

How it works

Thursday, January 13, 2022 3:33 PM

General overview:

recon_ty_v2.m

This is the main code to which I've made some modifications. Originally, it was a product of TY.

This function creates three panels of a single image's specified fluorescent channel. The first image is a construction of just the image with a max projection from all z-slices. The second image is a reconstruction from gaussian spatzcell objects. The final image is also a reconstruction from gaussian spatzcell objects, but with the whole spot area represented as an ellipse shaded a solid color.

recon_bestcv.m

This is similar to the code above. The only difference here is in the first panel. Specifically, this will load in a single z-slice *for each cell*. The z-slice will be determined by the highest CV slice for each cell. Another edit is a new function parameter, **cnum_focus**. This holds the channel where we will be doing the process of finding the best slice. A use case would be where the fluorescent channel is difficult in terms of the highest CV returns a bad image, so the phase channel is used instead.

recon_bestcv_max3proj.m

This inherits all the changes from **recon_bestcv.m** with 1 change that brings it closer to the original. In this version, after finding the best z-slice a max projection is then taken with the z slice immediately above and immediately below the best. This helps to reconstruct spots who exist in different z slices than the best.

recon_bestcv_max3_proj_norm.m

This code is still a work in progress, but perhaps a future user will improve on its idea. The change in this code is to normalize the image before taking the 3 slice max projection. The idea behind this being that if there were any bleaching effects working through the slices, it would not affect the projection.

reconstructimage_v7_w_channel.m

This creates only a single panel which hosts the 1st image from **recon_bestcv.m**, using the fluorescent channel as the focus channel, along with red circles which denote the centers of the spatzcell objects.

reconstructimage_v7_w_phase.m

This creates only a single panel which hosts the 1st image from **recon_bestcv.m**, using the phase channel as the focus channel, along with red circles which denote the centers of the spatzcell objects.

Detailed overview:

Because the code for all of these files are so similar to each other, I'll simply describe the process of one,

recon_bestcv_max3proj.m

1

First, we load in the cell masks and create the perimeters of all cells. Once this is done we create the figure and the frame of the three panels to hold the reconstructions.

2

Looking at the first panel, we must get the max projection. To do so we do the familiar process of loading in cell masks, the fluorescent channel images, and the focus channel images. From here we loop over all z-slices and all cells in the focus image and calculate the CV for each cell in that slice. Once this is done, we again loop over all z-slices and all cells in the fluorescent image. If in the current cell this z slice holds the highest CV, we take the highest pixel value, for each pixel in the cell's area, across the fluorescent images immediately above to immediately below this slice.

This is then plotted in the first panel

3

Looking at the second panel, we load in **peakdata** from the spatzcell data and use their spot positions, sizes, and intensities to plot them as gaussians on an empty image.

4

For the last panel we combine what we do in step 2 and step 3. We get the max projection like in step 2, and we then load in the **peakdata** like in step 3. However, rather than creating a gaussian spot with decaying intensity, we draw an ellipse of solid blue color and plot it on the max projection.

Pseudocode:

%% STEP 1

```
LcFull = Load in cell segmentation (...);
ALL_cellPerim = Get cell perimeter (LcFull);
N = Get number of cells (LcFull);
Figure();
ha = subplot(1, 3);
```

%% STEP 2

```
axes(ha(1));
MaxP = zeros(size(LcFull));
all_cv = zero matrix (z_range rows, Cell number columns);
Loop over each z_slice {

    FluorImage = Load in fluor image (...);
    Loop over each cell {

        all_cv(z_slice, cell number) = CV(FluorImage(current cell, current z_slice));
    } End loop over each cell
} End Loop over each z_slice
Loop over each z_slice {

    FluorImage_previous = Load in fluor image (...);
    FluorImage_current = Load in fluor image (...);
    FluorImage_next = Load in fluor image (...);
    Loop over each cell {

        If max_index(all_cv) == current z_slice {

            MaxP = MaxP + max(FluorImage_previous, FluorImage_current, FluorImage_next) * (LcFull == current cell)
        } End if best z_slice
    } End loop over each cell
} End Loop over each z_slice
imshow(MaxP, LUTS);
spy(ALL_cellPerim);
```

%% STEP 3

```
axes(ha(2));
spy(ALL_cellPerim);
BG_in = median(MaxP(logical(LcFull)));
rec_img = zeros(size(LcFull));
rec_img(logical(LcFull)) = BG_in;
Load in peakdata (...);
Loop over all real spots in peakdata {
```

```

    x_rg = x pixels around spot center in predetermined distance to gain intensity (spot_center);
    y_rg = y pixels around spot center in predetermined distance to gain intensity (spot_center);
    fluo_ = pixel intensity to add from spot's gaussian intensity distribution (x_rg, y_rg, spot_center);
    rec_img = rec_img + fluo_;
} End loop over all real spots in peakdata
imshow(rec_img, LUTS);

```

%% STEP 4

```

axes(ha(1));
MaxP = zeros(size(LcFull));
all_cv = zero matrix (z_range rows, Cell number columns);
Loop over each z_slice {

    FluorImage = Load in fluor image (...);
    Loop over each cell {

        all_cv(z_slice, cell number) = CV(FluorImage(current cell, current z_slice));
    } End loop over each cell
} End Loop over each z_slice
Loop over each z_slice {

    FluorImage_previous = Load in fluor image (...);
    FluorImage_current = Load in fluor image (...);
    FluorImage_next = Load in fluor image (...);
    Loop over each cell {

        If max_index(all_cv) == current z_slice {

            MaxP = MaxP + max(FluorImage_previous, FluorImage_current, FluorImage_next) * (LcFull == current cell)
        } End if best z_slice
    } End loop over each cell
} End Loop over each z_slice
imshow(MaxP, LUTS);
Load in peakdata (...);
Loop over all real spots in peakdata {

    If this spot area isn't extremely large {

        DrawEllipse(spot_center, spot_axis, spot_angle);
    } End if not large spot area
} End loop over all real spots in peakdata
spy(ALL_cellPerim);

```