How to Display Perfusion Maps
with PMA Colormaps

Yao Xiao
SMILE Lab, UF
2/18/2019

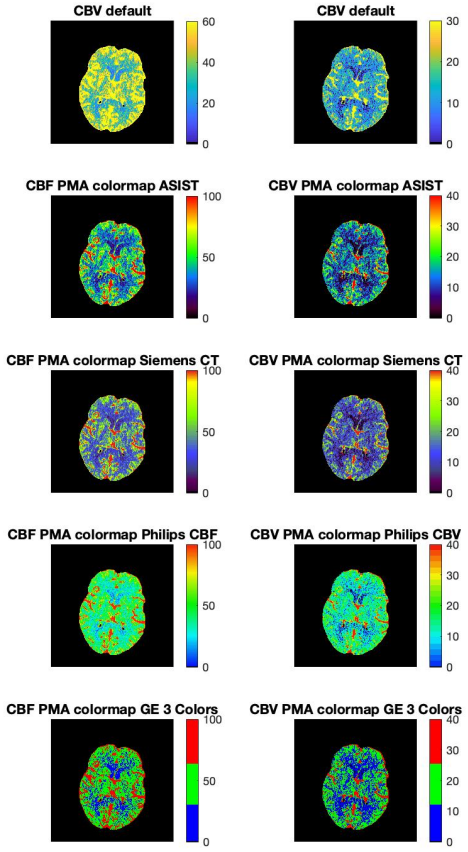
Assume we already calculated the perfusion maps through some methods. Let's call the generated perfusion maps as CBF and CBV (2D matrix).

Requirements: (everything is in the folder /Display_PMA_Colormaps_Yao)

- Full PMA Color Lookup Table (CLT) from PMA software, *PMA_lut.csv*
- Function *select_colormap.m* % select the PMA colormap that you want to display
- Function *ctshow_pma.m* % display images with defined colormap
- *CBF.mat*, *CBV.mat*, *mask.mat* % required data for displaying (pre-calculated)

Example:
Run *example_ctshow_pma_colormap_full.m*

Resulted image:



PMA Software: http://asist.umin.jp/index-e.htm

Code details:

```
%% example_ctshow_pma_colormap_full.m
%
% Display perfusion maps by using PMA colormaps
% 1) Load the full PMA color lookup table
% 2) Select the Colormap that want to use
% 3) Load precalcualted perfusion maps, CBF, CBV
% 4) Display images with different color maps
%
% Yao Xiao
% Feb 14, 2019 @ SMILE | UF
%
%% settings
close all; clear; clc;

addpath(genpath(cd));

path_data = './data';
path_save = './results';

if ~exist(path_save,'dir'), mkdir(path_save); end

%% load the full PMA color lookup table
clt_pma = readtable(fullfile(path_data,'PMA_lut.csv'));

%% select colormaps from the full table
CLT_ASIST = select_colormap(clt_pma,'ASIST');
CLT_Siemens_CT = select_colormap(clt_pma,'Siemens_CT');
CLT_Philips_CBF = select_colormap(clt_pma,'Philips_CBF');
CLT_Philips_CBV = select_colormap(clt_pma,'Philips_CBV');
CLT_GE_3_Colors = select_colormap(clt_pma,'GE_3_Colors');

%% load precalculated perfusion maps
load(fullfile(path_data,'CBF.mat')); % precalculated CBF
load(fullfile(path_data,'CBV.mat')); % precalculated CBV
load(fullfile(path_data,'mask.mat')); % precalculated mask

%% display images
figure;

% default colormap
p1 = subplot(5,2,1);
[~,cm1,~] = ctshow_pma(CBF,mask,[0 60],'default');
title('CBV default');

p2 = subplot(5,2,2);
[~,cm2,~] = ctshow_pma(CBV,mask,[0 30],'default');
title('CBV default');

% PMA colormap ASIST
p3 = subplot(5,2,3);
[~,cm3,~] = ctshow_pma(CBF,mask,[0 100],'pma',CLT_ASIST);
title('CBF PMA colormap ASIST');

p4 = subplot(5,2,4);
[~,cm4,~] = ctshow_pma(CBV,mask,[0 40],'pma',CLT_ASIST);
```

```
title('CBV PMA colormap ASIST');

% PMA colormap Siemens_CT
p5 = subplot(5,2,5);
[~,cm5,~] = ctshow_pma(CBF,mask,[0 100],'pma',CLT_Siemens_CT);
title('CBF PMA colormap Siemens CT');

p6 = subplot(5,2,6);
[~,cm6,~] = ctshow_pma(CBV,mask,[0 40],'pma',CLT_Siemens_CT);
title('CBV PMA colormap Siemens CT');

% PMA colormap Philips_CBF & Philips_CBV
p7 = subplot(5,2,7);
[~,cm7,~] = ctshow_pma(CBF,mask,[0 100],'pma',CLT_Philips_CBF);
title('CBF PMA colormap Philips CBF');

p8 = subplot(5,2,8);
[~,cm8,~] = ctshow_pma(CBV,mask,[0 40],'pma',CLT_Philips_CBV);
title('CBV PMA colormap Philips CBV');

% PMA colormap PGE_3_Colors
p9 = subplot(5,2,9);
[~,cm9,~] = ctshow_pma(CBF,mask,[0 100],'pma',CLT_GE_3_Colors);
title('CBF PMA colormap GE 3 Colors');

p10 = subplot(5,2,10);
[~,cm10,~] = ctshow_pma(CBV,mask,[0 40],'pma',CLT_GE_3_Colors);
title('CBV PMA colormap GE 3 Colors');


% display color for different maps
colormap(p1,cm1); % default
colormap(p2,cm2); % default
colormap(p3,cm3); % ASIST
colormap(p4,cm4); % ASIST
colormap(p5,cm5); % Siemens_CT
colormap(p6,cm6); % Siemens_CT
colormap(p7,cm7); % Philips_CBF
colormap(p8,cm8); % Philips_CBV
colormap(p9,cm9); % PGE_3_Colors
colormap(p10,cm10); % PGE_3_Colors

% save image to file
saveas(gcf,fullfile(path_save,'result.jpg'));
saveas(gcf,fullfile(path_save,'result.fig'));



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function CLT = select_colormap(clt_select)
% obtain CLT (PMA colormap) from the enntire color lookup table
%
% OUTPUTS:
%   CLT     - Color Lookup Table
%
% INPUTS:
```

```
%   clt_select  - Colormap Name in PMA
%
% 2/17/2019 - Yao Xiao @ SMILE | UF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function CLT = select_colormap(clt_pma,clt_select)

switch clt_select

    case 'ASIST'
        clt_col = clt_pma.ASIST;

    case 'Blue_Red'
        clt_col = clt_pma.Blue_Red;

    case 'Gray'
        clt_col = clt_pma.Gray;

    case 'Inv_Gray'
        clt_col = clt_pma.Inv_Gray;

    case 'AJS'
        clt_col = clt_pma.AJS;

    case 'AZE'
        clt_col = clt_pma.AZE;

    case 'GE_3_Colors'
        clt_col = clt_pma.GE_3_Colors;

    case 'GE_Puh_Thalium'
        clt_col = clt_pma.GE_Puh_Thalium;

    case 'GE_Rainbow'
        clt_col = clt_pma.GE_Rainbow;

    case 'GE_Inv_Rainbow'
        clt_col = clt_pma.GE_Inv_Rainbow;

    case 'Hitachi_Block'
        clt_col = clt_pma.Hitachi_Block;

    case 'Hitachi_Pallete'
        clt_col = clt_pma.Hitachi_Pallete;

    case 'LAS'
        clt_col = clt_pma.LAS;

    case 'Philips_CBF'
        clt_col = clt_pma.Philips_CBF;

    case 'Philips_CBV'
        clt_col = clt_pma.Philips_CBV;

    case 'Philips_MTT'
```

```
        clt_col = clt_pma.Philips_MTT;

    case 'Philips_TTP'
        clt_col = clt_pma.Philips_TTP;

    case 'Siemens_CT'
        clt_col = clt_pma.Siemens_CT;

    case 'Siemens_MR'
        clt_col = clt_pma.Siemens_MR;

    case 'Terarecon'
        clt_col = clt_pma.Terarecon;

    case 'Toshiba_CT_Rainbow'
        clt_col = clt_pma.Toshiba_CT_Rainbow;

    case 'Toshiba_CT_Rainbow_Red'
        clt_col = clt_pma.Toshiba_CT_Rainbow_Red;

    case 'Toshiba_MRI'
        clt_col = clt_pma.Toshiba_MRI;

    case 'ZIO_NIH'
        clt_col = clt_pma.ZIO_NIH;

    case 'ZIO_TR'
        clt_col = clt_pma.ZIO_TR;

    case 'ZIO_TRW'
        clt_col = clt_pma.ZIO_TRW;

    case 'Tmax'
        clt_col = clt_pma.Tmax;
end

% extract CLT from the entire color lookup table
CLT = zeros(256,3);
CLT(:,1) = clt_col(1:256);
CLT(:,2) = clt_col(259:514);
CLT(:,3) = clt_col(517:772);
CLT = CLT./256;

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%function [c,cm,cb] = ctshow_pma(im, mask, c, cflag, CLT)
% displays CT image by using CLT (PMA colormap) and rescaling the region in
the mask
%
% OUTPUTS:
%   c       - Output gray value range to be displayed
%   cm      - Output colormap
%   cb      - Output colorbar
```

```
%
% INPUTS:
%   im      - Images to be scaled and displayed
%   mask    - region of interest in the image to be scaled and displayed
%   c       - User specified gray value range to be displayed
%   cflag   - Indicator for graylevel/color display
%               default:default colormap, pma:PMA colormap
%   CLT     - Color Lookup Table
%
% 2/14/2019 - Yao Xiao @ SMILE | UF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [c,cm,cb] = ctshow_pma(im,mask,c,cflag,CLT)

%Read inputs
if nargin < 2 || isempty(mask)
    mask = true(size(im,1),size(im,2));
end
if nargin < 3 || isempty(c)
    thresh = 0.05;
    [n,xout] = hist(double(im(mask&im>0)),1000);
    r = find(n >= max(n) * thresh);
    c = real([xout(r(1)) max(xout(r(end)),xout(r(1)+1))]);
end
if nargin < 4
    cflag = 'default';
end

%Show image
im(~mask) = 0;
if ismatrix(im)
    imshow(im,c);
else
    im = permute(im,[1 2 4 3]);
    montage(im,'DisplayRange',c);
end
axis tight;
axis equal;

switch cflag
    case 'default'
        colormap('default');
        cmap = colormap;
        cmap(1,:) = [0 0 0];
        cm = colormap(cmap);
        cb = colorbar;

    case 'pma'
        cm = colormap(CLT);
        cb = colorbar;
end

end
```