Qdots in PAAm at 100fps

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 = '100fps qdots in PAAm from March dataset';
disp(monkier)
```

```
100fps qdots in PAAm from March dataset
```

Set the current directory for data saving purposes.

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

This Report generated at

```
disp(datetime)
```

```
   21-Mar-2025 12:18:22
```

Define the source folder here:

```
main_folder =  "C:\Users\al3xm\Documents\_Local_Data\25.03.12_PAG_HILO_Beads\10ms"
```

```
main_folder = 
"C:\Users\al3xm\Documents\_Local_Data\25.03.12_PAG_HILO_Beads\10ms"
```

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

```
paths = 1×2 string
"C:\Users\al3xm\Documents\_Local_Data\25.03.12_PAG_HILO_Beads\10ms\1st2…  "C:\Us ⋯
```

```
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=0.01 % 1/(fps)
```

```
dt = 0.0100
```

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 16143 tracks in the file.
found 16331 tracks in the file.
found a total of 16331 tracks in the directory
Warning: Scaling factor active!
Warning: plotting only tracks longer than threshold length
Computing MSD of 17672 tracks... 17Done.
```

```
ma1 =ma1.fitMSD(0.25); % set to short clipping factor to find diffusive rate
```

```
Fitting 17672 curves of MSD = f(t), taking only the first 25% of each curve... 17Done.
```

```
ma1 = ma1.fitLogLogMSD(0.25); % set to long clipping factor for linearity measure
```

```
Fitting 17672 curves of log(MSD) = f(log(t)), taking only the first 25% 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=3.5801microns^2/seconds
```

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

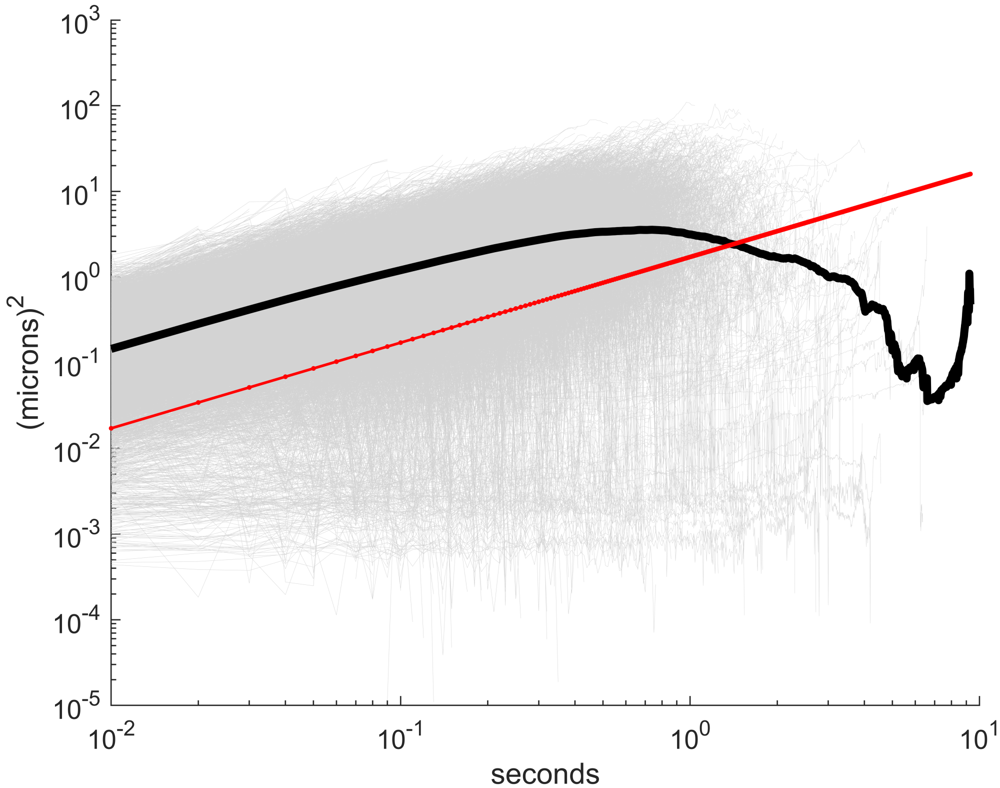
```
The time scaling factor is gamma=0.83881.
```

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

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

```
excpected diffusion coefficient:0.4292 (microns)^2/seconds
```

