

Seed Color ImageJ Manual (unfinished version)

Ian McNish
University of Minnesota
last modified 22-Feb-2017

Contents

1	Introduction	1
2	Using ImageJ	1
2.1	Downloading Fiji	1
2.2	Downloading tutorial materials from GitHub	1
2.3	Downloading Ellipse Split	3
2.4	Open the script	4
2.5	Setting parameters in ImageJ	6
2.5.1	Using cleanup mode	6
2.5.2	Choosing the file pathway	7
2.5.3	Enter control patch coordinates	7
2.5.4	Enter seed coordinates	8
2.5.5	Enter control patch reflectance data	8
2.5.6	Changing initial photo file type	9
3	Running the script in Fiji	10
4	Using R	10
5	Tips for taking photos	10
6	ImageJ process walkthrough	11
6.1	Section 3: importing the photo and splitting channels	11
6.2	Section 4: radiometric calibration	12
6.3	Section 6: background masking	13
6.4	Section 7: using Ellipse Split to create seed ROIs	13
7	FAQ	14
8	Bibliography	14
Index		15

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.

The screenshot shows a GitHub repository page for the user 'mcnis003' with the repository name 'image-analysis'. At the top, there are buttons for 'Watch' (0), 'Star' (0), and 'Fork' (1). Below the header, there are tabs for 'Code' (selected), 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. A description below the tabs reads 'Scripts for image analysis using Stereo Pipeline, ImageJ, SQL, and R languages.' There are buttons for 'Edit', 'imagej', 'r', 'phenotyping', 'sql', and 'Manage topics'. Below this, there are statistics: 18 commits, 1 branch, 0 releases, 1 contributor, and MIT license. A 'Clone or download' button is prominently displayed. The main area lists files and commits:

File/Commit	Date	Time Ago
mcnis003 pull 13-feb-2017 Merge branch 'master' of https://github.com/mcnis003...	13-feb-2017	Latest commit 754b9ab 22 hours ago
old_versions	13-feb-2017	additions 22 hours ago
seed_color_tutorial	13-feb-2017	additions 22 hours ago
LICENSE	Create LICENSE	6 days ago
README.md	readme.md version 1 7-feb-2017	7 days ago
headrow_color_v1.txt	version 1	15 days ago
seed_color_imagej_v09.txt	13-feb-2017 additions	22 hours ago

Then select the 'Download ZIP' button.

mcnis003 / image-analysis

Code Issues Pull requests Projects Wiki Pulse Graphs Settings

Scripts for image analysis using Stereo Pipeline, ImageJ, SQL, and R languages.

imagej r phenotyping sql Manage topics

18 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

old_versions	13-feb-2017 additions	Clone with HTTPS Use SSH	
seed_color_tutorial	13-feb-2017 additions	https://github.com/mcnis003/image-analysis	
LICENSE	Create LICENSE	Open in Desktop	Download ZIP
README.md	readme.md version 1 7-feb-2017	7 days ago	
headrow_color_v1.txt	version 1	15 days ago	
seed_color_imagej_v09.txt	13-feb-2017 additions	22 hours ago	

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:

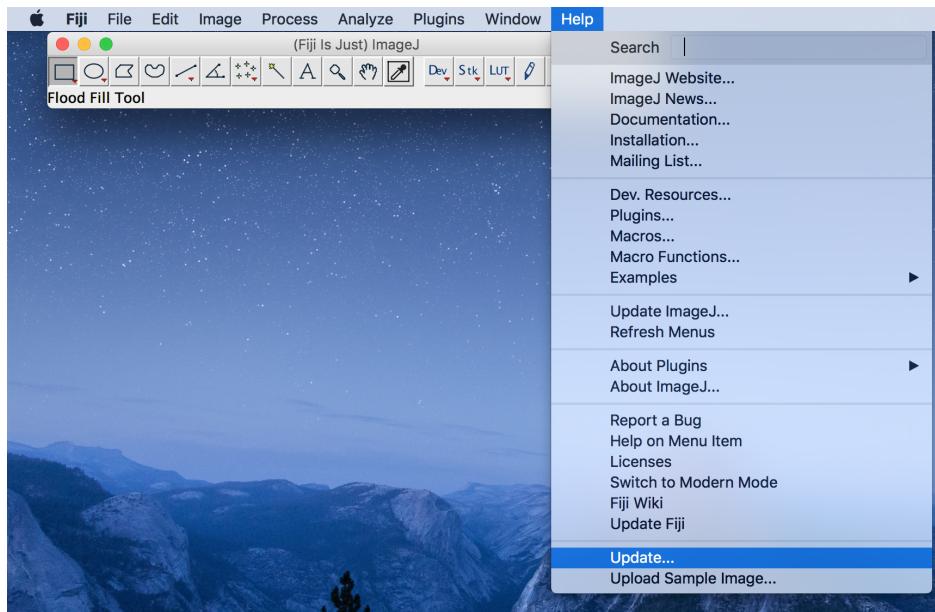
```
git clone https://github.com/mcnis003/image-analysis.git
```



