Colocalization tutorial

I. Introduction

Fluorescence microscopy is helpful for determining when two proteins colocalize in the cell. At the most basic level, images can be inspected by eye. Quantitative image analysis offers even more insight into colocalization. Colocalization, in image analysis terms, is defined as the likelihood of signals from different fluorophores to be present in corresponding pixels. There are a few ways to measure colocalization in CellProfiler; these will be discussed in **Methods**, below.

Before attempting to measure or quantify colocalization, it is important to inspect the images for common pitfalls, such as cross-talk and bleed-through. Cross-talk and bleed-through is the incomplete isolation of one channel from another. This can occur when the two fluorophore's excitation or emission spectra, respectively, partially overlap, or if the emission filters are not selective enough. We recommend papers by Bolte *et al* 2006 and Scriven *et al* 2008 for a more detailed description of these terms, as well as methods to correct for them. Scriven also explains how spatial under-sampling can cause errors in quantitative colocalization measures. It is important to use caution when binning pixels to increase intensity during image acquisition if you plan to measure colocalization later for this same reason.

II. Methods

Methods for measuring colocalization can be divided into two methods: pixel-based and object-based. Pixel-based methods look at all pixels in an image and determine various correlations between the channels. Object-based methods first segment the images into objects, and then make some comparison between the objects in the channels. CellProfiler allows both pixel- and object-based colocalization measurements, which will be described in the example pipeline.

For pixel-based approaches, the Pearson correlation coefficient is the most common colocalization measurement. Bolte *et al* review methods by Manders, Costes, van Steensel and Li; all methods look at pixel intensities in the two channels being considered for colocalization. CellProfiler computes the linear correlation between the two channels, and also the slope a of the line $y = ax + b$, where y and x are the two image intensities, indicating the overall relative intensity of the two images. In addition, CellProfiler also computes Manders' coefficients, Costes' automated threshold, Overlap and Rank Weighted Colocalization (RWC; Singan *et al* 2011) coefficients between the two channels.

Bolte *et al* also describe object-based approaches, which allow more spatial information to be inferred about colocalization, helpful in determining in which subcellular compartment(s) of the cell the stains may colocalize. Proper segmentation is very important in object-based methods, since different segmentations will give different

colocalization results. Pre-processing of the image, including filtering or illumination correction, may be necessary for proper segmentation.

Once images have been properly segmented (that is, the cells and subcellular compartments correctly identified in both channels), Bolte describes several methods by which to compare the colocalization. The first is to compare the centroids of the identified objects and define colocalized objects as those whose centroids are within some specified distance (usually, less than the optical resolution of the microscope). Another method is to multiply the two channels by one another after preprocessing and then to segment the structures of interest; the result of the multiplication is composed of only pixels present in both channels.

The CellProfiler colocalization pipeline demonstrates how to carry out the colocalization methods mentioned above. Here, we explain in detail the settings in the pipeline and how to adjust them to achieve good colocalization measurements.

Image Loading and Pre-processing

For this example, we will be measuring colocalization events between a histone-modified nucleosome that is labeled with a Cy3 dye and an antibody labeled with a Cy5 dye that is sensitive to the histone modifications. Even though the features of interest in this case are simple spots, the principles described here are applicable to cells and subcellular features as well.

Module 1: Load Images
Load each channel (or stain) as a separate image. If you have a color image composed of different stains, you'll need a **Color To Gray** module after **Load Images** to separate the incoming image into its component channels.

In this example pipeline, we call the two images *OrigStain1* and *OrigStain2*.

Module 2 and 3: Correct Illumination – Calculate

> **Before proceeding:** *Please gain a basic understanding of the options within "Correct Illumination – Calculate" and "Correct Illumination – Apply" by reading the help for the module: select it in the module panel at the top left of the CellProfiler window, then click the Click the Help button ("?") under the module panel on the left.*

The input image is named *OrigStain1*; the output illumination correction function (ICF) is named *IllumStain1*. The ICF is calculated using the *Regular* method which uses the foreground pixels to estimate the ICF. Since the ICF made using the *Regular* method is applied by division (see next module), it needs to have a minimum value of 1 to maintain the range of the correction and avoid division by zero, so we enable rescaling. Choose *Each*

to calculate the ICF to correct each image individually; if the input images are acquired as a large collection of images during the same acquisition run with identical acquisition settings, chose *All*. Since the foreground and background intensities appear to be fairly smooth, with no obvious large-scale heterogeneities, the smoothing method chosen is *Fit Polynomial*. Please see our tutorial "Calculating and applying illumination correction for images" on our Tutorials page for more details.

