Wfmm User's Guide

This user guide documents the basics of how to use the code provided for implementing the Bayesian wavelet-based functional mixed models methodology introduced in Morris and Carroll (2006). The code implements the Markov Chain Monte Carlo (MCMC) procedure described in Section 5 of the paper and outputs posterior samples for many model quantities.

Sample program call (from DOS window):
wfmm *input*.mat *output*.mat > *log_file.log*

The wfmm executable takes two arguments: an input filename or full filepath (represented above by *input*.mat)and and output filename (*output*.mat). Log information is written to standard out, which can be redirected to a log file.

Input

The input file ("*input*.mat" in the call) is in Matlab file format and can specify the raw data matrix **Y** and structures **model**, **basis_specs**, **MCMCspecs**, and **PostProcessSpecs** that specify the model and parameters controlling the processing.

Array dimensions of input and output variables are determined by the following parameters:

N - Number of data curves (rows in Y).
T - Number of samples of each data curve (columns of Y).
K - Number of wavelet coefficients for each curve (columns of D, the dwt of Y).
*K** - Number of wavelet coefficients retained after wavelet compression (K if no compression).
c - Number of error parameters in S.
J – Number of wavelet levels+1, which is number of wavelet groups retained after dwt.
m - Number of random effects in model.
H - Number of levels of random effects.
p - Number of fixed effects in model.
q - Number of quantiles specified for computation.
l - Number of contrasts to compute in postprocessing.
f - Number of peffects to compute in postprocessing.

Y: *N*-by-*T* matrix, each row containing one of the observed functions on an equally-spaced grid of length *T*. This is the only variable that is required. Defaults will be taken for everything else if they are omitted.
For 2d images, each row is a column-stacked image, i.e. if the image is a matrix A with dims(N1, N2), then each column from j = 0 to N2-1 will be copied successively into a single row of Y.

```
ind = 0;
for(j = 1:N2)
     for(i = 1:N1)
           ind = ind + 1;
           vY(ind) = V(i, j);
```

```
        end
    end
```

For 3d volumes represented by a 3d array V(i, j, k) with dims(N1, N2, N3), i varies most often followed by j, then k.

```
ind = 0;
for(k = 1:N3)
    for(j = 1:N2)
        for(i = 1:N1)
            ind = ind + 1;
            vY(ind) = V(i, j, k);
        end
    end
end
```

The opposite algorithm is used to reconstruct images and volumes from beta and U in the output.

model: Matlab structure indicating details of the model. The default value is taken if the field is left out. The following elements can be specified:

Field name	Type	Description	Default
X	*N-by-p* matrix	Desired covariates for the p fixed effect functions in the model.	Nx1 column of 1's
Z	*N-by-m* matrix	Contains covariates for each set of random effect functions. If omitted, it is assumed that it is a fixed effects model. Z can also be stored with all the levels concatenated together columnwise.	
C	*N-by-c* matrix	Columns indicate the functions that share a common residual error S.	*1 col of 1's*
m	Vector of length H	If more than one level of random effects is desired, this vector specifies the number of random effects at each level. Sum of elements must equal m, number of columns of Z.	

Notes on model

The previous method to specify multiple levels of random effects is still supported. In this format, Z is specified as a cell array of length H where each cell is a Z matrix containing just the covariates for that level. When concatenated together columnwise, they are equivalent to the Z matrix defined above. If Z is defined this way, then m vector is not used.

basis_specs: This Matlab structure, formerly called wavespecs, has been expanded to include other transforms as well as the discrete wavelet transform. The wavespecs name is still recognized and will be process as before, but is deprecated. The following elements can be specified (The default value is taken if the field is left out) :

Field name	Type	Description	Default
transformtype	string	Selects transform to be applied to data. "none" – In this case no transform is performed and the input data matrix is treated like the coefficient matrix	"wavelet"

