

# Geometric morphometrics

## Landmark data preparation

Manuel F. G. Weinkauf

26–27 August 2022

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Setting up the R session</b>	<b>2</b>
<b>3</b>	<b>Reading the dataset</b>	<b>3</b>
<b>4</b>	<b>Procrustes superimposition</b>	<b>4</b>

## 1 Introduction

In this exercise sheet, you will learn how to prepare landmark data for further analyses. For this exercise, we will start using a dataset of landmarks from 2162 planktonic Foraminifera shells from a Red Sea sediment core (Fig. 1).

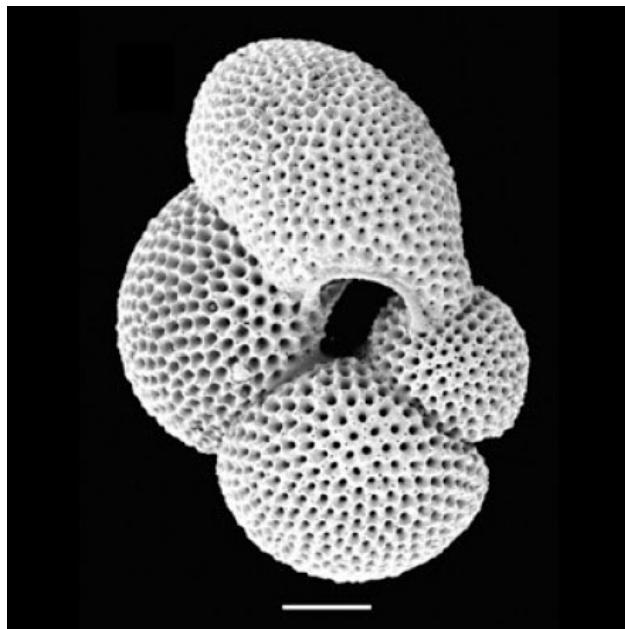


Figure 1: Depiction of a planktonic foraminifer of the species *Trilobatus sacculifer*. The scale-bar equals 0.1 mm. From <http://foraminifera.eu/>.

From all specimens, 12 landmarks were extracted as shown in (Fig. 2). The IDs of the specimens follow the

scheme ‘Top-Bottom\_nn’, with ‘Top’ as top of the sampling interval, ‘Bottom’ as bottom of the sampling interval, and ‘nn’ as running number within the sampling interval. For instance, ‘1432-1432.5\_2’ is specimen no. 2 from the sampling interval 1432–1432.5 cm within the sediment core.

