Models using continuous-time Markov chains

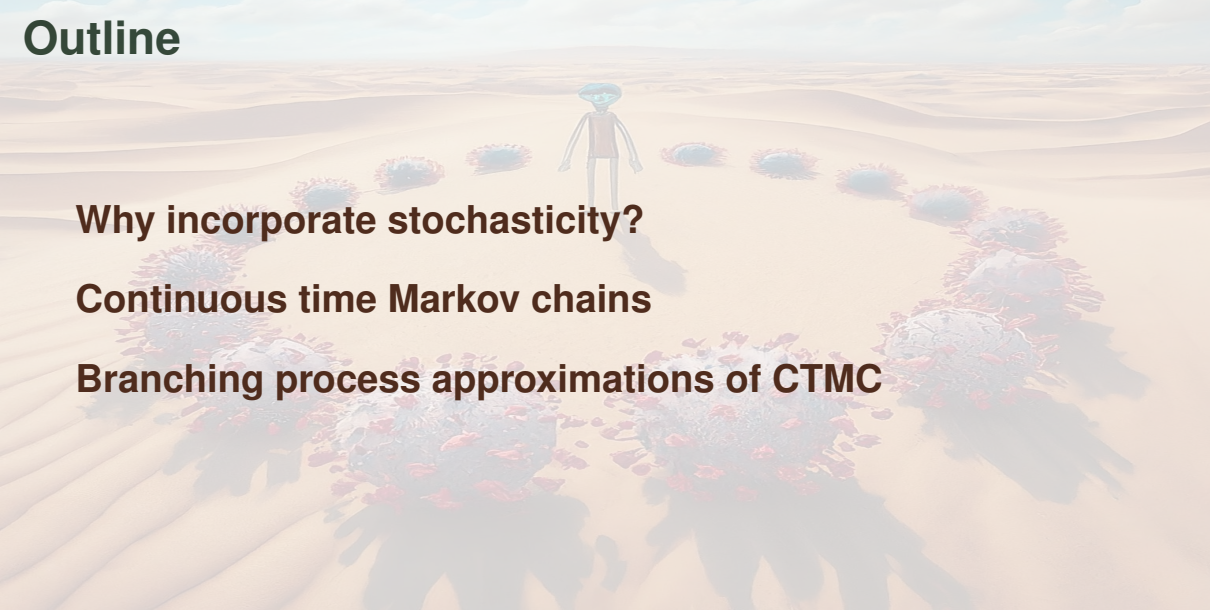
MATH 8xyz – Lecture 23

Julien Arino
Department of Mathematics @ University of Manitoba
Maud Menten Institute @ PIMS
julien.arino@umanitoba.ca

Winter 20XX

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis. We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

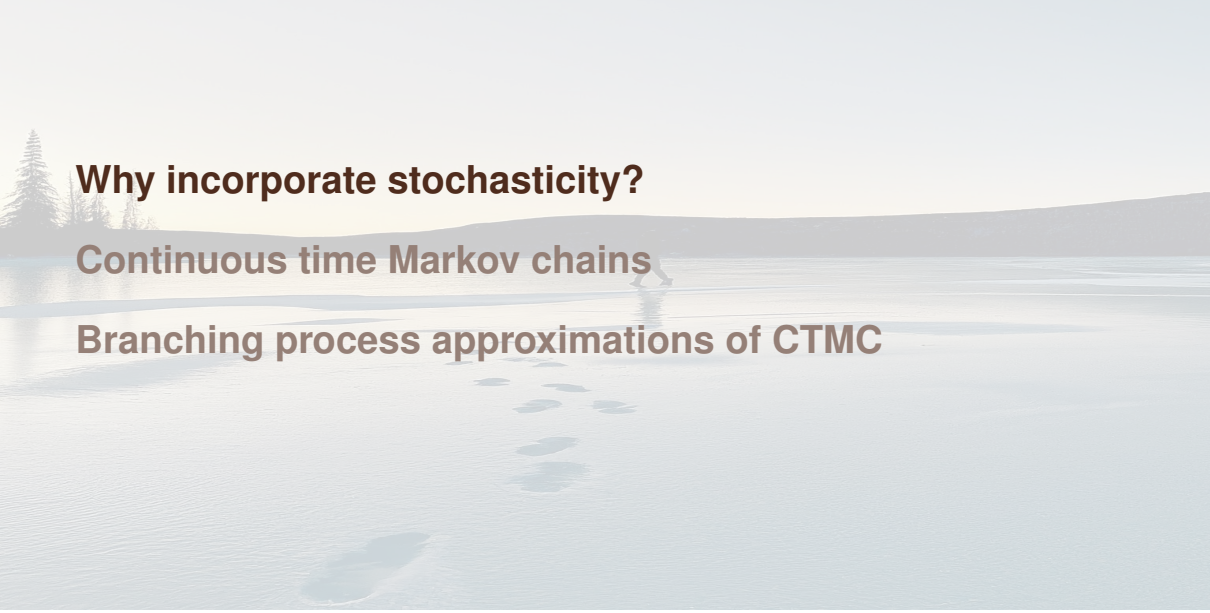
Outline



Why incorporate stochasticity?

Continuous time Markov chains

Branching process approximations of CTMC



Why incorporate stochasticity?

Continuous time Markov chains

Branching process approximations of CTMC

At the beginning of the COVID-19 crisis

- I was working under contract with the Public Health Agency of Canada on *COVID-19 importation risk assessment*

- Produced daily report with list of countries most likely to next report cases of COVID-19

- Used ensemble runs of a fitted global deterministic metapopulation model

- Very very long days (18-20 hours, 7 days a week)

- including a lot of time waiting for the “cluster” to finish

$\Longrightarrow$ PHAC gave me money for a cluster (yay Threadrippers!!!)

$\Longrightarrow$ Also thought about whether my model was really adequate as our focus switched from thinking about movement on a planetary scale to movement within Canadian provinces

What is wrong with deterministic models?

- I pointed out yesterday that SARS-CoV-2 is one *single* realisation of a stochastic process

- Deterministic models "operate on averages" over a large ($\to \infty$) number of realisations

- If we want to get a better sense of what could happen, not only on average, then we need to see what can indeed happen

My new focus – Introductions

- I started thinking in particular about **introductions** (or importations) of pathogens into new populations

- Indeed, introductions are an obligatory step in spatial spread

First piece of evidence

In real life, introductions of pathogens does not always follow the patter

$$\{\mathcal{R}_0 < 1 \implies \rightarrow \text{DFE} \mid \mathcal{R}_0 > 1 \implies \text{epidemic or} \rightarrow \text{EEP}\}$$

Aging and Disease
www.aginganddisease.org

Early access date: October 28, 2022
http://dx.doi.org/10.14336/AD.2022.0820

Short Communication

SARS-CoV-2 in Nursing Homes: Analysis of Routine Surveillance Data in Four European Countries

Tristan Delory[1,2*], Julien Arino[3], Paul-Emile Haÿ[4], Vincent Klotz[4], Pierre-Yves Boëlle[1]

[1] Sorbonne Université, INSERM, Institut Pierre Louis d'Épidémiologie et de Santé Publique, IPLESP, F-75012, Paris, France. [2]Centre Hospitalier Annecy Genevois, France. [3]Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada. [4]Groupe Colisee.

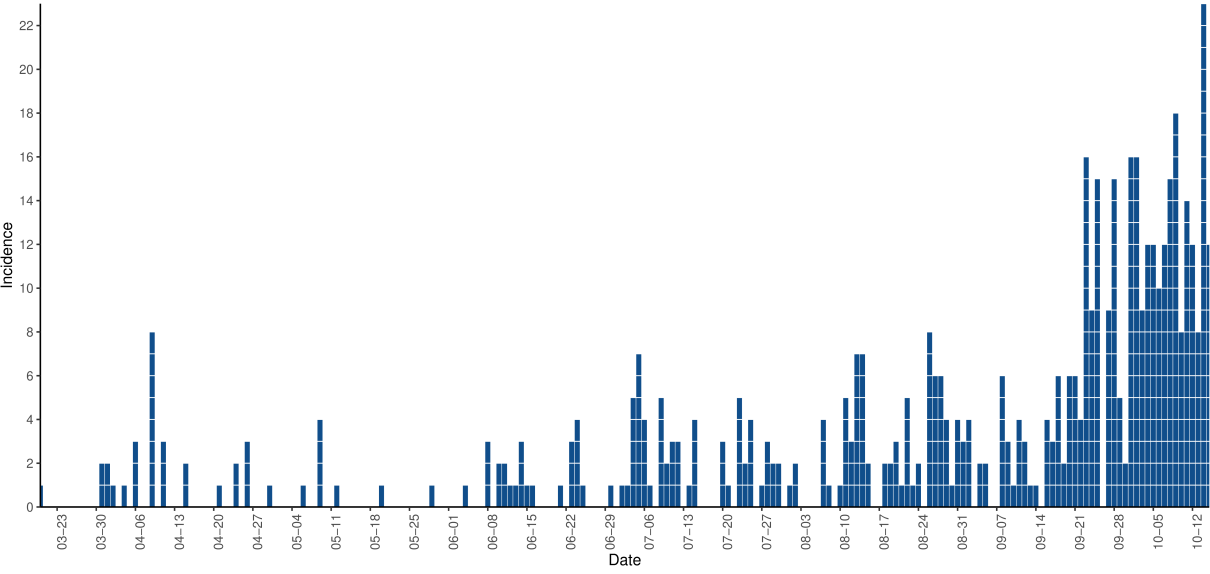
Table 1. Effect of vaccination scaling-up on the probability of successful viral introduction.

Period	Failed N = 136	Successful N = 366	aOR*	95%CI	P-value
Before vaccination	94 (69.1%)	311 (85.0%)	Ref		
January 15 to January 31	12 (8.8%)	37 (10.1%)	0.89	0.42 – 1.92	0.770
February 01 to February 15	17 (12.5%)	14 (3.8%)	0.23	0.10 – 0.52	<0.001
February 16 to February 28	13 (9.6%)	4 (1.1%)	0.08	0.02 - 0.29	<0.001

* Adjusted on study period, country, staffing ratio, cumulative attack rate at onset of introduction, and number of PCR per 1000-residents or 1000-staff members, at onset of introduction, and nursing home maximal capacity.

