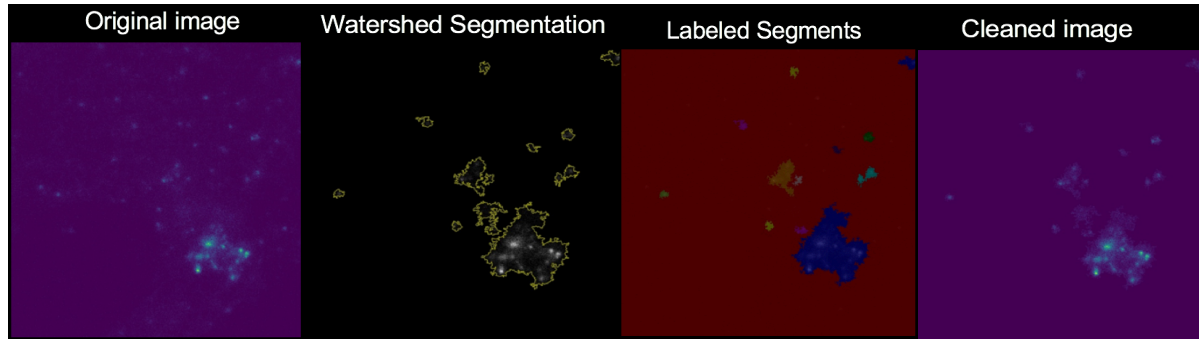# Computational methods to analyze the aggregation patterns of germ plasm ribonucleoparticles in zebrafish

## *Region-based segmentation of each z-section*

Confocal microscopy was used to obtain the z-stack of germ plasm labeled for RNA thought to comprise RNPs. The change in the fluorescent signal throughout the image was used to create an elevation map by which regions of fluorescence (peaks)could be distinguished from the background (valleys) [1]. The image was then segmented by applying a watershed transformation, wherein valleys are filled until the paths between the peaks forms a watershed and identifies regions of elevation [2]. Any remaining holes were filled, smaller segments were removed, and the final segments were labeled for better visualization (Fig. 1, above).

