How to save plots on your hard drive in IsoriX?

The IsoriX core team

2017-07-07

Welcome to **IsoriX**, in this vignette we present the steps required for you to save your isoscape or assignment maps on your hard drive.

Before starting

Please read the vignette ***Workflow***, if you haven't done so. You can access it by simply typing:

```
vignette("Workflow", package="IsoriX")
```

We assume that you are still in the same R session than the one created during the simple workflow example. That implies that **IsoriX** is loaded and that the objects we will use below have already been created.

Saving IsoriX plots using Cairo

There are many ways to save the plots we created during the workflow example. If you are not new to R you may already know some function to save plots (`pdf()`, `png()`...) that are readily available in R. Unfortunately, we encountered often some little annoying artefacts using these functions, such as white lines in the middle of the isoscape.

Our favorite way to create plots is thus to use the package **Cairo**. So, start by loading this package (after installing the package if you don't have it):

```
library(Cairo)
```

Typing `?Cairo` will show you all possible file formats you can save your plots into (png, jpg, tiff, pdf...). Here we will show how to save our isoscape both as a png and as a pdf.

Example: saving the isoscape

During the workflow example you have created the following isoscape:

```
isoscape
```

```
## ### stack containing the isoscape

## Loading required package: raster

## Loading required package: sp

## Warning: package 'sp' was built under R version 3.4.1

## class       : RasterStack
## dimensions  : 514, 1115, 573110, 8  (nrow, ncol, ncell, nlayers)
## resolution  : 0.08333333, 0.08333333  (x, y)
## extent      : -31.55847, 61.35819, 28.06653, 70.89986  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## names       :          mean,  mean.predVar, mean.residVar,  mean.respVar,          disp,  disp.predVa
## min values  : -123.52350359,    6.66572055,   67.02201337,   89.97938852,   67.02201337,    0.012748
## max values  :  5.801003e+00,  2.246382e+02,  1.559210e+03,  1.580116e+03,  1.559210e+03,  6.066897e-0
```

```
##
##
## ### first 5 locations of the dataset      coordinates values
## 1   (35.3, 36.98)  -9999
## 2 (-27.19, 38.77)  -9999
## 3  (35.77, 32.09)  -9999
## 4  (36.14, 30.33)  -9999
## 5 (-27.19, 38.79)  -9999
## NULL
```

Our goal is to save the plot of this isoscape on the hard drive so we can include it in a paper, poster or presentation. To do so we first store the plot into an R object that we call here `plot.mean.isoscape`:

```
plot.mean.isoscape <- plot(x = isoscape, plot = FALSE)
```

Note that we used the argument `plot = FALSE` so that the plot does not show up, you can also keep the default setting that will once again display the plot on your screen. It does not make any difference by when you handle many plots, it may be good to know that you don't have to display them (display takes time).

Let us first save this plot as a png file:

```
CairoPNG(filename = "Myisoscape.png", height = 1080, width = 1920, res = 100)
plot.mean.isoscape
```

```
dev.off()
```

```
## pdf
##   2
```

So you see, it is simple isn't it! You first initialize the creation of the plot with the function `CairoPNG()`, then you call your plot, then you tell your computer that you are done with `dev.off()`.

You just need to specify a filename, and the resolution of the file. The height and width are here considered to be in pixels. Here we chose the so-called Full-HD or 1080p resolution (1080x1920) because we wanted to observe the isoscape carefully on a monitor of that resolution. If your screen is Full-HD, try it and display the plot in full screen to get better results (if the plot does not match the definition of your screen, things can get ugly), if your screen is not, try another resolution. If you don't know what resolution your screen has, you can visit (http://www.whatismyscreenresolution.com/).

The parameter `res` is very useful as it rescales the line and fonts in the plot. So if every thing is too small increase the value and if everything looks too bold and ugly, decreases it.

If, like us, you don't indicate the path in the filename. The file will be now in your working directory which you can easily see by typing `getwd()`!

Let us now save this plot as a pdf file:

```
CairoPDF(file = "Myisoscape.pdf", height = 10, width = 15)
plot.mean.isoscape
```

```
dev.off()
```

```
## pdf
##   2
```

The arguments slightly differ with the function creating pngs. The argument setting the name of the file is now called `file` and not `filename`, the resolution has now to be provided in inches and the argument `res` does not exist for pdfs, but default results are usually fine.

Changing font size? the example of the IsoriX welcome picture

You can influence the font size while exporting your plots. Here is for example how we created the welcome picture of the **IsoriX** page on GitHub (www.github.com/courtiol/isorix_project):

```
plot.welcome.isoscape <- plot(x = isoscape,
                              palette = list(n.labels = 15,
                                             range = c(-130, 10),
                                             digits = 1))
```

```
CairoPNG(file = "isoscape.png",
         units = "in",
         width = 10,
         height = 8,
         pointsize = 24*92/72,
         dpi = 96)

plot.welcome.isoscape
```

```
dev.off()
```

```
## pdf
##   2
```

Saving IsoriX plots without Cairo

If you really don't manage to use **Cairo** or if you don't want to use it, you can always use the simple out-of-the-box alternative offered by R. You just need to use different function calls:

- `CairoPNG()` –> `png()`
- `CairoPDF()` –> `pdf()`

Nothing else changes, the inputs should be the same and you still need to close the graphic device using `dev.off()` after ploting.

Troubles with Cairo under MacOS?

One noticeable source of trouble comes with the installation of the package **Cairo**. On MacOS the package **Cairo** does not always install/load successfully. Problems happen when the program `xquartz` is not present on the system. This program used to be shipped by defaults on old version of MacOS but it is no longer the case. So if you are in trouble, please start by installing xquartz outside R.

The End

That is all for now! Here are the information of the R session we used:

```
sessionInfo()
```

```
## R version 3.4.0 (2017-04-21)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.5
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
```

```
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] raster_2.5-8 sp_1.2-5     Cairo_1.5-9  IsoriX_0.6   knitr_1.16
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.11        magrittr_1.5        rasterVis_0.41
##  [4] viridisLite_0.2.0   lattice_0.20-35     stringr_1.2.0
##  [7] tools_3.4.0         rgdal_1.2-8         parallel_3.4.0
## [10] grid_3.4.0          latticeExtra_0.6-28 htmltools_0.3.6
## [13] yaml_2.1.14         rprojroot_1.2       digest_0.6.12
## [16] RColorBrewer_1.1-2  evaluate_0.10.1     rmarkdown_1.6
## [19] stringi_1.1.5       compiler_3.4.0      backports_1.1.0
## [22] hexbin_1.27.1       zoo_1.8-0
```