The second **Correct illumination – Calculate** module uses the same settings as for Module 2, with the input image set to *OrigStain2* and the output image is named as *IllumStain2*.

Module 4: Correct Illumination – Apply
The input image is selected as *OrigStain1*, the output image is named *CorrectedStain1* and the illumination function is selected as IllumStain1. As mentioned above, if Regular is chosen as the ICF creation method, the ICF should be applied by division.

To apply the second ICF to *OrigStain2*, the "Add another image" button was pressed to reveal another group of settings for a second image. Select *OrigStain2* as the input image, name the output image *CorrectedStain2*, name the illumination function *IllumStain2*, which is applied by division.

Module 5: Align (optional)
Since accurate colocalization requires accurate positioning of the features in both images, it is sometimes worth using this module to align the images. If aligning the images, it is important to remember that there needs to be sufficient overlap in image features, other than the features suspected of overlapping, in order to align them. For example, attempting to align two images in which there is little to no colocalization will probably result in poor alignment.

In this case, a 1 pixel adjustment is needed to align the images, and the aligned images are named *Stain1* and *Stain2* (Fig. 1).
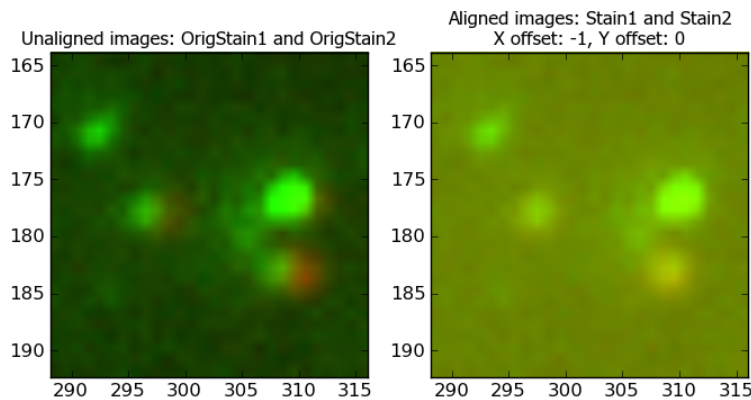
Unaligned images: OrigStain1 and OrigStain2

Aligned images: Stain1 and Stain2
X offset: -1, Y offset: 0

Figure 1: Zoom-in of the input and output images as displayed by the **Align** module. Note that the input images are shaded green and red, so that aligned spots appear as orange.

Pixel-based Correlation

Here, we will measure correlation by comparing the pixel intensities in each image and determining the correlation between them.

Module 6: Measure Correlation

The **Measure Correlation** module measures the correlation and colocalization coefficients including Manders, Overlap and Rank-Weighted Colocalization (RWC) between *Stain1* and *Stain2* across the entire image. It also provides Manders' coefficient based on Costes' automated threshold computations. The overall image correlation can give a general sense of how colocalized the features within the images are. Additional colocalization measures described in Singan *et al* 2011 and reviewed in Bolte *et al* 2006 are provided to give wider options for quantitation.

The correlation coefficient is the normalized covariance (covariance divided by the product of standard deviation of pixels in each image). This measurement is equivalent to Pearson's correlation, as described in Methods. Correlation ranges from -1 (complete inverse correlation) to +1 (complete correlation). Thus, the closer to one the correlation measurement is, the more correlated the two images are and the higher the amount of colocalization.

For this pair of images, the correlation coefficient is 0.46 and the slope is 2.22. Note that if the unaligned images are used, the correlation coefficient drops to 0.26, indicating that the alignment was necessary. The slope indicates here that the average pixel intensity in one image is more than twice that in the other image.

