

OptoGraphs: Creating .gif images for Optomotor and Optokinetic responses in larval fish.

Erin Frolli • Fuiman Lab
frolli.erin@utexas.edu
May 11, 2017

Contents

1. Introduction	1
2. OptoGraphs with ImageJ	1
3. OptoGraphs with ImageMagick	3

1. Introduction

Two sets of R functions were designed to create two simple assays for testing the Optomotor and optokinetic response of larval fish, Stripes assay and the Circular assay. The Stripes assay was created to play alongside a costume designed race way and the Circular assay was created to fit a 6 well plate.

This short vignette describes the two methods for creating a .gif image in R. With the first method, described in the OptoGraphs with ImageJ, several .png images are created and ImageJ to import all the .png images a .gif or .avi files can be created. The second method, described in OptoGraphs with ImageMagick, utilizes ImageMagick software by creating a .gif through the command line. Please note that each assay has code for both types of saving methods.

2. OptoGraphs with ImageJ

In this section we will describe how to make an animated .gif using the code for the Circular assay. Note that the steps are very similar with the Stripes assay (see the Example code for details). Make sure you have ImageJ already downloaded on your computer. You can get it for free here:

<https://imagej.en.softonic.com/download>

We will start by opening R. Then load OptoCirclesF_Nogif() function. Don't forget to use the correct file path if it is different from your working directory.

```
> # Note: write the file path to the folder that you have the R file saved in.  
> source(' C:/Users/Desktop/OptoCirclesF_Nogif.R')
```

Start by creating or designating a folder for the several images that will be outputted by this function. Then we will set our working directory to our specific folder.

```
> #Set your working directory to the parent folder.
```

```

> setwd('C:/Users/Desktop')
> # Note you can skip this next step if you have already created destination
  folders. You can continue onto the next line of code.
>
> # Create a destination folder
> dir.create("ClockwiseCircles") # Name this folder whatever you like
> dir.create("CounterClockwiseCircles") # Name this folder whatever you like
>
> # Then set our directory to the new folder
> setwd('C:/Users/Desktop/ClockwiseCircles')

```

Now you are ready to run the function. This will create a series of images in that designated folder that are mostly identical but slightly different in angle orientation.

```

> OptoCirclesF_Nogif(20) # Clockwise

```

This creates a series of images with 20 wedges of equal size and when you collate the images they will rotate clockwise. Note: You can change the number of wedges with any number, but if you chose an odd number it will make it an even number in order to maintain the color pattern.

Repeat the above steps if you want to create counter clockwise images also.

```

> # Then set our directory to the new folder
> setwd('C:/Users/Desktop/CounterClockwiseCircles')
> OptoCirclesF_Nogif(20, Reverse = TRUE) # Counter Clockwise

```

****** Note you can also change the colors of the graph in order to test color contrast or color vision.

```

> OptoCirclesF_Nogif(20, ColorsR= c("black", "grey"), Reverse = TRUE) # Counter
  Clockwise

```

Once you are happy with the images you have created its time to make the **.gif** or a video. We will do this in ImageJ. So open ImageJ. Then to upload the images we just created:

Go to File → Import → Image Sequence...

Navigate to the folder that has all the images in it. Then click on the first image in the sequence. It will have a number that looks like this: 0001plot.png

Next a popup window will open. Make sure the following are done:

Number of images: 36

Note for the stripes make sure this number matches the number of images you have in the folder.

Starting image: 1

Increment: 1

Scale images: 100%

Convert to RGB: is checked

Sort names numerically: is checked

Once all of the above are satisfied press the OK button. This should upload your images into a single window. You can press the play button to see your images in action as a sequence. To adjust the speed of the sequence:

Go to Images → Stacks → Tools → Animation options...

A new popup window will open. You can change the speed in the **Speed box**. This is number of frames per second. So the smaller the number the slower it goes and visa versa. Then press OK.

Once you are satisfied with your image you can save it in two formats. First format is a video format.

Go to File → Save As → AVI...

There will be a pop up box where you can adjust the compression and frame rate.

