

Introduction

Ce TP a pour but de se familiariser avec le traitement des points homologues (landmarks) sous R. Le jeu de données consiste en 5 statères (monnaies grecques), prises 2 fois en photo (les images sont dans le dossier 'Statere'). La première partie du TD a pour but d'expliquer comment faire l'acquisition des points homologues, la deuxième partie est consacrée à leur traitement par Analyse Procruste Généralisée (GPA).

	a	b
LeRider_022		
LeRider_026		
LeRider_083		
LeRider_091		
LeRider_092		

Acquisition des points homologues

A. Possibilités

Depuis les années 90, plusieurs logiciels de morphométrie géométrique ont été créés :

IMP (?) – non mis à jour

TPS (F. Rohlf) – sur Windows ; Contient un software pour l'acquisition de points (TPSdig2)

NTSYS (extersoftware) – logiciel payant,

Shape (Hiroyoshi Iwata) – pour les contours fermés (EFA)

MorphoJ (Chris Klingenberg) - cross-platforms (Java)

l'EVAN Toolbox (Evan consortium) – toutes les plateformes

Sous R, plusieurs packages permettant de traiter les informations morphométriques sont disponibles :

shapes (I. Dryden)

Rmorph (M. Baylac) - non dispo sur le site du CRAN

geomorph (D. Adams) - le plus récent

Morpho - le plus récent (3D)

B. Initialisation

Avant de commencer, définissons notre répertoire de travail 'Seance 3', c'est à dire le dossier où se trouvent les fichiers dont on aura besoin

```
setwd("C:/Users/XXX/Desktop/Seance 3/")
```

Nous allons travailler avec le package 'geomorph' car il permet l'acquisition des points homologues. Sinon il faudrait soit acquérir les données avec TPSdig2, soit avec par exemple ImageJ puis importer les données sous R. Ici, nous travaillons seulement sous R en utilisant le package 'geomorph'.

```
install.packages("geomorph")  
library(geomorph)
```

En plus, nous auront besoin de quelques fonctions complémentaires. Elles sont stockées dans le fichier 'archoMorpho.R'. On peut les charger aussi.

```
source("archoMorpho.R")
```

C. Acquisition des points homologues

Avant de commencer la digitalisation, il faut définir le nombre de points homologues avec lesquels nous voulons travailler.

On travaille avec des images qui sont en 2D, donc chaque point aura 2 coordonnées (x et y).

```
n.dim <- 2
```

Sur les statères (monnaies grecques), on va travailler avec 16 points :

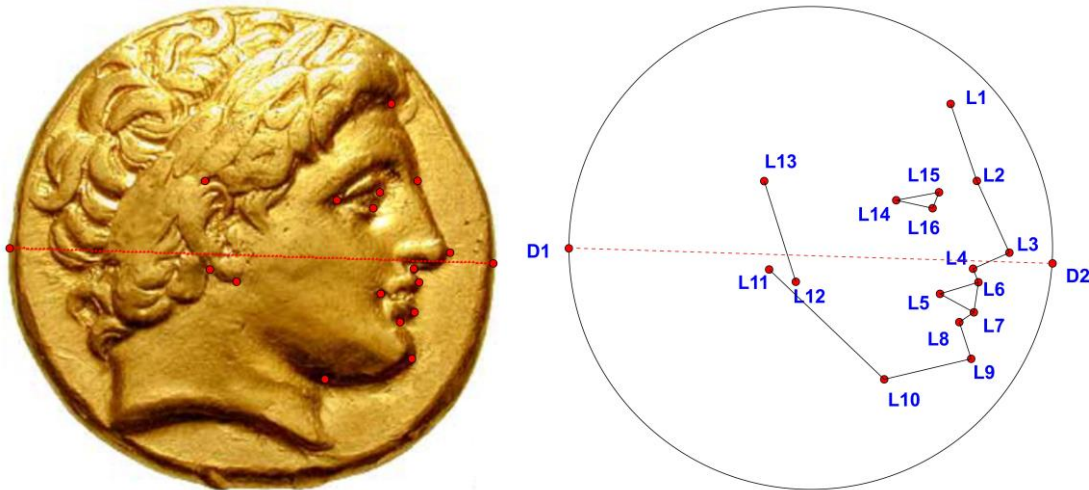
```
n.land <- 16
```

Des points homologues peuvent être digitalisés par la fonction 'digitize2d'. Faisons donc un test sur des images 'aLeRider_091@16.jpg' et 'aLeRider_091@16_b.jpg', en sauvegardant les résultats dans le fichier 'test.tps'.

```
digitize2d(c("aLeRider_091.jpg", "aLeRider_092.jpg"), n.land, 1, "test.tps")
```

Note : si une erreur se produit ('Error in digstart:length(filelist) : result would be too long a vector'), il faut d'abord effacer le fichier 'test.tps' du dossier 'Seance 3'.

Les 16 points homologues (landmarks) sont définis par de simples « clics gauches » de la souris. Avant de commencer la digitalisation des points, il faut néanmoins définir l'échelle. Comme les images ne contiennent pas de réelles échelles, on va définir l'échelle à l'aide du diamètre des statères. Cliquer donc sur deux points, qui définissent le diamètre de la monnaie (D1 et D2 sur l'image) et ensuite, vectorisez les 16 landmarks d'après le schéma.



