

MORFOMETRIA

Leonardo Carvalho

2023-10-08

ANALISE DE MORFOMETRIA EM ESTRUTURAS BIOLÓGICAS

Para a instalação dos Pacotes Necessários, sigam os comandos:

```
install.packages("fftwtools")  install.packages("BiocManager")  install.packages("pliman")  BiocManager::install("EBImage")
```

Caregando os pacotes:

```
library(pliman)

## The legacy packages maptools, rgdal, and rgeos, underpinning the sp package,
## which was just loaded, will retire in October 2023.
## Please refer to R-spatial evolution reports for details, especially
## https://r-spatial.org/r/2023/05/15/evolution4.html.
## It may be desirable to make the sf package available;
## package maintainers should consider adding sf to Suggests:.
## The sp package is now running under evolution status 2
##       (status 2 uses the sf package in place of rgdal)

## | ======
```

```
## | Tools for Plant Image Analysis (pliman 2.0.1)      |
## | Author: Tiago Olivoto                               |
## | Type `citation('pliman')` to know how to cite pliman   |
## | Visit 'http://bit.ly/pkg\_pliman' for a complete tutorial |
## | ======
```

```
library(EBImage)

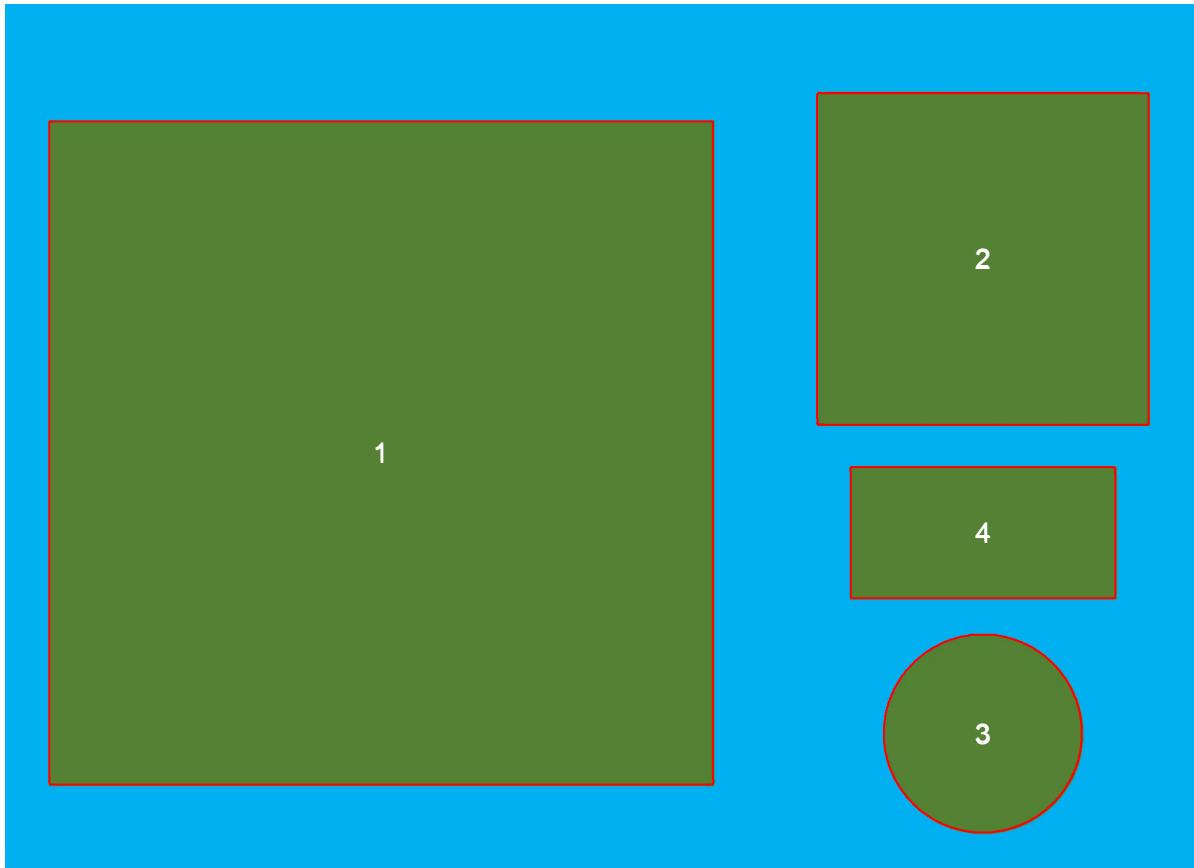
##
## Attaching package: 'EBImage'
## The following object is masked from 'package:pliman':
##   otsu
```

Exemplo ilustrativo inicial

```
img <- image_pliman("objects_300dpi.jpg")
plot(img)

results <- analyze_objects(img, marker = "id")
```

```
plot_measures(results, measure = "id")
```



```
get_measures(results,
             id = 1,
             area ~ 100)
```

```
## -----
## measures corrected with:
## object id: 1
## area      : 100
## -----
## Total      : 39.984
## Average    : 13.328
## -----
##   id      x      y      area  area_ch perimeter radius_mean radius_min
## 2 2 1736.509 452.2488 25.04020 24.97763 21.93099 2.86502 2.49467
## 3 3 1736.574 1295.3292 7.02212 7.01414 9.90335 1.49122 1.48384
## 4 4 1736.974 938.5020 7.92176 7.87974 11.90601 1.66580 0.98630
##   radius_max radius_sd diam_mean diam_min diam_max major_axis minor_axis
## 2     3.52427  0.31334  5.73003  4.98934  7.04853  2.03859  2.03731
## 3     1.50042  0.00297  2.98244  2.96769  3.00085  1.05505  1.05386
## 4     2.22199  0.42412  3.33159  1.97260  4.44398  1.48163  0.87143
##   caliper length  width radius_ratio theta eccentricity form_factor
## 2 7.02024 5.14072 5.13225       1.41272 1.54244      0.03540  0.65423
```

```

## 3 2.99261 2.98969 2.98711      1.01117 -0.73169      0.04757      0.89973
## 4 4.43256 3.99330 1.97558      2.25285  0.00004      0.80875      0.70226
##   narrow_factor asp_ratio rectangularity pd_ratio plw_ratio solidity convexity
## 2      1.36561  1.00165      1.05364  3.12397      2.13482  1.00251      0.68261
## 3      1.00098  1.00086      1.27177  3.30926      1.65697  1.00114      0.92114
## 4      1.11000  2.02133      0.99588  2.68604      1.99468  1.00533      0.83651
##   elongation circularity circularity_haralick circularity_norm coverage
## 2     0.00165  19.20784      9.14347      0.65202      0.10675
## 3     0.00086  13.96678      502.58210      0.89510      0.02994
## 4     0.50528  17.89416      3.92766      0.69779      0.03377

```

Exemplo 1

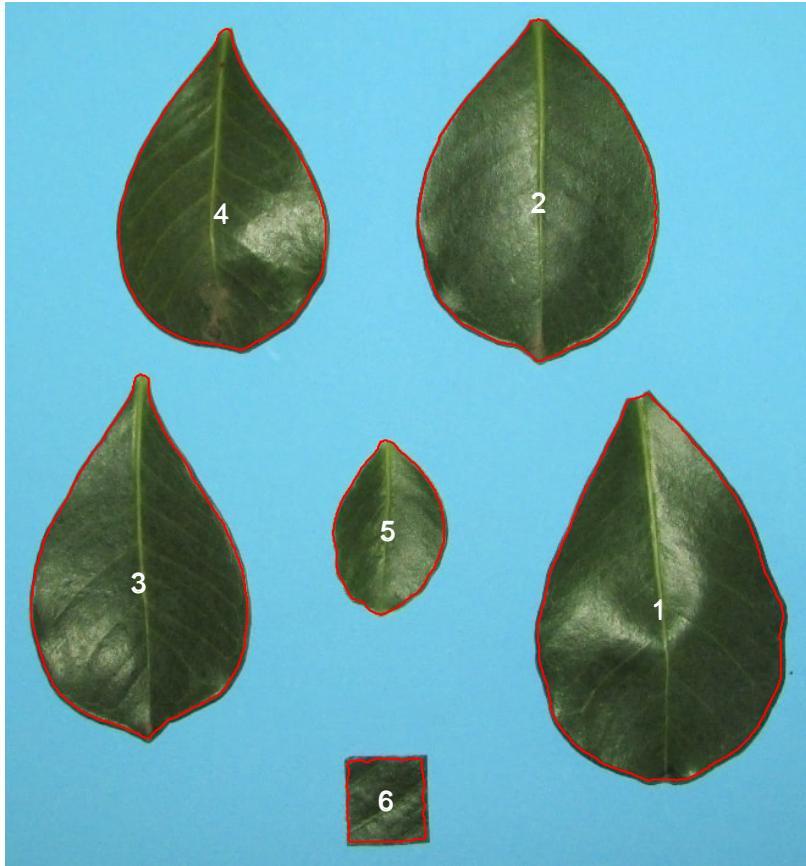
```

folhas <- image_pliman("la_leaves.jpg")
plot(folhas)

resultado <- analyze_objects(folhas, marker = "id")

plot_measures(resultado, measure = "id")

```



```

area <- get_measures(resultado,
                     id = 6,
                     area ~ 4)

