

Qdots in PAAm at 200fps

To run *and* export, paste the following command. Otherwise, export manually. You must first define monkier to do this.

```
% export("TrackMate_Live_Script_guide.mlx",
strcat('C:\Users\al3xm\Documents\GitHub\HISTia\TrackMate Reports\',monkier),
HideCode=false, OpenExportedFile=true, Run=true, FigureResolution=1600)
```

Name this data analysis run

```
monkier = '200fps Qdots In PAAm March 13 Dataset';
disp(monkier)
```

200fps Qdots In PAAm March 13 Dataset

Set the current directory for data saving purposes.

```
cd('C:\Users\al3xm\Documents\GitHub\HISTia')
```

This Report generated at

```
disp(datetime)
```

25-Mar-2025 10:20:49

Define the source folder here:

```
main_folder = "C:\Users\al3xm\Documents\_Local_Data\25.03.12_PAG_HILO_Beads\5ms"
```

```
main_folder =
"C:\Users\al3xm\Documents\_Local_Data\25.03.12_PAG_HILO_Beads\5ms"
```

```
paths = FileFinder(main_folder, '.xml')
```

```
paths = 1x14 string
"C:\Users\al3xm\Documents\_Local_Data\25.03.12_PAG_HILO_Beads\5ms\1... "C:\Users\ ..."
```

```
if ~isfolder(main_folder)
    error('please enter a valid directory')
elseif isempty(paths)
    error('please ensure the folder contains .xml ParticleTrack exports')
end
```

If you want to scale the tracks, set the scaling factor here in pixels/distance unit. Otherwise, set scaling to -1 to avoid scaling.

```
scaling = 1 % units per input unit
```

```
scaling = 1
```

Define the units

```
SpaceUnits = 'microns';
```

```
TimeUnits = 'seconds';
```

Enter the timestep in time units. Ensure that you use the inverse of the fps, not exposure.

```
dt=0.005 % 1/(fps)
```

```
dt = 0.0050
```

what is the minimum track length you want to consider?

```
MinTrackLength = 7
```

```
MinTrackLength = 7
```

Now, generate the MSD analyzer structure. Note that this filters out any tracks with a track length of less than 7 by default, though this can be changed.

```
ma1 = TrackMateImport(main_folder, true, MinTrackLength, scaling);
```

```
Warning: Directory already exists.
```

```
found 2030 tracks in the file.  
found 1475 tracks in the file.  
found 2060 tracks in the file.  
found 2279 tracks in the file.  
found 4530 tracks in the file.  
found 4410 tracks in the file.  
found 1951 tracks in the file.  
found 1815 tracks in the file.  
found 1829 tracks in the file.  
found 1777 tracks in the file.  
found 1911 tracks in the file.  
found 1577 tracks in the file.  
found 1701 tracks in the file.  
found 1541 tracks in the file.  
found a total of 1541 tracks in the directory
```

```
Warning: Scaling factor active!
```

```
Warning: plotting only tracks longer than threshold length
```

```
Computing MSD of 15212 tracks... 15Done.
```

```
ma1 = ma1.fitMSD(0.5); % set to short clipping factor to find diffusive rate
```

```
Fitting 15212 curves of MSD = f(t), taking only the first 50% of each curve... 15Done.
```

```
ma1 = ma1.fitLogLogMSD(0.5); % set to long clipping factor for linearity measure
```

```
Fitting 15212 curves of log(MSD) = f(log(t)), taking only the first 50% of each curve... Done.
```

```
d_est = mean(ma1.lfit.a) * (1/dt); % starts with units^2 per frame, divide by  
frames/time unit.  
% This gives the estimated slope of the MSD in units^2/second  
gamma_est = mean(ma1.loglogfit.alpha); % Gives time scaling factor of the MSD.  
dimensionality_modifier = 4; % Conversion to use conventional x^2 = 4Dt metric for  
2D diffusion.  
fprintf(strcat("The diffusion coefficient is D=", string(d_est/  
(dimensionality_modifier)), ' ', SpaceUnits, '^2/', TimeUnits));
```

```
The diffusion coefficient is D=5.0721microns^2/seconds
```

```
fprintf(strcat("The time scaling factor is gamma=", string(gamma_est), '.'));
```

