Seed Color ImageJ Manual (unfinished version)

Ian McNish
University of Minnesota
last modified 15-Feb-2017

Contents

1 Introduction

This document is meant to serve as a manual for the ImageJ[1] script seed_color_imagej and R[2] script seed_color_r created by Ian McNish. This script was developed to collect color and size data from oat seed samples. This script was originally developed on an Apple MacOS version 10.11 system using Fiji version 2.0.0. Testing was also completed on a Windows system. In this manual I will explain how this script can be used to analyze seed photos using example photos available on . I will also attempt to explain how the script works so that other researchers can modify my code to fit their own needs.

2 Using ImageJ

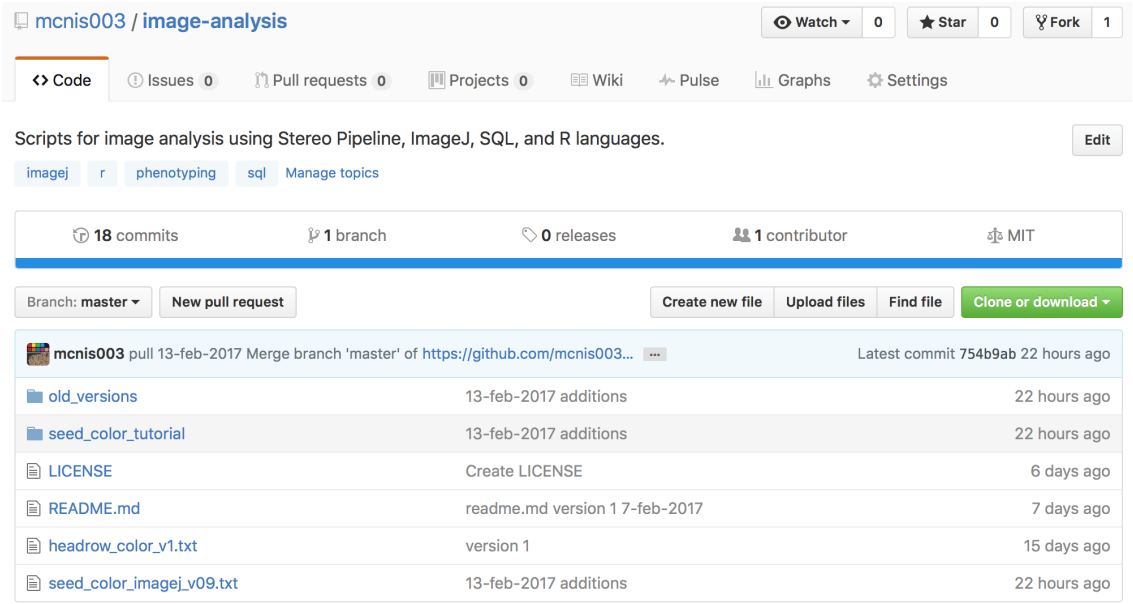
2.1 Downloading Fiji

I recommend using Fiji to run this script. Fiji can be found at this website `https://fiji.sc`. It is possible to use the script on a vanilla version of ImageJ, but it does require you to separately download the ImageJ macro 'Ellipse Split'. This is easily accomplished using the 'Update' feature in Fiji. Instructions for using using 'Ellipse Split' in ImageJ can be found at `http://imagej.net/Ellipse_split`.

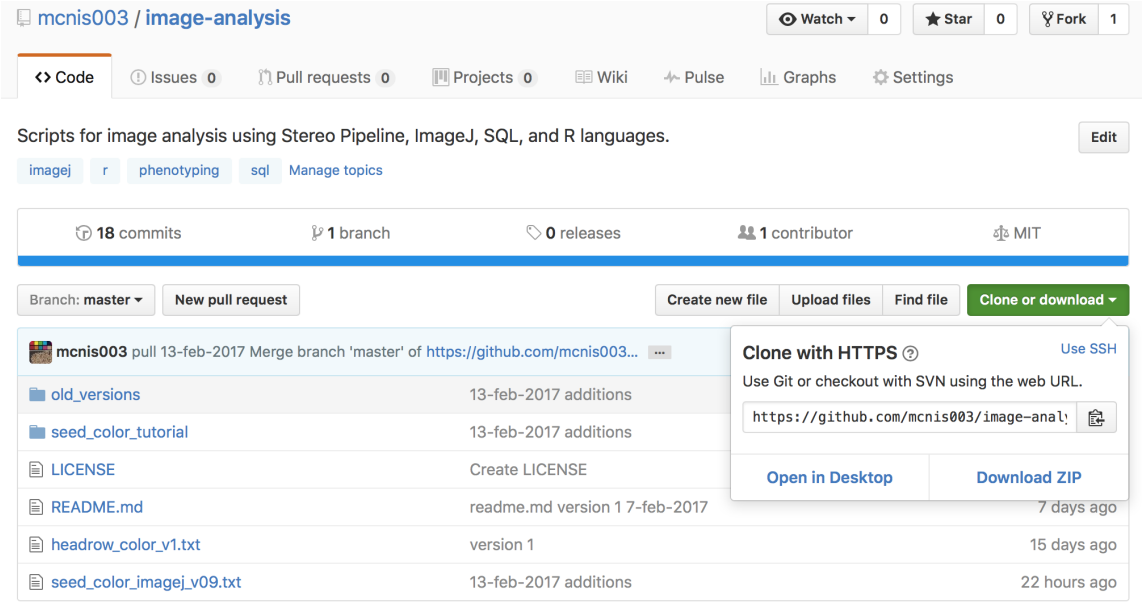
2.2 Downloading tutorial materials from GitHub

