Run2dmorph (Ver.2016.09.21) Manual

Allison Hsiang

2016-11-22

I Introduction

Run2dmorph is the 2D-data extraction module of the *AutoMorph* software package developed by Pincelli Hull and team [1]. *AutoMorph* is available for download on GitHub. The *run2dmorph* module was written by Allison Hsiang and is described in detail in Hsiang *et al.* [2].

The basic pipeline of *run2dmorph* is as follows: the user supplies the program with a directory of RGB images containing individual objects (one object per image; all potentially confounding information, such as labels, should be excluded). *Run2dmorph* expects light objects on a dark background. If the user is using *run2dmorph*'s companion *AutoMorph* modules *segment* and *focus*, the 'focused_unlabeled' folder outputted by *focus* should be used as the input directory of images for *run2dmorph*. The images are run through a series of filters (Fig. 1) that converts the original color image into black and white, with black indicating background and white indicating the object. The 2D outline is then extracted from this black and white image, along with measures of enclosed area, eccentricity, major and minor axis length, perimeter length, rugosity, and aspect ratio.

Run2dmorph, like all *AutoMorph* modules, is run using the command line. On Mac OSX, you must install the GNU Coreutils command line tools (more information here) in order to use the essential UNIX commands (*e.g.*, 'ls' and cat') and run the *run2dmorph* binary executable. If you are unfamiliar with the command line, we recommend searching for introductory tutorials online and familiarizing yourself before diving in (here are some suggestions). A good golden rule when dealing with the command line as a beginner is: never input a command if you don't know exactly what it will do!

Run2dmorph is run using a plain-text input file, referred to here as a 'control file'. Explanations for the user-controlled variables in the control file, and how to set them, are presented in section IV. By default, *run2dmorph* will output greyscale versions of the original images with the extracted outline overlaid for checking outline extraction fidelity, a CSV file containing the user input parameters, and a CSV of all 2D shape measurements. Optional output includes x,y-coordinates for the extracted outlines and intermediate images of the output of each image processing filter step.

If *run2dmorph* fails to extract an object, it is usually due to the input parameters being poorly suited to the object image. In this case, *run2dmorph* will skip the object and move on to the next. If any objects are skipped in this way, *run2dmorph* will also output a TXT file listing all skipped objects, and generate a folder nested within the output directory named ’no_outline_extracted’ that contains a copy of all skipped objects to facilitate easy re-processing.

II Installation

II.1 Prerequisites

Run2dmorph currently requires MATLAB (version R2014b or higher) and only runs on Unix-like systems, including Mac OSX. Removal of dependence on proprietary software, as well as PC compatibility, is currently in development. MATLAB must be available in your system’s path and callable via the command ’matlab’. On Mac OSX, this can be achieved by creating a symbolic link to the MATLAB binary executable:

```
ln -s /IPATH/MATLAB_VERSION.app/bin/matlab /usr/local/bin/matlab
```

where IPATH is the installation location of MATLAB, for example /Applications, and VERSION is your MATLAB installation version number, for example R2016b. After entering your system password when prompted, you will now be able to use MATLAB from anywhere in your system, so *run2dmorph* will be able to call MATLAB as necessary.

II.2 Setup

Once you have downloaded the *AutoMorph* software package, you will find *run2dmorph* in the ’run2dmorph’ folder. The ’bin’ folder contains the *run2dmorph* executable, while the ’src’ folder contains the MATLAB source code and a default control file, in which the user sets *run2dmorph*’s various parameters.

We recommend adding the *run2dmorph* executable to your path, so that *run2dmorph* can be called from anywhere in your system. To do this on Mac OSX, open the Terminal program (located at /Applications/Utilities/Terminal), and type the following command at the prompt:

```
sudo cp AMPATH/AutoMorph/run2dmorph/bin/run2dmorph /usr/local/bin
```

where AMPATH is the location of your *AutoMorph* installation. For example, if *AutoMorph* is located in /Applications, the full command would be:

```
sudo cp /Applications/AutoMorph/run2dmorph/bin/run2dmorph /usr/local/bin
```

Enter your system user password when prompted and hit enter. You can now use *run2dmorph* from any location on your computer.

