First, load the folder

```
main_folder =
"C:\Users\al3xm\Documents\_Local_Data\25.02.11_HILO\specimen2_24hrs_later\10ms_timel
apse_XMLs"
```

```
main_folder =
"C:\Users\al3xm\Documents\_Local_Data\25.02.11_HILO\specimen2_24hrs_later\10ms_timelapse_XMLs"
```

Now, generate ma file.

```
ma1 = TrackMateImport(main_folder)
```

```
Warning: Directory already exists.
found 8072 tracks in the file.
found 6565 tracks in the file.
found 6844 tracks in the file.
found 7153 tracks in the file.
found 7665 tracks in the file.
found 7290 tracks in the file.
found 7651 tracks in the file.
found 8263 tracks in the file.
found 8412 tracks in the file.
found 8527 tracks in the file.
    85

       30628
Warning: plotting only tracks longer than threshold length
Computing MSD of 30713 tracks... 30Done.
ma1 = 
  msdanalyzer with properties:

      TOLERANCE: 12
         tracks: {30713×1 cell}
          n_dim: 2
    space_units: 'µm'
     time_units: '20ms frame'
            msd: {30713×1 cell}
          vcorr: []
           lfit: []
      loglogfit: []
          drift: []
```

Prompt the user for the timestep **in seconds**

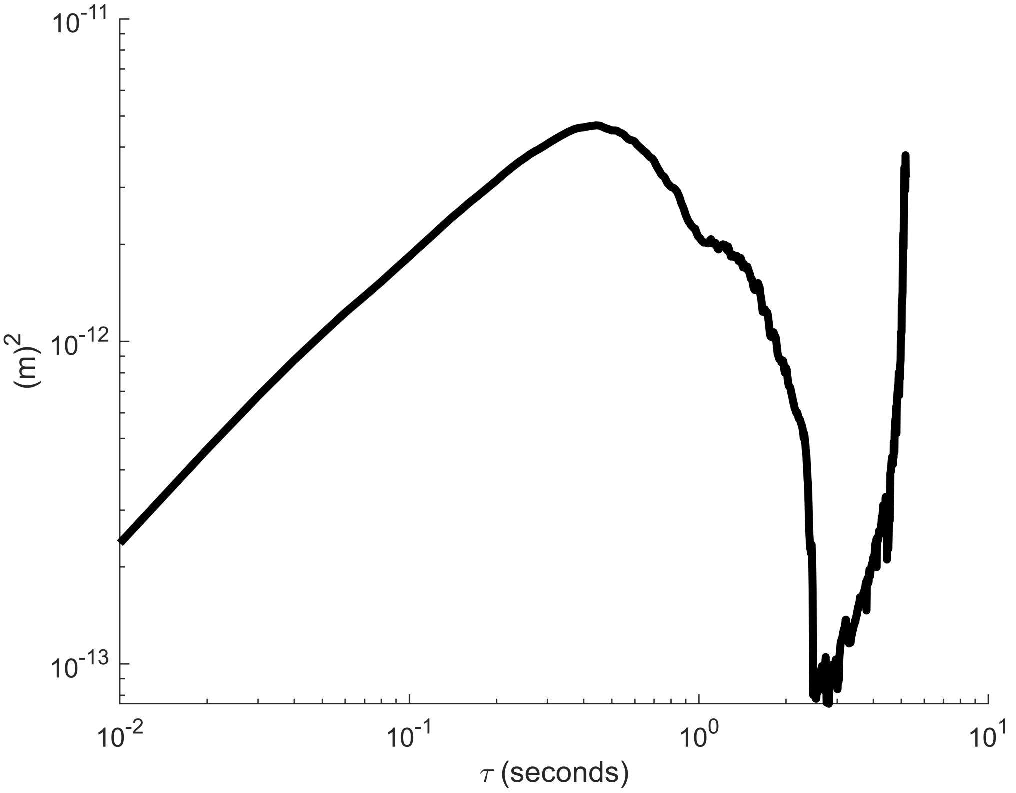
```
dt="0.01"
```

```
dt = 
"0.01"
```

Now, create FigMeanMSD.

```
FigMeanMSD(ma1, dt,-1, false, 10^-12)
```

```
Warning: Beware extremely large timesteps
```



```
ans = 519×5
10⁴ ×
         0         0         0    1.9879         0
    0.0000    0.0000    0.0000    1.8210   -0.0000
    0.0000    0.0000    0.0000    1.6022   -0.0000
    0.0000    0.0001    0.0000    1.4564   -0.0000
    0.0000    0.0001    0.0001    1.3494   -0.0000
    0.0000    0.0001    0.0001    1.2620   -0.0000
    0.0000    0.0001    0.0001    1.1857   -0.0000
    0.0000    0.0001    0.0001    1.1124   -0.0000
    0.0000    0.0002    0.0001    1.0297   -0.0000
    0.0000    0.0002    0.0001    0.9375   -0.0000
      ⋮
```

toNow, create CenterTracks

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

```
CenterTracks = 30713×1 cell
```