Notez, que la fonction vous guide. Elle va toujours demander si votre digitalisation est correcte :

Keep scale (y/n)?

Keep Landmark 1 (y/n/a)?

Il faut toujours répondre. En acceptant, vous poursuivez au point suivant, en rejetant, vous pouvez le corriger. A la fin, la fonction va vous poser la question, si vous voulez procéder de même avec le deuxième individu.

Continue to next specimen (y/n)?

Acceptez, et faites la digitalisation du deuxième spécimen.

Une fois fini, vous pouvez ouvrir (en 'Bloc-notes') le fichier 'test.tps' qui a été créé et regarder sa structure :

LM donne le nombre de landmarks (donc 16)

Les lignes suivantes représentent les coordonnées x, y des 16 landmarks

SCALE correspondant à l'échelle

ID correspond à l'identifiant c.-à-d. au nom du fichier que vous avez digitalisé.

Vous avez ensuite les mêmes informations pour le deuxième individu.

```

test.tps - Bloc-notes
Fichier Edition Format Affichage ?
LM=16
233.009650482524 236.188309415471
248.400420021001 190.826041302065
267.841392069604 147.893894694735
246.780339016951 138.173408670433
227.339366068348 123.592679633982
249.210460523026 130.073003650182
246.780339016951 113.062153107655
238.6799339967 106.581829091455
245.160258012901 84.7107355367768
193.317665883294 72.5601280064003
124.464223211161 138.173408670433
140.665033251663 130.883044152208
122.034101705005 190.826041302065
200.60803040152 179.485474273714
225.719285964298 184.345717285864
222.479123956198 174.625231261563
SCALE=0.00343576160875509
ID=aLeR1der_091@16.jpg

LM=16
232.453326456937 238.004125838061
247.770500257865 187.875193398659
266.568849922641 147.49355337803
247.074265085095 137.746260959257
227.57968024755 124.517792676637
247.770500257865 130.783909231563
247.074265085095 116.859205776173
238.719443011862 107.111913357401
246.378029912326 89.706034038164
192.071686436307 71.6039195461578
123.14440433213 137.746260959257
137.069107787519 130.783909231563
124.536874677669 192.052604435276
203.211449200619 178.127900979887
225.490974729242 186.48272305312
221.313563692625 174.646725116039
SCALE=0.00343576160875509
ID=aLeR1der_091@16_b.jpg

```

Qualité de digitalisation (comparaison des deux sessions) et différences entre les monnaies

Maintenant nous allons faire une étude plus sérieuse. Avant de commencer une étude morphométrique, il faut s'assurer que les landmarks ont été précisément localisés sur les images. Dans l'exemple suivant on va travailler avec 5 statères, puis estimer la qualité de la digitalisation. Pour se faire, les landmarks sont placés deux fois par individu. Ensuite on va:

- (i) examiner visuellement les différences entre les deux sessions.
- (ii) tester statistiquement la significativité de ces différences.
- (iii) calculer et visualiser la précision de localisation de chaque landmark pour identifier les points difficilement localisables.
- (iv) tester s'il y a des différences entre les monnaies.
- (v) visualiser ses différences.

1. Digitalisation

Avant de commencer, nous allons définir le dossier contenant les images :

```
setwd("...")
```

Nous allons traiter toutes les images dans un même temps. Listons tous les fichiers dans le dossier Statere ('rmq: getwd()') donne l'adresse du dossier), et stockons le nom de toutes ces images 'jpg' du dossier Statere dans l'objet 'img'

```
img <- list.files(path=getwd(), pattern='.jpg')  
print(img)
```

```
[1] "aLeRider_022.jpg" "aLeRider_026.jpg" "aLeRider_083.jpg" "aLeRider_091.jpg" "aLeRider_092.jpg"  
[6] "bLeRider_022.jpg" "bLeRider_026.jpg" "bLeRider_083.jpg" "bLeRider_091.jpg" "bLeRider_092.jpg"
```

Il y a donc 10 images de 5 statères prises en photo deux fois. Construisons trois vecteurs contenant ces informations.

```
n.img <- 10  
n.session <- 2  
n.ind <- n.img/n.session
```

...et commençons la prise de données

```
digitize2d(img,n.land,1, "statere.tps")
```

La fonction a créé le fichier 'statere.tps'. Pour vous en assurer, ouvrez-le en 'Bloc-notes' et vérifiez que toutes les coordonnées ont bien été stockées. Le fichier doit contenir 10 paragraphes (pour 2 x 5 individus), et aucune valeur ne doit être 'NA' ou 'NaN'. Si vous rencontrez un problème, essayez de le corriger.

2. Charger tps

Si tout est bon, on peut lire le fichier 'tps' qui vient d'être créé sous R en appelant la fonction 'readland.tps'

```
xy.pts <- readland.tps("statere.tps", specID = "ID", warnmsg = T)
```

Ensuite créons un facteur 'ind' et un facteur 'session'

Juste pour rappel : un facteur est une variable catégorielle qui définit des groupes. Par exemple si l'on compare la taille des hommes à celle des femmes, le facteur (ou 'grouping factor' ou 'grouping vector')

est 'genre'. Les niveaux ('levels') d'un facteur sont ses valeurs ('modes'). Par exemple le facteur 'genre' possède deux niveaux – 'Homme' et 'Femme'.