III Quick Run

Once *run2dmorph* is installed, it can be run via the command line using the following command (assuming the *run2dmorph* executable in your path):

```
run2dmorph <path to control file>
```

While running, *run2dmorph* will output status messages on its progress. It will also indicate when an outline cannot be extracted for a particular object (this is usually due to inappropriate image processing parameters, namely **Intensity Range In**, **Intensity Range Out**, **Gamma**, and **Threshold Adjustment**).

IV Control File and Parameters

The control file serves as the means by which the user supplies the necessary parameters for *run2dmorph* to operate. A default version of the control file can be found in AutoMorph/run2dmorph/src. Within the control file, each parameter is briefly described. If the parameter requires user input, it is marked as such; otherwise, the default value of the parameter is listed (although the user can, of course, change these as they see fit). It is recommended that the user read through the entire control file before using *run2dmorph*. The control file should be edited using a plain text editor (*e.g.*, TextWrangler for Mac OSX) to avoid interpretation issues and must use Unix (LF) encoding.

For parameters that are not required, the default value can be used by setting the parameter equal to '[]', as such:

```
output_dir = []
```

A listing of the parameters for *run2dmorph* follows, with required parameters marked with a *:

***Installation Directory**: the full path of the system's *AutoMorph* installation. This path cannot be a relative path. For example, if the installation of *AutoMorph* is in /Applications/Morphometrics, **../Morphometrics** would not be not acceptable as input, even if the user runs *run2dmorph* from the /Applications directory. The full path (**/Applications/Morphometrics**) would be required.

***Image Directory**: the full path (*not* relative) of the folder that contains the image files from which the 2D outline and measures will be extracted.

Output Directory: the full path (*not* relative) of the folder to which *run2dmorph*'s output will be saved. If the folder does not exist, *run2dmorph* will create it. If this parameter is left as default, *run2dmorph* will generate a folder named 'morph2d' inside the directory specified under 'Directory'.

***Image Extension**: the file extension of the images to be processed by *run2dmorph.* This should be formatted with the dot - so **.tif** is correct, but **tif** is not.

***Sample ID**: a name designating the identity of the current run, which will be appended before all files that *run2dmorph* outputs. We recommend using a Sample ID that corresponds to the identity of images in the directory to be processed by *run2dmorph.*

Microns Per Pixel X: a number specifying a conversion factor for pixel size in the width of the input image. *Run2dmorph* will scale the width of all images by this factor before extracting outlines and shape measures. Despite the parameter name, the conversion factor need not be microns; the user should choose the conversion factor most appropriate for their input images. The default value is 1 micron per pixel.

Microns Per Pixel Y: as above, but for the height of the image. The default value is 1 micron per pixel.

Get Coordinates: a boolean (*i.e.*, TRUE or FALSE) specifying whether *run2dmorph* will output x,y-coordinates for the outlines extracted from each image. The default value is TRUE.

Save Intermediates: a boolean that determines whether *run2dmorph* will output intermediate images of the output of every image processing filtering step (*i.e.*, the images shown in Fig. 1). The default value is FALSE; we suggest turning this option on when users are conducting initial exploration of the most appropriate filter parameter values and when troubleshooting poor outline extraction.

Intensity Range In: a range specified in the format [$\boldsymbol{low_{in}}$ $\boldsymbol{high_{in}}$] that gives the input intensity range for the gamma filtering step. This is generally used to increase the contrast between the background and the object by lightening darker values in the input image by mapping the $\boldsymbol{low_{in}}$ value to the $\boldsymbol{low_{out}}$ value and the $\boldsymbol{high_{in}}$ value to the $\boldsymbol{high_{out}}$ value. More information can be found in the documentation for the *imadjust* function of the MATLAB image processing toolbox. The default range is [0 0.2].