```

```

## -----
## measures corrected with:
## object id: 6

```

```

## area      : 4
## -----
## Total     : 145.634
## Average   : 29.127
## -----
## -----
str(area)

## Classes 'measures' and 'data.frame': 5 obs. of  34 variables:
## $ id          : num  1 2 3 4 5
## $ x           : num  536 438 110 177 314
## $ y           : num  498 164 476 173 434
## $ area        : num  41.79 35.81 31.65 27.77 8.61
## $ area_ch     : num  41.97 35.95 32.53 28.16 8.62
## $ perimeter   : num  26.2 23.5 23.9 21.6 11.6
## $ radius_mean : num  3.7 3.39 3.29 3.04 1.66
## $ radius_min  : num  2.77 2.9 2.39 2.32 1.32
## $ radius_max  : num  5.28 4.59 4.9 4.41 2.28
## $ radius_sd   : num  0.661 0.419 0.656 0.524 0.265
## $ diam_mean   : num  7.4 6.78 6.58 6.09 3.33
## $ diam_min    : num  5.53 5.79 4.78 4.63 2.63
## $ diam_max    : num  10.56 9.18 9.81 8.82 4.55
## $ major_axis   : num  3.07 2.67 2.79 2.51 1.36
## $ minor_axis   : num  2.173 2.134 1.861 1.796 0.995
## $ caliper     : num  9.71 8.54 9.13 8.06 4.38
## $ length      : num  9.73 8.57 9.13 8.06 4.37
## $ width       : num  6.19 5.98 5.43 5.18 2.8
## $ radius_ratio: num  1.91 1.59 2.05 1.91 1.73
## $ theta        : num  1.52 -1.53 1.56 1.56 -1.46
## $ eccentricity: num  0.705 0.601 0.745 0.699 0.682
## $ form_factor  : num  0.767 0.813 0.699 0.749 0.803
## $ narrow_factor: num  0.998 0.997 1 1 1.003
## $ asp_ratio    : num  1.57 1.43 1.68 1.55 1.56
## $ rectangularity: num  1.44 1.43 1.56 1.5 1.42
## $ pd_ratio     : num  2.69 2.75 2.61 2.68 2.65
## $ plw_ratio    : num  1.64 1.62 1.64 1.63 1.62
## $ solidity     : num  0.996 0.996 0.973 0.986 0.999
## $ convexity    : num  0.936 0.933 0.912 0.91 0.898
## $ elongation   : num  0.364 0.302 0.405 0.357 0.358
## $ circularity  : num  16.4 15.5 18 16.8 15.6
## $ circularity_haralick: num  5.59 8.1 5.01 5.81 6.29
## $ circularity_norm   : num  0.761 0.806 0.692 0.742 0.789
## $ coverage      : num  0.0954 0.0818 0.0723 0.0634 0.0197

plot(folhas)
plot_measures(area, measure = "diam_max")

```

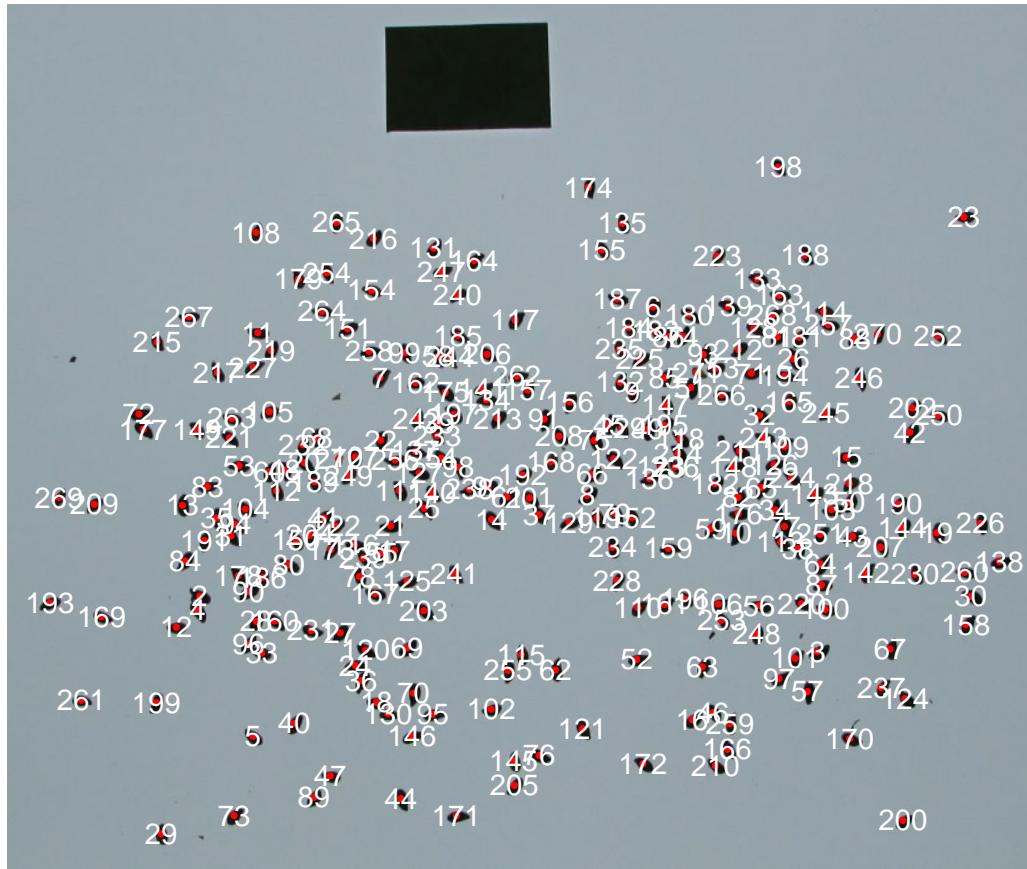


Exemplo 2

```
graos <- image_pliman("flax_grains.jpg")
plot(graos)

flax <- analyze_objects(graos,
                        index = "GRAY",
                        reference = TRUE,
                        reference_larger = TRUE,
                        reference_area = 6,
                        marker = "point",
                        marker_size = 0.5,
                        marker_col = "red", # padrão é branco
                        show_contour = FALSE) # Padrão é TRUE

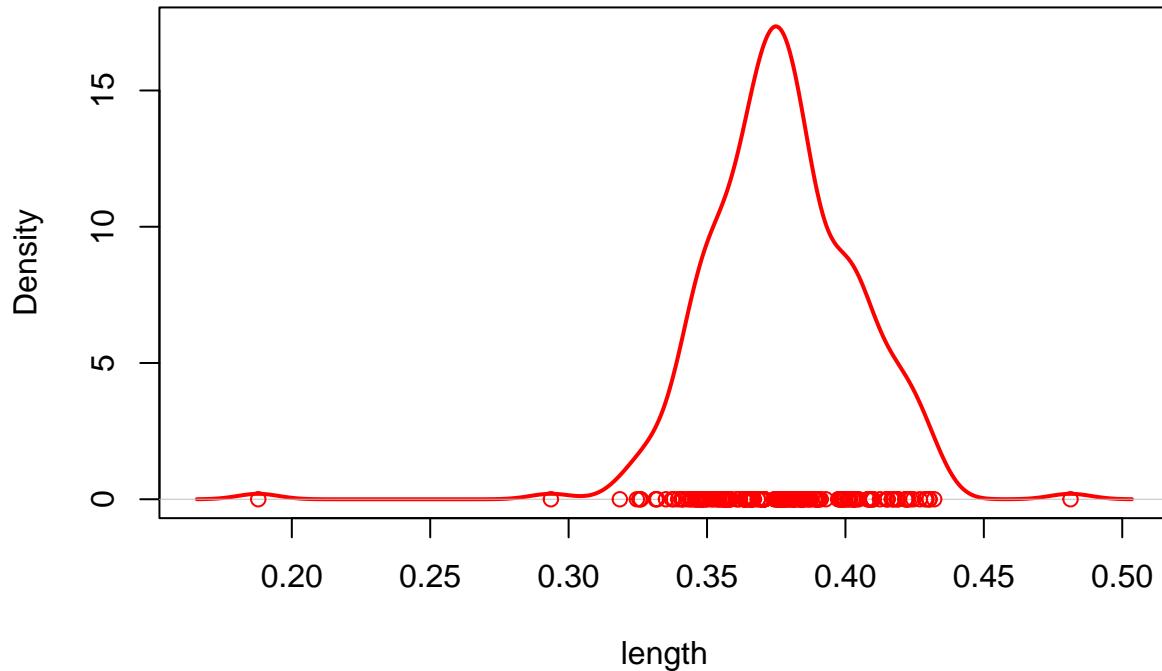
plot_measures(flax)
```



```
flax$statistics
```

```
##      stat      value
## 1      n 2.710000e+02
## 2  min_area 1.868474e-02
## 3 mean_area 6.169980e-02
## 4 max_area 7.968492e-02
## 5   sd_area 6.534106e-03
## 6 sum_area 1.672064e+01
## 7 coverage 5.396059e-02
```

```
plot(flax, measure = "length")
```