Dans notre cas, le facteur est 'session' et il possède deux niveaux ('1' pour la première et '2' pour la deuxième digitalisation d'un individu) :

```
session <- gl(n.session,n.ind)
```

Le facteur 'ind' va ensuite contenir les noms des fichiers :

```
ind <- as.factor(rep(1:n.ind,n.session))
```

On peut créer des facteurs en utilisant la fonction 'substring' appliquée sur les noms d'images.

Pour savoir comment cette fonction marche, essayez de lancer la ligne suivante

```
substring("Abrakadabradabra", 3, 6)
```

```
[1] "raka"
```

elle nous a donné "raka" – c'est-à-dire de la troisième à la sixième lettre du mot "Abrakadabradabra"

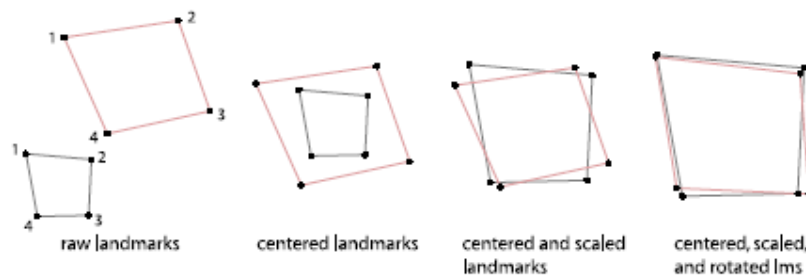
```
ind <- as.factor(substr(img,2,12))
```

```
session <- as.factor(substr(img,1,1))
```

3. Superposition Procruste (GPA: Ajustement Procruste généralisé)

L'analyse de Procruste a pour but de standardiser la position, la taille et la rotation des objets.

Fig. 3 The three steps of Procrustes superimposition: translation to a common origin, scaling to unit centroid size, and rotation to minimize the sum of squared Euclidean distances among the homologous landmarks. The resulting landmark coordinates are called Procrustes shape coordinates



En supprimant ces trois effets (position, taille et rotation), on observe uniquement les différences de formes existantes entre les objets. Ces différences sont numériquement exprimées par les « distances de Procruste » (Procrustes distances).

L'Analyse de GPA en R peut être fait grâce à la fonction 'gpagen'

```
gpa <- gpagen(xy.pts, ShowPlot=FALSE)
```

On peut examiner ce que GPA contient en utilisant la fonction 'str' qui nous retourne la 'structure' d'un objet R

```
str(gpa)
```

L'objet 'gpa' contient deux listes : \$coords et \$csize

```
gpa$coords
```

...nous donne un 'array' qui contient les landmarks standardisés par GPA pour chaque individu

```
gpa$csize
```

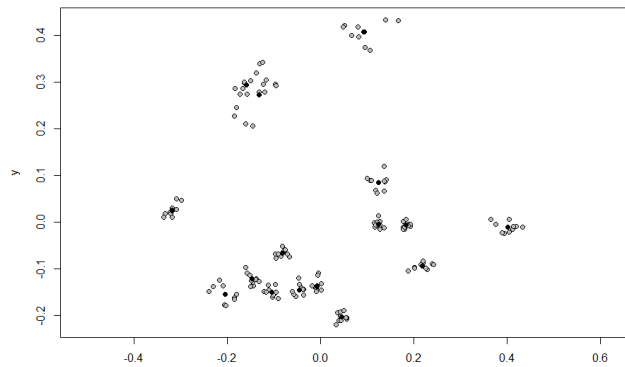
...nous donne l'information sur 'le centroid size' pour chaque individu. Centroid size est la somme des distances entre le centroïde et chacun des landmarks

3.1. Visualisation des résultats

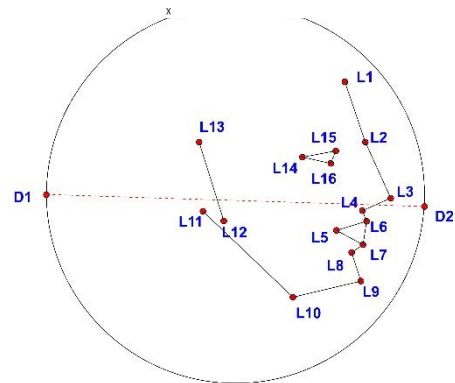
Pour visualiser les résultats, on peut appeler la fonction 'plotAllSpecimens'

```
plotAllSpecimens(gpa$coords, pointscale=1, meansize=1)
```

Les points en gris correspondent aux landmarks tandis que les points noirs à la 'meanshape'. L'image peut néanmoins paraître un peu illisible - pour voir les visages qu'on a défini, il faut avoir beaucoup d'imagination... Pour mieux visualiser les visages, on pourra ajouter les lignes qui de liaison entre certains landmarks. On va donc créer une matrice 'myLinks' telle que pour chaque ligne, on définit les liaisons entre landmarks.



Sur l'image, on voit par exemple que le 1er landmark est lié au 2eme, le 2eme au 3eme, le 3eme au 4eme, le 5eme au 6eme, etc.