Manders and Overlap coefficients address specific biases introduced in Pearson's correlation including channel intensity biases. Manders coefficients are computed based on the proportion of signal overlap, independent of pixel intensities. The Rank-Weighted Colocalization coefficient combines both the pixel intensity information (correlation) and overlap information (co-occurrence) using a weighting scheme (Singan *et al* 2011) to measure colocalization. The colocalization coefficients ranges from 0 to +1.

Setting an appropriate threshold to eliminate background is critical for measuring Manders and Overlap coefficient. The threshold can be set as percentage of maximum intensity in each channel. A value of 15 suggests that background intensity values that are 15% or lower than the maximum intensity in that channel is considered noise and an intensity subtraction of this value across the entire image is used to eliminate the background before computing the Manders, Overlap and RWC colocalization coefficients.

Note that if you are not interested in object-based correlation, you could stop here and remove the remaining modules up to the final **Export To Spreadsheet** module. The

Measure Correlation module can also be used to calculate correlation and colocalization coefficients within objects.

Object-Based Correlation

Similarly, correlation and colocalization measurements for individual objects can also be obtained. However, to determine colocalization on per-object basis, the objects within the image must be identified. We first segment the image features into objects, then make comparisons between the individual objects in the channels.

Module 7 and 8: Identify Primary Objects
The input image is selected as *Stain1*, with the output objects named *Objects1*. The typical diameter is set as [3,15] for the min/max size we expect the objects to be. We chose to discard small and large objects, which tend to be spurious, and discard those objects at the border because we will be making area-based measurements on complete objects.

The chosen thresholding method can greatly affect segmentation. Here, you want to select a method that will accurately identify the features of interest as foreground. Depending on the background level and properties of the stain, you may need to try several different methods and corresponding settings to obtain good segmentation. Please see the help for **Identify Primary Objects** for more information on the thresholding methods available.

In this case, three-class Otsu thresholding was used, with the middle class set to background, as this approach often works well in images with few foreground pixels.

Optimizing the settings to distinguish clumped objects is important for per-object measures of colocalization. For example, if you wish to measure colocalization only in the nuclei or cytoplasm, each sub cellular compartment must be properly segmented to provide an accurate measurement. Since the spots are punctate with the brightest region in the center, both the *Intensity* declumping method and the *Intensity* method to draw the dividing lines will work well. More guidance on these settings can be found in the help for this module.

As a quality control measure, we chose to save the outlines of the identified objects and name the outline image *Stain1Outlines*.

The **Identify Primary Objects** settings for module 8 are identical to Module 7, but here we identify the objects from *Stain2*.

Now that we have identified both sets of objects, there are several approaches to measuring colocalization.
(1) *Measure Correlation module:* Use the Measure Correlation Module with the option to measure colocalization within objects and choose the objects over which the measurements need to be calculated. Colocalization coefficients are calculated over the object regions between the pair of images. See modules 6, 9.

David Logan 3/22/2016 4:15 PM
Comment [5]: Redundant

(2) *Object overlap:* Objects which touch or overlap each other are considered to be colocalized. See module 10.

(3) *Object centroid distance:* In this case, only objects whose centers are N pixels apart or less are considered to be colocalized. The key here is to reduce the detected objects down to objects that share the same center location but are only N pixels wide; if these reduced objects touch or overlap, they are considered colocalized. In this example, we are considering only those objects in which the centers are within 2 pixels of each other. See modules 11-14.

Module 9: Measure Correlation
The Measure Correlation module used in Module 6 for whole images can also be used for quantifying colocalization within objects. For the pair of images *Stain1* and *Stain2*, colocalization coefficients are measured for regions defined by *Objects1* and *Objects2* independently.

Module 10: Relate Objects
For option (1) above, this module establishes a "parent-child" relationship between two sets of objects. A "parent" object in one image is one that touches, overlaps or encloses a "child" object in the other image. (Note that the particular assignment of parent or child is not important here). *Objects1* objects that touch or overlap with an *Object2* object are considered to be colocalized and will be assigned as a parent to a corresponding child. All others have no children and are labeled accordingly.

In addition, the distance between object centroids may also be obtained with this module by enabling the "Calculate distances?" setting.