All the materials used in this tutorial can be found at `https://github.com/mcnis003/image-analysis/tree/master/seed_color_tutorial`. I will likely develop new versions of this script in the future. The newest version of these scripts can be found at `https://github.com/mcnis003/image-analysis.git`.

The materials can be downloaded directly from the website. First, navigate to the webpage provided above. Then, select the 'Clone or Download' button on the left side of the page.

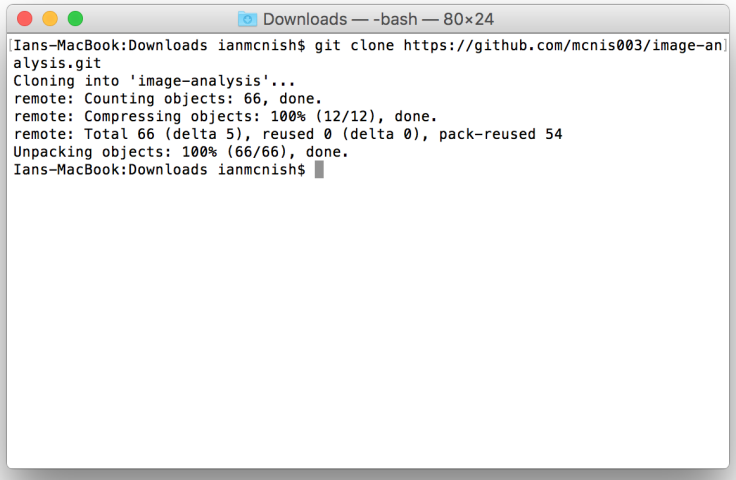


Then select the 'Download ZIP' button.



Alternatively you can also download the files using Terminal commands. A similar system is available on Windows, but I will not supply instructions here. First open . I normally do this using the 'Spotlight' feature in MacOS 10.11. You may need to refer to other sources to fully understand Terminal. I will only provide instructions on how to download materials from in this manual. While in an appropriate directory use the following commands:

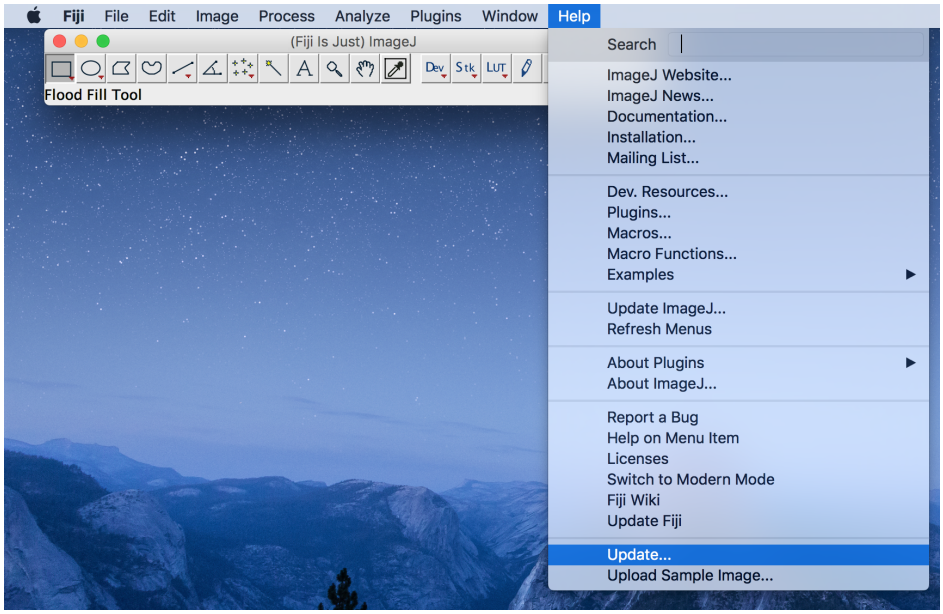
```
git clone https://github.com/mcnis003/image-analysis.git
```