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

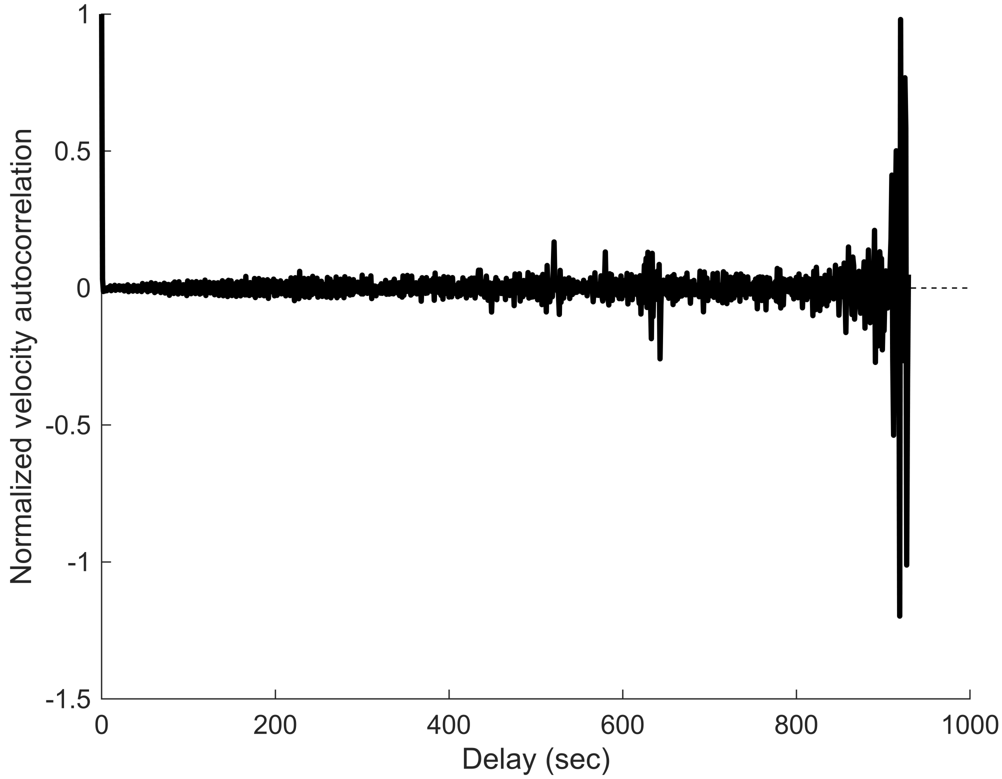



```
ans = 931×5
10^3 ×
         0         0         0    7.5483         0
    0.0000    0.0001    0.0001    6.6567    0.0000
    0.0000    0.0003    0.0002    5.7667    0.0000
    0.0000    0.0004    0.0003    5.2336    0.0001
    0.0000    0.0005    0.0004    4.8570    0.0001
    0.0001    0.0007    0.0005    4.5401    0.0001
    0.0001    0.0008    0.0006    4.2725    0.0001
    0.0001    0.0009    0.0007    4.0417    0.0001
    0.0001    0.0010    0.0008    3.8579    0.0001
    0.0001    0.0011    0.0009    3.7038    0.0002
      ⋮
```

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

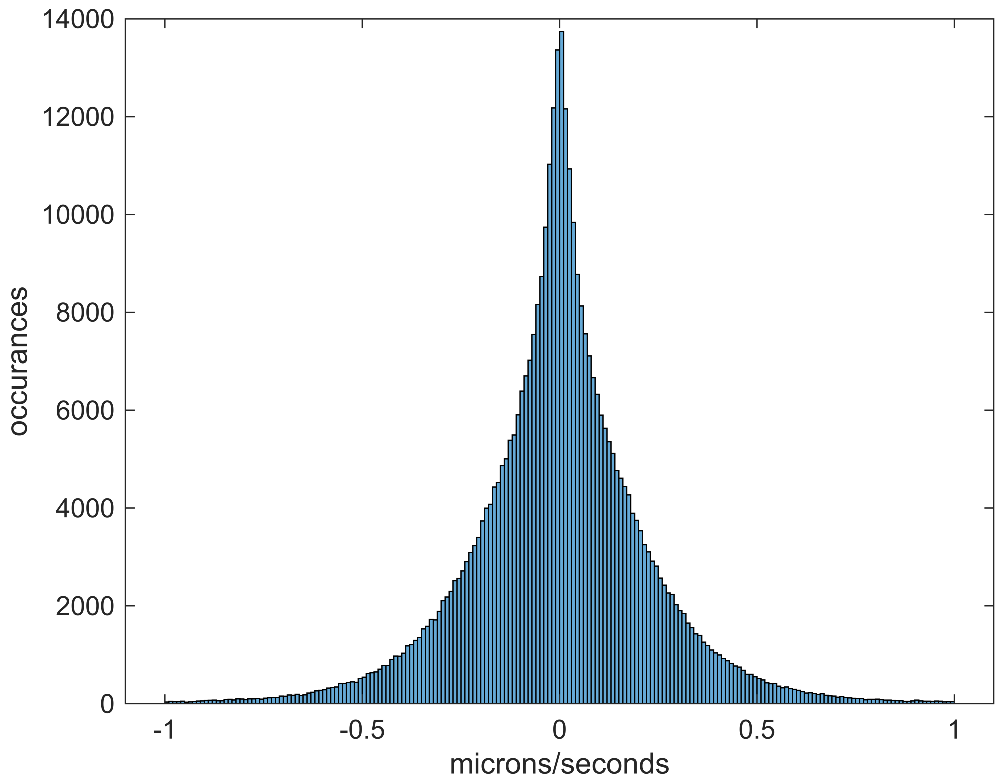
```
figure;
ma1.plotMeanVCorr;
```

```
Computing velocity autocorrelation of 17672 tracks...
```



