em-Clarity

tutorial version 1.0.4

Table of Contents

0.0 How to use this guide

The purpose for this document is to provide the minimal set of instructions needed to begin processing sub-tomogram data using emClarity. Other sources of information include the emClarity wiki and the emClarity mailing list. Of particular interest are a number of tutorial videos which discuss the details, both practical and technical, behind each of the steps you will execute during the tutorial. The first video on that page, "Overview on info related to emClarity"

If at any point you are confused, or something seems to not work as you expect, the videos are a good place to start; feel free to post to the mailing list any questions that you cannot resolve on your own.

Sections 1.0 – 2.2 are needed for all new users. The remainder of section, which details tilt-series alignment and getting and formatting the required files, may be skipped to get straight into the sub-tomogram workflow.

Expected Results:

You should be able to get to an ~ 9.0 Å structure by following the tutorial as written. Some extra modifications to the processing are needed to fully reproduce the results we report in the paper describing emClarity "***High resolution in situ structural determination of heterogeneous specimen.***" Currently on bioarxiv and under peer-review at your favorite neighborhood methods journal.

Diagnostic info:

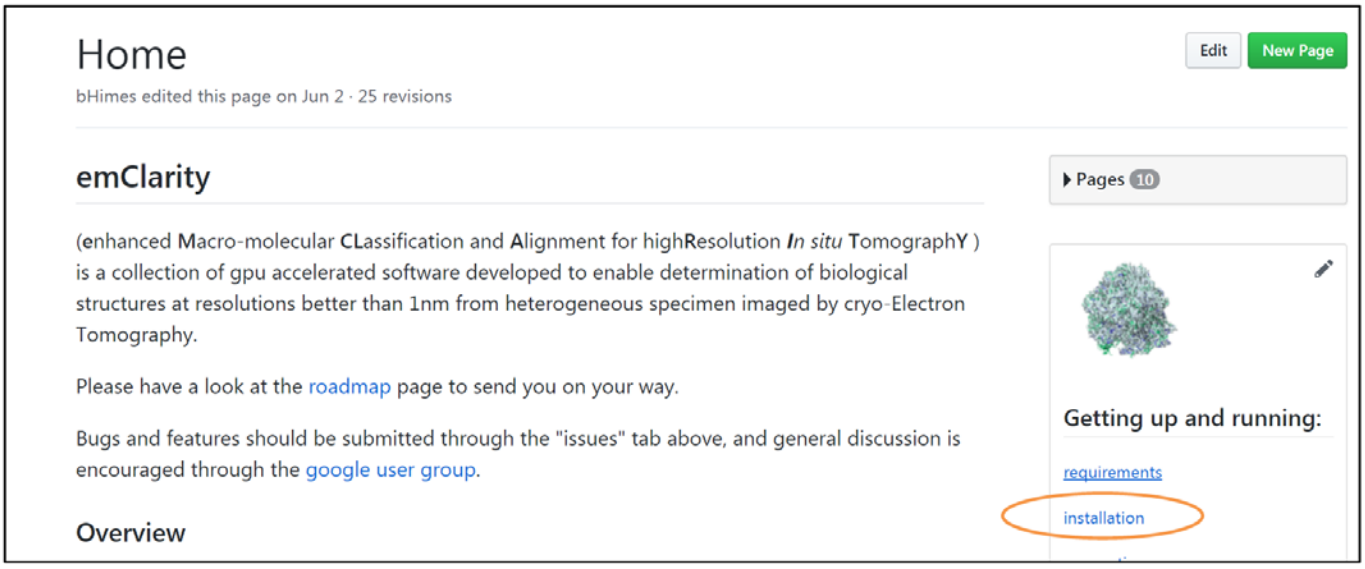
Since emClarity is still new and shiny, we leave produce quite a bit of diagnostic information to help with trouble shooting. The largest items will be aligned stacks from previous rounds of tomoCPR in the "aliStacks" directory, as well as reconstructions in the "cache" directory. Anything in the "cache" will be automatically created when needed, so feel free to delete those data when you are finished.

1.0 Setup

1.0 Obtaining the dependencies

1.0.1 Navigate to the emClarity wiki

1.0.2 Click on the installation link on the sidebar to find a section describing the



most up to date software and versions required.

