Wings 4 and CPR:
Curve finding, fitting

David Houle
Kim van der Linde
Eladio Marquez
dhoule@bio.fsu.edu
October 2014

Wings 4, takes images, detects curves that fit an *a priori* template, and then fits spline curves to them, as shown in Figure 1.

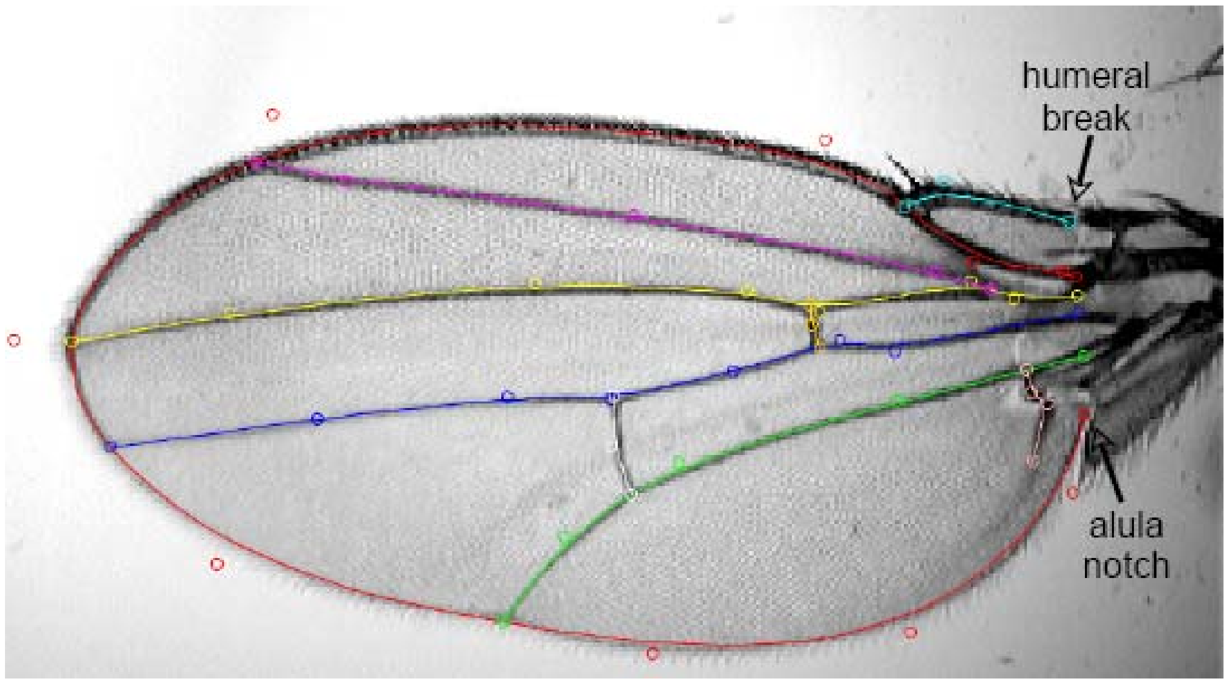


Figure 1. Nine B-splines fit to veins and edges of a Drosophila wing. Circles denote the positions of spline control points. The two starting landmarks that must be furnished to Wings are shown by the arrows. Starting landmark number 1 (in the ASC file) is the humeral break, and number 2 is the alula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

Wings4 is a JAVA program that will run on either Windows or Apple machines. 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.

What's a spline, and why use them?

Splines are continuous curves, so they can represent the edges of objects. They are essentially equations that give the location of the curve at any point between its end points. For an object like a fly wing, the most obvious features are the veins and edges of the wing, which are curves, so a spline can capture those well.

Wings4 fits quadratic B-splines, one choice from the family of possible spline equations. The data used to compute our spline curves are the spatial coordinates (x and y here) of a set of *control points*. For the wing in Fig. 1, the control points are the small circles. Note that they are generally not on the curve, with the exception of those at the end of a spline curve. For example, there are 5 control points on the green curve in Fig. 1, one at each end, and three in the middle that are not precisely on the curve. The relationship between control points and the curves is discussed in the section B-splines in Wings below. For you to use Wings, what you need to understand is that Wings fits curves to a wing image by moving control points to maximize the overlap between the spline curves and veins and edges of the wing captured in a digital image. Further technical details are in the section Understanding Splining below.

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. A video that explains this part of our Wingmachine is at https://www.youtube.com/watch?v=Nq-OIGKzdLk. 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 orientation landmark positions on the wing. We have not furnished our software for this, as this is hardware-dependent. We use, and would recommend the fully programmable package ImagePro for those with money to spend, or the free FIJI (ImageJ) package (http://fiji.sc/Fiji).

The two orientation 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 **Error! Reference source not found.**.

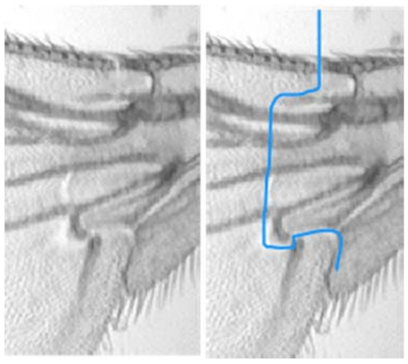


Figure 2. Approximate position of the wing hinge, shown by the blue line on the right image. Note that the alula is slightly folded from what is shown in Fig. 1, so as to obscure the location of the notch. The clear spot in the hinge, just above where the notch is a good marker to look for.

Note the clear spot that 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 in Houle et al. 2003.

The starting files

The raw images should be TIFF files. We use the extension ".tif". You also need an ASCII format file, with the extension .ASC, 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_02_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 orientation landmark 1 (humeral break, see Fig. 1).
3. The y coordinate in pixels of orientation landmark 1.
4. The x coordinate in pixels of orientation landmark 2 (alula notch, see Fig. 1).
5. The y coordinate in pixels of orientation landmark 2.
6. Columns 6, 8 and 9 can be used for any information you wish to record (with no internal spaces).
7. The date the image was recorded in any numeric format with separators (e.g. \, _)
8. Anything (see column 6).
9. Anything (see column 6).
10. Sex of the fly imaged.
11. The scale of the image in mm/pixel.
12. Other columns can follow column 11.

The Wings program will merely copy any of the information in columns 6 and 8-9, and 12 or more 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 location as 'Tallahassee, Florida', this will be read as two separate columns by Wings.