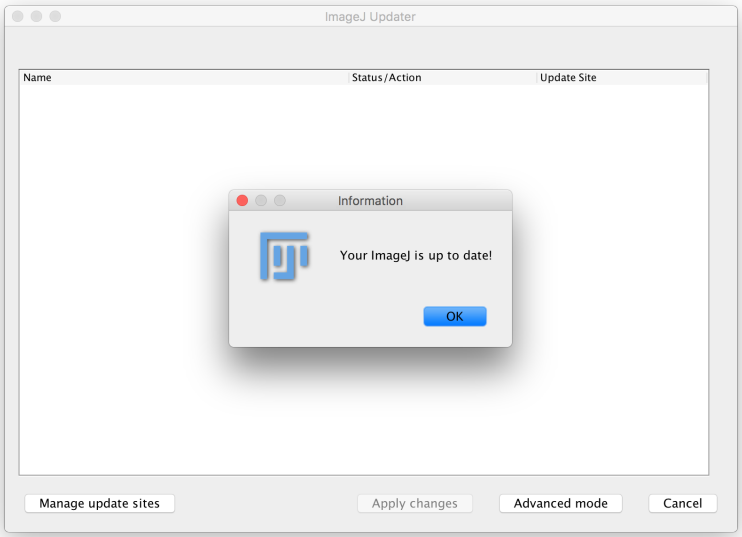
2.3 Downloading Ellipse Split

Downloading ‘’ is easiest when using Fiji. ‘Ellipse Split’ works in vanilla ImageJ but the installation process is slightly more calculated. I will not provide instructions for downloading ‘Ellipse Split’ for use in vanilla ImageJ.

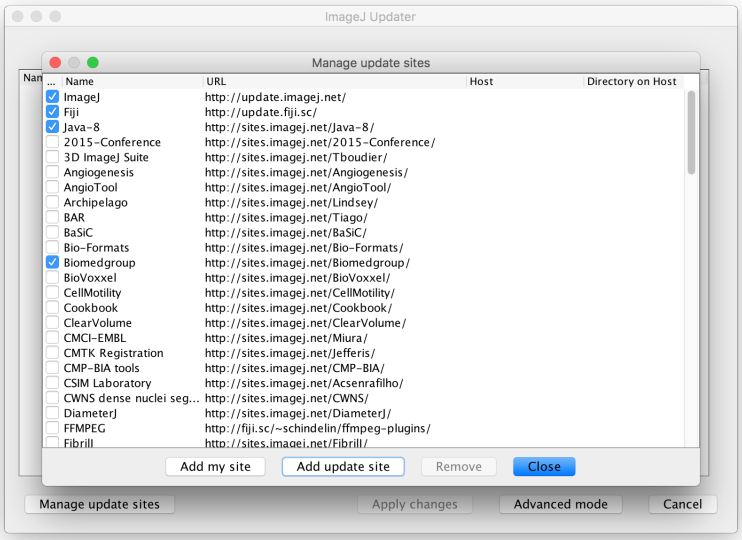
1. To download ‘Ellipse Split’, navigate the the Fiji ‘Help’ menu and select ‘Update...’



2. After Fiji has completed updating press ‘OK’. Then select ‘Manage update sites’.



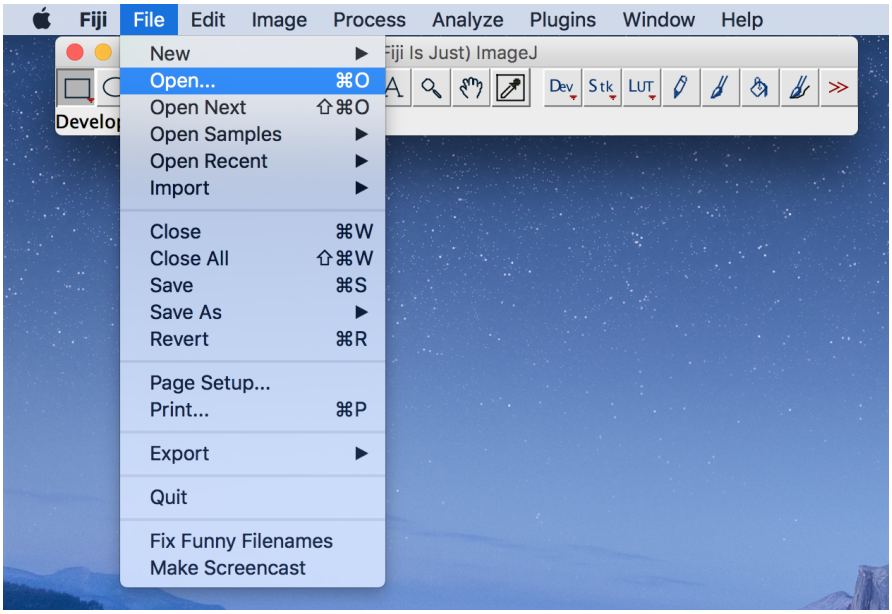
3. Select ‘’ from the list and select ‘Add update site’. ‘’ should then be available in you ‘Macro’ menu.



