DICE User Manual

Predictive Science Inc.

June 14, 2019

Contents

1 Introduction

The **DICE** package implements a global model of both directly transmitted and vector-borne diseases within populations with arbitrary spatial resolution, with the objective of making best use of available data at different spatial scales. Building on previous work which was limited to influenza and a single spatial region [12,13], **DICE** uses either mechanistic compartmental models or statistical techniques to fit and when relevant forecast the time evolution of the disease. The deterministic compartmental models (SIR, SEIR etc.), are applied on various interacting spatial scales, and use a robust Markov-Chain-Monte-Carlo (MCMC) fitting procedure that can quickly characterize disease incidence profiles, of either coupled or uncoupled regions, in real-time providing estimates for:

- Within, and if applicable between, region epidemic transmissibility (measured by the basic reproduction number R_e and the next-generation matrix [10], if applicable)
- individual level epidemic severity (as described by the proportion p_C of infections that result in clinical cases)

The statistical techniques are based on Seasonal Autoregressive Integrated Moving Average (SARIMA) models which have received the most attention in this field. **DICE** uses a MySQL database which includes incidence profiles for a large number of directly transmitted and vector-borne diseases along with population and climate data (precipitation, temperature and specific humidity). These data sets are provided on spatial scales of various, increasing, resolutions. For each data source/disease, the incidence, population and climate data use the same spatial and temporal scales. The latter can be either daily, weekly, monthly or yearly. School opening/closing schedules for the US are also included in the database. This dataset, along with the specific humidity data, are needed for three of the four different models for the time dependence of the basic reproduction number $R_e(t)$ that the user can choose from. The school and climate data can also be used as covariates when fitting/forecasting the incidence using the SARIMA statistical methods. **DICE** also includes a compact data set of Military Data for the 2009-2010 H1N1 pandemic season. This Military data set includes weekly numbers of ILI cases along with school vacation schedule and specific humidity for 50 military bases. The **DICE** MySQL database can be explored using a series of web based tools described here `http://predsci.com/id_data/`.

Upon completion of the MCMC fitting or statistical procedure, the **DICE** package analyzes the results and produces an extensive set of publication-quality plots (in a PDF or PNG format) and tables (in .csv format). The complete history of the run and the fitting procedure is saved as an '`RData`' file which the user can later load and further analyze.

2 Quick Start Guide

The **DICE** package is accessible as a GitHub repository. Before downloading the repository, there are a few software requirements:

2.1 Installation Instructions

In a command line environment navigate to the directory 'mydir' where you would like the source code for **DICE** to be located

```
cd /.../mydir
```

and clone from GitHub

```
git clone https://github.com/predsci/DICE.git
```

This creates a sub-directory "DICE" that contains all of the files required to compile. For a global install, execute the provided python compile script:

```
cd DICE
./compile.py
```

For a local install, execute the following commands:

```
cd DICE
/bin/rm -rf DICE_1.0.tar.gz dice/src/*.o dice/src/*.so
R CMD build DICE
R CMD INSTALL -l /.../mylib DICE
```

where 'mylib' is the local directory in which you would like to store the R library. A common problem encountered here is that for Mac OS X, the fortran libraries `-lgfortran`, `-lquadmath`, and `-lm` are not located where R expects them. This will result in an error like:

```
ld: library not found for -lgfortran
```

following a call to `clang`. This can be corrected by opening "/Library/Frameworks/R.framework/Resources/etc/Makeconf" and changing the "FLIBS" variable to point at the location of libgfortran.a and libquadmath.a in the user's local installation. In one case this meant changing

```
FLIBS = -L/usr/local/lib/gcc/x86_64-apple-darwin13.0.0/4.8.2 -lgfortran
    -lquadmath -lm
```

to

```
FLIBS = -L/opt/local/lib/gcc5/gcc/x86_64-apple-darwin15/5.3.0 -lgfortran
    -lquadmath -lm
```

for R to find the fortran libraries installed with gcc5.3 via MacPorts. When unsure about the correct directory, it is suggested to search your file system for 'libgfortran.a' and try the corresponding directory. Similarly, the "F77" variable in Makeconf dictates which fortran compiler is called. Depending on the version of R, this defaults to "F77 = gfortran-A.B". In the case mentioned above (gcc compiler installed via MacPorts), changing

```
F77 = gfortran-4.8
```

to

```
F77 = gfortran
```

allowed R to call the fortran compiler correctly.

To check that the package installed, open an R environment and enter

```
require("DICE")
```

This will load the **DICE** package as well as a number of dependency packages. No errors on load generally indicates a successful compile. The following section presents some example scripts to begin running **DICE**.

2.2 Examples

In this section we define the minimal, most important, set of parameters that the user will need to set for a **DICE** run. We focus on the parameters that control the main features of the package and discuss what values to choose for them for a few useful examples. For a more detailed description of the parameters the users should consult sections 3 and 5.

The **DICE** package is currently set to run using either a MySQL database or by directly accessing the the CDC influenza-like-illness database for the most up-to-date data. The user

should first choose a disease to study (e.g. dengue, influenza), a country (or region) and season, and a method (mechanistic or statistical). Once these have been selected other choices can be made that are specific to the method used and the disease. For example, in the case of a mechanistic method (and provided that incidence data is available) the user should choose if to model the country/region directly or as a coupled or uncoupled aggregate of higher resolution sub-regions. The user can also select a specific compartmental model (e.g. SIR, SEIR and models that describe both the human and the vector states explicitly). Additional selections control the details of the MCMC fitting procedure. We recommend using our web based tools to explore the database.

2.2.1 Setting Up and Running DICE

After selecting the *disease* (e.g. 'flu'), country (e.g. 'usa'), and *method* ('stat' or 'mech') the user should choose a season by setting the parameter *year* to the start year of the season (e.g. for the 2015-2016 influenza season use: $year = 2015$). The spatial region we want to model and at what level of granularity we want to do it are controlled by the parameters mod_level and fit_level, respectively. In the case of CDC data (first example below), data is publicly available on the national, regional (the regions are the ten HHS regions, see figure 2), and state levels. In the first example below we set $mod\_level = 2$ and $fit\_level = 3$. This choice means that we are modeling the national incidence data using the ten HHS regional-level incidence data. We also want the HHS regions to be coupled and therefore set: $isingle = 0$. (With these choices the national level curve we are modeling is a result of both within and between region infections.) The following command produces a quick exploratory **DICE** run:

```
> output <- runDICE(disease = 'flu', RegState = 'usa', method = 'mech',
            year = 2017, mod_level = 2, fit_level = 3, isingle = 0)
```

If we want to keep all of our choices the same but decouple the ten HHS regions so that our prediction for the national CDC profile is a weighted sum of our best predictions for each of the ten HHS regions, we only change the parameter *isingle* from zero to one:

```
> output <- runDICE(disease = 'flu', RegState = 'usa', method = 'mech',
            year = 2017, mod_level = 2, fit_level = 3, isingle = 1)
```

With this choice, infection can only happen within each HHS region and there is no interaction between the regions. As explained in detail in section 3 this still allows each region to have it's own set of optimal parameters, e.g. epidemic onset time, force of infection etc..

To demonstrate how to model a single HHS region using its' state level data we use the CDC dataset, $disease =$ "flu", and set: $mod\_level = 3$ and $fit\_level = 4$. We choose the 2014-2015 season, $year = 2014$, HHS region number 9 ($RegState = 9$) with coupling between the states that make-up this region ($isingle = 0$):

```
> output <- runDICE(disease = 'flu', RegState = 9, year = 2014, mod_level = 3,
            fit_level = 4, isingle = 0)
```

In these two examples the **DICE** code was using the default SIR compartmental model. To use an SEIR model for the flu set $epi\_model =$ "$SEIR$" or $epi\_model = 2$:

```
> output <- runDICE(disease = 'flu', RegState = 'usa', method = 'mech',
            year = 2017, mod_level = 2, fit_level = 3, isingle = 1,
            epi_model = 'SEIR')
```

For both SIR and SEIR like models, four different models are used in **DICE** to describe the basic reproduction number R_e. Below we set it to use a model that depends on the weekly averaged specific humidity and the school vacation schedule ($model = 3$):

```
> output <- runDICE(disease = 'flu', RegState = 'usa', method = 'mech',
            year = 2017, mod_level = 2, fit_level = 3, isingle = 0,
            epi_model = 'SIR', model = 3)
```

For more on the different models for the basic reproduction number R_e see sub-section 3.5.

To demonstrate the use of statistical models ($method$ = "*stat*") we use the 2010 Dengue data from Brazil:

```
output <- runDICE(disease = 'dengue', data_source=NULL, year=2010,
          mod_level = 2, fit_level = 3, RegState = 'BR', method = 'stat',
          arima_model = NULL)
```

Since mod_level and fit_level were set to 2 and 3 respectively, the statistical model first fits the nation level incidence directly followed by the incidence of each region. The regional fits are then combined (with proper weights) to form an indirect fit to the national level data. When the *arima_model* variable is set to *NULL*, **DICE** uses a pre-defined set of SARIMA models and chooses the best one for national incidence and for each of the regions. To select a specific arima model the user needs to define the *arima_model* list, e.g.,

```
arima_model = list(p = 1, d = 0, q = 0, P = 3, D = 1, Q = 0)
output <- runDICE(disease = 'dengue', data_source=NULL,
          year=2010, mod_level = 2, fit_level = 3, RegState = 'BR',
          method = 'stat', arima_model = arima_model)
```

When an ARIMA model is defined, the user can also select a covariate and a lag time for the covariate.(The units of the lag time are the same as those of the incidence data.) Any of the climate data (temperature, precipitation and specific humidity) can be used as a covariate. In the following example we model the weekly Dengue data for Singapore with a specific SARIMA model (key word 'arima_model') and with precipitation as a covariate (key word 'covar') with a lag time (key word 'covar_lag') of two weeks:

```
arima_model = list(p = 1, d = 0, q = 0, P = 1, D = 1, Q = 0)
output <- runDICE(disease = 'dengue', data_source=NULL,
          year=2016, mod_level = 2, fit_level = 2, RegState = 'SG',
          method = 'stat', arima_model = arima_model,
          covar = 'precip', covar_lag = 2)
```

The Dengue data can be modeled with mechanistic methods with the vector states modeled either explicitly or assuming that they are in a steady state (and only the human states, SIR or SEIR, are modeled). For example, setting the key word epi_model to either 3 or 'vsir', selects an SIR model for the humans and an SI model for the vector. Below we run this model uncoupled ($isingle = 0$) for the 2015 Dengue season in Mexico.

```
output <- runDICE(disease = 'dengue', data_source=NULL,
          year=2015, mod_level = 2, fit_level = 3, RegState = 'MX',
          method = 'mech', epi_model = 'vsir', isingle = 0)
```

Finally, for statistical modeling of the CDC influenza data (2017-2018 season) with a $(011)(010)_{12}$ SARIMA model and specific humidity as a covariate (with no lag time) use:

```
arima_model = list(p = 0, d = 1, q = 1, P = 0, D = 1, Q = 0)
output <- runDICE(disease = 'flu', data_source='cdc', year=2017,
          mod_level = 2, fit_level = 3, RegState = 'usa', method = 'stat',
          arima_model = arima_model, covar = 'sh', covar_lag = 0)
```

By default all the output files and plots of a **DICE** run are written into a sub-directory whose name is built using the run information: country and spatial level, disease, year etc.. **DICE** will create this sub-directory (if it does not already exist) under the working directory. The user can choose the name of this sub-directory by setting the keyword *subDir* when calling **DICE**. In this example we set it to 'tests':

```
> output <- runDICE(disease = 'flu', data_source='cdc',  year = 2017,
            mod_level = 3, fit_level = 4, RegState = 3, isingle = 0,
            model = 3, subDir = 'tests')
```

The **DICE** package produces plot files in either 'pdf' or 'png' formats with the default being 'pdf'. To control the format of the output plots the user needs to set the keyword *device* which is defined as an array in **DICE** so that multiple file formats can be created in a single run. To produce plot files in a 'png' format for example use:

```
> output <- runDICE(disease = 'flu', year = 2015, mod_level = 2, fit_level = 3,
      RegState = 'usa', isingle = 1, model = 4, device = 'png')
```

To produce both 'pdf' and 'png' file formats for each plot use:

```
(disease = 'flu', year = 2017,  mod_level = 2, fit_level = 3,
      RegState = 'usa', isingle = 1, model = 4,  device = c('pdf','png'))
```

DICE supports two methods for plotting the results: one using the basic plot set of commands and one that uses the ggplot2 package. By setting the keyword 'plot' to 1 or 2 the user can select between these two methods (basic plot or ggplot2, respectively). By default the keyword plot is set to 1 (which is the faster option). If the user sets it to 0, **DICE** will not produce any plots.

For publication-level results the user would want to ensure that the posterior densities of the parameters optimized by **DICE** are converged. This is done by increasing the number of MCMC chains we run (from it's default value of $nreal = 1$) to 3 or even 5 and by increasing the number of steps/trials ($nMCMC$) in each chain to 10^6 or more. The example below demonstrates how to run three MCMC chains ($nreal = 3$) each with 10^6 steps ($nMCMC$=10^6) modeling the 2015-2016 national-level CDC ILI data ($disease =$ 'flu', $data\_source =$ 'cdc', $year = 2015$, $mod\_level = 2$) using HHS region-level data ($fit\_level = 3$) with coupling between the regions ($isingle = 0$):

```
> output <- runDICE(disease = 'flu', data_source='cdc', RegState = 'usa',
            year = 2015, mod_level = 2, fit_level = 3, isingle = 0, nreal = 3,
            nMCMC = 1e6)
```

To run **DICE** in a forecast or predictive mode you can set the number of weeks (or months) the code uses in the fit (key word 'nfit') to be lower than the number of weeks/months in the season (for weekly incidence $nData$ is typically 52 but sometimes 53). In the example below we use only the first 35 weeks of CDC data for the $2013-2014$ season to fit HHS region number 5 ($mod\_level = 3$, $RegState = 5$) using uncoupled ($isingle = 1$) state level data ($fit\_level = 4$):

```
> output <- runDICE(disease = 'flu', data_source='cdc', RegState = 5,
            year = 2013, mod_level = 3, fit_level = 4,
            isingle = 1, nMCMC = 1e6, nfit = 35)
```

The lower limit for this parameter is 10 for weekly data. If the user sets it to less than that, **DICE** will reset it to 10 and print a warning message to the screen. When running on the current season, using all publicly available CDC data, the **DICE** package will always run in a

predictive/forecasting mode, since it calculates the profile for an entire season which is typically 52 weeks long.

The next three parameters control details of the MCMC fitting/forecast procedure. First, the user can choose to use an informative Gaussian prior distribution for the model parameters (as opposed to the default log-uniform distribution). This option is currently available *ONLY* for the uncoupled runs of CDC flu data and it uses the posterior distribution of the model parameters from the season that is most similar to the current season as the prior for the season we are modeling. To use a prior distribution set the key word *prior* to 1:

```
> output <- runDICE(disease = 'flu', data_source='cdc', RegState = 'usa', year = 2017,
            mod_level = 2, fit_level = 3, isingle = 1, nMCMC = 1e6, prior = 1)
```

Second, early in the season when running in a forecast mode it is often useful to use data augmentation and make maximum use of prior data within a mechanistic framework. **DICE** can augment any incidence data using average historic data (key word 'da' set to 1) or using the season that is most similar to the current season (key word 'da' set to 2).

```
> output <- runDICE(disease = 'flu', data_source='cdc', RegState = usa', year = 2017,
            mod_level = 2, fit_level = 3, isingle = 1, nMCMC = 1e6, da = 2)
```

By default, **DICE** sets $da = 0$, i.e. no data augmentation. Third, to allow the MCMC chains to properly explore parameter space the user can change the temperature of the MCMC procedure from its default value of 1 ($Temp = 1$) to 5, 10 ($Temp = 5$, or $Temp = 10$) or any other number:

```
> output <- runDICE(disease = 'flu', data_source='cdc', RegState = usa', year = 2017,
            mod_level = 2, fit_level = 3, isingle = 1, nMCMC = 1e6, da = 1,
            Temp = 10)
```

2.3 Batch Runs

In many cases, the user would prefer running the package from the command line using a script and the `Rscript` or `R CMD BATCH` command. After cloning **DICE** as described in sub-section 2.1 the user will see a directory called 'examples' with example scripts in it. In particular, the 'batch-example.R' and 'dengue-example.R' scripts (setup to run on flu and dengue data respectively) set default values for all of the parameters described above, but also allow the user to modify these selections by editing the script itself or while calling it from the command line. To see the 'help' for this script use:

```
> cd examples
> Rscript batch-example.R
```

To run the script, the user must call it with one or more parameters. For example:

```
> Rscript batch-example.R year 2014
```

will run **DICE** on the 20014-2015 season with all other parameters set to their default values in 'batch-example.R'. To change these values either edit the script file itself or set the parameters from the command line, e.g.

```
> Rscript batch-example.R data_source cdc year 2015 nfit 45 model 3 nMCMC 1e6
          nreal 3 isingle 1 device png
```

To run **DICE** on Dengue data from Brazil for the 2012 season using a statistical model use:

```
> Rscript dengue-example.R year 2012 method stat
```

These scripts are useful when **DICE** is running in a 'production-like' mode with loops through multiple selections for the data type, region, year, model etc..

