MATLAB code for scripts used in single molecule experiments

1) Extracting traces from video files

```
function TraceExtractor
%Acknowledge:
%We are grateful for the community on MathWorks forum for providing us
technical code solutions for certain challenges.
%Particularly we acknowledge here Tristan Ursell and his script for "Read
large image stack (TIFF)" used in our script.

%Videos pre-processing:
%1. before trace extraction we subtract the background using ImageJ
function "Process/Subtract Background" with rolling ball radius set to 5
%2. make sure that the first frame of the video is time projection image to
easily locate the spots (we create time projection image using ImageJ
function "Image/Stacks/Z project/STD")

%Post-processing:
%1. make sure that the first time points in extracted traces are removed
(since they are coming from time projection image used to locate spots).

close all
clc
XYN=[];
deltat=0.052; %time between frames

    path=uigetdir('C:\Users\'); % put videos source folder
    path0='C:\Users\'; % put save trace folder

    % load entire video stack
    fnamebasis=input('Filenamebasis of video file (default: "data00")',
's');
    if isempty(fnamebasis);fnamebasis='data00';end
    fnameo=input('Sequence number of video file (default: "0")', 's');
    if isempty(fnameo);fnameo='0';end
    fname2=[path '\' fnamebasis fnameo '.tif'];
    [stack_out,Nframes]=imread_big(fname2); %credits to: Tristan Ursell,
Read large image stack (TIFF),Feb 2017, code is available in MathWorks
forum

   % adjust histogram
    mip10 = stack_out(:,:,1);
    figure(1)
    imhist(mip10)
    maxhist=input('Max for histogram stretching? (default=255 [0, 255]) ');
    if isempty(maxhist);maxhist=255;end
    mip10h = imadjust(mip10,[0 maxhist/255],[0 1]);
    figure(2);
    im=image(mip10h);
    MAP=gray (256);
    colormap(MAP);
    title(['Frames: ' int2str(Nframes)]);
    uistack(im,'bottom');
    hold on;

    for iii=1:1000 % maximal 1000 AOI per step! Adjust if there is more!
          diffx=3;
          diffy=3;
```

```
          [X,Y]=ginput(1)
          XY=[floor(X),floor(Y)];
          X_ROI=[floor(X)-diffx/2 floor(X)-diffx/2 floor(X)+diffx/2
floor(X)+diffx/2 floor(X)-diffx/2];
          Y_ROI=[floor(Y)-diffy/2 floor(Y)+diffy/2 floor(Y)+diffy/2
floor(Y)-diffy/2 floor(Y)-diffy/2];
          plot(floor(X),floor(Y),'r.','MarkerSize', 10);
          plot(X_ROI, Y_ROI,'b-');
          hold on;
          if X<0;
              break
          end
          if X>0
              if isempty(XYN);
                  XYN(:,:,1)=XY;
              else
                  XYN(:,:,size(XYN,3)+1)=XY;
                  XYN
              end
          end
    end

    ndots=size(XYN,3)

    for num=1:ndots % loop over selected dots
        XY=XYN(:,:,num);
        X0=XY(1);
        Y0=XY(2);

        for frnr=1:Nframes % loop over all frames
            time=(frnr-1)*deltat;
            XX=stack_out(:,:,frnr);
            ZZ=double(XX);
            rect=[X0-1.5 Y0-1.5 2.5 2.5];
            YY2=imcrop(ZZ,rect);
            imint=mean2(YY2);

            if num==1 % for first dot processed add time coloumn (first
column)
                totintc(frnr,1)=time;
            end

            totintc(frnr,num+1)=imint; %average total intesity ROI=3x3

        end
    end


     save([path0 'intc vs t' fnameo '.mat'],'totintc');

end
```

2) Collect dwell times

```
function DwellTimeCollector

    path0='C:\Users\'
    % Load the traces file (obtained with TraceExtractor)
    fnamebasis2=input('Filenamebasis for center trace (default: "intc vs
t")', 's');
    if isempty(fnamebasis2);fnamebasis2='intc vs t';end
    fnameo2=input('Trace sequence number (default: "0")?', 's');
    if isempty(fnameo2);fnameo2='0';end
    loadname2=[path0 fnamebasis2 fnameo2 '.mat'];

    load(loadname2,'-mat');

    %Put folder path for results
    pathR='C:\Users\';
    savename=[pathR fnamebasis2 fnameo2];

% A figure
fig = figure('menubar','none');

% Add menus with Accelerators
mymenu = uimenu('Parent',fig,'Label','Hot Keys');
uimenu('Parent',mymenu,'Label','Zoom','Accelerator','w','Callback',@(src,ev
t)zoom(fig,'on'));%zoom the particular dwell time (hold mouse left button
when selecting area) to easly select start and end time data point
uimenu('Parent',mymenu,'Label','Pan','Accelerator','p','Callback',@(src,evt
)pan(fig,'on'));%move around the trace
uimenu('Parent',mymenu,'Label','Data','Accelerator','d','Callback',@(src,ev
t)GetDataFromFigure(fig));%you need to hold "Shift" button when you
collecting the time points!
uimenu('Parent',mymenu,'Label','Next','Accelerator','a','Callback',@(src,ev
t)LoadNextTrace(fig,totintc));%load next trace of the video
uimenu('Parent',mymenu,'Label','Save','Accelerator','g','Callback',@(src,ev
t)SaveDataPoints(fig,savename));%Saving Results: 1) "time_p" file with all
time points collected, 2) "dwell_on" file with caclulated dwell time
durations (time_p(n+1)-time_p(n)), 3) "dwell_off" file with caclulated
waiting time durations between dwells (time_p(n+3)-time_p(n+1))
uimenu('Parent',mymenu,'Label','Show','Accelerator','s','Callback',@(src,ev
t)ShowData(fig));% checking the current number of time points collected in
current trace. It is detrimental to be odd number before saving the result
of the trace.

%ploting trace
cla
ha(1)= plot(1);
p=plot(totintc(:,1),totintc(:,2),'-');
title(['totint not corrected, trace 1 out of ' num2str(size(totintc,2)-
1)]);
grid on
axis tight;
p.DataTipTemplate.FontSize = 1;
p.DataTipTemplate.DataTipRows(1).Label = '';
p.DataTipTemplate.DataTipRows(2).Label = '';
p.DataTipTemplate.DataTipRows(1).Format = '%0.0f';
p.DataTipTemplate.DataTipRows(2).Value = '';

end
```