Laplacian of Gaussian blob detection
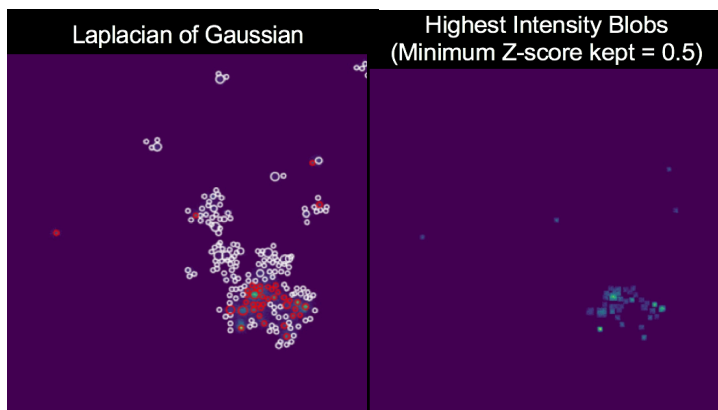
To detect discrete areas of high intensity in each segment, a Laplacian of Gaussian blob detection algorithm was used. The radius and location of these high intensity spots was measured and identified regions were considered spherical RNPs (red circles, Fig. 2, left). After obtaining the x, y, z coordinates of all RNPs from the z-stack, RNPs were plotted in 3D space (Fig. 3, below).
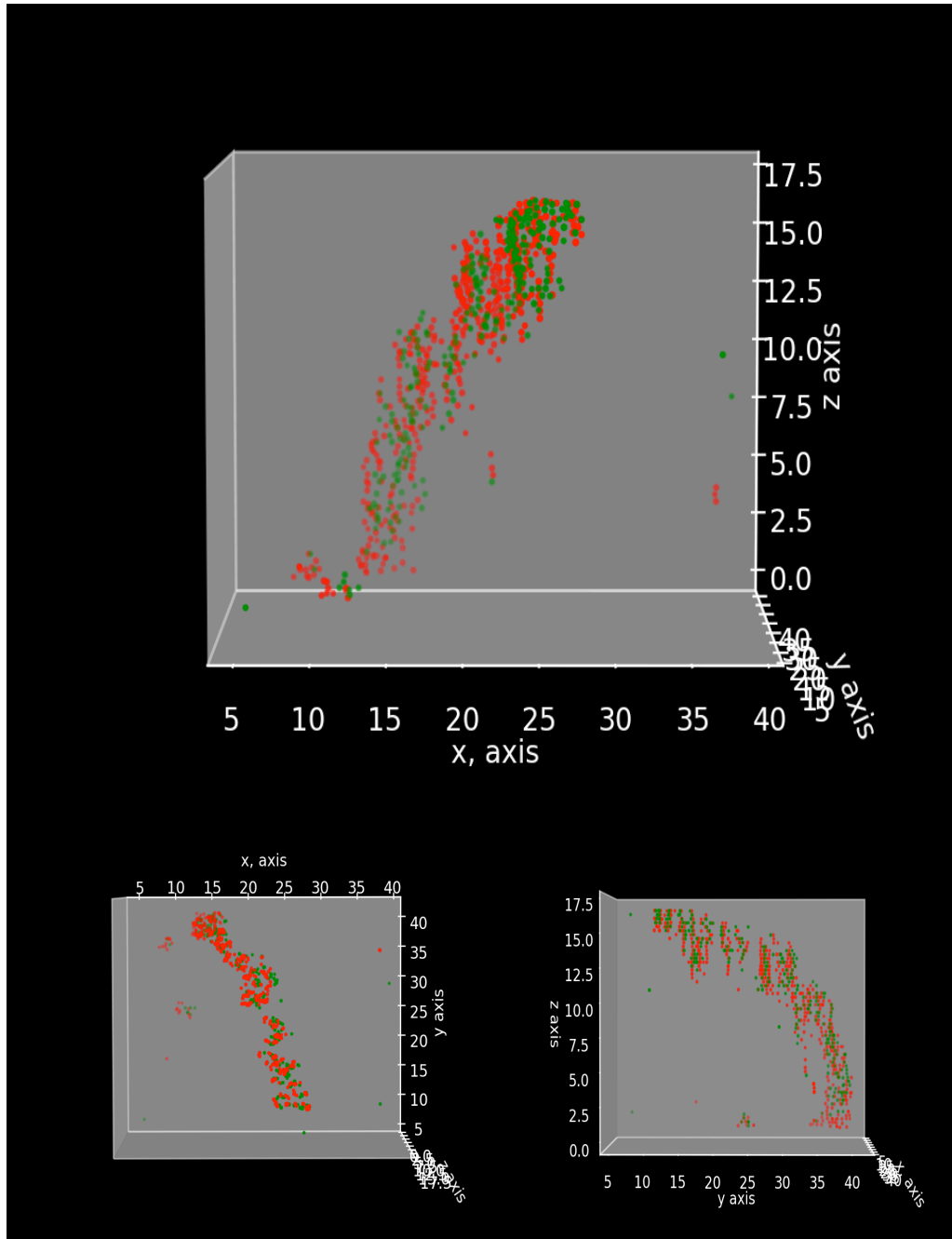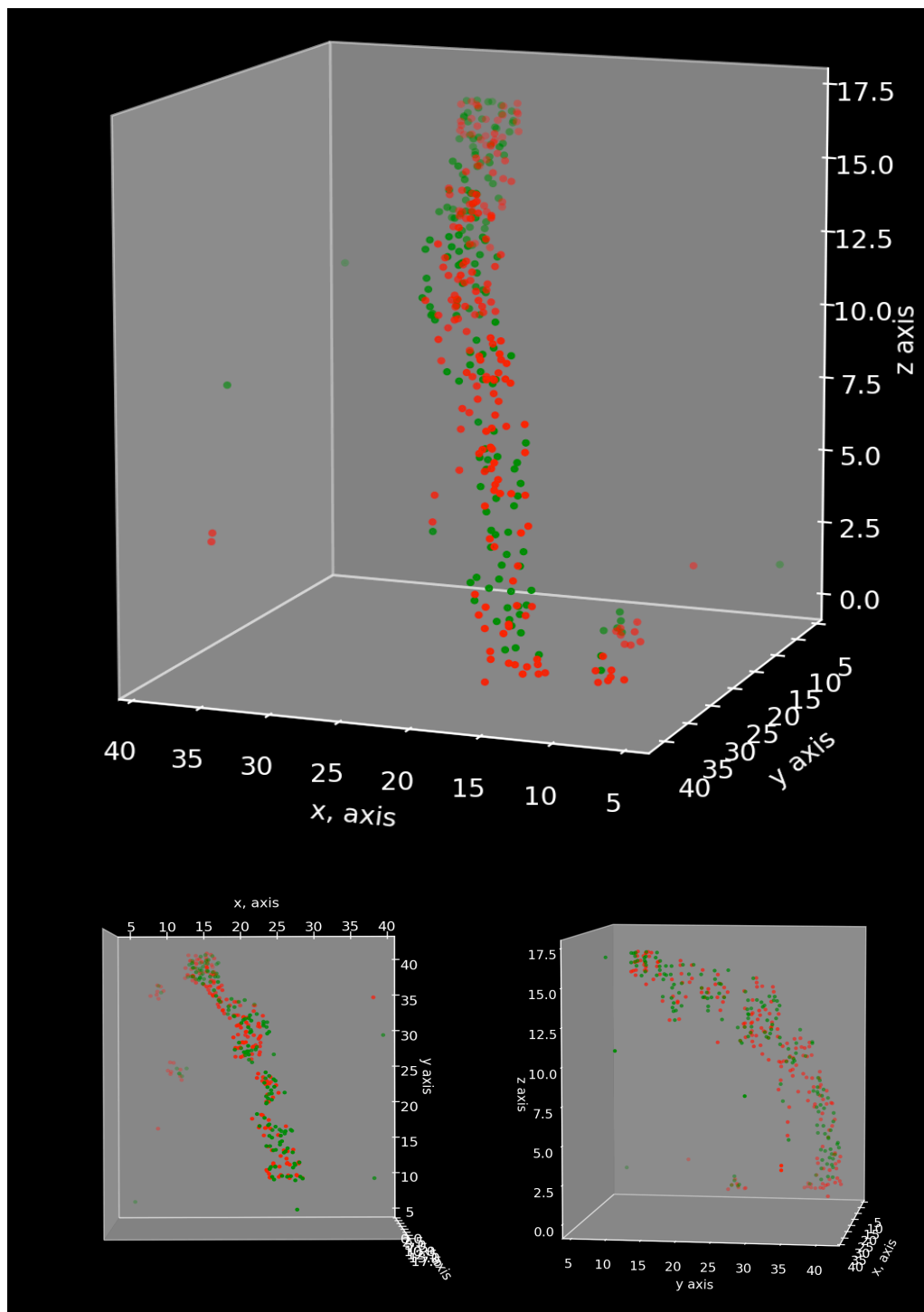
Different views of the 3D model