Note 1: You must use the version of the matlab MCR as specified.

Note 2: To download the IMOD version, you must right click and "save link as."

1.1 Installing the dependencies

1.1.1 Both IMOD and chimera can be installed locally without admin rights

```
$ ./currentVersionOfIMOD.sh -skip -dir /Path/to
```

```
$ mkdir MyMatlab_MCR ; cd MyMatlab_MCR # whatever you want to call it
$ unzip /path/to/mcrdownload.zip
$ ./install -mode silent -agreeToLicense yes -destinationFolder MyMatlab_MCR
```

1.1.2 When the matlab MCR is finished, you will be provided with a line to append to your LD_LIBRARY_PATH. You could to this by modifying your .bashrc file, but it is preferable to copy this line into a text file called **mcr_bash.sh** saved in the directory where you installed the MCR. The script you will use to run emClarity will source this file on start-up so that the appropriate libraries can be found.

1.1.3 Each of these are well documented, so rather than anticipate every situation, have a go and if you get stuck feel free to ask for help.

1.2 Download and install emClarity

The code is kept one level above the wiki. You can obtain the files needed to run emClarity by cloning the repository using git. This of course assumes you have **git** installed, if not you will need to do that first. In the directory you wish to install emClarity run the following.

```
$ git clone --depth=1  https://github.com/bHimes/emClarity.git
```

Note: Git stores all old versions, so using the "depth" flag prevents you from downloading all the old binaries, which each contain a lot of "extra" libraries so that they are self contained, i.e. are "big."

This will create a directory called emClarity which will have a few items in it, which are described in the accompanying README. If at any time you want to update to a newer version, simply run:

```
$ git pull --depth=1   https://github.com/bHimes/emClarity.git
```

Modify the line in the emClarity script to point to the mcr_bash.sh file you created in the installation of the matlab MCR.

Modify also the line to point to the installation directory.

Note 1: The file emClarity is just a shell script that points to the binary which will have a 7 character suffix which is beginning of the hash identifying the particular version (commit) that generated the file. This will also show up in all log files, making trouble shooting more straight forward.

Note 2: If you are running on a distributed computing system, you can use the example "runMatfile.sh" in the docs folder as a template, which essentially creates a run script like emClarity but with additional details relevant to queue submission.

Finally run the following to have emClarity check your installation for you.

```
$ emClarity check
```

This program will create a log file that will list the locations of imod and chimera as well as available gpus. emClarity itself has a few checks built in to make sure it can run.

2.0 Obtaining tutorial data

Goal:

Obtain the two of the seven tilt-series available from EMPIAR-10045.

Note: an important feature of cryoSTAC is the ability to work with a small data set, and then scale the process. It is best practice to work the whole way through the pipeline with as small a set as possible, partially scaling up to make sure everything holds, and then processing your full data.

If you are trying to reproduce the results we have published, you will of course nee to download all seven tilt-series.

Process:

2.1 open a terminal and navigate to your workspace. Preferably on a solid state drive.

2.2 create a working directory, and a raw data subdirectory.

Note: it is okay if the raw data have complicated naming conventions, but to prevent needless headaches, we will rename everything in a simple format later on.

```
$ cd  < Myname/wherever  >

$ mkdir emClarity_tutorial emClarity_tutorial/rawData emClarity_tutorial/fixedStacks
```

2.3 Tilts 005 and 008 are the mostly flat, so start with those. It is advisable to use the "Aspera Connect" or alternatively you can pull directly from the ftp server.

```
$  cd emClarity_tutorial/rawData


$ for iTomo in 05 08; do wget -b \
ftp://ftp.ebi.ac.uk/pub/databases/empiar/archive/10045/data/ribosomes/Tomograms/${iTomo}/IS002_291013_0${iTomo}.mrc  ; done
```

2.4 Have a look at the raw data.

```
$  imod  -bin 4 IS002_291013_00?.mrcs
```

- bin decimates (downsamples) the data, but only in 2d since imod treats this as a stack of images. Using -B 4 would bin in 3d.
- the single character wild card "?" allows us to open all matching names in one window.
- in the default (zap) window,a left click on the image will play through the stack, allowing you to use motion to asses the quality.
- using the 1 and 2 keys (above Q W) you can toggle through the different images.

