Instructions for *CM Engine* use

1) Download *CM Engine* from SourceForge (http://cm-engine.sourceforge.net/) or from the Rothstein Lab website (http://www.rothsteinlab.com/cm-engine.zip).

2) Download *ImageJ* (http://rsbweb.nih.gov/ij/).

3) Start *ImageJ*, install *CM Engine* as a macro (instructions here: http://rsbweb.nih.gov/ij/docs/menus/plugins.html#macros).

4) To start *CM Engine,* select it from *ImageJ*'s menu bar (Plugins -> Macros -> ScreenMill – CM Engine [c]).

5) Upon starting *CM Engine,* a dialog box will show up asking you to select a directory. This directory should be the parent directory of your images. The parent directory should only contain multi-plate scans or sub directories called "rough_crops" or "fine_crops". See **Supplementary Figure 3** and the Supplement labeled "***CM Engine* - Image orientation and naming conventions**" for further information.

6) After selecting a parent directory, you will be presented with an input dialog (**Figure 1**, note appearance may differ slightly based on operating system).

    Input dialog explanation:

    Colony Measurement mode: Default value = "Standard". For further information see supplement labeled **"*CM Engine - Colony* Measurement Modes"**.

    Plate Density: 384 or 1536 formats are currently supported.

    File name to save measurements to (log file): default = "colonyAreas".

    Running Mode: Default = "Standard". To see your images as they are being processed, along with popup messages that display relevant information as images are being processed, select "Debug". Note that debug mode is MUCH slower and memory intensive then Standard mode.

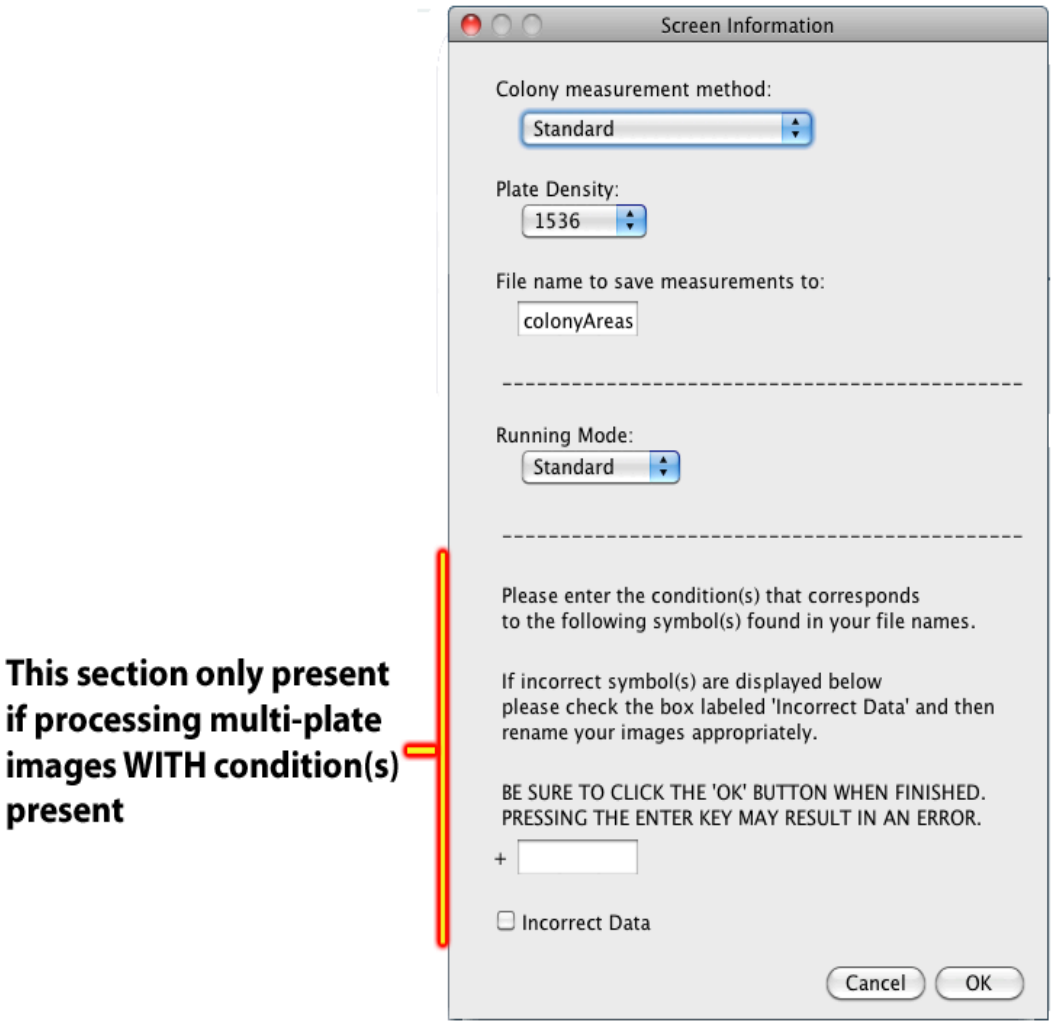
    Condition Input: This input is only present when multi-plate images containing conditions are detected by *CM Engine.* Enter identifiers for any conditions present in your multi-plate images. See the Supplement below labeled "Image orientation and naming conventions" for further information on conditions.