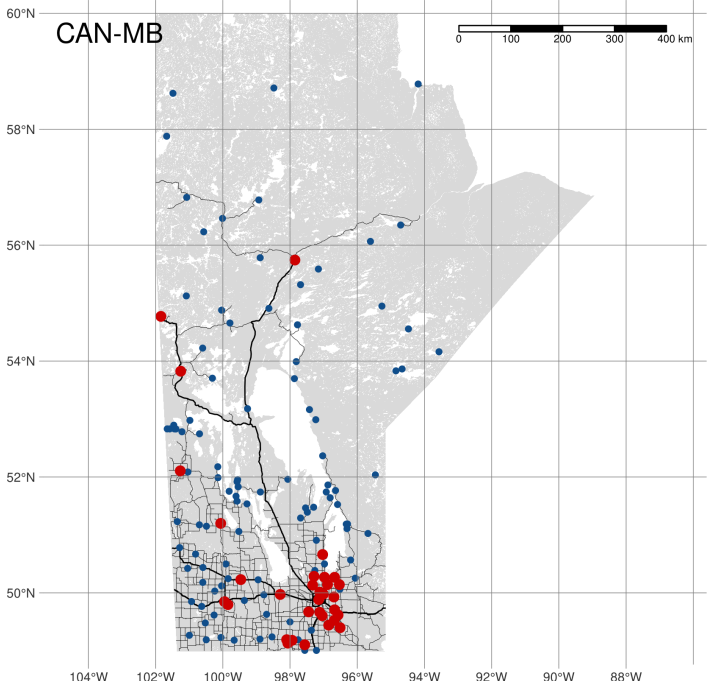
Second piece of evidence

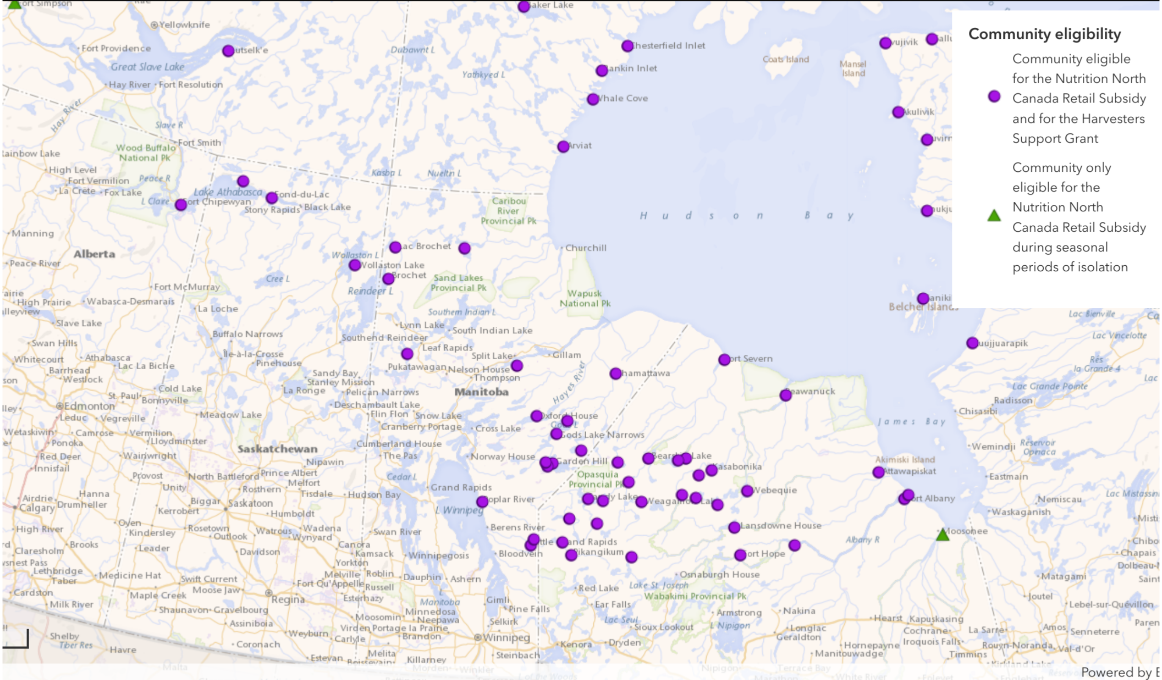
The start of an outbreak can be extremely slow, with very few cases for quite a while


Incidence
0
2
4
6
8
10
12
14
16
18
20
22
03–23
03–30
04–06
04–13
04–20
04–27
05–04
05–11
05–18
05–25
06–01
06–08
06–15
06–22
06–29
07–06
07–13
07–20
07–27
08–03
08–10
08–17
08–24
08–31
09–07
09–14
09–21
09–28
10–05
10–12
Date

Why this is relevant

Far from the only reason, but as an example: Canada has remote/isolated communities that are vulnerable to introductions of pathogens


CAN-MB
0
100
200
300
400 km
60°N
58°N
56°N
54°N
52°N
50°N
104°W
102°W
100°W
98°W
96°W
94°W
92°W
90°W
88°W


Community eligibility
Community eligible for the Nutrition North Canada Retail Subsidy and for the Harvesters Support Grant
Community only eligible for the Nutrition North Canada Retail Subsidy during seasonal periods of isolation
Hudson Bay
James Bay
Alberta
Saskatchewan
Manitoba
Yellowknife
Fort Providence
Great Slave Lake
Hay River
Fort Resolution
Fort Smith
Wood Buffalo National Pk
Lake Athabasca
Fond-du-Lac
Stony Rapids
Black Lake
Wollaston Lake
Reindeer L
Sand Lakes Provincial Pk
Caribou River Provincial Pk
Churchill
Wapusk National Pk
Arviat
Whale Cove
Rankin Inlet
Chesterfield Inlet
Baker Lake
Coats Island
Mansel Island
Belcher Islands
Lynn Lake
South Indian Lake
Southern Indian L
Leaf Rapids
Split Lake
Gillam
Pukatawagan
Nelson House
Thompson
Fort Severn
Peawanuck
Attawapiskat
Fort Albany
Moosonee
Kasabonika
Webequie
Lansdowne House
Fort Hope
Oxford House
Gods Lake Narrows
Garden Hill
Norway House
Cross Lake
Poplar River
Berens River
Bloodvein
Pikangikum
Red Lake
Sioux Lookout
Kenora
Dryden
Winnipeg
Brandon
Regina
Saskatoon
Prince Albert
Edmonton
Calgary
Red Deer
Lethbridge
Medicine Hat
Fort McMurray
La Loche
Flin Flon
The Pas
L Winnipeg
Kuujjuarapik
Radisson
Chisasibi
Wemindji
Eastmain
Waskaganish
Nemiscau
Timmins
Cochrane
Kapuskasing
Hearst
Rouyn-Noranda
Val-d'Or
Powered by E

Northern Manitoba chiefs call for immediate federal action on health-care crisis

Recent deaths linked to inadequate medical care include mother of 5 from Manto Sipi Cree Nation, chief says

CBC News · Posted: Apr 03, 2023 3:20 PM CDT | Last Updated: April 3, 2023

'A lengthy process to get help here'

Wasagamack is one of four First Nations communities that make up Island Lake, an area in northeastern Manitoba dotted with hundreds of small islands.

Island Lake has a population of at least 15,000, according to Scott Harper, the grand chief of Anisininew Okimawin, which represents the four communities.

Despite having a population roughly the size of Thompson, and having diabetes and hospitalization rates well above provincial averages, Island Lake has no hospital of its own. The region is accessible only by air, boat and an unreliable winter road.

The nursing station in Wasagamack First Nation, which has about 2,300 people, according to federal government data, typically operates short-staffed, with only two or three of five registered nurses working on any given rotation and a fly-in doctor who comes weekly.

For First Nation and Métis Communities

Remote describes a **geographical area** where a community is **located over 350 km** from the **nearest service centre having year-round access** by land and/or water routes normally used in all weather conditions

Isolated means a **geographical area** that has **scheduled flights** and good telephone service, but is **without year-round access** by land and/or water normally used in all weather conditions

Remote-Isolated means a **geographic area** that has **neither scheduled flights nor year-round access** by land and/or water routes normally that can be used in all weather conditions, irrespective of the level of telephone and radio service available

For Inuit communities

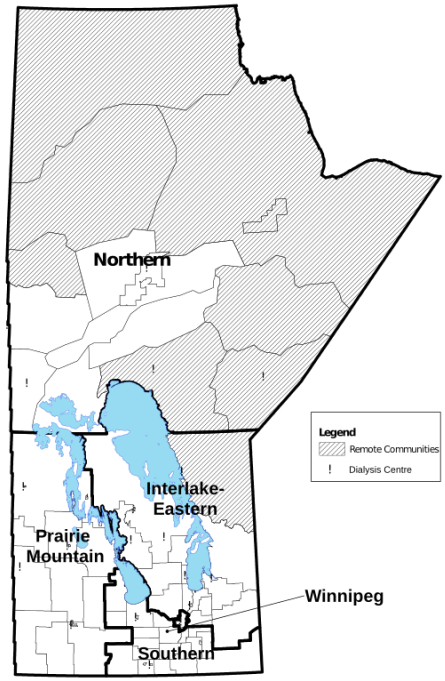
Inuit Communities to be referred to as **Inuit Nunangat**, not remote and isolated communities to respect the unique language and culture of Inuit regions, as well as the common challenges in social determinants of health, access to care, and infrastructure found across all Inuit communities

MB remote communities

Remote communities *are communities in Manitoba that* ***do not have permanent road access*** *(i.e., no all-weather road), are* ***more than a four-hour drive*** *from a major rural hospital (and a dialysis unit),* ***or have rail or fly-in access only****. This includes Norway House, Lynn Lake, Leaf Rapids, Gillam, and Cross Lake. If most communities in a health district are designated as "remote", the entire district is designated as "remote". In Manitoba, remote districts include:*

- *Northern Health Region: NO23, NO13, NO25, NO16, NO22, NO26, NO28, NO31, and*
- *Interlake-Eastern Health Region: IE61.*

Chartier M, Dart A, Tangri N, Komenda P, Walld R, Bogdanovic B, Burchill C, Koseva I, McGowan K, Rajotte L. Care of Manitobans Living with Chronic Kidney Disease. Winnipeg, MB. Manitoba Centre for Health Policy, December 2015


Northern
Interlake-
Eastern
Prairie
Mountain
Winnipeg
Southern
Legend
Remote Communities
Dialysis Centre

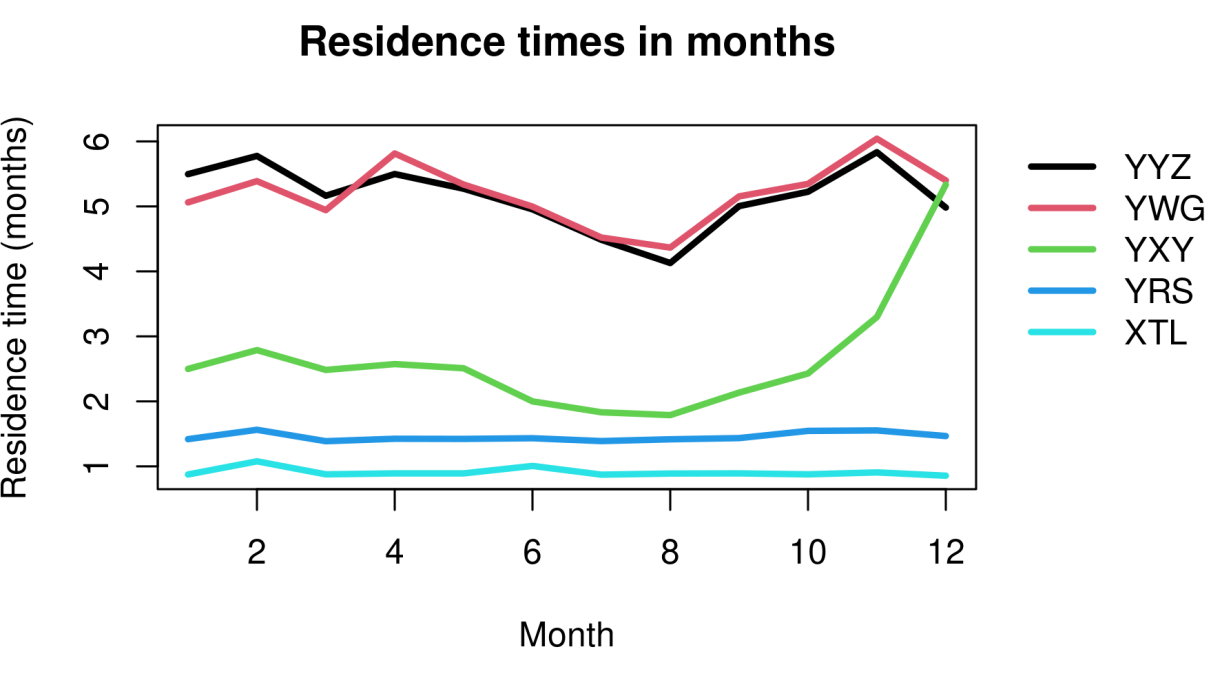
Travel to/from remote or isolated communities

How do you think this compares to travel in non-remote/isolated communities ?