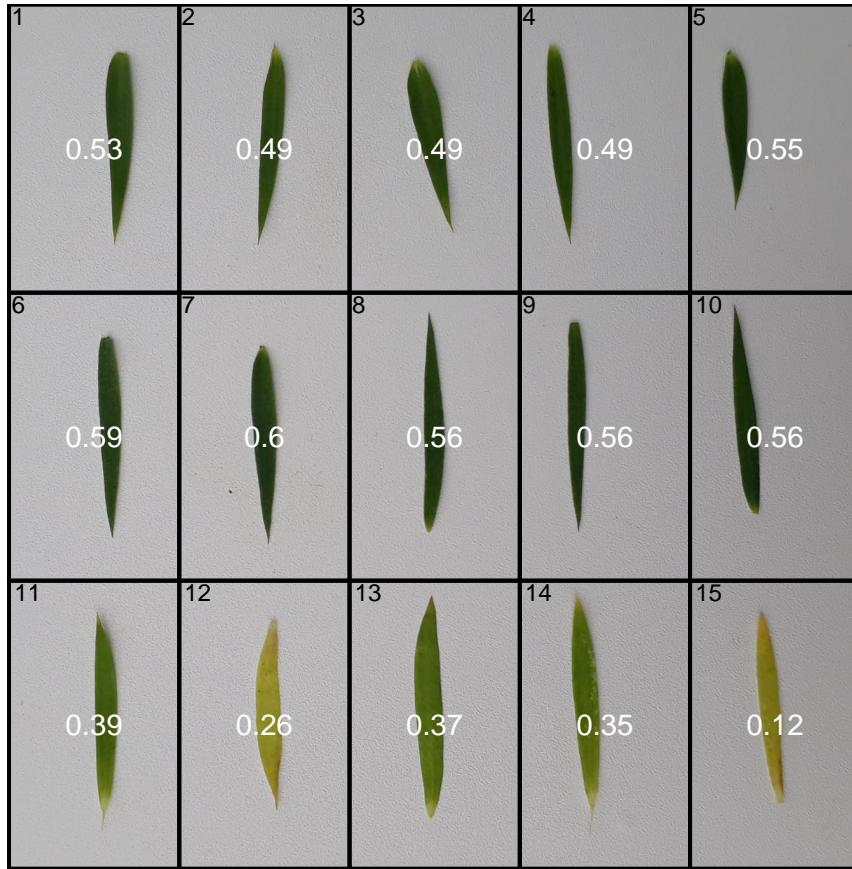


```
nrow(flax)
## NULL
```

Exemplo 3

```
flax <- image_pliman("flax_leaves.jpg")
plot(flax)

res <- analyze_objects_shp(flax,
                           nrow = 3,
                           ncol = 5,
                           plot = FALSE,
                           object_index = "DGCI")
plot(flax)
plot(res$shapefiles)
plot_measures(res, measure = "DGCI")
```



```
str(res)
```

```
## List of 16
## $ results
##   ..$ img
##   ..$ id
##   ..$ x
##   ..$ y
##   ..$ area
##   ..$ area_ch
##   ..$ perimeter
##   ..$ radius_mean
##   ..$ radius_min
##   ..$ radius_max
##   ..$ radius_sd
##   ..$ diam_mean
##   ..$ diam_min
##   ..$ diam_max
##   ..$ major_axis
##   ..$ minor_axis
##   ..$ caliper
##   ..$ length
##   ..$ width
##   ..$ radius_ratio
##   ..$ theta
##   ..$ eccentricity
: 'data.frame': 50 obs. of 35 variables:
 : chr [1:50] "shp1" "shp2" "shp2" "shp3" ...
 : num [1:50] 1 1 2 1 1 2 3 4 1 1 ...
 : num [1:50] 174 419 410 663 869 ...
 : num [1:50] 197 135 251 197 200 ...
 : num [1:50] 9067 3034 4561 8246 1794 ...
 : num [1:50] 9159 3152 4756 8392 1932 ...
 : num [1:50] 641 283 484 623 193 ...
 : num [1:50] 79 35.6 57.5 74.6 24.8 ...
 : num [1:50] 16 15.9 12.4 18.9 16.4 ...
 : num [1:50] 151.6 59.2 112.3 143.7 40.1 ...
 : num [1:50] 40.57 13.22 28.52 36.75 6.02 ...
 : num [1:50] 158.1 71.1 115.1 149.1 49.6 ...
 : num [1:50] 32 31.8 24.8 37.8 32.7 ...
 : num [1:50] 303.2 118.4 224.6 287.4 80.3 ...
 : num [1:50] 87.4 35.5 63.1 81.6 22.1 ...
 : num [1:50] 16 13.3 12 15.9 12.8 ...
 : num [1:50] 284 112 193.4 264.4 76.2 ...
 : num [1:50] 297.1 117.9 205.4 271.6 76.2 ...
 : num [1:50] 45.5 35.8 36.1 48.1 34.2 ...
 : num [1:50] 9.47 3.72 9.05 7.61 2.45 ...
 : num [1:50] -1.55 -1.52 -1.45 1.36 1.51 ...
 : num [1:50] 0.983 0.927 0.982 0.981 0.817 ...
```

```

##   ..$ form_factor      : num [1:50] 0.277 0.476 0.245 0.267 0.603 ...
##   ..$ narrow_factor    : num [1:50] 0.956 0.95 0.942 0.973 1 ...
##   ..$ asp_ratio         : num [1:50] 6.54 3.29 5.69 5.65 2.23 ...
##   ..$ rectangularity   : num [1:50] 1.49 1.39 1.63 1.58 1.45 ...
##   ..$ pd_ratio          : num [1:50] 2.26 2.53 2.5 2.36 2.54 ...
##   ..$ plw_ratio         : num [1:50] 1.87 1.84 2 1.95 1.75 ...
##   ..$ solidity           : num [1:50] 0.99 0.963 0.959 0.983 0.928 ...
##   ..$ convexity          : num [1:50] 0.894 0.826 0.792 0.838 0.801 ...
##   ..$ elongation         : num [1:50] 0.847 0.696 0.824 0.823 0.551 ...
##   ..$ circularity        : num [1:50] 45.4 26.4 51.3 47.1 20.8 ...
##   ..$ circularity_haralick: num [1:50] 1.95 2.69 2.02 2.03 4.12 ...
##   ..$ circularity_norm   : num [1:50] 0.268 0.455 0.233 0.258 0.574 ...
##   ..$ coverage            : num [1:50] 0.0747 0.0249 0.0374 0.0677 0.0147 ...
## $ statistics           :'data.frame': 105 obs. of 3 variables:
##   ..$ img : chr [1:105] "shp1" "shp1" "shp1" "shp1" ...
##   ..$ stat : chr [1:105] "n" "min_area" "mean_area" "max_area" ...
##   ..$ value: num [1:105] 1 9067 9067 9067 NA ...
## $ object_rgb            : NULL
## $ object_index           :'data.frame': 50 obs. of 3 variables:
##   ..$ img : chr [1:50] "shp1" "shp2" "shp2" "shp3" ...
##   ..$ id  : num [1:50] 1 1 2 1 1 2 3 4 1 1 ...
##   ..$ DGCI: num [1:50] 0.531 0.468 0.516 0.49 0.499 ...
## $ efourier               : NULL
## $ efourier_norm          : NULL
## $ efourier_error         : NULL
## $ efourier_power         : NULL
## $ efourier_minharm       : NULL
## $ veins                  : NULL
## $ shapefiles             :List of 4
##   ..$ shapefiles:List of 15
##     ...$ plot_1 :'data.frame': 5 obs. of 3 variables:
##     ...$ plot: chr [1:5] "1" "1" "1" "1" ...
##     ...$ x   : num [1:5] 0 268 268 0 0
##     ...$ y   : num [1:5] 0 0 454 454 0
##     ...$ plot_2 :'data.frame': 5 obs. of 3 variables:
##     ...$ plot: chr [1:5] "2" "2" "2" "2" ...
##     ...$ x   : num [1:5] 268 536 536 268 268
##     ...$ y   : num [1:5] 0 0 454 454 0
##     ...$ plot_3 :'data.frame': 5 obs. of 3 variables:
##     ...$ plot: chr [1:5] "3" "3" "3" "3" ...
##     ...$ x   : num [1:5] 536 805 805 536 536
##     ...$ y   : num [1:5] 0 0 454 454 0
##     ...$ plot_4 :'data.frame': 5 obs. of 3 variables:
##     ...$ plot: chr [1:5] "4" "4" "4" "4" ...
##     ...$ x   : num [1:5] 805 1073 1073 805 805
##     ...$ y   : num [1:5] 0 0 454 454 0
##     ...$ plot_5 :'data.frame': 5 obs. of 3 variables:
##     ...$ plot: chr [1:5] "5" "5" "5" "5" ...
##     ...$ x   : num [1:5] 1073 1341 1341 1073 1073
##     ...$ y   : num [1:5] 0 0 454 454 0
##     ...$ plot_6 :'data.frame': 5 obs. of 3 variables:
##     ...$ plot: chr [1:5] "6" "6" "6" "6" ...
##     ...$ x   : num [1:5] 0 268 268 0 0
##     ...$ y   : num [1:5] 454 454 907 907 454