Note: While we can and do fix the tilt-series alignment, the same conditions that cause the automated tracking to fail during data collection, are often those that also preclude high resolution work. If the tracking is very poor, the data are likely not suitable.

The remainder of section 2 can be skipped by copying the expected tilt series alignment results.

```
$   mkdir emClarity_tutorial/fixedStacks
$   cp -r /path/to/emClarity/docs/tiltAlignments/*  emClarity_tutorial/fixedStacks
$    cd emClarity_tutorial/fixed Stacks
$    nTomo=1; ls ../rawData/*.mrcs | while read iTomo ;  do ln -s ../rawData/${iTomo} tilt${nTomo}.fixed
     nTomo=$(( $nTomo + 1)) ; done
```

2.5 Coarse alignment

While these example tilt series are already aligned, we need to generate a model file that tells emClarity where the gold fiducials are, so they can be removed, and additionally collect a few other files. If you are unfamiliar with this process, or run into any trouble, the tutorial video contains many additional details.

2.5.1 Setup – fixing the header

It is important at this stage to fix the information in the header that is incorrect. The pixel size is not stored correctly for whatever reason. This is something you should always confirm with your own data as well.

The imod command

```
$ alterheader IS002_291013_005.mrcs
```

will open up an interactive dialog where you must do both:

1) Type in "cel" and enter the true pixel size (2.17 Angpix) * (celldimenion) and cell angles

   2.17*NX 2.17*NY 2.17 *NZ 90 90 90

   Strike enter to return to the menu.

2) Type in "done"

I'm sure this could be scripted, and if anyone would like to contribute such a script that would be great!

2.5.2 Setup – creating the alignment directory

From your rawData directory, create a working directory for the imod/etomo alignment. This program writes many files to disk, so working in its own directory makes cleaning up afterward much easier.

```
$  mkdir imodAli ;  cd imodAli
```

Make links to the raw data, changing the names to something simple tilt1 … tilt2 etc.

Also change the file extension to either .mrc or .st which etomo prefers.

```
$ ln -s ../IS002_291013_005.mrcs tilt1.st
```

Then open the etomo interface

```
$ etomo
```

2.5.3 Running the alignment

- Select build a new tomogram.
- Select your data set, single-axis, scan header (gold fiducials are 10nm here) and also change the image rotation angle to **–4.** We didn't fix this in the header because it should never be this far off in your own data that is totally unaligned.
- Select "Create com scripts"

A new screen will pop up that covers the important steps.

- Pre–processing isn't necessary for this data set, but you should check your own, particularly if you have CCD data.
- Bin the coarse aligned stack by 4 unchecking "convert to bytes"
- Use all the available fiducial markers
- Run fine alignment, selecting local alignments
- Skip tomogram positioning
- Create full aligned stack
- Under the erase beads tab, use erase beads 3d setting thickness to 3000
- Select "align and build stack", "run find beads 3d", "project model into 2d"

2.5.4 Gathering the results

Please pay careful attention to the naming conventions in this section!!

There are four files needed.

The refined tilt angles

```
$ mv tilt1_fid.tlt ../fixedStacks/tilt1.tlt
```

The 2d transformations

```
$  mv tilt1_fid.xf ../fixedStacks/tilt1.xf
```

The local alignments

```
$ mv tilt1local.xf ../fixedStacks/tilt1.local
```

The gold bead model (for erasing later)

```
$ mv tilt1_erase.fid ../fixedStacks/tilt1.erase
```

We do not want the final aligned stack that imod applied the transformations to (tilt1.ali) instead there are two alternate choices for the "fixed" data.

1 – if you didn't do any preprocessing (xray removal) instead we will link to the raw data

```
$ cd ../fixedStacks ; ln -s ../rawData/IS002_291013_005.mrcs tilt1.st
```

2 – if you DID remove xrays

```
$  mv tilt1_fixed.st ../fixedStacks/tilt1. fixed
```

```
$  imodtrans -i ../fixedStacks/tilt1.erase ../fixedStacks/tilt1.erase tmp.mod
$  model2point -float tmp.mod ../fixedStacks/tilt1.earse2
```

Repeat this for each of your tilt series, incrementing the name tilt1 tilt2 …tiltN. In practice you can often run multiple alignments concurrently, so in that case you might have named your temporary directory imodAli_1 imodAli_2 …etc.