3 Mechanistic Modeling

The **DICE** package has been designed to enable epidemic modeling on an arbitrary spatial scale with or without coupling between the regions/patches, for both directly transmitted and vector-borne diseases. In practice, our modeling is limited by the availability of data. Specifically, for influenza modeling, data is publicly available from the CDC at the national, regional and state levels, where the regions are the ten HHS regions shown in figure 2. Historic state-level data is publicly available only for the past few years. State level data is also available from the Google Flu Trend project (which has been discontinued) for the 2003-2014 flu seasons. Dengue (and other vector-borne disease) data was obtained in various forms (csv files, pdf tables, etc.) from a number of country level sources (typically the web site of the country's health department) as well as health organizations: PAHO and WHO. Our discussion of mechanistic models begins with SIR and $SEIR$ compartmental models using the CDC flu data. These models are typically used to describe directly transmitted diseases. Under the assumption of a steady state solution for the vector states, these models can also be used to describe the dynamics of vector-borne diseases. We use the Dengue data to discuss compartmental models that include the vector states explicitly: SIR_H/SI_V and $SEIR_H/SEI_V$. Some aspects of the discussion, related to the spatial resolution of the data are common to both directly transmitted and vector-borne diseases. For example, our discussion of the SIR and SEIR models focuses on modeling US national influenza data using the HHS regional data or modeling a specific HHS region using state level data. We will refer to these data-set pairs, nation-region and region-state, as the low-high spatial resolutions. This idea applies also to the vector-borne case. In Section 3.3 we briefly discuss a single-population model for waterborne diseases, e.g. Cholera.

DICE gives the user the ability to model the low resolution data directly or as the aggregate of many higher spatial resolution fits. By aggregating the individual fits of the underlying higher-resolution regions (coupled or uncoupled), the result is often able to explain various features in the lower resolution incidence data that cannot be explained with a S-I-R (or S-E-I-R) model fit directly to the lower-resolution spatial region.

Our MCMC fitting procedure always begins with a direct fit of the low-resolution data without using any higher spatial resolution data. This fit uses a deterministic compartmental model (e.g. S-I-R, S-E-I-R etc.) described below with the objective of minimizing the maximum likelihood. The likelihood is calculated by comparing the predicted and observed weekly/monthly incidence profiles. Next **DICE** will use the higher spatial resolution data (and information such as the population of the spatial region, it's school vacation schedule and average weekly specific humidity) to try and predict the low resolution data. This prediction can be done by either: (1) Sequentially optimizing each high-resolution spatial region independently (using the predicted and observed ILI profile of a single region) and then calculating the low-resolution profile as the weighted sum of the individual high-resolution profiles (with the weights given by the relative population of each region), or (2) assuming that the low resolution profile results from inter-dependent dynamics among st the high-resolution spatial regions. In this, numerically more demanding scenario, the S-I-R/S-E-I-R dynamics of the high-resolution spatial regions are coupled and together generate the lower-resolution profile which is optimized. In the following two sections we describe the uncoupled and coupled schemes.

3.1 Direct Transmission - Uncoupled Models

When modeling the incidence of an uncoupled set of populations, a deterministic compartmental S-I-R (or S-E-I-R) model is used with a time dependent transmission rate $\beta_j(t)$ that is is unique to each spatial region j [8]:

$$\frac{dS_j}{dt} = -\beta_j(t)\frac{S_j I_j}{N_j}, \tag{1}$$

$$\frac{dI_j}{dt} = \beta_j(t)\frac{S_j I_j}{N_j} - \frac{I_j}{T_g}, \tag{2}$$

$$\frac{dR_j}{dt} = \frac{I_j}{T_g}. \tag{3}$$

Here S_j represents the number of susceptible individuals in region j, I_j is the number of infectious individuals in region j, R_j is the number of recovered individuals in region j, and $N_j = S_j + I_j + R_j$ is the total population of the jth region. The transmission rate $\beta_j(t)$ can be expressed as a function of generation time T_g and reproduction number $R_{ej}(t)$: $\beta_j(t) = R_{ej}(t)/T_g$. The time-parameter t_{0j}, which is unique for each region j, denotes the onset time of the epidemic in the region. It is used to set the initial conditions for the uncoupled S-I-R equations as follows:

$$\begin{aligned} S_j(t_{0j}) &= N_j - I_j(t_{0j}) \ , \\ I_j(t_{0j}) &= I_j(t_{0j}) \ , \\ R_j(t_{0j}) &= 0 \ . \end{aligned}$$

In the case of an S-E-I-R model a latent period, describing individuals who are exposed but not yet infectious, is introduced. These individuals are referred to as 'Exposed' and described by the letter 'E' in the S-E-I-R model:

$$\frac{dS_j}{dt} = -\beta_j(t)\frac{S_j I_j}{N_j}, \tag{4}$$

$$\frac{dE_j}{dt} = \beta_j(t)\frac{S_j I_j}{N_j} - \sigma E_j, \tag{5}$$

$$\frac{dI_j}{dt} = \sigma E_j - \frac{I_j}{T_g}, \tag{6}$$

$$\frac{dR_j}{dt} = \frac{I_j}{T_g}. \tag{7}$$

The average duration of the latent state is described by $\frac{1}{\sigma}$. The S-I-R (and S-E-I-R) equations model the total population, but the data is the number of weekly observed cases or incidence rate (I^R). The weekly incidence rate is calculated from the continuous S-I-R (and S-E-I-R) models by discretizing the rate-of-infection term, $\beta_j(t)\frac{S_j I_j}{N_j}$ in the case of S-I-R:

$$I_j^R(t_i) = B_j + p_j^C \int_{t_i-1-\Delta_t}^{t_i-\Delta_t} \beta_j(t)\frac{S_j(t) I_j(t)}{N_j} dt \ , \tag{8}$$

scaling by percent clinical p_C^j, and adding a baseline B_j. p_j^C is the proportion of infectious individuals that present themselves to a clinic with ILI symptoms and B_j is a constant that estimates non-S-I-R or false-disease cases. The integral runs over one week determining the number of model cases for week t_i. Δ_t approximates the time delay from when an individual becomes infectious to when they visit a sentinel provider for disease symptoms. Unless otherwise specified, it is assumed that $\Delta_t = 0.5$ weeks.

In the case of an S-E-I-R model we use:

$$I_j^R(t_i) = B_j + p_j^C \int_{t_i-1-\Delta_t}^{t_i-\Delta_t} \sigma E_j dt \ , \tag{9}$$

Eqs. (8) and (9) are how **DICE** relates its internal, continuous S-I-R/S-E-I-R models to the discrete incidence data. In section 6 we describe the procedure used for fitting this property to a diseases profile.

Under the assumption that the vector (e.g. a mosquito) dynamics are fast compared to those of humans (i.e., a significantly faster life-cycle) and quickly reach a quasi-equilibrium state/-concentration the simple S-I-R or S-E-I-R models can be used to approximate the dynamics of vector-borne diseases. **DICE** supports this approximation but it also supports two models that describe the spread of infection between humans via mosquitoes (or other vectors, such as ticks, that take a single blood meal from a human/host and then move on). These models are described in the next sub-section.

3.2 Indirect Transmission - Uncoupled Models

Vector-borne diseases cannot be directly passed between humans/hosts, they require an intermediate (vector) for transmission. When a susceptible vector bites an infectious host it has a probability of becoming infected and soon after infectious. If while infectious it bites a susceptible host/human it has a certain probability of infecting the human. The simplest model that describes this scenario includes S-I-R compartments for the host/human and S-I for the vector. Transmission occurs only when a healthy vector is infected, when it bites an infectious human. The $\mathrm{SIR_H/SI_V}$ system describing this scenario includes three equations for the human/host:

$$\frac{dS_{H,j}}{dt} = -bT_{HV}\frac{S_{H,j}I_{V,j}}{N_{H,j}}, \tag{10}$$

$$\frac{dI_{H,j}}{dt} = bT_{HV}\frac{S_{H,j}I_{V,j}}{N_{H,j}} - \frac{I_{H,j}}{T_g}, \tag{11}$$

$$\frac{dR_{H,j}}{dt} = \frac{I_{H,j}}{T_g}. \tag{12}$$

where subscripts 'H' and 'V' denote the human and vector states, 'b' is the biting rate of the mosquito and T_{HV} is the probability that an infected mosquito biting a susceptible human transmits the infection. Given the fast life cycle of the vector we only include the susceptible and infectious vector states:

$$\frac{dS_{V,j}}{dt} = -bT_{VH}\frac{S_{V,j}I_{H,j}}{N_{H,j}} - \mu_V(S_{V,j} - kN_{H,j}), \tag{13}$$

$$\frac{dI_{V,j}}{dt} = bT_{VH}\frac{S_{V,j}I_{H,j}}{N_{H,j}} - \mu_V I_{V,j}, \tag{14}$$

In the above equations T_{VH} is the probability of transmission from human to vector, 'k' is the ratio of vectors to humans, and μ_V is the vector birth/death rate. As noted before, the explicit description of vector-states basically doubles the number of equations, and it introduces new parameters. **DICE** is able to fit the new parameters introduced in the Eqs. 10-14: T_{HV}, T_{VH}, b, k and μ_V. However given the large number of parameters we already fit, we recommend keeping these parameters fixed at their default values (set by **DICE**). As in the case of the simpler direct S-I-R model, weekly/monthly incidence is calculated by discretizing the rate of infections term, $bT_{HV}\frac{S_{H,j}I_{V,j}}{N_{H,j}}$:

$$I_j^R(t_i) = B_j + p_j^C \int_{t_i-1-\Delta_t}^{t_i-\Delta_t} bT_{HV}\frac{S_{H,j}I_{V,j}}{N_{H,j}}dt\ , \tag{15}$$

For vector transmission the analogue of the S-E-I-R model is an S-E-I-R/S-E-I model for the human/vector states with an average duration of latent states for both human and vector: $\frac{1}{\sigma_H}$ and $\frac{1}{\sigma_V}$. Four equations describe the human states:

$$\frac{dS_{H,j}}{dt} = -bT_{HV}\frac{S_{H,j}I_{V,j}}{N_{H,j}}, \tag{16}$$

$$\frac{dE_{H,j}}{dt} = bT_{HV}\frac{S_{H,j}I_{V,j}}{N_{H,j}} - \sigma_H E_{H,j}, \tag{17}$$

$$\frac{dI_{H,j}}{dt} = \sigma_H E_{H,j} - \frac{I_{H,j}}{T_g}, \tag{18}$$

$$\frac{dR_{H,j}}{dt} = \frac{I_{H,j}}{T_g}. \tag{19}$$

and three describe the vector states:

$$\frac{dS_{V,j}}{dt} = -bT_{VH}\frac{S_{V,j}I_{H,j}}{N_{H,j}} - \mu_V(S_{V,j} - kN_{H,j}), \tag{20}$$

$$\frac{dE_{V,j}}{dt} = bT_{VH}\frac{S_{V,j}I_{H,j}}{N_{H,j}} - E_{V,j}(\sigma_V + \mu_V), \tag{21}$$

$$\frac{dI_{V,j}}{dt} = \sigma_V E_{V,j} - \mu_V I_{V,j}, \tag{22}$$

In both the uncoupled and coupled modes (and for both direct and vector transmission) we begin by fitting the low-resolution incidence data directly using all the information we have on the low-resolution spatial region , and with the relevant set of equations, e.g. for an S-I-R model we use eqs. (1-3) whereas for S-E-I-R/S-E-I we use eqs. (16-22) . In the uncoupled case, we then continue and optimize each high-resolution spatial region independently (using the same set of equations) and calculate the low resolution profile indirectly as a weighted sum of the independent high resolution profiles of the regions/patches that compose it. The weight of each region is determined by it's relative population. In this uncoupled mode infection can occur only within a region/patch, there is no interaction (mobility) between regions/patches and the observed low-resolution incidence profile is the result of independent dynamics within decoupled higher-resolution spatial regions/patches.

3.3 Indirect Transmission - Waterborne Disease

The **DICE** package includes a single population model for waterborne diseases, e.g. Cholera. The model includes the familiar S-I-R compartments and an additional compartment 'B' which describes the Bacteria (e.g. Vibrio Cholerae) concentration in the water supply. The model assumes that:

- Water sanitation leads to death of Bacteria in the aquatic environment.
- Water treatment is the only control strategy (e.g. control strategies such as vaccination are ignored).
- The waterborne disease occurs in a relatively short period of time and it has low mortality, which is ignored.

The following new parameters describe the bacteria dynamics and transmission mechanism:

- ϵ - rate of exposure to contaminated water
- m - bacteria growth rate
- n - bacteria loss rate

- κ - pathogen concentration in water which yields a 50% of probability of infection
- δ - death rate of bacteria due to water treatment
- α contribution of each infected person to the Bacteria population in the water (sanitation)
- φ - net death ratio of bacteria $\varphi = m - n$

The resulting, single population, S-I-R-B equations are:

$$\frac{dS}{dt} = -\frac{\epsilon B}{\kappa + B} S, \tag{23}$$

$$\frac{dI}{dt} = \frac{\epsilon B}{\kappa + B} S - \frac{I}{T_g}, \tag{24}$$

$$\frac{dR}{dt} = \frac{I}{T_g}, \tag{25}$$

$$\frac{dB}{dt} = \alpha I - \delta B - \varphi B. \tag{26}$$

With T_g being the usual rate of recovery. **DICE** does not keep track of the recovered population ('R' compartment) but it does report the bacteria population as a function of time, in addition to the usual number of cases. The bacteria population is also plotted along with the incidence data and fit.

Currently, there are two Cholera data sets to which this model can be applied: Somalia (2017) and Zambia (Oct. 2017 to May 2018). The 'SIRB' model can be invoked with the keyword epi_model set to 'SIRB' or 5.

