

# Package ‘multifluo’

July 12, 2018

**Type** Package

**Title** Dealing with Several Images of a Same Object Constituted of Different Zones

**Version** 1.1

**Date** 2018-07-12

**Author** Caroline Peltier

**Maintainer** Caroline Peltier <caroline.peltier@agrosupdijon.fr>

**Description** Deals with several images of a same object, constituted of different zones. Each image constitutes a variable for a given pixel. The user can interactively select different zones of an image. Then, multivariate analysis (PCA) can be run in order to characterize the different selected zones, according to the different images. Hotelling (Hotelling, 1931, <doi:10.1214/aoms/1177732979>) and Srivastava (Srivastava, 2009, <doi:10.1016/j.jmva.2006.11.002>) tests can be run to detect multivariate differences between the zones.

**License** GPL-3

**Imports** ellipse, imager, grDevices, graphics,stats,utils,agricolae

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-07-12 13:30:02 UTC

## R topics documented:

multifluo-package . . . . .	2
binning . . . . .	3
calc.ell . . . . .	4
count . . . . .	4
create.zones . . . . .	5
df.scaled . . . . .	6
dfboolTest . . . . .	6
dfTest . . . . .	6
difftable . . . . .	7

erase . . . . .	8
find.colors . . . . .	9
generic.plot . . . . .	9
gp . . . . .	10
hotelling.test . . . . .	10
img . . . . .	11
img.zone . . . . .	11
interieur.para . . . . .	12
intpca . . . . .	13
lifetime . . . . .	14
newgp . . . . .	14
pca . . . . .	15
pcapic . . . . .	16
plotimage . . . . .	16
plotpca . . . . .	18
plotzone . . . . .	19
reshaping . . . . .	20
sri.test . . . . .	21
summaryrshp . . . . .	21
translation . . . . .	22
zone.img . . . . .	23
zone.mat . . . . .	23
zones . . . . .	24
<b>Index</b>	<b>25</b>

---

multifluo-package	<i>Dealing with Several Images of a Same Object Constituted of Different Zones</i>
-------------------	--

---

## Description

Deals with several images of a same object, constituted of different zones. Each image constitutes a variable for a given pixel. The user can interactively select different zones of an image. Then, multivariate analysis (PCA) can be run in order to characterize the different selected zones, according to the different images. Hotelling (Hotelling, 1931, <doi:10.1214/aoms/1177732979>) and Srivastava (Srivastava, 2009, <doi:10.1016/j.jmva.2006.11.002>) tests can be run to detect multivariate differences between the zones.

## Author(s)

Caroline Peltier

Maintainer: Caroline Peltier <caroline.peltier@agrosupdijon.fr>

## See Also

[create.zones,reshaping,plotimage](#)

**Examples**

```
data(count)
count.img=plotimage(mat=count,lim=c(0,5000))
data(df.scaled)
resPCA=pca(data=df.scaled[,-1], zone="zone",pixel="pixel")
couleurs=find.colors(rownames(resPCA$IndivCoord))
plotpca(resPCA)
```

---

binning

*Bins the pixels*

---

**Description**

Reduces the size of an image by binning the pixels.

**Usage**

```
binning(dataset, n.bin, fun = "mean")
```

**Arguments**

dataset	The dataset to be binned (in a matrix form)
n.bin	number of pixels to be binned (a natural number)
fun	if "mean", the value of the binned pixel is the average of the values of the initial pixels if "sum", the value of the binned pixel is the sum of the values of the initial pixels

**Value**

The binned dataset

**Examples**

```
data(gp)
binning(gp,2)
```

---

calc.ell	<i>Calculating agreement ellipses</i>
----------	---------------------------------------

---

### Description

calculates the coordinates of agreement ellipses

### Usage

```
calc.ell(sup.ind,vep,axes=c(1,2),conf=0.9,ell.t="bar",trt="zone",sup="pixel",calc="Chi")
```

### Arguments

sup.ind	matrix of supplementary individuals
vep	eigenvectors
axes	selected axes (as a vector)
conf	confidence interval
ell.t	"barycentric" or "individual"
trt	name of trt
sup	name of the individuals around a same trt
calc	"Chi", "F" or "SAS"

---

count	<i>count</i>
-------	--------------

---

### Description

Spermatozoa bulls were labelled with a fluorescence probe (Laurdan) and then observed with a microscope and its related software Symphotime. Three data were recorded per pixels: the counts of photons (indicating the number of photons perceived on this pixel after simulation), the lifetime of fluorescence (indicating the time of first perceived photon after stimulation) and the generalized polarization (indicating a ratio between a green-light simulation and a blue-light simulation) These matrices have as many rows as observed pixels in lines, and as many columns as observed pixels in columns. This specific data represents the count of photons per pixel.