		"wavelet" – dwt as in previous versions "PC" – Principal Components "custom" – User supplies matrix transform "PCw"- Performs PC followed by wavelet transform on residual. PC and wavelet coefficients are concatenated to give D. "wPC" – Applies wavelet transform followed by PC on wavelet coefficients	
The following fields apply to the wavelet transform			
wavelet	string	Wavelet basis to use (see below for options)	"db4"
nlevels	+integer	Number of levels of decomposition.	Optimal number
boundary	string	Boundary correction method used.	"periodic"
extended_mode	*0-or-1*	Whether (1) or not (0) to keep extra boundary wavelet coefficients.	1
alphawav	+double	Fraction of "energy" retained during wavelet compression (1 being no compression).	1.0
t	*0-or-*+ integer	Number of functions that must be greater than threshold P in order to be retained during compression.	0
dims	Vector of length Ndims	Dimensions for 2d or 3d data sets. Length gives number of dimensions.	1d
highpass	*0-or-*+ integer	Number of most detailed levels discarded.	0
lowpass	*0-or-*+ integer	Number of least detailed levels discarded.	0
The following fields apply to the 2d rectangular wavelet transform			
rectangular	*0-or-1*	Selects square (0) or rectangular(1) 2D dwt transform.	0
wavelet2	string	Wavelet basis to use for dimension 2	"db4"
nlevels2	+ integers	Number of levels of decomposition for dim 2	Optimal number
boundary2	string	Boundary correction method used for dim 2	"periodic"
The following fields apply to the Principal Components transform			
alphaPC	0 <= alphaPC <= 1	Fraction of energy retained in PC compression (1 being no compression).	1.0
PCpartitionbase		Base for automatic partition of PC coefficients.	10.
The following fields control partitioning for Principal Components, "none", and "custom" transform types.			
partitions	Int Vector	Each element contains number of wavelet coefficients in a partition.	empty
npartitions	+ integer	Number of equal partitions	0

The following fields store matrices for "custom" transform; also store matrix transforms for "PC"			
phi_inv	*T-by-K* matrix	Matrix for transform D = Y*phi_inv	empty
phi	*K-by-T* matrix	Matrix for inverse transform Y = D*phi. Software attempts to compute inverse of phi_inv if this is left blank.	empty
The following fields are written out in wavespecs by dwt processing. They do not need to be specified on input and should not be modified except for Kj when wavelet = "none".			
Kj	Vector of length *J*	Number of wavelet coefficients in the approximation and n levels of details.	
T	+integer	Number of data points in curve	
J	+integer	Number of entries in Kj; wavelet levels +1	
K	+integer	Number of wavelet coefficients	
Kstar	+integer	Number of wavelet coefficients retained after compression	
DIndex	Vector of variable length	Index of each retained wavelet coefficient in original uncompressed array.	

Notes on basis_specs

Wavelet Bases: The current version accepts both columns of abbreviations for the wavelet bases specified below. Here are the available wavelets, and the corresponding notations for wfmm and Matlab:

```
WFMM wavelets     Matlab equivalent
    "haar",       "db1" or "haar"
    "d4",         "db2"
    "d6",         "db3"
    "d8",         "db4"
    "d10",        "db5"
    "d12",        "d6"
    "d14",        "db7"
    "d16",        "db8"
    "d18",        "db9"
    "d20",        "db10"
    "s4",         "sym2"
    "s6",         "sym3"
    "s8",         "sym4"
    "s10",        "sym5"
    "s12",        "sym6"
    "s14",        "sym7"
    "s16",        "sym8"
    "c6",         "coif1"
    "c12",        "coif2"
    "c18",        "coif3"
    "c24",        "coif4"
    "c30"         "coif5"
```

Notes on **partitions**:

