

Quantitative Analysis of 3-D Colocalization and  
2-D Correlation of Two-Color Fluorescence Images

## **Introduction**

The trend in neuroscience and other fields of biology over the past 10 years to rely upon images produced by advanced microscopy techniques has not been met with an availability of tools for the careful quantitation of such images. All too frequently, an image is shown in a publication as being a ‘representative example’ of a reported effect, yet no statistical summary accompanies the image. In certain cases, the effect is sufficiently obvious that an analysis of the image may not provide additional information. However, given the tremendous variability in biological systems, it is very important that the beauty and stunning detail captured at the subcellular level with today’s microscopes is substantiated with a rigorous analysis of properties of interest within the images.

Of particular relevance in this regard is the analysis of the spatial co-distributions of two or more proteins within a cell as identified through fluorescence immunocytochemical techniques. In neurons, for example, it is frequently reported that a given protein, which may exhibit a punctate or particulate distribution, tends to colocalize with markers of synaptic sites. The clustering of a particular protein at a synapse could be biologically meaningful since, presumably, in order to affect the function of the synapse in some way, the protein should be located in close proximity to synaptic sites. Furthermore, the distribution of proteins may change as a function of the state of the neuron (e.g. the well-characterized movement of AMPA receptors into and out of the membrane in response to synaptic plasticity), and it is useful to understand quantitatively how the colocalization properties of sets of proteins may differ across a range of experimental conditions.

A number of commercially and freely available programs provide the ability to analyze colocalization in two dimensions, employing a simple scatter plot of the pixel intensities of two-color immunostained images, with the colocalization index represented as a regression fit to the scatter plot. This method of quantifying colocalization is certainly a step in the right direction, but potentially interesting spatial information in the original images is lost in the final output of this type of analysis. Also, with the prevalence of confocal and multi-photon microscopy, it has become necessary to analyze colocalization in three dimensions rather than two. One software package exists with such capabilities (Volocity, from Improvision), although at a cost of more than \$10,000.00, it is not practical for many users. Furthermore, at last assessment, the Volocity program was very slow to perform a colocalization analysis on images of dimensions 512x512x10, a routine image size. In order to facilitate the analysis of 3-D colocalization, and to make the technology accessible to all researchers, I have developed a program using MATLAB (The Mathworks), which has been compiled as a stand-alone application to run on any Microsoft Windows-based computer. While the program is very useful for quantifying the colocalization properties of protein particles in two-color fluorescence images, there are a number of limitations to the program in this early stage. The MATLAB source code, which can be compiled using MATLAB or run as a graphical user interface from within the MATLAB framework, is included in Appendix A. The program, including an assessment of its limitations, is discussed in the 3DParticles section below.

In addition to colocalization, clusters of proteins may exhibit biologically relevant spatial correlation. Even if the signals for two proteins never colocalize, it is interesting in

many instances to characterize and quantify the spatial codistributions of the particles. This case is illustrated by considering the ultrastructural detail of dendrites on central mammalian neurons, with the main dendritic shaft containing small protrusions called spines onto which a majority of excitatory synapses are made (see, for example, Figure 1.2). Due to rapid rates of active transport in neurons, measured at up to 5  $\mu\text{m/s}$  (Brown, 2003), it is not necessary for a protein to be located within the spine in order to affect synaptic function on a rapid timescale. Indeed, the movement of proteins and cellular machinery such as ribosomes to the base of dendritic spines has been postulated as a molecular mechanism underlying synaptic plasticity (Ostroff et al., 2002; Shi et al., 2001; Shi et al., 1999; Steward and Levy, 1982). The quantitation of the 2-D spatial correlation of synaptic markers and postsynaptic proteins of interest, however, is lacking.

We have developed a strategy for a normalized 2-D cross-correlation analysis that greatly facilitates the quantitative description of the spatial codistributions of clusters of proteins within linear segments of dendrites and axons. Because there exists some random probability that two punctate signals exhibit spatial correlation, our algorithm estimates the random correlation that would be expected in a given image pair, and then subtracts the magnitude of that random correlation from the raw 2-D cross-correlation signal. The result is a 2-D correlation output that quantitatively describes the spatial coincidence of protein clusters in the cell, normalized to the correlation that might be randomly present simply due to the abundance of “on” signal in each thresholded image. The algorithm, as well as examples of the output of the Matlab script on simple test images, are illustrated in the 2DXcorr section below.

