



CFQuant

User manual

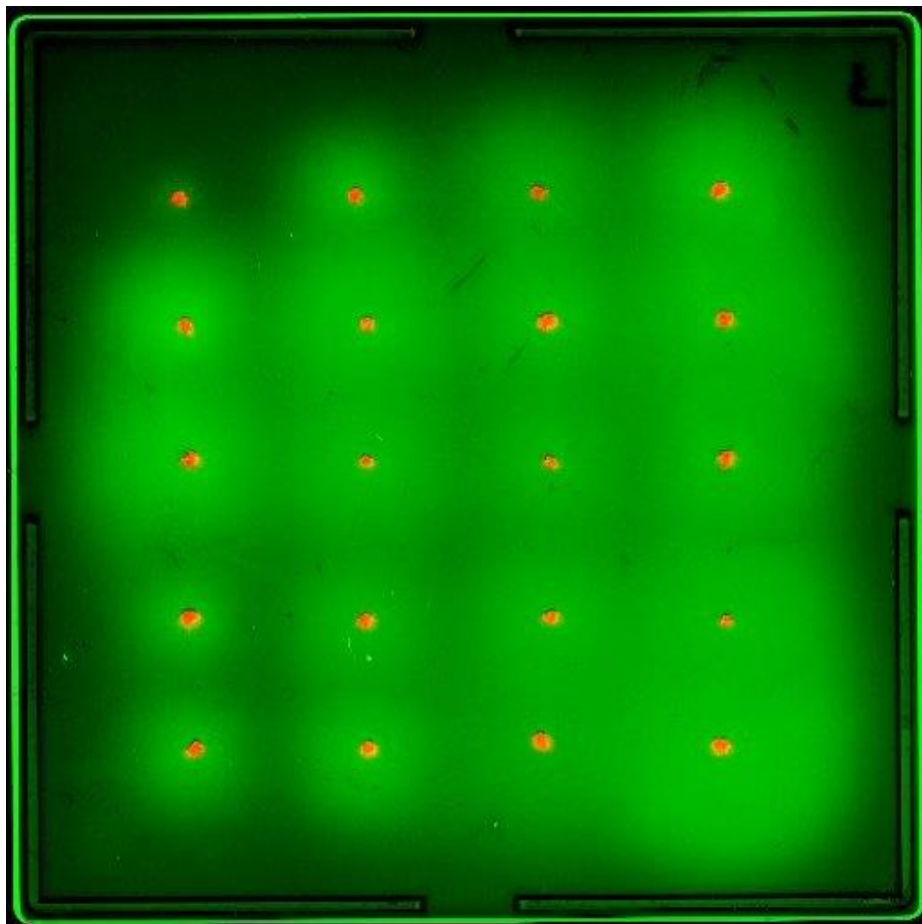


Table of Contents

| | |
|--|----|
| General description..... | 3 |
| Image requirements..... | 3 |
| Output files..... | 4 |
| Result analysis..... | 4 |
| Software pipeline | 4 |
| Initiation | 4 |
| Work modes..... | 4 |
| Colony identification | 5 |
| Halo identification..... | 5 |
| User input and interaction | 7 |
| Initiation screen..... | 7 |
| Colony inspection screen (single image mode) | 8 |
| Halo inspection screen (single image mode) | 9 |
| Error handling | 10 |

General description

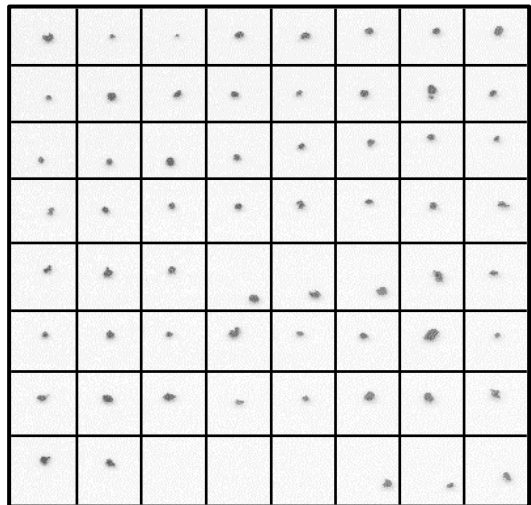
This software analyzes images of cell colonies that emit a luminescent/fluorescent signal (from here on generalized as “luminescent”), in the form of a luminescent halo around the colony. CFQuant automatically identifies both the colonies and their emitted luminescent halos and returns quantitative data on features such as area, pixel sum, etc. CFQuant can either analyze one image at a time, allowing the user to check its results and correct any possible errors, or perform batch processing, where multiple images are analyzed with no user interaction.

