Running fast_best_match.m

January 26, 2015

1 Introduction

The MATLAB script fast_best_match.m performs a reconstruction of the particle images using the best-match alignment strategy and subspace approximations for fast SSD calculations. The general flow of the algorithm is:

1. Get user-set parameters from fast_best_match_config.txt
2. Do outer loop until iteration number > numruns
   (a) Set up the subspace for the particle images (only in 1st iteration)
   (b) Set up the initial model parameters and subspace for the structure projections
   (c) Set up other parameters, including translation domain setup, calculation of norms of approximated images...
   (d) Do inner loop (reconstruction using approximated images and projections) until convergence
      i. Calculate the norms of the approximated projections
      ii. Calculate the inner products between all image and transformed projection basis elements
      iii. Calculate the SSDs and find the best alignment parameters
      iv. Update noise variances
      v. Update projection subspace
      vi. Update projection coefficients
      vii. Update the reconstruction
   (e) Calculate reconstruction using final alignment parameters and original images
3. Reconstruct with largest possible mask (for fairer fsc calculations) and, if desired, save aligned images

In summary, the inner loop calculates the alignment and reconstruction using the approximated images and approximated projections. The outer loop reruns the inner loop reconstruction using the final reconstruction result from the last outer loop iteration as the new initial model.

All input parameters are specified in fast_best_match_config.txt. At the absolute minimum, the user must specify the file for the particle images and an initial 3D model.

The following describes the parameters that can be set using the config file.

2 Particle Data and Parameters

- imfile: The name of the file of the particle images. This must be a .mat file that includes a matrix variable that contains the N particle images (imvar, see below) and a variable called "ctfinds" which is an (N,1) array that notes the index of the ctf class for each image. If a PCA has already been performed on the data, the variables must be saved as "coeffim" for the image basis, "scoreim" for the representation of each image in the image basis, and "latentim" for the eigenvalues. If these variables do not yet exist in imfile, the script will run a PCA and append the results to imfile.
- imvar: The variable name for the stack of particle images in imfile.
- substep: Parameter for subsampling the number of particle images. Number of images from the stack in imfile will be reduced by a factor of substep. (This parameter was used for testing the code to cut down on the amount of data and run faster). If substep is set to 0, no subsampling is performed. Default: 0
- reconhalf: Flag for running a reconstruction using half the particle images. Every other image will be included, starting from reconstartind (see below). 0 = use all the images, 1 = use half the images. Default: 0
- reconstartind: Index of the first image for doing a half-set reconstruction. Set to either 1 (to get image indices 1:2:N) or 2 (to get image indices 2:2:N).

Example usage for reconstructing half the data:
imfile = directory/imagefilename.mat
imvar = noisyims
reconhalf = 1
reconstartind = 1

3 Initial Model

For the initial model, the user must either specify 1) an initial model .mat file that already contains the parameters in the right format, or 2) an initial structure file with associated parameters set in the config file.

- initprojfile: The name of the file with the initial model and related parameters. This would have already been generated by a previous run of the algorithm that used the same initial structure and parameters.
- structfile: The name of the file with the initial 3D structure. This can be a .mrc file or a .mat file that contains just the structure.
- sampdeg: The angular sampling in degrees for the projection directions. If sampdeg = 0, coordfile must be specified.
- coordfile: The file with the coordinates for the projection directions. If coordfile is not specified, then sampdeg must be specified.
- ctffile: The file with the ctfs. The ctfs must be stored in one variable as a stack of images with the same size as the particle data. If no ctffile is specified, it is assumed that the images were already ctf coffected, and no ctf is applied.

- ctfvar: The name of the variable storing the ctfs in ctffile. If no variable is specified, it is assumed that the file contains only the one variable for the ctfs.
- savename: The name of the file in which to save the initial model and parameters. If no name is specified, structfile will be used, with identifying parameters added to the name of the file.
- structvoxelres: The voxel size of the 3D structure in Angstroms. If structfile is a .mrc, then structvoxelres need not be specified as the voxel size is determined from the file header. If structfile is a .mat, then structvoxelres must be specified.
- downsample: Parameter for downsampling the structure in structfile by a factor of downsample. Must be set correctly so that the structure size matches the particle image size.
- lpf: Low-pass filter cutoff in Angstrom for low-pass filtering the initial structure. If lpf is not specified, no low-pass filtering is performed. Default: 0 (no low-pass filtering)
- pf: Flag for phase-flipping the ctfs. 0 = do not phase-flip, 1 = phase-flip. Default: 0.
- sigma: Determines width of the Gaussian used in low-pass filtering the structure. Default: 1.
- addnoise: Flag for adding noise to the initial projections. 0 = no added noise, 1 = add noise. Default: 0.
- SNR_dB: Target SNR in dB for initial projections when random noise is added. Default: 0.
- pixfromedge: The number of pixels between the edge of the image and the edge of the spherical mask used for calculating the SSDs and projections/reconstructions. Default: 1