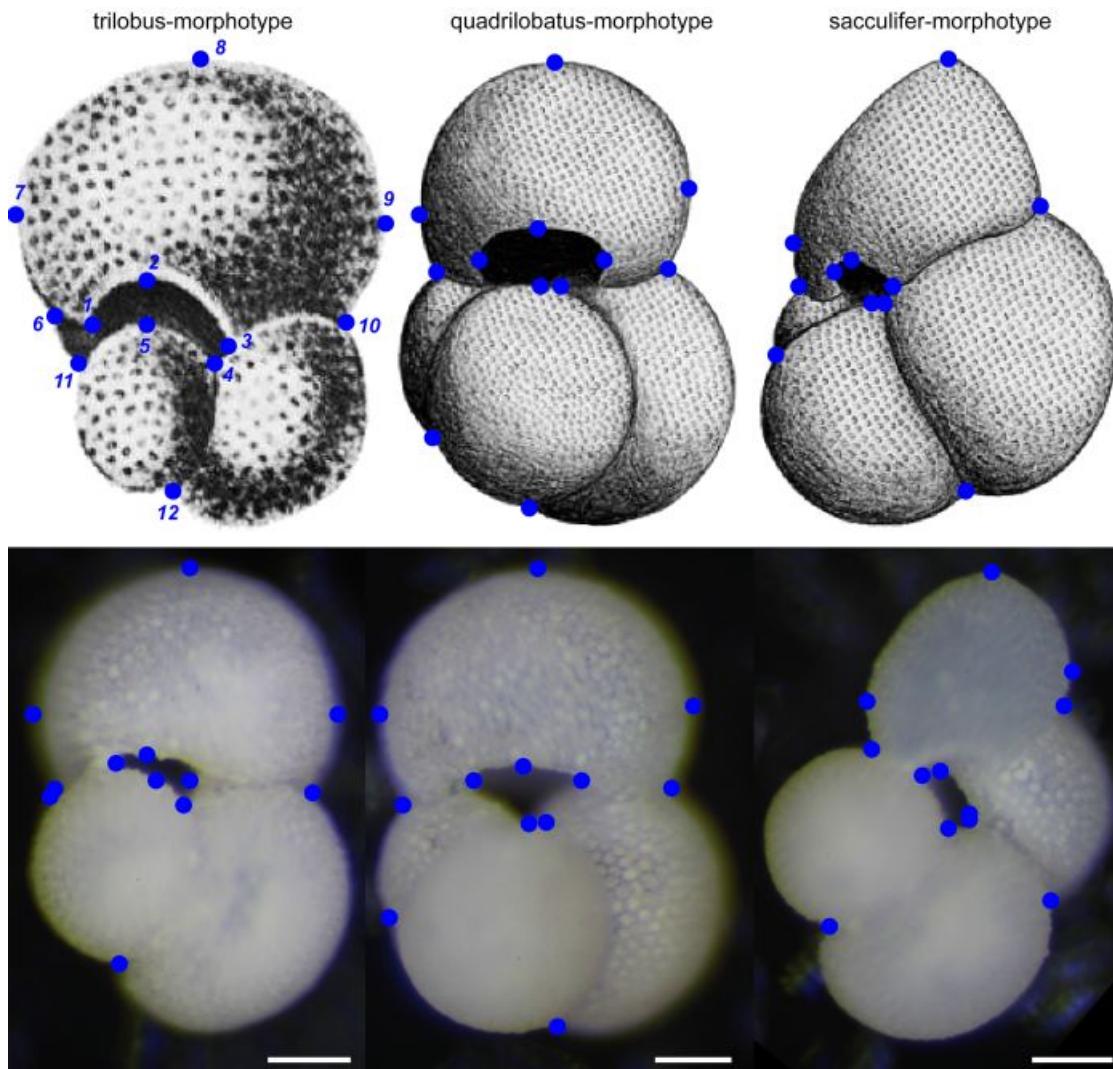


Figure 2: Scheme of the landmark extraction from *Trilobatus sacculifer* shells, exemplarily shown within the three morphotypes of the species. Scale bars equal 0.1 mm. From Weinkauf et al. (2019) An extinction event in planktonic Foraminifera preceded by stabilizing selection. *PLoS ONE* 14 (10): e0223490.

For more information on the data, you are referred to

Weinkauf, M. F. G., Bonitz, F. G. W., Martini, R., and Kucera, M. (2019) An extinction event in planktonic Foraminifera preceded by stabilizing selection. *PLoS ONE* 14 (10): e0223490. doi: [10.1371/journal.pone.0223490](https://doi.org/10.1371/journal.pone.0223490)

## 2 Setting up the R session

For landmark data preparation, we will mainly use the R-package ‘shapes’, although both ‘geomorph’ and ‘Morpho’ offer comparable functionality and you are free to try them as well.

Beside, we will use my own code included in the ‘MorphoFiles\_Function.r’ and ‘GeometricMorphomet-

rics\_Functions.r' source files.

```
setwd("C:/R_Data/Erlangen_Morphometrics/Session4_LandmarkDataPreparation")
library(shapes)
library(geomorph)
library(Morpho)
source("MorphoFiles_Function.r")
source("GeometricMorphometrics_Functions.r")
```

