

# ROME 1.1.2 Documentation

Copyright © 2017 | Intel® Parallel Computing Center for Structural Biology at Dana-Farber Cancer Institute, in association with Peking University.

## 1 How to install

### 1.1 Prerequisite

The Intel compiler is required for compiling ROME 1.1.2 code. To download Intel compiler, please visit : <https://software.intel.com/en-us/intel-parallel-studio-xe>

To obtain free Intel software tools, please visit : <https://software.intel.com/en-us/qualify-for-free-software>

### 1.2 Compiling and Installing

Uncompress the tarball containing the sources, and go to the Makefile directory. the rome execution binary file can be compiled by some compilers, such like icpc, mpiicpc, mpicxx.

To compiling with icpc

```
export ROME_CC=icpc
make
```

To compiling with mpicxx

```
export ROME_CC=mpicxx
make
```

To compiling with mpiicpc

```
export ROME_CC=mpiicpc
make
```

To compiling rome\_sml with OFFLOAD mode

```
export ROME_OFFLOAD=true
export ROME_CC=mpiicpc
make
```

To compiling rome\_map2d and rome\_map3d with single precision

```
export ROME_FLOAT=true
export ROME_CC=mpiicpc
```

make

The execution binary file will be generated on bin folder in current directory. type

```
./bin/rome_deep2d -help
```

to see how to use.

Notice:if you want to rebuild the code,you need to clean first

```
make clean
```

## 2 what's new in Version 1.1.2

We fixed some bugs and made general improvements to make ROME better for you. Please continue to send your feedback to help us improve this software in future updates.

- 1) Added Resmap to calculate local resolution
- 2) Added mask in rome2d and rome 3D
- 3) Select direction from prior angle
- 4) Fixed some input and output bugs
- 5) Fixed some OMP and memory bugs
- 6) Fixed a image read bug and a metadata access bug
- 7) Fixed some bugs in rome reconstruction
- 8) Escaped some local searching and only select non-zero probabilities while sorting, to speedup the 3D reconstructions
- 9) Removed ScaleCorrection output in star file, and added all relion metadata labels, to be compatible with relion.

## 3 List of Programs

### 3.1 rome\_map2d usage:

```
-i <metadata file>  
    Input metadata file with images needed to align  
-o <metadata file>  
    Output metadata  
-K <1>  
    Number of classes needed to classify  
-iter <100>  
    Maximum number of iterations to perform  
-angpix <1.0>  
    Pixel size (in Angstroms)  
-pool <50>  
    Number of images to be processed together for each EM step  
-random_seed <33>  
    Number of the random seed generator  
-offset_step <2>
```

Sampling rate (before oversampling) for origin offsets (in pixels)  
-offset\_range <10>  
Search range for origin offsets (in pixels)  
-psi\_step <10>  
Sampling rate (before oversampling) for the in-plane angle

## 3.2 rome\_sml usage:

-i <metadata file>  
Input metadata file with images needed to align  
-o <metadata file>  
Output metadata  
-K <1>  
Number of classes needed to classify  
-iter <100>  
Maximum number of iterations to perform  
-angpix <1.0>  
Pixel size (in Angstroms)  
-alpha <0.01>  
Value of the variance of prior model Gaussian distribution  
-updatebeta <1>  
Update the variance of noise whether or not, 0 or 1  
-precision <10e-12>  
The condition of GTM convergence  
-nummic <-1>  
The number of mic cards used to compute  
-loadmic <0.3>  
The percentage of job put to compute in mic card  
-weightedsum <1.0>  
Probability threshold (0~1) for weighted class averaging  
-search <0>  
Whether perform second round of classification based on GTM on each class or not

## 3.3 rome\_deep2d usage:

-i <metadata file>  
Input metadata file with images needed to align  
-o <metadata file>  
Output metadata  
-ml2d\_K <1>  
Number of classes needed to classify based on maximum likelihood method  
-sml\_K <1>  
Number of classes needed to classify based on GTM  
-ml2d\_iter <100>  
Maximum number of iterations to perform based on maximum likelihood method  
-sml\_iter <100>  
Maximum number of iterations to perform based on GTM  
-angpix <1.0>  
Pixel size (in Angstroms)  
-pool <50>  
Number of images to be processed together for each EM step

```
-random_seed <33>
    Number of the random seed generator
-offset_step <2>
    Sampling rate (before oversampling) for origin offsets (in pixels)
-offset_range <10>
    Search range for origin offsets (in pixels)
-psi_step <10>
    Sampling rate (before oversampling) for the in-plane angle
```

## 3.4 Useful tools

### 3.4.1 rome\_tool -classaverage

Compute class averaging from a given file,usage:

```
-i <metadata file>
    Input metadata file with images(*.star)
-o <metadata file>
    Output metadata
-K <1>
    Number of classes needed to classify
-angpix <1.0>
    Pixel size (in Angstroms)
-averageBeta <1>
    The variance of noise when doing weighted class averaging
-averageAlpha <0.01>
    The variance of prior model Gaussian distribution when doing weighted
class averaging
```