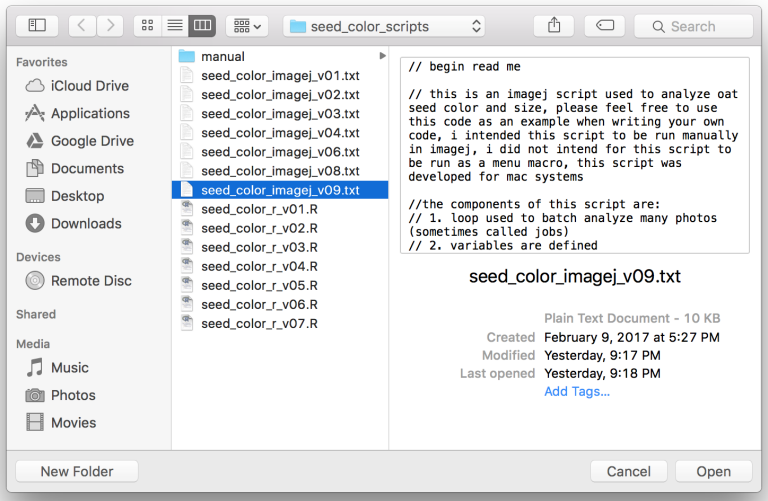
2.4 Open the script

Now that we have Fiji, Ellipse Split, and the materials from downloaded we are ready to open the script in . Do this through the open command in Fiji.

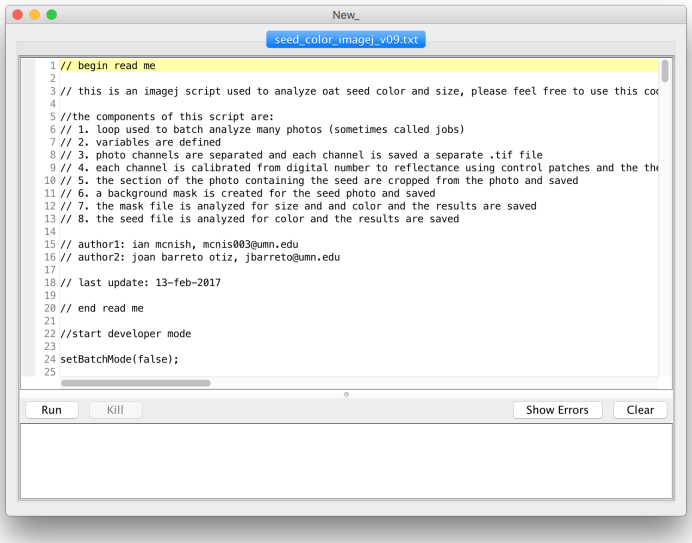
1. Select ‘Open’ in the ‘File’ menu.



2. Select the script you wish to open.



3. The script will then open in 's editing environment. The entire script may be run using the 'Run' button or by pressing 'command'+'R' on Mac. To run a portion of the code select the lines you wish to run a press 'command'+'shift'+'R'.



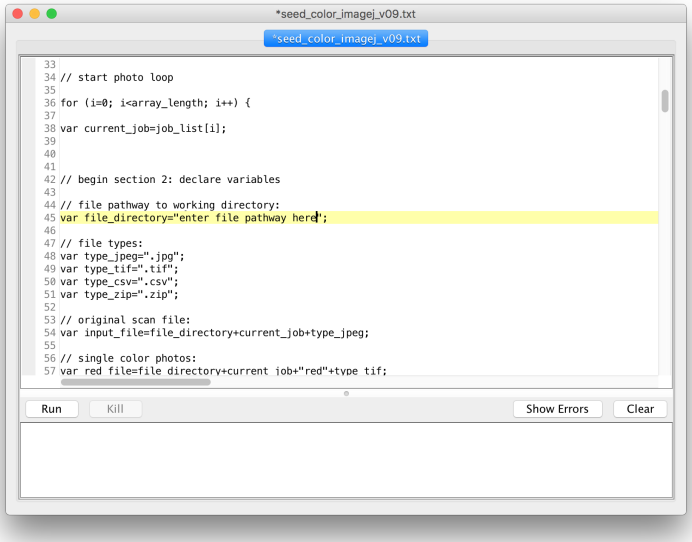
2.5 Setting parameters in ImageJ

Parameters other than the file directory should not need to be changed to run the Tutorial photos available on . However, you will definitely need to change the parameters for photos you collect. The

script works by dividing the image into separate zones. Each zone contains elements that are used for different processes. The position of the zones will depend on how you arrange your items in the photo, the equipment you use to take the photo, and the settings of your equipment. I suggest you test a couple different arrangements, and then strictly stick to the arrangement you think is most optimal.

