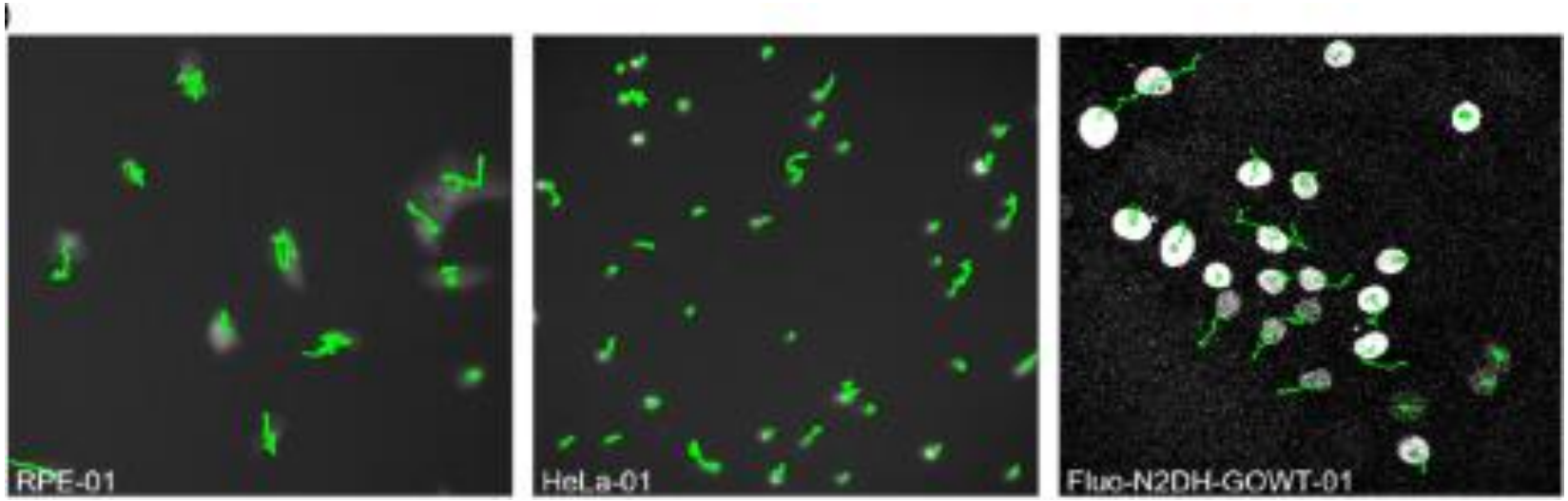
Week 11

Cell tracking techniques



Last week
Open images with matlab
Reading image and video files
Thresholding
Convolution

Segmentation

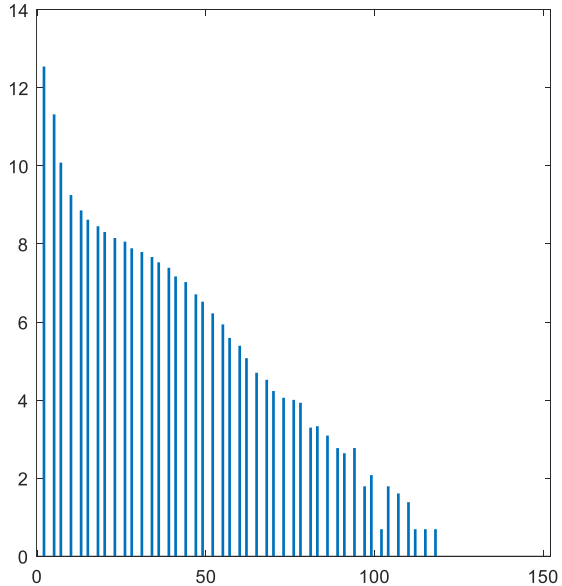
1. Thresholding
2. Watershed

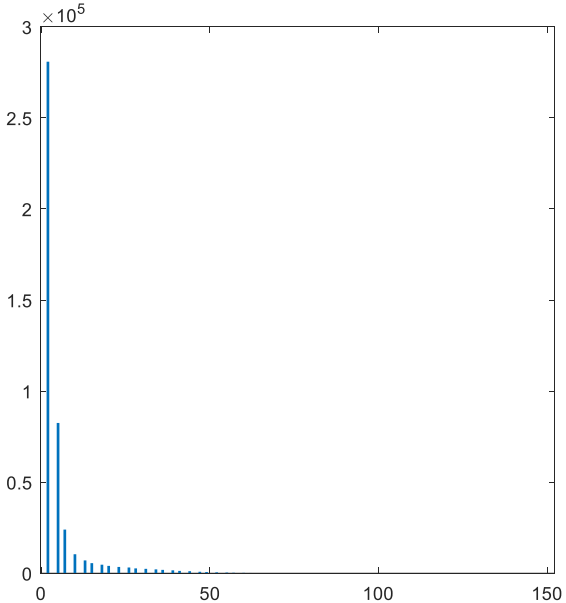
Image histograms

Functions: reshape, find, histcounts

```
%%
x= reshape(imgLD{1}(:,:,1),1,[]);
y=sort(x,'ascend')

[r,c,l]=find(y>0)

edges=1:1:Lcutmax
n=histcounts(y(1,c(1,1):end),edges)

figure(2)
bar(edges(1:151), log(n))
```

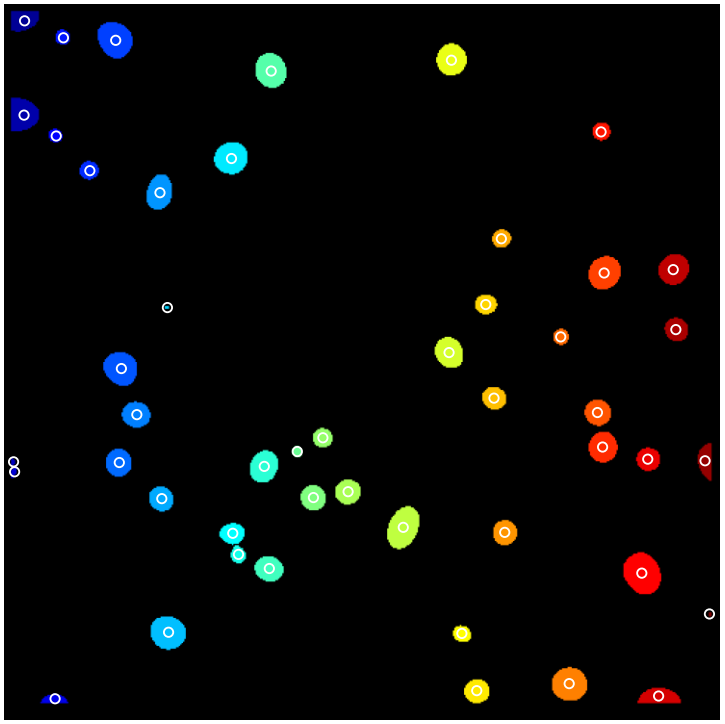
```
[r,c,l]=find(y>0)

edges=1:1:Lcutmax
n=histcounts(y(1,c(1,1):end),edges)
%%
figure(2)
subplot(1,2,1)
bar(edges(1:151), log(n))
subplot(1,2,2)
bar(edges(1:151), n)
%%
```