Residence time (the lake ecology version): theoretic time an average water or comparable molecule spends in a lake, considering inflow into and outflow from the lake

Think of residence times in these communities: what is the average time a person spends in a remote or isolated community before leaving it?

The **residence time in a location** is the total number of trips inbound into and outbound from location over a duration of time (1 month here) divided by the normal population in the location


Residence times in months
Residence time (months)
1
2
3
4
5
6
2
4
6
8
10
12
Month
YYZ
YWG
YXY
YRS
XTL

The paradox of travel to/from remote/isolated communities

Travel volumes small but movement rates high

ICs are highly connected to the urban centre(s) they are subordinated to

Further reinforced in Winnipeg by urban indigenous population (102,075 or 12.45% of metro population), meaning many family connections exist


Why incorporate stochasticity?
Continuous time Markov chains
Branching process approximations of CTMC

Continuous time Markov chains

From discrete to continuous time

Discrete-Time Markov Chains (DTMCs)

A system transitions between states at fixed, discrete time steps ($n = 0, 1, 2, \ldots$)

- The future depends only on the present state (Markov Property)
- Governed by a **transition probability matrix** P, where P_{ij} is the probability of moving from state i to j in one step

Continuous-Time Markov Chains (CTMCs)

A system can transition between states at **any point in time**

- Time spent in a state is a **continuous random variable**
- The "holding time" in any state i follows an **exponential distribution** parameterised by an *exit rate* q_i
- This is a direct consequence of the Markov Property being applied to continuous time (exponential is only continuous distribution that is "memoryless")

Transition rates

Dynamics of a CTMC defined by **transition rates**, not probabilities

Definition 1 (Transition Rates)

For two states $i \neq j$, the rate $q_{ij} \geq 0$ is the instantaneous rate of transition from state i to state j

- For a small time interval Δt, the probability of transitioning from i to j is approximately $q_{ij}\Delta t$
- Total **exit rate** from state i is $q_i = \sum_{j \neq i} q_{ij}$
- Time spent in state i is an exponential random variable $T_i \sim \mathrm{E}(q_i)$

The generator matrix

Generator matrix (Q-Matrix)

Collect all transition rates into a single matrix Q

- Off-diagonal: $Q_{ij} = q_{ij}$ for $i \neq j$ (The rate of going from i to j)
- Diagonal: $Q_{ii} = -q_i = -\sum_{j \neq i} q_{ij}$. (The negative of the total exit rate from i)

A key property is that all rows of Q sum to zero: $\sum_j Q_{ij} = 0$.

Kolmogorov equations

Let $P(t)$ be the matrix where $P_{ij}(t) = \mathbb{P}(X(t) = j | X(0) = i)$. How does $P(t)$ evolve over time?

Kolmogorov forward equations

Describes rate of change of probability of ending up in a target state j

$$\frac{d}{dt}P(t) = P(t)Q$$

In element form:

$$P'_{ij}(t) = \sum_k P_{ik}(t)Q_{kj}$$

Solution to the KFE

The solution is the **matrix exponential**

$$P(t) = e^{tQ} = \sum_{k=0}^{\infty} \frac{(tQ)^k}{k!}$$

The generator matrix Q “generates” the process’s evolution

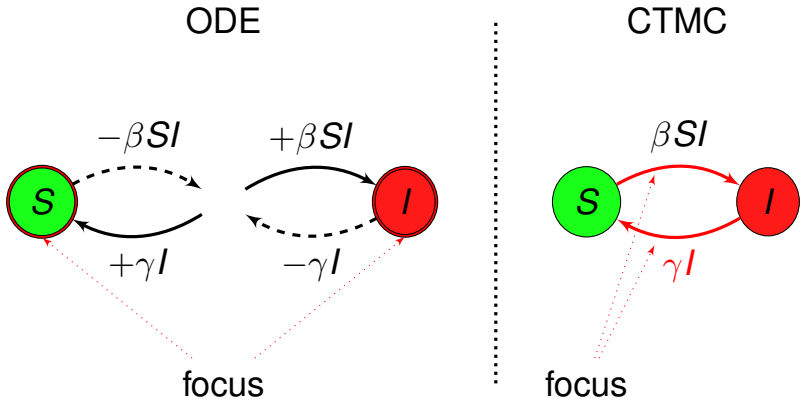
Continuous time Markov chains

Converting your compartmental ODE model to CTMC

Easy as π :)

- Compartmental ODE model focuses on flows into and out of compartments
- ODE model has as many equations as there are compartments
- Compartmental CTMC model focuses on transitions
- CTMC model has as many transitions as there are arrows between (or into or out of) compartments

ODE to CTMC : focus on different components



SIS without demography

Transition	Effect	Weight	Probability
$S \to S-1,\ I \to I+1$	new infection	βSI	$\frac{\beta SI}{\beta SI+\gamma I}$
$S \to S+1,\ I \to I-1$	recovery of an infectious	γI	$\frac{\gamma I}{\beta SI+\gamma I}$

States are S, I

SIS with demography

Transition	Effect	Weight	Probability
$S \to S+1$	birth of a susceptible	b	$\frac{b}{b+d(S+I)+\beta SI+\gamma I}$
$S \to S-1$	death of a susceptible	dS	$\frac{dS}{b+d(S+I)+\beta SI+\gamma I}$
$S \to S-1,\ I \to I+1$	new infection	βSI	$\frac{\beta SI}{b+d(S+I)+\beta SI+\gamma I}$
$I \to I-1$	death of an infectious	dI	$\frac{dI}{b+d(S+I)+\beta SI+\gamma I}$
$S \to S+1,\ I \to I-1$	recovery of an infectious	γI	$\frac{\gamma I}{b+d(S+I)+\beta SI+\gamma I}$

States are S, I

Kermack & McKendrick model

Transition	Effect	Weight	Probability
$S \to S-1,\ I \to I+1$	new infection	βSI	$\frac{\beta SI}{\beta SI+\gamma I}$
$I \to I-1,\ R \to R+1$	recovery of an infectious	γI	$\frac{\gamma I}{\beta SI+\gamma I}$

States are S, I, R

Continuous time Markov chains

Gillespie's algorithm

- A.k.a. the stochastic simulation algorithm (SSA)
- Derived in 1976 by Daniel Gillespie
- Generates possible solutions for CTMC
- Extremely simple, so worth learning how to implement; there are however packages that you can use (see later)

Gillespie's algorithm

Suppose system has state $\mathbf{x}(t)$ with initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$ and *propensity functions* a_i of elementary reactions

set $t \leftarrow t_0$ and $\mathbf{x}(t) \leftarrow \mathbf{x}_0$
while $t \leq t_f$

- $\xi_t \leftarrow \sum_j a_j(\mathbf{x}(t))$
- Draw τ_t from $T \sim \mathcal{E}(\xi_t)$
- Draw ζ_t from $\mathcal{U}([0,1])$
- Find r, smallest integer s.t. $\sum_{k=1}^{j} a_k(\mathbf{x}(t)) > \zeta_t \sum_j a_j(\mathbf{x}(t)) = \zeta_t \xi_t$
- Effect the next reaction (the one indexed r)
- $t \leftarrow t + \tau_t$

Drawing at random from an exponential distribution

If you do not have an exponential distribution random number generator.. We want τ_t from $T \sim \mathcal{E}(\xi_t)$, i.e., T has probability density function

$$f(x, \xi_t) = \xi_t e^{-\xi_t x} \mathbf{1}_{x \geq 0}$$

Use cumulative distribution function $F(x, \xi_t) = \int_{-\infty}^{x} f(s, \xi_t)\, ds$

$$F(x, \xi_t) = (1 - e^{-\xi_t x}) \mathbf{1}_{x \geq 0}$$

which has values in $[0, 1]$. So draw ζ from $\mathcal{U}([0, 1])$ and solve $F(x, \xi_t) = \zeta$ for x

$$\begin{aligned} F(x, \xi_t) = \zeta &\Leftrightarrow 1 - e^{-\xi_t x} = \zeta \\ &\Leftrightarrow e^{-\xi_t x} = 1 - \zeta \\ &\Leftrightarrow \xi_t x = -\ln(1 - \zeta) \\ &\Leftrightarrow \boxed{x = \frac{-\ln(1 - \zeta)}{\xi_t}} \end{aligned}$$

Gillespie's algorithm (SIS model with only I eq.)

set $t \leftarrow t_0$ and $I(t) \leftarrow I(t_0)$
while $t \leq t_f$
- $\xi_t \leftarrow \beta(P^\star - i)i + \gamma i$
- Draw τ_t from $T \sim \mathcal{E}(\xi_t)$
- $v \leftarrow [\beta(P^\star - i)i, \xi_t]/\xi_t$
- Draw ζ_t from $\mathcal{U}([0,1])$
- Find *pos* such that $v_{pos-1} \leq \zeta_t \leq v_{pos}$
- switch *pos*
    - 1: New infection, $I(t+\tau_t) = I(t) + 1$
    - 2: End of infectious period, $I(t+\tau_t) = I(t) - 1$
- $t \leftarrow t + \tau_t$

Sometimes Gillespie goes bad

- Recall that the inter-event time is exponentially distributed
- Critical step of the Gillespie algorithm:
  - $\xi_t \leftarrow$ weight of all possible events (*propensity*)
  - Draw τ_t from $T \sim \mathcal{E}(\xi_t)$
- So the inter-event time $\tau_t \to 0$ if ξ_t becomes very large for some t
- This can cause the simulation to grind to a halt

Example: a birth and death process

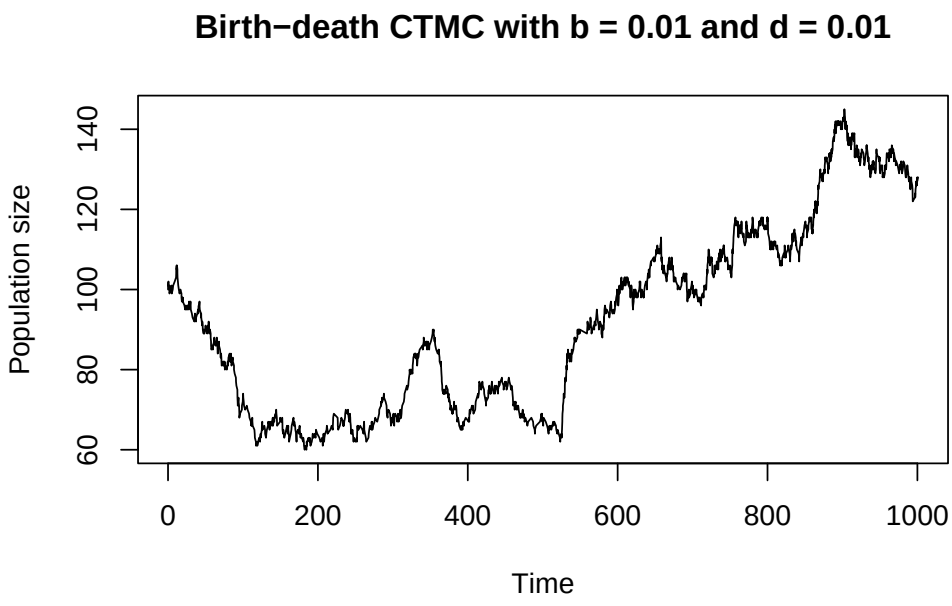
- Individuals born at *per capita* rate b
- Individuals die at *per capita* rate d
- Let's implement this using classic Gillespie