### Usage

```
data("count")
```

### Format

A data frame with 203 lines on the following 204 columns.

**See Also**[gp,lifetime](#)**Examples**`data(count)`

---

`create.zones`*Creating a list of consecutive zones of images*

---

**Description**

Creates a list of consecutive zones of images, interactively (by pointing directly control points on an image) or manually (by entering coordinates of control points). The control points can describe either a rectangle (two diagonal points), a parallelogram(three consecutive points representing an angle of the parallelogram) or a circle (center of the circle, then a point of the circle).

**Usage**`create.zones(img=NULL,shape="rect",n.zones=NULL,interact=TRUE,pts=NULL,name.zones=NULL)`**Arguments**

<code>img</code>	<code>img</code> is an image (cimg type) from imager stemming from "plot.image"
<code>shape</code>	When "rect", the shape would be a rectangle: the selection tool is a rectangle by clicking and maintaining the mouse on the screen. When "para", the shape would be a parallelogram: the selection tool is the consecutive input of three points of one angle of the parallelogram. When "circle", the selection tool is the consecutive input of the center of the circle and one point on the circle. When "various" and interactive, the user is asked for the shape on the next selection.
<code>n.zones</code>	number of zones to be input
<code>interact</code>	if TRUE, the interactive tool is launched to determine the control points of the different zones. If FALSE, these control points are defined in <code>pts</code>
<code>pts</code>	matrix of points corresponding to the selected zones
<code>name.zones</code>	if not NULL, corresponds to the names of the zones to select.

**Value**

A list containing as many component as the number of zones. Each component is also a list containing : the matrix of control points (`pts`),the name of the selected zone (`name`) and the name of the selected shape (`shape`) If "rect is chosen", `pts` contains the two points of the diagonal of the rectangle, If "para" is chosen, `pts` to the three points of the selected angle of the parallelogram If "circle" is chosen, `pts` contains the center of the circle and one point in the circle

**See Also**

[plotzone](#), [plotimage](#)

**Examples**

```
data(zone.mat)
zones=create.zones(pts=zone.mat,shape="para",interact=FALSE)
data(count)

img=plotimage(count,lim=c(1000,4000),m=count>1000)
zonesRect=create.zones(img,shape="rect",n.zones=3)
zonesPara=create.zones(img,shape="para",n.zones=3)
zonesCircle=create.zones(img,shape="circle",n.zones=3)
zonesVarious=create.zones(img,shape="various",n.zones=3)
```

---

df.scaled

*df.scaled*

---

**Description**

Example of data.frame required for the PCA (see [pca](#))

**Examples**

```
data(df.scaled)
## maybe str(df.scaled) ; plot(df.scaled) ...
```

---

dfboolTest

*dfboolTest*

---

**Description**

Reshaped dataset with the boolean option (return.bool=TRUE)

---

dfTest

*dfTest*

---

**Description**

reshaped dataframe with lifetime maximised at 8

---

difftable	<i>Table of multivariate differences</i>
-----------	--

---

### Description

Built a table crossing modalities of a trestmanet variable about multivariate differences

### Usage

```
difftable(mat,vep=NULL,axes=c(1,2),var.col=NULL,trt="zone",test="hotelling")
```

### Arguments

mat	the matrix must have one column named as trt, and
vep	when vep is not NULL, a preliminary operation multiplies mat [,var.col] by the vep matrix (useful to do the test only on the first principal components of mat).
axes	vep can contain all eigenvectors. If it is the case, axes allows the user to select only some eigenvectors (1:2 for the first two eigenvectors, 1:5 for the first five eigenvectors...)
var.col	indicates the number of columns of the numerical variables. If NULL, all columns but trt are selected.
trt	name of the column
test	"hotelling" or "sri". Indicates the type of test to detect multivariate differences. "hotelling" is the usual hotelling T2 test whereas "sri" corresponds to the Srivastava's Test, allowing a test of differences even if there is more variables than observations.

### Value

A table containing the p-values is returned.

