CONVERTING IMAGEJ OUTPUT TO LOCAL PLOT COORDINATES USING R

R. CONDIT

SHORT_TITLE: ImageJ END_SHORT_TITLE

DESCRIPTION: This tutorial demonstrates use of an R program designed to read all quadrat maps created by imageJ, converting the digitized pixel to coordinates within a quadrat (local coordinates). The commands shown here lead step by step through the process. The input imageJ files must be named in a very specific way, and there must be 4 separate files for each quadrat, in the case of 10x10 m sections of a single 20x20 m quadrat.

This tutorial is also available for download as a pdf from `https://ctfs.arnarb.harvard.edu/Public/CTFSRPackage/tutorials/imageJTutorial/imageJ.pdf`. END_DESCRIPTION

1. Directory and file naming

All files must be stored in one folder, and that overarching folder can be given any name. You need to give the entire path of this folder as a parameter to the function. The map files may all be saved in one folder, divided into column subfolders, or even into quadrat sub-subfolders. In a laptop running Unix, for example, the maps may be in the folder '/home/condit/data/other_plots/rabi'. On a Windows system, the entire path would be something like 'c:/data/other_plots' etc. A Mac uses :: instead of slashes. In the following examples, I assign this folder to a variable called *mapfolder*. You will have to name *mapfolder* to the name appropriate for your computer.

We suggest you save the files using the folder structure detailed below, which makes it easy to find a particular map or quadrat. It is not absolutely necessary to use this structure, but you do need to use the following file naming convention: 1. you have to use the same prefix for all the resulting quadrat files 2. they should all use the same delimiter (comma- or tab-delimited) 3. the quadrat names have to have 4 digits (i.e. pad with zeros for single digit columns or rows) 4. the subquadrat suffixes should be named consistently (i.e. the first suffix refers to the lower left subquadrat, second suffix to the upper left subquadrat, third suffix to the upper right subquadrat, and the fourth to the lower right subquadrat)

Within *mapfolder*, we suggest you make one folder for each column of data. These folders can be named 'Column##', for example 'Column01', 'Column02', etc. Within every one of the Column folders, there can be one subfolder per quadrat, and each can be named Quadrat_##, for example Quadrat_0101, Quadrat_2309, etc. Within those quadrat subfolders, there should be four imageJ files named Map_####_#.txt, for example Map_0101_1.txt, corresponding to the maps of the four subquadrats. The first number (usually 4 digits) is the quadrat number, matching the number for the Quadrat folder it is in. The second number may be a 1, 2, 3, or 4, indicating which of the 4 subsections of the map is included in the file.

Date: June 27, 2011.

mapfolder

- Column01
  - Quadrat_0101
    - Map_0101_1.txt
    - Map_0101_2.txt
    - Map_0101_3.txt
    - Map_0101_4.txt
  - Quadrat_0102
  - Quadrat_0103
  - Quadrat_0104
  - Quadrat_0105
    - Map_0105_1.txt
    - Map_0105_2.txt
    - Map_0105_3.txt
    - Map_0105_4.txt
  - etc
- Column02
  - Quadrat_0201
    - Map_0201_1.txt
    - Map_0201_2.txt
    - Map_0201_3.txt
    - Map_0201_4.txt
  - Quadrat_0202
  - Quadrat_0203
  - Quadrat_0204
  - Quadrat_0205
  - etc
- etc

2. EXECUTING

Download and Source the CTFS R Package following instructions at the home page.

The conversion program is called *fullplot.imageJ.* As a test, run it first with just one folder by setting the path to that folder. The other arguments are:

- Colrange specifies the range of the columns and rowrange the range of the rows, as per the first two digits and last two digits respectively, of our quadrat naming convention
- Gridsize is the size of each individual map.
- When include.subdir=F, only the folder specified in the path parameter is searched for map files. If include.subdir=T, all subfolders under the folder specified in the path parameter are searched also.
- Four arguments name map corners; they must match the labels in the imageJ output for the four corners of each map. They must be identical for every map, else the program cannot work. It does not matter where these four rows appear in the file; they can be any rows and in any order.
- The corner labels are caps-insensitive, so the labels p1 and P1 will be considered identical.
- The prefix used to name the map files should be consistent. It can be Map_ or just plain q, for example.
- The subquadrat suffixes should be consistent, and named clockwise from the lower left subquadrat.
- Setting outfile=NULL means the data will NOT be written to a file, only returned as the R object named coords.col02.