This software was written using MATLAB (The MathWorks, Inc., Release 2017b). It is available for download at <https://github.com/eyaldaf/CFQuant>. A folder of example images and the results of their automated analysis are also available at the link.

Image requirements

CFQuant requires two images from each plate: One image of the colonies, and a second image of the luminescent halos. Both images should be taken with the same camera, only with different filters. Do not change the focus or location of the camera or the location of the plate between the two images. This is crucial, as the location of the colonies is used when identifying the halos. Alternatively, the same RGB image can be used twice, with one color channel selected for the colonies and a different color channel selected for the halos.

CFQuant has two methods of colony detection – arrangement-based and scatter-based. Scatter-based detection can work with any colony arrangement, but for using the more accurate arrangement-based detection, the colonies should be arranged on a grid of rows and columns, similar to the plate shown on the right. The arrangement does not need to be precise, but rows and columns should be roughly recognizable. This arrangement is specified by the user upon initiation (see *User input and interaction*).



The colony identification is based on the colonies being either the darkest or the brightest objects in the image. Therefore, if the colonies are darker than the background, make sure no other objects in the image are darker than the colonies. Alternatively, if the colonies are lighter than the background, make sure no other objects in the image are lighter than the colonies. Scatter-based detection can handle some noise that is brighter/darker than the colonies, but this is still recommended. This is also recommended for the halos in the halo image, though objects outside the halo areas should not affect the analysis.

The images can be in any format supported by Matlab, including TIF, BMP, GIF, PNG and JPEG. The images can be either grayscale or RGB (CFQuant will convert RGB images to grayscale images using the channels selected by the user). The images can have any bit depth, but CFQuant reduces images to values between 0-511 (effectively nine bits per pixel, but stored at 16-Bit) for faster processing. Nonetheless, it is recommended to use 16-Bit images if possible. It is also recommended to not edit the images, as editing may alter the pixel values and lead to inaccurate results.

For batch processing, there are a few extra requirements: The images should be arranged in two separate folders, one for the colony images and another for the halo images. Additionally, the names of

the colony image and halo image from the same plate should be identical (for example, both being named "plate1.tif"). This name will also be used for the result files (see *Output files*).

Output files

For each analyzed image, CFQuant returns six output files:

The measurements of the colonies and halos are saved as two separate CSV files: Name_Colonies.csv and Name_Halos.csv ("Name" is the name the user specified when saving the data, or the name of the images if batch process was used). These files contain the measured area, sum of pixels, mean pixel value and maximum pixel value of each colony or halo, respectively, with each row corresponding to the same colony in both files. For the sum, mean and maximal pixel there are two additional normalized values: One is normalized by the selected threshold value and the second is normalized by the minimal threshold value (see *Colony identification* and *Halo identification*). The threshold values are also saved to these files.

CFQuant also creates three PNG images: Two of them, Name_Colonies.png and Name_Halos.png, show the borders of the identified colonies or halos, respectively. The third image, Name_ResultRows.png, shows the order of the colonies in the .csv result files, starting with 1 for the first row. By default, the colonies are ordered from top to bottom and from left to right, but manually creating new colonies (in the *Colony inspection screen*) can scramble that order.

Finally, CFQuant also creates a MATLAB result file – Name.mat. This file can be used to return to the halo inspection screen and make changes to the identified halos. To do so, initiate the software again, click on "edit old results" and choose this file. You can then either overwrite the old results or create a new set of result files.

Result analysis

CFQuant returns multiple measurements for both the colonies and the halos. When running the software for the first time, we recommend using colonies with known values of your trait of interest. That way, you will be able to find the features that best correlate with your trait of interest. We expect that dividing one of the features from the halo file by one of the features from the colony file will produce the best results, as this normalizes for colony size.

Software pipeline

Initiation

Upon initiation, CFQuant first receives input from the user (see *Initiation screen*). This includes the images, and in arrangement-based detection, this input also includes the approximate arrangement of the colonies. For arrangement-based detection, the colonies should be arranged in a grid of rows and columns, as described in the *Image requirements*. The user can also select between the two work modes of the software.

