

The WINGMACHINE: Drosophila wing splining and editing with Wings and CPReader

David Houle
dhoule@bio.fsu.edu
August 27, 2009

This set of programs, The WINGMACHINE, takes images of Drosophilid fly wings and fits spline curves to them, as shown in Figure 1, then collects the results and performs some basic registrations. The software here runs under Windows. It could probably be adapted for Mac or Unix, but we have not done so. We are happy to furnish the source code for this software if you would like to try improvements or porting of this system to other operating systems. Please contact us at dhoule@bio.fsu.edu.

The WINGMACHINE currently consists of two JAVA programs: Wings, which splines the images, and CPReader, which reads the resulting data for each individual and puts that data into one file with each individual on a different line. Wings calls an underlying C program, FINDWING, which is largely hidden from the user.

Recording images

The wing images themselves can be digitized with a wide variety of hardware setups. Ours is described in the paper Houle et al., 2003 BMC Evolutionary Biology 3:25. Perhaps the most useful part of our setup is a very simple suction device for immobilizing the wing of a fly to enable imaging on a live specimen. We use Optem macroscopes with digital cameras or a combination of an analog camera and a frame-grabber. These do not need to be of high quality, as the spline software works on a low resolution, grey scale version of the image. It is also remarkably forgiving of many kinds of imperfections, such as dirt, small tears in the wing. Others, such as hairs that look like a vein, will cause problems.

You will need to write or buy your own software to associate the wing image with the locations of two starting landmark positions on the wing, and to reduce the images to the sizes Wings can use. We have not furnished our software for this, as this is hardware-dependent. We use, and would recommend the program ImagePro for those with money to spend, NIHImage is another possibility that is free, although we have not found this easy to use in the Windows environment.

The two landmarks needed are at the distal side of the humeral break, and the notch in the sinus between the alula and the trailing lobe of the wing, as shown in Figure 1.