### 3-D Particles Analysis

3DParticles is a simple program for the quantitative analysis of colocalization in three-dimensional confocal or two-photon micrographs of fluorescently-immunolabeled biological specimens. A number of assumptions about the images must be true if the results of the program are to be meaningful. These considerations are to be taken into account for any high-resolution immunofluorescence microscopy experiments, and have been discussed in detail elsewhere, but some mention of the more relevant points is in order. Of primary concern is the optical thickness of each section for each color in the sample to be analyzed. Due to the differential chromatic aberrations from different wavelengths of fluorophores used in immunolabeling experiments (typically red and green fluorescent markers are used), the thickness of an optical section for each color will vary for a given confocal pinhole aperture. On systems with a continuously variable confocal pinhole, this can easily be circumvented by selecting an appropriate pinhole for each channel that results in optical slices of equal thickness. With some older confocal systems, however, it may be necessary to pre-process the images in order to negate any pixel shift before running the colocalization analysis. In general, if the user is comparing between experimental and control groups, such chromatic aberrations should average out between groups, but it is always best to ensure that the images are, as accurately as possible with light microscopy, representing the true positions of the proteins being analyzed. A generalized method for detecting and correcting chromatic aberrations in lateral as well as axial planes has been previously published (Kozubek and Matula, 2000), and will not be discussed further here.

Another issue of interest with regards to the fidelity of colocalization analysis in general is that of signal cross-talk, or bleedthrough. Because the excitation and emission spectra of fluorophores used in immunocytochemistry typically have some degree of overlap, it is important that the user select fluorophores, filter sets, and image acquisition parameters that minimize or eliminate signal bleedthrough from one channel to another. Using advanced microscopy systems such as the Zeiss Meta detector, or the Leica AOBS system, it is possible to completely separate the fluorophores based on their spectral characteristics. However, even in the absence of such high-tech spectral detectors, the experimenter should take care that appropriate filter sets and detector settings are used to minimize bleedthrough, which can create spurious colocalization signal even if the proteins in the cell never occupy the same general space. In any case, the amount of bleedthrough should be measured, and the amount of signal in each channel that results from such bleedthrough must be subtracted from the image, or a threshold must be set that does not include pixel intensity values within the range of the cross-talk.

One potential limitation of the 3DParticles program is the need to threshold the images in order to run the analysis. While this is an inevitable consequence of the morphological analysis performed by the algorithm, issues of thresholding are to be taken seriously. Morphological image analysis, almost by necessity, involves thresholding the images in order to define particles or other objects of interest. While it is possible to perform such segmentation without the use of direct thresholding techniques (for example, using the watershed segmentation algorithm), it is standard practice to convert a grayscale image into a thresholded binary image as the first step in many analysis routines. In the

best case scenario, thresholding will simply remove the noise, leaving behind the real data of interest in the image. In most instances, however, what is considered to be real is subjectively determined. While it may be true that the human eye is an exquisitely tuned machine for picking out the relevant objects of data in a noisy image, it is unsatisfying to use such subjective criteria in an analysis algorithm as "it looks good." Not only does it lack rigor, but it requires more user interaction with the analysis procedure, which ultimately increases the probability that user bias will interfere with the result. This is likely one reason for the vast body of literature on the subject of image thresholding in the computer science and electrical engineering communities.

Fortunately, there is a widely accepted method for analytically determining the optimal threshold in a grayscale image. This is an iterative technique known as Otsu's Method (Otsu, 1979), and is the basis for the initial threshold settings applied to the images when they are first loaded into the 3DParticles. While this is certainly not the only method for image segmentation, it is built in to MATLAB, so it is convenient. An interesting and highly-educational several hours may be spent on the internet by searching Google with the terms:

"image thresholding" otsu

Although not perfect, Otsu's method does at least provide an analytical starting point for defining image threshold cutoffs. Some limitations to the method involve the fact that it assumes a bimodal pixel intensity distribution, with one mode being the noise and the other being the data of interest. In images with high signal to noise ratio, such as tends

to be the case in confocal microscopy where the background is frequently equal to zero, the bimodal intensity assumption may not hold. This is why the 3DParticles program is equipped with a means of altering the threshold values (i.e., the sliders beneath each axis). There is still a subjective component to the routine as written, but one can simply chose to use the automatically generated threshold settings if an entire lack of subjectivity is desired. At the very least, this method provides some consistency in the threshold determination.

### **Running 3DParticles**

Launching the program from within the Matlab environment opens a program window as shown in Figure 4.1A. The user loads 3-D image stacks by clicking on the “Load 3D Images” button. The program assumes the image stacks are saved as TIFF files. Once both images are loaded, three-dimensional surface renderings of each image are plotted into the two axes, with the top axis showing the first image that was loaded. As a starting point, the images are initially plotted using Otsu’s method to generate a threshold. If desired, the user may then change the threshold for each image using the sliders beneath each image axis, or by entering a value between zero and one in the box to the left of each slider. For a quick look at a two-dimensional representation of the images, which may be useful in determining an appropriate threshold setting, the user may click on the “2D Projections” button.

