Beginner's Guide to THUNDER

Mingxu Hu (hmx13@mails.tsinghua.edu.cn)
Xueming Li's Lab

Method Article

Keywords: cryoEM, image processing, THUNDER

Posted Date: July 16th, 2018

DOI: https://doi.org/10.1038/protex.2018.060

Abstract

Electron cryo-microscopy \(cryoEM) is now a powerful tool in determining atomic structures of biological macromolecules under nearly natural conditions. THUNDER is a particle-filter algorithm based cryoEM image processing software. Here, we describe a protocol for using THUNDER to analysis cryoEM images in purpose of achieving a 3D model. A JSON file is used for setting parameters in THUNDER. Meaning of each attribute is discussed in this protocol. The .thu file format is defined for storing information of each particle image, including CTF parameters, rotation, translation, defocus adjustment and grading weight. The definition of this .thu file is introduced in this protocol. Finally, the procedure of running THUNDER and examining the output of THUNDER is contained in this protocol.

Introduction

Electron cryo-microscopy \(cryoEM) is now a powerful tool in determining atomic structures of biological macromolecules under nearly natural conditions. In single particle cryoEM, micrographs, each of which contains hundreds of single particles, are captured by cameras. Using softwares like CTFFind or GCTF, CTF parameters are determined. Moreover, by particle picking, single particle images are extracted from micrographs. Here, THUNDER uses single particle images to obtain a 3D model of the target macromolecule. THUNDER needs several files for running, as shown in Figure 1. It parses a JSON parameter file for obtaining parameters of THUNDER. This JSON parameter file contains the crucial information of running, including whether it is refinement, 2D classification or 3D classification, and the path of .thu file and initial model. A typical workflow of THUNDER contains several steps. Firstly, a 2D classification is performed to discard the obvious bad particles. Secondly, a 3D classification is performed to analysis the homogeneity of the dataset. By selecting the good class\(es) from the result of 3D classification, 3D model of the targeted macromolecule can be obtained by refinement.

Equipment

Hardware Requirement A computer cluster with shared storage is recommended for running THUNDER on. MPI environment should be set up on this computer cluster. Moreover, if multiple users share this computer cluster, a job scheduler, like LSF, is highly recommended. If you want THUNDER runs on a GPU, GPUs with CUDA compatibility 3.0 or more are required. **Software Requirement** _C/C++ compiler supporting C++98 standard along with MPI wrapper_ _cmake_ We recommend gcc and Intel C/C++ compiler as C/C++ compiler. Moreover, gcc42 has been tested as the oldest supporting version of gcc. OpenMPI and MPICH both can be used as MPI standard. In Tsinghua, we use openmpi-gcc43 as the C/C++ compiler for compiling THUNDER. cmake is a tool for configuring source code for installation. openmpi-gcc43 is open-source software, which can easily installed using yum on CentOS and apt-get on Ubuntu. cmake has been already installed in most Linux operating systems. If not, it can also be conveniently installed by yum on CentOS and apt-get on Ubuntu. CUDA 8 or higher version of CUDA is required for installing GPU version of THUNDER. **Get THUNDER Source Code and Install THUNDER** THUNDER is a open-source software package, source code of which is held on Github. You may download the source code at https://github.com/thuem/THUNDER. You may compile and install THUNDER under the tutorial in the manual directory of this package.

