Code, Data and Instructions for Reproducing Results

A Spatio-Temporal Modeling Framework for Surveillance Data of Multiple Infectious Pathogens with Small Laboratory Validation Sets

Contents

1 General Information

This package contains data and code for producing the results in the manuscript "A Spatio-Temporal Modeling Framework for Surveillance Data of Multiple Infectious Pathogens with Small Laboratory Validation Sets". It can be freely downloaded from `https://drive.google.com/open?id=19FG7fg_mKx5f3vOjAFN5SowivxoAaAMR`. Our C code was compiled by gcc 5.2.0 and R code was run in R 3.3.3. The required R packages (versions) are: `caTools` (1.17.1), `maptools` (0.9-2), `mvtnorm` (1.0-6), `mgcv` (1.8-23), `RColorBrewer` (1.1-2), `spdep` (0.7-4) and `splines` (3.3.3).

2 Directory Structure

The root unzipped directory (currently named "`code_v2`" but can be changed without affecting compiling or running programs) contains this README file and the following subdirectories:

- "`data/`" contains the hand, foot, and mouth disease (HFMD) surveillance data in five provinces (Hunan, Fujian, Guangxi, Guangdong, and Jiangxi) of China during 2009 and other related data files such as shape files for mapping. The subdirectoy "covariates" contains all the covariates used in simulation and case study.

- "`simulation/`" contains the code and summarized results for the simulation studies presented in the paper. The complete posterior samples are over gigabites, and thus only a subset of the samples and summary results are provided to produce the figures. Subdirectories and their use are explained below.

- "real_data_analysis/" contains the code and results for the case study presented in the paper. Similarly, a subset of the samples and summary results are provided to produce the figures. Subdirectories and their use are explained below.

3 Data

- RData files:
  - "2009_south5.RData" is the main data file that contains the case counts (Yv) stratified by severity in 69 prefectures and 53 weeks, the lab tested case counts (Z) stratified by severity and virus in 69 prefectures and 53 weeks, and the neighborhood structure of the 69 prefectures (nb.prefecture.south5). Yv is a three dimensional array. The three dimensions represent prefecture, severity, and week. Z is a four dimensional array. The dimensions are prefecture, severity, week, and virus. As the real surveillance data is owned by China CDC, random noises are added to the true counts. Please see Section 6 of this document for the R code making a slightly perturbed version of the real surveillance data. However, counts in Yv are within 6% of the real data. Counts in Z are the same as the real data. Researchers who need the exact real data may contact the Division of Infectious Disease of China CDC.
  - "south5_province_shape.RData" is the province-level shapefile.
  - "south5_shp.RData" is the prefecture-level shapefile.
- "data/covariates/": Files in this subdirectory contains temperature (temp), wind speed (ws), relative humidity (rh), and population density (pop_density) for the prefectures studied. File names ended with "std" indicate the variables are standardized

to have mean zero and standard deviation 1.

- Other `txt` files: These files are needed by the `C` code in subdirectories "`simulation/code/`" and "`real_data_analysis/code/`".
  - "`2009_south5_Yv_realdata.txt`": txt version of the surveillance cases counts `Yv` in "`2009_south5.RData`"
  - "`2009_south5_Z_realdata.txt`": txt version of the lab-validation counts `Z` in "`2009_south5.RData`"
  - "`2009_south5_Z_agg_realdata.txt`": `Z` aggregated by province
  - "`basis_modified.txt`": spline basis functions used for modeling temporal effects
  - "`south5_nb_eval.txt`" and "`south5_nb_evec.txt`": eigenvalues and eigenvectors of $\mathbf{D} - \mathbf{W}$ where $\mathbf{W}$ is the prefecture-level adjacency matrix and $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the row sums of $\mathbf{W}$.
  - "`south5_nb_eval_prov.txt`" and "`south5_nb_evec_prov.txt`": eigenvalues and eigenvectors of $\mathbf{D} - \mathbf{W}$ where $\mathbf{W}$ is the province-level adjacency matrix and $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the row sums of $\mathbf{W}$.
  - "`south5_nb.txt`": number of neighbors and neighbors of each prefecture.
  - "`south5_provincecode.txt`": province ID for each prefecture. 1: Hunan, 2: Guangdong, 3: Guangxi, 4: Fujian, 5: Jiangxi.
  - "`south5_region2prefecture.txt`": prefecture ID for each prefecture.