### See Also

[hotelling.test](#), [sri.test](#)

### Examples

```
data(df.scaled)
difftable(df.scaled[,-5],trt="zone")
```

---

erase

*Eraser*

---

### Description

Suppresses a zone (determined on an image) from a given selection stemming from a zone image

### Usage

```
erase(zone.img, img, interact = TRUE, shape = "rect", pts = NULL)
```

### Arguments

zone.img	the zone image to be modified
img	the image on which the zone to suppress is selected
interact	if TRUE, the zone is determined interactively. Else pts is used as a matrix of control points (see <a href="#">create.zones</a> )
shape	"rect", "para" or "circle"
pts	Matrix containing the control points (see <a href="#">create.zones</a> )

### Value

returned a zone image without the selected zone

### See Also

[img.zone.plotimage](#)

### Examples

```
data(zone.img)
plotimage(zone.img, z=TRUE)
data(img)
zone.img2=erase(zone.img, img, interact=TRUE, shape="rect")
plotimage(zone.img2, z=TRUE)
```



---

find.colors	<i>find.colors</i>
-------------	--------------------

---

**Description**

Finds different colors for a corresponding to the first character of each value of a vector.

**Usage**

```
find.colors(vect, color = NULL)
```

**Arguments**

vect	the vector to find colors for.
color	can be a vector of colors (with the same size as the number of groups), if NULL, rainbow scale is chosen

**Value**

vector of colors related to vect

**Examples**

```
find.colors(c("V1", "V3", "R22"))
```

---

generic.plot	<i>generic.plot</i>
--------------	---------------------

---

**Description**

Plots

**Usage**

```
generic.plot(type = "R", width = 7, height = 7, name = "output", CALLFUN = NULL)
```

**Arguments**

type	"R", "png" ...
width	width of the file
height	height of the file
name	name of the file
CALLFUN	function to plot

gp    gp

---

### Description

Spermatozoa bulls were labelled with a fluorescence probe (Laurdan) and then observed with a microscope and its related software Symphotime. Three data were recorded per pixels: the counts of photons (indicating the number of photons perceived on this pixel after simulation), the lifetime of fluorescence (indicating the time of first perceived photon after stimulation) and the generalized polarization (indicating a ratio between a green-light simulation and a blue-light simulation) These matrices have as many rows as observed pixels in lines, and as many columns as observed pixels in columns. This specific data represents the Generalized polarization data

### Usage

```
data("gp")
```

### See Also

[count](#), [lifetime](#)

### Examples

```
data(gp)
```

---

hotelling.test                          *Hotelling's T2 Test*

---

### Description

Tests multivariate differences between two samples with Hotelling's T2 Test

### Usage

```
hotelling.test(matx, maty)
```

### Arguments

matx                          First sample (matrix with observations as lines and variables as columns)  
maty                          Second sample (matrix with observations as lines and variables as columns)

### Value

Returns a list containing the stat and the p-value

**References**

Hotelling, H. (1931). "The generalization of Student's ratio." *Annals of Mathematical Statistics* 2. (3): 360-378

**See Also**

[difftable](#), [sri.test](#)

**Examples**

```
A=matrix(rnorm(13.15),13,15)
B=matrix(rnorm(13.15),13,15)
hotelling.test(A,B)
```

---




---



---




---

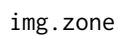
**Description**

An example of image based on the data(count)

**Examples**

```
data(img)
```

---




---



---

*Converts a zone object (point list) into a matrix*

---

**Description**

Converts a zone object (point list) into a matrix : each pixel is attributed to a zone. This matrix can also be plotted with [plotimage](#). This matrix is to be related to the other image matrices, for example in the [reshaping](#) function, to be used in [pca](#).

**Usage**

```
img.zone(zone,d=NULL,mask=NULL,edge.only=NULL,wo.edge=NULL,graph=FALSE,img=NULL,lim=30)
```

**Arguments**

zone	A zone created with <a href="#">create.zones</a>
d	dimension of the matrix to create
mask	boolean matrice can be specified here
edge.only	if TRUE, only the edges of the images are kept. They are defined with "limit" for their width. This requires that the parameter img is fulfilled
wo.edge	if TRUE, the edges of the images are removed. They are defined with "limit" for their width. This requires that the parameter img is fulfilled
img	image on which the edges are selected
graph	if TRUE, the selected edges are plotted
lim	sensitivity of the scale detection