3.4 Coupled Models

In the case of a coupled modeling of the low-resolution spatial data, we begin as before by fitting the low-resolution data directly using the S-I-R or S-E-I-R equations presented in section 3.1. This gives us the direct low resolution S-I-R (or S-E-I-R) fit to the data which may or may not be able to describe the dynamics properly. Next, following Mills and Riley [10], we describe a scenario where the low spatial resolution profile is the result of the dynamics within and between higher resolution regions/patches that compose the larger low resolution region/patch. In this coupled scenario the rate at which a susceptible person in patch j becomes infected depends on: (1) the risk of infection from those in the same patch j, (2) the risk of infection from infected people from patch i who traveled to patch j, and (3) the risk of infection encountered when traveling from patch j to patch i. In the uncoupled case only the first term, which we expect to be dominant for nation/region/state level data, is present. To account for these three mechanisms, Mills and Riley [10] defined the force of infection, or the average rate that susceptible individuals in region/patch i become infected per time step, as:

$$\lambda_i(t) = \sum_{i=1}^{D} \beta_j m_{ij} \frac{\sum_{l=1}^{D} m_{lj} I_l}{\sum_{p=1}^{D} m_{pj} N_p} \tag{27}$$

where D is the total number of regions/patches and in our-case β_j is allowed to depend on time: $\beta_j(t)$. Given this force of infection we can write the coupled S-I-R equations for each region/patch as:

$$\frac{dS_j}{dt} = -\lambda_j(t) S_j, \tag{28}$$

$$\frac{dI_j}{dt} = \lambda_j(t) S_j - \frac{I_j}{T_g}, \tag{29}$$

$$\frac{dR_j}{dt} = \frac{I_j}{T_g}. \tag{30}$$

Equations (28-30) are the coupled version of eqs. (1-3) and they reduce to them in the limit of no mobility between regions/patches. In this case the mobility matrix m_{ij} is the identity matrix so that $\lambda_i(t) = \beta_i(t)\frac{I_i}{N_i}$ and we recover eqs. (1-3).

The mobility model we use, which follows Mills and Riley [10] directly, defines each element of m_{ij} as the probability for an individual from region/patch i, given that he/she made a contact, that this contact was with an individual from region/patch j:

$$m_{ij} = N_j \kappa(r_{ij}) \frac{1}{\sum_k N_k \kappa(r_{ik})} \tag{31}$$

where $\kappa(r_{ik})$ is the interaction kernel between regions/patches. This kernel is expected to depend on the geographic distance between the regions (r_{ij}) and Mills and Riley [10] have chosen to use a variation of the off-set power function for this kernel:

$$\kappa(r_{ij}) = \frac{1}{1 + (r_{ij}/s_d)^\gamma} \tag{32}$$

where s_d is a saturation distance in km and the power γ determines the amount of mixing between the regions/patches: The lower is γ the more mixing we have. In our MCMC procedure these two parameters are optimized.

The weekly or monthly incidence is calculated as before by discretizing the rate of infection term:

$$I_j^R(t_i) = B_j + p_j^C \int_{t_i-1-\Delta_t}^{t_i-\Delta_t} \lambda_j(t) S_j dt \ , \tag{33}$$

scaling by the percent clinical and adding a baseline term.

We have extended the Mills and Riley model to include the exposed but not yet infectious state, 'E', whose duration is controlled by $\frac{1}{\sigma}$:

$$\frac{dS_j}{dt} = -\lambda_j(t) S_j, \tag{34}$$

$$\frac{dE_j}{dt} = \lambda_j(t) S_j - \sigma E_j, \tag{35}$$

$$\frac{dI_j}{dt} = \sigma E_j - \frac{I_j}{T_g}, \tag{36}$$

$$\frac{dR_j}{dt} = \frac{I_j}{T_g}. \tag{37}$$

The force of infection is calculated using equation 27 and the incidence is given by:

$$I_j^R(t_i) = B_j + p_j^C \int_{t_i-1-\Delta_t}^{t_i-\Delta_t} \sigma E_j dt \ , \tag{38}$$

Currently, **DICE** does not include a coupled model with explicit modeling of the vector states. Hence, when modeling vector-borne disease the user can select between the following options:

- Uncoupled, SIR with a quasi-equilibrium assumption for the vector states: epi_model=1, isingle = 1
- Uncoupled, SEIR with a quasi-equilibrium assumption for the vector states: epi_model=2, isingle = 1
- Uncoupled with explicit modeling of vector states, SIR/SI: epi_model=3, isingle = 1
- Uncoupled with explicit modeling of vector states, SEIR/SEI: epi_model=4, isingle = 1

- Coupled, SIR with a quasi-equilibrium assumption for the vector states: epi_model=1, isingle = 0
- Coupled, SEIR with a quasi-equilibrium assumption for the vector states: epi_model=2, isingle = 0

At each step in the MCMC procedure we compute the coupled equations for all the regions/-patches and then use equation (33) or (38) to calculate the weekly/monthly infection rate of the region/patch. We then construct the low-resolution incidence rate as the weighted sum of these individual high-resolution incidence rates (with the weight of each region/patch j determined by $N_j / \sum_{j=1}^{D} N_j$). It is important to note that in this procedure the individual high-resolution region/patch incidence rates are *never* optimized directly. We only optimize their weighted sum using the low-resolution region/patch incidence rate as our target. This is unlike the uncoupled procedure where we optimize *each* high-resolution region/patch separately and then use our best results (as well as the posterior density) to provide a best estimate (and a range) for the low-resolution incidence rate. This is a key difference because it means that even in the absence of high resolution incidence data we can still try and improve our prediction for the lower resolution incidence data by using all what we know about the high-resolution regions/patches: location, population, school vacation schedule and specific humidity. How these factors may affect the reproduction number is explained in the next sub-section.

3.5 Time-Dependent Reproduction Number

The seasonality of influenza in temperate climates suggests that the reproduction number R_e is time dependent. When modeling the flu, **DICE** allows the user to apply five different models for the time dependent reproduction number, $R_e(t)$. To implement this, we write the transmission term in the most general way as a product of the basic reproduction number, R_0, multiplied by the various time dependent terms:

$$R_e(t) = R_0 \cdot F_1(t) \cdot F_2(t) \cdot F_3(t) \tag{39}$$

The first time dependent term $F_1(t)$, allows for a dependence of the transmission rate on specific-humidity (SH). In a series of papers [15–17] Shaman et al. and others [1,9] have argued that both transmission and survival of the influenza virus are affected by specific humidity. In temperate regions specific humidity has a seasonal oscillation with a minimum in the winter and a maximum in the summer. We follow Shaman et al. [17] and relate the SH, $q(t)$, to the reproduction number as:

$$F_1(t) = 1 + \Delta_R \cdot \mathrm{e}^{-a \cdot q(t)} \tag{40}$$

In the above equation, and unlike the work published by others, the values of the parameters a and Δ_R are fitted. The SH term $F_1(t)$ is included in models 1 and 3. The details of data collection and processing is discussed later in section 5.

The second time dependent term $F_2(t)$ allows for dependence of the transmission rate on the school vacation schedule. During the 2009-2010 H1N1 pandemic health officials around the world had to consider the potential benefits of reducing transmission by closing schools, against the high economic and social costs of such a drastic measure. In our formulation the transmission rate depends on weekly school closures $p(t)$ as follows:

$$F_2(t) = 1 - \alpha \cdot p(t) \; . \tag{41}$$

The value of $p(t)$ represents school closures in a region/state for the week t. Based on the proportion of schools closed and number of days closed, $p(t)$ is assigned a value in the range $[0, 1]$. For example in week t_i, if all schools are closed for the entire week then $p(t_i) = 1$. However, if all schools have Monday and Tuesday off (missing 2 of 5 days), then $p(t_i) = 0.4$.

Similarly, if 3 of 10 schools have spring break (entire week off), but the other 7 schools have a full week of class then $p(t_i) = 0.3$. And of course if all schools have a full week of class then $p(t_i) = 0$.

DICE fits the effect of school closure by optimizing the parameter α, which is in the range $0-1$. Larger values of α indicate a more significant lowering of R_e as a result of school closures. Conversely small values of α indicate that the school vacation schedule is not an important factor in determining the ILI profile. The school factor is 'turned-on' by selecting $model = 2$ or 3.

The third, and last, time-dependent term, $F_3(t)$, has a simple "box-like" shape and it allows the user to model an arbitrary behavior modification that can drive the transmission rate up or down for a limited period of time:

$$F_3(t) = 1 + \Delta \cdot H(t) \tag{42}$$

where

$$H(t) = \begin{cases} 1 & \text{when } t_s \leq t < t_f \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

Here Δ is the magnitude of change to R_e, t_s is the start time for the change, and t_f is the final time. By allowing the parameter Δ to be in the range $[-1, 1]$, **DICE** can model an increase or decrease in transmission due to behavior modification, multi-strain epidemics, etc. $F_3(t)$ is selected by choosing $model = 5$. This model has been successfully used to describe the complex H1N1 dynamics in military installations during the 2009-2010 pandemic season, see refs. [12,13]

Finally, the user can choose to use a simple S-I-R model with a constant transmission rate ($model = 4$):

$$R_e(t) = R_0 \ . \tag{44}$$

3.6 Summary of Models

As described above, there are a number of ways to define $R_e(t)$. The **DICE** package allows the user to select from five different $R_e(t)$ models. Table 1 contains a brief description of each model and specifies which parameters are being optimized.

Table 1: **DICE** Models

Model #	Description	Optimized Parameters
1	Specific Humidity only	$R_0, \Delta_R, a, t_0, B, pC$
2	School Vacation only	R_0, t_0, B, pC, α
3	School and specific humidity terms	$R_0, \Delta_R, a, t_0, B, pC, \alpha$
4	Constant R_e	R_0, t_0, B, pC
5	Stepped-R_e (see eqs (42) & (43))	$R_0, t_0, B, pC, \Delta, t_s, dur$

Several example $R_e(t)$ profiles using the 2014-2015 SH and SV data are shown in figure 1 (green lines in all four panels). Here, the specific humidity and school vacation schedule from CDC region 5 is used. The figure depicts sample $R_e(t)$ profiles for models 1, 2, 3, and 5. The profile for model 4 is constant ($R_e(t) = R_0$) and therefore has been omitted from the illustration. In these figures, the cyan markers denote scheduled school breaks and the height of the marker indicates the proportion of student-days missed for a given week. A marker height of 0.5 indicates that all schools are out for the entire week and lower values indicate less days/schools on break. Where all schools are in session for the entire week, no marker has been plotted. (This does not take into account sick-days for individual students.) The brown, sin-like jagged line denotes the weekly averaged SH for HHS region 5 during the 2014-2015 season. In the northern hemisphere is is lower in the winter and has a negative correlation with the ILI profile. (This line is the same in all four panels - since they all depict the same HHS

region.) The The top-left illustrates a simple two-value step function $model = 5$. The primary value of $R_e(t)$ is R_0, then for dur weeks starting at t_s, the value of $R_e(t)$ increases(decreases) to $R_0 * (1 + \Delta)$. In the top-right plot, model 1 is considered. Here $R_e(t)$ depends only on the specific humidity. A $R_e(t)$ profile resulting from model 2 (school vacations) appears in the bottom left. The relationship between specific humidity and reproduction number is from eq. (40). Finally, the lower-right plot shows the combined effects of school closures and specific humidity present in model 3.

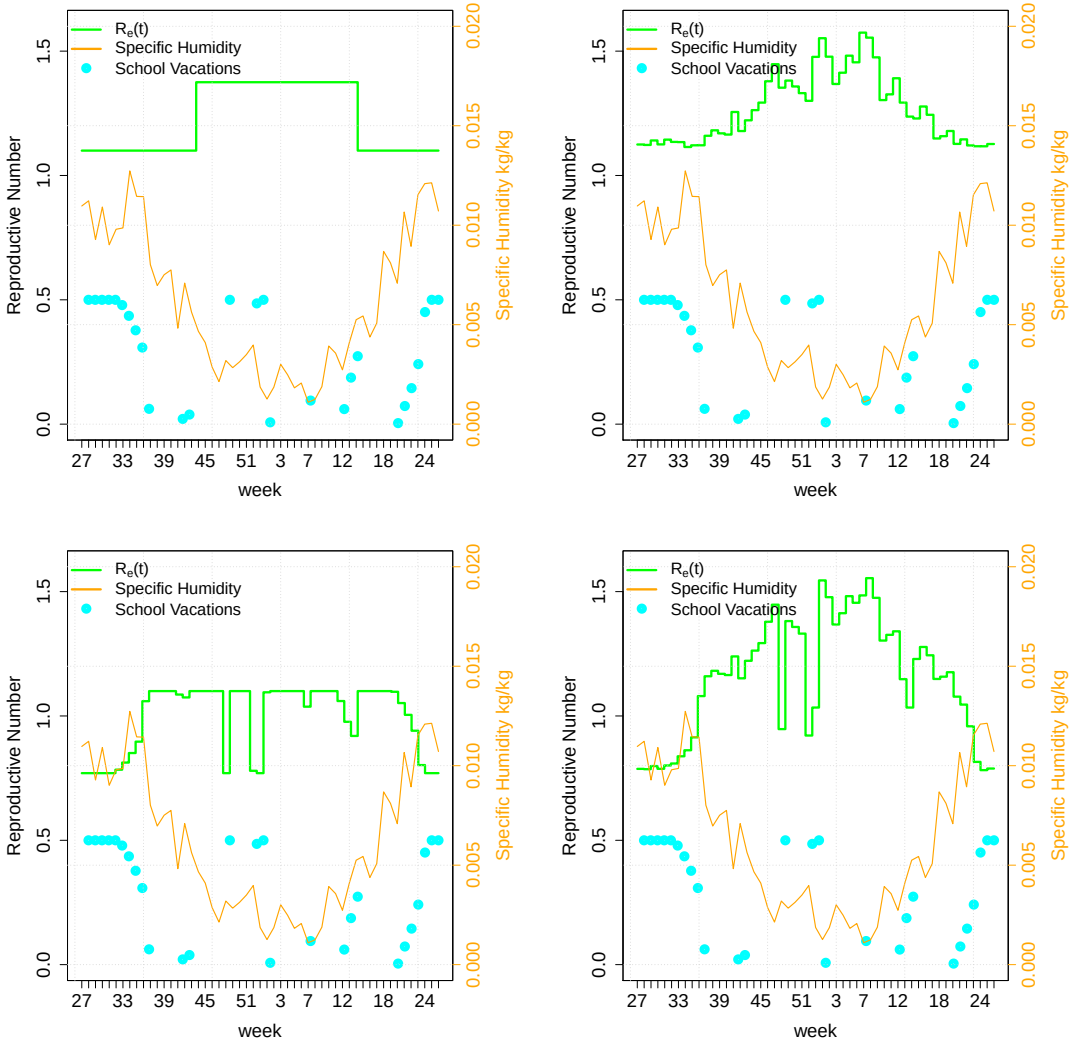

Figure 1: $R_e(t)$ profiles for the model parameters: $R_0 = 1.1, \Delta_R = 0.6, a = 300, \alpha = 0.3, t_s = 18, dur = 23, \Delta = 0.25$. (Top Left) model 5: stepped-$R_e(t)$. (Top Right) model 1: specific humidity only. (Bottom Left) model 2: school only. (Bottom Right) model 3: school and specific humidity terms.

A summary of the parameters recognized by the **DICE** package is found in table 2. The "Optim Range" column is the range of values that the optimizer is allowed to vary over. For the parameters that control $R_e(t)$, we show which model uses which parameters. Both the default and range of values for the baseline parameter B depend on B_{est}. For a single flu-year (ex. Summer 2010–Summer 2011), B_{est} is the average of the first five weeks of ILI data.

Table 2: **DICE** Model Parameters

Parameter	Description	Optim Range
$R_{\text{primary value}}$	Baseline reproduction number (models 1–5)	0.5–3.0
T_g	Average infectious period (default: $T_g = 3$ days)	NA
t_0	Infection start week	0.0–$(nweeks/2)$
$I(t_0)$	Initial number of infected	1–100
B	Baseline: mag. of non-SIR ILI data	$(0.1 * B_{est})$–$(10 * B_{est})$
pC	Percent clinical	10^{-6}–1.0
α	Reduction in R_e for school closure (models 2, 3)	10^{-6}–1.0
Δ_R	Max change in R_e from specific humidity (models 1, 3)	10^{-6}–2.0
a	Exponential for specific humidity R_e-term (models 2, 3)	1–500
Δ	Increase/Decrease in R_e (model 5)	-1.0–1.0
t_s	Start time of change in R_e (model 5)	0–40
dur	time duration: $t_f = t_s + dur$ (model 5)	0.01–40

3.7 Synthetic Examples

To test the quality of both the coupled and uncoupled fitting procedures we provide a script that creates synthetic profiles (either coupled or uncoupled), fits them and compares the input and output values of all the parameters that were fitted. For now, the code has been set up to work for Model number 4 - a fixed value for the force of infection.

