Results

Nastaran Ghorbani

2024-02-07

teeth-scriptus-pricei2(aka shape)

hotelling test

```
load("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei2/data/teethdata_scriptus_pricei.RData")


pvals_hotelling = list()
for (i in c("LM1", "LM2", "LM3", "UM1", "UM2", "UM3")){



  PC_combined <- read.csv(paste0("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei2/data/matlab/

  library(Hotelling)
  PC_combined$g <- c(rep(1,length(data[[i]][["scriptus"]])),
                     rep(2,length(data[[i]][["pricei"]])))

  results <-  hotelling.test(.~g, data = PC_combined)
  results$stats$statistic


  #Permutation
  nsim <- 1000
  null <- c()
  for (j in 1:nsim){
    temp <- PC_combined
    temp$g <- sample(temp$g, length(temp$g),replace = FALSE)
    null[j] <-  hotelling.test(.~g, data = temp)$stat$statistic
  }


  pvals_hotelling[[i]] <- mean(null >= results$stats$statistic)
}
```

```
## Loading required package: corpcor
```

```
pvals_hotelling
```

```
## $LM1
## [1] 0
##
## $LM2
## [1] 0
##
## $LM3
## [1] 0
##
## $UM1
## [1] 0
##
## $UM2
## [1] 0
##
## $UM3
## [1] 0
```

dist-based-test

```
load("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei2/data/teethdata_scriptus_pricei.RData")
pvals <- list()
for (toothtype in c("LM1","LM2","LM3","UM1","UM2","UM3")){print(toothtype)

  n_scriptus <- length(data[[toothtype]][["scriptus"]])
  n_pricei <- length(data[[toothtype]][["pricei"]])

  class <- c(rep("scriptus",n_scriptus),rep("pricei",n_pricei))


  #Run this script first in matlab: pairwise_dist_scriptus_pricei.m
  #Pariwise distances
  #First rows are scriptus and last rows are pricei
  ddd <- read.csv(paste0("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei2/data/matlab/pairwise_
  ddd <- as.matrix(ddd)

  #Distrance based permutation testing.
  #Based on the test defined in Soto et al 2021
  #1. Use distances based on the shapes projected into the tangent space.
  #to 2. Use distances in the size-shape space.  (I just need a function that computes distance between


  Dbar11 <- sum(ddd[class == "scriptus",class == "scriptus"])/(n_scriptus^2)
  Dbar22 <- sum(ddd[class == "pricei",class == "pricei"])/(n_pricei^2)
  Dbar12 <- sum(ddd[class == "pricei",class == "scriptus"])/(n_pricei*n_pricei)

  S <- ((n_scriptus*n_pricei)/((n_scriptus+n_pricei)))*(2*Dbar12 - (Dbar11 + Dbar22))

  #Now permute
  Sperm <- c()
  nsim <- 100000
  for (i in 1:nsim){
```

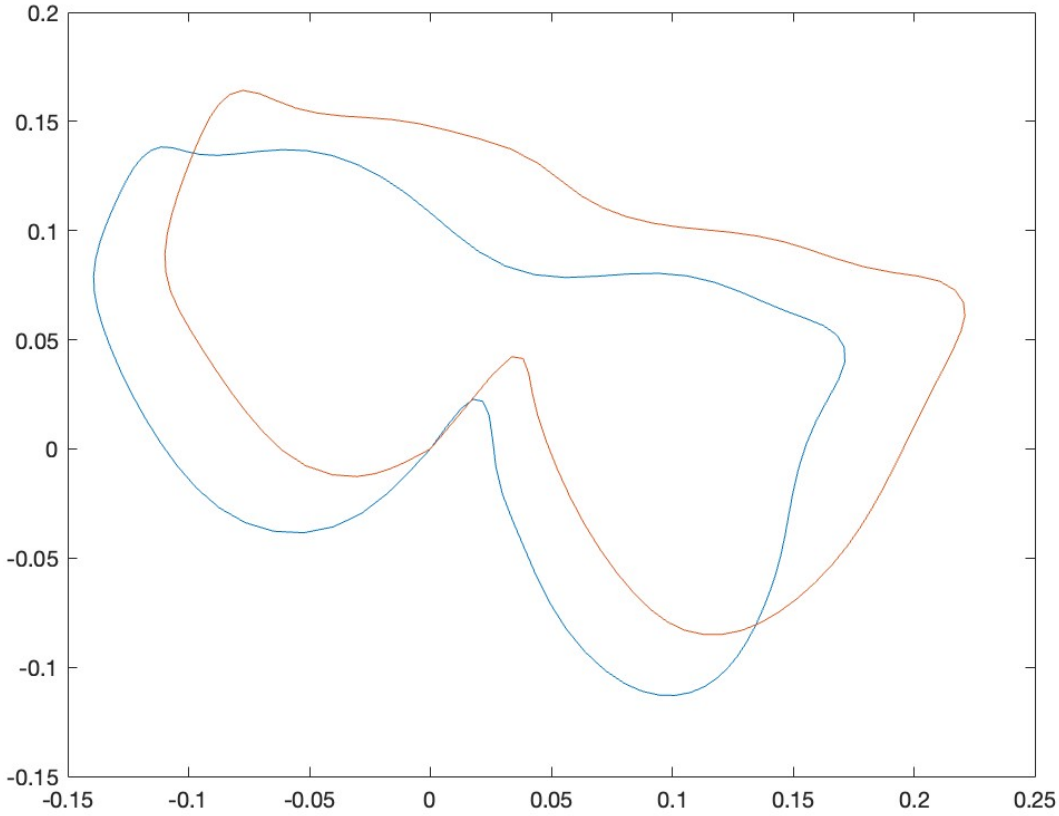
```
    class_perm <- sample(class,length(class),replace = FALSE)
    Dbar11 <- sum(ddd[class_perm == "scriptus",class_perm == "scriptus"])/(n_scriptus^2)
    Dbar22 <- sum(ddd[class_perm == "pricei",class_perm == "pricei"])/(n_pricei^2)
    Dbar12 <- sum(ddd[class_perm == "pricei",class_perm == "scriptus"])/(n_pricei*n_pricei)

    Sperm[i] <- ((n_scriptus*n_pricei)/((n_scriptus+n_pricei)))*(2*Dbar12 - (Dbar11 + Dbar22))
  }

  pvals[[toothtype]] <- mean(Sperm >= S)

}
```

```
## [1] "LM1"
## [1] "LM2"
## [1] "LM3"
## [1] "UM1"
## [1] "UM2"
## [1] "UM3"
```

