EMPFTCC.DOC

(last update Aug 24, 1993)

CONTENTS

A. INTRODUCTION
B. PROGRAM INPUT
C. EXAMPLE EMPFTCC BATCH JOB
D. FINAL NOTES FOR PURDUE USERS
E. FLOW CHART FOR EMPFTCC PROGRAM

A. INTRODUCTION

EMPFTCC is used after EMPFT to determine theta,phi,omega orientation and x,y origin parameters for images of particles (DEFAULT is for icosahedral particles). This uses cross-correlation procedures to compare "raw" (unfiltered or filtered) image data against a data base of reference projection images produced from a 3D reconstruction or model (e.g. from x-ray crystallography).

EMPFTCC is a much faster and more powerful orientation search procedure than the EMICOFV program which has been used for many years to determine initial orientation parameters for icosahedral particles. EMICOFV uses a brute-force common-lines procedure. Note however, that EMPFTCC CANNOT substitute for EMICOFV-type routines if there is no model 3D data from which reference data base can be produced. Thus, EMICOFV (or some equivalent) is still needed when starting a structure determination for a new particle of unknown structure. A key to the success of the EMPFTCC procedure is that comparisons made between the raw and model data can include 100% of the available data, or less if you choose (in filtering mode to reduce the effects of noise). In common lines procedures, the fraction of available data rapidly drops off as a function of resolution. Thus, in principle, the EMPFTCC technique is not resolution limited.

EMPFTCC converts each raw image from Cartesian to polar format. Each image is subdivided into a series of annuli (usually each 1.0 pixel wide). All annuli are sampled NROT times, with the interpolation performed at a fine enough interval to prevent loss of pixel resolution. In this way, the annuli corresponding to the lowest radii in the particle image are over-sampled by a large amount. A Fourier Transform (PFT) of the image is calculated by Fourier transforming each separate annulus. The PFT therefore provides the Fourier transform of the density distribution at each

radius of the particle. This provides a particularly useful representation of the data which allows rapid and sensitive determination of the particle orientation. The program achieves high speed because it determines the particle parameters in several stages rather than in a single, brute force step. Theta and phi are determined first, with an ambiguity in the sign of phi (corresponding to an ambiguity in the hand of the particle). Next, the sign of phi (relative hand) and the omega angle are determined by a real-space, rotational cross-correlation between the projected model and the image. This correlation is typically performed only for a restricted range of particle radii, IRAD1,IRAD2, selected by the user. It is best, for example, to set IRAD2 to a value close to or just inside the outer particle edge. Setting IRAD2 too large adds unnecessary background noise and hence reduces the sensitivity of the search procedure. Also, experience shows that values of IRAD1 >> 0 help because the low radii in projected images typically contain a significant contribution from the nucleic acid core (for mature virions) which does not generally carry much signal corresponding to the icosahedral orientation.

Correlations can be computed with Fourier-filtered image data (RES_MIN,RES_MAX) and also for data in which the low-order Fourier Bessel terms are omitted: the Jo terms are usually left out of the calculations because they are so dominant and don't convey orientation information.

Currently, EMPFTCC outputs a variety of data including the standard EMICO.DAT data format (in the file EMPFTCC.DAT) and statistics on the correlation coefficients as a function of radius (EMPFTCC.RADS) and resolution (EMPFTCC.RES1 and EMPFTCC.RES2). The variable, ILIST, is used to designate what output is produced.

Two different correlation coefficients are listed out in the EMPFTCC.DAT file after the THETA,PHI,OMEGA,X,Y parameters: the first is the global correlation coefficient between the PFTs of the raw and reference images (it incorporates the effects of any radii cutoffs in real-space and resolution limits in reciprocal space). The second is a global real-space correlation coefficient between the raw and reference images, which is only calculated for data between radii IRAD1 and IRAD2.

The program EMPFTREF is a combination of the algorithms in EMPFT and EMPFTCC, which can be used essentially like the two separate programs, or as a refinement routine, which allows finer searches of selected theta,phi regions.

******DISCLAIMER******

To date (Aug 93) this program and those like it (EMPFT, EMICOPFT, EMPFTREF, and EMICOPFTCC) have worked remarkably well under a wide variety of conditions. These routines seem to work flawlessly AS LONG AS THE MODEL IS "ON THE RIGHT TRACK". If your model is ridiculous or "trash", then don't expect miracles! Also, we currently find that, for icosahedral particle data which shows very weak enantiomorphic features, the model MUST INCLUDE some (correct) enantiomorphic character or the refinement will lead to a 3D map that exhibits mirror line symmetry about the equatorial line. This is NOT a problem for icosahedral particles with handed surface lattices (e.g. T=7 papovairuses) because the arrangement of morphological units is clearly enantiomorphic even at very low resolution (>5nm). In many instances, enantiomorphic features do not become apparent until much higher resolutions (<3-4nm), and therefore proper refinement of data cannot proceed until the model incorporates information at the higher resolutions. In tricky situations, it is still advisable to use programs like EMICOGRAD with small data sets (5 particles or less) to try and make sure that the particles are oriented with respect to a consistent hand. A crude 3D reconstruction computed from such a limited data set, although noisy, may give a much better model for further refinement with the EMPFT* programs.

