Joint Downscaler Software for Speciated $PM_{2.5}$

Colin Rundel, Erin Schliep, Alan Gelfand, David Holland

March 26, 2014

Contents

1 Preliminaries

This section describes important details with regard to the structure and organization of this software.

1. The current version of this software is tailored for the specific data files used for our analysis of Speciated $PM_{2.5}$ in Rundel, et al. 2014. This dependency is primarily in the form of assumptions about the structure of the input data from the FRM, CSN, IMPROVE, and CMAQ sources, namely in the form of data file formats and data column naming conventions. We will indicate in this manual where and when we make these assumptions in order to help identify where modifications are necessary to accommodate new data sources.
2. The numerical prefix of each R script indicates the order in which each file must be run as later files require the output of the earlier files to function correctly.
3. All R scripts should be run from the directory containing them as they make assumptions about the location of the data, include, and output folders relative to that location. This can also be achieved within R by setting the working directory of the session to the containing directory.
4. The computational time and space requires for the model are substantial. Each model run (52 weeks) produces about 5 - 6 gigabytes and takes approximately 8 hours to fit and 8 hours to predict each weekly model on a high end 8 core system.

2 Files

2.1 0-Setup.R

This script is responsible for installing all R packages necessary for the proper function of the subsequent R scripts. As of this writing this includes the following packages and their current versions.

- coda (0.16-1)
- fields (6.9.1)
- lubridate (1.3.3)
- plyr (1.8.1)
- raster (2.2-31)
- Rcpp (0.11.1)
- RcppArmadillo (0.4.100.2.1)

It should only be necessary to run this script once for any given R installation as the packages do not need to be reinstalled.

2.2 1-PrepData.R

2.2.1 Raw Data

This script is responsible for processing and standardizing the raw data from each of the three monitoring networks (FRM, CSN, IMPROVE) as well as CMAQ data. Each input file or set of files is handled by a separate section of the script, each of which is indicated by comments. For each input source the goal is to aggregate the raw data such that processed output data files will contain no more than one reported observation for each unique pair of location and date (multiple observations at the same location and date are collapsed by taking the average ignoring any missing values.)

The data is also subsetted to only contain columns for variables of interest in or latter analyses, specifically

- site
- longitude
- latitude
- date
- sulfate
- nitrate
- ammonium
- oc
- ec
- carbon
- soil
- pm25.

The R data frames of the full data (no processing beyond reading into R) and the aggregated and subsetted data are saved in an Rdata file in the Data subfolder using network name as the file prefix (frm.Rdata, csn.Rdata, improve.Rdata, and cmaq.Rdata). The names of the input and output files can be modified using the INPUT and OUTPUT variables at the beginning of the script.

The code for loading each raw input data is specific to each particular input file as different files use different naming conventions for columns, include different data, use different coding schemes to indicate missingness, etc. The use of different input data will necessitate the modification of the relevant section of this file such that subsetted / aggregated data is saved in a format that contains all of the columns given above.

2.2.2 Weekly Aggregation

Due to the temporal misaligned of the monitoring data we have decided to aggregate data to weekly observations. This is done in the same way that duplicate observations were removed, by calculating the average of all observations from the same location and week. We define January 1st as occurring in the first week and that each new week starts on each subsequent Sunday.

This script section is also responsible for matching each network observation to the nearest CMAQ grid cell, sites without a close CMAQ match are excluded. The matched data from both the network and CMAQ is reshaped to construct data matrices with weekly dates along the rows (52) and locations across the columns, any missing observations are set to NA. These matrices are construct for all species and collected in a list stored in data_sp. These reshaped results for CMAQ are stored in cmaq_sp and the network location data in locs. These revised data are save in combied_frm.Rdata, combined_csn.Rdata, and combined_improve.Rdata in the Data subfolder.

2.2.3 Formatting / Hold outs

This final stage involves modifying the combined data files into the format that is used by the model implementation. This involves splitting the combined data by week and network, C, I, and F are the aggregated data from CSN, IMPROVE, and FRM for the given week. Q_C, Q_I, and Q_F are then the matched CMAQ data for each network and s_C, s_I, and s_F the latitude and longitude of each station.