The X and Y coordinates assume that the upper left corner of the image is 0,0, and that going down is actually positive for the Y direction. Weird, huh?

The final type of starting file needed are template splines that Wings uses as a starting point in fitting each wing. These files have exactly the same format as the files that contain the data recovered from each image, which Wings calls .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 give a file extension '.cp#' where # is a single digit from 0 through 9. 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 spline fits. The principles of how to choose a good model are described in the Section Understanding Splining. 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 6 shows such an 'extreme' model that can be effective.

Installing Wings4

To use Wings4 you must install JAVA (JDK7 build 7), the latest version of the Java Runtime Environment. This is a free download available from http://www.java.com/en/download/. The Wing programs can be downloaded as a zip file from http://bio.fsu.edu/~dhoule/Software/Wings/. Check back there for updated programs. The current version is wings4.0 Beta 2.8.zip. To install, extract the entire zip file to a subdirectory of your choosing. This will make a new folder called Wings4, plus a few subdirectories that the program expects to find. Make a desktop shortcut to Wings4.jar before you forget where you put the program.

To start the program, click on Wings4.jar in the Wings4 subdirectory. This will first start a java virtual machine (which can take 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 ..\Wings4\supportFiles, which was created when you extracted the ZIP file.

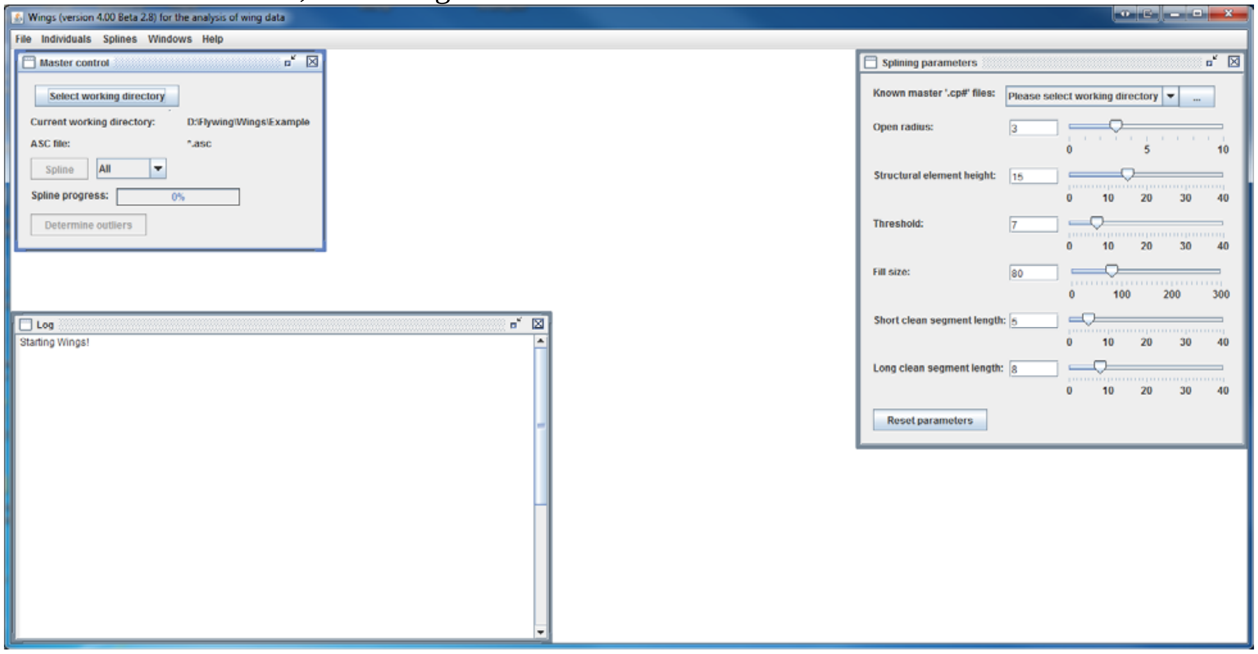
Before starting the program

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 Wings Example Set/DSmodel.cp2.

Wings program usage

The program for doing the splining is called Wings4.jar. Click on this file to start the program.

1. The program pops up two small windows, and asks you to input your name. This associates the name of the spliner with results produced in this session of Wings. Names can have spaces. Hit return.
2. You will see a window, containing three other window. Here is what it looks like:


