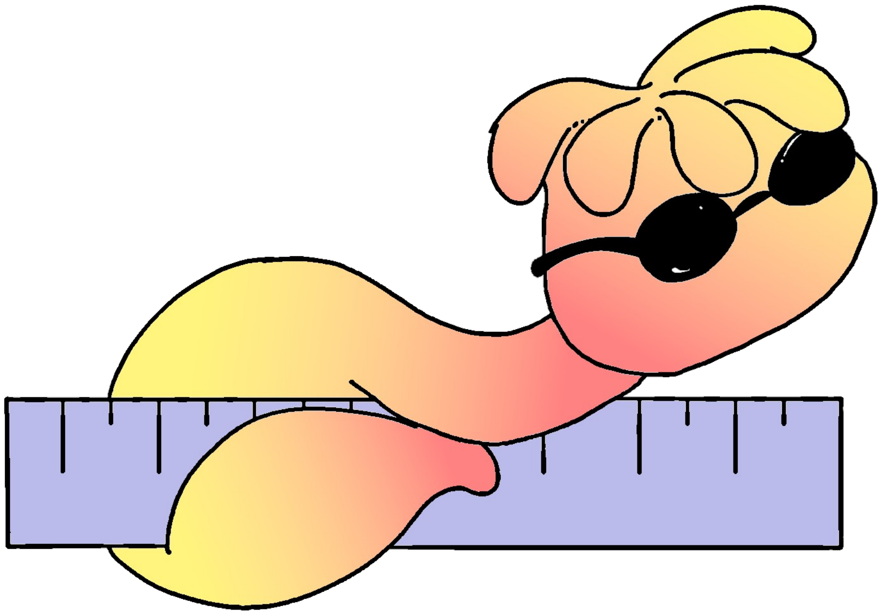
WormBox

version 0.9 beta



A tool set for automating measurements with Fiji/ImageJ

Bruno C. Vellutini & Fernando P.L Marques

- 2011 -

https://github.com/nelas/WormBox/zipball/masterAbout WormBox:

WormBox is a plugin to FIJI/ImageJ that was written with the intent to help you to automate the uptake of measurements from sets of images. This macro is based on plotting landmarks upon images from which a table with linear distances will be generated. You can also use it to count structures (i.e., meristic variables). We have not tested how effective time wise this plugin will be in comparison to the traditional ways most people obtain measurements from specimens. We expect that it will vary a lot among cases. Ultimately, it will all depend on how easy (and fast) you can produce images from which you can extract the information you need. If that is not an issue, we believe that you will find that this application will speed up data gathering. You should also consider that, even if you do not speed up your work by using this application, using it will provide you with full documentation of your measurements – which usually is not the case by traditional methods. Be that as it may, give it a try. We will be happy if it turned out to be a good tool for your research.

To download the most recent version of the plugin, press here.

System requirements:

As this application makes full usage of Fiji/ImageJ, which can be downloaded from http://fiji.sc/wiki/index.php/Fiji, as long as you meet the system requirements of Fiji/ImageJ, this application will work. Thus, before you start you should install Fiji/ImageJ. After installing Fiji/ImageJ, you should verify whether there are updates for Fiji and/or ImageJ available. To accomplish that, open the casting menus of Help in the main menu of Fiji/ImageJ and select the options to update both applications (Figure 1).

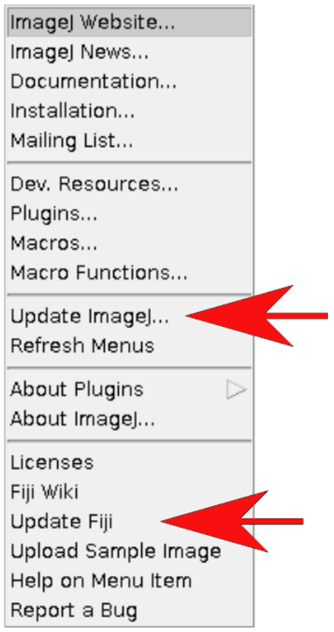


Figure 1. Options of updates in the main menu of Fiji/ImageJ.

NOTE: WormBox will not work in ImageJ as distributed because this plugin requires Jython. However, you can make ImageJ capable of dealing with **WormBox** by installing Jython in your system (see here). We have not tried to do that and advise you to install Fiji/ImageJ, which seems to a much easier solution.

Installing the plugins:

After downloading **WormBox.zip** and decompressing the ZIP file, you will find two files within the folder: WormBox_Tools.txt and WormBox_Analyzer.py. These files are the scripts of **WormBox** and will have to be moved to specific locations within Fiji/ImageJ's folders. Thus, move WormBox_Tools.txt to Fiji.app/macros/toolsets/ and WormBox_Analyzer.py to Fiji.app/plugins/.