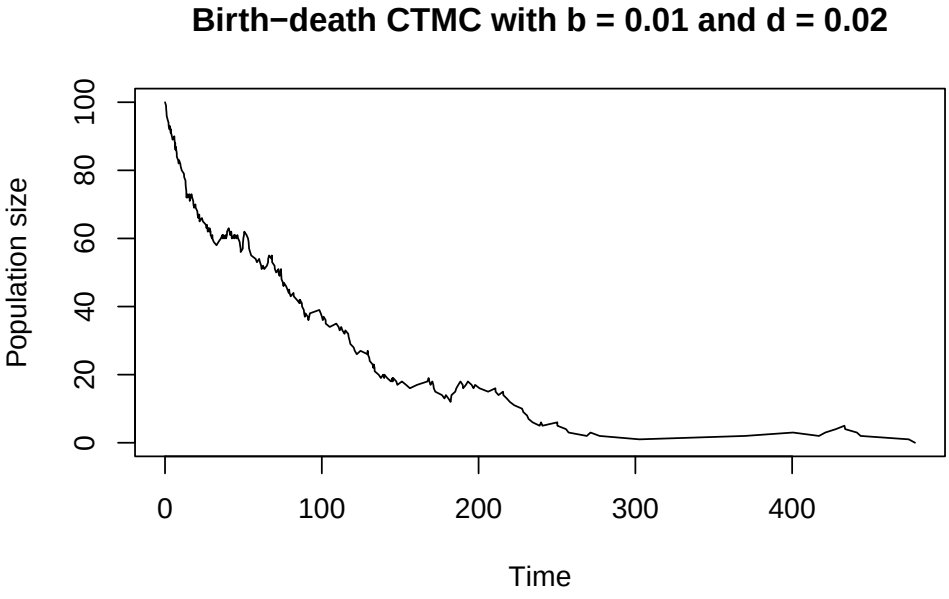
(See `simulate_birth_death_CTMC.R` on course GitHub repo)

Gillespie's algorithm (birth-death model)

set $t \leftarrow t_0$ and $N(t) \leftarrow N(t_0)$
while $t \leq t_f$
- $\xi_t \leftarrow (b+d)N(t)$
- Draw τ_t from $T \sim \mathcal{E}(\xi_t)$
- $v \leftarrow [bN(t), \xi_t] / \xi_t$
- Draw ζ_t from $\mathcal{U}([0,1])$
- Find *pos* such that $v_{pos-1} \leq \zeta_t \leq v_{pos}$
- switch *pos*
    - 1: Birth, $N(t+\tau_t) = N(t) + 1$
    - 2: Death, $N(t+\tau_t) = N(t) - 1$
- $t \leftarrow t + \tau_t$

```
birth_death_CTMC = function(b = 0.01, d = 0.01) {
  t_0 = 0    # Initial time
  N_0 = 100  # Initial population

  # Vectors to store time and state. Initialise with initial condition.
  t = t_0
  N = N_0

  t_f = 1000  # Final time

  # Track the current time and state (could just check last entry in t
  # and N, but will take more operations)
  t_curr = t_0
  N_curr = N_0
  while (t_curr<=t_f) {
    xi_t = (b+d)*N_curr
    if (N_curr == 0) {
      break # Avoid error with rexp when xi_t = 0
```

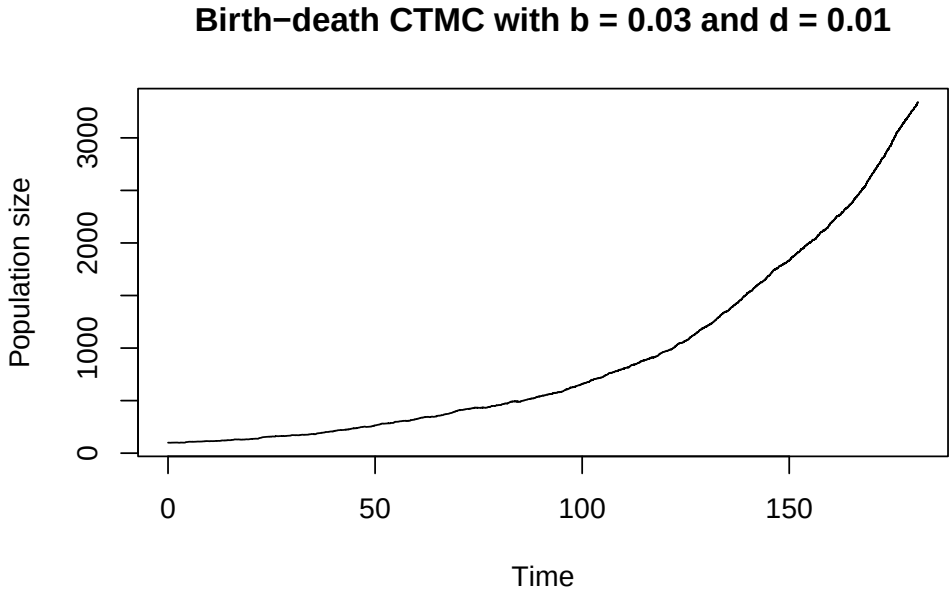

Birth−death CTMC with b = 0.01 and d = 0.01
Population size
60
80
100
120
140
0
200
400
600
800
1000
Time


Birth−death CTMC with b = 0.01 and d = 0.02
Population size
0
20
40
60
80
100
0
100
200
300
400
Time

$b = 0.03$ & $d = 0.01$...

We want to run the function with these parameter values but I know in advance this will not work well, so let's tweak the function a bit. We add a test:

```
if (t[length(t)]-t[(length(t)-1)] < 1e-8) {
      # If the time step is too small, stop the simulation
      message("Stopping simulation because time step is too small")
      break
    }
```

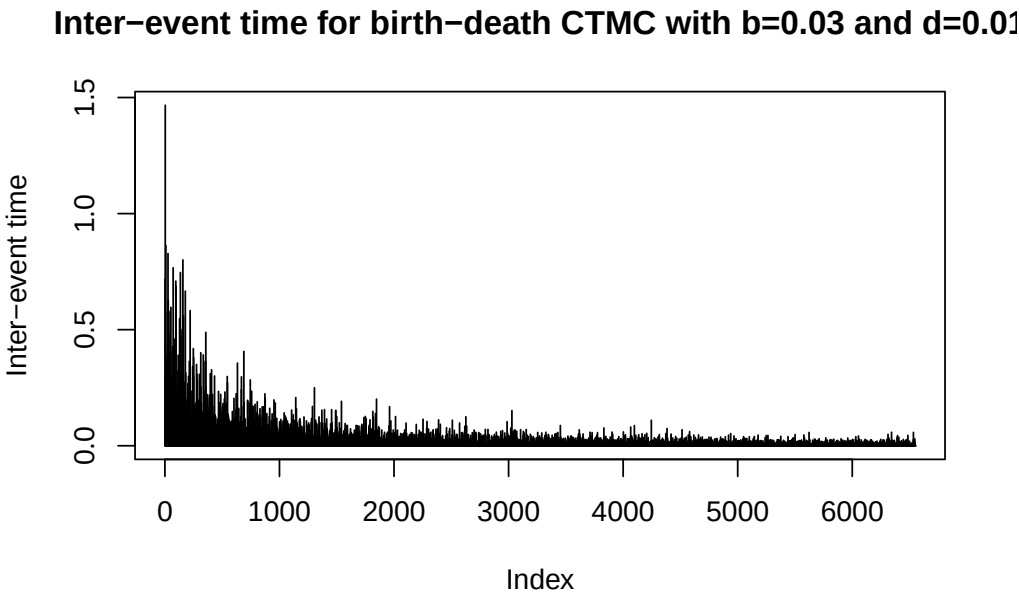

Birth−death CTMC with b = 0.03 and d = 0.01
Population size
0
1000
2000
3000
0
50
100
150
Time

Last one did not go well

- Wanted 1000 time units (days?)
- Interrupted at 181.1034837 because of the test
  (Slide with $b < d$: sim stopped because the population went extinct, I did not stop it!)
- At stop time
  - $N = 3338$
  - $|N| = 6549$ (and $|t|$ as well, of course!)
  - time was moving slowly

```
tail(diff(results$t))
```

```
## [1] 1.154357e-02 1.856236e-03 1.200474e-02 3.011079e-02 1.526377e-02
## [6] 3.306244e-07
```

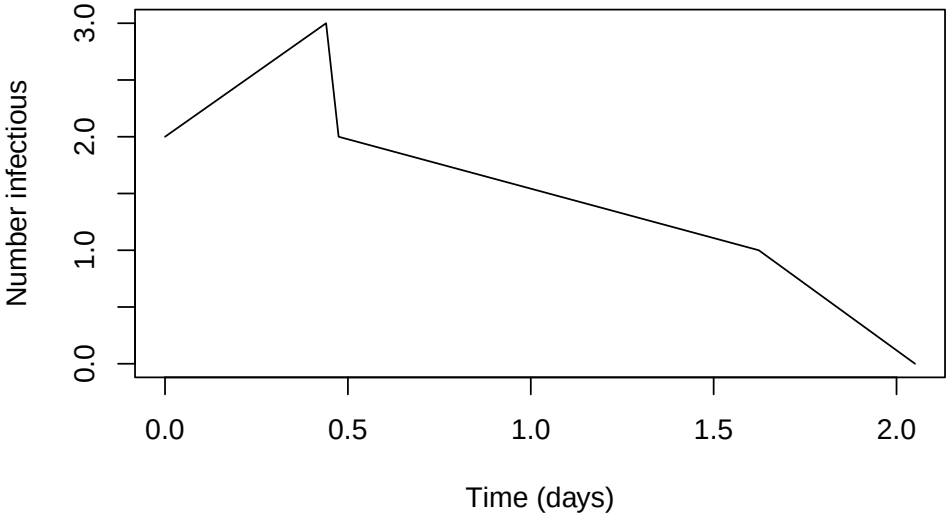

Inter−event time for birth−death CTMC with b=0.03 and d=0.01
Inter−event time
0.0
0.5
1.0
1.5
0
1000
2000
3000
4000
5000
6000
Index