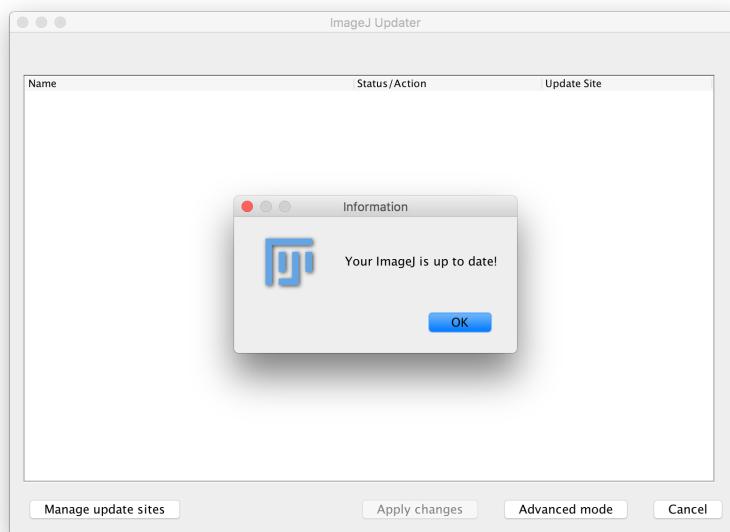
2.3 Downloading Ellipse Split

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.

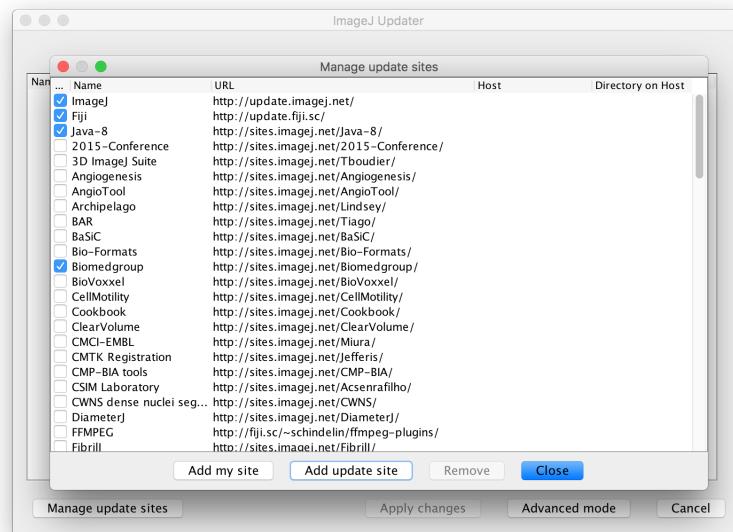
1. To download ‘Ellipse Split’, navigate the the Fiji ‘Help’ menu and select ‘Update...’



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



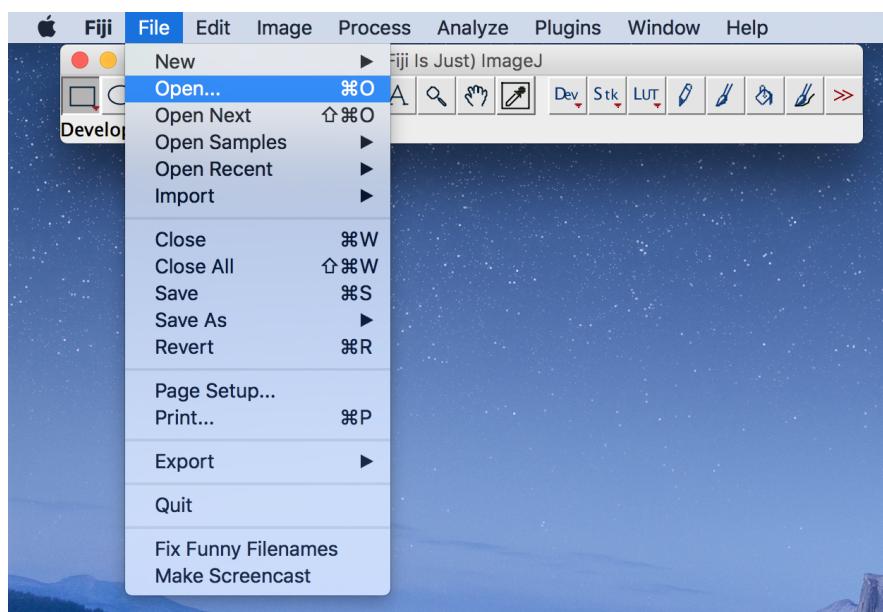
3. Select ‘’ from the list and select ‘Add update site’. ‘’ should then be available in you ‘Macro’ menu.



