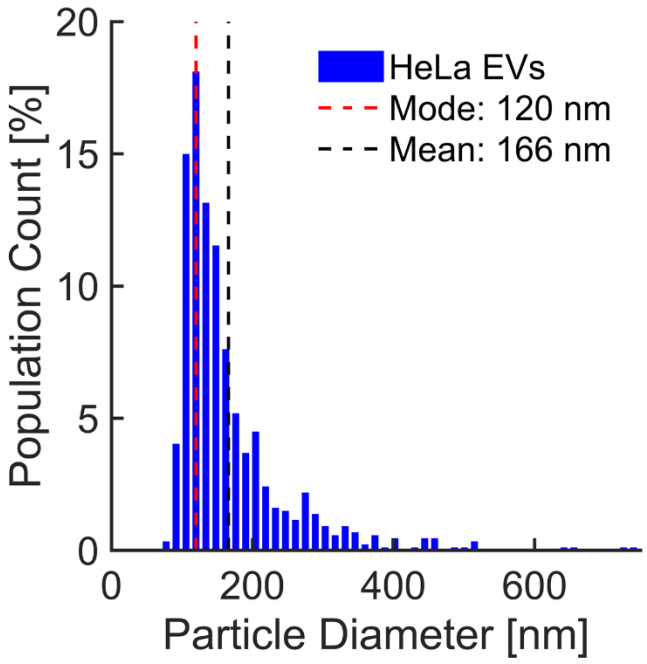
Electronic Supplementary Material (ESI) for Lab on a Chip.

Supporting Information:

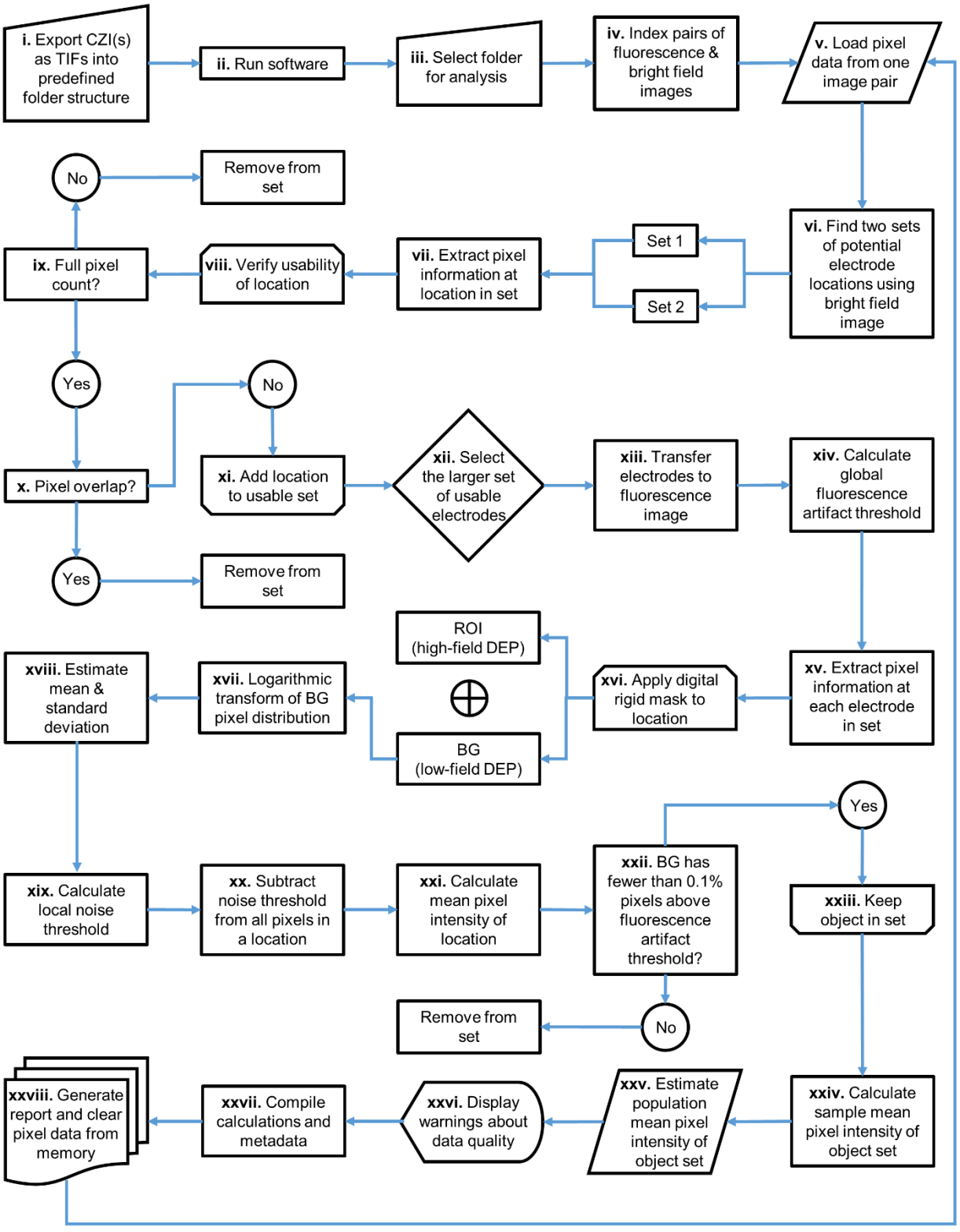
Automated Fluorescence Quantification of Extracellular Vesicles Collected from Blood Plasma using Dielectrophoresis

Kyle T. Gustafson,[1,2] Katherine T. Huynh,[1,2] Daniel Heineck,[1,2] Jesus Bueno,[1,2] Augusta Modestino,[1,2] Sejung Kim,[1,2] Austin Gower,[1] Randall Armstrong,[1] Carolyn E. Schutt,[1,2] Stuart D. Ibsen[1,2,*]

[1]Cancer Early Detection Advanced Research Center, Knight Cancer Institute, Oregon Health & Science University
[2]Department of Biomedical Engineering, School of Medicine, Oregon Health & Science University
*Corresponding author: ibsen@ohsu.edu

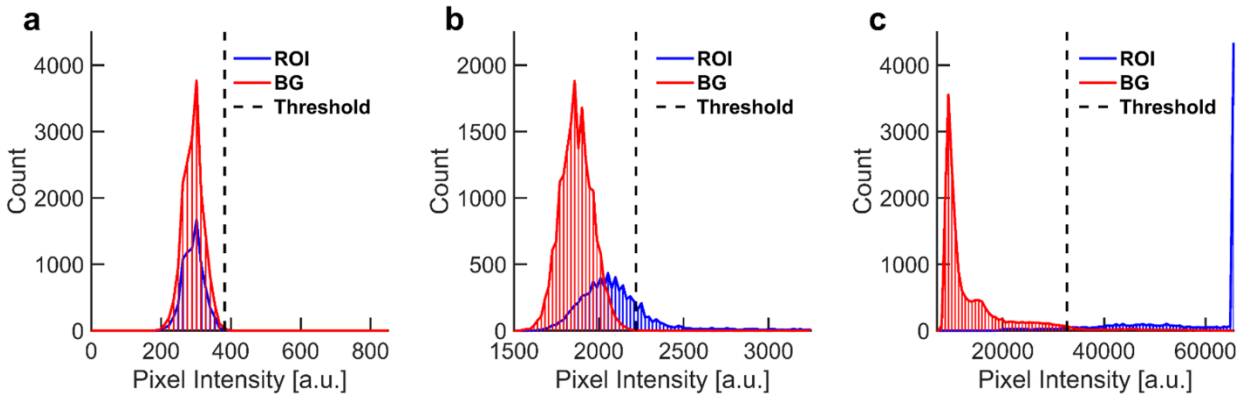


Supplemental Figure 1 – Size distribution of EVs collected, purified, and concentrated from HeLa cell culture media: A histogram showing the size distribution of a sample of EVs harvested from cell culture media as the percentage of the measured sample population versus particle diameter (mean particle diameter of 166 nm; n = 867 EVs). The mode particle diameter (120 nm) fell within the range of diameters associated with exosomes (50 – 150 nm), suggesting that these EVs were successfully enriched for exosomes. The concentration of EVs was measured to be approximately 4×10^{11} EVs/mL.



