PyLattice

V1.0

2018-04-26

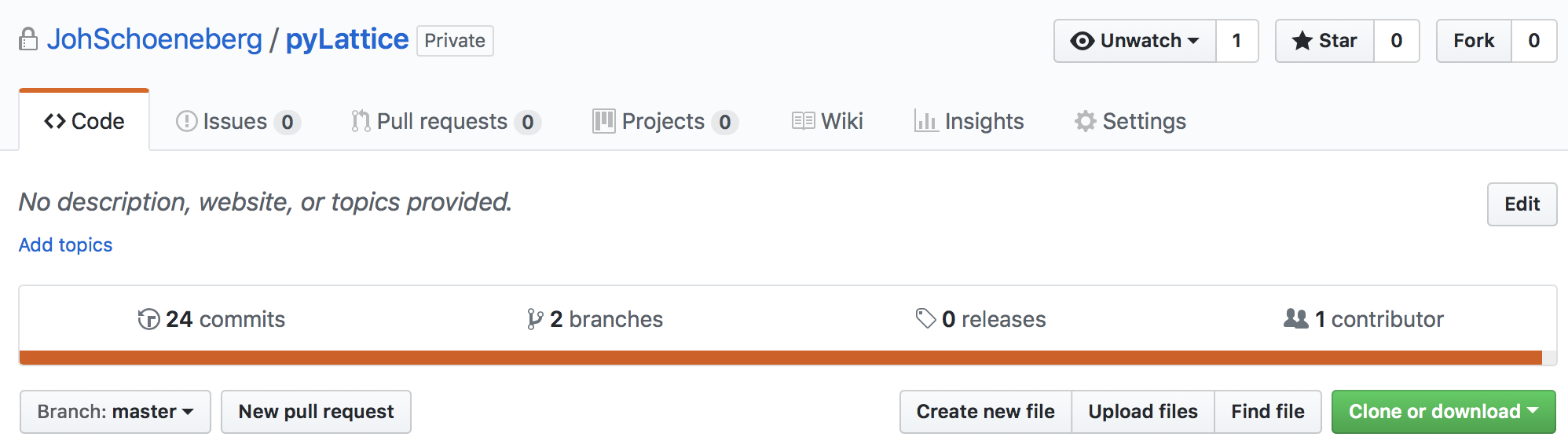
Joh Schöneberg

User Manual

# 3D Particle Detection & Tracking

## Installation

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

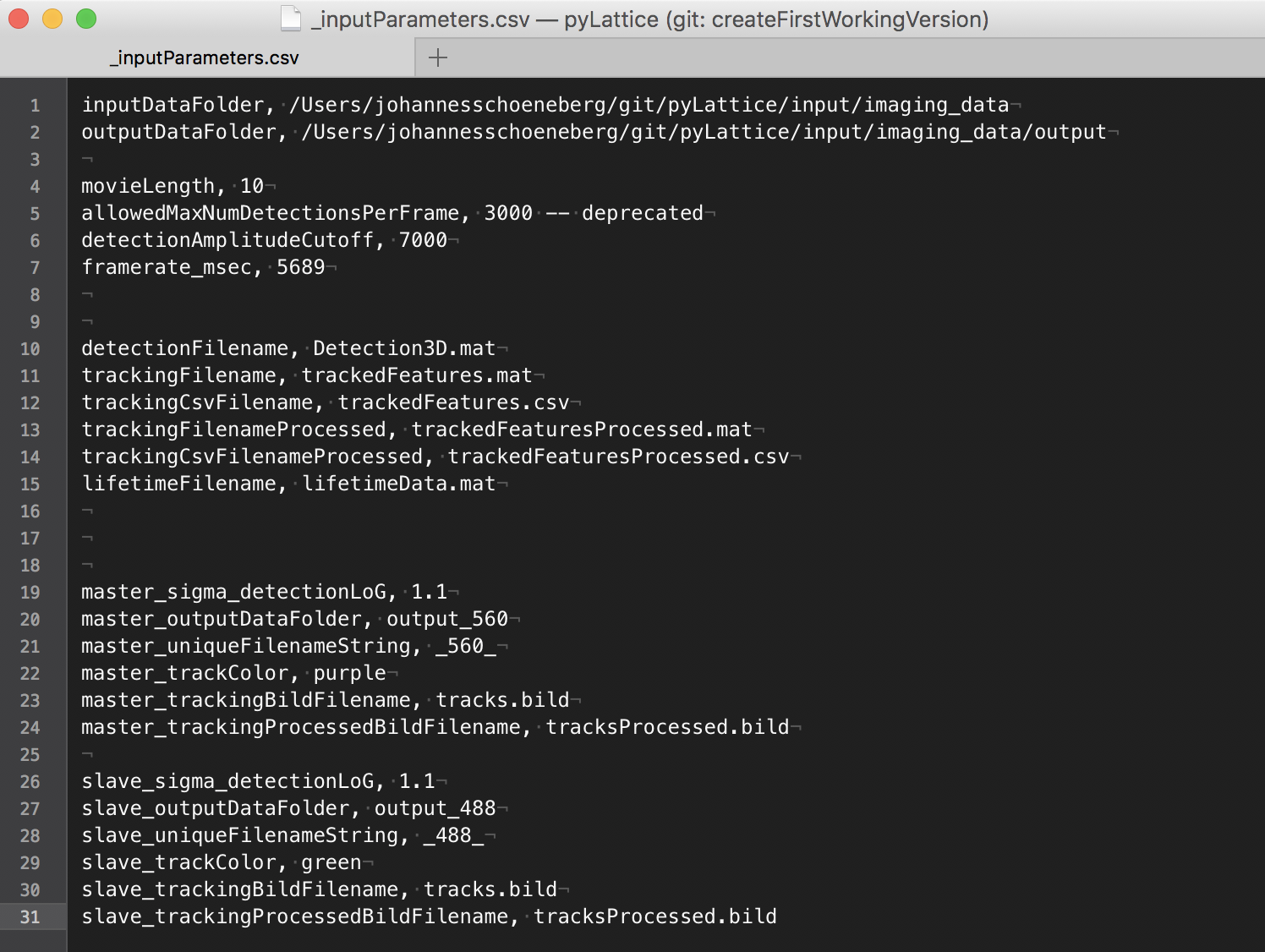


Requirements:

* Jupyter notebook installation (e.g. Anaconda)
* For tracking, a recent Matlab installation is required (Matlab 2017 was used for testing)

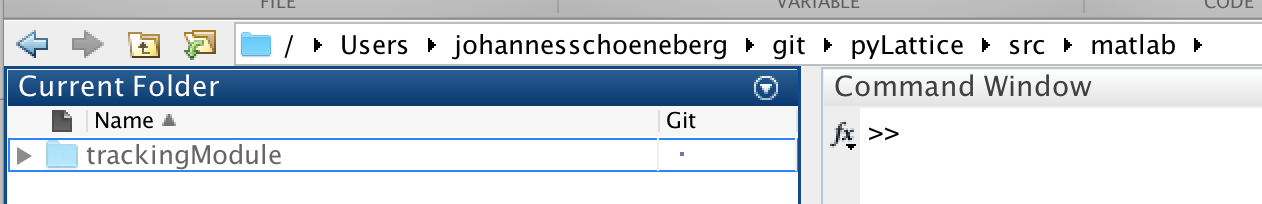
## Adjust Input Parameters

* Navigate to pyLattice/input/imaging\_data
* Open \_inputParameters.csv
* Change the inputDataFolder and outputDataFolder variables such that they match your filesystem (top two lines)

