



Frealign: An Exploratory Tool for Single-Particle Cryo-EM

N. Grigorieff¹

Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA, United States

¹Corresponding author: e-mail address: niko@grigorieff.org

Contents

1. Introduction and Philosophy	192
2. Frealign Elements at a Glance	193
2.1 Running Frealign	193
2.2 Required Input	194
2.3 Optional Input	197
2.4 Output	197
2.5 Naming Conventions	198
3. Algorithms	199
4. Typical Application Scenarios	201
4.1 Refinement of a Structure Generated with Different Software	201
4.2 3D Reconstruction Using Parameters from a Previous Frealign Run	208
4.3 3D Classification	208
4.4 Selecting or Merging Particles from Different Classes	210
4.5 Generating an Initial Map	211
4.6 Using Masks	213
4.7 Asymmetric Refinement	216
4.8 Processing Images of Helical Structures	219
5. Tuning Options	220
6. Related Software	223
References	224

Abstract

Frealign is a software tool designed to process electron microscope images of single molecules and complexes to obtain reconstructions at the highest possible resolution. It provides a number of refinement parameters and options that allow users to tune their refinement to achieve specific goals, such as masking to classify selected regions within a particle, control over the refinement of specific alignment parameters to accommodate various data collection schemes, refinement of pseudosymmetric particles, and generation of initial maps. This chapter provides a general overview of Frealign functions and a more detailed guide to using Frealign in typical scenarios.



1. INTRODUCTION AND PHILOSOPHY

Frealign (Grigorieff, 1998, 2007) is an image processing tool that can be used to calculate and refine three-dimensional (3D) structures of macromolecular assemblies that are calculated from images collected on an electron microscope. Its development began in 1996 at the MRC Laboratory of Molecular Biology (Cambridge, UK) with the aim to implement a fast and accurate projection matching algorithm, and to calculate 3D reconstructions that are fully corrected for the contrast transfer function (CTF) of the microscope. Besides the author, a number of other people have contributed to the development of Frealign in various ways, including Tim Grant, Richard Henderson, Dmitry Lyumkis, Alexis Rohou, Charles Sindelar, Alex Stewart, Douglas Theobald, Christine Villeneuve, and Matthias Wolf (Lyumkis, Brilot, Theobald, & Grigorieff, 2013; Sindelar & Grigorieff, 2012; Stewart & Grigorieff, 2004; Wolf, DeRosier, & Grigorieff, 2006), and a GPU-accelerated version was developed by Cheng and his group (Li, Grigorieff, & Cheng, 2010). Its primary application has remained the refinement of particle alignments and 3D reconstruction optimized to reveal details at the highest possible resolution. Other features were added over the years, including refinement of microscope defocus and magnification, correction for the Ewald sphere curvature (Wolf et al., 2006), processing of helical particles (Alushin et al., 2010), 3D classification (Lyumkis et al., 2013), and density masking. Furthermore, algorithms and run scripts were developed to take advantage of parallel computing environments to speed up processing. These developments have made Frealign one of the fastest, most versatile image processing tools for the refinement of single-particle structures, yielding some of the best-resolved reconstructions to date.

Frealign is freely available for download from the Grigorieff lab web page (<http://grigoriefflab.janelia.org/frealign>). Its primary purpose is to serve as a platform for the development of new image processing algorithms and to support projects in the Grigorieff lab and at the MRC Laboratory of Molecular Biology. Little attention has therefore been devoted to user friendliness and documentation, which require time and resources that were instead devoted to support the primary mission of Frealign. However, some help is provided by the Frealign user forum on the Grigorieff lab web page, which allows users to ask questions and read up on previously answered questions.

The more narrowly defined scope and purpose of Frealign distinguish it from software packages that are designed to offer a complete set of tools for

single-particle image processing, including movie processing, CTF determination and correction, particle selection, initial map generation, 2D and 3D classification, refinement, and reconstruction. Using Frealign requires the use of other software to carry out many of the steps necessary to arrive at a 3D reconstruction. Tools for some of these steps have also been developed as stand-alone applications in the Grigorieff lab, such as Signature (Chen & Grigorieff, 2007), CTFFIND (Mindell & Grigorieff, 2003; Rohou & Grigorieff, 2015), Unblur/Summovie (Grant & Grigorieff, 2015b), and magnification distortion correction (Grant & Grigorieff, 2015a), which are also freely available to users. Some of these stand-alone applications will briefly be described at the end of this chapter. However, the main emphasis of the chapter will be on the use of Frealign, describing typical application scenarios and providing practical advice on how to achieve the highest possible resolution. This differs from an earlier paper on Frealign (Grigorieff, 2007) that focused more on algorithmic features.



2. FREALIGN ELEMENTS AT A GLANCE

2.1 Running Frealign

The Frealign distribution is available from the Grigorieff lab web page (<http://grigoriefflab.janelia.org/frealign>) and contains compiled versions of the programs for 64-bit Linux and Mac OS systems. Installation requires unpacking of the archive and adding the path to the compiled programs and run scripts to the user environment.

Frealign has been developed to run on Linux and Mac OS workstations and is run from a command line inside a terminal. Many of the available commands can be listed by issuing the `frealign_help` command. These include `frealign_run_refine` and `frealign_calc_reconstructions`, commands that will start the refinement of a structure or calculate a 3D reconstruction using parameters from a previous Frealign run. The progress toward completion of a task can be followed by monitoring the file `frealign.log`. Apart from these high-level tasks (run scripts) there are also more primitive commands that call up one of the compiled programs that are part of the Frealign distributions, for example, `bfactor.exe`, which allows users to apply a low-pass filter and sharpen a 3D reconstruction using a specified B-factor (see below).

Structure refinement in Frealign is performed iteratively. Each iteration takes input files from the previous cycle and produces new output files that

can serve as input for a new cycle. The user can specify the number of cycles to run and how many computing resources to dedicate to the job (see below).

2.2 Required Input

Most data used by Frealign is stored either in text files or image files. While the most extensively used and tested image format is the MRC format (Crowther, Henderson, & Smith, 1996), images stored using the Spider (Frank et al., 1996) and IMAGIC (van Heel, Harauz, Orlova, Schmidt, & Schatz, 1996) formats are also supported. A schematic overview of required and optional input and output files is shown in Fig. 1, together with some of the functional features discussed below. To run Frealign, the user has to set up a text file called `mparameters` (Fig. 2) that specifies

- the computing architecture, for example, cluster type and number of CPUs to use;
- the main Frealign control parameters, including the number of refinement cycles to run;
- data-specific parameters, such as input parameters and microscope settings;
- optional parameters to tune refinement; and
- masking parameters.

Each input line in `mparameters` is annotated to help users choose appropriate settings. Many of the settings involve flags that can be set to “T” (for “true”) or “F” (for “false”) to turn a feature on or off. Besides `mparameters` the user also has to supply a particle image stack and a particle parameter file—a text file providing some information about each image in the stack. The images in the stack should be uniformly scaled, ie, the average background (solvent) density should be set to zero and its variance to a specified value. In practice, it is sufficient to set the average density of each particle image to zero and its variance to a specified value. This scaling procedure includes the particle density and will therefore be less accurate than a procedure that only considered the background in the calculation. However, in tests comparing results with particles scaled with one of these two procedures, no noticeable differences were observed.

The particle parameter file contains one line of text for each image in the stack. It is therefore important to make sure that the number of lines (excluding comment lines) in a parameter file matches the number of images in the stack. Each line of text normally contains the following information:

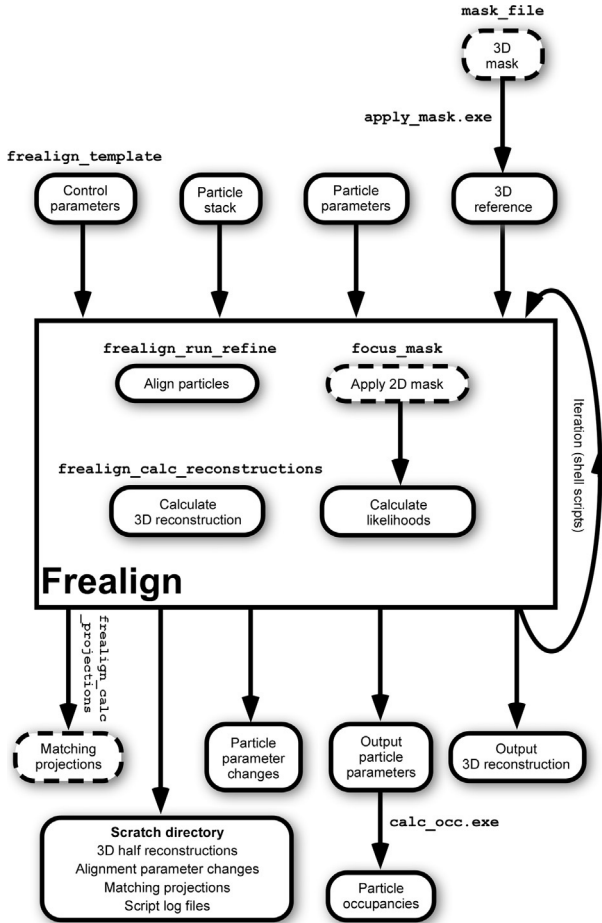


Fig. 1 Schematic overview of Frealign functions, input and output data. Some of the Frealign commands mentioned in the text are shown in Courier font above the functions or files they relate to. Features and files that are optional and not always used are shown with *dashed* borders. Frealign is run by a number of shell scripts that prepare input and output data, manage parallel execution, and perform iterations when more than one cycle is run.