The image in each axis may be rotated, zoomed, and panned across the screen using the left, right, and middle mouse buttons, respectively. This can prove to be useful when there are particles very close to one another, and complete separation is desired. By



rotating and zooming into the images, it is possible to set a threshold that effectively separates closely packed particles while maintaining as much image data as possible. If, after rotating the image, the orientation of the image becomes unclear, the default position can be reset with the “Reset Axes” button. The reset button functions only to restore the axes to the default 3-D position, and does not affect the zoom level.

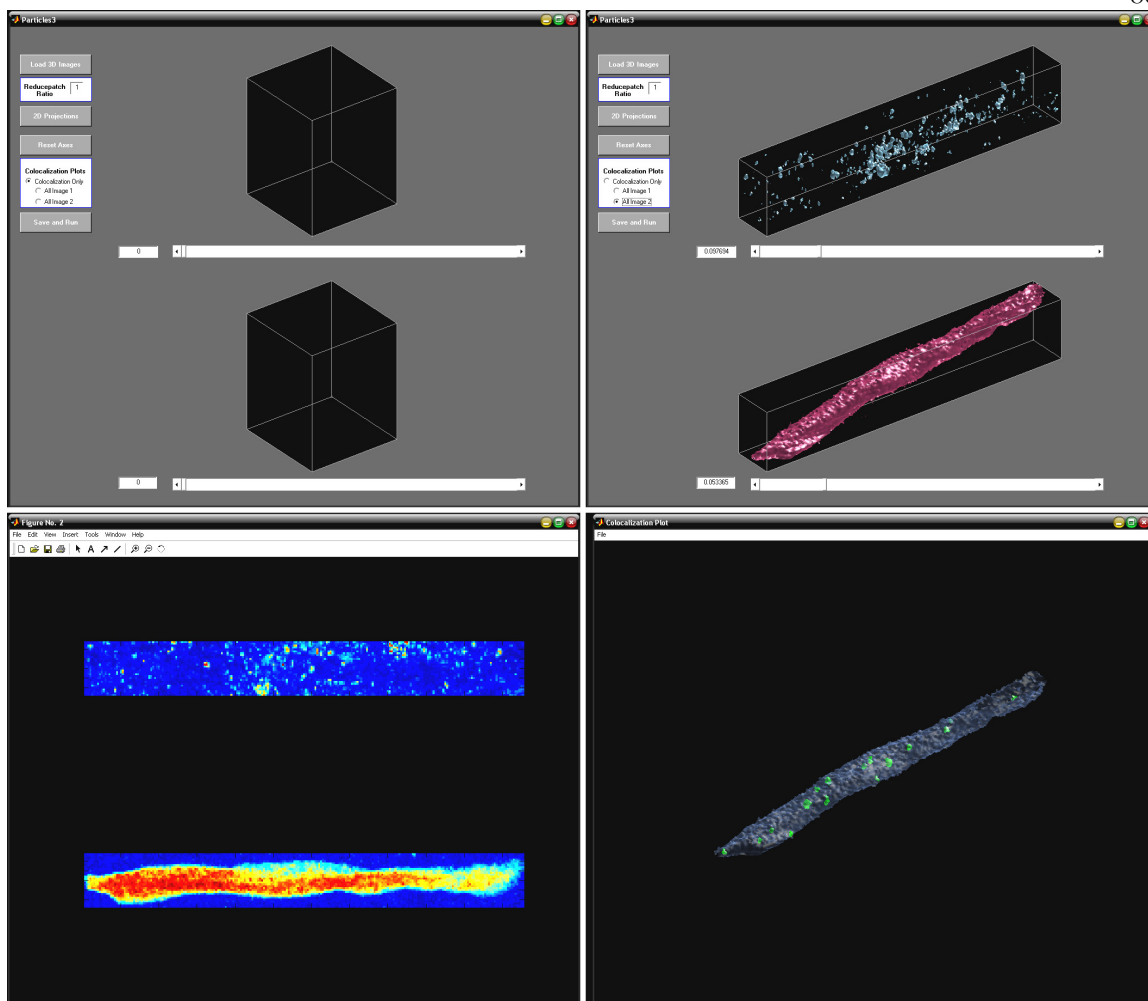
Selection of three different colocalization plots is possible by selecting the desired output using the various radio buttons in the “Colocalization Plots” panel. The default setting, Colocalization Only, generates a plot showing only the particles in each image that are overlapping with one another. The other two options allow the user to plot all of the particles of one image or the other, along with the colocalized particles from the other image. Such a plot is useful in the case where a neuronal process has been filled with GFP or some other soluble dye, and the user is interested in only visualizing the particles from the other channel that are colocalized with the filled process. Once the appropriate plot style is selected, the routine is completed by clicking the “Save and Run” button, which results in the display of some progress indicator windows followed by the selected colocalization plot.