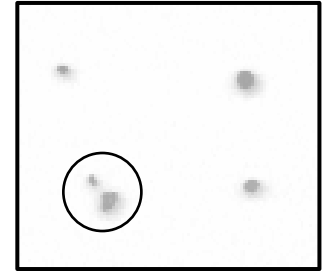
Work modes

CFQuant has two work modes: In single image mode, one image is analyzed each time, and the user can view the detected colonies and halos and correct any possible error. In batch processing mode, multiple

images are analyzed automatically without user interaction, though the results for the halos can still be manually adjusted after the analysis is done (using the "Edit old results" button, see *Initiation screen*).

Colony identification

Two identification methods exist for this software, chosen by the user upon initiation. Arrangement based-detection requires input from the user about the colony arrangement, but in return offers more precise detection, as well as the ability to identify colonies that split to two or more fragments, like the example on the right. Scatter-based detection cannot identify these splits and is also somewhat less precise, however it requires no input from the user



aside from the images themselves. Both identification methods rely on the colonies being brighter or darker than the background (see *Image requirements*) to separate the colonies from the background.

Colony identification begins with a background removal step, after which only several foreground areas are left in the image. Typically, the number of remaining foreground areas exceeds the user-specified number of colonies. From this point, arrangement based-detection determines whether the excess foreground areas are split colonies that should be re-joined or persisting background areas that should be removed. To do so, CFQuant compares the arrangement of the foreground areas with the user-specified arrangement, and then either joins close foreground areas or deletes low-value ones until only the specified number of colonies remain. In scatter-based detection, the colony number is unknown, so the software deletes background noises based on their shape, size and pixel values. Regardless of method, CFQuant then determines the background threshold value (i.e. the value below which pixels are considered part of the background).

Once the colonies are identified, the user can view the results and make changes, if necessary (see *Colony inspection screen*). These changes include removing a split area from the colony or adding one, deleting a colony or creating a new colony.

After user inspection, the software calculates the features of the colonies, which include the area, pixel sum, mean pixel value and maximal pixel value of each colony. The values for the sum, mean and maximal pixel are also normalized in two ways: one where the pixel values are normalized by reducing the threshold value from them, and another where the pixel values are normalized by reducing the lowest possible threshold that still keeps all the colonies separated from them.

Halo identification

The software must overcome several problems for automatic identification of luminescent halos. These include the gradual decrease of the luminescence, overlaps between halos, and inherent image noise. All these problems prevent simple edge detection algorithms from giving meaningful input about the location of the halos.

To address these problems, CFQuant relies on the shape of the halos, which is expected to be roughly circular, with the colony at the center. Since the colony and halo images are of the exact same location (see *image requirement*), the centers of the colonies can be used as an approximation for the centers of the luminescent halos. CFQuant determines the edges of each halo by gradually lowering the background threshold value for that halo, and selecting the lowest one (i.e. the one resulting in the largest halo area) where the area around the colony centers still resembles a circle. The decision is made based on the distance of each edge-point from the colony center being roughly equal, while still allowing some noise in the distance and center location. This sets a specific background threshold for each halo, which helps with the detection of halos with different intensities. To address the possibility of halo overlaps, CFQuant

recognizes not only complete circles but also partial sectors (at least 144° in total). This also allows the software to recognize halos with more ellipsoid shapes.

Once the halos are identified, the user can view the results and make changes, if necessary (see *Halo inspection screen*). These changes include increasing or decreasing the area of a halo (by changing the background threshold value of that halo), deleting a halo entirely or creating a halo for a colony with no identified halo.