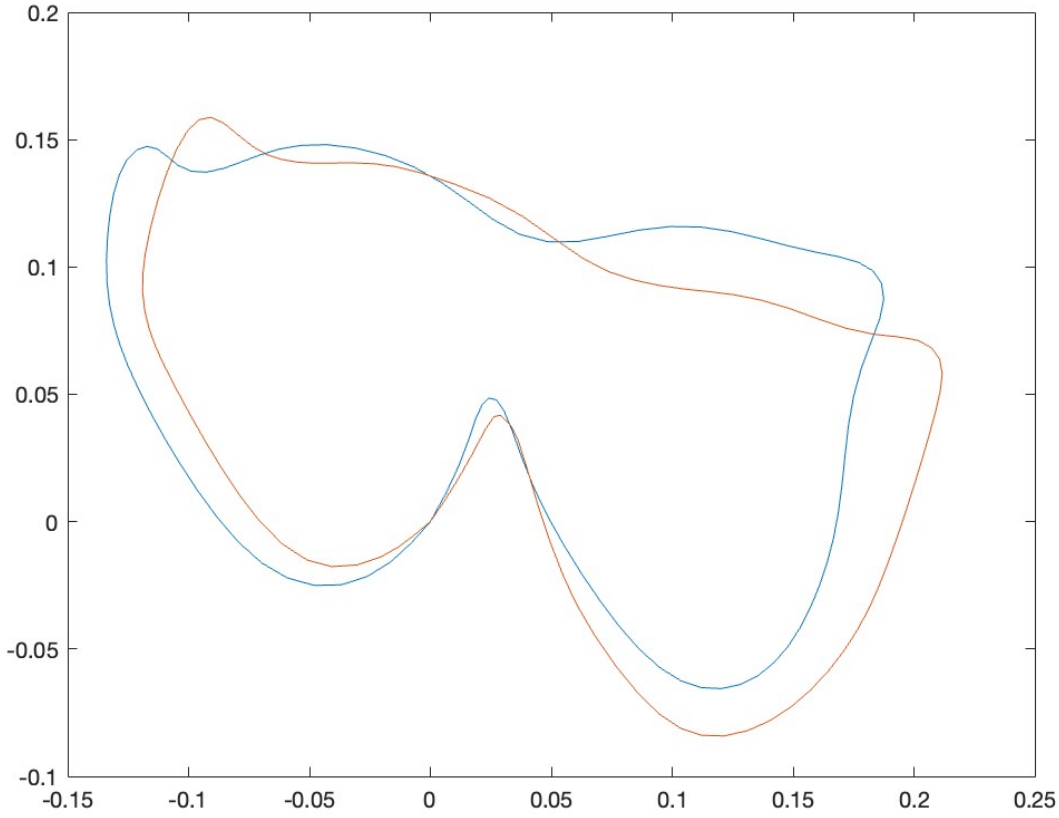
```
pvals
```

```
## $LM1
## [1] 0
##
## $LM2
## [1] 0
##
## $LM3
## [1] 9e-05
##
## $UM1
## [1] 0
##
## $UM2
## [1] 0
##
## $UM3
## [1] 0
```