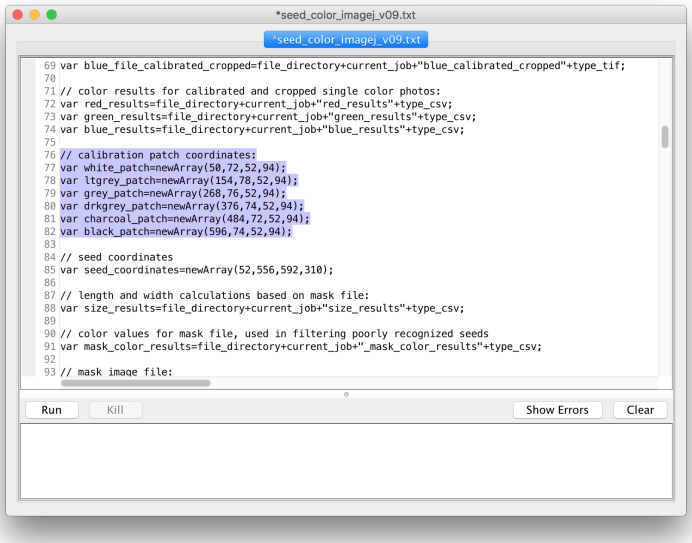
2.5.1 Choosing the file pathway

You will need to change the directory defined on line 45 of the script to the directory containing your photos. The result files will also be written to this directory. The behavior of this part of the script is similar to the setwd() command in R.



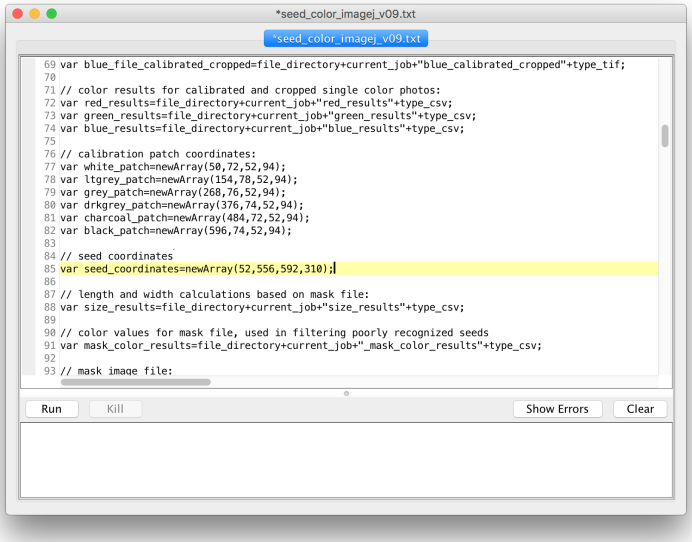
2.5.2 Enter control patch coordinates

The coordinate of you control patches must be entered in this part of the script. These must be entered as an array because each point in the array is specifically referenced later in the script. If you use more or less control patches you will need to modify this section and the calibration sections later in the script. You will need to determine if your application requires .



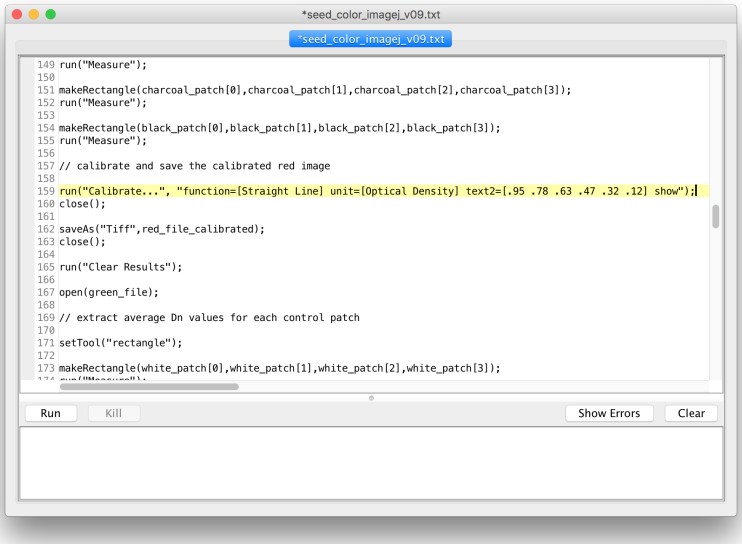
2.5.3 Enter seed coordinates

Enter the coordinates of the seeds here. Seeds that are on the edge of the region you draw will not be included in the analysis if you use ‘’ with the default settings from my script.



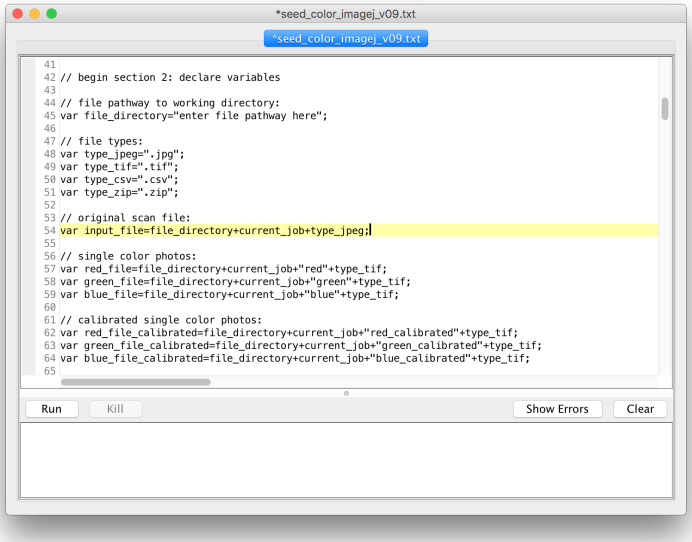
2.5.4 Enter control patch reflectance data

Enter the control patch expected reflectance data here. You will need to do this three times (lines 152, 193, and 227).



