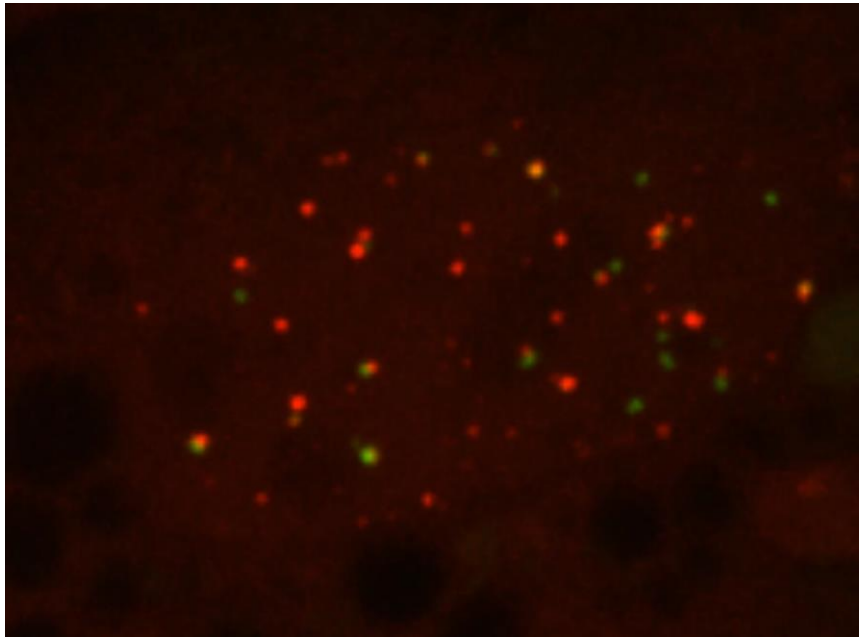
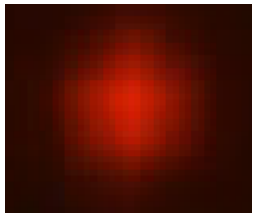
Automated Image Processing

October 9, 2023

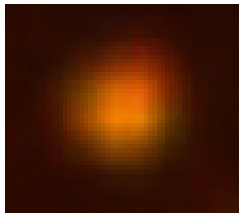
Data Situation



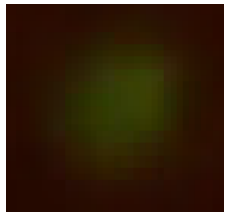
Color	Represents	Full Name
Red Puncta	Telomere	
Green Puncta	PML	ALT telomere-associated promyelocytic leukemia
Yellow Puncta	APB(telomere +PML)	ALT-associated promyelocytic leukemia nuclear bodies



Red Puncta



Yellow Puncta



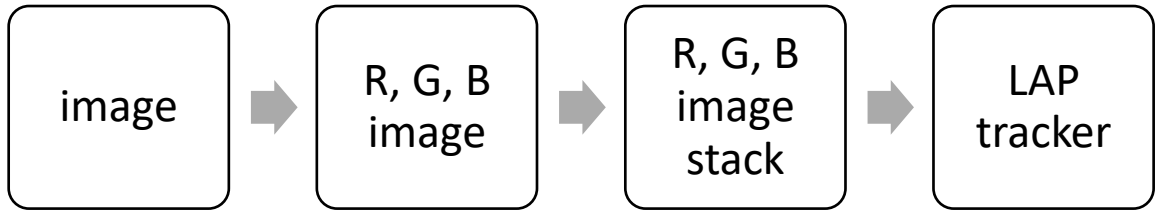
Green Puncta

Reference Papers for background on the biology behind the experiment

1. Nuclear body phase separation drives telomere clustering in ALT cancer cells
2. An Image Analysis Pipeline for Quantifying the Features of Fluorescently-Labeled Biomolecular Condensates in Cells

Goal: Detect the following fusion using tracking puncta analysis

Color	Represents
Red+Red	Telomere+Telomere
Yellow+Yellow	APB+APB
Red+Yellow	Telomere+APB



Trackmate-7

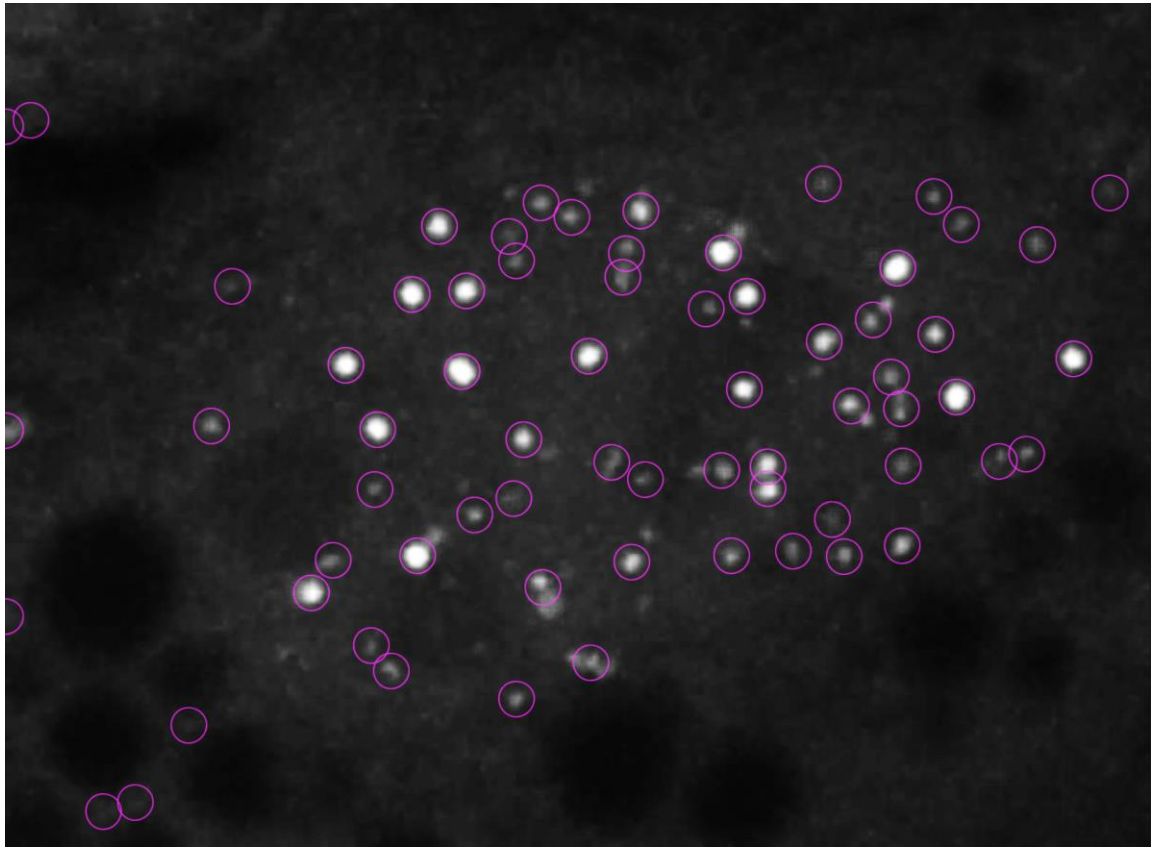
Reference Papers for background on the Tracking analysis

1. Robust single-particle tracking in live-cell time-lapse sequences
2. Segmentation and quantification of subcellular structures in fluorescence microscopy images using Squassh
3. Time-lapse confocal imaging datasets to assess structural and dynamic properties of subcellular nanostructures
4. Recent advances in optical microscopic methods for single-particle tracking in biological samples
5. TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines

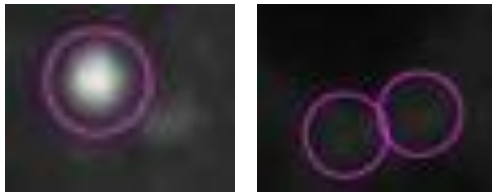
Trackmate 7 basic function

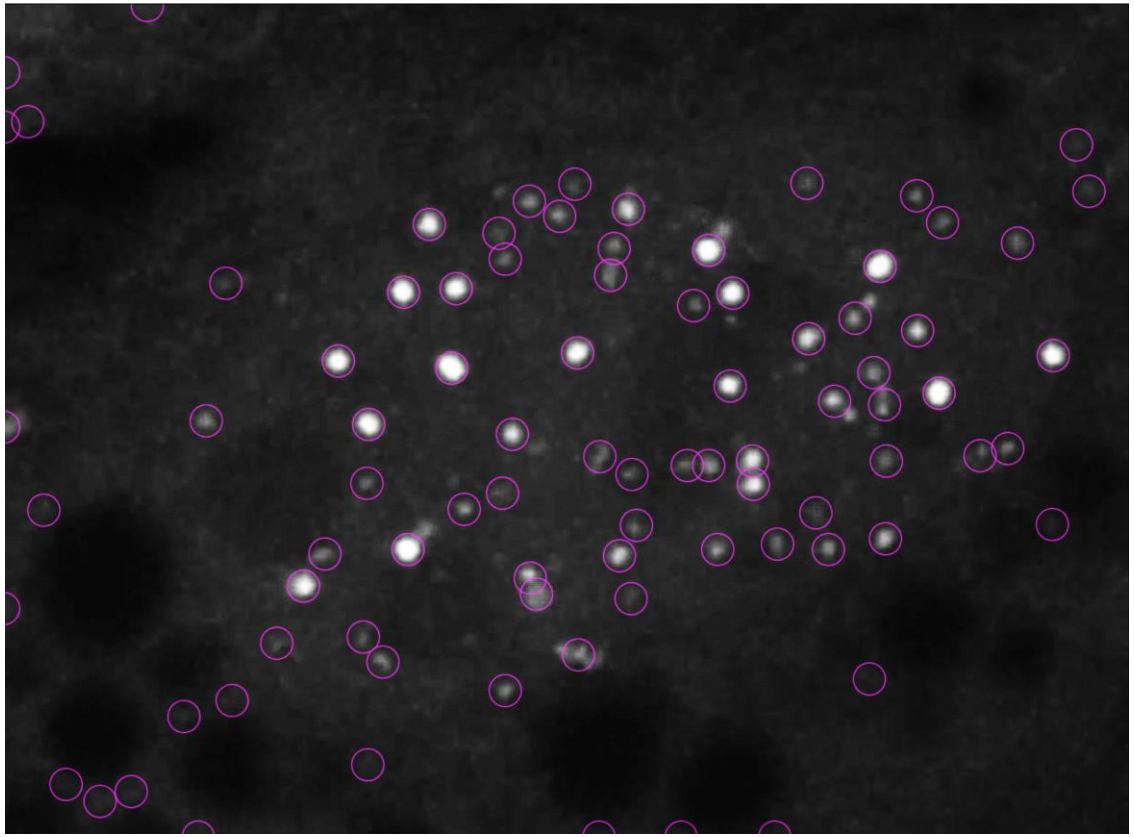
Detector		
DoG detector	Approximation of the LoG operator by differences of Gaussian (DoG)	Spots found too close are suppressed. This detector can do sub-pixel localization of spots using a quadratic fitting scheme.
Hessian detector	Compute the determinant of the hessian matrix	It can be configured with a different spots size in XY and Z. It can also return a normalized quality value, scaled from 0 to 1 for the spots of each time-point.
Label image detector	This detector creates spots by importing regions from a label image.	A label image is an image where the pixel values are integers. Each object in a label image is represented by a single common pixel value (the label) that is unique to the object.
LoG detector	Laplacian of Gaussian	The maxima in the filtered image are searched for, and maxima too close from each other are suppressed. A quadratic fitting scheme allows to do sub-pixel localization.
Manual annotation	Manual segmentation	
Mask detector	creates spots from a black and white mask	all the pixels in the designated channel that have a value strictly larger than 0 are considered as part of the foreground, and used to build connected regions.

LAP tracker



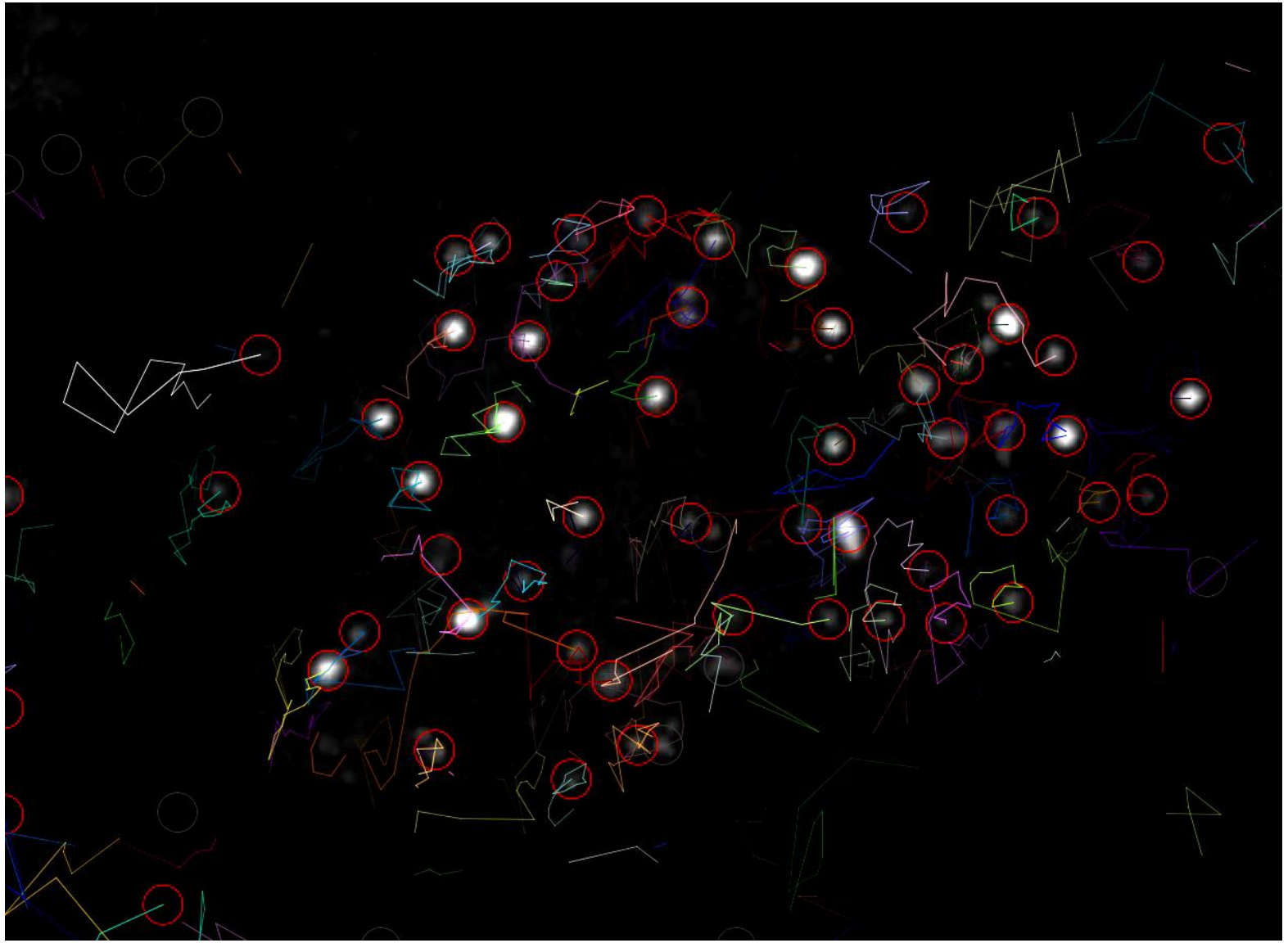
- Estimated object diameter : 39 pixel
- Quality threshold : 0.08
- 67 puncta - 6 not a puncta = 61





- Estimated object diameter : 36 pixel
- Quality threshold : 0.08
- 86 puncta – 14 not a puncta = 72

LAP tracker



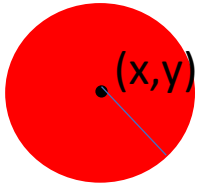
- Initial thresholding - not applied
- Segment gap closing - 20 pixel
- Max frame gap - 2 frame
- Each example for R,G, B channel
- Basic function- Merging detection exist but fails to detect our puncta fusion

Impletmenting merging function Using the result obtained from the LAP tracker in TrackMate

1. def readLAPfile: Obtain x, y coordinate from LAP tracker

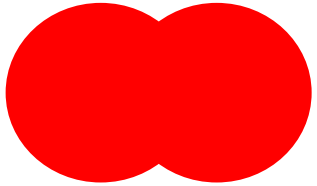
```
  </FeatureDeclarations>
▼<AllSpots nspots="2411">
  ▼<SpotsInFrame frame="0">
      <Spot ID="23561" name="ID23561" VISIBILITY="1" RADIUS="19.5" QUALITY="0.08574102073907852" POSITION_T="0.0" POSITION_X="28.232280561963265" POSITION_Y="129.48178326094316" FRAME="0
      <Spot ID="23562" name="ID23562" VISIBILITY="1" RADIUS="19.5" QUALITY="0.08515263348817825" POSITION_T="0.0" POSITION_X="0.0" POSITION_Y="130.0" FRAME="0" POSITION_Z="0.0"/>
      <Spot ID="23563" name="ID23563" VISIBILITY="1" RADIUS="19.5" QUALITY="0.10878407955169678" POSITION_T="0.0" POSITION_X="900.5869813824542" POSITION_Y="199.4815610967736" FRAME="0"
      <Spot ID="23564" name="ID23564" VISIBILITY="1" RADIUS="19.5" QUALITY="0.08986377716064453" POSITION_T="0.0" POSITION_X="1216.9184795089582" POSITION_Y="208.64993413783785" FRAME="0
      <Spot ID="23565" name="ID23565" VISIBILITY="1" RADIUS="19.5" QUALITY="0.11995470523834229" POSITION_T="0.0" POSITION_X="1023.0424248780397" POSITION_Y="213.05221874892715" FRAME="0
```

2. def drawPuncta: For each frame, define area of a circle for each x, y coordinate from LAP tracker based on the object diameter(ex 39 pixel)



Object diameter(pixel)

3. def detectOverlap: Detect all the circle that overlaps



Label 1, Label 2

3. def trackPath : Function keep track of the puncta that have been detected in detectOverlap to see whether they stop being tracked, whether their tracking path continues. (Currently working on this process)

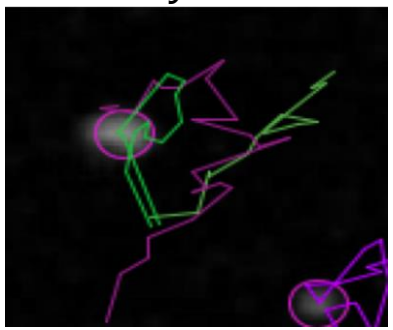
4. def countMerge : Count the number of events where one track disappears, and the other puncta track continues, or where their puncta coordinates coincide over a period of time.

