Running Reconstruction Functions

Nicha C. Dvornek

June 26, 2015

1 Introduction

This manual describes the operation of the three reconstruction functions in the cryo3d/script directory: subspaceEM.m, fast_best_match.m, and fast_multi_ref.m. The functions subspaceEM.m and fast_best_match.m are for running single particle reconstructions from homogeneous data, while fast_multi_ref.m is for running multi-reference reconstructions from heterogeneous data. All these methods use subspace approximations for the particle images and the projection images to speed up the reconstruction runtime.

subspaceEM.m runs a fast single particle reconstruction using the EM algorithm, which gives weights to each of the possible image alignments. The algorithm is described in detail in the JSB article "SubspaceEM: A fast maximum-a-posteriori algorithm for cryo-EM single particle reconstruction" by Dvornek, Sigworth, and Tagare. The flow of the algorithm is similar to that described below for fast_best_match.m, but with the appropriate steps and calculations to solve the problem using an EM framework. Please see the paper for further details.

fast_best_match.m runs a fast single particle reconstruction using the best-match alignment strategy. 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.

Finally, fast_multi_ref.m runs fast multi-reference reconstructions from heterogeneous particle data. Under the EM framework, each image is assigned a weight of coming from each structure, and the best-match projection direction is found for each image using these weights. The flow of the algorithm is similar to that presented above for fast_best-match.m, with an additional initialization step for creating the initial structures. First, the particle stack is randomly divided into the desired number of structures to reconstruct. For each random set of images, a best-match alignment and reconstruction to a single (low-pass filtered) input structure is performed to create each initial structure.

2 Configuration File

All input parameters are specified in a configuration text file that is passed as a parameter to the reconstruction function, for example, config_file.txt. At the absolute minimum, the user must specify the file for the particle images and an initial 3D model. For an example configuration file for fast_multi_ref.m, see cryo3d/doc/fast_multi_ref_config_eg.txt.

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

2.1 Particle Data and Related Parameters

- imfile: (REQUIRED) Filename of the particle image stack in MRC format.
- imreconfile: Filename of the particle image stack in MRC format used to create the final reconstruction and alignment based on the algorithm output using the stack in imfile. If no file is specified, then imreconfile = imfile.
- ctffile: Filename of the CTF data in .mat format, corresponding to the particle stack in imfile. The CTF data file must contain 1) a 3D matrix called *ctfs* which specifies the CTF for each defocus class as a 2D image in Fourier space stacked along the third dimension, and 2) a 1D vector called *ctfinds* which specifies the index of the defocus class for each particle image. That is, if the ith particle image belongs to the jth defocus class, then ctfinds(i) = j, which corresponds to CTF image ctfs(:,:,j). The file output by fit_ctfs.m contains these variables and can be used as input for ctffile. If no CTF data file is specified, it is assumed that the particle images have already been CTF-corrected.
- 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/particle_stack.mrcs
reconhalf = 1
reconstartind = 1

2.2 Initial Model

For the initial model, the user must specify an initial structure file and possibly associated parameters set in the config file.

- structfile: (REQUIRED) 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.
- maskfile: A user-specified mask for the structure in MRC format.
- sampdeg: The angular stepsize in degrees for uniform sampling of the projection directions on the sphere. If sampdeg = 0, coordfile must be specified.
- coordfile: The file in .mat format with the coordinates for the projection directions, stored in variable named coord_axes. If coordfile is not specified, then sampdeg must be specified. The file output by coordinate_axes.m can be used as input for coordfile.
- 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.
- 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
- pf: Flag for phase-flipping the ctfs. 0 = do not phase-flip, 1 = phase-flip. Default: 0.
- 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.

Example usage:
structfile = directory/initvolume.mrc
coordfile = directory/coords.mat
pixfromedge = 4

2.3 System Parameters

- maxmem: Maximum amount of memory in MB, available to run the algorithm. Default: System physical memory available - 1GB.
- numthreads: Number of MATLAB workers to use in parallel on parfor calls. Default: 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. Note: This flag does nothing in fast_best_match.m and fast_multi_ref.m.

