

Archaeological Geometric Morphometrics and R

As part of the #StayHomeButStudy Workshop Series

Dr Christian Steven Hoggard (University of Southampton, United Kingdom)

Introduction

This guide provides a “hands-on” step-by-step introduction for the application of geometric morphometric (GMM) methodologies in archaeological science, as conducted in the R Environment. Using a published dataset this workflow will guide the reader through four key GMM procedures: 1) data importing (and creation), 2) data transformation, 3) data analysis and 4) data visualisation. A Japanese translation of this documentation is also provided.

I will first demonstrate the actions or functions on Zoom and then allow time for all participants to run the function. Should there be any queries then please let us know in the Slack workspace.

This practical constitutes the second workshop of the #StayHomeButStudy event, organised by Dr. Atsushi Noguchi, and is tailored for Japanese archaeologists, researchers and enthusiasts.

About the Code, Packages and Data

The data used throughout this guide originates from Ivanovaitė et al (2020): “*All these Fantastic Cultures? Research History and Regionalization in the Late Palaeolithic Tanged Point Cultures of Eastern Europe*”, published in open-access on the European Journal of Archaeology (<https://doi.org/10.1017/ea.2019.59>). The data can be found on my GitHub repository (<https://github.com/CSHoggard/-Eastern-Europe-Tanged-Points>), in addition to the Open Science Framework (<https://osf.io/agrwb/>).

All code, and data, including the markdown document (in HTML and PDF format) for this practical can be found on GitHub (https://github.com/CSHoggard/-japanworkshop2020tree/master/workshop_2).

The GMM procedure detailed below is grounded on two-dimensional outline analysis. In conducting outline analysis for this practical the following packages (including their imported packages) are necessary:

- * **Momocs** (Version 1.2.9) <https://cran.r-project.org/web/packages/Momocs/index.html>
- * **tidyverse** (Version 1.3.0) <https://cran.r-project.org/web/packages/tidyverse/index.html>

Software Installation

Following the installation of R or RStudio, we can now install the required packages:

```
if(!require("Momocs")) install.packages('Momocs', repos='http://cran.us.r-project.org')
if(!require("tidyverse")) install.packages('tidyverse', repos='http://cran.us.r-project.org')
if(!require("rio")) install.packages('rio', repos='http://cran.us.r-project.org')
```

To download the data into R/Rstudio we can use the `import` function from the `rio` package, and extract the data from the GitHub repository:

```
database <- rio::import("https://github.com/CSHoggard/-workshopjapan2020/raw/master/workshop_2/database")
tpsdata <- rio::import("https://github.com/CSHoggard/-workshopjapan2020/raw/master/workshop_2/tpslines")
```

The tidyverse and Momocs packages may take time to install given the size of the file, so please be patient. *Please ensure that these are downloaded prior the workshop.* Once installed, the packages can be activated through the `library()` function:

```
library(Momocs)
library(tidyverse)
```

About the Data

This data was composed to assess the robustness of cultural taxonomies in the Final Palaeolithic period of Eastern Europe, as portrayed through tanged point variants. It consists of 250 tanged point outlines and produced in the TPS Suite (<https://life.bio.sunysb.edu/morph/soft-dataacq.html>), using the *outline object* function. As this dataset was produced in the TPS Suite the file format is *.tps*. In their composition these outlines are semilandmarks, an algorithm-produced series of equidistant points are each shape. A database of all examples and their respective cultural assignment is also provided.

Importing Geometric Morphometric Data: Alternative Approaches

There are a number of ways which landmark and outline morphometric data can be imported into the R Environment. Here, for ease and replicability, the outline data (in *.tps* format) was stored on a GitHub repository and directed into the R environment, utilising the `Momocs::import_tps()` function in a *.rds* file. Other ways to import *.tps* data (if saved locally) include the `geomorph::readland.tps()` and rewriting tools in Momocs e.g. `Momocs::rw_rule()`. Data from stereomorph can also be imported through the `Momocs::import_StereoMorph_ldk()` and `Momocs::import_StereoMorph_curve()` functions.