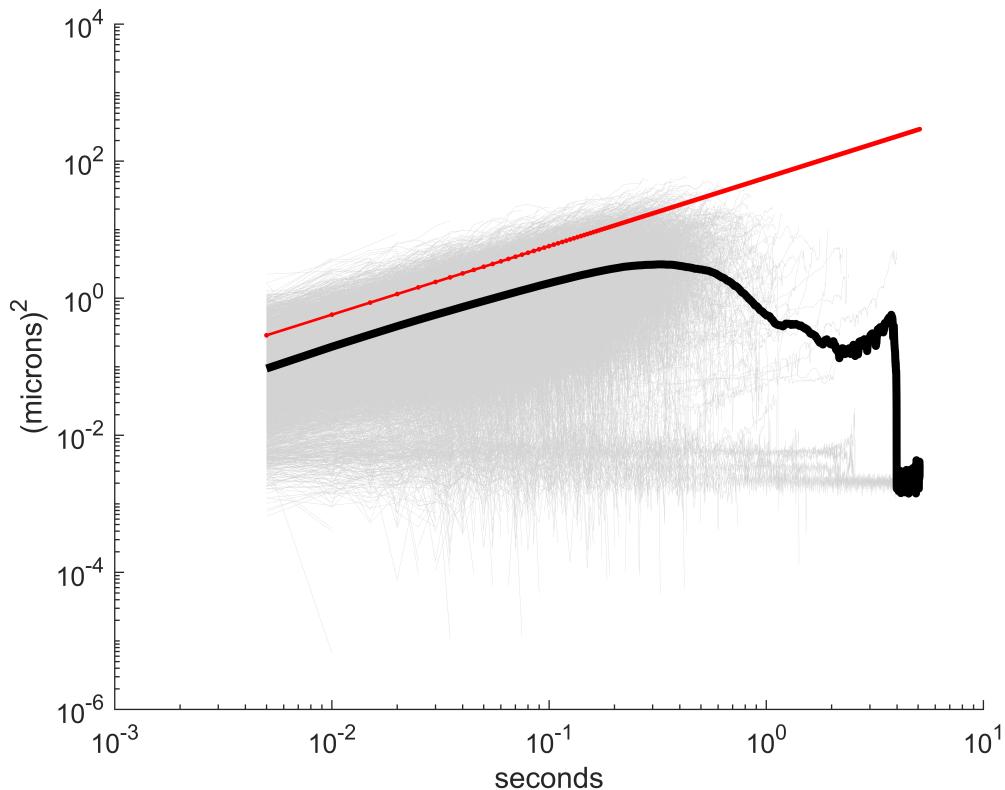
The time scaling factor is gamma=0.8673.

Now, we plot the MSD. If you want to plot an expected value, set it. Otherwise, set D to -1.

```
D_expect=14.4; %D=0.4292 for 1um beads in water.  
fprintf(strcat('expected diffusion coefficient: ', string(D_expect), ' (' ,  
SpaceUnits, ')^2/' , TimeUnits))
```

expected diffusion coefficient:14.4 (microns)^2/seconds

```
FigMeanMSD(SpaceUnits, TimeUnits,ma1, dt,D_expect,true)
```



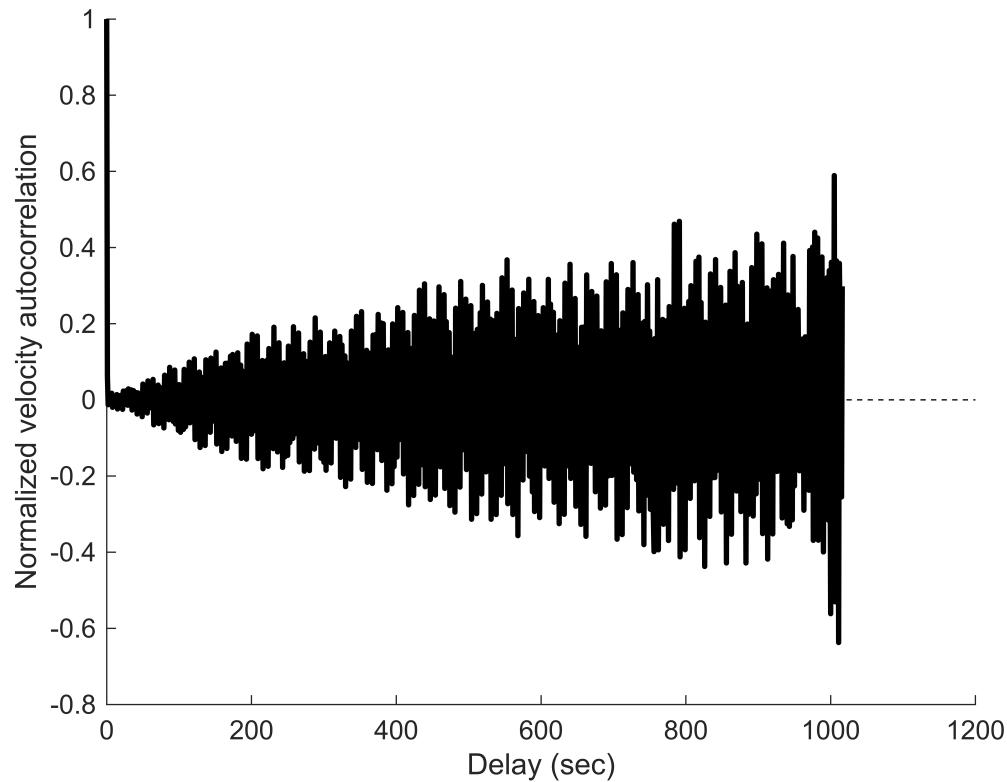
```
ans = 1018x5  
103 x  
0 0 0 6.6313 0  
0.0000 0.0001 0.0001 5.9191 0.0003  
0.0000 0.0002 0.0001 5.1901 0.0006  
0.0000 0.0003 0.0002 4.7515 0.0009  
0.0000 0.0004 0.0003 4.4323 0.0012  
0.0000 0.0005 0.0004 4.1536 0.0014  
0.0000 0.0006 0.0004 3.9057 0.0017  
0.0000 0.0007 0.0005 3.6721 0.0020  
0.0000 0.0007 0.0006 3.4610 0.0023  
0.0000 0.0008 0.0006 3.2896 0.0026  
:  
:
```