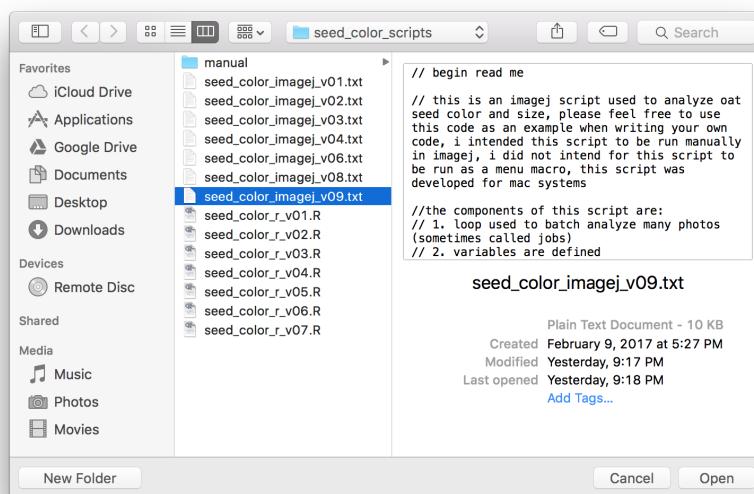
2.4 Open the script

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.

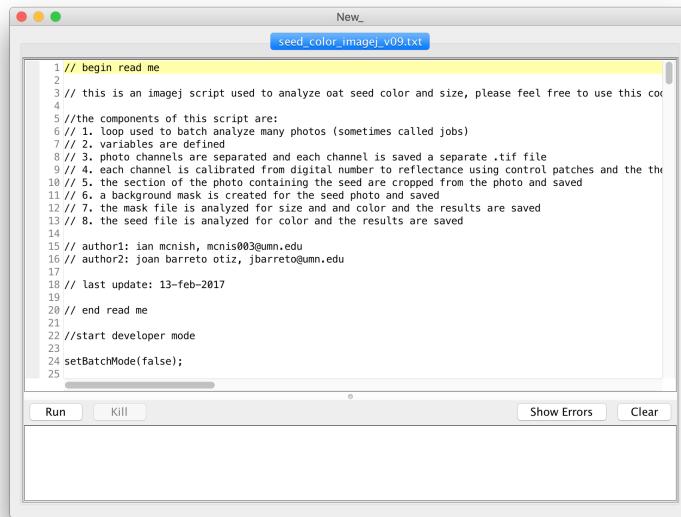
1. Select ‘Open’ in the ‘File’ menu.



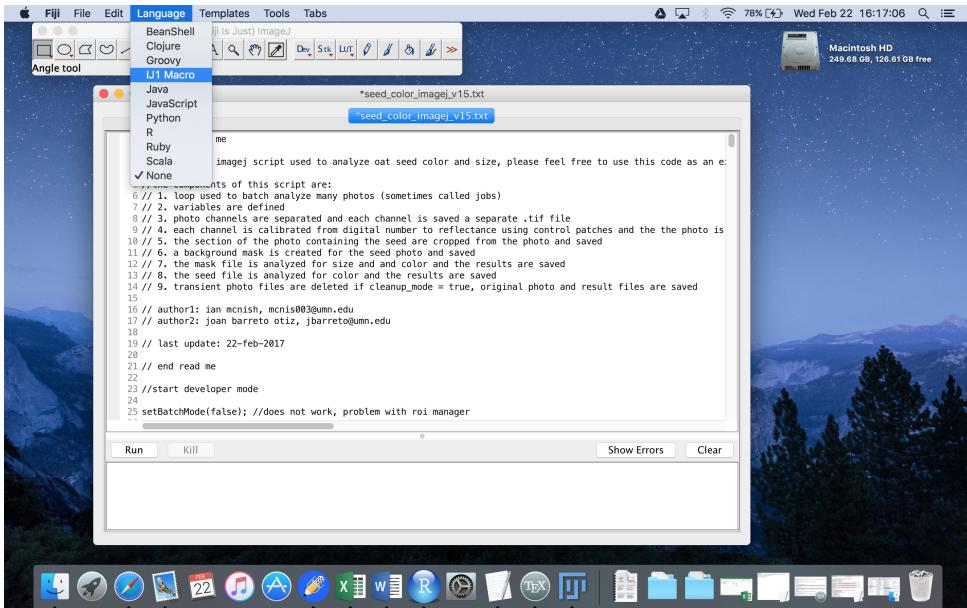
- Select the script you wish to open.



- 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’.



- Finally, the language of the script must be selected. Choose ‘IJ1 Macro’ from the ‘Language’ menu.



