

# Changes in facial shape throughout pregnancy - A computational exploratory approach

## Contents

<b>Housekeeping</b>	<b>1</b>
Load libraries . . . . .	1
Custom functions . . . . .	1
<b>SHAPE DATA</b>	<b>2</b>
Data prep . . . . .	2
Load data . . . . .	2
Generalized Procrustes analysis . . . . .	3
Principal component analysis . . . . .	4
Calculate Asymmetry . . . . .	6
Averageness . . . . .	7
Sexual dimorphism . . . . .	7
Data prep . . . . .	7
Calculating vector score . . . . .	9
Compile shape scores . . . . .	10
Descriptives . . . . .	10

## Housekeeping

### Load libraries

```
library(tidyverse)
library(geomorph)
```

Export sessionInfo()

```
writeLines(capture.output(sessionInfo()), "sessionInfo.txt")
```

### Custom functions

```
# Select PCs according to Broken Stick Model (MacArthur 1957)
# Takes output from plotTangentSpace or prcomp as argument
# Returns selected PCs and saves number of selected PCs in variable called "n.PCs.[argument name]"

# Adapted from function "evplot" (Francois Gillet, http://adn.biol.umontreal.ca/~numerica/ecology/numec
selectPCs <- function(PCA.output){

  ev <- PCA.output$sdev^2
  n.ev <- length(ev)
  bsm <- data.frame(j = seq(1:n.ev), p = 0)

  bsm$p[1] <- 1 / n.ev
```

```

for (i in 2:n.ev) bsm$p[i] <- bsm$p[i - 1] + (1/(n.ev + 1 - i))
bsm$p <- 100 * bsm$p / n.ev

test <- cbind(100 * ev / sum(ev), bsm$p[n.ev:1])
n.PCs <- sum(test[, 1] >= test[, 2])

arg_name <- deparse(substitute(PCA.output)) # Get argument name
var_name <- paste("n.PCs", arg_name, sep = ".")
assign(var_name, n.PCs, .GlobalEnv)

if (!is.null(PCA.output$pc.summary$importance)) {

  return(PCA.output$pc.summary$importance[, 1:n.PCs])

} else {

  temp <- summary(PCA.output)
  return(temp$importance[, 1:n.PCs])

}
}

```

## SHAPE DATA

For more details see:

- Holzleitner, I. J., Lee, A. J., Hahn, A. C., Kandrik, M., Bovet, J., Renoult, J. P., Simmons, D., Garrod, O., DeBruine, L. M., & Jones, B. C. (2019). Comparing theory-driven and data-driven attractiveness models using images of real women's faces. *Journal of Experimental Psychology. Human Perception and Performance*, 45(12), 1589–1595. <https://doi.org/10.1037/xhp0000685>
- Marcinkowska, U. M., & Holzleitner, I. J. (2020). Stability of women's facial shape throughout the menstrual cycle. *Proceedings of the Royal Society B: Biological Sciences*, 287(1924), 20192910. <https://doi.org/10.1098/rspb.2019.2910>

## Data prep

### Load data

Read shape data

```
data.temp <- readland.tps("dataFiles/_DATA.tps", specID = "ID", warnmsg = FALSE)
```

Check number of images for each woman

```

check <- tibble(picture = dimnames(data.temp)[[3]]) %>%
  mutate(id = gsub("-", "_", picture)) %>%
  separate(id, c("id", "image"), sep="_") %>%
  group_by(id) %>%
  add_tally()

```

Data dimensions

```

sample <- check %>%
  filter(n==3)

```

```

data <- data.temp[, , dimnames(data.temp)[[3]] %in% sample$picture]
ids <- dimnames(data)[[3]]

# Defining dimensions of data
k <- dim(data)[1] # number of landmarks
d <- dim(data)[2] # number of dimensions
n <- dim(data)[3] # number of images (3 sessions per woman)