**Value**

Zone object (with control points) is given as a matrix /image

**See Also**

[create.zones](#), [plotzone](#), [reshapimg](#), [erase](#), [plotimage](#)

**Examples**

```
data(zones)
data(count)
zone.imgRect=img.zone(zone=zones,d=dim(count))
plotimage(mat=zone.imgRect,z=T)
```

---

*interieur.para*

*interieur.para*

---

**Description**

Detects if a point is inside a given parallelogram

**Usage**

```
interieur.para(x, y, xOrigine, yOrigine, xA, yA, xB = NA, yB = NA, shape = "para")
```

**Arguments**

x	abscisse of the point to test
y	ordinate of the point to test
xOrigine	abscisse of a summit of the parallelogram (origin)
yOrigine	ordinate of a summit of the parallelogram (origin)
xA	abscisse of a summit of the parallelogram linked to the origine
yA	ordinate of a summit of the parallelogram linked to the origine
xB	abscisse of the other summit of the parallelogram linked to the origine
yB	ordinate of the other summit of the parallelogram linked to the origine
shape	"para"

---

intpca

*internal pca*


---

**Description**

runs pca

**Usage**

```
intpca(vars, ind, map="b", si=NULL, ic=NULL, sic=NULL, ell=NULL, seg=NULL, epd=NULL, x="", y="")
```

**Arguments**

vars	variables
ind	individuals
map	if "bil", biplot is plotted with supplementary individuals projected, and labels of supplementary individuals are written if "bip", biplot is plotted with supplementary individuals projected, and points of supplementary individuals are plotted if "b", biplot is plotted if "til", two-way pca is plotted with supplementary individuals projected, and labels of supplementary individuals are written if "tip", two-way pca is plotted with supplementary individuals projected, and la points of supplementary individuals are plotted if "t", two-way pca is plotted
si	supplementary individuals
ic	color of individuals
sic	color of supplementary individuals
ell	"bar" or "ind" or "none"
seg	TRUE or FALSE
epd	expand
x	x label
y	y label

---

`lifetime`*lifetime*

---

**Description**

Spermatozoa bulls were labelled with a fluorescence probe (Laurdan) and then observed with a microscope and its related software Symphotime. Three data were recorded per pixels: the counts of photons (indicating the number of photons perceived on this pixel after simulation), the lifetime of fluorescence (indicating the time of first perceived photon after stimulation) and the generalized polarization (indicating a ratio between a green-light simulation and a blue-light simulation) These matrices have as many rows as observed pixels in lines, and as many columns as observed pixels in columns. This specific data represents the lifetime data

**Usage**

```
data("lifetime")
```

**See Also**

[gp,count](#)

**Examples**

```
data(lifetime)
```

---

`newgp`*Translated Generalized Polarization Data*

---

**Description**

This matrix represents the translated gp data

**See Also**

[gp](#)

**Examples**

```
data(newgp)
```

---

pca *pca*

---

### Description

Calculates Principal Component Analysis with agreement ellipses

### Usage

```
pca(data, zone = "zone", pixel = NULL)
```

### Arguments

data	should contain a column named as zone, and another named pixel
zone	name of the column containing the zone
pixel	name of the column containing the pixel name (required when ellipses, individual projections or tests are asked in the PCA)

### Value

A list containing	
B	matrix of zone covariance
IndSup	supplementary individuals
EingenVectors	eigen vectors obtained by the PCA
EingenValues	eigen values obtained by the PCA
IndivCoord	coordinates of the individuals (here, zone means)
VarCoord	coordinates of the variables
NbdimSig	number of significant dimensions

### References

Peltier, C., Visalli, M. and Schlich, P. (2015), Canonical Variate Analysis of Sensory Profiling Data. *J Sens Stud*, 30: 316 328. doi:10.1111/joss.12160

### See Also

[plotpca reshaping](#)

### Examples

```
data(df.scaled)
resPCA=pca(data=df.scaled[, -1], zone="zone", pixel="pixel")
```

---

pcapic

*Pictures from the first component of PCA*

---

### Description

This function returns a picture where each pixel value corresponds to its coordinate on the first component of PCA (not scaled)

### Usage

```
pcapic(dfbool, d = c(203, 204), lim = NULL)
```

### Arguments

dfbool	result from <a href="#">reshaping</a> , with return.bool=TRUE
d	d is the final size of the image. It has to correspond to the initial images taken to run the <a href="#">reshaping</a>
lim	limit of the colorscale

### Value

Returns the resulted matrix with d dimensions, and plots the image with pca.