Procedure

JSON Parameter File Preparation The JSON parameter file contains the parameter of THUNDER, where typical JSON parameter file is shown as Figure 2. It follows JSON file standard. The meaning of each attribute of the JSON parameter file is listed below. _Number of Threads Per Process_ It stands the number of threads used in each MPI process. Usually, THUNDER hosts one process in each node, for the purpose of minimizing the usage of memory by sharing memory property of threading. As only one process hosted in each node, occupying all cores of CPU\(s) in each node is accomplished by threads. This value should be set to the number of cores in each node, in this circumstance. _2D or 3D Mode_ It stands for whether THUNDER acts in 2D or 3D. This attributes only takes string 2D or 3D, for setting THUNDER in 2D mode or 3D mode, respectively. If the user desires to perform 2D classification, this attribute should be set to 2D. On the other hand, if the user aims to perform 3D refinement or 3D classification, this attribute should be set to 3D. _Global Search_ It stands for whether THUNDER performs global search or not. _Local Search_ It stands for whether THUNDER performs local search or not. _CTF Search_ It stands for whether THUNDER refines defocus of each single particle or not. _Number of Classes_ It stands for the number of classes the user desires that the dataset classified into. For refinement, this attribute is set to 1. _Pixel Size \(Angstrom)_ It stands for the size of each pixel in Angstrom. _Radius of Mask on Images \(Angstrom)_ It stands for the radius of mask, which will be masked on each single particle image. This attribute should be set to slightly larger than the radius of the targeted macromolecule. _Estimated Translation \(Pixel)_ It stands for the standard deviation of in-plane translation of single particle images. As the translation is unknown before running THUNDER, this attribute is estimated by the user. Larger this attribute is, the wider THUNDER will search in in-plane translation, the longer computing time will be. _Initial Resolution \(Angstrom)_ It stands for the resolution THUNDER starts its iteration. If the user aims to eliminate the bias of initial model, he or she should set this attribute to a low resolution. _Perform Global Search Under \(Angstrom)_ It stands for the resolution threshold where search type switches from global search to local search. _Symmetry_ It stands for symmetry type of the macromolecule. For example, C5 stands for 5-fold rotation symmetry along Z axis; D7 stands for 7-fold rotation symmetry along Z axis and 2-fold rotation symmetry along X axis; T stands for tetrahedral symmetry; O stands of octahedral symmetry; I stands for icosahedral symmetry. If the macromolecule is asymmetrical, it should be set to C1. _Initial Mode_ It stands for the path where THUNDER reads in the initial model. _.thu File Storing Paths and CTFs of Images_ It stands for the path where THUNDER reads in the .thu File. If THUNDER starts from global search, the path and CTF information of each single particle image will be read in. If THUNDER starts from local search or CTF search, besides path and CTF information, the rotation and translation information of each single particle image will be read in as well. _Prefix of Particles_ It stands for a prefix added before the path of each single particle image, in order to help THUNDER find the correct path of image. _Prefix of Destination_ It stands for a prefix added before the path of output file. It helps THUNDER redirects the output files.

Calculate FSC Using Core Region It stands for whether THUNDER calculates FSC of the reference\(s) masked by a sphere or not. _Calculate FSC Using Masked Region_ It stands for whether THUNDER calculates FSC of the masked region of the reference\(s), where the mask is provided by the user. _Particle Grading_ It stands for whether THUNDER uses particle grading to weight the contribution of each single particle image during reconstruction. The JSON parameter file also contains more parameters. However, those parameters are designed for professional users, as beginners can just use default values. Users can find a full description of parameters in JSON parameter file in THUNDER's website. **.thu File Format** THUNDER uses .thu file for inputing and output information of each image, including CTF parameters, classification, rotation, translation, defocus adjustment and grading weight. .thu file is a simple space-separate plain text file with .thu as suffix. Each column of it stands for an attribute, as listed below. 1. voltage of the electron microscope \(in volt) 2. first defocus value defined in CTFFind[1] 3. second defocus value defined in CTFFind 4. theta of defocus defined in CTFFind 5. spherical aberration in Angstrom 6.amplitude constrast 7. phase shift in radian 8. path of the single particle image 9. path of the micrograph which the single particle image belong to 10. coordinate X in micrograph 11. coordinate Y in micrograph 12. group ID 13. class ID 14. 1st element of the unit quaternion for rotation 15. 2nd element of the unit quaternion for rotation 16. 3rd element of the unit quaternion for rotation 17. 4th element of the unit quaternion for rotation 18. 1st standard deviation of rotation 19. 2nd standard deviation of rotation 20. 3rd standard deviation of rotation 21. translation in X 22. translation in Y 23. standard deviation of translation in X 24. standard deviation of translation in Y 25. defocus factor to adjust defocus 26. standard deviation of defocus factor 27. particle grading score **Running THUNDER** The only file THUNDER directly reads in is the JSON parameter file. By parsing it, THUNDER will fetch .thu file for information of each particle and read in the initial model. Further, THUNDER will try to obtain each image by the path described in .thu file. Those files should should be prepared before running THUNDER. THUNDER is a multiple-process software, where at least 3 processes should be assigned for it. Typically, one process should be hosted on each node. Threading is used for occupying all cores in each node. Multiple processes should be governed by MPI environment. Users are able to use _mpirun_ or _mpiexec_ tool run MPI jobs of THUNDER. If a job scheduler is available on the cluster, users can entrust it with THUNDER MPI jobs.