For these data sets “good” locations for each week are established by selecting only locations that have no missing observations. The “bad” locations are then included in the validation set which also includes a randomly selected 10% of all locations for each network. Therefore this validation set varies from week to week. These validation data are collected into P, Q_P, and s_P respectively and removed from the original data. These resulting data are saved in settings.Rdata in the Data subdirectory.

2.3 2-GenRast.R

This script is responsible for producing simplified versions of the CMAQ data that are used for continental scale prediction and plotting. The output is a collection of weekly (cmaq_weekly_rasts.Rdata) or seasonal (cmaq_seasonal_rasts.Rdata) rasters which have aggregated from the original 12 km grid cells to a 0.5° by 0.5° grid. This scale is adjustable through the ratio variable defined at the beginning of the script, smaller ratio values produce higher resolution rasters. Note that increasing the raster resolution will dramatically increase the computation time needed for prediction.

2.4 3-Joint_downscaler.R

This script is responsible for performing modeling fitting and validation prediction for the weekly data. It compiles and loads the C++ code (downscaler_joint_tobit.cpp) responsible for model fitting. A description of the common options and output is given below.

2.4.1 Options

- **verbose** - a logical value indicating if model results should be saved, useful for testing in order to avoid over writing existing results.
- **verbose** - a logical value indicating if the MCMC should include verbose output which includes parameter values and acceptance rates for parameters updated via Metropolis-Hastings.
- **predict** - a logical value indicating if predictions should be made for the hold out values.
- **bivar** - a logical value indicating if the bivariate variation of the joint model should be fit. If bivar is true then it is necessary to specify which species to considered, this is done using bisp1 and bisp2 which should be set to the index of the desired species (e.g. 1 - sulfate, 2 - nitrate, 3 - ammonium, 4 - soil, 5 - carbon).
- **fix_phi** - a logical value indicating if spatial range parameters ϕ_1, ϕ_2, ϕ_3, ϕ_4, ϕ_5, and ξ should be updated.
- **fix_beta1** - a logical value indicating if the β_1^i for $i \in (1, \ldots, 5, o)$ should be fixed. Default values are specified by the beta1 vector.
- **nburn** - an integer value for the number of MCMC burn-in iterations.
- **niter** - an integer value for the number of MCMC iterations to save.
- **nthin** - an integer value for the multiplier of MCMC iterations, the total iterations is equal to (nburn + niter) $\times$ nthin.
- **weeks** - an integer vector of the week indexes to fit.

2.4.2 Output

Model fitting results are saved in the output subdirectory, an additional directory is created within output based on the given options. Each weekly result is saved in a separate Rdata file named using the week's index. Each data file contains a list r which contains all model settings, posterior samples (post), posterior predictive samples (pred), and acceptance rates (accept).

2.4.3 downscaler_joint_tobit.cpp

This file represents the C++ implementation of the core MCMC algorithm for fitting the joint model.

2.5 4-Prediction.R

This script is responsible for performing the large scale prediction of joint model results. It compiles and loads the C++ code (downscaler_joint_predict.cpp) responsible for predicting at all locations in the CMAQ rasters from 2-GenRast.R. A description of the common options and output is given below.

2.5.1 Options

- **n_samples** - an integer value for the number of posterior samples to use for continental scale prediction. Reducing n_samples decreases the overall computational time needed.
- **runs** - a character vector of directory names in the output directory that will be used for prediction.

2.5.2 Output

For each weekly result a set of rasters will be produced and saved (e.g. 1_rast_predict.Rdata) in the same output directory. After all weekly results are processed the rasters are then aggregated by season and saved (e.g. 1_rasts.Rdata).

2.5.3 downscaler_joint_predict.cpp

This file represents the C++ implementation of the core prediction algorithm for the joint model.

2.6 Additional Files

- *include/adapt_mcmc.hpp* - tools for adaptive MCMC based on Vihola 2011.
- *include/assert.hpp* - run time assert tools for error checking.
- *include/distance.hpp* - C++ implementations of great circle distance.
- *util/distance_util.R* - R / C++ implementations of efficient distance functions.