2.1) sub-function GetDataFromFigure

```
function GetDataFromFigure(fig)
dcm_obj = datacursormode(fig);
set(dcm_obj,'enable','on');
end
```

2.2) sub-function ShowData

```
function ShowData(fig)
dcm_obj = datacursormode(fig);
mypoints = getCursorInfo(dcm_obj);
pos=cell2mat({mypoints.Position}');
datapoints=pos(:,1);
size(datapoints,1)
end
```

2.3) sub-function SaveDataPoints

```
function SaveDataPoints(fig,savename)
clc
global n;

dcm_obj = datacursormode(fig);
mypoints = getCursorInfo(dcm_obj);
pos=cell2mat({mypoints.Position}');
time_p_cur=sort(pos(:,1),'descend');

persistent time_p;
persistent dwell_on;
persistent dwell_off;

if exist('time_p')==0; time_p=[];end

stp = size(time_p);
stpc = size(time_p_cur);
a = max(stp(1),stpc(1));
time_p = [[time_p;zeros(abs([a 0]-stp))],[time_p_cur;zeros(abs([a,0]-
stpc))]];


assignin('base','time_p',time_p)
save([savename '_time_p.mat'],'time_p');

dwell_on_cur=[];
for i=1:2:(length(time_p_cur)-1)
    dwell_on_cur(end+1,:)=time_p_cur(i)-time_p_cur(i+1);
end

if exist('dwell_on')==0; dwell_on=[];end

sdo = size(dwell_on);
sdoc = size(dwell_on_cur);
a = max(sdo(1),sdoc(1));
dwell_on = [[dwell_on;zeros(abs([a 0]-sdo))],[dwell_on_cur;zeros(abs([a,0]-
sdoc))]];
```

```
assignin('base','dwell_on',dwell_on);
save([savename '_dwell_on.mat'],'dwell_on');

dwell_off_cur=[];
for i=2:2:(length(time_p_cur)-1)
    dwell_off_cur(end+1,:)=time_p_cur(i)-time_p_cur(i+1);
end

if exist('dwell_off')==0; dwell_off=[];end

sdf = size(dwell_off);
sdfc = size(dwell_off_cur);
a = max(sdf(1),sdfc(1));
dwell_off = [[dwell_off;zeros(abs([a 0]-
sdf))],[dwell_off_cur;zeros(abs([a,0]-sdfc))]];

assignin('base','dwell_off',dwell_off);
save([savename '_dwell_off.mat'],'dwell_off');

if isempty(n); trace=1; else trace=n-1;end

disp(['You saved trace:' num2str(trace)]);

end
```

2.4) sub-function LoadNextTrace

```
function LoadNextTrace(fig,totintc)
global n;
if isempty(n);n=2;end
n=n+1;
if n>numel(totintc(1,:))
    disp('You processed all the traces in this video!');
    close all;
    return
end
time=totintc(:,1);
tintc=totintc(:,n);

% New trace plotting
cla
clear plot(1);
ha(1) = plot(1);
p=plot(time,tintc,'-');
title(['totint not corrected, trace ' num2str(n-1) ' out of '
num2str(size(totintc,2)-1)]);
grid on
axis tight
p.DataTipTemplate.FontSize = 1;
p.DataTipTemplate.DataTipRows(1).Label = '';
p.DataTipTemplate.DataTipRows(2).Label = '';
p.DataTipTemplate.DataTipRows(1).Format = '%0.0f';
p.DataTipTemplate.DataTipRows(2).Value = '';
end
```

3) Script for selecting hotspots

```
hotspot=10; %select hotspot with minimum 10 dwell times in matrix dwell_on
(created by DwellTimeCollector)

dwell_on_hot=[];
stp = size(dwell_on);

for i=1:stp(2)
    spots=size(nonzeros(dwell_on(:,i)));
    if spots(1)>=hotspot
        dwell_on_hot(:,i)=dwell_on(:,i);
    end
end
 dwell_on_hot( :, ~any(dwell_on_hot,1) ) = [];
```

3) Script for creating cumulative dwell time distribution

```
%create cumulative dwell time distribution (cdf, as in Amrute-Nayak et al.,
PNAS, 2014)

CD=[];
cd=1;
dt=0.052;
X0=sort(nonzeros(dwell_on_hot));%you need hotspots matrix "dwell_on_hot"
created by HotSpots script

for i=0:dt:max(X0)
    c=0;
    for j=1:1:length(X0)
        if i<X0(j)
            c=c+1;
        end
    end

    CD(cd)=c;
    cd=cd+1;
end
x=[0:dt:max(X0)]';  %time vector
CDt=CD';            %cdf of dwell times

%Time vector (x) and cdf (CDt) can be than transferred to function fitting
%program like GraphPad Prism.
```