Intensity Range Out: a range specified in the format [$\boldsymbol{low_{out}}$ $\boldsymbol{high_{out}}$] that gives the output intensity range for the gamma filtering step. The default range is [0 1].

Gamma: a number that specifies the gamma value for the gamma filtering step. Gamma values <1 result in a concave-down gamma correction curve (*i.e.*, the intensity range mapping is weighted toward brighter output values); gamma values >1 result in a concave-up gamma correction curve (*i.e.*, the intensity range mapping is weighted toward darker output values); and gamma = 1 results in a linear mapping (Fig. 2). The default value is 2.

Threshold Adjustment: a number that specifies an adjustment value that will be added to the threshold value that is automatically determined by the MATLAB function *im2bw* during conversion of the image from greyscale to black and white. A higher threshold adjustment value corresponds to higher tolerance during the conversion (*i.e.*, a broader, lighter range of grey will be considered ?white? in the final black and white image). Note that while this adjustment value can be negative, the *total* threshold adjustment value cannot be negative. The default value is 0.

Smoothing Sigma: an integer that controls the amount of smoothing applied to the edge outline before rugosity calculation; this is the controlling parameter for the canny smoothing algorithm, as implemented under the MATLAB *edge* function. A higher sigma value corresponds to higher amounts of smoothing. The default value is 7.

Noise Removal Limit: a number that specifies the size (in percentage of total image size) below which isolated objects will be removed from the image during the noise removal step. For instance, a noise removal limit of 0.05 will result in all objects smaller than 5% of the total image size being removed. The default value is 0.05.

Downsample: a boolean specifying whether x,y-coordinates extracted by *run2dmorph* should be downsampled from the default number of points (equal to the number of pixels in the outline extracted using the MATLAB function *bwboundaries*). The default value is TRUE. This parameter is only applicable when the **Get Coordinates** parameter is TRUE.

Number of Points: an integer specifying the number of x,y-coordinates to be outputted if the **Downsample** parameter is set to TRUE. For example, setting this value to 100 will result in 100 outline x,y-coordinates outputted for every object processed. The default value is 100.

Draw AR: a boolean specifying whether images showing the minimum bounding box (MBB) that encloses the extracted outline of each object, and the resulting aspect ratio that is calculated from the MBB, should be outputted. The default value is TRUE.

V Hands-On Example Run

A set of example images for testing *run2dmorph* can be dowloaded on Zenodo here. A control file that can be used with these images is located at /AutoMorph/run2dmorph/examples/run2dmorph_control_4sq_example.txt. The user need only change the **Installation Directory** and **Image Directory** parameters to run the example. Step-by-step instructions – which assume that the user's installation of *AutoMorph* is in /Applications, and that the user has added *run2dmorph* to their system's path – follow:

1. Open run2dmorph_control_4sq_example.txt in a plain-text editor and change the **Installation Directory** and **Image Directory** to the appropriate paths:

```
# Installation Directory:
install_dir = /Applications

# Image Directory:
directory = /Applications/AutoMorph/run2dmorph/examples/4sq
```

2. At the command line prompt, navigate to the folder containing the control file:

```
cd /Applications/AutoMorph/run2dmorph/examples
```

3. Start *run2dmorph* by entering the following command:

```
run2dmorph run2dmorph_control_4sq_example.txt
```

Run2dmorph will proceed to extract outlines from the example objects; when extraction is complete, the command line prompt will reappear. The output files will be located in: /Applications/AutoMorph/run2dmorph/examples/4sq/morph2d.

VI Troubleshooting

Based on in-house usage of *run2dmorph*, the most common issues that arise concern incorrect specification of parameters in the control file. In general, if *run2dmorph* encounters an error before outline extraction begins, something is likely wrong with the control file formatting or encoding. In this case, the user is advised to:

- Check that the encoding of the control file is set to Unix (LF) and that the control file is in plain text format
- Check that the correct input directory containing the images is specified
- Check that the parameters are formatted correctly; in particular, a common issue is specifying parameters inside brackets (*e.g.*, specifying **Microns Per Pixel X** as [**0.975**] instead of **0.975**). The only exceptions to this are the **Intensity Range In** and **Intensity Range Out** parameters, which *should* be specified within brackets as a range (*e.g.*, [**0 0.2**] instead of **0 0.2**).