Anticipated Results

The running status of THUNDER will be output as a log format in file _thunder.log_. The reference\(s) will be output as file\(s) as _Reference_XXX_A_Round_XXX.mrc_ and _Reference_XXX_B_Round_XXX.mrc_ during 3D refinement, as the two half maps. Meanwhile, during 3D classification, the average maps of the two half maps will be output as _Reference_XXX_Round_XXX.mrc_. Finally, during 2D classification, the average maps will be output in an image stack as _Reference_XXX_Round_XXX.mrcs_. Classification, rotation, translation and defocus adjustment of each particle image will be saved in a .thu file in each iteration, as file _Meta_Round_XXX.thu_, the format of which has been introduced above. FSC\(FRC) curve of all reference\(s) in each iteration will be output as _FSC_Round_XXX.txt_. The first column of it stands for the frequency in pixel, and the second column of it stands for the frequency in Angstrom. The following column\(s) stand for FSC\(FRC) curves for each reference, respectively. Classification information will be summarized and output in file _Class_Info_Round_XXX.txt_. The first, second and third column of it stands for the index, resolution and the percentage of images belonged of each class.

References

1. Rohou, A, and N. Grigorieff. "CTFFIND4: Fast and accurate defocus estimation from electron micrographs." Journal of Structural Biology 192.2\(2015):216-221.

