

How it works

Thursday, January 13, 2022 3:35 PM

General overview:

This algorithm works to link together spatial data of two different spatzcell data channels and of one cluster data channel.

This code will loop through all samples and their images and load in each cell's 'LcFull' segmentation data and spatzcell 'peakdata'. Then, it loads in the cluster channel fluorescent image and finds the best z-slice to load in for each cell. After this, it loads in the cluster data from cluster recognition. Now that all the fluorescent data is loaded in, it loops over each spatzcell channel to ask questions as to its spatial relationship between the other channels. In addition, for each spatzcell channel a fake version of a spot is run where the spot's location is randomly chosen in the same cell.

Detailed overview:

Step 1

The algorithm will be run in a loop over each image and looped over all samples. We will load in cell masks, **peakdata** from the two spatzcell channels while avoiding spot data outside a cell and imaginary spot data, and the cluster masks from cluster recognition.

Step 2

For the cluster's fluorescent channel, we wish to have a single matrix as the working image rather than a matrix for each z-slice. For this, we loop over each z-slice for this image. Inside of this z-slice loop we find the CV of all of the cells in the fluorescent channel of interest. This is so that we can find the best z-slice for each cell. Now we begin to construct our image by looping over each z-slice and over all of the cells in the fluorescent channel of interest. If this slice holds the highest CV of a cell then we load in the fluorescent image of the cell to our single image matrix.

Step 3

Now, for a single spatzcell data channel we loop over each spot in the image twice: once for the spot's real location and once for the fake location, which is randomly placed in a coordinate inside the cell. In each loop we ask if the spot is in the low threshold cluster and how far from the boundary of the cluster it is. We do this again for the high cluster.

Step 4

Then, we turn to the other spatzcell channel, dubbed channel 2 now, and asks what the closest distance is between the current spot and a spot from channel 2. It also records some data of that spot from channel 2. We also do another version of this question and ask the total number of spots from channel 2 in a predetermined radius from our spot. We also record the total intensity of the channel 2 spots in this area. This is done for two different radii.

Step 5

Lastly, we calculate the total cluster fluorescence around the current spot in three different radii. We also record the concentration of the fluorescence in this area.

Step 6

We now loop over channel 2's spots doing the same exact steps including the loop for fake spots and the distance to another channel 2 spot.

Pseudocode:

```
Initialize_all_params (...);
Loop over samples {

    Loop over images within samples {

        Display time taken and current progress ();
```

%% STEP 1

```
cellmasks = Load in cell segmentation (...);
Channel1 = Load in spatzc cell channel1 peakdata (...);
peakdata_channel1 = remove spots outside cell and imaginary spots (Channel1);
Channel2 = Load in spatzc cell channel2 peakdata (...);
peakdata_channel2 = remove spots outside cell and imaginary spots (Channel2);
all_high_cluster_mask = load high threshold cluster data (...);
all_low_cluster_mask = load low threshold cluster data (...);
```

%% STEP 2

```
All_cv = zero matrix (z_range rows, Cell number columns);
max_fluor = zero_matrix(dimensions(LcFull));
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 = Load in fluor image (...);
    Loop over each cell {

        If max_index(all_cv) == current z_slice {

            max_fluor = max_fluor + FluorImage * (LcFull == currentcell);
        } End if best z_slice
    } End loop over each cell
} End Loop over each z_slice
```

```
Loop over spatzc cell channel 1 spots {
```

```
    Display current progress ();
```

%% STEP 3

```
Loop over fake or real positions {

    If dealing with fake positions {

        cell_coor = (cellmasks == current cell number);
        coor = random coordinate from cell_coor (cell_coor);
        x = get x from coordinate (coor);
        y = get y from coordinate (coor);
    } Else {

        x = get x from channel1 (peakdata_channel1);
        y = get y from channel1 (peakdata_channel1);
    } End if dealing with fake positions
    low_cluster_mask = all_low_cluster_mask * (cellmask == current cell number);
    If spot inside low cluster (x, y, low_cluster_mask) {

        Inside = host low cluster number (x, y, low_cluster_mask);
    } Else {
```

```

        Inside = -1;
    } End if spot inside low cluster
    If inside > 0 {

        perim = get perim of low cluster (low_cluster_mask, inside);
        mini = min(distance(x, y, perim));
    } Else {

        perim = get perim of all clusters in the cell (low_cluster_mask);
        mini = min (distance(x, y, perim));
        closest_low_clusters_number = min index(distance(x, y, perim));
        Inside = closest_low_clusters_number * -1;
    } End if inside == 1
    Save data(inside, mini);

```

Repeat from "If dealing with fake positions {" but for high cluster data instead of low cluster data.

%% STEP 4

```

mini = min (distance(x, y, peakdata_channel2));
spot = min index(distance(x, y, peakdata_channel2));
int = get spot intensity(peakdata_channel2, spot);
rolling_int = sum of intensities in radius(x, y, peakdata_channel2, radii([1, 2, 3]));
rolling_spotnum = number of spots in radius(x, y, peakdata_channel2, radii([1, 2, 3]));

Save data(current cell number, current frame number, channel1 spot number, channel 1 spot intensity);
Save data(mini, spot, int, rolling_int, rolling_spotnum);

```

%% STEP 5

```

Loop over radii {

    spot_mask = area(x, y, radii);
    channelcolor_fluor = total intensity in radi(max_fluor, spot_mask);

    Save data(channelcolor_fluor, channelcolor_fluor/area(spot_mask));
} End loop over radii

} End loop over fake or real positions
} Loop over spatzc cell channel 1 spots

```

%% STEP 6

```

Loop over spatzc cell channel 2 spots {

```

Do the contents of loop "Loop over spatzc cell channel 1 spots {" but for channel 2 spots and peakdata.

```

    } End loop over spatzc cell channel 2 spots
} End loop over images within samples
} End loop over samples

```