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 Using cleanup mode

The cleanup variable can be set to either ‘true’ or ‘false’. Cleanup mode is enabled when the cleanup variable equals ‘true’. This script creates many intermediate files when processing photos. These files can occupy a large amount of space. When cleanup mode is enabled, the script will delete many of these files. The only files that will not be deleted are the original photo files and the data result files. As a safety precaution the delete function in ImageJ will only function if the file pathway includes a directory named “ImageJ”. Cleanup mode is probably not useful when you are first learning to use the script, or when you are troubleshooting.

```

36
37 for (i=0; i<array_length; i++) {
38
39 var current_job=job_list[i];
40
41 // begin section 2: declare variables
42
43 //set cleanup mode, if = TRUE then the script will delete several of the intermediate files at the end of the
44
45 var cleanup_mode = true;
46
47 // file pathway to working directory:
48 var file_directory="enter file pathway here";
49
50 // file types:
51 var type_jpg=".jpg";
52 var type_tif=".tif";
53 var type_csv=".csv";
54 var type_zip=".zip";
55
56 // original scan file:
57 var input_file=file_directory+current_job+type_jpg;
58
59 // single color photos:
60 var red_file=file_directory+current_job+"red"+type_tif;
61 var green_file=file_directory+current_job+"green"+type_tif;
62 var blue_file=file_directory+current_job+"blue"+type_tif;

```

2.5.2 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.

```

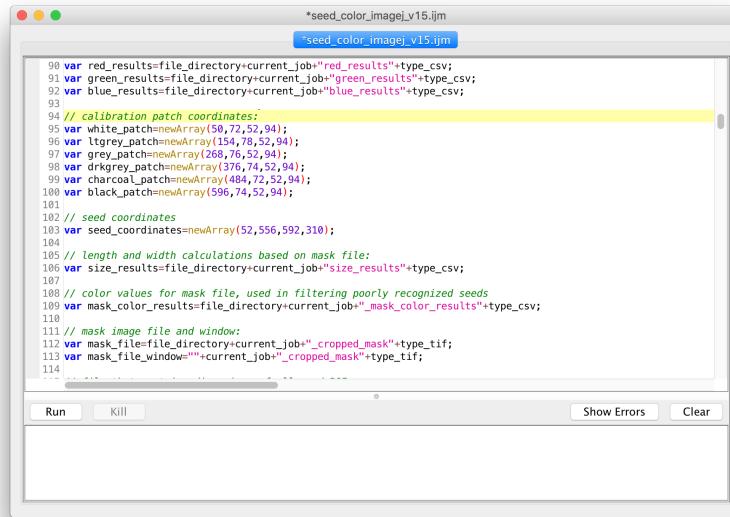
36
37 for (i=0; i<array_length; i++) {
38
39 var current_job=job_list[i];
40
41 // begin section 2: declare variables
42
43 //set cleanup mode, if = TRUE then the script will delete several of the intermediate files at the end of the
44
45 var cleanup_mode = true;
46
47 // file pathway to working directory:
48 var file_directory="enter file pathway here";
49
50 // file types:
51 var type_jpg=".jpg";
52 var type_tif=".tif";
53 var type_csv=".csv";
54 var type_zip=".zip";
55
56 // original scan file:
57 var input_file=file_directory+current_job+type_jpg;
58
59 // single color photos:
60 var red_file=file_directory+current_job+"red"+type_tif;
61 var green_file=file_directory+current_job+"green"+type_tif;
62 var blue_file=file_directory+current_job+"blue"+type_tif;

```

2.5.3 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

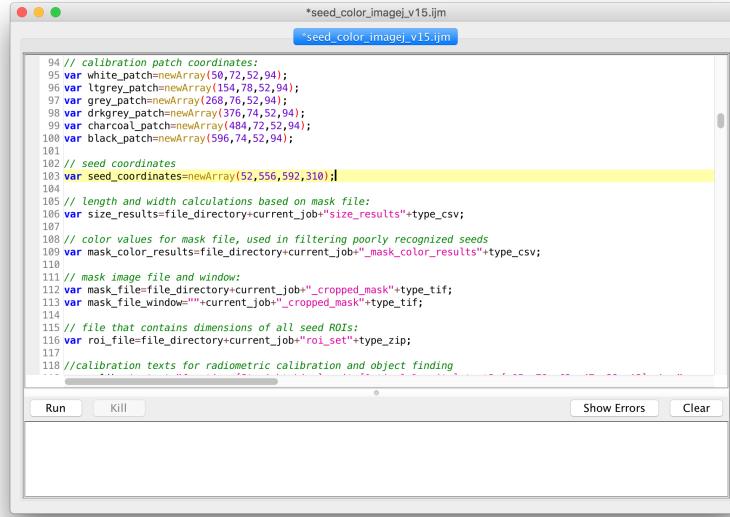
script. You will need to determine if your application requires .



```
*seed_color_imagej_v15.ijm
90 var red_results=file_directory+current_job+"red_results"+type_csv;
91 var green_results=file_directory+current_job+"green_results"+type_csv;
92 var blue_results=file_directory+current_job+"blue_results"+type_csv;
93
94 // calibration patch coordinates:
95 var white_patch=newArray(50,72,52,94);
96 var ltgrey_patch=newArray(154,78,52,94);
97 var grey_patch=newArray(268,76,52,94);
98 var drkgrey_patch=newArray(376,74,52,94);
99 var charcoal_patch=newArray(484,72,52,94);
100 var black_patch=newArray(596,74,52,94);
101
102 // seed coordinates
103 var seed_coordinates=newArray(52,556,592,310);
104
105 // length and width calculations based on mask file:
106 var size_results=file_directory+current_job+"size_results"+type_csv;
107
108 // color values for mask file, used in filtering poorly recognized seeds
109 var mask_color_results=file_directory+current_job+"_mask_color_results"+type_csv;
110
111 // mask image file and window:
112 var mask_file=file_directory+current_job+"_cropped_mask"+type_tif;
113 var mask_file_window=""+current_job+"_cropped_mask"+type_tif;
114
```

