**Computer code (Matlab Mathworks)**

***% The following is an image processing function that takes signal from both channels of a dLGN***

***% image and signal and background masks to calculate percentage overlap. The thresholds***

***% increase by 2.5% per step. If start\_threshold\_num = 4 and threshold\_num = 7, then the***

***% starting threshold is 4\*2.5 = 10%, and increase by 2.5% each time for 6 more thresholds,***

***% and the highest threshold is 25%.***

function [ipsi\_area\_vector, overlap\_area\_vector, roi\_area, section\_overlap\_fraction\_vector, log\_ratio\_vector, log\_ratio\_variance] = dLGNprocessing(contra\_raw\_int, ipsi\_raw\_int, sg\_mask, bg\_mask)

threshold\_num = 7;

start\_threshold\_num = 4;

contra\_raw = double(contra\_raw\_int);

ipsi\_raw = double(ipsi\_raw\_int);

sg\_indicator = sg\_mask > 0;

bg\_indicator = bg\_mask > 0;

roi\_area = sum(sum(sg\_indicator));

% create indicator matrices using masks for background region and signal

% region.

contra\_bg = mean(contra\_raw(bg\_indicator));

ipsi\_bg = mean(ipsi\_raw(bg\_indicator));

% calculate background values of contra and ipsi channels.

contra\_bg\_rmv = contra\_raw - contra\_bg;

contra\_bg\_rmv\_crt = contra\_bg\_rmv .\* (contra\_bg\_rmv > 0);

contra\_bg\_rmv\_crt\_norm = contra\_bg\_rmv\_crt / max(contra\_bg\_rmv\_crt(sg\_indicator));

ipsi\_bg\_rmv = ipsi\_raw - ipsi\_bg;

ipsi\_bg\_rmv\_crt = ipsi\_bg\_rmv .\* (ipsi\_bg\_rmv > 0);

ipsi\_bg\_rmv\_crt\_norm = ipsi\_bg\_rmv\_crt / max(ipsi\_bg\_rmv\_crt(sg\_indicator));

% create versions of both channels that have background subtracted, with

% values lower than background set to zero, and also normalized to the max

% values in the signal regions.

overlap\_area\_vector = zeros(1,threshold\_num);

ipsi\_area\_vector = zeros(1,threshold\_num);

% create a vector recording overlap areas. The element in the nth column

% is the area in the region of interest that has values from both channels

% exceeding n\*2.5 percent of the max value.

for n = start\_threshold\_num:(start\_threshold\_num + threshold\_num - 1)

overlap\_area\_vector(n - start\_threshold\_num + 1) = sum(sum( (contra\_bg\_rmv\_crt\_norm >= n\*0.025) & (ipsi\_bg\_rmv\_crt\_norm >= n\*0.025) & sg\_indicator ));

ipsi\_area\_vector(n - start\_threshold\_num + 1) = sum(sum((ipsi\_bg\_rmv\_crt\_norm >= n\*0.025) & sg\_indicator));

end

section\_overlap\_fraction\_vector = overlap\_area\_vector / roi\_area;

contra\_non\_zero = contra\_raw + (contra\_raw == 0);

ipsi\_non\_zero = ipsi\_raw + (ipsi\_raw == 0);

contra\_norm = contra\_non\_zero / max(contra\_non\_zero(sg\_indicator));

ipsi\_norm = ipsi\_non\_zero / max(ipsi\_non\_zero(sg\_indicator));

contra\_to\_ipsi\_log\_ratio = log( contra\_norm ./ ipsi\_norm );

log\_ratio\_vector = contra\_to\_ipsi\_log\_ratio(sg\_indicator);

log\_ratio\_variance = var(log\_ratio\_vector);

end

***% The following is a script that calls the function above and calculate percentage overlap for a group of***

***% animals. This script assumes that the image files are named by first labeling whether is a signal mask,***

***% a background mask, or a raw image of contra/ipsilateral signal, then specifying genotype, age, side of***

***% the brain, then specifying animal number and section number. The user can specify from which***

***% section the analysis starts and how many sections to analyze. The percentage overlap across multiple***

***% thresholds in multiple animals will be stored in total\_overlap\_fraction\_vector.***

sgmask\_file\_format = 'sgmask\_%s\_%s\_%s\_%d\_%d.tif';

bgmask\_file\_format = 'bgmask\_%s\_%s\_%s\_%d\_%d.tif';

contra\_file\_format = 'contra\_%s\_%s\_%s\_%d\_%d.tif';

ipsi\_file\_format = 'ipsi\_%s\_%s\_%s\_%d\_%d.tif';

threshold\_num = 7;

genotype = 'homko';

age = 'p10';

side = 'l';

animal\_number = 6;

section\_number = 7;

start\_section = 2;

section\_overlap\_fraction\_matrix = zeros(section\_number, threshold\_num);

section\_overlap\_area\_matrix = zeros(section\_number, threshold\_num);

section\_ipsi\_area\_matrix = zeros(section\_number, threshold\_num);

section\_roi\_area\_vector = zeros(section\_number, 1);

pooled\_log\_ratio\_vector = [];

log\_ratio\_variance\_vector = zeros(section\_number, 1);

for n = start\_section:(start\_section + section\_number - 1)

sgmask\_file = sprintf(sgmask\_file\_format, genotype, age, side, animal\_number, n);

bgmask\_file = sprintf(bgmask\_file\_format, genotype, age, side, animal\_number, n);

contra\_file = sprintf(contra\_file\_format, genotype, age, side, animal\_number, n);

ipsi\_file = sprintf(ipsi\_file\_format, genotype, age, side, animal\_number, n);

[section\_ipsi\_area\_matrix(n,:), section\_overlap\_area\_matrix(n,:), section\_roi\_area\_vector(n), section\_overlap\_fraction\_matrix(n,:), log\_ratio\_vector, log\_ratio\_variance\_vector(n)]...

= dLGNprocessing(imread(contra\_file), imread(ipsi\_file), imread(sgmask\_file), imread(bgmask\_file));

pooled\_log\_ratio\_vector = [pooled\_log\_ratio\_vector; log\_ratio\_vector];

end

total\_overlap\_volume\_vector = sum(section\_overlap\_area\_matrix);

total\_roi\_volume = sum(section\_roi\_area\_vector);

total\_ipsi\_volume\_vector = sum(section\_ipsi\_area\_matrix);

total\_overlap\_fraction\_vector = total\_overlap\_volume\_vector / total\_roi\_volume;

total\_overlap\_of\_ipsi\_fraction\_vector = total\_overlap\_volume\_vector ./ total\_ipsi\_volume\_vector;