### See Also

[reshaping](#)

### Examples

```
data(dfboolTest)
pcapic(dfbool=dfboolTest,lim=c(-1,1))
```

---

plotimage

*Plotting an image*

---

### Description

Plots an image with its colorscale.

### Usage

```
plotimage(mat,lim=NULL,nc=1000,m=NULL,z=FALSE,add=FALSE,cols="rainbow",lc=c(0,0.7),p="l")
```



**Arguments**

mat	matrix representing the image. Thus, a numerous matrix with 256 rows and 256 columns will represent an image with 256*256 pixels. The colour of each pixel will be determined thanks to the value of the related pixel in the matrix, and the values of the limit scales.
lim	minimum and maximum value for the color scale. Ex: lim=c(0,10)
nc	number of colors
m	boolean matrix with the same size than mat, selecting pixels which will plotted (TRUE) or not (FALSE) (mask)
z	TRUE when mat is a zone matrix stemming from <a href="#">img.zone</a>
add	when TRUE, the graph is drawn over the existing graph. If FALSE, a new graph is plotted.
cols	"rainbow" by default, but other scales are also available : "gray.scale", "cm.colors", "topo.colors", "terrainc" These scales correspond to the usual colors scales in R.
lc	c(0,0.7) by default. Parameter corresponding to the start color and the end color in the rainbow scale,a number in [0,1] (start in rainbow function)
p	Further parameters to plot. if "n", nothing more is plotted, if "a", axes are plotted, if "l", color legend is plotted, if "al" axes and color legend are plotted. If "d" is added to this parameter (giving "ad", "ld", "ald" or "nd", no smoothing of the pixel is computed (discrete)

**Value**

Returns the created image as a cimg (type cimg from the imager package)

**Examples**

```
data(lifetime)
data(count)
data(gp)
count.img=plotimage(mat=count,lim=c(0,5000))

# with m
count.img=plotimage(mat=count,lim=c(1000,4000),m=count>1000)
plotimage(mat=lifetime,lim=c(2.8,3.5))
gp.img=plotimage(mat=gp,lim=c(-0.2,0.8))

#add

resT=plotimage(mat=lifetime,lim=c(2.8,3.5),m=count>1000)
resGP=plotimage(mat=gp,lim=c(2.8,3.5),add=TRUE)
```

---

 plotpca

*Plotting a Principal Component Analysis*


---

## Description

Plots the Principal Component Analysis from the `pca` function, and customizes the output with

## Usage

```
plotpca(x, map="b", n="a", l=0.05, col=NULL, calc="Chi", epd=NULL, et="bar", dif="n", dax="all")
```

## Arguments

<code>x</code>	results of <code>pca</code>
<code>map</code>	if "bil", biplot is plotted with supplementary individuals projected, and labels of supplementary individuals are written if "bip", biplot is plotted with supplementary individuals projected, and points of supplementary individuals are plotted if "b", biplot is plotted if "til", two-way pca is plotted with supplementary individuals projected, and labels of supplementary individuals are written if "tip", two-way pca is plotted with supplementary individuals projected, and la points of supplementary individuals are plotted if "t", two-way pca is plotted
<code>n</code>	number of axes to plot. By default, "a", which calculates an automatic theoretical number of dimensions (when the contribution of loadings is higher than 1)
<code>l</code>	by default 0.05. Risk to be taken in the agreement ellipses
<code>et</code>	"bar" for the confidence ellipses around the product means "ind" for the representation of a proportion of 1-alpha of the panelist scores
<code>col</code>	select the colors of the different individuals
<code>calc</code>	"Chi", "F" or "Sas". Indicates the type of ellipse to be calculated.
<code>epd</code>	when a biplot is chosen, the arrows can be oversized.
<code>dif</code>	if "n", no tests are run if "h", hotelling test is run to detect differences between zones. if "hc", the p-value is corrected with the bonferroni Correction ( $l$ is divided by $k(k+1)$ where $k$ is the number of zones) if "s", srivastava test is run to detect differences between zones. if "sc", the p-value is corrected with the bonferroni Correction ( $l$ is divided by $k(k+1)$ where $k$ is the number of zones) When a test is run, zones not significantly difference are related by a segment
<code>dax</code>	if "all", the test is run on all the Principal Components. If "sig", it is done only on the significant Principal Components. <code>dax</code> can also be a vector determining the index of Principal Component to be chosen for the test (ex <code>dax=c(1,2)</code> ).