### 3.4.2 rome\_tool -convert

Convert image data file from given format (SPIDER form .dat or RELION form .mrcs) to formats (SPIDER form .dat or RELION form .mrcs) you want,usage:

```
-i <images stack>
    Input file name(*.mrcs or *.dat)
-o <images stack>
    Output file name(*.dat or *.mrcs)
```

### 3.4.3 rome\_tool -select

Gather images from plenty of images into one stack,usage:

```
-i <metadata file>
    Input metadata file with images(*.star)
-o <metadata file>
    Output metadata
```

### 3.4.4 rome\_tool -adjust

Shift and rotate images based on translations and rotation angle in star file,usage:

```
-i <metadata file>
    Input metadata file with images(*.star)
```

-o <metadata file>  
Output metadata

### 3.4.5 rome\_tool -applyfilter

Perform low-pass filtering, usage:

-i <image stack>  
Input file name (\*.mrcs)  
-o <image stack>  
Output file name (\*.mrcs)  
-filter  
Filter radius in frequency

### 3.5 rome\_map3d usage:

-i <metadata file>  
Input metadata file with images needed to align  
-o <metadata file>  
Output metadata  
-ref <\*.mrc>  
3D reference file  
-particle\_diameter  
Diameter of the circular mask that will be applied to the experimental images (in Angstroms)  
-K <1>  
Number of classes needed to classify  
-iter <100>  
Maximum number of iterations to perform  
-angpix <1.0>  
Pixel size (in Angstroms)  
-ini\_high  
Resolution (in Angstroms) to which to limit refinement in the first iteration  
-pool <50>  
Number of images to be processed together for each EM step  
-random\_seed <33>  
Number of the random seed generator  
-offset\_step <2>  
Sampling rate (before oversampling) for origin offsets (in pixels)  
-offset\_range <10>  
Search range for origin offsets (in pixels)  
-oversampling  
Adaptive oversampling order to speed-up calculations (0=no oversampling, 1=2x, 2=4x, etc)  
-healpix\_order  
Healpix order for the angular sampling (before oversampling) on the (3D) sphere: hp2=15deg, hp3=7.5deg, etc  
-tau2\_fudge  
Regularisation parameter (values higher than 1 give more weight to the data, 4 for 3D)  
-sym  
Symmetry group

## 3.6 rome\_reconstruct usage:

```
-i <metadata file>
    Input metadata file with images needed to align
-o <*.mrc>
    Name for output reconstruction
-angpix <1.0>
    Pixel size (in Angstroms)
```

## 3.7 rome\_res usage:

```
-i <metadata file>
    Input metadata file needed to calculate local resolution
-i1 <metadata file >
    Input half1 metadata file needed to calculate local resolution
-i2 <metadata file >
    Input half2 metadata file needed to calculate local resolution
-minRes <0.0>
    minimal resolution (A)
-maxRes <5.0>
    maximal resolution (A)
-vxSize <0.0>
    Voxel size of input map (A),
-stepRes <1.0>
    step size (A)
-variance <0.0>
    estimate variance
-pValue<0.05>
    P-value for likelihood ratio test
```

# 4 ROME GUI

ROME has a display program (called relion\_viewer). It could be launched from the command-line "python rome\_viewer.py".

## 4.1 Menu lists

- Open STAR File : open \*.star file for particle picking
- Open MRCS File : open \*.mrscs file for general view
- Save selected classes into STAR File : save selected classes into a \*.star File
- Save selected classes to TIFF File : save selected classes to \*.tiff files
- Select All : select all classes in current page
- Clear Selection : unselect all classes in current page
- Reverse Selection : reverse select classes in current page
- Show All Classes : show all classes in all pages

- Show Non-Empty Classes : only show non empty classes in all pages
- Sort Classes by Images Population : sort all classes by its images population
- Next Page : go to next page
- Previous Page : go to previous page
- First Page : go to first page
- Last Page : go to last page
- Each Page Classes Number : choice how many classes show in each  
page(100,300,500,1000)
- Scalable Viewing Mode : classes viewer window is scalable, so you can adjust the  
windows size
- Scrolled Viewing Mode : classes viewer window is fix, you can set the classes' photo  
size
- ImageViewer : using to view images belonging to each class

## 4.2 Particle picking

In particle picking mode, users should put ".star and .mracs" file that you got from folder "rome\_ml2d", "rome\_sml" or "rome\_deep2d" and the original input mracs data in same directory. Then open the "\*.star" file in rome\_viewer. Finally "ImageViewer" is needed to pick particle.

