12/8/2022

Project 8

BME 7450

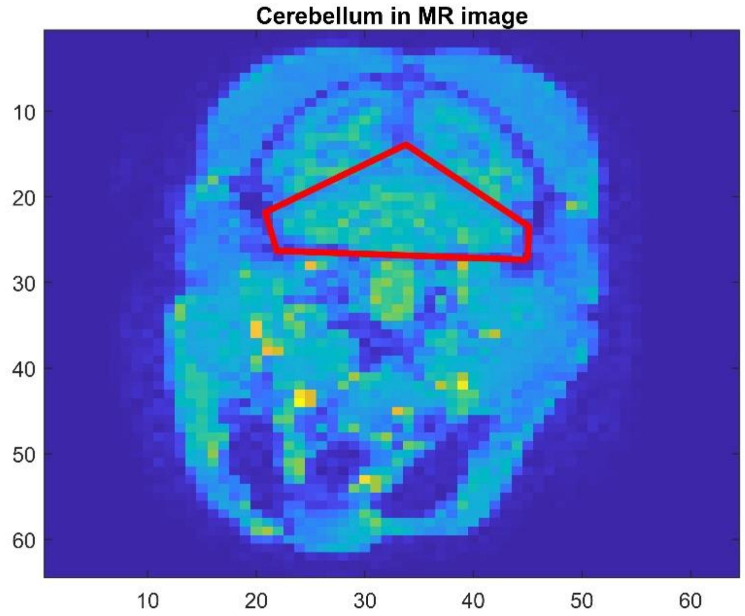
Submitted by,

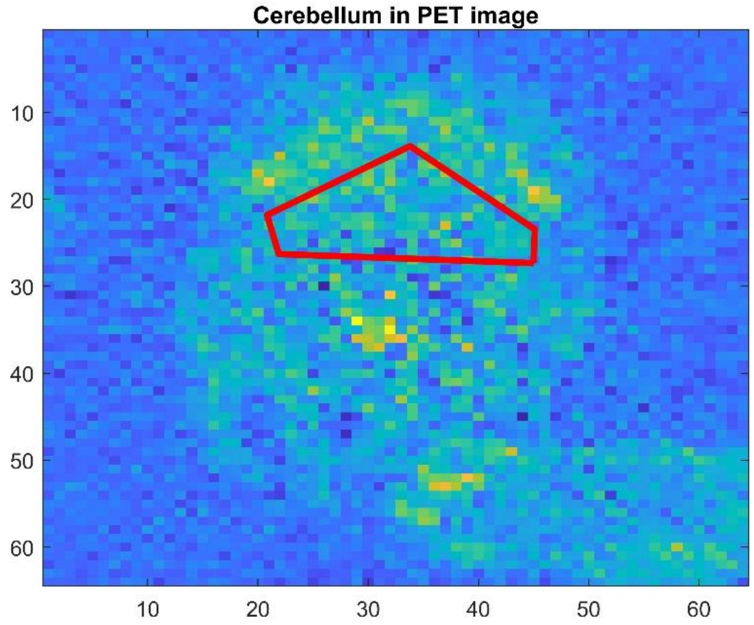
Rana Mozumder

DEPARTMENT OF BME, VANDERBILT UNIVERSITY

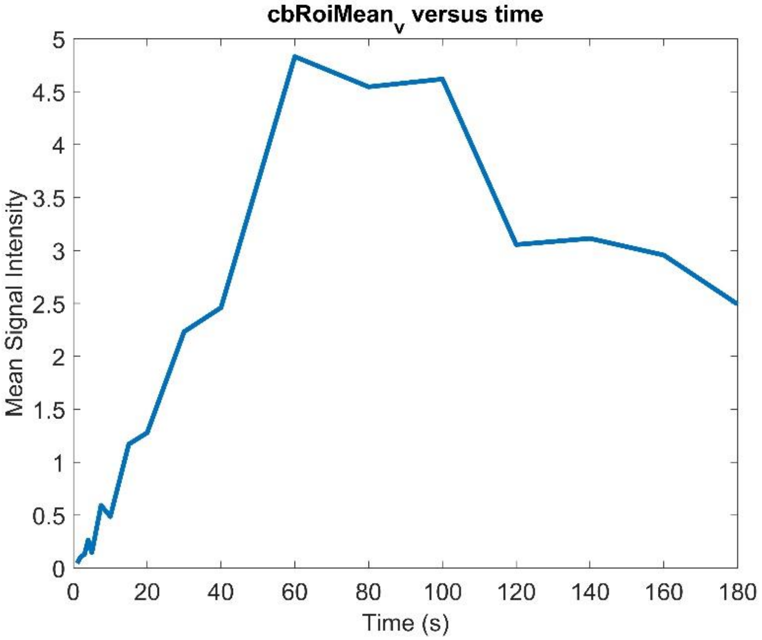
VANDERBILT
UNIVERSITY