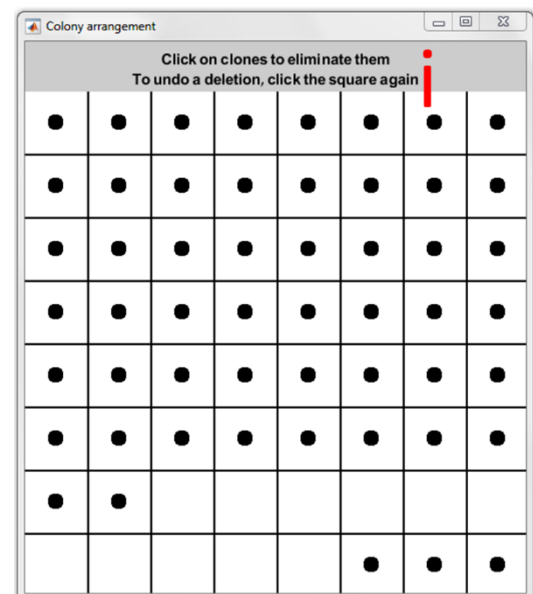
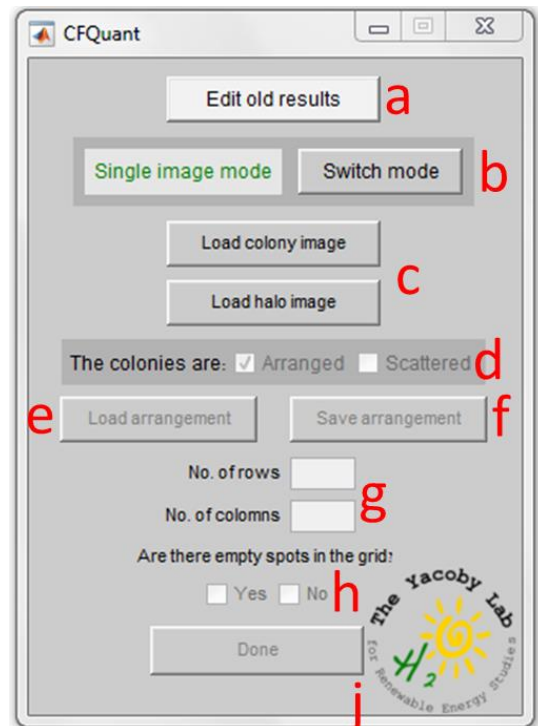
After user inspection, CFQuant calculates the features of the halos, which include the area, pixel sum, mean pixel value and maximal pixel value of each halo (same as the colonies). Since areas where two halos overlap are expected to have higher pixel values than areas with no overlap, the pixel values in these areas are adjusted. For each overlapping halo individually, the values in the overlap area are calculated based on that halo's expected area under every threshold: Each pixel of the overlapping area is assigned the value of the highest threshold under which it is expected to still be within the halo's area, rather than the actual (potentially higher) value of that pixel. This value can be different between the two halos, based on their intensity outside the overlap area. Like the colonies, the values for the sum, mean and maximal pixel of the halos are also normalized in two ways: one where the pixel values for each halo are normalized by reducing the threshold value of that halo from them, and another where the lowest threshold value amongst all halos is reduced from the pixel values of every halo.

User input and interaction

Upon initiation, CFQuant will open with the initiation screen. After input from the user, the software will begin the identification and progress to the next screens in the order they are shown below.

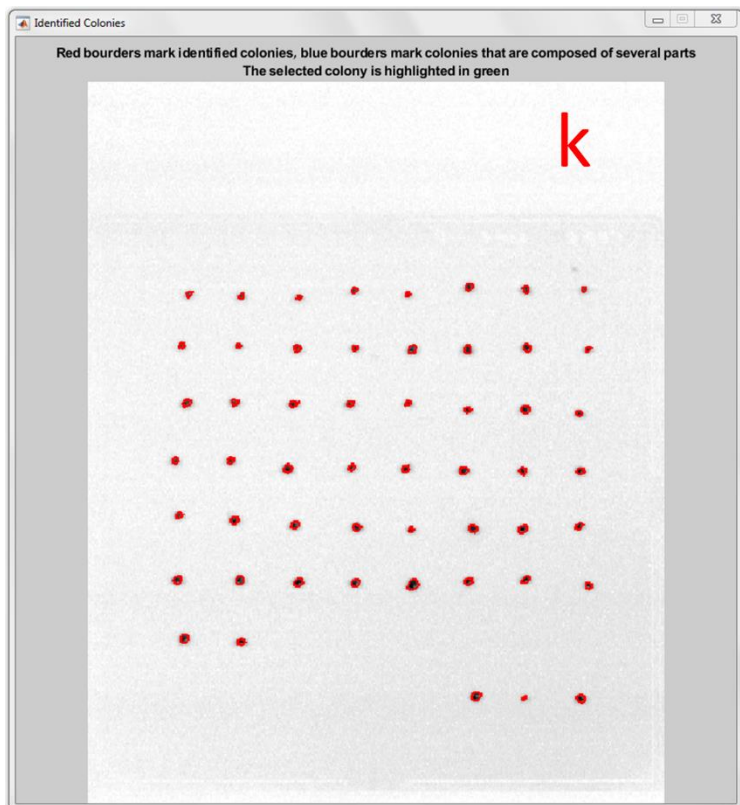
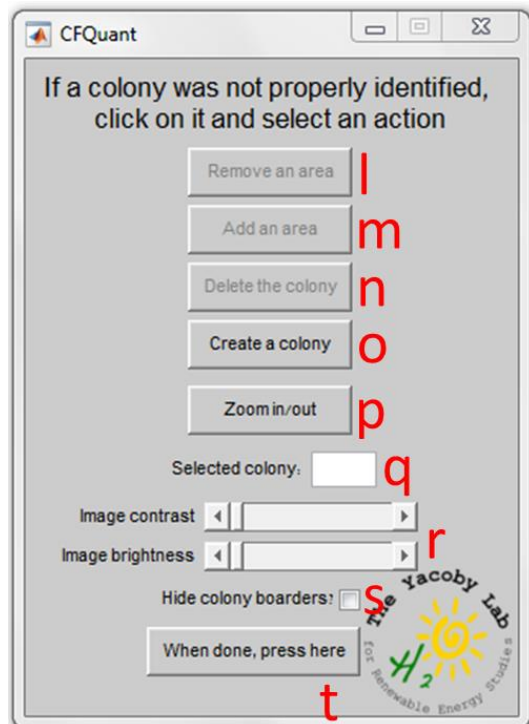
Initiation screen