Using mergeDetect function, we can detect

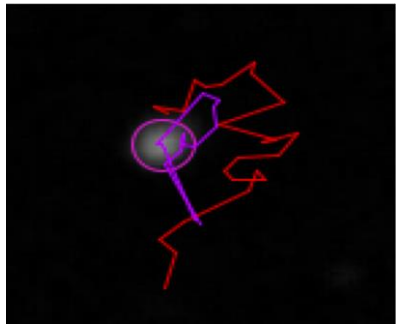
Fusion of Red + Red puncta→ But what about yellow puncta?

Solution 1 : Change RGB image to HSV(Hue, Saturation, Value-library for this function exist)
lower_yellow = np.array([30, 100, 100])
upper_yellow = np.array([60, 255, 255])
yellow_mask = cv2.inRange(hsv_image, lower_yellow, upper_yellow) – make a yellow mask
yellow_result = cv2.bitwise_and(image, image, mask=yellow_mask) - extract yellow part from image
Use the result in trackmate LAP tracker -> apply the result to Merge detect function

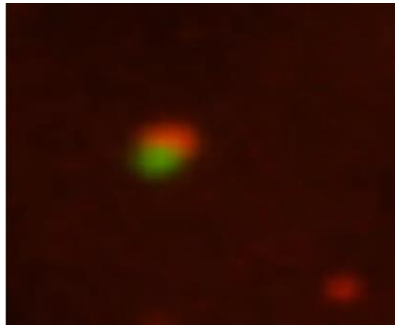
Solution 2: Draw RGB map based on their intensity. Find corresponding RGB of each pixel in the image and check whether the combination of RGB is in the yellow boundary. Have to check whether we can extract intensity



Red Channel
Time 28

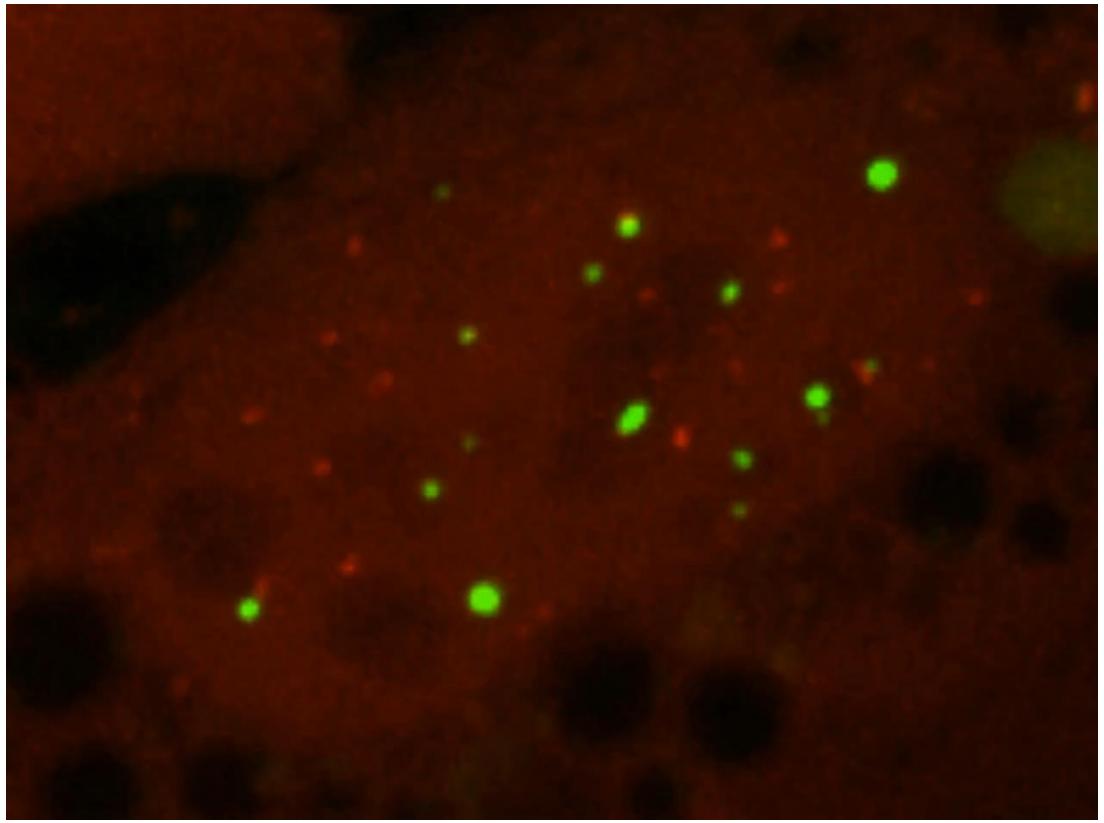


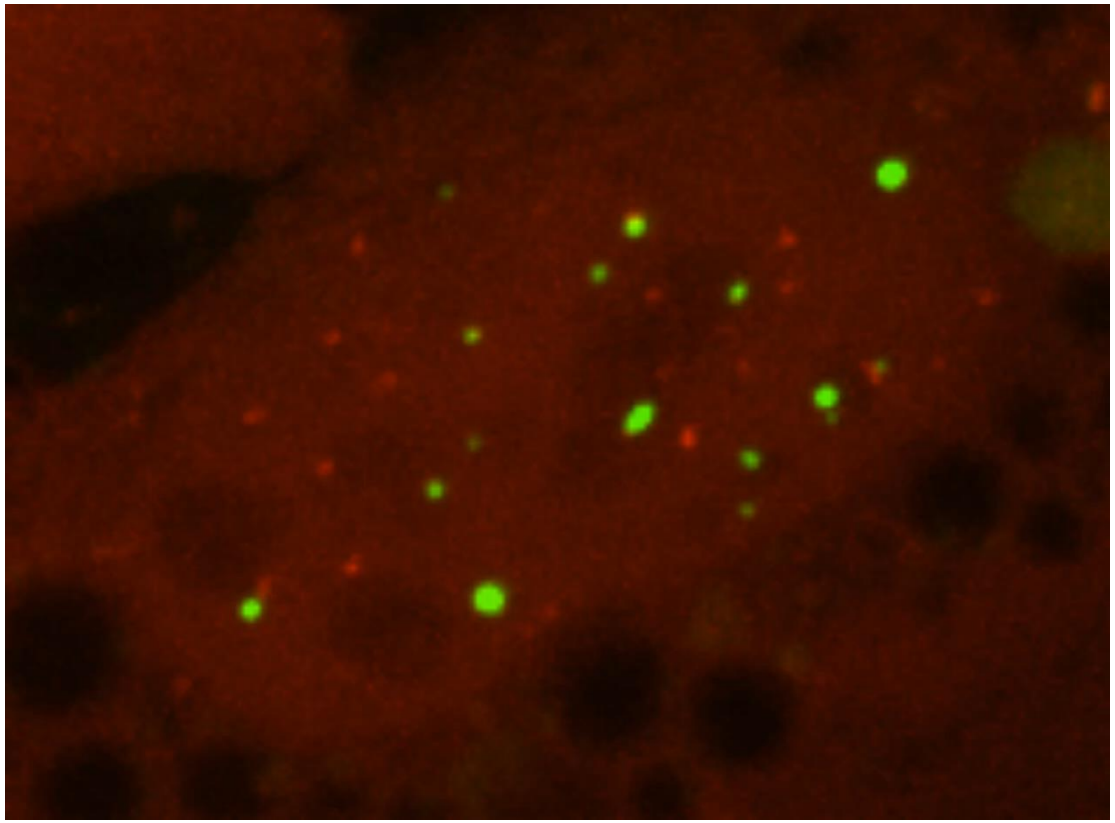
Green Channel
Time 28



Original Image
Time 28

Preprocessing Images

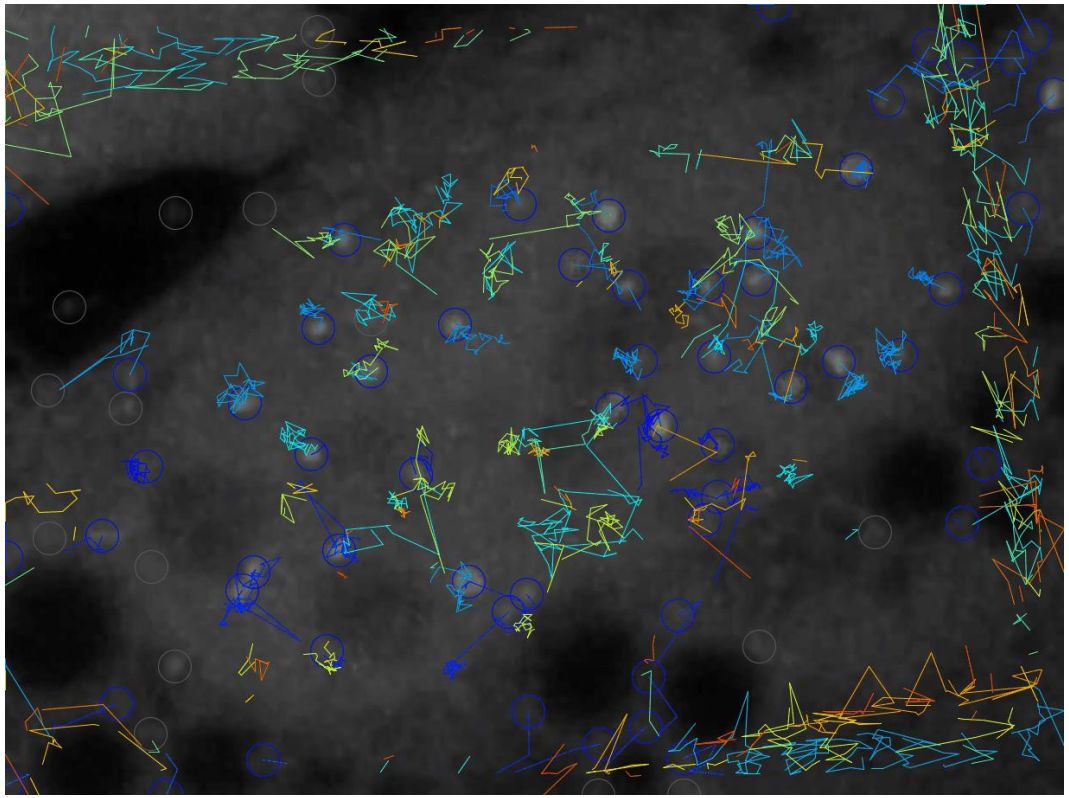




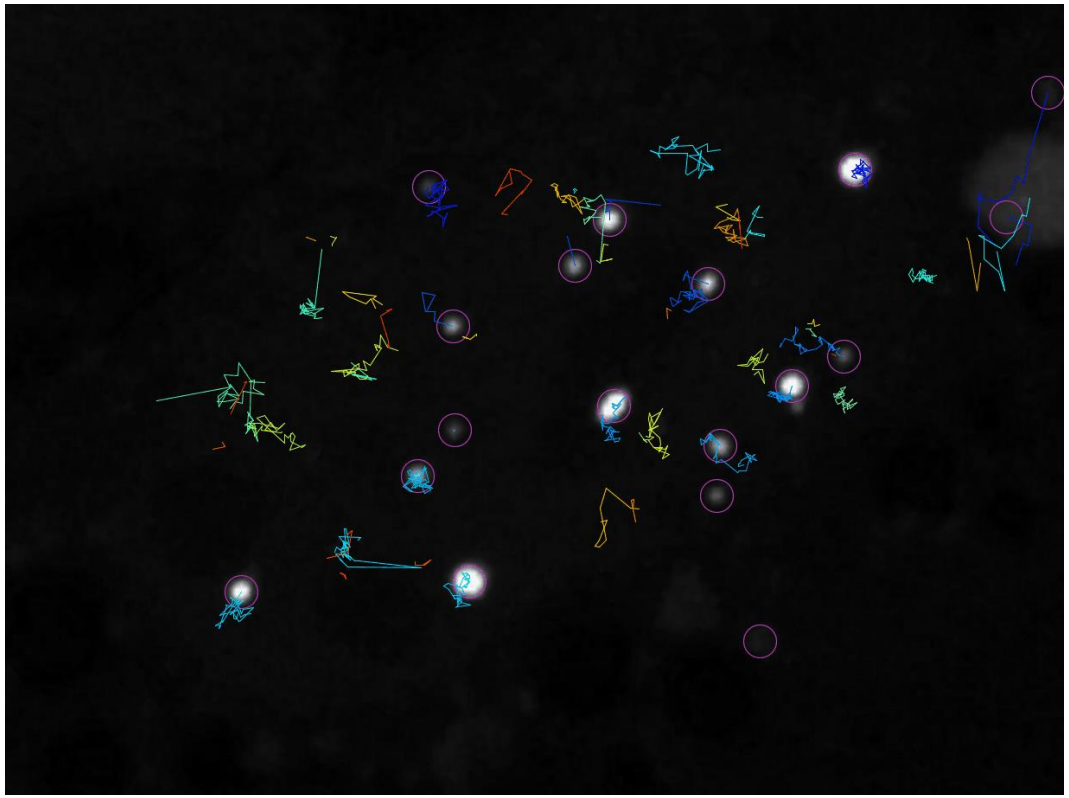
Rigid transformation to a "canonical" form -> Choose the first image but should choose the one for which the average of the punca coordinates is almost as the center of the center of the image

Trackmate using transformed video

Divided it into 38 frames and into R, G, B channel

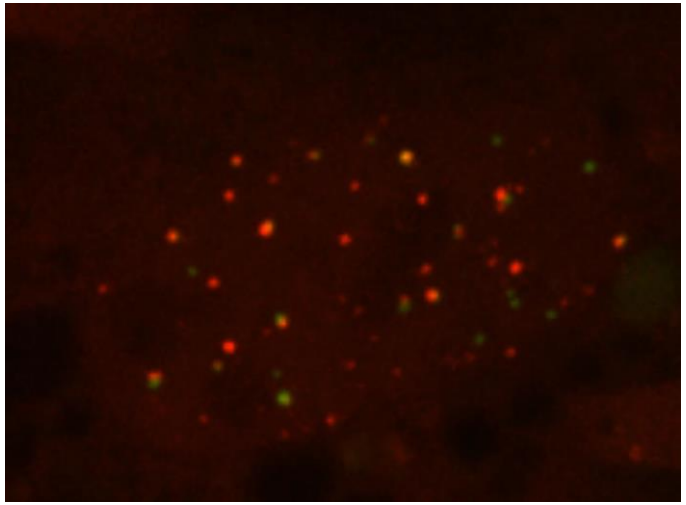


Red Channel

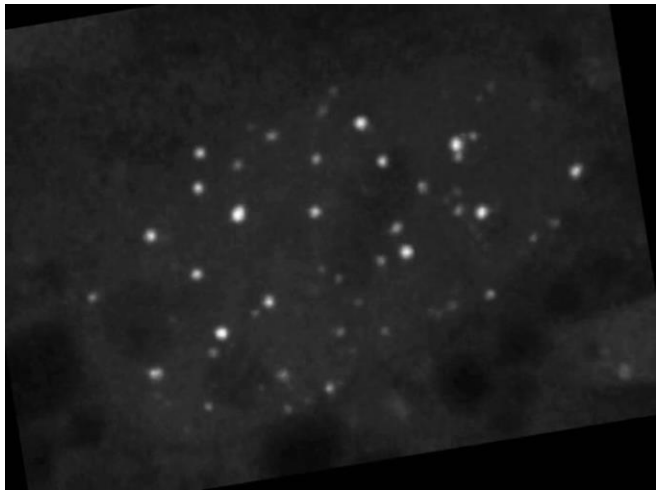


Green Channel

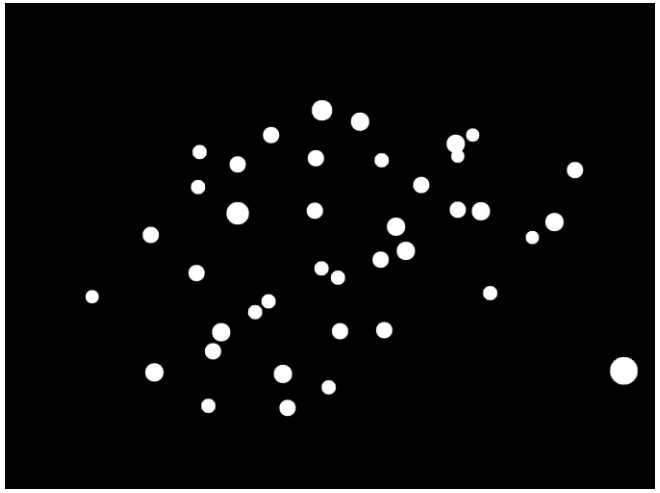
Preprocessing images



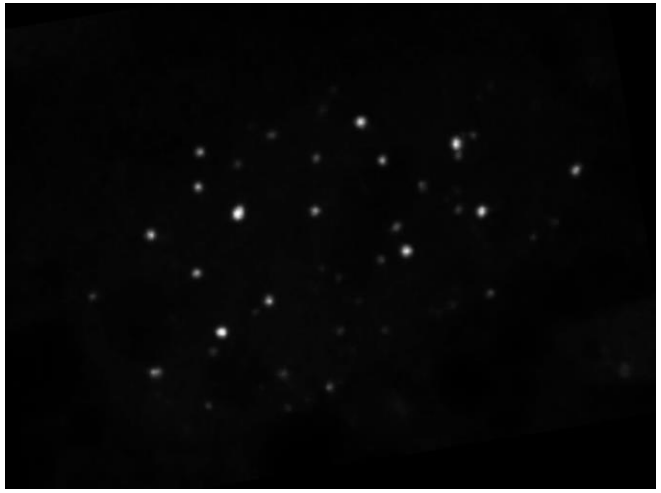
Original Image



Red channel Image



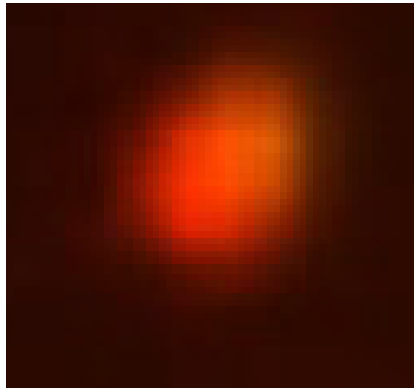
Blob masking(0,1)