```

```

## ... .$.plot_7 :'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "7" "7" "7" "7" ...
## ... ... $.x : num [1:5] 268 536 536 268 268
## ... ... $.y : num [1:5] 454 454 907 907 454
## ... .$.plot_8 :'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "8" "8" "8" "8" ...
## ... ... $.x : num [1:5] 536 805 805 536 536
## ... ... $.y : num [1:5] 454 454 907 907 454
## ... .$.plot_9 :'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "9" "9" "9" "9" ...
## ... ... $.x : num [1:5] 805 1073 1073 805 805
## ... ... $.y : num [1:5] 454 454 907 907 454
## ... .$.plot_10:'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "10" "10" "10" "10" ...
## ... ... $.x : num [1:5] 1073 1341 1341 1073 1073
## ... ... $.y : num [1:5] 454 454 907 907 454
## ... .$.plot_11:'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "11" "11" "11" "11" ...
## ... ... $.x : num [1:5] 0 268 268 0 0
## ... ... $.y : num [1:5] 907 907 1361 1361 907
## ... .$.plot_12:'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "12" "12" "12" "12" ...
## ... ... $.x : num [1:5] 268 536 536 268 268
## ... ... $.y : num [1:5] 907 907 1361 1361 907
## ... .$.plot_13:'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "13" "13" "13" "13" ...
## ... ... $.x : num [1:5] 536 805 805 536 536
## ... ... $.y : num [1:5] 907 907 1361 1361 907
## ... .$.plot_14:'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "14" "14" "14" "14" ...
## ... ... $.x : num [1:5] 805 1073 1073 805 805
## ... ... $.y : num [1:5] 907 907 1361 1361 907
## ... .$.plot_15:'data.frame': 5 obs. of 3 variables:
## ... ... $.plot: chr [1:5] "15" "15" "15" "15" ...
## ... ... $.x : num [1:5] 1073 1341 1341 1073 1073
## ... ... $.y : num [1:5] 907 907 1361 1361 907
## ... $.bbox :'data.frame': 5 obs. of 2 variables:
## ... ... $.x: num [1:5] 0 1341 1341 0 0
## ... ... $.y: num [1:5] 0 0 1361 1361 0
## ... $.nrow : num 3
## ... $.ncol : num 5
## ... - attr(*, "class")= chr "image_shp"
## $ mask :Formal class 'Image' [package "EBIImage"] with 2 slots
## ... ..@ .Data : int [1:1341, 1:1361] 0 0 0 0 0 0 0 0 0 0 ...
## ... ..@ colormode: int 0
## ... ... $.dim: int [1:2] 1341 1361
## $ index : chr "R"
## $ object_index_computed: chr "DGCI"
## $ final_image :Formal class 'Image' [package "EBIImage"] with 2 slots
## ... ..@ .Data : num [1:1341, 1:1361, 1:3] 0.718 0.698 0.69 0.69 ...
## ... ..@ colormode: int 2
## ... ... $.dim: int [1:3] 1341 1361 3
## $ final_image_masked :Formal class 'Image' [package "EBIImage"] with 2 slots
## ... ..@ .Data : num [1:1341, 1:1361, 1:3] 1 1 1 1 1 1 1 1 1 1 ...

```

```

## ...@ colormode: int 2
## ...$ dim: int [1:3] 1341 1361 3
## - attr(*, "class")= chr "anal_obj"

res$results

##      img id      x      y area area_ch perimeter radius_mean radius_min
## 1  shp1  1 173.8875 197.2161 9067 9159.0 641.450793  79.028840 16.0106560
## 2  shp2  1 419.4291 134.7420 3034 3152.0 283.024387  35.566278 15.9119564
## 3  shp2  2 409.5526 251.2449 4561 4756.5 483.894444  57.548280 12.4058530
## 4  shp3  1 663.4812 197.3267 8246 8392.0 623.060967  74.566090 18.8763422
## 5  shp4  1 868.9454 200.1027 1794 1932.5 193.296465  24.797913 16.3736278
## 6  shp4  2 865.6604 160.1930 882 1013.5 144.426407  16.099585 2.2318799
## 7  shp4  3 861.9256 112.2877 1997 1973.5 209.669048  27.099277 13.1184023
## 8  shp4  4 875.6012 279.0875 2893 3185.5 392.338095  45.012722 11.0730407
## 9  shp5  1 1145.2620 178.7497 6506 6689.5 540.722871  66.621395 16.3124837
## 10 shp6  1 155.9354 584.9626 3267 3254.5 275.083261  35.988973 15.2281802
## 11 shp6  2 160.1674 711.8887 4803 4813.0 465.953319  57.378911 12.4049831
## 12 shp7  1 400.2638 596.0386 2934 2912.5 252.982756  33.191050 15.9888562
## 13 shp7  2 405.3649 721.2340 4967 5015.0 458.639610  57.016572 13.1670321
## 14 shp7  3 402.1105 642.1812 433 415.5 98.899495  12.980241 4.5339498
## 15 shp8  1 668.3367 674.2159 7824 7898.0 712.965512  87.081666 12.9710678
## 16 shp9  1 894.7255 645.2546 7029 7061.5 685.367532  84.672813 12.7154172
## 17 shp10 1 1167.8752 730.5821 3357 3538.0 340.338095  40.254948 13.6497265
## 18 shp10 2 1155.9773 602.2516 4492 4580.0 465.651804  56.487813 11.6390767
## 19 shp11 1 151.3224 1112.2437 8139 8200.0 657.793939  79.812618 14.4915353
## 20 shp12 1 409.5224 1071.9279 5048 5402.0 430.663997  50.562390 13.2937101
## 21 shp12 2 413.7442 1184.0553 3083 3475.0 383.977705  38.574153 10.8120465
## 22 shp12 3 418.3652 973.5014 76 80.5 42.213203  4.615324 2.8476022
## 23 shp13 1 660.2912 1036.8183 5530 5531.0 406.296465  51.834269 17.4423583
## 24 shp13 2 665.5701 1174.5687 5456 5462.0 392.882251  49.857300 17.3184442
## 25 shp14 1 909.4298 1092.8077 9405 9533.5 693.534055  83.024967 17.8581377
## 26 shp15 1 1202.6975 1137.3417 1940 2036.0 211.852814  25.433231 14.0512918
## 27 shp15 2 1198.8367 1078.6869 578 601.5 108.112698  13.734770 7.3602514
## 28 shp15 3 1198.8884 1097.4149 694 707.5 114.698485  14.638374 7.2890726
## 29 shp15 4 1193.8355 1030.5558 2531 2965.0 347.835570  32.544216 9.5760265
## 30 shp15 5 1207.7567 1191.6629 1089 1549.5 244.722871  20.896257 7.6271200
## 31 shp15 6 1216.5087 1226.9660 212 281.5 102.497475  10.180855 2.6995537
## 32 shp15 7 1186.6845 962.5903 123 154.0 76.012193  6.458577 2.8468190
## 33 shp15 8 1239.0963 910.7407 11 7.5 11.656854  1.584170 0.3277774
## 34 shp15 9 1218.3333 1255.5667 12 6.0 11.828427  1.659186 0.4249183
## 35 shp15 10 1331.3000 965.5278 13 9.0 13.071068  1.794408 0.4859127
## 36 shp15 11 1203.7697 1228.1212 13 9.0 13.656854  1.847557 0.2432521
## 37 shp15 38 1286.0500 973.8333 7 3.5 8.242641  1.261610 0.6250000
## 38 shp15 69 1249.3556 993.1111 7 3.0 9.242641  1.325692 0.4006168
## 39 shp15 73 1236.8000 994.1667 8 3.5 8.656854  1.290907 0.1250000
## 40 shp15 89 1339.3000 952.5833 8 4.5 9.414214  1.304104 0.1000000
## 41 shp15 93 1280.3000 908.8333 7 2.5 7.828427  1.189185 0.3750000
## 42 shp15 105 1288.8000 921.3333 7 4.0 11.485281  1.578584 0.0000000
## 43 shp15 144 1318.2000 947.8000 11 10.5 16.899495  1.783724 0.2108185
## 44 shp15 160 1293.3000 982.4167 7 3.5 8.242641  1.282114 0.3750000
## 45 shp15 183 1252.0667 1010.2667 8 4.0 9.656854  1.345600 0.3513642
## 46 shp15 216 1296.5778 979.5556 8 3.5 12.485281  1.692346 0.3748278
## 47 shp15 263 1239.3000 920.0833 7 3.5 8.656854  1.261610 0.6250000
## 48 shp15 282 1319.0667 1208.6667 7 3.0 6.828427  1.135047 0.4040610