    Incorrect Data: This input is only present when multi-plate images containing conditions are detected by *CM Engine.* Check this box if there is a mistake in the conditions displayed in the previous input.

7) After entering information in the input dialog box, click "Ok" and image processing begins. Please note that image processing is memory intensive. If

ImageJ crashes due to an out of memory error, close *ImageJ*, restart it, and reload and rerun *CM Engine*. In the event of a crash, no data will be lost. When *CM Engine* is rerun, it will restart where it stopped at the time of the crash.

Figure 1:



CM Engine **- Image orientation and naming conventions**

Image file names convey all the necessary information about plates for each *ScreenMill* component. The convention in *ScreenMill* is that the filenames of plate images convey information about experimental treatments as well as information about the strains contained on a plate. Therefore, images that are input into *CM Engine* must conform to orientation and naming conventions to ensure that comparisons between plate data are performed appropriately. Single plate images ("rough crops" and "fine crops") must be oriented horizontally with position A1 in the upper left-hand corner (see **Figure 2**). Multi-plate image orientation is described in the next section.

For all images, file names are comprised of four parameters: a query identifier, a plate number(s), a condition(s) (if present) and a file extension. One of the queries is designated as the comparer (e.g., control) and is defined as the query to which all other queries will be compared (the user is prompted for this designation by the *DR Engine)*.

File name parameter definitions:

- Query = common identifier for all plates within one set. For example, if the plates in a set were control plates, then Query could be named "Control".
- Plate Number = typically an integer.
- Condition = descriptor for any additional treatment of the plate (e.g., drug or inducer).
- File Extension = file extension of the image. A lossless format, such as tif, is recommended, however, other image formats may be used (e.g., jpg or gif).

For example, a screen in which the *TOP1* gene and a mutant allele (*top1-TA*) were over-expressed in the yeast deletion library, some of the rough crop tif files could be named as follows:

- Control,1.tif
- Control,1,Cu.tif
- Control,2.tif
- Control,2,Cu.tif
- Top1,1.tif
- Top1,1,Cu.tif
- Top1,2.tif
- Top1,2,Cu.tif
- top1-TA,1.tif
- top1-TA,1,Cu.tif
- top1-TA,2.tif
- top1-TA,2,Cu.tif

In this example "Control," "Top1" and "top1-TA" are query identifiers, "1" and "2" are plate numbers and "Cu" is a condition. In this case, Cu indicates the presence of copper, which was used to induce the over-expression of either *TOP1* or *top1-TA*.

If "Control" were designated as the comparer in the *DR Engine*, then the following comparisons would be made:

- Control,1.tif to Top1,1.tif
- Control,1.tif to top1-TA,1.tif

- Control,1,Cu.tif to Top1,1,Cu.tif
- Control,2.tif to Top1,2.tif
- Control,2,Cu.tif to Top1,2,Cu.tif
- Control,1,Cu.tif to top1-TA,1,Cu.tif
- Control,2.tif to top1-TA,2.tif
- Control,2,Cu.tif to top1-TA,2,Cu.tif

In this example, to compare *TOP1* to *top1-TA*, *TOP1* would be designated the comparer in the *DR Engine*.

