Markerfree: GPU-accelerated robust marker-free alignment for high-resolution cryo-electron tomography

Contents

S1 Overview **2**

S2 Installation **2**

S2.1 Prerequisites 2

S2.2 Installtion of Markerfree 4

S3 Explanation of parameters and useful examples **4**

S3.1 Parameter explanation of Markerfree 4

S3.2 Explanation of the content in the output folder 5

S3.3 The number of projections 6

S3.4 Examples 7

S1 Overview

We introduce Markerfree, a novel, fully automated software designed for fiducial marker-free alignment. Our approach prioritizes accuracy, robustness, and complete automation to streamline the alignment process. The software first performs pixel value adjustments to eliminate negative values and normalize image luminance across different tilt angles, ensuring consistent image quality. Next, it employs a hybrid strategy combining cross-correlation and common-line methods for coarse alignment, effectively reducing large deviations in translation and rotation parameters to facilitate precise global alignment. Finally, we perform global alignment. This process begins with correcting the tilt angles, followed by iterative optimization of translation and rotation parameters using projection matching and common-line methods. The accuracy of projection matching is further enhanced through filtering and weighting of the projection images. This comprehensive workflow ensures high-precision alignment while maintaining full automation.

This manual introduces the instructions for installation (see section S2), the detailed explanation of software parameters and several useful examples(see section S3).

S2 Installation

The sections below explain how to download and install Markerfree on your computer.

S2.1 Prerequisites

Note that Markerfree depends on and uses several external programs and libraries.

- **Linux or Unix-like operating systems**
- **GCC (version $\geq$ 8.4)**

```
sudo apt install gcc g++
```

- **CMake (version $\geq$ 3.10)**

```
sudo apt-get install cmake
```

- **MPI (OpenMPI version ≥ 3.1)**

```
wget https://download.open-mpi.org/release/open-mpi/v3.1/openmpi-3.1.6.tar.gz
tar -xzf openmpi-3.1.6.tar.gz
cd openmpi-3.1.6
./configure --prefix=/usr/local
make -j$(nproc)
sudo make install
export PATH=/usr/local/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

- **CUDA (version ≥ 11.1)**

```
sudo apt install -y cuda-11-1
echo 'export PATH=/usr/local/cuda-11.1/bin:$PATH' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=/usr/local/cuda-11.1/lib64:$LD_LIBRARY_PATH' >> ~/.bashrc
source ~/.bashrc
```

S2.2 Installtion of Markerfree

We store the public release versions of Markerfree on GitHub, a site that provides code-development with version control and issue tracking through the use of git. We will not describe the use of git in general, as you will not need more than very basic features. Below we outline the few commands needed on a UNIX-system, please refer to general git descriptions and tutorials to suit your system. To get the code, you clone or download the repository. We recommend cloning, because it allows you very easily update the code when new versions are released. To do so, use the shell command-line:

- **Download**