NOTE: MAC users could use the terminal to accomplish this task.
1. open the Terminal application.
2. move to the folder where WormBox.txt and WormBox_.py are located (e.g., cd Desktop/WormBox)
3. execute 'cp WormBox.txt /Applications/Fiji.app/macros/toolsets/'
4. execute 'cp WormBox_.py /Applications/Fiji.app/plugins/'

To make life a little bit easier for Mac and Linux users, we wrote a script in PERL that will copy the files to their appropriated directories. Open the Terminal application, move to WormBox folder and execute the script:

```
your_computer/WormBox$ perl install.pl
```

Planning your project:

WormBox uses landmarks to compute linear distances. Thus, before you start, you should plan your project so that you have a clear idea of how many landmarks you will need to obtain the measurements you want. You can add more landmarks throughout the process, but we advise you to really make an effort to plan ahead. We will illustrate the basics of **WormBox** by using an example in which our goal is to obtain measurements from a set of hooks for *Potamotrygonocestus* (a onchobothriid cestode). With this example we hope to demonstrate how **WormBox** can be helpful.

You should start by defining the landmarks that would allow you to calculate the distances (measurements) you need (Figure 2). In the example below, we want to obtain the measurements from handles and prongs of the internal and lateral bothridial hooks. Accordingly, we defined 8 landmarks (Figure 2). **WormBox** allows you to set as many as landmarks as you need (or wish), but you should always be careful and minimize the amount of landmarks for practical reasons.

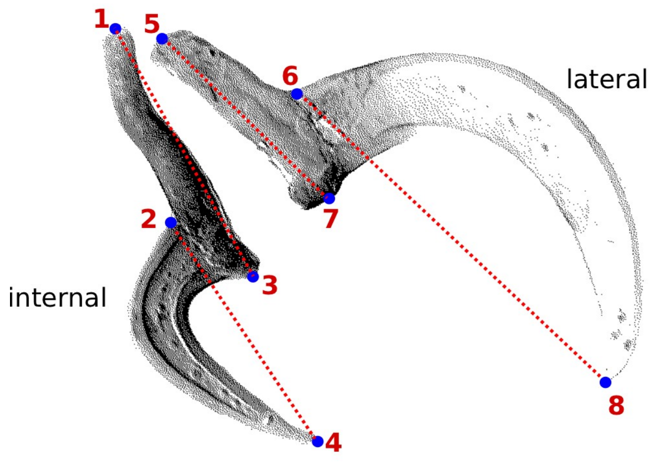
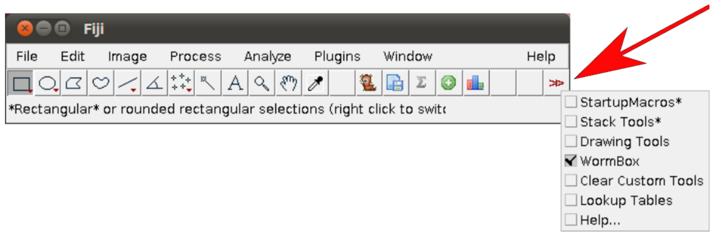


Figure 2. Definition of landmarks upon handles and prongs of internal and lateral bothridial hooks of *Potamotrygonocestus*.

Getting started with WormBox:

Once you have a good idea of the measurements you want to take and the landmarks required to do that, you should compile within a folder all the images you want to extract the measurements. We provide a folder called "hooks" that we will use in the first part of this tutorial. The images in this folder have the hooks from which to obtain the measurements defined in Figure 2. Ultimately, the images should be all scaled, preferentially using ImageJ or Fiji/ImageJ. **WormBox** will inform you whether it recognized the scales in your images. If **WormBox** does not recognize the scale, it will ask you to set up the scale. Here are the steps you should take to fully complete the first part of this tutorial:

Step 1. Start Fiji/ImageJ and enable **WormBox** in the switch to alternate macro tool sets (Figure 3).



Rectangular or rounded rectangular selections (right click to switc

Figure 3. Selecting WormBox tool sets.

Once the plugin is enabled, 5 new icons will become available to you with the following function:

Initialize landmark setup and/or load ROI (Region of Interest) to images.

