Practical II: epidemiological surveillance and modelling (spatial)

3rd/4th February 2025

Pathogens move between different geographical units because of movements of animals or humans or other hosts. Spatial patterns of infectious disease transmission arise from heterogeneity in the landscape in which transmission occurs. In large and dense cities, for example, transmission intensity tends to be higher compared to rural areas. Spatial dispersal in ecology and epidemiology is often approximated by an exponential distribution, with the probability of transmission decaying with increasing distance $\propto exp(-d/a)$, where a is a shape parameter or Gaussian kernel and d is the distance between locations of interest.

For infectious diseases spreading among human populations, spatial dispersal rarely follows a continuous, uniform expansion. Instead, it is often shaped by the spatial organisation of cities and towns. Emerging infectious diseases, for example, tend to appear first in large urban agglomerations connected via air travel and subsequently spread to other major cities and rural counties, following patterns of local human movement. For the UK, for instance, the SARS-CoV-2 Omicron variant was first detected in London, a major hub for international travellers.

In this practical we will learn how to model continuous spatial spread using spatial kernels, before investigating the impact of human mobility on the spatial dispersal of infectious diseases. The code has been adapted from Chapter 11 of Epidemics: Models and Data Using R (https://link.springer.com/book/10.1007/978-3-031-12056-5) by Ottar Bjornstad, and additional code can be found here: https://github.com/objornstad/epimdr/blob/master/rcode/chapter11code.r (https://github.com/objornstad/epimdr/blob/master/rcode/chapter11code.r). There will be a 15 minute break after ca. 1h ½.

All the code for the practical is provided here, and you can copy & paste it in your own R script. Some of the questions will require you to make small changes in the code.

Load the packages of interest `epimdr`, `ggplot2`, `ncf`, `deSolve`, `dplyr`, `plot.matrix`, `reshape2`, `cowplot` using `library(<package_name>)`. You might have to install some of the packages yourself using `install.packages(<package_name>)`.

```
library(epimdr)
library(ggplot2)
library(ncf)
library(deSolve)
library(plot.matrix)
library(reshape2)
library(dplyr)
library(cowplot)
```

If you are unable to load the `epimdr` package using the code above, please re-install the htmltools package (a dependency of `epimdr`).

```
remove.packages('htmltools')
install.packages('htmltools')
```

Practical II Questions

Q1: What determines the spatial spread of infectious diseases?

Contrast spread of plant diseases vs. human infectious diseases. Here a visualisation of human movement patterns in the UK (https://www.science.org/cms/10.1126/science.abj0113/asset/c322ab29-d54d-42ef-8412-4e6789f1fcbd/assets/images/large/science.abj0113-f1.jpg) and dynamics of measles diffusion in the UK (https://www.nature.com/articles/s41559-020-1186-6/figures/3).

And here you will find a figure describing the continuous diffusion of Aedes albopictus in the USA and Europe (https://www.nature.com/articles/s41564-019-0376-y/figures/1).

Q2: Plot the coordinates and infection status (0 for uninfected, 1 for infected, with different colours for the timing of infection) of a fungal rust pathogen (see picture below) on the Filipendula ulmaria wild plant. Summarise the data (number of infected locations in 1994, 1995 and how many remained uninfected).

Plant Infected With Rust

Context: Triphragmium ulmariae is a species of rust fungus in the family Sphaerophragmiaceae. It causes meadowsweet rust gall which develops as a chemically induced swelling, arising from the lower surface of the Filipendula ulmaria leaves. It has implications for the survival of the meadowsweet seedlings.

