R package 'LogicMaps'

Martin Elstner

2015–04–12

1 Basics

Package "EBImage" from BioConductor is needed since the build-in image processing capabilities in R are poor.

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("EBImage")

> library("EBImage")
> source("LogicMaps.R")
```

We use here the EBImage routines to read and display images, and we also store values in `Image` objects provided by EBImage. Nevertheless, most functions provided by LogicMaps work also with appropriate matrices.

The package provides three example files: `blue.tif`, `green.tif` and `light.tif`.

To read the images use

```
> blue <- readImage(files="blue.tif")
> green <- readImage(files="green.tif")
> light <- readImage(files="light.tif")
```

A basic viewing environment is provided by

```
> display(blue)
```

which opens the image in the standard web browser. Along with writeImage, the standard R command `image` allows the user to save modified images to files.

Since we are interested only in the intensity values for a distinct color in a single image, we can reduce them to gray scale images in the respective channel:

```
> blue <- channel(blue, mode="blue")
> green <- channel(green, mode="green")
```

Such a gray scale image could be written to a jpg file of the size 400×400 with the command

```
> jpeg(filename = "green.jpg", height = 400, width = 400)
> image(green, col = c("black", "white"), zlim = c(0,1), axes = F)
> dev.off()
```

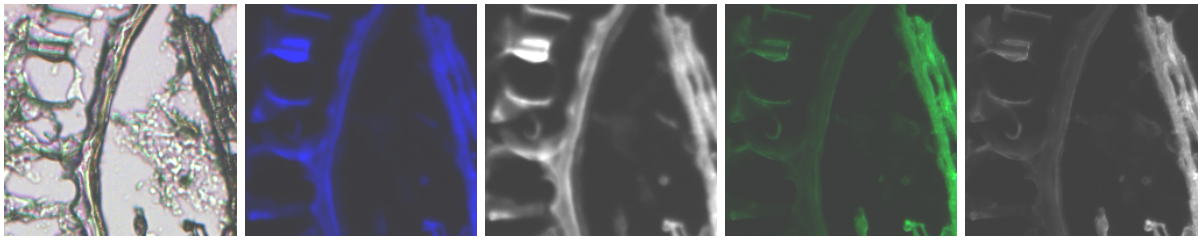

Figure 1: Sample image files. Left, the light microscopy image, followed by the blue fluorescence channel and the gray-scale representation and the same for the green fluorescence channel.

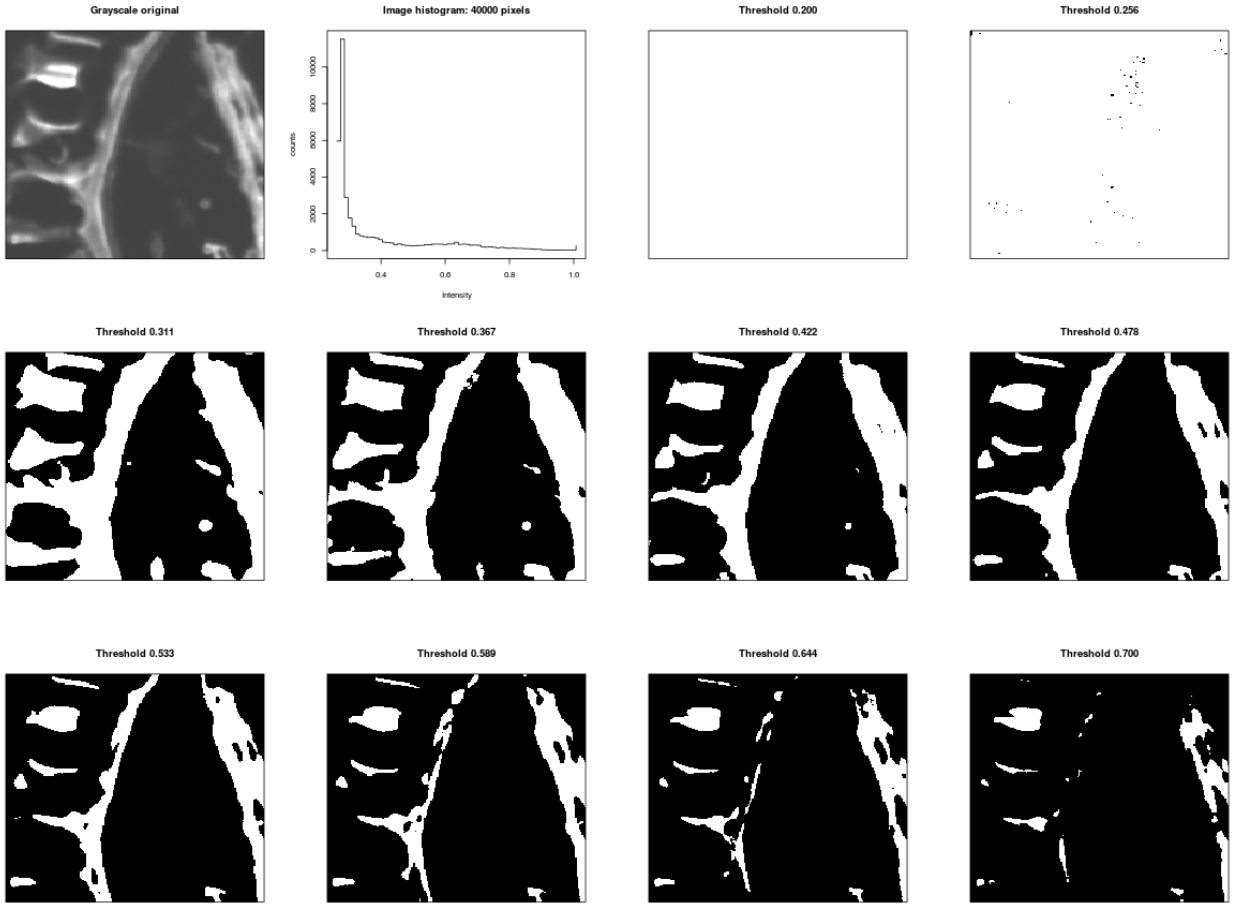

Figure 2: Threshold determination. The function `ShowThresholdDep` generates an image matrix containing the original gray-scale image and histogram (default). With the `thresh`-option the user controls the range and the number of tested thresholds (from, to, intercepts).

2 Data exploration

2.1 Threshold determination

The function `ShowThresholdDep` allows the user to evaluate or find appropriate threshold values for image binarization.

```
> ShowThresholdDep(pic, thresh = c(0, 1, 4),
+ pic.col = c("black", "white"),
+ out.size = c(row = 2, col = 3, width = 1000, height = 700),
+ file = "threshold.png", keep.orig = TRUE, make.hist = TRUE)
```

3 Mask images

```
> green <- green > 0.5
> blue <- blue > 0.5

> operated <- RunLogicOperation(blue, green, "inh")
> masked <- MaskRGBImage(light, operated)
> writeImage(masked, "light_masked.png")
```