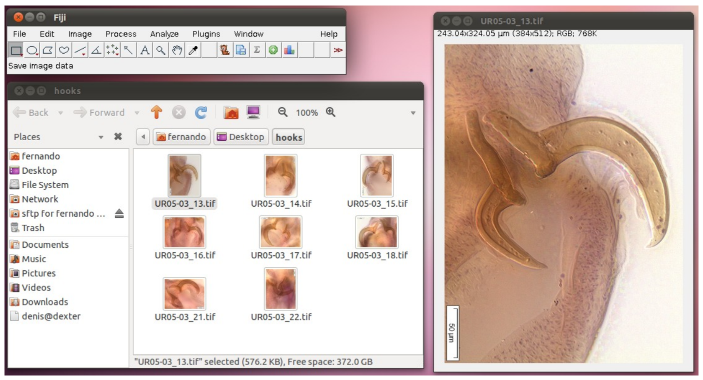
Save landmark positions.

Count meristic variable.

Add new landmarks

Run **WomBox** Analyzer, to compile measurements to *.cvs file.

Step 2. Open the first image of your working folder. You can do that by either dragging the image into the Fiji/ImageJ main menu area (gray area) or by using File/Open.



Save image data

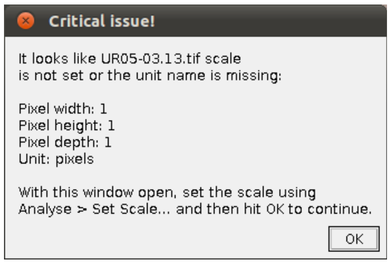
"UR05-03_13.tif" selected (576.2 KB), Free space: 372.0 GB

243.04x324.05 µm (384x512); RGB; 768K

Figure 4. Opening images in Fiji.

NOTE: Files should be in *.tif format and the extension should be in lower case. **WormBox** will recognize all major images formats recognized by Fiji/ImageJ. However, if the original image is not in *.tif format, or if the extension is written in upper case, **WormBox** will make a copy of the image and save it as *.tif after processing.

Step 3. Scaling images. If your images are not scaled by Fiji/ImageJ or, although scaled using a different software, Fiji/ImageJ did not recognize the scale, **WormBox** will require that a scale is provided and the warning window below will be presented.



```
It looks like UR05-03.13.tif scale
is not set or the unit name is missing:

Pixel width: 1
Pixel height: 1
Pixel depth: 1
Unit: pixels

With this window open, set the scale using
Analyse > Set Scale... and then hit OK to continue.
```

Figure 5. Request for providing a scale by **WormBox**.

To scale the image in Fiji/ImageJ you will have to follow these steps:

i. Select the *straight* tool to draw lines.

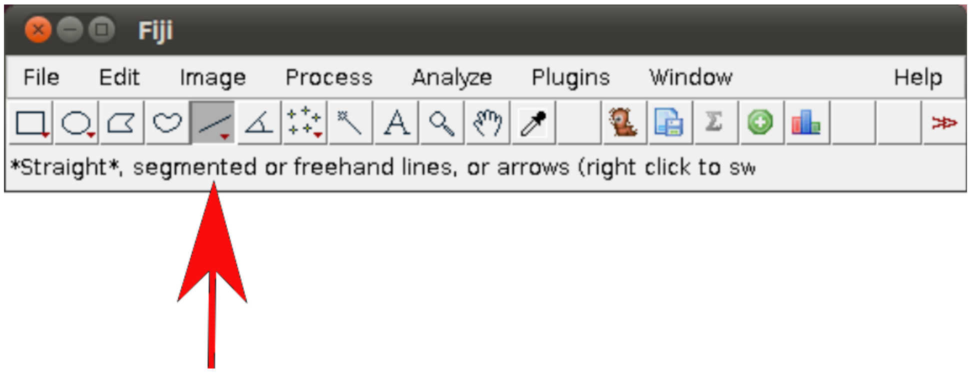
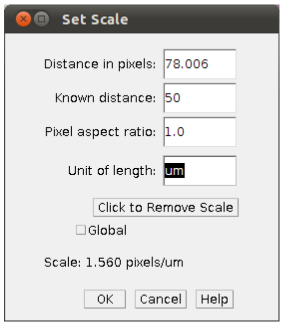

Fiji
File
Edit
Image
Process
Analyze
Plugins
Window
Help
Straight, segmented or freehand lines, or arrows (right click to sw

Figure 6. Selecting the straight line tool in ImageJ.

ii. Draw a straight line using as reference the scale bar embedded into the image.
iii. In the main menu, select Analyze/Set Scale and provide the known distance (i.e., 50) and the unity of the length (i.e., um)



Distance in pixels: 78.006
Known distance: 50
Pixel aspect ratio: 1.0
Unit of length: um
☐ Global
Scale: 1.560 pixels/um

Figure 7. Set scale menu in ImageJ.

NOTE: It might be easier to have all images scaled before setting up landmarks. But that is not mandatory, you can provide the scales as you work in each image.