R code adapted from Chapter 11, Epidemics: Models and Data in R, Ottar N. Bjornstad (ISBN 978-3-319-97487-3) (https://www.springer.com/gp/book/9783319974866).

```
data(filipendula)

cols <- c('Uninfected' = '#3a506b','Infected in 1994' = '#ffbc42',
          'Infected in 1995' = '#d81159', 'All plants' = "grey")

a1 <- ggplot() +
  geom_point(aes(X, Y,fill = 'All plants', color = 'All plants'),
             data = filipendula, shape = 21, size = 2,
             alpha = 0.5) +
  theme_bw() +
  xlim(0,700) + labs(x= "Distance in meters", y= "Distance in meters") +
  ylim(0,700) +
  scale_fill_manual(name = 'Infection Status',
                    values = cols) +
  scale_color_manual(name = 'Infection Status',
                     values = cols) +
  theme(legend.position='none')

a2 <- ggplot() +
  geom_point(aes(X, Y,fill = 'All plants', color = 'All plants'),
             data = filipendula, shape = 21, size = 2,
             alpha = 0.5) +
  geom_point(aes(X, Y,fill = 'Infected in 1994', color = 'Infected in 1994'),
             data = filipendula[filipendula$y94==1 & filipendula$y95 == 1, ],
             shape = 21, size = 2,
             alpha = 0.5) +
  theme_bw() +
  xlim(0,700) + labs(x= "Distance in meters", y= "Distance in meters") +
  ylim(0,700) +
  scale_fill_manual(name = 'Infection Status',
                    values = cols) +
  scale_color_manual(name = 'Infection Status',
                     values = cols) +
  theme(legend.position='none')

a3 <- ggplot() +
  geom_point(aes(X, Y,fill = 'All plants', color = 'All plants'),
             data = filipendula, shape = 21, size = 2,
             alpha = 0.5) +
  geom_point(aes(X, Y,fill = 'Infected in 1994', color = 'Infected in 1994'),
             data = filipendula[filipendula$y94==1 & filipendula$y95 == 1, ],
             shape = 21, size = 2,
             alpha = 0.5) +
  geom_point(aes(X, Y,fill = 'Infected in 1995', color = 'Infected in 1995'),
             data = filipendula[filipendula$y94==0 & filipendula$y95 == 1, ],
             shape = 21, size = 2,
             alpha = 0.5) +
  theme_bw() +
  xlim(0,700) + labs(x= "Distance in meters", y= "Distance in meters") +
  ylim(0,700) +
  scale_fill_manual(name = 'Infection Status',
                    values = cols) +
  scale_color_manual(name = 'Infection Status',
                     values = cols) +
  theme(legend.position = 'top')
```

```
legend_plot <- get_legend(a3)

a3_new <- a3 +
  theme(legend.position='none')

plot_grid(a1,a2,a3_new, legend_plot, ncol = 3,
          rel_heights = c(6, 1))
```

Q3: Calculate the distance between each pair of X and Y coordinate using the dist function in R. Visualise the distribution of distances and describe them (unit of distances is meters). Please also explain why we need a distance matrix.

The distance between two points $[x_1, y_1]$ and $[x_2, y_2]$ calculated using the formula: $d(x_1, y_1, x_2, y_2) = \sqrt{}((x_2 - x_1)^2 - (y_2 - y_1)^2)$.

We first have to calculate the distance between each pair of observations of Filipendula ulmaria across the island.

```
dst = as.matrix(dist(filipendula[,c("X","Y")]))
hist(dst)
```

Q4: Calculate the force of spatial infection based on infected and uninfected plants. The likelihood of a uninfected plant becoming infected depends on the spatial force of invasion, which is given by its connectivity to all infected plants:

The force of infection experienced by an uninfected plant i is proportional to $\sum_j z_j exp(-d_{ij}/a)$, where z_j is the disease status $[0, 1]$ in the previous year of any plant j that is not i, and d_{ij} is the distance between plant i and j. a is the shape parameter which governs how fast the force of infection (foi) decays with increasing distance. Also note that the summation is performed over all plants.