```

```

## 49 shp15 287 1326.3000 938.1190 16 12.5 18.313708 2.420345 0.5153882
## 50 shp15 296 1227.3000 919.4167 7 3.0 9.071068 1.339804 0.5000000
##   radius_max radius_sd diam_mean diam_min diam_max major_axis minor_axis
## 1 151.589329 40.5666357 158.057680 32.0213120 303.178658 87.371069 15.9621459
## 2 59.206727 13.2182269 71.132556 31.8239127 118.413454 35.518726 13.3202749
## 3 112.321971 28.5239188 115.096560 24.8117061 224.643943 63.075467 12.0460749
## 4 143.696158 36.7504708 149.132181 37.7526844 287.392316 81.576915 15.9220365
## 5 40.132013 6.0170291 49.595826 32.7472555 80.264025 22.098482 12.7510837
## 6 22.293752 4.1160594 32.199170 4.4637598 44.587503 12.295531 11.1727844
## 7 44.217002 9.1533776 54.198555 26.2368046 88.434003 26.066020 11.7597813
## 8 88.733069 21.3889491 90.025444 22.1460814 177.466139 48.714183 10.4551740
## 9 127.065717 33.4158588 133.242789 32.6249674 254.131433 73.200930 13.9448904
## 10 59.581144 13.8899396 71.977946 30.4563603 119.162288 35.987639 13.8668698
## 11 110.140699 29.1800007 114.757821 24.8099662 220.281397 63.224086 12.0255285
## 12 55.330214 11.2974861 66.382100 31.9777124 110.660427 31.898032 14.5346479
## 13 109.489358 28.5138425 114.033144 26.3340643 218.978716 62.413228 12.9102300
## 14 20.264743 5.2745405 25.960482 9.0678997 40.529485 13.198071 4.6724241
## 15 169.595995 46.8714231 174.163332 25.9421355 339.191990 98.135435 12.0976238
## 16 165.984561 46.1023355 169.345627 25.4308344 331.969123 95.784026 10.8226674
## 17 73.958306 17.0214462 80.509896 27.2994530 147.916611 41.788606 12.7663690
## 18 109.128576 29.0925696 112.975626 23.2781533 218.257152 62.523665 11.2294584
## 19 148.392265 40.8745582 159.625235 28.9830705 296.784529 88.557714 13.9870388
## 20 87.042216 22.4619305 101.124779 26.5874202 174.084432 53.314499 14.7431611
## 21 72.945472 17.1734573 77.148307 21.6240929 145.890945 40.254068 12.7134949
## 22 7.135801 1.0385369 9.230647 5.6952045 14.271601 3.810200 2.7987497
## 23 95.287302 22.6466110 103.668538 34.8847166 190.574605 54.185425 16.1949024
## 24 85.400523 21.1241776 99.714599 34.6368883 170.801046 51.576085 16.4528267
## 25 154.678288 42.1477909 166.049934 35.7162754 309.356576 91.824890 15.3301194
## 26 42.839042 6.9323884 50.866462 28.1025836 85.678083 22.745991 13.3143347
## 27 18.938672 3.6338481 27.469539 14.7205027 37.877344 12.331114 7.0470089
## 28 19.754562 3.0850432 29.276748 14.5781452 39.509124 12.195959 8.6584715
## 29 57.039731 13.1330971 65.088431 19.1520530 114.079462 32.808103 12.4366694
## 30 35.247405 6.8765918 41.792514 15.2542400 70.494809 19.592892 9.9918786
## 31 18.711168 4.3269394 20.361710 5.3991073 37.422336 10.589553 3.1659464
## 32 10.460023 2.0266053 12.917154 5.6936381 20.920045 6.128204 2.8639619
## 33 2.972324 0.7882039 3.168339 0.6555548 5.944648 1.625472 0.6574358
## 34 2.974428 0.7861627 3.318371 0.8498366 5.948856 1.707015 0.6368228
## 35 3.093452 0.8804882 3.588817 0.9718253 6.186904 1.880171 0.6288964
## 36 3.360051 0.8590554 3.695113 0.4865043 6.720101 1.891305 0.7194720
## 37 2.211476 0.6699374 2.523221 1.2500000 4.422952 1.358164 0.3738520
## 38 2.320068 0.6835965 2.651383 0.8012336 4.640136 1.437581 0.3258847
## 39 2.294695 0.7114841 2.581815 0.2500000 4.589390 1.388138 0.4271383
## 40 2.282542 0.8091929 2.608207 0.2000000 4.565085 1.461324 0.3931068
## 41 2.095382 0.6049444 2.378371 0.7500000 4.190764 1.266004 0.3627787
## 42 2.828427 1.0044740 3.157169 0.0000000 5.656854 1.824248 0.2685478
## 43 3.402613 1.0558772 3.567448 0.4216370 6.805227 1.960827 0.6143118
## 44 2.095382 0.6238663 2.564228 0.7500000 4.190764 1.340702 0.4323117
## 45 2.519063 0.7577209 2.691201 0.7027284 5.038126 1.470043 0.3999506
## 46 3.364865 0.9975046 3.384692 0.7496556 6.729729 1.917948 0.3001190
## 47 2.211476 0.6699374 2.523221 1.2500000 4.422952 1.358164 0.3738520
## 48 1.857143 0.5535981 2.270094 0.8081220 3.714286 1.158684 0.4565870
## 49 4.692081 1.3170906 4.840689 1.0307764 9.384162 2.675023 0.5732588
## 50 2.500000 0.7210514 2.679608 1.0000000 5.000000 1.441170 0.4159668
##   caliper      length       width radius_ratio      theta eccentricity

```

```

## 1 284.015845 297.106189 45.4577649 9.468027 -1.54708223 0.9831699
## 2 112.004464 117.872101 35.8427750 3.720896 -1.52423279 0.9270162
## 3 193.380454 205.388617 36.0870477 9.053950 -1.45474970 0.9815942
## 4 264.367169 271.625852 48.0517727 7.612500 1.36458518 0.9807678
## 5 76.164296 76.161216 34.2070333 2.451015 1.51409526 0.8167359
## 6 43.139309 36.180302 35.4421894 9.988778 -0.22585841 0.4174789
## 7 80.081209 78.652657 31.2329984 3.370609 1.48166369 0.8924463
## 8 152.029602 158.416268 32.4817890 8.013433 1.48508146 0.9766970
## 9 233.308808 249.126869 39.3321990 7.789477 1.53508183 0.9816869
## 10 112.008928 110.203318 35.2580131 3.912558 1.52035269 0.9227817
## 11 206.349219 206.908696 34.1059523 8.878746 1.56412700 0.9817444
## 12 99.824847 97.896903 39.0772267 3.460549 1.55733598 0.8901541
## 13 196.308431 202.836347 38.7125393 8.315417 1.54577106 0.9783725
## 14 40.162171 39.111168 11.1363899 4.469556 -0.09115646 0.9352366
## 15 326.000000 337.011374 33.6345799 13.074945 1.56532152 0.9923726
## 16 315.101571 327.109367 29.1434180 13.053804 1.55866675 0.9935961
## 17 131.939380 132.832984 33.7456786 5.418299 1.47058744 0.9521925
## 18 206.784912 214.032629 33.5361545 9.376051 1.47166251 0.9837391
## 19 284.395851 296.019307 35.4900555 10.239927 1.55006241 0.9874483
## 20 170.355511 172.018807 43.3538930 6.547624 -1.52960056 0.9610047
## 21 125.399362 131.364300 40.0309540 6.746685 1.46346551 0.9488154
## 22 14.212670 14.199094 10.0421861 2.505898 0.84135428 0.6785638
## 23 174.797025 174.702281 45.2327704 5.462983 -1.54076299 0.9542909
## 24 159.201131 159.681937 43.5141375 4.931189 1.53779694 0.9477544
## 25 303.237531 304.257358 40.6424766 8.661502 1.50413311 0.9859654
## 26 73.824115 72.708518 37.5880208 3.048762 1.54422235 0.8107817
## 27 36.715120 36.093311 19.2171297 2.573101 -0.12488720 0.8206146
## 28 39.357337 35.461378 22.7552318 2.710161 0.02630802 0.7042562
## 29 106.677083 104.918459 40.3410435 5.956513 1.49389459 0.9253667
## 30 67.911707 66.087039 32.2251135 4.621326 -1.56529391 0.8601894
## 31 35.171011 35.168492 11.6288676 6.931208 1.32399453 0.9542629
## 32 20.248457 20.093867 10.3557071 3.674284 1.34145355 0.8840771
## 33 5.385165 5.289406 2.3443279 9.068118 1.37915425 0.9145564
## 34 5.385165 5.348681 2.1916684 7.000000 -0.26403722 0.9278064
## 35 5.830952 5.789154 2.0164448 6.366271 0.66022645 0.9423997
## 36 6.082763 5.985367 2.2743129 13.813037 1.50094189 0.9248178
## 37 4.242641 4.192672 1.1811461 3.538361 0.63176876 0.9613690
## 38 4.472136 4.469351 0.9885564 5.791240 0.42835281 0.9739670
## 39 4.472136 4.465655 1.3637649 18.357560 0.51749254 0.9514815
## 40 4.472136 4.383820 1.2267079 22.825424 0.26458228 0.9631381
## 41 4.123106 4.122062 1.2287402 5.587685 0.26747754 0.9580641
## 42 5.656854 5.632580 1.0155251 Inf 0.87807214 0.9891052
## 43 6.403124 6.395846 2.6448257 16.140012 0.84837057 0.9496568
## 44 4.123106 4.041522 1.3761035 5.587685 0.44423989 0.9465859
## 45 5.000000 4.999951 1.4008746 7.169379 0.64794324 0.9622782
## 46 6.403124 6.402557 1.0473747 8.977095 0.68805539 0.9876812
## 47 4.242641 4.192672 1.1811461 3.538361 0.93902757 0.9613690
## 48 3.605551 3.597548 1.3652029 4.596194 0.52136094 0.9190862
## 49 8.602325 8.584931 2.1133625 9.103975 0.55664494 0.9767678
## 50 4.472136 4.440346 1.3853321 5.000000 0.58295227 0.9574403

## form_factor narrow_factor asp_ratio rectangularity pd_ratio plw_ratio
## 1 0.2769152 0.9559405 6.535873 1.4895537 2.258504 1.872499
## 2 0.4759680 0.9502203 3.288587 1.3925060 2.526903 1.841230
## 3 0.2447759 0.9415344 5.691477 1.6250535 2.502292 2.003906