Partitions are defined as contiguous subgroups of coefficients that are assigned a Pi and Tau parameter for smoothing. Partitioning is taken care of automatically for wavelets. Each level has its

own Pi and Tau parameter. For other types of transforms, partitions can be specified manually by supplying a partitions vector. Each element of the vector specifies the number of contiguous elements in the partition from 0 to Kstar – 1 and must sum to Kstar. Software will check this and throw an exception if there is an error.
If partitions vector is not supplied, it checks for an npartitions argument; if present, it splits the coefficients into npartitions number of equal partitions (remainder goes to the last partition). If npartitions is not supplied, it is assumed to be 1. For principal components, an automatic method is used if none of the manual methods are prescribed. This algorithm examines the logbase(lambda) of the retained eigenvalues, which are sorted in descending order and groups them into bins of 0.5. If a bin doesn't have more than 1 element, it is combined with the following bin (except for the last bin, which would be combined with the previous bin. The base for the logarithm is specified by the PCpartitionbase field and is defaulted to 10. Decreasing the base value has the effect of creating more bins.

.

Boundary: The following strings are recognized for the boundary field:

Boundary Condition Type	Description
"zero"	Boundaries are zero-padded
"periodic"	
"reflection"	Boundary values are reflected around the end of data.
"interval"	

MCMCspecs: Matlab structure describing details of MCMC. The default value is taken if the field is left out. The following scalar elements can be specified:

Field name	Description	Default
B	Number of MCMC samples to obtain.	1000
burnin	Burn-in length; number of initial samples to discard.	1000
thin	Thinning parameter; e.g. if 10, then keep every 10 samples in MCMC.	5
propvar_omega	Multiple of var(MLE) to use in proposal variance for variance components in step 2 of the MCMC.	1.5
nj_nosmooth	Number of lowest frequency wavelet levels for which we want a vague prior (no smoothing).	2
The following parameters are simply for numerical stability:		
minp	Minimum value for any π_{ij}.	10^{-14}
minT	Minimum value for T_{ij}.	10
bigT	Value to use for T_{ij} when vague prior desired (no smoothing).	1000
maxO	Maximum odds ratio (prevents overflow).	10^{20}
minVC	Minimum value of variance component (prevents instability of variance components wandering near zero).	10^{-6}
VC0_thresh	Minimum size for important variance component.	10^{-6}

delta_omega	Multiple for prior on omega: "number of datasets of information" in prior (see discussion in Morris, et al. (2003) JASA, 98:591-597).	10^{-4}
omega_MOM_maxiter	Maximum number of iterations in finding MOM starting values for variance components.	100
omega_MOM_convcrit	Convergence criteria for iterative procedure for finding MOM starting values for variance components.	10^{-3}
time_update	Number of iterations between updates to the log file during MCMC loop.	100.
missing_data	Flag indicating whether to process normal data Y or imputed data Vstar (0=normal, 1=imputed).	0.
update_pi_tau	Flag indicating whether to allow MCMC updating of pi and tau variables Vstar (0=no updating, fix with empirical Bayes estimates, 1=do MCMC updating).	0.
pi_prior_var	Prior variance for pi when updating pi and tau.	0.06
tau_prior_var	Prior variance for tau when updating pi and tau.	1000

Notes on **nj_nosmooth:**
All transformtypes except "PCw" use a scalar value for nj_nosmooth. In this one exception, two values for the PC coefficients and wavelet coefficients may be required. In this case, nj_nosmooth is specified as a vector with the first element applied to the PC components and the second element applied to the wavelet components. If only one element is specified, it is applied to the wavelet portion and all PC coefficients are smoothed.