```
1 git clone https://github.com/icthrm/Markerfree.git
```

- **Compilation**

```
1 cd ./Markerfree/
2 mkdir build
3 cd ./build
4 cmake ..
5 make -j16
```

S3 Explanation of parameters and useful examples

S3.1 Parameter explanation of Markerfree

Markerfree: command line arguments

==== Required options ====

- -INPUT (-i) <input_filename>: Input MRC file for reconstruction.

  -INPUT input.mrc or -i input.mrc

- -OUTPUT (-o) <output_filename>: Output MRC filename for results.

  -OUTPUT output.mrc or -o output.mrc

- -TILEFILE (-a) <tilt_angle_file>: Tilt angle filename.

  -TILEFILE angles.rawtlt or -a angles.rawtlt

- -GEOMETRY (-g) <seven_integers>: Geometry information: offset, tilt axis angle, z-axis offset, thickness, projection matching reconstruction thickness, output image downsampling ratio, GPU ID (default: 0 if only one GPU is available).

  -GEOMETRY 0,30,0,100,200,1,0 or -g 0,30,0,100,200,1,0

- -NPROJ (-p) <Number of projections>: The number of images used during the projection matching phase defaults to 10. Users can adjust this setting as needed, though the default value suffices in most cases. For users seeking greater accuracy, we provide a solution to simulate the optimal number of images for their specific dataset in S3.3.

- -Savemode (-s) <save_format>:

  * -s 0: Save parameters as txt format (tilt axis angle and translation parameters).

  * -s 1: Save parameters as xf file format (affine transformation matrix).

- -help (-h): Display help information, no extra argument needed.

We have supplied examples of the input file format on Github (https://github.com/icthrm/Markerfree.0.git), which users can downloaded and viewed.

S3.2 Explanation of the content in the output folder

- 'BBb_fin.xf'

  – Transformation matrix file defining the linear transformation (affine or rigid-body) to be applied to the image data.

- 'BBb_fin.txt'

  – Parameter file (text format) containing translation (x, y) and rotation angle values for geometric transformation.

- `'BBb_fin.mrc'`
  - The final aligned tilt series.

S3.3 The number of projections

Although 10 projections performed consistently well across our systematic simulation studies, we fully agree that this value may shift for different acquisition schemes. To make our approach broadly applicable, we provide a general and extensible workflow that allows users to determine the appropriate number of projections for their own dataset:

Step 1. Dataset-specific structural simulation. Clean 3D phantoms are generated using PolNet (Martinez-Sanchez *et al.*, 2024). These phantoms are configured to match the structural characteristics of the target dataset, including membrane-like or filamentous geometries.

Step 2. Noise characterization from raw data. To extract realistic noise statistics, NT2C (Li *et al.*, 2022) is applied to the raw micrographs:

(a) A coarse CNN denoiser (developed in-house) enhances the raw images.

(b) The enhanced images are divided into overlapping patches.

(c) Local SSIM is computed for each patch.

(d) High-SSIM patches are identified as pure-background regions.

(e) The corresponding raw patches are collected to form a noise-sample library representing the dataset's intrinsic noise distribution.

Step 3. Incorporation of imaging parameters. To ensure physical accuracy, the forward simulation incorporates imaging parameters extracted from the dataset, including: CTF, defocus, total electron dose and dose-dependent attenuation, detector MTF/NNPS, and pixel size. These parameters are applied to the clean projections prior to noise synthesis.

Step 4. Realistic noise synthesis and transfer. The GAN module in NT2C learns the statistical distribution of the extracted noise samples. A contrast-guided reweighting mechanism transfers this learned noise onto the CTF-modulated clean projections, generating synthetic tilt series that replicate both the structural content and noise characteristics of the real dataset.

Step 5. Calculate the optimal result. A set of high-precision computation codes is available on GitHub to demonstrate alignment residuals for datasets containing fiducial markers.

S3.4 Examples

When using Markerfree for alignment, the required user inputs include the initial micrograph, its tilt angle, output paths for aligned images, geometry parameters, and the save mode for rotation and translation files.

Other parameters are optional and users can choose according to their needs (see subsection S3.1). The following are serveral useful examples of input from the command line:

The software will process the tilt series alignment with default parameters: no manual geometry adjustments, no image downsampling, using primary GPU (ID 0), and save rotation and transformation parameters in text format.

```
./Markerfree -i BBb.st -o BBb_ali.mrc -a BBb.rawtlt -g 0,0,0,0,300,1,0 -s 0
```

The software will process the tilt series alignment with default parameters: no manual geometry adjustments, no image downsampling, using primary GPU (ID 0), and save rotation and transformation matrices in xf format.

```
./Markerfree -i BBb.st -o BBb_ali.mrc -a BBb.rawtlt -g 0,0,0,0,300,1,0 -s 1
```

The software will process the tilt series alignment with default parameters: no manual geometry adjustments, apply 2x downsampling to output, using primary GPU (ID 0), and save rotation and transformation matrices in txt format.

```
./Markerfree -i BBb.st -o BBb_ali.mrc -a BBb.rawtlt -g 0,0,0,0,300,2,0 -s 0
```

References

Li, H., Zhang, H., Wan, X., Yang, Z., Li, C., Li, J., Han, R., Zhu, P., and Zhang, F. (2022). Noise-transfer2clean: denoising cryo-em images based on noise modeling and transfer. *Bioinformatics*, **38**(7), 2022–2029.

Martinez-Sanchez, A., Lamm, L., Jasnin, M., and Phelippeau, H. (2024). Simulating the cellular context in synthetic datasets for cryo-electron tomography. *IEEE Transactions on Medical Imaging*, **43**(11), 3742–3754.