The script for the synthetic case is in the 'examples' directory: 'example-synthetic.R'. The script first sets all the parameters that the synthetic example expects and can not be changed. These are the *disease*, mod_level and fit_level which should be set to 'flu', 2 and 3, respectively. The year (i.e. flu season), is basically irrelevant (see below), and is simply set to 2015. Then come the parameters that the user can change, with the two most important ones first: coupled or uncoupled case and the number of MCMC steps (recommended value is 10^5 or more). Other parameters include the plotting method, plotting device and sub-directory name for all the output files of the run. The script loads the **DICE** package and uses all the chosen options to retrieve the 'cdc' dataset for the 2015-2016 season: national and ten HHS regions. Other run parameters are then set and the script calls the fitting routine (either coupled or uncoupled). In the fitting routine the real data for each of the ten HHS regions is replaced by synthetic data, created using either a coupled of uncoupled S-I-R model, with randomly chosen values for all the parameters of model 4. The national synthetic data is computed as a weighted average of the ten HHS regions. **DICE** then proceeds to fit the synthetic data using either the coupled or uncoupled procedure. The output of the synthetic run (graphic files, csv files and binary RData files) is the same as that of a regular run with an additional graphic file showing the input values of the basic reproduction number, R_0, and the percent clinical, pC, for each of the ten HHS regions (in red vertical lines), along with their posterior densities from the MCMC chain. In the case of a coupled run we also show the input and densities of the two parameters that govern the coupling between regions: The saturation distance s_d, and the power γ that determines the amount of mixing between regions (see also Equation 32). This additional graphic file has a unique name (see Section 7 for more details on the naming convention for files) which begins with the **FileType** 'syn-'.

4 Statistical Modeling

4.1 SARIMA Models

Whereas the main objective of **DICE** is to enable mechanistic modeling of (both direct and vector borne) epidemics on an arbitrary spatial scale with or without coupling between regions,

it also supports statistical modeling that is based on Seasonal Autoregressive Integrated Moving Average (SARIMA) models. These techniques are numerically fast but unlike the mechanistic model, described in detail in section 3, it does not give us as much insight into the underlying physics of the disease. For example, it can not tell us anything about the force of infection or the percent clinical. But in some cases it can provide some information about the importance (or lack thereof) of multi-year seasonality. The selection of statistical vs. mechanistic modeling is determined by the key word 'method' which can be set to either 'stat' or 'mech'. When the SARIMA method is selected only a few parameters need to be set: (1) The user can select a specific SARIMA model by defining the 'arima_model' key word which is of type list (see below). (2) If a specific 'arima_model' is selected the user can also select a covariate (key word 'covar') and a (3) lag time for the covariate (key word 'covar_lag').

In general terms, SARIMA models can all be described as:

$$(p, d, q)(P, D, Q)_s \tag{45}$$

where p specifies the number of lags used in the model, d represents the degree of differencing, and q determines the number of error terms to include in the model. The next three parameters specify the "seasonal" aspects of the model, with s being the length of the season (12 months or 52/53 weeks in our case). For example, the selection $(1, 0, 0)(1, 0, 0)_{12}$ corresponds to a model that includes a non-seasonal autoregressive (AR) term of order one, a seasonal AR(1) term, no differencing, no moving average terms and a seasonal period of 12 months. Essentially, this is an AR model with predictors at lags of 1, 12, and 13 months. For each disease (and data source) **DICE** determines the length of the season (based on the temporal resolution of the data) hence the user need only define the $(p, d, q)(P, D, Q)$ values as a list, e.g.

```
> arima_model = list(p = 1, d = 0, q = 0, P = 1, D = 0, Q = 0)
> output <- runDICE(disease = 'dengue', data_source=NULL,
          year=2010, mod_level = 2, fit_level = 3, RegState = 'BR',
          method = 'stat', arima_model = arima_model)
```

If the user does not provide a specific SARIMA model to **DICE** it is set to NULL and the code will use an automated process to select the best SARIMA model for each region (see next section for more details). After selecting a SARIMA model the user can select as a covariate any of the climate data provided with **DICE**: temperature, specific humidity or precipitation (keyword 'covar' set to 'temp', 'sh', or 'precip', respectively). In the case of the CDC ILI data, the user can also select the school vacation schedule as a covariate: 'covar = school'. By default, the covariate variable has a lag time of zero but this can be changed by setting the key word 'covar_lag' to '1' or '2' or more. (The lag time has the same time units as the incidence data, e.g. week, month.)

4.2 SARIMA Training and Forecasting

When a statistical procedure is selected, with a specific SARIMA model, **DICE** first trains the model on (at most) the past ten years of incidence data (up to the latest available data point or the latest requested data point) and then uses the resulting parameters to make a forecast for the next 4 data points. The statistical procedure can only be run in an uncoupled form and the logic of the procedure is identical to that of the mechanistic models. Thus, the low resolution incidence data is first fitted and forecasted directly. **DICE** then fits the high resolution data of each region/patch and these results are used with the proper weights to provide an indirect fit/forecast for the low resolution data. The resulting plots show the fits/forecasts of the high resolution incidence data along with the direct and indirect fits/forecasts of the low resolution data. If the SARIMA model was defined by the user, the same model will be used in all of these fits along with the user selected covariate and covariate lag time. If however 'arima_model' is set to 'NULL' **DICE** will use an automatic procedure to first train and then select the best SARIMA

model for each (both high and low resolution) region. This procedure, which is numerically more demanding, is pre-defined in **DICE** by setting upper limits on all the SARIMA model parameters and it does not allow the use of a covariate.

5 DICE Data

The data incorporated into the **DICE** database includes incidence across many diseases and countries, climate data derived from satellite imagery, population estimates, and school vacation schedules. The following section details how data was collected and processed into the database. Also included is an example of how to sift through the incidence data by location, disease, and/or data source.

5.1 Incidence Data

Incidence data was primarily collected from public-facing urls at country level Health Ministries. A smaller fraction was collected from secondary sources including World Health Organization (WHO), Pan American Health Organization (PAHO), and manuscript supplementary information. For best use with **DICE**, we focused on ingesting data reported monthly or more often than monthly. While there is some annual data in the database, lower frequency datasets have generally been avoided.

A sampling of data sources are shown in table 3. The complete table of data sources can be downloaded at anytime using a few lines of code:

```
> library(DICE)
> myDB = OpenCon()
> data_sources = dbReadTable(conn=myDB, name="data_sources")
```

References in the full table contain a url when available as well as a semicolon delimited list of data columns. For example, the RIO_weekly data source has columns 'cases' and 'pop' indicating that this source contains both number of dengue cases and population for each region/sub-region and period in the dataset. Similarly, the CDC source contains both percent ILI and number of ILI cases. Cadence #'s are explained in table 4 below. The *by-date* entry refers to data where

Table 3: Sample Data Sources

Source	Disease	Source Abbv	Cadence	Col Names
US-Centers for Disease Control	flu	CDC	2	ILI;cases
Brazil Ministry of Health	dengue	BRA_MH	3	cases
Rio Ministry of Health	dengue	RIO_weekly	2	cases;pop
Sri Lanka Ministry of Health	dengue	SRI_monthly	3	cases

each individual case is listed with a date. Currently, **DICE** converts this type of data to daily totals (cadence=1) for fitting.

Table 4: Cadence Key

Cadence #	Cadence Desc
0	by-date
1	daily
2	weekly
3	monthly
4	yearly

The **DICE** database has too many data sources to include full descriptions here, but a few select sources are detailed below.

5.1.1 CDC ILINet

The CDC Influenza-like Illness surveillance Network (ILINet) Human and Health Services (HHS) region and national data were downloaded from the CDC-hosted web application FluView http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html. Figure 2 shows which states are grouped into each HHS region. The circle in each region denotes the population density weighted location of the centroid of the region. The radius of each circle is proportional to the weight of the region which is determined by it's relative population (shown next to the centroid).

DICE is capable of downloading CDC ILI data using the R-package **cdcfluview**. This ensures that **DICE** is using the most up-to-date CDC data available. If the data is temporarily unavailable, or cannot be accessed, **DICE** will use the data it has for the current season and notify the user that internal data, which may not be up-to-date, is used. See the *CDC_server* flag in the 'Data Selection Variables' below.

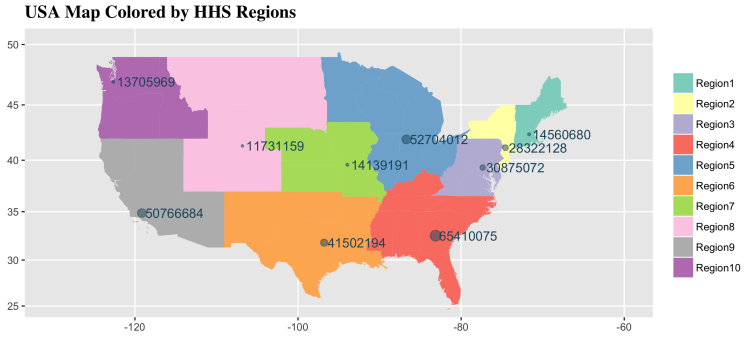

Figure 2: HHS Regions labeled with population.

The CDC Patient ILINet consists of more than 2,900 outpatient healthcare providers in all 50 states, Puerto Rico, the District of Columbia and the U.S. Virgin Islands reporting more than 30 million patient visits each year. Each week, approximately 2,000 outpatient healthcare providers around the country report data to CDC on the total number of patients seen and the number of those patients with influenza-like illness (ILI) by age group (0-4 years, 5-24 years, 25-49 years, 50-64 years, and $\geq$ 65 years). For this system, ILI is defined as fever (temperature of 100F or greater) and a cough and/or a sore throat without a *known* cause other than influenza. Sites with electronic health records use an equivalent definition as determined by public health authorities. The CDC Influenza surveillance data collection is based on a reporting week that starts on Sunday and ends on Saturday of each week. With the exception of the 2009 influenza pandemic, each flu season starts on Week 27 and ends on Week 26 of the following year. For 2009, the flu season started Week 13, 2009 (3-30-2009) and ended Week 26, 2010 (6-28-2010). For more information on ILINet see: http://www.cdc.gov/flu/weekly/overview.htm.

Because **DICE** requires an absolute number of incidents per week, CDC ILINet data is

converted from percent ILI to approximate total ILI incidents. We estimate the absolute number of weekly ILI cases by dividing the weighted percent of ILI cases in the CDC data ("X.WEIGHTED.ILI" column) by 100 and multiplying it by the total weekly number of patients. Total weekly patients is estimated as: (total population)*(2 outpatient visits per person per year)*(1 year/52 weeks).

We arrive at the estimate of 2 outpatient visits per year using two surveys published by the CDC: National Ambulatory Medical Care Surveys (NAMCS) [18] and National Hospital Ambulatory Medical Care Surveys (NHAMCS) [19]. In the 2011 issues, the surveys estimated ambulatory visit rates of 3.32 visits per capita per year for Physician's offices, .43 for hospital emergency departments and .33 hospital Outpatient Departments. We can sum these rates to get an outpatient visit per capita-year of 4.08. The rates for 2008 are 3.20, .41, and .37 respectively, which sums to 3.98 outpatient-visits per year. In 2006 Schappert and Burt did a secondary study [14] of the 2002 NAMCS and NHAMCS reports and calculated an ambulatory rate of 3.8 visits per capita-year. We further estimate from the surveys that approximately one half of outpatient visits are at a medical practice who's specialty is ineligible for ILINet participation. Thus we arrive at the estimation (4 outpatient visits per capita-year)*(1 ILI-reportable visit/2 visits) = 2 ILI-reportable visits per capita-year. By incorporating this best-estimate, we hope to improve the accuracy of p_C posteriors.

5.1.2 Google Flu Trends

Google Flu Trends data (at the national, state and city level) was obtained directly from the GFT web site: http://www.google.org/flutrends/us/data.txt. GFT attempts to quantify ILI cases through a proprietary model using key terms from search queries (originally discussed in [6]; see also [3, 4]). GFT ILI estimates were publicly accessible from Week 39, September 28th, 2003 to Week 32, August 9th, 2015. The GFT data is aggregated to the same ten HHS regions as for the CDC data and the flu seasons were also defined using the same dates. In addition, some of the GFT data is available at state levels. To create a state-level dataset that spans all available years, the GFT data in **DICE** consists of two different GFT model updates. The 2003-2013 flu seasons are covered by GFT data from the 2013 model-update. Then the 2014-2015 season is covered by GFT data from the 2014 model-update.

5.2 Climate Data

In every geographic region for which the **DICE** database has incidence data, climate data is automatically pre-calculated. Generally this consists of temperature, precipitation, and specific humidity processed from data generated by the National Oceanic and Atmospheric Administration's (NOAA) National Centers for Environmental Prediction (NCEP)/National Center for Atmospheric Research (NCAR) Reanalysis-2 project. This data is processed to a single population-weighted value for each region at a cadence of daily/weekly/monthly to match the cadence of incidence reporting.

Additionally, specific humidity is tracked for the contiguous United States using the National Atmosphere and Space Agency's (NASA) North American Land Data Assimilation System (NLDAS)-2 dataset. This is legacy code that does not incorporate population-weighting.

5.2.1 NOAA Climate Data

NOAA's NCEP/NCAR Reanalysis-2 project [7] (see for example: http://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCEP-NCAR/.CDAS-1/.DAILY/.Diagnostic/.above_ground/.qa/) provides climate data 4-times daily, on a spatial grid of 1.875° for the entire world from 1979-present (2 week to 2 month lag). The original grid is first averaged to daily values. Then the Database of Global Administrative Areas (GADM) is used to define boundaries while Socioeconomic Data and Applications Center (SEDAC) maps are

used to approximate population distributions. By overlaying GADM boundaries on both the Reanalysis-2 and SEDAC grids, **DICE** is able to calculate a population-weighted daily climate time-series for each region. Currently, **DICE** expects climate time-series data on the same cadence as incidence data, so daily climate data has been averaged to weekly/monthly/yearly wherever weekly/monthly/yearly incidence exists.

5.2.2 NASA Specific Humidity

Our secondary source of specific humidity is calculated from the NLDAS-2 database published by NASA http://ldas.gsfc.nasa.gov/nldas/NLDAS2model.php/ [11, 20]. The NLDAS-2 database provides hourly specific humidity (measured 2-meters above the ground) for the continental US at a spatial grid of 0.125° which we average to daily and weekly SH. The weekly data is then spatially-averaged for the states and CDC regions. Because they are not covered by the NLDAS grid, Alaska and Hawaii always use specific humidity from the NOAA Reanalysis dataset.

5.3 School Vacation Schedules

Currently school vacation schedules have been collected only for the United States. For each state, school vacation schedules were approximated by averaging the public school vacation schedule from the three largest cities in that state for the 2014-15, 2015-16, and 2017-2018 school years. Approximations for region schedules are determined by a population-weighted average of state schedules. The same process is then applied to the regions to recover a national school schedule. The value of $p(t)$ represents school closures in a state/region for the week t. Based on the proportion of schools closed and number of days closed, $p(t)$ is assigned a value in the range $[0, 1]$. For example in week t_i, if all schools are closed for the entire week then $p(t_i) = 1$. However, if all schools have Monday and Tuesday off (missing 2 of 5 days), then $p(t_i) = 0.4$. Similarly, if 3 of 10 schools have spring break (entire week off), but the other 7 schools have a full week of class then $p(t_i) = 0.3$. And of course if all schools have a full week of class then $p(t_i) = 0$.