Wings (version 4.00 Beta 2.8) for the analysis of wing data

Master control

Current working directory: D:\Flywing\Wings\Example

ASC file: *.asc

Spline progress: 0%

Log

```
Starting Wings!
```

Splining parameters

Known master '.cp#' files: Please select working directory

Open radius: 3

Structural element height: 15

Threshold: 7

Fill size: 80

Short clean segment length: 5

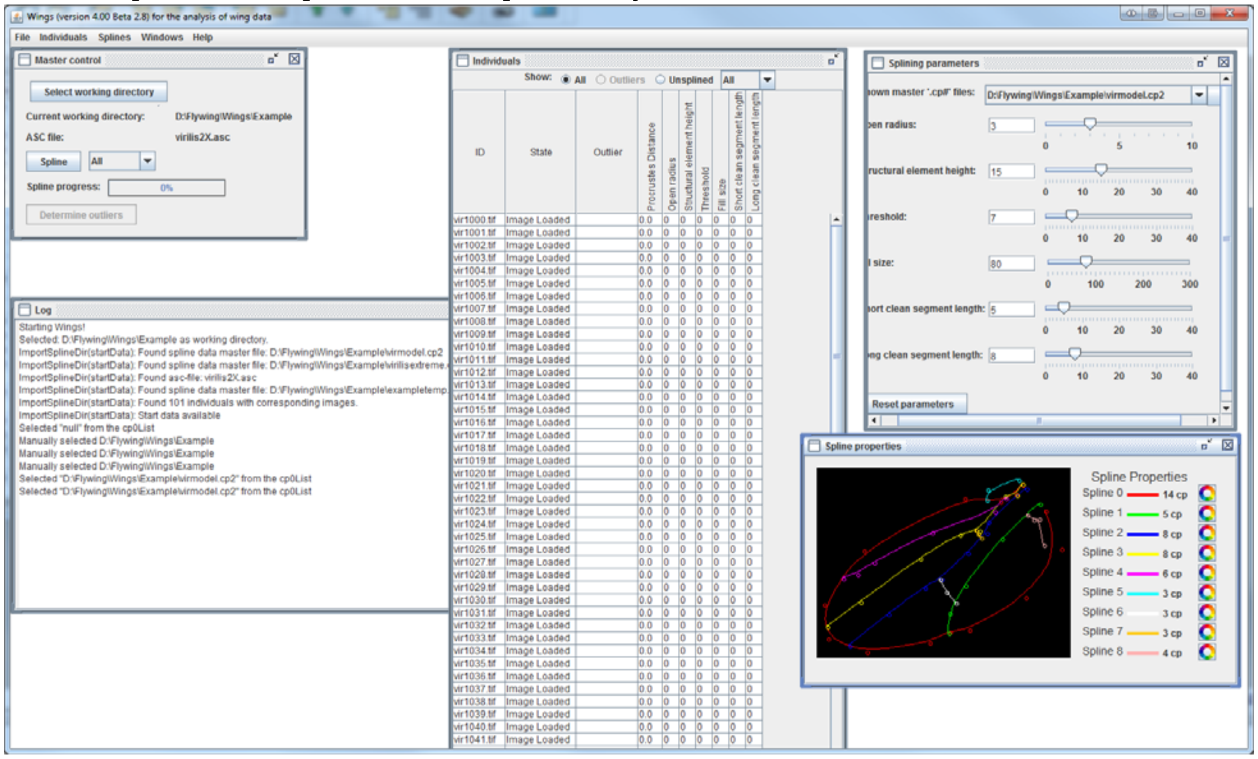
Long clean segment length: 8

Figure 3. Main screen of Wings when a session is first started.

The **Master Control** window is used to navigate to your files, choose an ASC file, spline images and detect outliers.
The **Splining Parameters** window is used to choose a template file, and basic image processing parameters.
The **Log** window gives you feedback on the progress of your session, including error messages and exceptions.

3. Hit the ***Select working directory*** button, then use the pop-up to navigate to the directory with the images to be splined and the corresponding ASC file. Wings will check your ASC file for format errors. If there is an error in your file, Wings will show you a message window describing the error, but there is currently no notice of the error after you close this window. No further progress is possible until an error-free ASC file is furnished. Edit your ASC file, then repeat your choice of working directory to reread the ASC file. Once an error-free file is read, and Wings will pop up a new window titled **Individuals** with a list of all the images, and their splining status.

4. Next, choose a master CP# file to use for the splining. Go to **Splining Parameters**, then check if your CP# file is already in the pull-down list. If not, click on the button marked '…'. If you don't see this button, expose the right side of the window by pulling the slider at the bottom of the window to the right. Once you hit the '…' button, you will see an Open File window that allows you to navigate to the subdirectory with the CP# file that you want to use. To actually choose the file, use the pull-down menu, which should now be populated with all the CP# files in this subdirectory. At this point, your screen should look like the following: The **Spline properties** window shows the positions of the spline control points and the splines for you chosen model.



Current working directory: D:\Flywing\Wings\Example
ASC file: virilis2X.asc
Spline: All
Spline progress: 0%

```
Starting Wings!
Selected: D:\Flywing\Wings\Example as working directory.
ImportSplineDir(startData): Found spline data master file: D:\Flywing\Wings\Example\virmodel.cp2
ImportSplineDir(startData): Found spline data master file: D:\Flywing\Wings\Example\virilisextreme.
ImportSplineDir(startData): Found asc-file: virilis2X.asc
ImportSplineDir(startData): Found spline data master file: D:\Flywing\Wings\Example\exampletemp.
ImportSplineDir(startData): Found 101 individuals with corresponding images.
ImportSplineDir(startData): Start data available
Selected "null" from the cp0List
Manually selected D:\Flywing\Wings\Example
Manually selected D:\Flywing\Wings\Example
Manually selected D:\Flywing\Wings\Example
Selected "D:\Flywing\Wings\Example\virmodel.cp2" from the cp0List
Selected "D:\Flywing\Wings\Example\virmodel.cp2" from the cp0List
```

Show: [✓ All] Outliers Unsplined All

ID	State	Outlier	Procrustes Distance	Open radius	Structural element height	Threshold	Fill size	Short clean segment length	Long clean segment length
vir1000.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1001.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1002.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1003.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1004.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1005.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1006.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1007.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1008.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1009.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1010.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1011.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1012.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1013.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1014.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1015.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1016.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1017.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1018.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1019.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1020.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1021.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1022.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1023.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1024.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1025.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1026.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1027.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1028.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1029.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1030.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1031.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1032.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1033.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1034.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1035.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1036.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1037.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1038.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1039.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1040.tif	Image Loaded		0.0	0	0	0	0	0	0
vir1041.tif	Image Loaded		0.0	0	0	0	0	0	0

own master '.cp#' files: D:\Flywing\Wings\Example\virmodel.cp2
pen radius: 3
ructural element height: 15
reshold: 7
l size: 80
ort clean segment length: 5
ng clean segment length: 8

Spline 0	🟥	14 cp
Spline 1	🟩	5 cp
Spline 2	🟦	8 cp
Spline 3	🟨	8 cp
Spline 4	🟪	6 cp
Spline 5	🟦	3 cp
Spline 6	⬜	3 cp
Spline 7	🟨	3 cp
Spline 8	🟥	4 cp

Figure 4. Wings4 screen once the ASC and spline model files have been chosen.