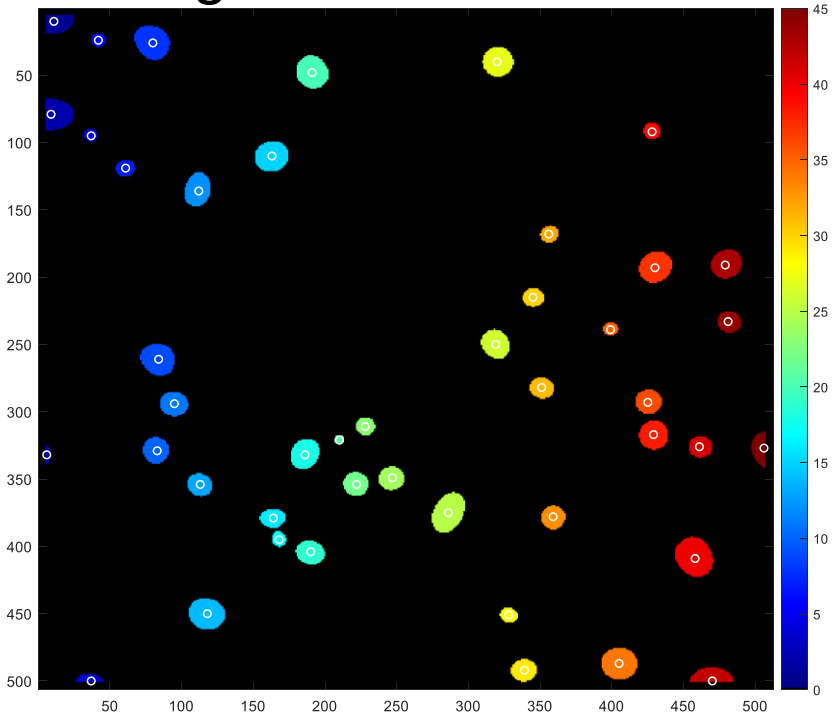


Segmented images

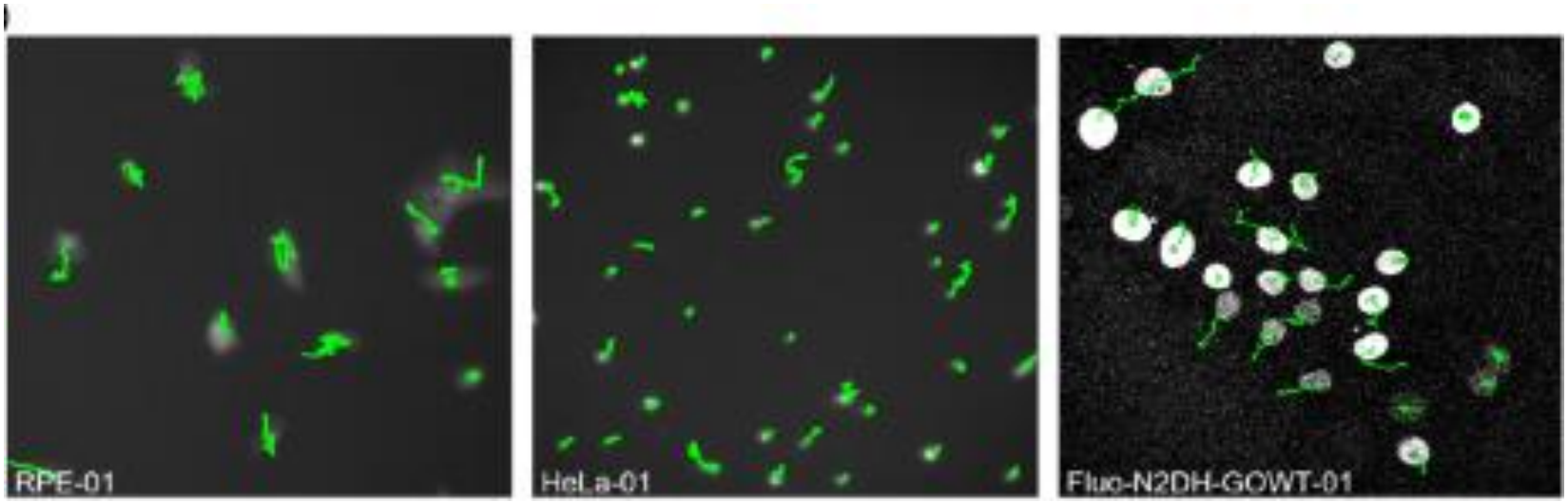
Watershed method



Traditional method Thresholding, gaussion filtering



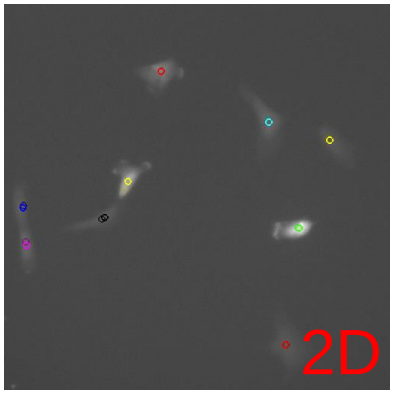
Cell tracking for cell types

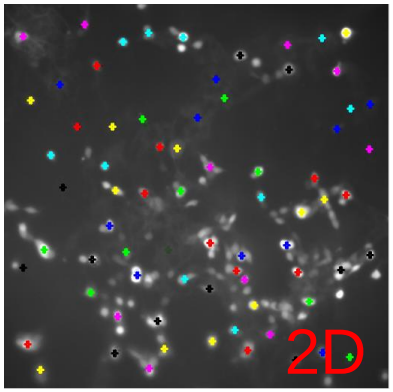


Qureshi et al Comp Biol and Medi 2022

Imaging and tracking at micro and nanoscale

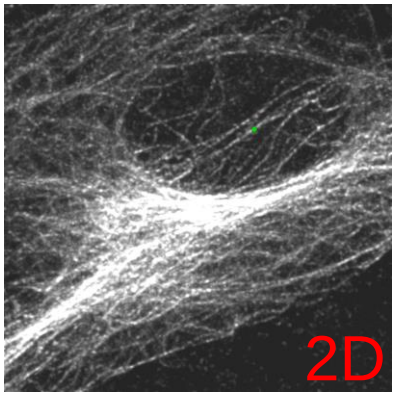
Micro: Average cell size from 10 to 100 µm

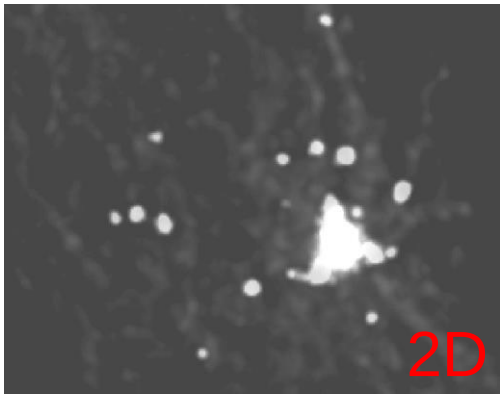




Shah et al Nature comm 2019

Nano: Subcellular components their average size from 40 to 300 nm





Conkar et al, Scientific reports, 2019

Data growth and challenges

- Sample preparation and imaging

- Analysis, cell properties or dynamics

High content image analysis

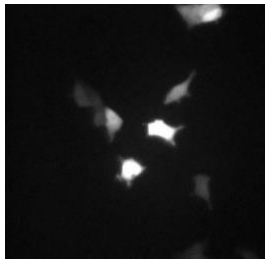
	Perimeter	Ellipticity	Intensity	…….
Cell 1	100	1.2	5	6
Cell 2	120	1.3	8	10
…….	88	2.4	6	13

- Exponential data growth: State or the art imaging methods provides GB to TB of data per run or day.

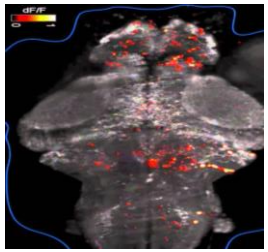
Desired features for image analysis pipelines

- Automated systems
- Minimum user intervention
- Reliable and accurate results
- Fast analysis time
- Handle the image&video data acquired with different imaging conditions

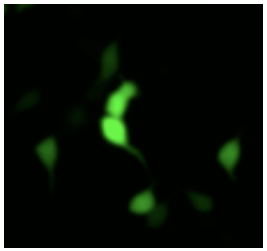
Why is the image processing intrinsically difficult?

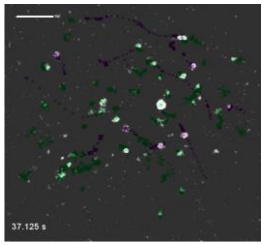


Intensity levels

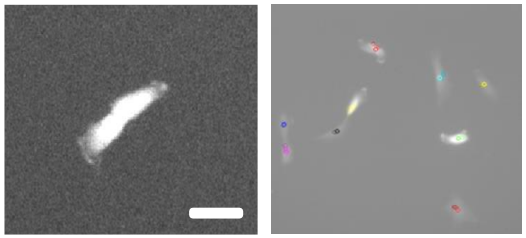


Ahrens et al 2013



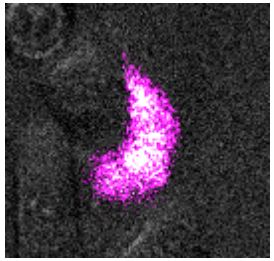


Conkar et al 2019



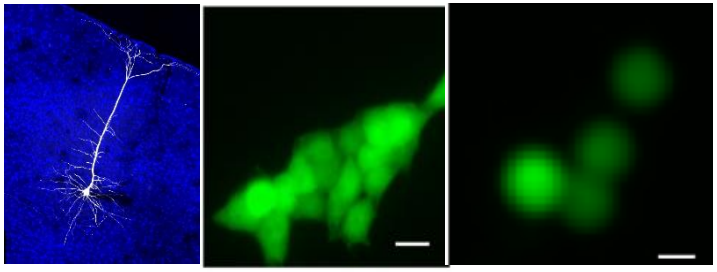
Cell splitting and apoptosis

Micro — Object size — Nano

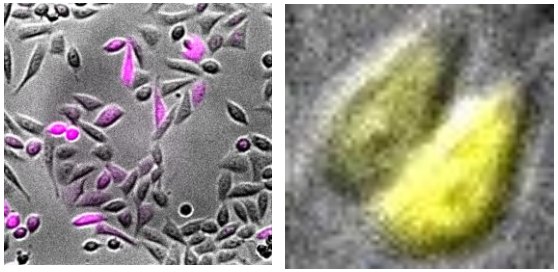


Photodamage

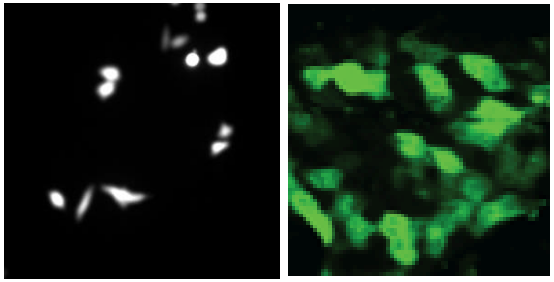
- Image analysis



Shape

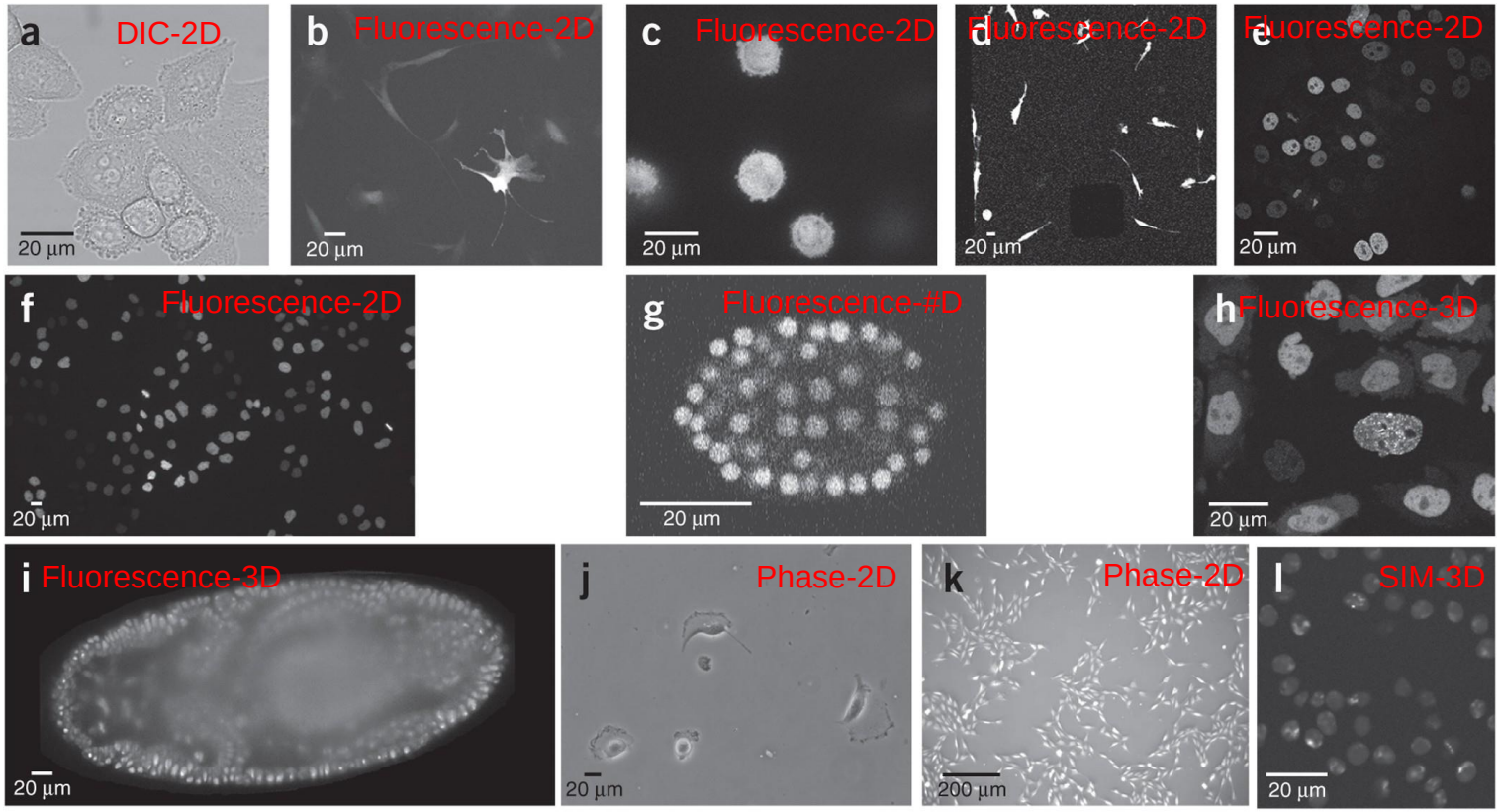


Uneven labeling



Confluency

Image data types are very diverse.



