This Report generated at

```
tstamp = datetime('today', 'InputFormat','yyyy-MM-dd');
display(datetime)
```

```
datetime

 27-Feb-2025 13:17:08
```

Define the source folder here:

```
main_folder = "\
\nsrg.cs.unc.edu\nanodata2\Alexander_Marshall\Data_Storage\25.02.27_OpenFlexure_part
icle_tracking_1um_beads_30fps\tracks"
```

```
main_folder = 
"\\nsrg.cs.unc.edu\nanodata2\Alexander_Marshall\Data_Storage\25.02.27_OpenFlexure_particle_tracking_1um_beads_30fps\
```

If you want to scale the tracks, set the scaling factor here. Otherwise, set scaling to -1 to avoid scaling.

```
scaling = 1
```

```
scaling = 1
```

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.
Warning: scaling factor active!
found 544 tracks in the file.
Warning: scaling factor active!
found 507 tracks in the file.
Warning: scaling factor active!
found 553 tracks in the file.
found a total of 553 tracks in the directory
Warning: plotting only tracks longer than threshold length
Computing MSD of 1030 tracks...  1Done.
```

You may override space and time units below as desired. Note that we currently default to frames in pretty much all cases.

```
SpaceUnits = ma1.space_units;
TimeUnits = 'seconds';
display(SpaceUnits)
```

```
SpaceUnits = 
'pixel'
```

```
display(TimeUnits)
```

```
TimeUnits = 
'seconds'
```

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

```
dt=0.033
```

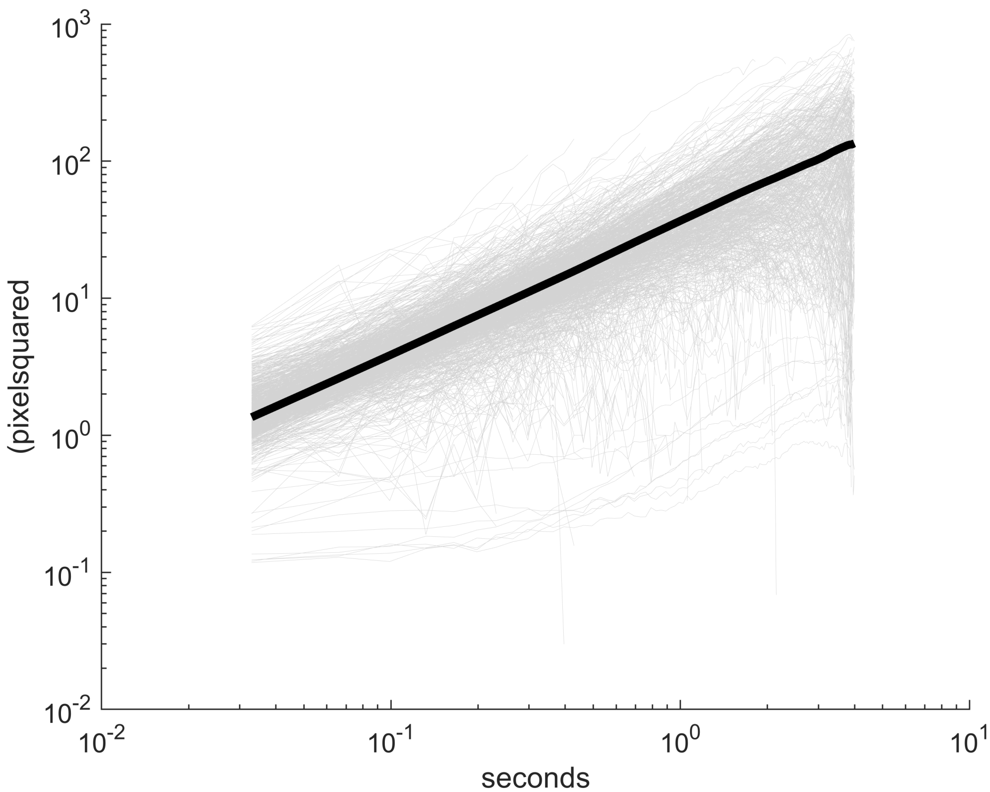
```
dt = 0.0330
```

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