5. You may alter the choice of wings to be splined in the **Master Control** window by choosing from the drop down list next to the *Spline* button. *Fraction* splines a proportion of all images, chosen randomly. This can be useful when you have a large set of images and want to test whether the chosen **Splining parameters** will yield good splines. *All* is the default once some images have been splined, and will respline all the images regardless of whether they have already been splined and edited, losing any previous data. *Unsplined* will spline just those without corresponding .cp files. *Selected* resplines just individuals selected in the Individuals window. *Outliers* resplines individuals that have been identified as outliers (see below). Outliers are marked in red in **Individuals**.
6. Click the *Spline* button in **Master Control**. Wings begins splining, and updates the Individuals information to reflect progress. Wings pops up a **Tiff viewer** window to show the quality of splines.
7. The **Tiff Viewer** shows 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. This window is activated as soon as any individuals from your list have been splined, that is while splining is still ongoing. The parameters window is still active, and changes you make in that window will cause subsequent splines to be fit with whatever parameters you set there on the fly. We recommend allowing the splining to finish before using **Tiff Viewer** for resplining (see below).
8. If you see a wing in Tiff Viewer that is damaged, folded or incomplete, such that it cannot be measured accurately, click ***Reject*** to exclude it from your data set. If you respline, however, Wings will forget past rejections.
9. The wing in the **Tiff Viewer** window will immediately be resplined if you adjust the parameters in the **Splining parameters** window. If you have an incorrectly splined individual (including those wings where splining failed completely), you can rapidly click through alternative parameter combinations and directly observe how this alters splining. This is a good way to go about finding a set of spline parameters that are effective at splining most individuals. Generally speaking, *Open radius* values between 3 and 5 in combination with *Threshold* values from 5 to about 25 are worth investigating, although sometimes the *Structural element height* and the other parameters can be important. Once you are done resplining an image, click ***Next*** to save the results. If you click ***Cancel***, it will discard any splining done in this window, and keep the old spline as the best.
10. To investigate in detail why splining goes awry, and the effects of the parameters on the intermediate results, click on ***Respline*** button. This brings up the **Debug** window showing the results of image processing steps. See the Understanding Splining section below for a detailed discussion of the splining algorithm, and how to interpret the intermediate results displayed. Note that the **Tiff Viewer** window is not updated with any resplining results from this window until it is closed.
11. Once you have done the best you can using splining, you may edit the details of each spline by hand. There are two routes to editing. First, you can right-click each line in the **Individual** window, then select ***Edit*** from the menu, which starts the **Spline Editor** window. Second, you can use the ***Determine Outliers*** button in the **Master Control** window, described below. In **Spline Editor,** edit by grabbing control points and dragging them to new locations. The effects of this on the splines are updated instantaneously, allowing you to increase the fit to the image. To make full use of the

advantages of automated data recovery it is important that your goal in editing should be consistency of results with the way the program works, rather than matching your own ideal. Some editing practice is helpful, as it is often necessary to rearrange several control points to fit a misfit region of the wing. For example, shifting several control points toward areas of high curvature may be necessary, with compensatory changes elsewhere. Currently Wings cannot select blocks of individuals for editing.

12. If there are more than 25 splined individuals of each sex, a rapid route to finding badly splined individuals is by invoking the ***Determine Outliers*** button in the **Master Control** window. Wings runs 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. This routine can take some time complete for large numbers of wings (the ***Determine Outliers*** button will look depressed during this process).
    a. When complete, the **Spline Editor** window will open and display up to 10 wings that are NOT outliers. Do not edit these, as any changes will not be saved. Check these first 10 wings to understand the typical splining behavior for this set of parameters and model file. If you are not happy with the splines for these 10, you should return to the splining stage and alter the model and parameters to improve the typical splines before proceeding with detailed editing of splines.
    b. Once you have examined the 10 'good' wings, click on ***Continue with Outliers*** in the lower right corner of **Spline Editor**. **Spline Editor** now shows the outlier splines from the most atypical to the more typical ones. Above the image is information about how many outliers there are, the number of the current outlier, and why the wing is an outlier. Two separate outlier detection routines are run – one on the landmarks (intersections of wing veins), and one on the outline of the wing (the red curve in **Spline Editor**). A wing is classified as either an outline or landmark outlier, or both, which gives you an idea of where the spline might need to be corrected. The 'outlier value' reflects how unusual the wing is, and is a Mahalanobis distance, the number of standard deviations to the centroid of the sample. Values less than 3 will usually not be worth correcting.
    c. Each outlier wing can be edited by grabbing a control point and dragging it to a new location to increase the fit to the image. Some practice is helpful at this, as it is often necessary to rearrange several control points to fit a misfit region of the wing. For example, shifting several control points toward areas of high curvature may be necessary, with compensatory changes elsewhere. Once you are satisfied with the fit, hit the right arrow button to proceed to the next image.
    d. If you start to see lots of images that do not need correction, you can discontinue your check by clicking the ***Finished*** button.
13. Regardless of whether you completed the ***Determine Outliers*** routine, you should perform multivariate outlier detection in a statistical package to identify unusual wings that have escaped your editing. CPR, our cp reading program provides basic visualizations useful for detecting major outliers. Manually check at least the most unusual outliers as described above.
14. When done, exit the program using the File Quit command to ensure that all the information is properly saved.

Understanding spline fitting

Successful splining requires three elements: a good image, a good spline model, and image processing parameters appropriate to the image. To understand what makes image, model and parameters good, it is important to understand how Wings fits a spline model to a wing. There are two basic parts to fitting: orientation of model to image and then optimization of fit.

The orientation step tries to make a crude match of the model (from your cp# file, your 'ideal' spline) to the actual wing image. The initial orientation landmarks recorded when you take the wing image, and recorded in the ASC file provide Wings with the correct orientation of the wing. The next challenge is to find a vein intersection that is as far away from the orientation landmarks as possible. Once this distal intersection point is chosen, Wings scales and rotates the model relative to the image. If the distal intersection is either landmark 2 or 3, then all is well. If not, then the model and image will match poorly and optimization will probably converge on the wrong answer.

All of the parameters in **Splining Parameters** are involved in the orientation step. To see their effects, it is best to look at images in the Debug window, which is accessed by right-clicking an image in the **Individual** window, or clicking ***Respline*** from the **Tiff Viewer**. The window looks like:

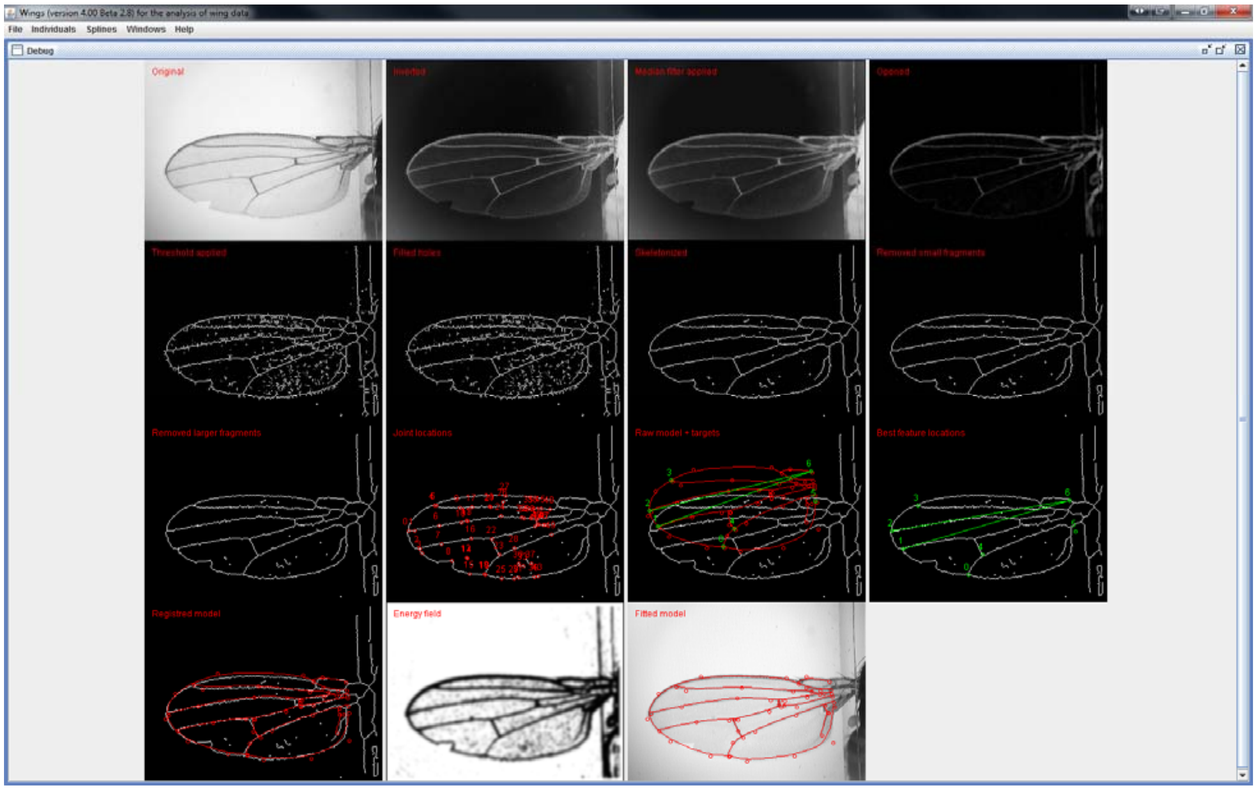


Figure 5. Debug window, showing the stages in orienting the model to image, and the fitting of model.

The program steps in orientation are:

1. Inverting the image (black becomes white, white black),
2. Application of a median filter.
3. An open operation, which both erodes messy edges of areas distinct from background, and the dilates the remaining foreground to match a pre-determined size square. The size of the square is controlled by the *Open Radius* parameter.

4. Thresholding remaining foreground areas that are above the intensity of the *Threshold* parameter.
5. Skeletonizing the white areas to single pixel width.
6. Filling small holes between fragments of white.
7. Reskeltonizing.
8. Removal of short fragments of white.
9. Removal of longer fragments of white.
10. Detection of joint locations – places ether two white lines intersect.
11. Wings then chooses the one or two intersections that are farthest from orientation landmark 1.
12. The model is then scaled, translated and rotated so that model landmark 6 is placed at orientation landmark 1, and model landmarks 2 and 3 are place on the farthest joints detected in the previous step.
13. An energy field is derived from the filtered image by detection of probable edges, and diffusion of such likely edges to provide a gradient from dark to light areas that extends a substantial distance away from the actual edge location.
14. The model control points are moved to increase the overlap of the spline curves with the dark areas energy field. When this process reaches a local optimum, the spline fitting is done.

In the end, there are four major modes of failure to spline.

First a set of parameters will not spline successfully if the structure of intersecting lines at the distal end of the wing is not detected. Check this in the Joint locations image in **Debug**. Any of the operations prior to joint detection may cause this loss of information.

Second, failure may result from noise in the distal parts of the images that cause false joints to be detected, say in the corner of the image, causing the splines to optimize on non-wing elements. Check this in the Joint locations image in **Debug**.

Third, if the energy field is either too far from the corresponding part of the superimposed model for a dark area to overlap the correct curve, or if a model curve ends up too close to the wrong part of the energy field, it will be attracted to the wrong solution.

Using a good starting spline model is one of the most important steps to getting repeatable, accurate spline fits. 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 nature of the energy field in the images will lead the spline routine to converge on the same type of improper solution, over and over. In such cases, it is often effective to choose a starting model that deviates from a good fit in one direction or another. Figure 6 shows such an ‘extreme’ model.