Figures

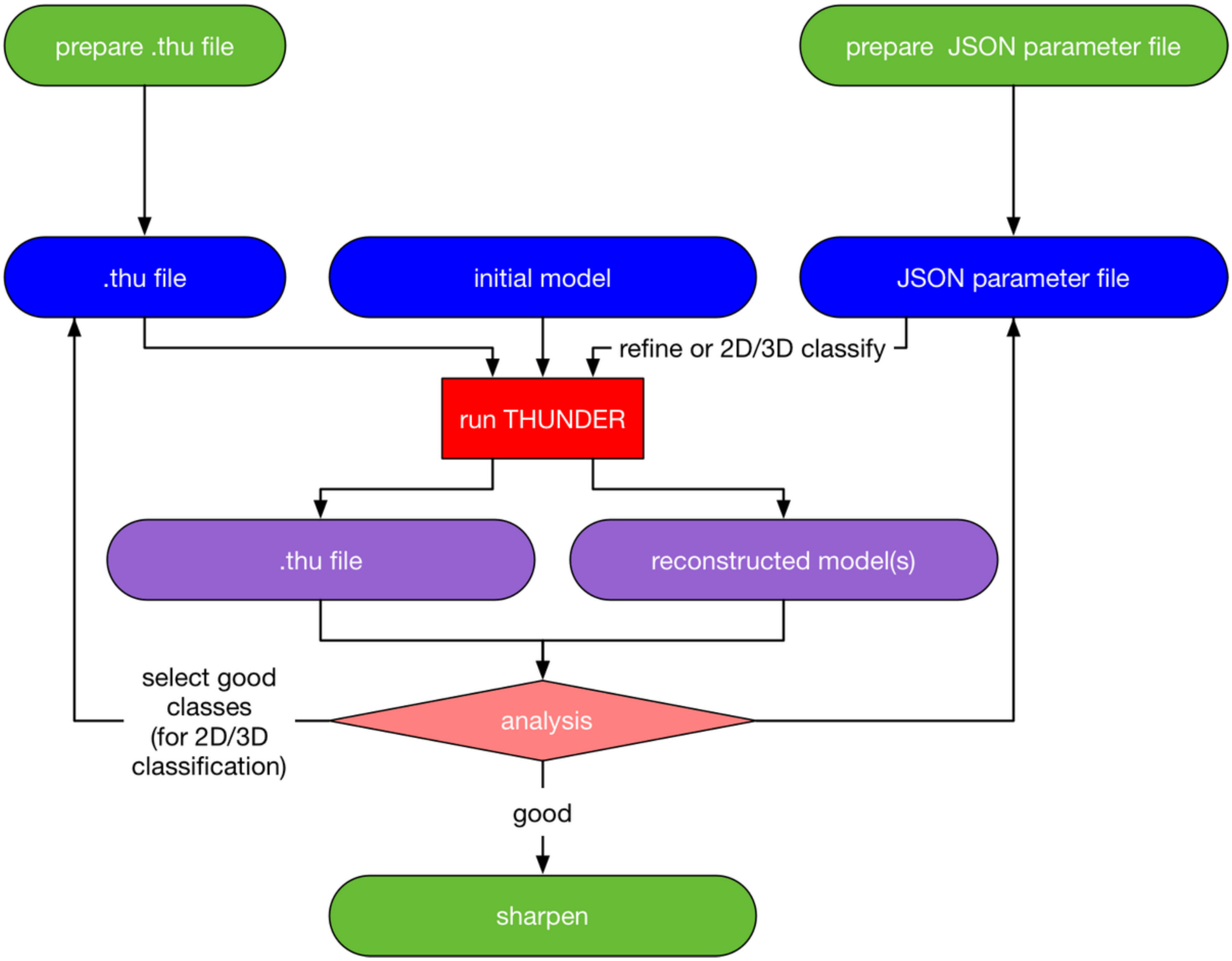
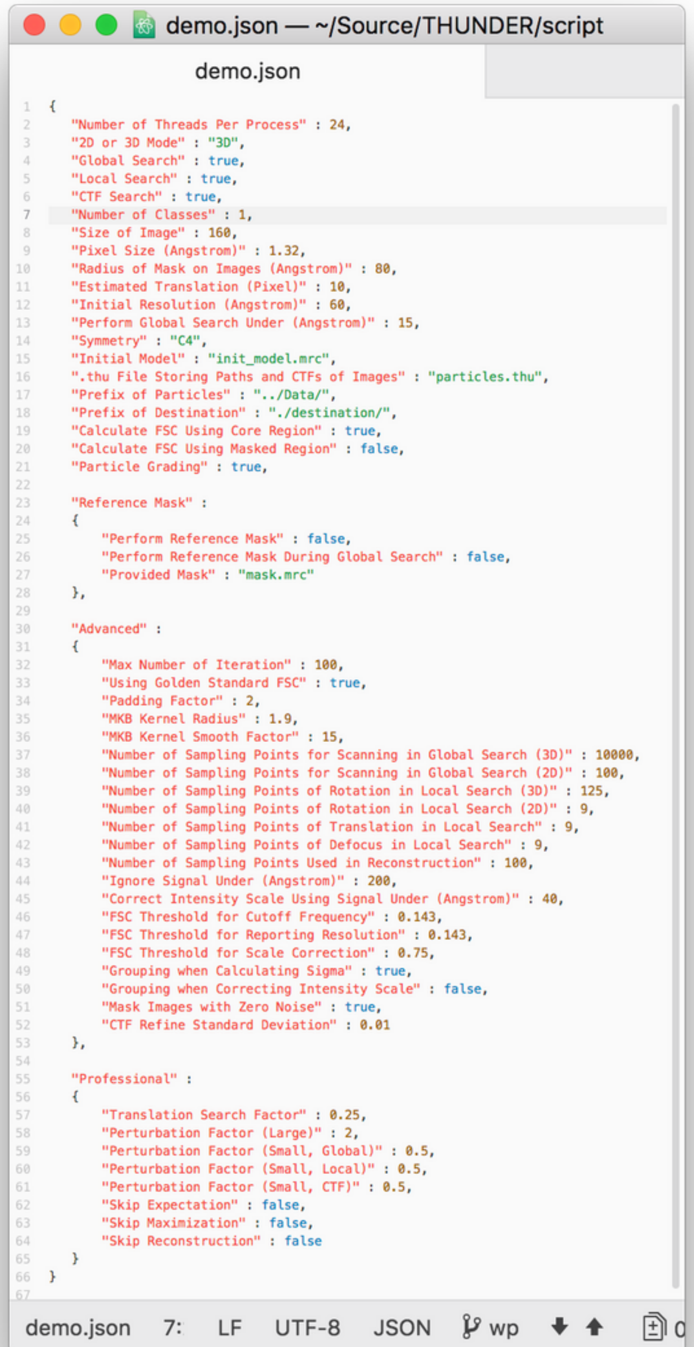