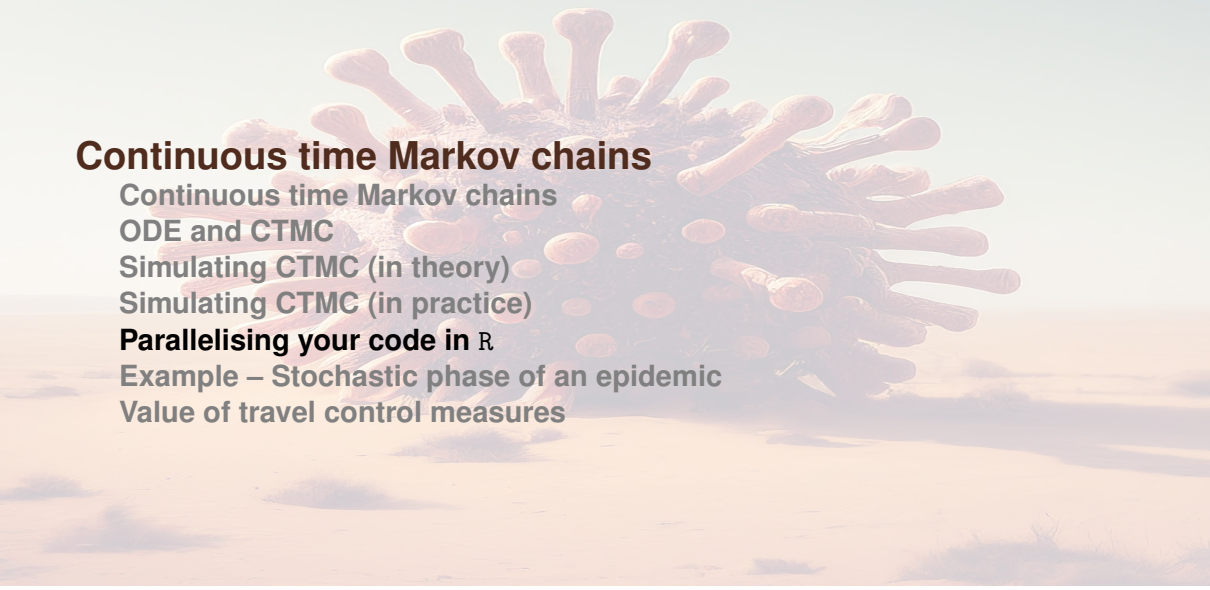
Continuous time Markov chains

Tau-leaping (and packages) to the rescue!

- *Approximation* method (compared to classic Gillespie, which is exact)
- Roughly: consider "groups" of events instead of individual events
- Good news: `GillespieSSA2` and `adaptivetau`, two standard packages for SSA in `R`, implement tau leaping

```r
library(GillespieSSA2)
Pop <- 1000
I_0 <- 2
IC <- c(S = (Pop-I_0), I = I_0)
gamma = 1/3
# R0=beta/gamma*S0, so beta=R0*gamma/S0
beta = as.numeric(1.5*gamma/IC["S"])
params <- c(gamma = gamma, beta = beta)
t_f = 100
reactions <- list(
  reaction("beta*S*I", c(S=-1,I=+1), "new_infection"),
  reaction("gamma*I", c(S=+1,I=-1), "recovery")
)
set.seed(NULL)


sol <- ssa(
  initial.state = IC,
```


Number infectious
0.0
1.0
2.0
3.0
0.0
0.5
1.0
1.5
2.0
Time (days)



Continuous time Markov chains

Parallelisation

To see multiple realisations: good idea to parallelise, then interpolate results. Write a function, e.g., `run_one_sim` that .. runs one simulation

Use some parallelisation mechanisms to run `run_one_sim` in parallel. One easy way to do it is to use a parallel version of `lapply`, which applies a function to a list

Here, I am showing parallelisation using a recent-ish package called `future` (and `future.apply`, which contains the relevant `lapply` equivalent)

I am also illustrating another SSA library that I find less tricky on Windows because the reactions are not precompiled: `adaptivetau`

```r
library(adaptivetau)
library(future.apply)
# It is useful to have the transitions, rates and
# names defined in a function
CTMC_SIS <- function() {
  # Define transitions for adaptivetau
  transitions <- list(
    c(S = -1, I = +1),  # new_infection
    c(S = +1, I = -1)   # recovery
  )
  # Define rate function
  rates <- function(x, params, t) {
    c(
      params[["beta"]] * x["S"] * x["I"],
      params[["gamma"]] * x["I"]
    )
  }
```

```
  event_names = c("new_infection", "recovery")
  return(list(transitions = transitions,
              rates = rates,
              event_names = event_names))
}

run_one_sim = function(CTMC, params) {
    IC <- c(S = (params$Pop-params$I_0),
            I = params$I_0)
    set.seed(NULL)
    sol <- ssa.exact(
        init.values = IC,
        transitions = CTMC$transitions,
        rateFunc = CTMC$rates,
        params = params,
        tf = params$t_f
    )
```

```
    # Interpolate result (just I will do)
    wanted_t =
      seq(from = 0, to = params$t_f, by = 0.01)
    interp_I = approx(x = sol[,"time"],
                      y = sol[,"I"],
                      xout = wanted_t)
    names(interp_I) = c("time", "I")
    sol$interp_I = interp_I
    # Return result
    return(sol)
}

# By default, use all available cores
plan(multisession)
## To use fewer workers, leaving one empty for
# instance
# plan(multisession, availableCores()-1)
```

```r
## To run sequentially
# plan(sequential)

# Set up parameters not needing computation
params <- list(gamma = 1/3,
               Pop = 1000,
               I_0 = 2,
               R0 = 1.5,
               t_f = 100, nb_sims = 50)
IC <- c(S = (params$Pop-params$I_0),
        I = params$I_0)
# R0=beta/gamma*S0, so beta=R0*gamma/S0
params =
  c(params,
    beta = as.numeric(params$R0*params$gamma /
                        IC["S"]))
# Run the simulation
```

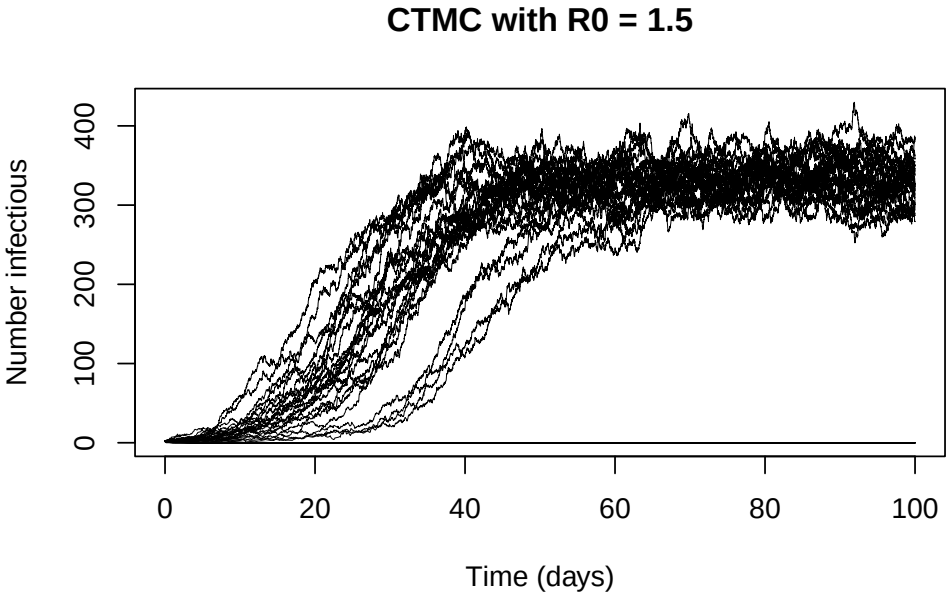
```
CTMC <- CTMC_SIS()
SIMS = future_lapply(
  X = 1:params$nb_sims,
  FUN =  function(x) run_one_sim(CTMC, params))
#  Liberate the workers!
stopCluster(cl)

## Error in stopCluster(cl):  could not find function "stopCluster"

# Find max y value for plot
y_max = max(unlist(lapply(SIMS, function(x) max(x$interp_I$I))),
            na.rm = TRUE)
# Now plot
plot(SIMS[[1]]$interp_I$time,
     SIMS[[1]]$interp_I$I,
     type = "l", lwd = 0.5,
     xlab = "Time (days)",
     ylab = "Number infectious",
```

```
     ylim = c(0, y_max),
     main = paste("CTMC with R0 =", params$R0))
for (i in 2:length(SIMS)) {
  lines(SIMS[[i]]$interp_I$time,
        SIMS[[i]]$interp_I$I,
        type = "l", lwd = 0.5)
}
```

Common part – The function we run I

Common part – The function we run II

```
run_one_sim = function(params) {
    IC <- c(S = (params$Pop-params$I_0), I = params$I_0)
    params_local <- c(gamma = params$gamma, beta = params$beta)
    reactions <- list(
        # propensity function effects name for reaction
        reaction("beta*S*I", c(S=-1,I=+1), "new_infection"),
        reaction("gamma*I", c(S=+1,I=-1), "recovery")
    )
    set.seed(NULL)
    sol <- ssa(
      initial_state = IC,
      reactions = reactions,
      params = params_local,
      method = ssa_exact(),
      final_time = params$t_f,
      log_firings = TRUE     # This way we keep track of events
```


CTMC with R0 = 1.5
Number infectious
0
100
200
300
400
0
20
40
60
80
100
Time (days)

Benefit of parallelisation

Run the parallel code for 100 sims between ‘tictoc::tic()‘ and ‘tictoc::toc()‘, giving ‘66.958 sec elapsed‘, then the sequential version

```
tictoc::tic()
SIMS = lapply(X = 1:params$number_sims,
              FUN =  function(x) run_one_sim(params))
tictoc::toc()
```

which gives ‘318.141 sec elapsed‘ on a 6C/12T Intel(R) Core(TM) i9-8950HK CPU @ 2.90GHz (4.75$\times$ faster) or ‘12.067 sec elapsed‘ versus ‘258.985 sec elapsed‘ on a 32C/64T AMD Ryzen Threadripper 3970X 32-Core Processor (21.46$\times$ faster !)

Continuous time Markov chains

Bulletin of Mathematical Biology (2022) 84:128
https://doi.org/10.1007/s11538-022-01077-5

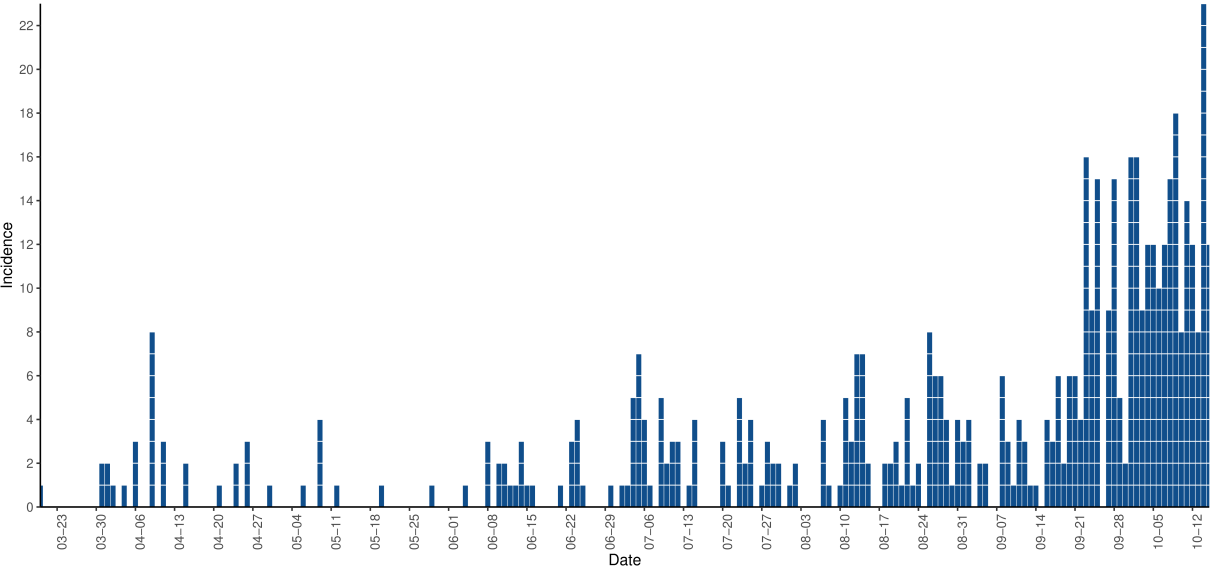
Society for Mathematical Biology

ORIGINAL ARTICLE

Effect of Movement on the Early Phase of an Epidemic

Julien Arino[1] · Evan Milliken[2]

