PlateReader manual

General

PlateReader is a program for counting and determining the size of colonies on agar plates. It is designed to be easy to use without any previous knowledge and requires little manual input.

Humans are very skilled at detecting shapes against a background, but this is a much more difficult task for computers. It is therefore important that the input image is of the best possible quality. Avoid resizing or changing formats, and ensure good lighting when taking the photo. Try to avoid reflections from the lamp if possible. Use a dark and evenly colored background under the plate. The default parameters in PlateReader are tweaked for a resolution of about 5000x3000 pixels. See the Parameter section if the resolution of your images is very different.

Lastly, a note on semantics. Throughout this manual we will refer to a part of the image which contains one or more partially overlapping colonies as a "blob". A "colony" refers to the fitted circular shape which is used for the count and size determination.

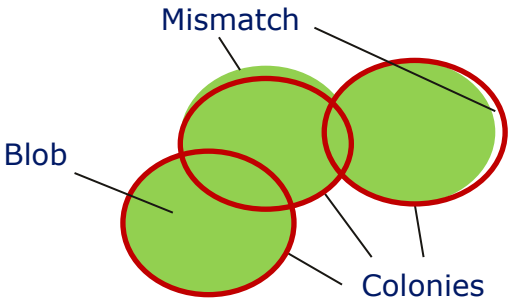


Figure showing a blob (green) with three fitted colonies (red circles). Note that there might be a non-perfect agreement between the fitted colonies and the blob. The main objective when fitting colonies is to make this mismatch as small as possible, while obeying rules about size, shape, and so on.

PlateReader is property of Novo Nordisk and should not be distributed outside of the company.

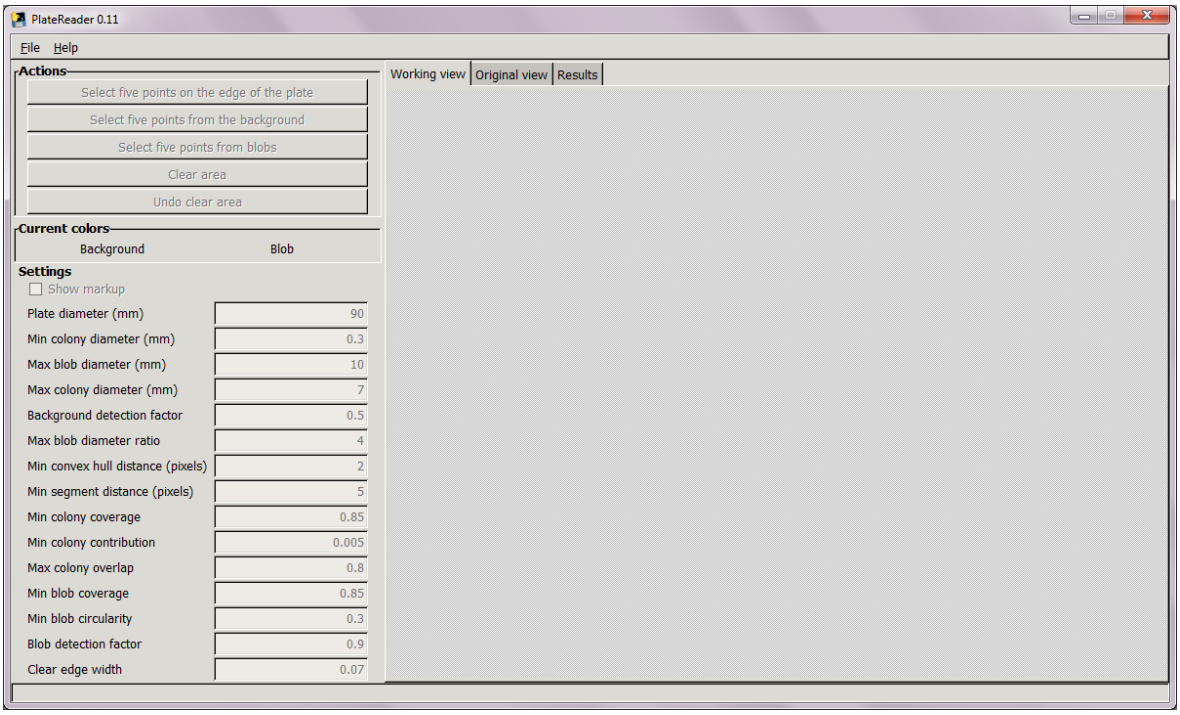
For source code, questions or comments please contact Rasmus Ågren (RAAG).