```

```

## 4 0.2669268 0.9732769 5.652775 1.5828406 2.356802 1.949029
## 5 0.6033711 1.0000404 2.226478 1.4522014 2.537888 1.751377
## 6 0.5313556 1.1923424 1.020826 1.4538652 3.347907 2.016495
## 7 0.5708464 1.0181628 2.518255 1.2301243 2.618205 1.908066
## 8 0.2361769 0.9596843 4.877080 1.7786532 2.580669 2.055223
## 9 0.2796240 0.9365060 6.333917 1.5061032 2.317627 1.874522
## 10 0.5425386 1.0163844 3.125625 1.1893327 2.455905 1.891109
## 11 0.2779954 0.9972960 6.066645 1.4692522 2.258081 1.933299
## 12 0.5760871 1.0196936 2.505216 1.3038648 2.534266 1.846938
## 13 0.2967296 0.9678168 5.239552 1.5808959 2.336322 1.898744
## 14 0.5563007 1.0268722 3.512015 1.0059058 2.462504 1.968245
## 15 0.1934201 0.9673264 10.019788 1.4487776 2.187011 1.923576
## 16 0.1880426 0.9632912 11.224125 1.3562505 2.175069 1.923824
## 17 0.3642001 0.9932727 3.936296 1.3352813 2.579504 2.043107
## 18 0.2603317 0.9661373 6.382146 1.5979144 2.251865 1.880899
## 19 0.2363748 0.9607341 8.340909 1.2907902 2.312952 1.984239
## 20 0.3420205 0.9903307 3.967782 1.4773544 2.528031 1.999622
## 21 0.2627673 0.9545924 3.281568 1.7056887 3.062039 2.240305
## 22 0.5359532 1.0009562 1.413945 1.8761835 2.970111 1.741377
## 23 0.4209678 1.0005423 3.862295 1.4289816 2.324390 1.847347
## 24 0.4441804 0.9969890 3.669656 1.2735377 2.467836 1.933513
## 25 0.2457158 0.9966481 7.486191 1.3148084 2.287098 2.010827
## 26 0.5431792 1.0153434 1.934353 1.4087470 2.869697 1.920757
## 27 0.6214187 1.0172278 1.878184 1.2000170 2.944637 1.954653
## 28 0.6629091 1.1098648 1.558384 1.1627261 2.914285 1.970202
## 29 0.2628779 1.0167618 2.600787 1.6722719 3.260640 2.394580
## 30 0.2285013 1.0276101 2.050793 1.9556128 3.603545 2.489243
## 31 0.2535826 1.0000716 3.024240 1.9291025 2.914260 2.190241
## 32 0.2675152 1.0076934 1.940367 1.6917577 3.753975 2.496330
## 33 1.0172784 1.0181038 2.256257 1.1272821 2.164623 1.527019
## 34 1.0777974 1.0068211 2.440461 0.9768779 2.196484 1.568684
## 35 0.9561611 1.0072200 2.870971 0.8979623 2.241670 1.674576
## 36 0.8758946 1.0162723 2.631725 1.0471229 2.245173 1.653436
## 37 1.2947179 1.0119182 3.549664 0.7074511 1.942809 1.533852
## 38 1.0297120 1.0006232 4.521088 0.6311722 2.066717 1.693440
## 39 1.3414659 1.0014514 3.274505 0.7612629 1.935731 1.485029
## 40 1.1343100 1.0201460 3.573646 0.6722083 2.105082 1.677955
## 41 1.4353535 1.0002532 3.354706 0.7235633 1.898672 1.463038
## 42 0.6668446 1.0043097 5.546470 0.8171466 2.030330 1.727602
## 43 0.4840109 1.0011380 2.418248 1.5378088 2.639258 1.869274
## 44 1.2947179 1.0201863 2.936932 0.7945075 1.999134 1.521449
## 45 1.0780242 1.0000099 3.569164 0.8755380 1.931371 1.508689
## 46 0.6449160 1.0000886 6.112957 0.8382345 1.949873 1.675892
## 47 1.1737827 1.0119182 3.549664 0.7074511 2.040440 1.610932
## 48 1.8865423 1.0022247 2.635175 0.7016261 1.893865 1.375936
## 49 0.5994835 1.0020262 4.062214 1.1339419 2.128925 1.711835
## 50 1.0690329 1.0071593 3.205258 0.8787649 2.028352 1.557083

## solidity convexity elongation circularity circularity_haralick
## 1 0.9899552 0.8939527 0.84699826 45.379852 1.948124
## 2 0.9625635 0.8255487 0.69591808 26.401715 2.690700
## 3 0.9588983 0.7918351 0.82429870 51.338266 2.017545
## 4 0.9826025 0.8380844 0.82309573 47.077973 2.028983
## 5 0.9283312 0.8006254 0.55086020 20.826936 4.121288
## 6 0.8702516 0.7781469 0.02040094 23.649645 3.911407

```

	circularity_norm	coverage			
## 7	1.0119078	0.9144376	0.60289964	22.013575	2.960577
## 8	0.9081777	0.7273545	0.79495926	53.207460	2.104485
## 9	0.9725690	0.8636903	0.84211980	44.940243	1.993706
## 10	1.0038408	0.8778358	0.68006396	23.162167	2.591010
## 11	0.9979223	0.9389099	0.83516424	45.203518	1.966378
## 12	1.0073820	0.8913393	0.60083286	21.813318	2.937915
## 13	0.9904287	0.9198568	0.80914397	42.349566	1.999610
## 14	1.0421179	0.8827536	0.71526317	22.589169	2.460924
## 15	0.9906305	0.9202176	0.90019749	64.969302	1.857884
## 16	0.9953976	0.9325963	0.91090620	66.827238	1.836627
## 17	0.9488412	0.7734072	0.74595407	34.504027	2.364955
## 18	0.9807860	0.6994329	0.84331289	48.270615	1.941658
## 19	0.9925610	0.8508373	0.88010898	53.162903	1.952623
## 20	0.9344687	0.5720481	0.74797004	36.741577	2.251026
## 21	0.8871942	0.6711963	0.69526763	47.823185	2.246150
## 22	0.9440994	0.6258196	0.29275867	23.446770	4.444063
## 23	0.9998192	0.8249664	0.74108655	29.851142	2.288831
## 24	0.9989015	0.7333754	0.72749493	28.291141	2.360201
## 25	0.9865212	0.8792919	0.86642073	51.141891	1.969853
## 26	0.9528487	0.7375976	0.48303140	23.134853	3.668755
## 27	0.9609310	0.8242435	0.46757088	20.222068	3.779676
## 28	0.9809187	0.8321562	0.35830944	18.956401	4.744950
## 29	0.8536256	0.6862254	0.61550099	47.803075	2.478031
## 30	0.7028074	0.4843328	0.51238376	54.994751	3.038752
## 31	0.7531083	0.6149950	0.66933846	49.555341	2.352900
## 32	0.7987013	0.5614295	0.48463345	46.974419	3.186894
## 33	1.4666667	0.7928932	0.55678808	12.352932	2.009847
## 34	2.0000000	0.8853363	0.59024133	11.659307	2.110486
## 35	1.4444444	0.6622913	0.65168575	13.142524	2.037970
## 36	1.4444444	0.5905090	0.62002115	14.346898	2.150684
## 37	2.0000000	0.6854941	0.71828319	9.705875	1.883177
## 38	2.3333333	0.8084417	0.77881433	12.203772	1.939290
## 39	2.2857143	0.7682099	0.69461030	9.367641	1.814387
## 40	1.7777778	0.9863153	0.72017371	11.078427	1.611610
## 41	2.8000000	0.8193490	0.70191129	8.754896	1.965776
## 42	1.7500000	0.6094757	0.81970515	18.844527	1.571553
## 43	1.0476190	0.7088742	0.58647757	25.962994	1.689329
## 44	2.0000000	0.6854941	0.65950859	9.705875	2.055110
## 45	2.0000000	0.7982107	0.71982231	11.656854	1.775852
## 46	2.2857143	0.7122725	0.83641305	19.485281	1.696580
## 47	2.0000000	0.6526945	0.71828319	10.705875	1.883177
## 48	2.3333333	0.9739112	0.62051850	6.661060	2.050308
## 49	1.2800000	0.7807444	0.75382882	20.961995	1.837645
## 50	2.3333333	0.6489147	0.68801261	11.754896	1.858125

```

## 10      0.52061788 2.685088e-02
## 11      0.26505934 3.947498e-02
## 12      0.55262341 2.402437e-02
## 13      0.28349717 4.067111e-02
## 14      0.49270513 3.545519e-03
## 15      0.18499013 6.406498e-02
## 16      0.17913410 5.755531e-02
## 17      0.34705873 2.748800e-02
## 18      0.24763965 3.678168e-02
## 19      0.22740075 6.689296e-02
## 20      0.32891012 4.133436e-02
## 21      0.24823544 2.524442e-02
## 22      0.40549088 6.223081e-04
## 23      0.40616164 4.528110e-02
## 24      0.42887508 4.467517e-02
## 25      0.23730320 7.701063e-02
## 26      0.51714019 1.588523e-02
## 27      0.566658760 4.732817e-03
## 28      0.61085067 5.682656e-03
## 29      0.24755804 2.072450e-02
## 30      0.20594490 8.917020e-03
## 31      0.19915801 1.735912e-03
## 32      0.19465536 1.007157e-03
## 33      0.41615934 9.007091e-05
## 34      0.44908223 9.825917e-05
## 35      0.44130513 1.0644474e-04
## 36      0.37057081 1.0644474e-04
## 37      0.36991941 5.731785e-05
## 38      0.22065257 5.731785e-05
## 39      0.50304972 6.550612e-05
## 40      0.28357750 6.550612e-05
## 41      0.41010100 5.731785e-05
## 42      0.09526351 5.731785e-05
## 43      0.11000247 9.007091e-05
## 44      0.36991941 5.731785e-05
## 45      0.33688255 6.550612e-05
## 46      0.12092176 6.550612e-05
## 47      0.33536648 5.731785e-05
## 48      0.67376511 5.731785e-05
## 49      0.26227404 1.310122e-04
## 50      0.30543798 5.731785e-05

```

Exemplo 4

```

soy <- image_pliman("soybean_touch.jpg")
plot(soy)