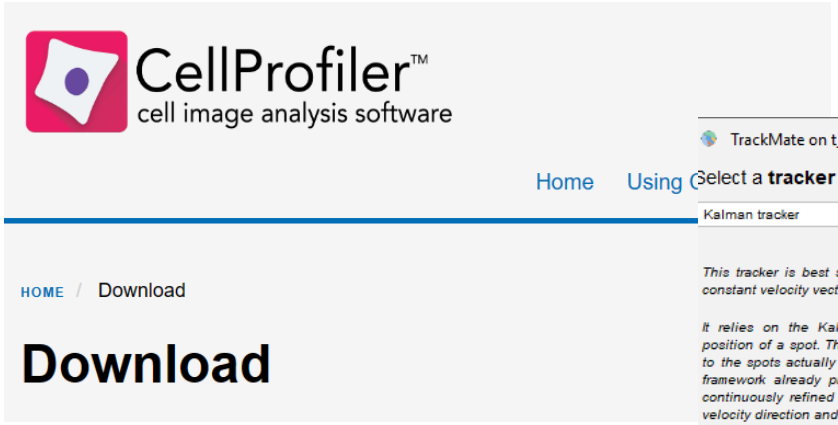
Ulman et al Nature Methods 2017

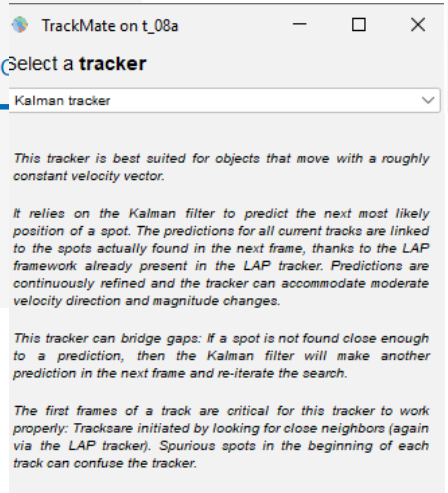
- There is no universal algorithm that can reliably analyze all of the images shown above.

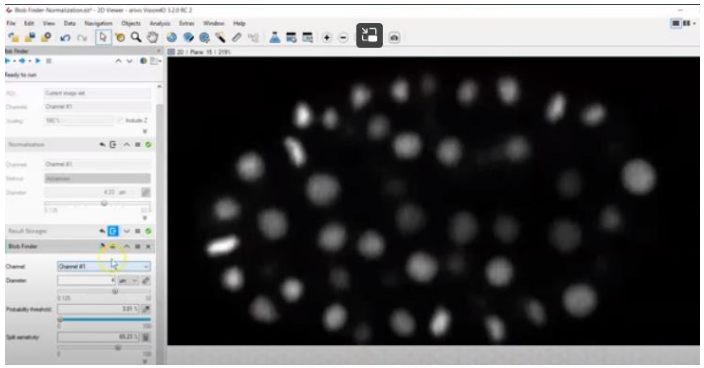
Current commercial or open source image analysis tools

Commercial programs
Imaris, Volovity, ArivisVision4D

Open source software
Cellprofiler, TrackMate, MtrackJ, CellTracker, ilastik







Comparison of open source image analysis tools

Table 1 Comparison of features in tTt/qTfy and existing software for cell tracking and fluorescent marker quantification in long-term time-lapse microscopy

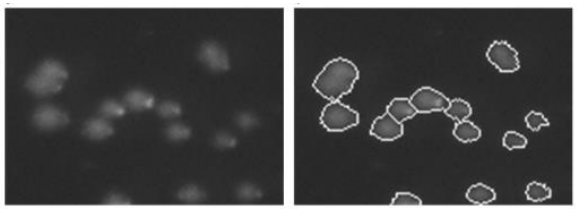
Feature required for	Tool feature	ImageJ[26] (v1.50e, Fiji distribution[15])	Icy[14] (v1.7.3.0)	CellProfiler[17] (v2.1.1)	LEVER[27] (v7.13.2)	tTt, qTfy
Cell tracking in long-term imaging experiments	Process amounts of image data too large to fit in memory	Limited[a]	No	Yes	Yes	Yes
	Support for multi-dimensional image data	Limited[b]	Limited[b]	Limited[b,c]	Limited[b,c]	Yes[d]
	Track cells over multiple overlapping fields of view	Limited[e]	No	Limited[e]	No	Yes[f]
	Import existing trees for manual inspection and correction	Limited[g]	No	No	No	Yes
Quantification of signals from fluorescent markers	Correction of uneven illumination in fluorescence images	Limited[h]	Limited[h]	Limited[h]	No	Yes
	Efficient manual inspection and correction of segmentation and quantification	Limited[i]	Limited[i]	Limited[i]	No	Yes[j]
	Integrated visualization of lineage trees and fluorescent marker time courses	No	No	No	No	Yes[j]
	Integrated workflow for tracking and quantification	No[k]	Yes	Yes	No	Yes
Usability	Expert knowledge required	Yes[k,l]	Yes[l]	Yes[m]	Yes[n]	No
	Processing of image data that cannot be analyzed with existing automatic methods	Limited[l,o]	Limited[l,o]	No[p]	No[p]	Yes

Nature Biotech,703, 2016

Classic and deep learning based cell segmentation methods

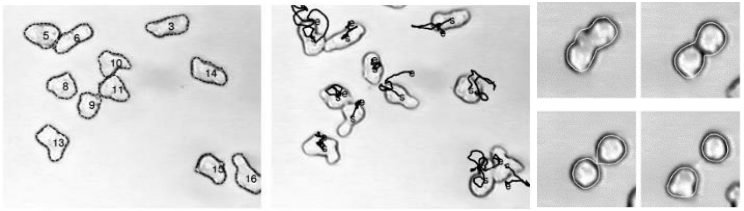
- Classic approaches: Background subtraction, watershed, active contour, n-pixel linkages.

Gradient watershed



Yang et al. Pattern Recog., 2014

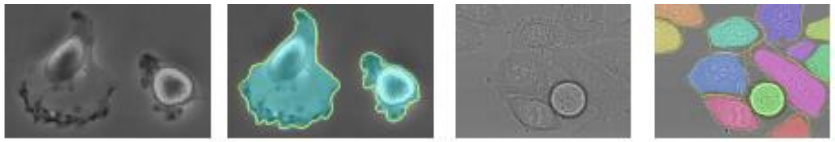
Active contour=snake method



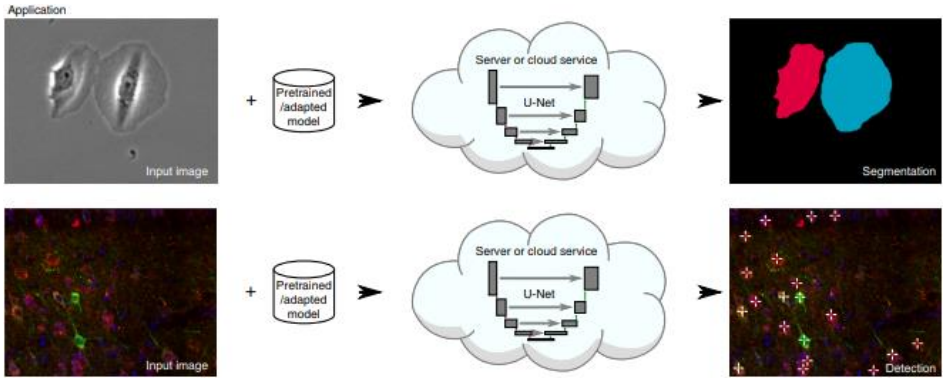
Zımmer et al. IEEE Medi Imaging., 2002

- Deep learning: Neuronal networks, U-net, deepCell.

Convolutional Neuronal Networks



Ronnberger et al. Medi Image, 2015

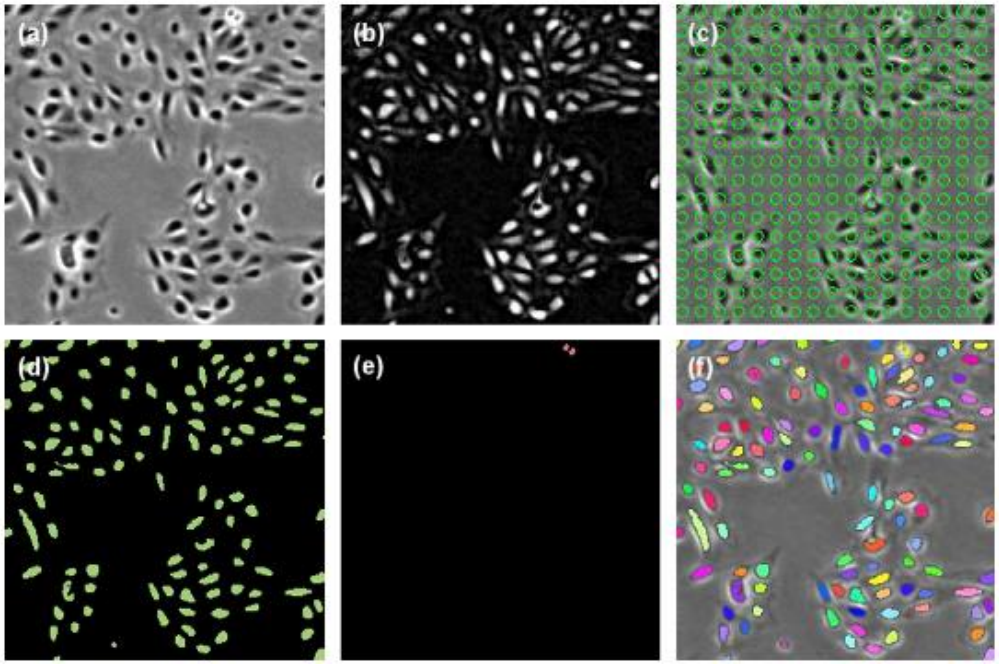


Falk et al. Nature methods, 2019

Cell tracking methods for time-lapse image acquisition

- Outline detection
- Model-based approaches (active contour)

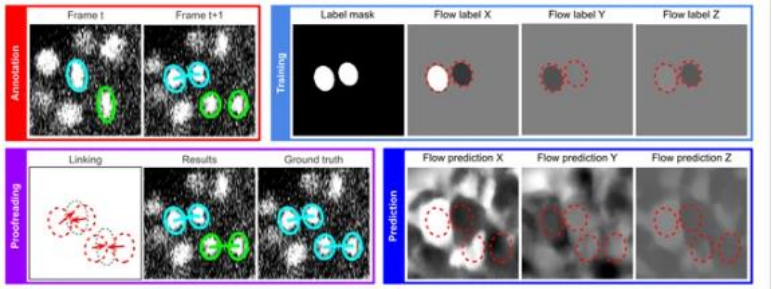
Major issues: Long processing time, tracking errors if cells do not overlap



Li et al Med. Image An. 2008

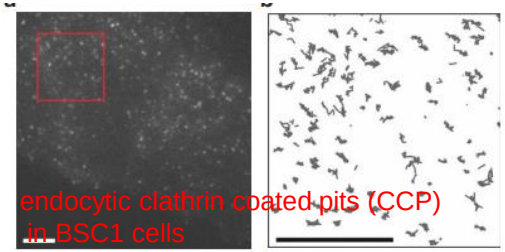
Other cell tracking algorithms

- Unet+optical flow+nearest neighbor



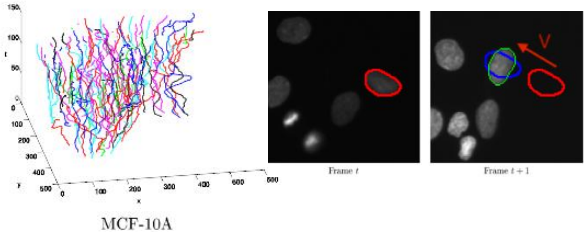
Sugawara et. al. elife, 2022

Linear assigment based tracking



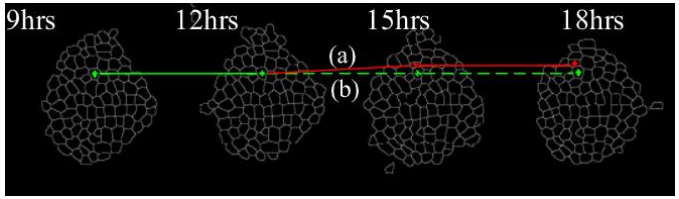
Jaqaman et al Nature methods 2008

- Kalman filter+maximum posterior probability



Assaf et. al. Med. Image Analysis. 2018

Local graph matching with Kalman filter



Liu et. al. Med. Image Vision Computing. 2017

Method name	Segmentation	Pre-track processing	Tracking
Adaptive cell tracking	n-pixel linkage, Bandpass filter	None (Automatic search radius)	Feature-based adaptive nearest neighbor method
[a]KTH-SE	Bandpass filter, global thresholding, watershed	Scaling factor	Global tracking linking, state-space optimization
[a]HD-Har-GE	Gaussian filtering, thresholding, cluster separation	Maximum displacement	Constrained distance based nearest neighbor method
[a]NOTT-UK	Local thresholding	Threshold distance	Distance based nearest neighbor method
[a]UZH-CH	Watershed segmentation with Otsu thresholding	Maximum distance	Distance based nearest neighbor method

Image processing is usually a difficult process. Can we make it simple?