PostProcessSpecs: Matlab structure controlling postprocessing. The default value is taken if the field is left out. The following scalar elements can be specified:

Field name	**Type**	**Description**	**Default**
L	*l-by-p* matrix	Specifies linear combinations of effects.	None
quantiles	vector of length *q*	Specifies quantiles either as whole numbers (0 to 1) or percent (0 to 100). Values less than 50% also specify alpha for Lbeta_upperCI and Lbeta_lowerCI.	.5,1,2.5,5 and complement
effect_size	vector of length *f*	Specifies effect sizes for computation beta_peffects = Pr(|theta|>effect_size).	0.3219, 0.585, 0.8074, 1
keep_beta_samples	*0-or-1*	Flag that specifies whether to output beta samples to *Input* _beta.dat. 0 = no output, 1 = output).	1
compute_U	*0-or-1*	Flag that specifies whether to compute random effects. 0 = do not compute, 1 = compute).	0

keep_U_samples	*0-or-1*	Flag that specifies whether to output U samples to *Input* _U.dat. (0 = no output, 1 = output).	0
LT	*T-by-g* matrix		

Notes on LT

LT allows users to specify an option that allows the user to request inference for linear combinations of the t's. This is done by simply matrix multiplying by the T x g matrix LT, e.g. for the posteriors of p x T matrix beta, beta*LT gives a p x g matrix on which additional statistical summaries are computed. The LT transform allows the user to specify g regions of interest, one with each column of LT.

Output

The output of the program is a Matlab data file ("*output*.mat" in the sample call), containing the following Matlab objects, as well as an *input*_Init.mat file containing results of the initialization phase of the computation. Error messages and status are written to standard output, which can be redirected to a log file. Processing status can be monitored by periodically typing the log file

The following variables are stored in the *input*_Init.mat file:

Variable name	**Type**	**Description**
model wavespecs MCMCspecs		Copies of input structures.
D	*N-by-K** matrix	Wavelet coefficients for observed data.
pi_MLE	*p-by-J* matrix	π_{ij} estimated by the Empirical Bayes procedure described in Section 4.4 of Morris and Carroll (2006), based on theta_MLE.
pi_MOM	*p-by-J* matrix	π_{ij} estimated by the Empirical Bayes procedure based on theta_MOM.
tau	*p-by-J* matrix	T_{ij} estimated by the Empirical Bayes procedure described in Section 4.4 of Morris and Carroll (2006), based on theta_MLE.
tau_MOM	*p-by-J* matrix	T_{ij} estimated by the Empirical Bayes procedure, based on theta_MOM.
omega_MOM	*(H+c)-by-K**	Method of moments starting values for the wavelet-space variance components q_{jk} and s_{jk} in model (3).
omega_MLE	*(H+c)-by-Kstar*	Profile maximum likelihood starting values for the wavelet-space variance components.
se_omega	*(H+c)-by-K**	Estimate of the variance of omega_MLE, to use in automatic proposal variances in Metropolis-Hastings procedure described in step (b) of Section 5 in Morris and Carroll (2006).
betastar_ns	*p-by-K** matrix	Non-shrunken estimate of wavelet coefficients for fixed effects conditioning on starting values of variance components, given by

		equation (5) in Morris and Carroll (2006).
Vbetastar_ns	*p-by-K** matrix	Variance of these wavelet-spaced estimates, given by equation (6) in Morris and Carroll (2006).
alpha_MLE	*p-by-J* matrix	Matrix containing starting values for shrinkages for wavelet coefficients for fixed effect functions, which are their posterior probabilities of being "nonzero". Condition on omega_MLE for variance components.
alpha_MOM	*p-by-J* matrix	same as alpha, only based on omega_MOM.
prior_omega_a, prior_omega_b	*(H+c)-by-K**	matrices containing the prior hyperparameters for the inverse gamma distributions on the wavelet-space variance components
Wv	structure	Structure containing, for each wavelet coefficient, the following statistics, using starting values of the variance components for Σ_{jk} • $XvX=X'(\Sigma_{jk})^{-1}X$ • $XvZ=X'(\Sigma_{jk})^{-1}Z$ • $XvD=X'(\Sigma_{jk})^{-1}D$ • $ZvZ=Z'(\Sigma_{jk})^{-1}Z$ • $ZvD=Z'(\Sigma_{jk})^{-1}D$ • $dvd=diag(D'(\Sigma_{jk})^{-1}D)$ • $L1=det(\Sigma_{jk})$ • $L2= (d_{jk}-X B_{jk})' (\Sigma_{jk})^{-1} (d_{jk}-X B_{jk})$ where Σ_{jk} is the marginal variance of d_{jk}

.