Installation

PlateReader is written in Python, but it is distributed in a compiled form which doesn't require that Python or any other software is installed on the target machine. Simply run setup.exe and then start PlateReader from the Start menu. For non-windows operating systems contact RAAG for instructions.

Interface

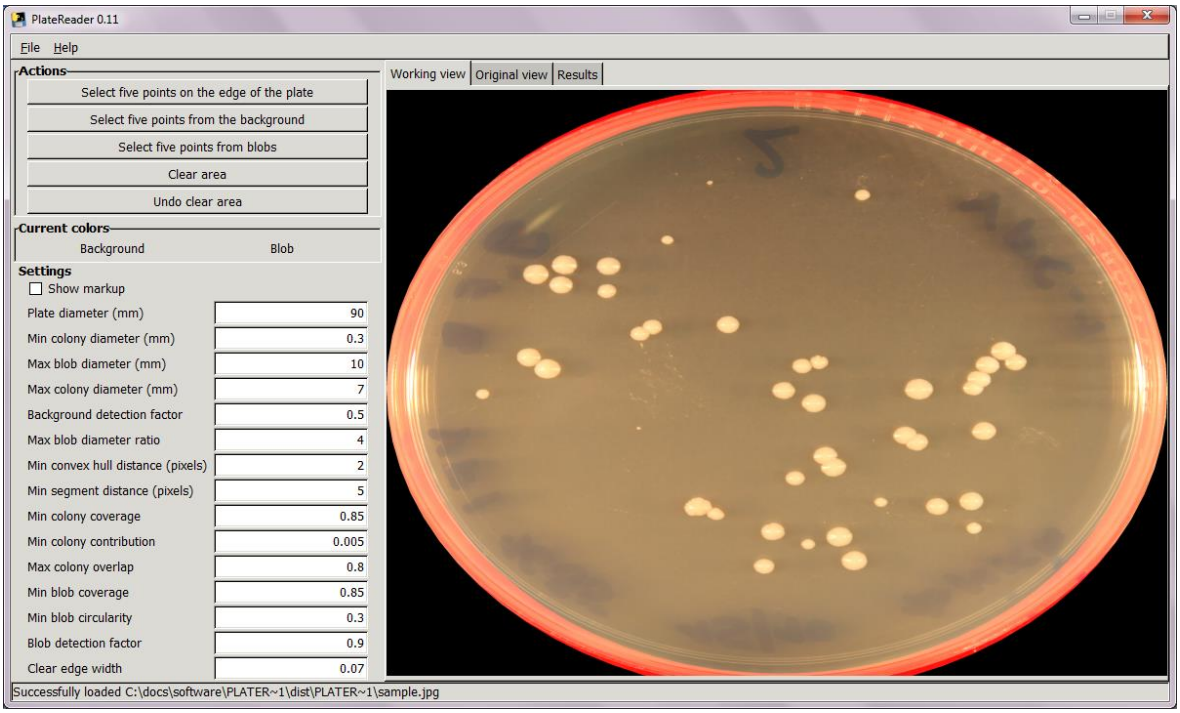
When you first open PlateReader you will see something like the figure below. This is the main interface.