Supplemental Figure 2 – Technical flow diagram for automated fluorescence quantification: Flow diagram outlining the computational sequence of the custom algorithm for fluorescence image

quantification of NPs isolated *via* DEP. This flow diagram was referenced along with **Figure 7** in the Materials & Methods Section of the main text.



Supplemental Figure 3 – Poor data quality can cause overlap and saturation of pixel distributions: **a)** A histogram showing counts versus pixel intensity for a representative electrode in a low-exposure fluorescence image of EVs collected from a low-concentration sample using DEP. The ROI (blue) and BG (red) pixel distributions overlapped completely, indicating that the combination of concentration and exposure time were too low to yield discernible fluorescence patterns after collection from plasma with DEP. This observation was confirmed by a fluorescence intensity value of 0.01 a.u., which was well below the lower limit of the dynamic range (1 a.u.). Accordingly, the contribution from the ROI for the average of electrode in this image was approximately 30%, which was the theoretical minimum value indicating the total absence of fluorescence signal. **b)** A pixel histogram for a single electrode in a representative image of nanoparticle collection that yielded discernible fluorescence at the ROI. The ROI and BG pixel distributions overlapped partially with the right-hand tail of the ROI distribution extending beyond the noise threshold. The fluorescence intensity for the corresponding image was 211 a.u., within the bounds of the dynamic range. **c)** The ROI and BG pixel distributions were almost entirely non-overlapping. The ROI pixel distribution saturated at the right end, suggesting substantial pixel saturation. The pixel saturation percentage of the average electrode used to calculate the fluorescence intensity for this image was approximately 7%. The fluorescence intensity calculated for this image was 5787 a.u., which far exceeded the upper limit of the dynamic range (2000 a.u.) for automated analysis.

S1.1: Data file preparation for analysis

Image files were exported using ZEN Software and prepared for analysis using the following parameters: [File type: Tagged Image File Format (TIFF); Convert to 8 bit (unselected); Compression (None); Resize (100%); Original Data (selected); Shift Pixel (unselected) Apply Display Curve and Channel Color (unselected); Use channel names (selected); Use Full Set of Dimensions (selected); Export to (…\ANALYZE ME); Create folder (selected); Generate xml file (unselected); Generate zip file (unselected)]. Each CZI was exported as an "image folder" of fluorescence channel TIFs and one bright field TIF. Image folders were placed into corresponding "experiment folders" for organizational purposes. Experiment folders were then placed into a folder labeled "ANALYZE ME". The folder structure was as follows: ANALYZE ME → Experiment Folder(s) → Image Folder(s) → images formatted as uncompressed, unedited, greyscale TIFs with 16-bit depth. Images were acquired using a 14-bit camera. The maximum value for images acquired with a 16-bit resolution camera would be $(2^{16} – 1) = 65535$. The maximum pixel value of a 14-bit image stored in a 16-bit format was $(2^{14} – 1)*2^2 = 65532$, not 65535. Therefore, saturated pixels in our images had intensity values of 65532.

S1.2: MATLAB® Code

S1.2.1: Main Function (v7_7)

```
%% 2.2.1 Algorithm Input Consisted of Pairs of Fluorescence and Bright Field Images
% Figure 7a, Supplemental Figure 2i-v

clear, clc, close all, format long;
iptsetpref('ImshowInitialMagnification', 'fit');

disp('Select the folder to analyze.')
select_a = uigetdir;
parent_directory = select_a;
cd(parent_directory)

experiments = dir;
experiments = experiments([experiments(:).isdir]==1);
experiments = experiments(~ismember({experiments(:).name}, {'.','..'}));
experiment_folders = string([]);

for l = 1:length(experiments)
    experiment_folders = [experiment_folders; experiments(l).name];
end

for a = 1:length(experiment_folders)
    experiment_directory = strcat(parent_directory, ...
        '\', char(experiment_folders(a)));
    cd(experiment_directory)

    images = dir;
    images = images([images(:).isdir]==1);
    images = images(~ismember({images(:).name}, {'.','..'}));
    image_folders = string([]);

    for p = 1:length(images)
        image_folders = [image_folders; images(p).name];
    end

    for q = 1:length(image_folders)
        image_directory = strcat(experiment_directory, ...
            '\', char(image_folders(q)));
        cd(image_directory)
        tifs = dir('**/*.tif');
        tif_files = string([]);

        for k = 1:length(tifs)
            tif_files = [tif_files; tifs(k).name];
        end

        fl_image_files = tif_files;

```

```
        for k = 1:length(tif_files)
            if contains(tif_files(k), 'Cy5')
                fl_image_Cy5 = tif_files(k);
            elseif contains(tif_files(k), 'DAPI')
                fl_image_DAPI = tif_files(k);
            elseif contains(tif_files(k), 'DsRed')
                fl_image_DsRed = tif_files(k);
            elseif contains(tif_files(k), 'EGFP')
                fl_image_EGFP = tif_files(k);
            elseif contains(tif_files(k), 'Bright')
                bf_image_file = tif_files(k);
                fl_image_files = ...
                    fl_image_files(fl_image_files ~= bf_image_file);
            else
            end
        end

        BF_IMAGE = imread(char(bf_image_file));

        for z = 1:length(fl_image_files)
            close all
            FL_IMAGE = imread(char(fl_image_files(z)));

%% 2.2.2 Electrode Locations were Identified using Bright Field Images
% Figure 7b, Supplemental Figure 2vi-vii

[centers_bright, radii_bright] = imfindcircles(BF_IMAGE, [25 60], ...
    'Sensitivity', 0.85, 'ObjectPolarity', 'bright', 'EdgeThreshold', 0.05);

% for Set 1...
fig_0 = figure('Name', 'Brightfield 0');
fig_0.InvertHardcopy = 'off';
imshow(BF_IMAGE);
axis xy, hold on;
viscircles(centers_bright, radii_bright, 'Color', 'red', ...
    'LineStyle', '-', 'EnhanceVisibility', false, 'LineWidth', 1);
ylabel('Bright Field');
title('Located Electrodes');
hold off;

[centers_dark, radii_dark] = imfindcircles(BF_IMAGE, [25 60], ...
    'Sensitivity', 0.85, 'ObjectPolarity', 'dark', 'EdgeThreshold', 0.05);

% for Set 2...
fig_1 = figure('Name', 'Brightfield 1');
fig_1.InvertHardcopy = 'off';
imshow(BF_IMAGE);
axis xy, hold on;
viscircles(centers_dark, radii_dark, 'Color', 'red', 'LineStyle', '-', ...
    'EnhanceVisibility', false, 'LineWidth', 1);
ylabel('Bright Field');
```

```
title('Located Electrodes');
hold off;

radii_proxy = 38; % determined empirically for 5X objective
square_edge_length = 161; % determined empirically for 5X objective
square_bottomleft_bright = centers_bright - (square_edge_length)/2;
square_topright_bright = centers_bright + (square_edge_length)/2;
square_bottomleft_dark = centers_dark - (square_edge_length)/2;
square_topright_dark = centers_dark + (square_edge_length)/2;

%% 2.2.3 Unusable Electrodes with Bright Field Artifacts were Removed from Analysis
% Figure 7c, Supplemental Figure 2viii-xii

% for Set 1...
[sort_bottomleft_corners_check_bright, sort_topright_corners_check_bright, ...
sort_centers_check_bright, sort_radii_check_bright] = ...
getXSortedCheckedData(centers_bright, radii_bright, ...
square_bottomleft_bright, square_topright_bright);

% for Set 2...
[sort_bottomleft_corners_check_dark, sort_topright_corners_check_dark, ...
sort_centers_check_dark, sort_radii_check_dark] = ...
getXSortedCheckedData(centers_dark, radii_dark, square_bottomleft_dark, ...
square_topright_dark);

% for Set 1...
[sort_corners_check_bright_DIM, ~] = size(sort_bottomleft_corners_check_bright);

xx1_sort_bright = sort_bottomleft_corners_check_bright(:,1);
xx2_sort_bright = xx1_sort_bright + square_edge_length;
yy1_sort_bright = sort_bottomleft_corners_check_bright(:,2);
yy2_sort_bright = yy1_sort_bright + square_edge_length;
X_sort_bright = ...
    [xx1_sort_bright, xx2_sort_bright, xx2_sort_bright, xx1_sort_bright, ...
    xx1_sort_bright];
Y_sort_bright = ...
    [yy1_sort_bright, yy1_sort_bright, yy2_sort_bright, yy2_sort_bright, ...
    yy1_sort_bright];

fig_2_sort_bright = figure('Name', 'Brightfield 2 - Bright');
fig_2_sort_bright.InvertHardcopy = 'off';
imshow(BF_IMAGE);
viscircles(sort_centers_check_bright, sort_radii_check_bright, 'Color', ...
    'red', 'LineStyle', '-', 'EnhanceVisibility', false, 'LineWidth', 1);
axis xy, hold on;

for a = 1:sort_corners_check_bright_DIM
    plot(X_sort_bright(a,:),Y_sort_bright(a,:),'y-','LineWidth', 1);
    text(sort_centers_check_bright(a,1) - square_edge_length/3, ...
        sort_centers_check_bright(a,2) - square_edge_length/3, ...
        sprintf('%2u', a), 'Color', 'white', 'FontSize', 8, ...
```

```
        'FontWeight', 'bold');
    end
    title('Isolated Electrodes - Bright');
    hold off;

    % for Set 2...
    [sort_corners_check_dark_DIM, ~] = size(sort_bottomleft_corners_check_dark);

    xx1_sort_dark = sort_bottomleft_corners_check_dark(:,1);
    xx2_sort_dark = xx1_sort_dark + square_edge_length;
    yy1_sort_dark = sort_bottomleft_corners_check_dark(:,2);
    yy2_sort_dark = yy1_sort_dark + square_edge_length;
    X_sort_dark = [xx1_sort_dark, xx2_sort_dark, xx2_sort_dark, xx1_sort_dark, ...
        xx1_sort_dark];
    Y_sort_dark = [yy1_sort_dark, yy1_sort_dark, yy2_sort_dark, yy2_sort_dark, ...
        yy1_sort_dark];

    fig_2_sort_dark = figure('Name', 'Brightfield 2 - Dark');
    fig_2_sort_dark.InvertHardcopy = 'off';
    imshow(BF_IMAGE);
    viscircles(sort_centers_check_dark, sort_radii_check_dark, 'Color', ...
        'red', 'LineStyle', '-', 'EnhanceVisibility', false, 'LineWidth', 1);
    axis xy, hold on;

    for a = 1:sort_corners_check_dark_DIM
        plot(X_sort_dark(a,:),Y_sort_dark(a,:),'y-','LineWidth', 1);
        text(sort_centers_check_dark(a,1) - square_edge_length/3, ...
            sort_centers_check_dark(a,2) - square_edge_length/3, ...
            sprintf('%2u', a), 'Color', 'white', 'FontSize', 8, ...
            'FontWeight', 'bold');
    end
    title('Isolated Electrodes - Dark');
    hold off;

    % for Set 1...
    [overlap_bright_all] = checkOverlap(sort_corners_check_bright_DIM, ...
        xx1_sort_bright, yy1_sort_bright, xx2_sort_bright, yy2_sort_bright);

    % for Set 2...
    [overlap_dark_all] = checkOverlap(sort_corners_check_dark_DIM, ...
        xx1_sort_dark, yy1_sort_dark, xx2_sort_dark, yy2_sort_dark);

    % for Set 1...
    [sort_centers_check_overlap_bright, sort_radii_check_overlap_bright, ...
        sort_bottomleft_corners_check_overlap_bright, ...
        sort_topright_corners_check_overlap_bright, ...
        sort_radii_check_overlap_DIM_bright] = ...
        removeOverlap(sort_centers_check_bright, sort_radii_check_bright, ...
        sort_bottomleft_corners_check_bright, sort_topright_corners_check_bright, ...
        overlap_bright_all);