```



Conta os objetos na imagem

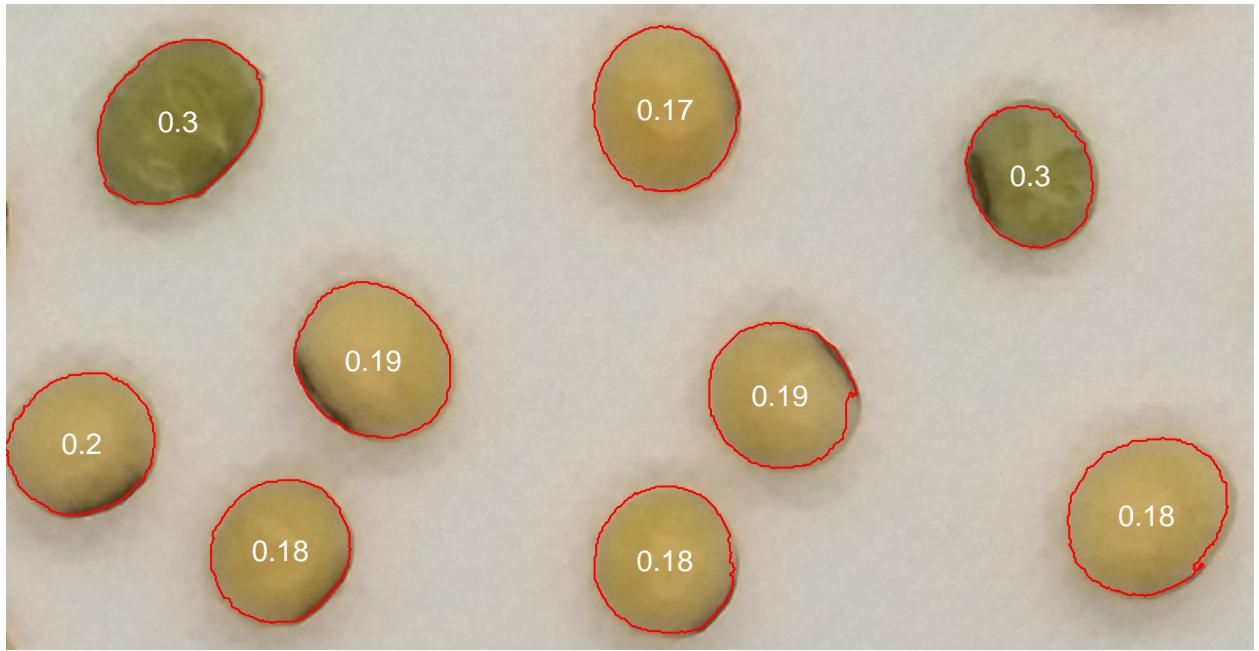
```
grains <- analyze_objects(soy)
```



Exemplo 5

```
img <- image_pliman("soy_green.jpg")
plot(img)
rgb <- analyze_objects(img,
                       object_index = "DGCI",
                       pixel_level_index = TRUE)

plot_measures(rgb, measure = "DGCI")
```

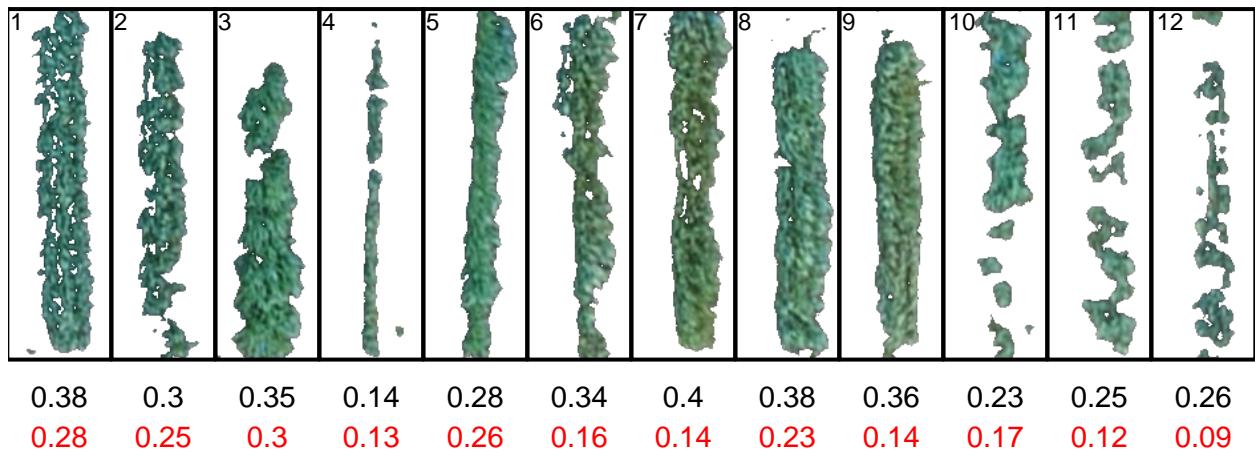


```
shp <- image_pliman("field_mosaic.jpg")
plot(shp)
```



Exemplo 6 - Cobertura vegetal

```
res <-  
  analyze_objects_shp(shp,  
    ncol = 12,  
    nrow = 1,  
    index = "HUE", # Segmenta o solo  
    object_index = "NGRDI")  
plot(res$final_image_masked)  
plot(res$shapefiles)  
plot_measures(res, measure = "coverage", col = "black", vjust = -160)  
plot_measures(res, measure = "NGRDI", col = "red", vjust = -190)
```



Exemplo 7 - Nível de Severidade

```
img <- image_pliman("sev_leaf.jpg")
plot(img)
```



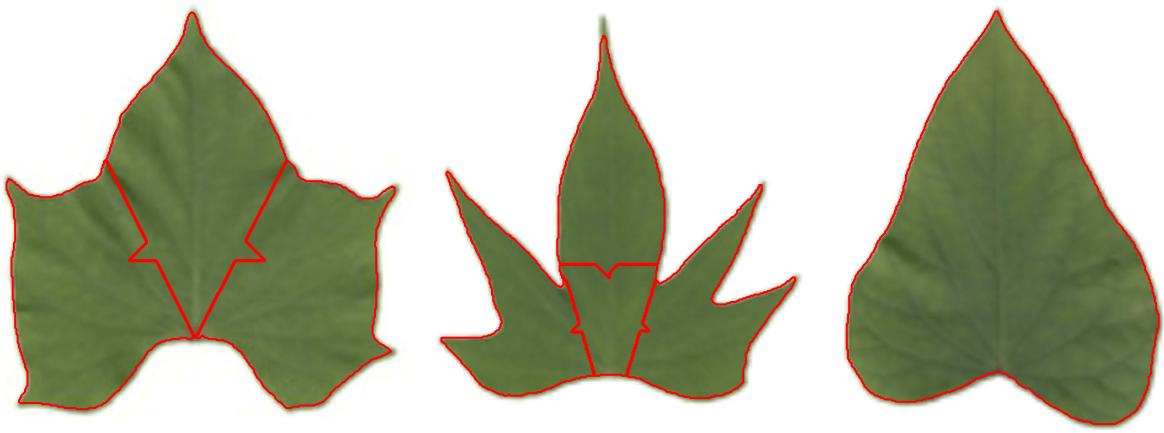
```
sev <- measure_disease(img = img,
                        index_lb = "B", # remove o background
                        index_dh = "NGRDI", # isola a area com doença
                        threshold = c("Otsu", 0),
                        plot = TRUE)
```



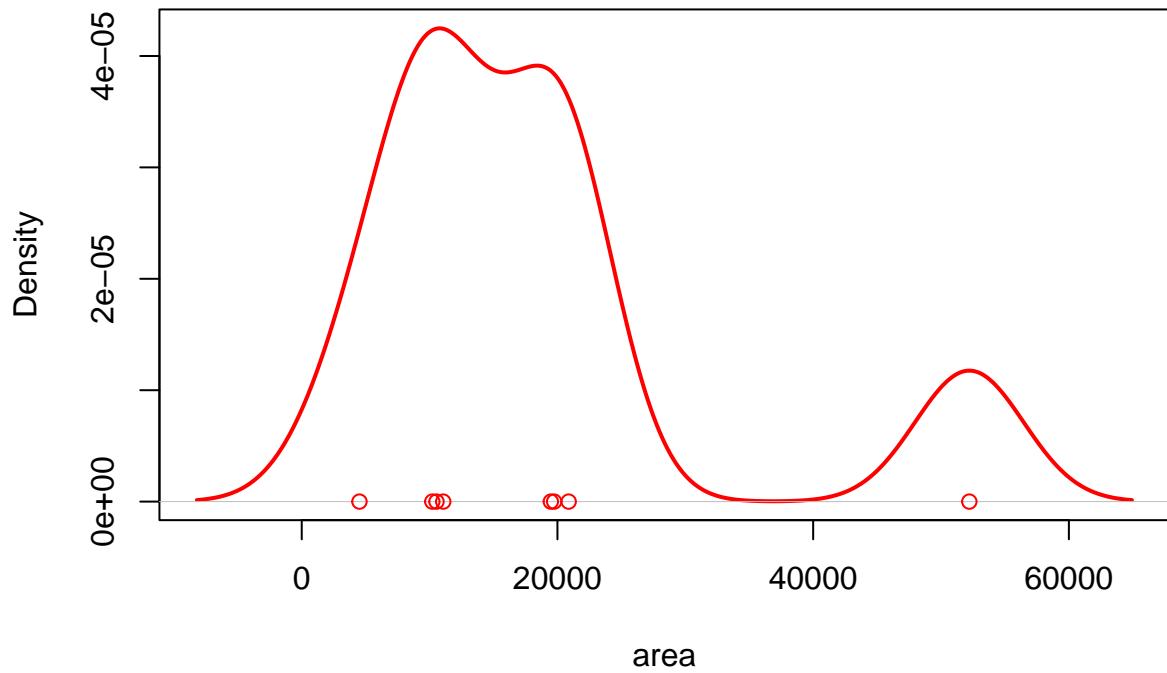
```
sev$severity  
##      healthy symptomatic  
## 1 92.78098    7.219017
```

Exemplo 8 - Batata Doce

```
img <- image_pliman("potato_leaves.jpg")  
plot(img)  
ana_i = analyze_objects(img)
```



