

# 20ms frames of qdots in PAG

This Report generated at

```
disp(datetime)
```

06-Mar-2025 03:14:43

Define the source folder here:

```
main_folder =
"C:\Users\al3xm\Documents\_Local_Data\25.02.11_HILO_PAG\specimen2_24hrs_later\20ms_x
mls"
```

```
main_folder =
"C:\Users\al3xm\Documents\_Local_Data\25.02.11_HILO_PAG\specimen2_24hrs_later\20ms_xmls"
```

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=0.02 % 1/(fps)
```

```
dt = 0.0200
```

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, scaling);
```

```
Warning: Directory already exists.
found 6387 tracks in the file.
found a total of 6387 tracks in the directory
Warning: Scaling factor active!
Warning: plotting only tracks longer than threshold length
Computing MSD of 2499 tracks... 2Done.
```

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

```
Fitting 2499 curves of MSD = f(t), taking only the first 25% of each curve... 2Done.
```

```
ma1 = ma1.fitLogLogMSD(0.75); % set to long clipping factor for linearity measure
```

```
Fitting 2499 curves of log(MSD) = f(log(t)), taking only the first 75% of each curve... Done.
```

```
d_est = mean(ma1.lfit.a) * dt; % gives the estimated slope of the MSD in units^2/second
gamma_est = mean(ma1.loglogfit.alpha) % Gives time scaling factor of the MSD.
```

```
gamma_est = 0.7218
```

```
fprintf(strcat("The diffusion coefficient is D=", string(d_est), ' ', SpaceUnits,
'^2/', TimeUnits));
```

The diffusion coefficient is D=0.0058656microns^2/seconds

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

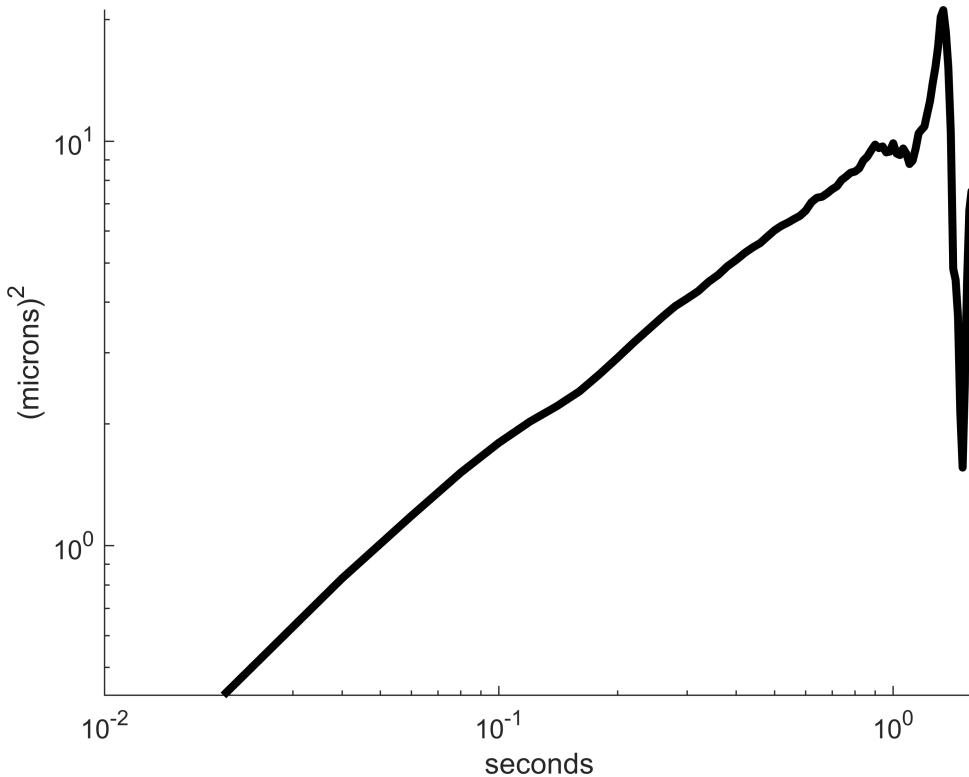
The time scaling factor is gamma=0.72177.