Example usage:
maxmem = 26624
numthreads = 6
dispflag = 0

2.4 Algorithm Parameters

- t: The threshold which determines the subspace dimensions based on Cattell's scree test. 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 t. Default: 0.0001
- numruns: Number of times to run the outer loop. Default: 2
- maxnumiter: Maximum number of times to run the inner loop if convergence criteria is not met according to convtol (see below). Default: 15
- convtol: Tolerance value for when the inner loop 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 current projection is less than convtol. Default: 0.01
- normprojint: Flag for whether or not to normalize the intensities of the projections to match the particle images. 0 = no normalization, 1 = normalize. Default: 1

Example usage:
t = 0.0001
numruns = 2
maxnumiter = 12
convtol = 0.01
normprojint = 1

2.5 Transformation Search Parameters

- rotstart: Starting angle in degrees for in-plane rotational search. Default: 0°
- rotstep: Stepsize in degrees for in-plane rotational search. Default: 2°
- rotend: Last angle in degrees for in-plane rotational search. Default: 359.9°
- transmax: Maximum translational search in pixels. The overall search domain covers a translational grid of +/- transmax pixels. Default: 4 (81 translational search values)
- transdelta: Stepsize in pixels for the translational search. Note only integral values are supported. Default: 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: 1

Example usage:
rotstart = 1
rotstep = 2

rotend = 359.9
transmax = 6
transdelta = 1
transwidth = 1

2.6 Parameters Specific to fast_best_match.m

- alignims: Flag to denote whether or not to save the final aligned particle images as output. 0 = don't align, 1 = align. Default: 0.

Example usage:
alignims = 1

2.7 Parameters Specific to fast_multi_ref.m

- numstruct: Number of structures to reconstruct. Default: 2.

Example usage:
numstruct = 4

3 Algorithm Outputs

3.1 Outputs Common to all Algorithms

- *imfile*_pca.mat: Results from PCA of the particle stack in imfile. If one of the reconstruction algorithms has already been run before with the same particle stack, then the current run uses the results saved in this .mat file and does not need to perform PCA again.

- *structfile*_#ctf_#d.mat: Initial model and parameters based on the structure in structfile. The first # is the number of defocus classes, and the second # is the number of projection directions. If phase flip flag pf = 1, then the file name would be appended with _pf. The file is saved in the same directory as structfile.

- run_#_iter_#.mat: Intermediate results after each run of the inner loop. Contains many variables, including the intermediate structure (structure), projection basis (projbasis), and projection coefficients (projcoeffs).

3.2 Outputs for subspaceEM.m

- subspaceEM_recon.mrc: The final 3D reconstruction. This reconstruction is performed using the largest mask possible.

- subspaceEM_recon_masked.mrc: If a user-specified mask is used, the algorithm outputs a masked version of the final reconstruction.

- *imfile*_subspaceEM_#_#_*rotstep*d_*transparams*t_#x.mat: Output after completing a run of the outer loop. The first two #'s denote the dimensions for the particle image and projection subspaces, respectively. *Transparams* is 3 numbers denoting the values for transmax, transdelta, and transwidth. The last # is how many times the outer loop has been run. Some of the important variables saved in the final output files are:

- recon: The final reconstruction using the largest mask possible.
- coord_axes: The coordinates of the projection directions.
- proj_est: Stack of the final projection (with CTF) estimates from the aligned images, used to calculate the final reconstruction. Projection images are ordered first by the projection direction, then by the CTF image. So if there are F defocus classes, the first F images in proj_est correspond to the first projection direction given by coord_axes. That is, proj_est(:,:,1) indices the projection image corresponding to the projection direction given by coord_axes(:,1) with CTF indexed by ctfs(:,:,1) applied, and proj_est(:,:,F) indices the projection image corresponding to the projection direction given by coord_axes(:,1) with CTF indexed by ctfs(:,:,F) applied. The size of the stack (3rd dimension) is then # directions $\times$ # CTFs.
- lpcs_vals: An array of the (non-zero) latent probabilities corresponding to the image alignment parameters indexed by iminds, projinds, rotinds, and transinds (see below). So, lpcs_vals(i) is the latent probability for the particle image indexed by iminds(i) aligned to the projection indexed by projinds(i), with rotation parameter indexed by rotinds(i) and translation parameter indexed by transinds(i).
- iminds: An array of particle image indices corresponding to the entries in lpcs_vals. length(iminds) = length(lpcs_vals).
- projinds: An array of indices (into proj_est) of the projection image (with CTF) corresponding to the entries in lpcs_vals. length(projinds) = length(lpcs_vals).
- rots: An array of the in-plane rotational angles searched.
- rotinds: An array of indices denoting the index of the angle in rots for the in-plane rotation corresponding to the entries in lpcs_vals. length(rotinds) = length(lpcs_vals).
- trans: A (# translations,2) matrix giving the translations searched.
- transinds: An array of indices giving the index of the translation in trans for the translation of corresponding to the entries in lpcs_vals. length(transinds) = length(lpcs_vals).
- alphas: A (# projections,1) array of estimated prior probabilities of a particle image be generated from each projection image (the mixing coefficients in the Gaussian mixture model).
- 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.
- structure: The reconstruction when the inner loop converges from the approximated images and the final alignment parameters.

