PyLattice

V0.2

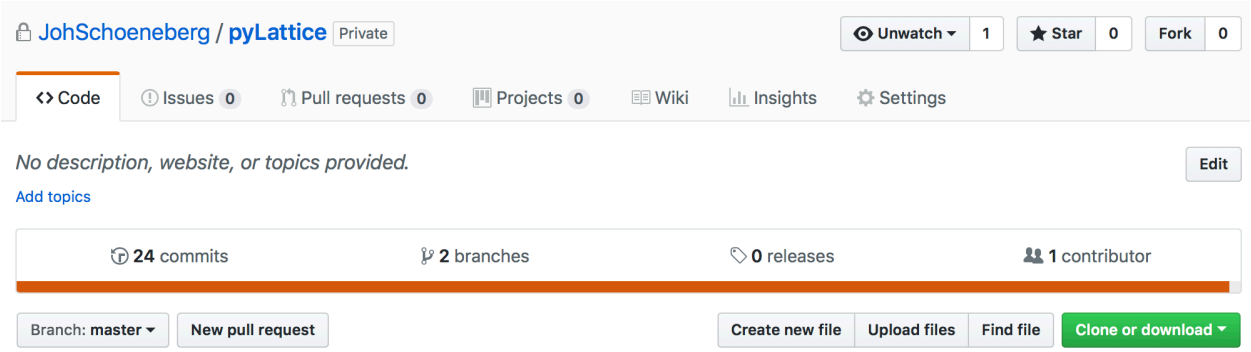
July 2018

Joh Schöneberg

User Manual

Installation

Download the git repository to your computer from https://github.com/JohSchoeneberg/pyLattice by using the green button as displayed below:



1. Installation

Requirements

- Python
- Jupyter notebook installation (e.g. Anaconda)
- A recent Matlab installation (Matlab 2017 was used for dev and testing)

Python

Some tools are written in Python and operate as Jupyter notebooks. They are located in PyLattice/src/python/ and grouped by task (see details below)

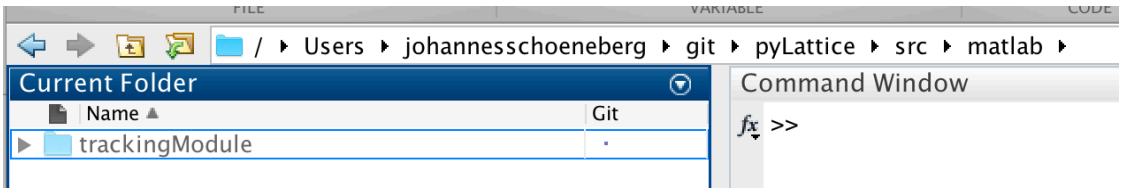
The best way to access and run the tools is through their Jupyter notebook interfaces:

- Start a Jupyter notebook server and navigate to the scr/python folder

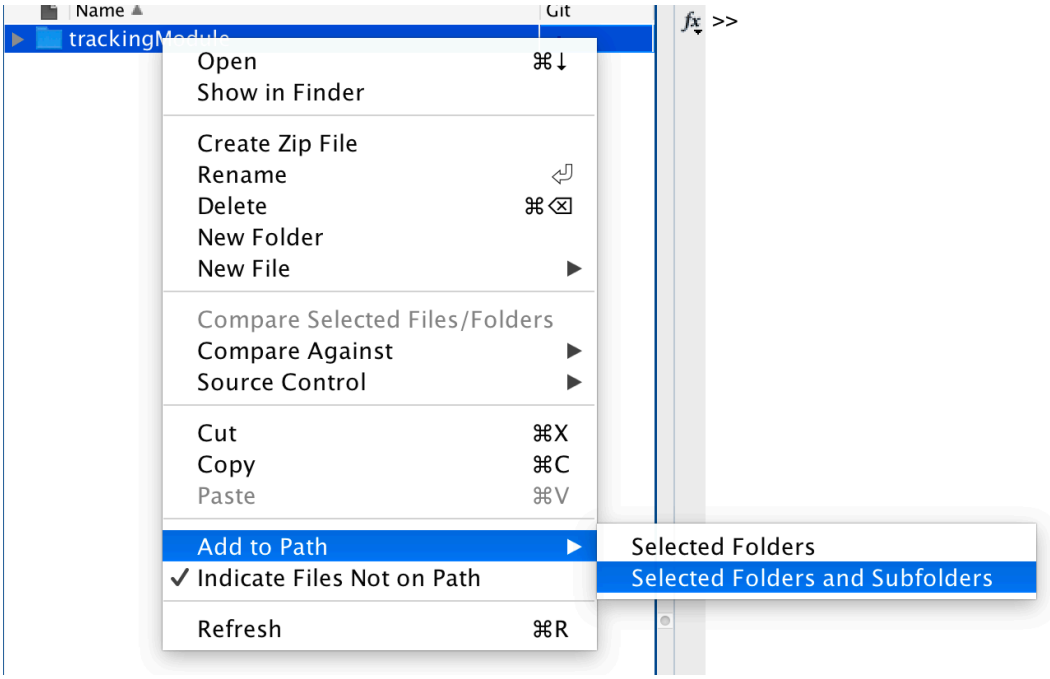
Matlab

The 3D particle detection and tracking modules are written in Matlab. Install them by:

- Open Matlab.
- Navigate to pyLattice/src/matlab/trackingModule.



- Add it to your path by rightclick -> Add to Path -> Selected Folders and Subfolders :



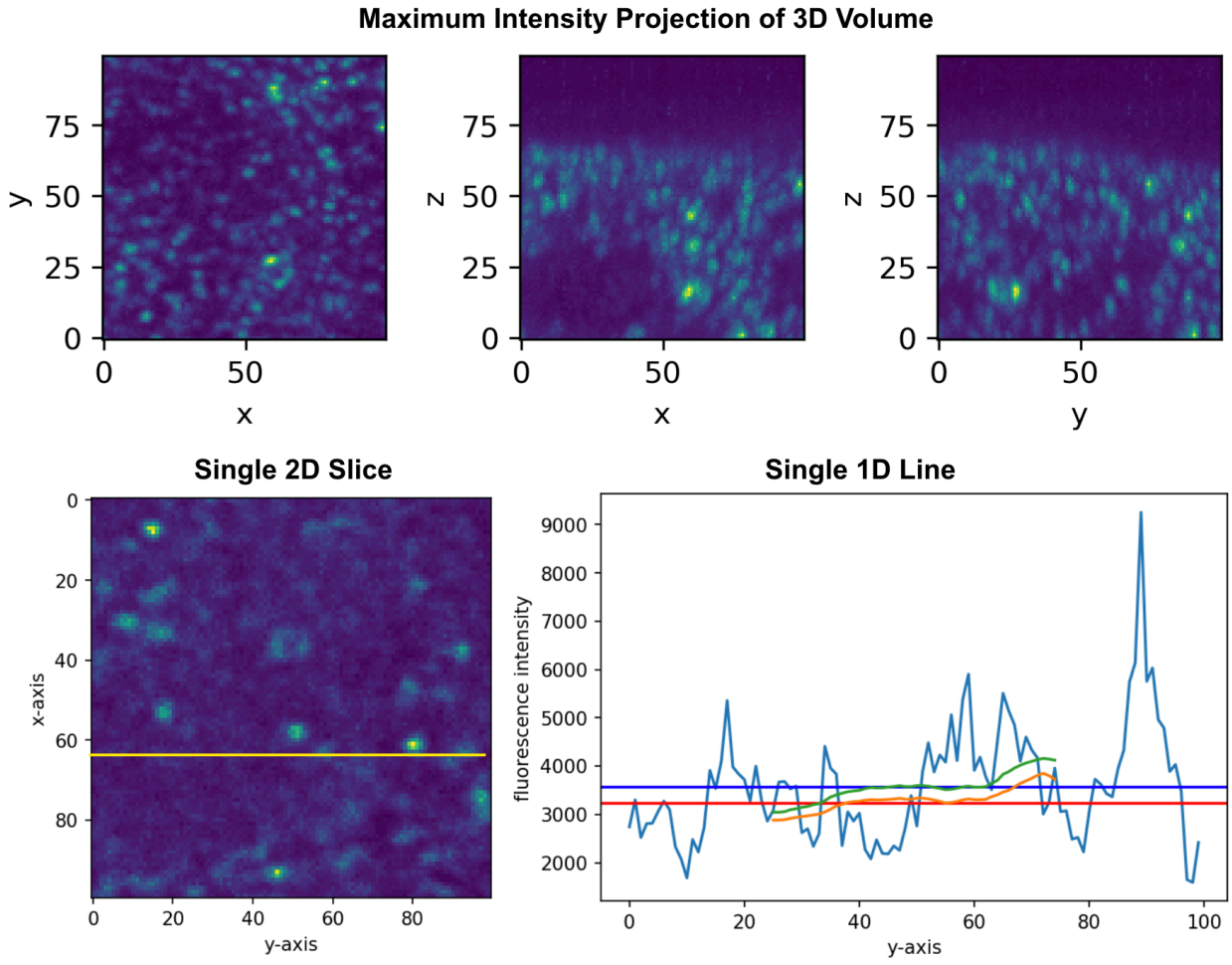
Lattice Light-Sheet Data Visualization and Preprocessing