```
D_expect=-1
```

```
D_expect = -1
```

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

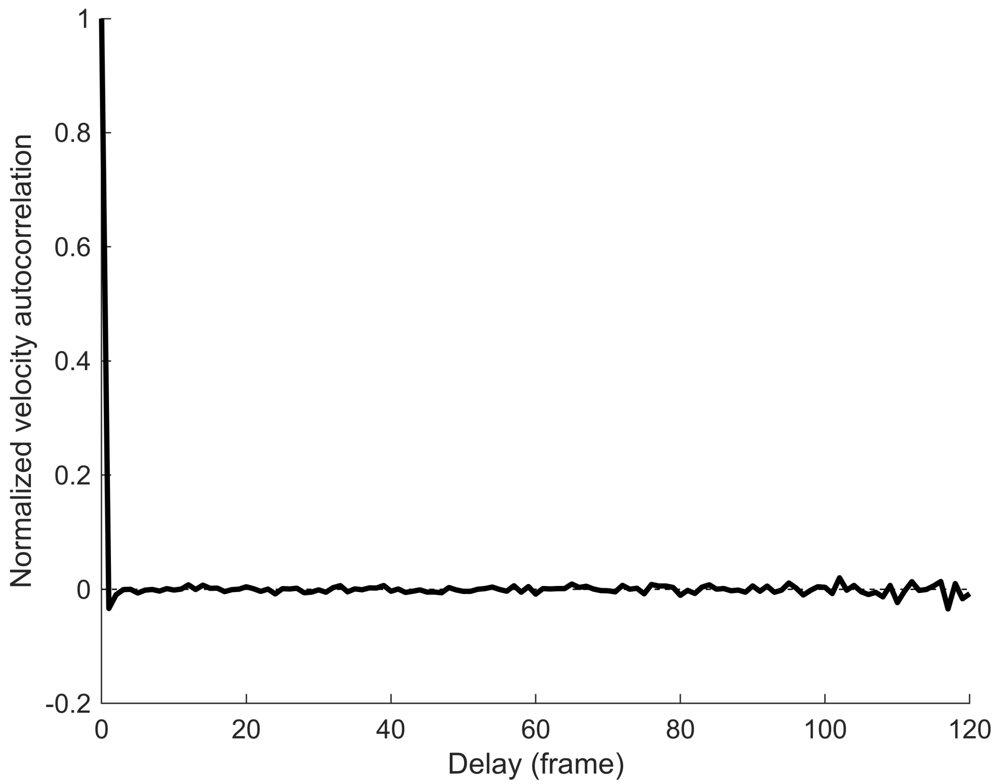



