Pre-processing of dataset from Titan II

1. Recover the dataset from the microscope

Log into **cbi-gateway-01** to access the microscope and transfer the data to **gateway** or **storage**

```
ssh <login>@cbi-gateway-01
ls /mnt/titan2/offloaddata/TemScripting/EF-Falcon/<dataset>
rsync -avz --progress /mnt/titan2/offloaddata/TemScripting/EF-
Falcon/<dataset> /mnt/storage/teams/<your_team_folder>/<project>/<dataset>
```

Path to gateway as seen from cbi-gateway-01 : */mnt/zfspool/teams/*

2. Convert the EER movies to TIF

Log into **cbi-compute-01** and transfer the scripts

eer2tif_parallel_loop.py

and

eer2tif_parallel.py

in your **home**

```
ssh <login>@cbi-compute-01
module load relion
cd /mnt/storage/teams/<your_team_folder>/<project>/<dataset>/ #go to the
folder with the dataset
module load python
python ~/eer2tif_parallel.py
```

Warning : conversion of eer to tiff file will show error this message : *TIFFReadDirectory: Warning, Unknown field with tag 65002 (0xfdea) encountered.*

3. Begin processing with relion

Load and start relion (from cbi-compute-01 or the team GPU node (phantom-node39 for Lamour-Ruff))

```
module load relion
relion& #start relion in the folder with movies
```

Job Relion/Import

Give the path to the *.tif , specify the pixel size, voltage and hit **RUN** You can **RESUME** the job when more movies are transferred and converted

Estimation the camera gain

Estimate the gain with at least 200 movies (redo if needed with a larger number)

```
relion_estimate_gain --i Import/job001/movies.star --j 8 --max_frames 10000
--random true --o estimated_gain.mrc
```

Explanation of options:

- –j: number of thread
- –max_frames: target number of frames to average (rounded to movies)
- –random: randomize the order of input movies before taking subset

Job Relion/Motion correction

Specify the dose per frame, the number of patches 5x5m, provide the gain file, and set several MPI (number of tasks in parallel) and several (as a multiple of the number of frames in the movies) threads (number of cpu per task)