Explain in non-technical language how the force of invasion in 1995 is defined in this context and how changing the shape parameter a can affect our estimation of the force of invasion.

```
max_dist <- 100
a <- rep(seq(3, 15, by = 3), each = max_dist)
d <- rep(seq(1, max_dist), length(unique(a)))
df <- cbind(a, d) %>%
    as.data.frame()
df <- df %>%
    mutate(foi = exp(-d/a))
ggplot(data = df, aes(x = d, y = foi)) + geom_line(aes(color = factor(a))) +
    scale_color_manual(values = c("#dad7cd", "#a3b18a", "#588157",
        "#3a5a40", "#344e41"), name = "a") + theme_bw()
```

Calculate force of invasion per location for 1995 based on data from 1994 for a given a. Try changing a and see how the mean foi changes.

```
a = 10
foi = apply(exp(-dst/a) * filipendula$y94, 2, sum)
mean(foi)
```

Q5: Now we like to compare how a spatial model estimating the dispersal compares to a model which is non-spatial (nullmod). To do so we create a second model in which the risk of infection is uniform across all locations. We then

compare this model to the spatial model using the anova function in R (more details available here: https://www.youtube.com/watch?v=wEY1M8Pg0Wg (https://www.youtube.com/watch?v=wEY1M8Pg0Wg)).

Here our outcome is infection status in year 1995 and we want to know which one out of the following is the better predictor, using logistic regression (because our outcome is binary):

- Likelihood of becoming infected which is calculated by the connectivity of an uninfected plant to infected plants (spmod in code)
- No information about connectivity between uninfected and infected plants (nullmod in code)

Explain the results of the anova test below. Please note, the residual deviance would be 0 if the model can perfectly explain the data. Lower residual deviance indicates a better model fit.

```
# Using a GLM framework (more details on GLM can be found here:
# http://www.simonqueenborough.info/R/statistics/glm-binomial)
lfit = glm(y95 ~ foi, family = binomial(), data = filipendula)
lfit$deviance/2

#################################################
a = seq(1,20, length = 1001)
llik = rep(NA, length(a))
for(i in 1:length(a)){
  foi = apply(exp(-dst/a[i])*filipendula$y94,2,sum)
  lfit = glm(y95~foi, family = binomial(), data = filipendula)
  llik[i] = lfit$deviance/2
}

#################################################
ahat = a[which.min(llik)]
foi = apply(exp(-dst/ahat)*filipendula$y94,2,sum)
spmod = glm(y95~foi, family = binomial(), data = filipendula)
nullmod = glm(y95~1, family = binomial(), data = filipendula)
# Correct the df of the spmod
spmod$df.residual = spmod$df.residual-1
anova(nullmod, spmod, test = "Chisq")
```

Q6: We are now interested in whether a Gaussian kernel may be better in approximating the dispersal. To do so we create another model in which the force of infection is calculated using the Gaussian shape rather than exponential (see introduction).

Calculate the log-likelihood for the Gaussian kernel and compare the two models using AIC (see more details on AIC here (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8187274/) section “Traditional null-hypothesis significance testing”) and visualise their kernels. Explain what the plot shows.

AIC is a function of:

- Model complexity (number of parameters used); it prefers a more parsimonious model
- Likelihood (how well the model reproduces the data)

Hence, the better model according to AIC is the one that explains the greatest amount of variation in the outcome using the fewest possible predictors. AIC = *2(number of parameters)- 2*log(Likelihood). The model with the lower AIC is better.

```
a2 = seq(1, 20, length = 1001)
llik2 = rep(NA, length(a2))
for (i in 1:length(a2)) {
    foi2 = apply(exp(-(dst/a2[i])^2) * filipendula$y94, 2, sum)
    lfit2 = glm(y95 ~ foi2, family = binomial(), data = filipendula)
    llik2[i] = lfit2$deviance/2
}
ahat2 = a2[which.min(llik2)]
foi2 = apply(exp(-(dst/ahat2)^2) * filipendula$y94, 2, sum)
spmod2 = glm(y95 ~ foi2, family = binomial(), data = filipendula)
spmod2$df.residual = spmod2$df.residual - 1

##################################################
curve((2/(ahat2 * gamma(1/2))) * exp(-((x/ahat2)^2)), 0, 10,
    col = 2, lty = 2, ylab = "Probability of invasion", xlab = "Distance between infe
cted and uninfected plants (Meters)")
curve((1/(ahat) * gamma(1)) * exp(-x/ahat), 0, 10, add = TRUE)
legend("topright", c("Exponential", "Gaussian"), lty = c(1, 2),
    col = c(1, 2))

##################################################
spmod$aic
spmod2$aic
```

Q7: Human mobility and its impact on spatial dispersal of epidemics.