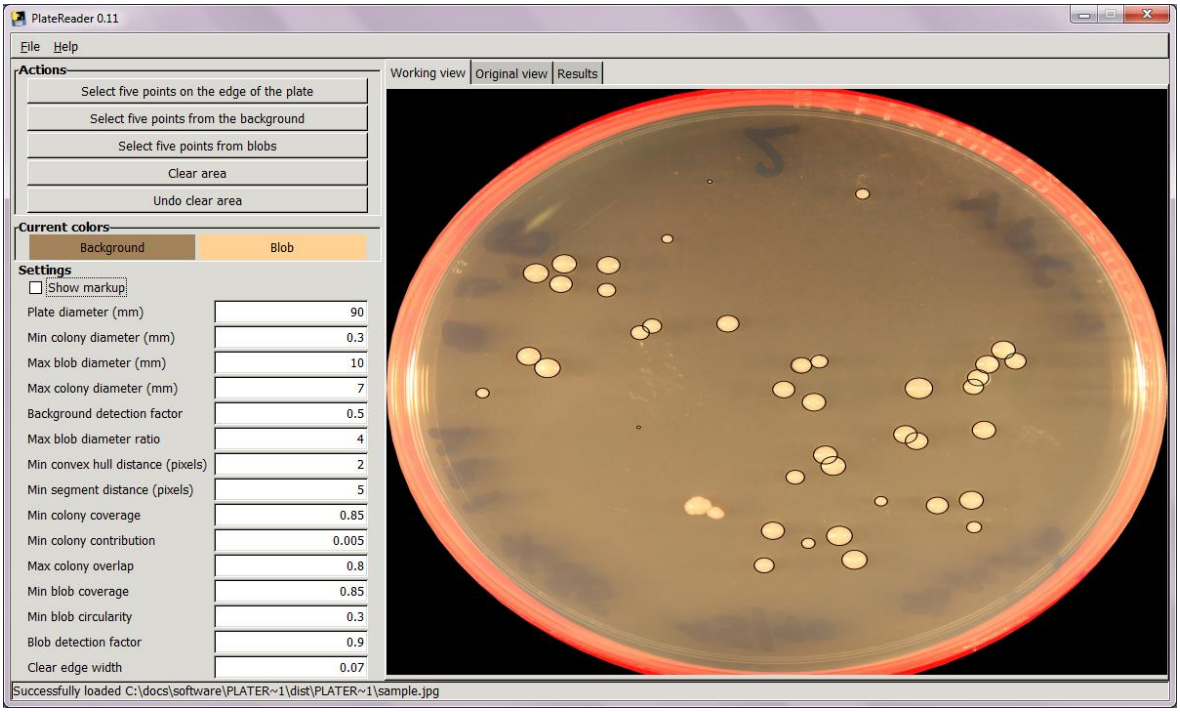
To the left is the Actions and Settings panels, in the center is the image area and to the top are the three view selectors Working view, Original view and Results. Start by loading a sample image by selecting the Help menu and Load sample image. To open any other image, use the File menu and then Load image.

This results in that the sample image is loaded and displayed. The first step is to define the perimeter of the plate in the image. This is mainly to be able to report colony sizes in

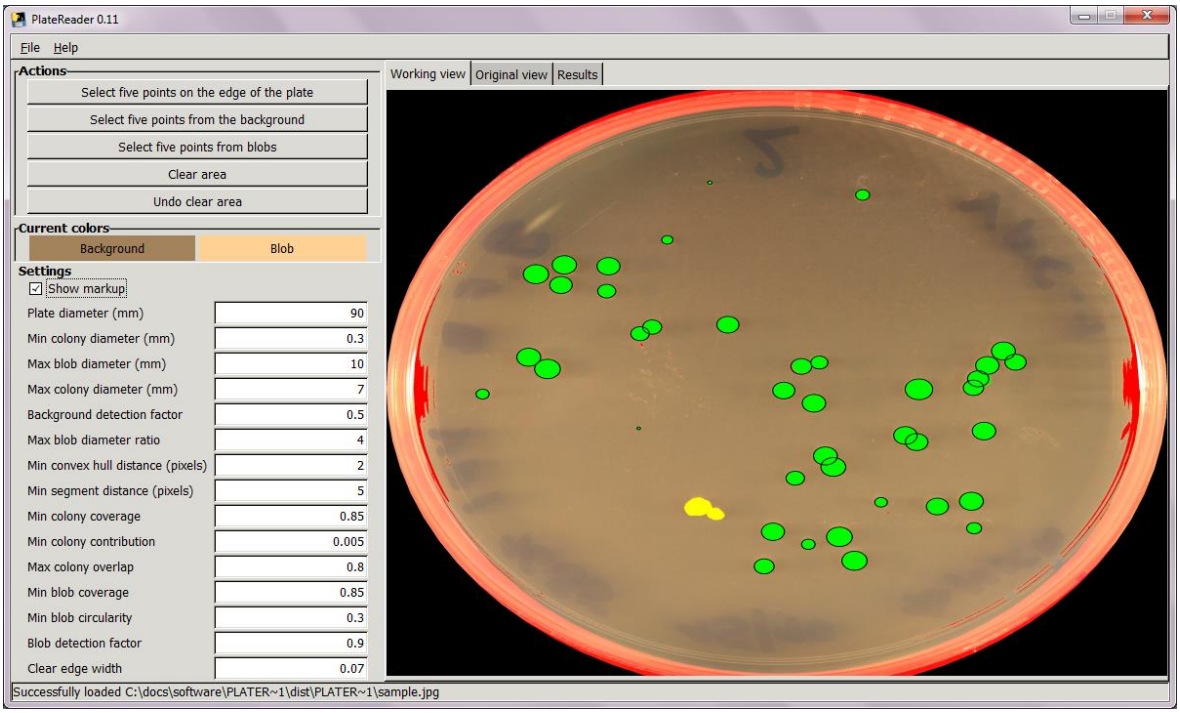
millimeters regardless of how zoomed in the photo is or how high the resolution is. Select the action “Select five points on the edge of the plate”. Then click on five points somewhat well spread around the edge of the plate.