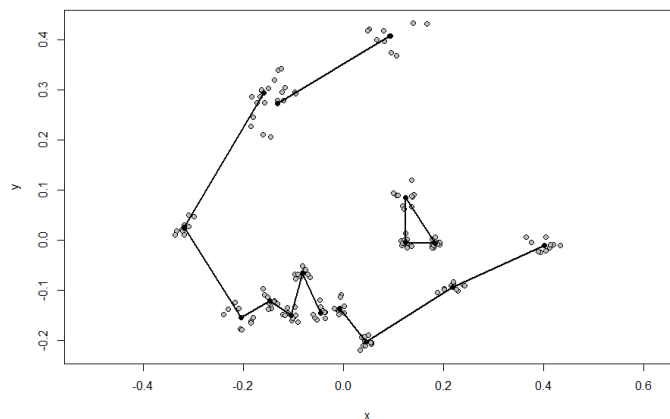
Ces liaisons sont donc listées dans la matrice 'myLinks'

```
myLinks <- cbind(c(1:3,5,5,7:10,12,14:16),c(2:4,6,7,8:11,13,15,16,14))
```

Une fois les liaisons définies, on peut de nouveau appeler la fonction 'plotAllSpecimens', mais maintenant en ajoutant le paramètre 'links' auquel on va associer nos liaisons 'myLinks'

```
for (i in 1:dim(myLinks)[1]) { lines(rbind(mshape(gpa$coords)[myLinks[i,1],], mshape(gpa$coords)[myLinks[i,2],]), lwd=2) }
```

De cette manière le visage est plus clair. Sur l'image, on peut observer les disparités des landmarks (en gris) autour de leurs moyennes (en noir). Ces disparités peuvent nous donner un premier indice sur la précision de digitalisation, mais il montre aussi la différence entre les individus. Par exemple sur l'image, on peut voir beaucoup de disparités autour d'oreille (L11, L12 et L13) et autour du landmark correspondant à l'œil (L14).



3.2. Calcul de la mean shape

La mean shape (c-à-d la forme moyenne de tous les individus) a été calculée automatiquement par la fonction 'plotAllSpecimens'. Elle apparaît en noir sur l'image. Pour connaître ses coordonnées, par la fonction 'mshape' est appliquée aux landmarks standardisés par GPA :

```
msh <- mshape(gpa$coords)
```

La matrice 'msh' contient les coordonnées de la mean shape de tous les landmarks, c.-à-d. des deux sessions. Pour voir cette mean shape, on peut simplement passer par la fonction 'plot':

```
plot(msh, asp=1)
```

3.3. Calcul de la mean shape pour chaque session

Pour visualiser les différences entre les deux sessions, on peut calculer les mean shapes des deux sessions séparément :

```
msh.1 <- mshape(gpa$coords[, ,which(session=="a")])  
msh.2 <- mshape(gpa$coords[, ,which(session=="b")])
```

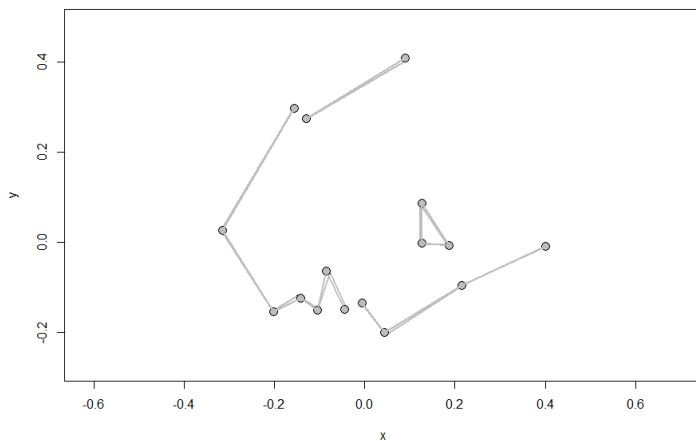
3.4. Visualisation de la différence de moyenne entre les sessions

Une fois les deux mean shapes calculées, il est possible de les représenter sur le même graphique. On peut passer par la fonction '`plotRefToTarget`' qui va :

- (i) prendre la deuxième mean shape ('msh.2')
- (ii) l'ajuster (par GPA) sur la première ('msh.1')
- (iii) puis réaliser un graphe

Ensuite on peut injecter la première mean shape ('msh.1') dans le graphique grâce à la fonction '`add.mshape.links`'

```
plotRefToTarget(msh.1,msh.2,method="vector",mag=1,links=myLinks)  
add.mshape.links(msh.1,myLinks,lwd=2,col="gray")
```



4. Transformation des données d'un array vers une matrice

Dans l'étape suivante, il s'agit de réaliser une ACP. La fonction '`prcomp`' ne travaille pas avec des 'array'. Il faut donc d'abord transformer ces données en une matrice avec $n \times (2k)$ dimensions où n est le nombre d'individus (ici $n = 10$ car on a 10 individus) et k est le nombre de dimensions des coordonnées (ici $k = 2$ soit les abscisses et les ordonnées). La façon la plus simple est de passer par la fonction '`two.d.array`'

```
shape <- two.d.array(gpa$coords)  
shape
```

Dans la matrice '`shape`' on a donc dans les lignes les individus et dans les colonnes les coordonnées des landmarks. Pour s'en assurer, on peut regarder le nombre des dimensions de la matrice '`shape`'

```
dim(shape)
```

```
[1] 10 32
```

On voit que la matrice a comme dimension : 10 (individus) x 32 (x et y coordonnées)

5. Analyse en composante principale (ACP)

Pour visualiser différemment les différences entre nos digitalisations, on peut passer par l'Analyse en Composantes Principales (ACP). Par l'ACP on va créer un espace morphométrique (morphospace), défini par les Composantes Principales (CP), dans lesquelles on va projeter nos individus. Les distances entre les individus dans ce morphospace traduisent les similarités de formes.