## 5 Users-created scripts to perform some task

### 5.1 Running the job of “rome\_map2d”

```
./bin/rome_map2d -i $input_fn -o $output_fn -K $map2d_classes -angpix  
$pixel_size -iter $map2d_iter  
-pool $nr_pool -offset_range $offset_range -offset_step $offset_step -  
psi_step $psi_step
```

### 5.2 Running the job of “rome\_sml”

```
./bin/rome_sml -i $input_fn -o $output_fn -K $sml_classes -angpix $pixel_size  
-iter $sml_iter  
-pool $nr_pool
```

### 5.3 Running the job of “rome\_deep2d”

```
./bin/rome_deep2d -i $input_fn -o $output_fn -map2d_K $map2d_classes -sml_K  
$deep2d_classes -angpix $pixel_size  
-map2d_iter $map2d_iter -sml_iter $sml_iter -pool $nr_pool -offset_range  
$offset_range -offset_step $offset_step  
-psi_step $psi_step
```

## 5.4 Running the job of “rome\_map3d”

```
./bin/rome_map3d -i $input_fn -o $output_fn -particle_diameter $dim -K  
$map3d_classes -angpix $pixel_size -K $nr_classes -ini_high $ini_high -iter  
$nr_iter -offset_range $offset_range -offset_step $offset_step  
-oversampling $oversampling -healpix_order $healpix_order -random_seed  
$random_seed -pool $nr_pool -tau2_fudge $tau2_fudge -sym C1 -zero_mask -  
flatten_solvent -norm -scale -firstiter_cc -ctf
```

## 5.5 Running the job of “rome\_reconstruct”

```
./bin/rome_reconstruct -o $output_fn -i $input_fn --sym C1 --angpix  
$pixel_size -ctf
```

## 5.6 Running the job of “rome\_res”

```
./bin/rome_res -res -i1 $mrc_file_1_name -i2 $mrc_file_2_name -minRes 10 -  
maxRes 20 -stepRes 1.0
```

## 5.7 Format conversion

```
./bin/rome_tool -convert -i $mracs_file_name -o $dat_file_name
```

## 5.8 Class averaging

```
./bin/rome_tool -classaverage -i $star_file_name -o $output_file_name -K  
class_number -angpix pixel_size
```

## 5.9 Low-pass filtering

```
./bin/rome_tool -applyfilter -i $mracs_file_name -o $output_file_name -filter  
filter_radius -angpix pixel_size
```

## 5.10 Rotate and shift images

```
./bin/rome_tool -adjust -i $star_file_name -o $output_file_name
```

# 6 Image I/O

ROME reads the following image file formats: MRC stacks (with extension .mracs) (this is the recommended image format) SPIDER individual images (with extension .spi) SPIDER stacks (with extension .spi). ROME writes individual images and image stacks



in MRC format. For SPIDER image format, users can use "rome\_tool" to convert SPIDER images or stacks to MRC format stacks. Individual images in stacks are indicated by an integer number (ranging from 1 to the number of images in the stack) followed by an "@" sign and the filename of the stack. For instance, the first three images in a stack file "test.mrcs" should read as:

```
1@test.mrcs
2@test.mrcs
3@test.mrcs
```

## 7 File formats and parameter conventions

We have inherited all the STAR conventions from RELION. ROME uses the STAR (Self-defining Text Archiving and Retrieval) format (Hall, Allen and Brown, 1991) for the storage of label-value pairs for all kinds of input and output metadata. The STAR format has been adopted by the crystallographic community in the form of CIF (Crystallographic Information Framework), and Bernard Heymann's BSOFT package was the first to use STAR in the field of 3D-EM. Also Xmipp-3.0 now uses the STAR format.

### 7.1 Metadata label definitions

#### **An example**

A STAR file that could be used as input for refinement in ROME that includes CTF information about each particle. A STAR input file should look like this:

```
data_
loop_
_rlnVoltage #1
_rlnDefocusU #2
_rlnDefocusV #3
_rlnDefocusAngle #4
_rlnSphericalAberration #5
_rlnAmplitudeContrast #6
_rlnImageName #7
200.000 38739.3 38739.3 0.00000 2.70000 0.100000 1@data1.mrcs
200.000 38739.3 38739.3 0.00000 2.70000 0.100000 2@data1.mrcs
200.000 38739.3 38739.3 0.00000 2.70000 0.100000 3@data1.mrcs
```

In each iteration of classification, ROME will write out orientations and class number of each particle into a STAR file. All other alignment parameters will be stored in this STAR file.

## 7.2 Orientations

### 7.2.1 In-plane rotation angle

The definition of in-plane rotation angle is in compliance with the Heymann, Chagoyen and Belnap (2005). Right-handed rotations are called positive. This is consistent with that is used in RELION.

### 7.2.2 Image center

When a 2D image is shifted and rotated, the center of this 2D image of dimensions  $x_{dim}$  x  $y_{dim}$  is defined by  $((int)x_{dim}/2, (int)(y_{dim}/2))$  (with the first pixel in the upper left being (0,0). Note that for both  $x_{dim}=y_{dim}=101$  and for  $x_{dim}=y_{dim}=100$ , the center will be at (50,50). This is the same convention as used in SPIDER and XMIPP. Origin offsets reported for individual images translate the image to its center and are to be applied BEFORE rotations in Fourier Space.

### 7.2.3 Contrast Transfer Function

CTF parameters are defined as in those used in CTFFIND3, also see the publication by Mindell et al (2003).