B. PROGRAM INPUT

1. FILE CONTAINING LATEST PARTICLE PARAMETERS (A FORMAT)
2. INPUT FILENAME FOR PACKED RAW DATA IMAGES (A FORMAT)
3. INPUT FILENAME FOR REFERENCE PFTS (A FORMAT)
4. INPUT FILENAME FOR REFERENCE PROJECTIONS (A FORMAT)
5. J1,IRAD1,IRAD2,RES_MIN,RES_MAX,ILIST,SIG (3I,2F,I,F)
6. IMG_NUM1,IMG_NUM2,NUM_STEP (3I FORMAT)

1. FILE CONTAINING LATEST PARTICLE PARAMETERS (A FORMAT)

For REFINEMENT mode operation, supply the name of the file that contains the latest particle parameters. Otherwise, leave a BLANK space for this entry to signify NON-REFINEMENT mode (i.e. when performing a whole or half-asymmetric unit search). In refinement mode, the current values of the particle origin (x,y) are used and the origin search with respect to a circular average is skipped.

2. INPUT FILENAME FOR PACKED RAW DATA IMAGES (A FORMAT)

This specifies the file that contains the raw image data.

3. INPUT FILENAME FOR REFERENCE PFTS (A FORMAT)

This specifies the file containing the series of reference PFTs produced by EMICOPFT from a 3D model or 3D reconstruction.

4. INPUT FILENAME FOR REFERENCE PROJECTIONS (A FORMAT)

This specifies the file that contains the series of reference projections produced by EMICOPFT from a 3D model or reconstruction.

5. JCUT,IRAD1,IRAD2,RES_MIN,RES_MAX,ILIST,SIG (3I,2F,I,F)

JCUT specifies the minimum rotational Bessel order (Jn) to include in the calculations. The DEFAULT (= 1) will omit the Jo term (this is recommended for the reasons given above). To include Jo, set JCUT to a NEGATIVE value.

IRAD1,IRAD2 specify the minimum and maximum radii (annuli) to be included in the calculations. The best way to determine optimum values for these parameters is to run EMICOPFTCC with the initial DEFAULT values (0,NANNULI-1) and ILIST = 1, and check the output file EMICOPFTCC.RADS to see what the correlation coefficients are like as a function of radius. You will typically see a large drop in the AVERAGE correlation coefficient near the outer edge of the particle (IRAD2) and you may also see generally lower correlations at low radii (IRAD1). Thus, IRAD1 and IRAD2 may need some fine tuning to optimize the orientations found by the program.

RES_MIN,RES_MAX specify the lower and upper resolution limits (in TPU), respectively, of the data to be included in the calculations. These values are roughly the same as those used in EMICOGRAD. Specifying a value of RES_MAX greater than the resolution of the reference data makes no sense and may generate meaningless correlation coefficient values. The DEFAULTS for RES_MIN and RES_MAX are 0.0 and SQRT(2*IDIM**2), where IDIM is the dimension of the Fourier transform. This value for RES_MAX is an ABSOLUTE UPPER LIMIT, and thus is an unrealistic value for real (i.e. noisy) data. Thus, your judgement is MOST IMPORTANT in the selection of this parameter.

ILIST is a switch used to generate various additional forms of output.

<CR> = EMICOPFTCC.DAT output only.

1 = EMICOPFTCC.RADS and EMICOPFTCC.RES also output.
2 = same as above plus asymmetric unit plots of correlation coefficients for each image output to terminal (*.LOG file).

SIG enables the use of a masking function that filters the PFT data on the basis of the variance of the PFT data. PLEASE BEWARE: This option currently DOES NOT WORK CORRECTLY!!! Don't use it!!! SIG specifies the threshold for the variance mask. Set SIG = 0.0 (DEFAULT), to disable the masking option.

6. IMG_NUM1,IMG_NUM2,NUM_STEP (3I FORMAT)

IMG_NUM1 and IMG_NUM2 specify the range of images to test. NUM_STEP (DEFAULT=1) can be used to skip images (e.g. NUM_STEP = 2 will skip over every other image). You may keep entering new values for IMG_NUM1,IMG_NUM2, and NUM_STEP. When IMG_NUM1 is set = 0, or when no input is provided for this variable, the program ends.