```

## Generalized Procrustes analysis

```

gpa.data <- gpagen(data, print.progress = FALSE)

# geomorph flips templates: rotate
rotate <- gpa.data$coords
temp <- array(0, c(k, d, n))
data.aligned <- array(0, c(k, d, n))

for (i in 1:n){
  temp[, , i] <- rotate[, , i] %*% matrix(c(-1, 0, 0, 1), 2, 2, byrow = T)
  data.aligned[, , i] <- temp[, c(2,1), i]
}

dimnames(data.aligned)[[3]] <- ids

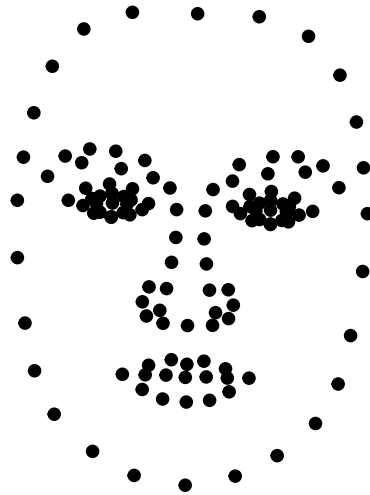
```

Visualize mean

```

mean <- gpa.data$consensus %*% matrix(c(-1, 0, 0, 1), 2, 2, byrow = T)
template <- mean[, c(2,1)]
plot(template, asp = 1, pch = 19, cex = .8, axes = FALSE, ylab = "", xlab = "")

```

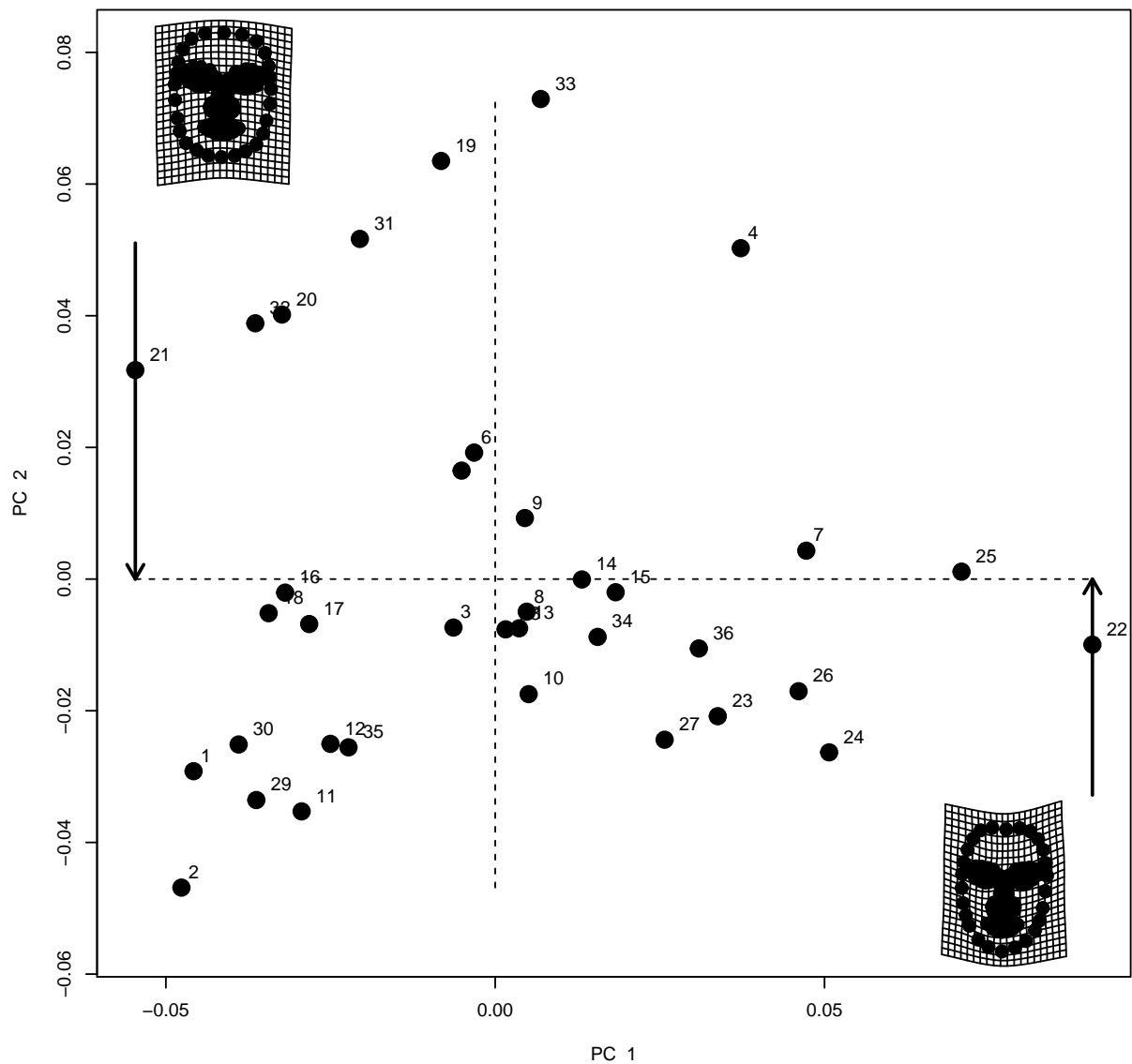


```
png(filename="template_lm124.png",res = 300, height = 1800, width = 1800)
par(mar = c(2, 0, 2, 0) + 0.1) # bottom, left, top, right
plot(template, asp = 1, pch = 19, cex = .8, axes = FALSE, ylab = "", xlab = "")
dev.off()
```

```
## pdf
## 2
```