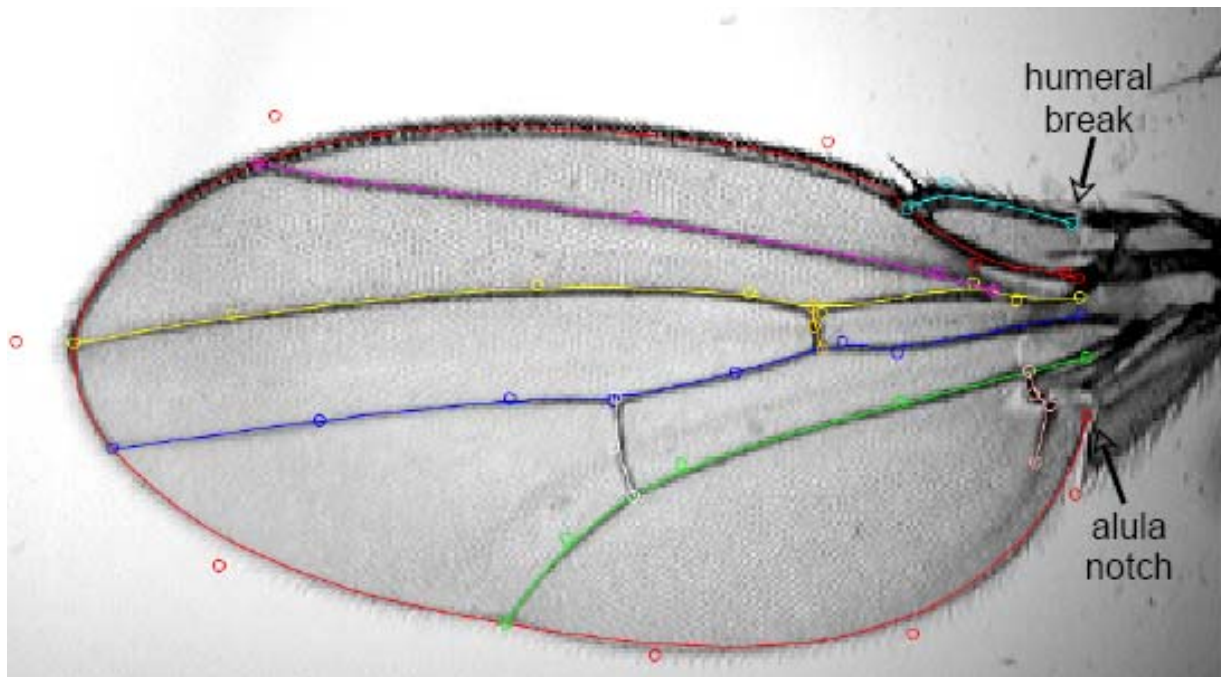


Figure 1. Nine B-splines fit to veins and edges of a *Drosophila* wing. Circles denote the positions of spline control points. The two starting landmarks that must be furnished to Wings are shown by the arrows. Starting landmark number 1 (in the ASC file) is the humeral break, and number 2 is the alula notch.

Note the clear spot which is part of the hinge, shown in Figure 2. This is a good marker for the location of starting landmark 2 on wings where the alula is folded over, which is a frequent occurrence with our suction-based wing grabber. A description of this piece of hardware is

The starting files

The raw images should be grey scale TIFF files. We use the extension “.tif”. Unfortunately, Wings is very limited in the formats it can read. It is critical that you understand these limitations before trying to use the program. Wings is designed to start with tiff files that are 632 X 480 pixels (298 kb) in size. The actual splining will only work on tiff files that are

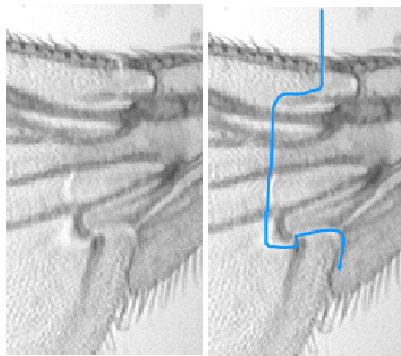


Figure 2. Approximate position of the wing hinge, shown by the blue line on the right image. Note that the alula is slightly folded from what is shown in Fig. 1, so as to obscure the location of the notch. The clear spot in the hinge, just above where the notch is a good marker to look for.

approximately 316 X 240 pixels (75) kb in size. There is a conversion routine in the Wings program to reduce the TIFF files to the 316 X 240 size. You must furnish Wings one of these two sizes of files. Many image packages allow conversion of larger files to these sizes.

You also need an ASCII format file, we call it the .ASC file, that lists the image file names, with accompanying information to allow the splining of the image. Lines in the .ASC file must look like this:

```
vir1000.tif 508 46 576 142 Jeff 15_Feb_01 Thu_PM virilis F 1.12E-03
```

The delimiter for columns is a space character. The information that Wings actually needs is in the first five columns, and the last two (the 10th and 11th columns). These are:

1. The name of the TIFF file.
2. The x coordinate in pixels of landmark 1 (humeral break, see Fig. 1).
3. The y coordinate in pixels of the landmark 1.
4. The x coordinate in pixels of the landmark 2 (alula notch, see Fig. 1).
5. The y coordinate in pixels of the landmark 2.
10. Sex of the fly imaged.
11. The scale of the image in mm/pixel.

The Wings program will merely copy any of the information in columns 6-9, so you can use those for any information about the specimen you like. There is no limitation on the length of these strings within columns. Remember that any space is treated as a delimiter, so if you enter a date as Feb. 15, 2001, this will be read as three separate columns by Wings.

This version of Wings is very limited in the expectations about the x and y coordinates and scale information in the .ASC file (columns 2-5, and 11). Wings expects these to correspond to the coordinates in images that are 632 X 480 pixels, regardless of whether you furnish the program 632 X 480 or 316 X 240 TIFF files. If you do not directly save your landmark coordinates and scale information for TIFF files at the 632 X 480 size, you must correct the values in the .ASC files so that they correspond. For example, if you record coordinates and scale for images that are 1264 X 960 pixels, then you should convert the coordinates and scale information in your .ASC file by dividing all your landmark coordinates by 2, and multiplying your scale information by 2. In the example data set, the images are 316 X 240, and the file virilis.asc has the landmark and scale information derived directly from those images. However, to spline these, you must instead use the .ASC file virilis2X.asc, which has the landmark coordinates multiplied by two and the scale divided by two.