In addition to the graphical output as shown in Figure 4.1, 3DParticles generates a tab-delimited text file named “rowdat.txt” located in the directory where the original images were stored. This file contains 25 variables that describe a number of the properties of particles in each image, as well as some information about the colocalization of those particles. If the analysis is run multiple times on a given image pair, or if there are

multiple image pairs in any given directory, the results of each analysis will be appended to the end of the rowdat.txt file. Below is a list of the 25 variables, along with brief definitions of each:

1. TH1: the threshold used for image 1
2. TH2: the threshold used for image 2
3. MeanV1: average particle volume for image 1
4. TotalV1: total amount of signal in the thresholded version of image 1
5. MeanPix1: average pixel intensity of particles in image 1
6. TotalPix1: total pixel intensity of particles in image 1
7. NumP1: number of particles in image 1
8. MeanP1frac: average fractional volume of particles in image 1 contributing to overlap
9. MeanV2: average particle volume for image 2
10. TotalV2: total amount of signal in the thresholded version of image 2
11. MeanPix2: average pixel intensity of particles in image 2
12. TotalPix2: total pixel intensity of particles in image 2
13. NumP2: number of particles in image 2
14. MeanP2frac: average fractional volume of particles in image 2 contributing to overlap
15. MeanVOL: average volume of overlapping particles
16. TotalVOL: total amount of signal in the overlapping image
17. NumPOL: number of particles in the overlapping image
18. TotalPixOL1: total intensity from image1 of particles that overlap
19. MeanPixOL1: mean intensity (per particle) from image1 of particles that overlap
20. NumOLdivTot1:  $\text{NumPOL}/\text{NumP1}$
21. VOLdivTotV1: total overlap volume as fraction of total image 1 particle volumes
22. TotalPixOL2: total intensity from image2 of particles that overlap
23. MeanPixOL2: mean intensity (per particle) from image2 of particles that overlap
24. NumOLdivTot2:  $\text{NumPOL}/\text{NumP2}$
25. VOLdivTotV2: total overlap volume as fraction of total image 2 particle volumes

The output format of rowdat.txt is suitable for import into Matlab or a spreadsheet program such as Excel for further statistical analysis.



**Figure 4.1** Screenshots of the 3DParticles Application

The program window is shown immediately after launching the application (top, left), and after images are loaded and thresholds adjusted (top, right). Also shown are 2-D projections of the loaded images (bottom, left), and the colocalization image generated using the “all image 2” option (bottom, right).

## 2-D Cross-Correlation

In our previous work (see Figure 2.10), we used a normalized one-dimensional cross-correlation routine to argue that GFP hotspots (or puncta) seen in the dendrites of cultured hippocampal neurons were correlated with the presence of synaptic markers and ribosomal proteins. This analysis bolstered our claim that the GFP was locally synthesized in the dendrites, near synaptic sites. Given the potential loss of information in taking the mean pixel intensity across the width of the dendrites, for example, due to the fact that some regions of the dendrite may have a synapse on either side while others have only a single synapse, we devised a routine to analyze the correlation in 2-D space.

The amount of correlation detected between two images using a standard two-dimensional cross-correlation algorithm is a function of the spatial distributions of particles within each image, as well as the total amount of signal detectable in each image. The effect of signal abundance on the expected correlation is most obvious at the upper limit: if one image or the other is completely filled with signal, there will be perfect 2-D correlation (as well as colocalization) between the images. Even in real biological samples, where the signal of interest frequently occupies less than one third of the image area, there is some probability that any correlation measured between a pair of images is simply due to the amount of space filled by the signal in each image. Here we describe an algorithm that normalizes the 2-D correlation by removing the average correlation detected when the particles from each image are randomly redistributed within the image frame. The result gives an estimate of the likelihood that the correlation detected is due to randomness, or if it is in fact due to some underlying process that is potentially more meaningful.

## Methodology

Here I describe, in detail, the steps involved in the 2DXcorr routine. Because the MATLAB code is included in Appendix B, the discussion here will be limited to a more intuitive description of the process. The specifics of each step, and the computation involved, are clearly identified in the comments of the code.

The routine begins by loading the images into the MATLAB workspace. The images are assumed to be in the TIFF format. It is also assumed that the images represent two independent signals (red and green fluorescent signal derived from two different proteins, for example) from a single, horizontal, linear dendritic segment. While the routine is easily generalized to cases in which these constraints are not met, the linearization of a single neuronal process greatly facilitates the analysis for two important reasons: first, there is no ambiguity about the orientation of the principal x and y axes, and second, much less memory is required for images pre-processed in this manner.