Within Momocs, outlines can be extracted from silhouette data through the `Momocs::import_jpg()` and `Momocs::import_jpg1()` functions. See their respective helpfiles for more details. These will be demonstrated at the end of the workshop.

Examining the Data

With our data now in the R Environment we can now call our *tpsdata* through the `base::View` functions. The `base::View()` function will highlight the three constituent parts of the *tps* file: the 1) *Coo* (coordinate data), 2) *cur* (the curve data if present), and 3) *scale* (the scale data if present). It is the *Coo* data which we will take forward, with the database, to examine shape variation among our tanged points. It is best not to call the *tps* data as R will stream all coordinate data for each example.

We can also inspect our database using the `head()` function, and examine the different components of our dataset.

```
head(database)
```

```
## # A tibble: 6 x 10
##   ID Site Context Longitude Latitude Country Archaeological_~ File_Name
##   <dbl> <chr> <chr>      <dbl>      <dbl> <chr>      <chr>      <chr>
## 1     1 Balt~ Baltaš~    24.0      54.0 Lithua~ Baltic Magdalen~ Bal.1
## 2     2 Baro~ Barouka    30.3      53.5 Belarus Grensk      Bor.1
## 3     3 Baro~ Barouka    30.3      53.5 Belarus Grensk      Bor.3
## 4     4 Baro~ Barouka    30.3      53.5 Belarus Grensk      Bor.4
## 5     5 Baro~ Barouka    30.3      53.5 Belarus Grensk      Bor.5
## 6     6 Baro~ Barouka    30.3      53.5 Belarus Grensk      Bor.6
## # ... with 2 more variables: Reference <chr>, Notes <chr>
```

We can observe that the group data we want to examine (*Archaeological_Unit*) is `<chr>`, of type 'character', and not `<fctr>` ('factor'), as required for our analysis. This can be corrected through the `base::as_factor()` function:

```
database$Archaeological_Unit <- as.factor(database$Archaeological_Unit)

is.factor(database$Archaeological_Unit) # check to see the data is now of type 'character'
```

```
## [1] TRUE
```

We can also inspect the number of different archaeological units within our dataset through the `base::summary()` function. This highlights the number of tanged points in each group. With certain taxonomic units rarely used this is reflected in the low sample sizes for certain groups e.g. Vyshegorian.

```
summary(database$Archaeological_Unit)
```

##	Baltic Magdalenian	Bromme (Eastern Europe)	Bromme (Western Europe)
##		36	9
##			49
##	Grensk	Krasnosillya	Perstunian
##		55	29
##			4
##	Pitted Ware (Type A)	Podolian	Vyshegorian
##		24	14
##			8
##	Wolkushian		
##		22	

GMM Procedure 1: Outline File Creation

Central to Momocs are a specific suite of shape classes for: 1) *outlines* (`OutCoo`), *open outlines* (`OpnCoo`) and *landmarks* (`LdkCoo`). While some operations in Momocs are generic and do not depend on one of these classes, many functions require your data to be one of these S3 objects. In this instance our tps data is comprised of outlines, and so we wish for our data to be `OutCoo`, as to enable `efourier` (elliptic Fourier), `rfourier` (radii Fourier) or `tfourier` (tangent angle Fourier) analyses. For this workshop, we're only going consider elliptic Fourier analysis (EFA).