```

```
% for Set 2...
[sort_centers_check_overlap_dark, sort_radii_check_overlap_dark, ...
    sort_bottomleft_corners_check_overlap_dark, ...
    sort_topright_corners_check_overlap_dark, ...
    sort_radii_check_overlap_DIM_dark] = ...
    removeOverlap(sort_centers_check_dark, sort_radii_check_dark, ...
    sort_bottomleft_corners_check_dark, sort_topright_corners_check_dark, ...
    overlap_dark_all);

% for Set 1...
xx1_sort_overlap_bright = sort_bottomleft_corners_check_overlap_bright(:,1);
xx2_sort_overlap_bright = xx1_sort_overlap_bright + square_edge_length;
yy1_sort_overlap_bright = sort_bottomleft_corners_check_overlap_bright(:,2);
yy2_sort_overlap_bright = yy1_sort_overlap_bright + square_edge_length;
X_sort_overlap_bright = [xx1_sort_overlap_bright, xx2_sort_overlap_bright, ...
    xx2_sort_overlap_bright, xx1_sort_overlap_bright, xx1_sort_overlap_bright];
Y_sort_overlap_bright = [yy1_sort_overlap_bright, yy1_sort_overlap_bright, ...
    yy2_sort_overlap_bright, yy2_sort_overlap_bright, yy1_sort_overlap_bright];

fig_2_sort_overlap_bright = figure('Name', 'Brightfield 2 - Bright');
fig_2_sort_overlap_bright.InvertHardcopy = 'off';
imshow(BF_IMAGE);
viscircles(sort_centers_check_overlap_bright, sort_radii_check_overlap_bright, ...
    'Color', 'red', 'LineStyle', '-', 'EnhanceVisibility', false, 'LineWidth', 1);
axis xy, hold on;

for a = 1:sort_radii_check_overlap_DIM_bright
    plot(X_sort_overlap_bright(a,:),Y_sort_overlap_bright(a,:),'y-','LineWidth', 1);
    text(sort_centers_check_overlap_bright(a,1) - square_edge_length/3, ...
        sort_centers_check_overlap_bright(a,2) - square_edge_length/3, ...
        sprintf('%2u', a), 'Color', 'white', 'FontSize', 8, 'FontWeight', 'bold');
end
title('Isolated Electrodes - Bright');
hold off;

% for Set 2...
xx1_sort_overlap_dark = sort_bottomleft_corners_check_overlap_dark(:,1);
xx2_sort_overlap_dark = xx1_sort_overlap_dark + square_edge_length;
yy1_sort_overlap_dark = sort_bottomleft_corners_check_overlap_dark(:,2);
yy2_sort_overlap_dark = yy1_sort_overlap_dark + square_edge_length;
X_sort_overlap_dark = [xx1_sort_overlap_dark, xx2_sort_overlap_dark, ...
    xx2_sort_overlap_dark, xx1_sort_overlap_dark, xx1_sort_overlap_dark];
Y_sort_overlap_dark = [yy1_sort_overlap_dark, yy1_sort_overlap_dark, ...
    yy2_sort_overlap_dark, yy2_sort_overlap_dark, yy1_sort_overlap_dark];

fig_2_sort_overlap_dark = figure('Name', 'Brightfield 2 - Dark');
fig_2_sort_overlap_dark.InvertHardcopy = 'off';
imshow(BF_IMAGE);
viscircles(sort_centers_check_overlap_dark, sort_radii_check_overlap_dark, ...
    'Color', 'red', 'LineStyle', '-', 'EnhanceVisibility', false, 'LineWidth', 1);
axis xy, hold on;
```

```
252
253     for a = 1:sort_radii_check_overlap_DIM_dark
254         plot(X_sort_overlap_dark(a,:),Y_sort_overlap_dark(a,:),'y-','LineWidth', 1);
255         text(sort_centers_check_overlap_dark(a,1) - square_edge_length/3, ...
256             sort_centers_check_overlap_dark(a,2) - square_edge_length/3, ...
257             sprintf('%2u', a), 'Color', 'white', 'FontSize', 8, 'FontWeight', 'bold');
258     end
259     title('Isolated Electrodes - Dark');
260     hold off;
261
262     % Supplemental Figure 2xii
263     num_usable_bright = length(sort_radii_check_overlap_bright);
264     num_usable_dark = length(sort_radii_check_overlap_dark);
265     num_usable_circles = 0;
266
267     if num_usable_bright >= num_usable_dark
268         xx1 = xx1_sort_overlap_bright;
269         xx2 = xx2_sort_overlap_bright;
270         yy1 = yy1_sort_overlap_bright;
271         yy2 = yy2_sort_overlap_bright;
272         X = X_sort_overlap_bright;
273         Y = Y_sort_overlap_bright;
274         sort_corners_check_DIM = sort_radii_check_overlap_DIM_bright;
275         sort_centers_check = sort_centers_check_overlap_bright;
276         sort_radii_check = sort_radii_check_overlap_bright;
277         sort_bottomleft_corners_check = sort_bottomleft_corners_check_overlap_bright;
278         sort_topright_corners_check = sort_topright_corners_check_overlap_bright;
279         use_these_circles = 1;
280         num_usable_circles = num_usable_bright;
281     elseif num_usable_bright < num_usable_dark
282         xx1 = xx1_sort_overlap_dark;
283         xx2 = xx2_sort_overlap_dark;
284         yy1 = yy1_sort_overlap_dark;
285         yy2 = yy2_sort_overlap_dark;
286         X = X_sort_overlap_dark;
287         Y = Y_sort_overlap_dark;
288         sort_corners_check_DIM = sort_radii_check_overlap_DIM_dark;
289         sort_centers_check = sort_centers_check_overlap_dark;
290         sort_radii_check = sort_radii_check_overlap_dark;
291         sort_bottomleft_corners_check = sort_bottomleft_corners_check_overlap_dark;
292         sort_topright_corners_check = sort_topright_corners_check_overlap_dark;
293         use_these_circles = 2;
294         num_usable_circles = num_usable_dark;
295     end
296
297     %% 2.2.4 A Rigid Digital Mask Differentiated the Region-of-Interest from ...
298     % Background at Each Electrode
299     % Figure 7d, Supplemental Figure 2xiii, xv
300
301     fig_3 = figure('Name', 'Fluorescence 1');
302     fig_3.InvertHardcopy = 'off';
```

```
imshow(FL_IMAGE);
axis xy;
ylabel('Fluorescence');
title('Unedited Monochrome Image');

fig_4 = figure('Name', 'Fluorescence 2');
fig_4.InvertHardcopy = 'off';
imshow(FL_IMAGE);
axis xy, hold on;
sub_image_matrix_fl = uint32([]);

for a = 1:1:sort_corners_check_DIM
    sub_image_matrix_fl(:,:,a) = uint32(FL_IMAGE(yy1(a):yy2(a), xx1(a):xx2(a)));
    plot(X(a,:),Y(a,:),'y','LineWidth', 1);
    text(sort_centers_check(a,1) - square_edge_length/3, ...
        sort_centers_check(a,2) - square_edge_length/3, sprintf('%2u', a), ...
        'Color', 'white', 'FontSize', 8, 'FontWeight', 'bold');
end
hold off;
title('Transferred Electrode Map');

%% 2.2.6 Unusable Electrodes with Fluorescence Artifacts were Removed from Analysis
% Figure 7h, Supplemental Figure 2xiv
[FL_IMAGE_sat_copy, sat_markers_x, sat_markers_y, ...
    threshold_FROM_sat_cutoff] = getSatCutoff_v2(FL_IMAGE);

fig_5 = figure('Name', 'Fluorescence 2 - Bright Artifacts');
fig_5.InvertHardcopy = 'off';
imshow(FL_IMAGE_sat_copy)
axis xy, hold on;
viscircles(sort_centers_check, sort_radii_check, 'Color', 'blue', ...
    'LineStyle', '-', 'EnhanceVisibility', false, 'LineWidth', 2);
for a = 1:1:sort_corners_check_DIM
    plot(X(a,:),Y(a,:),'y','LineWidth', 1);
    text(sort_centers_check(a,1) - square_edge_length/3, ...
        sort_centers_check(a,2) - square_edge_length/3, sprintf('%2u', a), ...
        'Color', 'white', 'FontSize', 8, 'FontWeight', 'bold');
end
plot(sat_markers_x, sat_markers_y, 'rx', 'MarkerSize', 4, 'Linewidth', 1)
hold off

%% 2.2.4 A Rigid Digital Mask Differentiated the Region-of-Interest from ...
% Background at Each Electrode
% Figure 7e, Supplemental Figure 2xvi

[rows_stack, cols_stack, rows_image, cols_image, sub_image_matrix_fl_ROI_ONLY, ...
    sub_image_matrix_fl_BG_ONLY, sub_image_matrix_fl_DIM, radius_inner, ...
    radius_outer, square_center, ring_pixels, ring_pixels_proxy] = ...
    applyMask_v2(sub_image_matrix_fl, radii_proxy, square_edge_length);

%% 2.2.5 Local Noise Thresholds were Calculated from Background Pixels and ...
```

```
% Subtracted from all Pixels at Each Electrode
% Figure 7f-g, Supplemental Figure 2xvii-xxi

[STAT_LEVEL, STD_BG_array, MEAN_BG_array, threshold_FROM_BG_array, ...
    sub_image_matrix_fl_thresh] = getElectrodeStats(sub_image_matrix_fl_DIM, ...
    sub_image_matrix_fl, sub_image_matrix_fl_BG_ONLY);

%% 2.2.6 Unusable Electrodes with Fluorescence Artifacts were Removed from Analysis
% Figure 7h, Supplemental Figure 2xxii-xxiii

[perc_sat_BG_matrix, squares_to_elim, sort_centers_check_elim, ...
    sort_radii_check_elim, sort_bottomleft_corners_check_elim, ...
    sort_topright_corners_check_elim, sub_image_matrix_fl_thresh_elim, ...
    counter_elim, sub_image_matrix_fl_elim] = ...
    elimSquaresAUTO_2(threshold_FROM_sat_cutoff, sub_image_matrix_fl_BG_ONLY, ...
    sub_image_matrix_fl_DIM, rows_stack, cols_stack, ring_pixels_proxy, ...
    sort_centers_check, sort_radii_check, sort_bottomleft_corners_check, ...
    sort_topright_corners_check, sub_image_matrix_fl_thresh, sub_image_matrix_fl);

elim_check = isnan(sub_image_matrix_fl_thresh_elim(1,1,:));

fig_8 = figure('Name', 'Fluorescence: Check Eliminated Electrodes');
fig_8.InvertHardcopy = 'off';
imshow(FL_IMAGE);
axis xy, hold on

for a = 1:sort_corners_check_DIM
    if sum(ismember(a,squares_to_elim)) == 1
    plot(X(a,:),Y(a,:),'r--','LineWidth', 2);
    text(sort_centers_check(a,1) - square_edge_length/3, ...
        sort_centers_check(a,2) - square_edge_length/3, sprintf('%2u', a), ...
        'Color', 'red', 'FontSize', 10, 'FontWeight', 'bold');
    else
    plot(X(a,:),Y(a,:),'y','LineWidth', 1);
    text(sort_centers_check(a,1) - square_edge_length/3, ...
        sort_centers_check(a,2) - square_edge_length/3, sprintf('%2u', a), ...
        'Color', 'white', 'FontSize', 10, 'FontWeight', 'bold');
    end
end
hold off

%% 2.2.7 Fluorescence Intensity was Calculated from Usable Electrodes
% Figure 7i, Supplemental Figure 2xxiv-xxviii

sum_stack_fl_thresh_elim = nansum(sub_image_matrix_fl_thresh_elim, 3);
sum_stack_norm_fl_thresh_elim = ...
    sum_stack_fl_thresh_elim/(sub_image_matrix_fl_DIM - counter_elim);

fig_9 = figure();
fig_9.InvertHardcopy = 'off';
for a = 1:10
```

```
405         sum_stack_norm_fl_elim_adjust = imadjust(uint16(...
406             sum_stack_norm_fl_thresh_elim), [0 0.05*a], [0 1]);
407         subplot(2, 5, a)
408         imshow(uint16(sum_stack_norm_fl_elim_adjust))
409     end
410     suptitle('Average of Noise-Subtracted Electrodes (contrast-enhanced)')
411
412     [CHECK_total_num_pix, pix_int_metric] ...
413         = consistencyChecks_2(sub_image_matrix_fl_ROI_ONLY, ...
414         sub_image_matrix_fl_BG_ONLY, rows_stack, cols_stack, ...
415         sum_stack_norm_fl_thresh_elim);
416
417     fig_10 = figure();
418     fig_10.InvertHardcopy = 'off';
419     surf(sum_stack_norm_fl_thresh_elim)
420     xlim([0 rows_stack])
421     ylim([0 cols_stack])
422     zlim([0 max(max(sum_stack_norm_fl_thresh_elim))])
423     zlabel('Pixel Intensity [a.u.]')
424     title('Mean Noise-Subtracted Fluorescence Intensity')
425
426     sum_stack_norm_fl_thresh_elim_ROI_ONLY = sum_stack_norm_fl_thresh_elim;
427     sum_stack_norm_fl_thresh_elim_BG_ONLY = sum_stack_norm_fl_thresh_elim;
428     ROI_pix_count = 0;
429     BG_pix_count = 0;
430
431     for m = 1:rows_stack
432         for n = 1:cols_stack
433             if ring_pixels(m,n) == 0
434                 sum_stack_norm_fl_thresh_elim_ROI_ONLY(m,n) = 0;
435                 BG_pix_count = BG_pix_count + 1;
436             elseif ring_pixels(m,n) == 1
437                 sum_stack_norm_fl_thresh_elim_BG_ONLY(m,n) = 0;
438                 ROI_pix_count = ROI_pix_count + 1;
439             else
440             end
441         end
442     end
443
444     sum_stack_ALL = sum(sum(sum_stack_norm_fl_thresh_elim));
445     sum_stack_ROI_ONLY = sum(sum(sum_stack_norm_fl_thresh_elim_ROI_ONLY));
446     sum_stack_BG_ONLY = sum(sum(sum_stack_norm_fl_thresh_elim_BG_ONLY));
447     check_again = sum_stack_ROI_ONLY + sum_stack_BG_ONLY;
448     check_count = ROI_pix_count + BG_pix_count;
449     check_pix_int_metric = check_again/check_count;
450
451     sum_stack_norm_ROI_ONLY = sum_stack_ROI_ONLY/check_count;
452     sum_stack_norm_BG_ONLY = sum_stack_BG_ONLY/check_count;
453
454     frac_ROI = sum_stack_norm_ROI_ONLY/check_pix_int_metric
455     frac_BG = sum_stack_norm_BG_ONLY/check_pix_int_metric;
```

```
456
457     num_used_electrodes = length(sub_image_matrix_fl_thresh) - counter_elim
458
459     [scores, sample_average, population_std_estimate, confidence_level, ...
460         uncertainty] = confidenceInterval(sub_image_matrix_fl_thresh_elim);
461
462     check_score = sample_average - pix_int_metric;
463
464     % Supplemental Figure 2xxvi
465     det_lim_warning = string('WARNING: Limit of Detection Surpassed');
466     noise_lev_warning = string('WARNING: High Noise Level');
467     num_elec_warning = string('WARNING: Low Statistical Power');
468     saturation_warning = string('WARNING: High Saturation Level');
469     confidence_warning = string('WARNING: Problem with Confidence Interval');
470
471     warning_list = string([]);
472
473     if pix_int_metric < 1.5
474         disp(det_lim_warning)
475         warning_list = cat(1, warning_list, det_lim_warning);
476     else
477     end
478
479     if frac_ROI < 0.90
480         disp(noise_lev_warning)
481         warning_list = cat(1, warning_list, noise_lev_warning);
482     else
483     end
484
485     if num_used_electrodes < 100
486         disp(num_elec_warning)
487         warning_list = cat(1, warning_list, num_elec_warning);
488     else
489     end
490
491     check_frac = frac_ROI + frac_BG;
492
493     sub_image_matrix_fl_elim_copy = sub_image_matrix_fl_elim;
494     sub_image_matrix_fl_elim_copy(sub_image_matrix_fl_elim >= 65532) = 1;
495     sub_image_matrix_fl_elim_copy(sub_image_matrix_fl_elim < 65532) = 0;
496     true_saturation_counter = nansum(nansum(nansum(sub_image_matrix_fl_elim_copy)));
497     true_saturation_frac = ...
498         true_saturation_counter / (rows_stack*cols_stack*num_usable_circles)
499
500     if true_saturation_frac >= 0.01
501         disp(saturation_warning)
502         warning_list = cat(1, warning_list, saturation_warning);
503     else
504     end
505
506     if abs(check_score) > pix_int_metric*0.01
```

```
507         disp(confidence_warning)
508         warning_list = cat(1, warning_list, confidence_warning);
509     else
510     end
511
512     if CHECK_total_num_pix  == 0 && abs(check_score) < pix_int_metric*0.01
513         script_output = ['Main Output: ',  num2str(pix_int_metric), ' +/- ', ...
514             num2str(uncertainty), ' a.u.'];
515         disp(script_output)
516     else
517         script_output = ['Main Output: ',  num2str(pix_int_metric), ' +/- ', ...
518             num2str(uncertainty), ' a.u.'];
519         disp(script_output)
520         disp('Ask Kyle for help...')
521     end
522
523     fig_11 = figure();
524     bins = 15;
525     scores_hist = histogram(scores(:,1), bins);
526     hold on
527     metric_x = [pix_int_metric, pix_int_metric];
528     metric_y = fig_11.CurrentAxes.YLim;
529     conf_bound_lower_x = metric_x - uncertainty;
530     conf_bound_upper_x = metric_x + uncertainty;
531     plot(metric_x, metric_y, 'k--', 'LineWidth', 2);
532     plot(conf_bound_lower_x, metric_y, 'r--', 'LineWidth', 2);
533     plot(conf_bound_upper_x, metric_y, 'r--', 'LineWidth', 2);
534     xlabel('Average Pixel Intensity by Electrode [a.u.]');
535     ylabel('Count');
536     hold off
537
538     fig_12 = figure();
539     bg_avgs_hist = histogram(double(MEAN_BG_array), bins);
540     hold on
541     BG_sample_average = mean(double(MEAN_BG_array));
542     avg_BG_average_x = [BG_sample_average, BG_sample_average];
543     avg_BG_average_y = fig_12.CurrentAxes.YLim;
544     plot(avg_BG_average_x, avg_BG_average_y, 'k--', 'LineWidth', 2);
545     xlabel('Average BG Pixel Intensity by Electrode [a.u.]')
546     ylabel('Count')
547     hold off
548
549     fig_13 = figure();
550     thresh_hist = histogram(threshold_FROM_BG_array, bins);
551     hold on
552     thresh_sample_average = mean(threshold_FROM_BG_array);
553     thresh_average_x = [thresh_sample_average, thresh_sample_average];
554     thresh_average_y = fig_13.CurrentAxes.YLim;
555     plot(thresh_average_x, thresh_average_y, 'k--', 'LineWidth', 2);
556     xlabel('Threshold by Electrode [a.u.]')
557     ylabel('Count')
```

```
hold off

% Supplemental Figure 2xxvii-xxviii
format long

Folder_Name = strcat(datestr(now, 'yyyymmddTHHMMSS'));
for_report = mkdir(Folder_Name);
Report_Folder = strcat(image_directory, '\', Folder_Name);
cd(Report_Folder)

Time_Stamp = Folder_Name;
Working_Directory_Path = image_directory;
Bright_Field_Image  = bf_image_file;
Fluorescence_Image = fl_image_files(z);
Script_Version = mfilename;
Eliminated_Electrodes  = num2str(squares_to_elim);
Statistical_Stringency = STAT_LEVEL;
Uncertainty =  uncertainty;
Lower_Confidence_Bound = conf_bound_lower_x(1);
Upper_Confidence_Bound = conf_bound_upper_x(1);
Pixel_Intensity_Metric = pix_int_metric;

summary_cell_1 = {Time_Stamp, Working_Directory_Path, Bright_Field_Image, ...
    Fluorescence_Image, Script_Version, Statistical_Stringency, ...
    Eliminated_Electrodes, Uncertainty, Lower_Confidence_Bound, ...
    Upper_Confidence_Bound, Pixel_Intensity_Metric};
summary_cell_1_names = {'Time_Stamp', 'Working_Directory_Path', ...
    'Bright_Field_Image', 'Fluorescence_Image', 'Script_Version', ...
    'Statistical_Stringency', 'Eliminated_Electrodes', ...
    'Uncertainty', 'Lower_Confidence_Bound', ...
    'Upper_Confidence_Bound', 'Pixel_Intensity_Metric'};
Summary_1 = cell2table(summary_cell_1, 'VariableNames', summary_cell_1_names);

Electrode = transpose(1:1:sort_corners_check_DIM);
Centers_X = sort_centers_check(:,1);
Centers_Y = sort_centers_check(:,2);
Radius = sort_radii_check;
Mean_of_BG = transpose(MEAN_BG_array);
STD_of_BG = transpose(STD_BG_array);
Threshold = transpose(threshold_FROM_BG_array);
Avg_Pixel_Intensity_By_Electrode  = scores(:,1);

summary_array_2 = [Electrode, Centers_X, Centers_Y, Radius, ...
    Mean_of_BG, STD_of_BG, Threshold, Avg_Pixel_Intensity_By_Electrode];
Summary_2 = array2table(summary_array_2, 'VariableNames', {'Electrode', ...
    'Centers_X', 'Centers_Y', 'Radius', 'Mean_of_BG', 'STD_of_BG', ...
    'Threshold', 'Avg_Pixel_Intensity_By_Electrode'});

Bright_vs_Dark_Circles = use_these_circles;
Mean_of_Radii = mean(double(Radius));
STD_of_Radii = std(double(Radius));
```

```
609     Average_of_Mean_of_BG = mean(double(Mean_of_BG));
610     STD_of_Mean_of_BG = std(double(Mean_of_BG));
611     Mean_Threshold = mean(double(Threshold));
612     STD_Threshold = std(double(Threshold));
613
614     summary_array_3 = [Bright_vs_Dark_Circles, Mean_of_Radii, STD_of_Radii, ...
615         Average_of_Mean_of_BG, STD_of_Mean_of_BG, Mean_Threshold, STD_Threshold];
616     Summary_3 = array2table(summary_array_3, 'VariableNames', ...
617         {'Bright_vs_Dark_Circles', 'Mean_of_Radii', 'STD_of_Radii', ...
618         'Average_of_Mean_of_BG', 'STD_of_Mean_of_BG', 'Mean_Threshold', ...
619         'STD_Threshold'});
620
621     ROI_Contribution_to_Metric = frac_ROI;
622     Number_of_Used_Electrodes = num_used_electrodes;
623     True_Saturation_Level = true_saturation_frac;
624
625     summary_array_4 = [Pixel_Intensity_Metric, ROI_Contribution_to_Metric, ...
626         Number_of_Used_Electrodes, True_Saturation_Level];
627     Summary_4 = array2table(summary_array_4, 'VariableNames', ...
628         {'Pixel_Intensity_Metric', 'ROI_Contribution_to_Metric', ...
629         'Number_of_Used_Electrodes', 'True_Saturation_Level'});
630
631     Excel_Report = strcat(Folder_Name, '.xlsx');
632     filename = [pwd '\' Excel_Report];
633     writetable(Summary_1, filename, 'FileType', 'spreadsheet', 'Sheet', 1);
634     writetable(Summary_2, filename, 'FileType', 'spreadsheet', 'Sheet', 2);
635     writetable(Summary_3, filename, 'FileType', 'spreadsheet', 'Sheet', 3);
636     writetable(Summary_4, filename, 'FileType', 'spreadsheet', 'Sheet', 4);
637
638     filename_warnings = [pwd '\' Time_Stamp '_Warnings.txt'];
639     warning_summary_file = fopen(filename_warnings, 'wt');
640     fprintf(warning_summary_file, '%s\n', warning_list);
641     fclose(warning_summary_file);
642
643     saveas(fig_0, strcat(Folder_Name, '_BF_0.jpg'));
644     saveas(fig_1, strcat(Folder_Name, '_BF_1.jpg'));
645     saveas(fig_2_sort_bright, strcat(Folder_Name, '_BF_2.jpg'));
646     saveas(fig_2_sort_dark, strcat(Folder_Name, '_BF_3.jpg'));
647     saveas(fig_2_sort_overlap_bright, strcat(Folder_Name, '_BF_4.jpg'));
648     saveas(fig_2_sort_overlap_dark, strcat(Folder_Name, '_BF_5.jpg'));
649     saveas(fig_3, strcat(Folder_Name,'_FL_1.jpg'));
650     saveas(fig_4, strcat(Folder_Name,'_FL_2.jpg'));
651     saveas(fig_5, strcat(Folder_Name,'_Bright_Artifacts.jpg'));
652     saveas(fig_8, strcat(Folder_Name, '_Eliminated_Electrodes.jpg'));
653     saveas(fig_9, strcat(Folder_Name, '_Stack_Sum.jpg'));
654     saveas(fig_10, strcat(Folder_Name, '_Stack_3D.jpg'));
655     saveas(fig_11, strcat(Folder_Name, '_Scores.jpg'));
656     saveas(fig_12, strcat(Folder_Name, '_BG_Avgs.jpg'));
657     saveas(fig_13, strcat(Folder_Name, '_Thresholds.jpg'));
658
659     save(strcat(Folder_Name, '_Variables.mat'))
```

```
660
661     radius_ROI_inner = radius_inner + 1;
662     radius_ROI_outer = radius_outer - 1;
663     plane_at_z_vertex_x = [cols_stack, 0, 0, cols_stack];
664     plane_at_z_vertex_y = [cols_stack, cols_stack, 0, 0];
665     plane_at_z_color = [0, 0, 0];
666
667         cd ..
668             end
669         cd ..
670         end
671         cd ..
672     end
673     close
```