Avec l'ACP, on va surtout réduire la dimensionnalité en négligeant les dimensions qui ne représentent que peu de variance. Ici la fonction 'prcomp' est utilisée.

```
pcs <- prcomp(shape)
```

Pour tirer la table des pourcentages de variance totale exprimée par les différentes Composantes Principales :

```
table.pca <- summary(pcs)$importance
table.pca
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	0.07439169	0.04027529	0.03562874	0.02007485	0.01562456	0.01189202	0.0100744
Proportion of Variance	0.58572000	0.17168000	0.13435000	0.04265000	0.02584000	0.01497000	0.0107400
Cumulative Proportion	0.58572000	0.75740000	0.89175000	0.93440000	0.96024000	0.97521000	0.9859500

	PC8	PC9	PC10
Standard deviation	0.009045725	0.007136713	4.221039e-17
Proportion of Variance	0.008660000	0.005390000	0.000000e+00
Cumulative Proportion	0.994610000	1.000000000	1.000000e+00

On voit, qu'au lieu d'utiliser 32 dimensions (c.-à-d. 32 x et y coordonnées), il nous suffit de prendre quelques (dans ce cas deux ou trois) CPs, qui contiennent l'essentiel de la variance originale.

Maintenant regardons graphiquement les résultats de l'ACP. Afin de tout voir sur la même image, on peut préparer un espace graphique 2x2 dans lequel on va injecter les résultats de l'ACP :

```
Layout(matrix(1:4,2,2))
```

On peut faire la même chose avec: `par(mfrow=c(2,2))` - c'est à vous de choisir...

Le premier graphique montre le pourcentage de variance exprimé par chaque CP.

Le deuxième graphique est un 'biplot' - c.à.d. la projection des individus et des variables sur les deux premiers axes (PC1 et PC2). Le 'biplot' est rarement utilisé dans les études morphométriques.

Le dernier graphique sera le Morphospace – la projection de PC1 versus PC2 avec les 2 sessions colorisées en bleu ou rouge.

D'abord barplot

```
barplot(table.pca[2,],ylab=rownames(table.pca)[2],xlab="PC",names.arg=colnames(table.pca))
```

Ensuit biplot :

```
biplot(pcs)
```

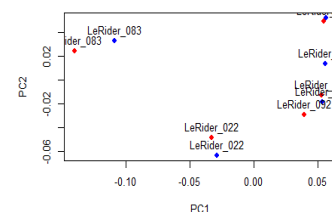
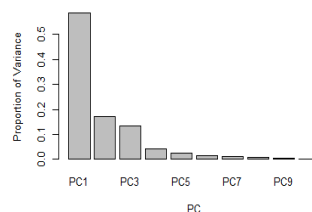
et à la fin PC1 vs PC2 :

```
plot(pcs$x,col=c(rep("blue",n.ind),rep("red",n.ind)),pch=19)
```

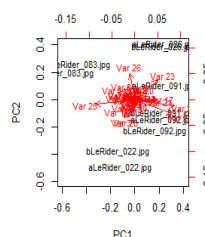
Pour savoir quel point correspond à quel individu, on peut injecter leurs noms

```
text(pcs$x[,1],pcs$x[,2],labels=ind,pos=3)
```

La majorité de la variance totale est portée par trois CPs (barplot).



Dans PC1 vs PC2 on observe les différences entre digitalisations. Il existe une différence entre les deux digitalisations du même individu, mais ces différences sont faibles face à la différence entre individus.



6. Tests des différences

On voit donc qu'il existe des différences concernant la digitalisation des individus durant les deux sessions. Pour savoir si ces différences sont significatives face aux différences entre les individus, on peut utiliser plusieurs tests statistiques.

En général, plusieurs options existent. On peut utiliser par exemple :

- 1) le Goodall-F test du package 'geomorph'
- 2) la MANOVA
- 3) le test de Hotelling

Dans tous les tests l'hypothèse nulle correspond à l'absence de différence entre les mesures.

6.1. le Goodall-F test de geomorph:

Goodall-F test ou Procrustes ANOVA sont similaires à l'Analyse de la variance à un facteur (ANOVA). De la même manière, le calcul de F est basé sur le rapport entre 'Between-group SS' et 'Within-group SS' (avec SS pour sum-of-square). La différence est que pour mesurer les SS, des "sum-of-squared Procrustes distances" sont utilisées. La distribution de probabilités des valeurs F pour le test est obtenue par permutation (pour plus d'informations envoyez '[help\("procD.lm"\)](#)').

En pratique, pour voir si le facteur (d'individu ou de la session) a de l'effet sur la forme, on peut passer par la fonction '[procD.lm](#)' qui calcule l'ANOVA Procrustes. Voici comment on applique la fonction dans R : `procD.lm(var.ind~var.dep)`

les variables indépendantes sont ici les landmarks standardisées par GPA (matrix '[shape](#)')

les variables dépendantes sont les individus ou la session (facteur '[ind](#)' ou '[session](#)')

Dans notre cas, l'effet d'individu est donc :