2.5.6 Obtaining an initial CTF estimation

First set up the microscope parameters in your parameter file, please see the video if the text descriptions are unclear.

For each tilt series run the following command. emClarity will simply send the job to the GPU with the most available memory – it is advisable to only one one job / GPU at this step. If you are not running on a distributed system with a queue manager, simply open up one shell for each GPU. E.g. for two gpus:

```
$  for iTiltSeries in 1 2 3 4 ; do emClarity ctf estimate tilt${iTiltSeries}; done
```

```
$  for iTiltSeries in 5 6 7 ; do emClarity ctf estimate tilt${iTiltSeries}; done
```

When this is finished, review the results, in particular the fit to the power spectrum which is reported in the file "fixedStacks/ctf/tilt(N)_psRadial.pdf"

An additional directory called "aliStacks" will be created where the tilt-series with the current transformation parameters applied will be placed. After each round of tomoCPR these are re-calculated and the suffix "_ali1_ … _ali2_ …" will be incremented and is = #tomoCPR iterations completed + 1.

3.0 Template-matching

If you have elected to skip the tilt-series alignment and initial CTF estimation by copying the expected results to your fixedStacks directory, you will need to first create all of the alignedStacks.

If you are not running on a distributed system with a queue manager, simply open up

one shell for each GPU. E.g. with two gpus:

```
$ for iTiltSeries in 1 2 3 4 ; do emClarity ctf update tilt${iTiltSeries} full; done
```

```
$ for iTiltSeries in 5 6 7 ; do emClarity ctf update tilt${iTiltSeries} full; done
```

3.0 Selecting regions in each tilt-series to process

It is often useful to select sub-regions from your tilt-series to be used for subsequent processing; either because there is only specimen in some areas, or alternatively to reduce the final size of reconstructions, which can easily be larger than 100Gb each if a 4k x 4k tilt series is reconstructed.

The following steps are accompanied by a short video demonstrating the process.

Run the script from you docs folder to set things up, either copy to your working directory, add it to your path, or specify the full path. Assuming that you have copied it to your working directory:

```
$ ./recScript2.sh -1
```

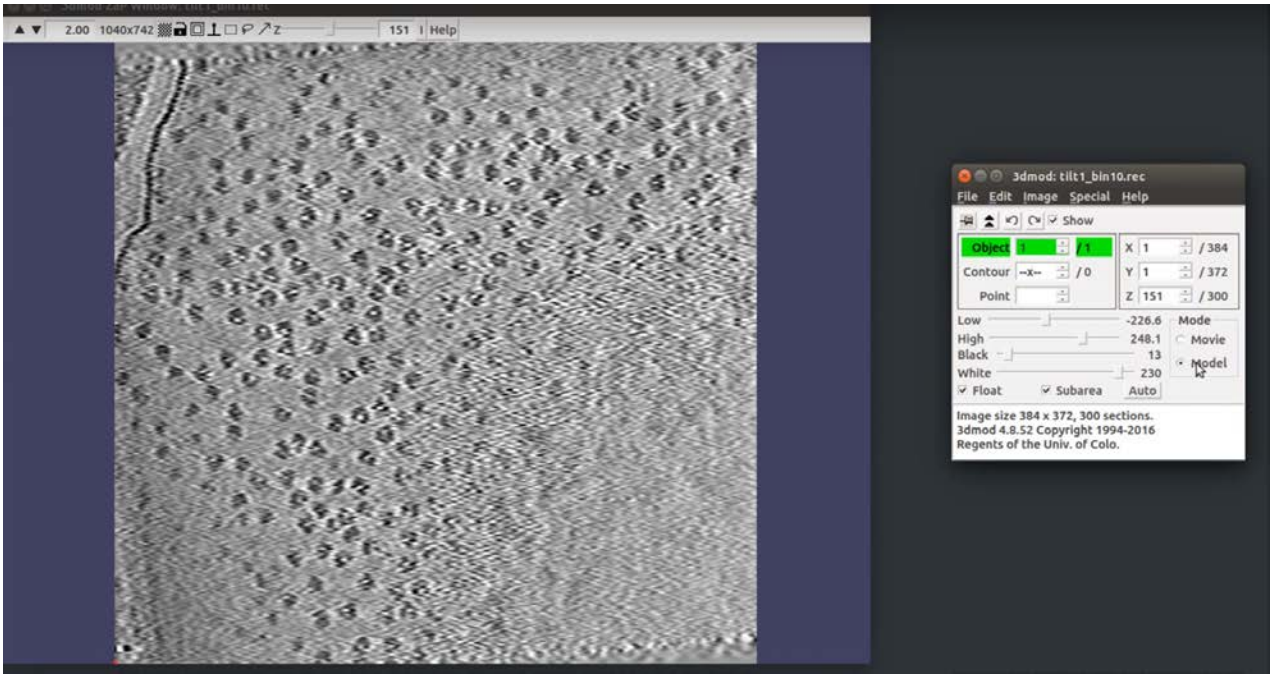
This will create a new directory, “bin10” which will have a bin10 tomogram for reconstructed for each tilt-series present in the fixedStacks directory.

