Tutorial: Protofilament Refinement

Garrett Debs
gedebs37@gmail.com

Charles Sindelar
charles.sindelar@yale.edu

June 22, 2020

Tl;dr- The quick version

pf_init_project: Initialize project parameters

pf_preprocess: Flip microtubule polarity and remove symmetry (if necessary)

pf_smooth (*Recommended*): Smooth microtubule coordinates

pf_wedge_masks: Prepare subtraction volumes

pf_microtubule_subtract: Generate protofilament particles

RELION Refine3D- Local refinements of protofilament particles. Processing can end here, all additional steps are optional and for trying to push analysis a little further.

---

pf_protofilament_subtract: Improved subtraction using protofilament refined coordinates

RELION Refine3D- Second local refinement step, with less background signal

pf_focused_classification: Subtract all but a region of interest for classification

RELION Class3D: Classification of your region of interest

pf_plot_distortions: Plot angle between adjacent protofilaments, to further analyze the microtubule structure

Contents

1 Introduction

The purpose of protofilament refinement is to further refine microtubule structures generated using cryo-Electron Microscopy. We have shown that under most circumstances, microtubules are not perfectly cylindrical as has previously been assumed. Therefore, in order to obtain the highest resolution structures possible, we want to refine only a portion of the microtubule wall at a time. In order to do this, we will refine each protofilament of a microtubule particle separately. This allows us to better represent the structure of the microtubule as a whole.

Currently this package will take you from a refined microtubule structure to a protofilament refined structure. Additionally you can perform focused classification of particles on your microtubule. For instance, you should be able to find sparsely decorating proteins, such as kinesin motors, bound to the surface of the microtubule. Lastly, we've provided a quick way to analyze the distortions within a microtubule, by plotting the angle between protofilaments following protofilament refinement. We hope that these tools will prove useful to anyone performing high-resolution structural analysis of microtubules.

It's worth noting that this may or may not be the best method to account for microtubule distortions during cryo-EM structure refinement. We highly encourage anyone to improve upon the method whether that includes building upon these techniques, or exploring their own avenues (utilizing RELION's multi-body refinement could be an interesting avenue of research).

The instructions in this tutorial assume that you have already refined your microtubule structure and have a functional version of RELION. Please note that this package does NOT directly refine any structures. All reconstructions, refinements, and classifications are done in RELION.

2 Installation

We find the easiest and most non-intrusive way to install this package is using anaconda (download link). This allows us to create a virtual environment with all the necessary python packages and without changing with your local environment. More importantly, it means that installation can be installed with root permissions, so any user can install the program. To install and run the package, run the following commands.

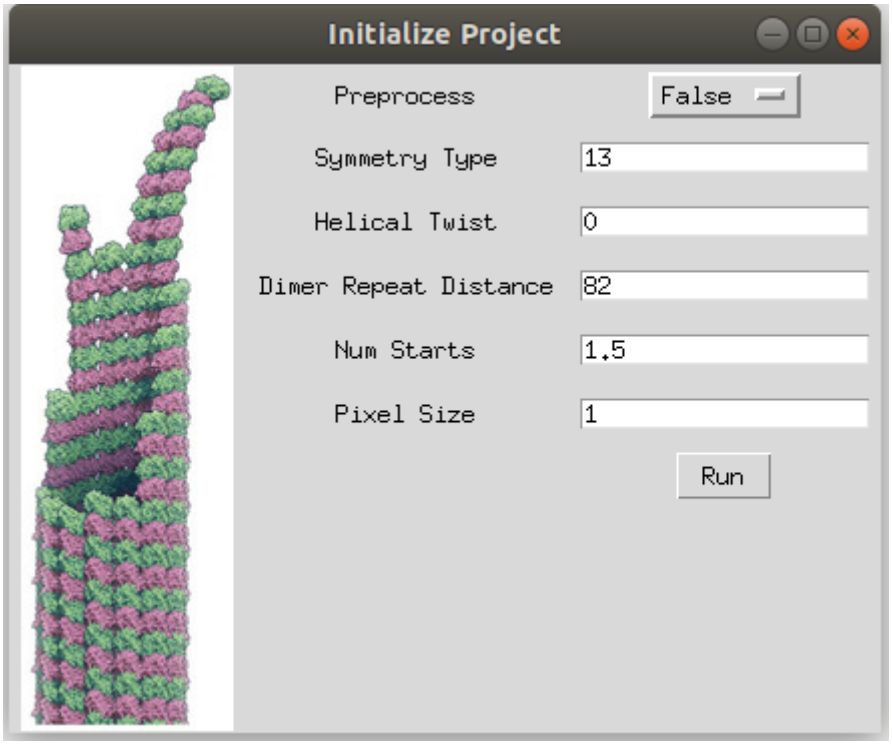
git clone https://gitlab.com/gedebs371/protofilament-refinement.git
cd protofilament-refinement
conda env create -f environment.yml
conda activate pf_refinement
python setup.py install

Assuming there were no errors during installation, you can begin using the program by activating the conda environment (i.e. **conda activate pf_refinement**) whenever you open a new terminal.