- One way is to use advance programs that can process these images minimum user intervention. Current methods require many parameters being predicted by the user.

Method name	Segmentation	Pre-track processing	Tracking	[b]Number of parameters
[a]KTH-SE	Bandpass filter, global thresholding, watershed	Scaling factor	Global tracking linking, state-space optimization	17
[a]HD-Har-GE	Gaussian filtering, thresholding, cluster separation	Maximum displacement	Constrained distance based nearest neighbor method	9
[a]NOTT-UK	Local thresholding	Threshold distance	Distance based nearest neighbor method	5
[a]UZH-CH	Watershed segmentation with Otsu thresholding	Maximum distance	Distance based nearest neighbor method	5

[a]These segmentation and tracking methods were previously described in detail.[2] [b]The total number of parameters used to evaluate the Fluo-N2DH-GOWT movies at CTC can be found at https://www.nature.com/articles/nmeth.4473.

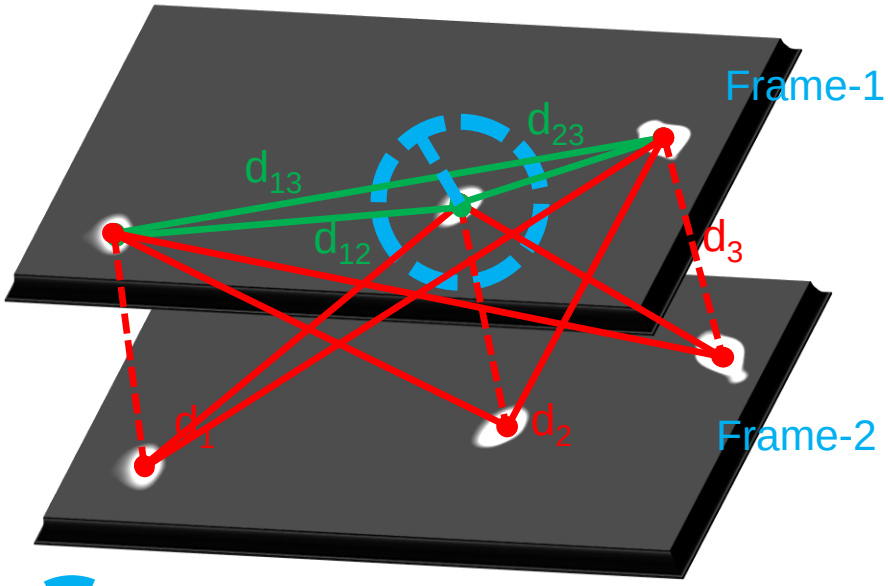
How many parameters are needed to determine the cell trajectories?

- Can we make the tracking simple and intuitive compared to other classical methods?
- Tracking section (many solutions are present from classic to AI driven segmentation)

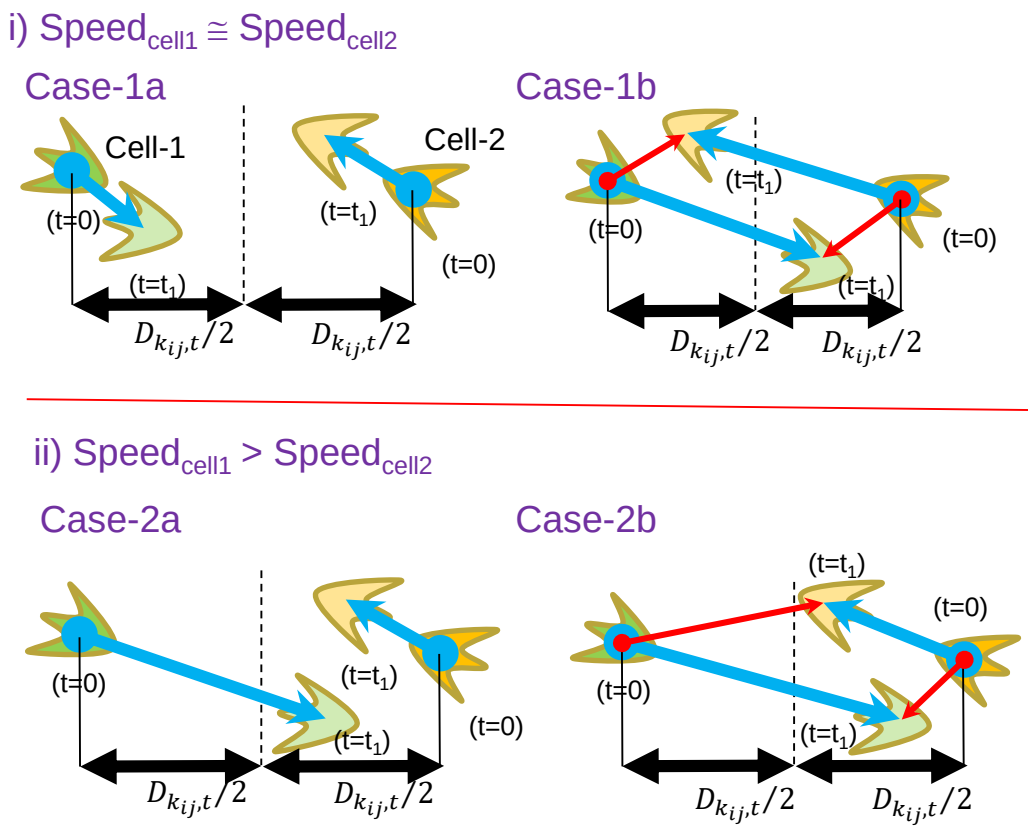
Key parameters for cell tracking

- Maximum search radius
- Gap time
- Filter size
- Intensity threshold
- Average cell size
- Loss time
- …….

Can we automatically determine the maximum search radius? *Ad*aptive *t*racking *algori*thm (Adtari).



Adaptive search radius (threshold), *Ar*(*t*)



$$Ar(t) = \begin{cases} 2 * \max\left(D_{k_i,(t,t+1)}\right), & \text{if } \max\left(D_{k_i,(t,t+1)}\right) < \min\left(D_{k_{ij},t}\right)/2 \\ \min\left(D_{k_{ij},t}\right), & \text{if } \max\left(D_{k_i,(t,t+1)}\right) \geq \min\left(D_{k_{ij},t}\right)/2 \end{cases}$$

<u>Adaptive tracking algorithm (*Adtari*)</u>

Ar(t) < Minimum distance between cells

($min(d_{12}^t, d_{13}^t, d_{23}^t)$)

Ar(t) > Maximum displacement of cells

($max(d_1^{t,t+1}, d_2^{t,t+1}, d_3^{t,t+1})$)

f_{match} = (*Ar(t)*, $dI_{ij}^{t,t+1}$, $dA_{ij}^{t,t+1}$, $dC_{ij}^{t,t+1}$) where i,j=1,2,3,….,n

Qureshi et al Comp Biol and Medi 2022

Adaptive tracking algorithm: Computation of search radius

Summary of adaptive tracking algorithm:

input $D(O_{ij}^{t,t+1})$, $S(O_{ij}^{t,t+1})$ D=distance information
S=shape attributes
for t=1,2,…,*N*
  for *i*=1,2,…..,O_k^t
    if sum($O_i^{t,match}$)==1 **and** sum($O_j^{t+1,match}$)==1
      f_{match}= | D(O_i^t,O_j^{t+1})<Ar(t) |
    else if sum($O_i^{t,match}$)>1 **or** sum($O_j^{t+1,match}$)>1
      f_{match}= | D(O_i^t,O_j^{t+1})<Ar(t) | * | S($O_i^{1\to t}$,O_j^{t+1})< S(t) |
    end
  end
  if $N_i^{match} > Nmin$ *and* $N_i^{lost} < Nmax$
    $M(t,p) = O_i^{match}$
  end
  update Ar(t), S(t)
end

	O_1^{t+1}	O_2^{t+1}	O_3^{t+1}
O_1^t	$D_{1,1}$	$D_{1,2}$	$D_{1,3}$
O_2^t	$D_{2,1}$	$D_{2,2}$	$D_{2,3}$
O_3^t	$D_{3,1}$	$D_{3,2}$	$D_{3,3}$

Ar(t)

	O_1^{t+1}	O_2^{t+1}	O_3^{t+1}
O_1^t	1	0	0
O_2^t	0	1	1
O_3^t	0	0	1

$$Ar(t) = \begin{cases} 2 * \max\left(D_{k_i,(t,t+1)}\right), & if\ \max\left(D_{k_i,(t,t+1)}\right) < \min\left(D_{k_{ij},t}\right)/2 \\ \min\left(D_{k_{ij},t}\right), & if\ \max\left(D_{k_i,(t,t+1)}\right) \geq \min\left(D_{k_{ij},t}\right)/2 \end{cases}$$

