nandb

An R package for performing number and brightness analysis as in Digman et al. 2008.

Installation

Platform Dependencies

This is slightly painful but you'll only have to do it once. First of all, if you're on **Ubuntu** (similarly for other debian **linux**), you need to do:

```
sudo apt-get update
sudo apt-get install libssl-dev libtiff5-dev libfftw3-dev
sudo apt-get install libcurl4-openssl-dev libxml2-dev
sudo apt-get install default-jre default-jdk libboost-all-dev
```

On **mac**, you need to go to https://support.apple.com/kb/dl1572?locale=en_US and download and install Java 6 (don't ask me why this is necessary). Then you need to install `Xcode` from the app store. Finally, you need to open a terminal and type `brew install boost`. If `brew install boost` doesn't work the google "install boost c++ library mac"; you need to have this done in order to use `nandb`.

On **Windows**, you need to go to https://cran.r-project.org/bin/windows/Rtools/ and install the latest version of Rtools.

All Platforms

Then, everyone, open R and run:

```
install.packages(c("devtools", "knitr", "rmarkdown",
                   "XML", "xml2", "gstat", "gridExtra"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("EBImage", "BiocParallel"))
```

If you get a message saying `Update all/some/none? [a/s/n]:`, type `a`. Then run

```
options(unzip = "internal")
devtools::install_github(paste0("rorynolan/",
                                c("filesstrings", "autothresholdr", "nandb")),
               auth_token = "50627d2ca9badc802ed993cb2b9a914034ab2246",
               build_vignettes = TRUE)
```

Done.

Problems

If you run into problems during your installation, it's most likely that your installation of `rJava` didn't work. Try running `install.packages("rJava")` and try to work through the errors there. If that still doesn't work, try googling "install rJava" for your operating system e.g. "install rJava Ubuntu". The second most likely culprit is `EBImage` so similarly have a google of "install EBImage". If you get these to work, then try the installation instructions again. If you still can't get it to work, feel free to contact me via the issues page associated with this repo.

Updates

To update the package, you just need to do the same thing:

```
options(unzip = "internal")
devtools::install_github(paste0("rorynolan/",
                                c("filesstrings", "autothresholdr", "nandb")),
                auth_token = "50627d2ca9badc802ed993cb2b9a914034ab2246",
                build_vignettes = TRUE)
```

To check if you need an update, check if the package has been updated since you installed it. To check your current version, use `packageVersion("nandb")`. To check if there's a newer version, go to the github page https://github.com/rorynolan/filesstrings (you're probably there right now) and look for the version in the DESCRIPTION file.

If you don't want to bother checking and you just want to make sure you have the latest version, just run those two lines of code.

Use

For the lowdown on how to use this package, you should read the package vignette (using `vignette("nandb")`); however, here's a quick example. First load the libraries:

```
library("nandb")
```

Then load the .tif file included with the package:

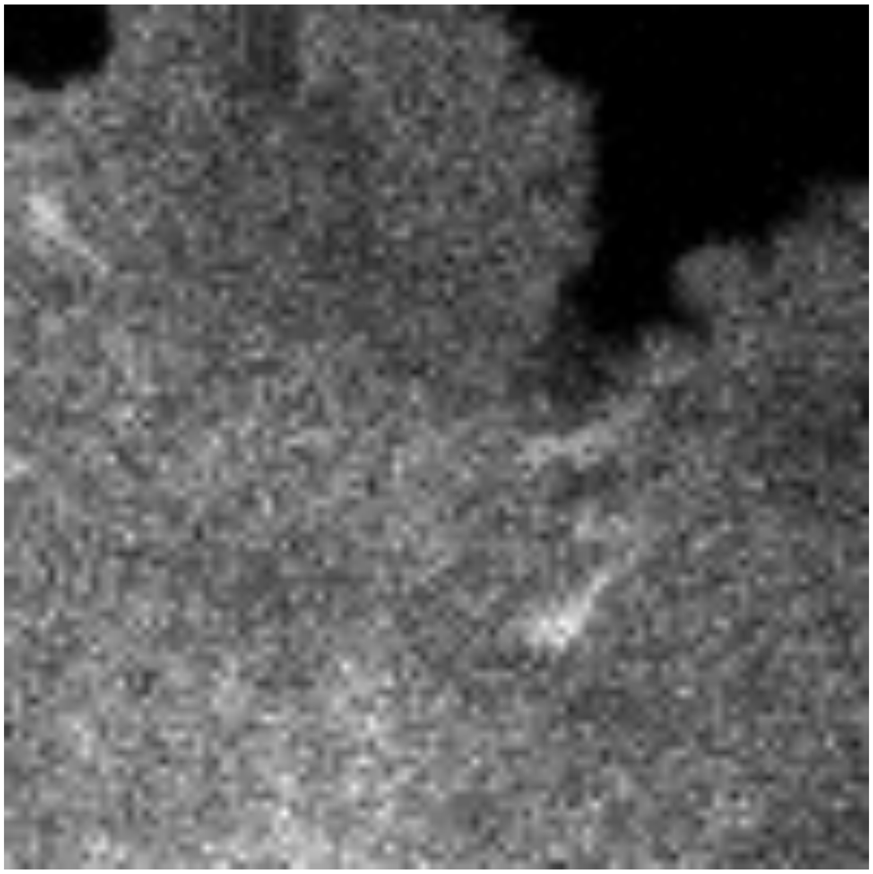
```
tif_file_location <- system.file("extdata", "50.tif", package = "nandb")
img <- ReadImageData(tif_file_location)
```