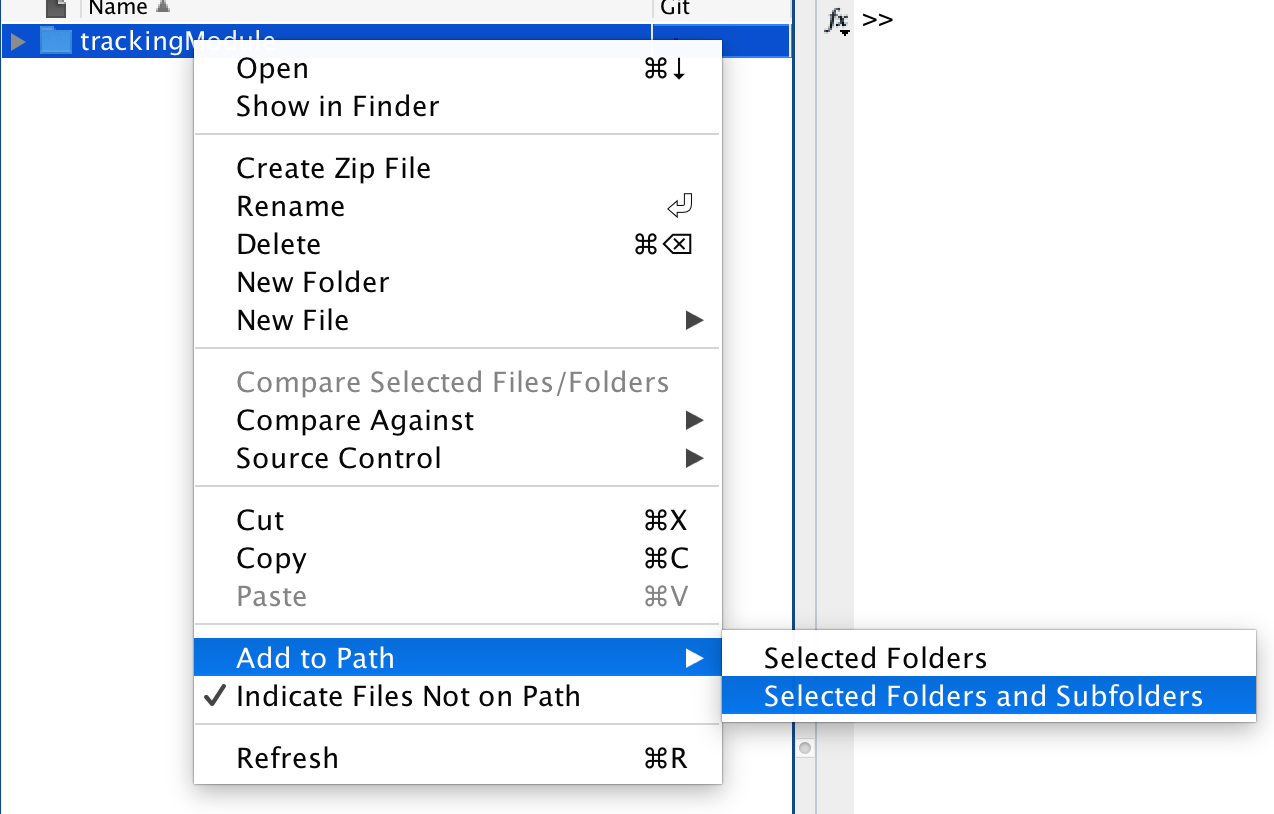


## Run the Matlab Tracking Module

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

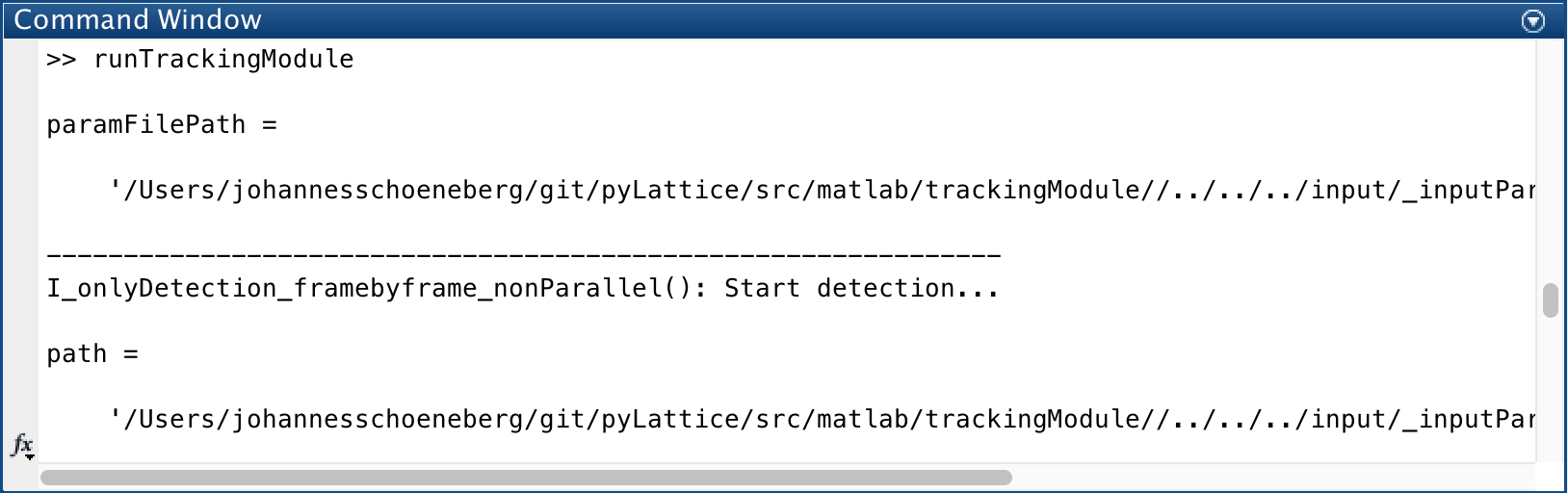


* Add it to your path by rightclick -> Add to Path -> Selected Folders and Subfolders :



* Open the trackingModule folder and open ‘runTrackingModule.m’
* Hit ‘Run’.