Gamma2

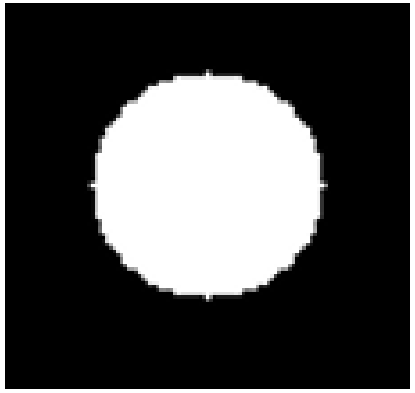
Masked

Preprocessing images to filter noise



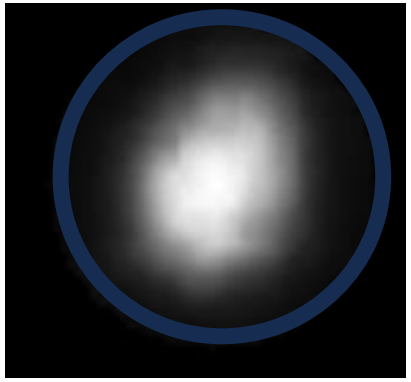
Original Image

Red channel Image



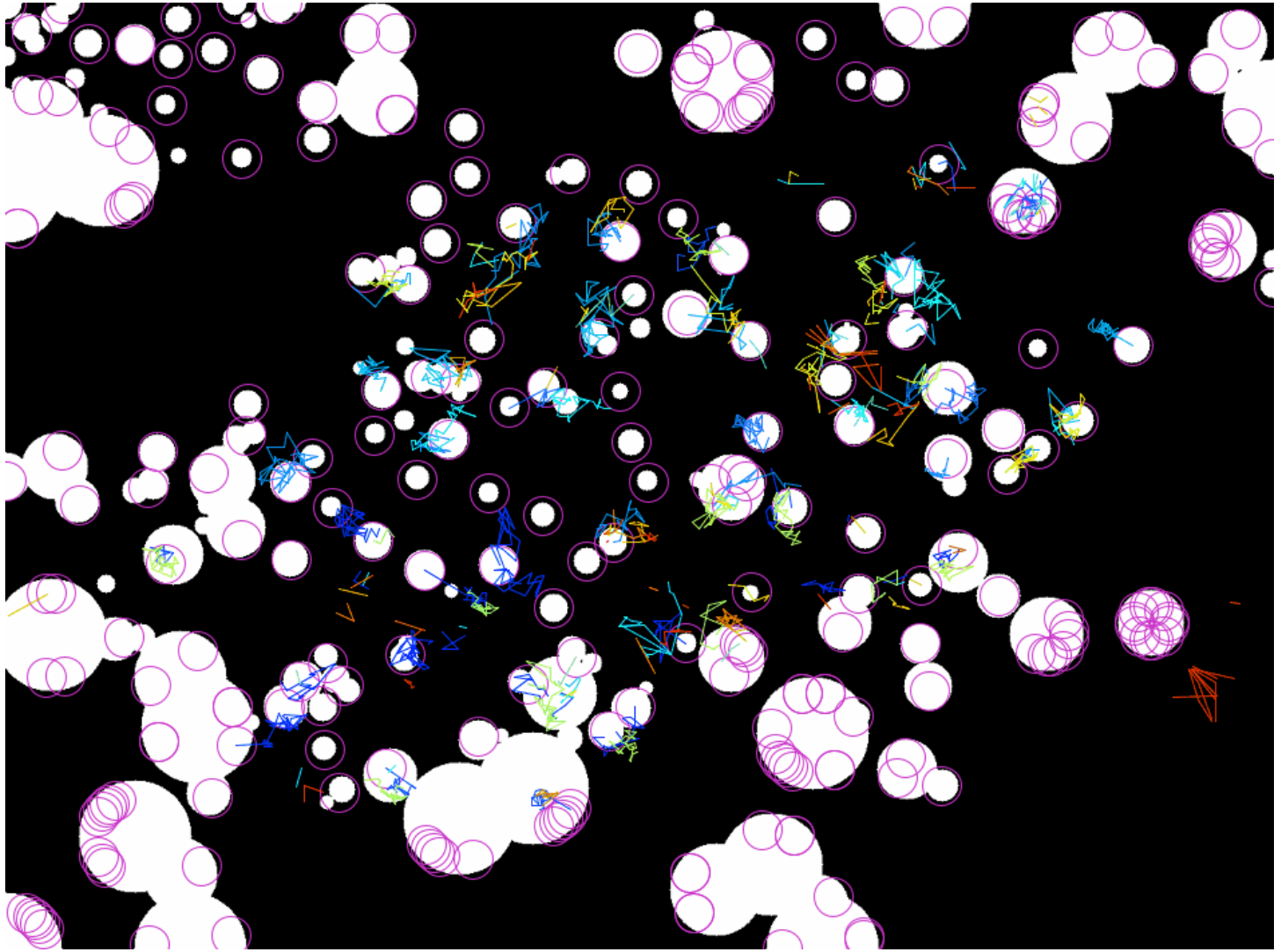
Blob masking

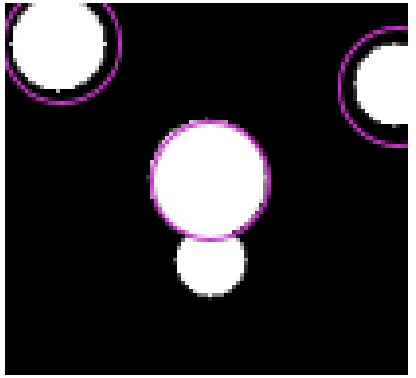
Gamma2



Masked

Blob Mask



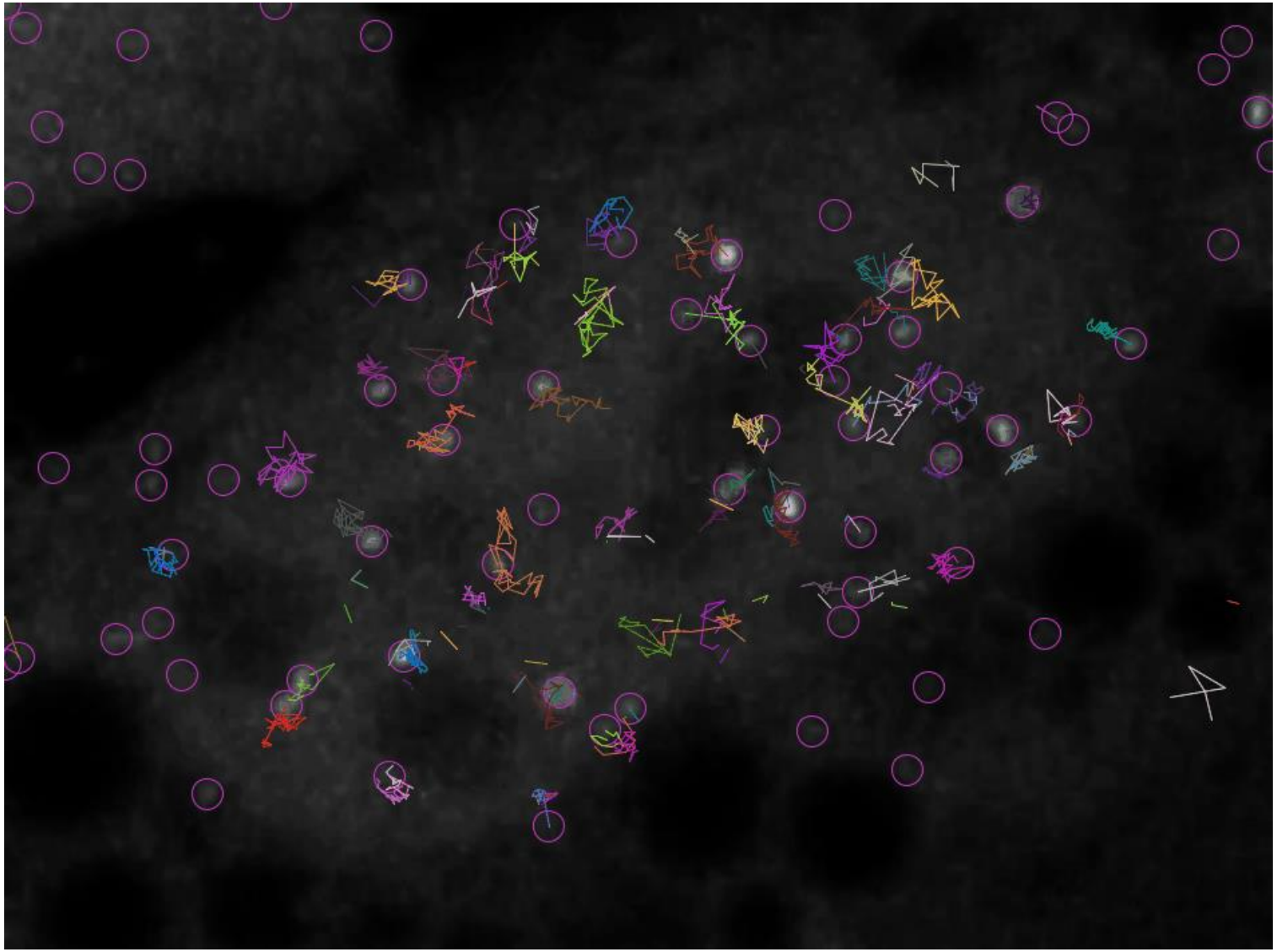


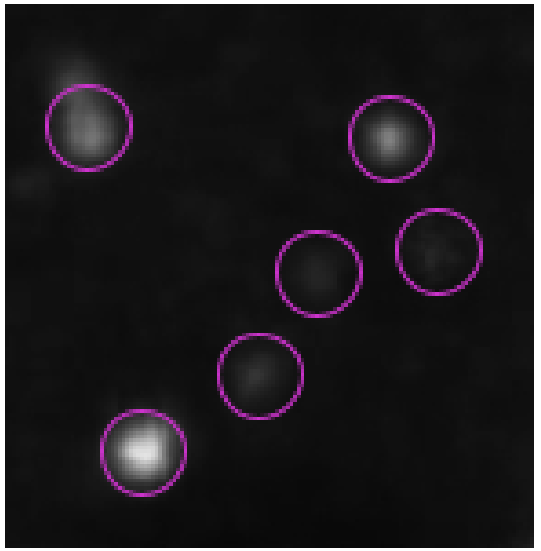
40 pixel

38 pixel

Size different cause fail in detection

Gamma2



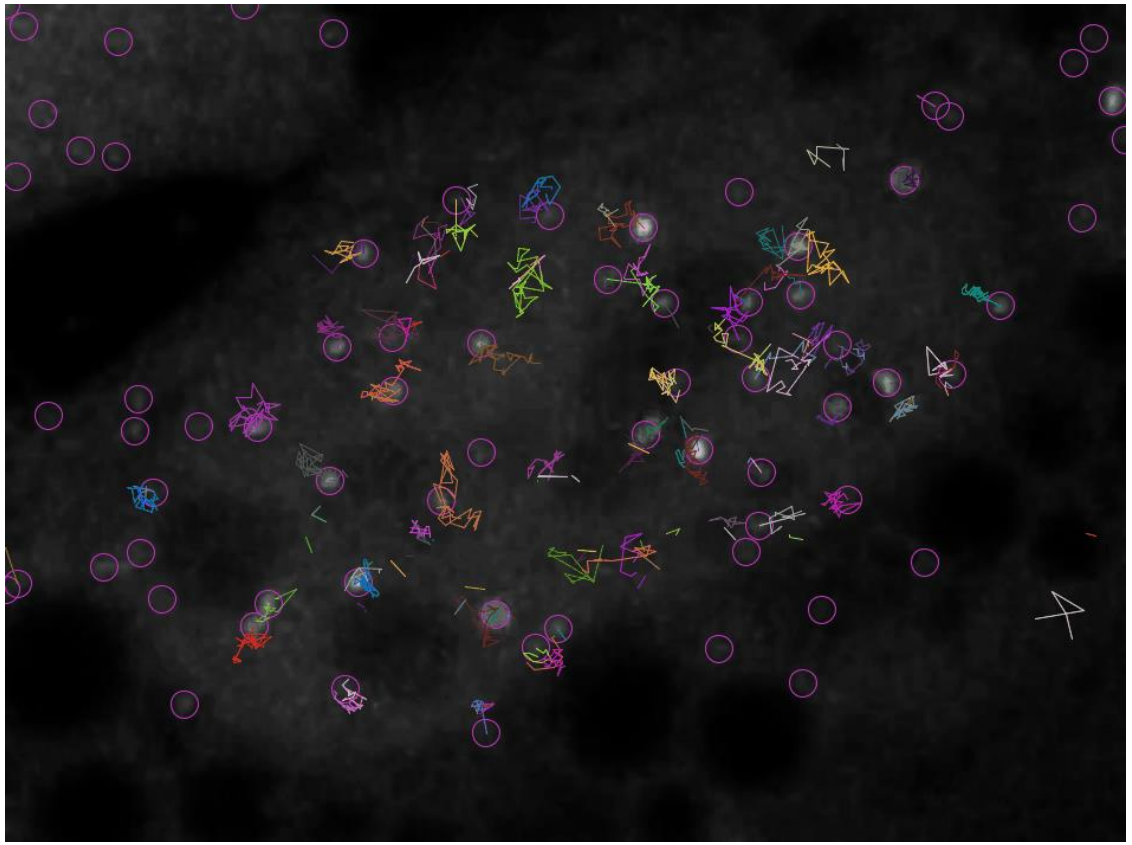


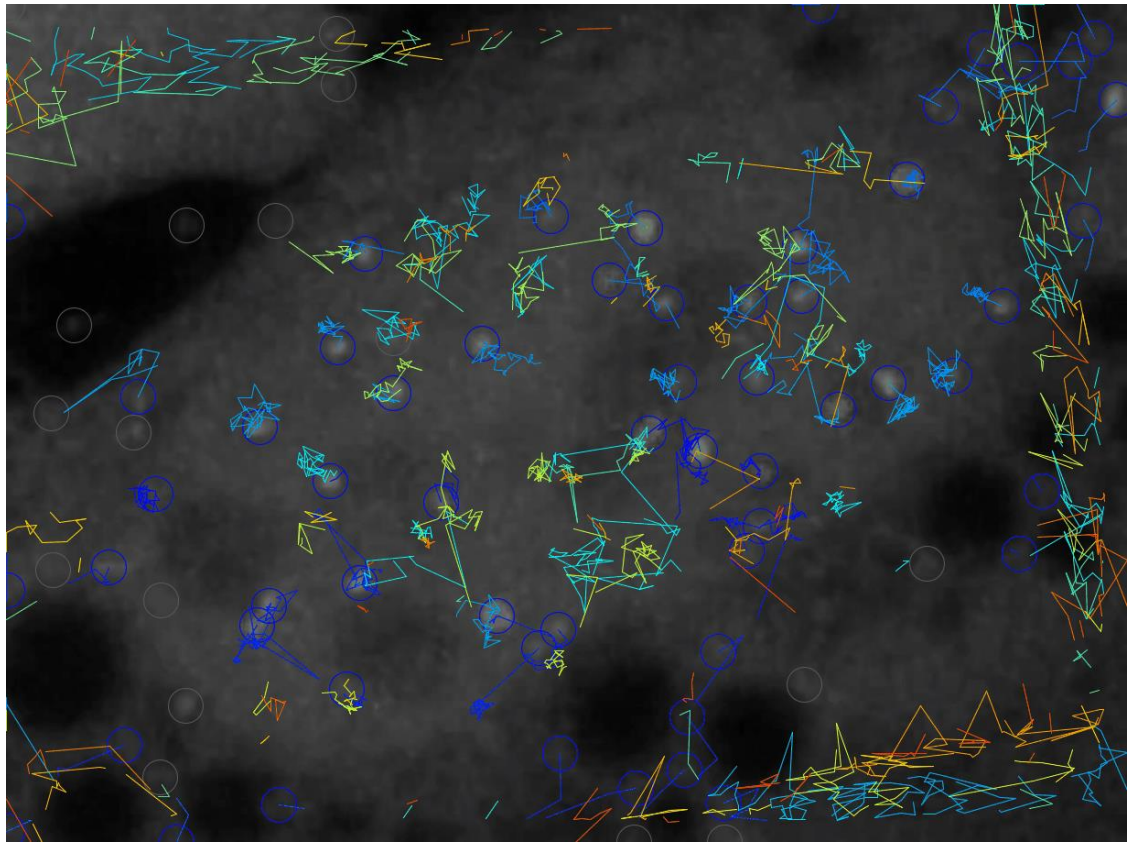
31 pixel

- Even with lower estimated object diameter, not that many noise

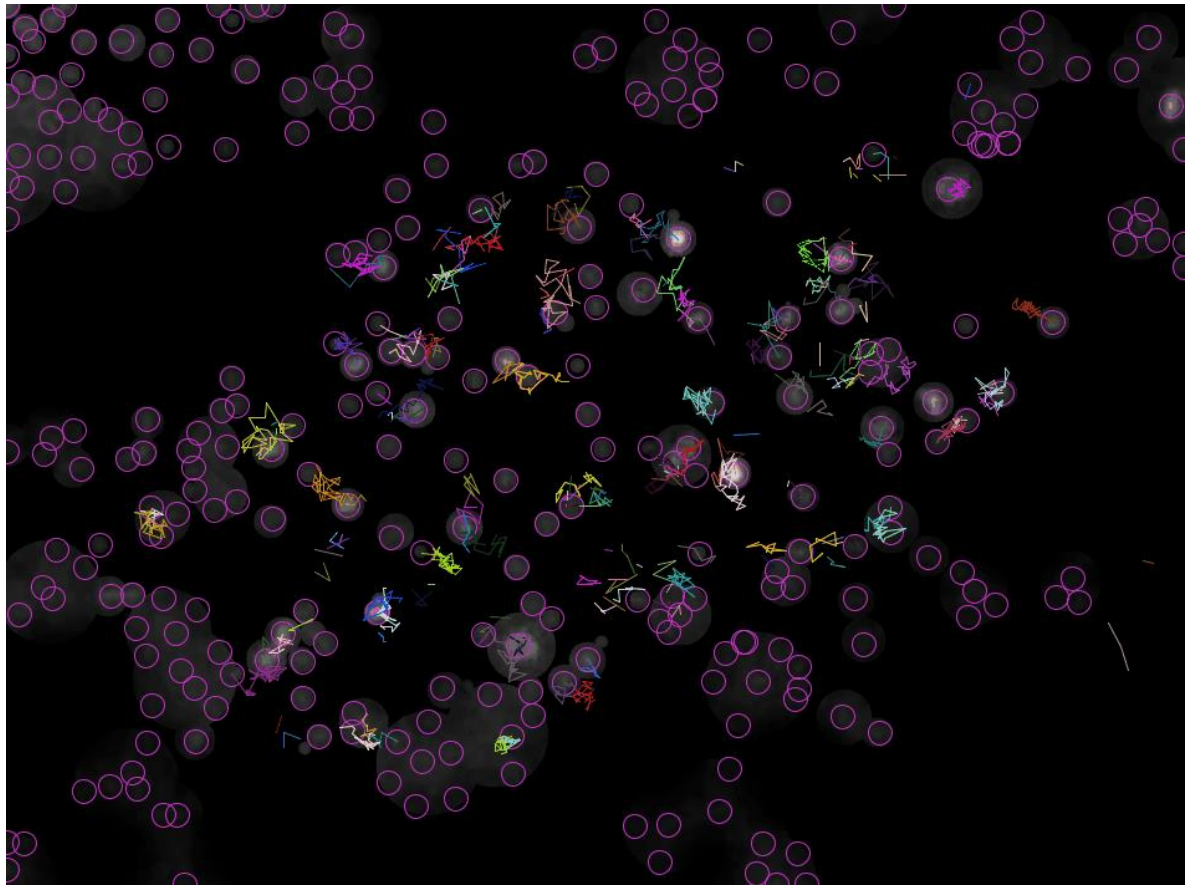
- Solved the edge problem of the original image