Other Single Plate Image Name Examples:

- "Cln3,4,Cu.tif" This is an example image name typical for a SDL screen. It identifies "Cln3" as the query in plate 4, and that copper is present in the plate (the Rothstein lab uses copper to induce gene expression when conducting SDL screens).
- "Control,plate1,.tif" In this example a generic label of "Control" has been entered as the query. This plate contains data from plate 1 and no condition is present.
- "glucose,1,.tif" In this example "glucose" is the query in plate 1 and no condition is present.

Multi-plate Images:

For proper multi-plate image processing by the *CM Engine*, plates must be in a grid with a small space separating them. All space between plates in multi-plate images must be black. Plates must be positioned vertically with coordinate A1 in the lower left hand corner. No orientation marks are required. Plate images are processed sequentially from left to right by row. (**Figure 2**).
Images of rectangular agar plates should be named using the following convention:

- One plate in image = "Query,PlateNumber1*conditionSymbol*.fileExtension".
- Two plates in image = "Query,PlateNumber1*conditionSymbol*,PlateNumber2*conditionSymbol*. fileExtension".
- Three plates in image = "Query,PlateNumber1*conditionSymbol*,PlateNumber2*conditionSymbol*,PlateNumber3*conditionSymbol*. fileExtension".
- Etc.

Note that there are <u>no</u> spaces between the commas in the file names.
File name parameter definitions:

- Query = common identifier for all plates within one set. For example, if the plates in a set were control plates, then Query could be named "Control". Only <u>one</u> query can be present per image.
- PlateNumber = typically an integer.
- *conditionSymbol* = Optional. Any combination of +, ~, ^, #, @ (excluding commas, see below for examples). When *CM Engine* is run, these symbols are detected and the user is prompted to provide a descriptive label for each

condition.

- fileExtension = file extension of the image. A lossless format, such as tif, is recommended, however, other image formats may be used (e.g., jpg or gif).

Example multi-plate image names:

TOP1TA,3,3+,4,4+.tif

The query in this image is "TOP1TA". The plate numbers are 3 and 4. The *conditionSymbol* present is "+", which, in this example, indicates the presence of copper, while no *conditionSymbol* after a plate number indicates the absence of copper.

In **Figure 2**, plate 3 would be in the top left corner, 3+ in the top right, 4 in the bottom left, and 4+ in the bottom right.

Control,1,1+,1~,1@.tif

The query in this image is "Control". The plate number is 1. The *conditionSymbols* present are "+", "~" and "@", which, in this example, indicate the presence of copper (+), nocodazole (~) and copper AND nocodazole (@).

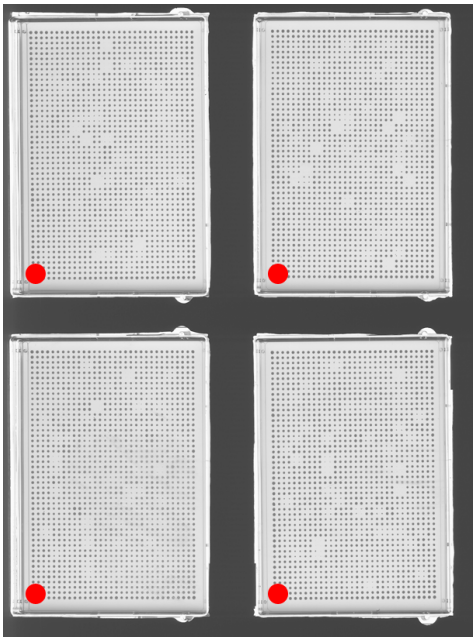
To expand the number of conditions, symbols may be used in combination: e.g., ++, ~+, @^ etc. The *CM Engine* will prompt the user to label these *conditionSymbols* properly.