```
ans = 122×5
         0         0         0  699.1324         0
    0.0330    1.3557    0.4548  680.7487   -0.1320
    0.0660    2.6043    0.7820  667.7748   -0.2640
    0.0990    3.8312    1.1360  661.9075   -0.3960
    0.1320    5.0543    1.5145  657.2216   -0.5280
    0.1650    6.2759    1.9159  653.1998   -0.6600
    0.1980    7.4671    2.3415  647.9699   -0.7920
    0.2310    8.6536    2.8166  642.2696   -0.9240
    0.2640    9.8423    3.2727  636.8660   -1.0560
    0.2970   11.0317    3.7663  631.7484   -1.1880
       ⋮
```

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

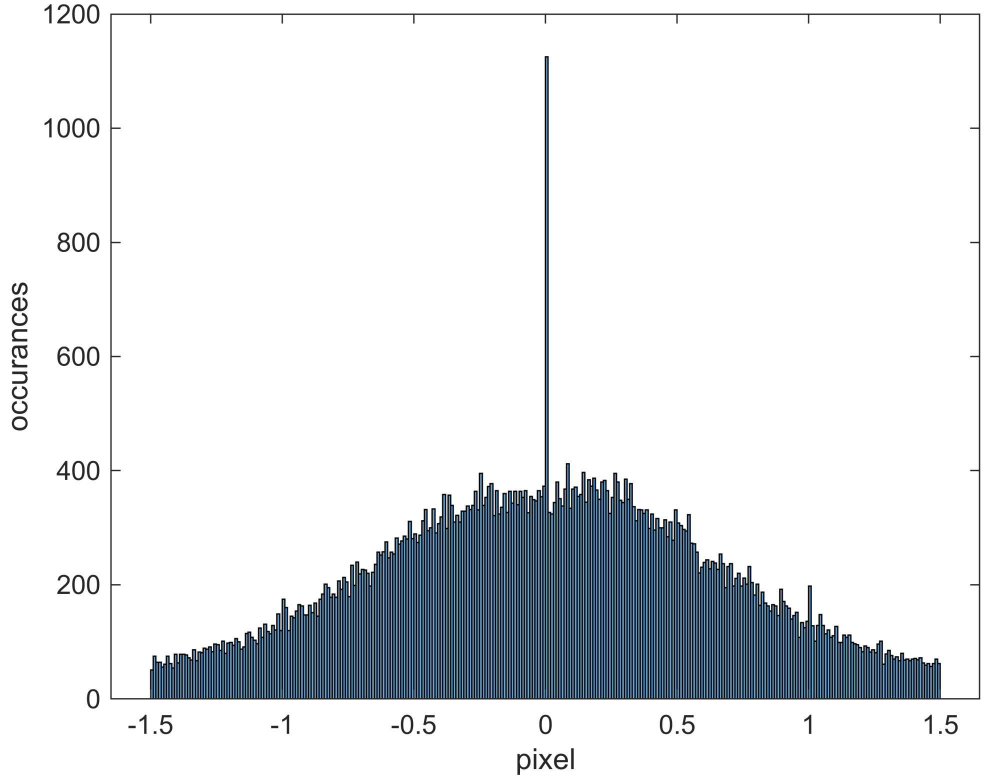
```
figure;
ma1.plotMeanVCorr;
```

```
Computing velocity autocorrelation of 1030 tracks... Done.
```



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

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

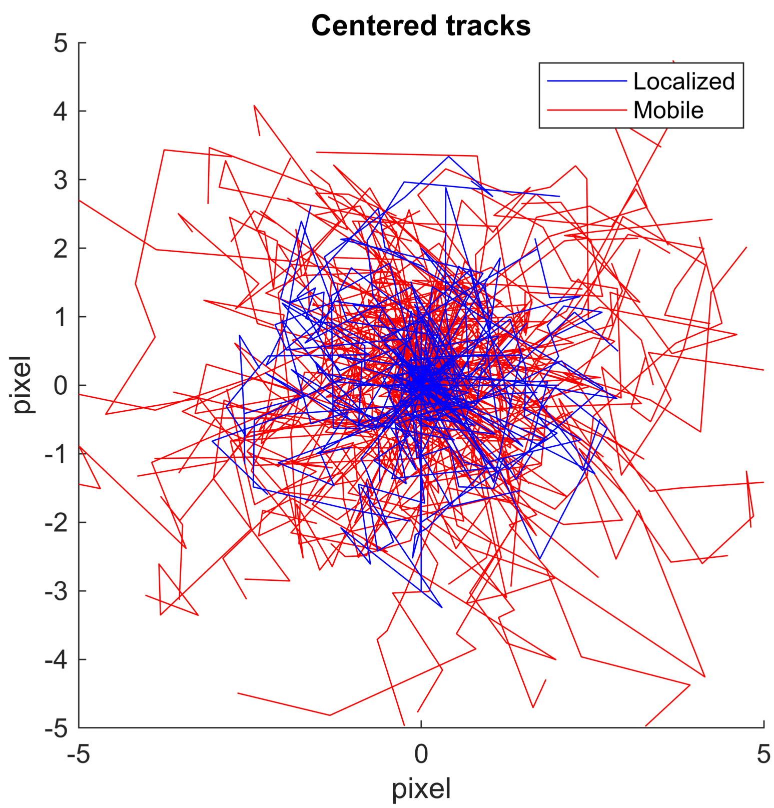


Now, we must call CenterTracks.

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

We may now create a short vs long track diagram.

