Option File for SEIR Model

2024-08-15

Overview

```
# This folder contains multiple pre-configured option files,
# providing users with the flexibility to select and utilize any of them as needed.
# Specifically, this option file is designed for general use,
# allowing users to work seamlessly with their own models alongside the other available options.
# It is recommended to review the tutorial before using this file to maximize its effectiveness.


# Set the calibration periods for the model.
# These are specified as a numeric vector, with each element
# representing a distinct period (in this case, 17 and 18)
# used for calibrating the model's parameters.
calibrationperiods <- c(17)


# Set the forecasting horizon for the model.
# This value indicates the number of time steps (in this case, 10)
# into the future for which predictions will be made.
# The forecasting horizon must be specified as a single numeric value.
forecastinghorizon <- 10


# Specify the name of the model to be used.
# In this case, the model is named "SEIR-test".
# This name can be referenced later in the code for clarity and organization.
model_name <- "SEIR-test"


# Define the names of the state variables for the SEIR model.
# These variables represent the different compartments in the model:
# "S" for susceptible individuals,
# "E" for exposed individuals,
# "I" for infected individuals,
# "R" for recovered individuals,
# and "C" for cumulative cases.
vars <- c("S", "E", "I", "R", "C")


# Define the model parameters as a vector.
# Each parameter plays a crucial role in the dynamics of the SEIR model:
#
# - beta: the transmission rate of the disease, with an expected range of (0, 10).
# - gamma: the recovery rate of infected individuals, with an expected range of (0, 10).
```

```
# - kappa: the incubation rate, representing the rate at which exposed individuals become infected,
#   with an expected range of (0, 10).
# - rho: the recovery proportion rate, indicating the fraction of infected individuals that recover,
#   with an expected range of (0, 1).
# - N: the total population size (not explicitly defined in the example).
# - i0: the initial number of infected individuals (not explicitly defined in the example).
params <- c("beta", "gamma", "kappa", "rho", "N", "i0")

# To define a time-dependent parameter, create a function as a string that depends on time
# t and elements from the params vector. If there are no any time-dependent functions
# in the model, keep the time_dependent_templates list empty.
# Example:
#time_dependent_param1 <- "return (params1 * exp(-params2 * t));" meaning beta * e^(-gamma*t)

time_dependent_templates <- list(

)

# Define the system of ordinary differential equations (ODEs) for the SEIR model.
# Each equation represents the rate of change for a specific state variable over time:
#
# - diff_var1: Rate of change of susceptible individuals (S).
# - diff_var2: Rate of change of exposed individuals (E).
# - diff_var3: Rate of change of infected individuals (I).
# - diff_var4: Rate of change of recovered individuals (R).
# - diff_var5: Rate of change of cumulative cases (C).
#
# The equations utilize the following parameters:
# - params1: Transmission-related parameter influencing S and E.
# - params2: Recovery-related parameter influencing I and R.
# - params3: Rate of exposure influencing E and I.
# - params4: Affects the cumulative cases.
# - params5: Total population size, impacting the transmission dynamics.
ode_system <- '
  diff_var1 = -params1 * vars3 * vars1 / params5
  diff_var2 = params1 * vars3 * vars1 / params5 - params3 * vars2
  diff_var3 = params3 * vars2 - params2 * vars3
  diff_var4 = params2 * vars3
  diff_var5 = params4 * params3 * vars2'

# Specify whether each parameter is fixed (1) or should be estimated (0) in the model.
# The 'paramsfix' vector indicates this for each parameter:
# - paramsfix[1]: For beta (transmission rate). 0 means it will be estimated.
# - paramsfix[2]: For gamma (recovery rate). 0 means it will be estimated.
# - paramsfix[3]: For kappa (incubation rate). 0 means it will be estimated.
# - paramsfix[4]: For rho (recovery proportion rate). 0 means it will be estimated.
# - paramsfix[5]: For N (total population size). 1 means it is fixed.
# - paramsfix[6]: For i0 (initial number of infected individuals). 0 means it will be estimated.
paramsfix <- c(0, 0, 0, 0, 1, 0)


# Define a list of expressions of interest that will be generated based on model parameters.
# Each expression provides important insights into the model dynamics:
```