The coordinate data (`coo`) must therefore be turned into outline data through the `Momocs::Out()` function. Once performed, we can then enter the object (here called 'shape') and examine its properties.

```
shape <- Out(tpsdata$coo, fac = database) # incorporating our database as our factors
shape # call the object
```

```
## Out (outlines)
## - 250 outlines, 1543 +/- 1370 coords (in $coo)
## - 10 classifiers (in $fac):
## # A tibble: 250 x 10
##   ID Site Context Longitude Latitude Country Archaeological_~ File_Name
##   <dbl> <chr> <chr>      <dbl>      <dbl> <chr>      <fct>      <chr>
## 1     1 Balt~ Baltaš~    24.0      54.0 Lithua~ Baltic Magdalen~ Bal.1
## 2     2 Baro~ Barouka    30.3      53.5 Belarus Grensk          Bor.1
## 3     3 Baro~ Barouka    30.3      53.5 Belarus Grensk          Bor.3
## 4     4 Baro~ Barouka    30.3      53.5 Belarus Grensk          Bor.4
## 5     5 Baro~ Barouka    30.3      53.5 Belarus Grensk          Bor.5
## 6     6 Baro~ Barouka    30.3      53.5 Belarus Grensk          Bor.6
## # ... with 244 more rows, and 2 more variables: Reference <chr>, Notes <chr>
## - also: $ldk
```

This tells us that in our Out file there are a total of 250 outlines, with a mean number of 1543 landmarks and 10 different factors (longitude, Latitude, Archaeological_Unit, etc.).

GMM Procedure 2: Outline Visualisation

Now our data is in the R environment, and in the appropriate class required for Momocs, we can examine the outline shapes. We can first look at all outlines through the `Momocs::panel()` function. Factors can also be coloured in using the `fac` argument in `Momocs::panel()`.

An example using the `Momocs::panel()` function is seen in Figure 1.

```
panel(shape, main = "", fac = 'Archaeological_Unit')
```

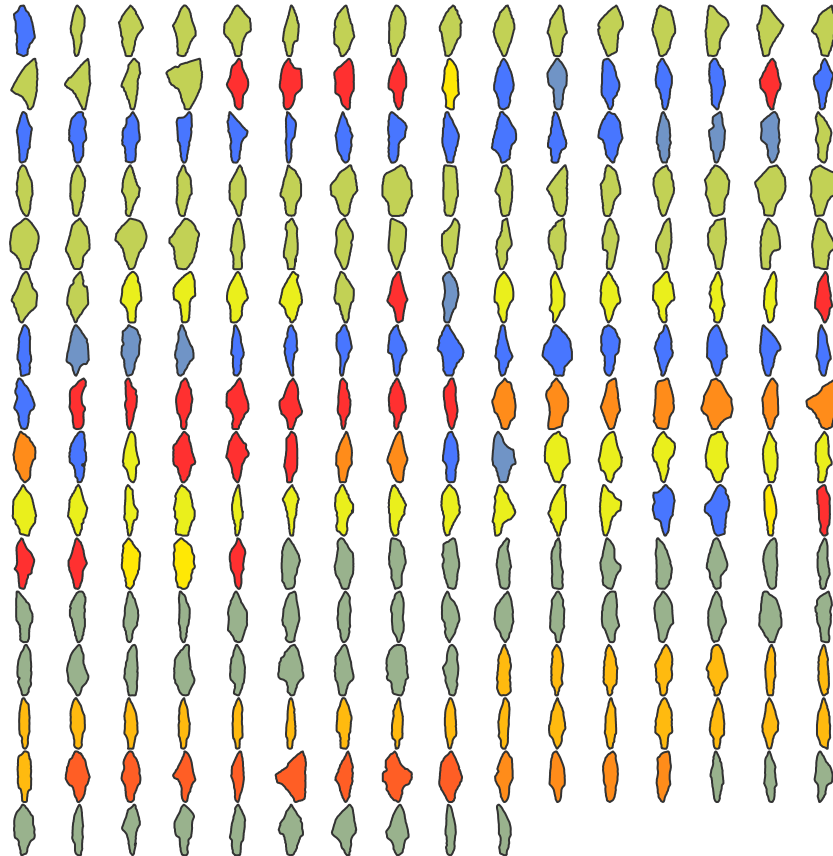


Figure 1: Showcasing the ‘panel’ function on all tanged points (colour correspond to different units)

An alternative to the `Momocs::panel()` function is `Momocs::mosaic()`, an updated image function (which will soon replace pane). This does include a legend, unlike the panel function, however the legend drawing options are limited, and are currently being improved for further package versions. We can also draw individual shapes of interest using the `Momocs::coo_plot()` function (Figure 2). A number of aesthetic or stylistic changes (including line colour and fill) are possible.

```
coo_plot(shape[1], col = "grey", main = "Artefact #1")
```