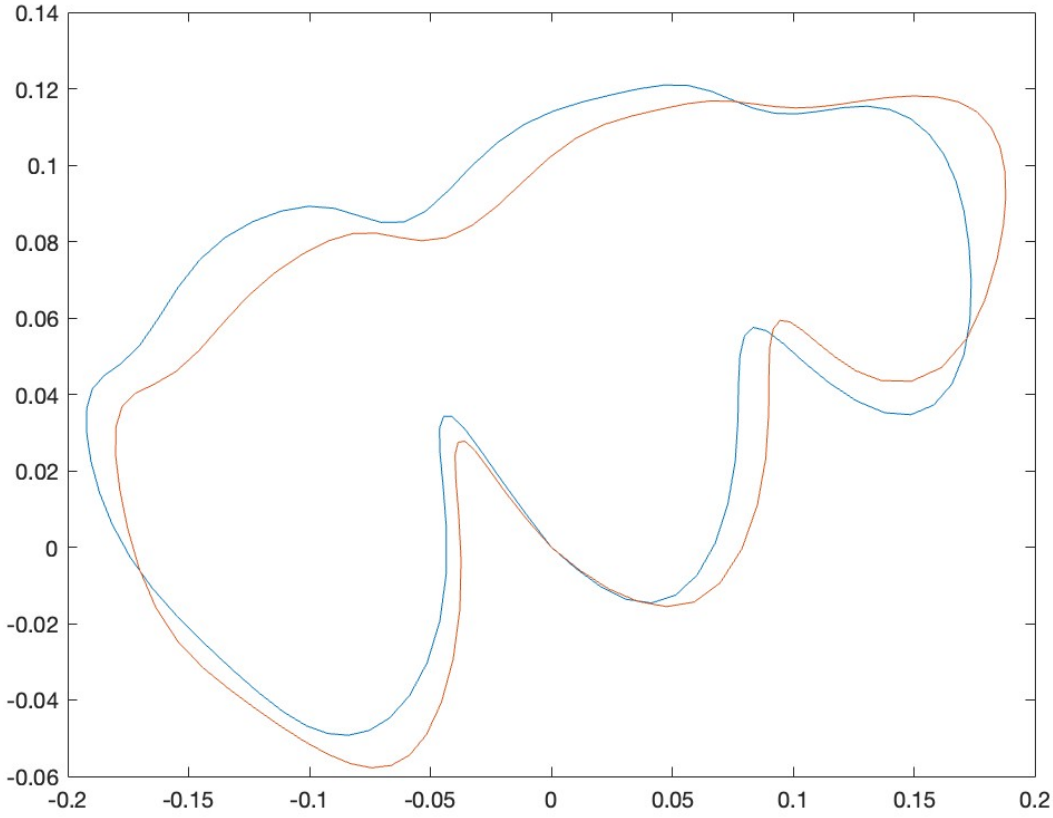
plots



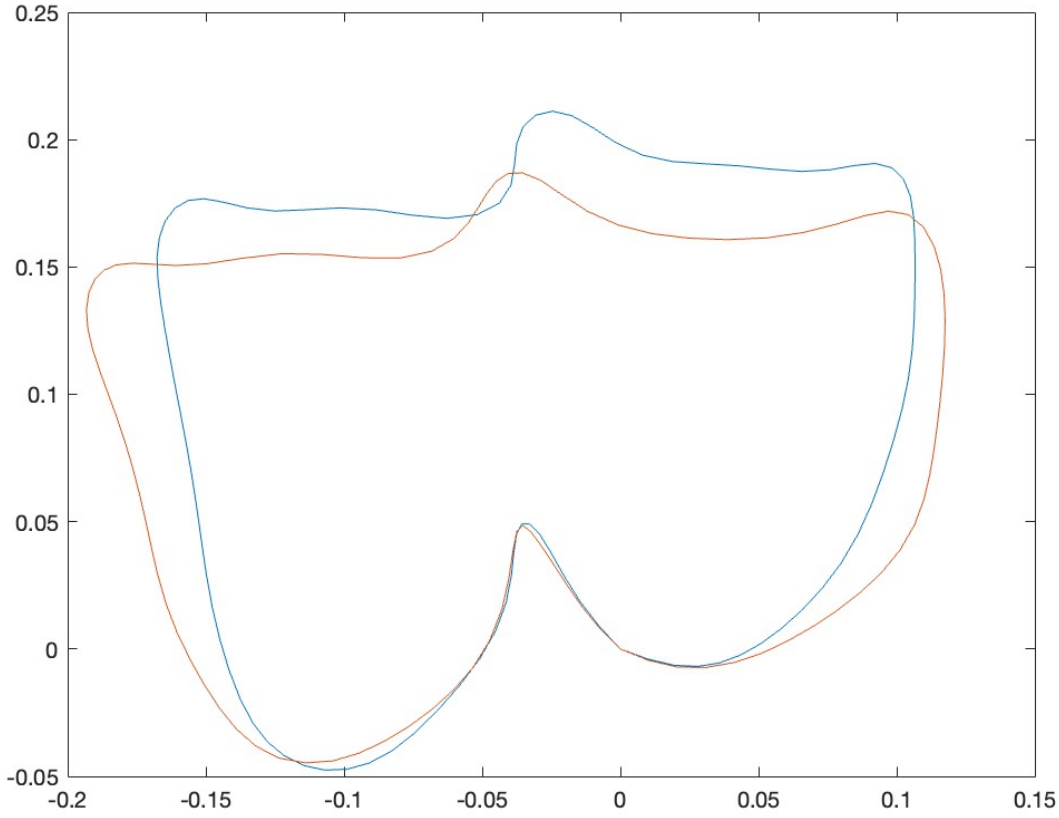

```
## LM1
```



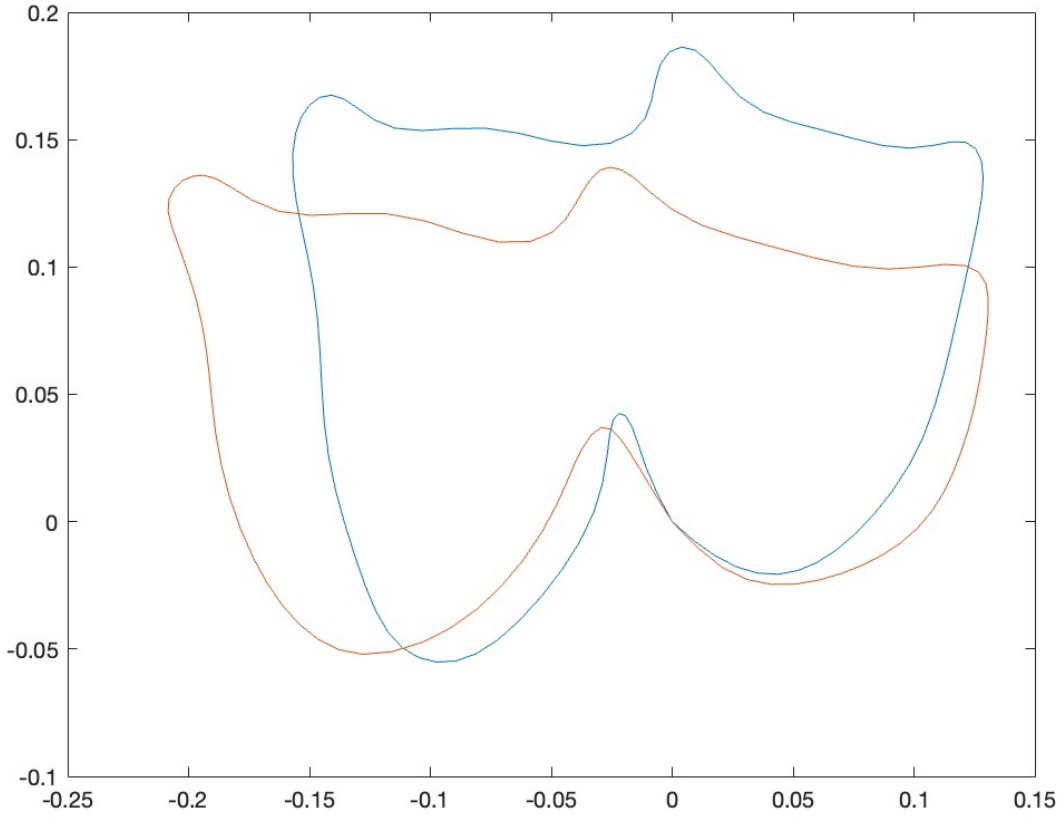
LM2