- **Figure showing the location of the cerebellum ROI on both MRI and mean PET images**

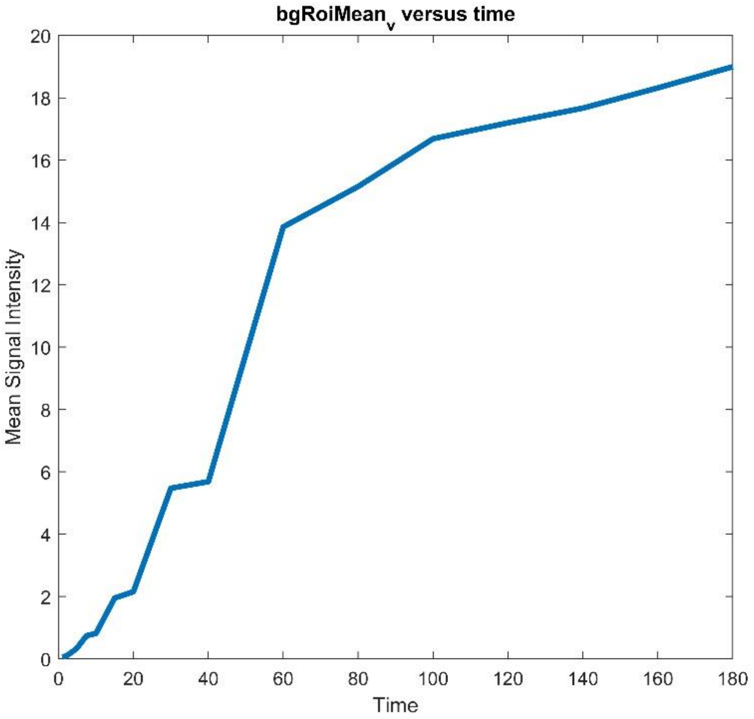




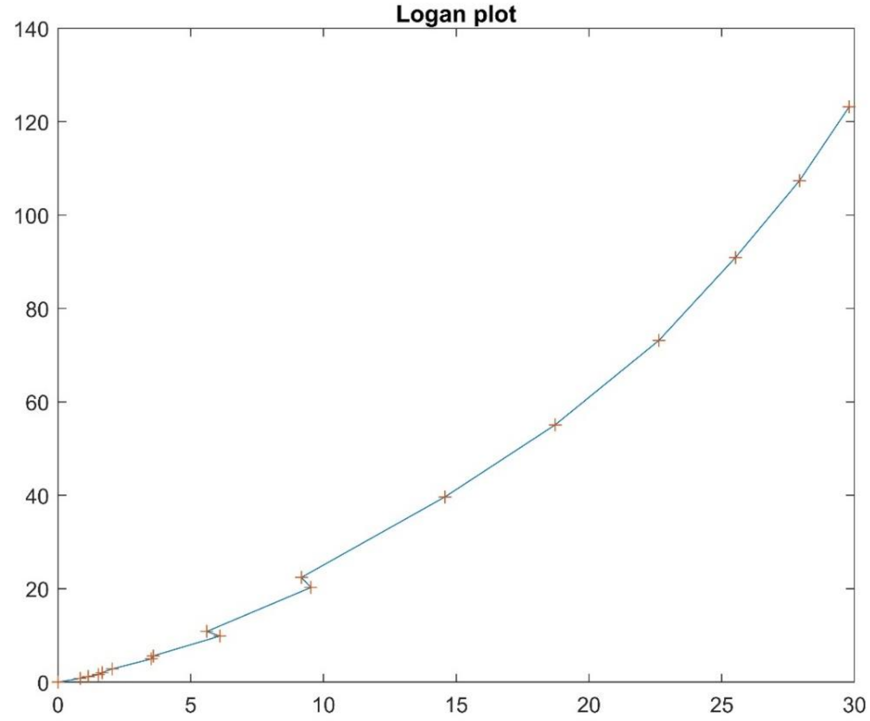
- **Plot of the cerebellum ROI mean pixel intensity versus time**



