

Dr. Bob on Colocalization or MSL Experiments In Learning Colocalization Using Image-J

Confocal microscopy is used to test whether two fluorescently labeled molecules are associated with one another. If one is labeled red and the other is labeled green, an image that combines them may show a yellow color where the red and green pixels overlap. Is this colocalization? Can we do any better than that? The following notes are my attempt to understand and convey the current state of colocalization in confocal microscopy.

These notes require the use of the software program Image-J. This is free, open source, software written by Wayne Rasband at NIH and others. It will run on any computer platform. You can download it from the following web address: <http://rsb.info.nih.gov/ij/>. You will also need several plug-ins for Image-J that must be downloaded separately. These are plug-ins from Tony Collins at the Wright Cell Imaging Facility in Toronto Canada. I have included the plug-ins on the MSL web page with this tutorial (www.med.unc.edu/microscopy). That page also contains images that you can use in this tutorial. If you wish to get the latest versions of the colocalization plug-ins, go to <http://www.uhnres.utoronto.ca/wcif>, under Software, under Individual Plugins, download and install into ImageJ the Colocalization_Test, Colocalization_Threshold, and Manders Coefficients plug-ins. Simply place the plug-ins in the Plug-ins folder in the ImageJ folder to install them. Also download any PDF or manual files for the plug-ins.

The experiments below require images made using Image-J. I have included already-made versions of the images on the web page. I indicate which images you should open at each experiment. (It may be more instructive if you create your own images for the experiments using Image-J.)

Let's get started:

Q. Can I use the yellow color in my overlay image as an indication of colocalization?

A. Where you see yellow, there probably is some colocalization, but there can be colocalization where you see no yellow. The intensity of the red and green signals determines the yellowness. And the intensity may result from more labeled compound or a higher PMT setting, among other things. **You cannot trust your eyes in colocalization!**

Here is a little experiment using Image-J to illustrate what I mean.

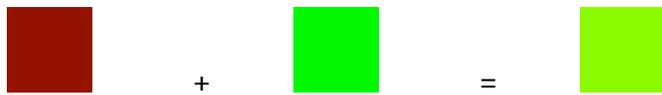
Experiment to show the effect of intensity in Red-Green overlap:

Using Image-J, open the images Red, Green, Red minus 128, and Red minus 192. These are 36 X 36 pixel RGB images, one filled with Red (255,0,0) and one filled with Green (0,255,0).

Add Red and Green together to make a new yellow image: Process/Image calculator/Add. Then add Red minus 128 and Green together. Finally add Red minus 192 and Green together. Look at the results.



Adding Red (intensity 256) and Green together to get Yellow



Reducing the Red intensity by 128 and adding to Green



Reducing the Red intensity by 192 and adding to Green

You know that the last image has both red and green, but it looks green. Simply changing the intensity of the red image changes the color of the colocalized image and, as you are about to learn, intensity in colocalization does not matter as long as it is above background. So, your real image from the microscope may have colocalization that you cannot see. Seeing yellow is a very subjective measure of colocalization. Something better is needed that does not depend on combining colors in the RGB color space.

Q. So, what's a better method?

A. The better method is to determine, mathematically, how interdependent the two different channels, taken at the same time are. The amount of interdependence, would determine the amount of colocalization. In statistics, interdependence of two variables is calculated using a correlation coefficient. In microscopy, the first attempt to quantify colocalization was by Manders who used the Pearson correlation coefficient (1). Let's do a simple experiment that calculates Pearson's correlation coefficient.

Experiment to calculate and test Pearson's Correlation Coefficient using Image-J:

In ImageJ open the images Overlap 0, Overlap 50, Overlap 20.

Here are three 36 X 36 pixel 8-bit images filled with black. Call the first one Overlap 0 the second Overlap 50 and the third Overlap 20. A 12 X 12 pixel box filled with 128 is in Overlap 0. Overlap 50 has the same size and intensity box positioned so that it will overlap the box in 0 by 50%. Overlap 20 has the same size and intensity box positioned so that it will overlap the box in 0 by 20%.