Unique match

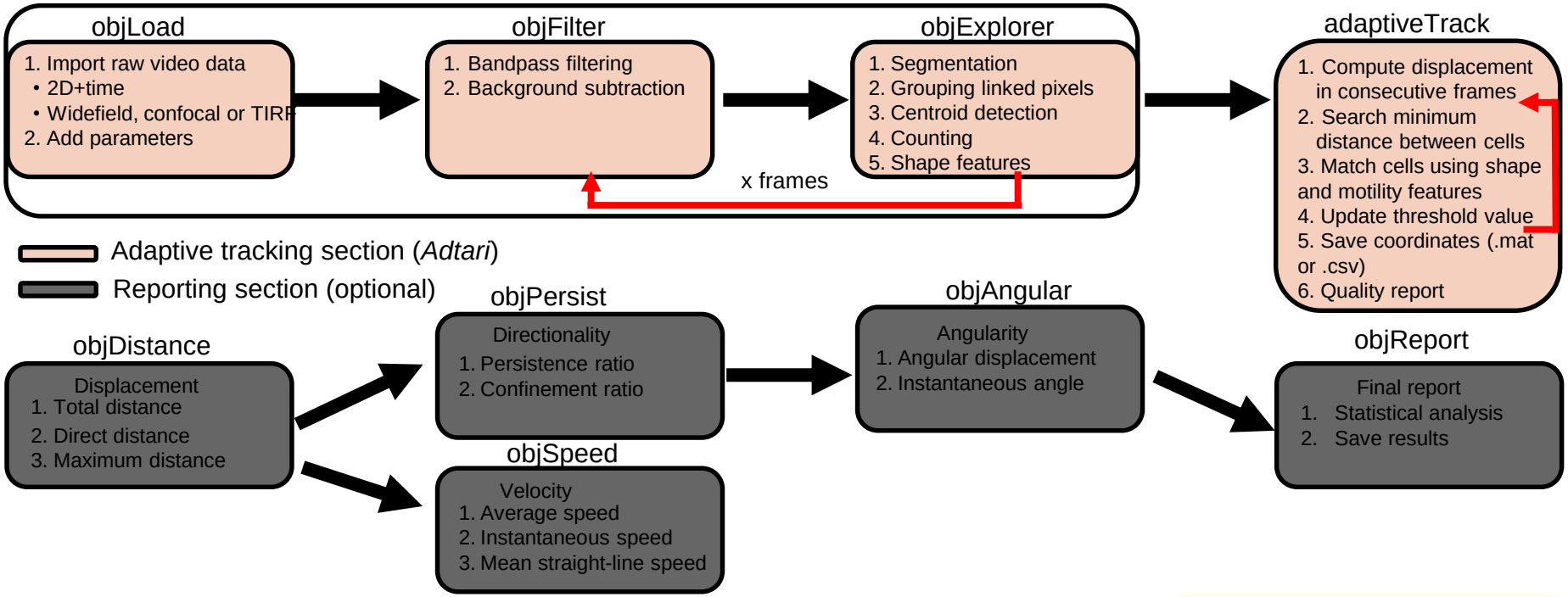
	O_1^{t+1}
O_1^t	1

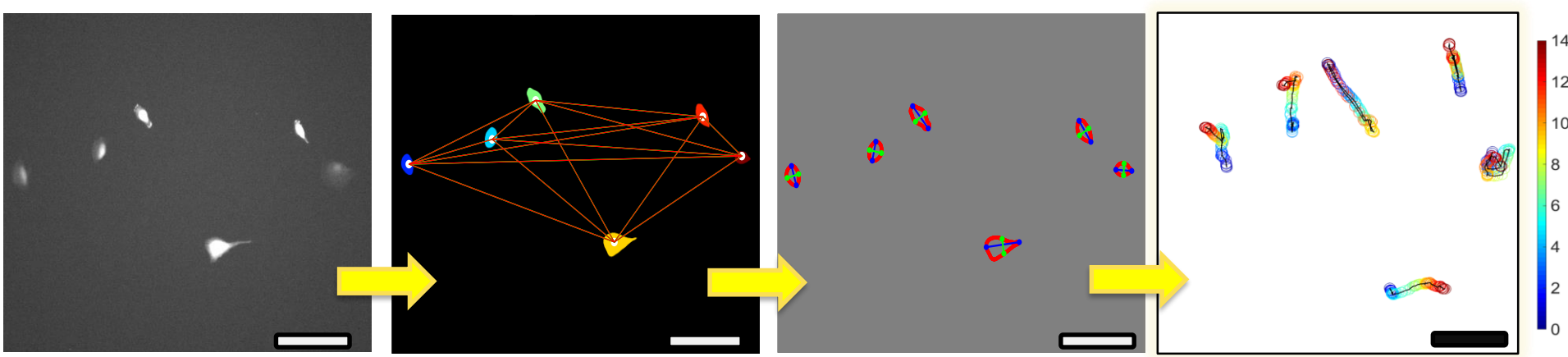
Nonunique match

	O_2^{t+1}	O_3^{t+1}
O_2^t	1	1
O_3^t	0	1

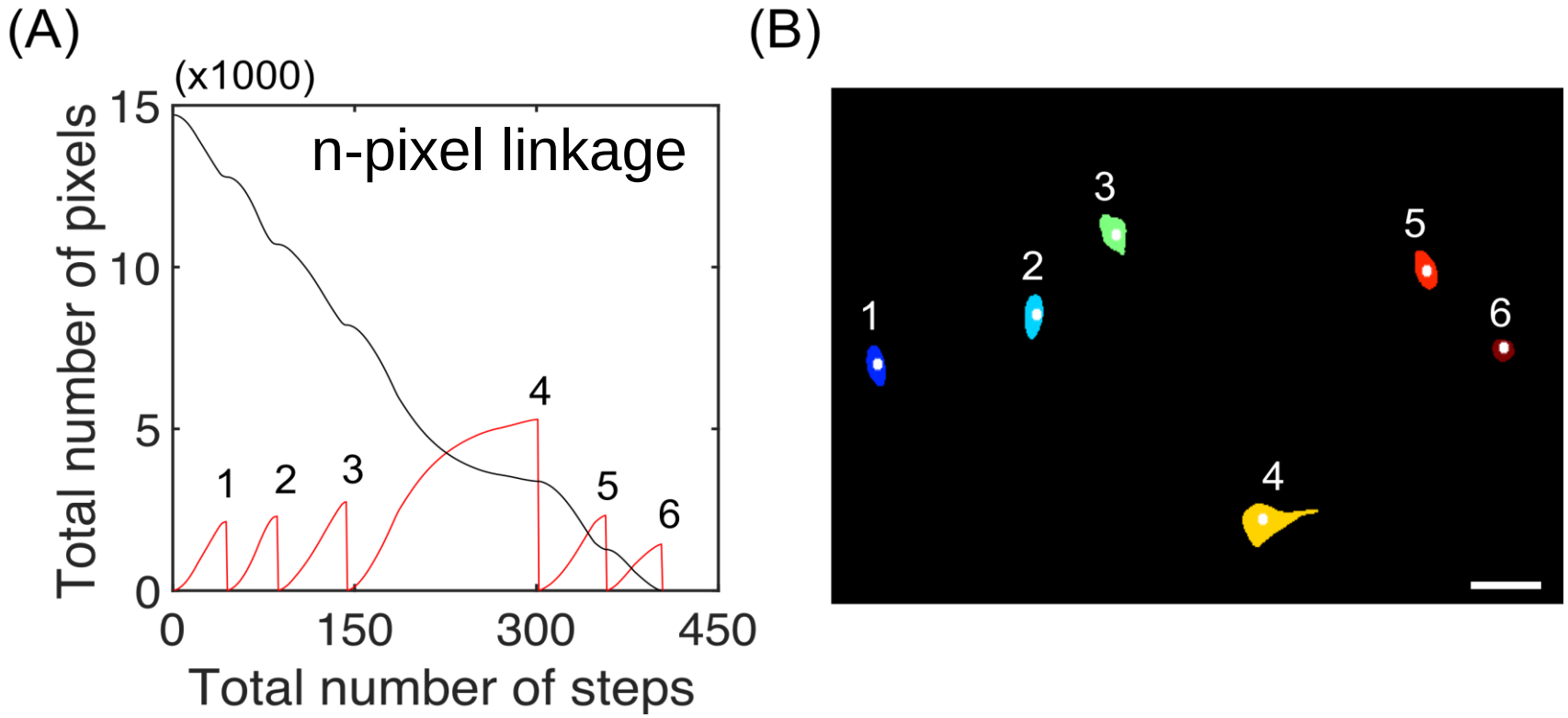
Qureshi et al Comp Biol and Medi 2022

Workflow of adaptive tracking method to determine cell trajectories and motility dynamics



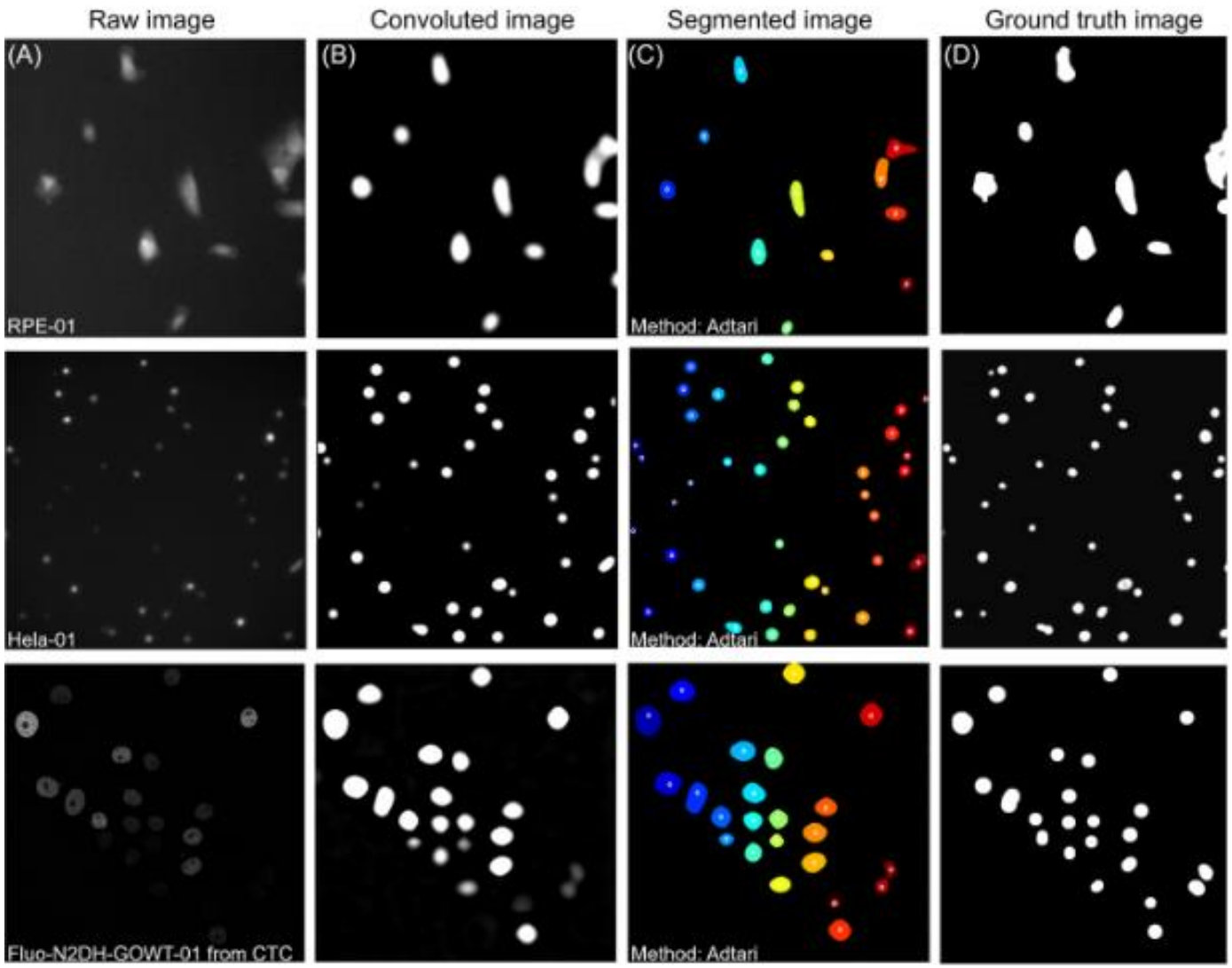


Cell segmentation analysis for detected pixels after applying threshold filter



Qureshi et al Comp Biol and Medi 2022

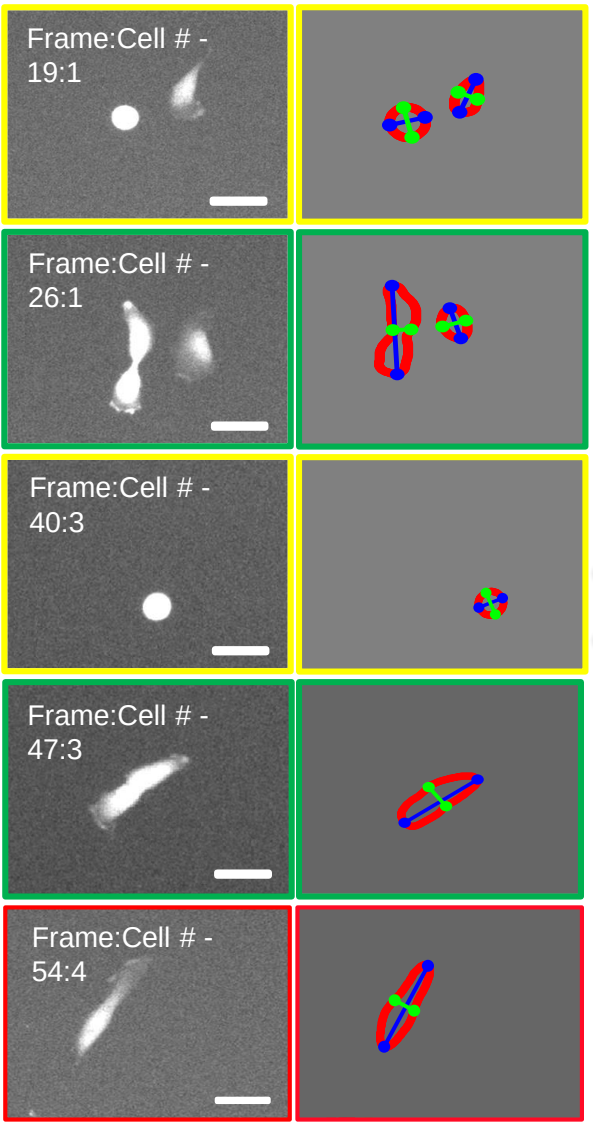
Segmentation results and extraction of key attributes for different cell types