latticeFrame_showFrame.ipynb

One of the first things when working with high resolution data is to visually look at the data. The notebook 'latticeFrame_showFrame.ipynb' provides the basic functionality to do that:

- Maximum intensity projection
- View individual 2D slices of the data
- View 1D lines of the data

For 3D volumetric renderings of the data, I recommend to use ChimeraX.

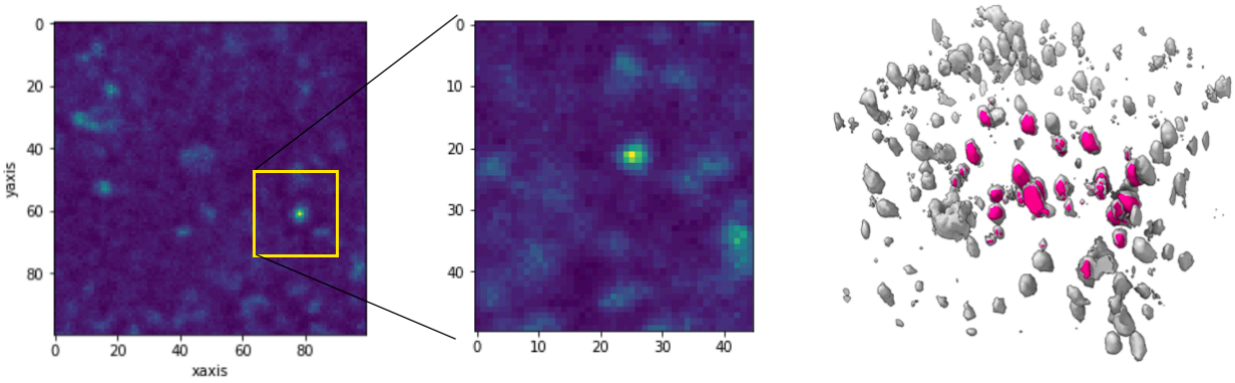


latticeMovie_cropRegion.ipynb

LLSM frames in respectively LLSM movies can reach substantial file sizes, cropping the movie can lead to faster processing times. Also, sometimes a certain region in the imaged 3D volume contains most of the relevant information. Or one simply wants to zoom in on a region.

This cropping tool allows to crop regions of interest out of large 3D LLSM movies:

Cropping a smaller 3D volume out of a larger LLSM frame

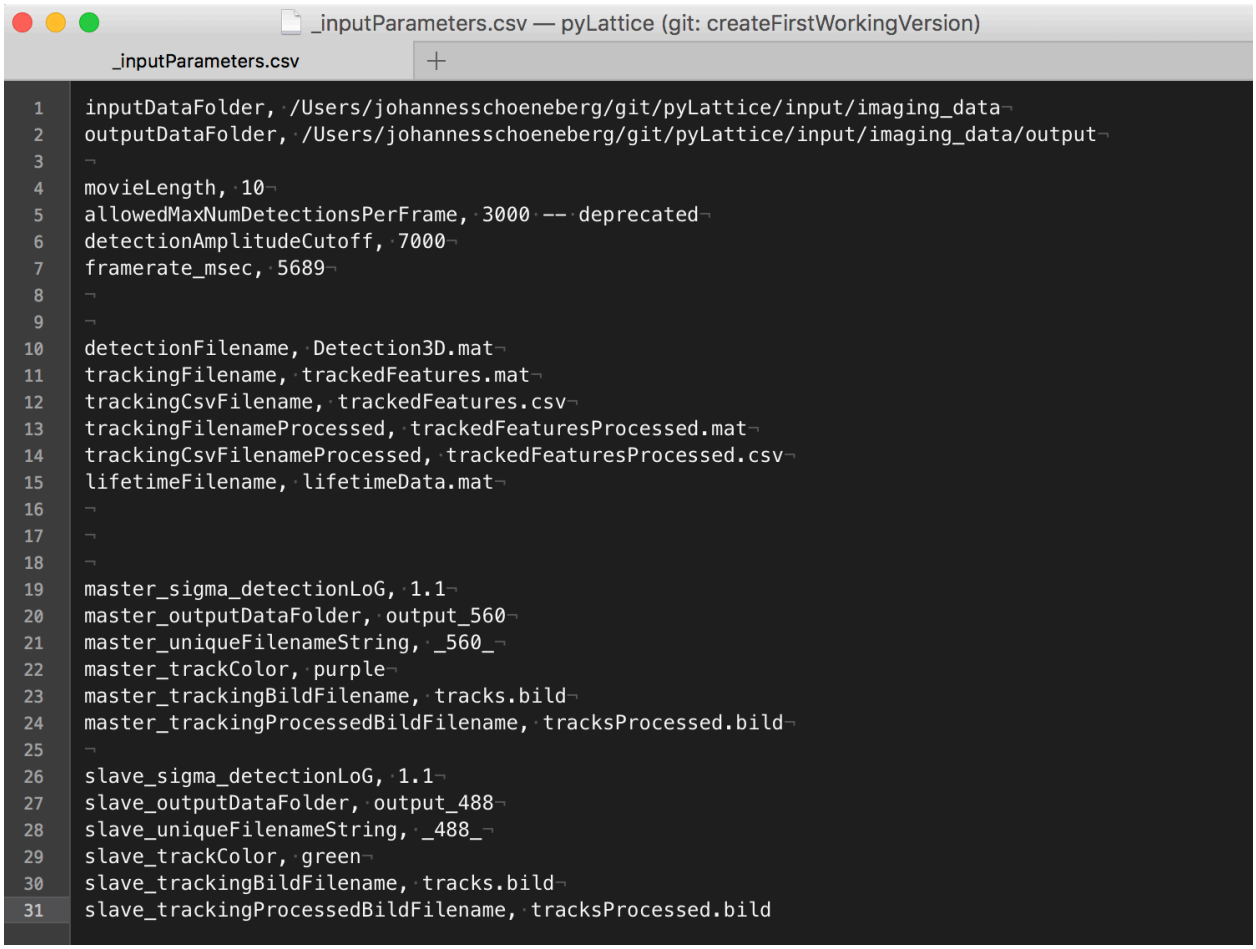


3D Particle Detection & Tracking

2. Adjust Input Parameters

- Navigate to pyLattice/input/imaging_data
- Open _inputParameters.csv

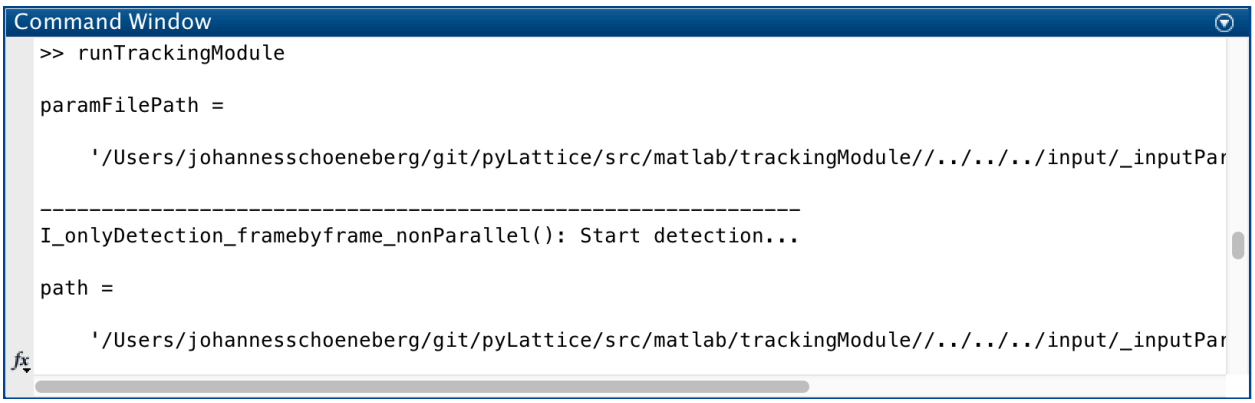
- Change the inputDataFolder and outputDataFolder variables such that they match your filesystem (top two lines)



3. Run the Matlab Tracking Module

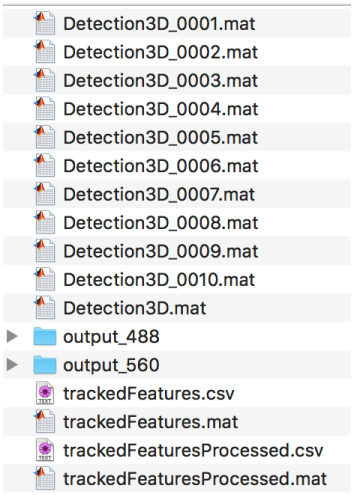
- Open the /src/matlab/trackingModule folder and open ‘runTrackingModule.m’
- Hit ‘Run’.

The Command window should now display the progress doing the tracking:



After a little while (few minutes) the tracking module should terminate.