The following variables are stored in *output*.mat

Variable name	**Type**	**Description**
model wavespecs MCMCspecs	structures	Input structures are included to store parameters used to generate these values.
D	*N-by-K** matrix	Wavelet coefficients for observed data.
betastar_ns	*p-by-K* matrix	Non-shrunken estimate of wavelet coefficients for fixed effects conditioning on starting values of variance components, given by equation (5) in Morris and Carroll (2006).
betastar_mean	*p-by-K** matrix	betans*alpha; shrinkage starting values for betas.
Lbetans_mean	*p-by-T* matrix	Inverse discrete wavelet transform of betastar_ns. Unsmoothed beta.
Lbeta_mean	*p-by-T* matrix	Posterior mean for each fixed effect function.
Lbeta_quantiles	*q-by-p*T* matrix	Pointwise quantiles specified by quantiles in PostProcessSpecs structure for each fixed effect function.
Lbeta_sd	*p-by-T* matrix	Pointwise quantiles for each fixed effect function.
Lbeta_peffects	*f-by-p*T* matrix	p effects of beta samples.
Lbeta_p0	*p-by-T* matrix	2*min [Prob{ Lbeta(t)>0 }, Prob{ Lbeta(t)<0 }]. In cases with vague priors approximates the frequentist p-values.

Lbeta_simbas	*p-by-T* matrix	Simultaneous band scores for beta
Lbeta_upperCI	*Alpha-by-p*T matrix*	Simultaneous credible interval upper bound
Lbeta_lowerCI	*Alpha-by-p*T matrix*	Simultaneous credible interval lower bound
omega_mean	*(m+c)-by-Kstar* matrix	Mean of MCMC omega samples.
omega_quantiles	q-*by-(H+c)*Kstar* matrix	Pointwise quantiles specified by quantiles in PostProcessSpecs structure for each random effects level and standard error c.
omega_sd	*(H+c)-by-Kstar* matrix	*S*tandard deviation of MCMC omega samples.
If MCMCspecs.update_pi_tau is 1, the following outputs are available:		
pi_mean	*p-by-J* matrix	Mean of the posterior samples of Pi.
pi_quantiles	*p-by-J* matrix	Quantiles of the posterior samples of Pi.
pi_sd	*p-by-J* matrix	**S**tandard deviations of the posterior samples of Pi.
tau_mean	*p-by-J* matrix	Mean of the posterior samples of Tau.
tau_quantiles	*p-by-J* matrix	Quantiles of the posterior samples of Tau.
tau_sd		Standard deviations of the posterior samples of Tau.
If PostProcessSpecs. compute_U is 1, the following outputs are available:		
U_mean	*m-by-T* matrix	Posterior mean for each random effect function.
U_ns	*m-by-T* matrix	Nonsmoothed curve for each random effect function.
U_quantiles	*q-by-m*T* matrix	**P**ointwise quantiles specified by quantiles in PostProcessSpecs struct for each random effect.
U_sd	*m-by-T* matrix	Standard deviation of MCMC random effect samples.
If functions are 1d and T < 1500, the following outputs are available:		
rho	Cell vector of length *(H+c)*	Data-space correlation *T-by-T* matrices corresponding to diagonal wavelet-space matrix formed from omega_mean.
sigma	Cell vector of length *(H+c)*	Data-space variance functions (vector of length T) corresponding to diagonal wavelet-space matrix formed from omega_mean.