Clustering

The y-z and x-z axes are susceptible to to point spreading of individual RNPs due to out of focus fluorescence in the focal plane, and as such can significantly impede the quantitative analysis of 3D models obtained from confocal microscopy [3]. An iterative, median based, constrained clustering approach was used to eliminate point spreading in the 3D model. RNP's were binned by size and clustered separately. RNPs were initially grouped based on their nearest neighbors.

RNPs within x ± radius, and y ± radius were grouped, and the z spread was determined for each group. After detecting all possible groups, the median spread in the z was determined and set as the z threshold for clustering. RNPs within the x, y, z threshold were represented as the center RNP. Other RNPs in the group were removed. See Fig. 4, below.


*Different views of the 3D model after clustering*

Straightening

To best assess aggregation patterns of RNPs, bent aggregates were straightened. The clustered coordinates were represented as an undirected, fully connected, weighted graph based on 3D coordinates. Edges connecting coordinates correspond to 3D distance. To determine the connectivity of coordinates through the aggregate, the Euclidean minimum spanning tree of the undirected weighted graph was computed using Prim's algorithm, whereby a minimal path through every node is calculated [4] (Fig. 5, above).

Minimum spanning tree

A breadth first search was used to rank order coordinates through the minimum spanning. A simple moving average with a window size of 100 was used to draw a smooth Bezier curve through the aggregate (Fig. 6, below). Vectors were used to represent the position of every coordinate from its closest point on the Bezier curve. The vectors were then plotted along the x axisto produced a straightened representation of the aggregate.
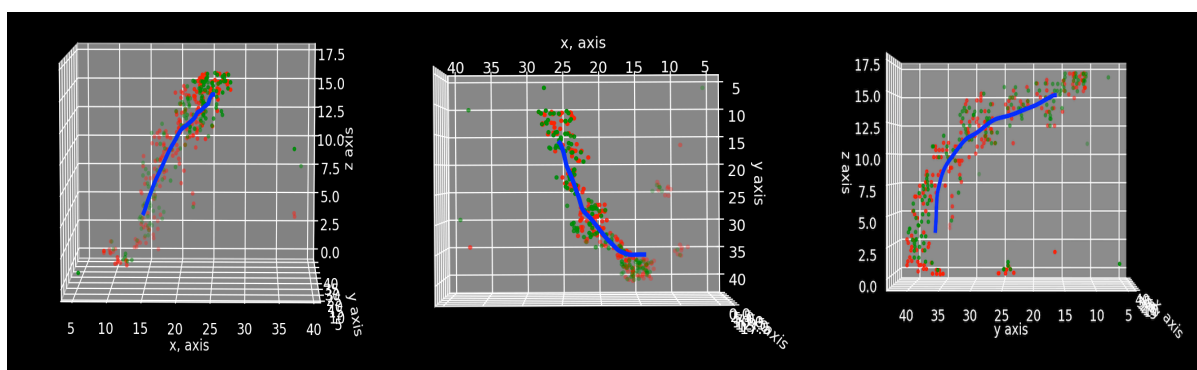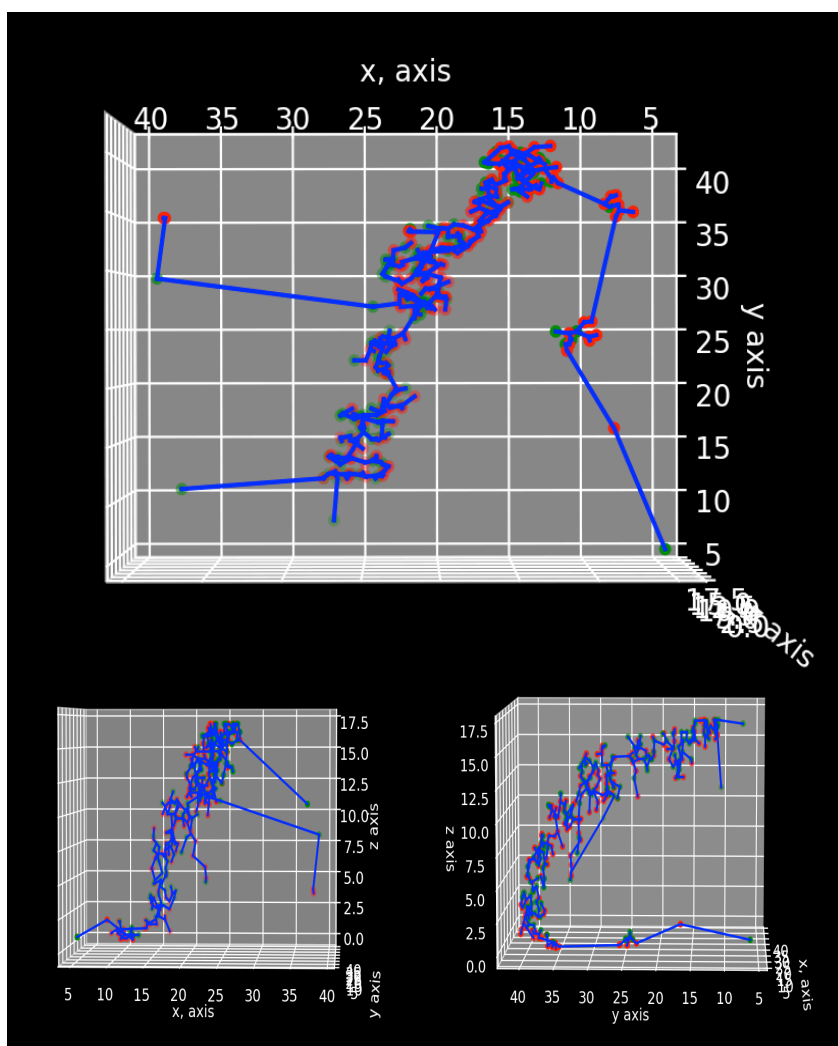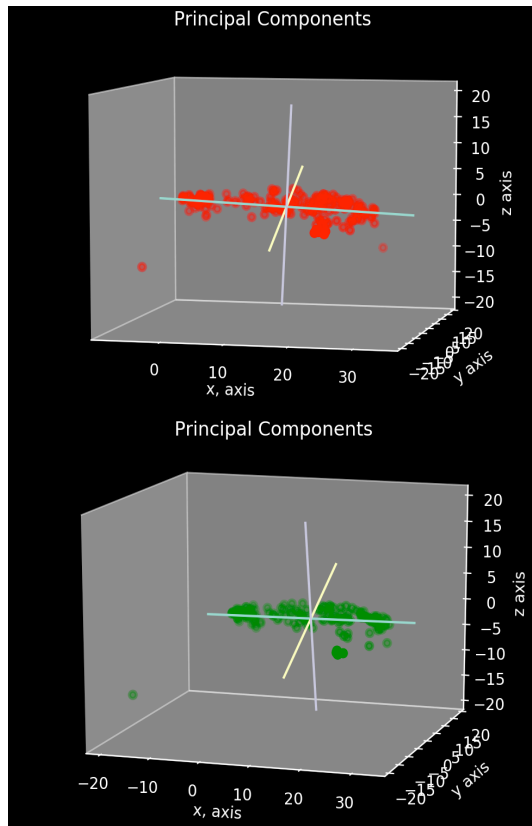