An alternative is to rewrite the JAVA code in WINGS to allow more flexible conversions. This is on our list to do, but we haven't gotten around to it yet. If you do make this more flexible, you are doing the Wings community a big service, so please let us know.

A third type of starting file needed are template splines that are used by the Wings program. These files have exactly the same format as the files that actually contain the data recovered from each image, which we generically call .cp files. Each time you spline new wings, you need to choose a template for splining, which is encoded in a cp file. By convention, when we adopt a cp file as a template, we name a template XXXX.cp2. There is actually no difference in format between a template and a regular cp file, just a name change.

Using a good starting spline model is one of the most important steps to getting repeatable, accurate splines back again. For some data sets, choosing a wing where the splines fit well, as in Figure 1 is sufficient. However, for many other data sets, the spline routine tends to make the same errors over and over again, essentially getting attracted to the wrong solution. In such cases, it is often effective to choose a starting model that deviates from a good fit in one direction or another. Figure 3 shows such an ‘extreme’ model that is effective for splining the example data set furnished with these programs.

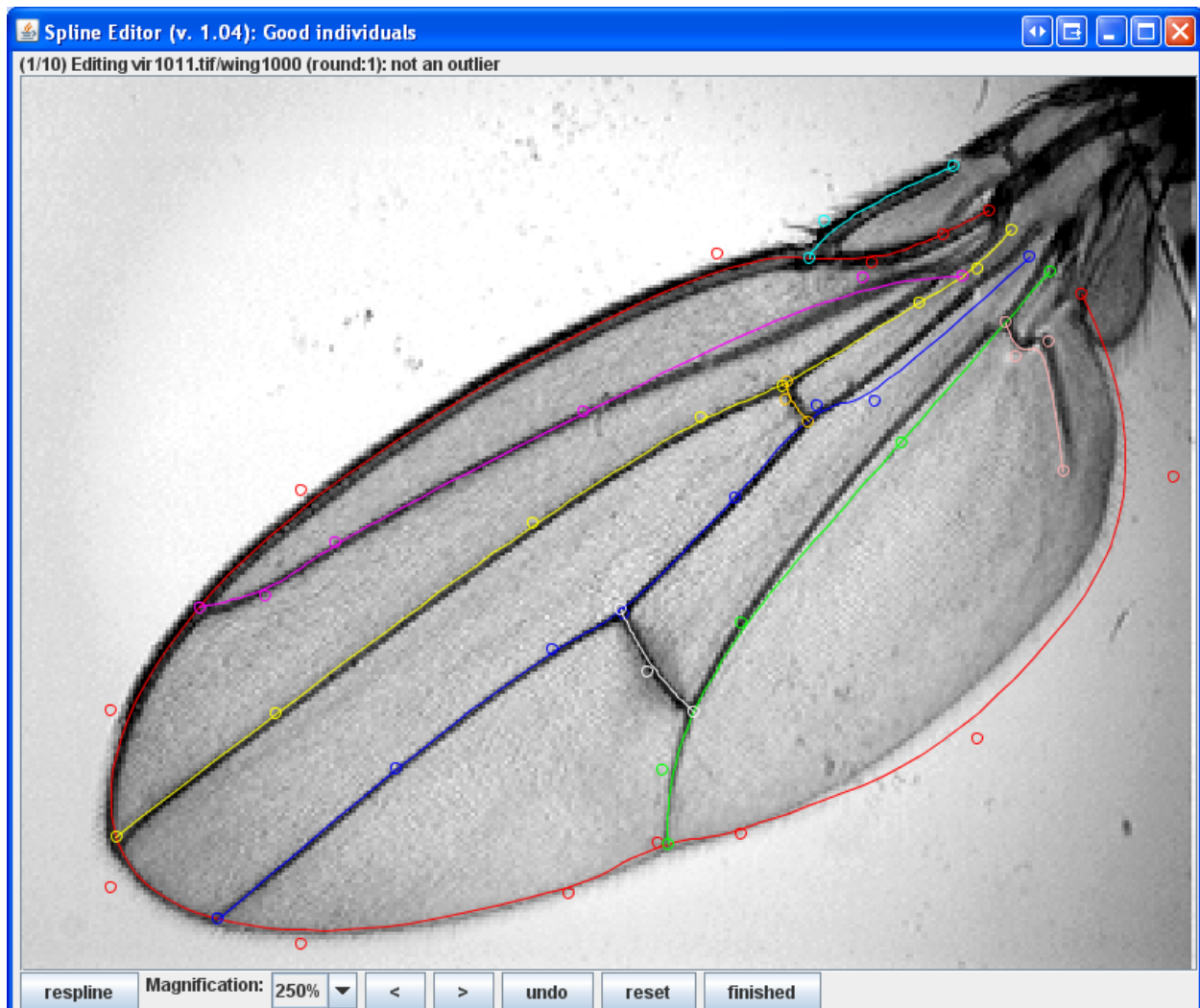


Figure 3. A spline altered to serve as an extreme model (cp2) file. To get this configuration, a fitted spline was edited in the **Edit manually selected outliers** section of Wings (that screen is shown here). By noting the wing number in the header for the image (wing1000), the corresponding .cp file can be identified. The original fitted spline is in file wing1000.cp, and subsequent modifications of the model during editing result in the addition of number, so, for example, the first time this cp model is edited, this results in the model wing1000.1.cp. Locate and copy the corresponding modified .cp then rename it with extension .cp2 to signal its use as a model file.