As can be seen, this crops the image and removes the parts outside of the plate. Then select the action “Select five points from the background” and click on five areas from the background. It is a good idea to choose parts which are as close to the blob color as possible. For example, if there are bright and dark areas (due to light coming in from the side when the photo was taken) it is probably good to select from the lighter part of the background. Then do the same for blobs by choosing “Select five points from blobs”.



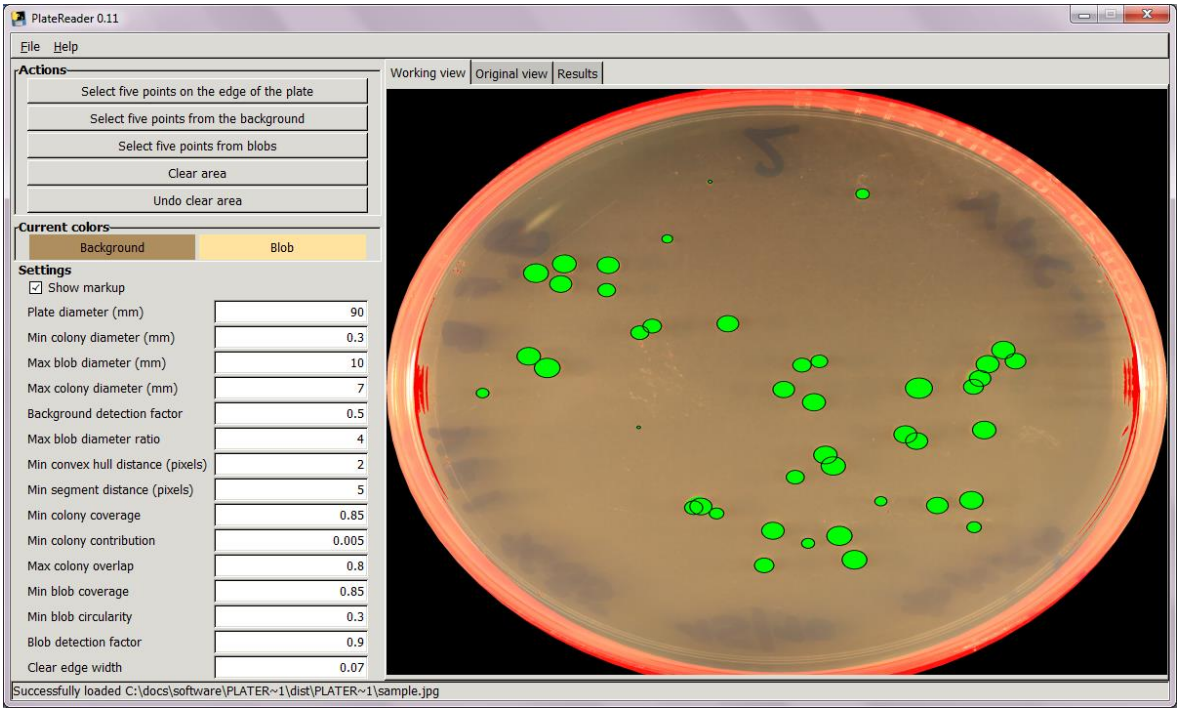
The “Current colors” panel should now show the colors you have chosen, and black circles should (hopefully) have been fitted to the colonies in the image. Click “Show markup” in the Settings panel to see how the software classified different objects in the image.