Notes on output file:

***quantiles:**

Lbeta_quantiles is specified to be stored as a q x p*T matrix where p*T length row is stored with T varying most often (*p-by-T* matrix stored "row-wise"). The same format also applies to quantiles for U, omega, pi, and tau. beta_peffects is specified to be stored as an *f-by-p*T* matrix. Again each row can be thought of as a *p-by-T* matrix stored row-wise.

LBetaLt_* outputs:

If PostProcessSpecs.LT is specified, the Lbeta_*statistic* outputs have a corresponding LbetaLt_*statistic* output that gives the summaries statistics for Lbeta*LT outputs.

Lbeta_simbas, Lbeta_upperCI, Lbeta_lowerCI:
Simultaneous credible intervals are computed from max(Zscore) over the T samples of each fixed effect sample, yielding n max_Zscores samples. The quantile of maxZscore corresponding to a specified alpha is then combined with point-wise mean and standard deviation values to give the simultaneous credible interval (Lbeta_upperCI and Lbeta_lowerCI for each fixed effect.
Lbeta_simbas is defined as minimum significance level alpha at which the simultaneous credible band excludes zero.
PostProcessSpecs.keep_beta_samples must be set to 1 in order to compute these outputs since this requires the beta samples to be read in from the *_beta.dat file.

MCMC samples are output in binary double precision format, one file for each variable with filename *Input _variablename*.dat**:**

Filename	**Description**
Input _wbeta.dat	File containing MCMC posterior samples for wavelet coefficients for fixed effects. *Kstar* samples for each fixed effect are stored together and all fixed effect data blocks for one iteration of MCMC are stored together.
Input _beta.dat	File containing MCMC posterior samples for data-space fixed effect functions. *T* samples for each fixed effect are stored together and all fixed effect data blocks for one iteration of MCMC are stored together.
Input _omega.dat	File containing MCMC posterior samples for variance components in wavelet space. *Kstar* samples for each error parameter or random effects level are stored together and all error parameter or random effects level data blocks for one iteration of MCMC are stored together.
Input _newtheta.dat	File containing Metropolis-Hastings acceptance probabilities for the set of variance components for each wavelet coefficient. *Kstar* samples for each error parameter or random effects level are stored together and all error parameter or random effects level data blocks for one iteration of MCMC are stored together.
*Input*_U.dat	File containing MCMC posterior samples for random effects. *T* samples for each random effect are stored together and all random effect data blocks for one iteration of MCMC are stored together.

Comments:

- The current interface assumes you create the input files and want to post-process the output files in Matlab.
- The current version of the code assumes:
    1. By default you want to estimate the shrinkage hyperparameters using the empirical Bayes method. These can be estimated as part of the MCMC by setting MCMCspecs.update_pi_tau to 1.
    2. You want vague proper priors for the variance components, centered at the starting values with information equivalent to delta_omega observations.
    3. The random effect functions are independent and identically distributed, so P=R=I

- This code yields MCMC samples for the quantities in the wavelet-space model, (3) in Morris and Carroll (2006), plus MCMC samples for the fixed effect functions B in the data space model (2).
- MCMC samples of Q_h and S_i matrices can be obtained by applying the 2-D IDWT to the corresponding diagonal wavelet-space matrices. They are generated only for T < 1000, since their large size will cause memory issues in large data sets.
- For large data sets, we recommend using the 64-bit executable. Approximate RAM and disk usage are given by the formulas below.