This `img` is a 3-dimensional array of image slices, where each element corresponds to a pixel with a value of integer counts. Looking at the dimensions of this object:

```
dim(img)
#> [1] 100 100  50
```

we can see that it is 50 slices each of which is a 100×100 image. We can view the first slice:

```
EBImage::display(EBImage::normalize(img[, , 1]), method = "raster")
```

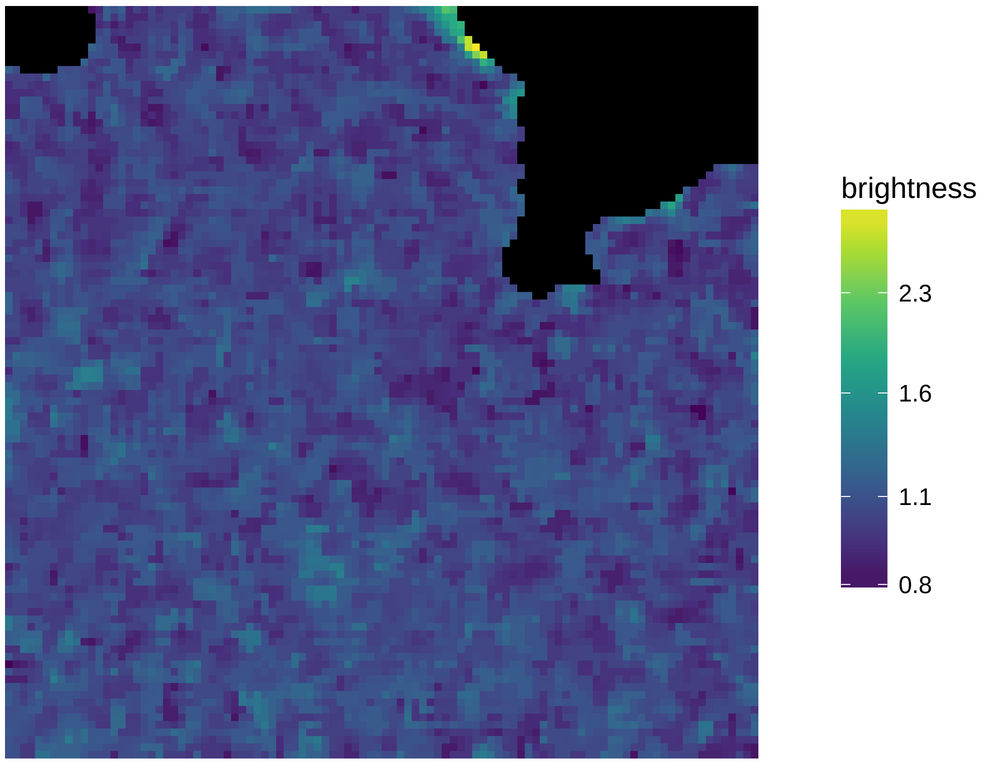


Now we can calculate the brightness image based on this image series, with an exponential filtering detrend with a time constant of (say) 10 frames via

```
img.brightness <- Brightness(img, tau = "auto", mst = "Huang", filt = "median")
```

Now we can see the (median filtered) version of the output.

```
MatrixRasterPlot(img.brightness, scale.name = "brightness", log.trans = TRUE)
```


brightness
2.3
1.6
1.1
0.8