Red corresponds to regions which were deemed unlikely to contain blobs, despite being similar in color to the selected blob color. Yellow corresponds to regions which are likely to represent blobs, but where the software couldn’t identify the colonies with enough precision. Green corresponds to regions which are likely to represent blobs, and where the software could identify the colonies. What we would like to see here is that all blobs are colored green, that circles have been fitted to the colonies correctly, and that there are no yellow objects. We would also like to be sure of that there are no “real” blobs hidden beneath the red areas, since that would exclude them from further analysis. You can switch to Original view to see that this is not the case here. Remember to switch back afterwards.

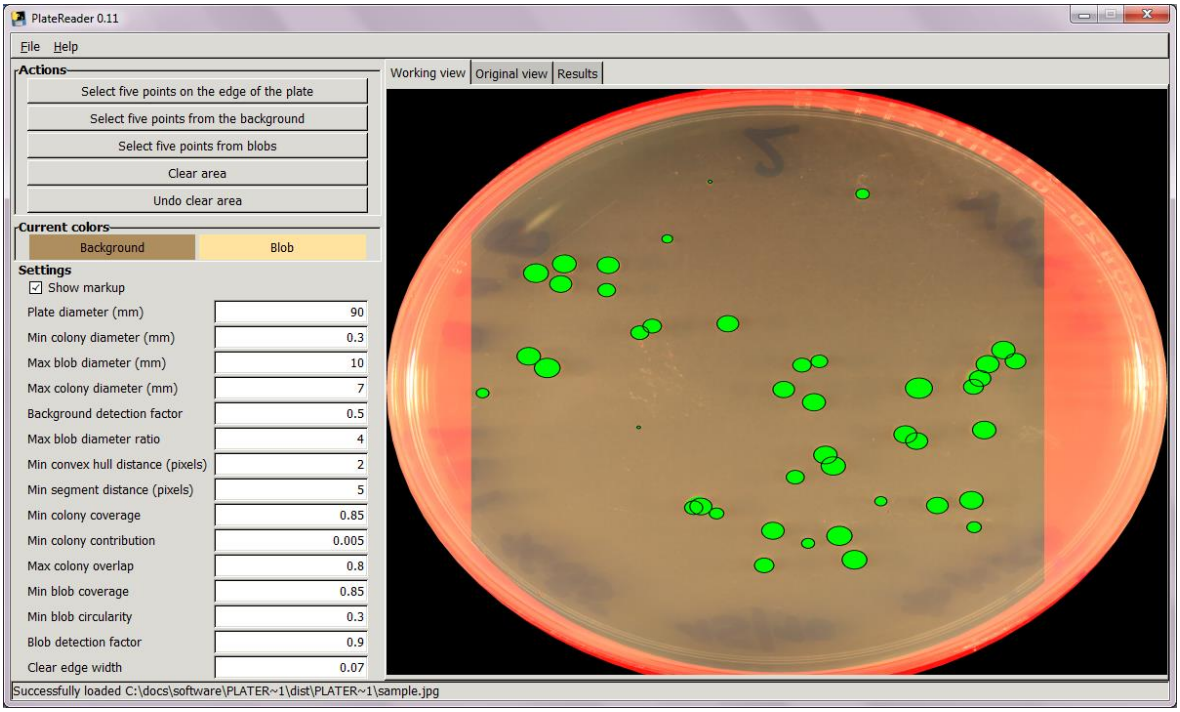
In this case there is one blob for which it wasn’t possible to identify the colonies. By default, PlateReader is very conservative when it comes to identifying a blob as correct (green). This is because it is often a larger error to wrongly assign a colony, compared to ignoring it and dealing with the yellow regions by averaging. Statistically, there should be little reason to think that colonies in yellow blobs are different from those in the green ones. There could be some size dependency since yellow blobs typically contain several colonies, and larger colonies are more likely to merge. However, this effect should be small and can be ignored. If we are mainly interesting in the size distribution of colonies we can therefore simply ignore the yellow blobs. If we are interested in counting colonies we can use the average size based on the green blobs to estimate the total count including the yellow ones. See Parameters for details regarding how to affect the classification of blobs.