Module 11 and 12: Expand Or Shrink Objects
For option (2) above, this module shrinks the objects identified in the *Stain1* image to a point and names the resultant points *ShrunkenObjects1*. The second **Expand Or Shrink Objects** does the same for *Stain2*.

Module 13 and 14: Expand Or Shrink Objects
We now expand the previously shrunken point objects by 2 pixels, i.e., two pixels are added to either side of the single-pixel object to create new objects which are 5 pixels across, as shown in Fig. 2. These new objects are named *ExpandedObjects1* and *ExpandedObjects2*.
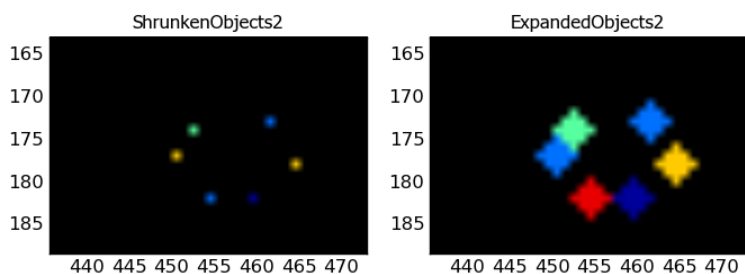
Figure 2: Result of using **Expand Or Shrink Objects** on the point objects created by module 11. The new objects are expanded by 2 pixels, making them 5 pixels across.

Module 15: Relate Objects

In this case, *ExpandedObjects1* are assigned (arbitrarily) to be parents, with *ExpandedObject2* as children. Therefore, objects in *ExpandedObjects1* which have children and ≤ 2 pixels apart are colocalized with objects in *ExpandedObjects2*.

Calculate and Export Measurements

Module 16 and 17: Classify Objects and Filter Objects

The **Classify Objects** module categorizes objects associated with a particular measurement. In this case, the chosen measurement is the child count for *Objects1*. The value 0.5 is chosen as the measurement since we want to distinguish between colocalized objects with children (child count of 1 or greater) versus those without (child count of 0).
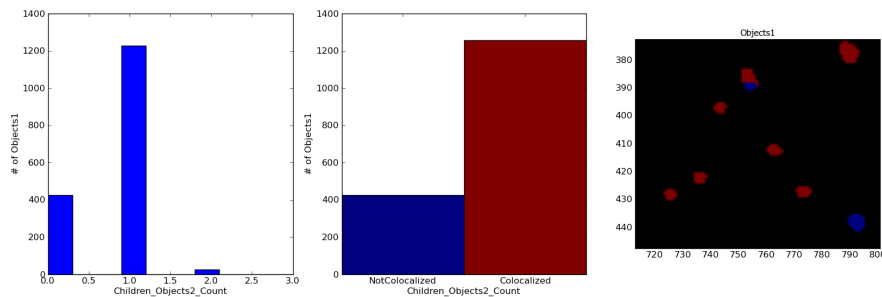
Figure 3: Output of the **Classify Objects** module showing histogram of measurement values (left), histogram of objects falling into specified measurement bin (middle) and zoom-in of objects colored by measurement bin.

The result of this module is an absolute count and percentage of objects that fall into the colocalized/non-colocalized bins, and an annotation to each *Objects1* object as to whether it falls into a particular bin or not (Fig. 3). This approach works well if you simply desire a yes/no categorical readout per object.

The **Filter Objects** module effectively does the same operation as **Classify Objects**, but rather than simply assigning a label to each object, **Filter Objects** removes all objects that do not pass a criterion. Using the same choice of measurement and cutoff as in **Classify Objects**, only those *Objects1* objects which fall into the colocalized bin are retained (the red objects in Fig. 3). This feature is useful if you want to perform additional operations or measurements on the remaining objects, since the objects that pass the filter are assigned as a new object class within CellProfiler.

These two modules can also be used to perform the same operations on the *ExpandedObjects1* object set.