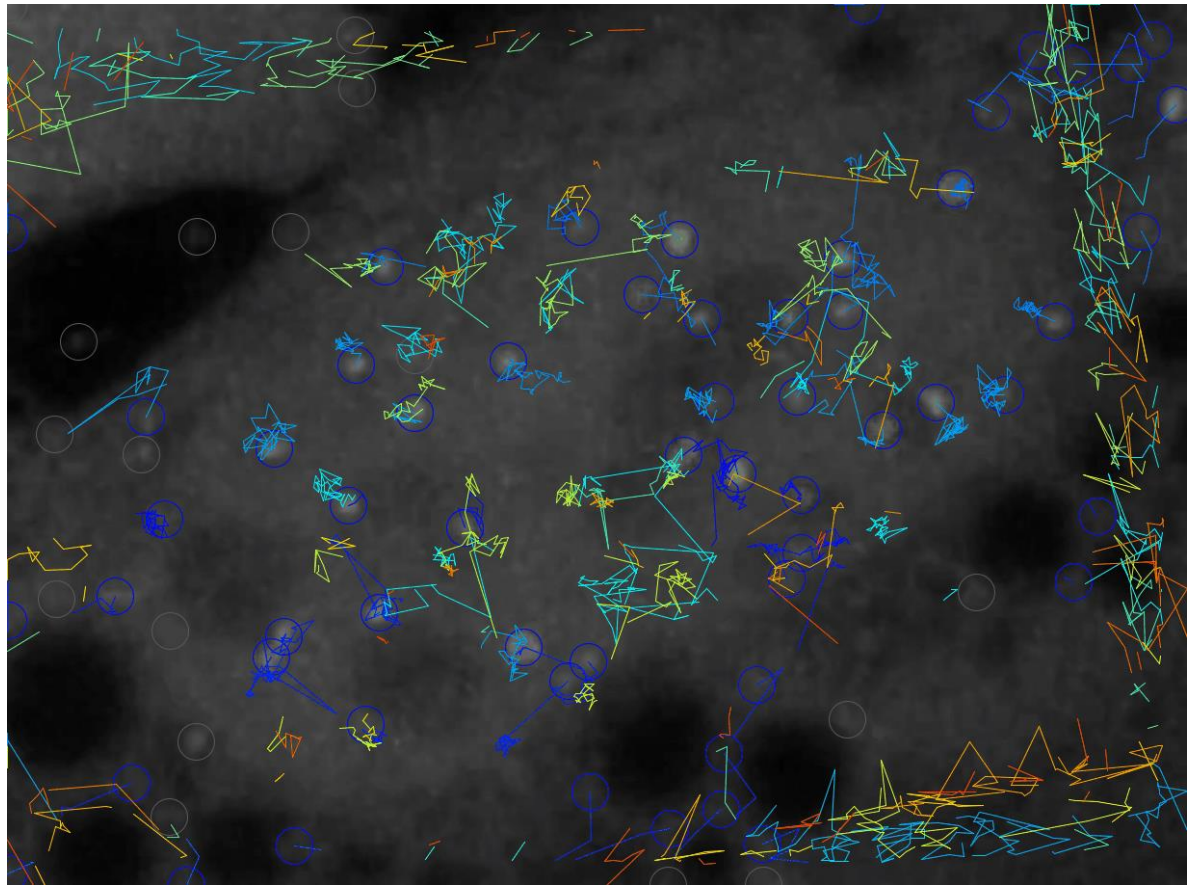
Gamma2





Masked



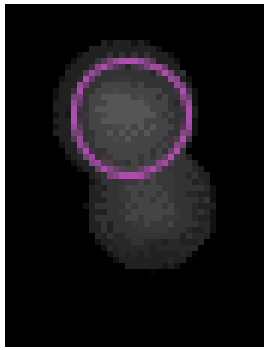
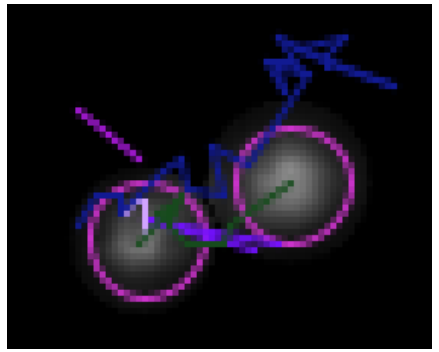
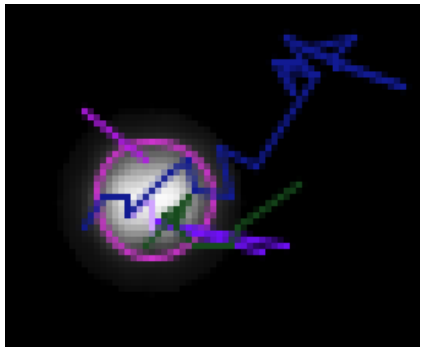


Trackmate merge function

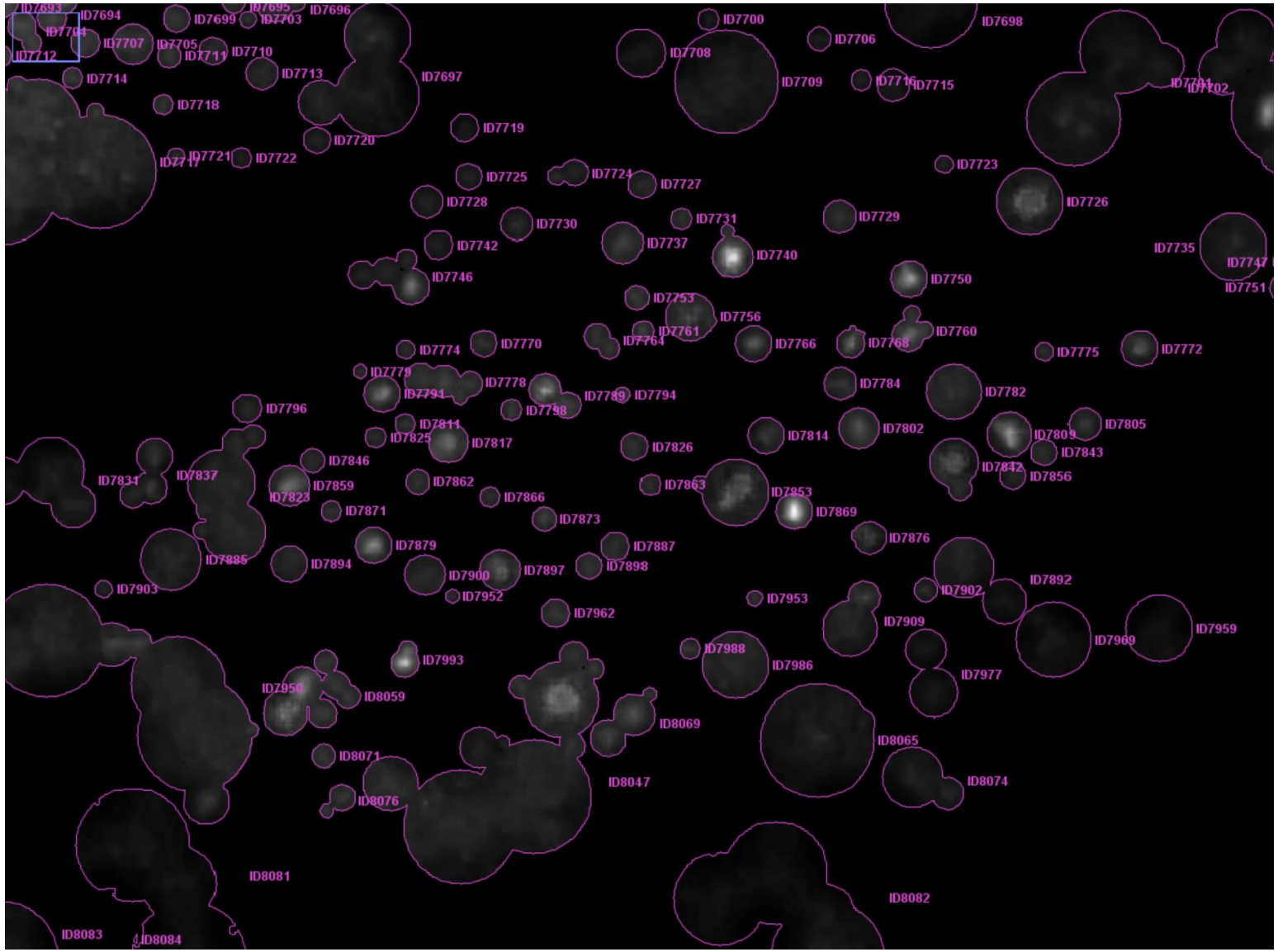
Bring a track → Detect labeled puncta neighbor of each frame → Compare prior and preceding frame → Method

- Detect it as merge if # increase
- Detect is as split if # decrease

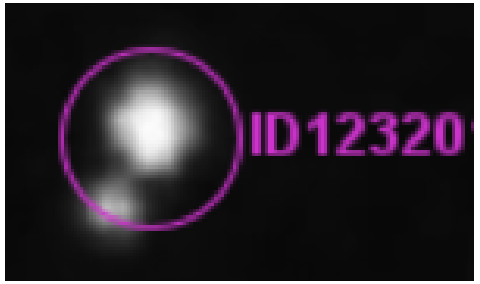
➔ Fail to detect puncta merge that were not originally in same trackpath, trackpath overlaps over certain frames.
➔ Trackmate fails to detect merging process

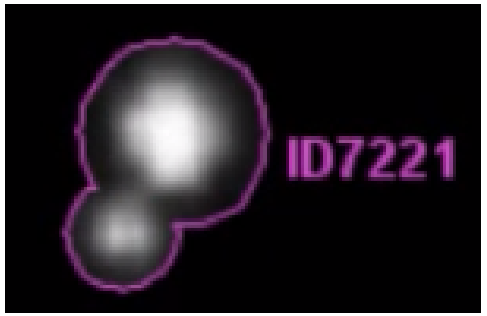
Puncta Merge Detection



1. Used a mask that can detect blob merging shape
2. Detect only mask in the image



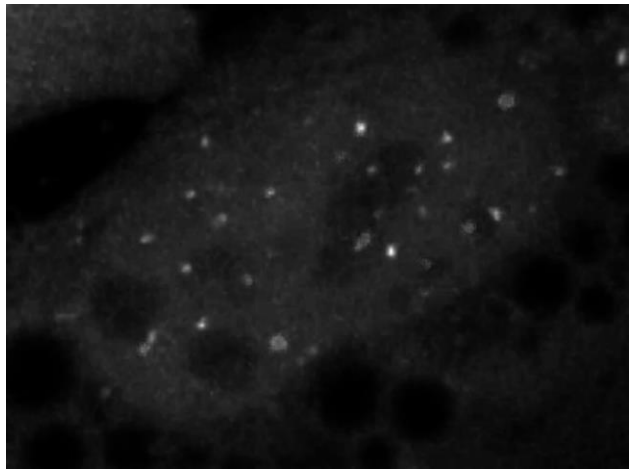
Before Mask Detection



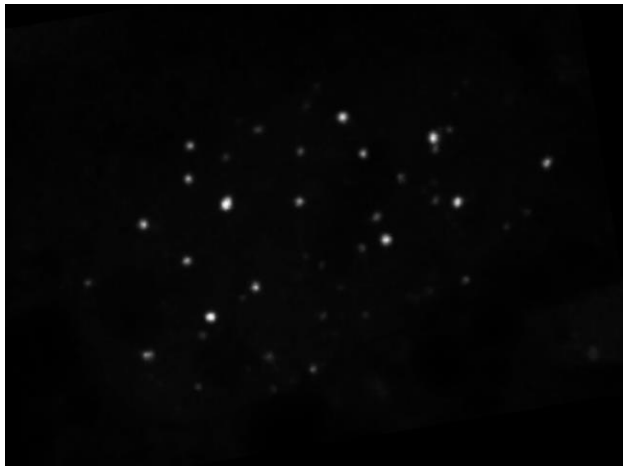
After Mask Detection

Frame_0 issue

Before



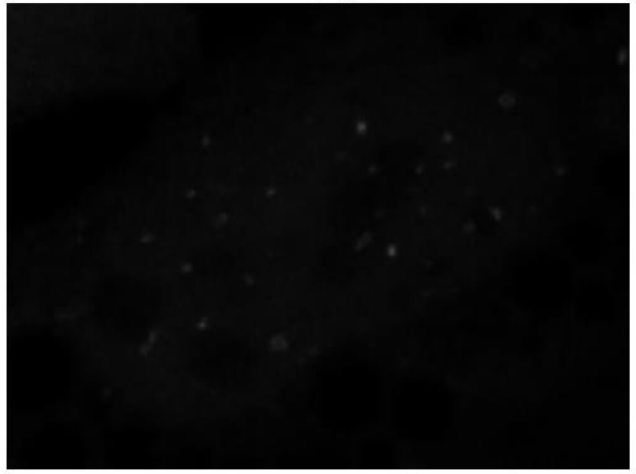
Frame_0



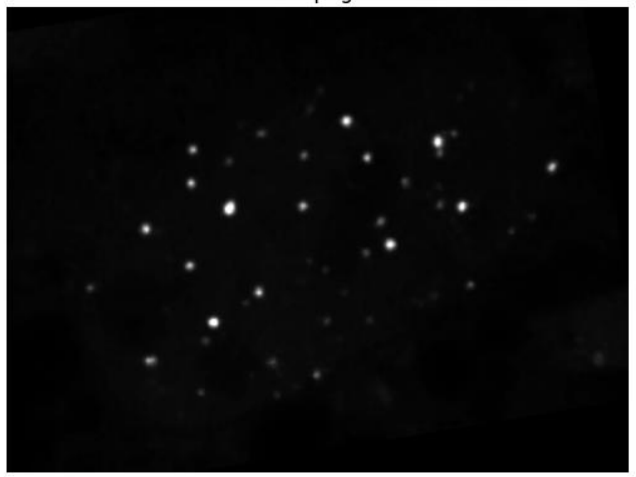
Frame_38

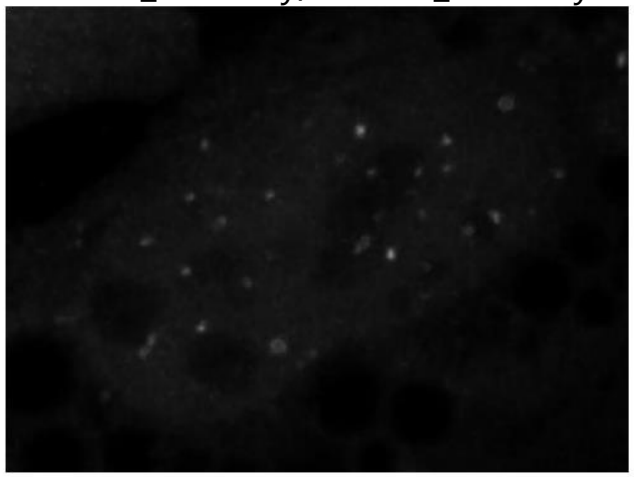
Applied enhancement factor:
medium_intensity/current_intensity