Figure 2 - *CM Engine* – Example images

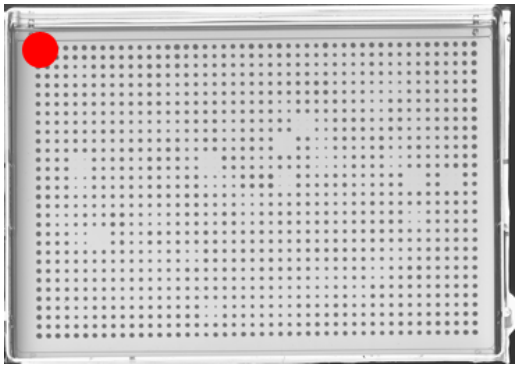
Note: images are not shown at 100% scale. Full size images may be downloaded from the Rothstein Lab website: http://www.rothsteinlab.com/tools/screen_mill/cm_engine

Red dots in the Figure indicate position A1. Red dots are for illustrative purpose only and should not appear on images processed by *CM Engine*. Images are assumed to be in this orientation by the *CM Engine*.

Multi-plate:

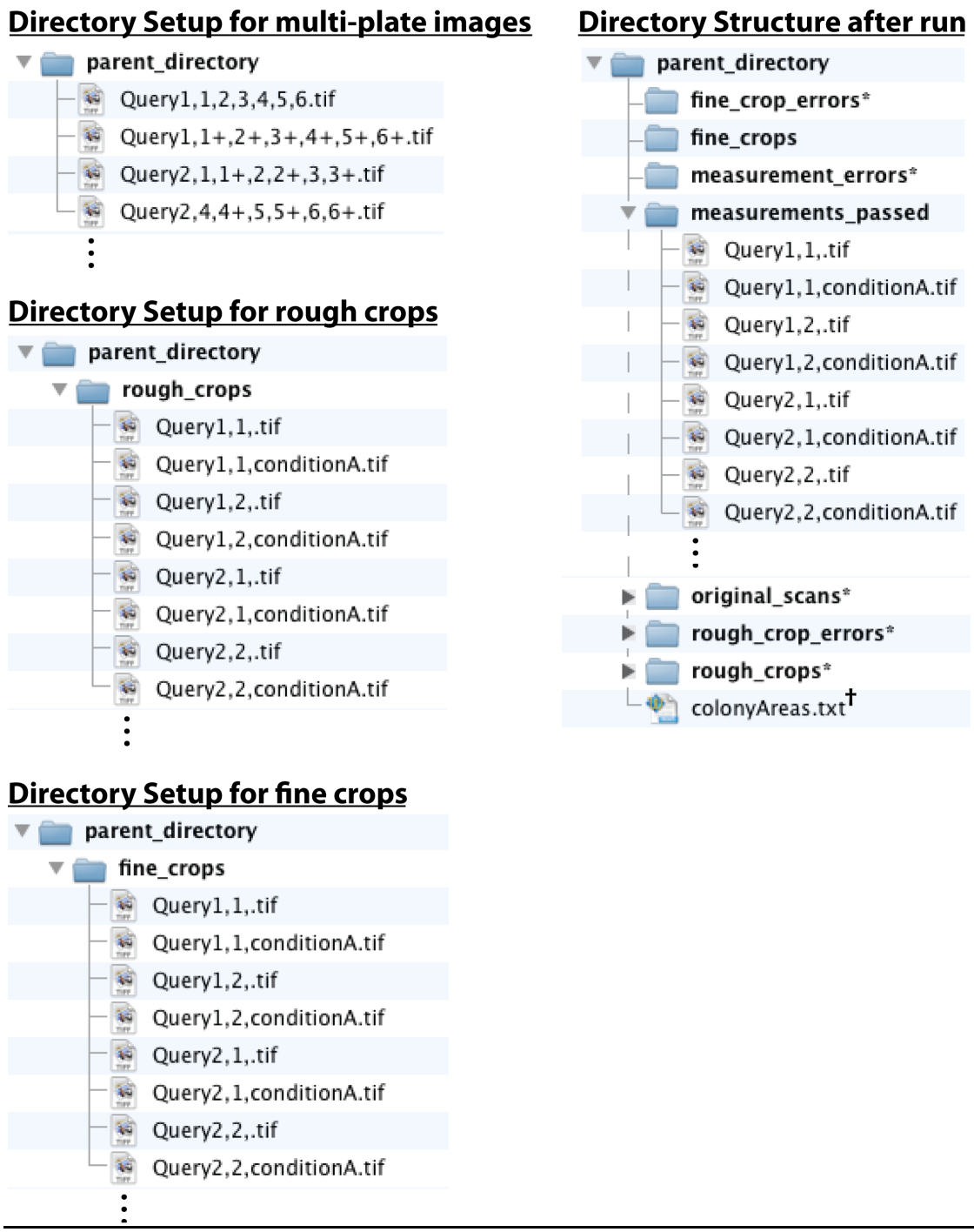


Rough Crop:



Fine Crop:

Figure 3 - Elaboration of *CM Engine* Directory Structure



---

† log file

*** depending on conditions, this file/directory may not be present**

CM Engine **- Colony Measurement Modes**

There are three colony measurement modes included within the *CM Engine*: Standard, Summation, and Background Subtracted. These modes have been optimized to work with colonies in grids of 8x12, 16x24, and 32x48 (rows x columns), although they may easily be adapted to other formats.

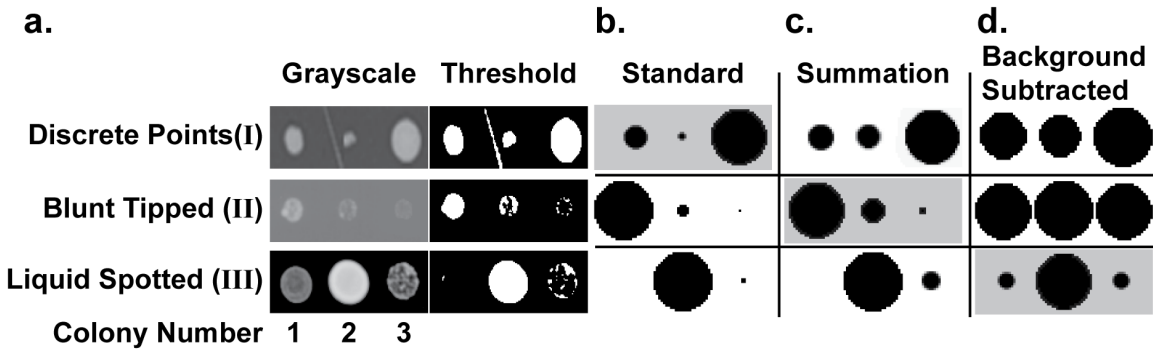
Standard mode is ideal for plates in which colonies grow in a contiguous circular pattern (e.g., discrete point pinning). A binary threshold filter is applied to each image to render plate images as black colonies (particles) with a white background. Every particle on a plate is then analyzed using the *ImageJ* function "Analyzes Particles", returning the centroid coordinates, particle size (number of pixels), and circularity (a value between 0 and 1.0, where 0 is the circularity of a straight line and 1.0 is the circularity of a perfect circle). This circularity parameter is used to help identify artifacts that may be present in the image. Any particle whose circularity is below a threshold value (e.g., 0.7) and whose area measurement is more then one standard deviation from the mean of particle area sizes on the plate is considered to be an anomaly. Examples of such artifacts include scratches on the plate or two colonies that have grown together due to excessive moisture on the plate. Any anomalies detected in an image are presented to the user as a list with a graphic of the image with the threshold filter applied. This presentation allows the user to manually remove artifacts using *ImageJ'*s built in editing tools, accepting any changes as user-verified data. Once these corrections are made, *CM Engine* assigns colony size values. The colony size assignment algorithm takes advantage of the fact that colonies are positioned on the plate in a grid format, meaning that the approximate position where each colony should lie is known. This information is correlated with the centroid coordinates that the particle analysis function of *ImageJ* returns. For each cell of the grid, every particle whose centroid lies within the cell is analyzed, but only the one with the largest area is assigned as the measurement value of that cell. The information of every other particle within that cell is discarded. After all particles are processed the algorithm determines if the image has successfully been quantified by comparing the number of particles assigned to the total number of particles in the image. If these two values do not deviate more than 25% from one another, then the plate has been successfully quantified and is moved into the "measurement_passed" folder. Otherwise a quantification error has occurred and the image is moved to the "measurement_errors" directory.