Step 4. Placing landmarks. If you have opened a image and **WormBox** has recognized the scale, you are ready to define and/or place the landmarks you desire to obtain the measurements you want. Press the button to load ROI from the **WormBox** tool set. If this is the first time you are using **WormBox** for the images within a given folder you will have to define the number of landmarks you will be working.

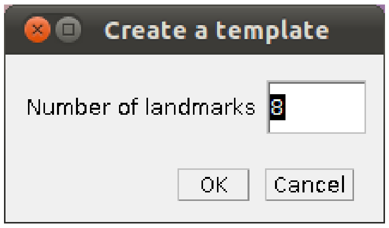


Figure 8. Defining the number of landmarks in **WormBox**.

As we mentioned above (Figure 2), we will need 8 landmarks for the measurements we are planning to take from the hooks. By selecting 8 and pressing “OK”, we will define a template with eight landmarks which will be used for all images in this folder (Figure 9A). A file called RoiSet.zip will be created and all landmarks will appear on the left part of the image.

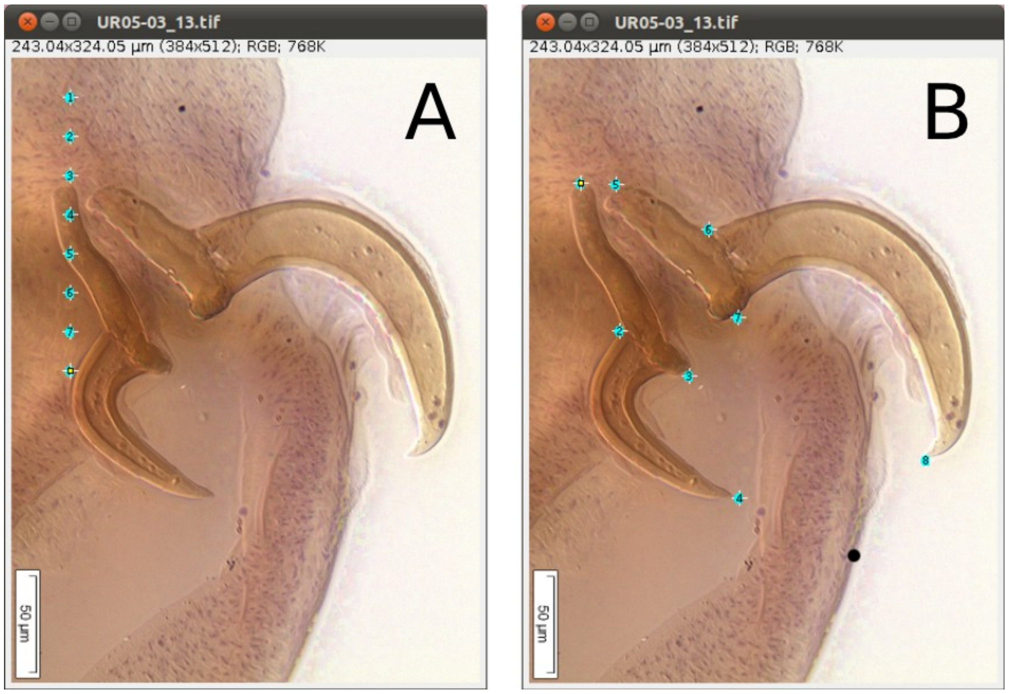
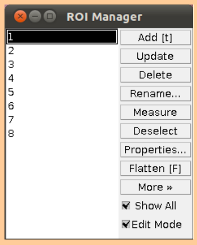

UR05-03_13.tif
243.04x324.05 µm (384x512); RGB; 768K
A
50 µm
UR05-03_13.tif
243.04x324.05 µm (384x512); RGB; 768K
B
50 µm

Figure 9. A, landmark pasted on image after defining numbers of landmarks or loading template. **B**, landmarks placed in desired positions.

Once the landmarks have been defined or loaded, all you have to do is to drag the landmarks to its right position using the mouse (Figures 2 and 9B) and then save it using the icon in the macro tool set menu of **WormBox**. After saving the image, **WormBox** will wipe the landmarks from the image. However, all information will be still there, saved in associated files with extensions *.txt and *.zip. These files will be used to compile the information after you process all images and **SHOULD NOT**

BE REMOVED – unless you want to wipe out all the data. If you desire to verify and or modify the positions of the landmarks, just reload the ROI using the icon in the macro tool set menu of **WormBox**.

NOTE: **WormBox** allows you to delete any landmark if you feel that a given measurement should not be taken from a given image.
All you have to do is to place the landmarks that you consider that should be included in the image and use the ROI manager to delete the ones that should not be taken into account. In its final stage, when **WormBox** will parse the images to compile the measurements, images for which one or more landmarks are missing, will still be used. **WormBox** will attribute "**NA**" for cells in which the distance could not be calculated by the lack of landmarks in the *.cvs file (see below).