Bezier curve
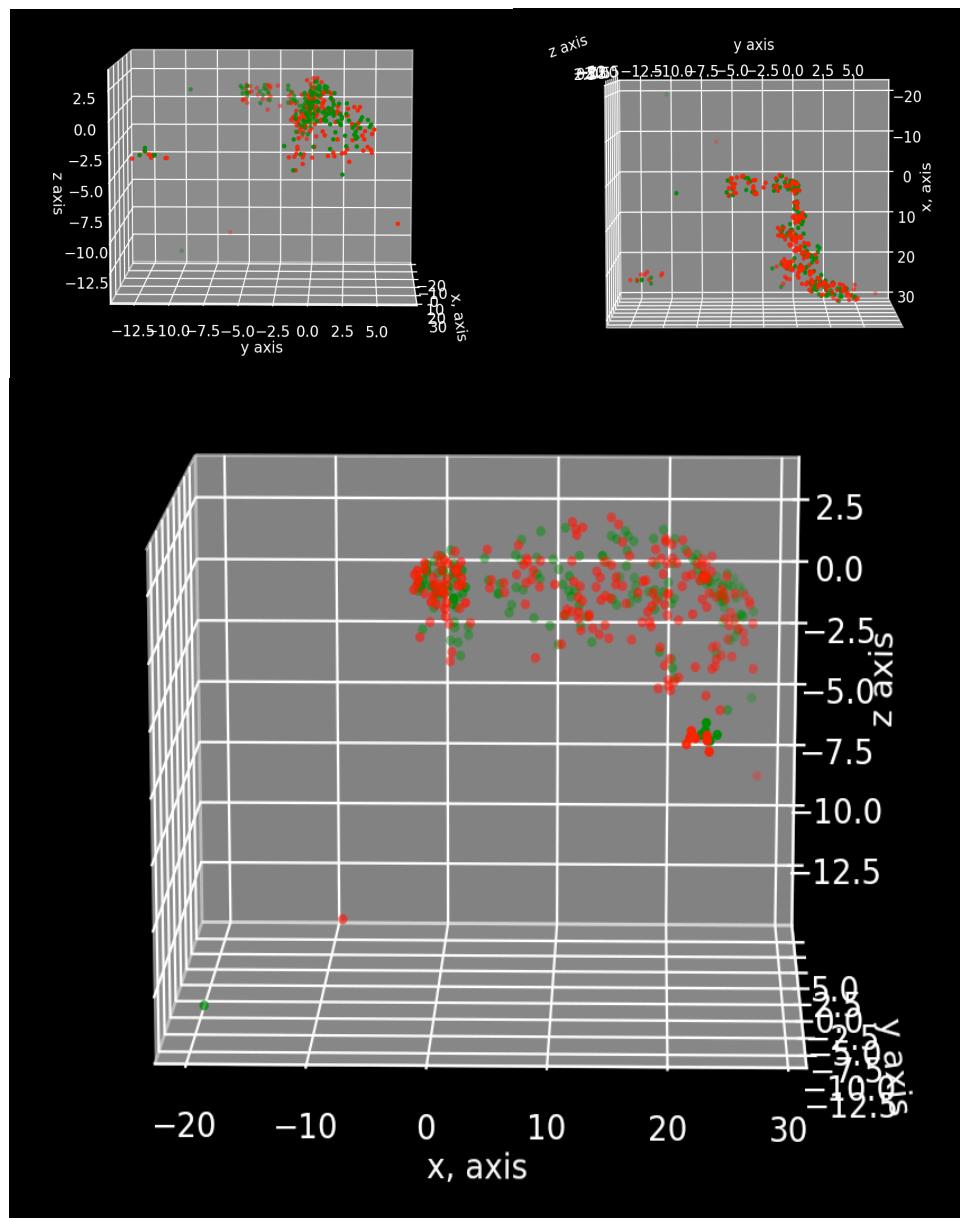
Straightened aggregate

Curve Fitting

For each channel, the first (PC1), second (PC2), and third (PC3) principal components were used to define the axes of the aggregate and center at the origin (Fig. 7, right). For every RNP, it's component in the PC1, PC2, and PC3 direction, C1, C2, and C3 was calculated to allow dimensional collapsing of the aggregate in every principal component direction. 2D plots of C1-C2, C1-C3, and C2-C3 were produced to facilitate the estimation of the aggregation pattern in 2D (Fig. 8, below).
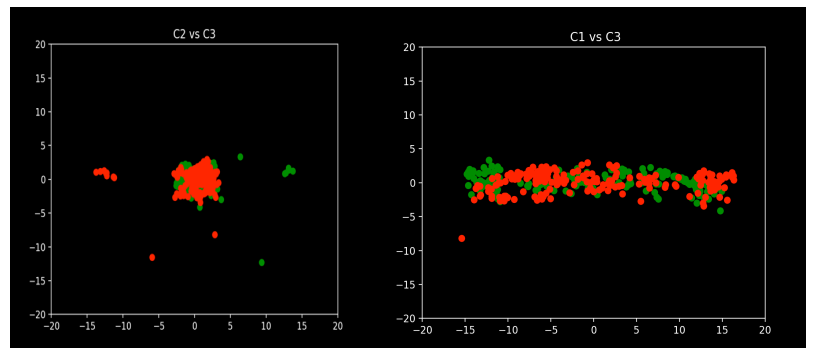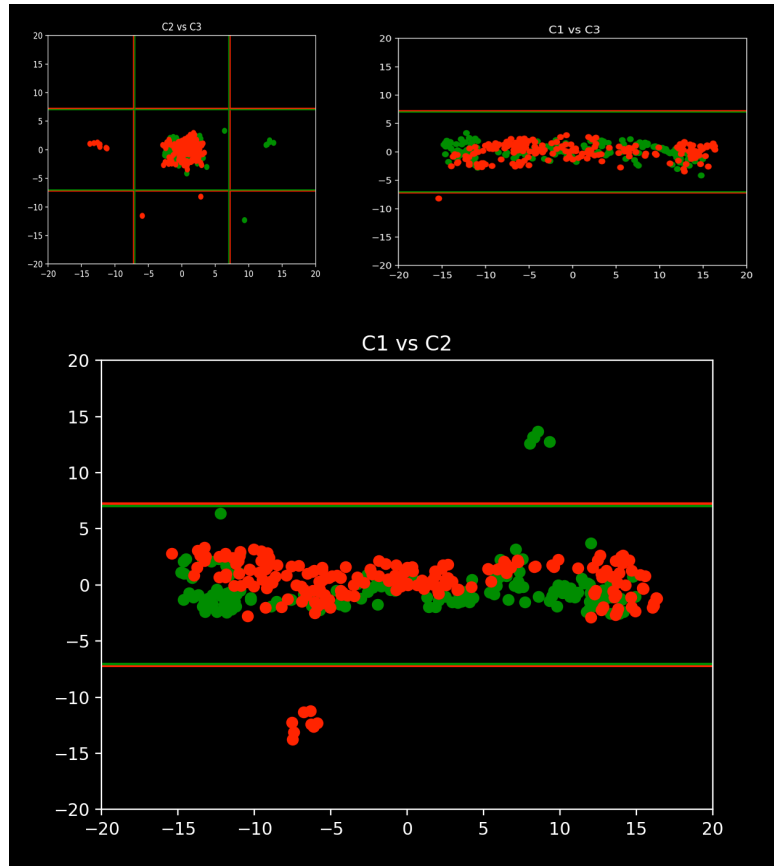