S1.2.2 Called Function 1 (getXSortedCheckedData)

```
%% 2.2.3 Unusable Electrodes with Bright Field Artifacts were Removed from Analysis
% Figure 7c, Supplemental Figure 2ix

function [sort_bottomleft_corners_check, sort_topright_corners_check, ...
    sort_centers_check, sort_radii_check] = ...
    getXSortedCheckedData(centers, radii, square_bottomleft, square_topright)

[~, index_x_sort] = sort(centers(:,1));
centers_x_sort = centers(index_x_sort, :);
radii_x_sort = radii(index_x_sort, :);
square_bottomleft_x_sort = square_bottomleft(index_x_sort, :);
square_topright_x_sort = square_topright(index_x_sort, :);
sort_bottomleft_corners_check = ...
    square_bottomleft_x_sort(square_bottomleft_x_sort(:,1) > 0 ...
    & square_topright_x_sort(:,1) <= 2752 & ...
    square_bottomleft_x_sort(:,2) > 0 ...
    & square_topright_x_sort(:,2) <= 2208,:);
sort_topright_corners_check = ...
    square_topright_x_sort(square_bottomleft_x_sort(:,1) > 0 ...
    & square_topright_x_sort(:,1) <= 2752 & ...
    square_bottomleft_x_sort(:,2) > 0 ...
    & square_topright_x_sort(:,2) <= 2208,:);
sort_centers_check = ...
    centers_x_sort(square_bottomleft_x_sort(:,1) > 0 ...
    & square_topright_x_sort(:,1) <= 2752 & ...
    square_bottomleft_x_sort(:,2) > 0 ...
    & square_topright_x_sort(:,2) <= 2208,:);
sort_radii_check = ...
    radii_x_sort(square_bottomleft_x_sort(:,1) > 0 ...
    & square_topright_x_sort(:,1) <= 2752 & ...
    square_bottomleft_x_sort(:,2) > 0 ...
    & square_topright_x_sort(:,2) <= 2208,:);

end
```