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

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



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
```

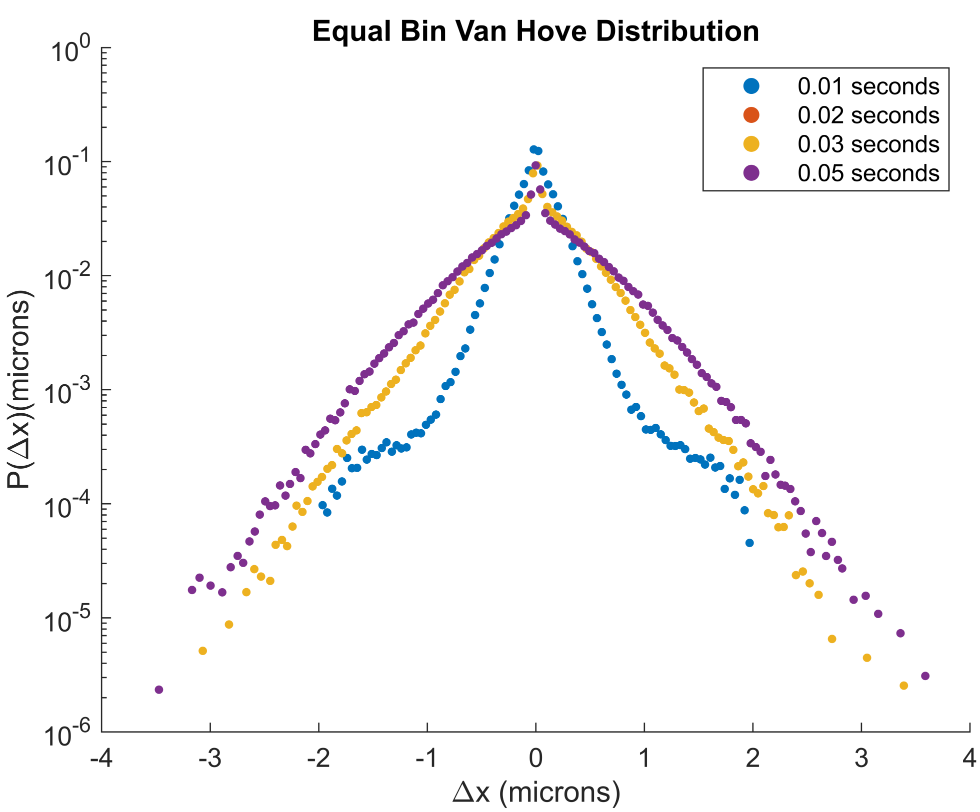
```
BinSize = 10
```

```
MinBin = 0.045 % In microns. I don't recommend going much below diffraction limit/4
```

```
MinBin = 0.0450
```

```
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: {5×1 cell}
       CenterPoint: {5×1 cell}
          EquiProb: {5×1 cell}
     TotStepsCount: [4×2 table]
           BinSize: 10
               tau: [1 2 3 5]
         FrameTime: 0.0100
            MinBin: 0.0450
```

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

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

```
ans = 
"C:\Users\al3xm\Documents\GitHub\HISTia\SuperStructs\100fps qdots in PAAm from March dataset2025-03-21.12.23.54.mat"
```

Save this output as a pdf for later viewing.