The results of the tracking can be found in the output folder that you specified in step 1.

There should now be the following files in your output folde:



The trackedFeaturesProcessed.csv is the final output of the tracking module that we will use for further processing.

Detected Puncta

The first step in particle tracking detects the particles (puncta).

1. Visualize Detected Puncta

detectedPuncta_oneFrame_plot.ipynb

This Jupyter notebook allows you to visualize the detected puncta from an individual frame.

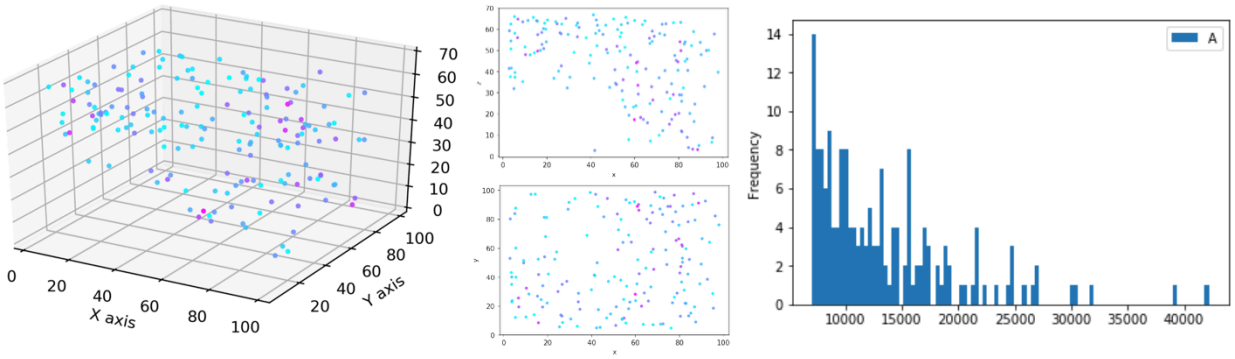


Fig 1. Plot the detected puncta with the 'detectedPuncta_oneFrame_plot.ipynb' notebook.

It provides code and functions to plot the detected puncta in 3D, 2D and has functionality for puncta selection based on intensity.