Now we create a plot of the mean velocity correlation as a function of time.

```
figure;
```

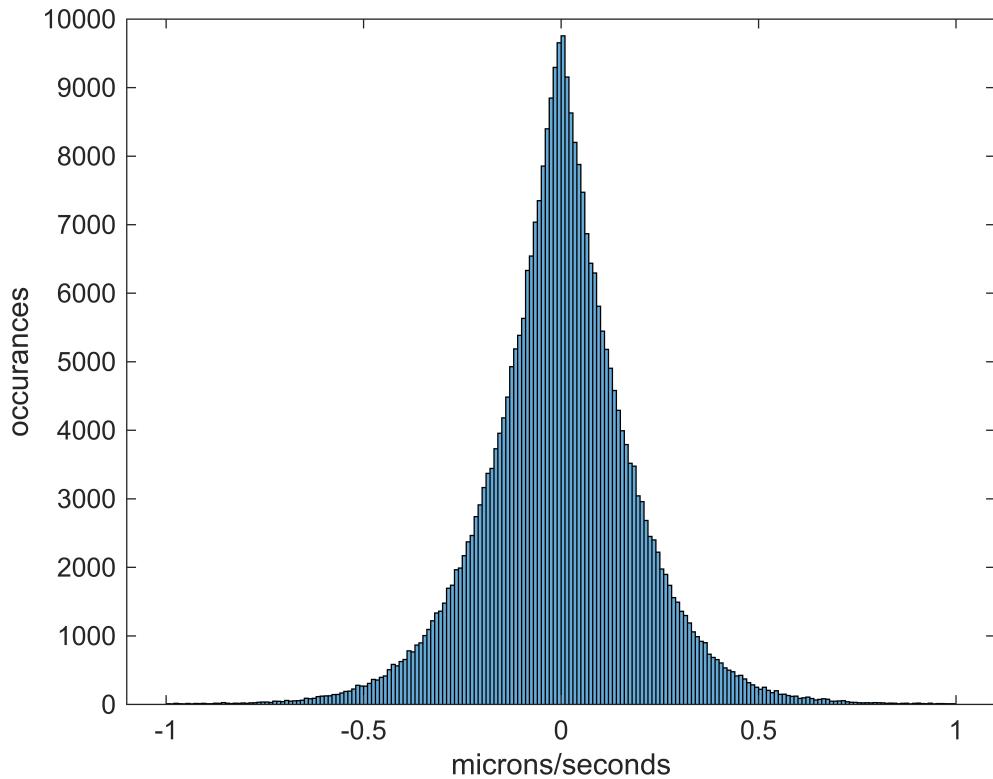
```
ma1.plotMeanVCorr;
```

Computing velocity autocorrelation of 15212 tracks...



We may also directly compute velocities and plot them as a histogram

```
v = ma1.getVelocities;
V=vertcat(v{:});
edges2 = -1:0.01:1;
histogram(V(:,2),edges2)
xlabel(strcat(SpaceUnits, '/', TimeUnits))
ylabel("occurrences")
```



Now, we must call CenterTracks.

```
CenterTracks=CreateCenterTracks(ma1.tracks);
```

We may now create a short vs long track diagram.

```
figure();
% UniDomainFigCenterJuxt(SpaceUnits, CenterTracks, 10,10, 10, 10, true, true, 1)
```

Now create a VanHovePlot. This function calls the new VanHove2.m so it is relatively fast and creates bins with equal widths, but it can still take a while to run.