- Euler angles (in degrees) and x, y translations (in \AA) describing the alignment of the particle;
- micrograph identifier, image magnification, and defocus;
- a parameter describing the membership of the particle to a class (occupancy);
- relative log likelihood for the particle given the map and alignment parameters;

```

Control parameter file to run Frealign
=====

This file must be kept in the project working directory from which the refinement scripts are launched.

Note: Please make sure that project and scratch directories (if specified) are accessible by all sub-processes that are run on cluster nodes.

# Computer-specific setting
cluster_type none ! Set to "sgs", "lsf", "slurm", "stampede", "pbs" or "condor" when running on a cluster, otherwise set to "none".
nprocessor_ref 16 ! Number of CPUs to use during refinement.
nprocessor_rac 16 ! Number of CPUs to use during reconstruction.
mem_per_cpu 2048 ! Memory available per CPU (in MB).

# Refinement-specific parameters
MODE 1 ! 1, 2, 3 or 4. Refinement mode, normally 1. Set to 2 for additional search.
start_process 2 ! First cycle to execute. Output files from previous cycle (n-1) required.
end_process 2 ! Last cycle to execute.
res_high_refinement 6.0 ! High-resolution limit for particle alignment.
res_high_class 8.0 ! High-resolution limit to calculate class membership (OCC).
nclasses 1 ! Number of classes to use.
DANG 0.0 ! Mode 3 and 4: Angular step for orientational search. Will be set automatically if set to 0.
ITMAX 200 ! Mode 2 and 4: Number of repetitions of grid search with random starting angles.

# Dataset-specific parameters
data_input particle ! Root name for parameter and map files.
raw_images /path/particle_stack
image_contrast N ! N or P. Set to N if particles are dark on bright background, otherwise set to P.
outer_radius 170.0 ! Outer radius of spherical particle mask in Angstrom.
inner_radius 0 ! Inner radius of spherical particle mask in Angstrom.
mol_mass 2500.0 ! Molecular mass in kDa of particle or helical segment.
Symmetry C1 ! Symmetry of particle.
pix_size 1.237 ! Pixel size of particle in Angstrom.
detap 5.0 ! Pixel size of detector in micrometer.
Aberration 2.0 ! Spherical aberration coefficient in millimeter.
Voltage 300.0 ! Beam acceleration voltage in kilovolt.
Amp_contrast 0.07 ! Amplitude contrast.

# Expert parameters (for expert users)
XSTD 0.0 ! Tighter masking of 3D map (XSTD > 0) or particles (XSTD < 0).
PBC 2.0 ! Discriminate particles with different scores during reconstruction. Small values (5 - 10) discriminate more than large values (50 - 100).
parameter_mask "1 1 1 1 1" ! Five flags of 0 or 1 (e.g. 1 1 1 1 1). Determines which parameters are refined (PSI, THETA, PHI, SHX, SHY).
refinementinc 4 ! When larger than 1: Alternate between refinement of OCC and OCC + angles.
refineshiftinc 4 ! When larger than 1: Alternate between refinement of OCC and OCC + angles + shifts.
res_reconstruction 0.0 ! High-resolution limit of reconstruction. Normally set to Nyquist limit.
res_low_refinement 0.0 ! Low-resolution limit for particle alignment. Set to particle dimension or larger.
thresh_reconst 0.0 ! Particles with scores below this value will not be included in the reconstruction.
thresh_refine 50.0 ! Mode 4: Score threshold above which search will not be performed.
nbootstrap 1000 ! Number of bootstrap volumes to calculate real-space variance map.
FMAG F ! T or F. Set to T to refine particle magnification. Not recommended in most cases.
FDEF F ! T or F. Set to T to refine defocus per micrograph. Not recommended in most cases.
FRITG F ! T or F. Set to T to refine astigmatism. Not recommended in most cases.
FRARF F ! T or F. Set to T to refine defocus for each particle. Not recommended in most cases.
FFILZ T ! T or F. Set to T to apply optimal filter to reconstruction. Recommended in most cases.
FMATCH F ! T or F. Set to T to output matching projections. Only needed for diagnostics.
FRBINT F ! T or F. Set to T to apply symmetry also in real space. Not needed in most cases.
FBOOST T ! T or F. Set to F if potential overfitting during refinement is a concern.
RBFactor 0.0 ! B-factor sharpening (when < 0) applied during refinement. Not recommended in most cases.
beam_tilt_x 0.0 ! Beam tilt in mrad along X-axis.
beam_tilt_y 0.0 ! Beam tilt in mrad along Y-axis.
mp_cpus 1 ! Number of CPUs to use for each reconstruction job.
restart_after_crash F ! T or F. Set to T to restart job after a crash.
delete_scratch T ! Delete intermediate files in scratch directory.
qsub_string_ref "" ! String to add to cluster jobs submitted for refinement (only for sgs, lsf, slurm and pbs clusters).
qsub_string_rec "" ! String to add to cluster jobs submitted for reconstruction (only for sgs, lsf, slurm and pbs clusters).
first_particle
last_particle
frealign_bin_dir
scratch_dir

# Masking parameters (for expert users)
mask_file
mask_edge 5 ! Width of cosine edge in pixels to add around mask. Set to 0 to leave mask unchanged.
mask_outside_weight 0.0 ! Factor to downweight density outside of mask (normally 0.0 - 1.0).
mask_filt_rn 0 ! Filter radius (in R) to low-pass filter density outside density. Set to 0.0 to skip filtering.
mask_filt_edge 5 ! Width of cosine edge in reciprocal pixels to add to filter function.
focus_mask "" ! Four numbers (in Angstroms) describing a spherical mask (X, Y, Z for mask center and R for mask radius).

```

Fig. 2 Template for the `mparameters` file containing all the control parameters required to run Frealign. Each keyword has an assigned value or string (text) and, in most cases, a comment line explaining how the parameter should be set. The control parameters are divided into different sections to help users customize the parameters according to their environment and project. Parameters listed in the expert section usually do not need to be changed until refinement and classification have converged. These parameters allow users to tune their refinement to improve resolution but they require some experience to be set correctly.

- standard deviation of the estimated background noise; and
- particle score.

At the beginning of a new project, when alignment parameters are not known, the user can supply an abbreviated particle parameter file that will contain only four numbers per line:

- an integer to identify the micrograph the particle came from and
- two defocus values and an astigmatic angle (in degrees) needed to describe the microscope CTF. These values are written out by the program CTFFIND (Mindell & Grigorieff, 2003; Rohou & Grigorieff, 2015), for example.

Frealign will convert this file into a full parameter file by adding random Euler angles, setting the particle translations to zero and filling in the rest of the parameters with nominal values that can then be updated in later refinement cycles.

2.3 Optional Input

In most cases, users will also supply a 3D reconstruction (or several reconstructions for multireference refinement and classification, see below) on input that was obtained either from a previous Frealign run or through other means. The 3D reconstruction is used by Frealign as a reference to refine the particle parameters (see later). If no 3D reconstruction is supplied, Frealign will calculate one from the input particle parameters. If these parameters are given in the abbreviated format, ie, no Euler angles and x,y translations are available, Frealign will calculate a 3D reconstruction using randomly assigned angles after converting the abbreviated parameter file to a full parameter file, thus creating a random startup structure (see below).

Finally, a 3D mask file can be specified in `mparameters` that Frealign will use to mask the input 3D reconstruction, allowing the user to focus on specific regions of the molecule during refinement (see below).

2.4 Output

Each refinement cycle performed by Frealign will generate a new particle parameter file and 3D reconstruction (if multiple references are used, parameter files and reconstructions are generated for each). A scratch directory used by Frealign to hold temporary files also contains 3D reconstructions calculated from half the data each. These are used by Frealign to calculate a Fourier shell correlation (FSC) curve to estimate the resolution of the final

3D structure (Harauz & van Heel, 1986). These half reconstructions can be ignored in a typical Frealign run but are sometimes useful if users would like to perform their own resolution estimation, for example, using ResMap (Kucukelbir, Sigworth, & Tagare, 2014). The FSC curve for each reconstruction can be found in a resolution table at the end of the particle parameter file belonging to the reconstruction, and plots can be generated using the command `frealign_plot_fsc`. The table also contains other information, for example, an adjusted FSC curve called `Part_FSC`, which estimates the resolution of the reconstruction (usually at the threshold value of 0.143, Rosenthal & Henderson, 2003) after removing all solvent noise. This adjusted curve should therefore be taken as the curve shown in publications to support resolution claims. While resolution estimates for reconstructions after removing solvent noise are often obtained by applying tight masks, Frealign obtains this estimate by considering the volume occupied by the particle (Sindelar & Grigorieff, 2012). The volume is calculated from the molecular mass of the particle provided by the user in `mparameters` using a conversion factor of 810 Da/nm^3 , equivalent to $1.25 \text{ \AA}^3/\text{Da}$ (Matthews, 1968). The volume is also used to calculate the spectral signal-to-noise ratio (SSNR) present inside the reconstructed particle density (called `Rec_SSNR` in the final resolution table). The particle SSNR (called `Part_SSNR` in the final resolution table) is used to apply an optimal filter to the final reconstruction (Sindelar & Grigorieff, 2012) if `FFILT` is set in `mparameters`. In addition to the SSNR filter, it is recommended to apply a negative B-factor to the final reconstruction to sharpen the density, and a low-pass filter with a cosine-edged cutoff to set terms beyond the resolution limit to 0. Both operations can be accomplished with `bfactor.exe` (see below).

2.5 Naming Conventions

Frealign uses the following naming scheme for alignment parameter files and 3D reconstructions: file names consist of a seed that should identify the particle (for example, 70S for the 70S ribosome), followed by “_M_rN” where M and N are integers that signify the refinement cycle number and reference, respectively. Typically, when initializing a Frealign run, only one reference is used (see below) and the refinement starts with cycle 1. Therefore M should be set to 0 and N to 1 (eg, “70S_0_r1”). The parameter files are expected to have the extension “.par” while particle image stacks and 3D reconstructions have extensions that depend on the file format. MRC/CCP4 files have the ending “.mrc,” Spider “.spi,” and IMAGIC “.hed” and “.img.”



3. ALGORITHMS

A description of the algorithms employed by Frealign to refine alignment parameters and perform classification is given in [Grigorieff \(2007\)](#) and [Lyumkis et al. \(2013\)](#). Briefly, Frealign performs projection matching to determine more accurate alignment parameters. Projections are calculated using the reference map provided on input and alignment parameters for each particle are updated according to the projection that generates the highest correlation coefficient. The user has a choice to search for projections with parameters close to those previously found (local search) by setting `MODE` to 1 in `mparameters`, or to perform a global parameter search with randomly generated parameters (`MODE` set to 2 and `ITMAX` set to the desired number of trials, eg, 100) or with parameters systematically chosen according to a search grid (`MODE` set to 3 and `DANG` set to the desired angular step in degrees, eg, 10). If `MODE` is set to 3, the user can also set `DANG` to 0 to let Frealign calculate an appropriate angular step, based on the particle radius and refinement resolution limit (see below) set by the user. The correlation coefficient calculated by Frealign is weighted according to the SSNR present in the particle images. The average particle SSNR is contained in the final resolution table (`Part_SSNR`, see above), together with the FSC curve and other statistics. It is important not to delete this table from the end of a parameter file because it is read by Frealign in the next refinement cycle.