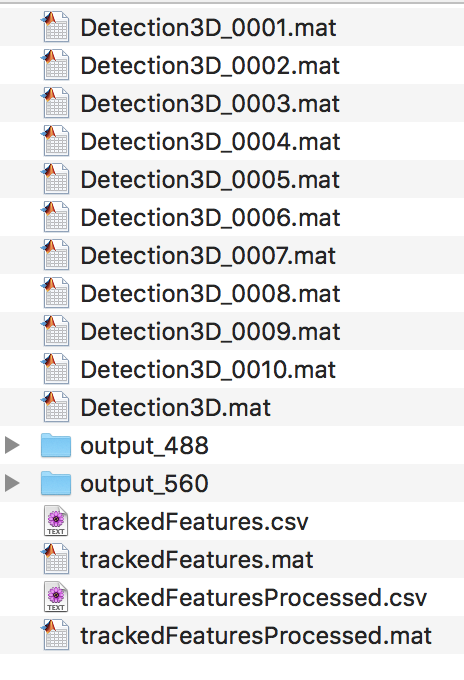
The Command window should now display the progress doing the tracking:



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:



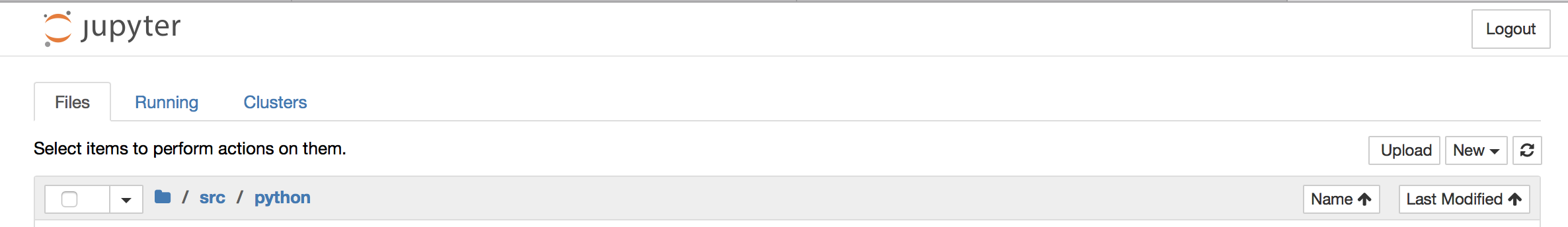
The trackedFeaturesProcessed.csv is the final output of the tracking module that we will use for further processing.