- a) **Edit old results:** This button is used to upload an old MATLAB result file that was saved in a previous run of CFQuant (see *Output files*). After uploading the file, the screen changes to the Halo inspection screen.
- b) **Switch mode:** Switches between single image mode and batch processing mode (see *Work modes*). The box on the left displays the current mode.
- c) **Colony and halo image loaders:** Uploads the images. In batch processing mode, these are replaced with buttons to upload the colony and halo folders. Other buttons are available only after uploading both images.
- d) **Colony arrangement:** Switches between arrangement-based detection (requiring colony arrangement input) and scatter-based detection (detection with no arrangement input).
- e) **Load arrangement:** Uploads a colony arrangement MATLAB file that was previously saved with the "Save arrangement" button.
- f) **Save arrangement:** This button saves the specified colony arrangement as a MATLAB file (.mat). This file can be uploaded on future uses of CFQuant instead of re-specifying the colony arrangement.
- g) **Row and column specification:** Used to input the number of rows and columns in the grid-like arrangement of the colonies (see *Image requirements*).
- h) **Empty spot specification:** Selecting yes will open the colony arrangement figure (h). Selecting no will allow the user to continue without specifying empty locations on the grid.
- i) **Colony arrangement figure:** If the user marked the "Yes" box in g, this figure allows them to specify the empty spots. Clicking on a spot in the figure will mark it as empty; clicking on it again will return it to normal.
- j) **"Done" button:** This button becomes available once the user specifies the empty spots in the grid, marks the "No" box in g or selects scatter-based colony detection. In single image mode, pressing it proceeds to the colony recognition phase. In batch mode, pressing this button prompts the user to select a folder to which the data will be saved, and then initiates the batch process with no more user interaction.