4 Simulation

4.1 Contents

- "simulation/code/": this folder stores all the code for conducting the simulation studies. The files are arranged in the order of data generation, inference, and presentation.
  - "set_epi_paras.R" sets the parameters for generating HFMD epidemics in the five provinces. This will call the multiNorm() function in the file "multinorm_lc.R" which generates a multivariate normal random vector with a linear constraint.
  - "set_lab_paras.R" sets the parameters for generating lab-validation data.
  - "data_gen.R" generates data for the scenario of one infectious week.
  - "data_gen_2infweek.R" generates data for the scenario of two infectious weeks.
  - "lab_agg.R" aggregates lab-tested cases by neighboring prefectures or by province.
  - "mcmc_template.c" and "mcmc_functions.h": MCMC code for data with one infectious week and non-aggregated lab data.
  - "mcmc_2infweek_functions.h" and "mcmc_2infweek_template.c": MCMC code for data with two infectious weeks and non-aggregated lab data.
  - "mcmc_agg_functions.h" and "mcmc_agg_template.c": MCMC code for data with one infectious week and aggregated lab data.
  - "datastructure.h" and "datastructure_agg.h" define data structures for the

MCMC inference with non-aggregated (1- or 2-week infectious period) and aggregated lab data, respectively.

- “distribution.h”, “mathfunc.h” and “matrix.h”: header files needed in C code, which provide routine statistical and mathematical function and matrix operations.
- “txt2RData.R”, “txt2RData_2infweek.R” and “txt2RData_agg.R” read the MCMC samples in txt files save them into RData files.
- “collect_bala_imba_results.R” summarizes the results for comparing balance and imbalance lab sampling designs.
- “collect_agg_results.R” summarizes the results from non-aggregated and two types of aggregated lab data.
- “collect_2infweek_results.R” summarizes the results from epidemics with one infectious week and epidemics with two infectious weeks.
- “collect_block_element_results.R” summarizes the results for comparing element-wise sampling (MBS) and block sampling (MIS).
- “collect_severe_results.R” summarizes the results from different lab test proportions for severe cases.
- “plot_figures.R” produces figures for the simulation study. Specifically, Figures 1, 2, S3-S22, and S29-S34.
- “identifiability_covariate_effects.R” conducts a small simulation study to show the identifiability issue for the covariate effects associated with the environment-to-human transmission. See Section 2 in the Web Appendix for details.

- “simulation/parameters/”: folder storing datasets of parameter values for simulating surveillance data including both epidemic data and lab-validation data.

- “simulation/sim_data/”: folder storing simulated epidemic and lab-validation data.

- “simulation/output_chains/”: folder storing all MCMC samples based on simulated data.

- “simulation/results_summary/”: folder storing summary results of MCMC samples for producing figures.

- “simulation/figures/”: folder storing figures.

- “simulation/parameters/provided/” and “simulation/results_summary/provided/”: parameter values and summary results used in our paper are given in these subdirectories. **Please do not change the datasets in these subdirestories, as they are created for reproducing the figures in the paper.**

4.2 Instructions for simulation

1. Copy the directories `data/` and `simulation/` to your working directory, or you can work directly from the unzipped root directory “`path/code_v2/`”, where “path” is the folder to which you unzipped the code and data package. Both C code and R code need to be run from this working directory.

2. Generate data

   (a) To set the parameters used for data simulation, execute the following commands in terminal:

```
Rscript simulation/code/set_epi_para.R
Rscript simulation/code/set_lab_para.R
```

"Rscript" may need to be replaced by platform-specific command for running R in batch mode. The generated parameter values will appear in the folder "simulation/parameters/" as R datasets, which are to be read in by "data_gen.R" and "data_gen_2infweek.R". The parameters used for the simulation studies in the paper are given in directory "simulation/parameters/provided/". They can be copied to the directory "simulation/parameters/" to reproduce the simulation results in the paper if needed.

(b) To generate one replicate of data, execute the following commands in terminal:

```
Rscript simulation/code/data_gen.R 1
Rscript simulation/code/data_gen_2infweek.R 1
Rscript simulation/code/lab_agg.R 1
```

The argument (1 as an example) after the file names specify the index of the data generated. To generate 100 replicates, set `nepi=100` in the file "simulation/code/batch_data_gen.bash" and execute the following command in terminal:

```
bash batch_data_gen.bash
```

Simulated datasets are saved in the directory "simulation/sim_data/".