5.4 Population Data

Population data has been collected from census bureaus, incidence data, and GADM/SEDAC overlay. In this context 'incidence data' refers to incidence reports that include a population estimate as an auxiliary value to the number of cases. For example the Rio de Janeiro Health Department dengue reports divide the municipality into 200 regions/sub-regions, each of which contains a yearly population estimate (ex. http://www.rio.rj.gov.br/web/sms/exibeconteudo?id=2815389). The third type of population data is computed by overlaying GADM administrative boundaries on the SEDAC population density map. In this way, we can produce a population estimate for any region of the world that is defined in GADM or can be represented as a collection of sub-regions that are defined in GADM. When retrieving population data for **DICE**, sources are prioritized in the following order: (1) auxiliary incidence data (2) census data (3) GADM/SEDAC.

DICE Data Structure

The **DICE** dataset is structured to organize global disease incidence data on a multitude of spatial scales. An important aspect of the data structure organization is the idea of spatial 'levels'. The levels for the United States are defined as seen in table 5 . Levels 3 and up are likely to be labeled differently in each country. For example, the levels in Australia are labeled as: 3-State/Territory, 4-Health Region, 5-Local Governing Unit, 6-City.

Table 5: Population levels

Level	Description
0	Global
1	Continent
2	Country
3	HHS Region
4	State
5	State Health Dist
6	County
7	City

A common application of **DICE** is to use the characteristics of sub-population profiles to build an epidemic forecast for the population of interest. For example one might analyze the 50 states (coupled or uncoupled) to recover a more accurate national forecast. For this purpose we have defined *fit level* as the population level that is being fit and *model level* as the level to be modeled/forecast. The model-level population may be a previously defined country/region or the user may select sub-regions (fit level) to build a super-region (model level) from. See the description of *fit_name* below for more information about creating a new region.

Data Selection Variables

The function *get.DICE.data*() provides easy access to all data in the **DICE** database. Whether the data is accessed directly through *get.DICE.data*() or simply by use of *runDICE*(), the list below provides useful descriptions of the input variables.

data_source - A string or integer specifying the data source. See section 5.1 about accessing the *source_abbv* column of the data_sources table. Additionally, the **DICE** functions *SummaryByCountry*() and *SummaryBySource*() are quite useful for determining source abbreviation.

mod_level - An integer value indicating the population-level of the incidence profile to model/forecast. Table 5 shows the population levels for the United States. Levels 3+ may vary by country. For example the levels in Australia could be defined as: 3-State/Territory, 4-Health District, 5-Local Government Area, 6-City.

mod_name - A named vector of strings specifying the model-level spatial patch. If `is.null(mod_name)`, the code reverts to using *RegState* (see next entry). To specify New York state, set `mod_name=c(NAME_2="United States", NAME_3="R1", NAME_4="New York")`. Here *NAME_X* is either the full name or abbreviation of the level-X patch. Replacing 'United States' with 'US' or 'R1' with 'Region 1' would result in the same outcome. Also, vector entries for *mod_name* should go from *NAME_2*,..., *NAME_n* where *mod_level*=n.

RegState - Determines which single region from *mod_level* is to be modeled. This is a legacy input that is limited to countries and United States regions/states. *mod_name* is the more robust input. Depending on the model level, *RegState* should adhere to the following format: *mod_level* = 2 - 3-letter ISO3 country code, *mod_level*=3 - an integer describing the HHS region, *mod_level*=4 - a 2-letter state code.

fit_level - An integer value indicating the population-level of the incidence profile(s) to be fit. $fit\_level \geq mod\_level$. See table 5. It is possible to run **DICE** for a single population by choosing $mod\_level = fit\_level$. This indicates to **DICE** that the *model* region is to be fit directly with no higher-level indirect fits.

fit_names - If *fit_names*=="all", then all regions at *fit_level* that are children of *mod_name* will be used. Otherwise, *fit_names* should be a vector of strings specifying the names or abbreviations to construct a region from. For example, if we wanted to construct a Pacific Coast Region for the contiguous United States we would use the inputs: *mod_level*=2, *mod_name*="US", *fit_level*=4, and *fit_names*=`c("CA", "OR", "WA")`.

year - Integer specifying the season year. Default season start and end dates are stored in the *season_se_dates* table in the database.

disease - String describing the which disease. See section 5.1 about accessing the *disease* column of the data_sources table.

db_opts - A list of database options. The *DICE_db* option determines which SQL database to retrieve data from 'PredSci' or 'BSVE'. The *CDC_server* option determines if incidence data should be retrieved directly from CDC servers. *CDC_server* should be set to TRUE only when *data_source*= 'cdc' or 7.

raw_col - String describing which data stream to use. When mulitple incidence metrics are present for a given location/disease/season/data source, *raw_col* allows the user to select which one to model. If no *raw_col* is specified, the code defaults to the first one as listed in the *col_names* column of the *data_sources* table.

Data Lookup

To get a snapshot of the incidence data currently available in the **DICE** database, use the functions SummaryByCountry() and SummaryBySource(). Following the example below, it becomes straightforward to determine which dataset you would like to use and fill the *disease* and *data_source* inputs. SummaryByCountry() is focused on summarizing the spatial levels of data by country, while SummaryBySource() breaks-down each country/disease pair by the data sources. The data structure returned by SummaryByCountry() is ordered by disease/-country/cadence. The output from SummaryBySource() is ordered as disease/country/root-location/data_source.

The complete list of data sources (when this document was compiled) is seen in Table 6 below.

Table 6: Complete list of DICE database data-sources as of 2019-06-14.

source_key	source_abbv	disease	source_desc
1	BRA_MH	dengue	Brazil national epidemiological data system (
2	RIO_monthly	dengue	Municipality of Rio de Janeiro Health departm
3	RIO_weekly	dengue	Municipality of Rio de Janeiro Health departm
4	Sri_monthly	dengue	Sri Lanka Ministry of Health: Epidemiology Un
5	Sri_weekly	dengue	Sri Lanka Ministry of Health: Epidemiology Un
6	MEX_portal	dengue	Mexico Federal Government portal
7	CDC	flu	United States Centers for Disease Control ILI
8	TYCHO	dengue	TYCHO-University of Pittsburgh
9	THA_MH	dengue	Thailand Ministry of Health - Weekly Epidemio
10	USA_SD	flu	San Diego County - Influenza Watch
11	USA_TN	flu	Tennessee Department of Health - Sentinel Sur
12	SGP_MH	dengue	Singapore Ministry of Health - Epidemiologica
13	SAU_MERS	mers_cov	Saudi Arabia MERS reporting
14	SARS_2003	sars	Country-level SARS incidence
15	WHO_flu	flu	WHO influenza incidence at country level and
16	SOM_MH	cholera	Somalia Ministry of Health
17	ZMB_MH	cholera	Zambia Ministry of Health
18	WHO_chlr	cholera	World Health Organization - annual cholera
19	BRA_yf	yellow_fever	Brazil Ministry of Health yellow fever
20	TWN_den	dengue	Taiwan CDC
21	CA_meas	measles	California measles reporting
22	FSM_meas	measles	Micronesia 2014 measles outbreak
23	JAM_chik	chik	Jamaica Data Portal
24	MDG_plag	plague	Ministry of Health of Madagascar
25	And_Lab	zika	Anderson Lab Digitization of PAHO records
27	WHO_ebo	ebola	WHO weekly ebola
28	COD_ebol	ebola	A Rosello - eLife 2015, Ebola linelist
29	CDC_lyme	lyme	CDC Monthly Average National Cases
30	CDC_deng	dengue	CDC Monthly Dengue reporting for American Sam
31	siraj_zika	zika	Siraj et al, Zika in Colombia, hosted on Drya
32	idn_dengue	dengue	"Prediction of Dengue Outbreaks based on Dise
33	isr_meas	measles	Israel Ministry of Health
34	nga_lassa	lassa	Nigeria Centre for Disease Control
35	GFT	flu	Google Flu Trends

Example 1: SummaryByCountry(), looking-up data for dengue in Brazil

```
> library(DICE)
> # Retrieve summary of available data
> test = SummaryByCountry()
> # This is a list by disease
> str(test,1)

List of 13
 $ dengue      :List of 13
 $ flu         :List of 139
 $ mers_cov    :List of 1
 $ sars        :List of 5
 $ cholera     :List of 160
 $ yellow_fever:List of 1
 $ measles     :List of 3
 $ chik        :List of 1
 $ plague      :List of 1
 $ zika        :List of 42
 $ ebola       :List of 4
 $ lyme        :List of 1
 $ lassa       :List of 1

> # The dengue list entry is a nested list by country
> str(test$dengue,1)

List of 13
 $ american samoa:List of 3
 $ brazil        :List of 4
 $ cambodia      :List of 3
 $ indonesia     :List of 3
 $ laos          :List of 3
 $ malaysia      :List of 3
 $ mexico        :List of 3
 $ philippines   :List of 3
 $ singapore     :List of 3
 $ sri lanka     :List of 4
 $ taiwan        :List of 3
 $ thailand      :List of 3
 $ vietnam       :List of 3

> # The dengue/brazil entry has four entries
> str(test$dengue$brazil,1)

List of 4
 $ tot_regions   : int 5702
 $ RegionsByLevel:'data.frame':        7 obs. of  2 variables:
 $ Weekly        :'data.frame':        4 obs. of  4 variables:
 $ Monthly       :'data.frame':        7 obs. of  4 variables:

> # The total number of regions for dengue/brazil
> test$dengue$brazil$tot_regions

[1] 5702
```

```
> # Show how the regions break-out by spatial level
> test$dengue$brazil$RegionsByLevel

  level num_regions
1     2           1
2     3           5
3     4          27
4     5        5467
5     6          11
6     7          31
7     8         160

> # A summary of data with weekly cadence
> test$dengue$brazil$Weekly

  level num_regions   min_date   max_date
1     5           1 2000-01-02 2018-09-02
2     6          11 2000-01-02 2018-09-02
3     7          31 2000-01-02 2018-09-02
4     8         160 2000-01-02 2018-09-02

> # A summary of data with monthly cadence
> test$dengue$brazil$Monthly

  level num_regions   min_date   max_date
1     2           1 2001-01-01 2012-12-01
2     3           5 2001-01-01 2012-12-01
3     4          27 2001-01-01 2012-12-01
4     5        5467 2001-01-01 2018-09-01
5     6          11 2011-01-01 2018-09-01
6     7          31 2011-01-01 2018-09-01
7     8         160 2011-01-01 2018-09-01
```

Example 2: SummaryBySource(), looking-up data for dengue in Thailand

```
> library(DICE)
> # recover the data summary
> test = SummaryBySource()
> # This is a list by disease
> str(test,1)

List of 13
 $ dengue      :List of 13
 $ flu         :List of 139
 $ mers_cov    :List of 1
 $ sars        :List of 5
 $ cholera     :List of 160
 $ yellow_fever:List of 1
 $ measles     :List of 3
 $ chik        :List of 1
 $ plague      :List of 1
 $ zika        :List of 42
 $ ebola       :List of 4
 $ lyme        :List of 1
 $ lassa       :List of 1
```

```
> # The dengue list entry is a nested list by country
> str(test$dengue,1)

List of 13
 $ American Samoa:List of 1
 $ Brazil        :List of 2
 $ Cambodia      :List of 1
 $ Indonesia     :List of 1
 $ Laos          :List of 1
 $ Malaysia      :List of 1
 $ Mexico        :List of 1
 $ Philippines   :List of 1
 $ Singapore     :List of 1
 $ Sri Lanka     :List of 1
 $ Taiwan        :List of 1
 $ Thailand      :List of 1
 $ Vietnam       :List of 1

> # The dengue/Thailand entry has only one root location
> str(test$dengue$Thailand,1)

List of 1
 $ Thailand:List of 1

> # This root location has only one data source (abbreviated 'THA_MH',
> # Thailand Ministry of Health)
> str(test$dengue$Thailand$Thailand,1)

List of 1
 $ THA_MH:List of 2

> # Show the data source info
> test$dengue$Thailand$Thailand$THA_MH$source_info

  source_key cadence disease                                        source
9          9       3  dengue http://www.wesr.moph.go.th/wesr_new/index.php
  source_abbv
9      THA_MH
                                                          source_desc
9 Thailand Ministry of Health - Weekly Epidemiological Surveillance Report
  data_cols col_names                 col_units bsve_use countries max_lev
9         2 cases;pop reported cases;population        1        TH       5
  min_lev
9       2

> # A summary of data by spatial level
> test$dengue$Thailand$Thailand$THA_MH$data_summary

  levels num_regions   min_date   max_date
1      2           1 2007-01-01 2018-12-01
2      3           4 2007-01-01 2018-12-01
3      4          13 2007-01-01 2018-12-01
4      5          77 2007-01-01 2018-12-01
```

From looking through the output of SummaryBySource() we see that we want to set the input variables disease='dengue' and data_source='THA_MH'. Now if we do not already have a region to model in mind, we can search through the regions by first grabbing the dengue lookup table. Each disease has a Look-Up Table (LUT) that defines all available regions as well as thier spatial hierarchy. The LUTs are named 'disease'_lut.

```
> # Open a connection to the DICE database
> myDB = OpenCon()
> # Downloand the Dengue look-up table
> dengue_lut = dbReadTable(conn=myDB, name="dengue_lut")
> str(dengue_lut)

'data.frame':        6557 obs. of  38 variables:
 $ identifier          : chr  "OC.AS" "SA.BR" "SA.BR.R1" "SA.BR.R1.RO" ...
 $ level               : num  2 2 3 4 4 4 4 4 4 4 ...
 $ NAME_1              : chr  "Oceania" "South America" "South America" "South America" ...
 $ NAME_2              : chr  "American Samoa" "Brazil" "Brazil" "Brazil" ...
 $ NAME_3              : chr  NA NA "RegiÃčo Norte" "RegiÃčo Norte" ...
 $ NAME_4              : chr  NA NA NA "RondÃt'nia" ...
 $ NAME_5              : chr  NA NA NA NA ...
 $ NAME_6              : chr  NA NA NA NA ...
 $ NAME_7              : chr  NA NA NA NA ...
 $ NAME_8              : chr  NA NA NA NA ...
 $ ID_1                : num  5 6 6 6 6 6 6 6 6 6 ...
 $ ID_2                : num  6 33 33 33 33 33 33 33 33 33 ...
 $ ID_3                : num  NA NA 1 1 1 1 1 1 1 1 ...
 $ ID_4                : num  NA NA NA 22 1 4 23 14 3 27 ...
 $ ID_5                : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ID_6                : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ID_7                : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ID_8                : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ABBV_1              : chr  "OC" "SA" "SA" "SA" ...
 $ ABBV_2              : chr  "AS" "BR" "BR" "BR" ...
 $ ABBV_3              : chr  NA NA "R1" "R1" ...
 $ ABBV_4              : chr  NA NA NA "RO" ...
 $ ABBV_5              : chr  NA NA NA NA ...
 $ ABBV_6              : chr  NA NA NA NA ...
 $ ABBV_7              : chr  NA NA NA NA ...
 $ ABBV_8              : chr  NA NA NA NA ...
 $ inc_key             : num  0 0 1 2 3 4 5 6 7 8 ...
 $ master_key          : chr  "00600000" "03300000" "03300001" "03300002" ...
 $ gadm_name           : chr  "American Samoa" "Brazil" NA "RondÃt'nia" ...
 $ gadm_lvl            : num  0 0 NA 1 1 1 1 1 1 1 ...
 $ clim_ident          : chr  "OC.AS" "SA.BR" "SA.BR.R1" "SA.BR.R1.RO" ...
 $ gadm_noaa_sedac_ident: chr  "AS" "BR" NA "BR.RO" ...
 $ gadm_lat            : num  -14.3 -10.74 -4.59 -10.9 -9.31 ...
 $ gadm_lon            : num  -170.7 -53.1 -59.2 -62.9 -70.4 ...
 $ gadm_area           : num  223 8500365 3848103 236376 152733 ...
 $ sedac_lat           : num  -14.31 -16.92 -4.19 -10.42 -9.53 ...
 $ sedac_lon           : num  -170.7 -45.8 -54.6 -62.8 -69 ...
 $ sedac_pop           : num  5.15e+04 2.02e+08 1.78e+07 1.67e+06 8.08e+05 ...

> # close connection
> dbDisconnect(myDB)
```

```
[1] TRUE
```