S1.2.3 Called Function 2 (checkOverlap)

```
%% 2.2.3 Unusable Electrodes with Bright Field Artifacts were Removed from Analysis
% Figure 7c, Supplemental Figure 2x

function [overlap_all] = checkOverlap(sort_corners_check_DIM, xx1, yy1, xx2, yy2);

overlap_BL = [];
overlap_BR = [];
overlap_TR = [];
overlap_TL = [];

for a = 1:sort_corners_check_DIM
    if a <= (sort_corners_check_DIM - 1)
        for k = (a+1):sort_corners_check_DIM

            if xx1(a) >= xx1(k) && xx1(a) <= xx2(k) && yy1(a) >= yy1(k) ...
                    && yy1(a) <= yy2(k)
                overlap_BL = [overlap_BL; a; k];
            else
            end

            if xx1(a) >= xx1(k) && xx1(a) <= xx2(k) && yy2(a) >= yy1(k) ...
                    && yy2(a) <= yy2(k)
                overlap_BR = [overlap_BR; a; k];
            else
            end

            if xx2(a) >= xx1(k) && xx2(a) <= xx2(k) && yy2(a) >= yy1(k) ...
                    && yy2(a) <= yy2(k)
                overlap_TR = [overlap_TR; a; k];
            else
            end

            if xx2(a) >= xx1(k) && xx2(a) <= xx2(k) && yy1(a) >= yy1(k) ...
                    && yy1(a) <= yy2(k)
                overlap_TL = [overlap_TL; a; k];
            else
            end

        end
    else
    end

overlap_all = unique([overlap_BL; overlap_BR; overlap_TL; overlap_TR]);

end
```