0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2

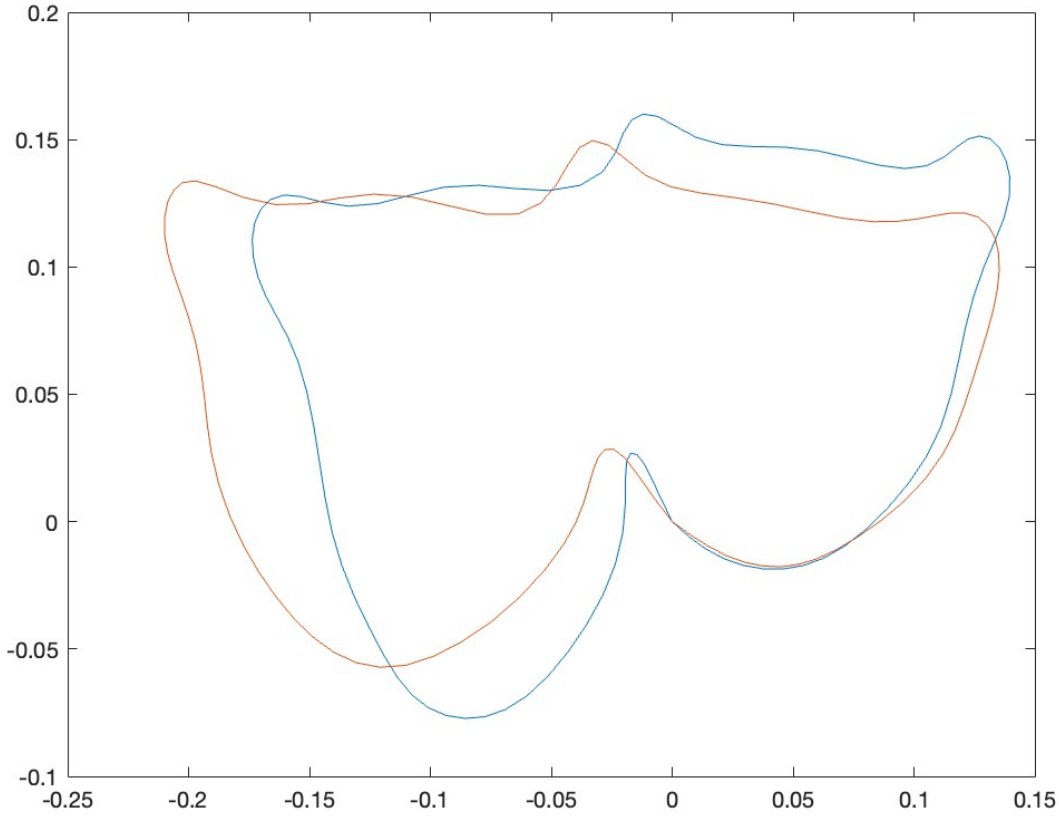
LM3




```
## UM1
```



UM2




```
## UM3
```

teeth-scriptus-pricei-size(aka shape & size)