```
# - R0: The basic reproduction number, calculated as the transmission rate (beta)
#   divided by the recovery rate (gamma).
# - recovery_time: The average time until an infected individual recovers,
#   calculated as the inverse of the recovery rate (gamma).
composite_expressions <- list(
  R0 = "beta / gamma",
  recovery_time = "1 / gamma"
)


# Specify the index of the model variable that will be fitted to the observed time series data.
# In this case, fitting_index is set to c(3,5), indicating that the model will fit the second
#  column in the data to the infectious (I) and the third column in the data to the
# cumulative cases variable (C) in the SEIR model.
fitting_index <- c(3,5)


# Boolean variable indicating whether the derivative of the model's fitting variable should be fitted t
# Here, a value of 1 (true) means the data will be fitted to dC/dt (the rate of change of cumulative ca
# while a value of 0 (false) would indicate that the data will be fitting to I (the number of infectiou
fitting_diff <-c(0,1)



# Select the type of error structure to be used in the model fitting.
# The options are as follows:
# 1. Negative binomial: Suitable for overdispersed count data.
# 2. Normal: Assumes normally distributed errors, typically used for continuous data.
# 3. Poisson: Assumes Poisson-distributed errors, commonly used for count data.
# In this case, errstrc is set to 3, indicating that the Poisson error structure will be used.
errstrc <- 1



# Define the input file name for the model data.
# The input file has the prefix "SanFrancisco" and the extension ".xlsx",
# indicating it is an Excel file containing the relevant data for the analysis.
cadfilename1 <- "SanFrancisco"


# String indicating the name of the disease associated with the time series data.
# In this case, caddisease is set to "SF1918", which refers to the
# influenza outbreak that occurred in San Francisco in 1918.
caddisease <- "SF1918"


# String indicating the type of data being used in the analysis.
# In this case, datatype is set to c("Infectious","newly infected people"), which signifies that the
# model will analyze the number of infectious and reported cases associated with the disease.
# Time series cases names
series_cases <- c("Infectious","newly infected people")
```

```r
# String indicating the unit of time for the data being analyzed.
# In this case, datetype is set to "Days", which indicates that the
# time series data is measured in days.
datetype <- "Days"