S1.2.4 Called Function 3 (removeOverlap)

```
%% 2.2.3 Unusable Electrodes with Bright Field Artifacts were Removed from Analysis
% Figure 7c, Supplemental Figure 2xi

function [sort_centers_check_overlap, sort_radii_check_overlap, ...
    sort_bottomleft_corners_check_overlap, ...
    sort_topright_corners_check_overlap, sort_radii_check_overlap_DIM] = ...
    removeOverlap(sort_centers_check, sort_radii_check, ...
    sort_bottomleft_corners_check, sort_topright_corners_check, overlap_all)

sort_centers_check_overlap = sort_centers_check;
sort_radii_check_overlap = sort_radii_check;
sort_bottomleft_corners_check_overlap = sort_bottomleft_corners_check;
sort_topright_corners_check_overlap = sort_topright_corners_check;

for a = 1:length(overlap_all)

    sort_centers_check_overlap(overlap_all(a),:) = NaN;
    sort_radii_check_overlap(overlap_all(a)) = NaN;
    sort_bottomleft_corners_check_overlap(overlap_all(a),:) = NaN;
    sort_topright_corners_check_overlap(overlap_all(a),:) = NaN;

end

sort_centers_check_overlap_x = sort_centers_check_overlap(:,1);
sort_centers_check_overlap_y = sort_centers_check_overlap(:,2);
sort_centers_check_overlap_x = ...
    sort_centers_check_overlap_x(~isnan(sort_centers_check_overlap_x));
sort_centers_check_overlap_y = ...
    sort_centers_check_overlap_y(~isnan(sort_centers_check_overlap_y));
sort_centers_check_overlap = ...
    [sort_centers_check_overlap_x, sort_centers_check_overlap_y];

sort_bottomleft_corners_check_overlap_x = ...
    sort_bottomleft_corners_check_overlap(:,1);
sort_bottomleft_corners_check_overlap_y = ...
    sort_bottomleft_corners_check_overlap(:,2);
sort_bottomleft_corners_check_overlap_x = ...
    sort_bottomleft_corners_check_overlap_x ...
    (~isnan(sort_bottomleft_corners_check_overlap_x));
sort_bottomleft_corners_check_overlap_y = ...
    sort_bottomleft_corners_check_overlap_y(~isnan(...
    sort_bottomleft_corners_check_overlap_y));
sort_bottomleft_corners_check_overlap = ...
    [sort_bottomleft_corners_check_overlap_x, sort_bottomleft_corners_check_overlap_y];


sort_topright_corners_check_overlap_x = sort_topright_corners_check_overlap(:,1);
sort_topright_corners_check_overlap_y = sort_topright_corners_check_overlap(:,2);
sort_topright_corners_check_overlap_x = sort_topright_corners_check_overlap_x ...
```

```
50      (~isnan(sort_topright_corners_check_overlap_x));
51  sort_topright_corners_check_overlap_y = sort_topright_corners_check_overlap_y ...
52      (~isnan(sort_topright_corners_check_overlap_y));
53  sort_topright_corners_check_overlap = [sort_topright_corners_check_overlap_x, ...
54      sort_topright_corners_check_overlap_y];
55  sort_radii_check_overlap = sort_radii_check_overlap(~isnan(sort_radii_check_overlap));
56
57  sort_radii_check_overlap_DIM = length(sort_radii_check_overlap);
58
59  end
```