3D classification is done using a maximum likelihood approach. Frealign calculates relative likelihoods (ie, likelihood values that are missing constants necessary to put the likelihoods on an absolute scale) of each particle given a 3D reference map and a set of alignment parameters. The logarithms of these values are listed in the particle parameter files (column `LogP`). For classification, multiple reference structures are used and likelihood values are calculated for each. At the end of a refinement cycle, these likelihood values are converted by the program `calc_occ.exe` (part of the Frealign distribution and executed automatically by the run script) into weights, so-called occupancies (column `OCC`) that determine the partitioning of each particle into the different classes. To speed up classification and improve convergence, the user can specify if Frealign should refine alignment parameters with every classification cycle, or if alignment parameter refinement should only be done every N th cycle. Typically, alignment parameter refinement should only be done every third or fourth cycle by setting the values for both

`refineangleinc` and `refineshiftinc` to 3 or 4 in `mparameters`. During cycles where alignment parameters are not refined, these parameters remain unchanged and the cycles complete more quickly. The user can also specify different numbers for `refineangleinc` and `refineshiftinc` to refine angles and shifts on different schedules but this is usually not necessary.

`Frealign` has been designed to reduce or entirely avoid overfitting of parameters. Overfitting is a well-known problem in single-particle work (Grigorieff, 2000; Stewart & Grigorieff, 2004). It manifests as features in the reconstruction that result from the alignment of noise rather than signal. If all particles in a dataset are aligned against the same reference, overfitting can also lead to inflation of the FSC and unrealistically high-resolution estimates. The latter problem can be mediated by calculating FSC curves from reconstructions that were refined entirely separately, thus not sharing a common reference (Grigorieff, 2002; Henderson et al., 2012). While this reduces the chances of generating inflated FSC values, it does not directly address overfitting. `Frealign` provides several ways to counter overfitting and with it, inflated FSC values:

- (a) *Weighting of the correlation coefficient.* The SSNR-weighted correlation coefficient used during projection matching (see above) aims at giving data with a stronger signal a higher weight than noisier data. This helps the signal drive the alignments and reduces the impact of the noise. A potential weakness of this approach is an incorrect measurement of the SSNR, which might be biased toward higher values like the FSC.
- (b) *Maximizing the absolute value of the correlation coefficient.* The user has the option to maximize an unsigned version of the weighted correlation coefficient (Stewart & Grigorieff, 2004) instead of the signed version by setting `FB00ST` to “F” in `mparameters`. When switching to the unsigned correlation coefficient, alignment of the strong low-resolution signal is usually unaffected (at very low resolution, below 30 Å, `Frealign` will always use a signed correlation to ensure proper centering of particles). However, as the SSNR decreases toward higher resolution, alignment may be driven more by the noise and, using the unsigned correlation coefficient, may end up being aligned in-phase (positive correlation) or out-of-phase (negative correlation). Subsequent averaging of images during 3D reconstruction will lead to strong attenuation of the incoherently aligned noise. Setting `FB00ST` to “T” may help the alignment in some cases and the user is encouraged to try different settings to get the best results (see below). However, setting

FB00ST to “T” also means that the chance of overfitting is increased. Careful validation of additional features appearing in the map is therefore necessary.

- (c) *Setting a refinement resolution limit.* In Frealign, the data used during refinement are bandpass filtered. The low- and high-resolution limits can be set in `mparameters` but users usually only change the latter while the former is left at 0 to let Frealign set the value automatically. It is good practice to monitor the progress of refinement and limit the high-resolution limit to a value well below the current resolution limit. For example, if the `Part_FSC` curve suggests a resolution of 8 Å, the resolution limit should probably not exceed 10 Å. Limiting the resolution during refinement (and classification) means that the FSC values at higher resolution will show little or no bias from noise overfitting. If, on the other hand, the FSC curves suggest that the resolution of the reconstruction increases more or less in parallel with the resolution limit set by the user, this is usually a strong sign that the refinement is unsuccessful and does not produce reliable structural details.



4. TYPICAL APPLICATION SCENARIOS

In this section, a few typical application scenarios are described to help users get started with Frealign. The processing steps for cases not described here may be derived from the scenarios below, giving users the flexibility to adapt to their own situations.

4.1 Refinement of a Structure Generated with Different Software

This will be the most common scenario in which the following input data exist:

- a 3D reconstruction,
- a particle image stack,
- a list of defocus parameters (two defocus values and an astigmatic angle) for each particle,
- a list of micrograph numbers detailing where each particle came from (optional), and
- a list of alignment parameters (Euler angles and x,y translations) for each particle (optional).

If particle alignment parameters are not available, the user will have to generate a startup parameter file with a list that contains four numbers per line

```

1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
1  21241.2  21446.7  32.56
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41
2  19113.7  19301.8  41.41

```

Fig. 3 Example of a startup parameter file containing the required data for 20 particles. Each line lists a micrograph number that the particle originates from, as well as defocus values and astigmatic angle determined for this micrograph. The defocus information can vary from particle to particle if more accurate information is available, for example, by using CTFFILT (Mindell & Grigorieff, 2003). If micrograph numbers are not known, they can be set to a constant number larger than 0 or to the particle number.

and one line per particle image (Fig. 3). In each line, the first number identifies the micrograph that the corresponding particle originates from and the last three numbers provide the defocus information (following CTFFIND conventions (Mindell & Grigorieff, 2003; Rohou & Grigorieff, 2015)). Numbers can be separated by commas or spaces; no other formatting is required. If micrograph numbers are not available, the user can set the identifiers for all particles to a constant number, for example, 1. The startup parameter file will be renamed by Frealign and replaced with a full Frealign-style parameter file (Fig. 4) that contains random Euler angles and 0,0 for the x,y translations.

If particle alignment parameters are also known, a Frealign-style parameter file should be generated from the file originating from the other software. Conversion scripts are available on the Frealign download page for different software. Users are cautioned, however, to make sure that the conversion worked correctly by checking the results for a few particles. The format of the parameter files originating from other software may change and these changes are usually not immediately accommodated in the conversion

C	PSI	THETA	PHI	SHX	SHY	MAG	FILM	DF1	DF2	ANGAST	OCC	LogP	SIGMA	SCORE	CHANGE
1	224.68	301.05	104.64	-0.99	-3.26	104011	1	21241.2	21446.7	32.56	100.00	-6342	0.8298	24.03	0.63
2	222.50	341.41	357.66	1.90	1.63	104011	1	21241.2	21446.7	32.56	100.00	-6444	0.8403	23.08	-0.06
3	192.03	97.21	178.83	-1.00	-0.38	104011	1	21241.2	21446.7	32.56	100.00	-7760	0.8809	17.92	2.41
4	127.34	143.60	123.36	1.25	2.54	104011	1	21241.2	21446.7	32.56	100.00	-7275	0.8725	18.48	-0.56
5	138.80	336.29	113.17	-1.37	-0.78	104011	1	21241.2	21446.7	32.56	100.00	-7426	0.8704	19.98	1.09
6	99.27	24.49	348.32	-0.52	-0.31	104011	1	21241.2	21446.7	32.56	100.00	-6681	0.8552	22.98	-1.34
7	317.37	195.53	124.72	0.30	-2.47	104011	1	21241.2	21446.7	32.56	100.00	-7145	0.8604	21.21	0.98
8	101.06	347.28	62.85	-1.95	-3.25	104011	1	21241.2	21446.7	32.56	100.00	-6974	0.8624	24.71	0.60
9	21.36	358.81	132.15	1.48	-1.59	104011	1	21241.2	21446.7	32.56	100.00	-8992	0.9527	20.33	1.89
10	334.18	264.77	261.20	-1.72	-1.30	104011	1	21241.2	21446.7	32.56	100.00	-7033	0.8573	22.35	-0.27
11	138.79	268.49	294.01	-1.09	-0.61	104011	2	19113.7	19301.8	41.41	100.00	-7000	0.8600	24.16	0.17
12	93.39	148.69	315.88	-3.19	-1.31	104011	2	19113.7	19301.8	41.41	100.00	-7273	0.8830	21.08	-0.98
13	132.31	228.06	266.01	-2.13	-0.40	104011	2	19113.7	19301.8	41.41	100.00	-7270	0.8675	23.83	1.77
14	178.55	200.04	86.08	-0.80	-0.83	104011	2	19113.7	19301.8	41.41	100.00	-7404	0.8740	20.65	0.16
15	22.43	177.64	118.28	1.60	-0.27	104011	2	19113.7	19301.8	41.41	100.00	-7102	0.8609	26.86	0.41
16	244.62	284.19	201.33	1.48	1.09	104011	2	19113.7	19301.8	41.41	100.00	-7206	0.8708	19.26	0.65
17	276.29	189.05	153.91	2.17	0.33	104011	2	19113.7	19301.8	41.41	100.00	-7605	0.8862	19.81	-0.43
18	182.05	332.75	208.28	0.25	2.47	104011	2	19113.7	19301.8	41.41	100.00	-7851	0.8958	22.53	-0.24
19	325.00	98.83	130.36	-0.41	-0.09	104011	2	19113.7	19301.8	41.41	100.00	-7463	0.8792	21.37	0.57
20	276.20	154.32	148.45	1.38	-1.24	104011	2	19113.7	19301.8	41.41	100.00	-7060	0.8579	17.51	0.97

Fig. 4 Example of a full Frealign alignment parameter file, after running Frealign with the startup file in [Fig. 3](#), containing Euler angles (PSI, THETA, PHI) and x,y translations (SHX, SHY), as well as micrograph numbers (FILM), magnification (MAG) and defocus (DF1, DF2, ANGAST) information, occupancies (OCC), log likelihoods (LogP), and scores (SCORE). The SIGMA column lists estimates of the standard deviation of the noise present in the particle images while CHANGE lists the change in the score compared with the previous refinement cycle.

scripts. It is recommended that users familiarize themselves with a scripting language (eg, shell script or Python) and then adapt the available conversion scripts to their own needs.

