STOPGAP 0.3.1 Basics

STOPGAP is a subtomogram package written in MATLAB and is largely compatible with TOM/AV3 motive lists and wedge lists. The general workflow is the same, and it has a few TOM dependencies, but at this point there is pretty much no code from TOM/AV3.

STOPGAP 0.3.1 is available on the modules as “STOPGAP/0.3.1”; it is recommended that you add the module to your .bashrc.

Workflow

The basic workflow of stopgap is to first generate a motive list and extract subtomograms. Then, subtomogram averaging jobs can be set up using stopgap_job_parser.sh, which parses input parameters into a correctly formatted .star file. The parser also tries to check for dependencies.

The paramfile is then run using run_stopgap.sh; this can be run locally or on the SGE-cluster. Remember to check that the total_cores parameter in run_stopgap.sh matches that in the param file.

Subtomogram averaging has 3 steps: subtomogram alignment, parallel averaging, and final averaging. Completion of these steps is saved into the param file at the end of each step. When all jobs are completed, the cores are released the software ends.

Executables

STOPGAP is run using 3 bash scripts: the first is for subtomogram extraction (extract_subtomo_local.sh), the second is for generating parameter files (stopgap_job_parser.sh), and the third is for running the subtomogram averaging workflow (run_stopgap.sh).

I would suggest that you copy these scripts into each folder that you want to perform averaging in.

Files and folders

STOPGAP job parameters are stored in the parameter file. This is a .star file that is generated by running stopgap_job_parser.sh.

The general folder layout I use is:

checkjobs:	folder for communication between nodes about iterations
combinedmotl:	folder for complete motivelists
complete:	folder for communication between nodes about steps
motls:	folder to hold temporary alignment files
otherinputs:	folder to hold masks and wedgelist
ref:	folder to hold references
stats:	folder to hold subtomogram grey value statistics
subtomograms:	folder to hold subtomograms

Volumes in stopgap are all .em files. Motive lists and basic wedgelists follow TOM/AV3 formatting. Advanced wedgelist which allow for proper CTF- and exposure-weighting during alignment and averaging are MATLAB .mat files; these can be generated from tomoman tomolists using the "stopgap_wedgemask_generate_defocus_wedgelist.m" script in matlab.

Scoring

STOPGAP uses real-space correlation-based alignment. There are three scoring functions: the fast-local correlation function (flcf), a weighted flcf (flcf-weighted), and a Pearson correlation.

The two flcf functions calculate a 3D correlation map which is used to determine the score and shift; shifts can be restricted using a ccmask (see below).

The Pearson correlation is combined with a maximization search to explicity refine the shifts. These are not restricted. This type of refinement can be immensely slow.

Masks

Generally, two types of masks are required for subtomogram averaging: an alignment mask and a cross-correlation mask (ccmask). The alignment mask is the mask applied to the reference and can be binary or include a Gaussian edge; the filename set in the parser using “maskname”.

The ccmask should be a binary mask; it’s shape defines the shifting with respect to the center of the box. The center is always (floor(boxsize)/2 + 1), so a box of 64 has a center of 33. If you wish to restrict shifts in all directions by 4 pixels, the ccmask should be a sphere with radius of 4 pixels. If you wish to restrict shifts in Z by 4 pixels, but allow 6 pixels in the XY-plane, use a cylinder with radius 6 and height 8.

Angular search

There are two modes of angular search: cone-search and arbitrary euler angles. Cone search is what is typically used in other subtomogram packages: the angincr (angular increment) and angiter (angular iterations) control searching the cone while the phi_angincr and phi_angiter control the in-plane angle. For the arbitrary euler search, you must define three euler axes and their angular increments and iterations.

Bandpass filter

The bandpass filter is set using integer Fourier pixels; lp and hp refer to lowpass and highpass while rad and sigma refer to the filter radius and falloff. To calculate Fourier pixels from resolution:

Fourier pixels = (boxsize*pixelsize)/resolution

In general, it's fine to use lp_sigma=3, hp_rad = 1, hp_sigma = 2. The lp_rad is the most important; for initial alignment chose something conservative like < 35 Angstroms. It's a good rule of thumb to keep the lp_rad at a resolution less than 0.5 of the FSC.