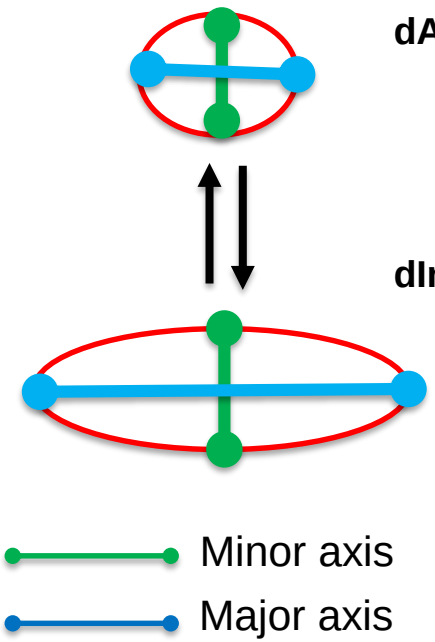


Features:

- Area
- Perimeter
- Ellipticity
- Axis length
- Intensity
- Circularity
- …….

Time course of geometric events: event map generation for identification of cell splitting

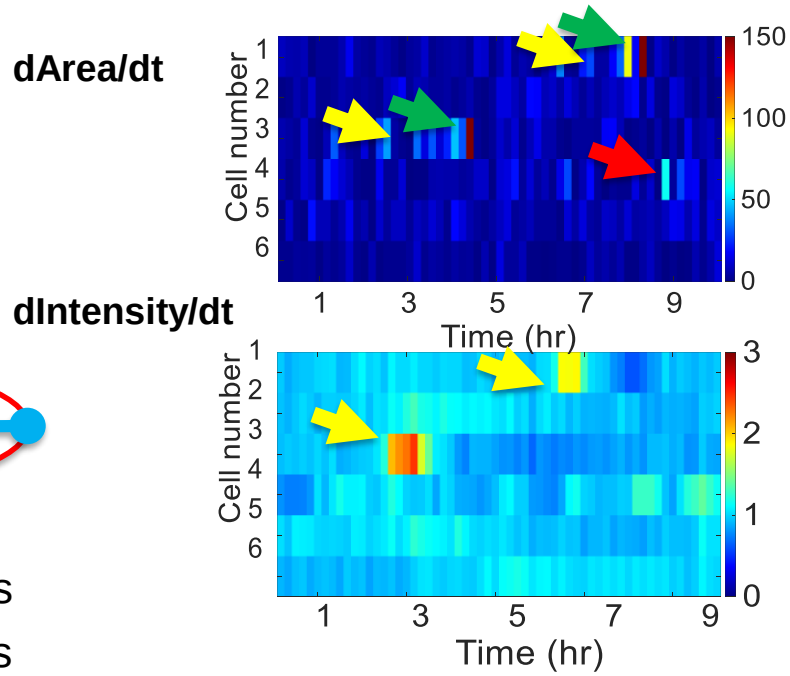




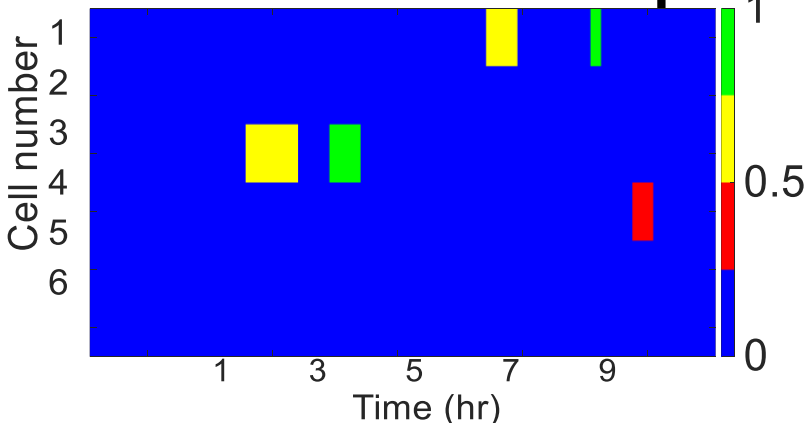
$$C_i^k = \frac{(P_i^k)^2}{4\pi(A_i^k)}$$

Circularity

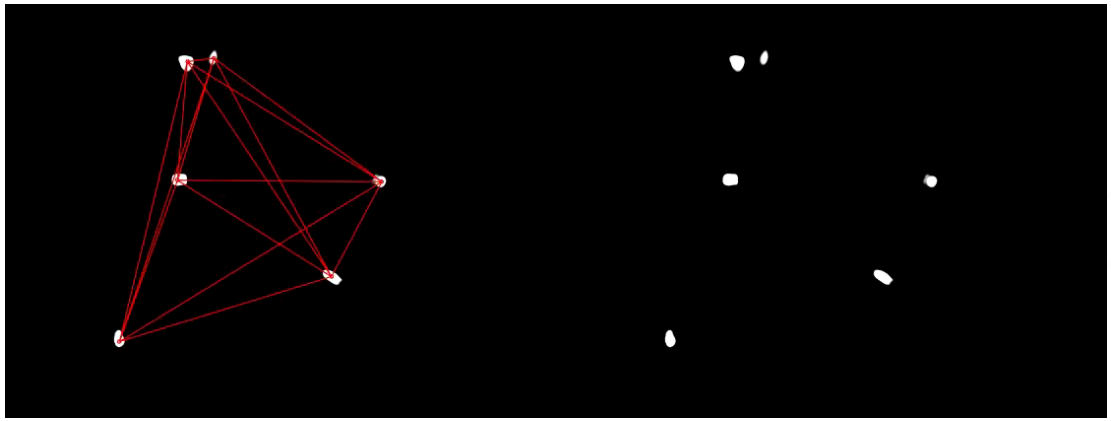
Elongation

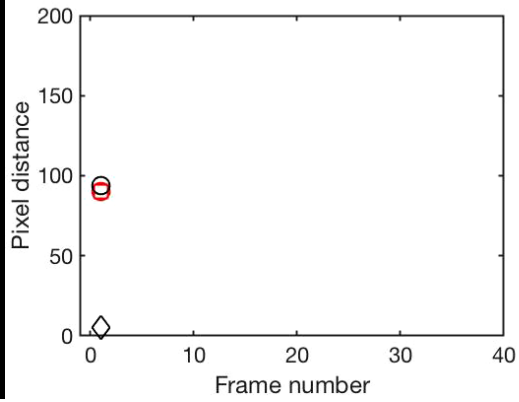


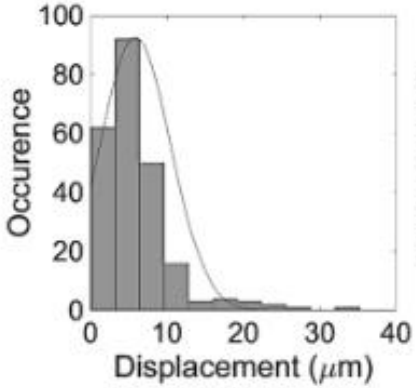
Geometric Event Map

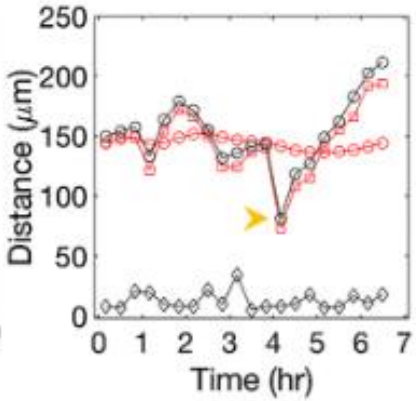


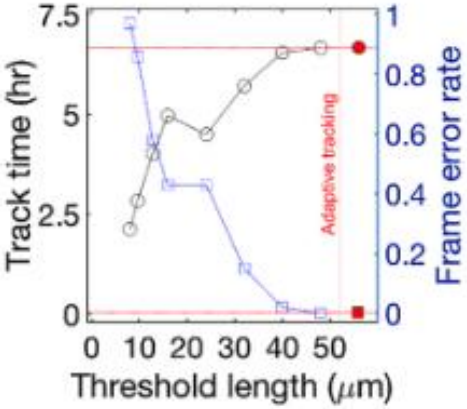
Example cell trajectories from adaptive tracking method