3.3 Outputs for fast_best_match.m

Files the algorithm produces:

- fbm_recon.mrc: The final 3D reconstruction. This reconstruction is performed using the largest mask possible.

- fbm_recon_masked.mrc: If a user-specified mask is used, the algorithm outputs a masked version of the final reconstruction.

- *imfile*_fastbm_#_#_*rotstep*d_*transparams*t_#x.mat: Output after completing a run of the outer loop. The first two #'s denote the dimensions for the particle image and projection subspaces, respectively. *Transparams* is 3 numbers denoting the values for transmax, transdelta, and transwidth. The last # is how many times the outer loop has been run. Most of the variables saved are the same as in the output file for subspaceEM.m. Some important new variables, and variables with the same name but slightly different structure, are:

  - aligned_ims: Stack of the images transformed to align to its best-match projection. Only saved if alignims set to 1 in config file.
  - projinds: An (N,1) array of indices of the best-match projection assignment for each image.
  - rotinds: An (N,1) array of indices denoting the index of the angle in rots for the best in-plane rotation of each image.
  - transinds: An (N,1) array of indices giving the index of the translation in trans for the best translation of each image.

- fbm_aligned_ims.mat: Aligned particle images and associated parameters. Output file is created if flag alignims is set to 1. This output file can be used as input for running Hemant's heterogeneity code (tested with CTF-corrected version). The variables saved are recon, aligned_ims, coord_axes, and projinds, as described above.

3.4 Outputs for fast_multi_ref.m

- fmr_recon_#.mrc: The final 3D reconstruction for the structure indexed by #. This reconstruction is performed using the largest mask possible.

- fmr_recon_#_masked.mrc: If a user-specified mask is used, the algorithm outputs a masked version of each final reconstruction indexed by #.

- *imfile*_fastmr_#_#_*rotstep*d_*transparams*t_#x.mat: Output after completing a run of the outer loop. The first two #'s denote the dimensions for the particle image and projection subspaces, respectively. *Transparams* is 3 numbers denoting the values for transmax, transdelta, and transwidth. The last # is how many times the outer loop has been run. Most of the variables saved are the same as in the output file for subspaceEM.m. Some important new variables, and variables with the same name but slightly different structure, are:

  - proj_est: Stack of the final projection (with CTF) estimates from the aligned images, used to calculate the final reconstruction. The order of the images in the stack is all projections with CTFs from the 1st structure, all projections from the 2nd structure, etc. Within each structure grouping, the order is the same as above, i.e., projections along 1st direction with all CTFs, projections along 2nd direction with all CTFs, etc. Size of the stack (3rd dimension) is then # directions × # CTFs × # structures.
  - projinds: An (# images × # structures,1) array of indices of the best-match projection assignment for each image. The order of the values in the array correspond to the indices for each structure for the 1st image, the indices for each structure for the 2nd image, etc.

- rotinds: An (# images × # structures,1) array of indices denoting the index of the angle in rots for the best in-plane rotation of each image. Same ordering as in projinds.
- transinds: An (# images × # structures,1) array of indices giving the index of the translation in trans for the best translation of each image. Same ordering as in projinds.
- alphas: A (# structures,1) array of estimated prior probabilities of a particle image being generated from each structure (the mixing coefficients of the Gaussian mixture model).
- structure_final: A 4D matrix storing the reconstructions at the end of the outer loop iteration, using the original particle images. The sth structure is given by structure_final(:,:,:,s).
- structure: A 4D matrix storing the reconstructions when the inner loop converges from the approximated images and the final alignment parameters. The sth structure is given by structure(:,:,:,s).