Using the C2-C3 plane, the average distance of the RNPs from the center was computed for each channel. Any RNP that was two standard deviations from the mean distance was removed to avoid skewing towards the outliers. The subsequent images show the boundaries within which RNPs were considered for further computation (Fig. 9, below).
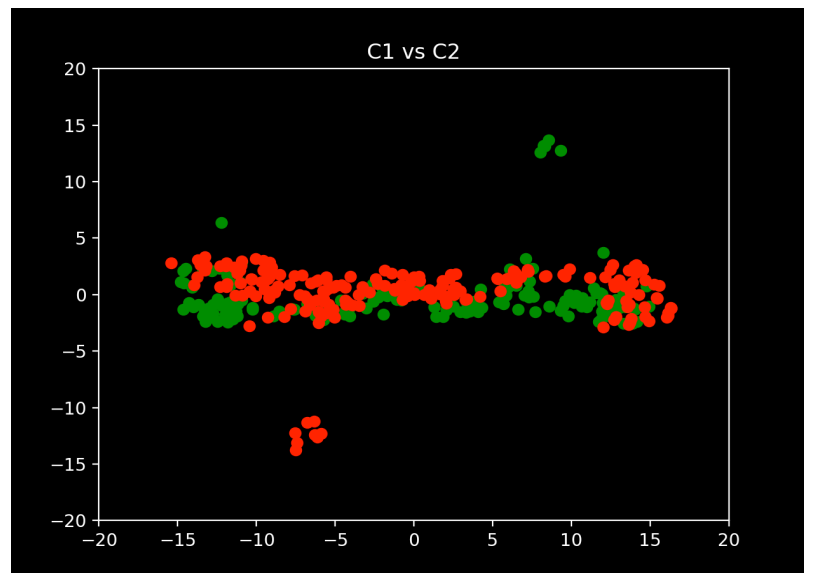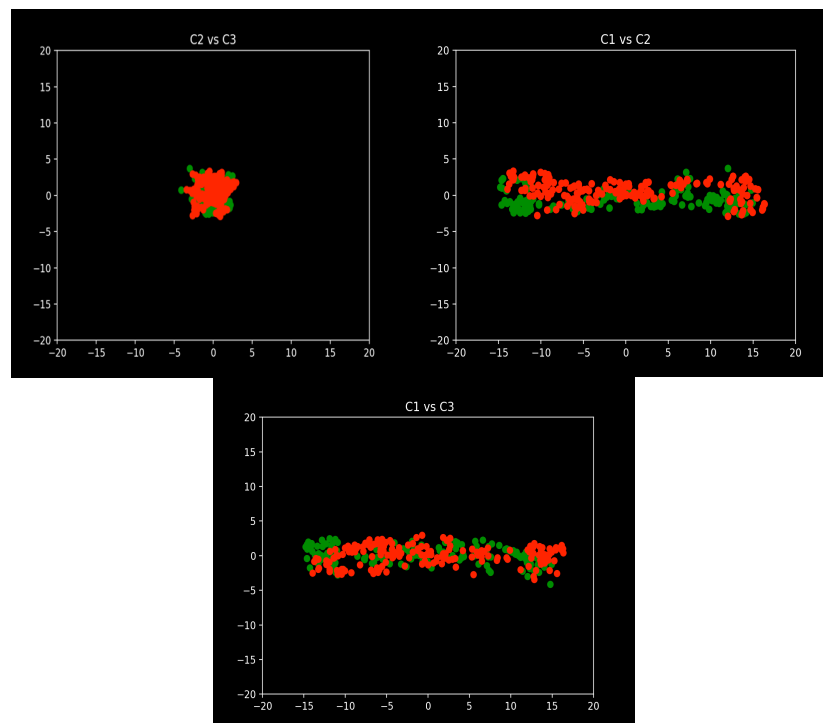