	1
1	21×4 double
2	11×4 double
3	12×4 double
4	11×4 double
5	9×4 double
6	15×4 double

	1
7	10×4 double
8	113×4 double
9	32×4 double
10	7×4 double
11	133×4 double
12	10×4 double
13	37×4 double
14	54×4 double
15	14×4 double
16	20×4 double
17	8×4 double
18	18×4 double
19	12×4 double
20	12×4 double
21	76×4 double
22	18×4 double
23	7×4 double
24	8×4 double
25	18×4 double
26	16×4 double
27	8×4 double
28	8×4 double
29	9×4 double
30	12×4 double
31	20×4 double
32	8×4 double
33	8×4 double
34	10×4 double
35	12×4 double
36	12×4 double
37	12×4 double
38	21×4 double
39	12×4 double

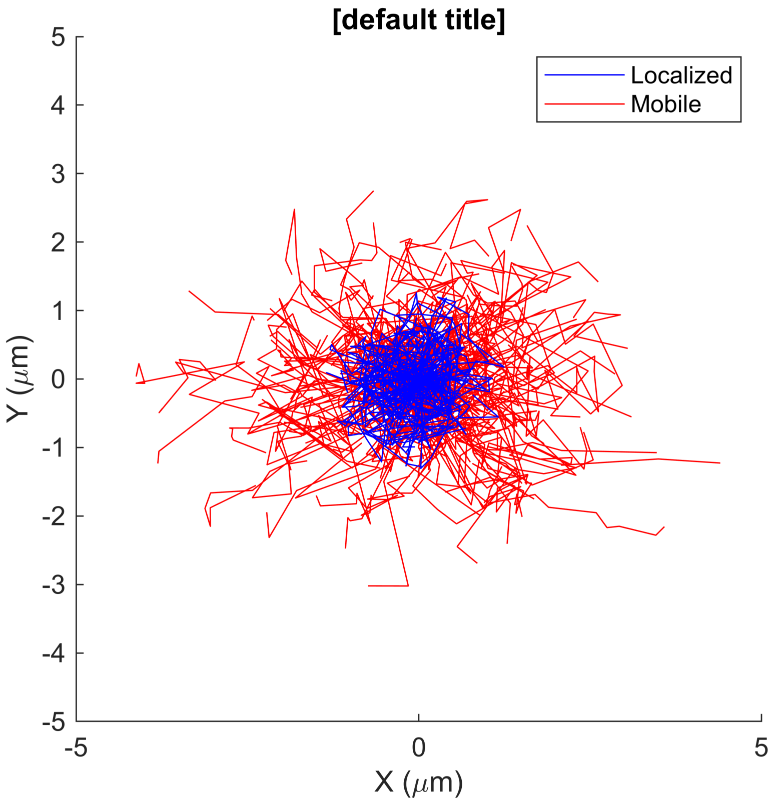
	1
40	9×4 double
41	84×4 double
42	20×4 double
43	30×4 double
44	10×4 double
45	7×4 double
46	15×4 double
47	17×4 double
48	18×4 double
49	7×4 double
50	118×4 double
51	31×4 double
52	7×4 double
53	20×4 double
54	27×4 double
55	15×4 double
56	7×4 double
57	38×4 double
58	85×4 double
59	27×4 double
60	96×4 double
61	9×4 double
62	27×4 double
63	57×4 double
64	22×4 double
65	11×4 double
66	13×4 double
67	40×4 double
68	12×4 double
69	22×4 double
70	7×4 double
71	55×4 double
72	33×4 double

	1
73	10×4 double
74	22×4 double
75	24×4 double
76	11×4 double
77	24×4 double
78	31×4 double
79	46×4 double
80	11×4 double
81	43×4 double
82	7×4 double
83	11×4 double
84	8×4 double
85	37×4 double
86	24×4 double
87	17×4 double
88	7×4 double
89	18×4 double
90	7×4 double
91	8×4 double
92	7×4 double
93	8×4 double
94	12×4 double
95	11×4 double
96	7×4 double
97	7×4 double
98	7×4 double
99	7×4 double
100	7×4 double

⋮

We may now create a short vs long track diagram

```
UniDomainFigCenterJuxt(CenterTracks, 10000,10000, 10, 10, true, true, 50)
```