The particle image stack must contain uniformly scaled images (see above) with an even box size. For both ice-embedded and negatively stained particles and underfocused images, the particles should be dark on lighter background (there is also an option to use the opposite contrast, see below). Finally, some parameters have to be set or adjusted in the `mparameters` file. A fresh `mparameters` file can be generated using the `Frealign` command `frealign_template`. Using a text editor, the following settings should be adjusted:

- `cluster_type`: should be set to the computing infrastructure used. If computations are done on a local workstation, this should be set to “none.”
- `nprocessor_ref`, `nprocessor_rec`: should be set to the number of CPUs to be used for parallelized computation during parameter refinement and reconstruction, respectively. While there is relatively little overhead when adding CPUs for refinement (using 1000–2000 CPUs should work without problems), the speedup from additional CPUs used for reconstruction will depend on the speed of the disk storage. Values for `nprocessor_rec` should probably not exceed 100 and are more typically 30–50. It is recommended to set the values for both parameters to a multiple of the number of classes used for classification (see below).
- `MODE`: should be set to 1 if valid particle alignment parameters are available on input, or 3 if only defocus values and micrograph information are provided. Setting `MODE` to 1 will perform a local search for improved alignments while 3 will perform a global search that will take significantly more time and should therefore only be run when necessary. There are other run modes available (2 and 4) that are less commonly used and not discussed here. If the user decides to run with `MODE` set to 3, it is recommended to work with binned data. Using `Frealign`'s tool `resample.exe` (or `resample_mp.exe` for multi-CPU environments), the pixel size of the particle image stack and 3D reference reconstruction can be changed to speed up processing. For example, if the native pixel size of a dataset is 1.5 Å but the global search with `MODE` set to 3 is performed at lower resolution, for example, at 20 Å (see below), the stack and 3D reference can be resampled to a pixel size of 9 Å, giving a Nyquist frequency limit of 18 Å, ie, just a little higher than the chosen resolution limit of 20 Å. At a later stage, processing can be switched back to a

smaller pixel size to enable higher resolution refinement. When changing the pixel size, it is important to also change the settings for `pix_size` and `dstep` in `mparameters` (see below). Furthermore, the 3D reconstruction from the previous cycle has to be recalculated using the new pixel size, or processed with `resample.exe` to generate a volume with the correct pixel size. Frealign recalculates the reconstruction automatically if the previous reconstruction is deleted (or moved to a different directory). Additional speedup can be obtained by reducing the margins around the particles (if possible) using the CROP tool available for download on the Grigorieff lab web page (see below). Images should not be cropped when refining at high resolution as this may lead to CTF aliasing loss of high-resolution signal (Rohou & Grigorieff, 2015).

- `start_process`, `end_process`: should be set to the first and last refinement cycle to be run. For example, if the initial parameter file carries cycle number 0 and 10 cycles should be run, the values for `start_process` and `end_process` should be set to 1 and 10, respectively.
- `res_high_refinement`: should be set to the desired resolution limit used during refinement. This will usually depend on the estimated resolution of the input 3D reference (see above). `mparameters` contains a second resolution limit, `res_high_class`, which determines the resolution used for classification. Users can keep this value at the default of 8 Å as Frealign will always adjust it internally to the value for `res_high_refinement` if that indicates a lower resolution than `res_high_class`.
- `nclasses`: determines the number of classes to be refined. This should be set to 1 when working with a parameter file that does not contain particle alignment parameters, or if the alignment parameters are not very accurate. Classification of particles into multiple classes is only recommended at a later stage of the refinement, when refinement with a single class does not improve the resolution further. To switch on classification, the user simply sets `nclasses` to a value larger than 1.
- `DANG`: determines the angular step size used in a global search (`MODE` set to 3). This should be set to the default of 0 to enable automatic step sizing by Frealign, based on the specified resolution limit (see above) and particle radius (see below). The user can specify a fixed value by entering a value larger than 0. A second parameter, `ITMAX`, is not used for `MODE` 1 and 3 and does not need to be changed by the user.
- `data_input`: the text string defining the seed used to generate the file names for alignment parameters files and reconstructions (see above).

- `raw_images`: name of the particle image stack. The path can either be relative to the working directory or absolute.
- `image_contrast`: normally set to N to indicate that particles appear dark against light background (see above). Users can also set this to P if the particle images have opposite contrast.
- `outer_radius`: determines the outer radius of the spherical mask to be applied to the final reconstruction, as well as the radius of a circular mask applied to the particle images during refinement. If this is set to a negative value no mask will be applied.
- `inner_radius`: determines the inner radius of the spherical mask to be applied to the final reconstruction. It is not used to mask the particle images. This parameter is normally set to 0 but if the particle is hollow or contains disordered density (eg, a clathrin coat or an icosahedral virus capsid), setting the inner radius to an appropriate value allows users to mask the inside of the particle and reduce noise.
- `mol_mass`: should be set to the total molecular mass of the particle. The value is given in kDa and determines how the FSC curve is scaled to calculate `Part_FSC`, which provides a more accurate resolution estimate of the particle density (see above).
- `Symmetry`: should be set to the assumed symmetry of the particle. The default is C1, ie, no symmetry.
- `pix_size`: should be set to the desired pixel size of the output reconstruction in Angstroms. Usually this is the same as the pixel size of the input particle images.
- `dstep`: should be set to the effective pixel size of the detector in micrometers. This is usually the physical pixel size of the detector, for example, 5 μm for the K2 detector (Gatan, Pleasanton, CA). However, if the particle images are binned, both the image pixel size and the effective detector pixel size change. For example, 2×2 pixel binning of the images doubles their pixel size (`pix_size`) and the effective detector pixel size (`dstep`). Users can check that they have set `pix_size` and `dstep` correctly by dividing `dstep` by `pix_size`. The result should be the particle magnification indicated in the alignment parameter file.
- `Aberration`, `Voltage`, `Amp_contrast`: should be set to the appropriate microscope parameters. `Aberration` and `Voltage` are given in millimeters and kilovolts, respectively. For example, for the FEI Titan Krios microscope, typical values for `Aberration`, `Voltage`, and `Amp_contrast` are 2.7, 300, and 0.07. If `Amp_contrast` is set to a negative number, CTF

correction is turned off in Frealign. However, this only works when `image_contrast` is set to N and the particle images have been pretreated to correct for the CTF (eg, phase flipping), yielding particles that appear light against a dark background.

The final list of files in the working directory needed to run Frealign includes `mparameters`, the particle parameter file (either with four values per line or a full Frealign parameter file with Euler angles, x,y translations and additional columns with occupancies, likelihood values, and scores), a 3D reference map and a particle image stack. For example, if the seed for the file names is 70S, and the stack is named `particle_stack.mrc`, the list of files includes: `mparameters`, `70S_0_r1.par`, `70S_0_r1.mrc`, `particle_stack.mrc` (assuming MRC file format). Issuing the `frealign_run_refine` command will then start the refinement. New parameter files and 3D reconstructions should appear in the working directory as refinement cycles are completed. Users can follow the status of the refinement by inspecting the file `frealign.log` (see above), as well as temporary files generated in the scratch directory, which is created in the working directory unless specified differently in `mparameters`.

Refinement progress can be monitored in several ways. Users are encouraged to check the resolution statistics appended to the end of the parameter files and verify that the resolution improves from cycle to cycle while observing the steps discussed above to avoid inflated resolution estimates. If there is no noticeable resolution improvement for a number of cycles (eg, 5), the refinement may have converged. Users can try increasing the refinement resolution limit if the resolution of the reconstruction is significantly higher than this limit (see above) and run a few more cycles to see if this leads to further improvement. To continue refinement, the numbers for `start_process` and `end_process` must be updated before issuing the `frealign_run_refine` command. Users can also check if the particle alignment parameters are still changing significantly between cycles. In the scratch directory, text files containing `.shft_` in their names list the changes in the parameters of the current cycle relative to the previous cycle. These files are normally deleted at the end of a cycle, so users have to check them while a cycle is running. Finally, the command `frealign_calc_stats` will display the average score, relative log likelihood per particle (and occupancy, see below) for a specified round. Both scores and likelihood values should increase during refinement until convergence has been reached. However,

if the resolution limit is changed during refinement, scores and likelihood values will be affected. Therefore changes in these values are only meaningful between cycles that use the same resolution limits.

4.2 3D Reconstruction Using Parameters from a Previous Frealign Run

To calculate (or recalculate) a 3D reconstruction using an existing particle parameter file containing Euler angles and x,y translations (ie, a full Frealign parameter file), the user has to set the following values in `mparameters`: `cluster_type`, `nprocessor_rec`, `nclasses`, `data_input`, `raw_images`, `image_contrast`, `outer_radius`, `inner_radius`, `mol_mass`, `Symmetry`, `pix_size`, `dstep`, `Aberration`, `Voltage`, and `Amp_contrast`. Details of how these values should be set are provided in the previous section. If the previous run used more than one reference/class, `nclasses` will have to be set accordingly. The command `frealign_calc_reconstructions` starts the calculation and a new reconstruction (or reconstructions if `nclasses` is larger than 1) will appear in the working directory when Frealign has finished. The parameter file (or files if `nclasses` is larger than 1) will be appended with the relevant resolution statistics. Therefore recalculating reconstructions multiple times will append multiple tables at the end of the parameter files. Only the last table in a parameter file will be used by Frealign in the next refinement cycle.

4.3 3D Classification

3D classification is typically done only after refinement with a single reference has converged, ie, there is no more improvement in resolution and particle parameters do not change much anymore from cycle to cycle (see above). To turn on classification, the value for `nclasses` (in `mparameters`) must be changed from 1 to the desired number of classes. After updating `start_process` and `end_process` to continue from the previous last cycle, refinement with classification is performed by issuing the `frealign_run_refine` command. Frealign will rename the particle parameter file and 3D reconstruction from the previous cycle and replace them with multiple parameter files and 3D reconstructions that result from randomly assigned particle occupancy values (`OCC` column in the parameter files). These occupancy values will be refined in subsequent cycles and, if successful, will indicate the membership of each particle to a class. Often, after refinement, occupancy values will be either 100 or 0, indicating that a particle is, or is not a member of a class, respectively. However, intermediate

values are also possible, indicating that the assignment to a class is not unique. For each particle, the sum of the occupancy values from all classes always adds up to 100.

Typically, several tens of refinement/classification cycles (eg, 50) have to be run before convergence is reached. As before, progress can be monitored by inspecting the resolution statistics at the end of the particle parameter files, changes in the `.shft_` files inside the scratch directory and by running the `frealign_calc_stats` command. This command will also calculate the average particle occupancy for each class. Upon successful classification, significant differences should emerge in the density maps representing the different classes. Users should inspect and compare density maps using display programs such as UCSF Chimera (Pettersen et al., 2004) and TIGRIS (<http://tigris.sourceforge.net>).

If classes represented by different reconstructions are available from a previous Frealign run or another source, classification can also be initiated using these reconstructions. This can be done by following these steps:

- save the particle parameter file and reconstruction from the single-class refinement in a safe place (or rename them),
- copy the saved particle parameter file multiple times to generate new parameter files for each of the classes. For example, if classification should proceed with three classes starting at cycle 101 and the seed for the file names is 70S, copy the original parameters file into `70S_100_r1.par`, `70S_100_r2.par`, and `70S_100_r3.par`. Similarly, copy the available reconstructions representing the previously obtained classes into `70S_100_r1.mrc`, `70S_100_r2.mrc`, and `70S_100_r3.mrc` (assuming MRC file format). Finally, set `nclasses` (in `mparameters`) to the number of classes used (here 3) and issue the command `frealign_run_refine`.

