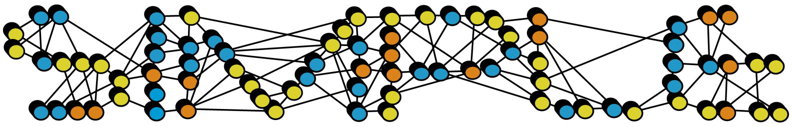

SIMPLE Workshop 2017

Dominika Elmlund, Hans Elmlund

January 17, 2017

Contributors:
cyril.reboul@monash.edu
dominika.elmlund@monash.edu
hans.elmlund@monash.edu
Adress:
Dept. Biochemistry and Molecular Biology
School of Biomedical Sciences
Monash University, Bldg. 77
Clayton, VIC, Australia, 3800
Webpage:
www.simplecryoem.com

1 What Will Be Covered

This part of the workshop will cover

1. DDD (Direct Detector Device) movie pre-processing
2. 2D analysis with PRIME2D
3. *ab initio* 3D reconstruction from class averages using PRIME3D

with the SIMPLE program package. We will use the latest unreleased version of SIMPLE, available to the workshop participants for downloading and beta-testing on the SIMPLE homepage `http://simplecryoem.com/beta.html`.

2 DDD Movie Pre-Processing

During image acquisition, beam induced motion, charging and stage drift introduce blurring in the integrated movies. The individual frames need to be aligned with respect to one another in order to restore high-resolution information. Because of radiation damage, the electron dose is kept low during image acquisition, resulting in a very low signal-to-noise ratio (SNR) of the individual movie frames. Until recently, our understanding of damage processes in cryo-EM was based primarily on results obtained with 2D protein crystals (REFS 44,45 REVIEW). The use of 2D crystals provides direct means for assessing the degree of damage by looking at the rate by which high-resolution diffraction spots are fading as a function of the accumulative electron dose. The disadvantage here is that it is impossible to separate spot fading due to loss of crystalline order from intrinsic molecular damage. Images of single rotavirus VP6 particles recorded with a total exposure of 100 electrons/Å2 indicated that single-particles are more robust to damage at low and intermediate resolution than previously thought (REF 46 REVIEW). In the same study, a dose-weighting scheme was introduced that maximizes the SNR of the integrated movie in a dose-dependent manner. This dose-weighting strategy has been adopted by most software packages and proven to be a powerful addition to the ever-growing set of single-particle analysis tools.

SIMPLE implements a program called unblur_movies for simultaneous motion-correction and dose-weighting, with the objective of maximising the SNR of the integrated movie in a motion- and dose-dependent manner. Input is a text file `filetab`, simply listing the individual movies in `*.mrc` format, the pixel-to-pixel (or sampling) distance `smpd` (in A), the number of CPU threads to use `nthr`, the body of the outputted files `fbody`, the exposure time of the movie `exp_time` (in s), the dose-rate at which the movie was acquired `dose_rate` (in e/A2/s) and the acceleration voltage of the electron microscope `kv` (in kV), in addition to a number of other optional parameters that we will not be concerned with here. If we execute `simple_exec prg=unblur_movies` in the prompt, we obtain brief instructions for how to run the program:

```
@!#> simple_exec prg=unblur_movies
USAGE:
bash-3.2$ simple_exec prg=simple_program key1=val1 key2=val2 ...

REQUIRED
filetab = list of files(*.txt/*.asc)
smpd    = sampling distance, same as EMANs apix(in A)

OPTIONAL
nthr      = nr of OpenMP threads{1}
fbody     = file body
lpstart   = start low-pass limit{15}
lpstop    = stop low-pass limit{8}
trs       = maximum halfwidth shift(in pixels)
exp_time  = expusure time(in s)
dose_rate = dose rate(in e/A2/s)
kv        = acceleration voltage(in kV){300.}
pspecsz   = size of power spectrum(in pixels)
numlen    = length of number string
startit   = start iterating from here
scale     = image scale factor{1}
frameavg  = nr of frames to average{0}
tomo      = tomography mode(yes|no){no}
```

Output consists of four files per movie (`movie1.mrc` in this example):

1. `<fbody>_intg1.mrc` is the frame-weighted, motion-corrected and dose-weighted sum

2. `<fbody>_forctf1.mrc` is the un-weighted sum of the aligned individual frames

3. `<fbody>_pspec1.mrc` is the power-spectrum of the uncorrected unweighted movie sum (left) and the corrected weighted movie sum (right)

4. `<fbody>_thumb1.mrc` is ta down-scaled version of `<fbody>_intg1.mrc`

The `*forctf*` output is created because, even though frame- and dose-weighting improves the SNR, it may degrade the contrast transfer function (CTF) signal and this file will therefore later be used to determine the CTF parameters of the integrated movie. The `*intg*` image is the one you will use for particle picking and extract your identified individual particle images from. The *pspec* and *thumb* outputs are diagnostic. Please, check so that the Thon rings are concentric and have similar radial intensity distribution in any given resolution shell after alignment (right part of power-spectrum)—if not, trash the image. Please, execute:

```
@!#> cd ~/workshop/SIMPLE
@!#> ls
1_DDD-movie-preproc/ 2_PRIME2D/           2_PRIME3D/
```