S1.2.5 Called Function 4 (getSatCutoff_v2)

```
%% 2.2.6 Unusable Electrodes with Fluorescence Artifacts were Removed from Analysis
% Figure 7h, Supplemental Figure 2xiv

function [FL_IMAGE_sat_copy, sat_markers_x, sat_markers_y, ...
    threshold_FROM_sat_cutoff] = getSatCutoff_v2(FL_IMAGE)

[rows_FL_IMAGE, cols_FL_IMAGE] = size(FL_IMAGE);
FL_IMAGE_sat_copy = FL_IMAGE;
sat_markers_x = [];
sat_markers_y = [];

use_for_sat_cutoff_FL_IMAGE = double(FL_IMAGE_sat_copy);
use_for_sat_cutoff_FL_IMAGE = use_for_sat_cutoff_FL_IMAGE(:);
use_for_sat_cutoff_FL_IMAGE(use_for_sat_cutoff_FL_IMAGE == 0) = NaN;
log_trans_for_sat_cutoff = log(use_for_sat_cutoff_FL_IMAGE);
log_trans_MEAN_for_sat_cutoff = nanmean(log_trans_for_sat_cutoff);
log_trans_VARIANCE_for_sat_cutoff = (nanstd(log_trans_for_sat_cutoff))^2;
MEAN_for_sat_cutoff = ...
    exp(log_trans_MEAN_for_sat_cutoff + log_trans_VARIANCE_for_sat_cutoff/2);
VARIANCE_for_sat_cutoff = ...
    exp(2*log_trans_MEAN_for_sat_cutoff + 2*log_trans_VARIANCE_for_sat_cutoff) - ...
    exp(2*log_trans_MEAN_for_sat_cutoff + log_trans_VARIANCE_for_sat_cutoff);
threshold_FROM_sat_cutoff = MEAN_for_sat_cutoff + 6*(VARIANCE_for_sat_cutoff)^0.5;

for k = 1:1:rows_FL_IMAGE
    for q = 1:1:cols_FL_IMAGE
        if FL_IMAGE(k,q) >= threshold_FROM_sat_cutoff
            FL_IMAGE_sat_copy(k,q) = 65535;
            sat_markers_x = [sat_markers_x, q];
            sat_markers_y = [sat_markers_y, k];
        else
        end
    end
end

end
```

S1.2.6 Called Function 5 (applyMask_v2)

```
%% 2.2.4 A Rigid Digital Mask Differentiated the Region-of-Interest from ...
% Background at Each Electrode
% Figure 7e, Supplemental Figure 2xvi

function [rows_stack, cols_stack, rows_image, cols_image, ...
    sub_image_matrix_fl_ROI_ONLY, sub_image_matrix_fl_BG_ONLY, ...
    sub_image_matrix_fl_DIM, radius_inner, radius_outer, square_center, ...
    ring_pixels, ring_pixels_proxy] = applyMask_v2(sub_image_matrix_fl, ...
    radii_proxy, square_edge_length)

[rows_stack, cols_stack] = ...
    size(sub_image_matrix_fl(:,:,1)); % determined empirically for 5X objective
radius_inner = radii_proxy - 28; % determined empirically for 5X objective
radius_outer = radii_proxy + 14; % determined empirically for 5X objective
square_center = [square_edge_length/2 + 1, square_edge_length/2 + 1];

[cols_image, rows_image] = meshgrid(1:cols_stack, 1:rows_stack);
ring_pixels = uint32(zeros(rows_stack, cols_stack));
array2D = (rows_image - square_center(1)).^2 + (cols_image - square_center(1)).^2;
ring_pixels_proxy = array2D >= radius_inner^2 & array2D <= radius_outer^2;
ring_pixels_proxy = uint32(ring_pixels_proxy);
ring_pixels = ring_pixels + ring_pixels_proxy;

[~,~, sub_image_matrix_fl_DIM] = size(sub_image_matrix_fl(1,1,:));
sub_image_matrix_fl_ROI_ONLY = sub_image_matrix_fl;
sub_image_matrix_fl_BG_ONLY = sub_image_matrix_fl;

for k = 1:sub_image_matrix_fl_DIM
    for m = 1:rows_stack
        for n = 1:cols_stack
            if ring_pixels(m,n) == 0
                sub_image_matrix_fl_ROI_ONLY(m,n,k) = 0;
            elseif ring_pixels(m,n) == 1
                sub_image_matrix_fl_BG_ONLY(m,n,k) = 0;
            else
            end
        end
    end
end

end
```