If there is a significant fraction of yellow blobs, or if the red areas overlap with relevant blobs, you can try to select new background/blob colors. It is typically a good idea to use at least some points from the yellow blobs when doing this, as it makes it more likely that they are correctly fitted.

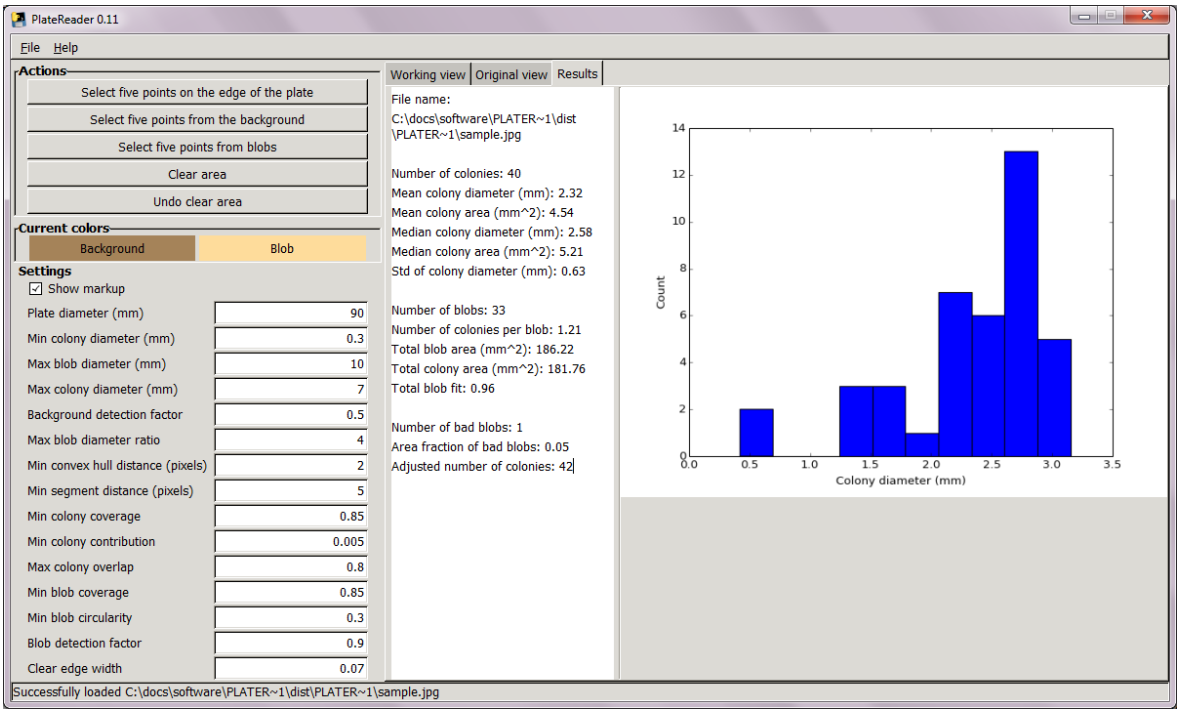


As can be seen, this led to that the yellow blob was now identified as consisting of three colonies, out of which two were largely overlapping. Due to the very irregular shape of this blob it's actually difficult to see whether there are two or three colonies, but given the current parameter settings the program determined that three was a better fit.

If there are regions which contain blobs wrongly identified as yellow or green you can exclude them from further analysis by choosing "Clear area" and selecting the area. This could for example be the case if there is text written on the plate and the interior region of the letter "O" is identified as a blob. In this case there are no such regions, but we can exclude the reflections to the left and right to illustrate.



Choosing suitable colors is somewhat of an acquired skill, so use trial and error until you get a feel for it. Switch to the Results view once you are satisfied with the colony detection. In this example we use the image with the yellow blob for illustration purposes.