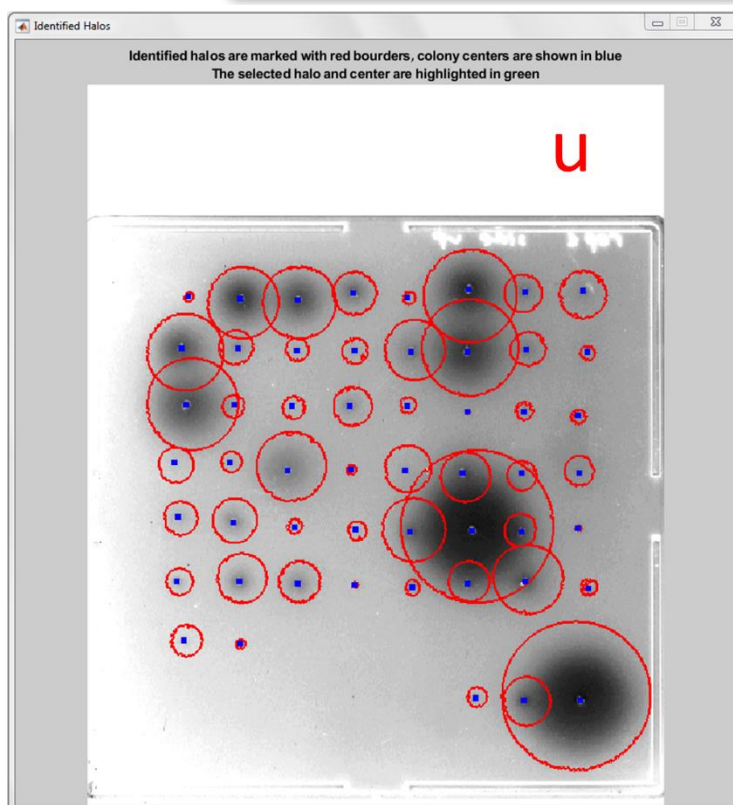
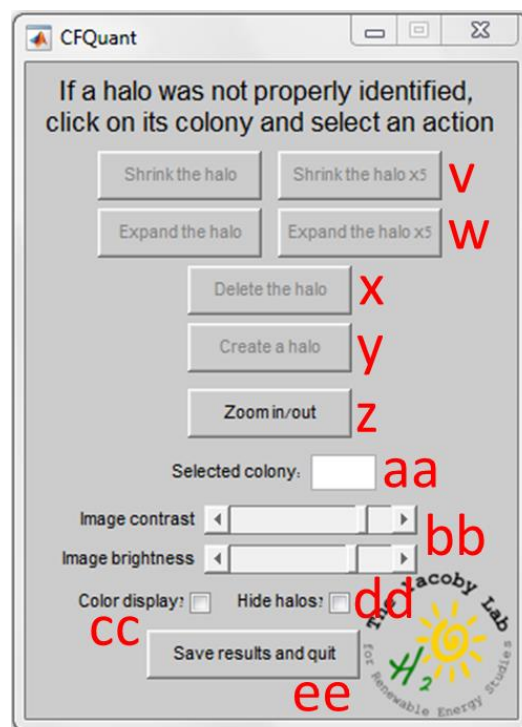
Colony inspection screen (single image mode)

- k) **Colony results figure:** This figure displays the colony image. The user can click on the image to select a colony they wish to edit.
- l) **Remove an area:** Only available if the user selected a colony that is split to several parts (denoted by blue borders when not selected). After pressing this button, the user can click on the colony figure to select the part they want to delete, or press cancel to switch action.
- m) **Add an area:** Available once the user selected a colony. After pressing this button, the user can click on an area in the colony figure to add it to the colony, or press cancel to switch action. Additionally, areas that already belong to another colony cannot be added. These areas must be removed from that colony or deleted first.
- n) **Delete the colony:** Completely deletes the selected colony.
- o) **Create a colony:** After pressing this button, the user can click on an unidentified area in the image to consider it as a new colony, or press cancel to switch action.
- p) **Zoom in/out:** While this button is selected, you can zoom in on the image by left clicking on it with the mouse, and zoom out by right clicking on it.
- q) **Selected colony:** Displays the selected colony number. Entering a different number will select the corresponding colony.
- r) **Brightness and contrast bars:** The sliders adjust the brightness and contrast of the image display (without changing the actual image).
- s) **Hide colony borders:** Ticking this box will hide the borders, allowing clearer view of the image. This option is automatically ticked off after the user makes any changes.
- t) **"Done" button:** Saves any changes the user made and proceeds to the halo identification.



Halo inspection screen (single image mode)

- u) Halo results figure:** This figure displays the halo image. The user can click on the image to select a colony whose halo they wish to create or edit. Once the colony is selected, the other buttons will become available.
- v) Shrink the halo:** Allows the user to gradually shrink the halo of the selected colony. Shrinking works by raising the pixel threshold for the halo by one (see *Halo identification*). Shrink x5 works the same as clicking on "Shrink the halo" five times
- w) Expand the halo:** Allows the user to gradually expand the halo of the selected colony. Expanding works by lowering the pixel threshold for the halo by one (see *Halo identification*). Expand x5 works the same as clicking on "Expand the halo" five times
- x) Delete the halo:** Completely deletes the halo, setting all of its values to zero. Data on the colony itself will still be saved.
- y) Create a halo:** Creates the smallest possible halo around the selected colony (If the colony had no identified halo).
- z) Zoom in/out:** While this button is selected, you can zoom in on the image by left clicking on it with the mouse, and zoom out by right clicking on it.
- aa) Selected colony:** Displays the selected colony number. Entering a different number will select the corresponding colony.
- bb) Brightness and contrast bars:** The sliders adjust the brightness and contrast of the image display (without changing the actual image).
- cc) Color display:** Switches between black and white display and color (heatmap) display for the images. The color display uses Matlab's parula colormap.
- dd) Hide halos:** Ticking this box will hide the halo borders, allowing clearer view of the image. This option is automatically ticked off after the user makes any changes.
- ee) "Done" button:** Saves any changes the user made and proceeds to save the data.