Estimating Disk and RAM Usage

N = Number of Functions
p = Number of fixed effect functions
T = Number of observations/function
B = Number of MCMC samples
K = Number of wavelet coefficients
K' = Number of non-thresholded wavelet coefficients
m = Number of random effect functions H = Number of levels of random effect functions
c = Number of strata for residual error functions

Disk Usage ≈ 8 [BK'(p+H+c+1) + pT(B+4) + 3NT+ 2NK' + K'(p^2+m^2+pm+m+3+7p+7(H+c))]

RAM Usage ≈ 8[2pT+(H+c+p+2)K'+T + (0.05B+4)pT +6T+ (4(H+c)+2p)K']

Parallel Processing

The processing has also been divided into three executables for initialization (wfmm1), MCMC loop (wfmm2), and postprocessing (wfmm3). This allows multiple MCMC chains to be run simultaneously using a grid computing resource like Condor, and have their results combined in the postprocessing step. Their command line arguments are:

wfmm1 *input*.mat > *log_file.log*

This takes the same *input*.mat as input and outputs a *input*_Init.mat file as described above.

wfmm2 *input*_Init.mat *output*

Takes the *input*_Init.mat file as input and outputs MCMC samples as *output_variablename*.dat binary files

wfmm3 *input*_Init.mat *output output_summary*.mat number_of_files

The following is an example of a parallel processing bat file for Condor using these three executables. It relies only on a command to submit jobs to the grid (condor_submit), and a command to wait until all of the submitted jobs have run (condor_wait):

```
wfmm1 %1.mat > %1_init.log
condor_submit -a Dataset=%1 -a ThreadNumber=%2 wfmm_condor.sub
condor_wait %1.log
wfmm3 %1_Init.mat %1_results %1_summary.mat %2 > %1_summary.log
```

%1 (first argument of the bat file) is filename of the input mat file, %2 (second argument of the bat file) is number of parallel jobs requested for the MCMC computation.

The condor_submit command also requires a submit file that describes the jobs. The filename and number of jobs parameters are passed to the condor submit file as parameters Dataset and ThreadNumber. An example file is shown below.

```
# A basic submit file

# On Windows the universe is vanilla
universe = vanilla

# Set the executable name here
executable = WFMM2.exe

# Set command line arguments here
arguments = $(Dataset)_Init.mat $(Dataset)_results_$(Process).mat

# Set requirements here (memory, OS, etc.)
requirements = (OpSys == "WINNT40" || OpSys == "WINNT50" || OpSys == "WINNT51")
   && (memory > 1000)

# List the input files here
transfer_input_files = $(Dataset)_Init.mat, Z:\bin\icudt24l.dll,
   Z:\bin\icuin24.dll, Z:\bin\icuio24.dll, Z:\bin\icuuc24.dll,
   Z:\bin\libmat.dll, Z:\bin\libmx.dll, Z:\bin\libut.dll, Z:\bin\libz.dll,
   Z:\bin\msvcp71.dll, Z:\bin\msvcr71.dll, Z:\bin\libguide40.dll

# Leave this alone
transfer_files = ALWAYS

# You can rename these files, but be sure they're defined
# These may be useful for debugging purposes
output = $(Dataset)_$(Process).txt
error = $(Dataset).err
log = $(Dataset).log

# Set the number of copies to submit here
queue $(ThreadNumber)
```

These files should be adaptable to any grid computing system.

References

Morris, JS and Carroll, RJ (2006, Wavelet-based functional mixed models, *Journal of the Royal Statistical Society, Series B,* 68(2): 179-199.

Morris JS, Arroyo C, Coull B, Ryan LM, Herrick R, and Gortmaker SL (2006), Using Wavelet-Based Functional Mixed Models to characterize Population Heterogeneity in Accelerometer Profiles: A Case Study. *Journal of the American Statistical Association*, 101(476): 1352-1364.

Morris, JS, Brown PJ, Herrick, RC, Baggerly KA, and Coombes, KR (2008), Bayesian Analysis of Mass Spectrometry Proteomic Data using Wavelet Based Functional Mixed Models, *Biometrics*, 64(2): 479-489.

Morris , JS, Baladandayuthapani, V, Herrick, RC, Sanna, P, and Gutstein, H (2011, Automated Analysis of Quantitative Image Data Using Isomorphic Functional Mixed Models, with Applications to Proteomics Data, *The Annals of Applied Statistics*, 5(2A), 894-923.