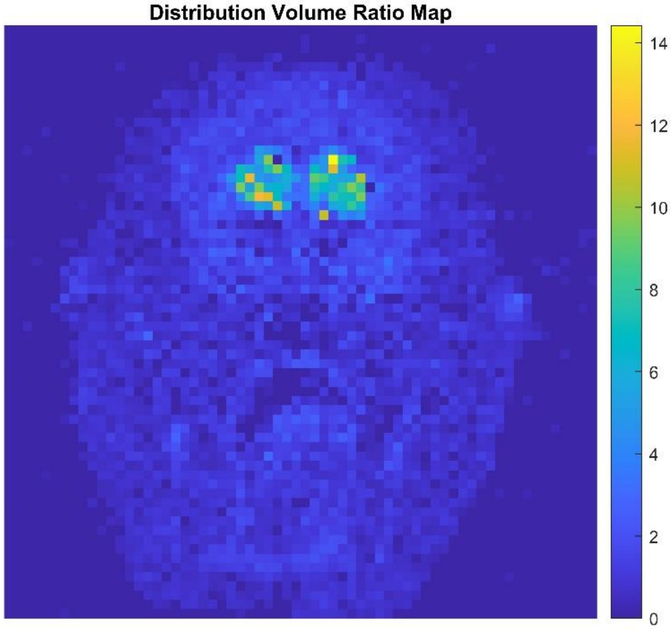
- **Plot of the basal ganglia ROI mean pixel intensity versus time**



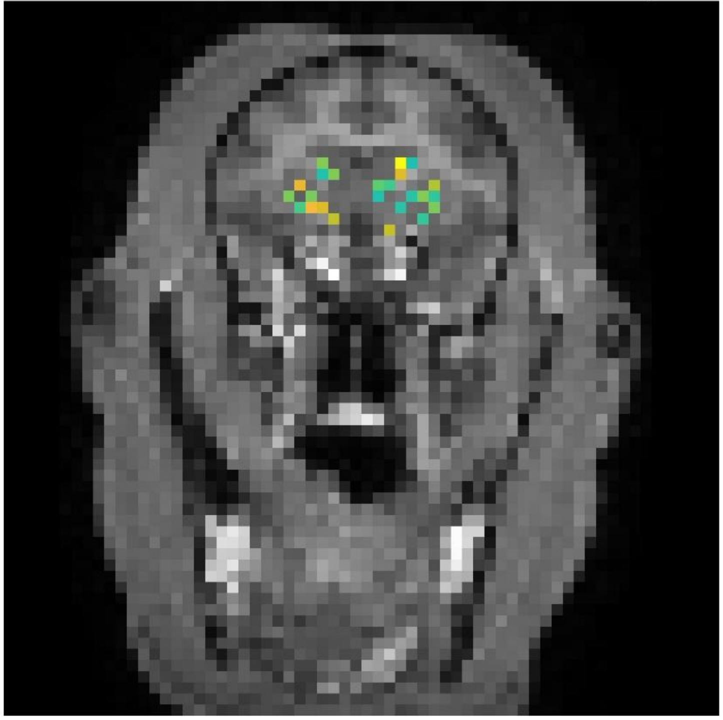
- **Logan plot for the basal ganglia ROI data**



- **DVR map and thresholded DVR values in the MRI anatomical image**



Thresholded DVR Values in MRI Anatomical Image



Questions:

1. *The plot of mean pixel intensity versus time is called a time-activity curve. The time-activity curve for the cerebellum ROI rises and falls, but the curve for the basal ganglia rises to a plateau (and stays high). Why do these regions have such different behavior?*

   Answer:

   In cerebellum, there is no dopamine binding site. Hence, when the agent is injected, its concentration goes up. However, since there is no receptor for the agent, it gets washed away with blood. Thus, the concentration again goes down.