Error handling

| During initiation | |
|--|---|
| Error message | Cause and possible solutions |
| This is not a result file. Please select the file created by the software when you saved the results | The user tried to upload an old result file but the file is missing some or all of the expected variables. Most likely caused by accidentally uploading the wrong file. |
| Unsupported file format | The selected image is in an unsupported format (see <i>Image requirements</i>). Convert it to a .tif file or a similar format. |
| Colony and halo images should have the same size | As stated in the warning, the colony and halo images must have the exact same size (and camera location, see <i>Image requirements</i>). |
| There are no files in the selected folder | In batch process, the user selected an empty folder. Make sure you select the correct folder (click on it once and press "Select Folder"). |
| None of the files in the selected folder belongs to a supported image file format | In batch process, no image file was found in the selected folder. Make sure the correct folder was selected and make sure the image files are in a supported format (see <i>Image requirements</i>). |
| No matching images found between the folders. Please note that batch processing requires each colony image and its matching halo image to have the same name | As stated in the error message, the colony images in the colony folder and the halo images in the halo folder must have the exact same names for the batch process (including file types). Make sure the names are indeed the same, and make sure the correct folders were selected. |
| No images with matching names and size were found. Please note that the colony and halo images should be of the exact same location, and should have the same height and width | In batch process, files with the same name were found in the two folders, but their sizes do not match. As stated in the warning, the colony and halo images must have the exact same size (as well as focus and camera location, see <i>Image requirements</i>). |
| Incorrect file | The user tried to load a colony arrangement, but the selected file does not contain a previously saved arrangement. |
| A minimum of two colonies are needed for this software | The current version of CFQuant cannot identify a single colony based on arrangement. Therefore, this message appears whenever a grid with less than two colonies is specified. |
| During colony identification | |
| Error message | Cause and possible solutions |
| The image appears to have less than # colonies | No threshold value for the colony image can produce as many colonies as specified in the arrangement. This can happen if the user specified an incorrect number of rows and columns, or if some or all of the colonies are not amongst the darkest or brightest elements in the image (see <i>Image requirements</i>). |
| The selected area is already a part of this colony | Occurs if the user attempts to add an area to a colony and then selects an area that already belongs to that colony. |
| The selected point is part of the background | Can occur during area addition or colony creation, if the selected pixel has a value that is lower than the lowest value that separates the existing colonies (see <i>Colony identification</i>). Make sure you did not accidentally click on a background pixel next to the area you wanted to pick. |

| | |
|--|--|
| The selected area is a part of another colony. To merge them, first delete one of the colonies | Occurs if the user attempts to add an identified area to another colony. As stated, CFQuant can only add areas to a colony if they are currently unidentified. If they belong to the wrong colony, they must first be removed or deleted (using "Remove an area" or "Delete the colony"). |
| The selected area is a part of an identified colony | Occurs if the user attempts to create a new colony, but clicks on an area that already belongs to an existing colony. |
| The selected area will become part of a different colony if the threshold will be lowered | Can occur during area addition, if the selected pixel has a value below the current threshold (see <i>Colony identification</i>), and when lowering the threshold (thus expanding the area of all colonies) it becomes part of a different colony than the specified one. Make sure you clicked on the correct area and selected the correct colony. |
| The selected area is not an independent colony, and will become part of an existing one if the threshold will be lowered | Can occur during colony creation, if the selected pixel has a value below the current threshold (see <i>Colony identification</i>), and when lowering the threshold (thus expanding the area of all colonies) it becomes part of an existing colony. Make sure you clicked on the correct area. |
| While saving results | |
| Error message | Cause and possible solutions |
| The specified file name belongs to an existing file that is currently open or otherwise uneditable. Please close the file and try again or choose a different name | When attempting to save the results of a single image analysis, the user selected a file name and folder that correspond to an existing file that cannot be edited. A common cause is that said file is already open in another software, in which case closing the file will be enough (note that the current contents of that file will be overwritten). Alternatively, you can always choose a different file name or folder. |