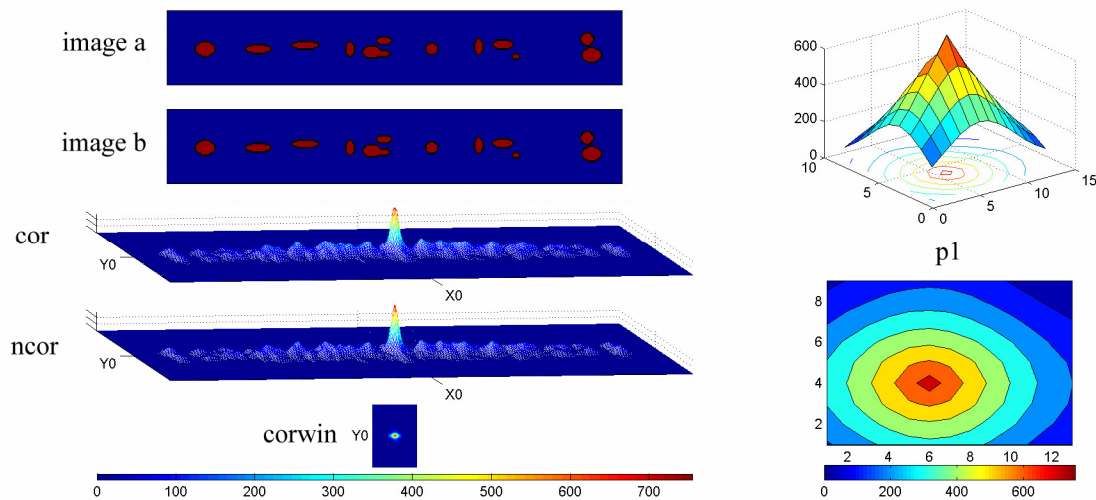
In order to fully understand the 2DXcorr algorithm, it is necessary to first review the general 2-D correlation function. Briefly, the 2-D correlation in the frequency domain is computed in MATLAB using the following code:

```
Fa = fft2(rot90(a,2));  
Fb = fft2(b);  
cor = real(iff2(Fa .* Fb));
```

where the 2-D Fourier transform (fft2) is computed on each image, **a** (rotated 180 degrees) and **b**. The final output, **cor**, is equal to the real component of the inverse 2-dimensional

FFT of the product of the transformed images. If the images are identical, the result is a special case of correlation known as the autocorrelation, shown graphically in Figure 4.2.

The autocorrelation determines the maximum possible correlation for a given image.



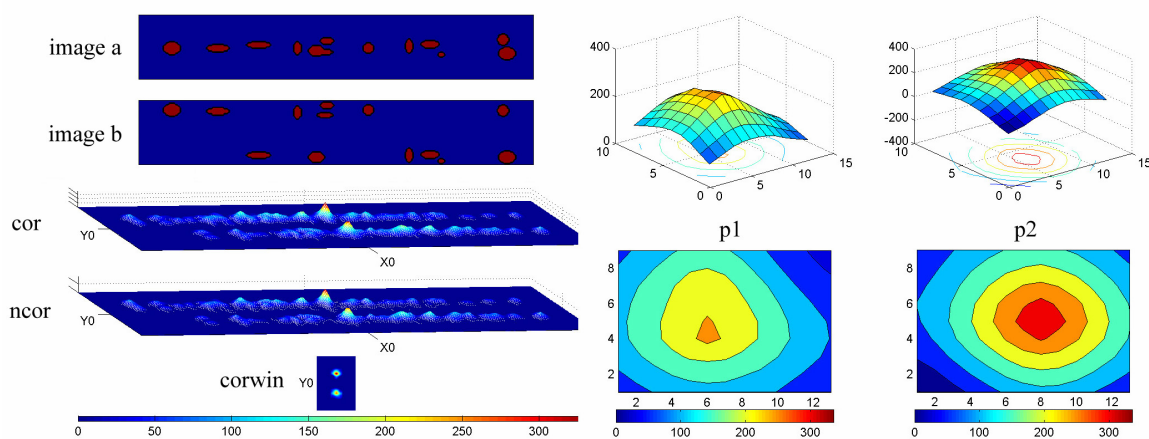
**Figure 4.2** The Autocorrelation Function

Shown at left are two identical simulated images (*image a* and *image b*), the raw 2-D autocorrelation output (*cor*), and the correlation after subtraction of the randomized correlations (*ncor*, see text). The correlation window (*corwin*) around the  $X=0$  lag position is also shown, and the correlation peak (*p1*) extracted from *corwin* is plotted as a surface (top, right) and a filled contour plot (bottom, right).

