Stripflow was written by Gang Dong, in collaboration with K. Palaniappan (University of Missouri Columbia) and Tobias Baskin (University of Massachusetts Amherst), under the auspices of NIH grant award EB00573-01A1. Address questions to Baskin@umass.edu. If you publish results based on using Stripflow, please cite Yang et al. 2016 Plant Cell and Environment.

Stripflow estimates the velocity profile along a growing root from two input images, separated by a user defined interval, usually around 60 sec. The name, Stripflow, derives from the fact that velocity is estimated for a narrow strip, usually as wide as the root and with a user defined length. Details of image analysis are provided in the accompanying file (Stripflow\_ principles.pdf). Velocity is reported as the component of motion locally tangent to the midline, and in a frame of reference where a point at the root tip is chosen as the origin. Thus, this point has a position and distance of zero and all motion is referred to this point. Although in principle any point could be used, for our work with *Arabidopsis thaliana* we use the quiescent center.

The program is written in MatLab. Stripflow comprises the following files:

estTransKeren.m

shift\_fft.m

stripflow\_estimator.m

read\_input\_v1\_0.m

read\_singlestack\_input\_v1\_0.m

The program is launched by running either of the read input files. The difference between them is as follows. If the entire root growth zone fits in the field of view, then use read\_singlestack\_input\_v1\_0 (hence ‘single-stack’ version). For higher resolution work, the root growth zone is imaged in a series of overlapping frames. In that case, use ‘read\_input\_v10 (hence ‘multi-stack’ version). Note that the single-stack version is mainly the multi-stack version with the features for multiple overlapping frames commented out.

A working folder contains a directory named ‘tiff’ and a file named input.txt. The former contains the images and the latter contains information that Stripflow needs to run.

**SINGLE STACK VERSION**

The tiff directory will contain two images (here, 4\_15-00.tif and 4\_15-01.tif).

The input.txt file will read (material in [] provided for explanation, delete in real file):

1 [Direction of growth: 0=right to left, 1=left to right. Roots must oriented horizontally in the image.]

0 [Always 0 for single-stack version.]

1 [Always 1 for single-stack version.]

1.85 [Calibration. In this example, 1 µm = 1.85 pixel.]

2 [Always 2 for single-stack version.]

[The blank line is required.]

stack1 [Begins provision of geometric parameters; see below.]

26 358, 500 358, 1000 356, 1500 364, 2000 368, 2324 370 [Coordinates in pixels of the midline.]

dlr 340 340 [Diameter in pixels of the root at left and right boundary.]

endm [Ends the provision of geometric parameters.]

60 [Time between images, in seconds]

2324 370 [Coordinates of the zero point of the velocity profile; usually the quiescent center.]

15-4-inag-1.txt [File name of the output data file. {*file name stem*}.txt, see below.]

4\_15-00.tif [File name of the first input image.]

4\_15-01.tif [File name of the second input image.]

As evident, the input.txt file provides certain coordinates. The convention here is that the origin (0, 0) is at the *lower left corner* of the image. Thus, y-values increase going upwards.

Although tedious, we find best results by providing midline points by hand. These are measured prior to running Stripflow, for example with ImageJ. There needs to be ~20 pixels overhang so the endpoints cannot be at the very edges of the image. For single-stack version, either the initial or final point is usually the quiescent center but this is not obligatory. When there are extraneous objects abutting the root, these can be masked out by setting the diameter less than that of the root.

When read\_singlestack\_input\_v1\_0 is opened, the user needs to specify the working folder (where the images and input.txt file are located). This is done by editing line 8, where the folder path is specified. In addition the user might wish to adjust the strip-width (line 5). When the program runs, an image of root with midline points and midline will be shown, as will the created velocity profile, and three files will be written to the working folder (specified in line 8):

{file name stem}-midline.png

{file name stem}-velo.png

{file name stem}.txt

The first file is an image where the midline points, the fit midline, and quiescent center point are overlain on the first input image. The second file is a plot of the velocity profile, with the strip-width in pixels shown along with the absolute velocity at the quiescent center. This value is used to invert the velocity profile and represents root growth rate. The third file contains the velocity profile data, with position given in microns from the quiescent center and velocity in microns per second. The string used as *file name stem* is specified in the input.txt file from the line that gives the output data file name.