00.png



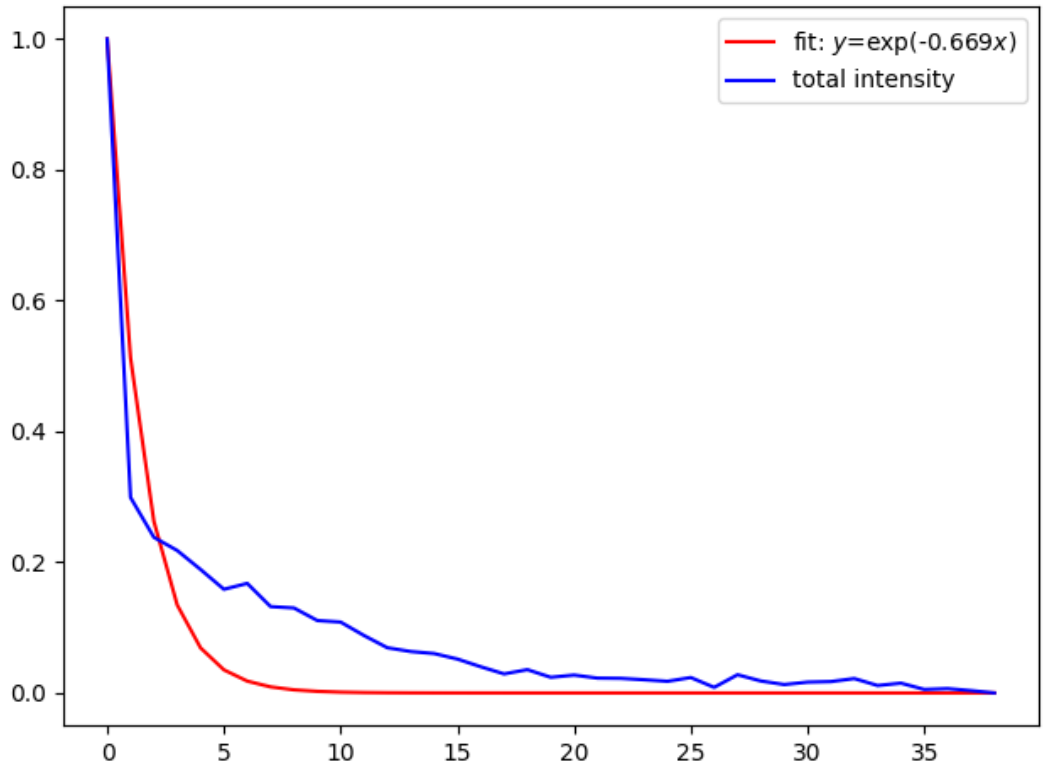
38.png

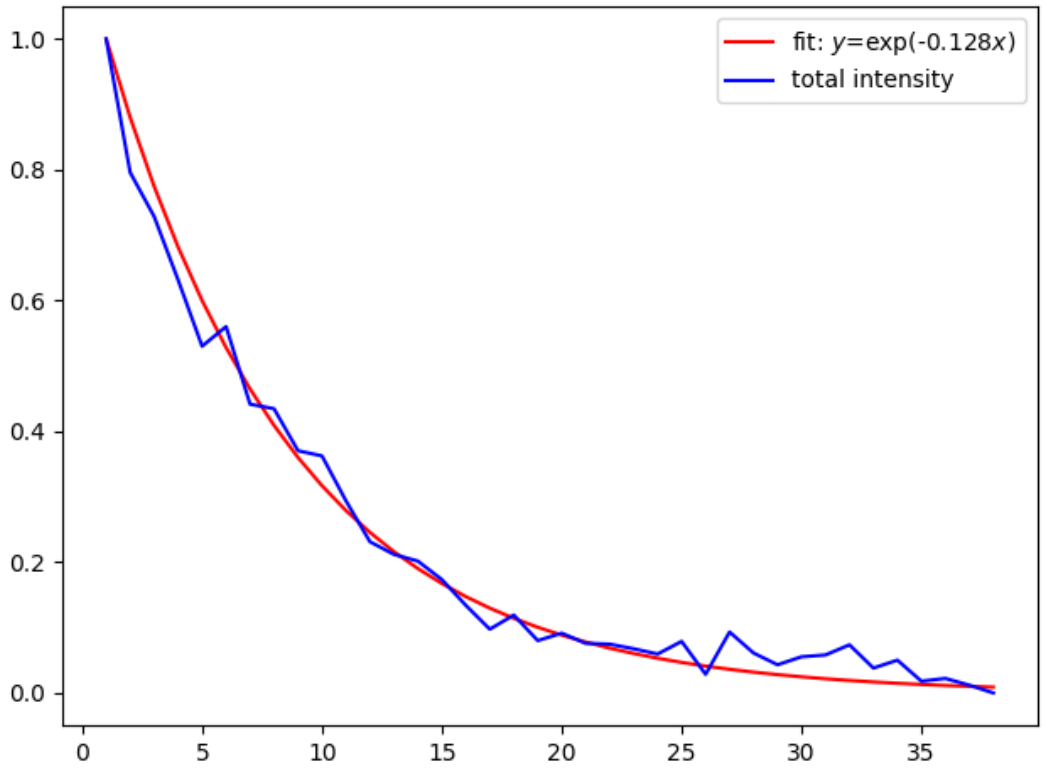




x2

Frame_0 issue





Result obtained from the LAP tracker in TrackMate

	A	B	C	D	E	F	G	H	I
	LABEL	ID	TRACK_ID	QUALITY	POSITION_	POSITION_	POSITION_	POSITION_	FRAME
9	ID63634	63634	7	3.707723618	893	323	0	0	0
0	ID63572	63572	7	6.187105179	893	268	0	0	0
1	ID61622	61622	7	14.86606598	928	298	0	1	1
2	ID61680	61680	7	7.88174057	914	355	0	1	1
3	ID61694	61694	7	9.20205307	936	366	0	1	1
4	ID64650	64650	7	2.094964981	938	306	0	2	2
5	ID64684	64684	7	8.42609024	915	337	0	2	2
6	ID64691	64691	7	11.57460022	940	346	0	2	2
7	ID59662	59662	7	10.12209702	920	327	0	3	3
8	ID59698	59698	7	2.739322662	954	361	0	3	3
9	ID60625	60625	7	14.71574783	925	312	0	4	4
0	ID60636	60636	7	13.37451935	962	325	0	4	4
1	ID58670	58670	7	12.88766479	940	310	0	5	5
2	ID62608	62608	7	11.31658936	963	300	0	6	6

➔ Split event

➔ Merge event

Trackmate merge detection + Merge function result
Prior function needs modification due to one-to-one matching of each puncta in a frame.

➔ Using Simple LAP tracker, we can obtain individual track without merging or splitting event

Things to consider

Simple LAP tracker : Track splitting and merging that allowed, resulting in having non-branching tracks

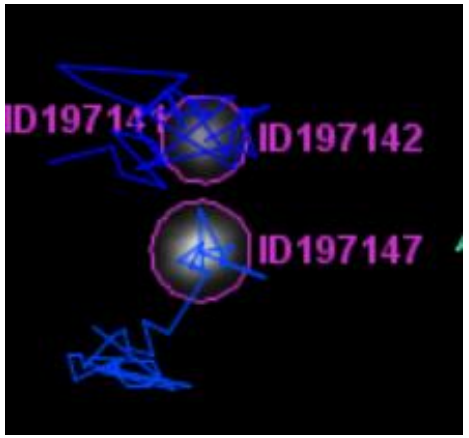
1. Misdetection : Threshold using medium radius - filter

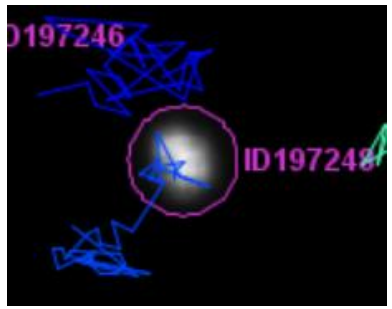
0.60164

0.93894

2. Circularity : $\frac{4\pi A}{P^2}$

A: Area, P: Boundary - check merge





3. Track that ends – check merge

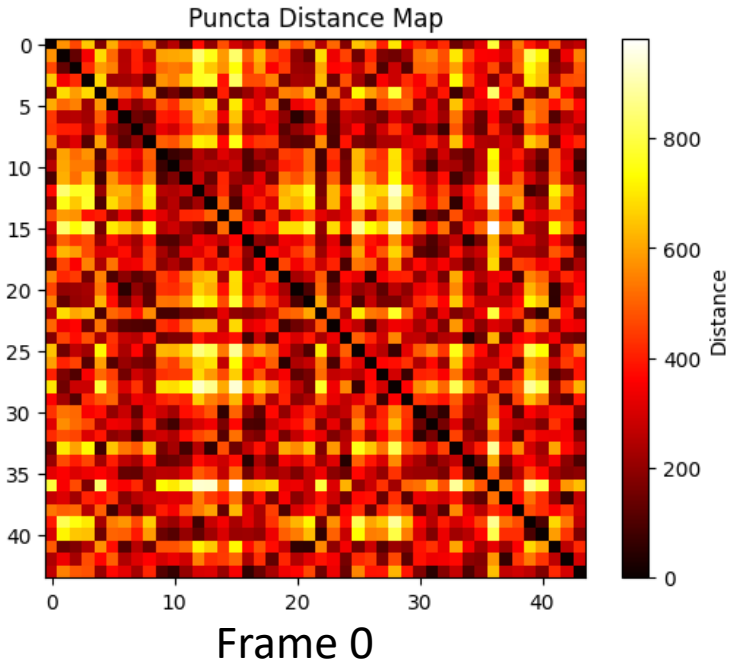
Two cases of merge

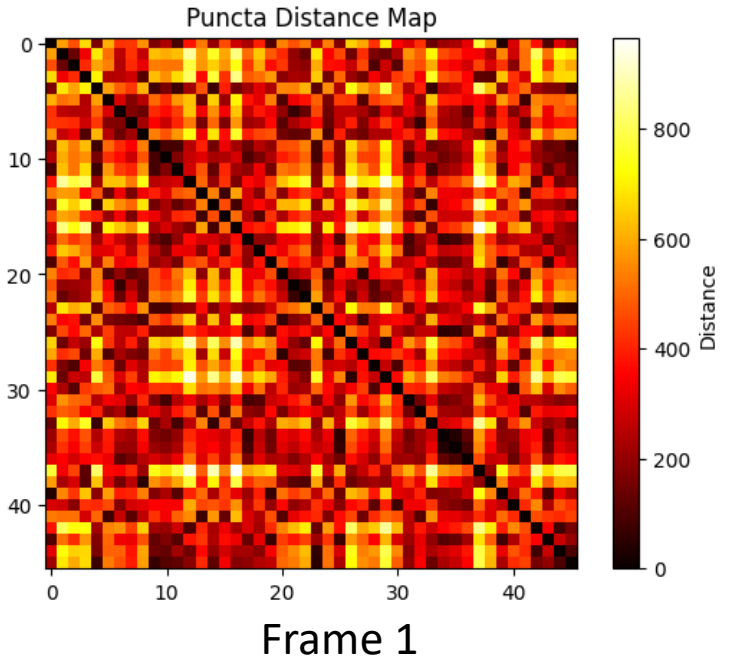
Impletmenting merging function Using the result obtained from the LAP tracker in TrackMate

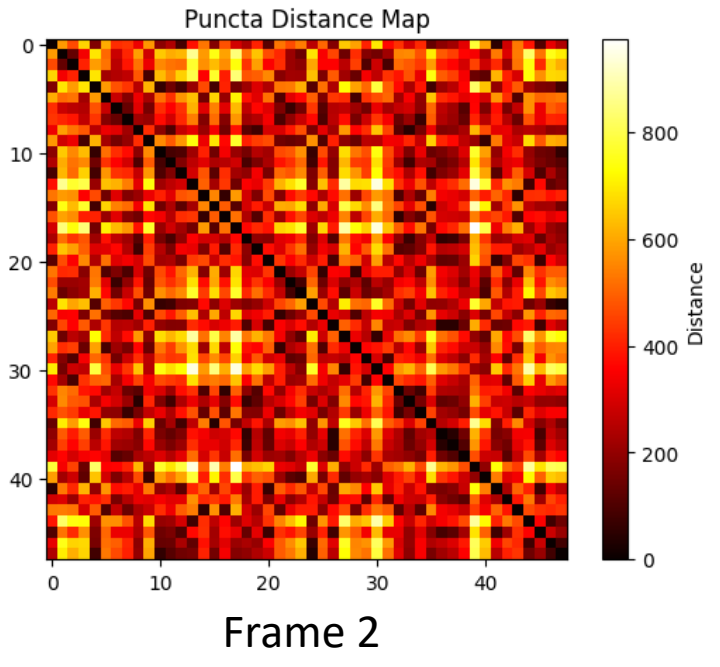
1. def readLAPfile: Obtain x, y coordinate from LAP tracker

```
  </FeatureDeclarations>
▼<AllSpots nspots="2411">
  ▼<SpotsInFrame frame="0">
      <Spot ID="23561" name="ID23561" VISIBILITY="1" RADIUS="19.5" QUALITY="0.08574102073907852" POSITION_T="0.0" POSITION_X="28.232280561963265" POSITION_Y="129.48178326094316" FRAME="0
      <Spot ID="23562" name="ID23562" VISIBILITY="1" RADIUS="19.5" QUALITY="0.08515263348817825" POSITION_T="0.0" POSITION_X="0.0" POSITION_Y="130.0" FRAME="0" POSITION_Z="0.0"/>
      <Spot ID="23563" name="ID23563" VISIBILITY="1" RADIUS="19.5" QUALITY="0.10878407955169678" POSITION_T="0.0" POSITION_X="900.5869813824542" POSITION_Y="199.4815610967736" FRAME="0"
      <Spot ID="23564" name="ID23564" VISIBILITY="1" RADIUS="19.5" QUALITY="0.08986377716064453" POSITION_T="0.0" POSITION_X="1216.9184795089582" POSITION_Y="208.64993413783785" FRAME="0
      <Spot ID="23565" name="ID23565" VISIBILITY="1" RADIUS="19.5" QUALITY="0.11995470523834229" POSITION_T="0.0" POSITION_X="1023.0424248780397" POSITION_Y="213.05221874892715" FRAME="0
```

2. def distancePuncta: For each frame, define the distance from puncta – distance map for 38 frames

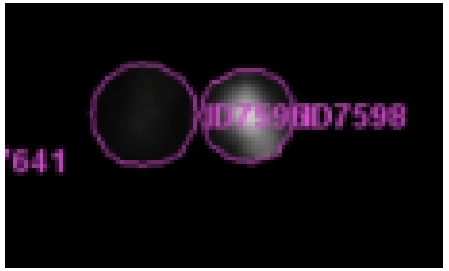




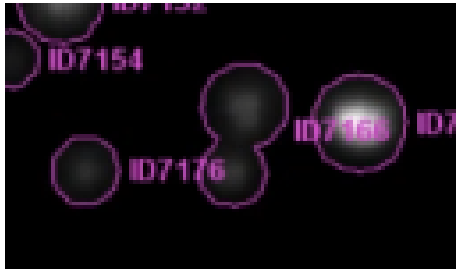


3. Find pairs of puncta with distance value lower than diameter1+diameter2 for more than three frames

Puncta Merge Detection



Frame 10

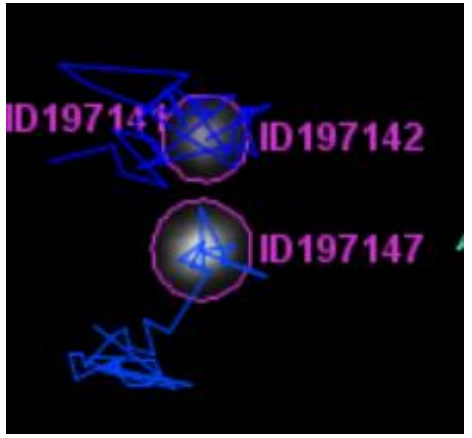


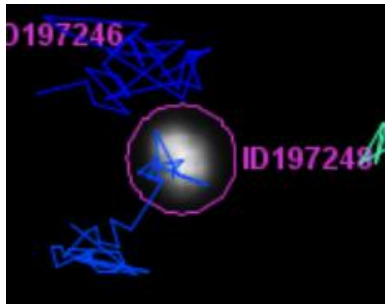
Frame 11

Assumption: In order for merging event to occur, existing neighboring puncta's track must end

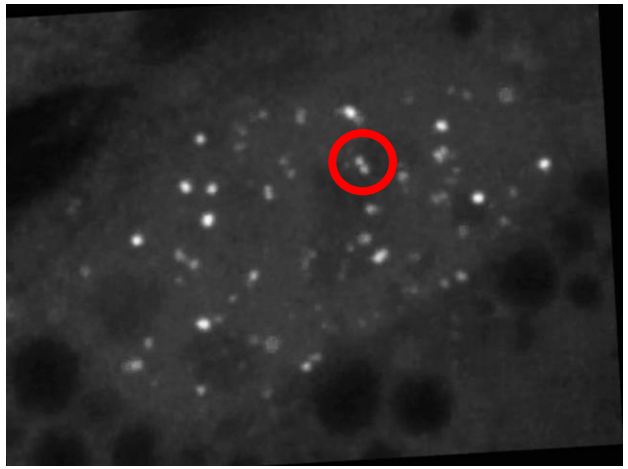
Algorithm: If puncta track ends, search for its neighboring puncta and check their area, circularity

-> Appearance of new puncta

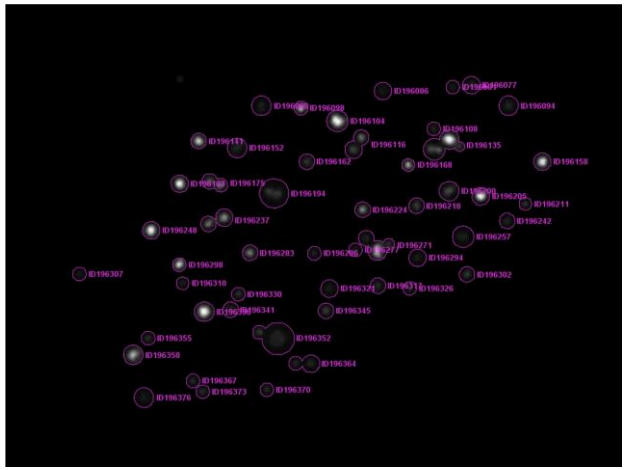




Evaluation



Manually mark merging event for each frame

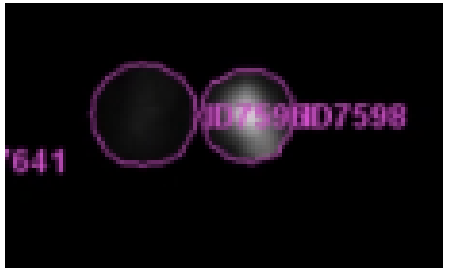


Find the matching puncta and its x,y coordinates in specific frame

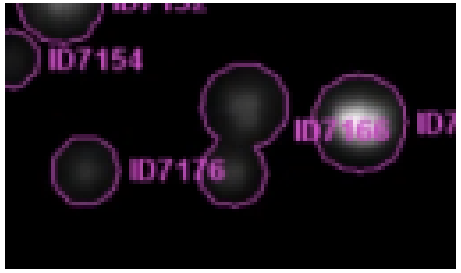
Compare and get the accuracy of the merge function