hotelling test

```
library(Hotelling)

load("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei-size/data/teethdata_scriptus_pricei-size.

pvals_hotelling = list()
for (i in c("LM1", "LM2", "LM3", "UM1", "UM2", "UM3")){



PC_combined <- read.csv(paste0("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei-size/data/matla

PC_combined$g <- c(rep(1,length(data[[i]][["scriptus"]])),
```

```
                            rep(2,length(data[[i]][["pricei"]])))

results <-  hotelling.test(.~g, data = PC_combined)
results$stats$statistic


#Permutation
nsim <- 1000
null <- c()
for (j in 1:nsim){
temp <- PC_combined
temp$g <- sample(temp$g, length(temp$g),replace = FALSE)
null[j] <-  hotelling.test(.~g, data = temp)$stat$statistic
}


pvals_hotelling[[i]] <- mean(null >= results$stats$statistic)
}



pvals_hotelling
```

```
## $LM1
## [1] 0
##
## $LM2
## [1] 0
##
## $LM3
## [1] 0
##
## $UM1
## [1] 0
##
## $UM2
## [1] 0
##
## $UM3
## [1] 0
```

dist-based-test

```
load("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei-size/data/teethdata_scriptus_pricei-size.
pvals <- list()
for (toothtype in c("LM1","LM2","LM3","UM1","UM2","UM3")){print(toothtype)

n_scriptus <- length(data[[toothtype]][["scriptus"]])
n_pricei <- length(data[[toothtype]][["pricei"]])

class <- c(rep("scriptus",n_scriptus),rep("pricei",n_pricei))
```

```
#Pariwise distances
#First rows are scriptus and last rows are pricei
ddd <- read.csv(paste0("/Users/nastaranghorbani/Documents/teeth-scriptus-pricei-size/data/matlab/pairwis
ddd <- as.matrix(ddd)

#Distrance based permutation testing.
#Based on the test defined in Soto et al 2021
#1. Use distances based on the shapes projected into the tangent space.
#to 2. Use distances in the size-shape space.  (I just need a function that computes distance between s


Dbar11 <- sum(ddd[class == "scriptus",class == "scriptus"])/(n_scriptus^2)
Dbar22 <- sum(ddd[class == "pricei",class == "pricei"])/(n_pricei^2)
Dbar12 <- sum(ddd[class == "pricei",class == "scriptus"])/(n_pricei*n_pricei)

S <- ((n_scriptus*n_pricei)/((n_scriptus+n_pricei)))*(2*Dbar12 - (Dbar11 + Dbar22))

#Now permute
Sperm <- c()
nsim <- 100000
for (i in 1:nsim){
class_perm <- sample(class,length(class),replace = FALSE)
Dbar11 <- sum(ddd[class_perm == "scriptus",class_perm == "scriptus"])/(n_scriptus^2)
Dbar22 <- sum(ddd[class_perm == "pricei",class_perm == "pricei"])/(n_pricei^2)
Dbar12 <- sum(ddd[class_perm == "pricei",class_perm == "scriptus"])/(n_pricei*n_pricei)

Sperm[i] <- ((n_scriptus*n_pricei)/((n_scriptus+n_pricei)))*(2*Dbar12 - (Dbar11 + Dbar22))
}

pvals[[toothtype]] <- mean(Sperm >= S)

}
```

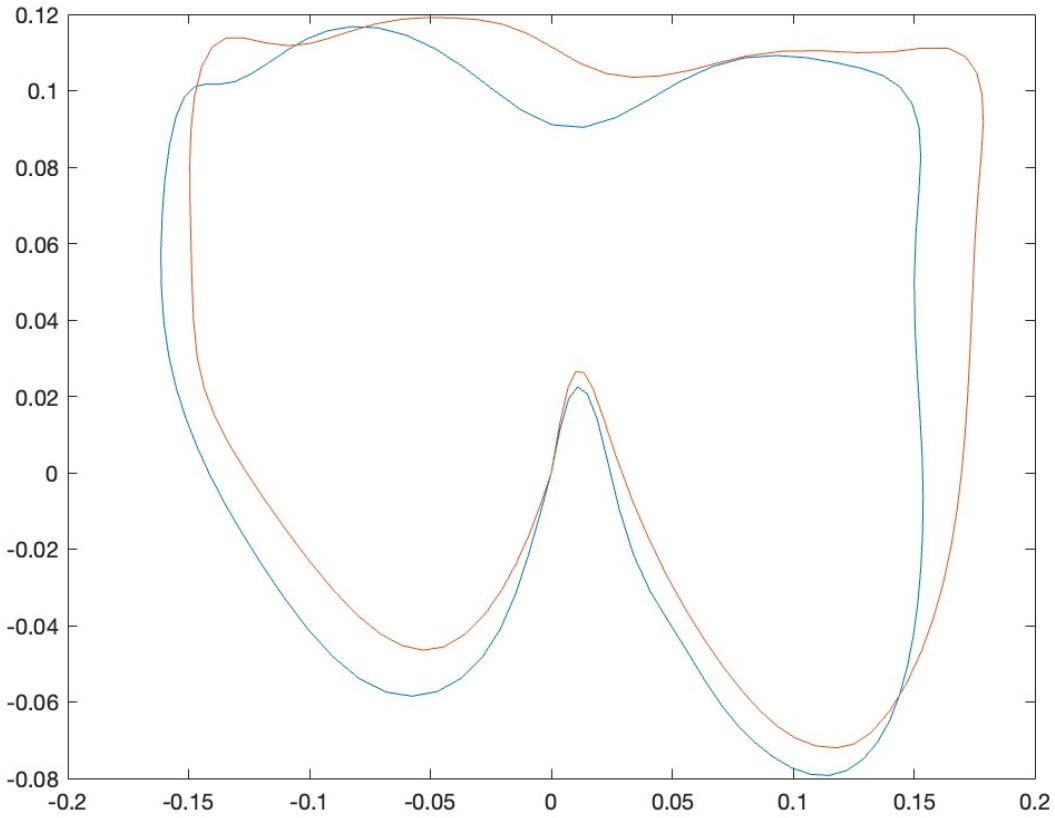
```
## [1] "LM1"
## [1] "LM2"
## [1] "LM3"
## [1] "UM1"
## [1] "UM2"
## [1] "UM3"
```

```
pvals
```