As before, the refinement should be run with `MODE` set to 1 unless previous alignment parameters are not available, in which case `MODE` should be set to 3 for the initial one or two cycles. It is important to remember that when more than one reference is used (`nclasses` set to a value larger than 1) Frealign will only refine the alignment parameters every N th round where N is specified by `refineangleinc` and `refineshiftinc` (see above). When initiating a refinement with `MODE` set to 3, `refineangleinc` and `refineshiftinc` should be temporarily set to 1 until the alignment parameters are deemed roughly correct and `MODE` is set to 1.

As a note of caution, users should be aware that differences in the densities that appear after several cycles of classification may reflect noise and not

real structural differences in the particles. Emerging features should always be evaluated on the basis of plausibility and what else is known about the sample. If the classification results are mainly driven by noise the average occupancy is often very similar for each class. Users must therefore be suspicious of classification results that suggest similar average occupancies of all classes.

4.4 Selecting or Merging Particles from Different Classes

When classification of a dataset has converged, it is often useful to continue refinement and/or classification using a subset of the data. For example, if classification yielded five classes and, after inspection of the densities, two of the classes are so similar that they are considered the same structure, particles from these two classes can be merged into one class. The reconstruction representing the combined class should then reach higher resolution since the total number of particles will be larger compared with the two original classes. To merge particles from several classes, Frealign comes with a tool called `merge_classes.exe`. Input prompts request the file names of the particle parameter files belonging to the classes to be merged, the particle image stack, and criteria for including a particle in the merged output. The criteria include minimum values for occupancy and score. Normally, a minimum occupancy of 50 and score of 0 should be used but users can include or exclude more particles by changing these numbers. `merge_classes.exe` will then generate a new particle parameter file and image stack (using file names specified by the user) that contain only the selected particles. These can then be used in additional refinement and classification cycles using Frealign.

A different tool, `select_classes.exe`, allows the user to select particles from a subset of classes for further refinement. Unlike `merge_classes.exe`, `select_classes.exe` produces several particle parameter files on output that are related to the parameter files provided on input. `select_classes.exe` will remove particles from these parameter files and image stack that do not belong to any of the selected classes. The new parameter files and image stack can then be used for further refinement and classification by Frealign. `select_classes.exe` can therefore be used to remove unwanted particles, for example, because they belong to a “junk” class or a good class that represents a state that should not be refined or classified further. The latter situation may occur when a sample containing many different conformations is classified. To isolate all the different conformations, one strategy would be to allow a larger number of classes during classification. However, this increases the need for computational resources and may reduce the chance of finding

smaller classes because particles may be misclassified to belong to a bigger class simply due to the better signal represented in this class (Yang, Fang, Chittuluru, Asturias, & Penczek, 2012). Therefore removing particles belonging to some of the bigger classes, or selecting one of the bigger classes and classifying it further into a smaller number of subclasses may result in new classes to emerge and will reduce the need for computational resources. When looking for new conformations that are not well represented in a dataset, it is often also useful to employ masking (see below).

4.5 Generating an Initial Map

Although there is no dedicated algorithm implemented in Frealign to generate an initial map, the following scheme can be applied. The user has to supply a startup parameter file with micrograph identifiers and defocus values for each particle (see above) and a particle image stack. It is recommended to limit the parameter list and stack to a subset of about 10,000 particles that have been selected from a larger stack based on defocus and some other “quality” criteria. The selected defocus range should be at the high end of the range used for the entire dataset. Particle quality can be ascertained either by manual picking of the particles by an experienced user, or by selecting particles based on 2D classification, for example, using ISAC (Yang et al., 2012). Furthermore, it is recommended to use the `resample.exe` tool (see above) to change to pixel size of the image stack to a value between 4 and 5 Å to speed up computation. The size of the particle images can be further reduced by using the CROP tool (see below) that is available for download on the Grigorieff lab web page and allows trimming of the margin around particles (users must make sure that the trimming does not cut into the particles). Finally, the refinement resolution limit should be set to a value between 30 and 40 Å (`res_high_refinement`), `nclasses` should be set to 1, `MODE` should be set to 3, the assumed particle symmetry should be specified (`Symmetry`), `refineangleinc` and `refineshiftinc` should be left unchanged (defaults are 4 and 4), and `FBOOST` should be set to T. Using the startup parameter file and stack (named according to Frealign’s naming scheme, see above), as well as an appropriately set `mparameters` file, the user should run one “refinement” cycle by issuing the `frealign_run_refine` command. Using the previous example, the needed files include `mparameters`, `70S_0_r1.par`, `particle_stack.mrc`, and `start_process` and `end_process` should both be set to 1. Frealign will rename the startup parameter file `70S_0_r1.par` and replace it with a full Frealign parameter file with

randomly set Euler angles and translations set to 0,0. The initial reconstruction generated by Frealign will therefore approximate a featureless sphere. When the initial refinement cycle has finished, between three and six classes should be specified by changing `nclasses` to the appropriate number, and seven more cycles should be run (`start_process` and `end_process` should be set to 2 and 8, respectively). As more refinement and classification cycles are executed, the spheres will gain features. When the specified number of cycles has been run, another eight cycles should be run but with a somewhat increased resolution limit, for example, increasing it from 40 to 30 Å. This should be repeated while increasing the resolution limit every time until a resolution of about 10 Å is reached. A possible schedule would therefore include

- cycle 1, resolution limit set to 40 Å, 1 class;
- cycles 2–8, resolution limit set to 40 Å, 3–6 classes;
- cycles 9–16, resolution limit set to 30 Å, 3–6 classes;
- cycles 17–24, resolution limit set to 20 Å, 3–6 classes;
- cycles 25–32, resolution limit set to 15 Å, 3–6 classes;
- cycles 33–40, resolution limit set to 10 Å, 3–6 classes.

When a resolution limit of about 10 Å has been reached, the class with the highest overall FSC should be selected and taken as a starting structure for another round of 40 refinement cycles that follow the same schedule. However, for this new round, the reconstruction representing the selected class with the highest FSC should be provided alongside the startup parameter file. Following the 70S example, for the next 40 cycles, refinement could start with cycle number 101, `70S_100_r1.par` should contain the startup parameters (this could simply be copied from `70S_0.par`, which should be available from the previous round of refinement cycles) and `70S_100_r1.mrc` should be a copy of (or symbolic link to) the selected best reconstruction. The schedule of classification with successively increasing resolution should be repeated until one of the classes shows an FSC curve indicating a resolution that extends significantly beyond 10 Å—an indication that the corresponding reconstruction contains reliable signal beyond 10 Å and therefore, that the structure is likely correct. Users can vary the number of cycles, classes, and resolution thresholds used in every new round. A larger number of cycles and classes may increase the chances of finding classes that represent the correct structure(s) but the computational cost increases and it may be necessary to use more than 10,000 particles for the trials to make sure there is a sufficient number of particles in each class (a minimum of about 2000 particles is recommended). An example of generating an initial model from an initial reconstruction calculated with

randomly assigned Euler angles is shown in [Fig. 5](#) for a cryo-EM dataset of L protein of vesicular stomatitis virus (VSV-L) that was published previously ([Liang et al., 2015](#)).

4.6 Using Masks

Frealign can employ two types of masks. The first involves a 3D volume stored in the same format as the input references. The mask file should contain positive and negative numbers outlining regions of the reference to be included or excluded by the mask, respectively. The mask file must be specified in `mparameters` using the `mask_file` key (if no filename is provided, Frealign assumes that a 3D mask is not used). Frealign will apply (ie, multiply) the mask to the input references before starting refinement after making the following modifications to the mask file:

- all positive mask densities are reset to 1;
- all negative mask densities are reset to 0 or another value specified in `mparameters`, keyword `mask_outside_weight`;
- a soft edge is added to the mask (ie, the region now containing voxels set to 1) using a width specified in pixels in `mparameters`, keyword `mask_edge` (usually set to 5);
- if specifying a value for `mask_outside_weight` larger than 0, the user can also specify if the density outside the mask should be low-pass filtered by setting keyword `mask_filt_res` to specify the filter resolution (a value of 0 turns the filter off) and `mask_filt_edge` to specify the width of a smooth edge (in units of Fourier voxels, usually set to 5) in `mparameters`.

This set of parameters provides the user with different masking strategies. Usually the goal of masking will be to set all densities outside the mask to 0 and retain unmodified density inside the mask so that refinement and/or classification is driven solely by the density inside the mask. However, depending on the size of the particle and mask, the fraction of density left after masking may be too small to achieve reliable particle alignment. The result may be larger alignment errors and loss of resolution in the reconstructions. Using different combinations of `mask_outside_weight` and `mask_filt_res`, it is possible to retain density outside the mask that is downweighted and/or low-pass filtered to prevent significant particle misalignment against the masked references. Depending on the filtering and weighting, the alignment is still driven by the density inside the mask at high resolution while particles are prevented from major misalignment by the low-resolution signal retained both inside and outside the mask. Users can experiment with different filtering and weighting applied to the density