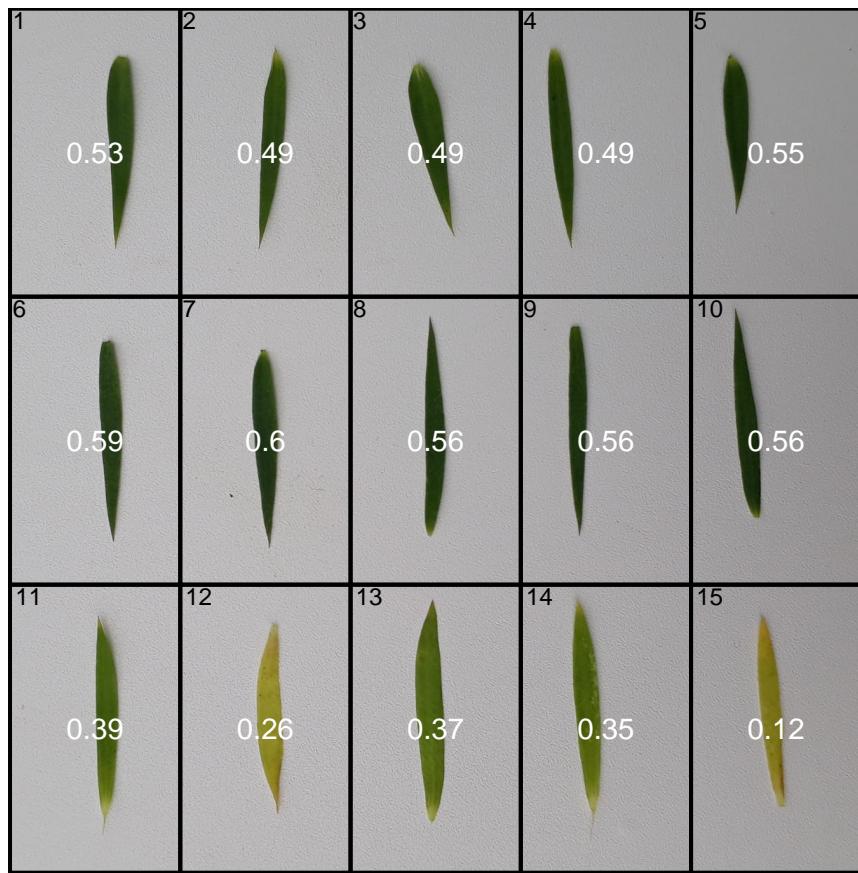
```
ana_i = analyze_objects_shp(img)
plot(ana_i)
```



```
medidas = get_measures(ana_i)
```

Exemplo 9 - Folhas Lanceoladas

```
flax <- image_pliman("flax_leaves.jpg", plot =TRUE)
res <-
  analyze_objects_shp(flax,
    nrow = 3,
    ncol = 5,
    plot = FALSE,
    object_index = "DGCI")
plot(flax)
plot(res$shapefiles)
plot_measures(res, measure = "DGCI")
```



A partir daqui foram usadas imagens que não estão no R, foram tiradas de Smartphone, portanto foi utilizada a função “image_import”. Adequem os nomes das imagens às suas imagens.

Exemplo 10 - Nível de Severidade - palmeiras de jardim

```

pal01 <- image_import("pal01.jpeg")
pal02 <- image_import("pal02.jpeg")

pal01_rgb <-
  analyze_objects_shp(pal01)

pal02_rgb <-
  analyze_objects(pal02,
    object_index = "B",
    pixel_level_index = TRUE)

```



```
plot(pal01)

sev01 <- measure_disease(img = pal01,
                           watershed = TRUE,
                           img_healthy = "pal_s",
                           img_symptoms = "pal_d",
                           invert = TRUE,
                           index_lb = "B", # remove o fundo
                           index_dh = "NGRDI", # isola a area com doença
                           plot = TRUE)
```



```
sev01$severity
```

```
##      healthy symptomatic
## 1 95.70899    4.291015

sev02 <- measure_disease(img = pal02,
                           invert = TRUE,
                           index_lb = "B", # remove o fundo
                           index_dh = "NGRDI", # isola a area com doença
                           plot = TRUE)
```



```
sev02$severity  
##      healthy symptomatic  
## 1 63.23319    36.76681
```

Exemplo 11 - Soja

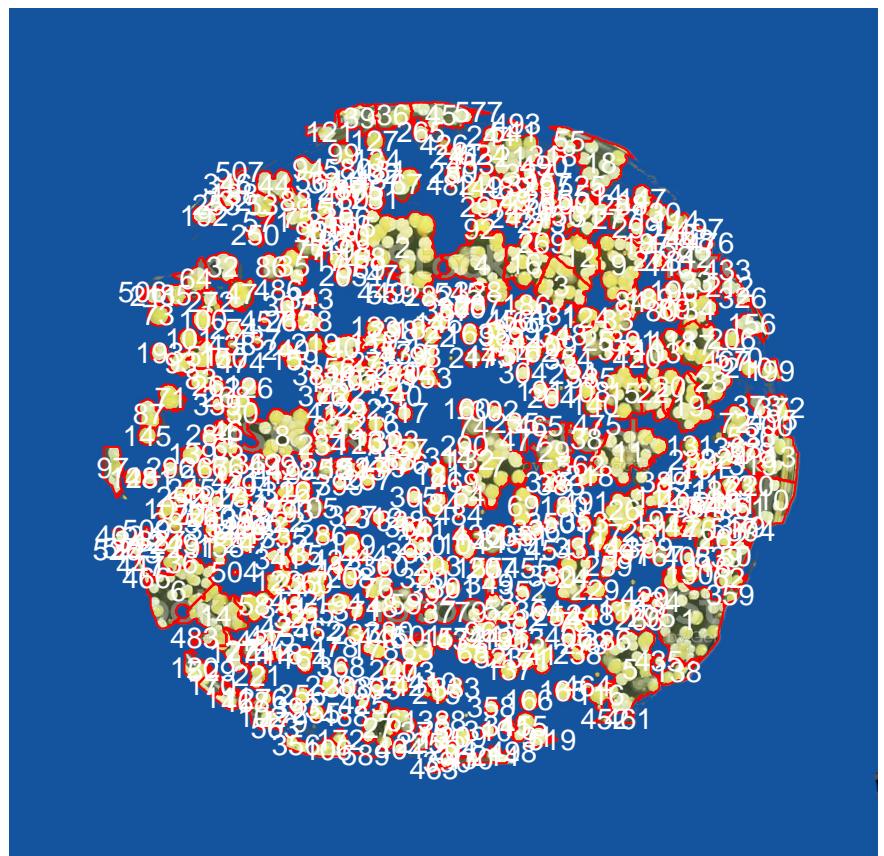
```
soy = image_import("soy_S.jpg")  
  
sev03 <- measure_disease(img = soy,  
                           img_healthy = "soy_Sintoma01",  
                           img_symptoms = "soy_padrao",  
                           invert = F,  
                           index_lb = "B",  
                           index_dh = "NGRDI",  
                           threshold = c("Otsu", 0),  
                           plot = TRUE)
```



```
sev03$severity
```

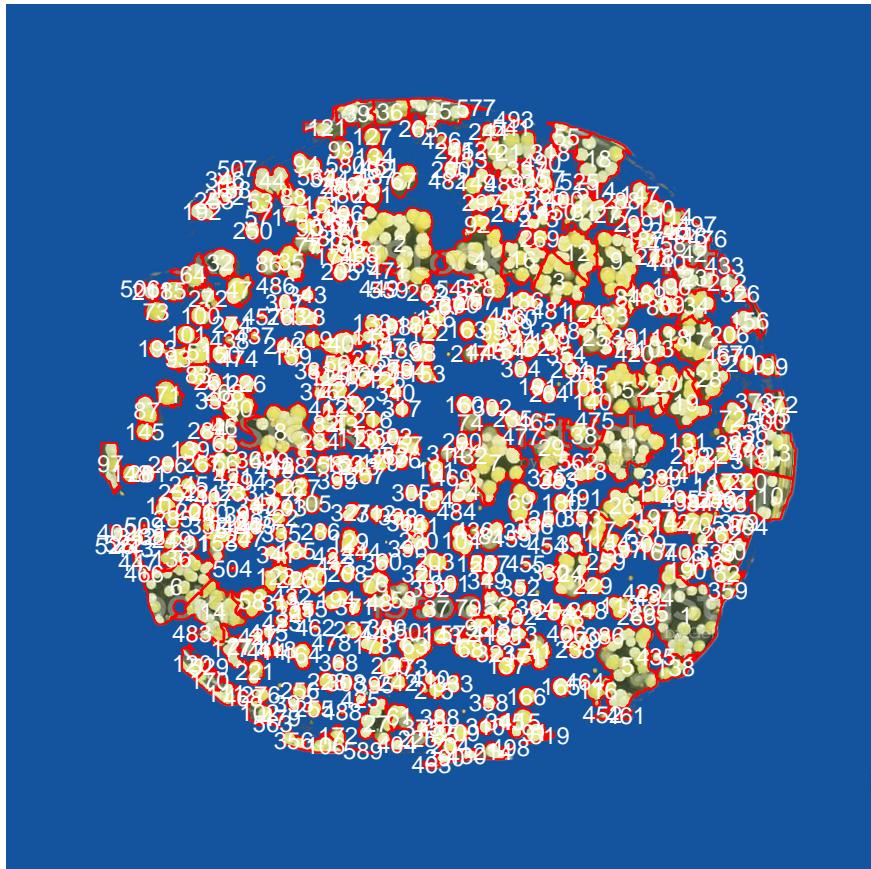
```
##      healthy symptomatic
## 1 48.60548    51.39452
```

```
bac <- image_import("colonias.jpg", plot =TRUE)
bac1 = analyze_objects(bac)
plot_measures(bac1)
```



Exemplo 12 - Bacterias

```
pick_count(bac)  
ids = object_id(bac)
```



```
nrow(ids$results)
```

```
## [1] 529
```