Overlap 0



Overlap 50



Overlap 20

In ImageJ, run the Manders Coefficients plug-in using the 0 and 50% images as input: Plugins/Manders Coefficients/Check only the box that says Current Slice Only.

Repeat the experiment with the Overlap 0 and Overlap 20 images as input. The results window should look like this -

Results							
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Overlap 0.tif and Overlap 50.tif	0.446	0.500	1.000	0.500	0.500	0; 0	0; 0
Overlap 0.tif and Overlap 20.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0

First, let's look at the Pearson's coefficient. It is in the results window under the Rr heading. It should be 0.446 in the 50% overlap and 0.113 in the 20% overlap.

The range for Pearson's is -1 to 1 with -1 being total negative correlation, 0 being random correlation, and 1 being total positive correlation. Looking at the results, obviously Pearson's is not telling us the percentage of overlapped pixels. The best description I have found for Pearson's is that it is telling us what percentage of variability in one channel is caused by variability in the other channel. You get at this number by squaring Rr and making it a percentage: for the 50% overlap images it is $0.446^2 = 19.9\%$ and for the 20% overlap images it is $0.113^2 = 2.8\%$. These numbers are difficult to interpret in terms of the actual colocalization we created.

Now let's add 64 to the entire area of Overlap 20 including the 12 X 12 pixel box.

In ImageJ open Overlap 20 plus 64.



Overlap 20 plus 64

Now re-run the plug-in using Overlap 0 and Overlap 20 plus 64.

Results							
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Overlap 0.tif and Overlap 50.tif	0.446	0.500	1.000	0.500	0.500	0; 0	0; 0
Overlap 0.tif and Overlap 20.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0
Overlap 0.tif and Overlap 20 + 64.tif	0.113	0.328	0.098	1.000	0.114	0; 0	0; 0

The result for Pearson's is still 0.113. Pearson's is not sensitive to the intensity of the background or to the intensity of the overlapping pixels. Both of these are very good properties.

Now let's look at the Manders overlap coefficient. It is in the results window under R.

Manders (1) created a Colocalization Overlap coefficient that can be more easily interpreted than the Pearson coefficient. Manders' Overlap coefficient R ranges between 0 and 1. Look at the Manders' Overlap coefficient labeled R in the results window. In the 50% overlap test R = 0.500 and in the first 20% test R = 0.200. Here, Manders' Overlap is telling us the percentage of pixels that are overlapped.

But, when we changed the intensity of the background and the overlapping pixels in the second 20% overlap test, Manders' Overlap gave a value of 0.328. Manders' is sensitive to differences in background intensity.

But, surprisingly, it is not sensitive to overlapping pixel intensities.

Experiment to show non-dependance of Manders on overlapped pixel intensity:

Using Image-J open the image Overlap 20 box 64. This is the 20% overlap image but with the 12 X 12 square filled with 64 instead of 128. The background is black.



Overlap 20 box 64

Run the Manders' plug-in using Overlap 0 and Overlap 20 box 64.

Results							
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Overlap 0.tif and Overlap 50.tif	0.446	0.500	1.000	0.500	0.500	0; 0	0; 0
Overlap 0.tif and Overlap 20.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0
Overlap 0.tif and Overlap 20 + 64.tif	0.113	0.328	0.098	1.000	0.114	0; 0	0; 0
Overlap 0.tif and Overlap 20 box 64.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0

R is now back to 0.200. Manders' Overlap coefficient is sensitive to changes in the background but not to differences in the intensity of overlapped pixels.

Finally, let's take a look at the Manders' Colocalization coefficients M1 and M2. M1 is the percentage of above-background pixels in Image-1 that overlap above-background pixels in Image-2 and M2 is the percentage of above-background pixels in Image-2 that overlap above-background pixels in Image-1. For the original 20% overlap test, both of them read 20%. But for the 20% overlap to which 64 had been added to the whole image M1=1 or 100% and M2 = 0.114 or 11.4% which are not interpretable. However, for the last 20% overlap test in which the backgrounds were both 0 but the box in Image-2 was set to 64, rather than 128, M1 and M2 were both 0.200 or 20%. So M1 and M2 are sensitive to background but not sensitive to overlapping pixel intensities.