Alison Kozol 4/21/2016 3:16 PM
Formatted: Font:12 pt, Bold Not , Check spelling and grammar

Alison Kozol 4/21/2016 3:16 PM
Deleted: 4

Alison Kozol 4/21/2016 3:16 PM
Deleted: 4

Alison Kozol 4/21/2016 3:16 PM
Formatted: Font:12 pt, Bold Not , Check spelling and grammar

The next couple of modules are used to determine the area occupied by the colocalized region as a fraction of the total area. The closer to one the proportion is, the more colocalized objects are present in the image.

Module 18: Mask Objects
In order to find only the overlapping regions, this module masks (i.e., "hides" from consideration) the pixels of the *Objects1* object set which are in common with the *Objects2* object set. The result of this operation is the colocalized area as a new object set, named *ColocalizedRegions* (Fig. 4).
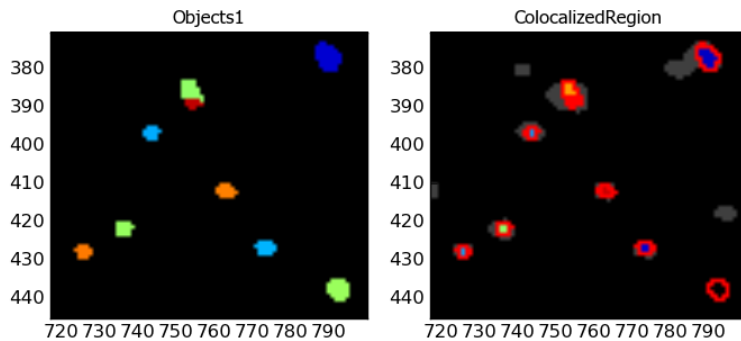

Figure 4: Output of the **Mask Objects** module. Left panel: the *Objects1* object set. Right panel: *Objects1* object set in red outline, *Objects2* objects set as gray blobs, colocalized regions as colored regions.

Module 19 and 20: Measure Image Area Occupied and Calculate Math
The **Measure Image Area Occupied** module measure various statistics associated with the defined area in an image. In this case, we are concerned with *Objects1* and *ColocalizedRegion*. The module counts all the pixels occupied by a given object set resulting in a total area occupied by each stained object. In **Calculate Math**, we divide the area occupied by *ColocalizedRegion* by the area occupied by *Objects1* to get a per-image fraction of colocalized pixels out of the total object pixels.

Module 21: Export To Spreadsheet
This module is used to export the full set of measurements obtained by the pipeline. Measurements such as object counts, colocalization percentages and area fractions are saved to a per-image file (that is, one value per image); measurements such as colocalized/non-colocalized status and centroid distances are saved to a per-object file (one value per object).

I. References

Bolte S, Cordelieres FP. A guided tour into subcellular colocalization analysis in light microscopy. *Journal of Microscopy* 224: 213-232 (2006).

Scriven DR, Lynch RM, Moore ED. Image acquisition for colocalization using optical microscopy. *American Journal of Physiology - Cell Physiology.* 294: C1119-22 (2008).

Manders EEM, Verbeek FJ, Aten JA. Measurement of co-localization of objects in dual-colour confocal images. *Journal of Microscopy* 169: 375-382 (1993).

Costes SV, Daelemans D, Cho, EH, Dobbin Z, Pavlakis P, Lockett S. Automatic and quantitative measurement of protein-protein colocalization in live cells. *Biophysical Journal* 86: 3993-4003 (2004).

van Steensel B, van Binnendijk EP, Hornsby CD, van der Voort HTM, Krozowski ZS, de Kloet ER, van Driel R. Partial colocalization of glucocorticoid and mineralocorticoid receptors in discrete compartments in nuclei of rat hippocampus neurons. *Journal of Cell Science* 109: 787-792 (1996).

Singan VR, Jones TR, Curran KM, Simpson JC. Dual channel rank-based intensity weighting for quantitative co-localization of microscopy images. *BMC Bioinformatics* 12: 407 (2011).

Li Q, Lau A, Morris TJ, Guo L, Fordyce CB, Stanley EF. A syntaxin 1, Gao and N-type calcium channel complex at a presynaptic nerve terminal: analysis by quantitative immunocolocalization. *The Journal of Neuroscience* 24: 4070-4081 (2004).