

NovaCTF

(A very brief tutorial)

NovaCTF is a freeware for 3D-CTF correction for electron microscopy, as described in this publication:

Turoňová, B., Schur, F.K.M, Wan, W. and Briggs, J.A.G. Efficient 3D-CTF correction for cryo-electron tomography using NovaCTF improves subtomogram averaging resolution to 3.4 Å.
doi.org/10.1016/j.jsb.2017.07.007

The source code for NovaCTF is distributed under an GPL v.3 license.

This very brief tutorial corresponds to version 1.0 of NovaCTF. For regular updates, FAQ, bug issues and fixes visit github page:

<https://github.com/turonova/novaCTF>

Install & Dependencies:

The program requires path to the include files from fftw3 as well as following fftw3 libraries:
fftw3 fftw3f

To compile the program in release mode run:

```
make includepath="path_to_fftw_include_files" libpath="path_to_fftw_libraries"
```

Example:

```
make includepath="/g/software/linux/pack/fftw-3.3.4/include" libpath="/g/software/linux/pack/fftw-3.3.4/lib /g/software/linux/pack/fftw-3.1.2/lib64"
```

Note #1: Typically both libraries should be in one folder and thus one library path should be sufficient (in case of EMBL fftw3f was compiled only for older versions of fftw and had to be thus linked separately).

Note #2: noveCTF also requires standard libraries with the standard path being /usr/lib64 – if the libraries are elsewhere open makefile and change the path on 10th line accordingly.

3D CTF Correction pipeline:

1. Generate defocus files:

Input: One file with defocus values and defocus step size in nm

Output: Number (N) of files with shifted defocus values (shifts correspond to chosen defocus step size)

The output files have the same name as input file with number attached to the end:

```
setfocus.txt -> setfocus.txt_0 setfocus.txt_1 ... setfocus.txt_(N-1)
```

The files have the same formatting as the input file.

Steps 2-4 have to be done for every defocus file generated in step 1:

2. CTF correction:

Input: Uncorrected tilt-series (does not have to be aligned) and one defocus file.

Output: Corrected tilt-series.

3. Additional processing of tilt-series, i.e. alignment, gold bead removal, taper...

Important: Prior next step, the tilt-series has to be flipped:

`clip flipyz input.ali output.ali`

4. Filtering

Input: Corrected tilt-series (have to be aligned and flipped at this point).

Output: Filtered tilt-series.

5. Reconstruction

Input: All CTF-corrected, aligned and filtered tilt-series.

Naming: `stack.ali_0`, `stack.ali_1` ... `stack.ali_(n-1)`

Output: Final tomogram.

Parameters

Most of the parameters used within NovaCTF are identical to those needed for corresponding functions in IMOD package (see parameters of `tilt`, `newstack`, `ctfphaseflip` for details). Similar to IMOD, parameters can be passed on command line or using text file (similar or the same as IMOD's `tilt.com`) or combination of both. In the latter case, the command line parameters have higher priority, i.e. you can override the parameters in `tilt.com` file by passing the same parameter on command line. Parameters listed in the following text show the minimal requirements for running given algorithms, more parameters can be used (see e.g. `tilt` parameters from IMOD). The parameters that are not recognized by the algorithm are simply ignored.

The example setup files for each of the algorithms implemented within NovaCTF are provided in `setup_examples` folder.

1. Generate defocus files:

Requires following parameters:

Algorithm defocus

InputProjections `input_stack.st`

FULLIMAGE 464 464

THICKNESS 140

TILTFILE `angles.tlt`

SHIFT 0.0 0.0

CorrectionType `phaseflip` | `multiplication`

DefocusFileFormat `ctffind4` | `imod`

DefocusFile defocus_file.txt
PixelSize 0.135 (in nm)
DefocusStep 15 (in nm)

Optionally:

CorrectAstigmatism 1 | 0
DefocusShiftFile file_with_additional_defocus.txt

Note #1: The input tilt-series and volume dimensions are necessary to properly set up the geometry for creating defocus matrix.

Note #2: The shift should be set to zero even if for reconstruction we want to shift the tomogram in z!!! We assume the defocus to be estimated at the center of mass which should correspond to the shifted tomogram and thus here the shift should be zero.

Note #3: For the specification on imod/ctffind4 formats see man pages of respective applications.

Note #4: Additional defocus shift is currently passed in a text file and is in voxels!!! If center of my tomogram in z is 70 (the tomogram thickness is 140 in voxels) and shift of -10 voxels is desired, the file with additional defocus will contain value 60.