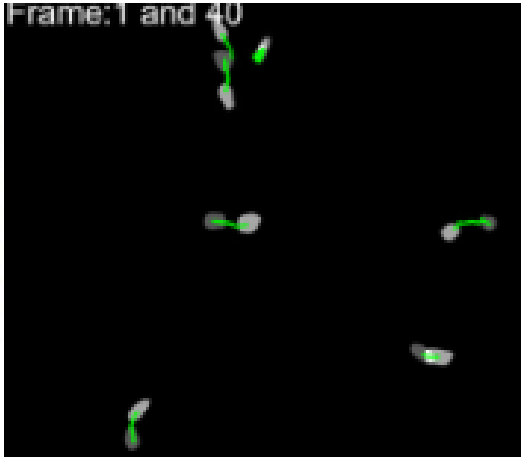






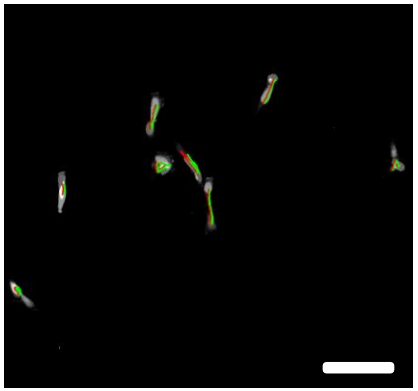


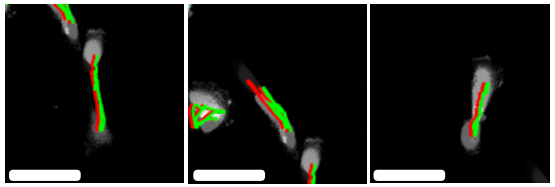




Qureshi et al Comp Biol and Medi 2022

Performance evaluation of *adaptive tracking* in time-lapse videos

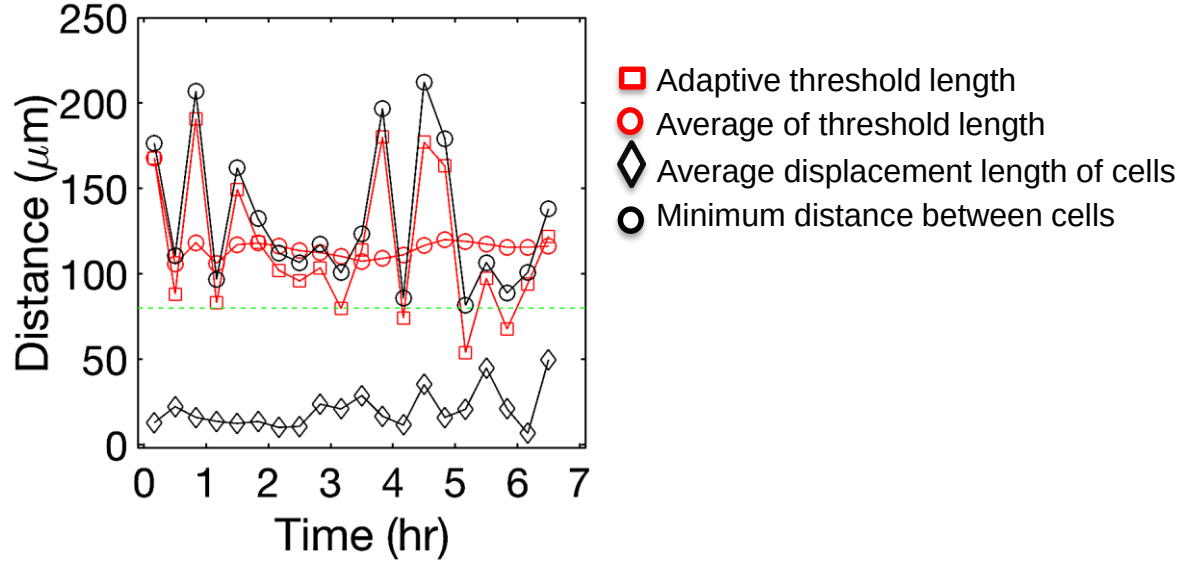


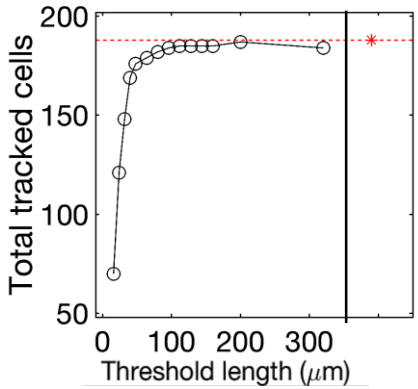


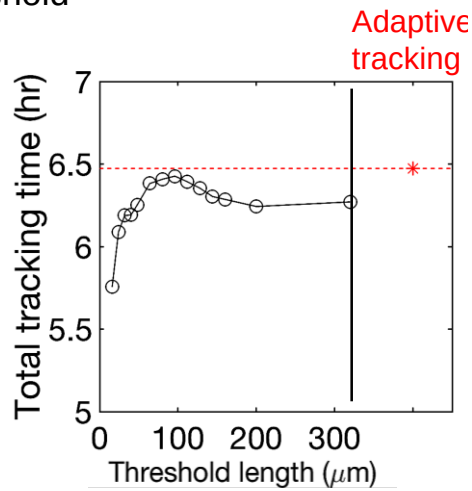
Ratio=1.25 Ratio=1.25 Ratio=1.0

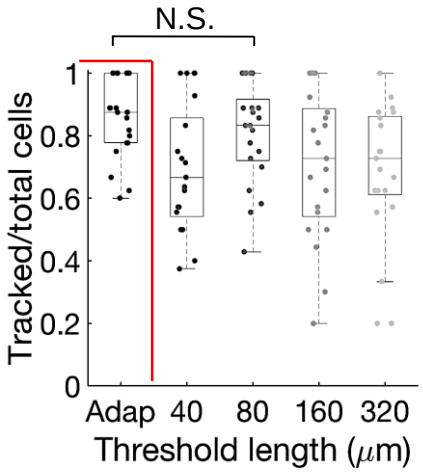
Adaptive tracking, variable threshold

Constant threshold

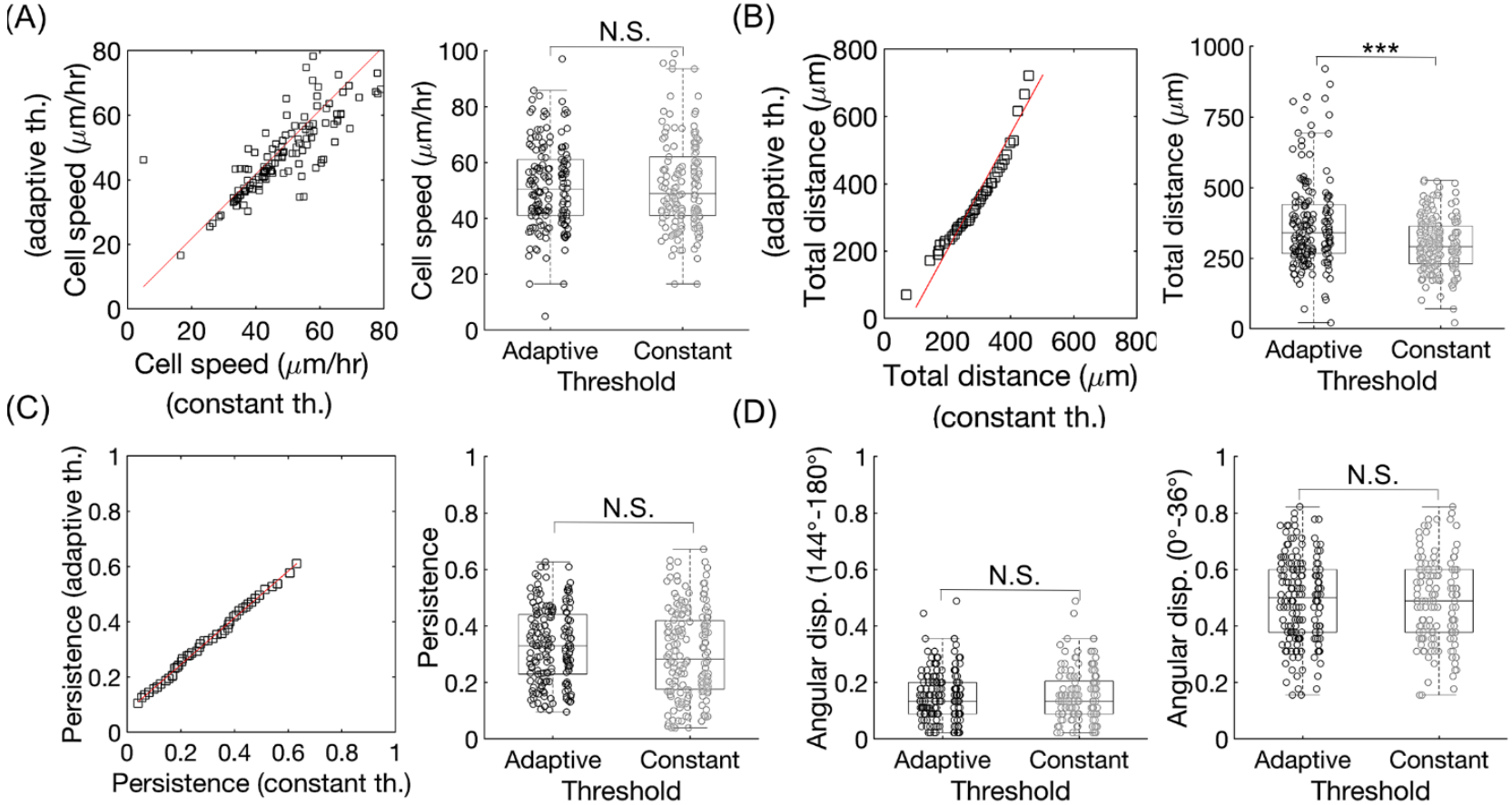




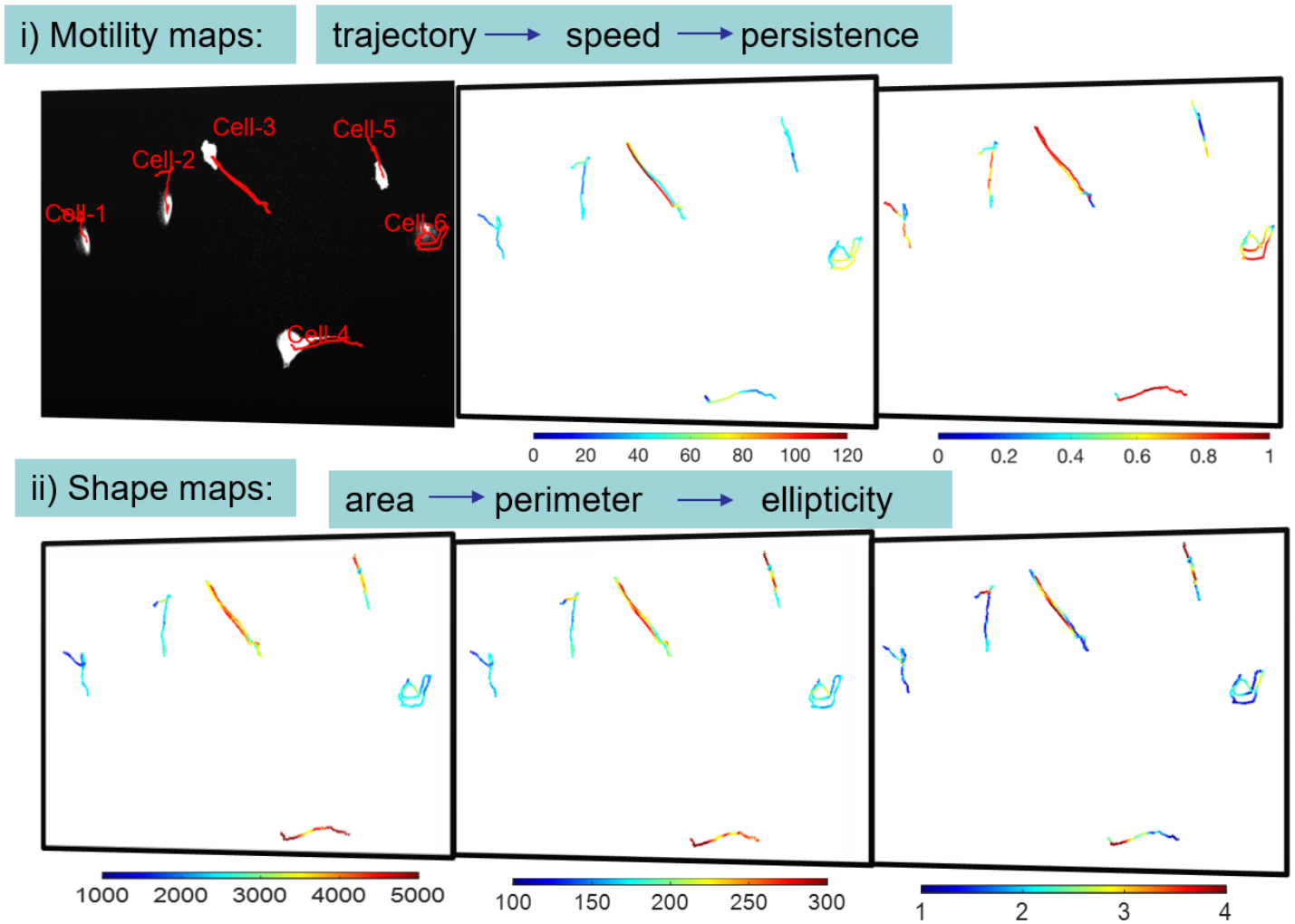




Correlation analysis of motility dynamics of GFP labelled RPE cell trajectories obtained from adaptive and constant distance methods