GMM Procedure 3: Outline Normalisation

Normalisation, as stressed by Claude (2008), has long been an issue in the elliptic Fourier process. Normalisation can be performed through the actual elliptic Fourier transformation (using what is known as the “first ellipse”). It is recommended to normalise (standardise) and align your shapes before the `Momocs::efourier()`

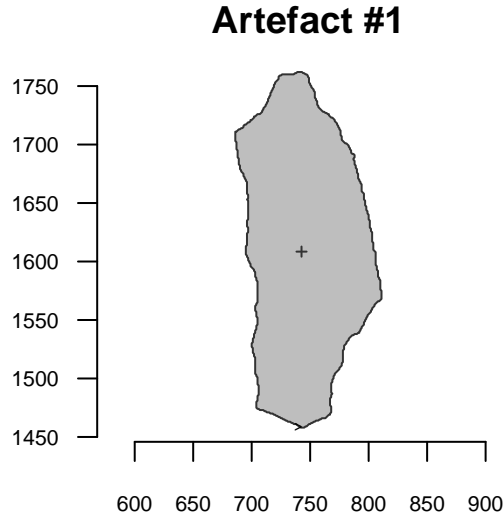


Figure 2: An individual tanged point (as through the `Momocs::coo_plot()` function)

process. Rotation was considered before outline digitisation, rotation could also be explored in Momocs through the `Momocs::coo_aligncalliper()` function. Here we will explore three transformation processes: 1) `Momocs::coo_center()`, 2) `Momocs::coo_scale()` and 3) `Momocs::coo_close()`.

These functions do the following:

- * `Momocs::coo_center()`: Centres coordinates on a common origin (common centroid).
- * `Momocs::coo_scale()`: Scaled the coordinates by their 'scale' (if provided) or centroid size.
- * `Momocs::coo_close()`: Closes unclosed shapes (precaution).

We can then use the `Momocs::stack()` function to inspect all outlines, now according to a common centroid and of a common scale (Figure 3):

```
shape <- coo_center(shape)
shape <- coo_scale(shape)
shape <- coo_close(shape)

stack(shape, main = "")
```

GMM Procedure 4: Elliptic Fourier Transformation

Elliptic Fourier Analysis (EFA) is one of a number of Fourier based methods of curve composition derived from the first series by Jean Baptiste Joseph Fourier (1768-1830), and developed by Giardina and Kuhl (1977) and Kuhl and Giardina (1982). In practice, a set of four parametric equations (grounded on sine and cosine transformations) are used to define the x and y Cartesian landmarks into curves (Fourier harmonic amplitudes). The coefficients (termed A,B,C and D), when summed together, represent the approximation of artefact form. This level of detail depends on the number of harmonics you use. The first harmonic (first ellipse) is responsible for rotation and defines an ellipse in the plane, with which all other harmonics fit onto. The greater the number of harmonics, the greater the level of detail, and the closer the curves resemble the shape. However, a considerable level of statistical noise is produced if there is too much detail, and so an appropriate level of harmonics are necessary.

When a level of harmonic power is determined, a series of procedures can be implemented to test how many harmonics are necessary:

- * `Momocs::calibrate_harmonicpower_efourier()`: This estimates the number of harmonics required for

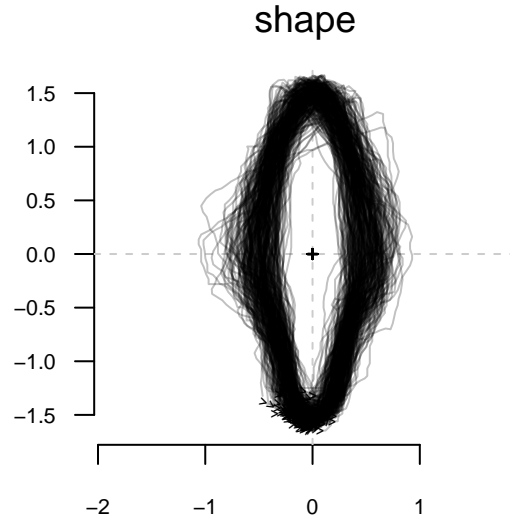


Figure 3: Normalised outlines of tanged points (n=250)

the elliptic Fourier process (and all other Fourier processes).