### 3 Reading the dataset

For reading data from a .tps-file, you can either use the ‘readland.tps()’-function from the package ‘geomorph’ or the ‘Read.TPS()’ function from ‘MorphoFiles\_Function.r’. Both work comparatively similar. However, ‘readland.tps()’ does not allow to remove NAs upon import and only imports the landmark data as an array. ‘Read.TPS()’ also imports the metadata (IDs, image names, scales) and allows to skip NAs; it imports the data in a structure best supported by the package ‘shapes’.

```
Sacculifer<-Read.TPS("SacculiferLandmarks.tps")
class(Sacculifer)

## [1] "list"
str(Sacculifer)

## List of 4
## $ ID      : chr [1:2230] "1432-1432.5_1" "1432-1432.5_2" "1432-1432.5_3" "1432-1432.5_4" ...
## $ Filenames: chr [1:2162] "1432-1432.5__2.ppm" "1432-1432.5__3.ppm" "1432-1432.5__4.ppm" "1432-1432...
## $ Scale    : num [1:2162] 0.277 0.277 0.277 0.277 0.277 ...
## $ LMDData  : num [1:12, 1:2, 1:2162] 60.4 90.6 133.4 125.2 91.5 ...
##   ..- attr(*, "dimnames")=List of 3
##     ...$ : NULL
##     ...$ : chr [1:2] "x" "y"
##     ...$ : chr [1:2162] "1432-1432.5_2" "1432-1432.5_3" "1432-1432.5_4" "1432-1432.5_5" ...
dim(Sacculifer$LMDData)

## [1] 12      2 2162
Sacculifer2<-readland.tps("SacculiferLandmarks.tps", specID="ID")

##
## No curves detected; all points appear to be fixed landmarks.
class(Sacculifer2)

## [1] "array"
dim(Sacculifer2)

## [1] 12      2 2230
```

The ‘class()’ and ‘str()’/‘dim()’ functions show us, that ‘Sacculifer’ is a list with the four elements ‘ID’ (vector), ‘Filenames’ (vector), ‘Scale’ (vector), and ‘LMDData’ (array). ‘Sacculifer2’ is an array, containing only the data available in ‘Sacculifer\$LMDData’. The length difference between ‘Sacculifer\$LMDData’ and ‘Sacculifer2’ is the result of ‘readland.tps()’ not removing NAs.

## 4 Procrustes superimposition

For landmark data, we simply use Procrustes superimposition to align all shapes and ensure that arbitrary shape differences are removed through rotation, translation, and scaling.

The ‘shapes’ package offers the function ‘procGPA’ to perform a generalized Procrustes superimposition of the landmark data. Should you ever be faced with organisms where a Pinocchio effect may be relevant, the ‘GRF()’ function in ‘GeometricMorphometrics\_Functions.r’ implements generalized resistant-fit Procrustes superimposition. Be warned that ‘GRF()’ takes very long to compute due to its iterative nature and may fail some shapes entirely. Setting AGoal to a higher values helps getting a result but sacrifices precision.

**EXERCISE 1:** Try out the different superimposition methods available in the packages and source codes and compare their results visually.

```
Sacculifer.GPA<-procGPA(Sacculifer$LMData)
```

```
## [1] "To speed up use option distances=FALSE"
## [1] "To speed up use option pcaoutput=FALSE"
Sacculifer.GRF<-GRF(Sacculifer$LMData, AGoal=0.01)
```

We can have a look at the success of the superimposition process by plotting the superimposed landmarks (Fig. 3).

```
Col.vec<-hcl.colors(n=dim(Sacculifer.GPA$rotated)[3], palette="viridis", alpha=0.5)

layout(matrix(1:4, 2, 2, byrow=TRUE))
par(mar=c(2.5, 2.5, 2.5, 2.5))
plot(Sacculifer$LMData[,1,1], Sacculifer$LMData[,2,1], xlim=c(0, 500), xlab="", ylim=c(0, 800), ylab="", asp=1, pch=16, col=Col.vec[1], main="Raw data")
for (i in 2:dim(Sacculifer$LMData)[3]){
  points(Sacculifer$LMData[,1,i], Sacculifer$LMData[,2,i], col=Col.vec[i])
}

plot(Sacculifer.GPA$rotated[,1,1], Sacculifer.GPA$rotated[,2,1], xlim=c(-150, 150), xlab="", ylim=c(-180, 160), ylab="", asp=1, pch=16, col=Col.vec[1], main="GPA data")
for (i in 2:dim(Sacculifer.GPA$rotated)[3]){
  points(Sacculifer.GPA$rotated[,1,i], Sacculifer.GPA$rotated[,2,i], col=Col.vec[i])
}

plot(Sacculifer.GRF$rotated[,1,1], Sacculifer.GRF$rotated[,2,1], xlim=c(-1.5, 1.5), xlab="", ylim=c(-1.6, 1.6), ylab="", asp=1, pch=16, col=Col.vec[1], main="GRF data")
for (i in 2:dim(Sacculifer.GRF$rotated)[3]){
  points(Sacculifer.GRF$rotated[,1,i], Sacculifer.GRF$rotated[,2,i], col=Col.vec[i])
}
```