In the example below, I set a variable *mapfolder* to the name of the directory where one of the column folders are found. You will have to set this to the folder on your computer.

```
> mapfolder='/home/laoz/maps/rabi/Column02/'
> coords.col02=
+   fullplot.imageJ(path=mapfolder,include.subdir=T,gridsize=c(20,20),outfile=NULL,corners=c('p
+                   prefix='q',colrange=c(0,49), rowrange=c(0,24),subquadsuffix=c('_1','_2','_3
Finished calculating coordinates for quadrat  0200
Finished calculating coordinates for quadrat  0201
Finished calculating coordinates for quadrat  0202
Finished calculating coordinates for quadrat  0203
Finished calculating coordinates for quadrat  0204
```

```
> head(coords.col02)
      tag        lx        ly quadrat
5  119417  0.391076  7.533889    0200
6  119418  3.568018  5.440052    0200
7  119419  5.915727 13.869409    0200
8  534491  9.041064 11.469232    0200
9  119392 10.718618 10.435609    0200
10 119393 12.041607 11.839611    0200
```

```
> dim(coords.col02)
[1] 67  4
```

If that works, then execute from the rabi folder. Every column will be read and converted. The program will read all files within the Quadrat subfolders.

```
> mapfolder='/home/laoz/maps/rabi/'
> coords=
+   fullplot.imageJ(path=mapfolder,include.subdir=T,gridsize=c(20,20),outfile=NULL,corners=c('p
+                   prefix='q',colrange=c(0,49), rowrange=c(0,24),subquadsuffix=c('_1','_2','_3
Finished calculating coordinates for quadrat  0001
Finished calculating coordinates for quadrat  0002
Finished calculating coordinates for quadrat  0004
Finished calculating coordinates for quadrat  0100
Finished calculating coordinates for quadrat  0101
Finished calculating coordinates for quadrat  0102
Finished calculating coordinates for quadrat  0103
Finished calculating coordinates for quadrat  0104
Finished calculating coordinates for quadrat  0200
Finished calculating coordinates for quadrat  0201
Finished calculating coordinates for quadrat  0202
Finished calculating coordinates for quadrat  0203
Finished calculating coordinates for quadrat  0204
Finished calculating coordinates for quadrat  0300
Finished calculating coordinates for quadrat  0301
Finished calculating coordinates for quadrat  0302
Finished calculating coordinates for quadrat  0303
Finished calculating coordinates for quadrat  0304
Finished calculating coordinates for quadrat  0400
Finished calculating coordinates for quadrat  0402
Finished calculating coordinates for quadrat  0403
Finished calculating coordinates for quadrat  0404
> head(coords)
      tag        lx        ly quadrat
5  634459  4.407679 11.355755    0001
51 119111  2.672999 12.058920    0002
52 119112  3.330483  2.417399    0004
53 119139  9.873418 19.237597    0100
6  119140 16.278481 14.443066    0100
54 119301  2.541646  7.389927    0101
> dim(coords)
[1] 114   4
> range(coords$lx)
[1]  0.1251708 19.7453001
> range(coords$ly)
[1]  0.1511969 19.9252093
```

All the values should be between 0 and 20. If this works, you can repeat, this time saving the results to a text file. The file will be saved into the folder specified in the path. The four columns of this file can then be uploaded into the tempLocations table in the database via a simple SQL script.

```
> coords=
+   fullplot.imageJ(path=mapfolder,include.subdir=T,gridsize=c(20,20),outfile='locations.txt',c
+                   prefix='q',colrange=c(0,49), rowrange=c(0,24),subquadsuffix=c('_1','_2','_3
Finished calculating coordinates for quadrat  0001
Finished calculating coordinates for quadrat  0002
Finished calculating coordinates for quadrat  0004
Finished calculating coordinates for quadrat  0100
Finished calculating coordinates for quadrat  0101
Finished calculating coordinates for quadrat  0102
Finished calculating coordinates for quadrat  0103
Finished calculating coordinates for quadrat  0104
Finished calculating coordinates for quadrat  0200
Finished calculating coordinates for quadrat  0201
Finished calculating coordinates for quadrat  0202
Finished calculating coordinates for quadrat  0203
Finished calculating coordinates for quadrat  0204
Finished calculating coordinates for quadrat  0300
Finished calculating coordinates for quadrat  0301
Finished calculating coordinates for quadrat  0302
Finished calculating coordinates for quadrat  0303
Finished calculating coordinates for quadrat  0304
Finished calculating coordinates for quadrat  0400
Finished calculating coordinates for quadrat  0402
Finished calculating coordinates for quadrat  0403
Finished calculating coordinates for quadrat  0404
```

3. Sweaving

These are the commands used to create the HTML help page; they are not involved with imageJ.

```
> Sweave(file="~/programs/R/gis/imageJ.Rnw",keep.source=TRUE)
> system("htlatex imageJ")
> system("pdflatex imageJ")
> system("mv imageJ* ~/programs/R/gis/imageJTutorial/")
> system("mv ~/programs/R/gis/imageJTutorial/imageJ.html ~/programs/R/gis/imageJTutorial/index.
```