* `Momocs::calibrate_reconstructions_efourier()`: This procedure calculates reconstructed shapes for a series of harmonic numbers. This process best demonstrates the harmonic process.

* `Momocs::calibrate_deviations_efourier()`: Calculate deviations from the original and reconstructed shapes for a series of harmonic numbers.

```
calibrate_harmonicpower_efourier(shape, id = 4, nb.h = 20, plot = FALSE)
```

```
## $gg
## [1] NA
##
## $q
##           h1      h2      h3      h4      h5      h6      h7      h8
## 4) Barouka 6.27197 77.4212 80.2774 97.35366 97.45275 98.0372 98.43572 99.14469
##           h9      h10     h11     h12     h13     h14     h15
## 4) Barouka 99.25864 99.40648 99.55404 99.68398 99.72722 99.81056 99.89088
##           h16     h17     h18 h19
## 4) Barouka 99.92353 99.93879 99.96744 100
##
## $minh
##   90%   95%   99% 99.9%
##    5    5    9   17
```

This first procedure highlights how much shape (harmonic power) is represented by the individual harmonics. Typically, the process is performed on all shapes, however one is used here to detail the components obtained from the function.

```
calibrate_reconstructions_efourier(shape)
```

This second function (Figure 4) best exemplifies the harmonic concept: as the number of harmonics increase, so the approximation of shape is closer to the digitised artefact.

```
calibrate_deviations_efourier(shape)
```

The third and final function (Figure 5) provides another means of examining the role of harmonic power on

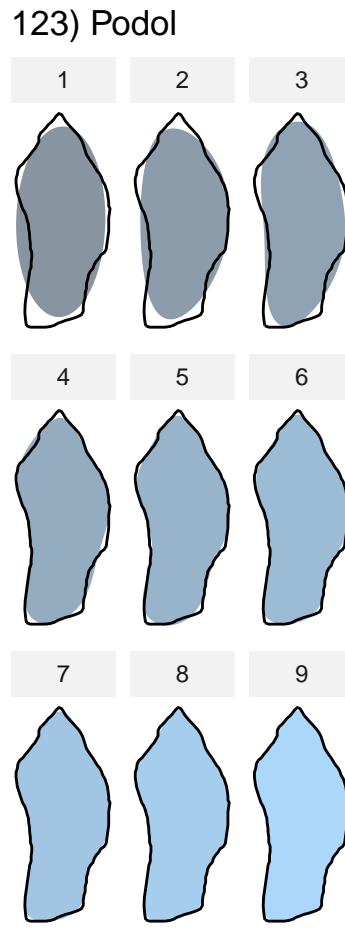


Figure 4: Harmonic estimation through the 'Momocs::calibrate_reconstructions_efourier()' function