Experiment to show the relationship between M1 and M2:

Using Image-J, open the image Overlap 50 plus 1 box. This is the Overlap 50 image with the overlap box filled with 128 with another box included in the image. The second box is a 6 X 6 pixel box that overlaps neither the box in this image nor the box in the Overlap 0 image. We fill the 6 X 6 box with 128.



Overlap 50 plus 1 box

Run the Manders' plug-in using Overlap 0 and Overlap 50 plus 1 box.

Results							
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Overlap 0.tif and Overlap 50.tif	0.446	0.500	1.000	0.500	0.500	0; 0	0; 0
Overlap 0.tif and Overlap 20.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0
Overlap 0.tif and Overlap 20 + 64.tif	0.113	0.328	0.098	1.000	0.114	0; 0	0; 0
Overlap 0.tif and Overlap 20 box 64.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0
Overlap 0.tif and Overlap 50 plus 1 box.tif	0.380	0.447	0.800	0.500	0.400	0; 0	0; 0

M1 = 50%, but M2 = 40%, and this is correct: Total gray pixels in Image 1 = 144, total grays in image 2 = 180 and 72 grays in Image 2 overlap grays in Image 1 or 40%.

Conclusion:

With a background of 0, M1 and M2 correctly report the percentage of overlap for each color and this is not dependant on intensity of the overlapped pixels.

Take a big breath and relax!

This brings us to the next point, that of setting a background threshold.

Up to now we have made our test images with a black background and we have seen that varying backgrounds cause trouble. Real images have varying background and specimen intensities. In order to calculate meaningful colocalization, the background of each image must be determined and adjusted to 0. The threshold value below which all values are set to 0 will be different for each of the two images being compared.

Experiment to test the effect of setting a threshold:

In Image-J, open the images Overlap 0 and Overlap 20 plus 64. In this image we added 64 to all of Overlap 20.

Run Manders' on Overlap 0 and Overlap 20 plus 64.

You get the strange values of M1 = 1.000 and M2 = 0.114 as before.

Now, set a threshold for Overlap 0 and Overlap 20 plus 64 making the background black:

In ImageJ click on each image and do: Image/Adjust/Threshold and check the Dark background box. For Overlap 0 set the top slider to 128 and the bottom slider to 255 and click Apply (the outer box should be black and the inner box white). For Overlap 20 plus 64, set the top slider to 65 and the bottom slider to 255 and click Apply (the outer box should be black and the inner box white).

Run the Manders' plug-in on the adjusted images.

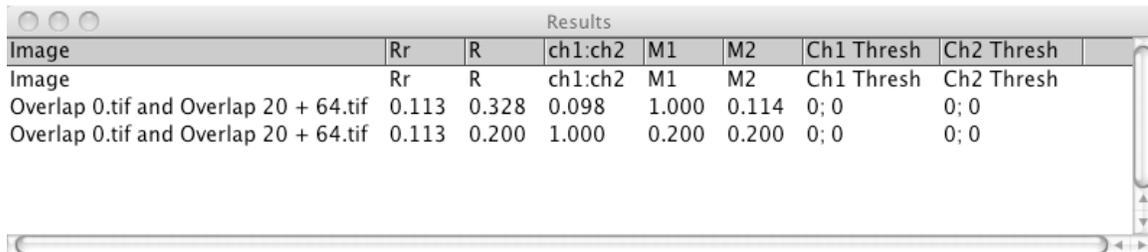


Image	Rr	R	ch1:ch2	M1	M2	Ch1 Thresh	Ch2 Thresh
Overlap 0.tif and Overlap 20 + 64.tif	0.113	0.328	0.098	1.000	0.114	0; 0	0; 0
Overlap 0.tif and Overlap 20 + 64.tif	0.113	0.200	1.000	0.200	0.200	0; 0	0; 0

M1 now = 20% and M2 = 20% which is the same answer as when we compared Overlap 0 and Overlap 20, both of which had black backgrounds.

If only setting a threshold on real images was that easy! Threshold choice is one of the major pitfalls of colocalization. Humans are no good at it, especially if they have a vested interest in the outcome.