Now, we can can manimputme the CreateVanHovePlots function in a few ways.

We can change the time steps, the number of particles assigned to each bin, and the minimum bin width.

The BinSize determines the number of particles which the Van

```
BinSize = 10
```

```
BinSize = 10
```

```
MinBin = 0.04 % In microns. I don't recommend going much below diffraction limit/4
```

```
MinBin = 0.0400
```

```
VanHoveStats= CreateVanHovePlots(SpaceUnits, TimeUnits, ma1.tracks, BinSize,  
[1,2,3,5], main_folder, MinBin, dt)
```

Calculating VanHove Distribution dt=1

5000
10000
15000

Calculating VanHove Distribution dt=2

5000
10000
15000

Calculating VanHove Distribution dt=3

5000
10000
15000

Calculating VanHove Distribution dt=5

5000
10000
15000

Cleaning up for dt=1

Cleaning up for dt=2

Cleaning up for dt=3

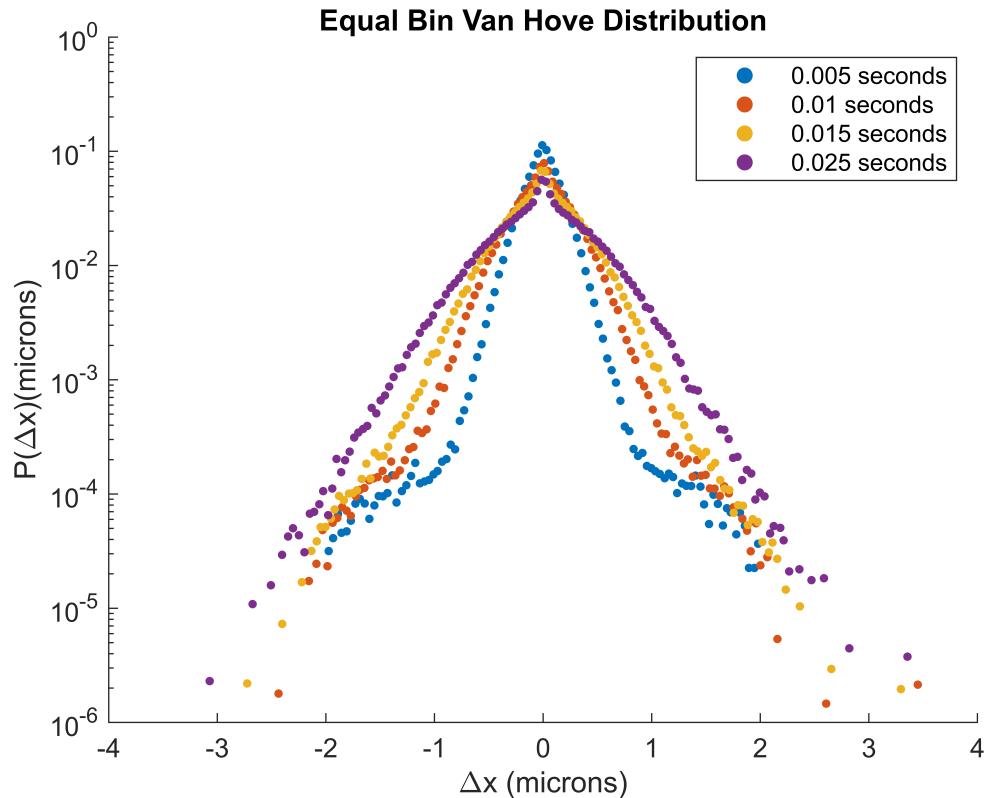
Cleaning up for dt=5

Sorting dt=1 into bins

Sorting dt=2 into bins

Sorting dt=3 into bins

Sorting dt=5 into bins



```
VanHoveStats = struct with fields:  
    Data: {5x1 cell}  
    CenterPoint: {5x1 cell}  
    EquiProb: {5x1 cell}  
    TotStepsCount: [4x2 table]  
    BinSize: 10  
    tau: [1 2 3 5]  
    FrameTime: 0.0050  
    MinBin: 0.0400
```

Save the data in SuperStruct format to the SuperStructs Github folder.

```
MasterSaveD(monkier,dt, ma1,main_folder, MinTrackLength, SpaceUnits, TimeUnits,  
VanHoveStats, cd)
```

```
ans =  
'C:\Users\al3xm\Documents\GitHub\HISTia\SuperStructs\200fps Qdots In PAAm March 13 Dataset'
```

Save this output as a pdf for later viewing.