3 Initializing the project

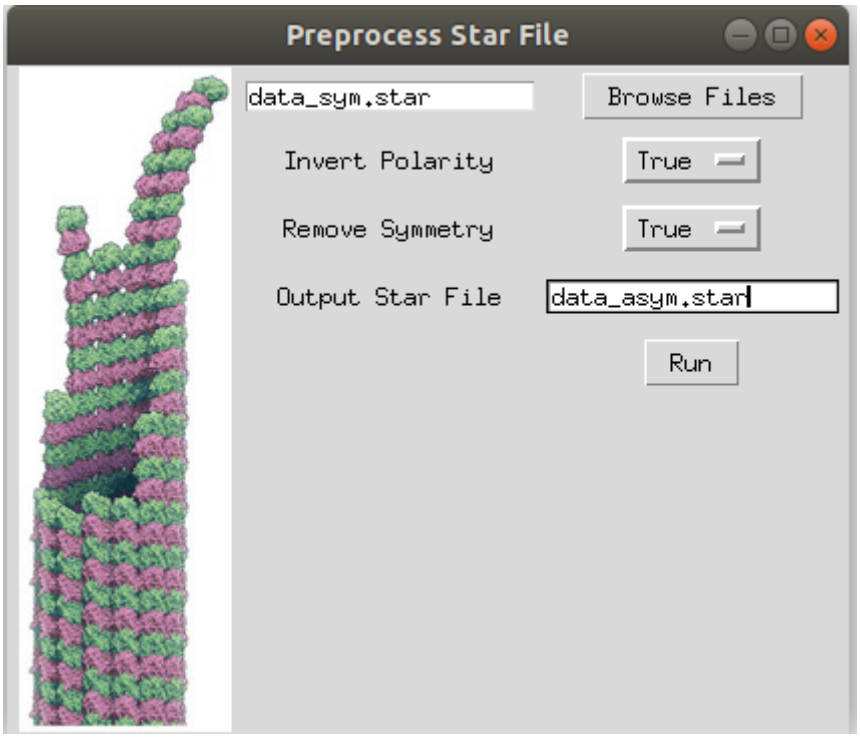
Before running protofilament refinement, we first need to set up our working directory, define a few parameters, and if necessary, preprocess the incoming data. Note that you need a new working directory for each microtubule symmetry type you analyze (i.e. different directories for 13 and 14 protofilament microtubules). The preprocessing step is performed to ensure that your microtubule volume and star files are in the orientation and format that we are expecting. First, make sure that the plus end of the microtubule is aligned with the positive Z axis. You can check this by comparing the polarity of your microtubule volume relative to one of our published models (EMDB-21924), or by utilizing the XYZ-axes.bild generated in chimera. Second, make sure that your star file contains only the asymmetrical microtubule coordinates (i.e. C1 symmetry). To initialize the project run the following command.

pf_init_project



Here, the symmetry type refers to the number of protofilaments, and the helical twist is assuming C1 microtubule symmetry (i.e. about 0 for a 13 protofilament microtubule). Additionally, the pixel size refers to the pixel size of the extracted particles, which may differ

from your micrograph pixel size if your data is binned. If you need to preprocess the data, select True in the preprocess tab and a new window will open.



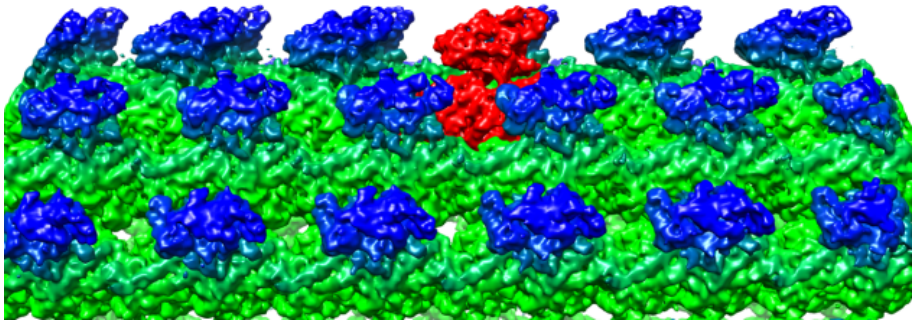
Input the star file corresponding to your best microtubule refinement, select True for any preprocessing methods you need to do, and the name of your desired output star file. The resulting star file will be what you use for the next steps of protofilament refinement. Additionally, if you plan on skipping the coordinate smoothing step, you should reconstruct the resulting star file for use in the upcoming subtraction steps.