```
UniDomainFigCenterJuxt(SpaceUnits, CenterTracks, 100,100, 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 = 30
```

```
BinSize = 30
```

```
MinBin = 0.2 % In microns
```

```
MinBin = 0.2000
```

```
[CenterPoint,TotStepCount,VanHoveData] = CreateVanHovePlots(SpaceUnits, TimeUnits,
ma1.tracks, BinSize, [1,2,5,10], main_folder, MinBin, dt)
```

```
 Calculating VanHove Distribution dt=1

1000
 Calculating VanHove Distribution dt=2

1000
 Calculating VanHove Distribution dt=5
```

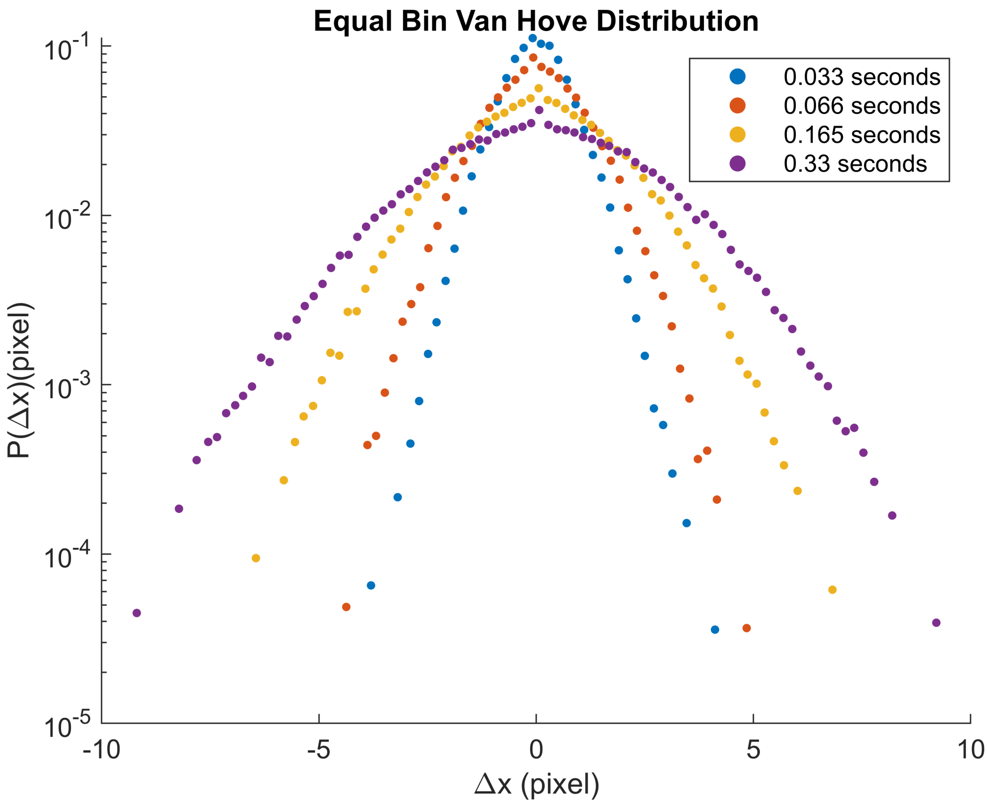
```
1000
 Calculating VanHove Distribution dt=10

1000
 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
```



CenterPoint = 1×10 cell

...

	1	2	3	4	5	6	7	8	9
1	1×35 double	1×43 double	[]	[]	1×61 double	[]	[]	[]	[]

TotStepCount = 4×2 table

	CellNumber	Value
1	1	68469
2	2	66955
3	5	64108
4	10	59485

VanHoveData = 1×10 cell

...

	1	2	3	4	5	6	7	8
1	68469×4 double	66955×4 double	[ ]	[ ]	64108×4 double	[ ]	[ ]	[ ]

Save this file as a pdf.