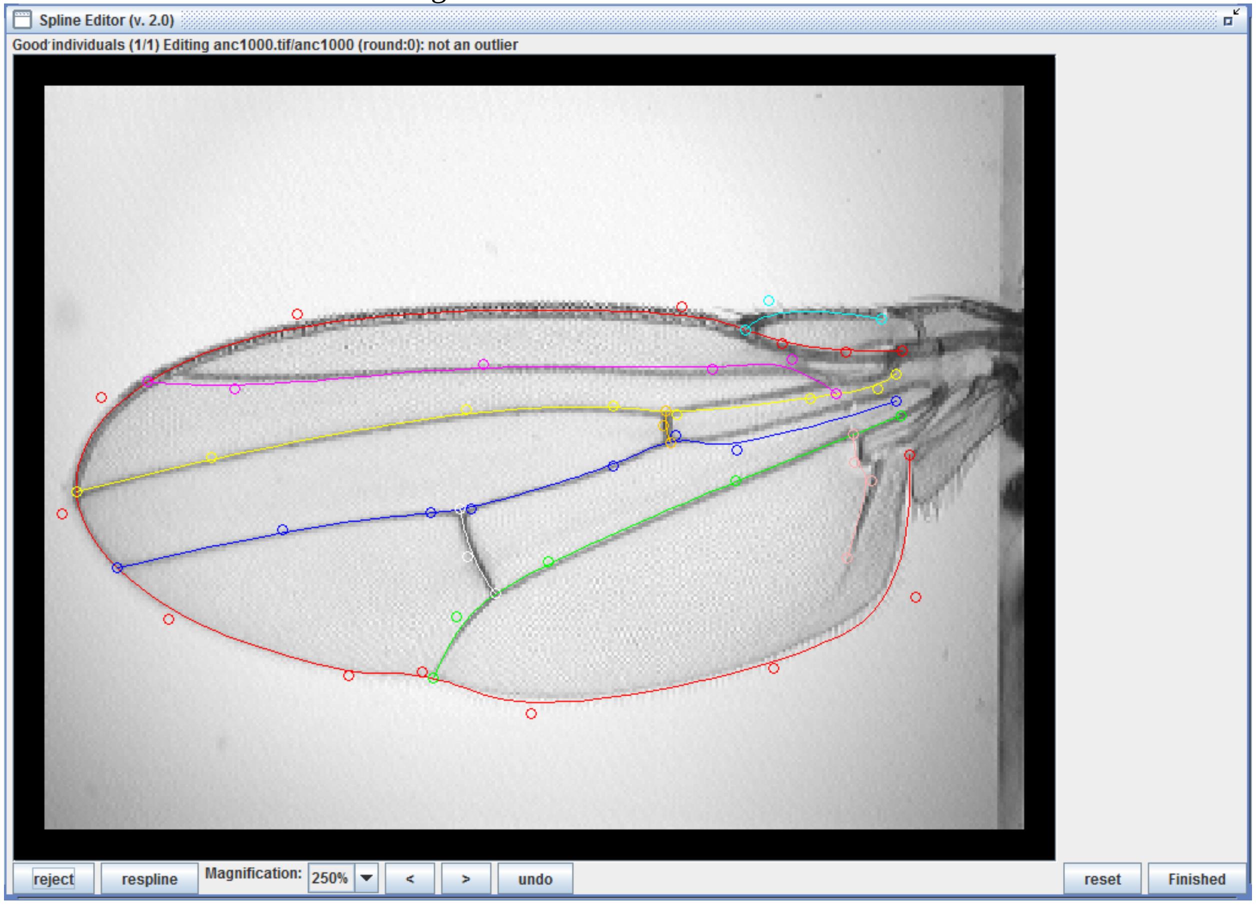

Spline Editor (v. 2.0)
Good individuals (1/1) Editing anc1000.tif/anc1000 (round:0): not an outlier
reject
respline
Magnification:
250%
<
>
undo
reset
Finished

Figure 6. Spline Editor window showing a set of splines altered to serve as an extreme model file. When finished editing a spline model to use as a model, right click on the corresponding individual in the Individuals window, and choose ***Save as model file***. Choose a file extension .cp# where # is a single digit, so the Wings will subsequently recognize that you intend to use it as a model.

This model would prevent, for example, vein 3 from being attracted vein1, vein 4 from being attracted to 3, and the outline curve from being attracted to vein 6.

The parameter most likely to change and potentially improve splining behavior is the threshold value. The default is 7, 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 5. 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.

Spline data format

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: crossb1573.tif 286.5 79.5 288.0 125.0 Yunily 03/04/02 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 in this case).   The next 14 lines give the *x* and *y* coordinates. There follow additional blocks giving the number of coordinates, then the *x*, *y* pairs. Coordinates are in pixels.

CPR: Cpreader

Our program for reading and working with spline data from .cp files is called CPR, short for CPReader. CPR incorporates the following functions:

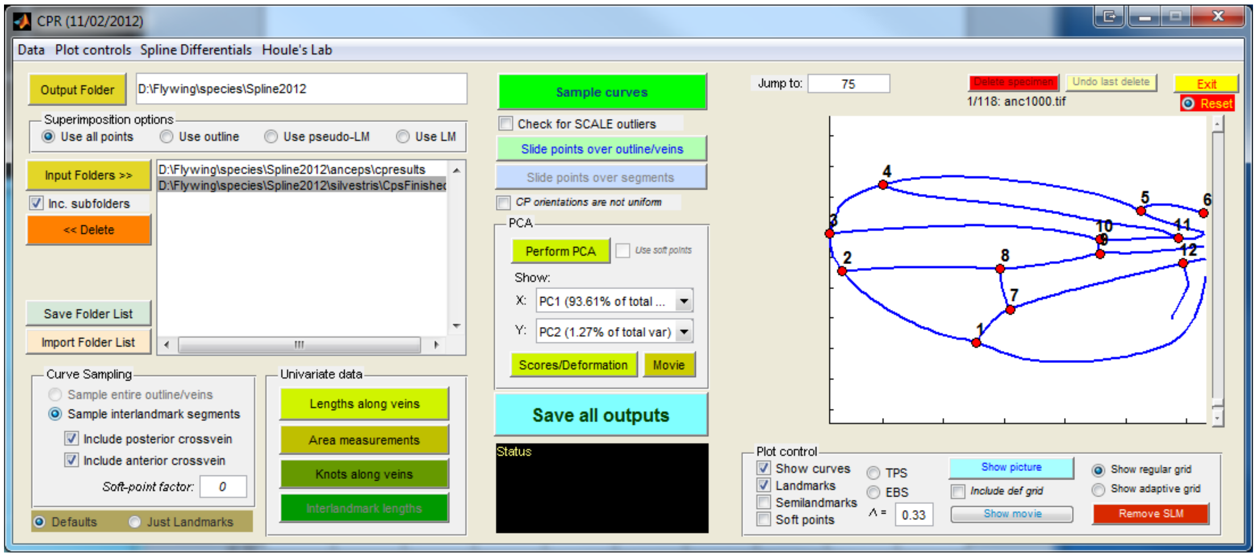
1. Collection of data from many .cp files in different subdirectories into a single large matrix, and saving the data matrix to a standard format.
2. Procrustes superimposition of curves to make them comparable and estimate overall size. Procrustes superimposition performs rigid scaling, translation and rotation to optimize fit of all curves to the mean curves, using a least-squares criterion.
3. Extraction of ‘semi-landmarks’ (sometimes called pseudolandmarks), the position of points along the wing veins between landmarks. The number of semi-landmarks can be controlled by the user.
4. Sliding of semi-landmarks to increase their comparability.
5. Extraction of univariate measures such as the distances between landmarks, lengths along curves, areas enclosed by veins, and the location of the knots.

6. Principal components analysis of the data, and visualization of the distribution along PC axes. This useful for finding major outliers.
7. Visualization of the splines, and deformations necessary to convert one set of splines into another. These visualizations are better implemented in the program Lory, also available from http://bio.fsu.edu/~dhoule/Software/ .

Installation of CPR requires just the CPR.exe file, plus Matlab files available from . . . .

CPR program usage

Click on the CPR.exe file.