This will display a histogram over the colony sizes together some statistics about the fitting. The first section deals with the number and size of the identified colonies (40 colonies with a mean diameter of 2.32 mm). This section is rather self-explanatory. The second section deals with the blobs and reports the number of blobs, the number of colonies per blob and the area covered by the blobs and colonies respectively. The areas should optimally be similar. Lastly, it shows the total blob fit. This is a value representing how good the colony fitting was (0.96 in this case). It is calculated as:

$$fit = 1 - \frac{A_{only colonies} + A_{only blobs}}{A_{blobs}}$$

That is, the sum of the area covered by only colonies and the area covered by only blobs divided by the total blob area. For a perfect fit with all green blobs there would be no regions where the blobs and colonies don't overlap, so this expression would be $1-(0+0)/A_{blobs}=1$. Note that the fit is calculated only for the green blobs.

The last section deals with the bad (yellow) blobs. In this case there is one bad blob and it takes up 5% of the total blob area. When this is taken into account the total colony count is predicted to be 42 (as compared to the 40 colonies calculated from only the green blobs). As discussed before, it is difficult to see whether the yellow blob consists of three or two intersecting colonies so this seems to be a reasonable prediction.

You can also export the results as a RTF (Rich Text Format) file by choosing File and Export to RTF. RTF files can be open in for example Microsoft Word or Wordpad.

Parameters

This section details the various parameters governing the colony identification. The default values are suitable in most cases, and this should be viewed as supplementary information for experienced users.

In order to understand the various parameters you would need to know something about how PlateReader performs the colony identification. There are two main steps; object identification and circle fitting.

1) Object identification

The purpose of this step is to transform the input image to a black and white version where relevant blobs are white and all other parts black. This is done in a number of steps:

1.1. First calculate the Euclidian distance from the blob color to the color of each pixel in the image. This results in a map where parts similar to the blob color have low values and dissimilar parts have high values

1.2. For each pixel in this map, calculate the variance based on the surrounding pixels. Blur the image by averaging several neighboring pixels before doing this in order to reduce the impact of noise. This results in a map where high-contrast areas (such as the border of a blob) have high values while uniformly colored areas (such as the background) have low values

1.3. Remove areas of the image which either have large values in the map from 1.1 or from 1.2. This leads to a map where objects such as blobs are left, but where the background and objects with different color have been removed

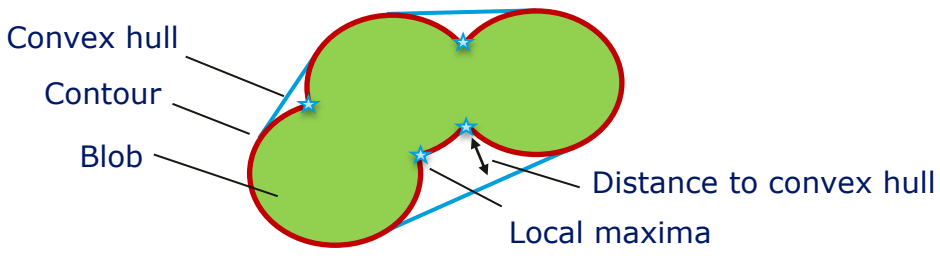
1.4. Calculate the contours for each object in the map from 1.3. A contour is a list of all the points on the edge of an object. Each blob should now be associated with one contour

1.5. At this point we have contours for each blob, but there are likely contours which represent other objects as well, such as glares on the plate or small speckles. Go through each contour and remove those with a size, color or shape which makes it unlikely to represent a blob

2. Circle fitting

2.1. For each contour, calculate what is called the convex hull. This is a type of geometric shape which contains all the points in the contour while having the property of convexity. The figure below illustrates the concept

2.2. For each point on the contour, calculate the shortest distance to the convex hull. Local maxima in this function represent intersection points between colonies (assuming colonies are circular)



2.3. For each contour segment between two local maxima, fit a circle using least squares

2.4. At this point we have a number of circles for each blob. The correct circle(s) is hopefully among them, but there will also be duplicate circles which should be removed. For example, there will be two circles fitted to the middle colony in the figure above; one when moving between the two upper maxima and one when moving between the two lower ones. There will also be circles fitted due to noise in the contour, which can result

in many small maxima. There incorrect circles are then filtered out based on a number of factors (see below)

2.5. In the last step the coverage of each blob is calculated, and blobs with insufficient coverage are classified as bad (yellow). All colonies for such blobs are ignored

Parameters

Note that you must press Enter after modifying a parameter value in order for it to be registered.