In addition, the notebook allows to convert the detected puncta into a *.bild file that can be visualized in the open source tool ChimeraX [1]:

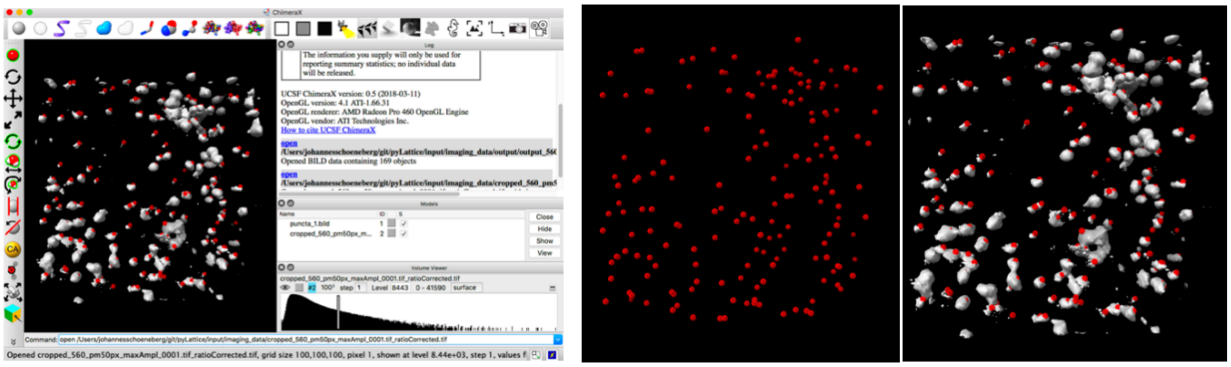
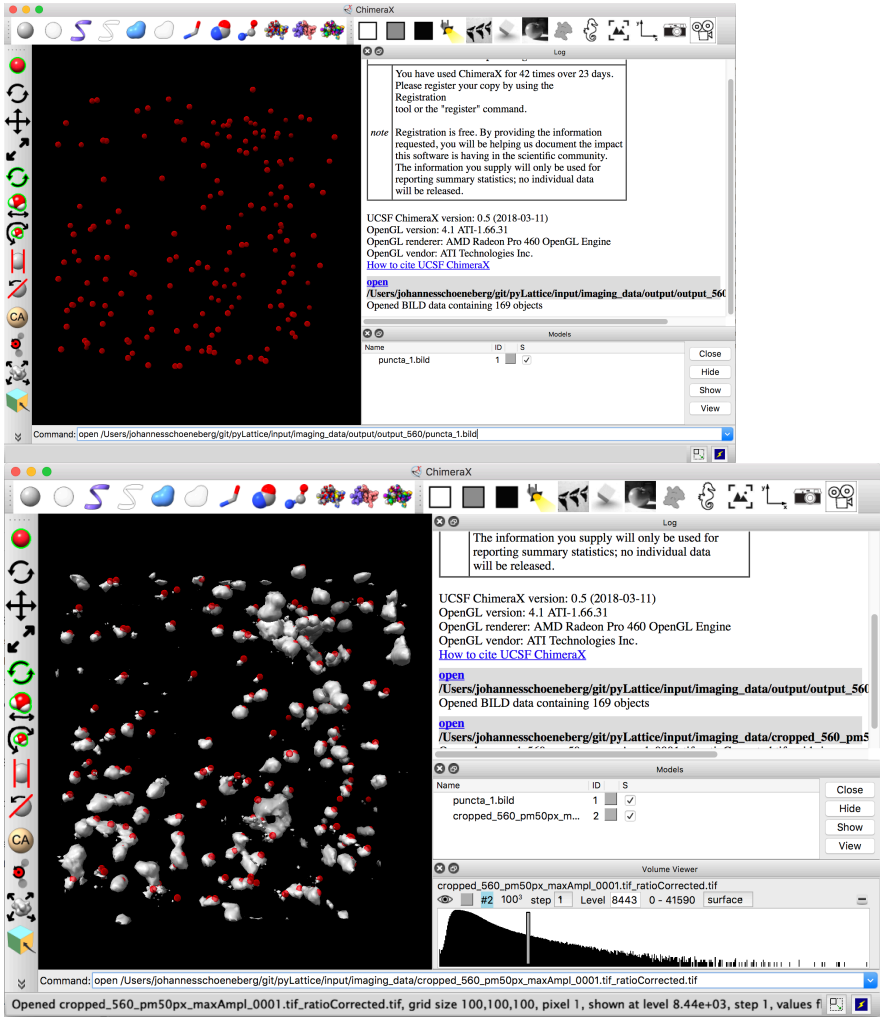
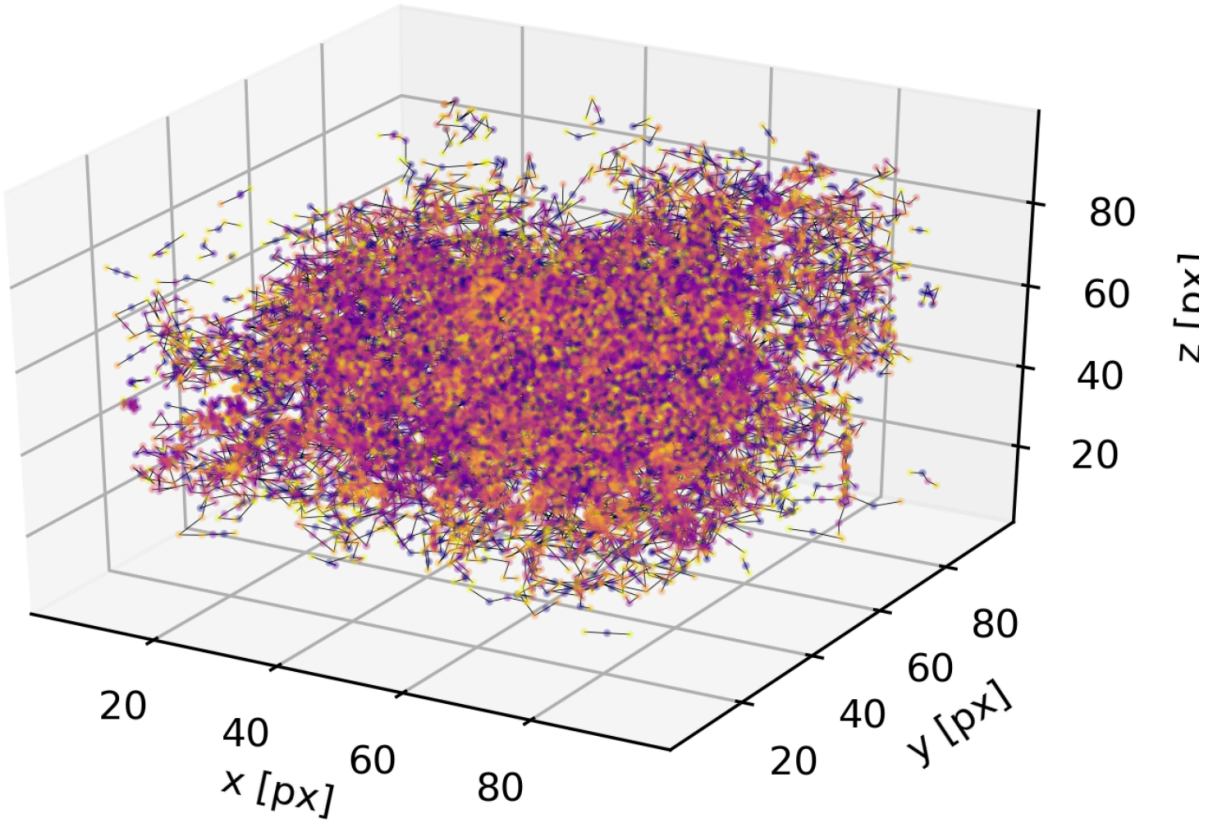


Fig 2. Display detected puncta using ChimeraX and overlay them with the raw data. Shown is ChimeraX's UI (left), detected puncta (middle) and overlay (right).



Run: detectedTracksProcessed_plotAll_3D.ipynb, the result should look like this:



References:

[1] UCSF ChimeraX: Meeting Modern Challenges in Visualization and Analysis. Goddard TD, Huang CC, Meng EC, Pettersen EF, Couch GS, Morris JH, Ferrin TE. *Protein Sci.* 2018 Jan;27(1):14-25. doi: 10.1002/pro.3235.