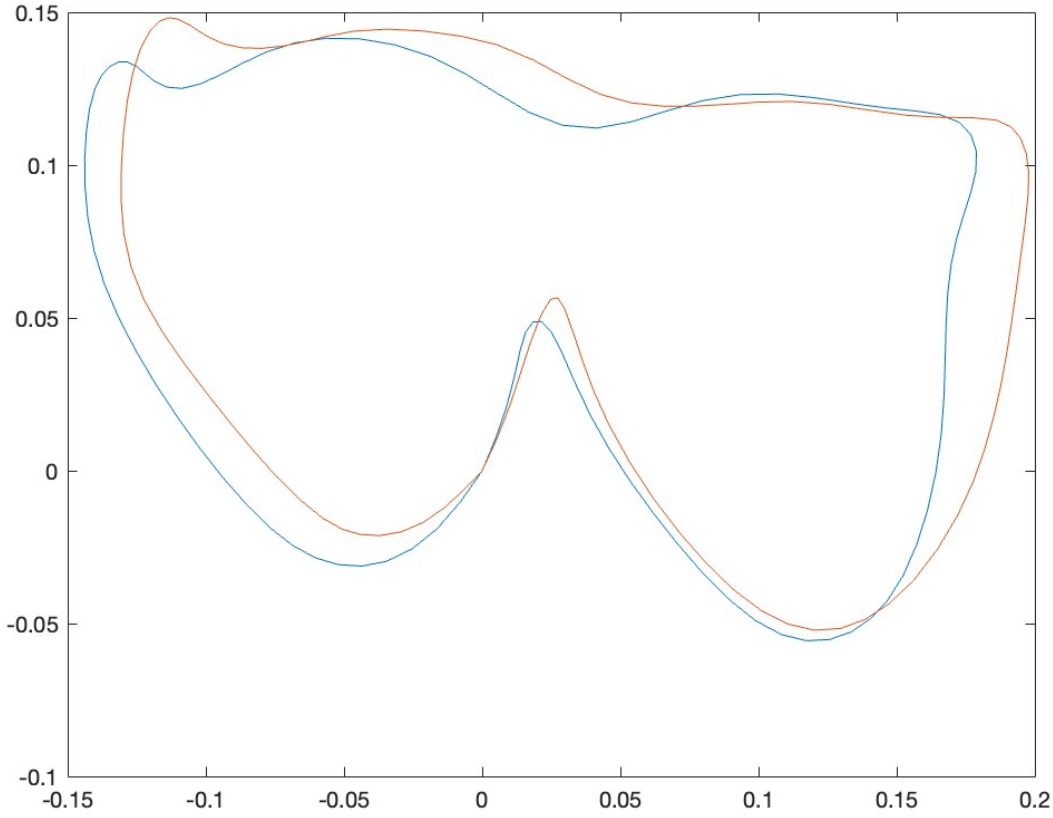
```
## $LM1
## [1] 1e-05
##
## $LM2
## [1] 0
##
## $LM3
## [1] 3e-05
##
```

```
## $UM1
## [1] 0
##
## $UM2
## [1] 0
##
## $UM3
## [1] 0
```