```
pAoV <- procD.lm(shape~ind,iter=999)
```

```
pAoV
```

```
Type I (Sequential) Sums of Squares and Cross-products
```

```
Randomization of Raw Values used
```

	df	SS	MS	Rsqr	F	Z	P.value
ind	4	0.078015	0.0195038	0.91744	13.891	2.0231	0.001
Residuals	5	0.007020	0.0014041				
Total	9	0.085036					

On voit donc que dans ce cas, on peut rejeter H0 – les individus sont différents.

On peut faire le même test pour savoir s'il y a des différences entre sessions :

```
pAoV2 <- procD.lm(shape~session,iter=999)
```

```
pAoV2
```

```
Type I (Sequential) Sums of Squares and Cross-products
```

```
Randomization of Raw Values used
```

	df	SS	MS	Rsqr	F	Z	P.value
session	1	0.001396	0.0013958	0.016415	0.1335	0.12209	0.9355
Residuals	8	0.083640	0.0104550				
Total	9	0.085036					

6.2. MANOVA

La Multivariate ANalysis Of VAriance (MANOVA) teste si les groupes sont significativement différents. La MANOVA peut être vue comme une extension du F-test (ANOVA). L'ANOVA teste s'il y a des différences significatives entre deux ou plusieurs groupes sur une seule variable. La MANOVA fait la même chose, mais sur plusieurs variables. Comme les variables de forme sont multidimensionnelles, la procédure est particulièrement bien adaptée.

Néanmoins, nous ne pouvons pas appliquer la MANOVA directement sur les CPs calculées à partir de la GPA parce que:

- (i) certaines CPs peuvent être nulles - c.-à-d. qu'elles ne contiennent aucune variance
- (ii) en appliquant la GPA, on perd (dans le cas 2D) 4 degrés de liberté et les calculs ne peuvent être réalisés.

Pour ces raisons, il est nécessaire de préparer les données.

De toutes les Composantes Principales obtenues par ACP, on va choisir celles qui sont non nulles (c.-à-d. celles qui contiennent au moins un peu de variance). Parce que les humains, machines et logiciels (R inclus) n'ont pas la même compréhension du "0", il faut d'abord regarder comment la plus petite valeur est définie dans le logiciel R. Cette valeur peut être obtenue grâce à la ligne suivante :

```
.Machine$double.eps
```

Il s'agit alors de sélectionner les CPs qui ont une variance non nulle

```
non.null <- which(pcs$sdev>.Machine$double.eps)
```

Ensuite on élimine 4 degrés de libertés, soit les 4 dernières valeurs

```
non.null <- non.null[-c((length(non.null)-3):length(non.null))]
```

Les CPs non nulles sont sélectionnées et on applique la MANOVA. La question est ici de savoir s'il y a des différences de formes entre les individus

```
mod.1 <- manova(pcs$x[,non.null]~ind)
summary(mod.1,test="Pillai") #default: pillai trace
      Df Pillai approx F num Df den Df    Pr(>F)
ind      4 3.8615   22.312     20   16 3.784e-08 ***
Residuals  5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

L'hypothèse nulle est rejetée.

De même manière, on examine s'il existe des différences de formes entre les deux sessions :

```
mod.1s <- manova(pcs$x[,non.null]~session)
summary(mod.1s,test="Pillai") #default: pillai trace
      Df Pillai approx F num Df den Df    Pr(>F)
session  1 0.21204  0.21527     5     4 0.9385
Residuals  8
```

6.3 Hotelling test d'après Claude (2008)

L'approche de Hotelling peut être aussi vue comme un type d'ANOVA. Dans notre cas, on se pose la question suivante : la 'session' a-t-elle un effet sur 'la forme'. Un modèle linéaire est défini: ' $lm(\text{forme} \sim \text{session})$ ' où '*forme*' contient les coordonnées et '*session*' est le facteur (ou effet).

Ce modèle permet ensuite de décomposer les différentes sources de variances qui se nichent dans la variance totale (exprimé par les SS). Cela nous permet de définir :

- La quantité de variance qui s'explique par le facteur (c.-à-d. par la session) => notée '*Ssef*' (comme Sum-of-squares of effect)
- La quantité de variance qui ne s'explique pas par le facteur => notée '*Sser*' (comme Sum-of-squares of error)

Pour chacune de ces variances, on définit le nombre de degrés de liberté (ddl):

- '*dfef*' (ddl pour effet) = $p - 1$ où p est le nombre de niveaux du facteur
- '*dferr*' (ddl pour error) = $n - p$ où p est le nombre de niveaux du facteur et n le nombre d'individus

Une fois ces quatre valeurs déterminées, on les introduit dans la fonction 'Hotellingsp'

Pour récapituler, la fonction 'Hotellingsp' nécessite les 4 paramètres suivants :

- 'Ssef': Sum of squares and cross-products of effect (dans notre cas, l'effet est soit 'ind', soit 'session')
- 'SSer': Sum of squares and cross-products of residual variation
- 'dfef': ddl pour l'effet
- 'dfer': ddl pour l'erreur

D'abord il faut appliquer le modèle linéaire. Comme facteur, on va d'abord regarder l'effet 'individu'.

```
library(MASS)
fact <- ind
n <- 10
```

```
p <- 5
mod1 <- lm(pcs$x[,1:4]~as.factor(factor))
```

Ensuite, il faut calculer les matrices de dispersion du facteur/effect et de l'erreur et les degrés de liberté

```
Ssef <- (n-1)*var(mod1$fitted.values)
SSer <- (n-1)*var(mod1$residuals)
dfef <- length(levels(factor))-1
dfer <- n - length(levels(factor))
```

Enfin, le test de Hotelling est calculé.

```
Hotellingsp(Ssef, SSer, dfef, dfer)
  dfeffect  dferror  T2  Approx_F  df1  df2  p
4.00000000 5.00000000 813.26078669 25.41439958 16.00000000 2.00000000 0.03849097
```

On rejette l'hypothèse nulle.