Now create a VanHovePlot. This function calls the new VanHove2.m function as well as several other functions to create a nice concise code to create what you want to see.

```
VanHoveData = CreateVanHovePlots(ma1.tracks, 300, [1,2,5,10], main_folder);
```

```
 Calculating VanHove Distribution dt=1

1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
18000
19000
20000
21000
22000
23000
24000
25000
```

```
26000
27000
28000
29000
30000
 Calculating VanHove Distribution dt=2

1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
18000
19000
20000
21000
22000
23000
24000
25000
26000
27000
28000
29000
30000
 Calculating VanHove Distribution dt=5

1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
18000
19000
20000
21000
22000
23000
24000
25000
```

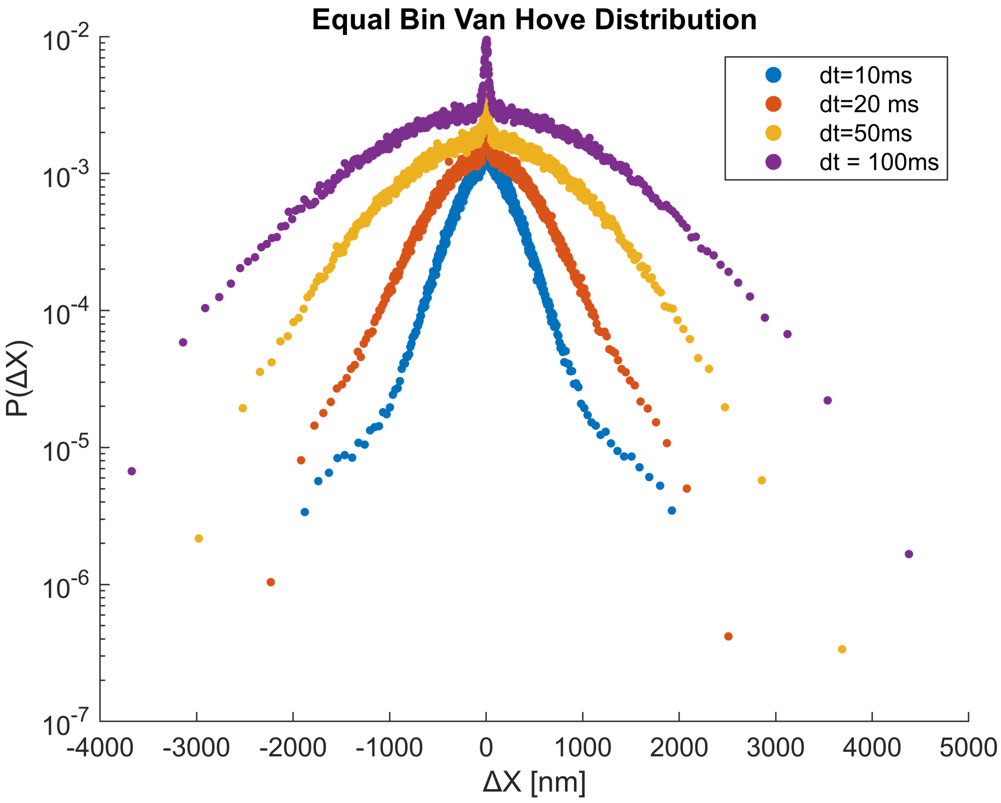
```
26000
27000
28000
29000
30000
 Calculating VanHove Distribution dt=10

1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
18000
19000
20000
21000
22000
23000
24000
25000
26000
27000
28000
29000
30000
 Cleaning up for dt=1

 Cleaning up for dt=2

 Cleaning up for dt=5

 Cleaning up for dt=10
Sorting dt=1 into bins
Sorting dt=2 into bins
Sorting dt=5 into bins
Sorting dt=10 into bins
Warning: Ignoring extra legend entries.
```