Figure 1

Flow Diagram of THUNDER




```
{
    "Number of Threads Per Process" : 24,
    "2D or 3D Mode" : "3D",
    "Global Search" : true,
    "Local Search" : true,
    "CTF Search" : true,
    "Number of Classes" : 1,
    "Size of Image" : 160,
    "Pixel Size (Angstrom)" : 1.32,
    "Radius of Mask on Images (Angstrom)" : 80,
    "Estimated Translation (Pixel)" : 10,
    "Initial Resolution (Angstrom)" : 60,
    "Perform Global Search Under (Angstrom)" : 15,
    "Symmetry" : "C4",
    "Initial Model" : "init_model.mrc",
    ".thu File Storing Paths and CTFs of Images" : "particles.thu",
    "Prefix of Particles" : "../Data/",
    "Prefix of Destination" : "./destination/",
    "Calculate FSC Using Core Region" : true,
    "Calculate FSC Using Masked Region" : false,
    "Particle Grading" : true,

    "Reference Mask" :
    {
        "Perform Reference Mask" : false,
        "Perform Reference Mask During Global Search" : false,
        "Provided Mask" : "mask.mrc"
    },

    "Advanced" :
    {
        "Max Number of Iteration" : 100,
        "Using Golden Standard FSC" : true,
        "Padding Factor" : 2,
        "MKB Kernel Radius" : 1.9,
        "MKB Kernel Smooth Factor" : 15,
        "Number of Sampling Points for Scanning in Global Search (3D)" : 10000,
        "Number of Sampling Points for Scanning in Global Search (2D)" : 100,
        "Number of Sampling Points of Rotation in Local Search (3D)" : 125,
        "Number of Sampling Points of Rotation in Local Search (2D)" : 9,
        "Number of Sampling Points of Translation in Local Search" : 9,
        "Number of Sampling Points of Defocus in Local Search" : 9,
        "Number of Sampling Points Used in Reconstruction" : 100,
        "Ignore Signal Under (Angstrom)" : 200,
        "Correct Intensity Scale Using Signal Under (Angstrom)" : 40,
        "FSC Threshold for Cutoff Frequency" : 0.143,
        "FSC Threshold for Reporting Resolution" : 0.143,
        "FSC Threshold for Scale Correction" : 0.75,
        "Grouping when Calculating Sigma" : true,
        "Grouping when Correcting Intensity Scale" : false,
        "Mask Images with Zero Noise" : true,
        "CTF Refine Standard Deviation" : 0.01
    },

    "Professional" :
    {
        "Translation Search Factor" : 0.25,
        "Perturbation Factor (Large)" : 2,
        "Perturbation Factor (Small, Global)" : 0.5,
        "Perturbation Factor (Small, Local)" : 0.5,
        "Perturbation Factor (Small, CTF)" : 0.5,
        "Skip Expectation" : false,
        "Skip Maximization" : false,
        "Skip Reconstruction" : false
    }
}
```


Figure 2

A Typical JSON Parameter File