Each region is assigned a(n) spatial level (level), name (NAME_'level'), abbreviation (ABBV_'level'), and number (ID_'level'). Each row defines a single region and contains the parent NAME(s), ABBV(s), and ID(s). Say we want to look at health zones (level 4) in Thailand. A list of level 4 names can be returned by:

```
> # First see a list of countries
> unique(dengue_lut$NAME_2)

 [1] "American Samoa" "Brazil"         "Indonesia"      "Cambodia"
 [5] "Laos"           "Sri Lanka"      "Mexico"         "Malaysia"
 [9] "Philippines"    "Singapore"      "Thailand"       "Taiwan"
[13] "Vietnam"

> # Next generate a list of levels available in Thailand
> unique(dengue_lut$level[dengue_lut$NAME_2=="Thailand"])

[1] 2 3 4 5

> # Now generate a list of level 4 regions in Thailand
> dengue_lut$NAME_4[dengue_lut$level==4 & dengue_lut$NAME_2=="Thailand"]

 [1] "Zone 1"       "Zone 2"       "Zone 3"       "Zone Bangkok" "Zone 4"
 [6] "Zone 5"       "Zone 6"       "Zone 7"       "Zone 8"       "Zone 9"
[11] "Zone 10"      "Zone 11"      "Zone 12"
```

Lets say we want to model Zone 5 using all of its provinces. If we want a list of provinces in Zone 5

```
> dengue_lut$NAME_5[dengue_lut$NAME_2=="Thailand" & dengue_lut$NAME_4=="Zone 5"
+     & dengue_lut$level==5]

[1] "Ratchaburi"          "Kanchanaburi"        "Suphan Buri"
[4] "Nakhon Pathom"       "Samut Sakhon"        "Samut Songkhram"
[7] "Phetchaburi"         "Prachuap Khiri Khan"
```

The next step is to inspect the lookup table row corresponding to Zone 5 so we know how to fill the *mod_name* input.

```
> dengue_lut[dengue_lut$NAME_2=="Thailand" & dengue_lut$NAME_4=="Zone 5" &
+     dengue_lut$level==4, ]

      identifier level NAME_1   NAME_2         NAME_3 NAME_4 NAME_5 NAME_6
5989 AS.TH.R2.Z5     4   Asia Thailand Central Region Zone 5   <NA>   <NA>
     NAME_7 NAME_8 ID_1 ID_2 ID_3 ID_4 ID_5 ID_6 ID_7 ID_8 ABBV_1 ABBV_2 ABBV_3
5989   <NA>   <NA>    3  228    2    5   NA   NA   NA   NA     AS     TH     R2
     ABBV_4 ABBV_5 ABBV_6 ABBV_7 ABBV_8 inc_key master_key gadm_name gadm_lvl
5989     Z5   <NA>   <NA>   <NA>   <NA>      55   22800055      <NA>       NA
      clim_ident gadm_noaa_sedac_ident gadm_lat gadm_lon gadm_area sedac_lat
5989 AS.TH.R2.Z5                  <NA> 13.81896 99.44633  45946.85   13.6808
     sedac_lon sedac_pop
5989  99.90137   5817362
```

The *mod_name* input requires names or abbreviations for the model region and its parent regions up to country level (level 2). From the above LUT information we can now define *mod_name*=`c(NAME_2="Thailand", NAME_3="Central Region", NAME_4="Zone 5")`, *mod_level*=4, and *fit_names*=`"all"`.

Using the information gleaned above we can now retrieve dengue data for the 2015 season in Zone 5 of Thailand:

```
> test_data = get.DICE.data(data_source = "THA_MH", mod_level = 4,
+     mod_name=c(NAME_2="Thailand", NAME_3="Central Region", NAME_4="Zone 5"),
+     fit_names="all", fit_level = 5, year = 2015, db_opts=
+     list(DICE_db="predsci", CDC_server=FALSE), disease="dengue")

Downloading incidence and climate data......Complete
Downloading climate data at appropriate cadence.
Formatting data......Complete

> # Inspect the output
> str(test_data,1)

List of 4
 $ mydata          :List of 20
 $ all_years_epi   :List of 5
 $ all_season_dates:'data.frame':        46 obs. of  7 variables:
 $ all_years_clim  :List of 5
```

Here, the 'mydata' entry is the primary data structure needed by DICE to perform the fitting and forecasting routines. 'all_years_epi', 'all_season_dates', and 'all_years_clim' are auxiliary data structures: all_years_epi - provides all historic incidence for the region(s) of interest, all_years_clim - provides all historic (1979-present) climate factors, all_season_dates - record of season start/end date definitions.

Week Indexing

To be consistent with week numbering and dates, we have adopted the CDC/MMWR week numbering system. This means that a week begins on Sunday and ends on Saturday. For the purpose of numbering weeks of a year this dictates that week number 1 is the week containing the first Wednesday of the year. Any January days occurring before week 1 are considered part of the final week (52 or 53) of the previous year (https://wwwn.cdc.gov/nndss/document/MMWR_week_overview.pdf).

6 Mechanistic Models: Fitting Procedure

The **DICE** incidence fitting procedure determines the joint posterior distribution for the model parameters using a Metropolis-Hastings Markov Chain Monte Carlo (MCMC) procedure [5]. It should be stressed that although we often refer to the MCMC algorithm in the context of an optimizer, it is better characterized as a probability distribution mapping routine. As such, the random walk will likely spend the majority of its time in the neighborhood of the optimal (most likely) solution. However it's purpose is to find a distribution of most-likely solutions, not necessarily *the* best solution.

After the user selects a compartmental model, and a model number for $R_e(t)$ (which triggers **DICE** to set TRUE/FALSE values for parameters that need/not be optimized), the user needs to tell **DICE** how many MCMC chains to run and how many steps to take in each chain. **DICE** will then randomly initialize the parameters that are optimized (using a log-uniform distribution

for all the parameters except the one that can be negative) integrate the coupled equations (S-I-R, S-E-I-R etc.) and incidence equations and generate a candidate incidence profile. The likelihood of this solution is calculated using the Akaike Information Criterion (AIC), which is a measure of the relative goodness of fit of a model:

$$\mathrm{AIC} = -2\log(L(\hat{\theta}|I_C)) + 2K \tag{46}$$

where $\log(L(\hat{\theta}|I_C))$ is the value of the maximized log-likelihood over the model parameters (θ), given the observed cases I_C. When the total number of parameters (K) is large relative to the sample size (n), the reduced Akaike Information Criterion is preferred:

$$\mathrm{AIC}_c = -2\log(L(\hat{\theta})) + 2K + \frac{2K(K+1)}{(n-K-1)} \tag{47}$$

and this is what **DICE** uses. The log-likelihood stems from a Poisson probability density

$$\log(L(\theta|I_C)) = \sum_i w_i \Big(I_C(t_i) \cdot \log\big(I_R(t_i,\theta)\big) - I_R(t_i,\theta) - \log\big(I_C(t_i)!\big) \Big) , \tag{48}$$

where $I_R(t_i,\theta)$ is the model point generated for week t_i as a result of parameter set θ. Additionally, the vector w has been added to allow the user to vary the weight given to each data point. To maximize $\log(L(\theta|I_C))$, the value of this likelihood is compared to a new likelihood calculated using a set of randomly displaced parameters in a standard rejection method to determine if the move is accepted or rejected. This MCMC procedure is executed as many times as the user has defined (this is the chain's length mentioned above) and **DICE** keeps the history of the chain parameters and AIC_c values. Once a chain is completed, it's history statistics and results are summarized and written to tables (csv format), a binary '`RData`' file and pdf/png plots. The detailed output provided by **DICE** enables the user to:

- Quickly look at the plots and evaluate the procedure/results.
- Generate detailed reports using the plots and tables prepared by **DICE**.
- Use the history of the MCMC chain to calculate any additional statistics/properties.

6.1 Informative Prior

In the previous section we described a traditional MCMC procedure which uses an uninformed prior (UP): a log uniform distribution for the parameters. Early in the flu season, before the ILI curve takes off, this fitting can result in peak intensities that are significantly larger/lower than expected (based on historic values) and/or peak weeks that are inconsistent with past values. One way to constrain the predictions, is to use an informed prior (IP). We have used each of the models supported by DICE to fit all previous seasons (starting from 2004) at both the national and regional levels. Using the history of the MCMC chain we then built a posterior distribution for each parameter and fitted it to a 1-D Gaussian. (This assumes that the model parameters are independent.) By repeating this procedure for each season and each model, we create a database of prior knowledge which can be used to inform the MCMC procedure. To allow for an informed prior that is less restrictive, we also allow the use of a heated log-likelihood where the temperature can be increased by a factor of five or ten. The informed prior option can *only* be used with the 'CDC' data type ('mod_level = 2, or 3' and 'fit_level = 2 or 3') and the uncoupled procedure. By default, **DICE** sets the 'prior' keyword to 0 uninformed prior. To use the informed (or heated informed, second example below) prior options, when calling the 'runDICE' function, use:

```
> prior = 1
> Temp = 1
```

or:

```
> prior = 1
> Temp = 10.
```

See also Table 7 below.

6.2 Data Augmentation

Another way to make maximum use of prior data within a mechanistic framework, which is implemented in **DICE**, is to use data augmentation (key word 'da'). Given a limited number of weekly (or monthly) data points for a season, we augment the available data using (y-shifted) average historic data ($da = 1$), or using the season that is most similar to the current season ($da = 2$). The augmented data is y-shifted so that it matches the last data point for the current season and it is given a lower weight in the MCMC procedure, determined by the value of the Pearson correlation between the current season and the data chosen for augmentation. The augmented data procedure can used for both the coupled and uncoupled fits and also using a heated augmented procedure (where the log-likelihood is again heated by a factor of five or ten). To use the data augmentation option with historic average data use:

```
> da = 1
```

or for the most similar season data:

```
> da = 2
```

when calling the 'runDICE' function. Table 7 lists all the prior, data augmentation and heating options in **DICE** [1], and more information about the use of informed priors can be found in our most recent work [2].

Table 7: Prior Options

Prior	keywords	Name	Description
0	prior=0	Uninformed Prior	log-uniform
1	prior=1	Informed Prior	Gaussian posterior
2	prior=1, Temp=10	Informed Prior	Heated Gaussian posterior
3	da = 1, Temp=1	Data Augmentation	Augment w/ historic/similar season
3	da = 2, Temp=10	Heated Data Augmentation	same as above & heated log likelihood

Forecasting Mode

DICE can be set to only fit the first $nfit$ data points of an incidence season. When $nfit$ is less than the number of weeks/months in the season, the resulting plot will show the profile fit as well as the 'predicted' incidence values for the remainder of the season. In this way **DICE** can be used to create an incidence forecast. For historic years, the actual incidence is plotted alongside the prediction curves allowing the user to gauge the accuracy of the forecast. The lower limit for $nfit$ is 10 for weekly data and 4 for monthly. If the user sets it to less than that, **DICE** will reset it to ten or $four$ and print a warning message to the screen.

[1] Only available for data type 'CDC'. prior=1 is only available uncoupled procedure.

Numeric Parameters

nreal - number of MCMC chains/realizations (integer). Generally, each realization is an MCMC optimization-chain with a unique set of initial parameter values. The default behavior for **DICE** is to randomly select initial values for the parameters-to-be-optimized using a uniform distribution over the range of values listed in table 2. For a quick run, set *nreal* to 1. For many data/model combinations, setting *nreal* to 5 will be enough to find the global minimum. Depending on the number of parameters being optimized, the prevalence of 'deep' local minima in the optimization-objective function, and the number of MCMC steps; a larger value of *nreal* may be required to find the global minimum without an excessively long chain.

nMCMC - number of 'random-walk' steps per MCMC chain/realization (integer). For a quick run, set this to 10^4. For most of the model/data combinations in **DICE**, 10^6 steps will be sufficient. However it is recommended to check the resulting parameter time-series plots for convergence. If any parameters are still trending up or down at the end of the chain, a larger value of *nMCMC* is likely needed.

nfit - number of weeks/months at the beginning of the dataset to fit. Must be $\leq$ than the number of weeks/months of data and ≥ 10 (or 4 for monthly data). By setting $nfit$ to be less than the number of data points, the user can compare the predicted incidence to the observed one for data points after $nfit$. When modelling a current season, even if $nfit$ equals the number of weeks/months of data the code will calculate incidence profiles for the entire season - providing a forecast for all future weeks/months.

prior - prior distribution for MCMC procedure (integer). Default is 0 for uninformed log-uniform prior. For data type 'CDC' use prior=1 for an informed prior given by the posterior distribution of the most similar past season.(prior = 1 is available only for the uncoupled procedure for the CDC ILI data.)

da - data augmentation (integer). Default is not to augment data ($da = 0$). To use the average historic curve for data augmentation set: $da = 1$, to use the most similar season: $da = 2$.

Temp - Integer, default it 1. Set to 5 or 10 to heat the Gaussian posterior or log-likelihood by a factor of five or ten.

7 Summary of DICE Output

This section presents an explanation of the outputs of **DICE** by showing sample results. A *runDICE* simulation produces a number of data and graphic files. We begin by describing the naming convention for these files. This is followed by a summary of the visual outputs and how they can be interpreted. The final subsection catalogs the contents of each data file. Our discussion does not include the modeling of the limited Military data. For that, the user should consult the **P-MEDDS** manual which preceded **DICE**.

We begin by describing the output of mechanistic runs. Each of the following subsections describes a file type that is the result of the function call similar to:

```
> runDICE(disease=`flu', data_source = `cdc', year=2015, mod_level=2, RegState="usa", fit_level
      nfit=41, epi_model = 1, model=3, isingle=1, nMCMC=1e5, nreal=1).
```

Output files created by a mechanistic **DICE** run follow the naming format

```
"FileType-disease-epi_name-Mod_Reg-Coupled-Year-model-nfit-realization.FileFormat"
```

where **FileType** is a word or phrase describing the contents of the file (ex. 'gaussian-fit', 'hist-prfl', 'hist-week', 'input', 'llk', 'mcmc', 'profiles', 'results'). See the subsections below for descriptions of these file types. The remaining portions of the filename describe a few vital data and model parameters. **epi_model** is the compartmental model name: sir, seir, vsir or vseir (epi_model = 1, 2, 3 and 4 respectively). **Mod_Reg** is the name of the forecast area (e.g. "United.States" or "Region1"). **Coupled** indicates whether the fit-level profiles are coupled

'cpl' ($isingle = 0$) or uncoupled 'uncpl' ($isingle = 1$). **FileType** can be either 'cdc' or 'gft' depending on the source of ILI data and disease name in all other cases (for example 'dengue' 'yf' for yellow fever). **Year** describes the disease season used (ex. '2015-2016'). The **model** used for the force of infection is designated by an integer 1 through 4. The number of weeks of data fit **nfit** is an integer denoting the number of data points fitted. In case more than one realization ($nreal > 1$) is run, the final portion of the name is an integer designating the **realization** number that produced the file. **DICE** outputs a number of **FileFormat**s including 'csv', 'RData', and 'pdf'. See the FileType descriptions below for some example file names.

7.1 Graphic Files

An instantiation of *runDICE*() will produce three visuals. The primary graphic (FileType 'results') details the incidence data and resulting model fits for all model and fit profiles. Also depicted, if $nfit < nData$, are the model predictions for forecast weeks. By selecting the keyword 'plot' to be 1 or 2 the user can choose if to use the basic R plotting commands or the newer 'ggplot2' option.