The diffusion of human pathogens rarely follows simple patterns that can be approximated by the exponential or Gaussian kernels discussed above. The patterns of human mobility are more complex and follow patterns of transportation and human population aggregation. For example, it is more likely to travel to and from major population aggregation than to rural areas, even if the rural areas are closer to an individual's home location. Consider the case of Oxford: on average people living in Oxford are more likely to travel to and from London than travel to let's say, Cirencester. Cirencester is closer to Oxford but travel there via train takes ~2h, vs. 1h to central London. Chichester's population is about 20k vs. London's population is currently estimated at > 8M. So it is no surprise that human mobility patterns today are not strictly determined by distance.

Models approximating these data usually take into account the population of the origin and destination location, distance or travel time between them, and any other variables that may influence travel patterns such as the attractiveness of a population (shopping and work opportunities in cities). For the purpose of our practical we will consider the simplest of these models which is the Gravity model (Erlander and Stewart 1990: https://books.google.co.uk/books/about/The_Gravity_Model_in_Transportation_Anal.html?id=tId3PU1leR8C&redir_esc=y (https://books.google.co.uk/books/about/The_Gravity_Model_in_Transportation_Anal.html?id=tId3PU1leR8C&redir_esc=y)). The gravity model posits that movement volume between two communities depends inversely on distance, d, but bilinearly on the size, N, of the communities considered. More generally the model can be written as: $T_{ij} = \frac{N_i^a N_j^b}{d_{ij}^c}$ where T_{ij} is the travel volume between locations i and j, N_i^a is the population at origin location i, N_j^b is the population of destination location j and d_{ij}^c is the distance between i and j. d_{ij}^c could also be the travel time between i and j. a, b, c are parameters fitted using empirical data.

Using a previously developed model by Viboud et al. 2006 (https://www.science.org/doi/10.1126/science.1125237) for the spatial spread of influenza in and between US cities, we will consider a simple version of the SIR model ignoring any susceptible recruitment (no births or deaths):

$$\frac{dS_i}{dt} = -(\beta I_i + \sum_{j \neq i} l_{j,i} I_j) S_i$$

$\frac{dI_i}{dt} = (\beta I_i + \sum_{j \neq i} l_{j,i} I_j) S_i - \gamma I_i$

$\frac{dR_i}{dt} = \gamma I_i$

where $l_{j,i} I_j$ is the gravity-weighted force of infection exerted by location j on location i. β is the transmission coefficient, I is number of infected individuals, S number of susceptibles, and γ is the recovery rate.

Please use links provided in the text above and any other online resources to describe in your own words the Gravity model and the basic intuition behind it.

Q8: We now translate the theory from above into code in R. This requires the R-package 'deSolve'.

Explain in your own words the basics of the SIR model and its key assumptions. Should you be unfamiliar with it, there will be another practical going into the theory behind it. Otherwise please use the literature or watch a short section of this video by Prof. Grenfell https://youtu.be/AzVnN5cCFk4 (https://youtu.be/AzVnN5cCFk4) (from minute ~4 to ~7). Some additional details can be found here (https://royalsocietypublishing.org/doi/10.1098/rstb.2020.0263).

```
SIR.space = function(t, y, pars){
  i = c(1:L)
  S = y[i]
  I = y[L+i]
  R = y[2*L+i]
  with(pars,{
    beta = beta[i]
    dS = -(beta*I + m*G%*%I)*S
    dI = (beta*I + m*G%*%I)*S - gamma*I
    dR = gamma*I
    list(c(dS, dI, dR))
  })
}
```

Q9: Now we want to estimate the interaction matrix for all US states using parameters estimated by Viboud et al. 2006, Science.