Received: 6 July 2021 / Accepted: 29 August 2022 / Published online: 23 September 2022


Incidence
0
2
4
6
8
10
12
14
16
18
20
22
03–23
03–30
04–06
04–13
04–20
04–27
05–04
05–11
05–18
05–25
06–01
06–08
06–15
06–22
06–29
07–06
07–13
07–20
07–27
08–03
08–10
08–17
08–24
08–31
09–07
09–14
09–21
09–28
10–05
10–12
Date

Investigating outbreak types using a simple CTMC SIS

$$\mathbf{X}(t) = \left(S^A(t), I^A(t)\right)$$

CTMC $\mathbf{X}(t)$ characterized by transitions

Description	Transition	Rate
Infection	$\left(S^A, I^A\right) \to \left(S^A - 1, I^A + 1\right)$	$\beta^A S^A I^A$
Recovery	$\left(S^A, I^A\right) \to \left(S^A + 1, I^A - 1\right)$	$\gamma^A I^A$

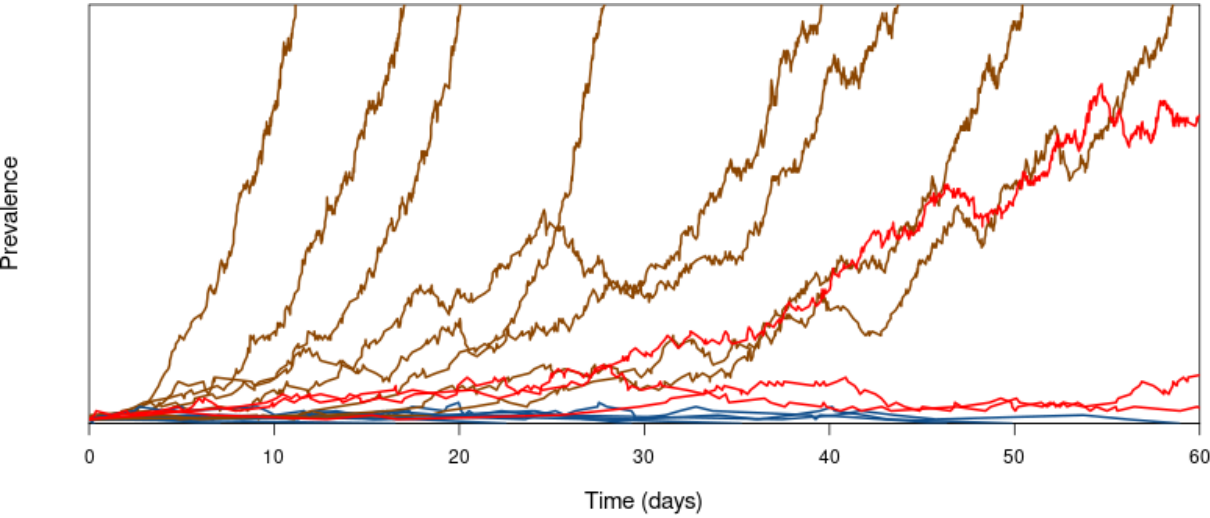
Investigating outbreak types using a simple CTMC SIS *with a twist*

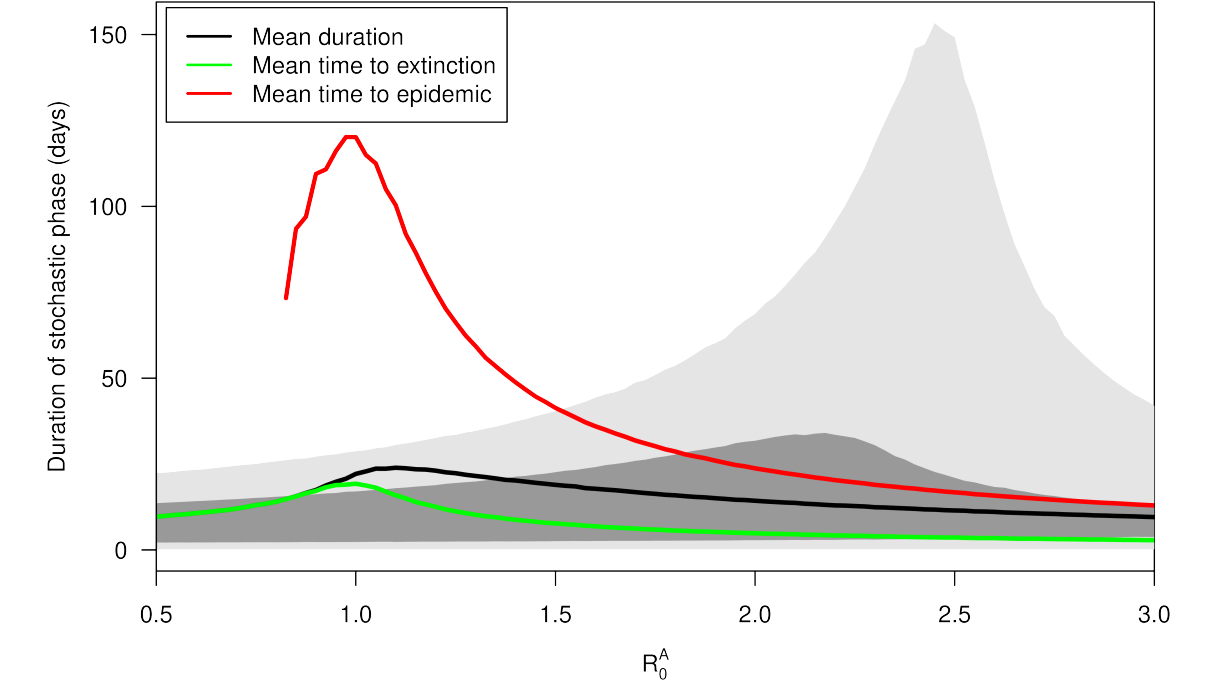
Regular chain of this type has $I = 0$ as sole absorbing state

We add another absorbing state: if $I = \hat{I}$, then the chain has *left* the stochastic phase and is in a quasi-deterministic phase with exponential growth

Doing this, time to absorption measures become usable additionally to first passage time ones

And the question becomes: how long does the chain "linger on" ("stutter") before it is absorbed? We define the inter-absorption trajectory as the stochastic phase

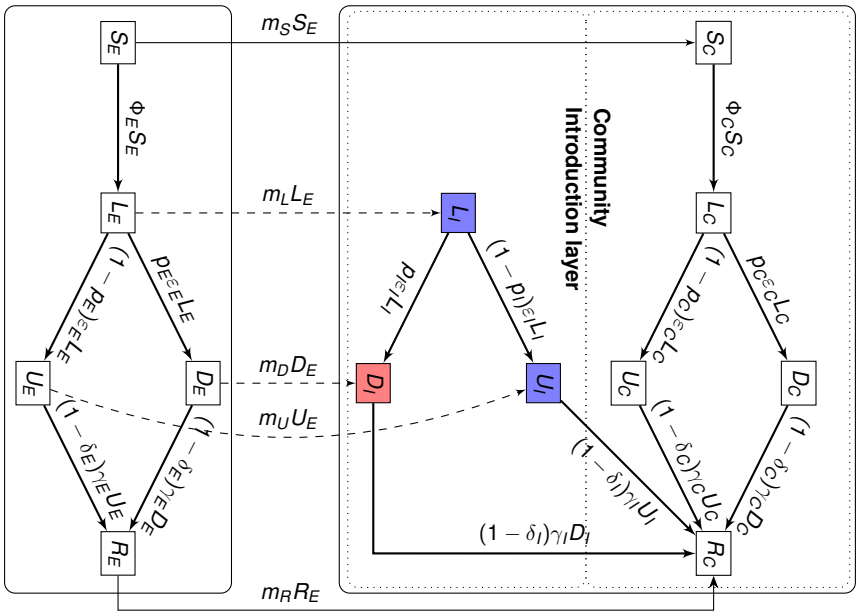

Prevalence
0
10
20
30
40
50
60
Time (days)

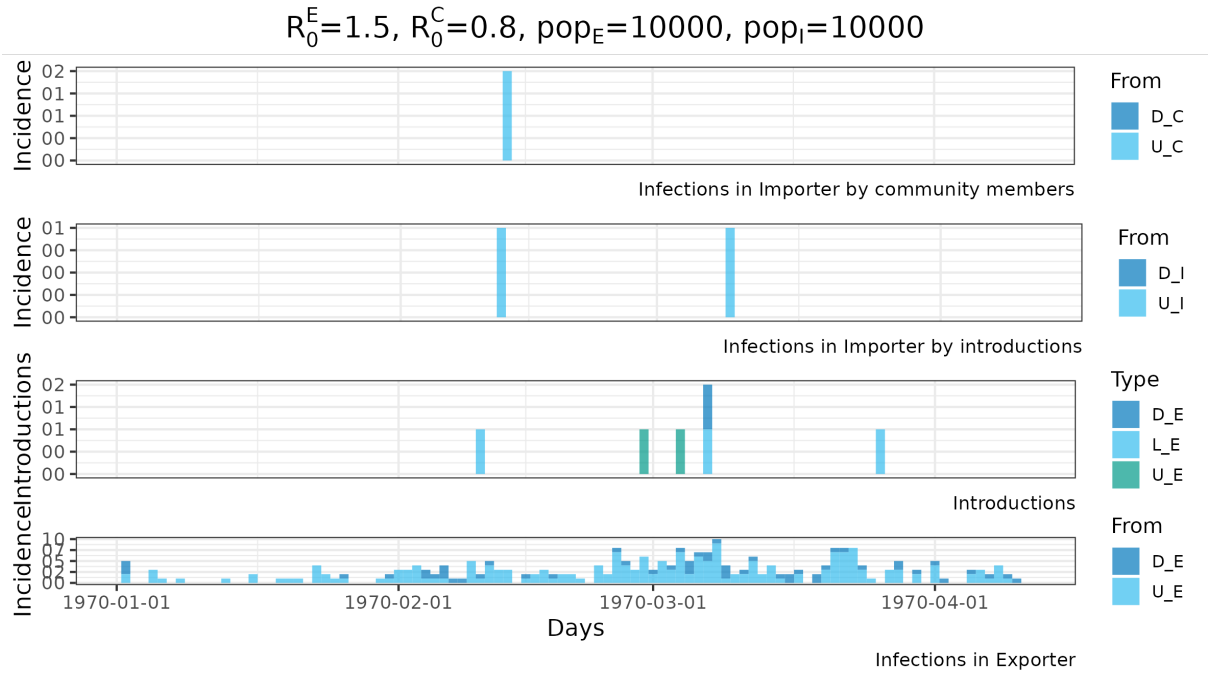

Mean duration
Mean time to extinction
Mean time to epidemic
Duration of stochastic phase (days)
150
100
50
0
0.5
1.0
1.5
2.0
2.5
3.0
R_0^A