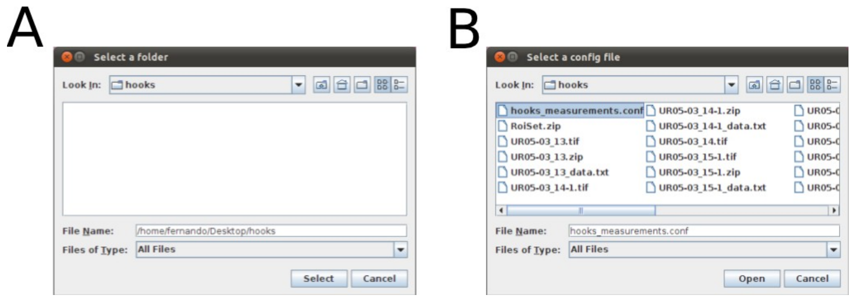
Step 5. Compiling the data from images. Once you have processed all images (in this particular example it should take less than 10 minutes to process all 8 images) you will have to write a configuration file to run the compiling tool of **WormBox**. The configuration file should be a text file with the following format[1]:

```
internal_handle:1,3
internal_prong:2,4
lateral_handle:5,7
lateral_prong:6,8
```

Each line should contain the name of the variable, followed by ":" and at least two landmarks separated by comma defining the two points from which **WormBox** will calculate the distance.

NOTE: You can use more than two land marks for a single variable. For instance, your configuration file could have a line like this: "variable_x:1,2,3,4". In this case, the measurement for "variable_x" will be the sum of the distances between $\overline{1\ 2}$, $\overline{2\ 3}$, and $\overline{3\ 4}$.

Once you have written your configuration file (e.g., hooks_measurements.conf) and saved it, run the **WomBox Analyzer** (see definition of icons above). First, **WormBox** will ask you the target folder in which your images have been processed (Figure 10A); and then the name of your configuration file (Figure 10B).



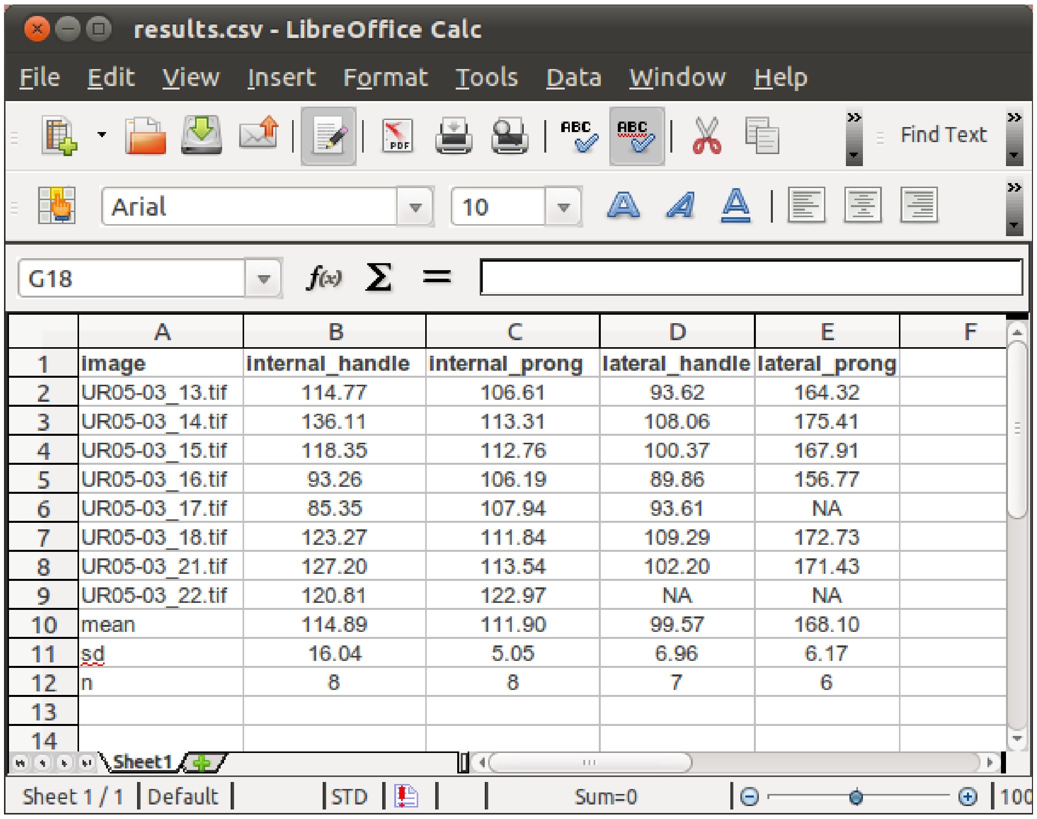
A — Look In: hooks
File Name: /home/fernando/Desktop/hooks
Files of Type: All Files