The goal is to define 6 points, xMin/xMax, yMin/Max, and zMin,zMax for each subregion you wish to study. This is most easily done by creating and IMOD model, where each of these points is represented by its own contour, and selected in this order. You should end up with (number of regions) * 6 contours in your model.

```
$ cd bin10

$ imod -S tilt1_bin10.rec
```

After selecting each point, either strike “N” to start a new contour, or move in z with page-up or page-down, which also creates a new contour.



Finally, save the model with the same name as the reconstruction, where the postfix has been changed from "rec" to "mod" in this case tilt1_bin1.mod.

You will do this for each of your 2 or 7 tilt series.

After all are completed, we need to convert these models into an input for emClarity, again using the recScript2.sh.

** It is helpful to keep a list of ~ how many particle you expect based on this visualization. Of course you can incorporate other a prior information here as well.

```
$ cd ../      # back to your project directory

$  for iTiltSeries in 1 2 3 4 5 6 7 ; do recScript2.sh  tilt${iTiltSeries}; done
```

This will create a directory called "recon" with files in it needed for emClarity. Those with the postfix ".coords" are the final barrier into the sub-tomogram workflow. If you decide after template matching to scrap a particular tilt-series, just delete this file prior to initializing the project meta data.

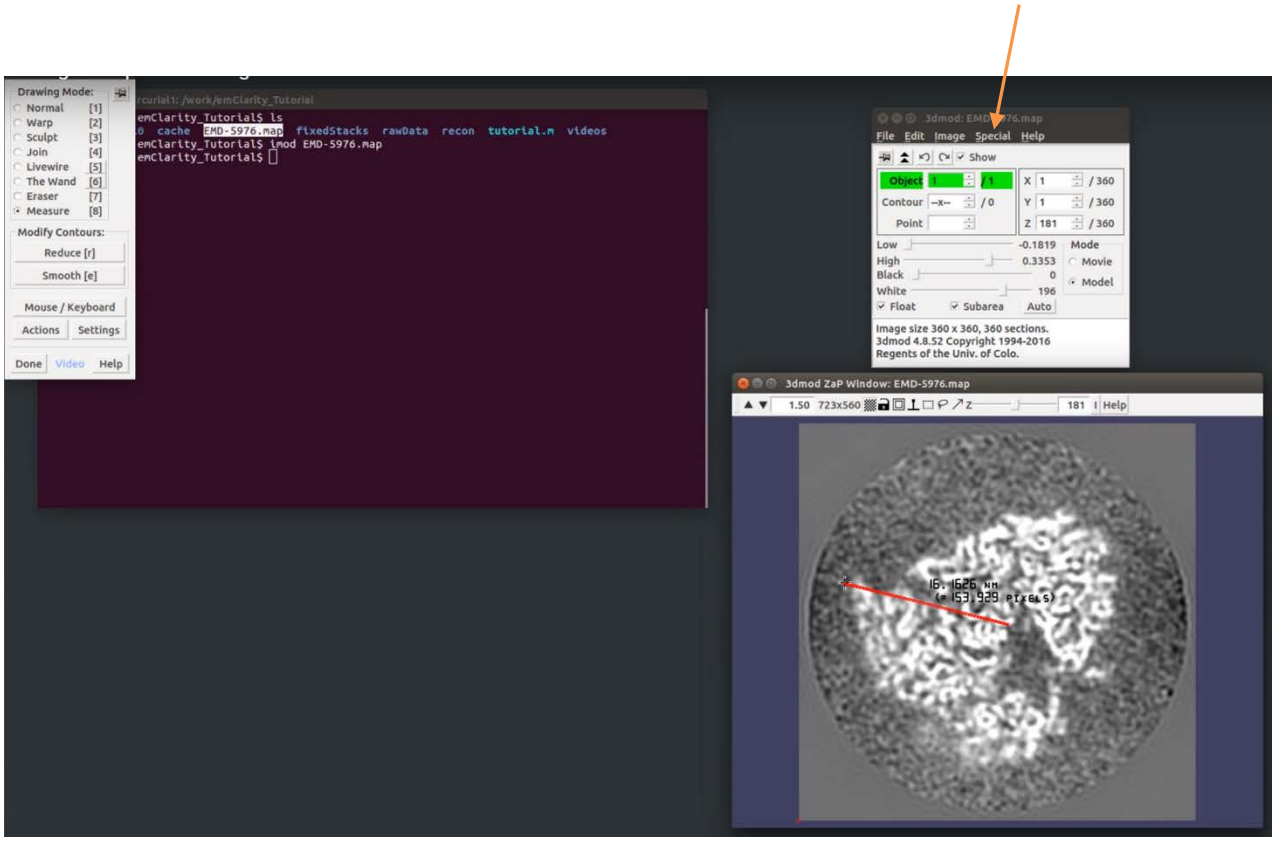
3.1 Choosing and preparing a reference

Download a suitable yeast 80s map to use as a reference, EMD-5976 for example. We assume we have some low resolution structure (< 40A) available.

Determine the particle radius in each dimension, this easily done in imod, selecting the Special menu, and then "drawing tools" as demonstrated in this video.

You will also need to rescale the map to make to match the data. Pixel sizes are in Angstrom

```
$  emClarity rescale EMD-5976.map riboReScale.mrc 1.05 2.17 GPU
```



3.2 Selecting template matching parameters

Select a threshold that is about 10% > than the expected number of particles. It is okay if this is not very accurate, as the results can be interactively edited in 3d and later through classification. It is best practice to clean up the data as much as is reasonable prior to relying on classification though.

For particles like the ribosome, there are no real constraints on the orientation, so searching a full grid in 12 or 15 degree increments is required.

This would be specified as [180,15,180,12] for +/- 180 in 15 degree steps (polar angle, which also determines the azimuthal angle) and +/- 180 degrees in 12 degree steps in the planar angle.

If you have a particle with C6 symmetry (with the symmetry axis corresponding to the Z-axis) you might search a more limited range [180,15,36,9]

The sampling rate should be chosen to give a running pixel size between 8-12 Angstroms