4 Smoothing Microtubule Coordinates

NOT YET IMPLEMENTED. While step is recommended to achieve the highest possible resolution, it is also the finicky step and can be skipped if strapped for time. However, this step should be performed if you would like to perform distortion analysis.

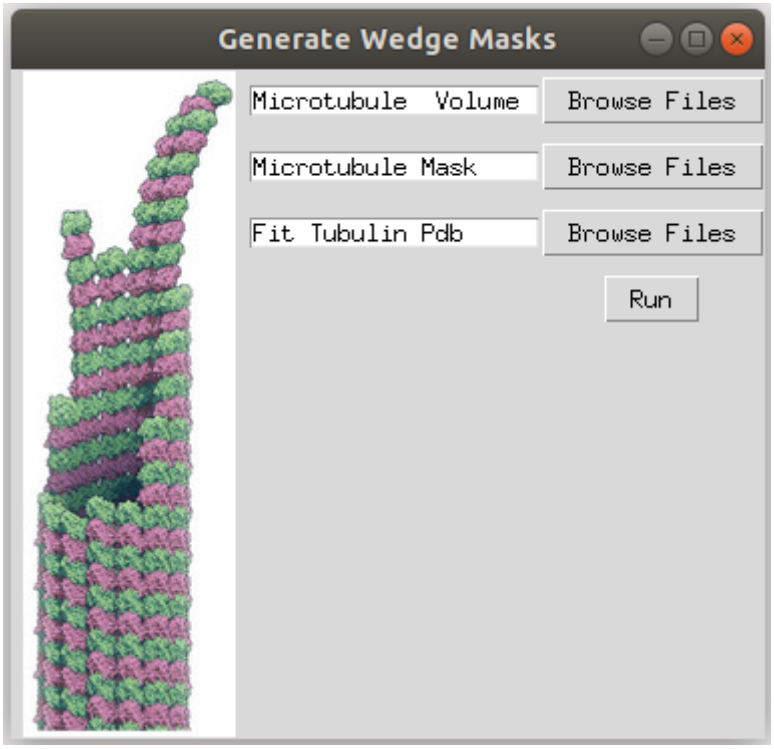
5 Wedge Masking

Now we're getting into the fun stuff! Our next step is generating a series of subtraction volumes, one for each protofilament. We do this by generate a wedge mask centered on each protofilament. To start, fit a tubulin PDB model in the center of the protofilament adjacent to the seam (as demonstrated by the red subunit below).



Save the fit PDB model relative to your microtubule volume. This pdb is used as a point of reference for where to center the wedge masks. Note, make sure you set the pixel size and center your microtubule volume prior to fitting the PDB. Now you're ready to generate your subtraction volumes by running the following command.

pf_patch_masks



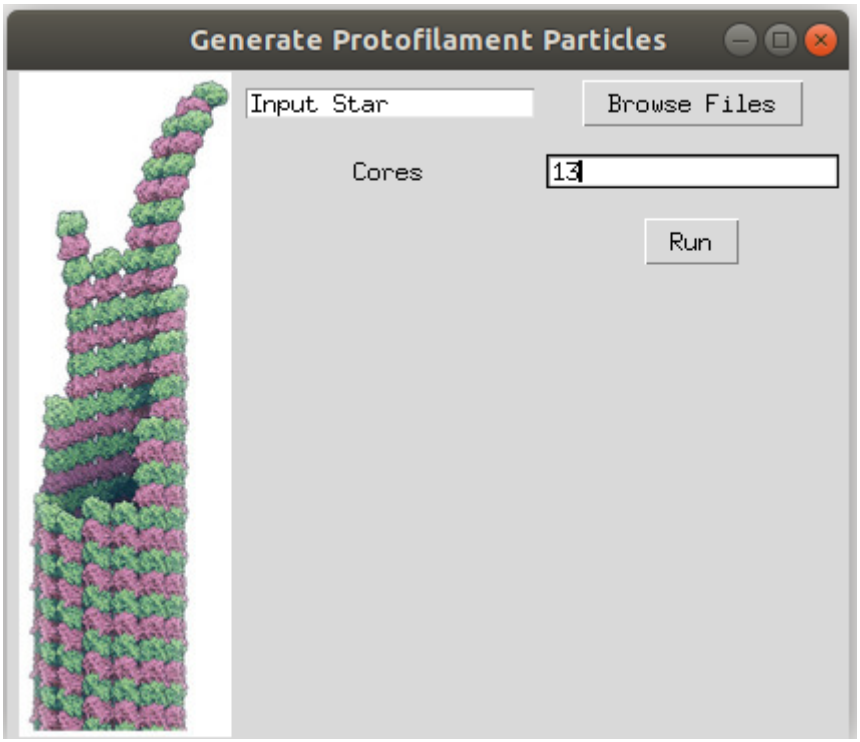
Enter the location of your microtubule volume, a microtubule mask, and the fit PDB files. Depending on your processing speed, this step should take anywhere from 1-10 minutes. The resulting subtraction volumes with be saved as *pf_masked.mrc* in a corresponding *pfn* directory, where *n* represent which protofilament has been removed. Open a couple of these volumes and make sure the protofilaments were removed correctly. There should not be any residual density where the removed protofilaments are. If there is a small amount of density on opposite sides of the protofilament at the plus and minus ends of the microtubule, the sign of helical twist may need to be flipped. This can be done by rerunning

