

```
%% Single molecule diffusion on surface
```

```
%The script is designed for working with an 'excel' file generated by MTrackJ  
%plug-in (ImageJ). So, references to column numbers are only valid in this case.  
%Change accordingly to data organization. In brief, column 2 is the  
%label of the particle, column 4 and 5, x and y coordinates respectively in  
%nanometers; and column 11 is the distance travel by the particle between  
%two consecutive frames.  
%The program calculates Diffusion constant, and alfa factor (as a  
%deviation measurement from 'normal' (strictly Brownian) diffusion) using  
%the Mean Square Displacement (MSD) at different window times for particles  
%moving in two dimensions.  
%It results with a plot for each trajectory and a plot of log10(MSD(deltat))  
%vs. log10(deltat). Finally writes in the original data file (in a different  
%sheet) the frequency of pauses, the average distance run between frames,  
%the diffusion constant, the label, and alfa factor.  
%Please, if you find any bug please send it to me  
%(h.sanchezgonzalez@erasmusmc.nl).
```

```
rawl=xlsread('data-example');
```

```
TX=[rawl(:,2) rawl(:,11) rawl(:,4:5)];
```

```
TX(isnan(TX))=0;%convert NaN into zero
```

```
singleproteins= struct('TR',[], 'FINAL', [], 'MSD1', [], 'scat', [],  
'Delta_tbueno', []);
```

```
for H=(unique(TX(:,1)))' % vector with the labeled particles
```

```
    %this organizes the data
```

```
    for a=1:size(TX,1);
```

```
        m=find(TX(:,1)==H);%gives to m the position in the first column of TX if  
it is equal to H
```

```
        singleproteins(1,H).TR= TX(m,2);%takes the values in x (nm run per  
frame) for each trajectory in a different matrix
```

```
        singleproteins(1,H).scat= TX(m,3:4); %this can be used for plotting the  
trajectory
```

```
    end
```

```
    if numel(singleproteins(1,H).TR)>1 %only works if there are more than  
one data point
```

```
        ...in the trajectory
```

```
singleproteins(1,H).Delta_tbueno=( [1:1:round(numel(singleproteins(1,H).scat(:,  
1))/4)]./2)';%first divided...
```

```
    % by 4 so only 1/4th of the time lag are...
```

```
    %considered; also here divided by 2...
```

```
    %because frame acquisition time was 0.5 sec, change accordingly
```

```
for intervalo=1:round(numel(singleproteins(1,H).scat(:, 1))/4);
```

```
    for inicio=1:round(numel(singleproteins(1,H).scat(:, 1))/4);
```

```
        singleproteins(1,H).MSD1(intervalo)=
```

```
mean(diff(singleproteins(1,H).scat(inicio:intervalo:end, 1))...
```

```
        .^2 + diff(singleproteins(1,H).scat(inicio:intervalo:end,
```

```
2)).^2);
```

```
end
```

```
end

if      numel(singleproteins(1,H).MSD1)>2 %only works if there are more than
two lag times !

    singleproteins(1,H).FINALa=[singleproteins(1,H).Delta_tbueno
singleproteins(1,H).MSD1'];
    singleproteins(1,H).FINAL=log10(singleproteins(1,H).FINALa);
    [singleproteins(1,H).p,singleproteins(1,H).ErrorEst] =
polyfit(singleproteins(1,H)...
        .FINAL(:, 1),singleproteins(1,H).FINAL(:, 2) ,1);
    [singleproteins(1,H).frit,singleproteins(1,H).delta] =
polyval(singleproteins(1,H)...
        .p,singleproteins(1,H).FINAL(:, 1) , singleproteins(1,H).ErrorEst);
    singleproteins(1,H).posit= singleproteins(1,H).TR>0;

singleproteins(1,H).speed=mean(singleproteins(1,H).TR(singleproteins(1,H).posit)
)/1;%change if different time frame
    singleproteins(1,H).mobility=
numel(singleproteins(1,H).TR(singleproteins(1,H).posit))...
    /numel(singleproteins(1,H).TR);
    singleproteins(1,H).pauses=1-singleproteins(1,H).mobility;
    singleproteins(1,H).label=H;

end

figure

%plot trajectory
subplot(121);
plot((singleproteins(1,H).scat(:, 1)), (singleproteins(1,H).scat(:,
2)));set(gca,'YDir','reverse');
title( {'fpauses=',sprintf('%7.2f',(singleproteins(1,H).pauses))});
xlabel('X (nm)');
ylabel('Y (nm)');

if isempty(singleproteins(1,H).p)==0

    %plot MSD

        subplot(122),plot(singleproteins(1,H).FINAL(:,
1),singleproteins(1,H).FINAL(:, 2),'+',...
            singleproteins(1,H).FINAL(:, 1),singleproteins(1,H).frit,'g-',...
            singleproteins(1,H).FINAL(:,
1),singleproteins(1,H).frit+2*singleproteins(1,H).delta,'r:',...
            singleproteins(1,H).FINAL(:, 1),singleproteins(1,H).frit-
2*singleproteins(1,H).delta,'r:');
        title(
{'D=',sprintf('%7.2f',((power(10,singleproteins(1,H).p(:,2)))/4)), ' nm^2/s',
...
        ' alfa=',sprintf('%7.2f',((singleproteins(1,H).p(:,1))))});
        % Annotate the plot
        xlabel('log10-time lag');
        ylabel('log10-MSD');
```

end

end

end

```
Statistics.pau=cat(1,singleproteins.pauses);
Statistics.sp=cat(1,singleproteins.speed);
Statistics.label=cat(1,singleproteins.label);
Statistics.D=cat(1,singleproteins.p);
Statistics.D_d=(power(10,Statistics.D(:,2)))./4;
Statistics.D_deviation=Statistics.D(:,1);

%export results to excel
xlswrite('data-example', Statistics.pau, 2, 'A1:A100');
xlswrite('data-example', Statistics.sp, 2, 'B1:B100');
xlswrite('data-example', Statistics.D_d, 2, 'C1:C100');
xlswrite('data-example', Statistics.label, 2, 'D1:D100');
xlswrite('data-example', Statistics.D_deviation, 2, 'E1:E100');
```