If *run2dmorph* runs but does not extract outlines, the problem likely lies in the image processing parameters; in this case, the user is recommended to make a test folder of 5-10 images from their full sample, turn on the **Save Intermediates** parameter, and adjust parameter values until acceptable outlines are extracted for the test objects before running *run2dmorph* on the entire sample.

References

[1] *AutoMorph* (https://github.com/HullLab/AutoMorph)

[2] Hsiang AY, Nelson K, Dobbins B, Elder LE, Liu Y, Hull PM. AutoMorph: Accelerating community morphometrics with 2D and 3D image processing and shape extraction. **Methods in Ecology and Evolution**. *In prep.*

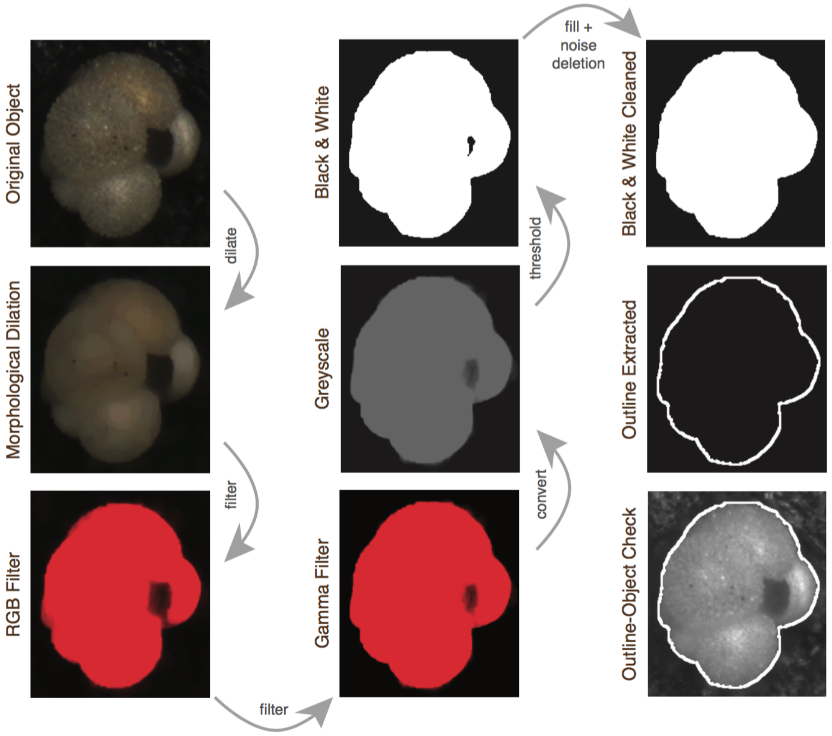

Figure 1: Visual pipeline of the image processing filters used in *run2dmorph*.

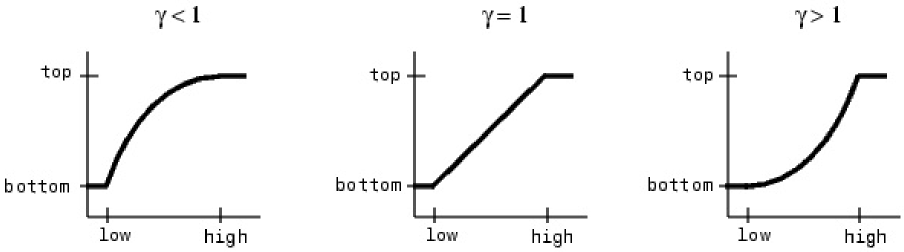

Figure 2: Relationship of intensity range in (x-axis) to intensity range out (y-axis) with varying values of gamma. Figure from MATLAB documentation.