## Principal component analysis

```
PCA <- plotTangentSpace(data.aligned, verbose = T, label = T)
```



```
####PCA <- gm.prcomp(data.aligned)
```

```
selectPCs(PCA)
```

```
##
## Standard deviation      0.03517135 0.02909763 0.0234919 0.01856749
## Proportion of Variance 0.27715000 0.18969000 0.1236400 0.07724000
## Cumulative Proportion  0.27715000 0.46684000 0.5904800 0.66772000
```

```
#Hence retaining 4 principal components, explaining 67% of variance
```

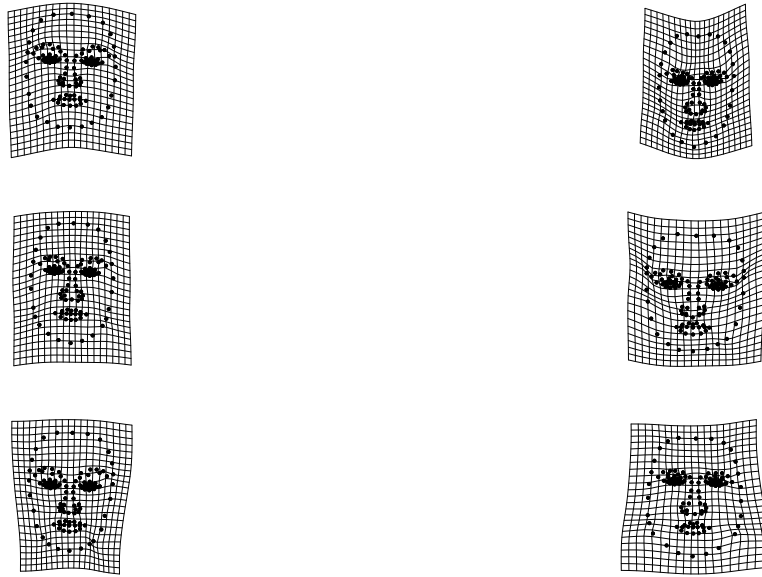


Figure shows PCs 1-3 in the current sample. Rows = PCs, columns = sample min/max.

```
scores <- PCA$pc.scores

col_names <- dimnames(scores)[[1]]
col_names <- gsub("_nc_1", "", col_names)
row_names <- dimnames(scores)[[2]]
PC.scores <- matrix(scores,
                    ncol = n,
                    byrow = TRUE,
                    dimnames = list(row_names, col_names)) %>%
  t() %>%
  as.data.frame() %>%
  rownames_to_column(var = "oc_id") %>%
  dplyr::select(1:(n.PCs.PCA + 1))
```

Save PC scores for subsequent analysis

## Calculate Asymmetry

Create empty array for next step

```
original <- data.aligned
mirrored <- array(0, c(k, d, n))
```

Mirror landmark template by changing sign of x-coords

```
for (i in 1:n){
  mirrored[,i] <- original[,i] %*% matrix(c(-1, 0, 0, 1), 2, 2, byrow = T)}
```