Depending on the way cells are deposited on plates, colony growth may not be represented by the largest particle within a cell of the grid layout, but instead by the growth of several particles within a cell. This type of growth may occur when colonies are pinned onto agar plates using flat pins (typical of hand-pinned experiments). In this situation, Standard mode is not optimal since it only considers the largest particle within a cell to represent colony growth. To overcome this deficiency, we developed another way to quantify colony sizes called Summation Mode. This mode is similar to Standard mode except that it assigns colony growth as the summation of all particles that lie within each cell of the grid (**Figure 4c**). Due to its nature, it does not attempt to determine if there are any artifacts on the plate.

The last measurement mode, Background Subtracted, is best for situations in

which screen images do not threshold well or in which the outline of colonies in plates are approximately equivalent, but the density differs. The latter situation is typically seen in screens which have colonies spotted onto agar plates from a liquid culture. In Background Subtracted Mode, images do not have a threshold filter applied but are instead converted to 8-bit grayscale images. The background gray value of each cell of the grid is then determined by taking the median gray value of pixels in the four corners of a cell. This background gray value is then subtracted from the cell, rendering the background with a gray value of 0 and the pixels corresponding to colony growth in the cell with a value above 0. The mean gray value of the cell is then reported as the quantification of colony growth, with larger / more dense colonies having larger values (**Figure 4d**). The limitation of this method it that everything within the cell above the background value will contribute to the growth quantification of that cell, even if this includes stray marks or other artifacts. However, Background Subtracted mode has the benefit of not having to apply a threshold filter to image prior to quantification. In addition, since the background value is calculated on a per cell basis, uneven lighting across an image does not affect quantification.

Figure 4 – *CM Engine* measurement modes:



(**a**) Images of colonies in grayscale before and after a threshold filter was applied. These images represent the three main ways cells are deposited onto agar plates in high-throughput growth experiments (discrete points, blunt tipped and liquid spotted). A different *CM Engine* measurement mode (Standard, Summation and Background Subtracted) has been developed to handle each of these colony deposition methods. The grayed boxes in **b**, **c** and **d** indicate the preferred measurement mode. Prior to analysis, each image is symmetrically divided into partitions, one for each colony. (**b**) Standard mode is designed for robot pinned colonies that come from a discrete point. Images are subjected to a threshold filter and only the largest particle of each partition is considered. Anomalies are presented to the user and may be removed manually. All colonies in image I are measured accurately since the linear artifact in partition 2 was detected and removed. In image II, only colony 1 is correctly measured since Standard mode chooses the largest particle in each partition, incorrectly measuring the non-contiguous growth in partitions 2 and 3. None of the colonies in image III were

measured accurately since the automatic threshold filter used in Standard mode is insensitive to density differences between partitions. (**c**) Summation mode is designed for colonies from blunt tipped pins. Images are subjected to a threshold filter and all particles within each partition are considered part of the colony. In this case, all colonies in image II are measured accurately. Colony 2 of image I was incorrectly measured due to the inclusion of the linear artifact in that partition. This mode also fails for image III since, like Standard mode, it is insensitive to density differences. (**d**) Background Subtracted mode is designed for cells spotted from a liquid culture. The background of the grayscale image is subtracted from each partition and the remaining pixel values are averaged to quantify cell density. In this mode, image III colonies are measured accurately, while image I was not due to the linear artifact in partition 2. Additionally, image II was not measured accurately due to the high background value of the grayscale image. For **b**, **c** and **d**, values are normalized to the largest colony size. Accuracy was determined by manual threshold and measurement as described in the documentation of *ImageJ* software.