- **Plate diameter:** The diameter of the Petri dish (and the image after it has been cropped to fit the plate perimeter). This is used for converting between diameter in pixels, which is what you get from the fitting, and diameter in mm

- **Background detection factor:** How far from the blob color a pixel can be and still be considered as a belonging to a blob. This is expressed as fraction of the distance from the blob color to the background color. A value of 0 therefore means that pixels have to have exactly the same color as the blob color, and a value of 1 means that also the background is considered to belong to blobs. This parameter can be raised if there is a color gradient in the background and you're having difficulties with choosing background/blob colors which fit all blobs. This is used in step 1.3

- **Max blob diameter:** The largest allowed diameter of a blob. This is used in step 1.5

- **Max blob diameter ratio:** The largest allowed ratio between the longest axis of a blob and the shortest. This ratio would be 1 for circular objects and larger than 1 for all others. For example, a blob consisting of four equally sized colonies, aligned on a line and only minimally overlapping, would have a ratio of 4. This is used in step 1.5

- **Min blob circularity:** The minimal circularity for a blob. Circularity is a measurement of the ratio between the square of the length of the perimeter of an object and its area. A circle has a circularity of 1 and all other shapes have lower values. A line with no area would have a value of 0. The purpose of this is to identify contours which have a reasonable blob diameter ratio (or they would have been removed in the previous step) but which have high complexity. One such example can be letters, which are sometimes printed or written on the plates, and which have to be separated from the interesting blobs. Take a blob containing the letter "H" for example. It has a low diameter ratio, but its circularity would be much too low for it to possibly represent a blob with intersecting colonies. This is used in step 1.5

- **Blob detection factor:** The minimal fraction of points in a blob which must be above the background detection factor. This is to filter out hollow objects which have a border of correct color, but an interior with another color. This is used in step 1.5

- **Min convex hull distance:** The minimal distance to the convex hull a point must have to be considered as a candidate for being a local maximum, and thereby a possible intersection point between circles. This is to reduce the effect of noise in the contour. This parameter could be decreased if an image is of low resolution. Note that fractional values are allowed. This is used in step 2.2

- **Min segment distance:** Minimal length of a segment between two local maxima in the circle fitting. If local maxima are closer than this then the smallest of them is ignored. This parameter could be decreased if an image is of low resolution. This is used in step 2.2

- **Min colony coverage:** Minimal fraction of a colony being inside the blob. This is to exclude circles with a bad fit to the blob. This is used in step 2.4

- **Min colony contribution:** Minimal fractional contribution of a colony to the total coverage of a blob when taking into account all other colonies. This is to exclude colonies which may fit well to the blob, but where the other colonies already cover those regions. This is to remove redundant or duplicate circles (such as discussed in step 2.4). This is an iterative process in which the worst circle is removed first. This is used in step 2.4

- **Min colony diameter:** The smallest allowed diameter of a colony. This is used in step 2.4

- **Max colony diameter:** The largest allowed diameter of a colony. This is used in step 2.4

- **Max colony overlap:** The largest allowed fractional overlap between a colony and all the other colonies. This is to remove colonies which may give a contribution to the overall fit to the blob, but which does so by overlapping to a large extent with the other colonies. Any arbitrary shape could be fully covered by an infinitely large number of circles, but that would give little information about the real number of colonies in the blob. This is used in step 2.4

- **Min blob coverage:** The minimal fraction by which a blob should be covered by colonies in order for it to be considered a good (green) blob. If the coverage is lower it is changed to a bad (yellow) blob and all colonies on it are ignored. As discussed in the Interface section, it is normally preferable to only include blobs with good fits in the calculations. However, there could be situations where it's relevant to fit as many colonies as possible with little regard for good/bad blobs. If that is the case then this parameter can be lowered. This is used in step 2.5

- **Clear edge width:** It is difficult to identify colonies very close to the edge of the plate due to glare and the perspective being warped. It can therefore be better to ignore this area to reduce the risk of false positives (which would otherwise have to be removed manually). This parameter clears areas which are within a certain distance of the plate edge. This width is given in fractions of the plate diameter. Note that this is applied when the plate shape is fitted. If not suitable you can either set it to 0.0 prior to doing the fitting, or you can unclear the section manually afterwards