Installing Wings 3.7, and CPReader

To use Wings and CPReader you must install Java (the Java Runtime Environment). This is a free download available from <http://www.java.com/en/download/index.jsp> . The Wing programs can be downloaded as a zip file from <http://bio.fsu.edu/~dhoule/downloads/> . Check back there for updated programs. The current version is wings3.71.zip. To install, extract the entire zip file to a subdirectory of your choosing. In the Houle lab, we will use a subdirectory called Morpho on the D drive. This will make a new folder called Wings3.7 (or whatever version is current), and a few subdirectories that the program expects to find.

Click on Wings37.jar in the Wings3.7 subdirectory. This will first start a java virtual machine (which takes a minute or two) and then starts the program. The first time you open this file, it will ask you to set file locations for the files it needs. They are in the subdirectory Wings3.7\supportFiles which should have been created when you extract the ZIP file. WINGSIOC.EXE reads data from the C: drive on Windows machines. WINGSIO.EXE reads data from the D: drive on Windows machines. Choose which one to use depending on whether your data is on the C: or D: drive accordingly. CPReader.jar is the file for reading the output of Wings, and requires no additional set up.

Before starting the program

Before starting the Wings program look at the directory structure where your wing images and .ASC file is stored. Locate the subdirectory containing your original tif files. Locate some template files, (.cp2) so that you can tell the program where to look for them. One such file comes in the zip file, and will be at
.. /Wings3.7/Example/virilis.cp2.

Wings program usage

The program for doing the splining is called Wings37.jar. Click on this file to start the program.