7.1.1 Incidence Profile Fit

The summary of results plots produced by the previously defined function call is found in the "United.States_2015_level_2_prior_0_Temp_1_da_0/" sub-directory with filename "results-cdc-sir-United.States-uncpl-2015-2016-4-41-1.pdf". An example result is shown in figure 3. Here the colored profiles (first 10) show individual ILI fits for the 10 HHS Regions. Weekly CDC ILI data points are plotted as black markers connected by black line segments. The CDC's definition of onset level is plotted as a dashed grey line and is constant in time. Notice that there are many fit-curves plotted for each region/nation. These are a sampling of MCMC trials and represent the likelihood of parameters using model 3 for the force of infection. The vertical grey bar indicates the 'forecast' weeks. As a result of $nfit = 41$, data in and to the right of the grey bar have not been used for the fitting procedure. Therefore the continuation of fitting curves works as a forecast for the remaining weeks. This is also indicated by the fit-curves transitioning to dashed-lines. Following the regional profiles are two different national fits. These have been magnified in figure 4. The plot on the left is an uncoupled, population-weighted average of the regional fits. On the right is the result of directly fitting the national data using model 3.

7.1.2 Peak Week Histogram

The FileType 'hist-week-max' are .pdf files that show the likelihood distribution of peak weeks at both the model and fit level. The images in figure 5 come from the file named "hist-week-max-cdc-United.States-uncpl-2015-2016-5-41-1.pdf". In these histograms the ILI incidence peaks have been binned by epidemic week number. The data has a single peak value and is therefore represented by a single blue bar. The MCMC fitting procedure results in a distribution that is depicted by green bars. Overlap of data and fit bars is appears as magenta.

7.1.3 Forecast Weeks Histogram

The FileType 'hist-prfl' are .pdf files that show the likelihood distribution for each of the forecast weeks at both the model and fit levels. Figure 6 shows the forecast epidemic weeks 15-18 for HHS Region 1. These images were extracted from the file "hist-prfl-cdc-United.States-uncpl-2015-2016-5-41-1.pdf". As was done in the CDC flu challenge, %ILI is separated into 0.5%-sized bins. These plots allow one to compare what **DICE** predicts for a forecast week to the actual observed %ILI for that week. Similar to the previous histogram, the data bar is blue and the forecast bars are green. Again, where there is overlap between the data bar and forecast distribution, the bar has been colored magenta. Note that the histograms have been normalized

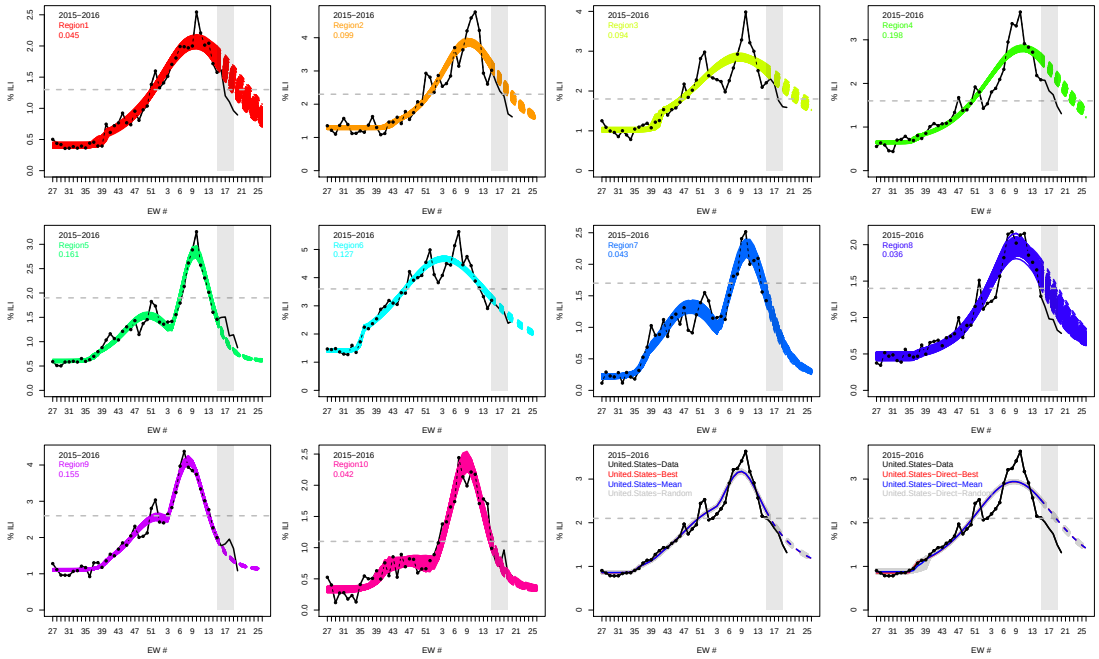

Figure 3: Sample **DICE** output as %ILI as a function of Epidemic Week (EW) using the following parameter values: disease = 'flu', year=2015, model=5, nfit=41, nMCMC=10^5.

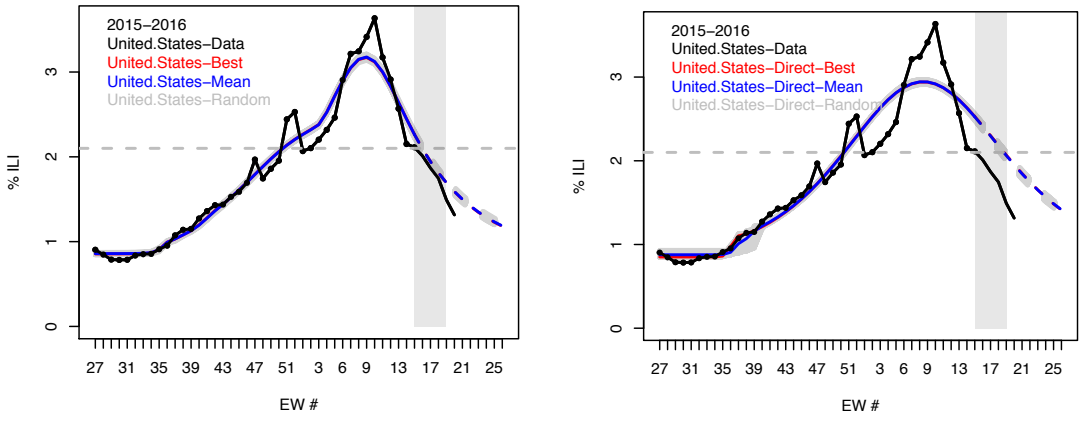

Figure 4: Comparing results of a direct fit (right) and region-fit (left) for 2015-2016 national ILI data.

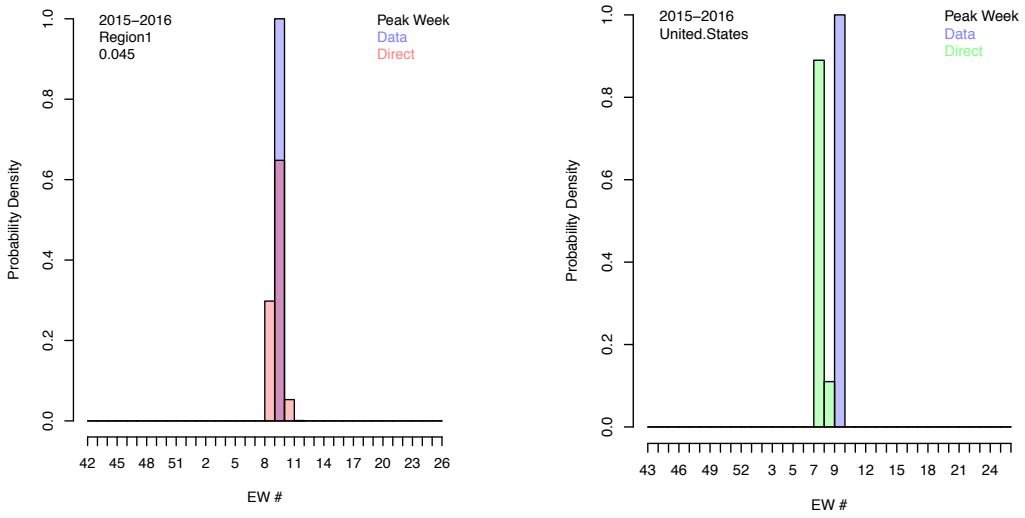

Figure 5: Sample histograms for the ILI peak-week distribution. For Region 1 (left) the blue bar depicts the peak data week and the green bars show the distribution of peak weeks found by MCMC fit. The overlap between the fit and data histograms appears as magenta. The national direct-fit (right) is colored similarly.

such that the area under the distribution is equal to 1. Thus the bin size of 0.5 dictates a max bar height of 2.

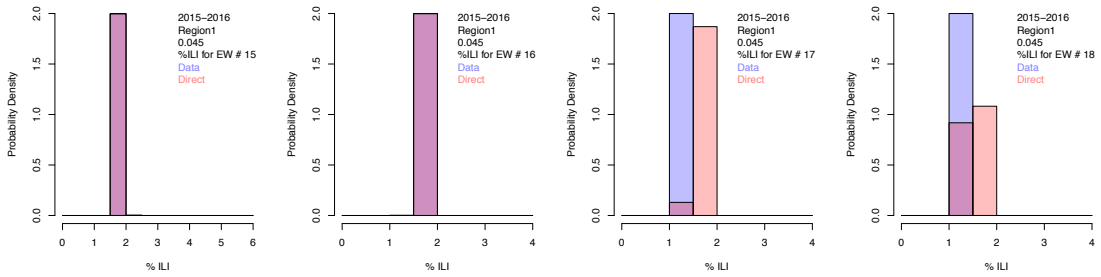

Figure 6: Sample histogram for the ILI forecast weeks of a Region 1 fit. Each plot represents the distribution of %ILI for a single forecast week. As with the previous figures, this results from the parameters: year=2015, disease = 'flu', model=5, nfit=41, nMCMC=10^5.

7.2 Incidence Profiles Fit-non ILI Example

The incidence profile fits/forecasts for diseases other than the flu show the same results as in the case of ILI data described in section 7.1.1. As an example we show in figure 7 the profiles of an uncoupled fit to Dengue data from Brazil for the 2009-2010 season using the five regions of Brazil ($fit\_level = 3$ and $mod\_level = 2$) and an S-E-I-R model ($epi\_model = 2$ or 'SEIR'). The Dengue data is monthly number of cases and it is shown with the black line/circles in each panel. The top row and middle left and middle center panels, show the fits to each of the five regions. The aggregate fit to the nation level data (middle right panel) is the sum of the region level fits and the bottom left panel shows the results of a direct fit to the nation level data. The grey line in each panel is a historic NULL model given by the average monthly number of Dengue cases in each region/nation.

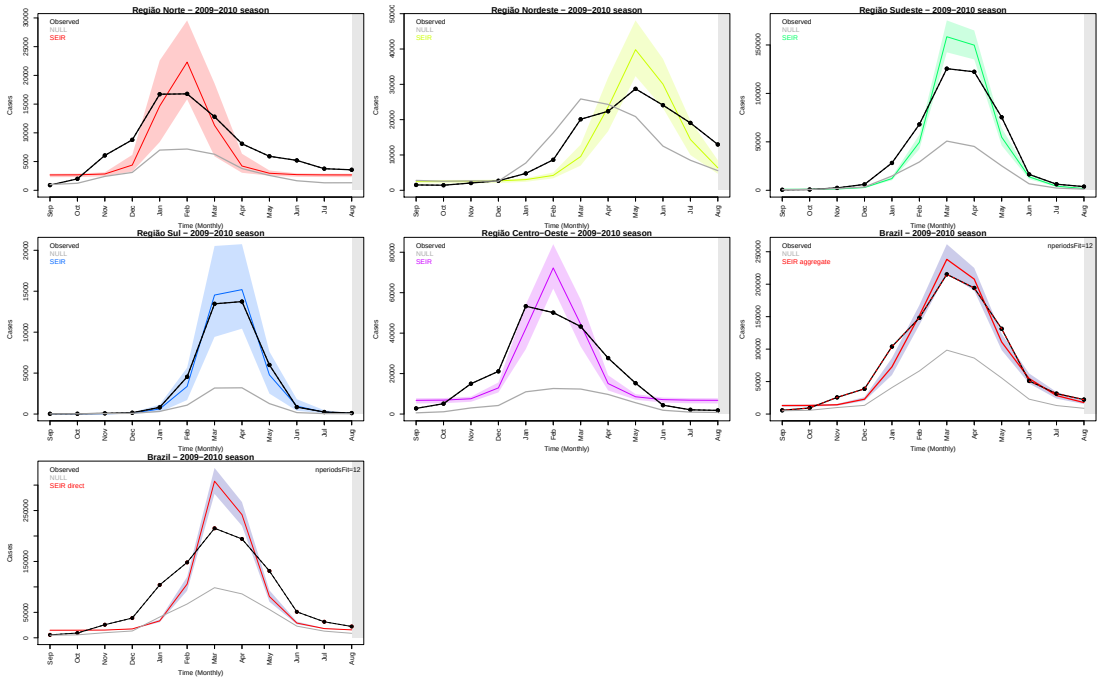

Figure 7: Example of fits to monthly Dengue data from Brazil using region and nation level data. These results are from the parameters: year=2010, disease='dengue', RegState = 'BR', mod_level = 2, fit_level = 3, isingle = 1, epi_model = 'SEIR', model=4, nfit=12, nMCMC = 1e6

7.3 Data Files

Here we briefly summarize the contents of the data files generated by a *runDICE*() simulation. The five types of data files produced by **DICE** are described below.

7.3.1 Input Data and Parameters

FileType 'input' records all of the model and data inputs to the fitting procedure. In this example the filename is "input-cdc-sir-United.States-uncpl-2015-2016-4-41-1.RData". This file contains two R objects, the first is called 'input' and details all model parameter values (generated from user inputs). The second object 'mydata' contains all data generated by **DICE** for the simulation.

7.3.2 MCMC Chain

The raw results from the MCMC fitting procedure are contained in the 'mcmc' FileType. For this example the filename is "mcmc-cdc-sir-United.States-uncpl-2015-2016-4-41-1.RData". The R object 'results.list' is a list of MCMC chains. There is one list entry for each population that was fit during the MCMC procedure. Each list entry is a representative sampling of the MCMC history. This takes a matrix form with each row corresponding to a sample and each column an optimized parameter. This file is the primary output of a simulation, and it is also useful for the purpose of restarting the chain.

7.3.3 ILI Profiles

The 'profiles' FileType is and .RData that stores a representative sample of ILI profiles generated by the MCMC chain. For this example it is named "profiles-cdc-sir-United.States-uncpl-2015-2016-4-41-1.RData". This file contains all information needed to plot the direct and aggregate profiles on the model-level.

7.3.4 Forecast Weeks LLK

When the forecast weeks overlap with observed data, the Log-Likelihood (LLK) is calculated and stored in FileType 'llk'. For the example simulation in this section the filename is: "ili-cdc-sir-United.States-uncpl-2015-2016-4-41-1.csv". This information is useful for quantitatively comparing forecasts of the aggregated and direct fits.

7.4 Season Targets

The file "stats-cdc-sir-United.States-uncpl-2015-2016-4-41-1.csv" contains the observed and predicted (best, mean and standard deviation) values for: onset week, peak week and peak intensity for the model level and fit level regions.

7.4.1 Gaussian Fit

The file "gaussian-fit-cdc-sir-United.States-uncpl-2015-2016-4-41-1.csv" contains a mean and variance for each optimized parameter. This procedure fits a single variable Gaussian to the parameters one-by-one. While this approximation of mean and variance is often quite useful, it should be noted that the distribution returned by the MCMC procedure is a function of many parameters. The parameters are interdependent, and thus a single variable Gaussian may be a poor approximation of the large-dimension distribution. Nevertheless, these fits are useful if the user wishes to use a prior distribution in the MCMC procedure. This has been explored in the context of the CDC challenge for the 2015-2016 flu season.