De la même manière on peut se demander, si la session produit un effet sur les différences entre les formes.

```
fact <- session
n <- 10
p <- 2
mod1s <- lm(pcs$x[,1:4]~as.factor(factor))
Ssef <- (n-1)*var(mod1s$fitted.values)
SSer <- (n-1)*var(mod1s$residuals)
dfef <- length(levels(factor))-1
dfer <- n - length(levels(factor))
Hotellingsp(Ssef, SSer, dfef, dfer)
  dfeffect  dferror  T2  Approx_F  df1  df2  p
1.00000000 8.00000000 0.03853439 0.04816799 4.00000000 5.00000000 0.99419865
```

7. Calcul de l'erreur de mesure d'après Bailey and Byrnes (1990)

L'erreur de mesure peut être exprimée par la 'mean square error' (Bailey and Byrnes 1990) :

$$ME = \frac{MS \text{ between (ou among)}}{MS \text{ within} * p} * 100 = \frac{\frac{SSer}{dfer}}{\frac{Ssef}{dfer} - \left(\frac{SSer}{dfer} * p\right)} * 100$$

La fonction 'meas.error' nécessite les mêmes paramètres que la fonction 'Hotellingsp'

```
mod <- lm(shape~ind)
SSCPind <- crossprod(mod$effects[which(mod$assign==1),,drop=F])
SSCPerr <- crossprod(mod$residuals)
SSind <- sum(diag(SSCPind))
SSerr <- sum(diag(SSCPerr))
dferr <- (n.session-1)*n.ind
dfind <- (n.ind-1)
meas.error(SSind, SSerr, dfind, dferr)
  $s2_among
[1] 0.009049879

  $s2_within
[1] 0.00140409

  $me
[1] 13.43117
```

On voit, que cette erreur est égale à 13 %.

8. Calcul de l'erreur (version par landmarks)

ME peut être calculée pour chaque landmark pris séparément. Cela permet de quantifier la précision avec laquelle les landmarks ont été localisés sur chaque objet (dans le cas des deux ou plusieurs sessions). Mais comme nous avons trouvé précédemment qu'il n'y avait pas de différence entre les deux sessions, nous allons nous concentrer sur l'erreur liée aux différences entre individus.

D'abord il s'agit de préparer une matrice vide 'ME' contenant 16 colonnes (pour les 16 landmarks) et trois lignes (pour conserver l'information concernant s2_among, s2_within et ME). Ensuite, on passe en revue chaque landmark pour calculer les deux sources de variation, ainsi que la ME (voir ci-dessous).

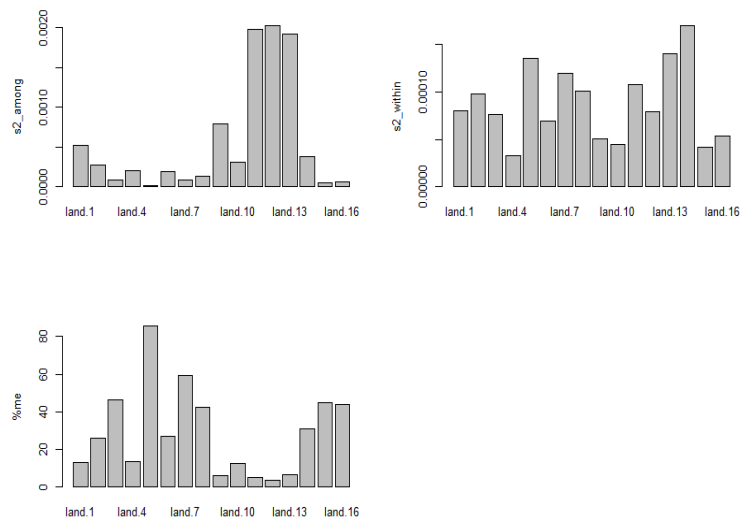
```
ME <- matrix(NA,3,n.land)
rownames(ME) <- c("s2_among","s2_within","%me")
colnames(ME) <- paste("land",1:n.land,sep=".")
land <- seq(1,n.land*n.dim,by=n.dim)
for (l in land){
  SSind <- sum(diag(SSCPind[l:(l+1),l:(l+1)]))
  SSerr <- sum(diag(SSCPerr[l:(l+1),l:(l+1)]))
  tmp <- meas.error(SSind,SSerr,dfind,dferr)
  ME[1,ceiling(l/n.dim)] <- tmp$s2_among
  ME[2,ceiling(l/n.dim)] <- tmp$s2_within
  ME[3,ceiling(l/n.dim)] <- tmp$me
}
ME
```