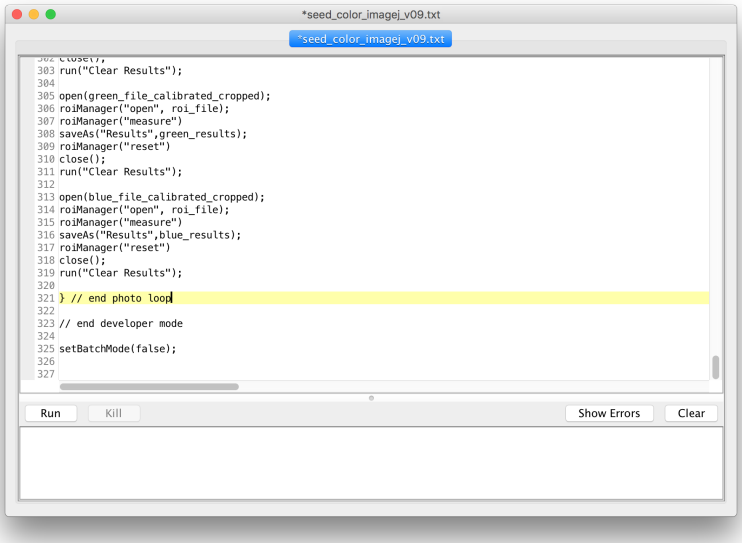
2.5.5 Changing initial photo file type

The Tutorial variable uses the .jpg file type to match the photos available on GitHub. The script has also been tested with TIFF format photos. You will need to change the extension here to match the type of photos you use here.

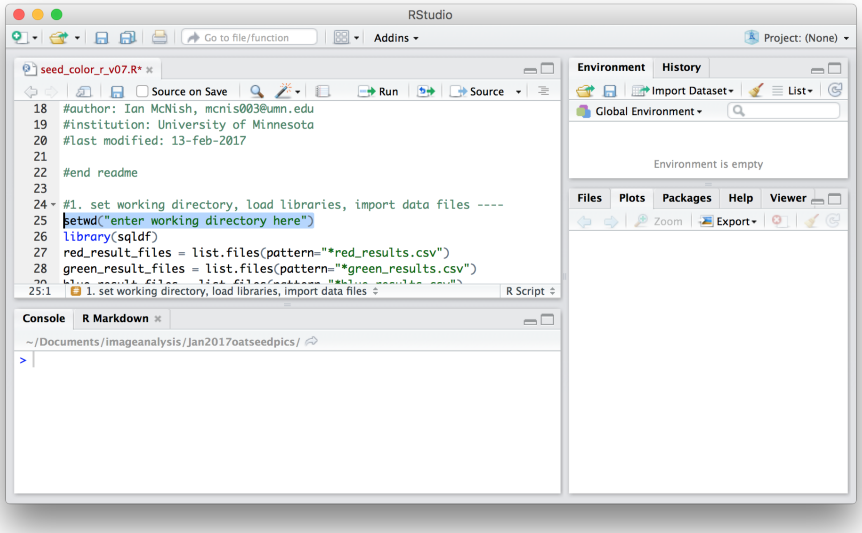


3 Running the script in Fiji

Once all your parameters are set you can run the script by pressing the 'Run' button in the development window or by pressing 'command'+'R' on Mac.



4 Using R



5 Tips for taking photos

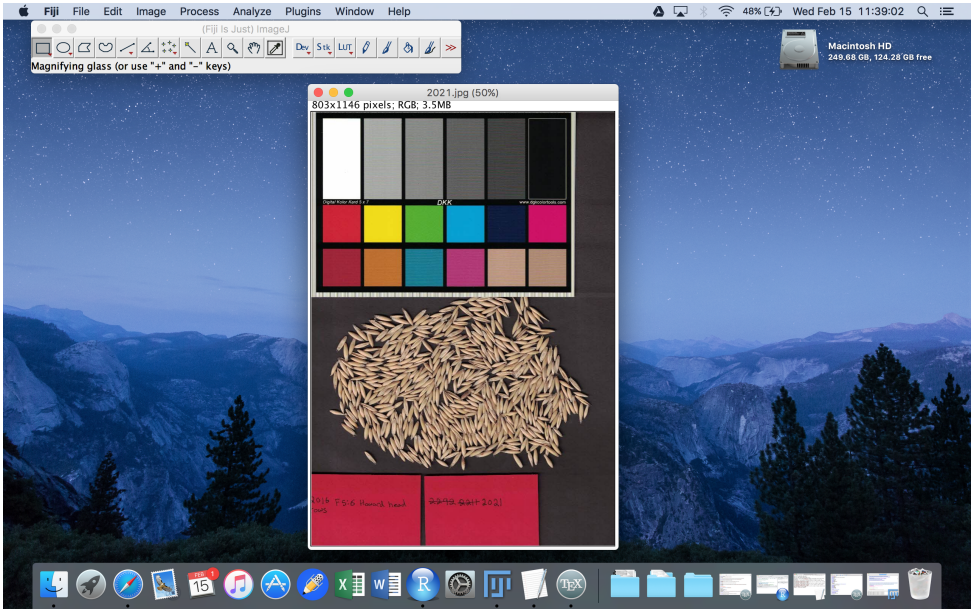
enter information about using this script