pf_init_project. If something is still off, consider rerefining the microtubule coordinates using C1 symmetry in RELION while searching for helical parameters.

6 Microtubule Subtraction

The microtubule subtraction step results in a series of protofilament particles, and is carried out by RELION. However, because we're subtracting multiple volumes, there are N different subtractions (with N being the number of protofilaments). To perform the subtractions, run the following command.

pf_microtubule_subtract



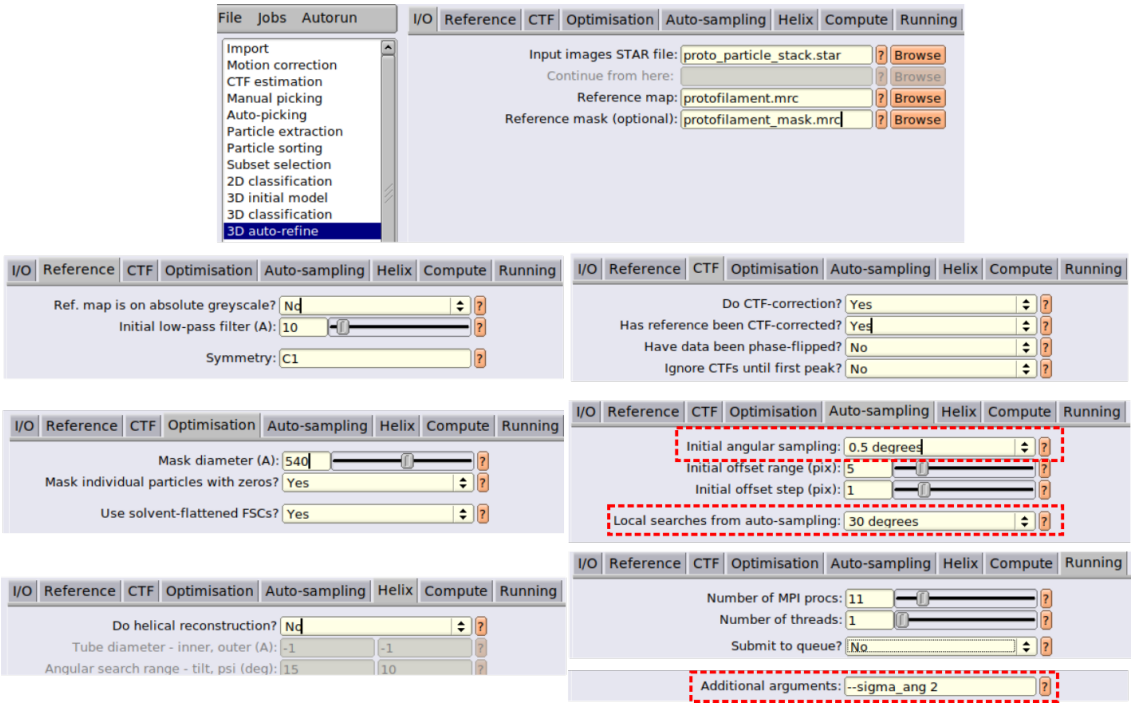
Here your star file will either be the Extracted particles.star file following smoothing, or the output starfile from preprocessing. Additionally, the cores will be the number of parallel subtractions you want to perform. I highly recommend running the subtractions in parallel, or this step will take a long time. You will need roughly 1 core and 10GB of memory per subtraction assuming an 420x420x420 volume size. I can run all subtractions in parallel for a 12 protofilament microtubule on our 20 core, 120GB memory node, but cannot run all the subtractions at the same time for a 14 protofilament microtubule for the same node. Unfortunately the software isn't sophisticated enough to estimate your exact memory requirements, so if you get strange errors when running subtractions in parallel, try running only 1 or 2 subtractions at a time to troubleshoot the problem.