# User-defined priors for each model parameter.
# For each parameter, define the prior distribution by appending _prior to the parameter name.
# Since all parameters are expected to have positive values, it is recommended to truncate
# the distribution at zero where applicable. The following distributions are suggested:
#
# - params1_prior: Prior distribution for beta (transmission rate).
#   - Normal distribution: "normal(0, 1)T[0,]" - Truncated normal with mean 0 and standard deviation 1.
#   - Log-normal distribution: "lognormal(0, 1)" - Suitable for positive values, representing a multipl
#   - Exponential distribution: "exponential(1)" - Useful for modeling rates, ensuring positivity.
#   - Uniform distribution: "uniform(0, 10)" - Assumes the parameter is equally likely to take any valu
#
# - params2_prior: Prior distribution for gamma (recovery rate).
#   - Normal distribution: "normal(0, 1)T[0,]" - Truncated normal with mean 0 and standard deviation 1.
#   - Log-normal distribution: "lognormal(0, 1)" - Suitable for positive values.
#   - Exponential distribution: "exponential(1)" - Represents the rate of recovery, ensuring positivity
#   - Uniform distribution: "uniform(0, 10)" - Assumes the parameter is equally likely to take any valu
#
# - params3_prior: Prior distribution for kappa (incubation rate).
#   - Normal distribution: "normal(0, 1)T[0,]" - Truncated normal with mean 0 and standard deviation 1.
#   - Log-normal distribution: "lognormal(0, 1)" - Represents the positive nature of incubation rates.
#   - Exponential distribution: "exponential(1)" - Ensures positivity while modeling rates.
#   - Uniform distribution: "uniform(0, 10)" - Assumes the parameter is equally likely to take any valu
#
# - params4_prior: Prior distribution for rho (recovery proportion rate).
#   - Normal distribution: "normal(0, 1)T[0,]" - Truncated normal with mean 0 and standard deviation 1.
#   - Beta distribution: "beta(1, 1)" - Suitable for proportions, defined between 0 and 1.
#   - Uniform distribution: "uniform(0, 1)" - Assumes the recovery proportion is equally likely to take
#
# - params5_prior: A fixed value of 1000000 for N (total population size).
#
# - params6_prior: Prior distribution for i0 (initial number of infected individuals).
#   - Normal distribution: "normal(0, 10)T[0,]" - Truncated normal with mean 0 and standard deviation 1
#   - Log-normal distribution: "lognormal(0, 10)" - Suitable for positive values.
#   - Exponential distribution: "exponential(10)" - Ensures positivity, suitable for initial counts.
#   - Uniform distribution: "uniform(0, 100)" - Assumes the initial count is equally likely to take any
params1_prior <- "normal(0, 1)T[0,]"
params2_prior <- "normal(0, 1)T[0,]"
params3_prior <- "normal(0, 1)T[0,]"
params4_prior <- "normal(0, 1)T[0,]"
params5_prior <- 1000000
params6_prior <- "normal(0, 10)T[0,]"




# Define the lower bounds for each model parameter.
# These values indicate the minimum allowable values for the parameters,
```

```
# ensuring that all parameters remain positive where applicable.
# - params1_LB: Lower bound for beta (transmission rate). Set to 0.
# - params2_LB: Lower bound for gamma (recovery rate). Set to 0.
# - params3_LB: Lower bound for kappa (incubation rate). Set to 0.
# - params4_LB: Lower bound for rho (recovery proportion rate). Set to 0.
# - params6_LB: Lower bound for i0 (initial number of infected individuals). Set to 0.
params1_LB <- 0
params2_LB <- 0
params3_LB <- 0
params4_LB <- 0
params6_LB <- 0

# Define the upper bounds for each model parameter.
# These values indicate the maximum allowable values for the parameters,
# helping to constrain the parameter estimates during the fitting process.
# - params1_UB: Upper bound for beta (transmission rate). Set to NA, indicating no upper limit.
# - params2_UB: Upper bound for gamma (recovery rate). Set to NA, indicating no upper limit.
# - params3_UB: Upper bound for kappa (incubation rate). Set to NA, indicating no upper limit.
# - params4_UB: Upper bound for rho (recovery proportion rate). Set to 1, as it is a proportion.
# - params6_UB: Upper bound for i0 (initial number of infected individuals). Set to NA, indicating no u
params1_UB <- NA
params2_UB <- NA
params3_UB <- NA
params4_UB <- 1
params6_UB <- NA




# Select the prior distribution for the model when using a normal or negative binomial
# error structure. The choice of prior distribution can influence the fitting process
# and should reflect prior knowledge about the parameters.
#
# - normalerror_prior: Prior distribution used when the normal error structure is specified.
#   In this case, a Cauchy distribution "cauchy(0, 2.5)" is chosen, which has heavier tails
#   than the normal distribution, allowing for greater variability and accommodating outliers.
#
# - negbinerror_prior: Prior distribution used when the negative binomial error structure is specified.
#   Here, an exponential distribution "exponential(5)" is selected, which assumes that the
#   counts of events follow a memoryless property, making it suitable for modeling overdispersed count
normalerror1_prior <- "cauchy(0, 2.5)"     # It has been defined but there is no need as errstrc = 1
negbinerror1_prior <- "exponential(5)"     # This is used for the first time series
negbinerror2_prior <- "exponential(5)"     # This is used for the second time series


# Select whether to estimate the initial condition for the model.
# Set vars.init to 0 if you want the initial condition to be estimated,
# allowing the model to determine the starting point based on the data.
# Alternatively, set it to 1 if you want to fix the initial condition,
# preventing it from being estimated during the fitting process.
vars.init <- 0
```

```r
# Enter the initial conditions for the model variables.
# The vector Ic specifies the initial values for each state variable:
# - Ic[1]: Initial condition for susceptible individuals (S) calculated as N - i0.
# - Ic[2]: Initial condition for exposed individuals (E), set to 0.
# - Ic[3]: Initial condition for infected individuals (I), set to i0 (initial infected count).
# - Ic[4]: Initial condition for recovered individuals (R), set to 0.
# - Ic[5]: Initial condition for cumulative cases (C), set to i0.
# If vars.init = 0, it is better to define each entry in the array as a string.
# If vars.init = 1, the entries should be numeric values.
Ic = c("N-i0", 0, "i0", 0, "i0")


# Number of Markov Chain Monte Carlo (MCMC) steps to be performed during the sampling process.
# The variable niter is set to 100, indicating that the MCMC algorithm will run for 100 iterations
# to sample from the posterior distribution of the model parameters.
niter <- 1000


# Number of Markov Chain Monte Carlo (MCMC) chains to be run during the sampling process.
# The variable num_chain is set to 2, indicating that two independent chains
# will be initiated to sample from the posterior distribution of the model parameters.
# Running multiple chains helps assess convergence and improves the reliability of the results.
num_chain = 2
```