9. Visualisation des sources de variance

Trois barplots sont dessinées: deux pour examiner les deux sources de variation et un pour représenter la ME.

```
par(mfrow=c(2,2))
for (i in 1:3)
  barplot(ME[i,],ylab=rownames(ME)[i])
```

Le premier graphique (S2_among) montre la variance entre les individus : les landmarks L9, L11, L12 et L13 varient beaucoup. Ces landmarks sont très bons pour différencier les individus. Sur le graphique de droite (S2_within), on a la variation résiduelle, c.-à-d. la variation qui n'était pas expliquée par la différence entre individus. Ces différences peuvent être causées par la digitalisation (mais aussi par le hasard !). Enfin, la troisième image présente la qualité des landmarks pour le problème qui nous occupe.



10. Visualisation des effets : individu et erreur

Pour visualiser les changements de positions des landmarks entre les individus, il faut d'abord obtenir l'axe majeur de variation de la matrice de variation entre les individus.

```
vec.ind <- matrix(prcomp(SSCPind/dfind)$rotation[,1],n.land,n.dim,byrow=T)
```

Ce vecteur '`vec.ind`' donne les directions et les quantités de « déplacement » des landmarks. Cette information peut être visualisée :

```
plot(vec.ind, type="n",asp=1)
for (i in 1:dim(vec.ind)[1]) {lines(rbind(c(0,0), vec.ind[i,]))}
```

Ensuite, les coordonnées de la mean shape ('`msh`') sont « déplacées » par ce vecteur ('`vec.ind`'). Notez, que la valeur 0.2 correspond à la magnitude du déplacement (au lieu de 0.2, on peut choisir n'importe quelle valeur). Les nouvelles coordonnées sont stockées dans l'objet '`target`'

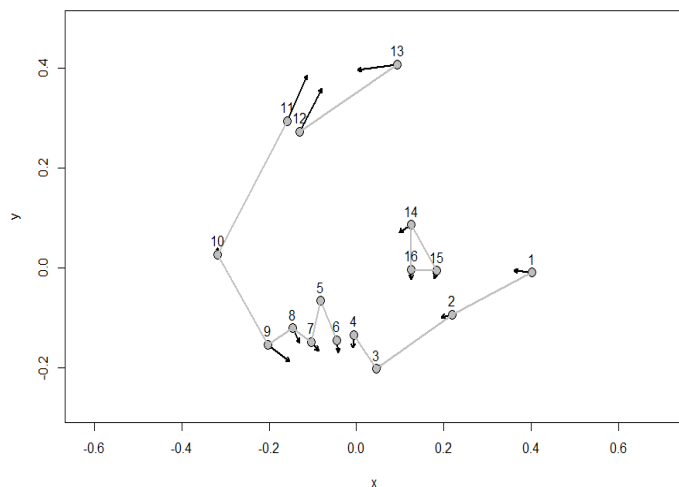
```
target <- msh + vec.ind * 0.2
```

Enfin, il s'agit de représenter à la fois '`msh`' et '`target`'

```
plotRefToTarget(msh,target,method="vector",mag=1)
add.mshape.links(msh,myLinks)
text(msh[,1],msh[,2],labels=1:n.land,pos=3)
```

Les directions et les magnitudes des déplacements des landmarks qui différencient mieux les individus sont clairement représentés. Les individus sont nettement différents au niveau des oreilles (L11-L13), et un peu au niveau du menton (L9).

La forme moyenne ('`msh`') est apparait en points gris et le '`target`' est défini par l'extrémité des flèches noirs. Ce graphique nous montre les mêmes tendances que le graphique (S2_among).



De la même manière, on peut visualiser l'axe majeur de variation de la matrice de variation résiduelle

```
vec.err <- matrix(svd(SSCPerr/dferr)$u[,1],n.land,n.dim,byrow=T)
target <- msh + vec.err * 0.2
plotRefToTarget(msh,target,method="vector",mag=1)
add.mshape.links(msh,myLinks)
text(msh[,1],msh[,2],labels=1:n.land,pos=3)
```

Remarque: Si nous avons travaillé sur les scores de l'ACP, alors nous aurions dû retourner dans l'espace des coordonnées via `mat.Ind <- eigenvec%*%MSCPind%*%t(eigenvec)`

Visualiser l'effet de la session

Pour un effet de session significatif, il est possible de visualiser les différences de mean shapes entre les deux sessions

```
msh.s1 <- apply(gpa$coords[,session=="a"],c(1,2),mean)
msh.s2 <- apply(gpa$coords[,session=="b"],c(1,2),mean)
target <- msh.s1 + (msh.s2-msh.s1)* 10
plotRefToTarget(msh.s1,target,method="vector",mag=1)
```

Deux types de denars

L'autre jeu des données contient trois lots de denars (monnaies romaines) découverts en Bohême. Le premier type de denar contient la tête de Antonius Pius (86-161 AD), le deuxième, celle de Marcus Aurelius (121-180 AD) et troisième, celle de Hadrien (76-138 AD).

Questions :

- 1) Trouvez s'il existe des différences entre deux types de monnaies (de votre choix).
- 2) Visualisez ces différences

Rédiger un rapport de 2 pages contenant une introduction, une description de la méthode, des résultats, et interprétez.