2.5.4 Enter seed coordinates

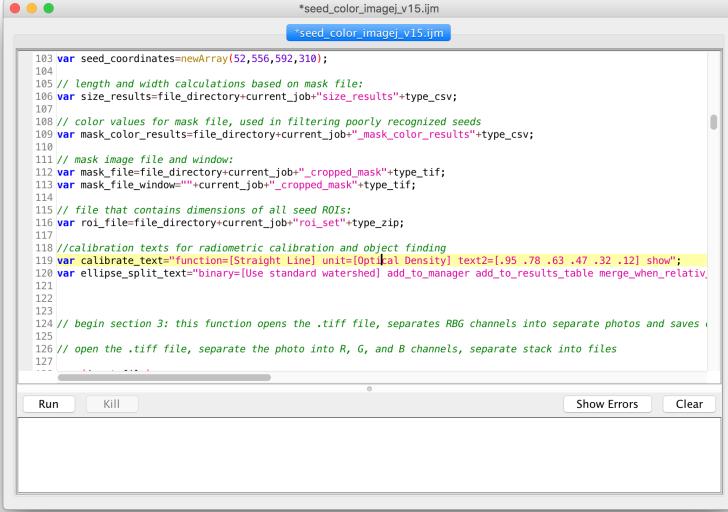
Enter the coordinates of the seeds here. Seeds that are on the edge of the region you draw will not be included in the analysis if you use ‘’ with the default settings from my script.



```
*seed_color_imagej_v15.ijm
90 var red_results=file_directory+current_job+"red_results"+type_csv;
91 var green_results=file_directory+current_job+"green_results"+type_csv;
92 var blue_results=file_directory+current_job+"blue_results"+type_csv;
93
94 // calibration patch coordinates:
95 var white_patch=newArray(50,72,52,94);
96 var ltgrey_patch=newArray(154,78,52,94);
97 var grey_patch=newArray(268,76,52,94);
98 var drkgrey_patch=newArray(376,74,52,94);
99 var charcoal_patch=newArray(484,72,52,94);
100 var black_patch=newArray(596,74,52,94);
101
102 // seed coordinates
103 var seed_coordinates=newArray(52,556,592,310);l
104
105 // length and width calculations based on mask file:
106 var size_results=file_directory+current_job+"size_results"+type_csv;
107
108 // color values for mask file, used in filtering poorly recognized seeds
109 var mask_color_results=file_directory+current_job+"_mask_color_results"+type_csv;
110
111 // mask image file and window:
112 var mask_file=file_directory+current_job+"_cropped_mask"+type_tif;
113 var mask_file_window=""+current_job+"_cropped_mask"+type_tif;
114
115 // file that contains dimensions of all seed ROIs:
116 var roi_file=file_directory+current_job+"roi_set"+type_zip;
117
118 //calibration texts for radiometric calibration and object finding
```

2.5.5 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).