6 ImageJ process walkthrough

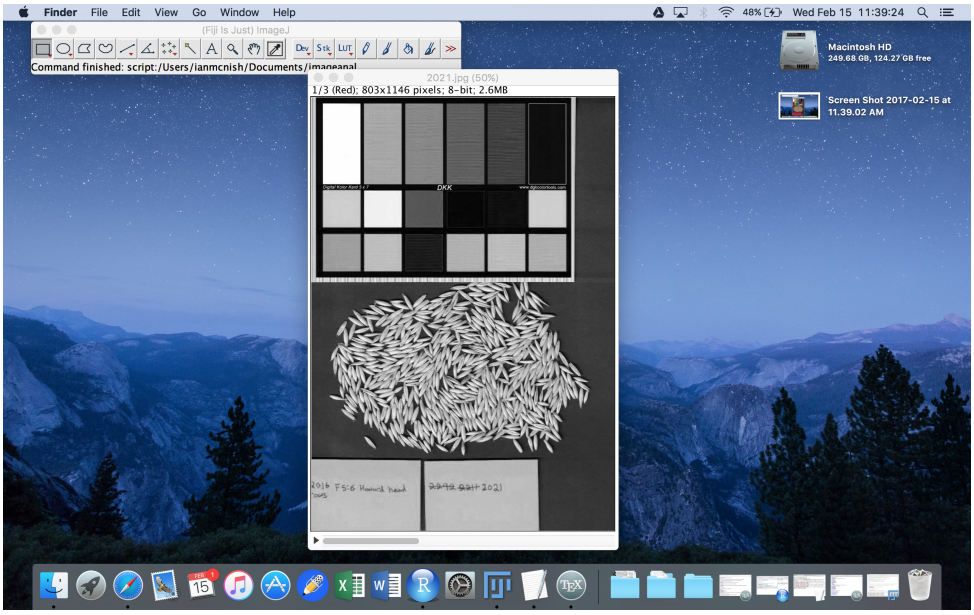
enter information about using this script

6.1 Section 3: importing the photo and splitting channels

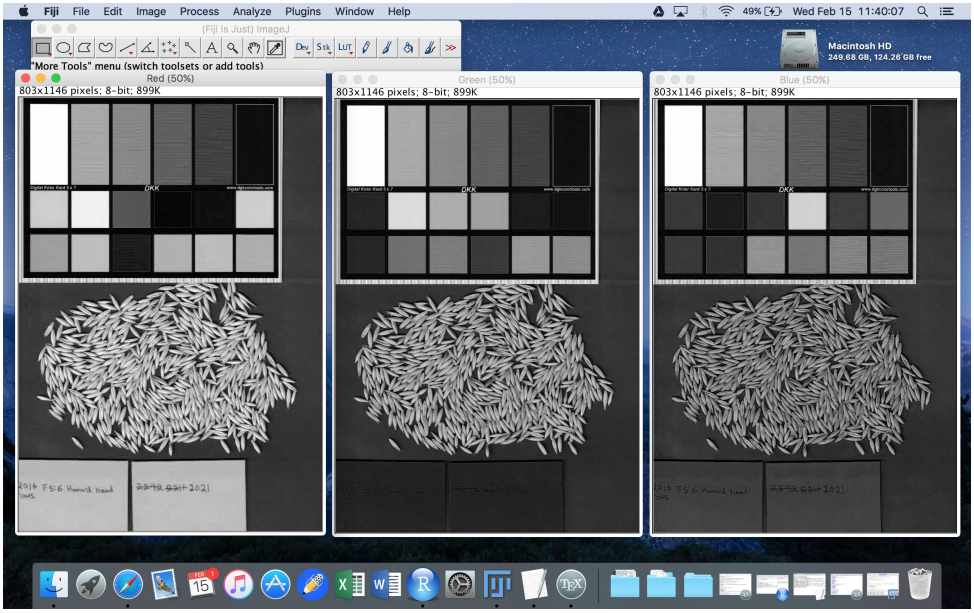
1. The full color photo is opened.