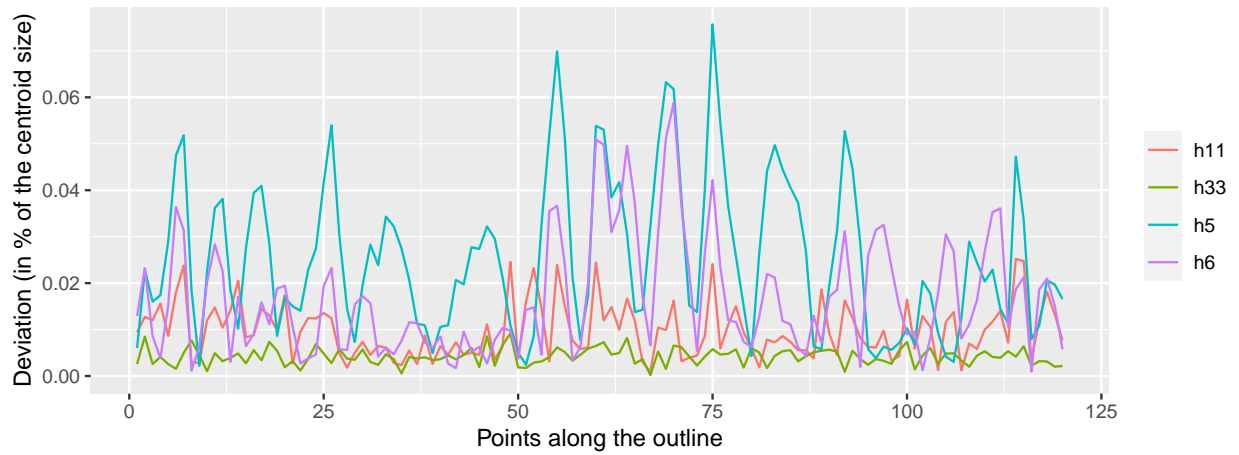


Figure 5: Harmonic estimation through the 'Momocs::calibrate_deviations_efourier()' function

deviation in shape./

Once we know how many harmonics are required we can use the `Momocs::efourier()` function to generate out `OutCoe` (outline coefficients) object.

```
efashape <- efourier(shape, nb.h = 11, smooth.it = 0, norm = TRUE)
```

GMM Procedure 5: Exploratory Procedure #1: PCA

With our elliptic fourier coefficients we can now begin the exploratory and analytical procedure. We will start by exploring the main theoretical differences in shape through a Principal Component Analysis (PCA). Please refer to the first workshop for a detailed explanation of PCA. We first need to convert out `OutCoe` class object to a PCA class object through the `Momocs::PCA()` function. We can then explore the main sources of shape variation through the `Momocs::PCcontrib()` function (Figure 6). The proportion can also be retrieved through calling the `Momocs::Scree()` function.

```
pcashape <- PCA(efashape)
PCcontrib(pcashape, nax = 1:5)
```

We can see through this function that Principal Component 1 (PC1), i.e. the main source of shape variation among the tanged points, range from thin tanged points to wider-tanged examples, and that Principal Component 2 (PC2), i.e. the second main source of shape variation, extends from left-exaggerated tangs to right-exaggerated tangs. This function can be set to display as many sources of shape variation as necessary./

While we can observe the main changes in artefact shape we are unsure how much variation these components account for. Using the `Momocs::Scree()` function we can find out that PC1 accounts for 57.7% of all shape variation, and that the first two axes account for 74.1% (almost three quarters of all shape variation within our dataset). 95% of all shape variation can be accounted for in the first ten principal components (an observation we will come back to afterwards)./

```
scree(pcashape)
```

```
## # A tibble: 44 x 3
##   axis proportion cumsum
##   <int>      <dbl> <dbl>
## 1     1      0.577  0.577
## 2     2      0.164  0.741
## 3     3      0.0806 0.821
## 4     4      0.0319 0.853
## 5     5      0.0307 0.884
## 6     6      0.0251 0.909
## 7     7      0.0195 0.929
## 8     8      0.0106 0.939
## 9     9      0.00918 0.948
## 10    10      0.00719 0.956
## # ... with 34 more rows
```

Now we know the main sources of shape variation, and the importance of each axis, we can now observe how each tanged point is reflected in the theoretical shape space through the `Momocs::plot_PCA()` function (Figure 7).

```
plot_PCA(pcashape, f = 'Archaeological_Unit', morphospace_position = "full_axes", zoom = 2, chull = FALSE)
```

References

For literature pertaining to outline analysis see:

* Claude, J. (2008). *Morphometrics with R*. Springer Publishing.