   In contrast, basal ganglia have dopamine binding sites. So, when the agent is injected, the concentration goes up and stays high, because the agent binds with the receptors. So, they don't get washed away with blood.

2. *Why isn't the Logan plot linear for all times? What determines the time at which it becomes linear?*

   Answer:

   Its because of the $C_R(t)/C_T(t)$ ratio where C_R is the agent concentration in reference region and C_T is the agent concentration in tissue. Since at first both the concentrations vary with time, so we get non-linear points. However, at long times, both the concentration in tissue and the concentration in the reference region reaches steady state. Hence, the $C_R(t)/C_T(t)$

ratio becomes constant, and the equation that produces the logan plot can be fit to a straight line. That's why at long times, we get linear points. So, this depends on how quickly the concentration in both tissue and reference region reaches the steady state.

3. *Can you see any structure in the peak of the DVR maps (or is the peak a uniform blob)? If you can distinguish separate 'hot spots' in the basal ganglia (in each hemisphere), what anatomical structures do you think these correspond to?*

   Answer:

   Yes, there are two distinct regions in the DVR maps. According to their location in the brain, I think they are caudate nucleus and putamen of the basal ganglia.

4. *What changes in the data acquisition or analysis would improve the accuracy of your results?*

   Answer:

   - Here, in the basal ganglia, we can see that the concentration doesn't achieve the steady state entirely. We have images taken with in 3 mins. So, I think it'd be better if we could have some more images until the concentration reaches steady state completely.
   - Resolution of the images could also be improved for better result.
   - Here I fit a straight line with the last four points of the Logan plot. It'd be better if we could fit to more points. It'd give us a better measurement of slope. Thus, the DVR map would be much more accurate.