2. The photo is then split into R, G, and B channels in a stack.

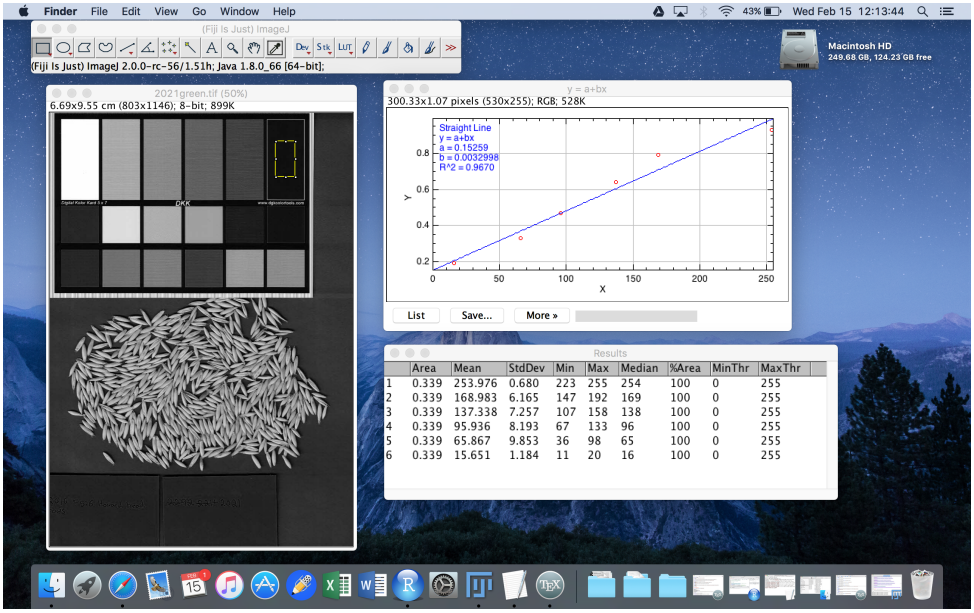


3. The stack of channels is then split into separate photos and each photo is saved. All photos are saved in TIFF format from this point onwards.



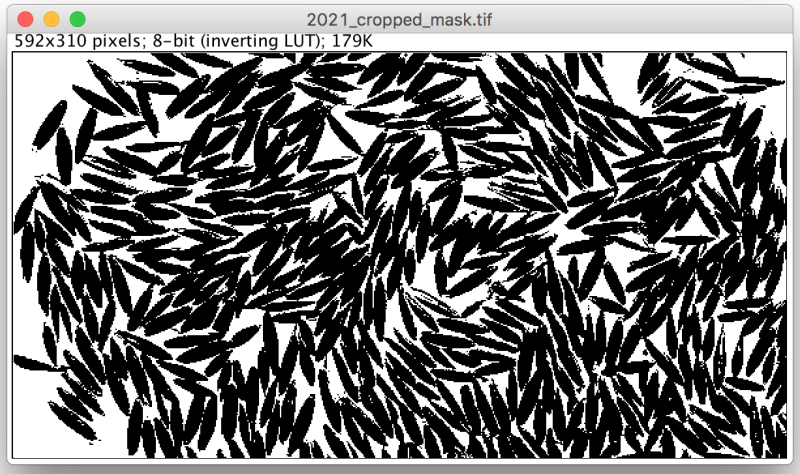
6.2 Section 4: radiometric calibration

Radiometric calibration is accomplished using a standard photo card placed in each photo combined with data collected with sensitive laboratory equipment. A detailed explanation of this method is provided by `http://ieeexplore.ieee.org/document/7098353/?reload=true`. Ultimately you will need to decide if you need to calibrate your photos. Radiometric calibration should be less important in circumstances with very controlled image acquisition. I provided the code in this script so that other researchers could use it if it is important in their process.



6.3 Section 6: background masking

For our application, background masking was not required. However, the mask is used by 'Ellipse Split' to produce the seed ROIs. I also use color data collected from the mask file to find instances where a seed ROI contains multiple seeds or large amounts of background. Masking may be necessary if your background is irregular.



6.4 Section 7: using Ellipse Split to create seed ROIs

Because our seeds are ellipse shaped, I used the ImageJ macro '' to find ROIs. If you are analyzing objects of another shape this method of locating the object will probably not work. For other shapes of object I recommend using the '' command.

7 FAQ

1. Q: how long does it take to analyze a photo with this program?
   A: Our application uses a 30mb photo with 100 objects and it takes 4 seconds per photo. The program is much faster if batch mode is enabled, but unfortunately I have been having problems with batch mode lately.

2. Q: How accurate is the script at finding seeds? A: In our experience, if items in the photo are well arranged the script will accurately find 85% of the seeds, 5% of the seeds will be misidentified in some way, and 10% of the seeds will not be identified.

8 Bibliography

[1] Schneider, C. A., Rasband, W. S., and W, E. K. Nih image to imagej: 25 years of image analysis. *Nature Methods 9*, 7 (2012), 671–675.

[2] Team, R. D. C. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

Index