Detect merge event
Frame #
label id_1, label id_2
x,y coordinate

Puncta Merge Detection



Frame 10

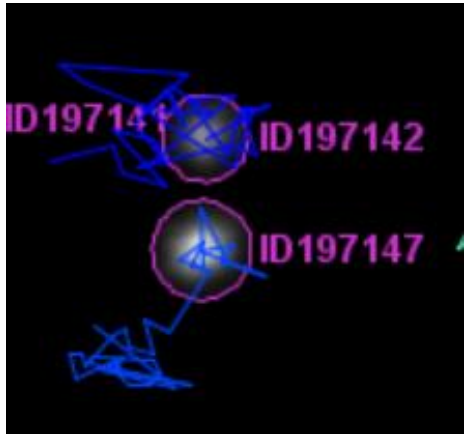


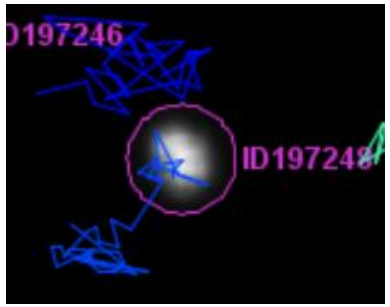
Frame 11

Assumption: In order for merging event to occur, existing neighboring puncta's track must end

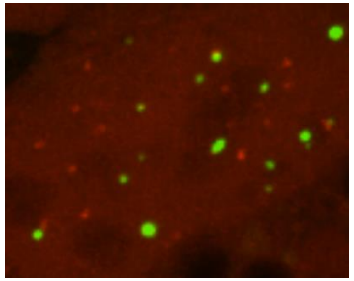
Algorithm: If puncta track ends, search for its neighboring puncta and check their area, circularity

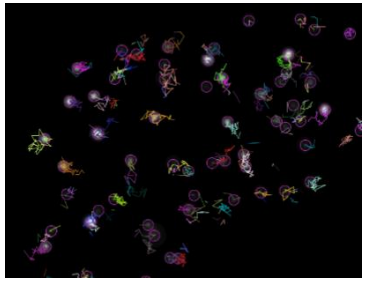
-> Appearance of new puncta





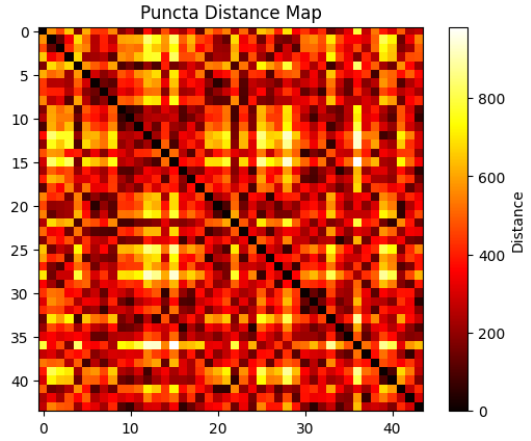
Puncta Merge Detection Function





Preprocessing

Simple LAP Tracker

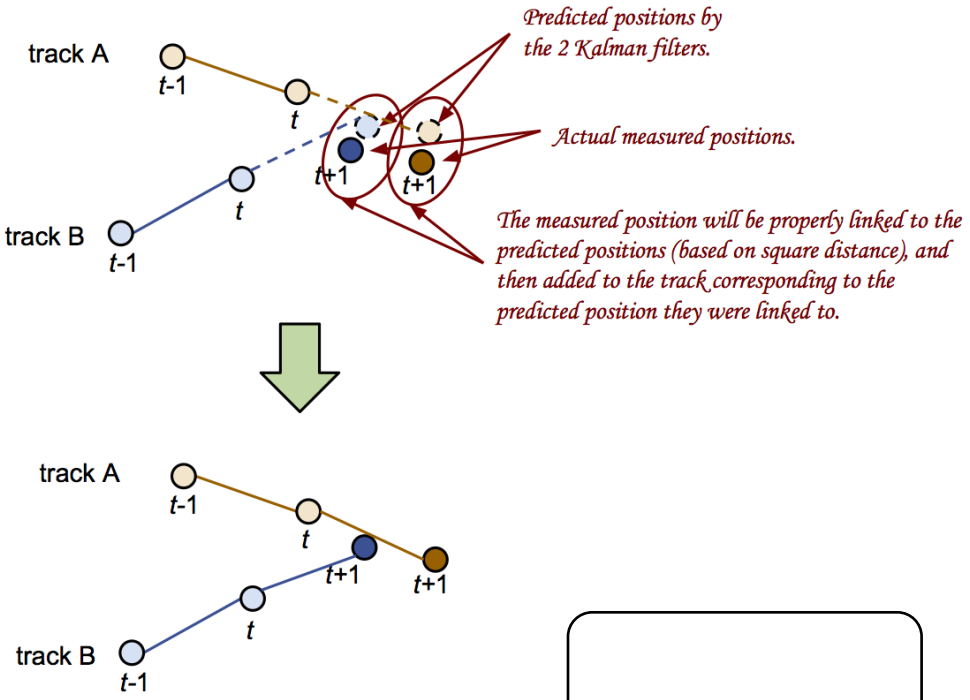


Compare radius sum and distance + Detect circularity

Merged over 3 consecutive frame

Merge

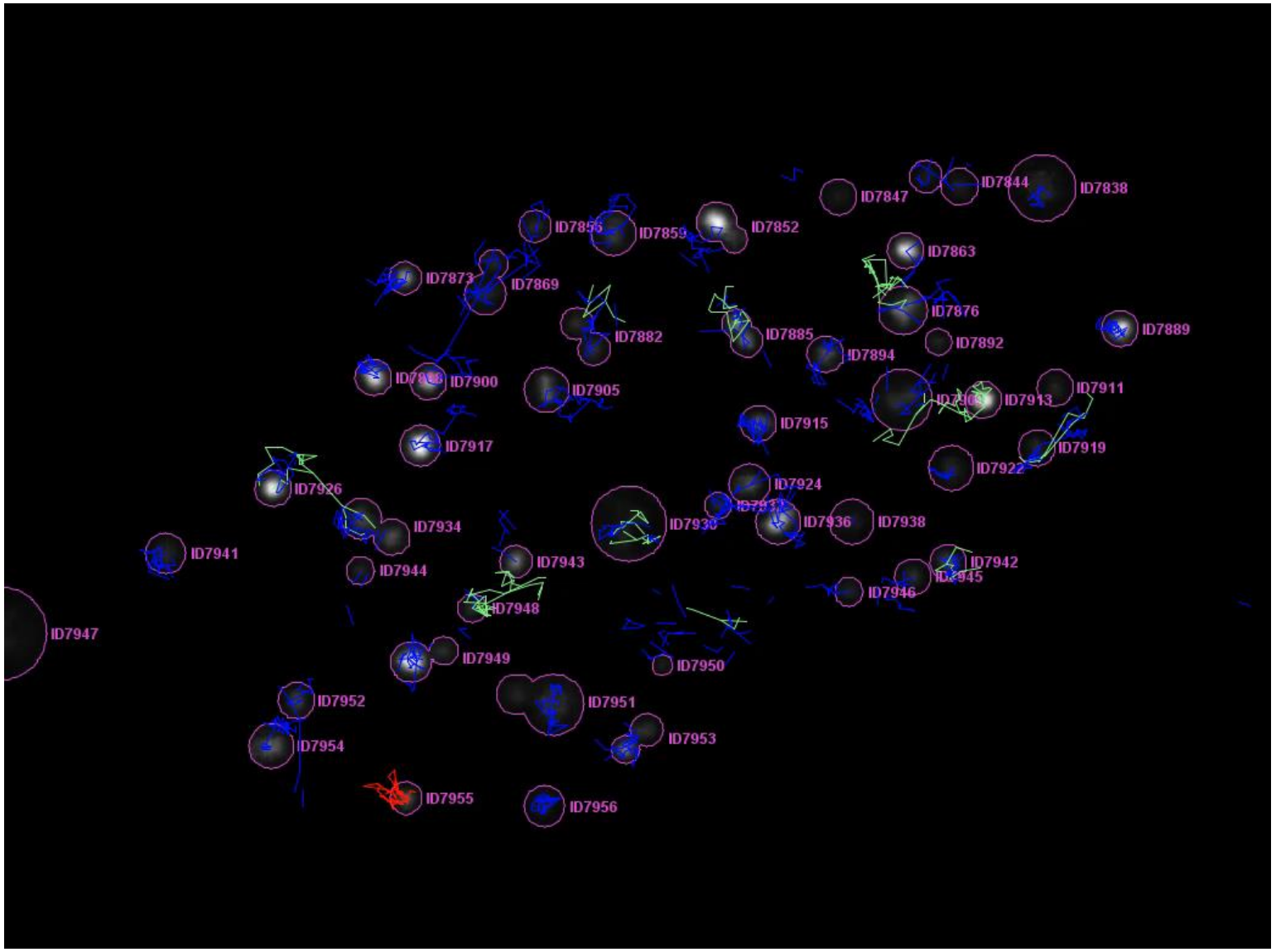
Advanced Kalman Tracker



- This tracker is an extended version of the Kalman tracker, that adds the possibility to customize linking costs and detect track fusion (merging) and track division (segments splitting).
- This tracker is especially well suited to objects that move following a nearly constant velocity vector. The velocity vectors of each object can be completely different from one another. But for the velocity vector of one object need not to change too much from one frame to another.
- Feature Penalties: Area, Circularity, Quality, Frame, Radius, Visibility, Manual spot color, Intensity(Mean Intensity, Median Intensity, Max Intensity, Sum Intensity, STD intensity),,, etc

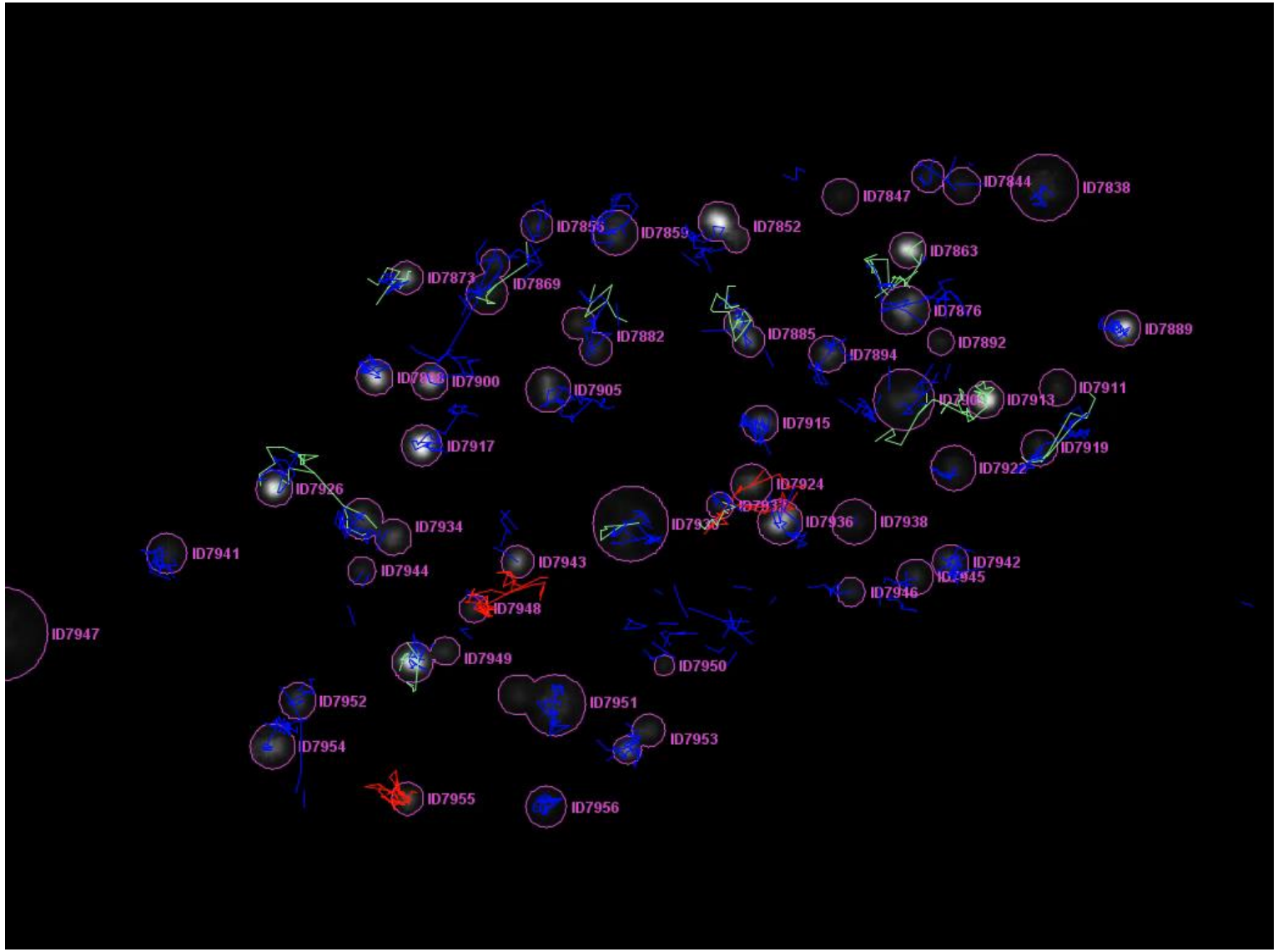
Initiated from a spot → Predict probable position → Predicted positions linked Jaqaman LAP framework(cost: square distnace) → Check for Occlusion

Kalman Tracker – Merging Event



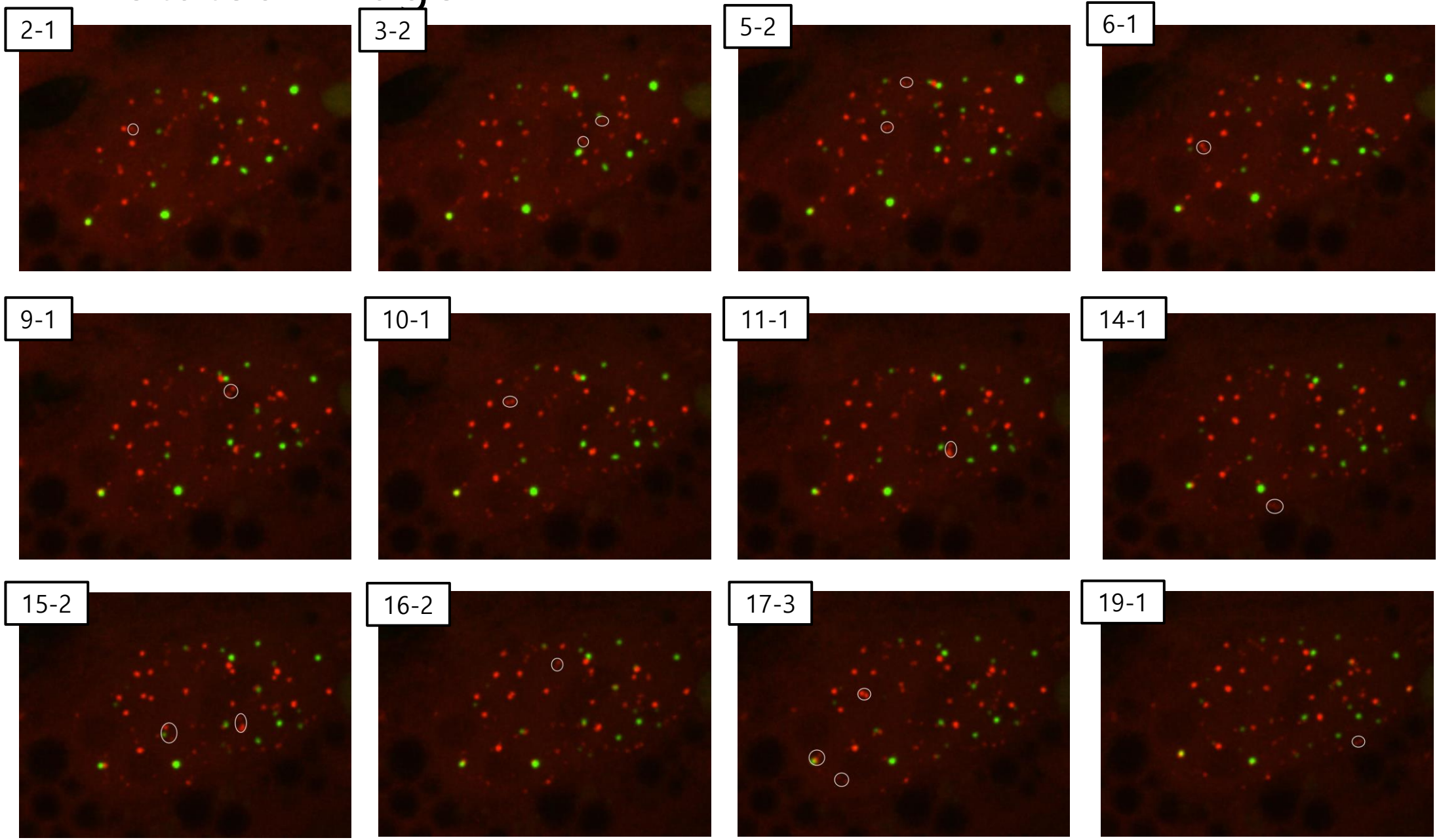
- Blue:0 merging event
- Green:1 merging event
- Red: 2 merging event