B — Look In: hooks
hooks_measurements.conf
RoiSet.zip
UR05-03_13.tif
UR05-03_13.zip
UR05-03_13_data.txt
UR05-03_14-1.tif
UR05-03_14-1.zip
UR05-03_14-1_data.txt
UR05-03_14.tif
UR05-03_15-1.tif
UR05-03_15-1.zip
UR05-03_15-1_data.txt
File Name: hooks_measurements.conf
Files of Type: All Files

Figure 10. A, menu for selecting target folder. **B**, menu to specify configuration file.

1 You should use plain text format here. Thus, do not use Word or any other text editor the leave hidden characters into the text. Linux users could use nano, vim, Geditand etc. Mac users could use TextEdit, Wrangler, nano, and etc.

Finally, you will have to specify the output file name. **WormBox** will generate a *.csv (comma-separated values) text file that can be opened in any spreadsheet software (e.g., Excel, OpenOffice, and etc) as demonstrated in Figure 11.



results.csv - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
Find Text
Arial
10
G18
Sheet1
Sheet 1 / 1 Default STD Sum=0

	A	B	C	D	E	F
1	image	internal_handle	internal_prong	lateral_handle	lateral_prong	
2	UR05-03_13.tif	114.77	106.61	93.62	164.32	
3	UR05-03_14.tif	136.11	113.31	108.06	175.41	
4	UR05-03_15.tif	118.35	112.76	100.37	167.91	
5	UR05-03_16.tif	93.26	106.19	89.86	156.77	
6	UR05-03_17.tif	85.35	107.94	93.61	NA	
7	UR05-03_18.tif	123.27	111.84	109.29	172.73	
8	UR05-03_21.tif	127.20	113.54	102.20	171.43	
9	UR05-03_22.tif	120.81	122.97	NA	NA	
10	mean	114.89	111.90	99.57	168.10	
11	sd	16.04	5.05	6.96	6.17	
12	n	8	8	7	6	
13						
14						

Figure 11. Summary results from **WormBox** showed by opening the *.csv file in OpenOffice Spreadsheet.

The summary results will contain a column with the image names and each variable will be displayed in a separated column. The three last lines of the file will compile the average, standard deviation and sample size of each variable. Cells with "**NA**" (as Non Available) denote measurements that were not taken due to the lack of associated landmark(s).

More on WormBox (Part 2):

Now we will explore some other features of **WormBox**. We provide a folder called "proglottids" upon which we base this second part of the tutorial. Within this folder you should find 3 images of loose proglottids of *Potamotrygonocestus*, all of which have been scaled using Fiji/Image.

In Figure 12 we depict the landmarks we wish to use to compile length and width of the proglottids, position of the genital pore, and testes diameters. We also will compute the number of testes for each proglottid. Thus, we will make use of the counting tool in WormBox, which we have neglected in the first part of the tutorial.

Except for the fact that we are using two sets of landmarks to compute testes diameter, and that the total legth of the proglottids will be calculated the sum of two distances (i.e., $\overline{1\ 2}$ and $\overline{2\ 3}$), you should have no problem with that if you went through the first part of the tutorial. It is true that the configuration file will have to specify that; which will be addressed latter. Thus, let's see how we count structures using **WormBox**.

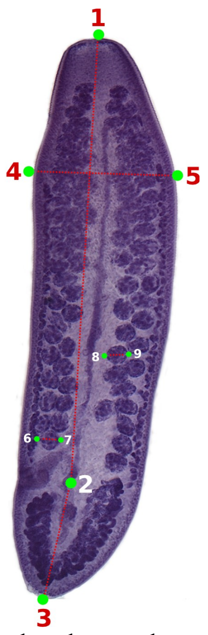
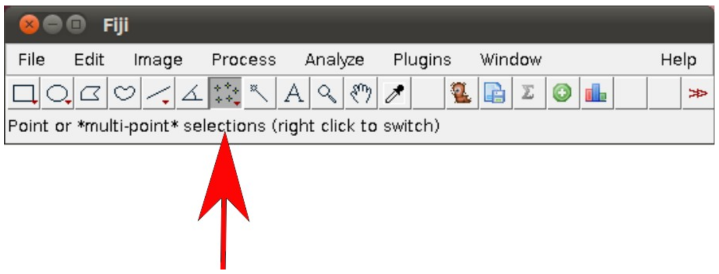


Figure 12. Landmarks on loose proglottids of *Potamotrygonocestus.*

Counting structures: Before starting the use of the counting toll of **WormBox**, you should:

i. Define the number of landmarks you need.
ii. Place them in the right positions.
iii. Save image (required)

The steps above have been described in the first part of the tutorial. If you have any difficulties, please go back and have a quick look above. To start counting, you have to enable the function to count meristic variables (icon with **Σ** sign). By doing that, you will turn on the multi-point tool of Fiji/ImageJ and a window named "action required" will appear.