Relabel landmarks

```
mirr.lm.order <- as.matrix(read.table("./dataFiles/mirr.lm.order.txt", header = F))
mirrored <- mirrored[order(mirr.lm.order),,]
```

Put original & mirrored data in one array

```
org.mirr <- array(0, c(k, d, n * 2))
org.mirr[, , 1:n] <- original[, , 1:n]
org.mirr[, , (n + 1):(n * 2)] <- mirrored[, , 1:n]
```

Run GPA

```
gpa.asymmetry <- gpagen(org.mirr, print.progress = FALSE)
```

Separate into two arrays

```
data.org.aligned <- gpa.asymmetry$coords[, , 1:n]
data.mirr.aligned <- gpa.asymmetry$coords[, , (n + 1):(n * 2)]
```

Calculate Procrustes distance between original and mirrored faces

```
asym <- numeric(n)
for (i in 1:n) {
  asym[i] <- sqrt(sum((data.org.aligned[, , i] - data.mirr.aligned[, , i])**2))
}
```

## Averageness

Calculated as Euclidean distance between PCs of individual faces and average face

```
pc.matrix <- PC.scores[, 2:ncol(PC.scores)] %>%
  as.matrix()
sample.mean <- t(as.matrix(colMeans(pc.matrix)))
avg <- numeric(n)
for (i in 1:n) {
  avg[i] <- sqrt(sum((pc.matrix[i, ] - sample.mean)**2))
}
```

Reverse averageness scores so higher value means more average

```
avg_r <- (avg - 1) * -1
```

```
asym_avg_scores <- tibble(
  "ID" = ids,
  "asymmetry" = asym,
  "averageness" = avg_r
) %>%
  separate(ID, c("ID", "image"), sep = "_|-")

write_csv(asym_avg_scores, "./dataFiles/asym_avg_scores.csv")
```

## Sexual dimorphism

Please note that to calculate the sexual dimorphism vector allowing to determine sexual dimorphism scores for individual women, we have used an external dataset (DATAFILE\_London.tps). The dataset and more details can be found on <https://osf.io/98qf4/>.

### Data prep

```
# Please note that we adjusted original dataset to the current analysis code by editing IMAGE ids to in

data.sexdim_temp <- readland.tps("dataFiles/DATAFILE_London_Ed.tps", specID = "imageID", warnmsg = FALSE)
```

```
# in the current data set, only LMs 1 to 124 are used; subset sexdim data set
data.sexdim <- data.sexdim_temp[1:124,,]
```

```
ids.sexdim <- dimnames(data.sexdim)[[3]]
```

```
n.sd <- dim(data.sexdim)[3]
```

```
n.all <- n + n.sd
```

GPA of faces including sexual dimorphism prototype faces

```
data.all <- array(c(data, data.sexdim),
  dim = c(k, d, n.all),
  dimnames = list(NULL, NULL, c(ids, ids.sexdim)))
```

```
gpa.sexdim <- gpagen(data.all, print.progress = FALSE)
```

```
# rotate data
```

```
rotate <- gpa.sexdim$coords
```

```
temp <- array(0, c(k, d, n.all))
```

```
sexdim.aligned <- array(0, c(k, d, n.all))
```

```
for (i in 1:n.all){
  temp[,i] <- rotate[,i] %*% matrix(c(-1, 0, 0, 1), 2, 2, byrow = T)
  sexdim.aligned[,i] <- temp[, c(2,1), i]
}
```

```
dimnames(sexdim.aligned)[[3]]<-c(ids, ids.sexdim)
```

Convert array to matrix for PCA using prcomp

```
# Data needs to be in format id, x1, y1, x2, y2, ...x132, y132
```

```
temp <- matrix(sexdim.aligned,
  ncol = (k*2),
  byrow = T)
```

```
x.coords <- temp[,1:k]
```

```
y.coords <- temp[, (k+1):(k*2)]
```

```
sexdim.matrix <- matrix(rbind(x.coords, y.coords),
  n.all,
  dimnames = list(dimnames(sexdim.aligned)[[3]], NULL))
```

```
sexdim.matrix.prototype <- sexdim.matrix[(n+1):n.all,]
```

```
sexdim.matrix.faces <- sexdim.matrix[1:n,]
```

