

Using ABC to Infer Parameters of a Simulated Zombie Epidemic

Henry Bourne, Emma Tarmey, Rachel Wood

June 2023

The REVISIT package

Need to create a package with appropriate folders and documentation - am happy to start on this

Documentation

Integration of R and C++

Simulations

```
# ----- Working Examples -----

# Replace source call with install.github() and library calls when package is finished!
# install.github("https://github.com/RaspberryEmma/Intractable-Models-Sim-Study")
# library(intractmodelinf)

source("intractmodelinf/R/generate_data.R")
suppressPackageStartupMessages({
  library(dplyr)
  library(ggplot2)
  library(knitr)
  library(tidyverse)
})

bristol.uni.N <- 29434
bristol.N     <- 467099
UK.N         <- 67081234

bristol.uni.example <- generate.SIR.data(total.N      = bristol.uni.N,
                                         initial.inf = 10,
                                         total.T      = 150)

bristol.example <- generate.SIR.data(total.N      = bristol.N,
                                     initial.inf = 10,
                                     total.T      = 150)

UK.example <- generate.SIR.data(total.N      = UK.N,
```

```

        initial.inf = 10,
        total.T      = 150)

bristol.uni.example %>% head() %>% knitr::kable()

```

t	S.t	I.t	R.t	N	b	k
1	29434	10	0	29434	0.6323166	0.3918715
2	29427	12	3	29434	0.6323166	0.3918715
3	29419	14	7	29434	0.6323166	0.3918715
4	29410	17	12	29434	0.6323166	0.3918715
5	29399	21	18	29434	0.6323166	0.3918715
6	29385	26	26	29434	0.6323166	0.3918715

```

# ----- Save Data -----

write.csv( bristol.uni.example, file = "data/bristol_uni_example.csv", row.names = FALSE
  ↪ )
write.csv( bristol.example,      file = "data/bristol_example.csv", row.names = FALSE )
write.csv( UK.example,          file = "data/UK_example.csv", row.names = FALSE )

```

Parallelisation

Generating SZR data

We simulate the zombie epidemic using the `generate.SIR.data()` function, which is given by the code below. For ease, this function has also been included in our `intractmodelinf` package

```

generate.SIR.data <- function(total.N = NULL, initial.inf = NULL, total.T = NULL) {

  # initial model conditions
  cond <- generate.start.cond(total.N = total.N, initial.inf = initial.inf)

  # constant model parameters
  N    <- total.N
  b    <- cond[5]
  k    <- cond[6]

  # results array to hold values of each function at times 1 to total.T
  results <- array(data = NA, dim = c(total.T, length(cond)+1) )
  results[1, ] <- c(1, cond)

  # vector to hold current sim values
  values.t <- NULL

  # run through SIR simulation for times 2 to total.T
  # results vector indices 1=S, 2=I, 3=R
  for (t in 2:total.T) {
    S.t <- as.integer( results[t-1, 2] + N*change.s(b, results[t-1, 2]/N, results[t-1,
  ↪ 3]/N) )
  }
}

```

```

    I.t <- as.integer( results[t-1, 3] + N*change.i(b, k, results[t-1, 2]/N, results[t-1,
↪ 3]/N) )

    R.t <- as.integer( results[t-1, 4] + N*change.r(k,      results[t-1, 3]/N) )

    values.t      <- c(t, S.t, I.t, R.t, N, b, k)
    results[t, ] <- values.t
  }

  # convert to data.frame to explicitly record column meaning
  results      <- as.data.frame(results)
  colnames(results) <- c("t", "S.t", "I.t", "R.t", "N", "b", "k")

  return ( results )
}

```

We use this to simulate an outbreak for 3 diff

Implementing ABC

Checking Results

```

bristol.uni.example <- read.csv("data/bristol_uni_example.csv", header = TRUE)
bristol.example     <- read.csv("data/bristol_example.csv", header = TRUE)
UK.example          <- read.csv("data/UK_example.csv", header = TRUE)

legend.colors <- c("Susceptible" = "red", "Infected" = "blue", "Recovered" = "green")

plot.SIR(bristol.uni.example)

```