Following microtubule subtraction, a new star file called *proto_particle_stack.star* will be

written. This contains particle information on each protofilament in the dataset. Reconstruct this star file and ensure that the resulting structures is a single protofilament. Additionally, you should use this map to make a protofilament mask within RELION, which will be used for refinement. I recommend lowpass filtering the map to 20Å when making the mask.

7 Protofilament Refinement: Round 1

This step is performed entirely in RELION, however, I'll share some screen shots on how I recommend you perform refinement. Because microtubule subtraction reduces the overall signal, an the resulting protofilament particles have a fairly low signal to noise ratio, I recommend performing local alignments only. An example of the parameters I usually use are shown below.

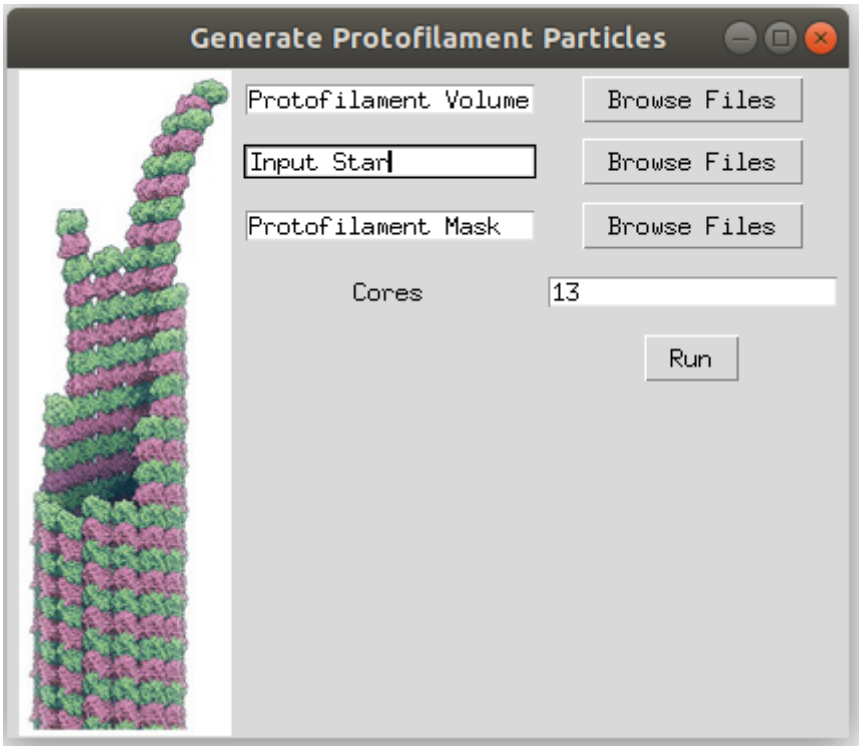


While most of input parameters are fairly straightforward, there are three crucial settings that are boxed in red. First, the angular sampling rate must be lower than the local search parameter in order to force local refinement. Second in the additional argument tab you'll want to specify the range of angles to search locally using the *–sigma_ang* option. I usually use a value of 2 or 3. Additionally, I usually like to specify the mask diameter to be slightly below the size of my particles, but this is probably not really necessary. Lastly, for the first couple times of doing this I would recommend not submitting the job to a queue so you can troubleshoot any errors, but again this is a matter of preference.

8 Further Processing: Protofilament Subtraction

Assuming protofilament refinement was successful and the overall resolution of the tubulin in your microtubule has increased, you now have much better alignment parameters for each protofilament in your dataset. To further push the resolution (and help with any classification you want to do), it's recommended you do a second series of subtractions, only this time, by subtracting one protofilament at a time. This should result in an improved subtraction with less residual signal, allowing for better alignments and/or classification. To perform protofilament subtraction run the following command.