Generate matrix using the code below, visualise it, and explain what it shows (also refer back to Question 7; $a = 0.3$, $b = 0.6$, $c = 3$, and $G = T_{ij}$ in question 7):

```
require(ncf)
data(usflu)
usdist = gcdist(usflu$Longitude, usflu$Latitude)

###################################################
gravity = function(a, b, c, pop, distance) {
    gravity = outer(pop^a, pop^b)/distance^c
    diag(gravity) = 0
    gravity
}
G = gravity(0.3, 0.6, 3, usflu$Pop, usdist)

class(G)
plot(log10(G), xlab = "US State", ylab = "US State")
```

Q10: What does G in the model above represent?

Q11: To estimate the spatial interaction matrix, what data do we need?

Q12: To simulate flu epidemic in the US, we need to define the reproduction number and recovery parameter. We assume all individuals to be susceptible for simplicity.

Generate model and outputs. Explain the figure and hypothesise why we observe subsequent waves of flu in different US states.

In the code below, we only visualised a selected few states.

```
gamma = 1/3.5
R0=1.8
beta = R0*gamma/usflu$Pop
m = 1/1000/sum(usflu$Pop)
parms = list(beta = beta, m = m, gamma  =  gamma, G = G)
L = length(usflu$Pop)

head(usflu)



S = usflu$Pop
R = I = rep(0, length(usflu$Pop))
usflu$State <- 1:nrow(usflu)
I[31] = 1 # State where to initialise the epidemic simulation
inits = c(S = S, I = I, R = R)


##################################################
times = 0:200
out = ode(inits, times, SIR.space, parms)
infected <- as.data.frame(out[,c(51:99)])
get_state_name <- as.data.frame(usflu)
names(infected) <- as.character(usflu$Acronym)
infected$time <- 1:201
infected_long <- melt(infected, id.vars="time")
state.name <- c("AL", "NY", "MT", "CA", "DE")
ggplot(infected_long %>%
    filter(variable %in% state.name), aes(time, value, col = variable)) +
    geom_point() + theme_bw() + labs(x = "Time", y = "Cases") +
    guides(colour = guide_legend(title = "States"))
```

Q13: Change initial conditions of the model and initialise epidemic in a rural US state (e.g., Montana (MT), relatively smaller population and further away from population centres). Describe the output in relation to an outbreak that originated in NY state. Make use of matrix G and usflu dataframe.

```
G_summary <- as.data.frame(rowSums(G))
colnames(G_summary) <- c('GravitySummary')
usflu <- cbind(usflu, G_summary)
```

Epidemic initialised in Alabama (AL). Smaller epidemic wave there and longer lag between Alaska and onset of epidemics in other states.

```
S = usflu$Pop
R = I = rep(0, length(usflu$Pop))
I[1] = 1
usflu$State <- 1:nrow(usflu)
inits = c(S = S, I = I, R = R)

#####################################################
# We are then ready to simulate and visualise the model.
require(deSolve)
times = 0:200
out = ode(inits, times, SIR.space, parms)
infected <- as.data.frame(out[, c(51:99)])
get_state_name <- as.data.frame(usflu)
names(infected) <- as.character(usflu$Acronym)
infected$time <- 1:201
infected_long <- melt(infected, id.vars = "time")
state.name <- c("AL", "NY", "MT", "CA", "DE")

ggplot(infected_long %>%
    filter(variable %in% state.name), aes(time, value, col = variable)) +
    geom_point() + theme_bw() + labs(x = "Time", y = "Cases") +
    guides(colour = guide_legend(title = "States"))
```

Q14: What are other factors that may explain the spatial synchrony/asynchrony of flu?