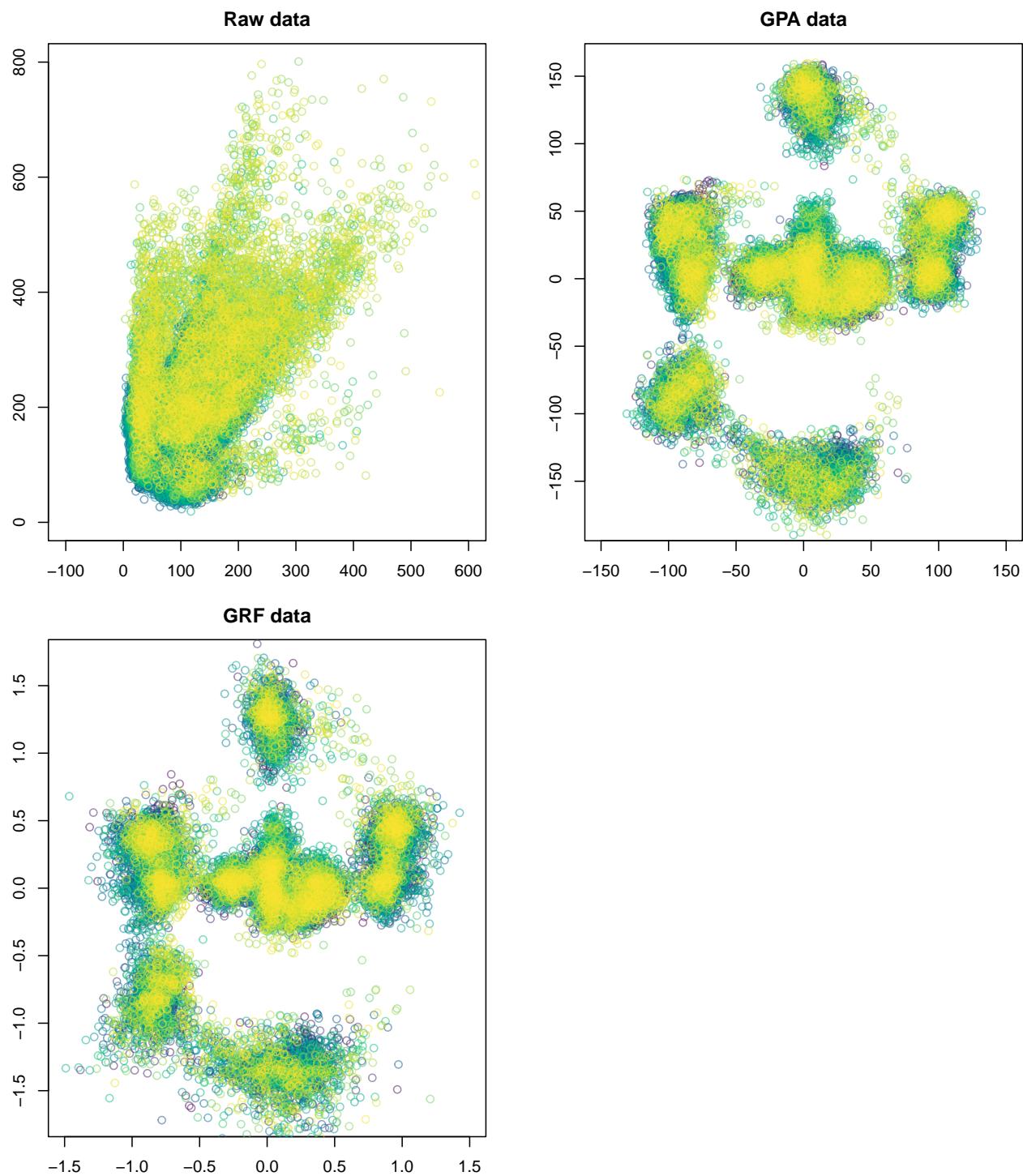


Figure 3: Comparison between raw landmark data and superimposition using generalized Procrustes superimposition and resistant-fit Procrustes superimposition.