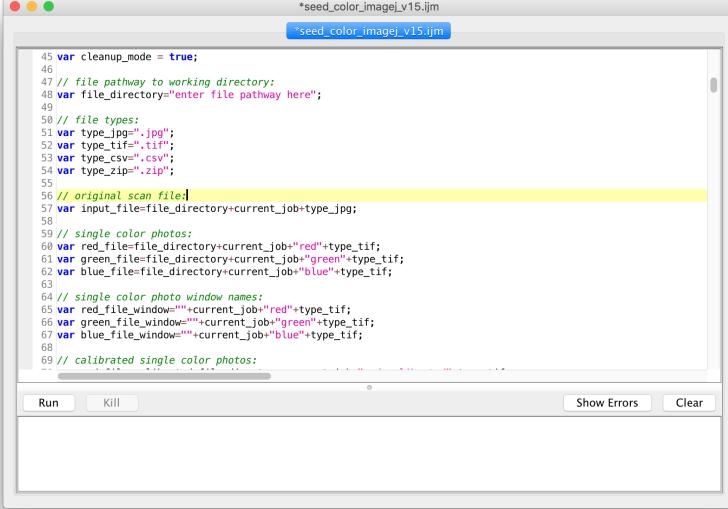
```

103 var seed_coordinates=newArray(52,556,592,310);
104
105 // length and width calculations based on mask file;
106 var size_results=file_directory+current_job+"size_results"+type_csv;
107
108 // color values for mask file, used in filtering poorly recognized seeds
109 var mask_color_results=file_directory+current_job+"_mask_color_results"+type_csv;
110
111 // mask image file and window:
112 var mask_file=file_directory+current_job+"_cropped_mask"+type_tif;
113 var mask_file_window=""+current_job+"_cropped_mask"+type_tif;
114
115 // file that contains dimensions of all seed ROIs;
116 var roi_file=file_directory+current_job+"roi_set"+type_zip;
117
118 //calibration texts for radiometric calibration and object finding
119 var calibrate_text="function=[Straight Line] unit=[Optical Density] text2=[.95 .78 .63 .47 .32 .12] show";
120 var ellipse_split_text="binary=[Use standard watershed] add_to_manager add_to_results_table merge_when_relativ,
121
122
123
124 // begin section 3: this function opens the .tiff file, separates RGB channels into separate photos and saves ,
125
126 // open the .tiff file, separate the photo into R, G, and B channels, separate stack into files
127

```

2.5.6 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.



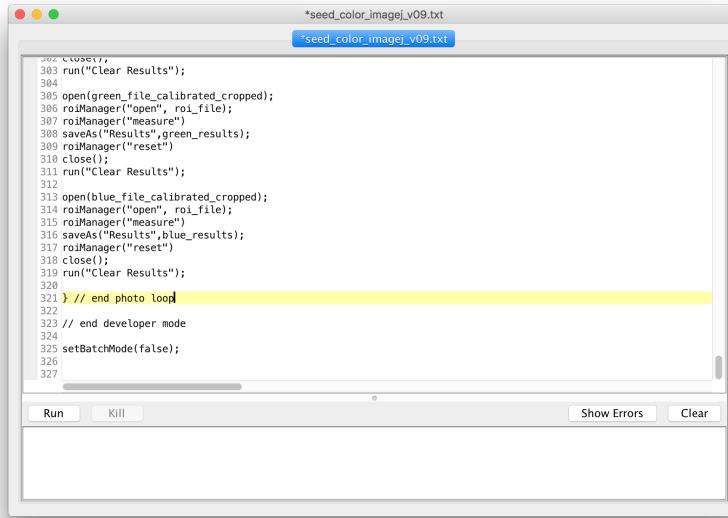
```

45 var cleanup_mode = true;
46
47 // file pathway to working directory;
48 var file_directory="enter file pathway here";
49
50 // file types:
51 var type_jpg=".jpg";
52 var type_tif=".tif";
53 var type_csv=".csv";
54 var type_zip=".zip";
55
56 // original scan files
57 var input_file=file_directory+current_job+type_jpg;
58
59 // single color photos:
60 var red_file=file_directory+current_job+"red"+type_tif;
61 var green_file=file_directory+current_job+"green"+type_tif;
62 var blue_file_directory=current_job+"blue"+type_tif;
63
64 // single color photo window names:
65 var red_file_window=""+current_job+"red"+type_tif;
66 var green_file_window=""+current_job+"green"+type_tif;
67 var blue_file_window=""+current_job+"blue"+type_tif;
68
69 // calibrated single color photos:

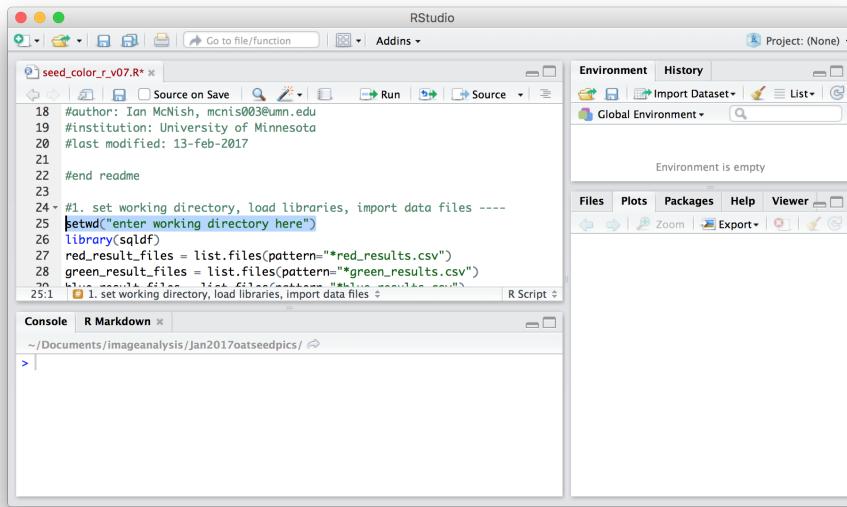
```