**MULTI-STACK VERSION**

When the root is imaged in overlapping frames, the multi-stack version is used. Here the user starts imaging either at the tip or at a basal region shootward of the end of the growth zone, captures a pair of images (referred to hence as a ‘stack’), moves the stage by approximately 4/5 of the field of view, captures another pair of images, and continues until the growth zone is spanned. To construct a complete velocity profile, Stripflow concatenates the profiles made from each stack. To do so, the distance moved between stacks must be known. There are two ways to provide this information. Either, the stage can be moved by a defined amount, for example by a motorized stage. If this method is used then the same distance must be moved between each stack. Alternatively, after capturing the pair of root images, the focal plane may be changed to a fixed background and a third image captured. Stripflow will then find the distance moved with reference to these background images. This cannot be done by using the root itself because growth changes the root between stacks. (Note, it has been many years since we have used the fixed-scene method of registration and the feature might not work properly now.)

The tiff directory for images will contain all of the images needed, named as shown:

stack0001frame0001.tif

stack0001frame0002.tif

stack0002frame0001.tif

stack0002frame0002.tif

stack0003frame0001.tif

stack0003frame0002.tif

…

For as many stacks as were captured. Note that this is when stage movement is known directly. The folder will also contain background images when those are used.

The input.txt file will look like this:

1 [Direction of growth: 0=right to left, 1=left to right.]

0 [Order of segments: 0 if first stack is tip, 1 if base.]

4 [Number of stacks.]

2.00 [Calibration. In this example, 1 µm = 2 pixel.]

2 [Mosaic method: 1 from background image, 2 from defined camera motion.]

500 [Camera motion in µm. Leave blank when mosaicking is from background image.]

09 22 02 [Time in h min sec of stack 1, frame 1.]

09 22 51 [Time in h min sec of stack 2, frame 1.]

09 23 32 [Time in h min sec of stack 3, frame 1.]

09 24 19 [Time in h min sec of stack 4, frame 1.]

stack1 [Begins provision of geometric parameters.]

36 698, 640 709, 1222 683 [Coordinates in pixels of the midline.]

dlr 239 213 [Diameter in pixels of the root at left and right boundary.]

stack2

21 693, 1532 693

dlr 240 252

stack3

36 677, 820 695, 1522 683

dlr 227 237

stack4

52 670, 472 702, 988 673, 1537 671

dlr 236 229

endm [Ends the provision of geometric parameters.]

30 [Time between images, in seconds.]

1274 679 [Coordinates of the zero point of the velocity profile; usually the quiescent center.]

hellokitty.txt [File name of the output data file. {*file name stem*}.txt.]

Note that at present, the input image file names are not specified in the input file and must be as shown above (it should be a simple matter to change this the in read\_input file. Details not explained here are explained above for the single stack version.

Output files are as stated above for the single stack version with the addition of an image of the root with midline points and midline for each stack.

First it is necessary to prepare a time stamp file that contains the time at which each image was captured. The times can often be obtained from the image file itself, for example in the tif file header, provided you have software that can read the header. Also the image capture software (camera software) should be able to display this information. Then, make a text file that contains the time of the first frame of the stack. Use the following format: 12 30 30

for 12:30 and 30 seconds. Save as “times.txt”.

Then run makeInputFile in matlab and enter data in the forms it provides. The following information is needed:

direction of root growth: whether left-to-right or right-to-left

the number of stacks in the sequence

the pixel size in microns

the interval between frames in each stack

Once Matlab is open, here is what you do:

Run makeInputFile

Then, choose a time stamp file (“Choose timestamps file”) and decide on a name for the input file (“Input file to create”), typically input.txt. This will determine the directory that will be searched in for image files. Make sure you have set the other parameters in the form. Then click “Go.”

For each stack in the sequence, you will need to repeat the following steps:

1. Select “left diameter”. Draw a diameter on the left side as follows: click on one side of the root, drag to the other side, and click again; then press enter (return). If you do not click the second time, the line will have length zero.