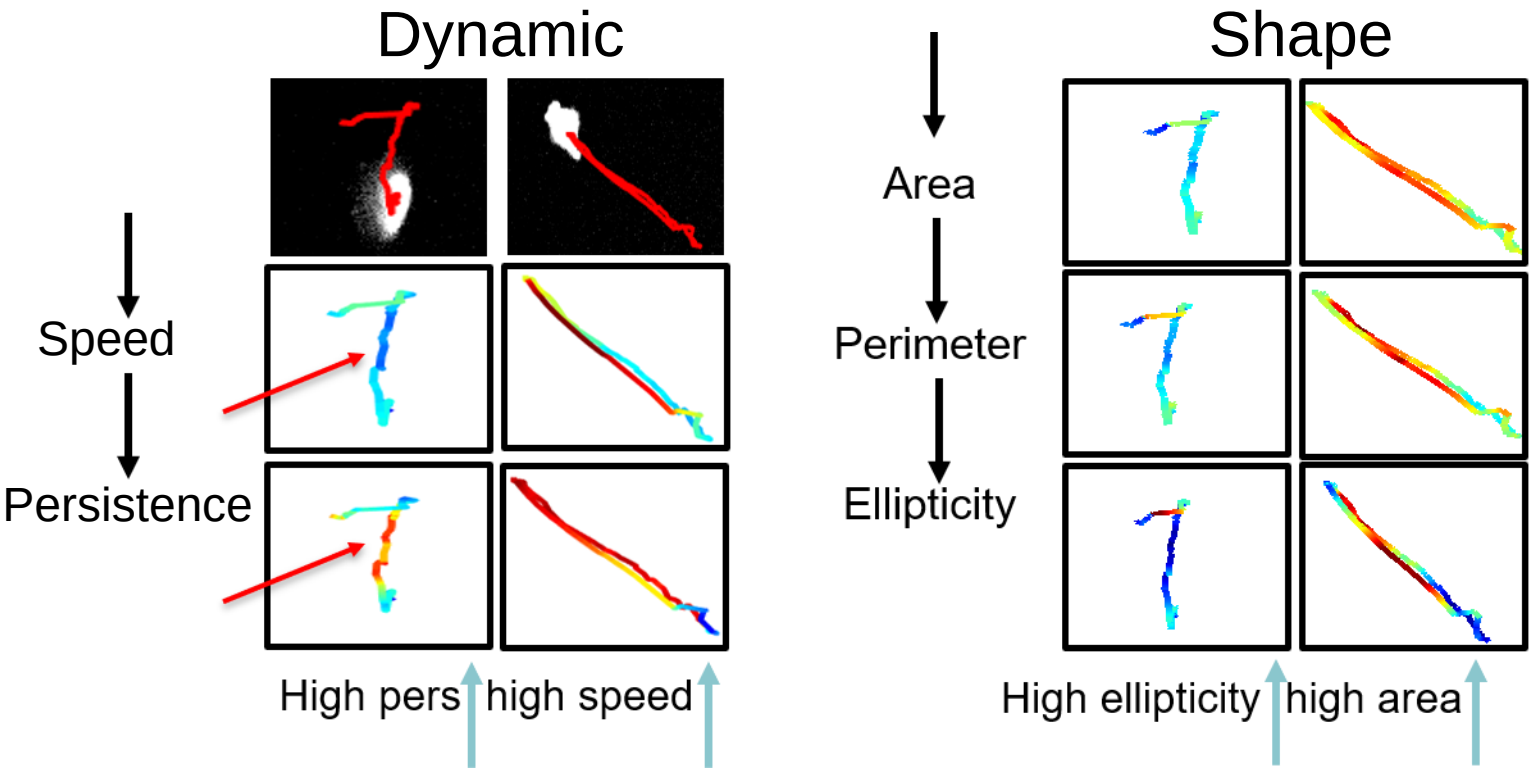


Layer-by-layer assessment of single-cells attributes: Motility maps

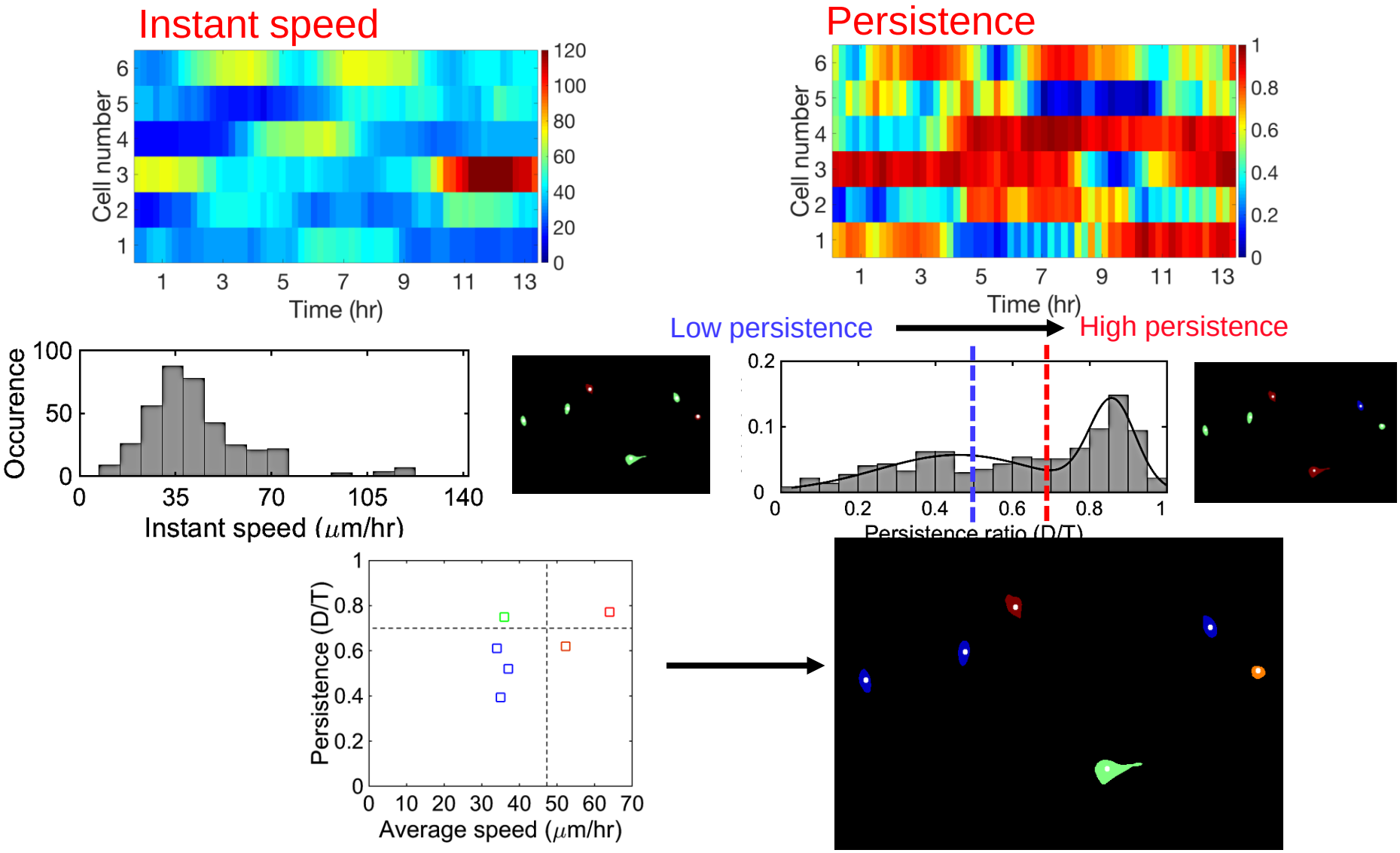


Qureshi et al Comp Biol and Medi 2022

Cell phenotype quantification: Single cell comparison of shape and motility features



Classification of cells based on instant speed and persistence



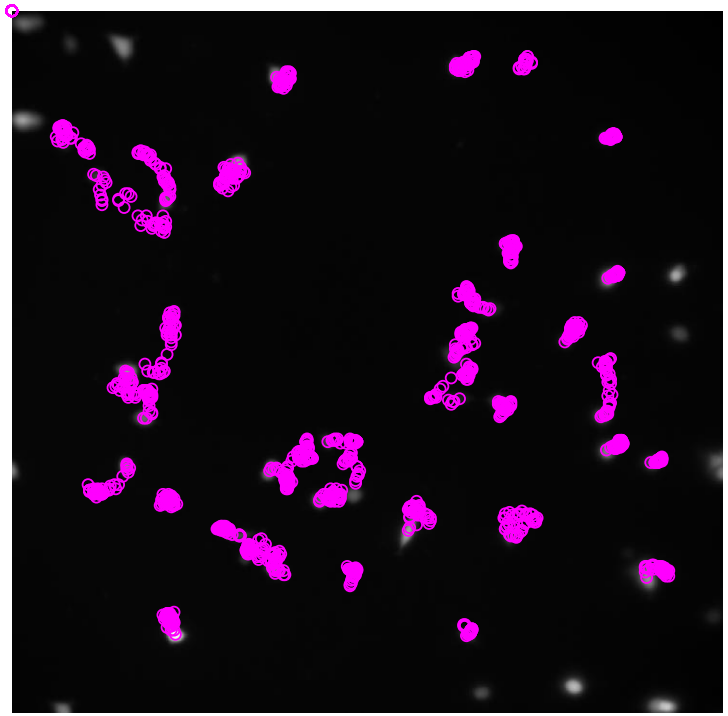
Example Codes

```
%%
% tracking methods


katdegeri=0.48
avecellsize=30;
[cllCntrsFn]=QuickExplorer(imageOut3,katdegeri,avecellsize);

%avecellsize=30
%cllCntrsFn]=QuickExplorer(imageOut, 2.8 ,avecellsize)
[Trackreport5]=adaptiveTrack15(cllCntrsFn,3,8)
size(Trackreport5{1,4})
```

```
%%


[Trackreport5]=adaptiveTrack15(cllCntrsFn,10,12)
size(Trackreport5{1,4})
```

```
%%
```

```
[Trackreport5]=adaptiveTrack15(cllCntrsFn,10,12)
size(Trackreport5{1,4})

%%
sel=1
se=40
figure(121)
%imshow(imageOut3{10,1}(:,:,1),[0 7])
hold on
for j=sel:sel;
imshow(imgLD{1,1}(:,:,j),[])
hold on
for i=1: size(Trackreport5{1,1},2);
   plot(Trackreport5{1,1}(sel:se,i),Trackreport5{1,2}(sel:se,i),'om', 'linewidth',1)
  hold on

end
pause(0.1)
j
end
```



Saving the data

```
%%
ns=14
Trackreportfinal{1,ns}=Trackreport3{1,1}
Trackreportfinal{2,ns}=Trackreport3{1,2}

Trackreportfinal{3,ns}=Trackreport3{1,3}
Trackreportfinal{4,ns}=Trackreport3{1,4}

Trackreportfinal{6,ns}=Afr
Trackreportfinal{7,ns}=ArY
Trackreportfinal{5,ns}='movie44'
save('ensondensitydata','Trackreportfinal')
%%
```