pf_protofilament_subtract

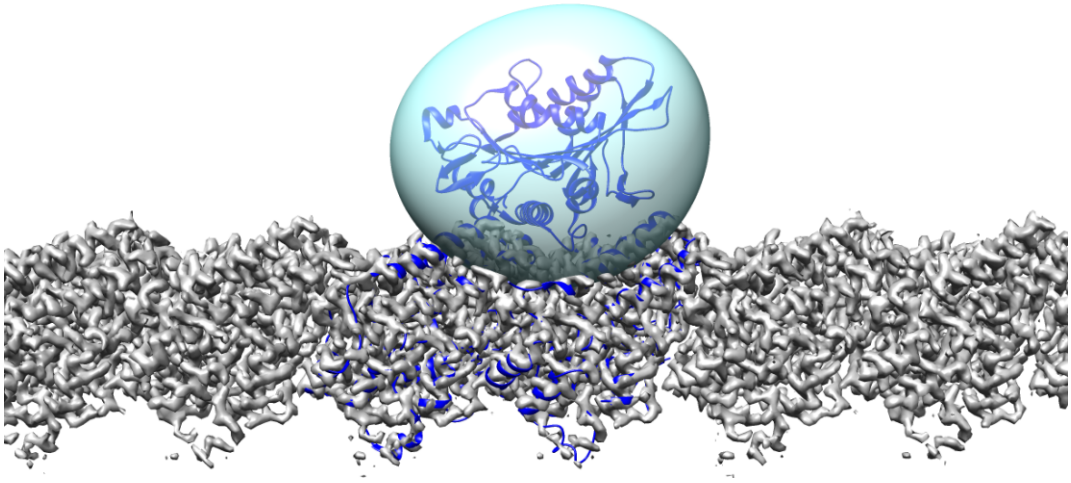


Here the protofilament volume and input star file should be the outputs from the final iteration of protofilament refinement (e.g. Refine3D/jobXXX/run_class001.mrc and Refine3D/jobXXX/run_data.star). The protofilament mask should be the mask used for protofilament refinement. If you don't run this step in parallel, it will likely be the slowest portion of the pipeline due to the number of subtractions you're performing ($X*N*(N-1)$ subtractions, where X is the total number of particles and N is the number of protofilaments). The output of this step that you are interested in will be the *protosubbed.star* file, which can be used for further processing.

9 Protofilament Refinement: Round 2

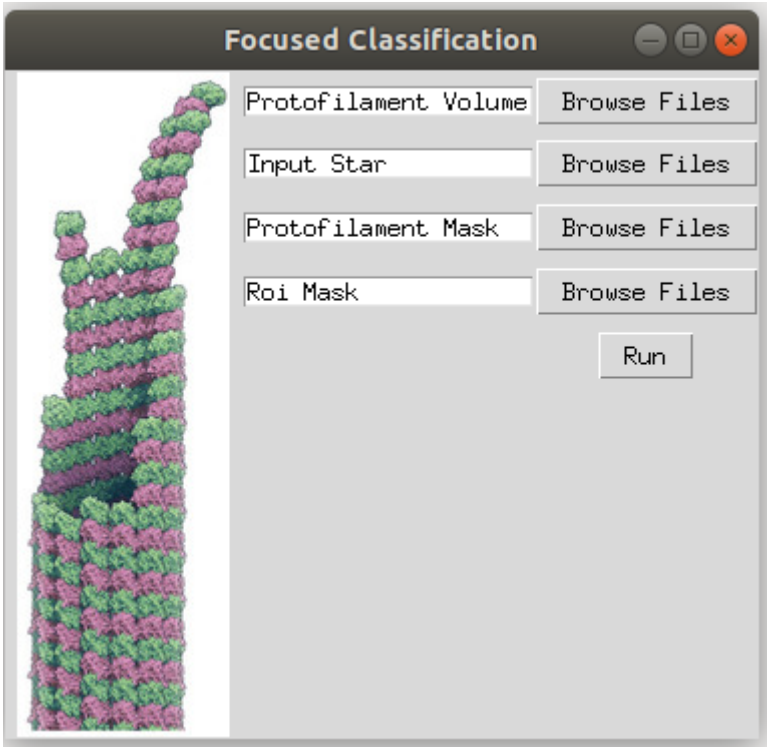
This basically the same as the first iteration of protofilament refinement, except you will use the *protosubbed.star* file as your input. However, you can (and probably should), input more restrictive alignment parameters (e.g. decrease *–sigma_ang* to 0.5 or 1).