3. Run MCMC for statistical inference

(a) Compile the `C` code by execute the following commands in terminal:

```
gcc simulation/code/mcmc_template.c -lm -O2 -o run_mcmc
gcc simulation/code/mcmc_2infweek_template.c -lm -O2 -o run_2infweek_mcmc
gcc simulation/code/mcmc_agg_template.c -lm -O2 -o run_agg_mcmc
```

"run_mcmc", "run_2infweek_mcmc", and "run_agg_mcmc" are executable files that take arguments and run MCMC.

(b) Run

- "run_mcmc" takes four arguments:
  - data_label specifies the data configuration, which can be one of the following
    - balanced_combo1: balanced design with 2% lab-validation
    - balanced_combo2: balanced design with 5% lab-validation
    - balanced_combo3: balanced design with 10% lab-validation
    - imbalanced_combo1: imbalanced design I/II with 2% lab-validation
    - imbalanced_combo2: imbalanced design I with 5% lab-validation
    - imbalanced_combo3: imbalanced design II with 5% lab-validation
    - imbalanced_combo4: imbalanced design I with 10% lab-validation
    - imbalanced_combo5: imbalanced design II with 10% lab-validation
    - balanced2_combo1: balanced design with 20% severe cases lab-tested
    - real: lab-validation design resembling the real surveillance data
  - index_epi specifies the dataset index. It is an integer between 1 and nepi where nepi is the total number of datasets generated for simulation.
  - index_chain specifies the MCMC chain index. It is an integer between

  1 and nchain, where nchain is the number of chains to be run per dataset. We set nchain=5 in the simulation studies.

  - Y_sampler specifies the sampling method for case counts, 1 for Markov basis sampling (MBS) and 2 for Metropolized independence sampling (MIS).

  For example, to run the 4th MCMC chain for dataset 1 with configuration balanced_combo1 using MIS, execute the following commands in terminal:

  ```
  ./run_mcmc balanced_combo1 1 4 2
  Rscript simulation/code/txt2RData.R balanced_combo1 1 4 2
  ```

  Note that the run time may take as long as about 50 hours. For a given data configuration and choice of sampling method, we run nepi*nchain programs in parallel on a high performance computing (HPC) cluster. "batch_mcmc.sbatch" in the directory "simulation/code/" is the file used for submitting job requests on HPC at University of Florida (UF) with a slurm scheduler.

- "run_2infweek_mcmc" takes three arguments:

  - data_label specifies the data configuration with two options: balanced_combo1 and imbalanced_combo1.
  - index_epi specifies the dataset index. It is an integer between 1 and nepi where nepi is the total number datasets generated for simulation.
  - index_chain specifies the MCMC chain index. It is an integer between 1 and nchain, where nchain is the number of chains to run per dataset. We set nchain=5 in the simulation studies.

  For example, to run the 4th MCMC chain for dataset 1 with configuration balanced_combo1, use

```
./run_2infweek_mcmc balanced_combo1 1 4
Rscript simulation/code/txt2RData_2infweek.R balanced_combo1 1 4
```

- "run_agg_mcmc" takes four arguments:
  - data_label specifies the data configuration with two options, balanced_combo1 and imbalanced_combo1.
  - index_epi specifies the dataset index. It is a integer between 1 and nepi where nepi is the total number of datasets generated for simulation.
  - index_chain specifies the MCMC chain index. It is an integer between 1 and nchain, where nchain is the number of chains to run per data set. We set nchain=5 in the simulation studies.
  - agg_method specifies the lab-data aggregation method. Aggregation by neighborhood is denoted by agg and aggregation by province is denoted by agg_prov.

  For example, to run the 4th mcmc chain for dataset 1 with configuration balanced_combo1 and aggregation by province, execute the following commands in terminal:

  ```
  ./run_2infweek_mcmc balanced_combo1 1 4 agg_prov
  Rscript simulation/code/txt2RData_agg.R balanced_combo1 1 4 agg_prov
  ```

All MCMC samples are output to the folder "simulation/output_chains/".