You see that we have one directory for each step. Each folder contains a subfolder `data` that contains the experimental cryo-EM data that we will process in addition to its associated parameters.

```
@!#> cd 1_DDD-movie-proc/
@!#> ls
data/
@!#> ls data/
info.txt   movie1.mrc movie2.mrc
```

The `info.txt` file lists the parameters associated with the data.

```
@!#> cat ./data/info.txt
molecule: proteasome
exp_time=7.6s
dose_rate=7e/A2/s
kv=300
cs=2.7
smpd=5.26
fraca=0.1
```

These movies are of proteasome molecules in ice, exposed for `exp_time=7.6` s with a dose rate of `dose_rate=7` e/A2/s. The electron microscope used to acquire these images has an acceleration voltage of `kv=300` kV and a spherical aberration constant of `cs=2.7` mm. We have downscaled these images for more rapid processing to a sampling distance of `smpd=5.26` A and we will assume 10% amplitude contrast when fitting the CTF `fraca=0.1`. First, we need to create a file table, listing the movies to be processed with `unblur_movies`.

```
@!#> ls data/movie* > movies.txt
```

Next, we execute the program.

```
@!#> simple_exec prg=unblur_movies filetab=movies.txt smpd=5.26 nthr=8
fbody=proteasome exp_time=7.6 dose_rate=7 kv=300
```

and we expect to see the following output in the terminal

```
>>> DONE PROCESSING PARAMETERS
>>> DONE BUILDING GENERAL TOOLBOX
>>> PROCESSING MOVIE:      1
>>> READING AND FOURIER TRANSFORMING FRAMES
  100%  |==================================================| done.
>>> WEIGHTED AVERAGE-BASED REFINEMENT
This % of frames improved their alignment:  100.
This % of frames improved their alignment:  100.
This % of frames improved their alignment:  100.
This % of frames improved their alignment:  100.
This % of frames improved their alignment:   47.
>>> LOW-PASS LIMIT UPDATED TO: 12.6667
This % of frames improved their alignment:    8.
```

```
>>> LOW-PASS LIMIT UPDATED TO: 10.3333
This % of frames improved their alignment:    8.
This % of frames improved their alignment:   53.
This % of frames improved their alignment:   42.
>>> LOW-PASS LIMIT UPDATED TO:  8.0000
This % of frames improved their alignment:    3.
This % of frames improved their alignment:    8.
>>> AVERAGE WEIGHT     :        0.0263
>>> SDEV OF WEIGHTS    :        0.0048
>>> MIN WEIGHT         :        0.0120
>>> MAX WEIGHT         :        0.0327
 50. percent of the movies processed
```

The first thing that happens is that all individual movie frames are read in and Fourier transformed. The algorithm is iterative and based on registration of the individual frames to a weighted average, where the weights are determined based on the agreement between the frames and the average. The algorithm automatically updates the resolution limit as the alignment accuracy improves, *i.e.* using only low-resolution Fourier components when the alignment errors are large and large movements need to be identified. It then successively updates the resolution limit as the alignment accuracy improves. When no improvements can be identified, the procedure stops. The average, standard deviation, minimum and maximum frame weight in the image series is reported upon completion of the alignment. The outputted image files are:

```
@!#> ls
data/                     proteasome_intg1.mrc    proteasome_thumb1.mrc
movies.txt                proteasome_intg2.mrc    proteasome_thumb2.mrc
proteasome_forctf1.mrc  proteasome_pspec1.mrc
proteasome_forctf2.mrc  proteasome_pspec2.mrc
```

The first images we inspect are the power spectra. For visualisation, we use `e2display.py`. Execute `e2display.py` in the current working directory, click on one of the `*pspec*` files and then `Show 2D`.

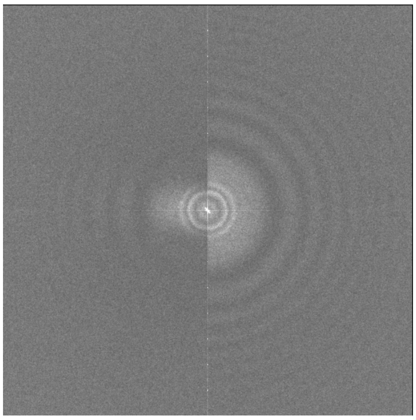

Figure 1: The left part of the power spectrum represents the movie integrated without applying the alignment parameters, whereas the right part is the aligned integrated movie. We see that the motion correction restores coherence, giving radially symmetric Thon rings extending to much higher resolution.