1. Once the program has started, you will see a large window. Here is what it looks like:

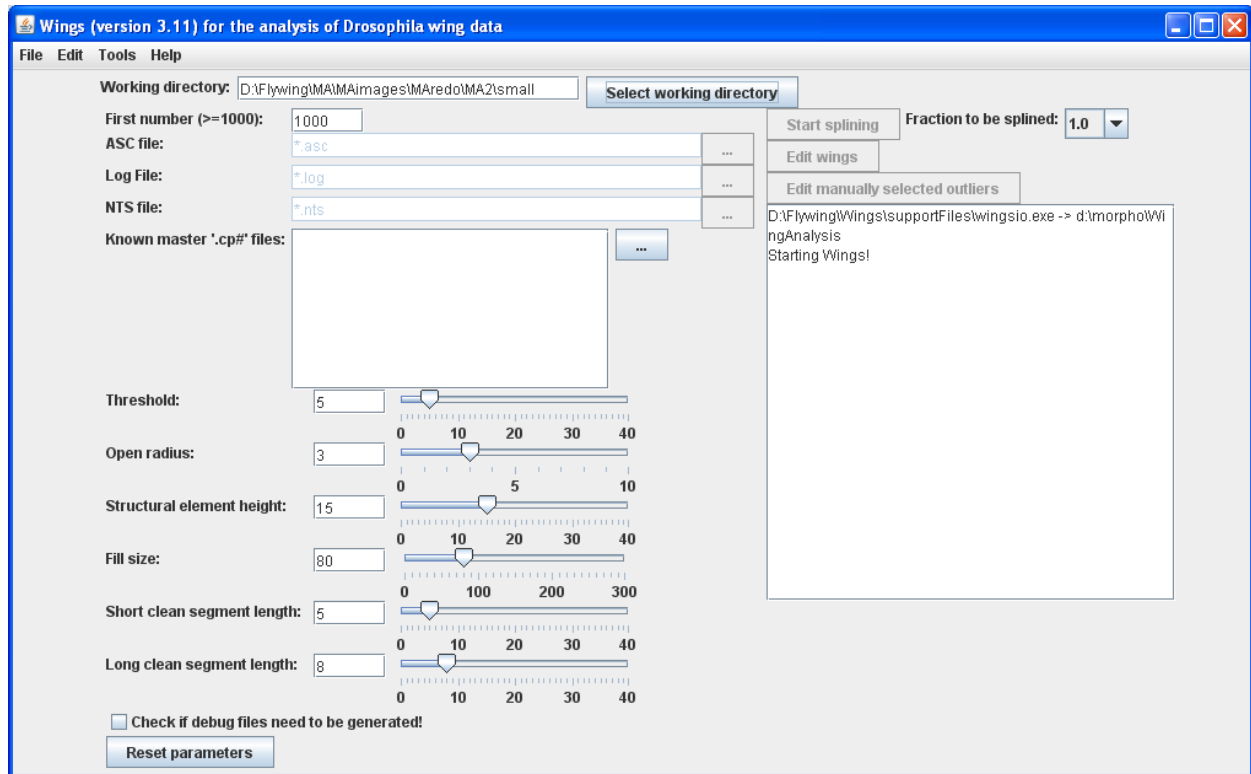


Figure 4. The main screen of Wings.

The first thing to do is select a working directory. This is the directory with the images to be splined.

2. If the tif files are 632 X 480, then the next thing that needs to be done is to convert them to the 316 X 240 format. To do this, go to the Tools menu and choose Reduce Images By Half. Verify the choice of working directory, and the program will make a new subdirectory called 'small' in the working directory, and copy the reduced tif images to that, along with a version of the ASCII file that correctly records the scaling of the images. If you try to spline 632 X 480 kb images, it will not work well, but it won't tell you what the problem is. Make sure you spline 316 X 240 pixel tiffs!
3. Next, choose a master cp2 file to use for the splining. Navigate to the subdirectory containing those files. Then, you must also highlight the cp2 file that you want to actually use.
4. Click the start splining button. Next to this button is a pull-down menu for the fraction of images to spline. Since the default parameters may not spline your batch of images well, an efficient approach is to start out splining a small fraction (it is quicker that way) to investigate whether the splining parameters or the model template file (.cp2) needs to be adjusted.
5. The program comes up with a side-by-side comparison of the splined and unsplined images. You can advance through these rapidly to get a sense for how well the average

image is splined. If you see a wing that is damaged, folded or incomplete, such that one could never expect to measure it accurately, you can click on reject to ban it from further analysis. To investigate how to change the parameters, click on Respline.