2D plots of the aggregate

Outlier Boundaries

Cleaned 2D plots

After the removal of outliers, least squares minimization was used to approximate a curve that best fits the aggregation pattern of RNPs. A range of initial estimates of radii (0-15), frequency (0-1), and phase (0-2π) were used for curve fitting to best approximate the global maximum fit. This was done for each channel in C1-C2 and C1-C3 (Fig. 10, below).
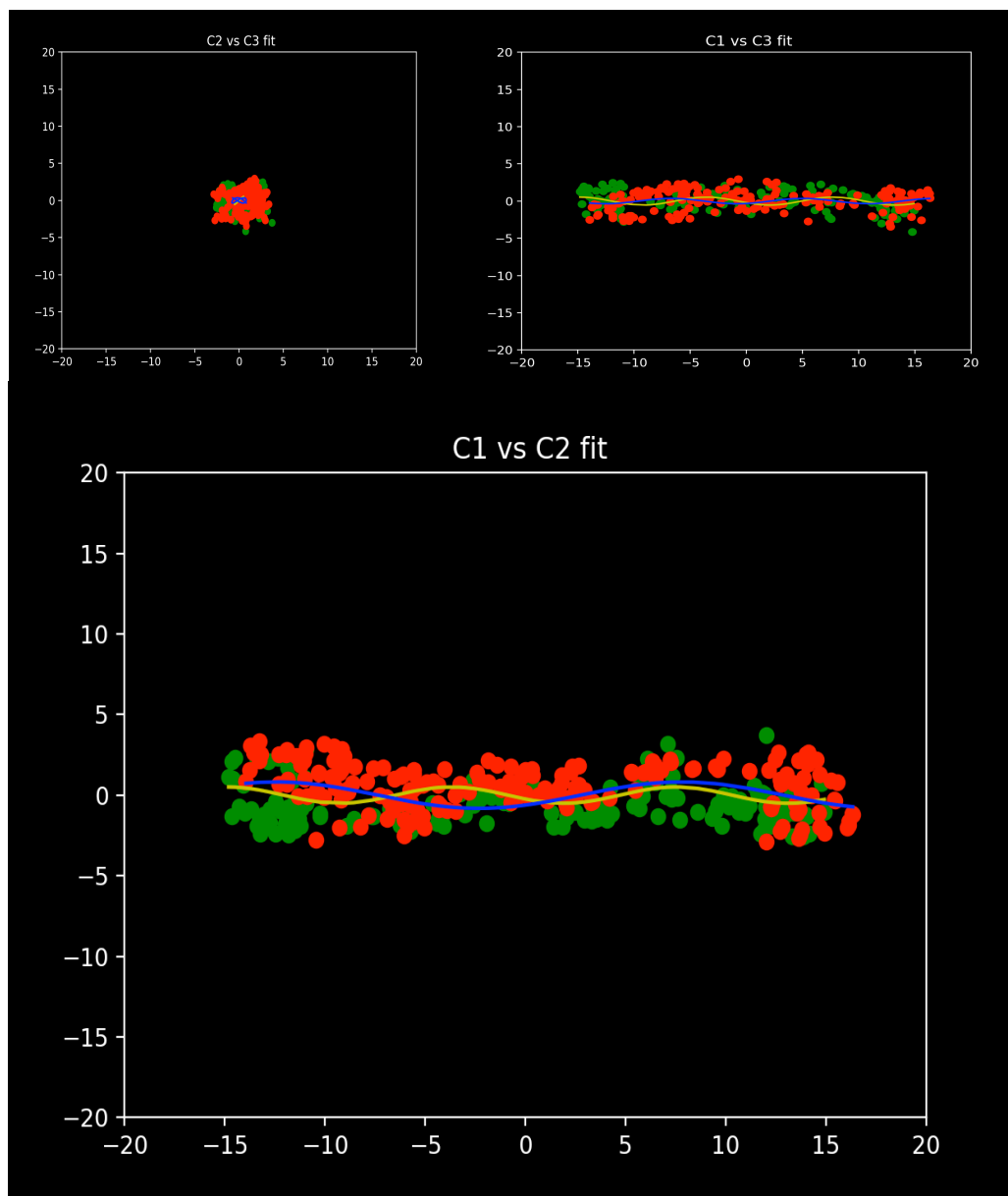
The curves fit in 2D for each channel were observed to be sinusoidal, suggesting a helical structure in 3D. A 3D fit was obtained by consecutively finding the centers of the fits in C1-C2 and C1-C3 for each channel.