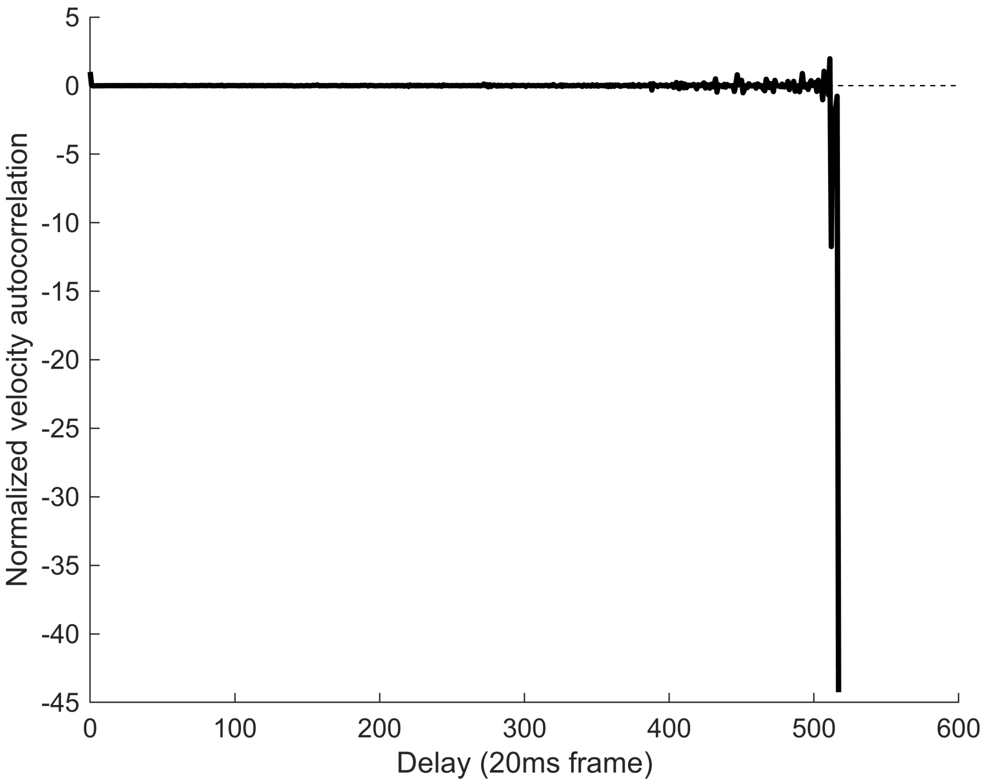


```
Warning: Over 1000 bins. Irregularities may occcur.
Warning: Minimum bin size under 1nm. Irregularities may occur.
```

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

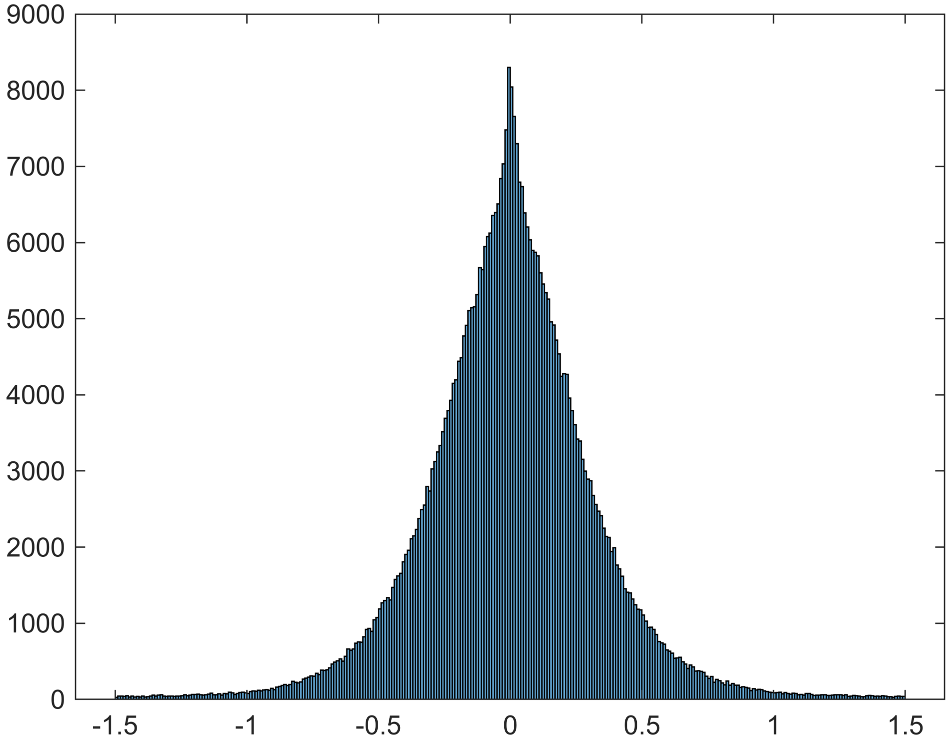
```
figure;
ma1.plotMeanVCorr;
```

```
Computing velocity autocorrelation of 30713 tracks...
```



Now we may write velocities as well.

```
  v = ma1.getVelocities;
    V=vertcat(v{:});
    edges2 = -1.5:0.01:1.5;
    histogram(V(:,2),edges2)
```


9000
8000
7000
6000
5000
4000
3000
2000
1000
0
-1.5
-1
-0.5
0
0.5
1
1.5