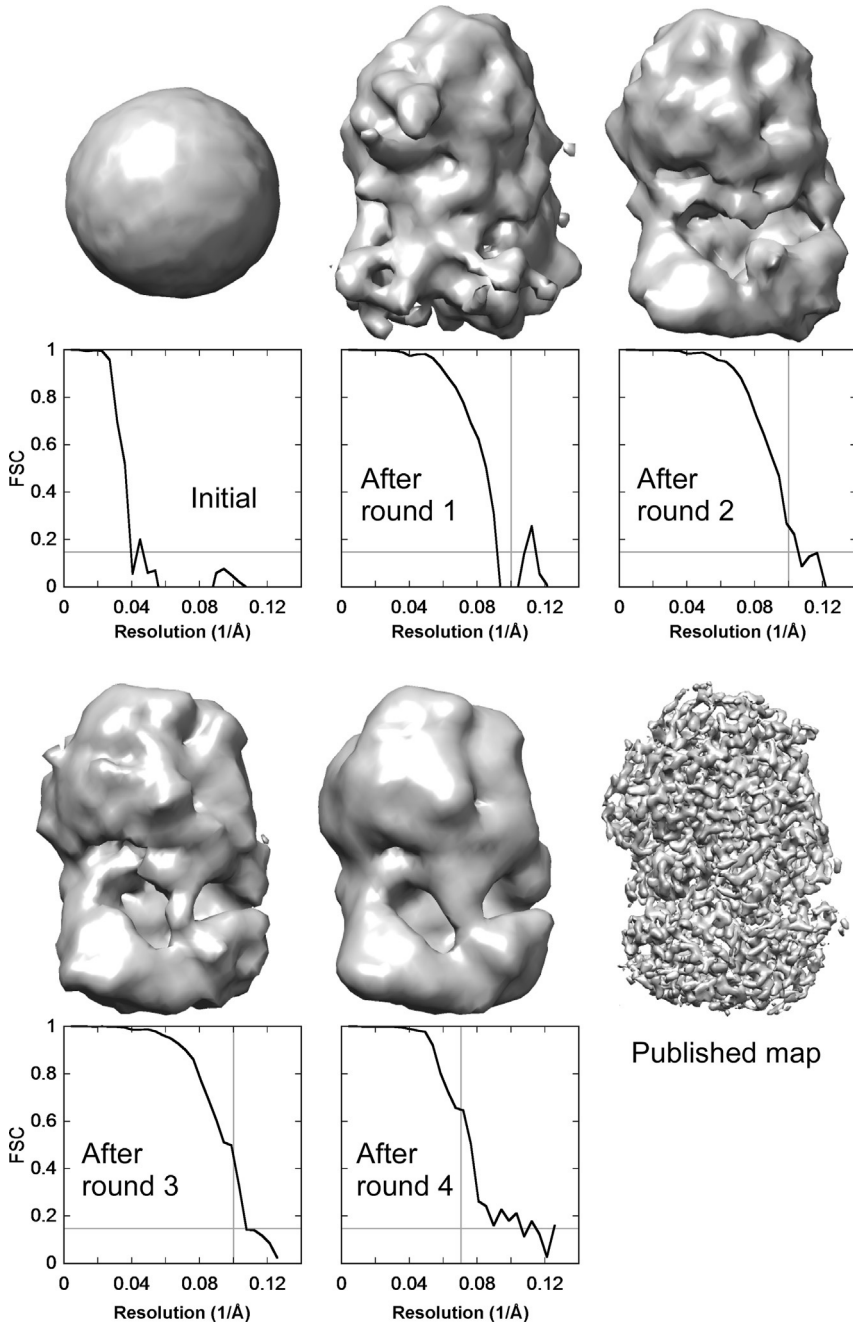


Fig. 5 See legend on opposite page.

outside the mask to achieve the best results. Preventing misalignment when using masks can also be accomplished by turning off the refinement of some of the alignment parameters (see below).

The second type of masking does not require a 3D volume on input. Instead, using the keyword `focus_mask` in `mparameters`, the user specifies the coordinates and radius of a sphere that is then used by Frealign to mask regions in the particle images. Therefore this second type of masking is applied to 2D images, not 3D volumes. For each particle image, the region inside the mask will be a disk that results from the thresholded projection of the specified sphere, in the direction of the view presented by the particle. This masking therefore depends on the correct alignment of the particles and should only be used once the alignments are reasonably accurate. The masked particle images will then be used to derive relative likelihood values that are used for classification (see above). The score function that is used for particle alignment is not affected by this type of masking and, therefore, the `focus_mask` option is only used for classification. Therefore unlike with 3D masking, the alignment accuracy should remain as high as without masking.

By defining a sphere around a region of interest, only structural variability in this region will be used to classify particles. Since the mask is applied in

Fig. 5 Example of an initial map generated from a reconstruction calculated using randomly assigned Euler angles. The dataset contained images of L protein of vesicular stomatitis virus (VSV-L) and led to a 3.8-Å reconstruction of this multienzyme (Liang et al., 2015). To initiate the startup procedure, the particle images were binned threefold to an effective pixel size of 3.711 Å and cropped to generate particle images of 60 × 60 pixels. A subset of 11,671 particles with an underfocus ranging between 1.7 and 2.5 μm were selected from the complete dataset (356,211 particles) and 40 rounds of multiresolution search and refinement were performed according to the scheme described in Section 4.5, using 5 classes and starting with a resolution limit of 40 Å that was gradually increased to 10 Å resolution. `FB00ST` was set to T for this startup procedure. Reconstructions at each stage are shown together with the calculated FSC curves, starting with the initial map obtained with randomly assigned angles (labeled "Initial"). The reconstruction with the best FSC curve from each round was selected to seed the next round of refinement and classification. The FSC curve gradually improved from round to round until the final round 4 in which only 25 refinement cycles were run and the resolution was limited to 14 Å. The FSC curve for the final round indicates a resolution of about 9.5 Å according to the 0.143 criterion (Rosenthal & Henderson, 2003) indicated by the horizontal gray line, thus significantly exceeding the resolution limit used in the refinement (indicated by the vertical gray line) and therefore reflecting an unbiased resolution estimate. The FSC curves in earlier rounds likely reflect some bias as the resolution limit during refinement exceeded to resolution indicated by the FSC. The final published map is shown in the last panel for comparison.

2D, parts of the volume overlapping with the region of interest in the views determined for the particles will be included inside the mask both in the particle image and projections of the reference structures. Differences between images and projections that drive classification will therefore result solely from true structural differences between the particles and references. This differs from 3D masking where density outside the masked region will also be missing in the projections of the masked reference structures. 2D masking may therefore increase the accuracy of classification, compared with 3D masking, an improvement that can also be achieved by subtracting constant parts of the 3D references from the particle images (Bai, Rajendra, Yang, Shi, & Scheres, 2016; Morais et al., 2003; Park et al., 2014) before determining their class memberships. However, unlike the approaches previously described, no shaped mask and no density subtraction in 2D or 3D is required. Masking in 2D offers the additional benefit of excluding noisy areas of the images that do not contain features to be classified, improving classification accuracy further. An example employing both 2D and 3D masking is shown in Fig. 6.

The `focus_mask` option can be used to “explore” different regions of a particle and obtain different classification results depending on which region the sphere is placed in. This opens up the possibility of classification based on different particle regions that display uncorrelated heterogeneity. Each classification task focuses on only one of the affected regions and separates particles based on variability in this region. If variability in one region is correlated with that in another region, classification based on the former will also separate variable features in the latter. The `focus_mask` option can therefore be used to simplify a classification problem and to test if structural variability in different regions of the complex are correlated.

4.7 Asymmetric Refinement

Symmetry present in a particle offers the advantage of additional averaging and the potential to increase the final resolution of a reconstruction. However, in many cases the individual particles will exhibit deviations from the nominal symmetry due to small distortions, disorder, conformational variability or the presence of pseudosymmetry that only becomes apparent at higher resolution. These deviations may limit the attainable resolution of the fully symmetrized reconstruction. One way to overcome this limitation is to treat the asymmetric subunits of each particle as separate entities (Ilca et al., 2015). Frealign provides an option to perform “asymmetric” refinement and reconstruction to explore if the alignment and classification of

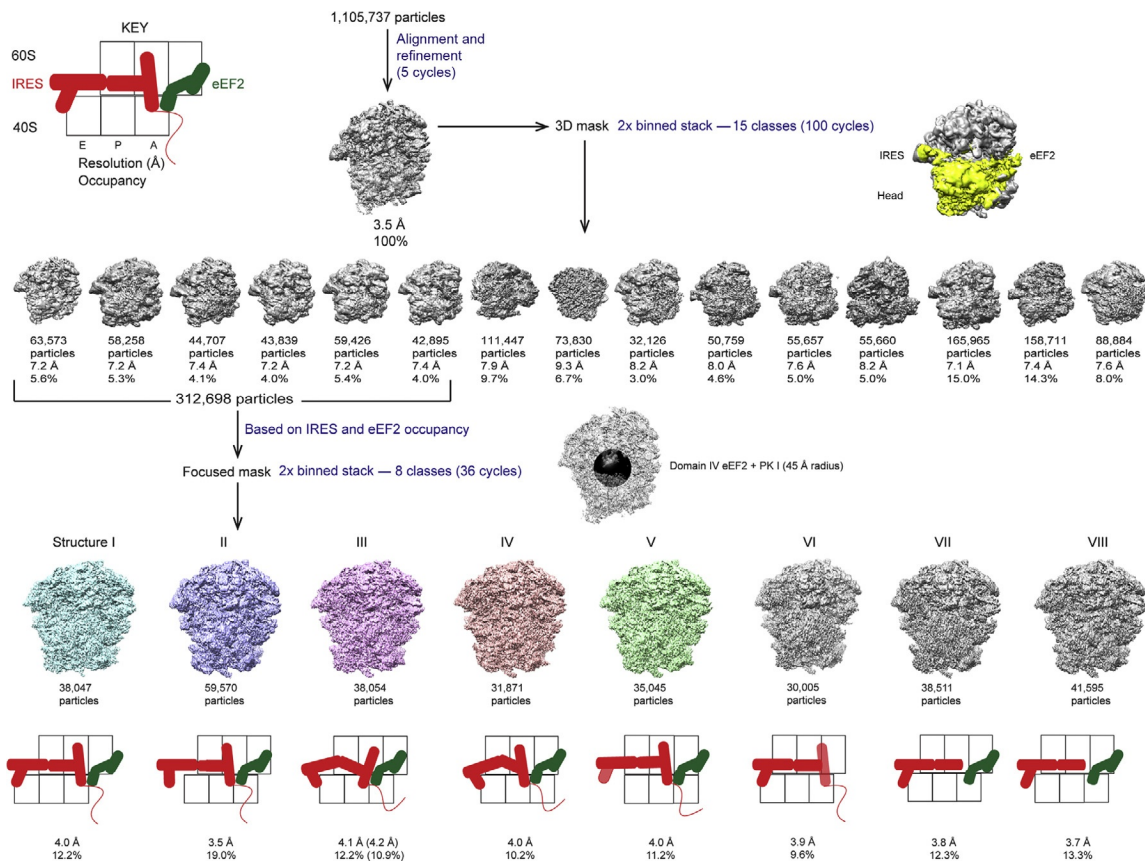


Fig. 6 See legend on next page.

asymmetric units improves the density. To switch to the asymmetric reconstruction mode, users can change the letter of the symmetry symbol from upper case to lower case. For example, to carry out asymmetric refinement of a structure with nominal C2 symmetry, users must specify “c2” (keyword `Symmetry` in `mparameters`). In subsequent refinement cycles, additional lines will appear in the alignment parameter file with alignment results for each symmetry-related orientation of the particle. In the c2 example, there would be two lines of parameters for each particle, thus doubling the number of lines in the parameter file. In the reconstruction step, `Frealign` inserts particle images into the 3D volume according to each line in the parameter file, without applying additional symmetry. Therefore if the alignment parameters on different lines belonging to the same particle differ, this will break the symmetry of the reconstruction. It is therefore possible to obtain a reconstruction that is not perfectly symmetrical when using asymmetric refinement.