**Examples**

```
data(df.scaled)
resPCA=pca(data=df.scaled[,-1], zone="zone",pixel="pixel")
couleurs=find.colors(rownames(resPCA$IndivCoord))
plotpca(x=resPCA)
```

---

plotzone

*Plotting the different selected zones on an image*

---

**Description**

plots the different selected zones on an image

**Usage**

```
plotzone(zone, name = FALSE, dim.img=NULL, d=200, cex=1, col="black", main="Zone", lwd=1)
```

**Arguments**

zone	zone stemming from <a href="#">create.zones</a>
name	if TRUE, the zone names are written above the zones
dim.img	dimensions of the image where the zones are drawn
d	the distance between the zone and the text is given in a portion of the image (divided by 200 by default)
cex	size of the text (by.default 1)
col	color of the zones and the text
main	main Title
lwd	width of the lines

**See Also**

[create.zones](#); [img.zone](#)

**Examples**

```
data(zones)
plotzone(zones)
```

---

reshaping	<i>Reshape image matrices as one sigle dataframe (useful to pca, for example)</i>
-----------	---

---

### Description

Reshapes a list of (numeric) image matrices, and a zone matrix related to one set of images, in order to obtain a dataframe, with each line corresponding to one pixel and each matrix corresponding to one single column.

### Usage

```
reshaping(zone.img,list.img,list.bool=NULL,reduction=TRUE,return.bool=FALSE,name.img=NULL)
```

### Arguments

zone.img	result of <a href="#">img.zone</a>
list.img	list of images to be compared
list.bool	potential list of boolean matrices reflecting conditions of the analysis. Each pixel which does not respect at least one of the boolean conditions will be removed from the resulting dataset.
reduction	if TRUE, each column of the related datafraime (corresponding to each numeric image matrix) is centered and reduced,else, it is only centered
return.bool	if TRUE, a list containing the resulting data (data) and the boolean matrix of the treated pixels (boolean) is returned
name.img	vector containing the names of the successive images of list.img which are going to be the names of the column of the resulting dataset)

### Value

The resulting data contains one column containing the zone, and then as many columns as components in list.img. The names of columns can correspond to the name.img vector (default), or to the names of the matrices in list.mat, else it will be Var1, Var2,...

### See Also

[img.zone](#),[pca](#),[create.zones](#),[summaryrshp](#)

### Examples

```
data(count)
data(newgp)
data(lifetime)
data(zone.img)
```

```
df37A=reshaping(zone.img,list(count,lifetime,newgp),name.img=c("Count","Lifetime","GP"))
```

---

sri.test	<i>Srivastava's Test</i>
----------	--------------------------

---

**Description**

Tests multivariate differences between two samples with Srivastava's Test

**Usage**

```
sri.test(matx, maty)
```

**Arguments**

matx	First sample (with observations as lines and variables as columns)
maty	Second sample (with observations as lines and variables as columns). matx and maty have to have the same number of variables.

**Value**

a list containing the p-value and the statistique (corresponding to T1 in the Srivastava's paper)

**References**

M. S. Srivastava and M. Du. A test for the mean vector with fewer observations than the dimension. *Journal of Multivariate Analysis* , 99(3):386 402, Mar. 2008.

**See Also**

[difftable](#), [hotelling.test](#)

**Examples**

```
A=matrix(rnorm(13.15),13,15)
B=matrix(rnorm(13.15),13,15)
sri.test(A,B)
```

---

summaryrshp	<i>Summarizing reshaped data</i>
-------------	----------------------------------

---

**Description**

Gives natural statistics (number of observations, average, standard deviations) for each variable and each zones. ANOVA of the model "variable~zone" is run, and difference tests are computed to find groups.