Kalman Tracker – Splitting Event

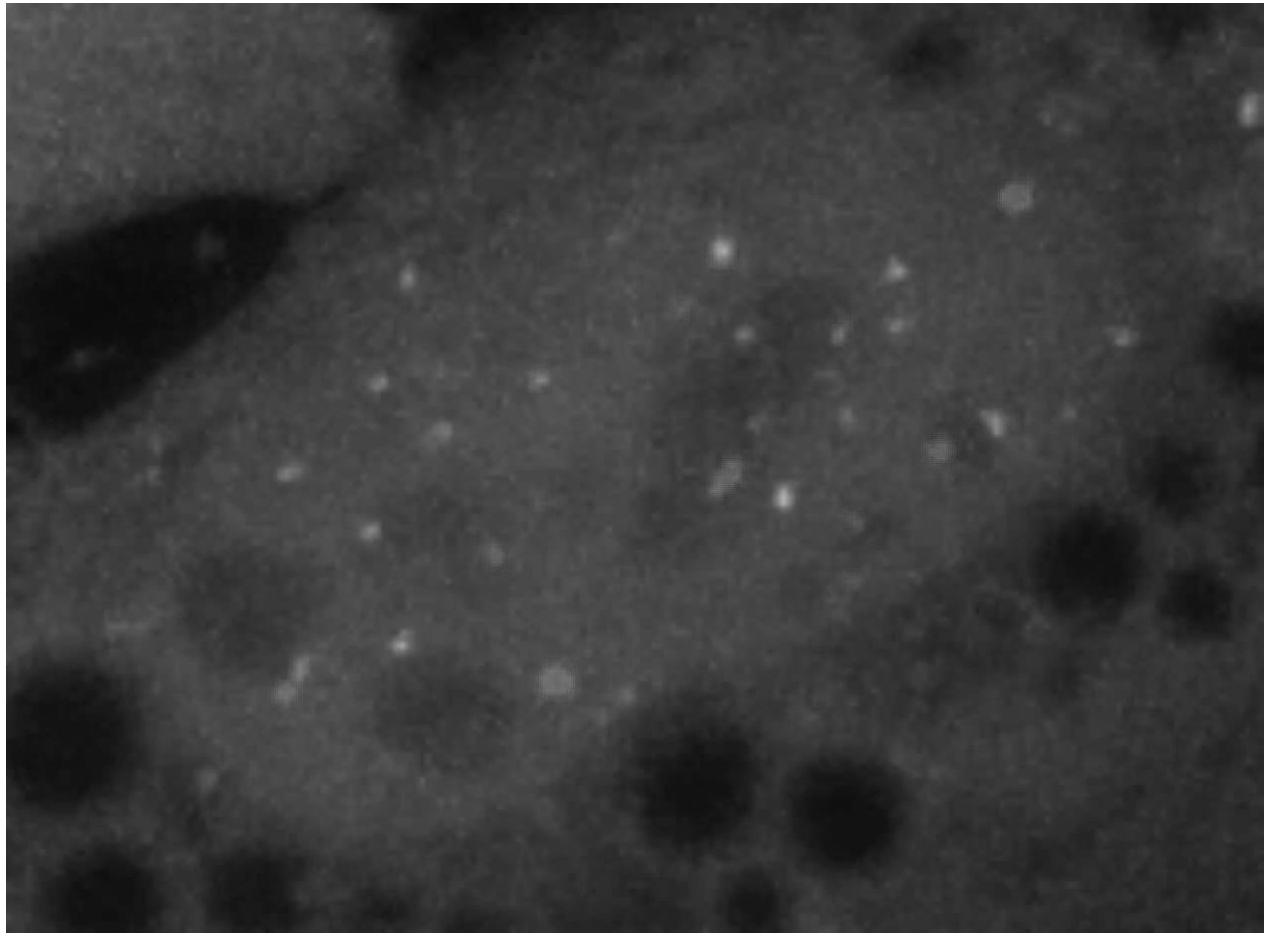


- Using Merging Event and Splitting Event, we can filter puncta that stays merged over 3 frames
- Applying various filter penalties can allow better detection

Annotated Image



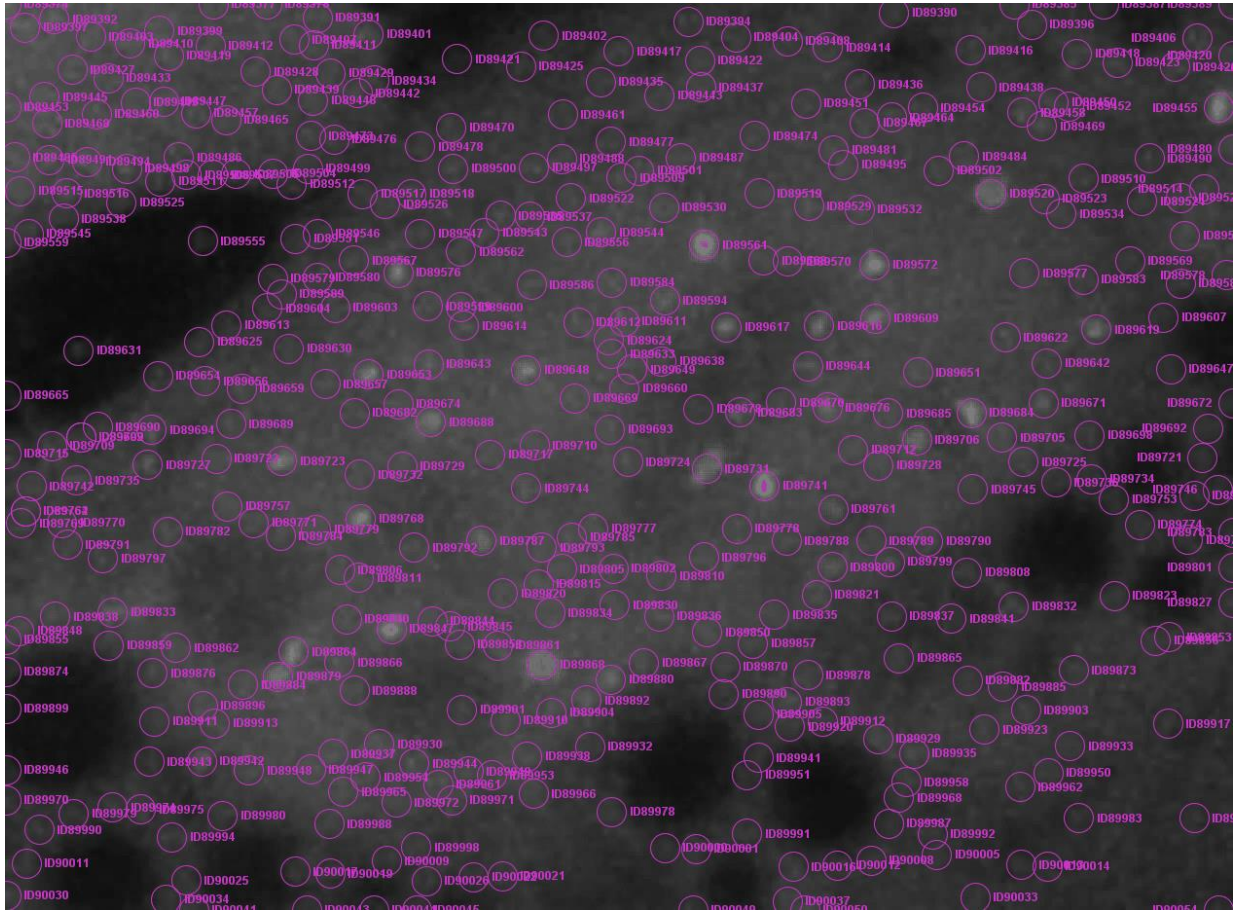
Data Situation

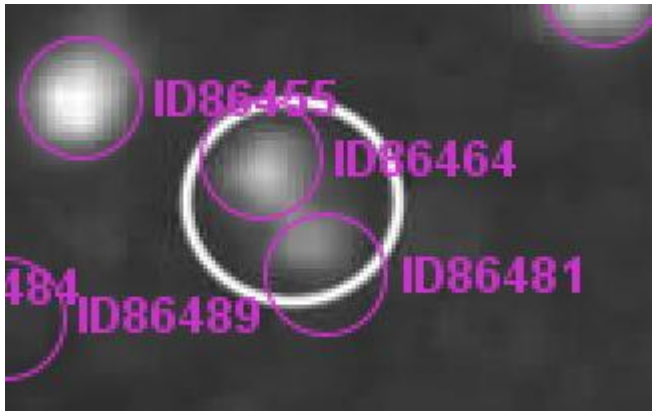


Kalman Tracker	Positive(11)	Negative(0)
True(23)	3	20
False(0)	8	0

Type	Num
Annotated	23
Kalman Tracker	11(3)
Distance based Tracking	7

Finding x, y coordinates of annotated merging event



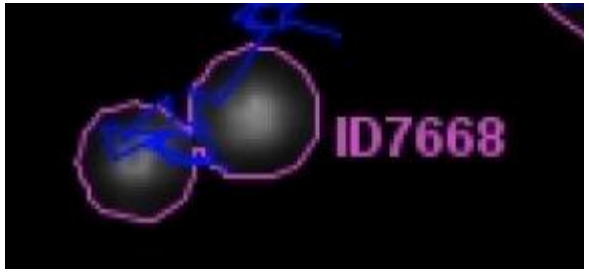


Annotated merging event: X, Y, Frame num, mean intensity

Label	Spot ID	Track ID	Quality	X	Y	Z	T	Frame	R	Visibility	Spot color	Mean ch1	Median ch	Min ch1
			(quality)	(pixel)	(pixel)	(pixel)	(frame)		(pixel)			(counts)	(counts)	(counts)
ID89374	893		0.1572335	0	0	0	0	0	15	1		42.353805	37	0
ID86455	86455		1.6881397	331.14259	436.87679	0	5	5	15	1		147.38584	137	75

Evaluation Function- Distance based Tracking

Compare the events in the same frame for merging puncta



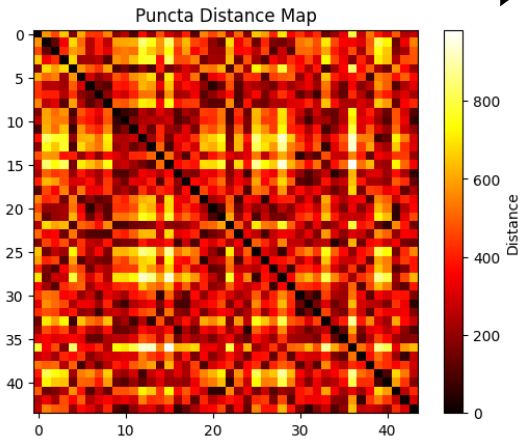
X,Y coordinate of merging puncta

X,Y coordinate of puncta

```
// Spot in pixel coordinates.
final double x = x0 + r * Math.cos( phi );
final double y = y0 + r * Math.sin( phi );

// But we want to create spots in image coordinates:
final double xpos = x * calibration[ 0 ];
final double ypos = y * calibration[ 1 ];
final double zpos = 0d;

// Create the spot.
final Spot spot = new Spot( xpos, ypos, zpos, SPOT_RADIUS, 1d / ( nspiral + 1d ) );
spots.add( spot );

// Loop to another spiral.
```

-> trackmate puncta x, y coordinate

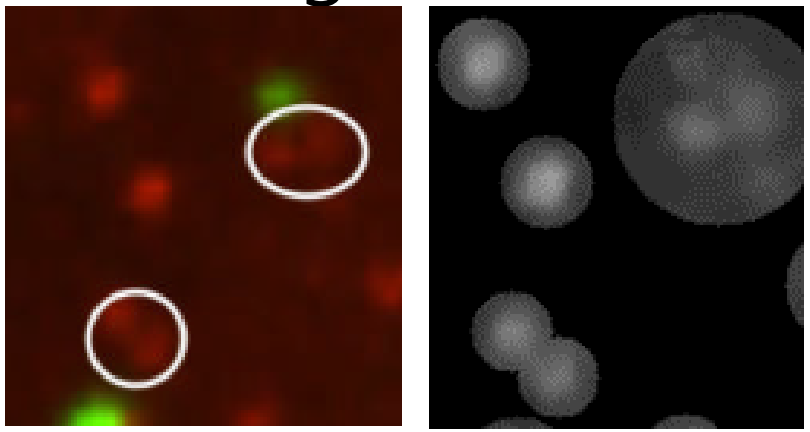
-> trackmate puncta x, y coordinate of two puncta



Make distance map, compare with distance value

Count the merging events

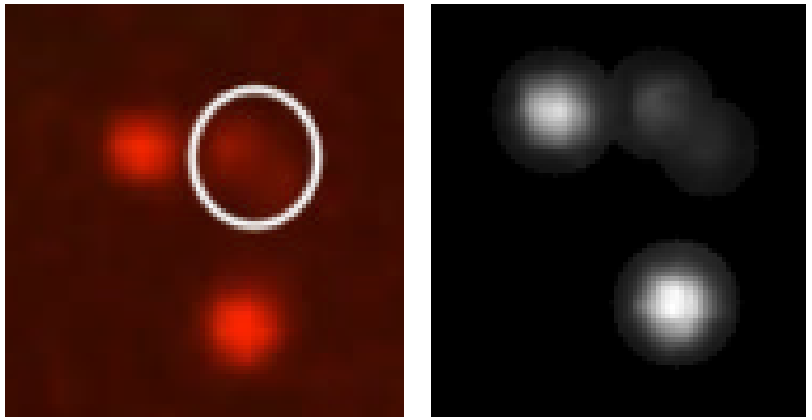
Checking the misdetection- Distance based Tracking

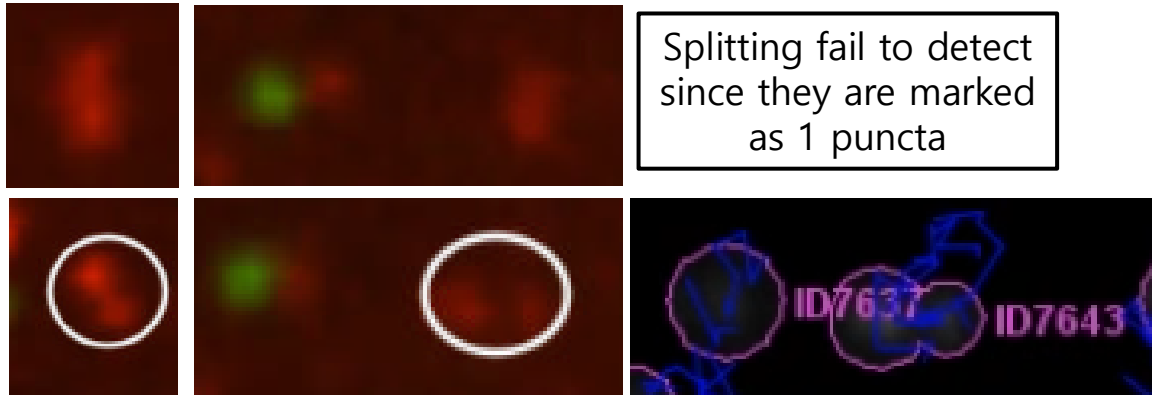


Masking failure -> Problem with preprocessing (1)

Mask Detecting -> Masked but misdetection;

Tracking -> Tracking failure; Problem with tracking algorithm





Splitting fail to detect since they are marked as 1 puncta

Starts with merging shape; recognize as 1 puncta (7)

Should be counted as separate puncta

Preprocessing: Apply circularity and mark it as separate puncta

Apply circularity and count merge

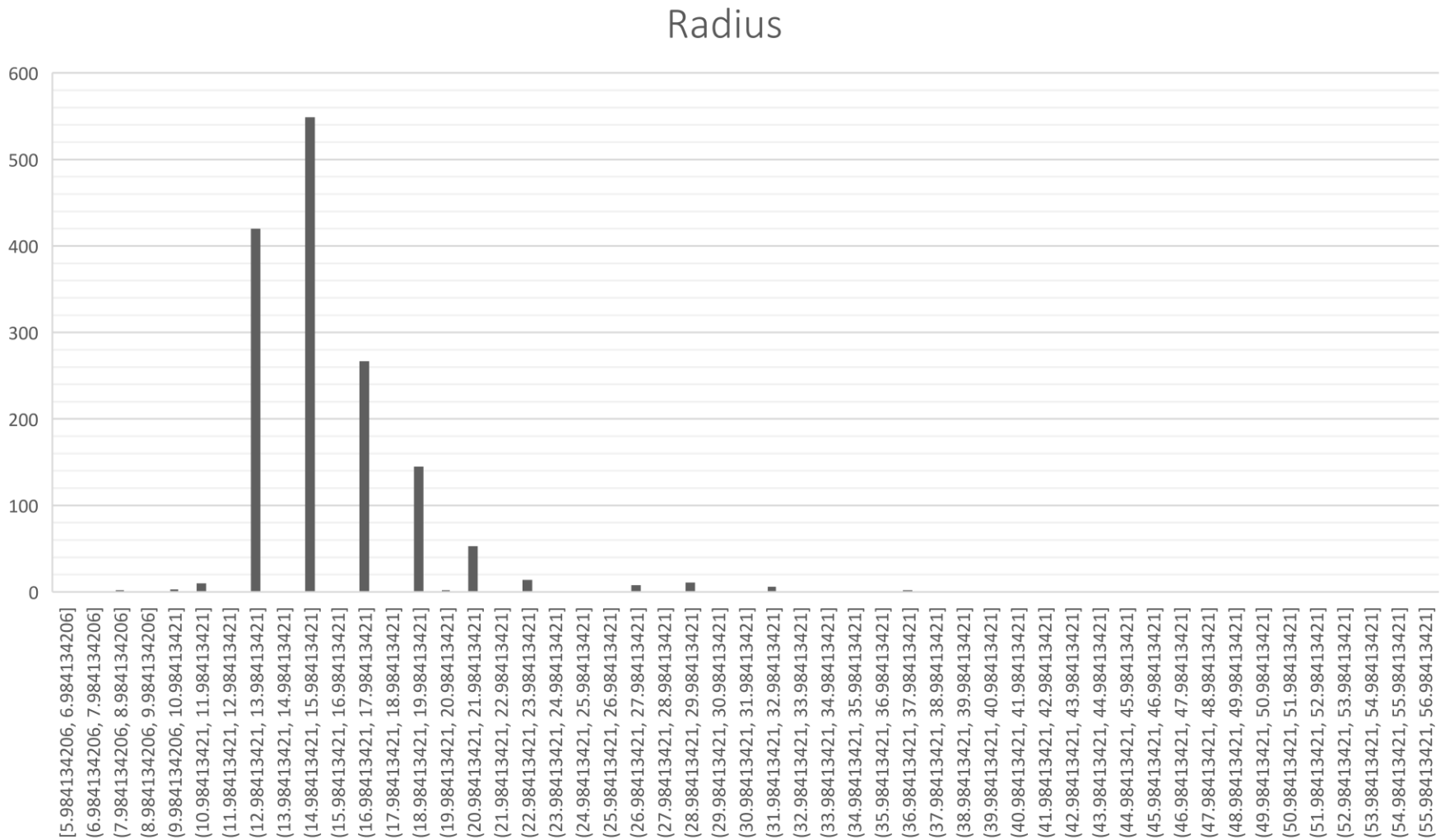
Parameters

Frame to Frame Linking	Explanation
Initial Search Radius	The initial search radius represents the initial search range used to locate objects or particles when tracking begins.
Search Radius	The search radius denotes the search range used to locate the object in the next frame, starting from its previous position.
Max Frame Gap	Max frame gap is a parameter defining the discontinuity of a track, indicating how many frames an object can be absent before the track is considered broken.