Matlab Code:

```
%%
clc; close all; clear all;
load('proj8Data.mat'); %loading data

%%
time_v = cumsum(dTime_v); % Time at the completion of each image
nTimes = length(time_v);

%%
% Displaying PET (and MRI) images at each time point:
figure
bgImageMax = max(bgPetImage_3d(:));

% Making 3D arrays of identical pages to display MR images in grayscale:
bgMrColor_3d = cat(3, bgMrImage_m, bgMrImage_m, bgMrImage_m)/max(bgMrImage_m(:));
cbMrColor_3d = cat(3, cbMrImage_m, cbMrImage_m, cbMrImage_m)/max(cbMrImage_m(:));

for timeIndex = 1:nTimes
    % Showing PET image of basal ganglia:
    subplot(2, 2, 1)
    bgPetImage_m = squeeze(bgPetImage_3d(:, :, timeIndex));
    imagesc(bgPetImage_m)
    set(gca, 'CLim', [0, bgImageMax])
    axis image; axis off
    title(['Time index = ', num2str(timeIndex)])
    % Showing PET image of cerebellum:
    subplot(2, 2, 2)
    cbPetImage_m = squeeze(cbPetImage_3d(:, :, timeIndex));
    imagesc(cbPetImage_m)
    set(gca, 'CLim', [0, bgImageMax])
    axis image; axis off
    % Showing MR image of basal ganglia:
    subplot(2, 2, 3)
    image(bgMrColor_3d)
    axis image; axis off
    % Showing MR image of cerebellum:
    subplot(2, 2, 4)
    image(cbMrColor_3d)
    axis image; axis off
    pause(1)
end

%%
figure
imagesc(cbMrImage_m)
title('Define cerebellum')
[cbRoiMask_m, cb_x, cb_y] = roipoly;
line(cb_x, cb_y, 'Color', 'r', 'LineWidth', 3)

figure
imagesc(mean(cbPetImage_3d, 3))
title('Cerebellum in PET image')
hold on
line(cb_x, cb_y, 'Color', 'r', 'LineWidth', 3)

%%
for time=1:nTimes
    cbRoiMean_v (1, time) = sum(cbRoiMask_m.*squeeze(cbPetImage_3d(:,:,time)),
'all')...
        /sum(cbRoiMask_m, 'all');
end

%%
figure
plot(time_v, cbRoiMean_v, 'LineWidth', 3)
title('cbRoiMean_v versus time')
xlabel('Time')
ylabel('Mean Signal Intensity')

%%
figure
imagesc(mean(bgPetImage_3d, 3))
title('Define Basal Ganglia')
[bgRoiMask1_m, bg_x, bg_y] = roipoly;
line(bg_x, bg_y, 'Color', 'r', 'LineWidth', 3)
hold on
[bgRoiMask2_m, bg_x, bg_y] = roipoly;
line(bg_x, bg_y, 'Color', 'r', 'LineWidth', 3)
bgRoiMask_m = bgRoiMask1_m + bgRoiMask2_m;

%%
for time=1:nTimes
    bgRoiMean_v (1, time) = sum(bgRoiMask_m.*squeeze(bgPetImage_3d(:,:,time)),
'all')...
        /sum(bgRoiMask_m, 'all');
end

figure
plot(time_v, bgRoiMean_v, 'LineWidth', 3)
```

```
title('bgRoiMean_v versus time')
xlabel('Time')
ylabel('Mean Signal Intensity')

%%
for time=1:nTimes
    if time == 1
        bgRoiX_v(1, time) = 0;
        bgRoiY_v(1, time) = 0;
    else
        bgRoiX_v(1, time) = trapz(time_v(1, 1:time), cbRoiMean_v(1,
1:time))/bgRoiMean_v(1, time);
        bgRoiY_v(1, time) = trapz(time_v(1, 1:time), bgRoiMean_v(1,
1:time))/bgRoiMean_v(1, time);
    end
end

figure
plot(bgRoiX_v, bgRoiY_v, '-', bgRoiX_v, bgRoiY_v, '+')

title('Logan plot')

%%
nFitPoints = 4;
meanImage = mean(bgPetImage_3d, 3);
maxPix = max(meanImage(:));
mask = zeros(64, 64);
for row=1:64
    for col=1:64
        if mean(squeeze(bgPetImage_3d(row, col,:)))>=0.05*maxPix
            mask(row, col) = 1;
        end
    end
end
figure
imagesc(mask)


%%
dvr_m = zeros(64, 64);
for row=1:64
    for col=1:64
        if mask(row, col) == 1
            pixels = squeeze((bgPetImage_3d(row, col,:)))';
            for time=1:nTimes
                bgRoiX1_v(1, time) = trapz(time_v(1, 1:time), cbRoiMean_v(1, 1:time),
2)/pixels(1, time);
                bgRoiY1_v(1, time) = trapz(time_v(1, 1:time), pixels(1, 1:time),
2)/pixels(1, time);
            end
            X = bgRoiX1_v(1, (end-nFitPoints+1): end);
            Y = bgRoiY1_v(1, (end-nFitPoints+1): end);
            s = polyfit(X, Y, 1);
            % Check for legal values:
            if (s(1) > 0)
                dvr_m(row, col) = s(1); % DVR = slope.
            else
                dvr_m(row, col) = 0;
            end
        end
    end
```

```
end


%%
% Displaying DVR map:
figure
imagesc(dvr_m)
colorbar
axis image
axis off
title('Distribution Volume Ratio Map')
colorMap_m = colormap;
nColors = size(colorMap_m, 1);
% Showing thresholded DVR on MRI:
figure
% Calculating a mask where the DVR values
% exceed half their maximum value. Name the mask dvrMask_m:
dvrMask_m = zeros(64, 64);
max_pix = max(dvr_m(:));
for row=1:64
    for col = 1:64
        if dvr_m(row, col)>0.5*max_pix
            dvrMask_m(row, col) = 1;
        end
    end
end


% Rendering pixels inside dvrMask in color. Use same colormap as above:
colorIndex_v = 1 + round((nColors-1) * dvr_m(:) / max(dvr_m(:)));
redDvr_m = dvrMask_m .* reshape(colorMap_m(colorIndex_v, 1), size(dvrMask_m));
greenDvr_m = dvrMask_m .* reshape(colorMap_m(colorIndex_v, 2), size(dvrMask_m));
blueDvr_m = dvrMask_m .* reshape(colorMap_m(colorIndex_v, 3), size(dvrMask_m));
% Displaying the rest of the image in gray:
maskedMrImage_m = (1-dvrMask_m) .* bgMrImage_m / max(bgMrImage_m(:));
color_3d = cat(3, maskedMrImage_m + redDvr_m, maskedMrImage_m + greenDvr_m,
maskedMrImage_m + blueDvr_m);
image(color_3d)
axis image
axis off
title('DVR map')
```