6. The Respline button brings up another side-by-side comparison labeled 'Resplining a single image'. You can then adjust the spline parameters on the main page, then click 'Respline this image' on the 'Resplining a single image' window. This will display how the change in parameters changes the splines. When you either find a good set of spline parameters, or give up trying to find anything better, then click 'Finished resplining'. This will save the most recent cp file from resplining as the best for that wing. If you click cancel, it will discard any splining done in this window, and keep the old spline as the best.
7. Choosing spline parameters is a bit of an art, and experience helps. Basically, they help the program find areas likely to be veins and vein intersection by thresholding, filling holes between thresholded areas, paring them down to lines, and then pruning off lines unlikely to connect to other lines. The parameter most likely to change and potentially improve splining is the threshold value. The default is 5, but raising the value as high as 25 or even more is sometimes helpful. Open radius is sometimes worth changing from 3 to maybe as high as 6. Structural element height is sometimes worth changing. The remaining parameters rarely seem to have much effect, but you should experiment with them for your data if you are having difficulty getting good splines. Usually a single set of parameters will do well for all the wing images recorded under the same lighting conditions. If a block includes more than one lighting setup, then actually splitting the block for processing, once you find the boundary of each set could be a reasonable alternative.
 - a. To see the effect that the spline parameters have on the process, check the box marked "Check if debug files need to be generated!" This produces a set of tiff images of the registration process results. This process is what is affected by the spline parameters. The program is looking for a clear vein intersection involving the edge of the wing and veins III or IV. This distance must be greater than half a screen length from the starting landmarks. Once this point is found, the spline model can be rotated and scaled to match, and the fitting process can proceed. If the program cannot find any candidate intersections, it will fail to produce any spline at all. This causes the gaps noted in (9a) below. Spline a small fraction when using the debug images.
8. Another trick for finding a good set of parameters is to make a small data set of say 20 wings, and make an ASCII file for it by removing lines. **Copy** the files to a new subdirectory before doing this! Then when you try different spline parameters you can see if a new set actually improves the average spline quality. The program by default splines a new random subset every time if the spline fraction is less than 1. This randomness makes it harder to judge if the new parameters are better or worse. One pitfall in this is if you take consecutive numbered images then they are all likely to have the same lighting, which may not cover the range of values in the whole set.
9. Once a good set of parameters is found, at the main window set the spline fraction to 1.0, then relick 'Start Splining' on the main window with the so all the wings will be splined.
 - a. Note that sometimes if the parameters are not just right, some of the wings will not spline at all. If this is so, the program will pop up a note that, for example

only 98 out of 100 wings were splined. If this happens, you can continue to adjust the spline parameters to see if that allows all images to spline. If this does not work, you may need to find out which images are not splining, and process them separately by taking just the matching lines from the asc file, and putting that and the unsplined images in a separate subdirectory and going through this process again. Or you could say 98 out of 100 is pretty good and move on.

10. To check the quality of fit, and respline or correct those that are not accurately splined, click Edit Wings. The program will behave differently depending on how many flies of each sex there are:
 - a. If there are more than 25 of each sex, Wings will run a robust outlier detection routine based on minimum volume ellipsoids on the whole data set. Wings are ranked by their Mahalanobis distance from the centroid. When you start to see lots of images that do not need correction, you can discontinue your check.
 - b. If there are less than 25 images of each sex, then Wings will show you all of the splined images for checking, in no particular order. You should check all of the images. An alternative is to do your outlier detection in another program, then use the 'Edit Manually Selected Outlier' routine described in (13).
11. In either case, Wings shows you an editable picture of all the splines and the control points. This screen is shown in Figure 3. To change the location of the splines, grab a control point with your mouse, and move it where it seems best. Alternatively, you may click on Respline to try adjusting the splining parameters and respline each image, as described in (6) above. Before doing your editing, make sure that you have looked at a good sample of well-splined images to get a good idea of what the program considers a good fit. In many cases the program will repeatably put splines in a location different from where you would place them. For example, the program usually places splines along the posterior edge of the wing inside the place where a human would place them. The ideal use of the software is get repeatable data, and the best way to do that is to let the program have its way. The alternative is to correct the placement of veins on the majority of images, which is extremely time consuming.
12. It is good practice to go through the Edit Wings process once or twice more to make sure you didn't miss any outlier wings. If there are two splines that are badly placed, you can easily miss one of them on the first pass. This is particularly painless when the automatic outlier detection routine (10b) can be used.
13. A final procedure for dealing with outliers is useful when you have started to analyse the data in another program, and come up with your own list of possible outliers to check. To use this list, you can go back into wings and choose 'Edit manually selected outliers'. This allows you to choose from the list of all wings. If you send a wing to the right side of the screen then the program will show you that wings and allow you to edit it, if you wish to do so.