Track Segment Merging	Explanation
Max distance	the maximum spatial separation between the end of one track segment and the beginning of another for them to be considered for merging.

Track Segment Splitting	Explanation
Max distance	the maximum spatial separation between the end of one track segment and the beginning of another for them to be considered for merging.

Choosing the radius



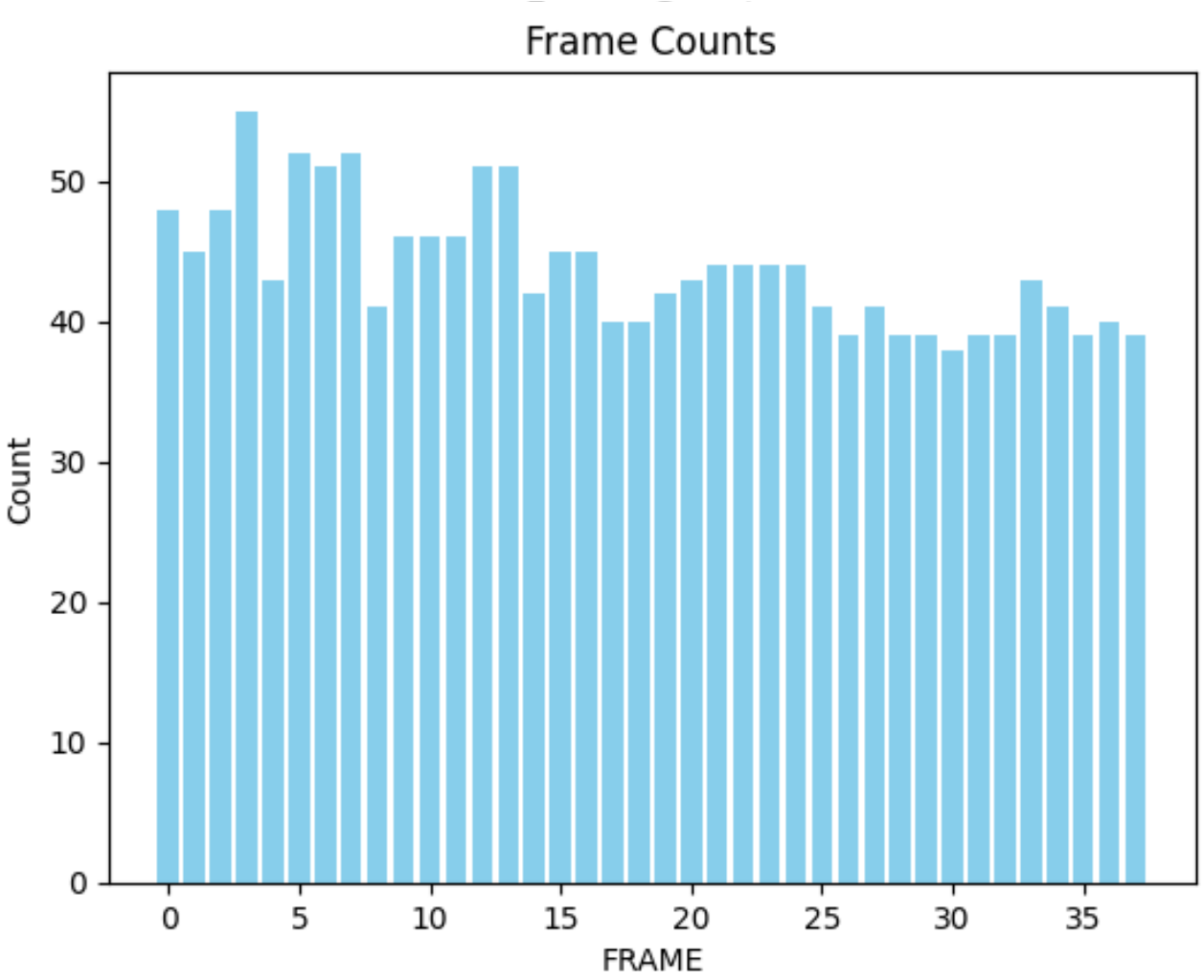
Circularity filter: 0.90

Average & Mode

Average : 17.249

Mode : 15.937

Choosing the quality threshold for puncta



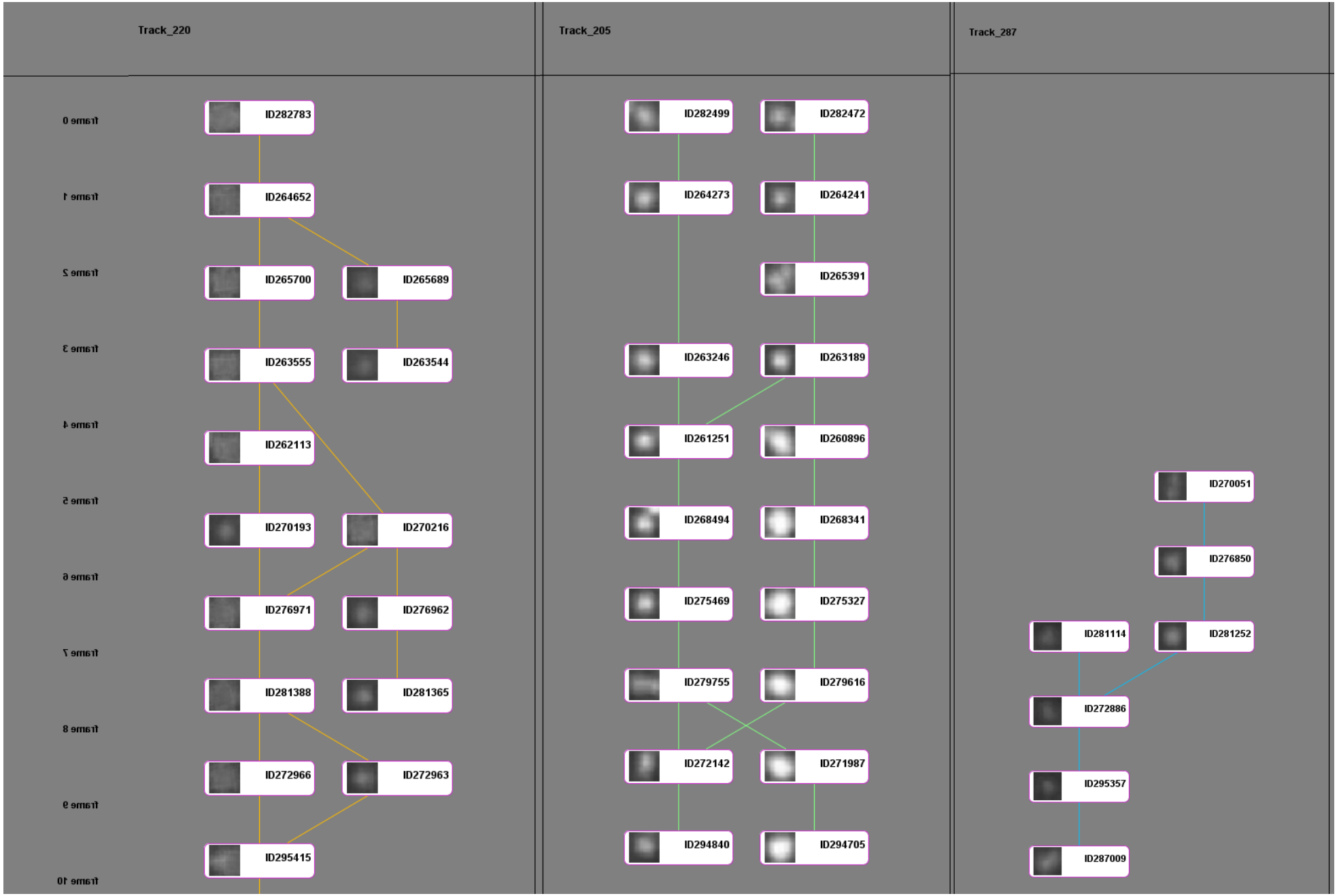
#puncta/frame: 43.815

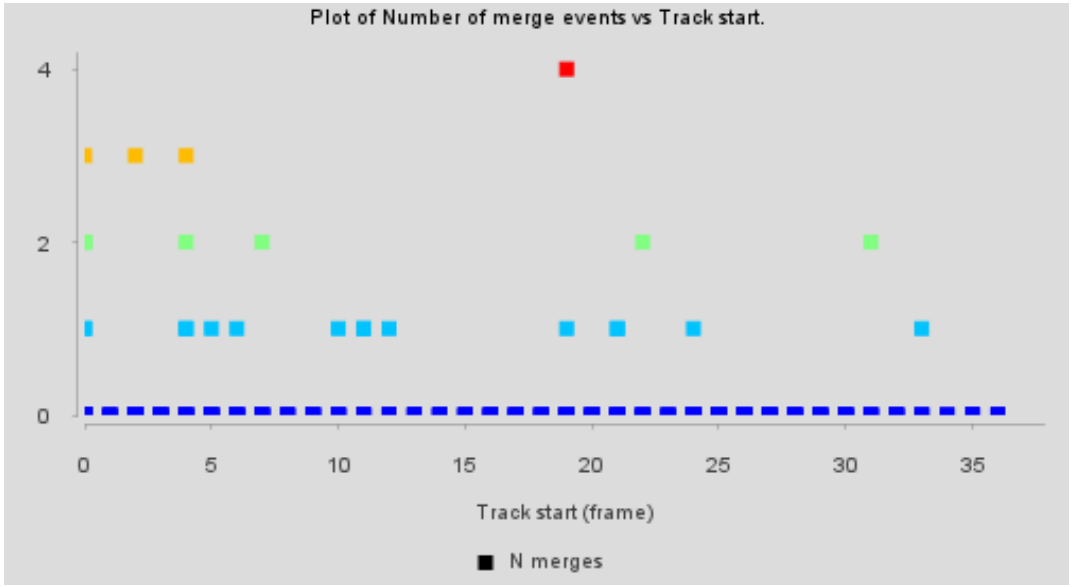
↓

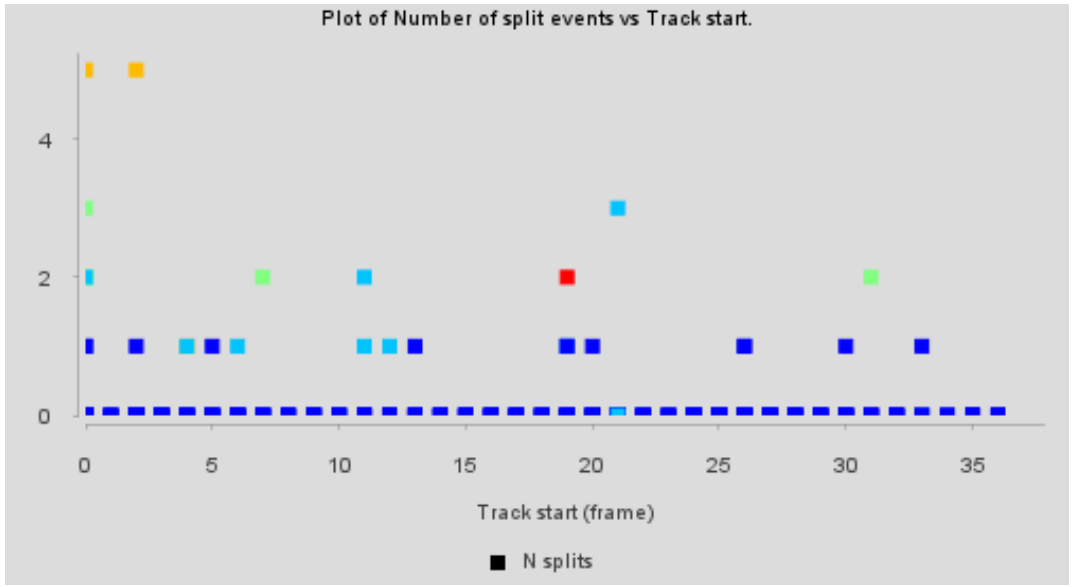
Choose quality threshold with 44 puncta

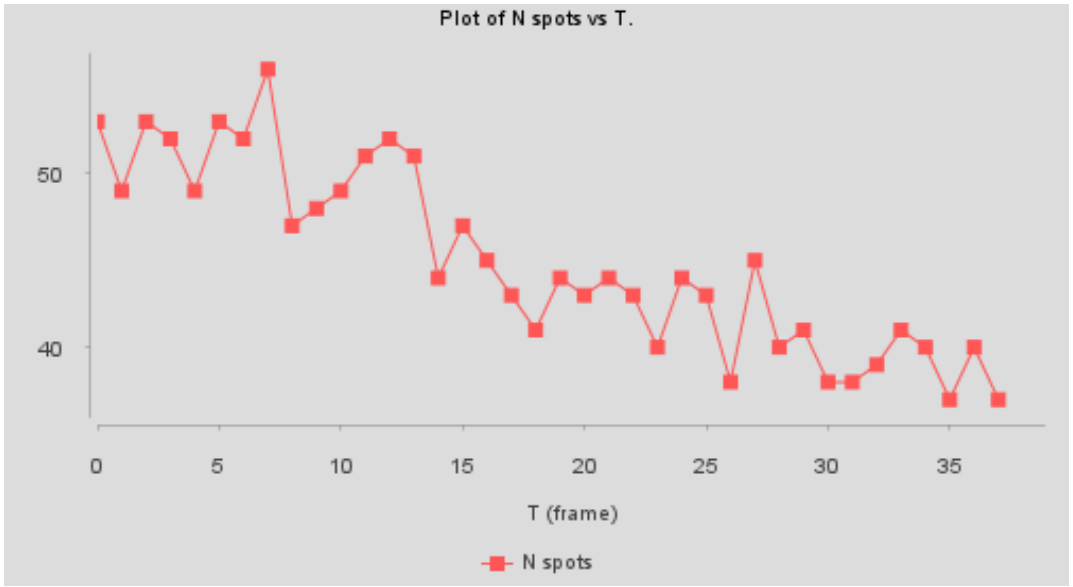
Parameter Result

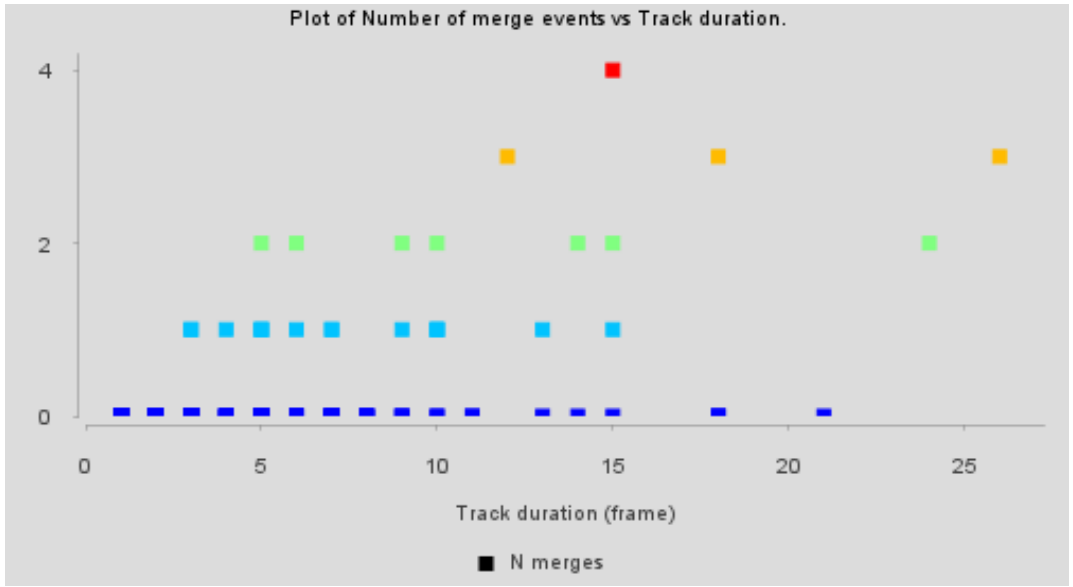
Quality Threshold	Initial Search Radius	Search Radius	Max Frame Gap	Max distance(Merge)	Max distance(Split)	Merge Event(dur)	Correspondence
0.08(example)	30	60	1	50	50	42	/23
0.53(44)	17.249(Mean)	51.747(x3)	1	51.747	51.747	24	/23
0.7(44)	15.937(Mode)	47.811(x3)	1	47.811	47.811	16	/23


Track_220
Track_205
Track_287
frame 0
frame 1
frame 2
frame 3
frame 4
frame 5
frame 6
frame 7
frame 8
frame 9
frame 10
ID282783
ID264652
ID265700
ID265689
ID263555
ID263544
ID262113
ID270193
ID270216
ID276971
ID276962
ID281388
ID281365
ID272966
ID272963
ID295415
ID282499
ID282472
ID264273
ID264241
ID265391
ID263246
ID263189
ID261251
ID260896
ID268494
ID268341
ID275469
ID275327
ID279755
ID279616
ID272142
ID271987
ID294840
ID294705
ID270051
ID276850
ID281114
ID281252
ID272886
ID295357
ID287009


Plot of Number of merge events vs Track start.
4
2
0
0
5
10
15
20
25
30
35
Track start (frame)
N merges


Plot of Number of split events vs Track start.
4
2
0
0
5
10
15
20
25
30
35
Track start (frame)
N splits


Plot of N spots vs T.
50
40
0
5
10
15
20
25
30
35
T (frame)
N spots


Plot of Number of merge events vs Track duration.
4
2
0
0
5
10
15
20
25
Track duration (frame)
N merges