The asymmetric mode can be used in two ways. In the first, particles with nominal symmetry experience distortion or disorder that displaces otherwise rigid subunits (or groups of subunits) from their symmetric positions. In this situation `Frealign` can align the subunits individually using a reference with an applied 3D mask (option `mask_file`, see above). For example, if a particle has a nominal C2 symmetry, a suitably designed 3D mask could be used to downweight or completely remove half of the reference density and align

Fig. 6 Example of a classification scheme using 2D and 3D masking. The dataset consisted of 80S ribosomes prepared with the Taura syndrome virus internal ribosome entry site (TSV IRES) and elongation factor 2 (eEF2) (Abeyrathne, Koh, Grant, Grigorieff, & Korostelev, 2016). The complete dataset of 1,105,737 images of 80S–IRES–eEF2 complex was initially aligned against a density map calculated from the atomic model of the non-rotated 80S ribosome bound with 2 tRNAs (PDB: 3J78, Svidritskiy, Brilot, Koh, Grigorieff, & Korostelev, 2014). This initial alignment was performed on data with a pixel size of 1.64 Å and limited to 20 Å resolution, resulting in a 3.5-Å resolution reconstruction. After 5 cycles of refinement the data were 2 × binned (by Fourier cropping using the `resample.exe` tool, new pixel size = 3.28 Å) and subjected to classification into 15 classes using a 3D mask that contained the IRES, eEF2, and head domain of the small subunit. Six of the resulting classes (312,698 particle images) contained density for the IRES and eEF2 and were further classified into eight classes. For this classification, a 2D mask was applied around the ribosomal A site to include IRES pseudoknot I and eEF2 domain IV. The figure shows this mask as a sphere which, when projected according to the orientation of a particle, results in a 2D mask correctly placed on the region of interest. In the case of the 80S–IRES–eEF2 complex, this focused classification resulted in the separation of different translocation states of the IRES, catalyzed by eEF2, as shown schematically below each reconstruction. The structures I–V containing clear density for the IRES and eEF2 are *highlighted in color*.

the particles against the remaining half. The particle images still contain both halves and Frealign will attempt to align each to the masked reference after applying symmetry-derived rotation matrices to transform each half into the correct frame of reference with respect to the masked reference. The resulting reconstruction should display improved density in the half of the reconstruction that survived the masking while the other half should display degraded density, depending on the amount of distortion/disorder present in the particles. It is important to note that the masking can also lead to increased alignment errors, degrading both halves of the density. Users should therefore carefully evaluate the asymmetric reconstructions and explore different masking options (see above).

The asymmetric refinement mode can also be used to classify symmetry-related regions (eg, subunits) in a particle if these differ in their conformation or composition (eg, ligand binding). In this case, classification ($n_{\text{classes}} > 1$, see above) has to be used together with asymmetric refinement. Using masking in 2D or 3D (see above), one of the symmetry-related regions has to be selected by the mask. Frealign will then assign each of the symmetry-related views of a particle to one of the classes using different occupancies in the alignment parameter file. In the resulting reconstructions, the region corresponding to the density inside the mask will contain the classification results while density outside this region will only show class-specific features if the variability inside the mask is correlated with that outside the mask.

Frealign currently does not offer a way to switch from asymmetric refinement back to regular refinement. Using their own scripts, users can reduce the “multiline” parameter files to “single-line” parameter files to continue with regular refinement. As a technical note, the multiple alignment parameters for each particle do not include the additional symmetry-derived transforms that generate each of the symmetry-related views. This means that the alignment parameters in multiline parameter files usually remain fairly similar to each other as differences only indicate the (typically small) differences in the alignments of the symmetry-related views.

4.8 Processing Images of Helical Structures

Frealign includes an option to impose helical symmetry on reconstructions calculated from segments of helical structures and filaments (see chapter “[Cryo-EM Structure Determination Using Segmented Helical Image Reconstruction](#)” by Sachse). The helical symmetry can be selected using symmetry symbol “H” or “HP.” When “H” is specified, Frealign resets the alignment parameters at the beginning of a new refinement cycle to center all helical segments to be within a single “asymmetric unit” of the helical

lattice. This will affect the Phi Euler angle, ie, the third angle listed in the alignment parameter file that determines the rotation of each segment around the helical axis, and the x,y translations. While this is useful in most cases, it prevents the refinement and reconstruction of pseudohelical structures, such as microtubules that have a seam. When symmetry “HP” is specified, the parameters are not reset and seam-sensitive alignments are preserved.

Frealign expects the helical axis to be aligned with the z -axis. The second Euler angle Theta, which describes the out-of-plane alignment of a helical segment, is therefore usually close to 90 degree. Most helical structures are also characterized by a persistence length that indicates how easy it is to bend them. The variability of the first Euler angle Psi, which describes the in-plane rotational alignment of a segment, is therefore limited depending on the value of the persistence length. While Frealign does not use persistence length as a parameter to restrict the angular alignment of segments, the user can define a `STIFFNESS` parameter as part of the helical symmetry section in `mparameters`. A large value for `STIFFNESS` will force the Psi and Theta angles to deviate less from the average values for a filament while a small value allows more variability. For this to work, it is important that segments belonging to the same filament are arranged consecutively in the alignment parameter file and image stack, and that the micrograph identifier (“Film” column in the parameter file) is the same for these segments. Other parameters that must be set when specifying helical symmetry are `ALPHA` (the rotation angle involved when going from one helical subunit to the next), `RISE` (the translation along the helical axis from one subunit to the next), `NSUBUNITS` (the number of unique subunits per segment; for overlapping segments that should only include the number of subunits in the non-overlapping parts), and `NSTARTS` (the number of helical starts present in the structure). Frealign provides a special `mparameters` template for helical image processing that includes these additional parameter keys. Users can access this template using the command `frealign_helical_template`. There is currently no option in Frealign to refine the helical symmetry parameters and users must therefore make sure that the `ALPHA` and `RISE` parameters are correct.



5. TUNING OPTIONS

Frealign offers a number of tuning options that allows users to optimize refinement and classification. These options are typically used only after refinement with the standard (default) parameters has converged,

ie, no further improvement in the reconstructed density is observed when running additional refinement cycles. The following options, which are listed in the expert section in `mparameters`, are available:

- `XSTD`: determines if the input 3D reference should be masked (values larger than 0) or the 3D reference should be used to generate 2D masks (values smaller than 0) to be applied to the particle images before alignment (the masks are not used when calculating reconstructions). The default is 0, which disables this feature. Frealign interprets the provided value as a multiple of the standard deviation of the input 3D reference and values between 2 and 5 are usually appropriate. Higher values will tighten the masks while lower values will loosen them, both in 2D and 3D. Masking can reduce the noise present in the 3D reference or particle images but users must be careful not to overtighten the masks and cut into the 3D reference density or 2D particle density. 3D masking can also be achieved using the `mask_file` option (see above), which is preferred over the `XSTD` option because it provides the user with more control over different aspects of the masking.
- `PBC`: determines the weighting of individual particle images using a B-factor during reconstruction. The applied B-factor is calculated as $B = 4 \times \text{score} / \text{PBC}$ (in \AA^2). Reconstructions will be corrected for the average applied weights and therefore, only relative weight differences between particles are significant. A large value for `PBC` (eg, 100) will effectively remove particle weighting while a small value (10 or smaller) will apply weighting. Users can recalculate reconstructions (using command `frealign_calc_reconstructions`) with different `PBC` values to see which value produces the best results.
- `parameter_mask`: determines which of the five alignment parameters (Psi, Theta, and Phi Euler angles, and x,y translations) will be refined. Normally, all parameters are refined and `parameter_mask` is set to "1 1 1 1 1." However, any of these flags can be set to "0," thereby forcing the corresponding parameter to remain constant during refinement. This allows users to reduce the degrees of freedom during refinement. For example, when images are collected as movies, it is possible to calculate movie sums with different numbers of frames, one with all frames to boost contrast and one with only the early frames to boost high resolution (Campbell et al., 2012). Frealign allows the use of different particle stacks for refinement and reconstruction by using the keywords `raw_images_ref` and `raw_images_rec` instead of `raw_images`. Using this simplified version of an exposure filter (Grant & Grigorieff, 2015b), it is possible to obtain

higher resolution. However, if the movie frames are not accurately aligned there might be small differences in the translational alignments of the particles between the refinement and reconstruction stacks (rotational differences are usually very small and can be ignored). Once the best possible alignments have been obtained using the stack corresponding to a higher exposure, one or two additional refinement cycles using the low-exposure stack can increase the resolution even further. Since only translational alignment is needed, `parameter_mask` should be set to “0 0 0 1 1” to avoid increased errors in the Euler angles due to the lower image SNR of the low-exposure stack. Keeping some of the parameters constant may also be useful in other situations, for example, when some of the alignment parameters are known from the experimental setup, such as in the random conical tilt (Radermacher, Wagenknecht, Verschoor, & Frank, 1987) or orthogonal tilt reconstruction (Leschziner & Nogales, 2006) methods.

- `thresh_reconst`: determines the particle score value below which particles will be excluded from the reconstruction. Normally this value is set to 0 to include all particles. However, if a large fraction of particles are damaged or otherwise compromised and do not contribute high-resolution signal, reconstructions can be improved by excluding them. These particle images tend to receive lower scores than particles that contribute the strongest signal. Users can therefore tune their reconstructions by testing different values for `thresh_reconst` and recalculating the reconstruction (using command `frealign_calc_reconstructions`).
- `FMATCH`: specifies that `Frealign` should also output matching projections after each refinement cycle. The matching projections are stored inside `Frealign`'s scratch directory and will have names that include the pattern `_reproject_`. It is recommended to keep `FMATCH` set to “F” and use the command `frealign_calc_projections` instead to generate match projections, after refinement is completed. Inspecting matching projections may be useful to verify that particle alignment was successful, especially when starting without a set of alignment parameters and `MODE` set to 3 (see above).
- `FBEAUT`: if set to “T,” `Frealign` will apply the specified particle symmetry also in real space. This will not improve the results of the refinement but may improve the appearance of symmetry in a reconstruction if some of the symmetry operators require interpolation to be represented in the orthogonal coordinate system used by `Frealign`. The feature is therefore usually only used when making figures for presentations and publications.