* Bonhomme, V., Picq, S., Gaucherel, C., & Claude, J. (2014). Momocs: Outline analysis using R. *Journal*

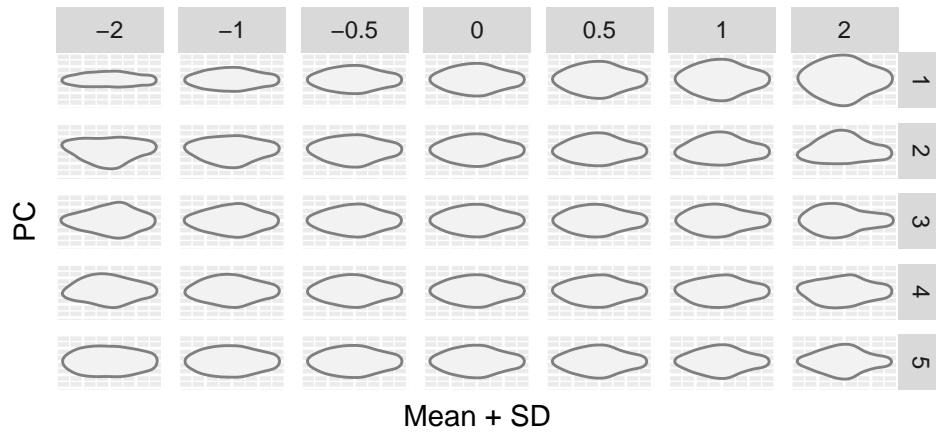


Figure 6: The first five principal components and their shape variation (measured in standard deviations)