The time has come to introduce the work of Costes (2).

But first, let's take a look at a different way of displaying the image data.

Data from the two images being compared can be displayed as a scatterplot. The scatterplot, is a graph in which the x and y axes represent the pixel intensity scale for Image-1 and Image-2 respectively (often called R and G). The intensity values for corresponding pixels in R and G are used as coordinates to place points in the scatterplot. The more points there are at any one location in the scatterplot the brighter that location becomes.

Demonstration of a scatterplot:

In Image-J, close all images.

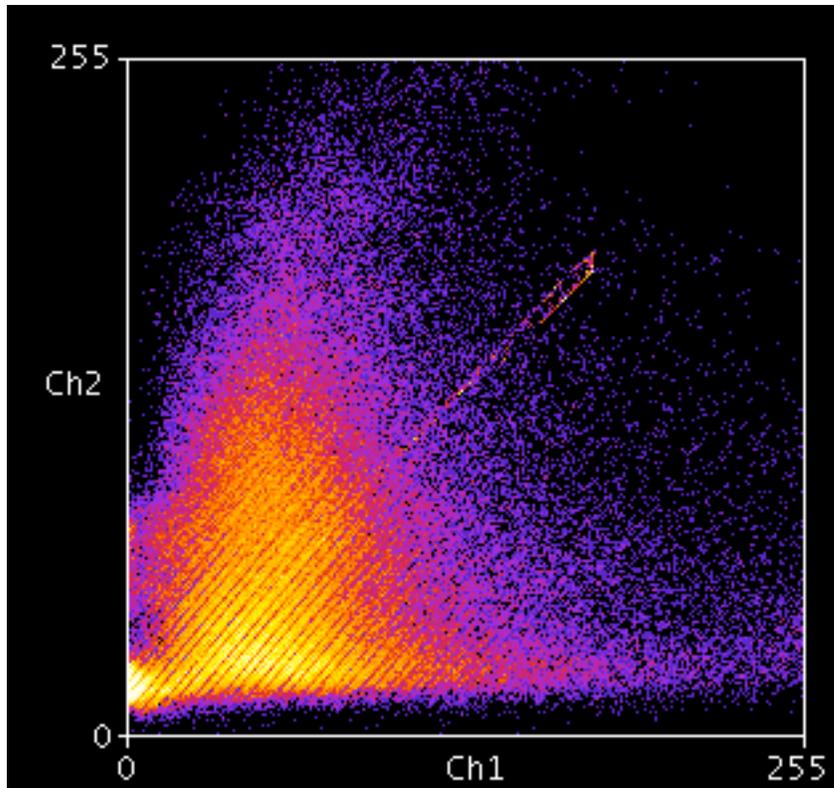
Open the Fluorescent Cells test image: File/Open Samples/Fluorescent Cells.

Separate the stack into separate images: Image/Stacks/Stack to Images.

Close the Blue image.

Run the Manders plug-in with R as Image-1 and G as Image-2 and checking only the Use Current Slice Only and Display Frequency Scatterplot boxes.

You will see a scatterplot of these two images.



Red and Green Frequency Scatterplot

Now back to Costes.

Costes' method calculates M1 and M2, but it uses a method for automatically setting the threshold based on the scatterplot and minimizing Pearson's coefficient. (More about this minimizing later.) Because of this dependence on Pearson's, Costes first tests the statistical significance of Pearson's. To do this, he calculates Pearson's for the two images. Then he tests this result against a number of trials in which the first image is compared to the second image, but the pixel values of the second image are "scrambled" to simulate random overlap. From this, a p-value is derived for the probability of real colocalization in the original images. This test must be passed at the 95% level of confidence or there is no reason to proceed. That is, there is no colocalization.

Experiment to test the statistical significance of Pearson's:

Using Image-J, open the Fluorescent Cells images as before keeping only the red and green images.

Run the Colocalization Test plug-in. Use Red for image 1 and Green for image 2, None for ROI or Mask. Choose the Costes randomization method. Use the default settings in the second window. Be patient, it takes the test some time to run because it has to create the randomized images.