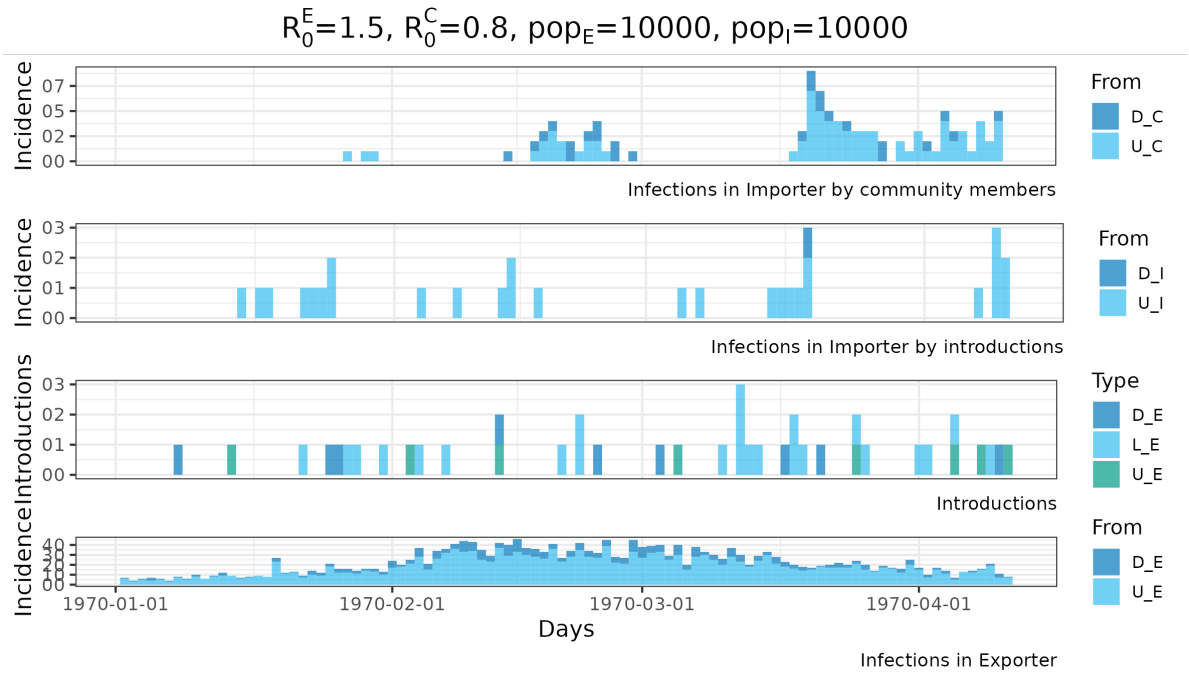
Problem of the value of the upper bound $\hat{l}$

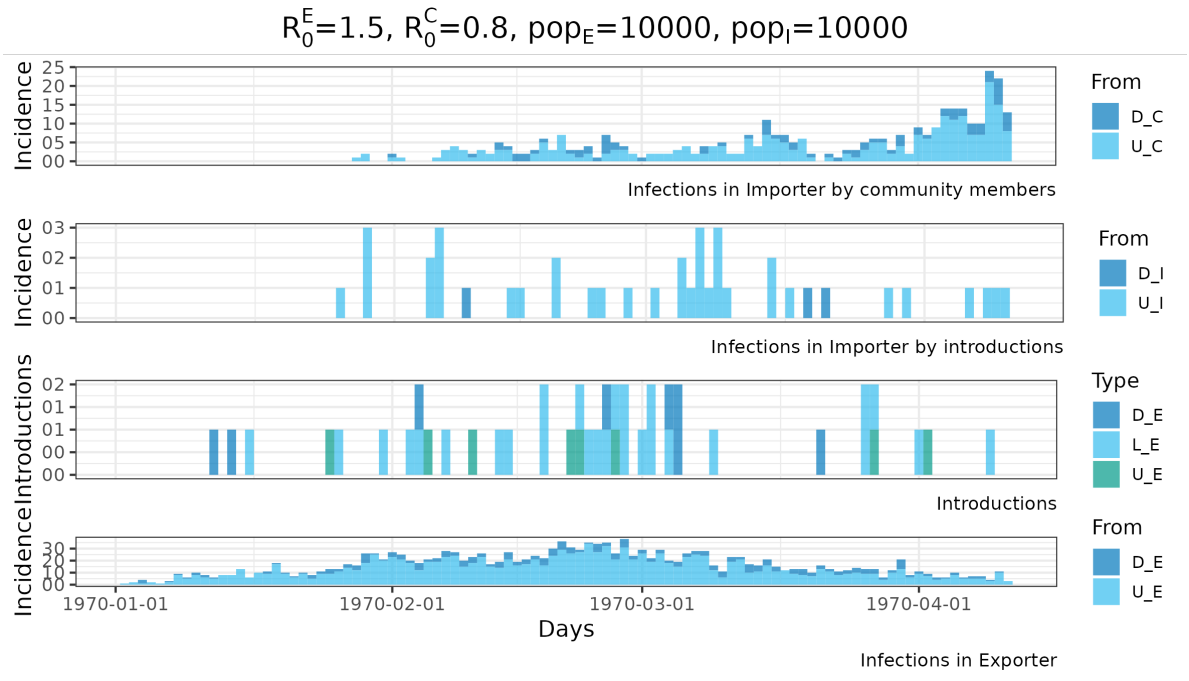
- Choose $\hat{l}$ too small and the stochastic phase will not last long
- Choose $\hat{l}$ too large and absorption will only be at the DFE
- So, how does one choose $\hat{l}$?
  - A formula of Whittle (1955)
  - Multitype branching process (MTBP)

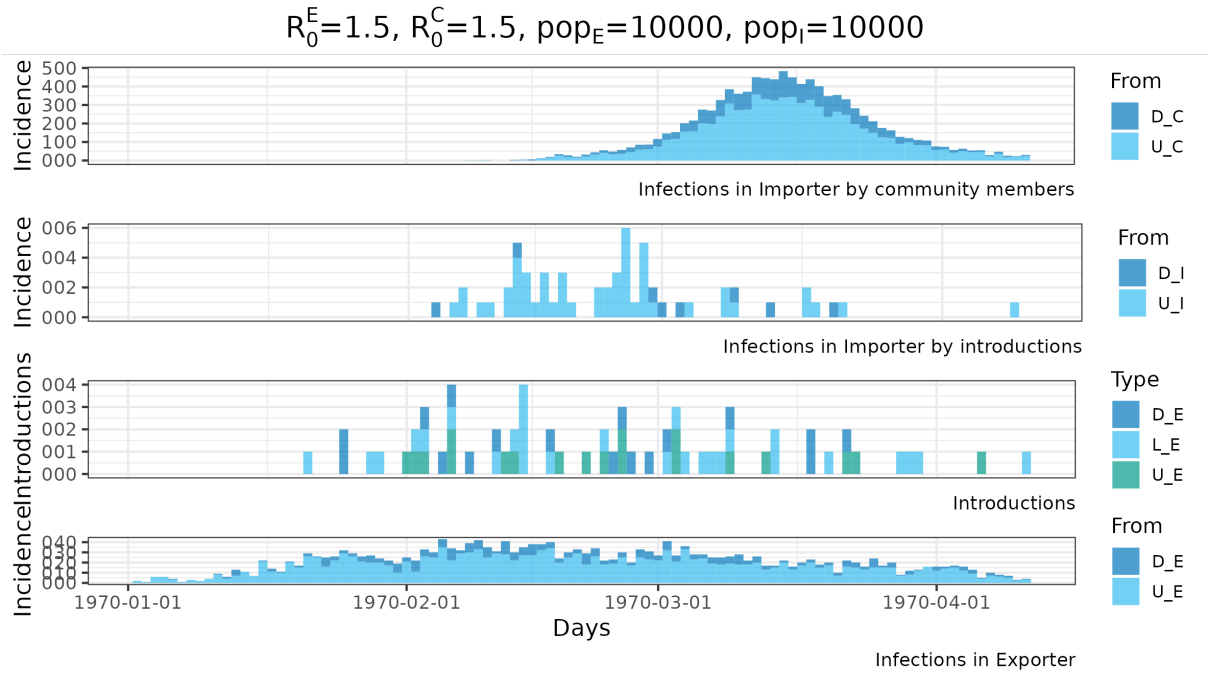
Continuous time Markov chains

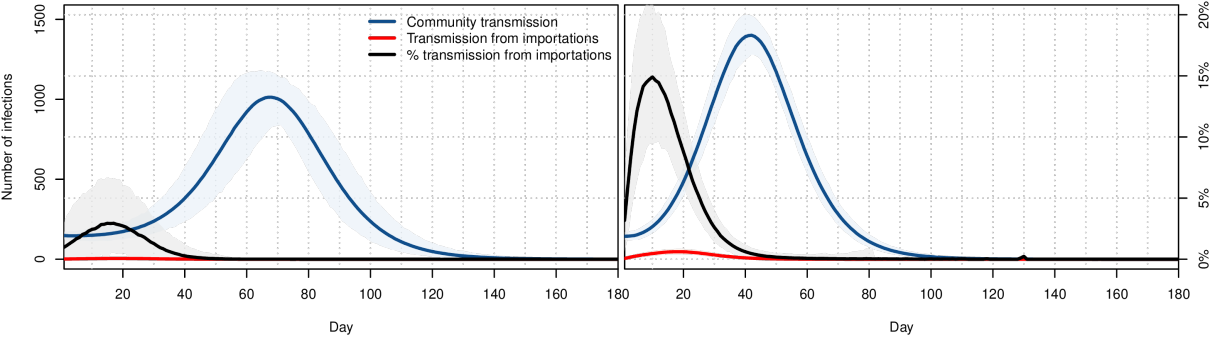

Exporter
Importer
Community
Introduction layer
$m_S S_E$
$m_L L_E$
$m_D D_E$
$m_U U_E$
$m_R R_E$
S_E
$\Phi_E S_E$
L_E
$p_{E\varepsilon E} L_E$
$(1-p_E)\varepsilon_E L_E$
U_E
D_E
$(1-\delta_E)\gamma_E U_E$
$(1-\delta_E)\gamma_E D_E$
R_E
L_I
$p_{I\varepsilon I} L_I$
$(1-p_I)\varepsilon_I L_I$
D_I
U_I
$(1-\delta_I)\gamma_I U_I$
$(1-\delta_I)\gamma_I D_I$
S_C
$\Phi_C S_C$
L_C
$p_{C\varepsilon C} L_C$
$(1-p_C)\varepsilon_C L_C$
U_C
D_C
$(1-\delta_C)\gamma_C U_C$
$(1-\delta_C)\gamma_C D_C$
R_C


R_0^E=1.5, R_0^C=0.8, pop_E=10000, pop_I=10000
Incidence
Infections in Importer by community members
From
D_C
U_C
Incidence
Infections in Importer by introductions
From
D_I
U_I
IncidenceIntroductions
Introductions
Type
D_E
L_E
U_E
IncidenceIntroductions
Infections in Exporter
From
D_E
U_E
1970-01-01
1970-02-01
1970-03-01
1970-04-01
Days