Point or *multi-point* selections (right click to switch)

Figure 13. Enabling the multi-point tool of Fiji/ImageJ.

Once the counting tool is on, all you have to do is to position your mouse on top of the structures you want to count and click the mouse. Here, we will count two sets of testes: poral and aporal ones. As you start clicking on top of the testes, let's say on the aporal side, landmarks in yellow will provide the counting (Figure 14).

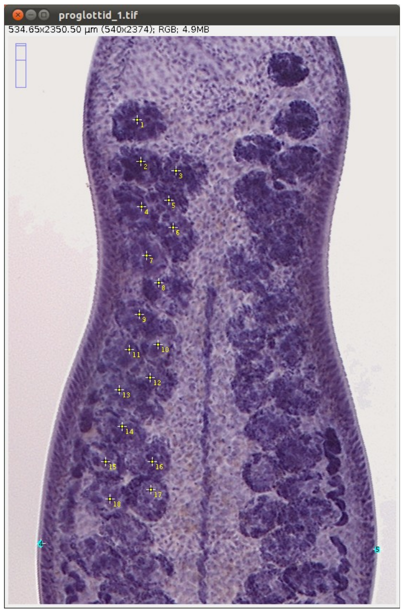


Figure 14. Image showing landmarks in blue and counting unities in yellow for partial counts of aporal testes.

After concluding counting the aporal testes, you should click on "OK" on the window for "Action Required" and this will prompt **WormBox** to ask you to name of the variable you just counted (e.g., aporal_testes).

NOTE: Watch out for spelling error here otherwise the script will consider them as different variables. It is important to inform you that the script is case sensitive. Thus, "testes" and "Testes" will be understood as two different variables.

Now, let us count the other set of testes, i.e., poral testes. You can start by loading all you have done thus far by pressing the icon to load ROI into image. If you load the ROI, you should have something like Figure 15.
The image should display all landmarks for calculating distances and counting, and the ROI manager should have a list of all landmarks and variables. Now proceed to count the poral testes by enabling the counting tool and repeating the steps you did before in which you count the aporal testes. You should repeat all the steps

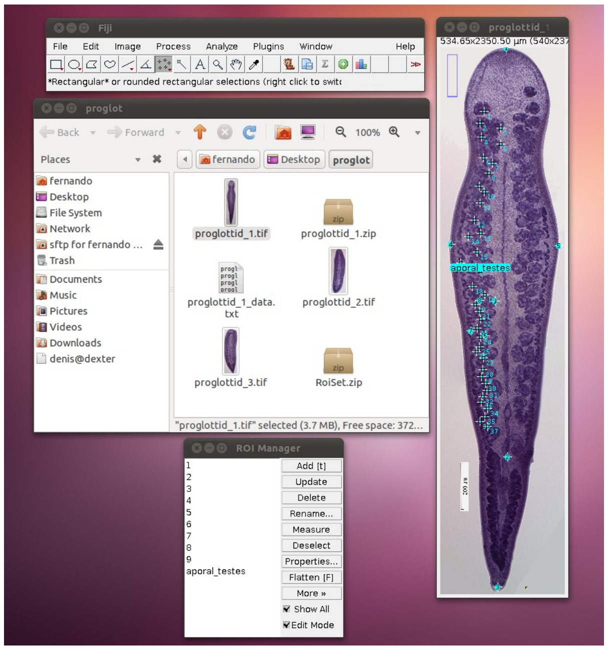


Figure 15. Summary information after loading ROI at this

above for the other remaining (2) images. point.

Compiling results (complex version): As explained above, you will need to specify a configuration file to parse all the information you implemented in each image. Here we will apply a configuration file a bit more sophisticated than before. Within the configuration file, we will not only obtain euclidean distances between landmarks but also specify new variables based on operations of those distances. Below we provide a example of configuration file to illustrate this capability of **WormBox**[2]:

```
#1  length:1,2,3
#2  width:4,5
#3  progl_ratio:${length}//${width}
#4  genital_pore_distance:2,3
#5  genital_pore_position:${genital_pore_distance}*100/${length}
#6  testes_dia:6,7
#7  testes_dia:8,9
#8  aporal_testes:count
#9  poral_testes:count
#10 total_testes:${aporal_testes}+${poral_testes}
```

Here is what this configuration file will do with the data you compiled for each image:

Lines 1 and 2 (i.e., #1 and #2) will calculate the length and width of your proglottid, which except for the fact that the length is being calculated by the sum of two distances (i.e., $\overline{1\ 2} + \overline{2\ 3}$), they are the kind of distances you calculated before in the first part of this tutorial.