Image	R(obs)	R(rand) mean±sd	P-value	R(rand)>R(obs)	Iterations	Randomisation	PSF width
Image	R(obs)	R(rand) mean±sd	P-value	R(rand)>R(obs)	Iterations	Randomisation	PSF width
Red and Green. slice: 1	0.400	-0.000±0.002	1.000	0/10	10	Costes X, Y	0.514 µm (1 pix.)

You get a p-value of 1 indicating >95% certainty that colocalization exists.

Once it is certain that there is colocalization, here's how Costes' method works: Costes creates a scatterplot from the two images. He then calculates an orthogonal line in the scatterplot with a slope determined by a least-squares fit based on orthogonal regression. (This is a standard statistical procedure that I will not cover here.) Next, starting at the brightest point on that line T, he calculates Pearson's coefficient for all pixels with values below T. Then T is sequentially lowered along the line and Pearson's recalculated until Pearson's equals 0 (i.e. random correlation). At this point on the orthogonal line the threshold has been reached for each image. It is found by looking up the respective coordinate values on the x and y axes. After setting the threshold for both images, M1 and M2 are calculated.

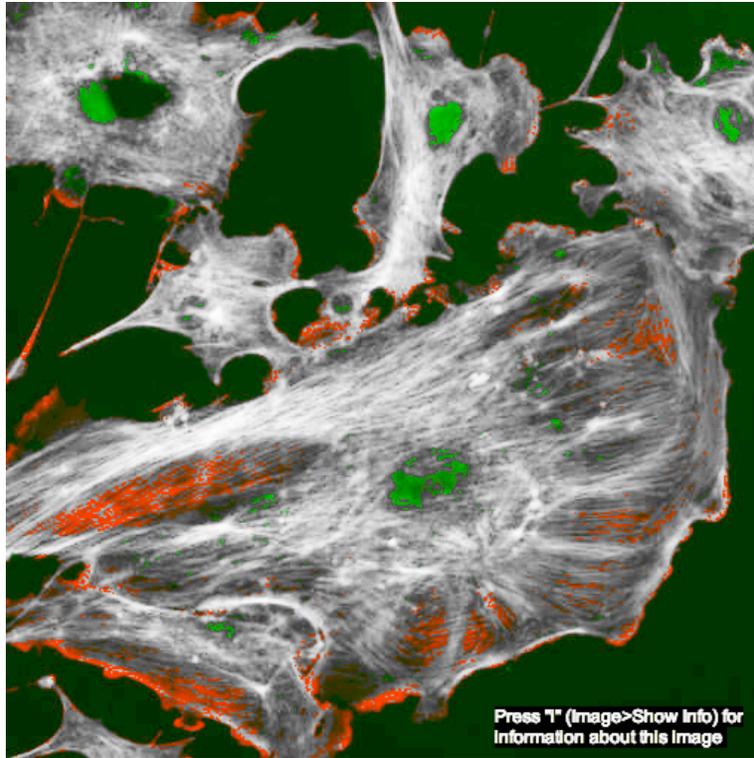
Experiment to Calculate Colocalization according to Costes:

Using Image-J, open the Fluorescent Cells images as before keeping only the red and green images.

Run the Colocalization Threshold plug-in with the red image as Image-1 and the Green image as Image-2. Use the default settings.

Ch1 thresh	Ch2 thresh	Rcoloc	R<threshold	M1	M2	tM1	tM2	Ncoloc	%Volume	%Ch1 Vol	%Ch2 Vol	
Ch1 thresh	Ch2 thresh	Rcoloc	R<threshold	M1	M2	tM1	tM2	Ncoloc	%Volume	%Ch1 Vol	%Ch2 Vol	
4.3	11	23	0.5722	-0.005	1.0000	0.9074	0.9106	0.8755	157984	60.27%	88.98%	91.4%

This plug-in generates a slew of values. (The plug-in is documented in the Colocalization Analysis PDF file on the web page.) The most important ones are tM1 and tM2. These are M1 and M2 using the calculated thresholds. In this example they are 0.9106 for tM1 and 0.8755 for tM2. This indicates that, in the thresholded images, 91.06% of the red pixels colocalize with green pixels and that 87.55% of the green pixels colocalize with red pixels. You can show the colocalized pixels in a new window by checking the box Show Colocalized Pixels, and you can choose to have those pixels all be white (Use Constant Intensity for Colocalized Pixels box checked) or let them be varying shades of gray based on the calculation $\sqrt{\text{Ch1} \times \text{Ch2}}$ as shown below. In this image, pixels that are red or green are not colocalized.