R_0^E=1.5, R_0^C=0.8, pop_E=10000, pop_I=10000
Incidence
07
05
02
00
Infections in Importer by community members
From
D_C
U_C
Incidence
03
02
01
00
Infections in Importer by introductions
From
D_I
U_I
Incidence Introductions
03
02
01
00
Introductions
Type
D_E
L_E
U_E
Incidence
40
30
20
10
00
1970-01-01
1970-02-01
1970-03-01
1970-04-01
Days
Infections in Exporter
From
D_E
U_E


R_0^E=1.5, R_0^C=0.8, pop_E=10000, pop_I=10000
Incidence
25
20
15
10
05
00
Infections in Importer by community members
From
D_C
U_C
Incidence
03
02
01
00
Infections in Importer by introductions
From
D_I
U_I
Incidence Introductions
02
01
01
00
00
Introductions
Type
D_E
L_E
U_E
Incidence
30
20
10
00
1970-01-01
1970-02-01
1970-03-01
1970-04-01
Days
Infections in Exporter
From
D_E
U_E

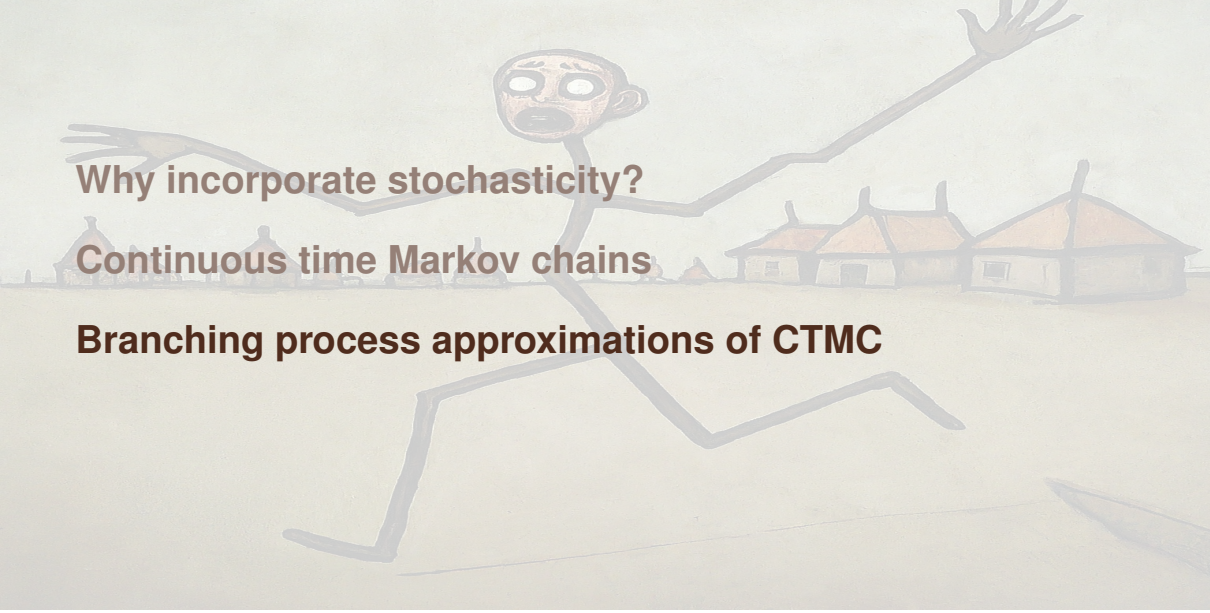

R_0^E=1.5, R_0^C=1.5, pop_E=10000, pop_I=10000
Incidence
500
400
300
200
100
000
Infections in Importer by community members
From
D_C
U_C
Incidence
006
004
002
000
Infections in Importer by introductions
From
D_I
U_I
IncidenceIntroductions
004
003
002
001
000
Introductions
Type
D_E
L_E
U_E
Incidence
040
030
020
010
000
1970-01-01
1970-02-01
1970-03-01
1970-04-01
Days
Infections in Exporter
From
D_E
U_E


Community transmission
Transmission from importations
% transmission from importations
Number of infections
0
500
1000
1500
0%
5%
10%
15%
20%
20
40
60
80
100
120
140
160
180
Day
Day

One last thought for the road

V. Chetail. Crisis without borders: What does international law say about border closure in the context of Covid-19? Frontiers in Political Science, 2 (12) (2020)

> *[..] a powerful expression of state's sovereignty, immigration control provides a typical avenue for governments to reassure their citizens and bolster a national sense of belonging, while providing an ideal scapegoat for their own failure or negligence.*



Why incorporate stochasticity?

Continuous time Markov chains

Branching process approximations of CTMC

What is a Branching Process?

The Core Idea

A branching process is a mathematical model for a population where individuals produce a random number of offspring and then die.

- Think of bacteria splitting, a virus spreading, or even the survival of family surnames.
- We start with an initial population, Z_0.
- Each individual in generation n produces a number of offspring for generation $n+1$.
- This "number of offspring" is a random variable. All individuals produce offspring according to the same probability distribution, independently of each other.

The Galton-Watson Process

Let Z_n be the size of the population in generation n. We typically start with $Z_0 = 1$. The population evolves according to the rule:

$$Z_{n+1} = \sum_{i=1}^{Z_n} X_{n,i}$$

- The term $X_{n,i}$ represents the number of offspring produced by the i-th individual in generation n.
- The variables $\{X_{n,i}\}$ are assumed to be **independent and identically distributed (i.i.d.)** integer-valued random variables.
- We call their common distribution $\{p_k\}_{k=0}^{\infty}$ the **offspring distribution**, where $p_k = P(X = k)$.

The Fundamental Questions

1. What is the long-term expected size of the population?
2. What is the probability that the population eventually dies out?

Mean Offspring

The fate of the population hinges on a single parameter: the mean of the offspring distribution

$$\mu = E[X] = \sum_{k=0}^{\infty} k \cdot p_k$$

Expected Population Size

Using the law of total expectation, we find the expected size of the next generation:

$$E[Z_{n+1}|Z_n] = E\left[\sum_{i=1}^{Z_n} X_{n,i} \middle| Z_n\right] = Z_n E[X] = Z_n \mu$$

Taking the expectation again, we get a simple recurrence:

$$E[Z_{n+1}] = \mu E[Z_n]$$

This implies:

The Three Regimes of Population Growth

The behavior of $E[Z_n] = Z_0 \mu^n$ suggests three distinct cases:

Subcritical ($\mu < 1$)	Critical ($\mu = 1$)	Supercritical ($\mu > 1$)
$E[Z_n] \to 0$. The population is expected to shrink. It goes extinct with probability 1.	$E[Z_n] = Z_0$. The population is expected to remain stable. Surprisingly, it still goes extinct with probability 1.	$E[Z_n] \to \infty$. The population is expected to grow exponentially. It has a non-zero probability of surviving forever.

Tool: The Probability Generating Function

To find the extinction probability, we need a powerful tool: the **probability generating function (PGF)** of the offspring distribution X.

$$G(s) = E[s^X] = \sum_{k=0}^{\infty} p_k s^k \quad \text{for } |s| \leq 1$$

Key Properties

- $G(1) = \sum p_k = 1$
- The mean can be found from the derivative: $G'(1) = \sum k p_k = \mu$.
- The PGF of Z_n is the n-th iterate of $G(s)$ with itself. If $G_n(s)$ is the PGF of Z_n, then $G_{n+1}(s) = G(G_n(s))$.

The Extinction Probability Equation

Let π_0 be the probability of eventual extinction, starting with $Z_0 = 1$.

$$\pi_0 = P(\text{population dies out}) = \lim_{n\to\infty} P(Z_n = 0)$$

Since $P(Z_n = 0) = G_n(0)$, and $G_{n+1}(0) = G(G_n(0))$, in the limit the extinction probability π_0 must satisfy the equation:

$$\pi_0 = G(\pi_0)$$

Theorem 2

*The extinction probability π_0 is the **smallest non-negative solution** to the equation $s = G(s)$.*

- If $\mu \leq 1$, the only solution in $[0, 1]$ is $s = 1$. So $\pi_0 = 1$.
- If $\mu > 1$, there is a unique solution in $[0, 1)$, which is the extinction probability $\pi_0 < 1$.

From Discrete to Continuous Time

Limitation of Galton-Watson

Generations don't happen in synchronized steps in the real world. Individuals give birth and die at random times.

This leads us to **Continuous-Time Markov Chains (CTMCs)**.

- The state of the system is the population size, $k \in \{0, 1, 2, \dots\}$.
- Instead of generations, we have transition rates:
  - λ_k: rate of birth when population is size k (moves to $k+1$).
  - δ_k: rate of death when population is size k (moves to $k-1$).
- Often, we assume these rates are linear: $\lambda_k = k\lambda$ and $\delta_k = k\delta$. This means individuals act independently.

Branching Process Approximation of a CTMC

The Key Insight

At the beginning of an outbreak (or for a very large population), the dynamics caused by a single individual are largely independent of others.

This allows us to approximate the start of a CTMC population process with a branching process.

Example: A Simple Epidemic (SIR Model)

- S: Susceptible, I: Infected, R: Recovered.
- An infected person meets others at a certain rate. If they meet a susceptible, a new infection may occur (an "offspring").
- The infected person recovers (or dies) at another rate, ending their infectious period.
- **Question:** How many new infections does a single infected person cause on average?

Case Study: The Basic Reproduction Number $\mathcal{R}_0$

Consider a single infected individual in a large population of susceptibles.

- Let β be the infection rate (rate of producing "offspring").
- Let γ be the recovery rate (rate of "dying").

The individual's infectious lifetime is an exponential random variable with mean $1/\gamma$.
The average number of secondary infections they cause is:

$$\mathcal{R}_0 = (\text{rate of infection}) \times (\text{average infectious period}) = \beta \times \frac{1}{\gamma} = \frac{\beta}{\gamma}$$

The Connection

$\mathcal{R}_0$ is precisely the **mean offspring number** μ for the embedded branching process that approximates the start of the epidemic.

Applying Branching Theory to Epidemics

The fate of the epidemic's initial phase is determined by $\mathcal{R}_0$:

- If $\mathcal{R}_0 \leq 1$ ($\mu \leq 1$): The number of infected individuals is a subcritical or critical process. The epidemic will die out with probability 1.
- If $\mathcal{R}_0 > 1$ ($\mu > 1$): The process is supercritical. There is a positive probability that the epidemic takes off and causes a major outbreak.

We can even calculate the probability of a major outbreak! It is $1 - \pi_0$, where π_0 is the extinction probability.
For this simple birth-death infection process, the PGF is $G(s) = \frac{\gamma}{\beta+\gamma} + \frac{\beta}{\beta+\gamma}s$.
Solving $s = G(s)$ gives the extinction probability:

$$\pi_0 = \frac{\gamma}{\beta} = \frac{1}{\mathcal{R}_0}$$

The probability of a major outbreak is $1 - 1/\mathcal{R}_0$.

Summary

- **Branching Processes** model populations with i.i.d. offspring generation.
- The fate of the population is determined by the **mean offspring number** μ. Extinction is certain if $\mu \leq 1$.
- The **extinction probability** π_0 can be calculated as the smallest non-negative fixed point of the probability generating function $G(s)$.
- The initial stages of many large-scale **Continuous-Time Markov Chains** can be approximated by a branching process.
- This allows us to apply the theory to real-world problems, like calculating an epidemic's **basic reproduction number** $\mathcal{R}_0$ and its probability of causing a major outbreak.

Bibliography I