Output Folder: D:\Flywing\species\Spline2012

Superimposition options: [✓ Use all points] Use outline Use pseudo-LM Use LM

Input Folders:
D:\Flywing\species\Spline2012\anceps\cpresults
D:\Flywing\species\Spline2012\silvestris\CpsFinished

[✓] Inc. subfolders

Curve Sampling: Sample entire outline/veins [✓ Sample interlandmark segments]
[✓] Include posterior crossvein
[✓] Include anterior crossvein
Soft-point factor: 0

[✓ Defaults] Just Landmarks

Univariate data

[ ] Check for SCALE outliers
[ ] CP orientations are not uniform

PCA
[ ] Use soft points
Show:
X: PC1 (93.61% of total ...
Y: PC2 (1.27% of total var)

Status

Jump to: 75
1/118: anc1000.tif
[✓ Reset]

Plot control
[✓] Show curves
[✓] Landmarks
[ ] Semilandmarks
[ ] Soft points
TPS EBS
Λ = 0.33
[ ] Include def grid
[✓ Show regular grid] Show adaptive grid

Figure 7. Main window of CPR. Two subdirectories of cp files have been read in using program defaults.

The main interface window appears, as shown in Figure 7.

1. Choose a destination folder for any output using the ***Output Folder*** button.
2. Choose one or more directories with the cp files that you wish to read in and analyze. Before inputting folders determine whether you want to include .cp files in all subfolders or not, and check the ***Inc. subfolders*** box appropriately. The list of folders chosen will appear in the window. You may delete folders from this list. If the list of folders to use is complex to construct, you may save the list to a file for later reuse.
3. The ***Curve Sampling*** section of the program at the lower left allows you to choose how many and what type of points on the curves that you wish to include in your output data. The major choices are:
    a. ***Just landmarks*** samples just 11 vein intersections, plus the humeral break, for a total of 12 Type I landmarks.
    b. ***Defaults*** enables sampling of points spaced along the curves, in addition to the landmarks. These additional points are called semi-landmarks, because their spacing along the curves is somewhat arbitrary. Choices for this are;
        i. ***Sample entire outline/veins*** treats each of the 9 curves as a separate entity, and then samples along each of them, without regard to intersection with other curves. This is currently not implemented.
        ii. ***Sample interlandmark segments*** samples semi-landmark points spaced equally along each curve segment between each pair of landmarks. Clicking on the ***Sample interlandmark segments*** text brings up a list (on two sequentially displayed boxes) of all those segments, along with the number of semi-landmarks to place on each segment. These choices can be edited by the user. Values of 0 are useful for excluding curve segments that are not reliably estimated in your data. In our work we have sometimes found that cross-vein position is subject to environmental variation, so we have made it possible to exclude or include cross-veins from the main screen, without resorting to the segment editing windows.
4. Choose the superimposition method desired. The choices alter which guide points that are used to determine superimposition parameters. Note that excluding points from serving as guides does not delete them from the data. All of the points chosen in the ***Curve Sampling*** section will be superimposed using the parameters estimated from the guide points. That is, the scaling, translation and rotation parameters will be applied to all points extracted. The choices of guide points are:
    a. ***Use all points***, including both landmarks and semi-landmarks, as guides.
    b. ***Use outline*** uses points on the outline curve as guides. This implements the recommended procedure in (Van der Linde and Houle 2009).
    c. ***Use pseudo-LM*** takes just the set of semi-landmarks as guides.
    d. ***Use LM*** takes just the 12 landmarks as guides.

5. Once you have made the above choices, click on the ***Sample curves*** button. This will read in all the .cp files in the selected subdirectories, sample semi-landmarks, superimpose the resulting data, and display the superimposed curves in the window on the right side of the screen.

6. You can use the window on the right hand side of the screen to see the aligned curves. Pulling the slider all the way to the bottom shows the superimposed points from all specimens.
7. ***Perform PCA*** does a principal components analysis, on the data, and opens a window with a two-dimensional plot of scores on two PCs. The user can choose to display plots of additional pairs of PC axes. Extreme points are numbered on the plots. Use the ***Jump to:*** box to display the curves for extreme individuals, and to find the name of the corresponding cp file. You can return to Wings to check and, if necessary, edit these individuals. The PCA section also can show either a ***Scores/Deformation*** plot or ***Movie*** of variation along the PC chosen as the X variable.
8. If you have chosen to work with semi-landmarks, you should slide the semi-landmarks to maximally comparable positions using the ***Slide points over outline/veins*** button. This is a computationally intensive step that can take some time. Data sets of hundreds of configurations finish in minutes, but if many thousands of individuals are included, this step can take days. There is currently no visual indication that this option is invoked, although eventually the cursor (when positioned over the CPR window) will show the processing configuration. A chime will sound when sliding has been completed. We recommend doing a PCA both before and after this step. A PCA analysis on the pre-slid data will still enable the detection of most outliers, so that you do not have to repeat this very time-consuming step.
9. ***Show picture*** in the Plot Control box shows an overlay of the curves on the current individual in the viewer.
10. Additional visualizations are implemented from the Plot Control box, but these are not as useful here as they are in the companion program Lory available from bio.fsu.edu/~dhoule .
11. Before exiting, click ***Save all outputs*** the large blue button. This saves the output to the directory chosen in step 1, with generic names and the .dat extension. The names are descriptive of their contents, so if one extracts curve sampling, then a file with all the landmarks and semi-landmarks will be written called Output_Landmarks+Veins+Outline.dat.

CPR output format

CPR output files are tab-delimited text files, with one wing per line. They include a header in the first line with variable names. Here is an example:

```
CPFile  File  O1x  O1y  O2x  O2y  Perp  Date  Time  Tags  Scale  Sex  x1  y1  . . .  x49  y49  CS

anc1000  D: \cpresults\anc1000.tif  540  151  558  239  Jeff  19_03_05  Sat_PM  1261.10  0.0045893  F  -0.06010178  -0.09720887
. . .   -0.03176999  -0.03889509   2.951
```

The variables in order are:

1. CPFile – the name of the cp file.
2. File – the path to the splined image.
3. O1x – The x coordinate in pixels of orientation landmark 1 (humeral break, see Fig. 1).
4. O1y – The y coordinate in pixels of orientation landmark 1.
5. O2x – The x coordinate in pixels of orientation landmark 2 (alula notch, see Fig. 1).
6. O2y – The y coordinate in pixels of orientation landmark 2.

