

## Instructions for running efficient simulations on the FASRC cluster using job arrays

### Step 1: Constructing the files

Create the following files:

- ❖ `sim_data.m`: Simulates one dataset.
  - Format is a Matlab function file:

```
function sim_data(n_sim)
    ...
end
```
  - Inputs:
    - `n_sim`: simulation iteration index, taken from the command line and bash script `sim_batchsubmit.sh`
  - Outputs: creates and saves one simulated dataset to a specified output directory.
  - Notes: within the function, there should be an `out_dir` variable that can be changed if a different output directory is desired.
  - For the wsOFMM simulations, an example of this type of file is `sim_equal_subpop.m`.
- ❖ `run_model.m`: Runs the model on a dataset.
  - Format is a Matlab function file

```
function sim_data(n_sim)
    ...
end
```
  - Inputs:
    - `scenario`: scenario index, taken from bash script `matlab_batchsubmit.sh`
    - `n_sim`: simulation iteration index, taken from the command line and bash script `matlab_batchsubmit.sh`
  - Outputs: runs the model on a dataset read in from an input directory, then saves the MCMC output and analysis results to a specified output directory
  - Notes: within the function, there should be an `in_dir` variable specifying input directory, and an `out_dir` variable specifying output directory.
  - For the wsOFMM model, an example of this type of file is `wsOFMM_main.m`
- ❖ `sim_batchsubmit.sh`: Bash script for simulation job array. This allows us to run `sim_data.m` many times in parallel, resulting in huge time savings when creating simulated dataset iterations.
  - Contains a call to the `sim_data(n_sim)` function, but uses the array ID, named `SLURM_ARRAY_TASK_ID`, to index the simulation iteration. The bash script includes the following:

```
#!/bin/bash
#SBATCH -J job_name           # Job name for the array
#SBATCH -o job_name_%A.out    # Save output to same file with job ID
#SBATCH -p shared             # Partition to submit to
#SBATCH -n 1                  # Number of cores
#SBATCH --mem 2000             # Memory request
#SBATCH -t 0-15:00:00         # Runtime (D-HH:MM:SS) days, hours, mins, secs
#SBATCH --mail-type=BEGIN,END,FAIL # Mail notifications
#SBATCH --mail-user=stephaniewu@fas.harvard.edu # Account to email
module load matlab
matlab -nodisplay -nosplash -r "sim_data(${SLURM_ARRAY_TASK_ID})"
```

- To submit a job array that will simulate 100 datasets, run:
 

```
sbatch --array=1-100 sim_batchsubmit.sh
```
  - For the wsOFMM model, an example of this type of file is `sim_batcharray.sh`
- ❖ `matlab_batchsubmit.sh`: Bash script for model job array. This allows us to run `run_model.m` over many iterations in parallel, as well as for various scenarios.
- Contains a call to the `run_model(scenario, n_sim)` function, but uses the array ID, named `SLURM_ARRAY_TASK_ID`, to index the simulation iteration. The bash script includes the following for running scenarios 1 through 4:
- ```
#!/bin/bash
#SBATCH -J job_name           # Job name for the array
#SBATCH -o job_name_%A.out    # Save output to same file with job ID
#SBATCH -p shared             # Partition to submit to
#SBATCH -n 1                  # Number of cores
#SBATCH --mem 2000            # Memory request
#SBATCH -t 0-15:00:00         # Runtime (D-HH:MM:SS) days, hours, mins, secs
#SBATCH --mail-type=BEGIN,END,FAIL # Mail notifications
#SBATCH --mail-user=stephaniewu@fas.harvard.edu # Account to email
module load matlab
for scenario in $(seq 1 4); do
    echo This is scenario ${scenario} iteration ${SLURM_ARRAY_TASK_ID}.${SLURM_ARRAY_TASK_ID}
    matlab -nodisplay -nosplash -r "wsOFMM_main(${scenario},${SLURM_ARRAY_TASK_ID})"
done
```
- To submit a job array that will simulate 100 datasets from scenarios 1 through 4, run:
 

```
sbatch --array=1-100 matlab_batchsubmit.sh
```
  - For the wsOFMM model, an example of this type of file is `matlab_batchsubmit.sh`

## Step 2: Running the files

1. Make sure the following files are in the same folder:
  - a. `sim_data.m`
  - b. `run_model.m`
  - c. `sim_batchsubmit.sh`
  - d. `matlab_batchsubmit.sh`
  - e. Any helper function files, such as `get_data_vars.m`
2. Make sure the `out_dir` variable in the `sim_data.m` file specifies the desired output location for the simulated datasets.
3. Submit simulation job array to create 100 simulated datasets:
 

```
sbatch --array=1-100 sim_batchsubmit.sh
```
4. Make sure the `in_dir` variable in the `run_model.m` file matches the `out_dir` variable in the `sim_data.m` file, and make sure the `out_dir` variable in the `run_model.m` file specifies the desired output location for the model results.
5. Submit a job array that will simulate 100 datasets from scenarios 1 through 4:
 

```
sbatch --array=1-100 matlab_batchsubmit.sh
```