PCA only of those faces that will constitute sexual dimorphism prototypes

```
PCA.sexdim <- prcomp(sexdim.matrix.prototype)
```

```
selectPCs(PCA.sexdim)
```

```
# 8 PCs explaining 74% of variance
```

Extract scores of faces that will constitute male and female prototypes

```
PC.scores.prototype <- PCA.sexdim$x[,1:n.PCs.PCA.sexdim] %>%
  as.data.frame() %>%
  cbind(ids.sexdim) %>%
  separate(ids.sexdim, c("id", "sex"), sep = 4) %>%
  mutate(sex = recode(sex, "f" = "0", "m" = "1"))
```



Predict PC scores of current sample based on sexdim PCA model

```
PC.scores.data <- predict(PCA.sexdim, newdata=sexdim.matrix.faces) %>%  
  as.data.frame() %>%  
  dplyr::select(1:n.PCs.PCA.sexdim)
```

### Calculating vector score

Pull out faces that define female and male prototypes, respectively

```
fem <- PC.scores.prototype %>%  
  filter(sex==0) %>%  
  dplyr::select(-sex, -id) %>%  
  as.matrix()  
  
mal <- PC.scores.prototype %>%  
  filter(sex==1) %>%  
  dplyr::select(-sex, - id) %>%  
  as.matrix()
```

Calculate male and female average for each PC (=column)

```
fem.mean <- t(as.matrix(colMeans(fem)))  
mal.mean <- t(as.matrix(colMeans(mal)))
```

Normalize difference vector between female and male

*#Please note that other than in Holzleitner et al. (2019), scores were scaled so that 0 would correspond to the average of the two prototypes*

```
temp <- fem.mean - mal.mean  
norm.dist <- sqrt(sum(temp ^ 2))  
norm.vec <- temp / norm.dist
```

For each face, subtract average fem PC score, divide by norm.distance, multiply result by norm.vector, sum elements

```
R <- nrow(PC.scores.data)  
pc.matrix <- as.matrix(PC.scores.data)  
sd.vector <- numeric(R)  
for (i in 1:R) {  
  z1 <- pc.matrix[i, ] - mal.mean  
  z2 <- z1 / norm.dist  
  z3 <- z2 * norm.vec  
  sd.vector[i] <- sum(z3)  
}
```

Save scores

```
sexDim_scores <- tibble(  
  "ID" = ids,  
  "sexdim" = sd.vector  
) %>%  
  separate(ID, c("ID", "image"), sep = "_|-")  
  
write_csv(sexDim_scores, "./dataFiles/sexDim_scores.csv")  
  
sexDim_scores <- read_csv("./dataFiles/sexDim_scores.csv")  
asym_avg_scores <- read_csv("./dataFiles/asym_avg_scores.csv")
```

## Compile shape scores

```
shapeScores_wide <- asym_avg_scores %>%
  left_join(sexDim_scores, by=c("ID", "image"))

shapeScores_long <- shapeScores_wide %>%
  gather(key = "type",
         value = "value",
         asymmetry, averageness, sexdim) %>%
  # Z-score shape scores
  group_by(type) %>%
  mutate(value.z = scale(value)) %>%
  ungroup()

write_csv(shapeScores_long, "./dataFiles/shapeScores_long.csv")
write_csv(shapeScores_wide, "./dataFiles/shapeScores_wide.csv")
```

## Descriptives

```
shapeScores_wide %>%
  dplyr::select(-image) %>%
  group_by(ID) %>%
  summarise_all(list(
    ~mean(.)
  )) %>%
  ungroup() %>%
  dplyr::select(-ID) %>%
  summarise_all(list(
    M = ~mean(.),
    SD = ~sd(.),
    min = ~min(.),
    max = ~max(.))) %>%
  gather(stat, val) %>%
  mutate(val = round(val, 4)) %>%
  separate(stat, into = c("Score", "stat"), sep = "_") %>%
  spread(stat, val) %>%
  dplyr::select(Score, M, SD, min, max) %>%
  knitr::kable()
```

Score	M	SD	min	max
asymmetry	0.0480	0.0089	0.0373	0.0646
averageness	0.9486	0.0111	0.9314	0.9656
sexdim	1.0304	0.4194	0.3446	1.6208