3.3 Running the job

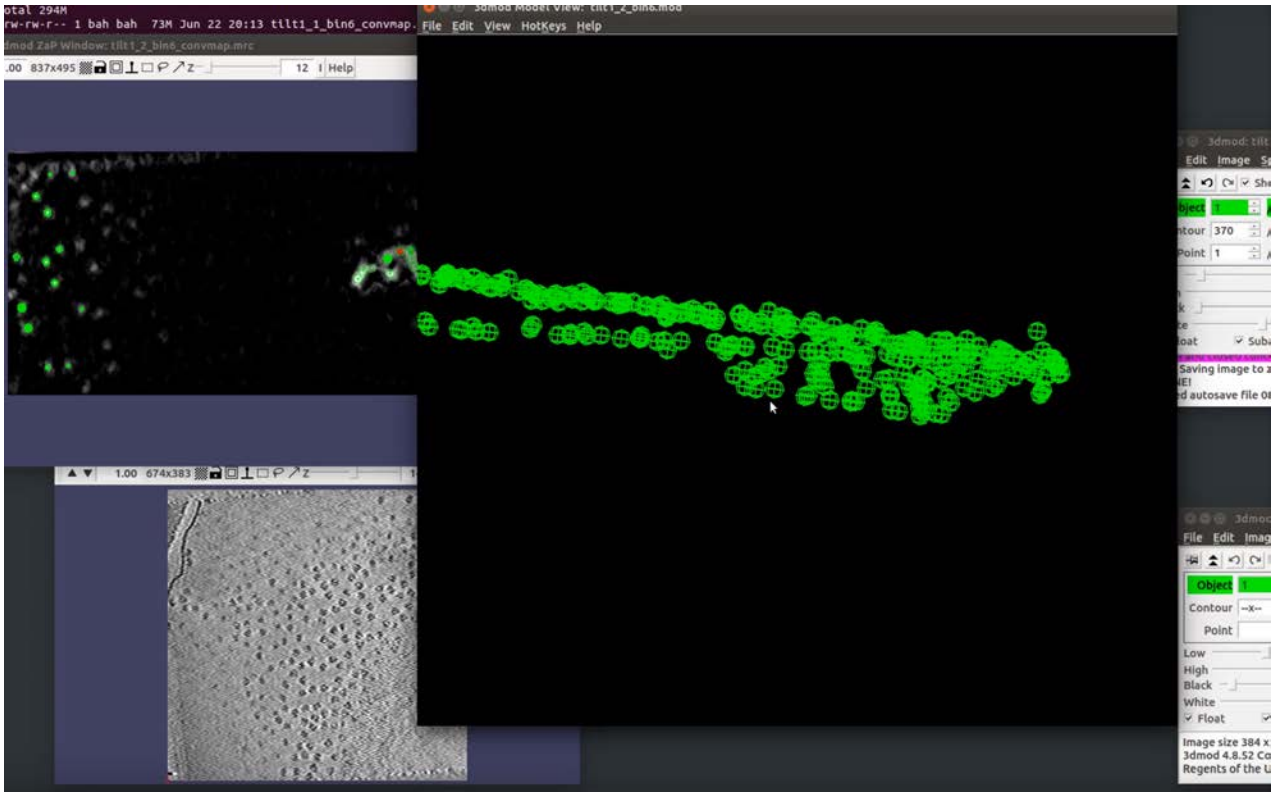
Similar to ctf estimation, each call to the template search program should be run on its own gpu.

```
$ for iTilt in 1 2 3 4 5 6 7; do for jTomo in 1 2 3 4 ; do emClarity templateSearch param0.m tilt${i} ${j} riboReScale.mrc 1 1 ; done ; done
```

Where the final two parameters are the symmetry and gpuIDX, the number of tomograms will match whatever you have selected from each given tilt series.

3.4 Analyzing and editing the results in 3d

The primary goal here is to remove false positives due to strong homogeneous features like carbon edges, membranes, or residual gold beads. In addition to the bin10 tomogram, which helps to visualize the sample, the template matching produces a “cumulative correlation” map which can be opened alongside a 3d model of the best peaks. Taken together, these tools make editing the results pretty effortless as demonstrated in the tutorial video.



4.0 Initial run

The final steps to getting up and running are described in this tutorial video.

4.4 Initialize the subTomoMeta

4.5 Create an initial average

4.6 Analyze results and adjust as necessary

I will continue to update the following sections as possible. The paper describing the methods is out for review, and will also be posted to bioarxiv shortly (link to follow)

Until more documentation can be generated, please use the mailing list to request information.

5.0 Alignment

5.5 General considerations

5.6 Using symmetry

5.7 Increasing sampling rate and convergence

5.8 Multi-reference alignment

6.0 Classification

6.1 PCA

6.1.6 Selecting regions to for analysis in real space

6.1.7 Running on all or some of the data

6.1.8 Analyzing eigenimages

6.1.9 Analysing variance maps

6.1.10 Selecting features for clustering

6.2 Cluster

6.2.1 Select algorithm

6.2.2 Run and analyze results

6.6 Removing classes from further analysis

6.7 Selecting classes to use as references

6.8 Skipping class average alignment and /or rawalignment