2. Select “right diameter”; draw a diameter on the right side of the root, as in step 1, then press return.

NOTE: The idea here in entering these diameters is to delimit the useable area of the root. If there are dark or blurry regions along the sides of the root, measure to the inside edge of the dark or blurry area. ALSO, be sure to make the line perpendicular to the root.

3. Select midline coordinates; click on two, three, or four points along the midline of the root in the image, then press enter.

4. If this is the last image, select “Quiescent center”; click on the quiescent center and press return. Otherwise, repeat from step 1.

NOTE: that for arabidopsis roots, we take as x=0 the position on the midline of the quiescent center. The position of the quiescent center is usually easy to see. However, you may choose the absolute tip of the root. Just be consistent.

Click “Run.” If everything went well, you should get a little dialog box that says “Input file created.” If not, you may need to start over.

A note about entering midline coordinates. If the root is straight in the frame, use two points; if it is uniformly curved, use three; if it has an inflection or a changing curvature, use 4. Try to put the points roughly evenly spaced along the root. If you put a point too close to the edge of the image, the program may stop working.

The input file generation program is fussy. If you go through the process in a different order, the program may stop working, and you might have to start over. If this is too troublesome, one can also prepare the input file manually, by measuring diameters and identifying points on the midline. You need to generate a text file that is in exactly the same format as the input.txt file provided with the examples. The content of the input.txt file is described in detail in prepare\_input.txt.

The input structure for stripflow is mostly inherited from RootflowRT. Each pair of input images, being images of the same position in world coordinates taken at two different times, are named as two "frames" in the same "stack". If line 8 gives the base directory as, say, "~/base," and there are **n** stacks, the assumed subdirectory structure is:

~/base/

input.txt

tiff/

stack0001frame0001.tif

stack0001frame0002.tif

stack0002frame0001.tif

stack0002frame0002.tif

...

stack000**n**frame0002.tif

The input file, named "input.txt" by default, contains meta data needed for the program, including scale factor, direction of motion, time interval between frames, and so on. The file follows the format for RootflowRT's input, and frankly is a little awkward.

The specification for the imput.txt file is at

http://www.bio.umass.edu/biology/baskin/research\_rootflow.htm#input

The relevant section of that page is also included for convenience here, as a separate file called: prepare\_input.txt.

There are 3 files responsible for the basic work of the algorithm; they should not require editing for routine work:

stripflow.m

estTransKeren.m

shift\_fft.m

The fourth file, read\_input.m, reads an input file providing metadata for the input images. When you have prepared the input.txt file, then from inside matlab, run read\_input.m. It needs to be edited to indicate the directory where the input and image files are located. If your setup follows the assumed conventions (described further below), all that needs to be edited is line 8, where the folder path is specified. Note that you edit it inside matlab after you open it. After it is edited, click run.

The output from stripflow goes into a text file (the name of which is specified in the input text file). As of now, stripflow's output consists of two space-separated columns of data, the distance from the root tip and the velocity.

As the program runs, several intermediate steps are displayed in windows for your examination but at the moment not saved. There are several things to note. For each stack, the program will display the points entered for the midline, the calculated midline, superimposed on one of the frames. Because the midline points are entered manually it is easy to make a mistake so it is important to check that the displayed points and calculated line are reasonable. In subsequent versions we will save these overlays so they can be inspected later. The program displays the velocity profile for each stack in real world coordinates. When all stacks are processed, the program displaces the complete profile with no velocity adjustments, the complete profile with velocity shifted for maximal contiguity, and finally the complete profile in x=0 v=0 coordinates. The data for this graph is what is currently saved in the output.txt file.

NOTE: The program was written to cope with a complete growth zone which requires several overlapping stacks to span. However, for many purposes, it is of interest to obtain data from a single stack. In this case, the input.txt file is prepared the same way but will have information for only one stack. AND, VERY IMPORTANT, run read\_singlestack\_input.m instead of read\_input.m.

Easily customizable things:

- edit line 9 to use an input file name other than 'input.txt'

- edit line 5 to change the width of the strip used in local velocity estimation ('stripwidth')