Run:

```
./novaCTF -Algorithm defocus -InputProjections input_stack.st -FULLIMAGE 464,464 -THICKNESS 140  
-TILTFILE angles.tlt -SHIFT 0.0,0.0 -CorrectionType phaseflip -DefocusFileFormat ctffind4  
-CorrectAstigmatism 1 -DefocusFile defocus_file.txt -PixelSize 0.135 -DefocusStep 15 -DefocusShiftFile  
file_with_additional_defocus.txt
```

Or:

```
./novaCTF -param setup_defocus.com
```

2. CTF Correction:

Algorithm ctfCorrection

InputProjections input_stack.st
DefocusFile defocus_file.txt_#n
OutputFile corrected_stack.st_#n

TILTFILE angles.tlt

CorrectionType phaseflip | multiplication
DefocusFileFormat ctffind4 | imod

PixelSize 0.135 (in nm)
AmplitudeContrast 0.07
Cs 2.7
Volt 300

Optionally:

CorrectAstigmatism 0 | 1

Run:

```
./novaCTF -Algorithm ctfCorrection -InputProjections input_stack.st -OutputFile corrected_stack.st_#n  
-DefocusFile defocus_file.txt_#n -TILTFILE agnles.tlt -CorrectionType phaseflip -DefocusFileFormat  
ctffind4 -CorrectAstigmatism 1 -PixelSize 0.135 -AmplitudeContrast 0.07 -Cs 2.7 -Volt 300
```

Or:

```
./novaCTF -param setup_ctfCorrection.com
```

Note #1: This has to be run on all input stacks – can be parallelized if necessary. One has to be careful with naming – the output stacks has to be numbered the same way as the defocus files generated in the previous step.

3. Additional processing

Before the reconstruction and filtering the stacks have to be aligned and flipped (i.e. before filtering using NovaCTF and reconstruction the XYZ orientation has to be changed to XZY):

```
clip flipyz corrected_stack.ali_#n corrected_stack_flipped.ali_#n
```

4. Filtering

Algorithm filterProjections

InputProjections corrected_stack_flipped.ali_#n

OutputFile filtered_stack.ali_#n

TILTFILE angles.tlt

StackOrientation xz

Optionally:

RADIAL 0.3 0.05

Note #1: For meaning of RADIAL parameter see IMOD's tilt man page.

Note #2: At the time we implemented novaCTF, IMOD was offering only radial filtering and so currently this is the only filtering option we provide. In IMOD version 4.9.2 and higher, there are more filters offered for reconstruction – if you want to use them, you can used IMOD directly to filter the input

stacks, just bear in mind you have to keep the numbering convention and you have to have XZY orientation of your stacks before you proceed with reconstruction.

Note #3: The StackOrientation parameter specifies the rotation of the input stack – since we flipped it before, we have to specify xz, the default value is xy for a non-flipped stack. We can filter non-flipped stack as well and flip afterwards (for the reconstruction it has to be flipped) but that does not correspond to the IMOD way of filtering in the reconstruction process.

Run:

```
./novaCTF -Algorithm filterProjections -InputProjections corrected_stack_flipped.ali_#n  
-OutputFile filtered_stack.ali_#n -TILTFILe angles.tlt -StackOrientation xz -RADIAL 0.3,0.05
```

Or:

```
./novaCTF -param setup_filter.com
```

5. Reconstruction

Algorithm 3dctf

InputProjections filtered_stack.ali (**no number here!!!**)

OutputFile tomogram.rec

TILTFILe angles.tlt

THICKNESS 140

FULLIMAGE 464 464

SHIFT 0.0 -20.0

PixelSize 0.135 (in nm)

DefocusStep 15 (in nm) **or** NumberOfInputStacks N

Optionally:

Use3DCTF 0 | 1 (default is 1)

Note #1: If no 3D-CTF is desired, i.e. we have only one input stack, we set InputProjections to its name and set Use3DCTF to 0. Otherwise, we put the base name **without the number and without the underscore** as InputProjections and the program will add the numbers automatically.

Note #2: The number of input stacks can be specified either in the same way as in the first step, i.e. by giving defocus step in nm, or since we already know how many tilt-series we have as NumberOfInputStacks.

Run:

```
./novaCTF -Algorithm 3dctf -InputProjections filtered_stack.ali -OutputFile tomogram.rec -TILTFILE  
angles.tlt -THICKNESS 140 -FULLIMAGE 464,464 -SHIFT 0.0,-20.0 -PixelSize 0.135 -DefocusStep 15
```

Or:

```
./novaCTF -param setup_reconstruction.com
```