As seen in Figure 4.2, plotting the 2-D autocorrelation output as a surface generates a large peak in the middle of the 2-D correlation space. If the dimensions of the original images are defined as  $(x,y)$ , the dimensions of the correlation space are equal to  $((2*x)-1,(2*y)-1)$ . Intuitively, this is because the correlation operation considers one image as a template, and then shifts the other image in 2-D space over the template, measuring the correlation at each point. The correlation axes are expressed in units of *lag*, where  $(X_0,Y_0)$  in the correlation space is the midpoint of each axis. The maximum-

amplitude peak of the autocorrelation, located at  $(X_0, Y_0)$ , indicates that the images do not need to be shifted in either dimension in order to generate the maximum correlation.

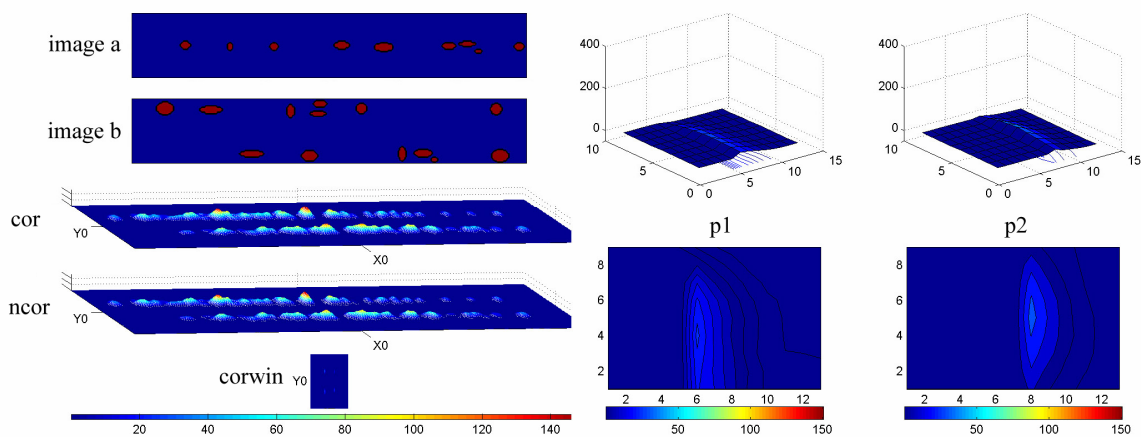
In the case where the images are not identical, although very highly correlated, the correlation function output becomes more complicated. Figure 4.3 shows the case where the test correlation images have been generated with the intent of emulating signal in the shaft of a dendrite (image a), and signal marking synapses in spines (image b). While the particles in each image are identical, the fact that the particles in image b are distributed away from the y-axis midline results in a correlation output that differs from the autocorrelation case. Specifically, given that the particles in image b are distributed on both sides of the y-axis midline, with the particles in image a largely confined to the midline, the 2-D correlation results in two distinct peaks, one on either side of the  $Y_0$  lag position.



**Figure 4.3** Highly Correlated Test Images

Shown are simulated test images, with *image a* representative of protein signal in the shaft of a dendrite, and *image b* designed to emulate protein signal at synapses on spines. Because particles in *image b* lie on either side of particles in *image a* (in the y-axis), the correlation function (cor) results in two peaks: *p1* at  $(X_0, -Y)$  and *p2* at  $(X_0, +Y)$ .

To illustrate the importance of the correlation window (*corwin*), Figure 4.4 shows an example where the particles are uncorrelated, as defined by our criteria. Because we are interested only in correlating particles in the dendritic shaft with synaptic signal that is nearby, we limit our analysis of the correlation output to a small window centered around  $(X_0, Y_0)$ . We have chosen a method that defines *corwin* based on the measured linear distances between adjacent particles. As specified in the code found in Appendix B, the dimensions of *corwin* in the X-lag axis are limited to  $2 \times (\text{mean}(d) + \text{std}(d))$ , where  $d$  is the linear distance between adjacent particle centroids, and  $\text{std}$  is the standard deviation of that mean. For example, if the mean distance were 10 pixels, with a standard deviation of 3, *corwin* would be set at  $(X_0 - 13, X_0 + 13)$ . Because the images are preprocessed for this routine such that only a single neuronal process occupies the image, the Y-lag window is self-limiting.