7. Perp – The 6th column of information from the ASC file. In the Houle lab, this is the name of the person that imaged the wing.
8. Date – The date the image was recorded.
9. Time – The 8th column of information from the ASC file. In the Houle lab, this is the time of day the wing was imaged.
13. Tags – The 9th column of information from the ASC file. In the Houle lab, this is the genotype of the fly.
14. Scale – The number of mm/pixel in the image file.
15. Sex – Sex of the fly imaged.

x1, y1, x2, y2, . . . x*n*, y*n* – Any number (*n*) of pairs of x and y coordinates corresponding to landmarks and semi-landmarks. The first 12 pairs are the Type 1 landmarks formed by vein intersections and the humeral break. The remaining *n*-12 pairs are semilandmarks.
CS – centroid size in mms. Centroid size is measured using all the *n* points.

Example data set

Included in the distribution is an example data set of 100 wings from a *Drosophila melanogaster* population, an ASC file with basic onformation about each specimen, and two example model (CP2) files. The population is from an unpublished experiment in which flies have been selected to differ from wild-type shape in the direction of upregulation of the gene *dachsous*. The file DachsousDN.asc has the actual pixel coordinates and scale in pixels/mm of the starting points appropriate to the size images furnished.

The two starting model files are named DSmodel.cp2 and DachsousExtreme.cp2. Try splining with each model file. DSmodel starts with a closer match to the typical wing, while DachsousExtreme.cp2 resembles the model shown in Fig. 6. This is useful to see how critical the choice of a model file is.

B-splines used in Wings

Splines are piecewise polynomial functions parameterized with the spatial locations of points. We use a parameterization based on *control points*, which are not necessarily on the curve that is being represented. *Knots* are points on the curve that are the mean position of adjacent pairs of control points. Piecewise means that the full curve is a combination of *n* short curves that are interpolated based on the subset of the control points that are nearest the section of curve being reconstructed. Assume that we have a curve running generally left to right. The control points are numbered sequentially from left to right from 0 to *n+1*. Wings uses a quadratic parameterization that uses the three closest control points to generate the curve. For example, consider the *i*th knot that is the mean of control points *i*-1 and *i*. To the left of this knot, the three control points used for interpolating the curve are *i*-2, *i*-1 and *i*. To the right of this point, we use points *i*-1, *i*, and *i*+1. Choosing a higher number of control points to represent a curve allows it to have a larger number of changes in curvature, that is to be more wiggly. The number of control points for each curve on the wing was chosen so that the resulting curve fits well over a variety of wings. Simple curves, such as the cross-veins, are represented by just three control points, while the more complex outline of the wing takes 14 control points.

Given the set of control points $\left\{\left(c_{x0}, c_{y0}\right), \left(c_{x1}, c_{y1}\right), \cdots, \left(c_{x(n+1)}, c_{y(n+1)}\right)\right\}$, the corresponding knots are

$$k_{x(i)} = \left(c_{x(i-1)} + c_{x(i)}\right)/2$$
$$k_{y(i)} = \left(c_{y(i-1)} + c_{y(i)}\right)/2.$$

The *i*th curve segment is runs from knot *i* to knot *i*+1. The location of the point $\left(x_{i\bullet p}, y_{i\bullet p}\right)$ that is proportion *p* of the length between knot *i* and knot *i*+1 is

$$x_{i\bullet p} = c_{x(i-1)}w_1 + c_{x(i)}w_2 + c_{x(i+1)}w_3$$
$$y_{i\bullet p} = c_{y(i-1)}w_1 + c_{y(i)}w_2 + c_{y(i+1)}w_3, \text{ where}$$
$$w_1 = \frac{(1-p)^2}{2}$$
$$w_2 = \frac{1}{2} + p - p^2$$
$$w_3 = \frac{p^2}{2}.$$

Each segment of curve can be reconstructed to the degree desired by sampling *p* in the interval p =0 to p =1, with the extreme values 0 and 1 returning the *i*th and *i*+1th knots respectively.

Wings' output returns the first and last knots (the actual ends of the curves) instead of the 0th and *n*+1th control points. This is convenient as the ends of many of the curves are Type I landmarks that can be used directly in geometric morphometric analyses. To actually compute curves (semi-landmarks in geometric morphometrics jargon) from Wings output, these end knots can either be converted to their corresponding control points as

$$c_{x0} = 2k_{x1} - c_{x1}$$
$$c_{y0} = 2k_{y1} - c_{y1}, \text{ and}$$
$$c_{x(n+1)} = 2k_{x(n)} - c_{x(n)}$$
$$c_{y(n+1)} = 2k_{y(n)} - c_{y(n)}$$

or the weights of the 1st and last curves adjusted to compensate for the combination of knots and control points.

Known issues and planned features

Wings issues – for KIM

1. Sometimes sex of an individual is not known. Relax the error checking on this to allow other codes than M and F – perhaps just a U with a note to that effect ('Error in sex – permissible values are M, F or U.') when there is an error.

   MVE error checking relies on proper assignment of sex. I ran into the same problem with pseudomales. This requires a bit more rethinking than just adding a third category.

   For the moment, the program will accept U, Unknown and Unasigned when sex is unknown. As it generally is a small class, MVE is useless, but they will show up in the table as unedited and unfinished and can be caught that way.

2. Implement an unusualness order for all individuals using Mahalanobis distance for data sets with too few for MVE.

   It can be done, but I think the approach with marking the finished images is a more effective way to achieve the same.

   Alternatively, I think determining the distance of the 'unfinished' images to the 'finished' images might be a far faster method to find outliers that can be done on the fly.

~~3.~~ Is memory still a problem? User manual item on memory and how to solve problems. Preset memory limit that the user can adjust upwards at their own risk.

4. Memory limitations. Sensible warning behavior when memory is exceeded.

Things that might be nice in the optimal world

5. Model creation module is not done.
6. General asc file column reading solution.

Authorship and support

Wings was written by Kim van der Linde, supported by NSF grants DEB 019129 and 0950002, and by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 RR021813. The underlying spline fitting algorithms were originally written by Feng Lu and supported by NSERC. CPR was written by Eladio Marquez, supported by NIH R01 GM094424, and NSF DEB-0950002.

Van der Linde, K., and D. Houle. 2009. Inferring the nature of allometry from geometric data. Evolutionary Biology OnLine First.