The Colocalization Thresholds plug-in operates on the entire image. There is no option to set a region of interest (ROI). Often it is advantageous to set an ROI for analysis – say to include just a single cell or to exclude an artifact.

Experiment to Calculate Colocalization on only part of an image:

Quit and restart Image-J to reset the Results window.

Using Image-J, open the Fluorescent Cells images as before keeping the Red and Green images.

Make about a 64 X 64 pixel selection in the red image and move it to a location that seems to also have data in the green image: Select the box drawing tool/Click and drag while watching the x and y dimensions in the info box. Click and hold inside the box while dragging it around.

Run the Colocalization Test plug-in : red as 1, green as 2, ROI in image 1, Fay method of randomization, and observe the p-value result.

If $p < 0.95$, move the box to another location and run the plug-in again.

If $p \geq 0.95$, select the green image by clicking on its title bar (do not click in the image space as that will remove the selection).

Do: Edit/Selection/Restore Selection to create a selection in the green image in the same location as that in the red image.

Next do: Image/Crop on each image. Remember, select the image by clicking on its title bar.

Run the Colocalization Threshold plug-in on the cropped images.

Take two big breaths and relax!

You now know something about quantitative colocalization in fluorescence microscopy and you have the software (Colocalization Test and Colocalization Threshold) to do the calculations. In reporting your data, include, for each selected area, the p-value for Pearson's coefficient, the tM1 and tM2 colocalization values and the colocalization image.

Finally, all this depends on having high quality images.

Microscopy technique:

Good microscopy technique is absolutely critical in acquiring images for colocalization. All pixel values should be on scale and the system optimized for minimum image noise. Care should be taken not to photobleach the sample. There should be no bleed-through of one channel into the other (images should be acquired sequentially). The images must all be in register in x, y, and z. Registration in z for large image stacks can be problematic as the optical section thickness changes with wavelength. A background correction is also essential to eliminate system variability in the image background (Model & Burkhardt 3, Brakenhoff et.al. 4). If calculations are to be made on z-stacks, then the z-stacks should be deconvolved. Help in acquiring high quality images for analysis is available in the MSL or any other light microscopy core facility at UNC.

Commercial software for colocalization:

There are several commercial software packages that do Colocalization calculations including most of the software that comes with confocal microscopes. A quick search of the term "colocalization" on the internet will turn up most of them. This set of notes should give you a good background for understanding them. Many of them allow you to draw a region of interest in the scatterplot while the results of your changes are displayed in the colocalization image. I believe that this is a bad technique since it amounts to you setting a threshold for each of the images. A good software package should first calculate a p-value on Pearson's coefficient, using some accepted randomization method for the second image, to see if colocalization

exists. If colocalization exists, the software should then set an automatic threshold, calculate tM1 and tM2 and produce a colocalization image.

References:

(1) Manders, Verbeek, Aten, 1993. Measurement of colocalization of objects in dual-color confocal images. *J. Microscopy* 169:375-382.

(2) Costes, Daelemans, Cho, Dobbin, Pavlakis, Lockett. Automatic and Quantitative Measurement of Protein-Protein Colocalization in Live Cells. *Biophysical Journal*: 86: June 2004: 3993-4003.

(3) Model, Brukhardt, 2001. A Standard for Calibration and Shading Correction of a Fluorescence Microscope, *Cytometry* 44:309-316.

(4) Zwier, Oomen. Brocks, Jalink, Brakenhoff: Quantitative image correction and calibration for confocal fluorescence microscopy using thin reference layers and SIPchart-based calibration procedures, *Journal of Microscopy*, 231, 1, 59-62, January 2008.