Example usage, for when the initial model .mat file already exists:

```
initprojfile = directory/initfilename.mat
```

Example usage, to create an initial model file given an initial structure and some specified parameters

```
structfile = directory/initvolume.mrc
sampdeg = 10
ctffile = directory/ctfs.mat
downsample = 2
lpf = 60
pixfromedge = 4
```

4 System Parameters

- maxmem: Maximum amount of memory in MB, available to run the algorithm. Default: Based on system physical memory available - 1GB
- numthreads: Number of MATLAB workers to use in parallel on parfor calls. Default: Based on default number of works in "local" cluster profile
- dispflag: Flag for displaying MATLAB figures during the run. 0 = don't display, 1 = display. Default: 0

Example usage:

```
maxmem = 26624
numthreads = 6
dispflag = 0
```

5 Algorithm Parameters

- f: The threshold which determines the subspace sizes. The 2nd difference of the eigenvalues is calculated and the number of components is chosen as the point at which the absolute value of the 2nd difference is less than f. Default value: 0.0001
- numruns: Number of times to run the outer loop. Default value: 2
- maxnumiter: Number of times to run the inner loop. Default value: 15
- convtol: Tolerance value for when the inner loops is considered converged. The algorithm is considered converged when for every projection, the norm of the difference between the last iteration and current iteration divided by the norm of the projection is less than convtol. Default value: 0.01
- normprojint: Flag for whether or not to normalize the intensities of the projections to match the particle images. Default value: 1

Example usage:

f = 0.0001
numruns = 2
maxnumiter = 12
convtol = 0.01
normprojint = 1

6 Transformation Search Parameters

- rotstart: Starting angle in degrees for in-plane rotational search. Default value: 0°
- rotstep: Stepsize in degrees for in-plane rotational search. Default value: 2°
- rotend: Last angle in degrees for in-plane rotational search. Default value: 359.9°
- transmax: Maximum translational search in pixels. The overall search domain covers a translational grid of +/- transmax pixels. Default value: 4 (81 translational search values)
- transdelta: Stepsize in pixels for the translational search: Default value: 1
- transwidth: The size of the neighborhood in which to search for the best translation in the current iteration. The neighborhood for the local translation search is +/- transwidth. Default value: 1

Example usage:

rotstart = 1
rotstep = 2
rotend = 359.9
transmax = 6
transdelta = 1
transwidth = 1

7 Output Parameters

- savemrc: Parameter for saving the final reconstruction as a .mrc. savemrc should be set to the voxel size of the reconstruction. The structure is saved to the file "fbm_recon.mrc". If savemrc is set to 0, the reconstruction is not saved as a .mrc. Default value: 0.
- alignims: Flag to denote whether or not to save the final aligned particle images as output. 0 = don't align, 1 = align. Default value: 0.

Example usage:
savemrc = 3.09
alignims = 1

8 Algorithm Outputs

Results are saved in .mat files at the end of each iteration of the inner loop (run_#_iter_#.mat) and at the end of the outer loop (imfile_fastbm_numimcoeffs_numprojcoeffs_rot_trans_#x.mat). Some of the important variables saved in the final output files are:

- recon: The final reconstruction using the largest mask possible.
- aligned_ims: Stack of the images transformed to align to its best-match projection. Only saved if alignims set to 1 in config file.
- coord_axes: The coordinates of the projection directions.
- projinds: An (N,1) array of indices (into coord_axes) of the best-match projection direction assignment for each image.
- rots: An array of the in-plane rotational angles searched.
- rotinds: An (N,1) array of indices denoting the index of the angle in rots for the best in-plane rotation of each image.
- trans: A (# translations,2) matrix giving the translations searched.
- transinds: An (N,1) array of indices giving the index of the translation in trans for the best translation of each image.
- ssdtimes: An array of the times for calculating the ssds in each iteration.
- itertimes: An array of the times for calculating each inner loop iteration.
- totaltime: Total wall time for calculating an iteration of the outer loop.
- structure_final: The reconstruction at the end of an outer loop iteration, using the original particle images and the given mask.
- structure: The reconstruction when the inner loop converges from the approximated images and the final alignment parameters.

The first four variables (recon, aligned_ims, coord_axes, and projinds) are required as input for running Hemant's heterogeneity code.