Compression: JPEG

Frame Rate: (Should be the same as your speed)

Then press OK. Then the save window will pop up. Rename and save your movie.

To save your images as a **.gif** file. First you need to change the image format:

Go to Image → Type.. → 8-bit color

Then save the file:

Go to File → Save As → Gif...

Then press OK. Then the save window will pop up. Rename and save your **.gif**.

3. OptioGraphs with ImageMagick

In this section we will describe how to make an animated **.gif** using the code for the Stripes assay. Note that the steps are very similar with the Circular assay (see the Example code for details). Note you will need ImageMagick an animation support package. If you don't already have ImageMagick you need to install it. We will go through the steps for downloading it for both Windows and Mac if you have done this already skip to the assay steps.

Windows

Start by opening R. This can be done on **PC computers** in R using the package installr:

```
# Install installr package & load it
> install.packages("installr")
> require(installr)
```

```
# Then install ImageMagick
> install.ImageMagick()
```

Mac OS

In order to install ImageMagick on a Mac you must have Homebrew or MacPorts installed. Here we describe the steps to download and use Homebrew.

Start by opening the **Mac Terminal**:

Finder → Go → Utilities → Terminal

Note make sure that you are logged in as the Administrator if applicable.

Type in the Terminal console:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Follow the prompts if it gives you any. To check if Homebrew is downloaded properly - you can test this by typing in **brew doctor** into the terminal console. This will pop up a summary report. Once you are satisfied that Homebrew is installed you can then download imageMagick. Type in the Terminal console:

```
brew install imagemagick
```

OptioGraph function

Follow any prompts as followed. Once imageMagick is installed open up R. Then open the OptoStripesF() function in the R console. Don't forget to use the correct file path if it is different from your working directory.

```
> file.edit('C:/Users/Desktop/OptoStripesF.R')
```

For **Windows** computers you need to find where ImageMagick downloaded in your file directory. Once you find it you need to update line 79 (for Stripes assay) or line 124 (for Circular assay) to match your file directory for the **convert function**. Then edit this line of code to match your file directory. Note only replace the area in **yellow**.

```
# Replace the yellow portion of line 79 to match your computer's directory
GifPath = paste('"C:/Program Files/ImageMagick-7.0.5-Q16/convert.exe" -delay
',Speed, ' *.png ',R,'Stripes',Barwidth,'_Speed',Speed,'.gif',sep = "")
```

For a Mac computer you will also need edit this line of code to only say the word **convert**. As shown below.

```
# Replace the yellow portion of line 79 to match your computer's directory
GifPath = paste('convert -delay ',Speed, ' *.png
',R,'Stripes',Barwidth,'_Speed',Speed,'.gif',sep = "")
```

Once you have the correct directory for ImageMagic make sure to source your new code file.

```
> # Note: write the file path to the folder that you have the R file saved in.  
> source('C:/Users/Desktop/OptoStripesF.R')
```

Again we will have to set our directory to a specific folder because the process will create several **.png** images before it creates the **.gif** image

```
> #Set your working directory to working folder  
> setwd('C:/Users/Desktop')  
> # Note you can skip this step if you have already created destination folders  
> # And just move onto the next line of code.  
>  
> # Create a destination folder  
> dir.create("Stripes") # Name this folder whatever you like  
>  
> # Then set our directory to the new folder  
> setwd('Stripes')
```

Now you are ready to run the function. This will create an animated **.gif** in that designated folder that will play on loop a series of strips moving across the image.

```
> # You can have the image movie either left or right  
> # Right handed .gif  
> OptoStripesF(20,10)  
> # Note the first variable is the bar width, the second variable is speed, and  
the third is reverse = FALSE.  
  
> # Left handed .gif  
> OptoStripesF(20,10,Reverse = TRUE)
```

Note: You can change the size of the stripes but the values have to be $0 < X \leq 50$. Also the speed variable the lower the value the faster the gif will animate. You can also change the colors of the graph in order to test color contrast or color vision.

```
> OptoStripesF(20,10,ColorSr= c("black","grey"),Reverse = TRUE) # left handed
```