10 Focused Classification



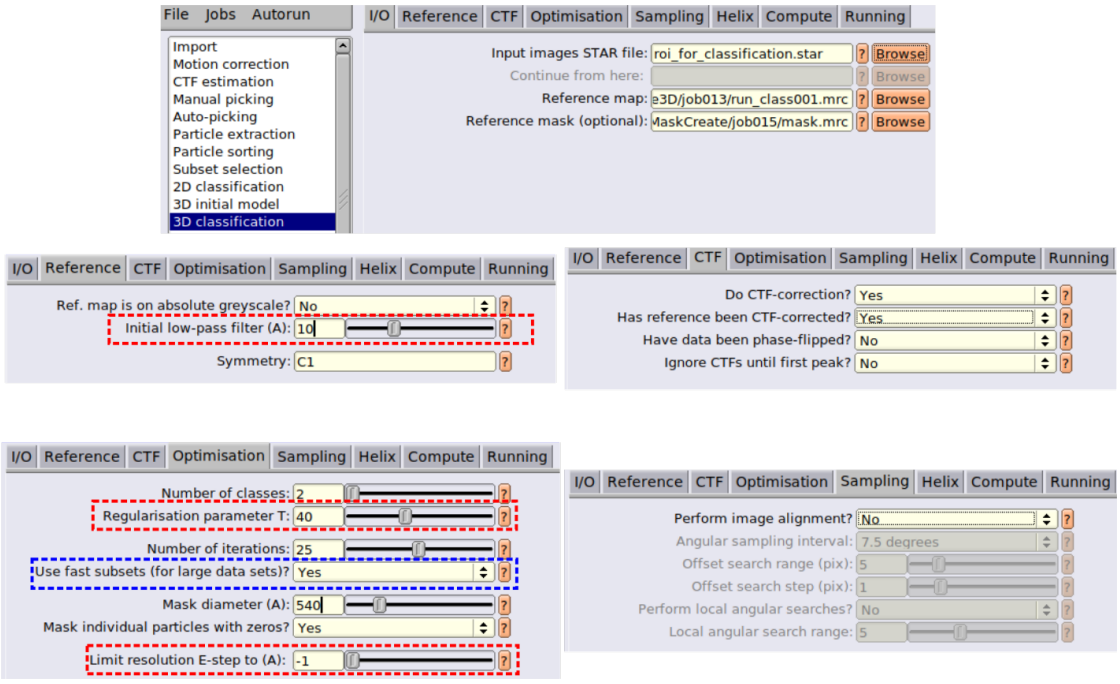
For focused classification we would ultimately like to generate particles containing only a region on interest. The first step of this process is to make a mask in 3D space surrounding your region of interest relative to your protofilament volume. For example, I have fit a PDB (3J8X) to middle tubulin dimer in my protofilament volume, saved only the kinesin portion of the structure, and then used EMANs **pdb2mrc** function, to generate a kinesin map in my region of interest. Next I'll make a mask of this map within RELION, which will be used for classification as well as subtraction. We'll call this mask the ROI mask (the cyan volume from the above image). Once you've made your ROI mask, you can generate your particles containing only the region of interest using the following command.

pf_focused_classification



Here the protofilament volume, protofilament mask, and ROI mask have already been described. Additionally the input star file should be your most recent or successful protofilament refinement output star file. Again RELION is used for the actual subtraction step. Unfortunately this process is not parallelized, and can take quite a while depending on the number of particles you have. The outputs from this step that you are interested in will be the *roi_for_classification.star* and *roi_for_init_classification_30k.star* files. The first file contains information for a particle stack containing only the region of interest for all your particles, whereas the second contains information for an approximately 30,000 particle subset of your dataset (if your dataset is larger than 30,000 particles). This second file may or may not be useful depending on your classification needs (described below).

The actual classification portion of the analysis is a little less robust and you may have to play around with the parameters a bit to get the classification to converge. I'll explain our methods and parameters used to find regions of the microtubule where kinesin was bound as an example, but this may or may not work for your needs.



Here, the most important feature is boxed in blue. I have found that when working with large datasets, my classification will not converge unless I select this parameter. Additionally, even when selecting the 'fast subsets' option, I have had difficulties with my data converging. If this proves to be the case, I will instead use the *roi_for_init_classification_30k.star* file for an initial round of classification. The resulting two classes can then be used to seed classification of the entire dataset (the fast subsets option does not need to be selected in this case).