7.5 Output Files For Statistical Runs

In the case of a statistical run the number of output files is significantly smaller and the directory and file name convention are simpler. As an example we use the 2010 Dengue data for Brazil which we model with a specific SARIMA model and precipitation as a covariate with a lag time of one month:

```
> arima_model = list(p = 1, d = 0, q = 0, P = 3, D = 1, Q = 0)
> covar = 'precip'
> covar_lag = 1
```

DICE is called using:

```
> runDICE(disease = 'dengue', RegState = "BR", year=2010, mod_level=2,
      fit_level=3, nfit=12, method='stat', arima_model = arima_model,
      covar= 'precip', covar_lag = 1)
```

all the output files appear in a directory called "Brazil_2010_level_2/" and the filename convention for the output files is:

```
"FileType-sarima-source-Mod_Reg-Year-SarimaModel-covar-lag-nfit.FileFormat"
```

where **source** is the name name of the data source, **Mod_Reg** is (as before) the name of the model region, **Year** is the season, **SarimaModel** is the detail of the sarima Model we chose, **covar** is the covariate name, **lag** is the lag time for the covariate and **nfit** is the number of data points fitted. The statistical run produces two graphic files. The first graphic file contains the results. It is named in this case: "results-sarima-bra_mh-Brazil-2009-2010-100-310-precip-lag-1-12.pdf" and shown in figure 8

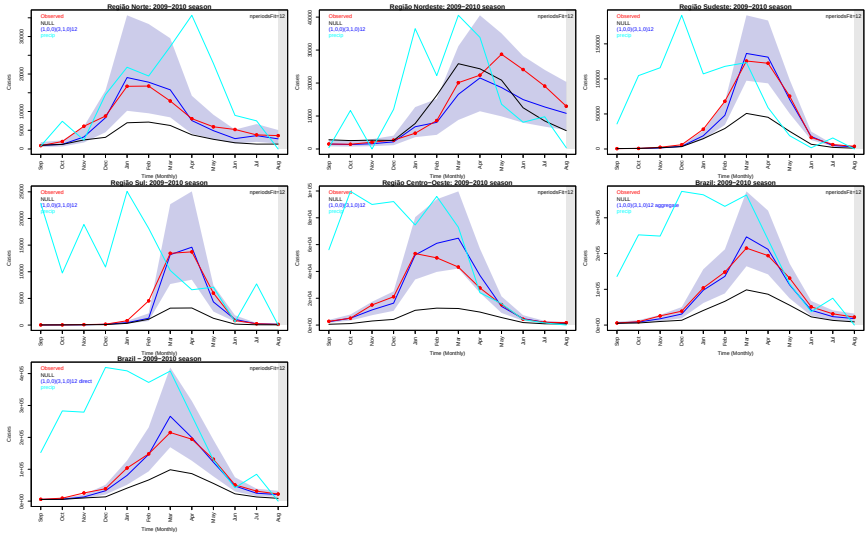

Figure 8: Sample of SARIMA fits to the region and national level Dengue Data for Brazil. The first five panels (top row and left/middle panels in the second row) show the fits to the five regions followed by the aggregate fit to the National curve (right panel in second row). The last panel (bottom row) is the result of the direct SARIMA fit to the national incidence. The red line in each panel is the data, blue line is the fit and the shaded region denotes the standard deviation. The black line in each panel is the historic average which we call the "NULL" model. The legend in each panel denotes the SARIMA model and the cyan line is the covariate (precipitation in this example).

The aggregate fits are obtained by simply adding the fits to the five regions. The second graphic file compares the fit/forecast to the observation by depicting the error. It is called in this case: "error-sarima-bra_mh-Brazil-2009-2010-100-310-precip-lag-1-12.pdf" and shown in figure 9:

The statistical runs produce a single binary file named in this example: "sarima-bra_mh-Brazil-2009-2010-100-310-precip-lag-1-12.RData". This file contains all the data used for this run, the details of the SARIMA model and tables with the results of the fit/forecast. This file can be loaded to produce additional plots.

Before concluding this section we note that if the user does not define a SARIMA model, and lets **DICE** select the best model for each region and the nation, the output file names replace the "SarimaModel-covar-lag" part with "auto-arima". In this case, the legend of each panel in the plot files will denote the specifics of the SARIMA model chosen by **DICE** for each region/nation.

8 DICE Key Words

The following is a list of all key words recognized by the **DICE** package:

- **db_opts** A list of database options. DICE_db Determines which SQL database the data

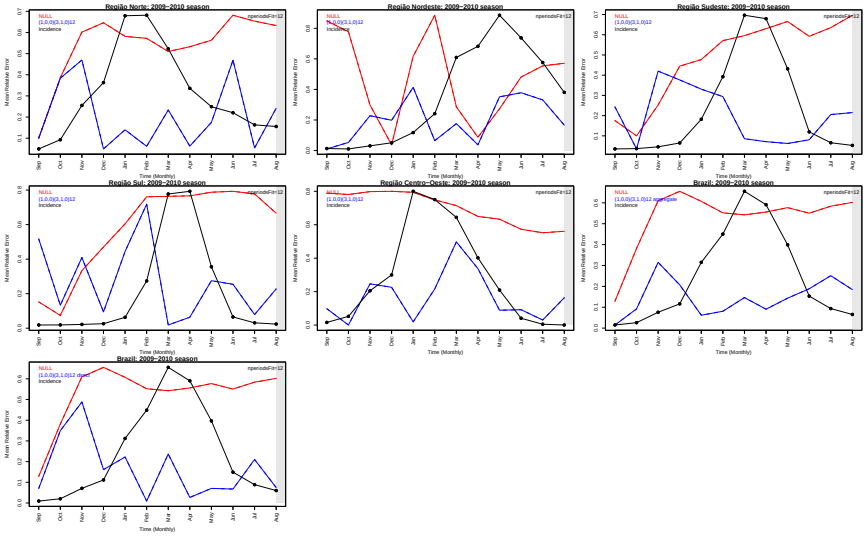

Figure 9: The error of the SARIMA fits shown in the previous figure. The black line/dots are the observed incidence and the blue line is the mean relative error (MRE) of the SARIMA fit. In red we show the MRE error of the historic NULL model. The panels are ordered as before: five regions followed by nation aggregate and nation direct (bottom panel).

is retrieved from. 'PredSci' is the default SQL database, 'BSVE' is in development. Additional flags are for outside sources of data (currently only the CDC Influenza-Like_Illness (ILI) is supported: CDC_server=TRUE).

- **disease** String - disease name. Options for modeling are: flu, dengue, yellow_fever, ebola, zika, cholera, chik, plague. To graphically explore the data see: predsci.com/id_data/. A full list of diseases in the DICE database can be found from an R-prompt by:

```
> library(DICE)
> myDB = OpenCon()
> data_sources = dbReadTable(con=myDB, name="data_sources")
> dbDisconnect(myDB)
> unique(data_sources$disease)
```

- **data_source** - Describes the data source for the incidence data. Default is 'cdc' (for `disease = 'flu'`). It can be selected by source_key (integer) or source abbreviation (string). Most disease/location combinations have only one data source. In this case, it may be easier to set data_source=NULL. However, when multiple data sources exist, setting data_source=NULL will essentially choose from the available sources at random. To determine a data source by graphical interface, see: predsci.com/id_data/. Looking-up the disease and location will result in a list of data sources that can be entered into DICE. Alternatively, all country/disease/data_source combinations are listed in the 'Data Sources Table' tab at the same url. To access the list of sources directly:

```
> library(DICE)
> myDB = OpenCon()
> data_sources = dbReadTable(con=myDB, name="data_sources")
> dbDisconnect(myDB)
> str(data_sources)
> data_sources$source_abbv
```

- **mod_level** - integer. Spatial level of the model data. The US/CDC covers the following levels: 2-United States, 3-HHS Regions, 4-State. Level 1 is always continent and level 2 is always country, but the levels of smaller divisions vary by country.
- **fit_level** - integer. Spatial level of data used for a coupled or uncoupled fit of the model data. Again, this varies by country and data availability, but fit_level must always be greater than or equal to mod_level.
- **mod_name** - Named character vector specifying the model-level spatial patch. If `is.null(mod_name)`, the code reverts to using `RegState`. To specify New York state, set `mod_name=c(NAME_2="United States", NAME_3="R2", NAME_4="New York")`. Here NAME_X is either the full name or abbreviation of the level-X patch. Replacing 'United States' with 'US' or 'R2' with 'Region 2' would result in the same outcome. Also, vector entries for mod_name should go from NAME_2,....,NAME_n where mod_level=n.
- **RegState** - name of location to model (default 'usa'). This is an older input that has been retained for backwards compatibility. It is suggested that users use mod_name instead. Single element: determines which single region from `mod_level` is to be modeled. Depending on the model level, `RegState` should adhere to the following format: `mod_level = 2` - 3-letter ISO3 code, `mod_level=3` - an integer describing the HHS region, `mod_level=4` - a 2-letter state code (US). RegState is limited to country-level data and US regions/states.
- **fit_names** A character vector indicating which fit-regions to use. If `fit_name='all'`, then DICE uses all child-regions of the model region with level equal to `fit_level`. The other mode for fit_name is to specifiy a subset of the fit regions to construct an aggregate representation of the model region. For example if `mod_level=c(NAME_2="US")`, `mod_level=2`, `fit_level=3`, and `fit_names=c("R1", "R2", "R3")`, DICE will create and model an Atlantic super-region from HHS regions 1, 2, and 3 (as opposed to using all 10 HHS regions). Similarly, if `mod_level=c(NAME_2="US")`, `mod_level=2`, `fit_level=4`, and `fit_names=c("WA", "OR", "CA")`, DICE will create and model a super-state of Pacific states.
- **nfit** Integer - Number of data points that will be fitted. Default is to fit all the data. This will be reset if nfit > nperiodsData (number of data points for the season).
- **method** String either 'mech' for compartmental mechanistic models or 'stat' for SARIMA models
- **epi_model** String or Integer - Name of mechanistic compartmental model: SIR, SEIR, (SIR)H/(SI)V , (SEIR)H/(SEI)/V, or SIRB integer 1,2,3,4,5 (case insensitive).
- **model** Integer - The model number, see Section 3.5 for more details (1-4 are supported for flu 4 for all other diseases). Relevant only when method = 'mech'.
- **Tg** Numeric - recovery time in days. If NULL it is set to three/eight days for flu/dengue. Relevant only when method = 'mech'.
- **isingle** Integer - 0: couple the fit spatial regions; 1: no coupling. Relevant only when method = 'mech'.
- **prior** Integer - if greater than zero use a prior for the MCMC procedure. Relevant only when method = 'mech'.
- **da** Integer 0, 1 or 2. Data augmentation options: 0-none, 1-using historic average and 2-using the most similar season. Relevant only when method = 'mech'.
- **Temp** Integer 1, 5, 10, 100 - Temperature for the MCMC procedure. Relevant only when method = 'mech'.

- **nMCMC** Integer - number of steps/trials in the MCMC procedure. Relevant only when method = 'mech'.
- **nreal** Integer - number of MCMC chains. Relevant only when method = 'mech'.
- **arima_model** - A List of ARIMA model parameters: list(p=, d=, q=, P=, D, Q=) can be set to NULL to trigger the `auto.arima` process. Relevant only when method = 'stat'.
- **covar** String, optional. Covariate for use in ARIMA fitting. Options are: 'sh', 'precip', 'temp'. Relevant only when method = 'stat'. Default is NULL - no covariate. In the case of CDC flu data school vacation can also be used as a covariate (`covar = 'sv'`).
- **covar_lag** Numeric lag time for optional covariate variable in time units of cadence of the data. Relevant only when method = 'stat' and `covar` is not NULL.
- **device** String Either 'pdf' or 'png' (default).
- **plot** TRUE, FALSE or EXTERNAL (or 0, 1, 2) allows the Users to implement their own plotting routines. Default is TRUE.
- **subDir** String - Name of output sub-directory where all plots and files will be written. Default is NULL -let the code build it.

References

[1] Alan I Barreca and Jay P Shimshack. Absolute humidity, temperature, and influenza mortality: 30 years of county-level evidence from the united states. *Am J Epidemiol*, 176(suppl 7):S114–S122, 2012.

[2] Michal Ben-Nun, Pete Riley, James Turtle, David Bacon, and Steven Riley. Forecasting national and regional influenza-like-illness for the usa. *PLoS Comput Biol*, to be submitted, 2018.

[3] Samantha Cook, Corrie Conrad, Ashley L Fowlkes, and Matthew H Mohebbi. Assessing google flu trends performance in the united states during the 2009 influenza virus a (h1n1) pandemic. *PloS one*, 6(8):e23610, 2011.

[4] Patrick Copeland, Raquel Romano, Tom Zhang, Greg Hecht, Dan Zigmond, and Christian Stefansen. Google disease trends: an update. *Nature*, 457:1012–1014, 2013.

[5] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC Interdisciplinary Statistics Series. Chapman & Hall, 1996.

[6] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012, 2009.

[7] Eugenia Kalnay, Masao Kanamitsu, Robert Kistler, William Collins, Dennis Deaven, Lev Gandin, Mark Iredell, Suranjana Saha, Glenn White, John Woollen, et al. The ncep/ncar 40-year reanalysis project. *B Am Meteorol Soc*, 77(3):437–471, 1996.

[8] M. J. Keeling and P. Rohani. *Modeling Infectious Diseases In Humans and Animals*. Princeton University Press, 2008.

[9] Anice C Lowen, Samira Mubareka, John Steel, and Peter Palese. Influenza virus transmission is dependent on relative humidity and temperature. *PLoS Pathog*, 3(10):e151, 2007.

[10] Harriet L. Mills and Steven Riley. The spatial resolution of epidemic peaks. *PLoS Comput Biol*, 10(4):1–9, 04 2014.

[11] Kenneth E Mitchell, Dag Lohmann, Paul R Houser, Eric F Wood, John C Schaake, Alan Robock, Brian A Cosgrove, Justin Sheffield, Qingyun Duan, Lifeng Luo, et al. The multi-institution north american land data assimilation system (nldas): Utilizing multiple gcip products and partners in a continental distributed hydrological modeling system. *J Geophys Res-Atmos*, 109(D7), 2004.

[12] Pete Riley, Michal Ben-Nun, Richard Armenta, Jon A Linker, Angela A Eick, Jose L Sanchez, Dylan George, David P Bacon, and Steven Riley. Multiple estimates of transmissibility for the 2009 influenza pandemic based on influenza-like-illness data from small us military populations. *PLoS computational biology*, 9(5):e1003064.

[13] Pete Riley, Michal Ben-Nun, Jon A. Linker, Angelia A. Cost, Jose L. Sanchez, Dylan George, David P. Bacon, and Steven Riley. Early characterization of the severity and transmissibility of pandemic influenza using clinical episode data from multiple populations. *PLoS Comput Biol*, 11(9):1–15, 09 2015.

[14] SM Schappert and CW Burt. Ambulatory care visits to physician offices, hospital outpatient departments, and emergency departments: United states, 2001-02. *Vital and Health Statistics. Series 13, Data from the National Health Survey*, (159):1–66, 2006.

[15] Jeffrey Shaman, Edward Goldstein, and Marc Lipsitch. Absolute humidity and pandemic versus epidemic influenza. *Am J Epidemiol*, 173(2):127–135, 2011.

[16] Jeffrey Shaman and Melvin Kohn. Absolute humidity modulates influenza survival, transmission, and seasonality. *PNAS*, 106(9):3243–3248, 2009.

[17] Jeffrey Shaman, Virginia E. Pitzer, Cécile Viboud, Bryan T. Grenfell, and Marc Lipsitch. Absolute humidity and the seasonal onset of influenza in the continental united states. *PLoS Biol*, 8(2):1–13, 02 2010.

[18] A Talwalkar, E Hing, and K Palso. National ambulatory medical care survey: 2011 summary tables.

[19] A Talwalkar, E Hing, and K Palso. National hospital ambulatory medical care survey: 2011 outpatient department summary tables.

[20] Youlong Xia, Kenneth Mitchell, Michael Ek, Justin Sheffield, Brian Cosgrove, Eric Wood, Lifeng Luo, Charles Alonge, Helin Wei, Jesse Meng, et al. Continental-scale water and energy flux analysis and validation for the north american land data assimilation system project phase 2 (nldas-2): 1. intercomparison and application of model products. *J Geophys Res-Atmos*, 117(D3), 2012.