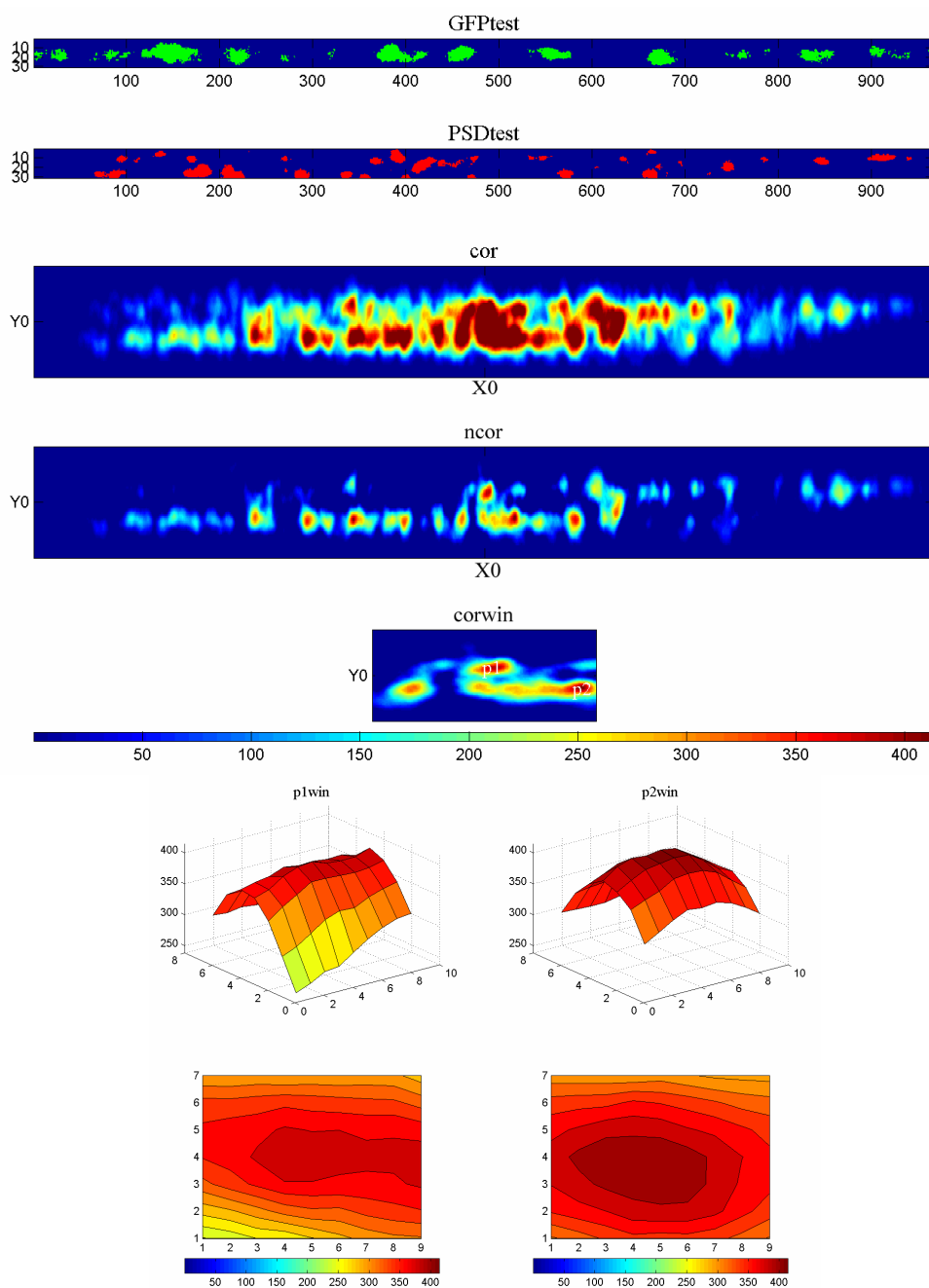


**Figure 4.4** Uncorrelated Test Images

Shown are two images (*image a* and *image b*), designed to show the effect of limiting the analysis to *corwin*. In this case, the images are uncorrelated as defined by the algorithm, because each particle in *image a* is directly in between particles in *image b*. The result is a set of correlation peaks that fall outside the dimensions of the *corwin* X-lag.