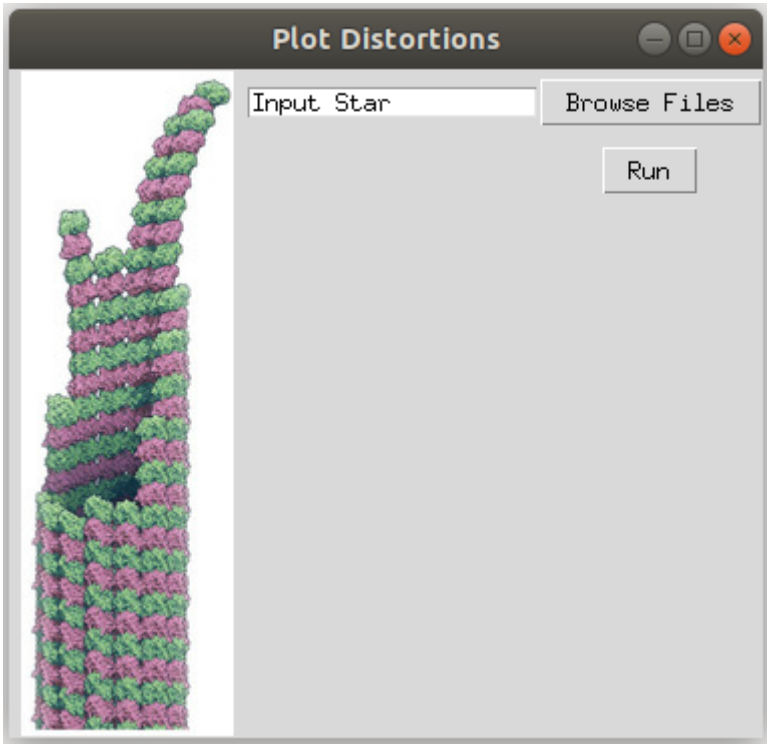
The next 3 important parameters are boxed in red above. These are very much dependent on what you are trying to classify. For instance, if you looking for large-scale structural rearrangements, you can use a high initial low-pass filter, and to speed up classification, you can limit the E-step resolution to a lower resolution as well. Lastly, you may want to play around with the regularization parameter (T). The smaller your region of interest, the higher you want this value you to be. Furthermore, if you are more interested in higher resolution features, you can continue to increase this value.

11 Distortion Analysis

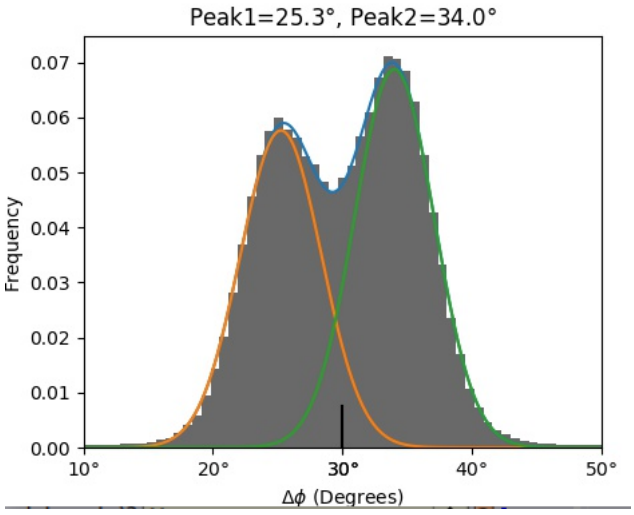
A final mode of analysis that can be performed is distortion analysis. The idea is to gain a better understanding of the lateral interactions occurring within a microtubule. One method of doing this is by calculating the angle between each adjacent protofilament in a microtubule. We can then look at the distribution of these angles over the entire dataset. To do this we've added a simple plotting script that will plot the distribution of these angles for all protofilaments pairs, as well as the seam and non-seam protofilament pairs.

Lastly we've added gaussian or double gaussian fits to these plots. (Note these fits are not perfect, and you may want to play with plotting the data yourself, see below). To perform distortion analysis, run the following command.

pf_plot_distortions



Here, the input file should be the resulting refined star file following either your first or second protofilament refinement. The fit and unfit results will be stored in the *plot_dphis* directory. An example plot is displayed below



Lastly, if you would like to play with the data more, feel free to use the calculated data

within python.

from pf_refinement import StarOp
data=StarOp('Refine3D/jobXXX/run_data.star')
data.delPhiXs()
data.dphixs ##*Dictionary containing distortion results for individual microtubules*
data.dphi_flat ##*NumPy array containing the aggregate data*
data.dphi_flat[data.num_pfs-1::data.num_pfs] ##*Distortion data at the seam*

12 Concluding Remarks

We hope you find that protofilament refinement provides a significant boost to the quality of your reconstructed microtubule volumes. This is currently the beta version of the program and we anticipate a few errors along the way. Feel free to contact us with any issues, or even ideas for further feature implementation. If any of the work you do with this program is published, please cite us at XXX.