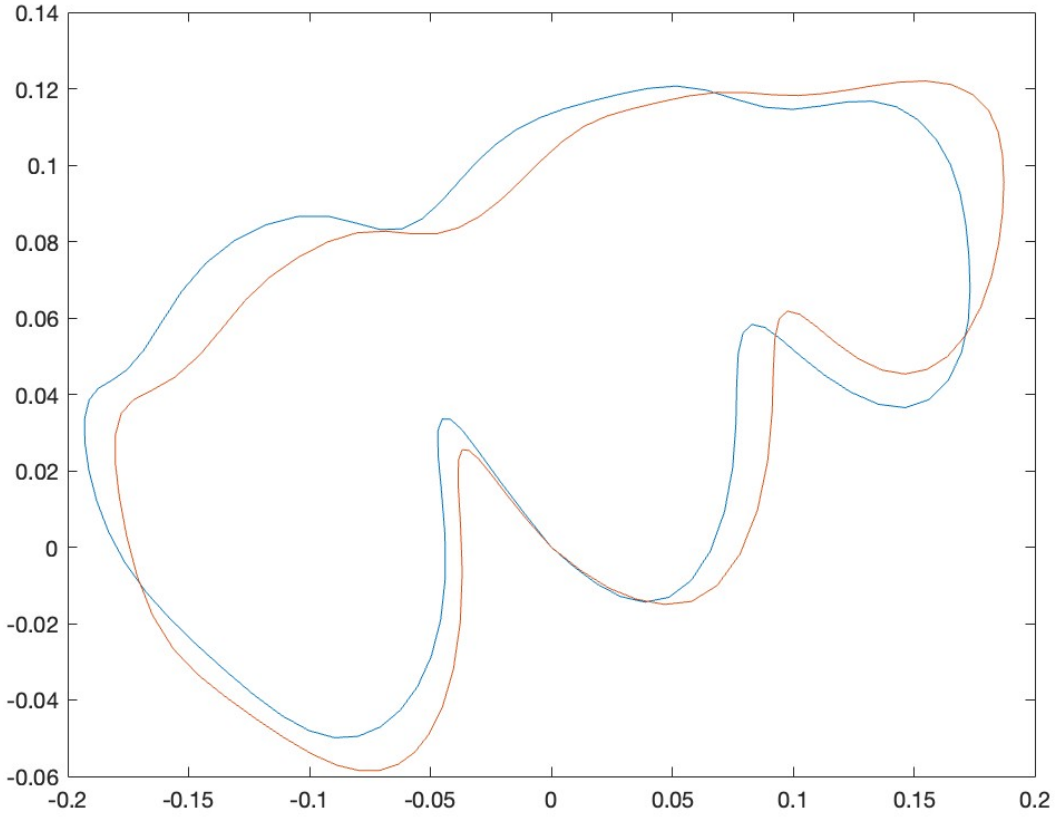
plots



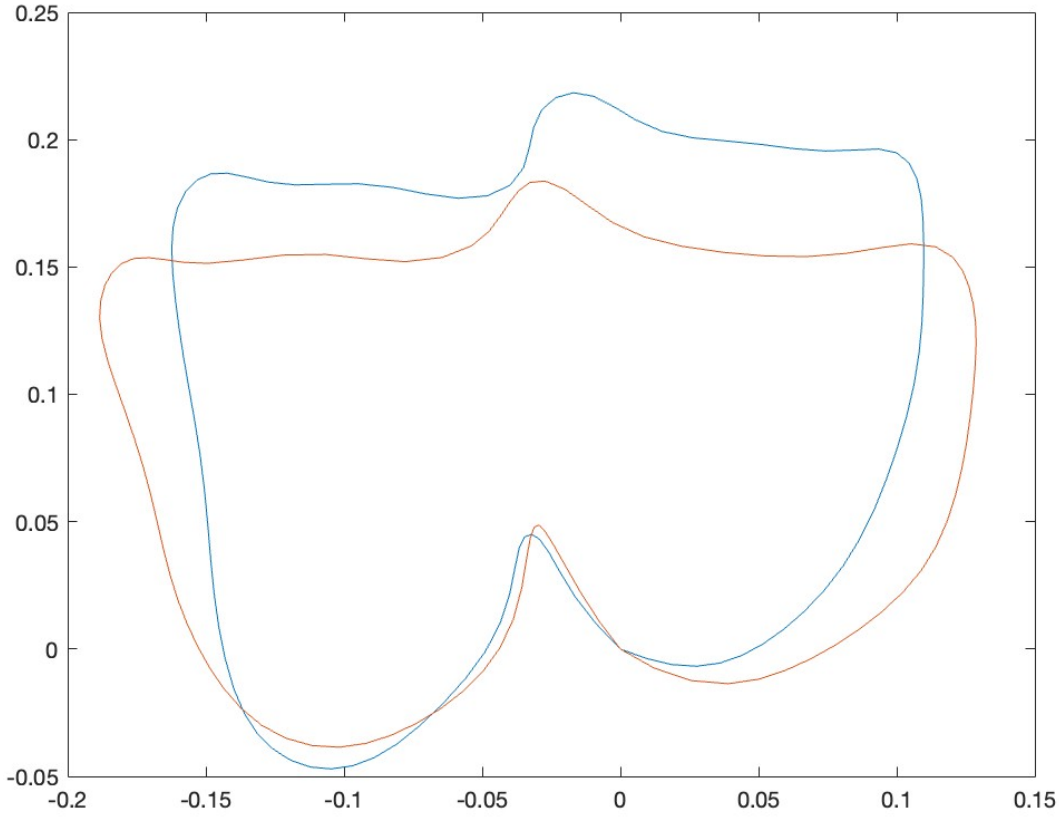

```
## LM1
```



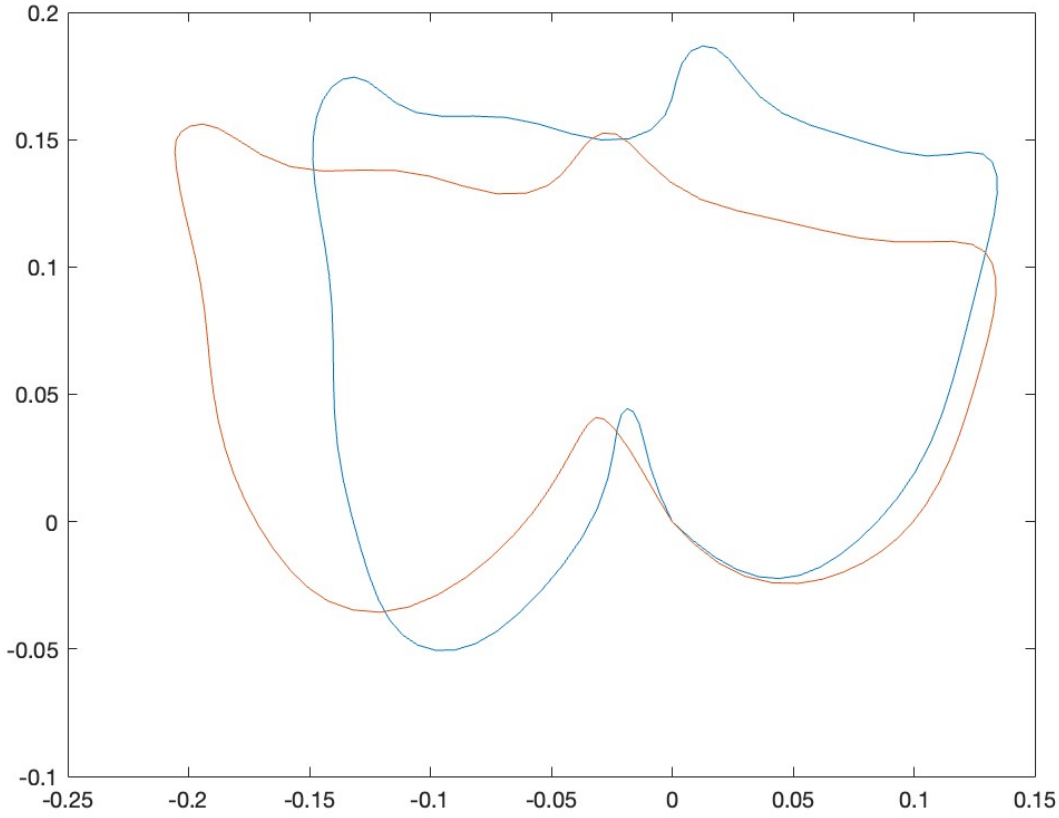