Line 3 provides something new. Here we defined a new morphometric variable,"`progl_ratio`", which is equal to the floored quotient of length and width of the proglottid; that is, its ratio. To operate with variables there are few rules you must follow:

i. You can only operate with variables that were defined before. That is, line 3 could not be written prior to lines 1 and 2.
ii. The syntax for operating variables should be "${name_of_variable}".
iii. The operator used in **WormBox** are basically those available in Python:

Operation	Result
x + y	sum of x and y
x - y	difference of x and y
x * y	product of x and y
x / y	quotient of x and y
x // y	(floored) quotient of x and y
pow(x, y)	x to the power y
x ** y	x to the power y

Other lines that make use of this property are lines 5 and 10 in which we calculate the genital pore position and the total number of testes, respectively.

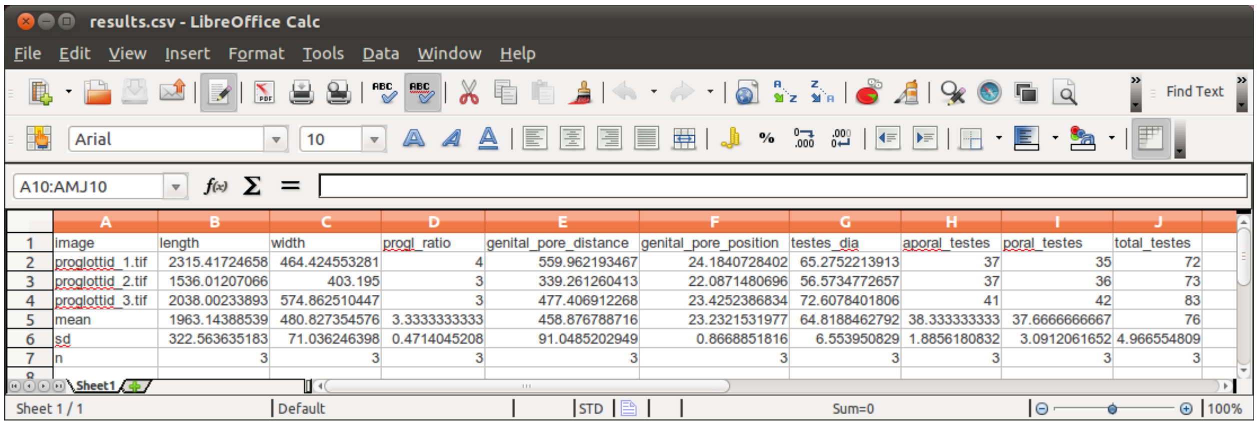
Next, we should comment lines 8 and 9. These two redundant lines specify the distances for testes diameter (i.e., $\overline{6\ 7}$ and $\overline{8\ 9}$). For those, when **WormBox** parse the information of an image, it

2 For explanation proposes, each line is specified by a number (e.g., #1), which should not be in the configuration file.

compiles the mean value between 2 or more pseudo-replicates.

Finally, meristic variable are defined by its name followed by "`:count`". Examples of that are the testes counts defined in "`aporal_testes`" and "`poral_testes`", lines 8 and 9, respectively.

The final results should be something like this:



	A	B	C	D	E	F	G	H	I	J
1	image	length	width	progl_ratio	genital_pore_distance	genital_pore_position	testes_dia	aporal_testes	poral_testes	total_testes
2	proglottid_1.tif	2315.41724658	464.424553281	4	559.962193467	24.1840728402	65.2752213913	37	35	72
3	proglottid_2.tif	1536.01207066	403.195	3	339.261260413	22.0871480696	56.5734772657	37	36	73
4	proglottid_3.tif	2038.00233893	574.862510447	3	477.406912268	23.4252386834	72.6078401806	41	42	83
5	mean	1963.14388539	480.827354576	3.3333333333	458.876788716	23.2321531977	64.8188462792	38.333333333	37.6666666667	76
6	sd	322.563635183	71.036246398	0.4714045208	91.0485202949	0.8668851816	6.553950829	1.8856180832	3.0912061652	4.966554809
7	n	3	3	3	3	3	3	3	3	3

Figure 16. Summary results from **WormBox** showed by opening the *.csv file in OpenOffice Spreadsheet.