**Usage**

```
summaryrshp(rshp, test = "hsd", p.adj = "none", alpha = 0.05)
```

**Arguments**

rshp	data resulting from <a href="#">reshaping</a>
test	"lsd" (corresponding to Least Significant Difference) or "hsd" (corresponding to the Tukey Test)
p.adj	LSD.test parameter to adjust the p-value for the multiple comparison
alpha	Threshold of significance for determining the gorups

**Value**

Returns a list containing, for each variable: the results of the ANOVA; the summary of the variable; and the usual statistics by zones (n,average, sd,group)

**See Also**

[reshaping](#)

**Examples**

```
data(df.scaled)
summaryrshp(df.scaled)
```

---

translation

*Translation of the data*

---

**Description**

Sometimes, a slight shift exists between two images of the same sample. This function allows them to be superimposed. The shift can be determined with a vector by entering its coordinates; or interactively by clicking on a specific point on a first image, and its corresponding point on the second image)

**Usage**

```
translation(dtt,dtc=NULL,x,y,interact="none",dtt.lim=NULL,dtc.lim=NULL,n.around=c(5,5))
```

**Arguments**

dtt	image matrix (for 512*512 image, the dimensions of this matrix are 512*512) to translate
dtc	potential image matrix (for 512*512 image, the dimensions of this matrix are 512*512) on which the dtt is superimposed
x	If interactive is FALSE, abscisse of vector of translation

y	If interactive is FALSE, ordinate of vector of translation
interact	if TRUE, dtt is displayed as an image, and the user has to select one point on it. Then, dtc is displayed as an image and the user has to select the related same point on it.
dtt.lim	limits of the scale of dtt (to change when the image is not contrasted enough to detect specific points)
dtc.lim	limits of the scale of dtc (to change when the image is not contrasted enough to detect specific points)
n.around	Tests every translation in a neighborhood of n.around[1] in x and n.around[2] of y (clicked by the user), and select the translation that maximizes the number of not TRUE common to dtc and dtt. It requires that both dtc and dtt are boolean matrices.

**Value**

The function returns a list containing the translated data ("data") and the vector of translation ("translation")

**Examples**

```
data(gp)
data(count)
GP37Anew=translation(dtt=gp,dtc=count,interact=TRUE,dtt.lim=c(-0.4,0.8),dtc.lim=c(1000,4000))
GP37Anew=translation(gp,x=7,y=-3)$data
```

---

zone.img	<i>Zone image</i>
----------	-------------------

---

**Description**

Example of image zone stemming from [zones](#)

**Examples**

```
data(zone.img)
```

---

zone.mat	<i>Zones as matrix</i>
----------	------------------------

---

**Description**

Example of zones as a matrix

**Examples**

```
data(zone.mat)
```

---

zones

*Zone example*

---

**Description**

Example of control points for zones

**Usage**

```
data("zones")
```

**Examples**

```
data(zones)
```



# Index

- \*Topic **\textasciitildekwd1**
  - pca, [15](#)
- \*Topic **\textasciitildekwd2**
  - pca, [15](#)
- \*Topic **datasets**
  - count, [4](#)
  - df.scaled, [6](#)
  - gp, [10](#)
  - lifetime, [14](#)
  - zone.mat, [23](#)
- \*Topic **package**
  - multifluo-package, [2](#)
- binning, [3](#)
- calc.ell, [4](#)
- count, [4](#), [10](#), [14](#)
- create.zones, [2](#), [5](#), [8](#), [12](#), [19](#), [20](#)
- df.scaled, [6](#)
- dfboolTest, [6](#)
- dfTest, [6](#)
- diffTable, [7](#), [11](#), [21](#)
- erase, [8](#), [12](#)
- find.colors, [9](#)
- generic.plot, [9](#)
- gp, [5](#), [10](#), [14](#)
- hotelling.test, [7](#), [10](#), [21](#)
- img, [11](#)
- img.zone, [8](#), [11](#), [17](#), [19](#), [20](#)
- interieur.para, [12](#)
- intpca, [13](#)
- lifetime, [5](#), [10](#), [14](#)
- multifluo (multifluo-package), [2](#)
- multifluo-package, [2](#)
- newgp, [14](#)
- pca, [6](#), [11](#), [15](#), [18](#), [20](#)
- pcapic, [16](#)
- plotimage, [2](#), [6](#), [8](#), [11](#), [12](#), [16](#)
- plotpca, [15](#), [18](#)
- plotzone, [6](#), [12](#), [19](#)
- reshaping, [2](#), [11](#), [12](#), [15](#), [16](#), [20](#), [22](#)
- sri.test, [7](#), [11](#), [21](#)
- summaryrshp, [20](#), [21](#)
- translation, [22](#)
- zone.img, [23](#)
- zone.mat, [23](#)
- zones, [23](#), [24](#)