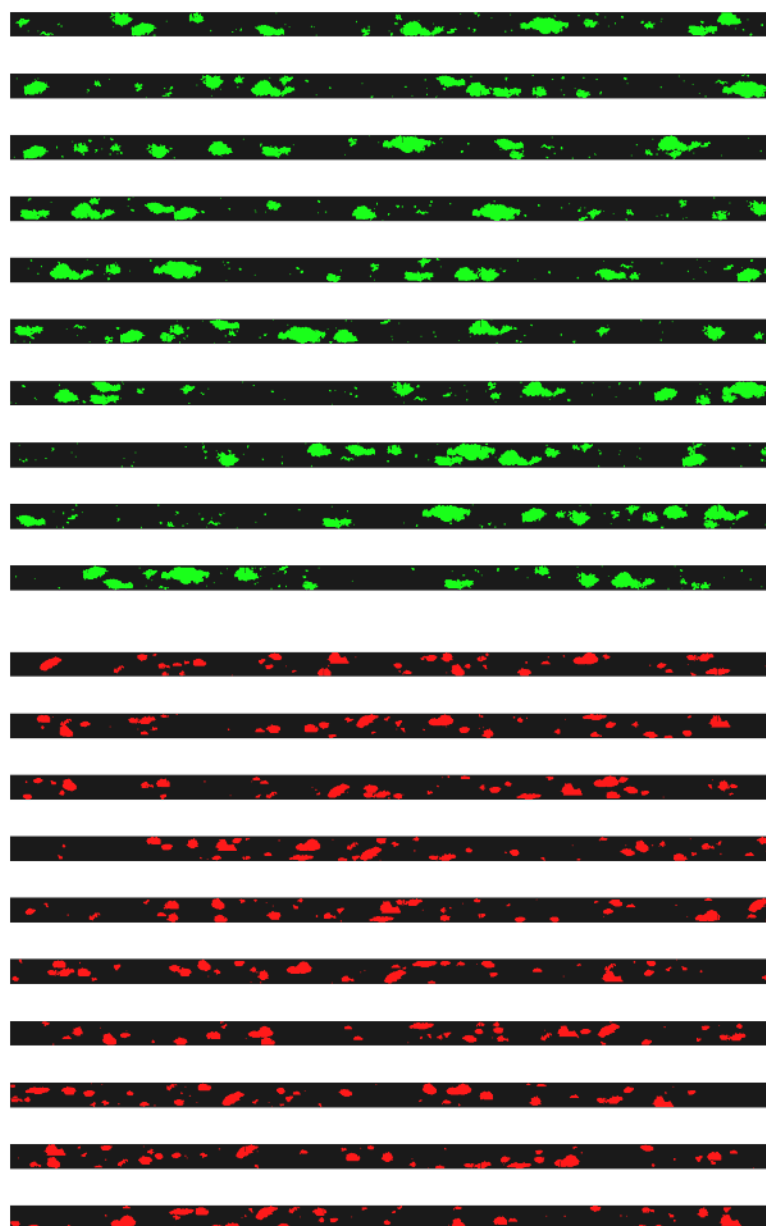


The results of an analysis on real image data are shown in Figure 4.5. The images in this case are the same as those used in Figure 1.10, where *image a* is the GFP signal, and *image b* shows synaptic locations as determined by PSD-95 staining. As a reference, a sample of the randomized versions of each image is plotted in Figure 4.6. From analysis of the 2DXcorr output, it is clear that a substantial portion of the raw correlation (*cor*) is attributable to the abundance of the two signals, which can be seen in the amount of correlation signal that has been subtracted as a result of the normalization routine (cf. *cor* to *ncor*). However, within the specified window of the normalized correlation (*corwin*), there are two significant peaks remaining (*p1win*, *p2win*). The sum of the two peaks, divided by the autocorrelation of the GFP image, gives a correlation score of 0.40. While this number by itself carries little meaning, further analysis reveals that of the randomized correlations, analysis from repeated trials of the correlation that ranks as the upper limit of the 95<sup>th</sup> percentile produces correlation scores ranging from 0.08 to 0.12.



**Figure 4.5** *myr*-dGFP / PSD-95 Cross-Correlation

Brief visual inspection shows that a large amount of the correlation signal (*cor*) has been subtracted away in the process of generating the normalized correlation (*ncor*). Integration of the two correlation peaks (*p1win* and *p2win*) gives a correlation score of 0.40. Refer to text for further details.



**Figure 4.6** Examples of Randomized GFP Puncta (top) and PSD-95 Puncta (bottom).

## Conclusions

The routines presented here, for quantitative analysis of 3-D colocalization and 2-D correlation, provide a simple, intuitive means of measuring and summarizing the most frequently-reported properties of immunostained particles. These programs are a good starting point for such analyses, although some refinement will improve the algorithms. Currently, the need to threshold the images is the greatest limitation. Developing methods that permit similar analyses while maintaining the full bit-depth of the original images will dramatically improve the robustness of these algorithms. Such modifications will further reduce the potential for user bias in the analysis process. Another benefit would be to allow for direct comparisons of colocalized and spatially correlated particles of similar intensities between the two images. Such comparisons may provide further information about the dynamics of the system.

The 2DXcorr program will also benefit from more a more advanced statistical analysis of the distributions of particles in the real versus the random data sets. For example, with an appropriately large set of randomized images (generating on the order of  $10^5$  randomized correlations), it may be possible to assign significance values to individual correlations using a statistical measure that computes the distance between the actual correlation and the distribution of random correlations. This will, in turn, further facilitate the between-group comparisons required in experiments using this routine.