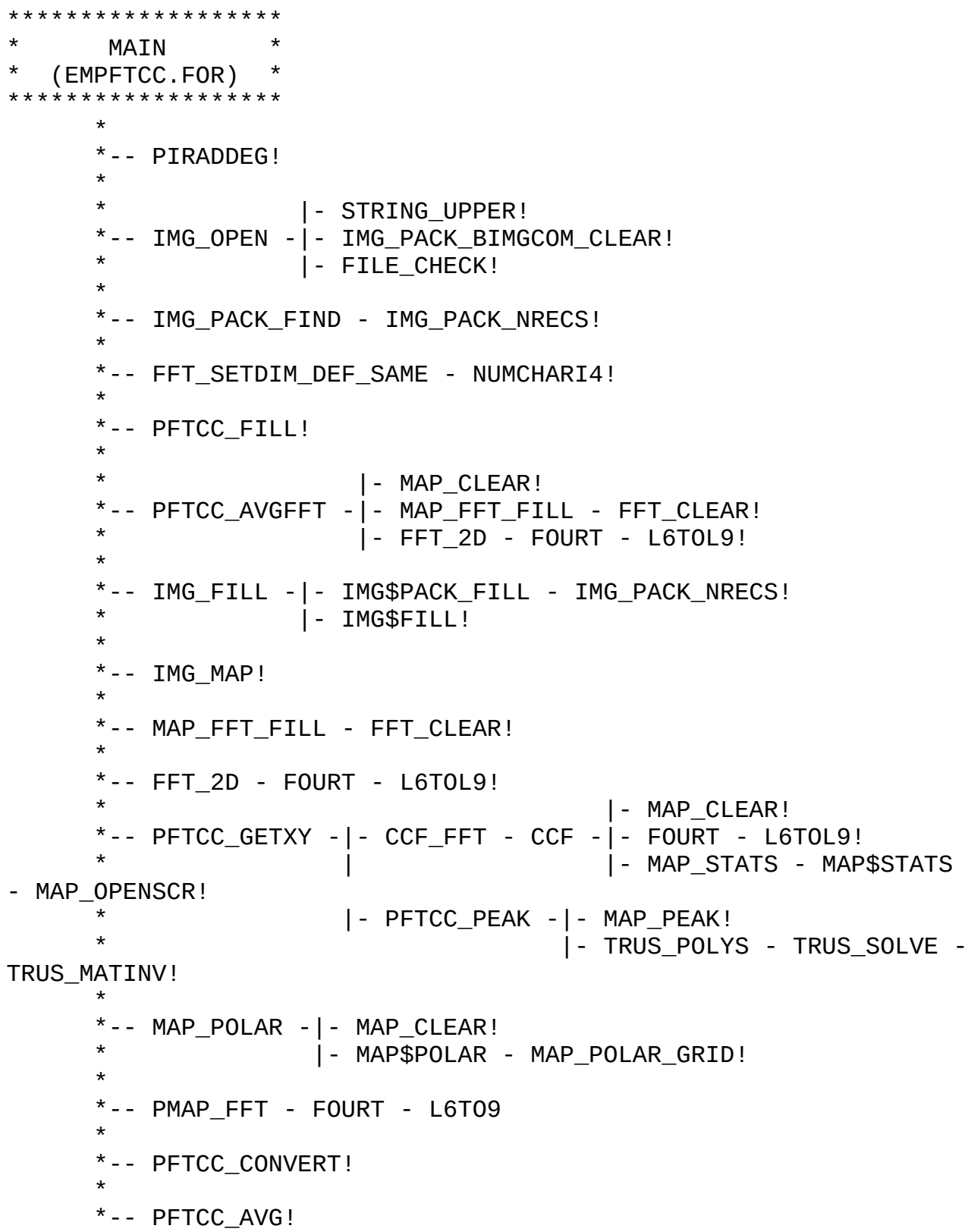
C. EXAMPLE EMPFTCC BATCH JOB

```
$ SET DEF JUSTEM$DKA0:[TSB.SCRATCH]
$ RUN JUSTEM$DKA0:[TSB.EXE]EMPFTCC
BERNAL2:[TSB.V.SV40]SV40_TILT0.DAT
BERNAL2:[TSB.V.SV40]SV40_TILT0.PCK
BERNAL2:[TSB.V.SV40]SV40_TILT0.PFTS
BERNAL2:[TSB.V.SV40]SV40_TILT0.PRJS
1, 21, 34, 6.4, 25.5, 1, 0.0
1, 96, 1
$ EXIT
```

D. FINAL NOTES FOR PURDUE USERS

1. The Alpha version of this program runs fastest.

E. FLOW CHART FOR EMPFTCC PROGRAM



```
*
*-- PFTCC_GETTP!
*
*-- PFTCC_GETPRJ!
*
*-- PFTCC_GETPO!
*
*-- MAP_YFLIP!
*
*             |- MAP$ROT - LIB$MOVC3!
*             |- MAP_ROTX!
*-- MAP_ROT -|- MAP_ROTX_SLOW -|- LIB$MOVC3!
*             |- MAP_ROTY!       |- MAP$ROT - LIB$MOVC3!
*             |- MAP_ROTY_SLOW -|- LIB$MOVC3!
*                                |- MAP$ROT - LIB$MOVC3!
*
*-- FFT_2D - FOURT - L6TOL9!
*
*-- PFTCC_RES!
*
*-- FFT_FLT -|- FFT_HIP!
*             |- FFT_LOP!
*
*-- FFT_2DBT -|- COPY_R4!
*              |- FOURT - L6TOL9!
*-- PFTCC_RAD!
```