# Detected Puncta

All data analysis tools are written in Python and are located in PyLattice/src/python/

The best way to access them is through their Jupyter notebook interfaces:

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



## Visualize Detected Puncta



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

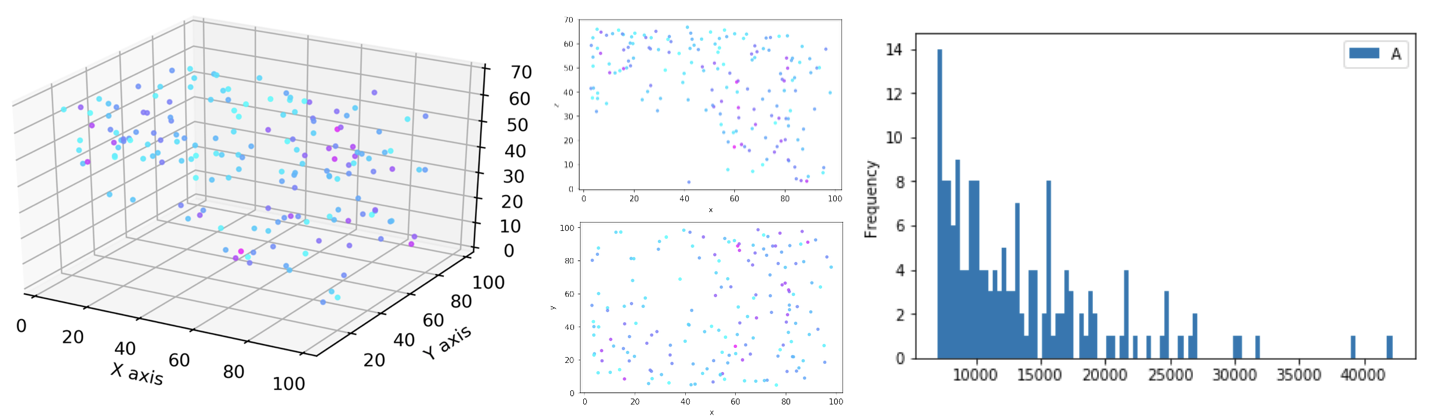


Fig 1. Plot the detected puncta with the ‘detectedPuncta\_oneFrame\_plot.ipynb’ notebook.

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](https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/bild.html) file that can be visualized in the open source tool [ChimeraX](https://www.rbvi.ucsf.edu/chimerax/) [1]:

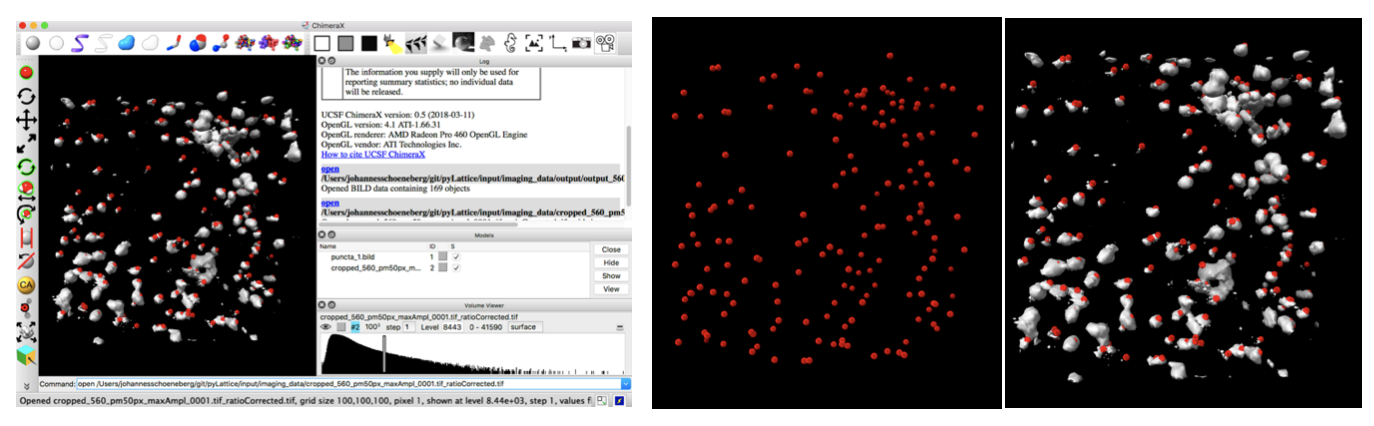
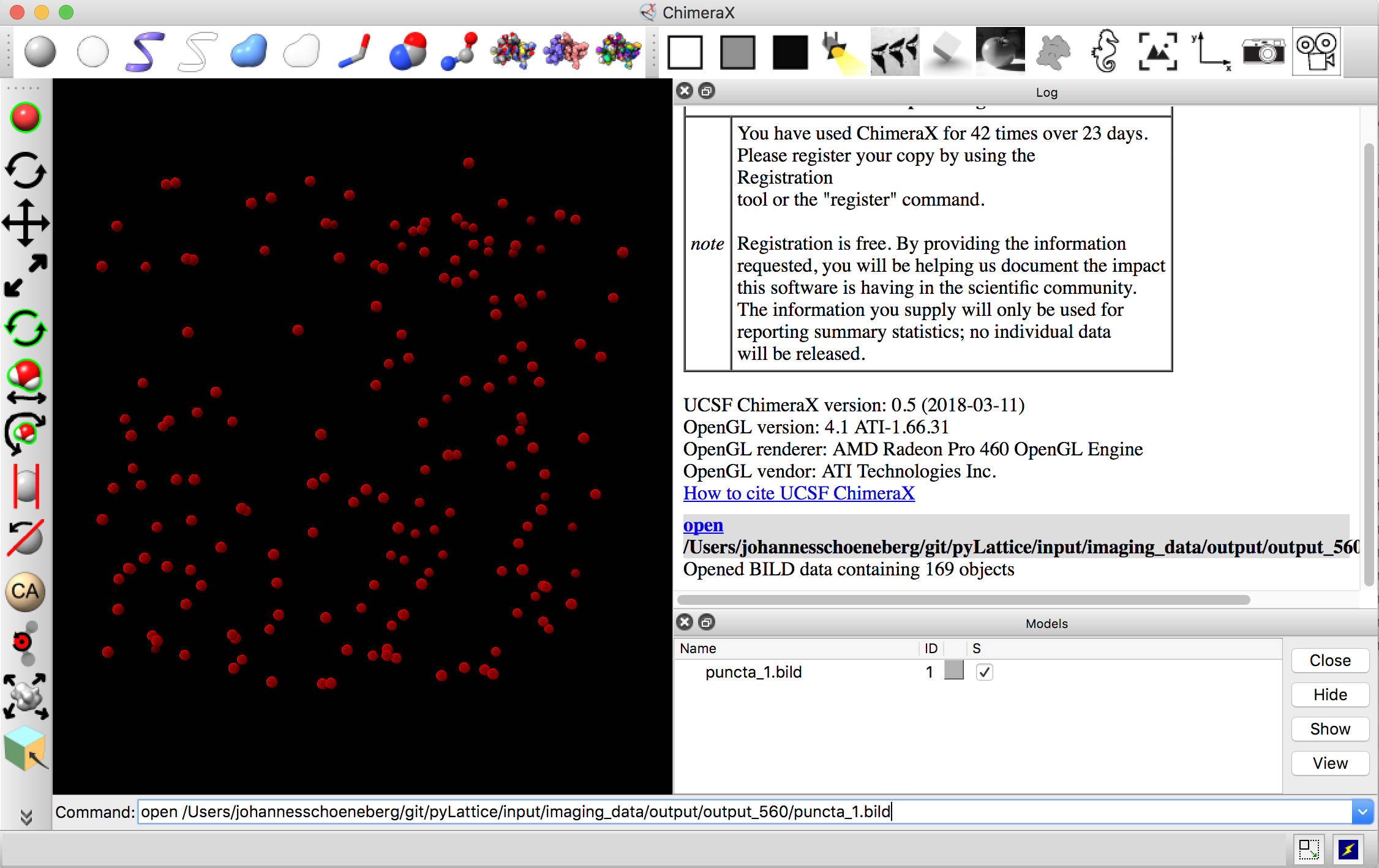
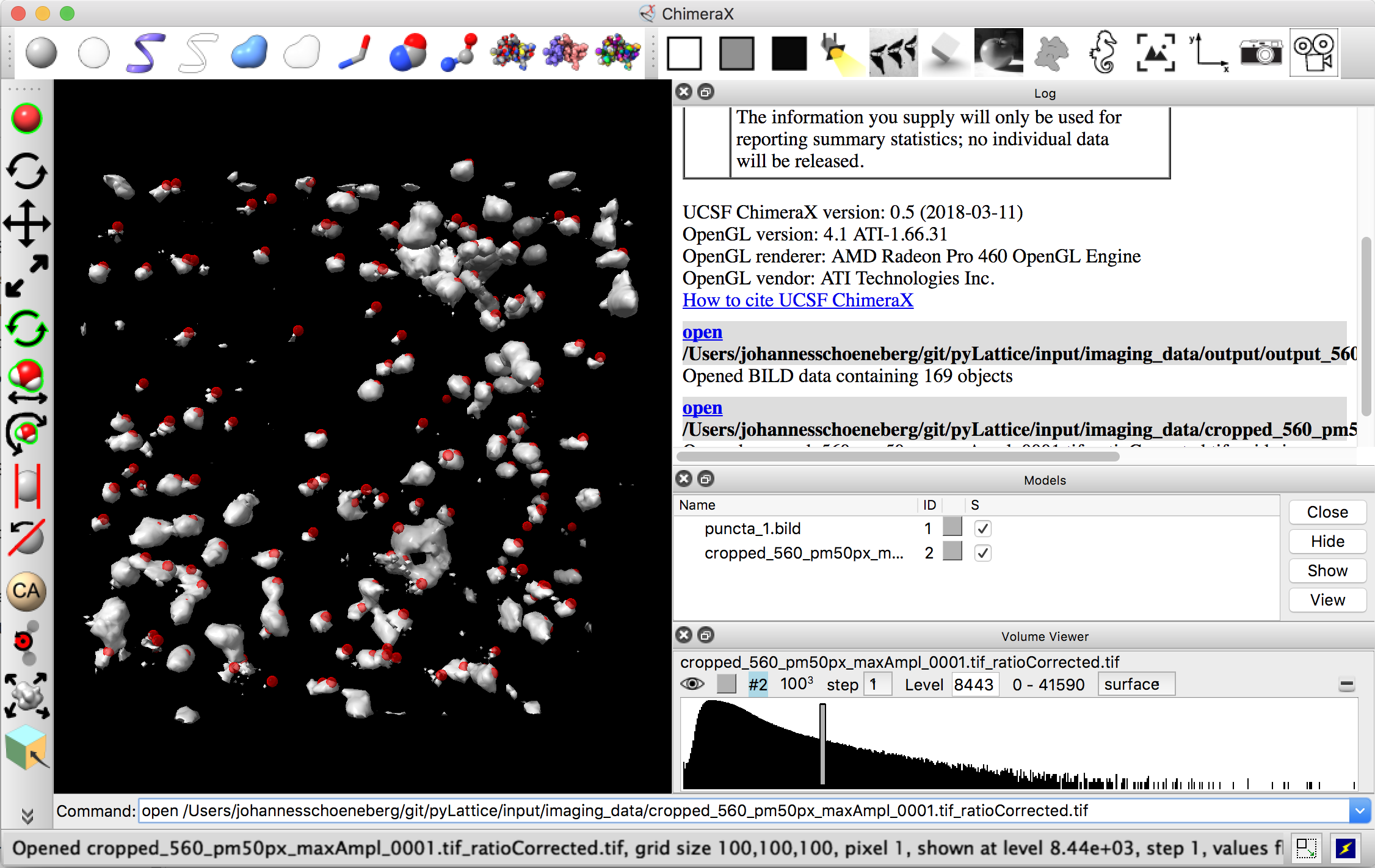
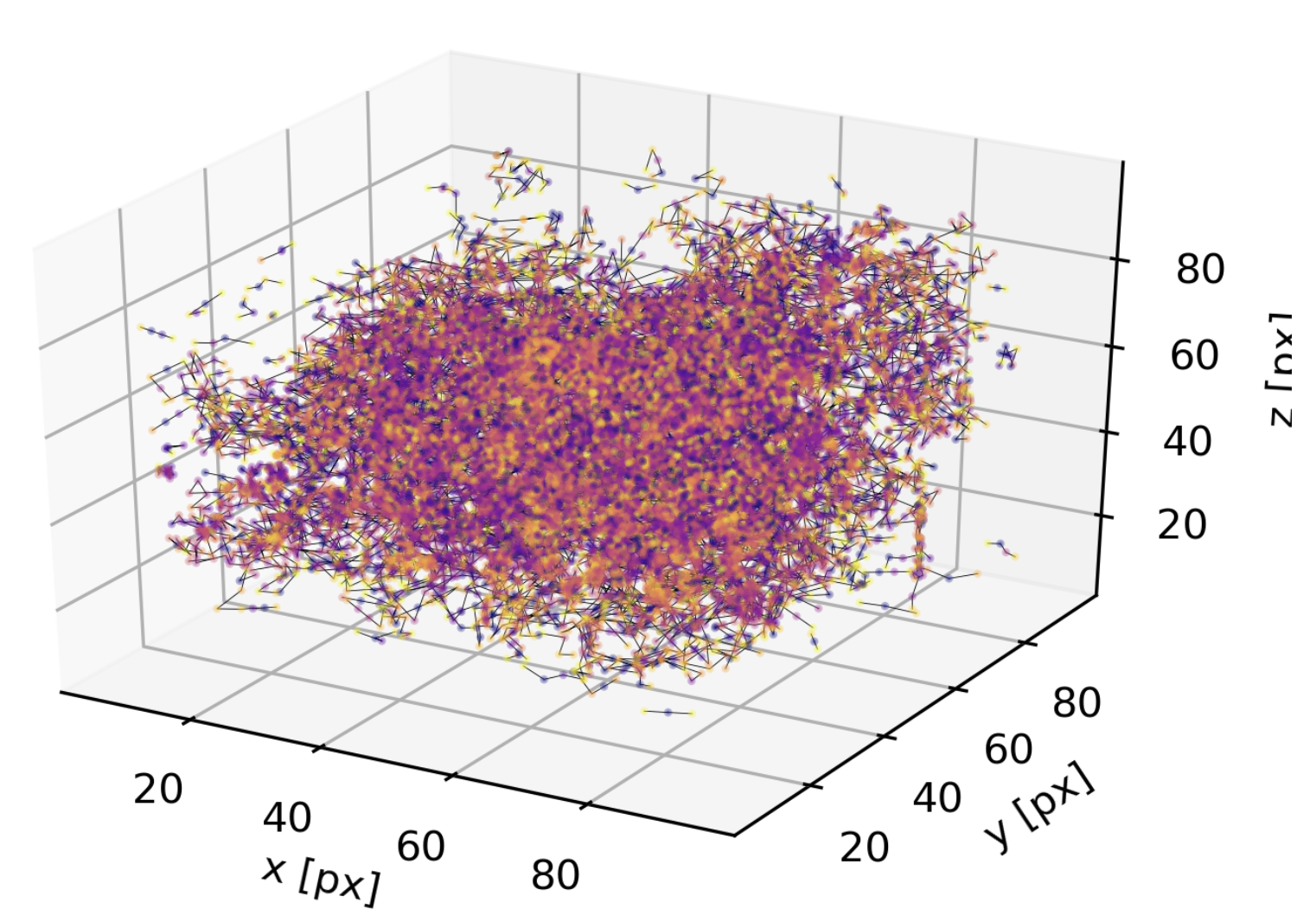


Fig 2. Display detected puncta using ChimeraX and overlay them with the raw data. Shown is ChimeraX’s UI (left), detected puncta (middle) and overlay (right).





Run: detectedTracksProcessed\_plotAll\_3D.ipynb, the result should look like this:



References:

[1] [UCSF ChimeraX: Meeting Modern Challenges in Visualization and Analysis.](http://www.ncbi.nlm.nih.gov/pubmed/28710774) 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.