LM2


0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2

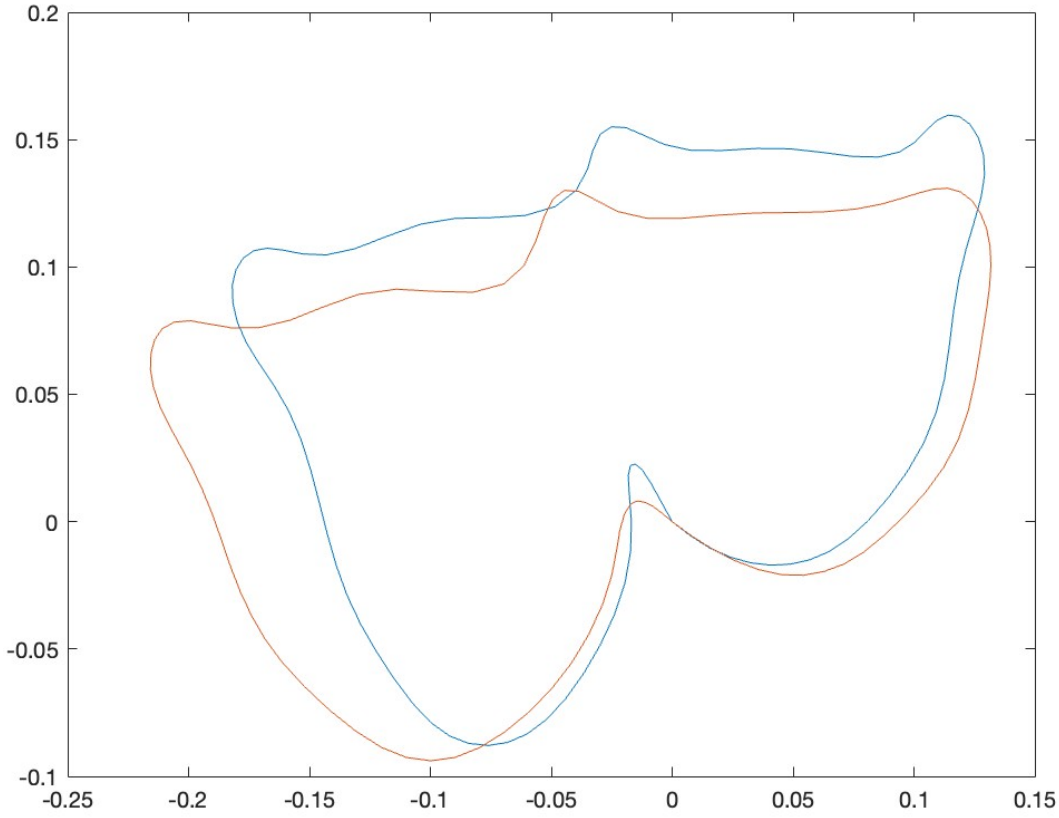
LM3




```
## UM1
```



UM2



UM3