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

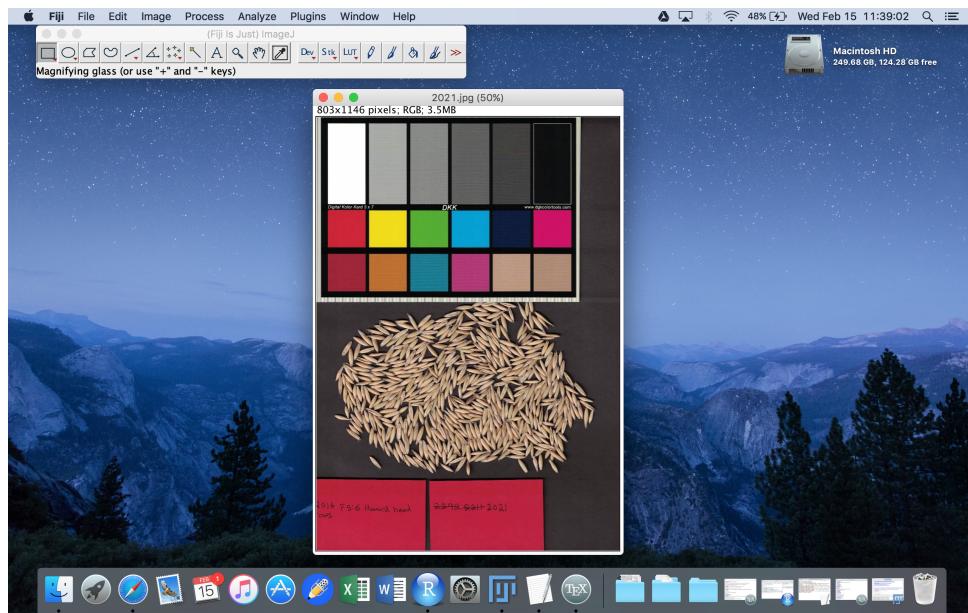
enter information about using this script

6 ImageJ process walkthrough

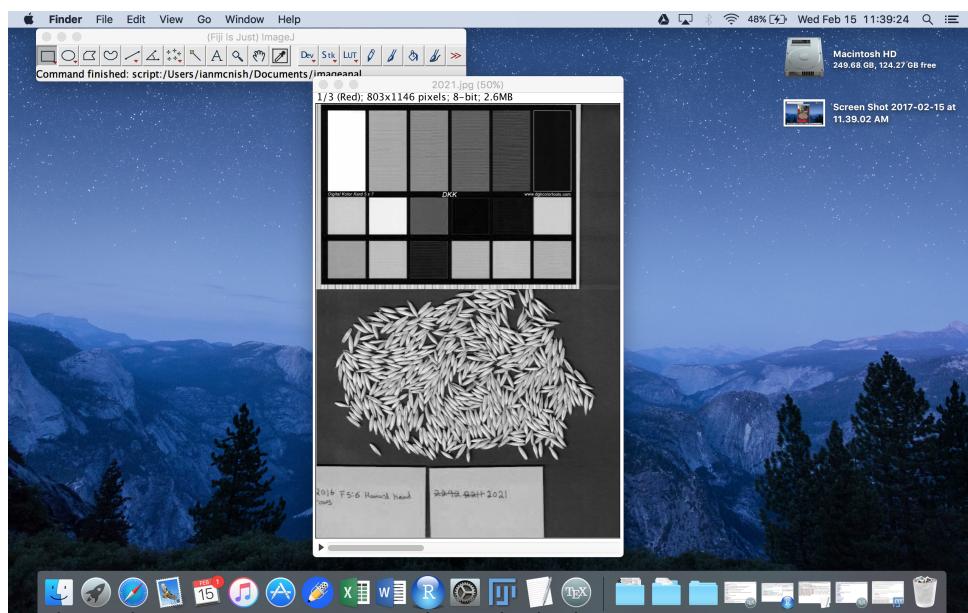
enter information about using this script

