Using the system (with troubleshooting notes at the end)

Startup and calibration:

To start the software, type `python3 AMiGUI.py`. You should get a welcome message on the terminal, the three axes should find their home position, and the GUI should open. It is essential that the positions where the four corner samples are centered and in focus be saved before automated imaging or manual viewing. The corner positions will be initially read from a configuration file (AMi`.config`), and the software with exit with an error message if this file is not present in the directory from which the software is launched. Those who use more than one type of multi-well plate will most likely want to create a separate configuration file for each type of plate. To read a different configuration file, enter the file name in the lower part of the GUI and then click "read config." Even if an appropriate configuration file is present, it is generally a good idea to check the positions of at least one of the corner drops. Clicking the four blue buttons drives the stage to the selected corner. (If you are using a plate with more than one sample well at each position, center and focus on the top-right sample at each corner.) The precise X, Y position of the stage and the focus (Z) can then be adjusted using the top two windows. Clicking farther from the center results in exponentially larger movements. Once a corner sample is centered and in focus click SET to save the position. Then move on to the next corner. To write the saved coordinates to the configuration file click "write config." A configuration file will also be automatically written to the directory with the images when automated imaging is initiated.

An additional calibration step is necessary when using plates with more than one sample at each position (i.e. 96-well plates with three sub-wells at each of the 96 positions). The position of the sub-wells should not be entered until all four of the top-right corner wells have been set. To calibrate the sub-wells, the number of samples per position should be entered into the box at the lower right of the GUI. Values greater than 1 in this box alter the behavior of the blue TL button. Right-clicking the button allows you to set the position of the 2nd sub-well, right-clicking it twice sets the position of the

3rd sub-well, etc. There is no need to describe the position of the sub-wells at the other corners. When more than one sub-well is present, the image names will have a lower case lettered suffix (i.e. D11a, D11b, ...). The sub-well information is saved to the configuration file when you click "`save file`."

Manual inspection:

As discussed in the main text, once the positions of the four corners (and sub-wells, if present) have been set, AMi can be used to manually inspect the plate. Left-clicking the Next and Prev buttons move the view forward or backwards. Right clicking these buttons moves the view up or down a row. The focus and position can be adjusted using the X,Y and Z windows, but these changes have no effect on subsequent Next or Prev operations. Clicking SNAP captures an image which is written to the directory specified by the plate_ID field in the lower part of the GUI. Snapped image files have a time-stamp within their name.

Automated imaging:

As discussed in the main text, automated imaging is initiated by clicking RUN. (If a run is started in error, click Stop/Close just once to stop the run. (Clicking twice will cause the program to end completely.) The plate_ID field and the screen field in the GUI, as well as the date, determines the location of the image files, and a script for depth of field enhancement is written along with the jpg images. The critical parameters for automated imaging are the number of z-stacked images per sample and the distance in z between these images. To determine the depth of field (this gets smaller as the magnification increases) choose a small, well defined object in one of your wells. Then translate in z until it is just barely in focus. The terminal window reports the x,y,z coordinates (in mm every time you make a movement. Record the z value, when the object you chose is just beginning to go out of focus. Then move z in the opposite direction until the object is again just barely in focus and record z again. The difference between these is the depth of field. A similar procedure can be used to determine the depth of your drops. It is best to set the z-spacing about 20% smaller than the depth of field. Also, the total thickness of the image slices (z-spacing times n_images) should be at at least 30% greater than

the thickness you determine optically. To test your chosen parameters right-click the SNAP IMAGE. This collects a series of images and writes a script file that combines them into a single z-stacked picture. Once the images in your stack are acceptable and the corners and sub-drops are all in focus, click RUN to initiate the auto-imaging. This procedure need only be done once, provided you do not reset the magnification, change lenses or change the volume of your drops.

Viewing the images

If you have chosen to take more than one image of each sample you will likely want to run the image enhancement script (i.e. `source processMCSG2.com`). Then view the images using a photo viewing application such as Ristretto Image Viewer. To view the images on a remote machine, first install ssh on the server with the images (`apt-get install ssh`). Then start the ssh service (`sudo service ssh start`). Then open the x-window enabled connection from the remote machine (`ssh -x pi@***.***.*.*`). The *s represent the IP number of the server.

Troubleshooting:

My automated imaging run starts fine, but then the images become increasingly off center.

Most likely, you accidentally centered on a non-corner drop when doing the calibration. Re-center the corner drops, and confirm that the microscope is actually imaging the corner drops by putting the tip of a pencil or some other pointy object into the field of view at each corner. Don't forget to save the configuration file after you've gotten the correct corners set.

It is also possible that one of the set screws connecting a motor to its screw has become lose. These can be tightened with a small Allen key. A drop of epoxy or locktite can be used to ensure that the screw does not come lose again.

There's nothing to focus on in one of the a corner drops.

Use the translation tool to move to a neighboring drop that does have something to focus on and push set. Then, center the corner drop in question and adjust X and Y and click set again. Alternatively, you can use an identical plate that does have something to focus on at the position in question. It may help to make a small mark with a magic marker.

I accidentally collected images with the wrong plate ID.

First, don't panic. All image directories have their own time stamp, so no data has been overwritten. It is not difficult to move the directory containing the data to the correct place. You can do this from the command line using the UNIX **`mv`** command. First change into the images directory

```
cd images
```

Then move the directory to the correct location

```
mv <wrong_sample_ID>/<wrong_plate_name>/<dir_name>
<correct_sample_ID>/<correct_plate_name>/.
```

The dot at the end of the line above is important. It means don't change the directory name.

How do I abort an automated imaging run once it has started?

Click the Stop/Close button in the AMiGUI window <u>just once</u>. It is normal for the program to capture a few more images before the run stops. Clicking Stop/Close during a run causes the run to stop but leaves the GUI running. At other times clicking this button causes the program to end completely. Clicking the button twice during a run will cause the run to stop and then AMiGUI will close. If the Stop/Close button does not work, you can also shut down the program by clicking the X in the upper right corner of the AMiGUI window.

AMiGUI won't start up and the machine is very close to one of the translation endpoints.

The program will not start up if an end-switch is activated. Turn the translation screws by hand to a position where no end switches are depressed. Then try restarting the software.

The machine stops and the software hangs for no apparent reason.

There are at least three possible causes:

1) It is possible that a screw or one of the bars are misaligned. To check this, turn the screws by hand. If you will feel significant resistance (particularly near the ends), this is probably the problem. If so, loosen the screws that hold the bar holders and/or the bearings in question, translate the screw by hand to the end that gives the problem, and then re-tighten the screw. It may be necessary to do this more than once.

2) Electrical interference could be causing one of the switches to register a momentarily closed state, thereby triggering an Alarm in the Grbl software. This is most likely to occur during movement, and the terminal window should show an Alarm message if this is the case. The solution is to make sure the switch wires are well separated from the current carrying wires.

3) It is possible that you have a bad SD card. The SD card can be tested by removing it from the Raspberry Pi and inserting it into another computer. These cards are inexpensive, and it is relatively easy to simply purchase a new card and then download and configure the software as discussed above.

The machine does not zero correctly at the start or when the reset button is clicked.

This is most likely an issue with a limit switch. Switches can be checked by unplugging them from the Arduino board and testing continuity with the switch open and depressed. The circuit should be closed when the switch is depressed.

The machine stops with a message about moving beyond its physical limits:

If you have extended the dimensions of the machine by using longer screws and bars, you may get software message saying that a physically-allowed move is not permitted. The software limits for the x, y, and z axes can easily be changed my editing the AMiGUI code. The relevant parameters are near the start of the python program.