CPREADER

The wings program will create a .cp file with data from each of the wings. The first part of a .cp file looks like this:

```
# Image: crossbl573.tif 286.5 79.5 288.0 125.0 Yunily 03_Apr_02 Wed_PM F39-21xM39-54v2
M 0.006985684.0 Resolution: 1.000000 1.000000
```

9

14


```

287.936387 123.536901
280.173680 147.223963
243.396758 173.045201
182.313180 178.688463
155.460223 171.009150
127.945452 168.800180
78.535084 149.530000
50.550815 113.646119
67.005519 77.287631
129.405146 54.736720
230.262215 65.291896
255.874894 82.691561
267.957581 89.166542
286.575417 92.234599
5
153.337939 171.337700
165.054074 155.883137
189.648997 137.097543
242.659892 118.950596
287.398487 111.165197
8
65.674047 133.003511
.
.
.

```

The first line of the cp file just copies over the information about this wing from the .asc file. The second line (with just the number 9 on it) gives the number of spline curves to follow. The next line gives the number of control points on that spline (14 – so this is cp2 format – see above). Then follows 14 x, y pairs of coordinates. These are in pixels.

The program cpreader does two things: First it collects the information from many cp files into a single large matrix. These files will contain all the information about the fly, plus the locations of landmarks and control points on the wings. Second the program will perform Procrustes registration on the data using different subsets of the data. For less than a few thousand wings, you can run CPReader by clicking on the CPReader.jar file. For larger data sets, you may need to increase the memory allocated to the program. To do this, go to the command line (Start, Run from the Start menu in Windows XP), and say

```
C:\WINDOWS\system32\java.exe -jar -Xmx1024m D:\Flywing\Wings\CPReader.jar
```

The -Xmx1024m command allocates extra memory to the process (1024 MB in this example), and allows the registration of data sets of tens of thousands of wings.

Unfortunately, the CPREADER program is pretty primitive at this point. It does not do error checking, or give informative error messages. To get some feedback on what the program is doing (and where it goes wrong) run the program at the command line with the -jar option, as above.

The interface window for CPREADER is shown in Figure 5.