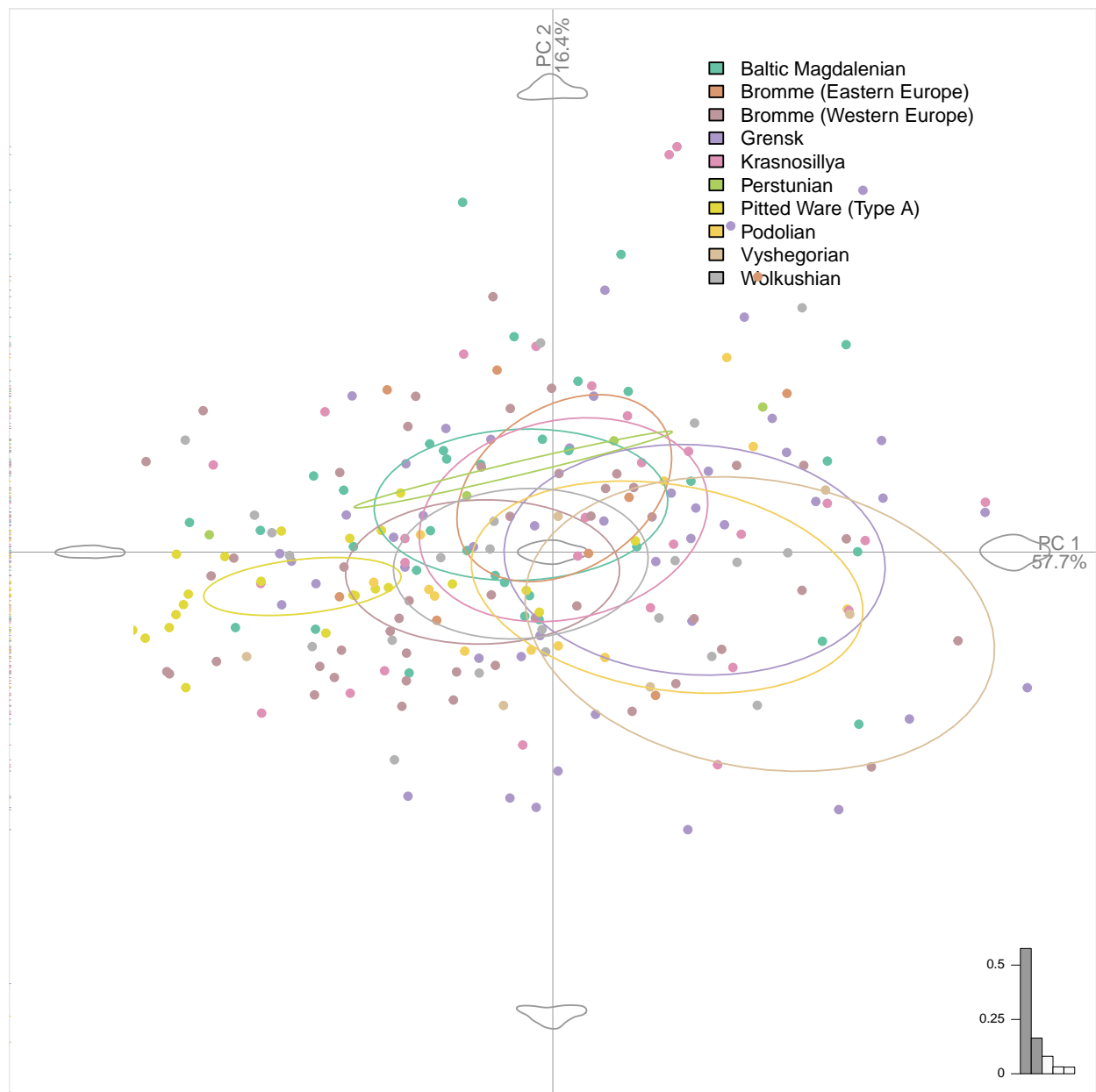


Figure 7: Principal Component Analysis (PCA 1 vs. PCA 2)

of *Statistical Software*, 56: 1–24.

* Caple, J., Byrd, J., & Stephan, C. N. (2017). Elliptical Fourier analysis: Fundamentals, applications, and value for forensic anthropology. *International Journal of Legal Medicine*, 131 (6): 1675–1690.

* Ferson, S., Rohlf, F. J., & Koehn, R. K. (1985). Measuring shape variation of two-dimensional outlines. *Systematic Zoology*, 34 (1): 59–68.

* Kuhl, F. P., & Giardina, C. R. (1982). Elliptic Fourier features of a closed contour. *Computer Graphics and Image Processing*, 18 (3): 236–258.

* Yoshioka, Y. (2004). Analysis of petal shape variation of *Primula sieboldii* by elliptic fourier descriptors and principal component analysis. *Annals of Botany*, 94 (5), 657–664.

* Zahn, C. T., & Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21 (3): 269–281.

For literature pertaining to GMM by the author (including code and data) see:

* Hoggard, C.S., Lauridsen, L. and Witte, K.B. (2019). The Potential of Geometric Morphometrics for Danish Archaeology: Two Case Studies. *Arkaeologisk Forum*, 40: 30-42. (<http://www.archaeology.dk/16738/Nr.%2040%20-%202019>). OSF: <https://osf.io/en5d2/>.

* Hoggard, C.S., McNabb, J. and Cole, J.N. (2019). The application of elliptic Fourier analysis in understanding biface shape and symmetry through the British Acheulean. *Journal of Paleolithic Archaeology*, 2 (2): 115-133. (<https://doi.org/10.1007/s41982-019-00024-6>). OSF: <https://osf.io/td92j/>.

* Ivanovaite, L., Swertka, K., Hoggard, C.S., Sauer, F. and Riede, F. (2020). All these fantastic cultures? Research history and regionalisation in the Late Palaeolithic tanged point cultures of Eastern Europe. *European Journal of Archaeology*. (<https://doi.org/10.1017/ea.2019.59>). OSF: <https://osf.io/agrwb/>.

* Vestergaard, C. and Hoggard, C.S. (2019). A Novel Geometric Morphometric (GMM) Application to the Study of Bronze Age Tutuli. *Danish Journal of Archaeology*, 8: 5-28. (<https://tidsskrift.dk/dja/article/view/112494/164318>). OSF: <https://osf.io/fcp43/>.

* Riede, F., Hoggard, C.S. and Shennan, S. (2019). Reconciling material cultures in archaeology with genetic data requires robust cultural evolutionary taxonomies. *Nature: Palgrave Communications*, 5 (1): 55. (<https://doi.org/10.1057/s41599-019-0260-7>). OSF: <https://osf.io/vtdf2/>.