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

```
D_expect=-1;
fprintf(strcat('expected diffusion coefficient:', string(D_expect), ' (',
SpaceUnits, ')^2/', TimeUnits))
```

expected diffusion coefficient:-1 (microns)^2/seconds

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



```
ans = 81x5
10^3 x
0 0 0 1.9352 0
0.0000 0.0004 0.0002 1.8496 -0.0001
0.0000 0.0008 0.0004 1.6849 -0.0002
```

```

0.0001  0.0012  0.0006  1.5581 -0.0002
0.0001  0.0015  0.0009  1.4715 -0.0003
0.0001  0.0018  0.0011  1.4054 -0.0004
0.0001  0.0020  0.0013  1.3528 -0.0005
0.0001  0.0022  0.0015  1.2996 -0.0006
0.0002  0.0024  0.0017  1.2173 -0.0006
0.0002  0.0027  0.0019  1.1204 -0.0007
:

```

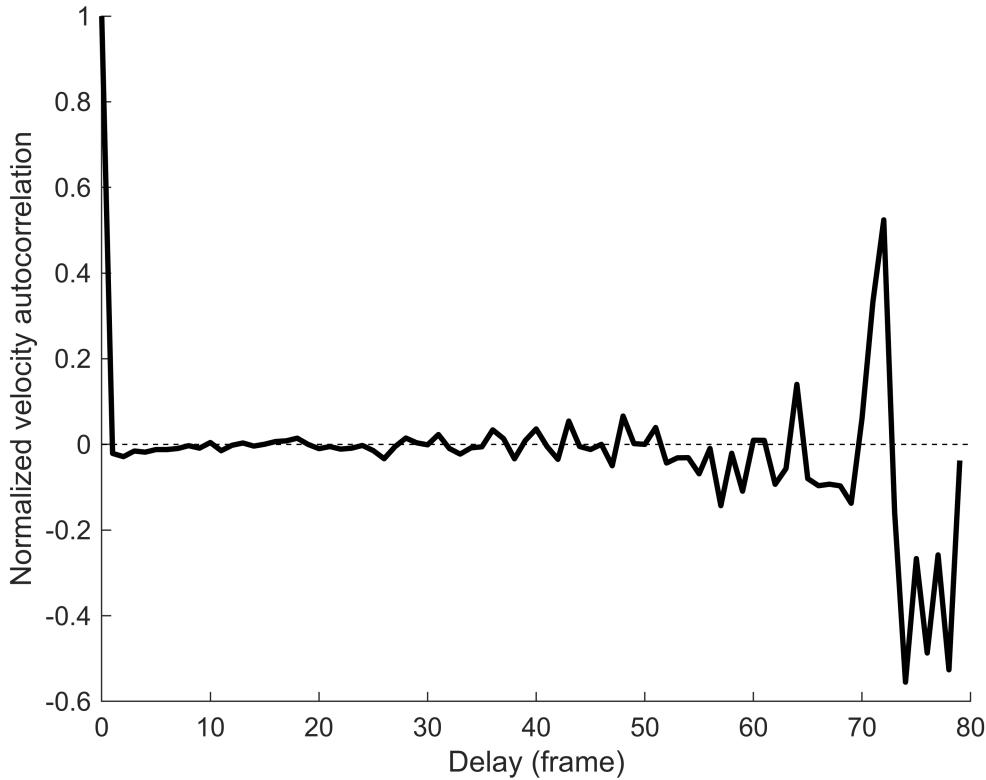
Now we create a plot of the mean velocity correlation as a function of time.

```

figure;
ma1.plotMeanVCorr;

```

Computing velocity autocorrelation of 2499 tracks... Done.