6.1 Section 3: importing the photo and splitting channels

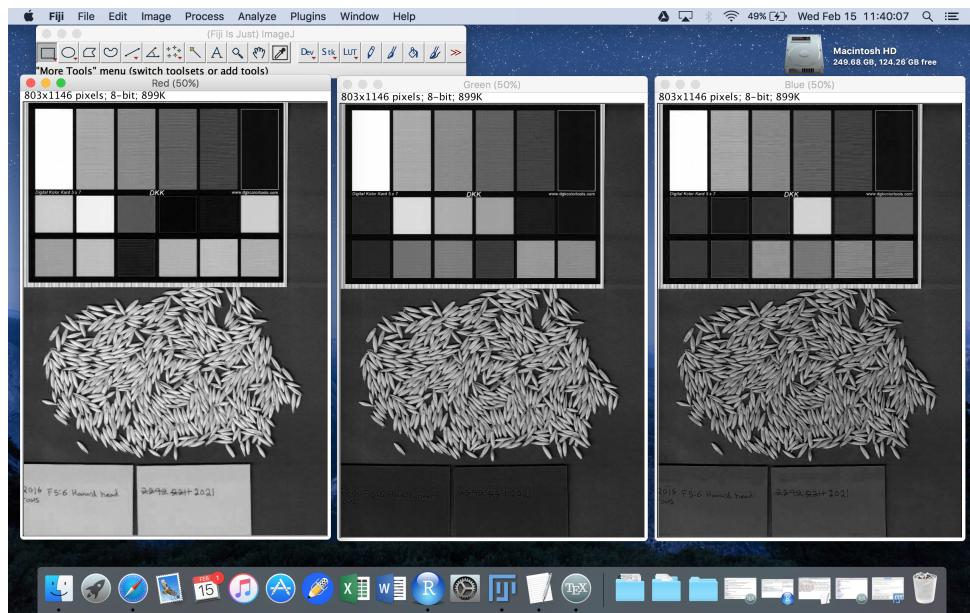
1. The full color photo is opened.



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

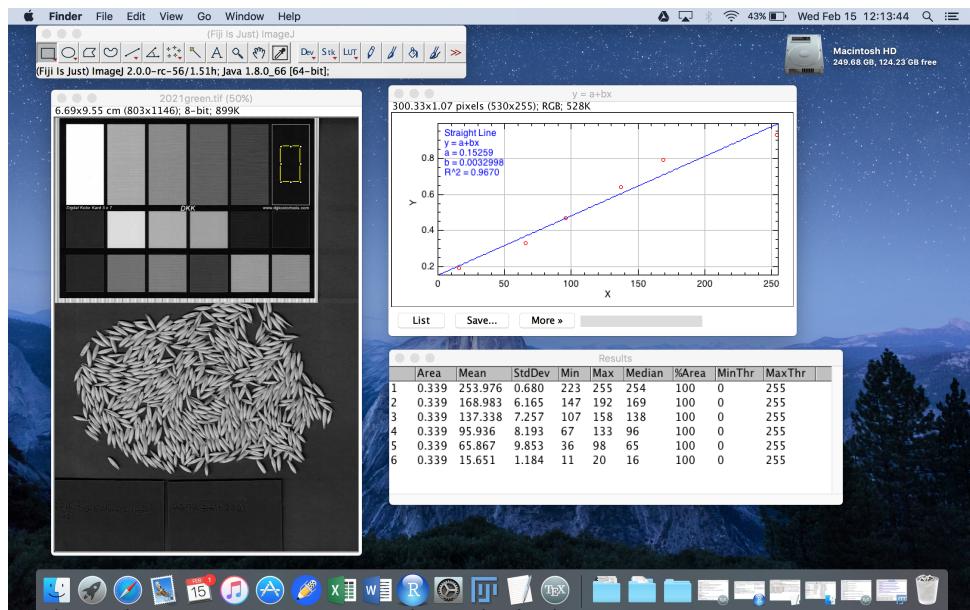


3. The stack of channels is then split into separate photos and each photo is saved. All photos are saved in TIFF format from this point onwards.



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.



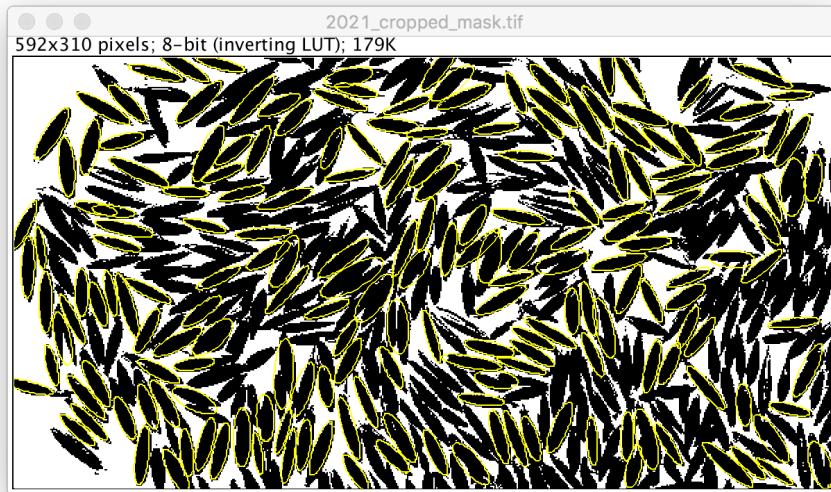
6.3 Section 6: background masking

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 “`Elliptical Selection`” 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 “`Find Maxima`” 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

Analyze Particles, 12
Biomedgroup, 4
Ellipse Split, 3, 4, 7, 12
Fiji, 4, 5
GitHub, 1, 2, 4, 5
radiometric calibration, 6
Terminal, 2