- **FB00ST**: determines if a signed (when set to “T”) or an unsigned (when set to “F”) correlation coefficient is maximized during particle parameter refinement. As explained earlier, refining with the signed correlation coefficient may improve alignments but also increases the chance of overfitting. Users should initially refine with **FB00ST** set to “F” and, after convergence, test if additional refinement cycles with **FB00ST** set to “T” improve the reconstruction. It is important to limit the resolution used during refinement and look for a clear improvement of the FSC curve beyond the resolution limit. Also, an improvement of the FSC should be accompanied by an improvement in the density that can be correlated with known structural features. Users may have to sharpen the reconstructed density using `bfactor.exe` (see below) to observe high-resolution features.
- **beam_tilt_x**, **beam_tilt_y**: allows users to specify a beam tilt that Frealign will include during CTF correction. The values must be given in units of milliradians. Frealign cannot refine these values. However, if the presence of beam tilt is suspected, users can try different values and recalculate the reconstructions (using command `frealign_calc_reconstructions`) to find the best values.
- **FMAG**, **FDEF**, **FASTIG**, **FPART**: setting any of these flags to “T” will enable refinement of the magnification, defocus, and astigmatism, optionally for individual particles (**FPART** set to “T”). Refinement of these parameters is usually not warranted, and users should not use these options in most cases.
- **RBfactor**: alters the correlation function used during refinement. This is not recommended and users should not use this option in most cases.



6. RELATED SOFTWARE

Besides Frealign, there are several other image processing tools that have been developed in the Grigorieff lab and that are freely available for download from the lab web page. Since some of these may be useful in combination with Frealign, they are briefly listed here for reference.

- *Signature*: software to display micrographs and select particles (Chen & Grigorieff, 2007). A semiautomatic mode is available that uses templates to identify particles using an algorithm first developed for FindEM (Roseman, 2004).
- *CTFFIND3/CTFTILT*: software used to determine accurate image defocus present in a micrograph (Mindell & Grigorieff, 2003) of untilted

and tilted samples. A recent update, CTFFIND4 (Rohou & Grigorieff, 2015), offers significant speedups over CTFFIND3.

- *BFACTOR*: a tool to low-pass filter images and 3D density maps, and to estimate and apply a B-factor to bring out high-resolution features in a map. This tool is also included with the FREALIGN distribution.
- *CROP*: a tool to cut out a region of density from a 2D image or 3D volume.
- *DIFFMAP*: a tool to perform amplitude scaling of one 3D map against another and to write out a difference map. The tool can be used for scaling of 3D maps even if these are not aligned with each other. While the difference map is not meaningful in this case, the scaled maps should have similar filtering and B-factor sharpening, making it easier to compare them in terms of quality and high-resolution features.
- *Unblur/Summovie*: software to align frames of movies collected on an electron microscope, and to apply exposure-dependent filtering to the frames to enhance high-resolution signal (Grant & Grigorieff, 2015b).
- *mag_distortion_estimate/correct*: a tool to measure and correct for magnification distortions present in electron microscope images (Grant & Grigorieff, 2015a).

REFERENCES

- Abeyrathne, P., Koh, C. S., Grant, T., Grigorieff, N., & Korostelev, A. A. (2016). Ensemble cryo-EM uncovers inchworm-like translocation of a viral IRES through the ribosome. *eLife*, in press. <http://dx.doi.org/10.7554/eLife.14874>.
- Alushin, G. M., Ramey, V. H., Pasqualato, S., Ball, D. A., Grigorieff, N., Musacchio, A., et al. (2010). The Ndc80 kinetochore complex forms oligomeric arrays along microtubules. *Nature*, *467*, 805–810.
- Bai, X. C., Rajendra, E., Yang, G., Shi, Y., & Scheres, S. H. W. (2016). Sampling the conformational space of the catalytic subunit of human gamma-secretase. *eLife*, *4*, e11182.
- Campbell, M. G., Cheng, A., Brilot, A. F., Moeller, A., Lyumkis, D., Veesler, D., et al. (2012). Movies of ice-embedded particles enhance resolution in electron cryo-microscopy. *Structure*, *20*, 1823–1828.
- Chen, J. Z., & Grigorieff, N. (2007). SIGNATURE: A single-particle selection system for molecular electron microscopy. *Journal of Structural Biology*, *157*, 168–173.
- Crowther, R. A., Henderson, R., & Smith, J. M. (1996). MRC image processing programs. *Journal of Structural Biology*, *116*, 9–16.
- Frank, J., Rademacher, M., Penczek, P., Zhu, J., Li, Y., Ladjadj, M., et al. (1996). SPIDER and WEB: Processing and visualization of images in 3D electron microscopy and related fields. *Journal of Structural Biology*, *116*, 190–199.
- Grant, T., & Grigorieff, N. (2015a). Automatic estimation and correction of anisotropic magnification distortion in electron microscopes. *Journal of Structural Biology*, *192*, 204–208.
- Grant, T., & Grigorieff, N. (2015b). Measuring the optimal exposure for single particle cryo-EM using a 2.6 Å reconstruction of rotavirus VP6. *eLife*, *4*, e06980.
- Grigorieff, N. (1998). Three-dimensional structure of bovine NADH:ubiquinone oxidoreductase (complex I) at 22 Å in ice. *Journal of Molecular Biology*, *277*, 1033–1046.

- Grigorieff, N. (2000). Resolution measurement in structures derived from single particles. *Acta Crystallographica. Section D, Biological Crystallography*, 56, 1270–1277.
- Grigorieff, N. (2002). Single particles always fit the mold. In *Paper presented at Frontiers in structural cell biology: How can we determine the structures of large subcellular machines at atomic resolution?* Asilomar, CA: The Biophysical Society.
- Grigorieff, N. (2007). FREALIGN: High-resolution refinement of single particle structures. *Journal of Structural Biology*, 157, 117–125.
- Harauz, G., & van Heel, M. (1986). Exact filters for general geometry 3-dimensional reconstruction. *Optik*, 73, 146–156.
- Henderson, R., Sali, A., Baker, M. L., Carragher, B., Devkota, B., Downing, K. H., et al. (2012). Outcome of the first electron microscopy validation task force meeting. *Structure*, 20, 205–214.
- Ilca, S. L., Kotecha, A., Sun, X., Poranen, M. M., Stuart, D. I., & Huiskonen, J. T. (2015). Localized reconstruction of subunits from electron cryomicroscopy images of macromolecular complexes. *Nature Communications*, 6, 8843.
- Kucukelbir, A., Sigworth, F. J., & Tagare, H. D. (2014). Quantifying the local resolution of cryo-EM density maps. *Nature Methods*, 11, 63–65.
- Leschziner, A. E., & Nogales, E. (2006). The orthogonal tilt reconstruction method: An approach to generating single-class volumes with no missing cone for ab initio reconstruction of asymmetric particles. *Journal of Structural Biology*, 153, 284–299.
- Li, X., Grigorieff, N., & Cheng, Y. (2010). GPU-enabled FREALIGN: Accelerating single particle 3D reconstruction and refinement in Fourier space on graphics processors. *Journal of Structural Biology*, 172, 407–412.
- Liang, B., Li, Z., Jenni, S., Rameh, A. A., Morin, B. M., Grant, T., et al. (2015). Structure of the L-protein of vesicular stomatitis virus from electron cryomicroscopy. *Cell*, 162, 314–327.
- Lyumkis, D., Brilot, A. F., Theobald, D. L., & Grigorieff, N. (2013). Likelihood-based classification of cryo-EM images using FREALIGN. *Journal of Structural Biology*, 183, 377–388.
- Matthews, B. W. (1968). Solvent content of protein crystals. *Journal of Molecular Biology*, 33, 491–497.
- Mindell, J. A., & Grigorieff, N. (2003). Accurate determination of local defocus and specimen tilt in electron microscopy. *Journal of Structural Biology*, 142, 334–347.
- Morais, M. C., Kanamaru, S., Badasso, M. O., Koti, J. S., Owen, B. A., McMurray, C. T., et al. (2003). Bacteriophage phi29 scaffolding protein gp7 before and after prohead assembly. *Nature Structural Biology*, 10, 572–576.
- Park, E., Menetret, J. F., Gumbart, J. C., Ludtke, S. J., Li, W., Whynot, A., et al. (2014). Structure of the SecY channel during initiation of protein translocation. *Nature*, 506, 102–106.
- Petersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., et al. (2004). UCSF Chimera—A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25, 1605–1612.
- Radermacher, M., Wagenknecht, T., Verschoor, A., & Frank, J. (1987). Three-dimensional reconstruction from a single-exposure, random conical tilt series applied to the 50S ribosomal subunit of *Escherichia coli*. *Journal of Microscopy*, 146, 113–136.
- Rohou, A., & Grigorieff, N. (2015). CTFFIND4: Fast and accurate defocus estimation from electron micrographs. *Journal of Structural Biology*, 192, 216–221.
- Roseman, A. M. (2004). FindEM—A fast, efficient program for automatic selection of particles from electron micrographs. *Journal of Structural Biology*, 145, 91–99.
- Rosenthal, P. B., & Henderson, R. (2003). Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. *Journal of Molecular Biology*, 333, 721–745.

- Sindelar, C. V., & Grigorieff, N. (2012). Optimal noise reduction in 3D reconstructions of single particles using a volume-normalized filter. *Journal of Structural Biology*, *180*, 26–38.
- Stewart, A., & Grigorieff, N. (2004). Noise bias in the refinement of structures derived from single particles. *Ultramicroscopy*, *102*, 67–84.
- Svidritskiy, E., Brilot, A. F., Koh, C. S., Grigorieff, N., & Korostelev, A. A. (2014). Structures of yeast 80S ribosome-tRNA complexes in the rotated and nonrotated conformations. *Structure*, *22*, 1210–1218.
- van Heel, M., Harauz, G., Orlova, E. V., Schmidt, R., & Schatz, M. (1996). A new generation of the IMAGIC image processing system. *Journal of Structural Biology*, *116*, 17–24.
- Wolf, M., DeRosier, D. J., & Grigorieff, N. (2006). Ewald sphere correction for single-particle electron microscopy. *Ultramicroscopy*, *106*, 376–382.
- Yang, Z., Fang, J., Chittuluru, J., Asturias, F. J., & Penczek, P. A. (2012). Iterative stable alignment and clustering of 2D transmission electron microscope images. *Structure*, *20*, 237–247.