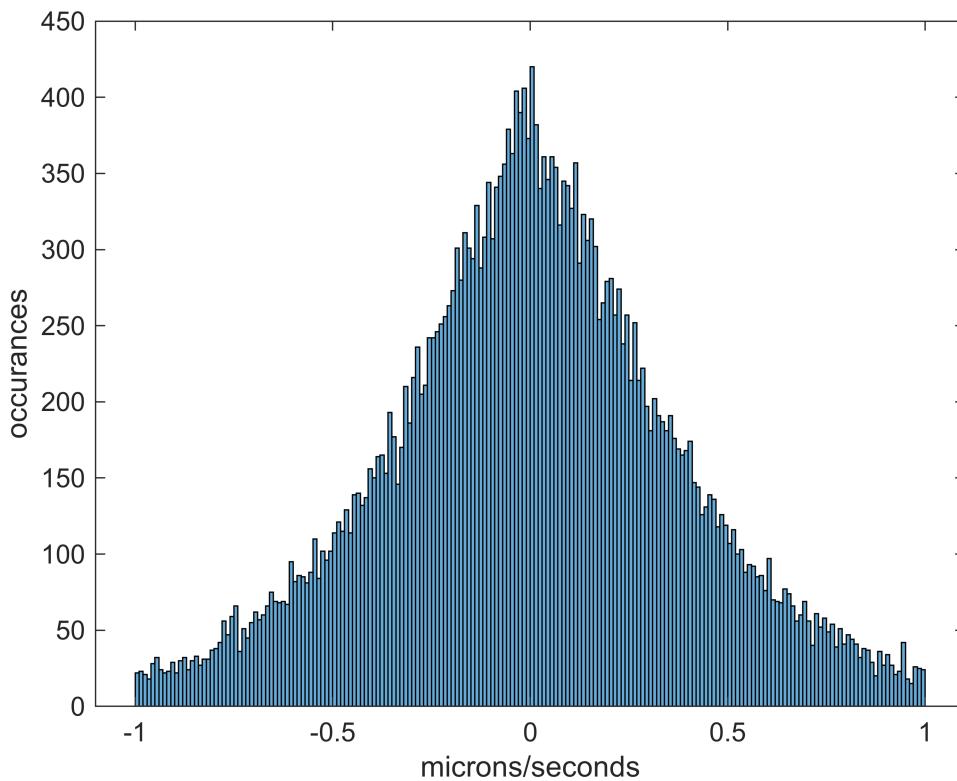


We may also directly compute velocities and plot them as a histogram

```

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 maniputme 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 = 0.1 % In microns
```

```
MinBin = 0.1000
```

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

Calculating VanHove Distribution dt=1

1000  
2000

Calculating VanHove Distribution dt=2

1000  
2000

Calculating VanHove Distribution dt=3

1000  
2000

Calculating VanHove Distribution dt=5

1000  
2000

Cleaning up for dt=1

Cleaning up for dt=2

Cleaning up for dt=3

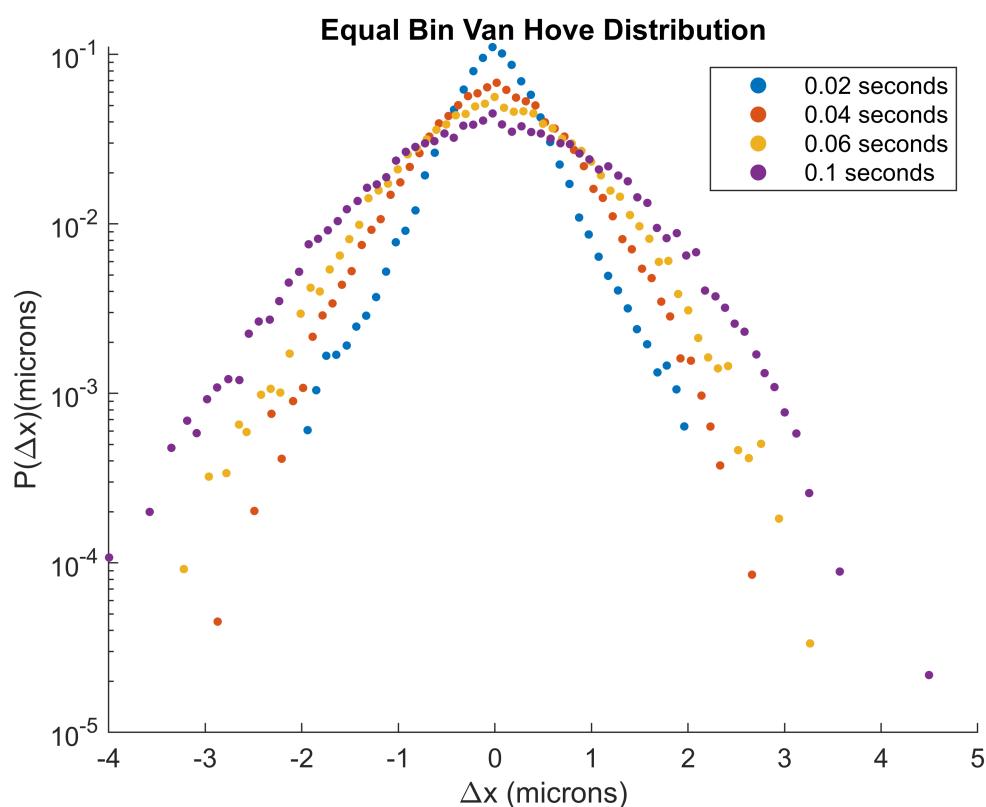
Cleaning up for dt=5

Sorting dt=1 into bins

Sorting dt=2 into bins

Sorting dt=3 into bins

Sorting dt=5 into bins



CenterPoint = 1×5 cell

	1	2	3	4	5
1	1×40 double	1×50 double	1×59 double	[]	1×69 double

**TotStepCount** = 4×2 table

	CellNumber	Value
1	1	26675
2	2	21049
3	3	17882
4	5	14031

**VanHoveData** = 1×5 cell

	1	2	3	4	5
1	26675×4 double	21049×4 double	17882×4 double	[]	14031×4 double

Save this file as a pdf.