2D Fits
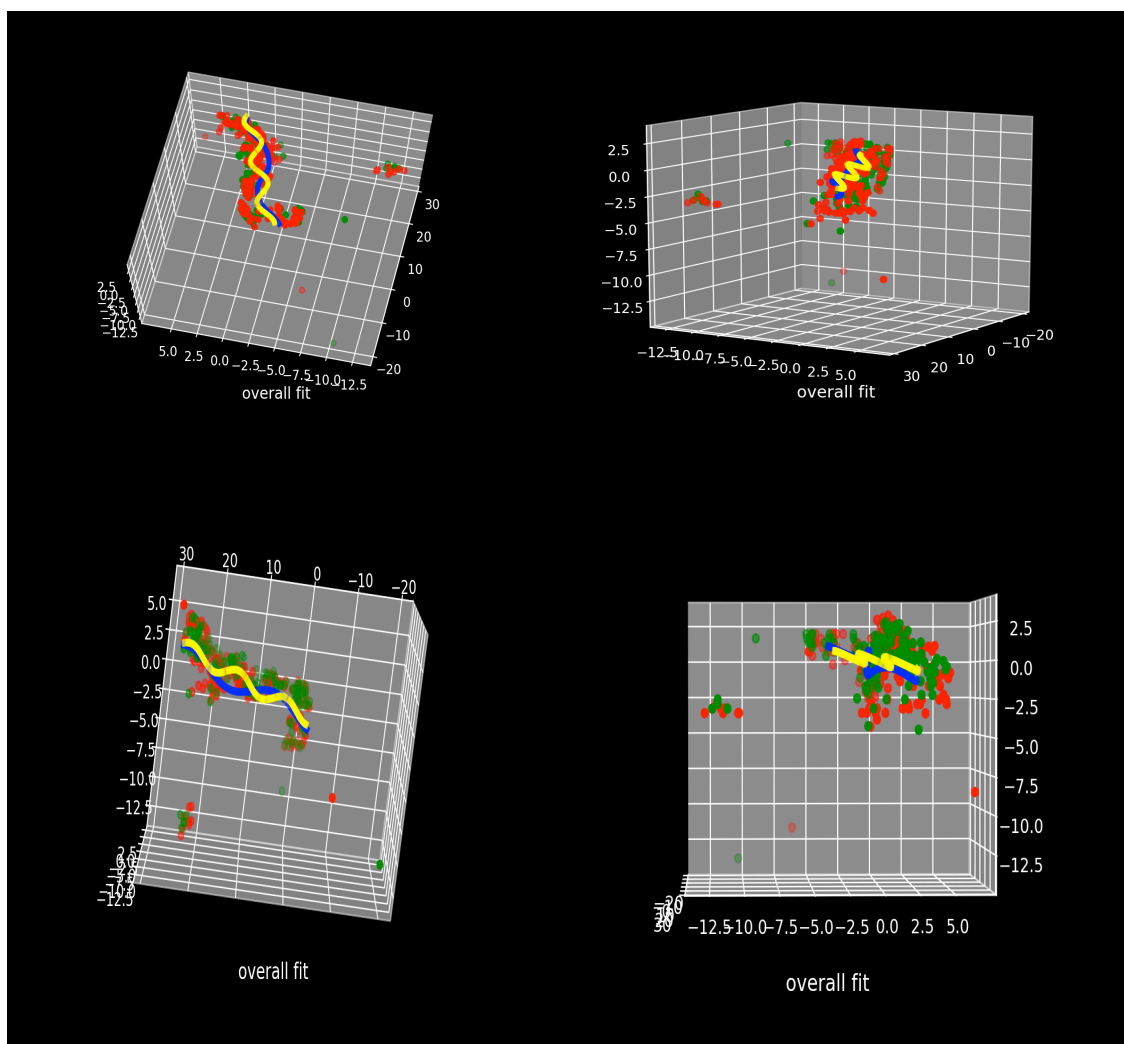
3D Fits

Results

The sample analyzed above is dnd (Channel1-blue fit), nanos (Channel2-yellow fit). The radius, frequency, phase, and pitch was measured separately for each channel in each 2D fit (to control for systematic variance in every direction). The diagonal of the covariance matrix of the parameters was used to estimate the standard error in each parameter.

Fit Parameters

	Channel 1		Channel 2	
	C1-C2	C1-C3	C1-C2	C1-C3
Radius	0.820886097	0.3561042334	0.5059933668	0.5336264958
Frequency	0.310562395	0.5362428903	0.5617256545	0.5583666489
Phase	3.857698549	3.5808463877	2.0617567875	1.9833707874
Pitch	20.60844806	11.888005138	10.810321506	10.975431152
	Channel 1 standard error		Channel 2 standard error	

	C1-C2	C1-C3	C1-C2	C1-C3
Radius	0.16246727	0.14848932	0.15910103	0.14531764
Frequency	0.01893829	0.03993911	0.03256164	0.02787106
Phase	0.17368325	0.38202156	0.30646248	0.26181347

On-going work
1. Removal of outliers before straightening, to get an accurate path through the aggregate.
2. Cluster visualization to test effectiveness of clustering
3. Data collection and analysis
4. Analysis of animal-vegetal relationships
5. Better 3D visualization

Future work
1. Estimating the Point Spread Function to determine the shape of individual RNPs.
2. Locating bends in the Bezier curve using Integer Programming, to analyze each part of the aggregate separately.
3. Analyze parts of the aggregate with and without vegetal RNPs separately.

References
1. Scikit-image, image processing in python, Image Segmentation documentation
2. OpenCV - Image Segmentation with Watershed Algorithm
3. Sarder and Nehorai, Deconvolution Methods for 3-D Fluorescence Microscopy Images
4. Lee, Curve Reconstruction from Unorganized Points