4. Collect results. After nepi*nchain runs for all data configurations finish, results are summarized for producing figures by executing the following commands in terminal:

```
Rscript simulation/code/collect_bala_imba_results.R
```

```
Rscript simulation/code/collect_agg_results.R
Rscript simulation/code/collect_2infweek_results.R
Rscript simulation/code/collect_block_element_results.R
Rscript simulation/code/collect_severe_results.R
```

The summary results are saved in the folder "`simulation/results_summary/`". The summary results from our simulation studies are provided in the folder "`simulation/results_summary/provided/`".

5. Figures are produced by

```
Rscript simulation/code/plot_figures.R
```

and output to the folder "`simulation/figures/`". The file "`plot_figures.R`" reads in summary results from the folder "`simulation/results_summary/provided/`" and some necessary parameters values from "`simulation/parameters/provided/`". If you want to use your own MCMC summary results and parameters values for plotting, you need to change the directories in "`plot_figures.R`" to "`simulation/results_summary/`" and "`simulation/parameters/`".

5 Real Data Analysis (Case Study)

5.1 Contents

- "`real_data_analysis/code/`": folder containing the code used for the case study.
  - "`mcmc_realdata_template.c`": main MCMC code. MCMC samples are output to "`real_data_analysis/results/`" as `txt` files.

  - “datastructure_agg.h”, “distribution.h”, “mathfunc.h” and “matrix.h”: header files needed to run the C code.
  - “txt2RData.R”: R code that converts txt files of the MCMC samples into RData files and save them in the folder “real_data_analysis/results/”.
  - “plot_figures.R”: R code for plotting figures 3, 4, S1 and S23-S28 as well as summarizing tables 2, S1 and S2.

- “real_data_analysis/results/”: folder storing MCMC samples, both txt and RData formats. Five hundred MCMC samples of our case study in RData format is provided in the folder “real_data_analysis/results/provided/”. “real_data_analysis/code/plot_figures.R” currently produces figures based on the provided samples.

- “real_data_analysis/figures/”: folder storing figures for the case study.

5.2 Instructions for data analysis

1. Copy the directories data and real_data_analysis to the working directory, or you can work directly from the unzipped root directory “path/code_v2/”, where “path” is the folder to which you unzipped the code and data package.

2. To compile the C code, execute

```
gcc real_data_analysis/code/mcmc_realdata_template.c -lm -O2 -o run_real_mcmc
```

   run_real_mcmc is an executable file that take an argument and run MCMC. The only argument is the chain index. It is an integer between 1 and nchain, where nchain is the number of chains to run for the case study. We set nchain=10 in the paper. For example, to run the 4th MCMC chain for the data analysis, execute the following commands in terminal:

```
./run_real_mcmc 4
Rscript real_data_analysis/code/txt2RData.R 4
```

We run 10 chains in parallel on a high performance computing cluster. "batch_real_mcmc.sbatch" in the folder "real_data_analysis/code/" is the file used for submitting job requests on the UF HPC with a slurm scheduler.

3. Produce figures by executing

```
Rscript real_data_analysis/code/plot_figures.R
```

6 Adding noise to the real surveillance counts

The following R code is not intended for readers to run but is to show how much noise was added to the real surveillance data.

```
rm(list=ls())
load("2009_south5_real.RData")
data.list <- ls()
set.seed(****)
x<-rbinom(prod(dim(Yv)),1,0.5)
x <- x*2 - 1
X<-array(x, dim(Yv))
k<-which(Yv>=20 & Yv <50, arr.ind=TRUE)
```

```
if(length(k)>0) Yv[k]<- Yv[k] + X[k]
k<-which(Yv>=50 & Yv <100, arr.ind=TRUE)
if(length(k)>0) Yv[k]<- Yv[k] + X[k]*3
k<-which(Yv>=100 & Yv <500, arr.ind=TRUE)
if(length(k)>0) Yv[k]<- Yv[k] + X[k]*5
k<-which(Yv>=500, arr.ind=TRUE)
if(length(k)>0) Yv[k]<- Yv[k] + X[k]*10

Z.sum <- apply(Z, 1:3, sum)
k<-which(Yv<Z.sum, arr.ind=TRUE)
if(length(k)>0) Yv[k]<-Z.sum[k]

save(list=data.list, file= "data/2009_south5.RData")
X<-matrix(as.vector(Yv), byrow=TRUE, nrow=dim(Yv)[3]*dim(Yv)[2], ncol=dim(Yv)[1])
write.table(X, file="data/2009_south5_Yv_realdata.txt",
            append=FALSE, row.names = FALSE, col.names = FALSE)
```