em-Clarity

tutorial version 0.0.2

Table of Contents

7.0

1.0 Setup

1.1 Obtaining the dependencies

1.1.1 Navigate to the emClarity wiki – http://github.com/bHimes/emClarity/wiki

1.1.2 Click on the installation link on the sidebar to find a section describing the most up to date software and versions required.

Note 1: For the matlab mcr, you MUST use the specified version, and it is highly recommended to use the versions of supporting software as listed.

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

1.2 Installing the dependencies

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

1.2.2 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 approriate libraries can be found.

1.2.3 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.

1.3 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 https://github.com/bHimes/emClarity.git
```

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 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 text file that points to the binary which will have a 7 character suffix which is the 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 easier.*
>
> *Note 2: If you are running on a distributed computing system then you can use the example "runMatfile.sh" in the docs folder as a template, which essentially creates a textfile like the emClarity text file 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

Forward:

For those unfamiliar with using the command line, please have a look at the video tutorial which accompanies this chapter to get a more in-depth explanation of what each command is doing. You can of course also just copy and paste the commands, but that will not help you in the future when you are attempting more convoluted experiments.

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 small data sets 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. This will save a substantial amount of your valuable time.*

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 fine if the raw data has complicated naming conventions, but to prevent needless headaches, we will rename everything in a simple way later on.*

```
$ cd  < Myname/wherever >

$ mkdir  rln_tutorial  rln_tutorial/rawData  rln_tutorial/fixedStacks
```

2.3 Tilts 005 and 008 are the most flat, so start with those. It is advisable to use the "Aspera Connect" which is described **here**. Alternatively you can pull directly from the ftp server.

```
$  cd rln_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.

Note: While we can fix the alignment, the same conditions that cause automated tracking to fail during data collection also preclude high resolution work.

```
$  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.

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. Both the pixel size and the origin are 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 "org" and enter the appropriate origin, which is

   0 0 0

   Strike enter to return to the menu.

3) 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
```

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

For now, please refer to the relevant tutorial video here

https://github.com/bHimes/emClarity/wiki/Tutorial-videos