S1.2.7 Called Function 6 (getElectrodeStats)

```
%% 2.2.5 Local Noise Thresholds were Calculated from Background Pixels and ...
% Subtracted from all Pixels at Each Electrode
% Figure 7f, Supplemental Figure 2xvii-xx

function [STAT_LEVEL, STD_BG_array, MEAN_BG_array, threshold_FROM_BG_array, ...
    sub_image_matrix_fl_thresh] = getElectrodeStats(sub_image_matrix_fl_DIM, ...
    sub_image_matrix_fl, sub_image_matrix_fl_BG_ONLY)

STAT_LEVEL = 3;

STD_BG_array = uint32(zeros(1, sub_image_matrix_fl_DIM));
MEAN_BG_array = uint32(zeros(1, sub_image_matrix_fl_DIM));
threshold_FROM_BG_array = uint32(zeros(1, sub_image_matrix_fl_DIM));
sub_image_matrix_fl_thresh = sub_image_matrix_fl;

for k = 1:sub_image_matrix_fl_DIM
    use_for_stats_BG = double(sub_image_matrix_fl_BG_ONLY(:,:,k));
    use_for_stats_BG = use_for_stats_BG(:);
    use_for_stats_BG(use_for_stats_BG == 0) = NaN;
    log_trans_BG = log(use_for_stats_BG);
    log_trans_MEAN_BG = nanmean(log_trans_BG);
    log_trans_VARIANCE_BG = (nanstd(log_trans_BG))^2;
    MEAN_BG = exp(log_trans_MEAN_BG + log_trans_VARIANCE_BG/2);
    VARIANCE_BG = exp(2*log_trans_MEAN_BG + 2*log_trans_VARIANCE_BG) ...
        - exp(2*log_trans_MEAN_BG + log_trans_VARIANCE_BG);
    threshold_FROM_BG = MEAN_BG + STAT_LEVEL*(VARIANCE_BG)^0.5;
    MEAN_BG_array(k) = MEAN_BG;
    STD_BG_array(k) = (VARIANCE_BG)^0.5;
    threshold_FROM_BG_array(k) = threshold_FROM_BG;
    sub_image_matrix_fl_thresh(:,:,k) = ...
        sub_image_matrix_fl(:,:,k) - threshold_FROM_BG;
end

end
```

S1.2.8 Called Function 7 (elimSquaresAUTO_2)

```
%% 2.2.6 Unusable Electrodes with Fluorescence Artifacts were Removed from Analysis
% Figure 7h, Supplemental Figure 2xxii-xxiii

function [perc_sat_BG_matrix, squares_to_elim, sort_centers_check_elim, ...
    sort_radii_check_elim, sort_bottomleft_corners_check_elim, ...
    sort_topright_corners_check_elim, sub_image_matrix_fl_thresh_elim, ...
    counter_elim, sub_image_matrix_fl_elim] = ...
    elimSquaresAUTO_2(threshold_FROM_sat_cutoff, sub_image_matrix_fl_BG_ONLY, ...
    sub_image_matrix_fl_DIM, rows_stack, cols_stack, ring_pixels_proxy, ...
    sort_centers_check, sort_radii_check, sort_bottomleft_corners_check, ...
    sort_topright_corners_check, sub_image_matrix_fl_thresh, sub_image_matrix_fl)

proxy_counter_BG = 0;

for m = 1:rows_stack
    for n = 1:cols_stack
        if ring_pixels_proxy(m,n) == 0
            proxy_counter_BG = proxy_counter_BG + 1;
        else
        end
    end
end

perc_sat_BG_matrix = [];

for k = 1:sub_image_matrix_fl_DIM
    sat_cutoff_counter = 0;
    perc_sat_BG = 0;
    for m = 1:rows_stack
        for n = 1:cols_stack
            if sub_image_matrix_fl_BG_ONLY(m,n,k) >= threshold_FROM_sat_cutoff
                sat_cutoff_counter = sat_cutoff_counter + 1;
            else
            end
        end
    end

    perc_sat_BG = sat_cutoff_counter / proxy_counter_BG * 100;
    perc_sat_BG_matrix = [perc_sat_BG_matrix, perc_sat_BG];

end

counter_elim = 0;
squares_to_elim = [];

for k = 1:sub_image_matrix_fl_DIM
    if perc_sat_BG_matrix(k) >= 0.1
        counter_elim = counter_elim + 1;
        squares_to_elim = [squares_to_elim, k];
```

```
50        else
51        end
52    end
53
54    sort_centers_check_elim = sort_centers_check;
55    sort_radii_check_elim = sort_radii_check;
56    sort_bottomleft_corners_check_elim = sort_bottomleft_corners_check;
57    sort_topright_corners_check_elim = sort_topright_corners_check;
58    sub_image_matrix_fl_thresh_elim = double(sub_image_matrix_fl_thresh);
59    sub_image_matrix_fl_elim = double(sub_image_matrix_fl);
60
61    for a = 1:length(squares_to_elim)
62       for m = 1:rows_stack
63          for n = 1:cols_stack
64             sub_image_matrix_fl_thresh_elim(m,n,squares_to_elim(a)) = NaN;
65             sub_image_matrix_fl_elim(m,n,squares_to_elim(a)) = NaN;
66          end
67       end
68       sort_centers_check_elim(squares_to_elim(a),:) = NaN;
69       sort_radii_check_elim(squares_to_elim(a)) = NaN;
70       sort_bottomleft_corners_check_elim(squares_to_elim(a),:) = NaN;
71       sort_topright_corners_check_elim(squares_to_elim(a),:) = NaN;
72    end
73
74    end
```

S1.2.9 Called Function 8 (consistencyChecks_2)

```
%% 2.2.7 Fluorescence Intensity was Calculated from Usable Electrodes
% Figure 7i,  Supplemental Figure 2xxiv

function [CHECK_total_num_pix, pix_int_metric] = ...
    consistencyChecks_2(sub_image_matrix_fl_ROI_ONLY, sub_image_matrix_fl_BG_ONLY, ...
    rows_stack, cols_stack, sum_stack_norm_fl_thresh_elim)

[counts_ROI, bins_ROI] = imhist(uint16(sub_image_matrix_fl_ROI_ONLY(:,:,1)), 3000);
counts_ROI(1) = 0;
[counts_BG, bins_BG] = imhist(uint16(sub_image_matrix_fl_BG_ONLY(:,:,1)), 3000);
counts_BG(1) = 0;
total_num_pix_from_stats = sum(counts_ROI) + sum(counts_BG);
total_num_pix_in_square = rows_stack*cols_stack;
CHECK_total_num_pix  = total_num_pix_from_stats - (rows_stack*cols_stack);
pix_int_metric = sum(sum(sum_stack_norm_fl_thresh_elim))/total_num_pix_in_square;

end
```

S1.2.10 Called Function 9 (confidenceInterval)

```
%% 2.2.5 Local Noise Thresholds were Calculated from Background Pixels and ...
% Subtracted from all Pixels at Each Electrode
% Figure 7g, Supplemental Figure 2xxi

%% 2.2.7 Fluorescence Intensity was Calculated from Usable Electrodes
% Figure 7i, Supplemental Figure 2xxv

function [scores, sample_average, population_std_estimate, confidence_level, ...
    uncertainty] = confidenceInterval(sub_image_matrix_fl_thresh_elim)

[~,~, n] = size(sub_image_matrix_fl_thresh_elim);
scores = zeros(n,2);

for c = 1:n
    electrode_thresh = sub_image_matrix_fl_thresh_elim(:,:,c);
    electrode_thresh = electrode_thresh(:);
    electrode_thresh_avg = mean(electrode_thresh);
    electrode_thresh_std = std(electrode_thresh);
    scores(c,1) = electrode_thresh_avg;
    scores(c,2) = electrode_thresh_std;
end

sample_average = nanmean(scores(:,1));
population_std_estimate = nanstd(scores(:,1));
confidence_level = 1.96;
uncertainty = (confidence_level)*(population_std_estimate)/(n^(1/2));

end
```