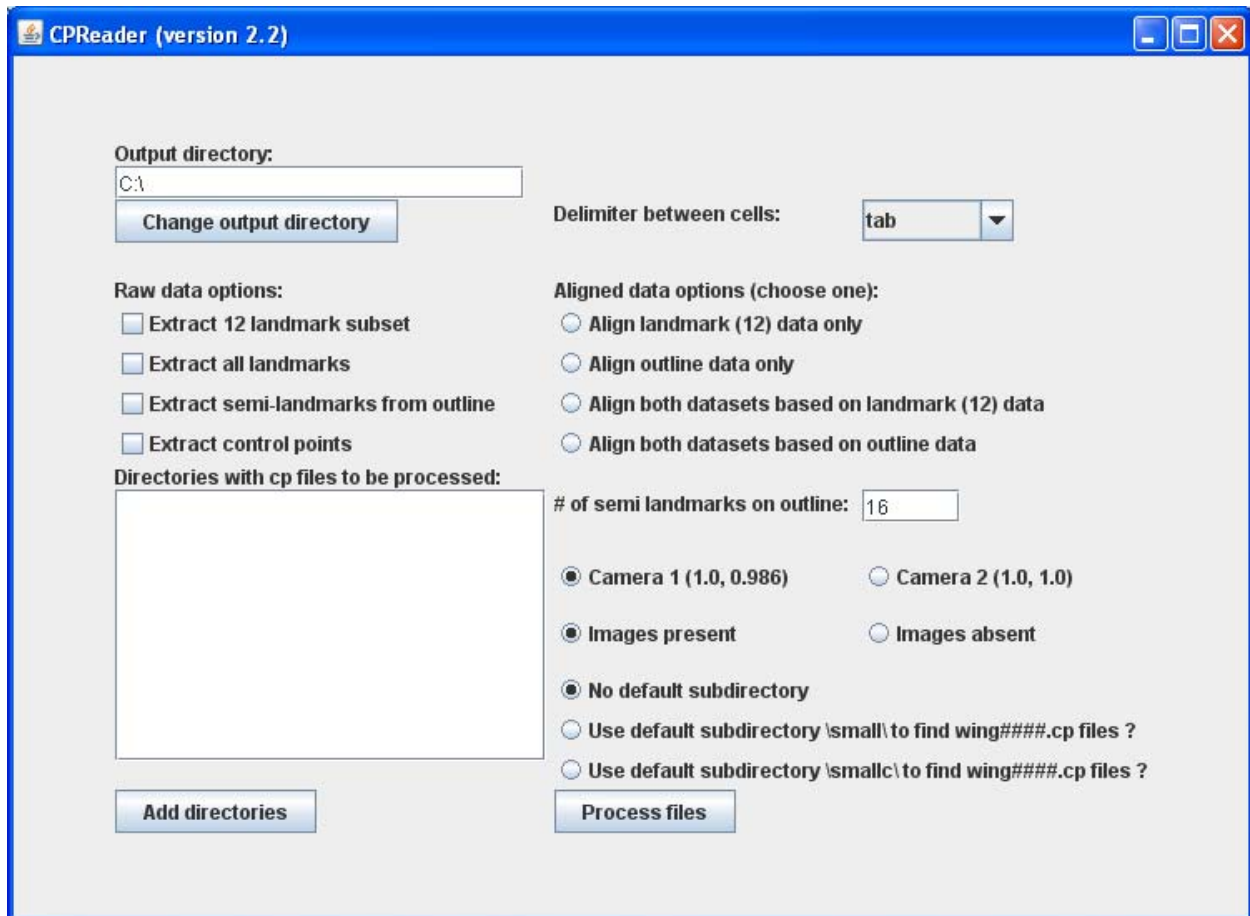


Figure 5. User interface of CPREADER.

The output directory is where the program will put the files it creates. On the left side of the screen are check boxes to determine what data is extracted. The 12 landmarks are the ends of splines internal to the wing. All landmarks add to the 12 internal landmarks, the ends of splines on the right side of the wing at the line connecting the starting landmarks. Semi-landmarks are evenly spaced points along the red curve in Figure 1, which is primarily the outline of the wing. If n semi-landmarks are extracted, then these are each $1/n$ th of the way along the length of the outline. Extract control points extracts the x,y coordinates of all the control points in the cp file.

The 'Add directories' button at the lower left brings up a folder navigation tool that allows any number of subdirectories to be examined for .cp files, and have their data extracted.

On the right side of the screen are options for registration (here called alignment) of the data. Either the 12 landmarks or the semi-landmarks on the outline can be used as the basis for registration. It is also possible to have the remainder of the points 'lifted along' with the registration points. We find it very useful to register based on the outlines, then analyze the landmark data. You will probably want to choose Camera 2 for your data, as this assumes that the pixels in your images are square. For one of our cameras this is not true.

When you hit the 'Process files' button, the button will turn dark, but no other change will be apparent until the program has finished. As the registration step can take quite a while when there are thousands of images, be patient. If the program runs successfully, the window will close at the termination of the registration step. If there is a problem, the 'Process files'

button will go back to being light, and the window will stay open. This probably means that there was an error.

Output files are given generic but decipherable names like Output.txt, Summary.txt, Output12.txt, Output12Aligned.txt, etc.

Example data set

Included in the distribution is an example data set of 100 wings from a *Drosophila virilis* population, two version of the ASC file, and two example starting model files. The file virilis.asc has the actual pixel coordinates and scale in pixels/mm of the starting points appropriate to the size images furnished. The file virilis2X.asc has these coordinates adjusted for images twice as large in both dimensions, and the scale reduced appropriately. Counterintuitively, it is the virilis2X.asc file that needs to be used for splining.

The two starting model files are named virmodel.cp2 and virilisextreme.cp2. Try splining with each model file. When virmodel is used, the proximal veins are usually misplaced, despite the fact that this model fit one wing quite well, as in Figure 1. When virilisextreme.cp2 is used, spines recovered are usually very good, despite the fact that virilisextreme fit its wing similarly to Figure 3. This is useful to see how critical the choice of a model file is.

Authorship and support

Wings and CPReader were written by Kim van der Linde, supported by NSF grant DEB 019129, and by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 RR021813. The underlying splining program was written by Feng Lu and supported by NSERC.