

# Prática de Computador III: Socialidade

Marco A. R. Mello

Universidade de São Paulo

Instituto de Biociências

Departamento de Ecologia

Tópicos Avançados em Ecologia de Animais (BIE0315)

Profs. José Carlos Motta Jr. & Marco A. R. Mello

Artigo de base: Guimarães Jr. et al. (2007, Physical Review E)

Agradecimentos: O primeiro autor do artigo de base tirou nossas dúvidas sobre os resultados de seu trabalho.

README

## Sumário

Preparativos

Modelo SIR básico

Modelo SIR com estrutura de rede

Para saber mais

## Preparativos

Leia o tutorial desta prática, disponível aqui em formato PDF e também no moodle da disciplina.

Aqui faremos simulações de modelos SIR usando não o NetLogo, mas pacotes de R.

Defina o diretório de trabalho como sendo a origem deste script.

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

Remova todos os objetos prévios.

```
rm(list= ls())
```

Limpe o console.

```
cat("\014")
```

Carregue os pacotes necessários.

```
library(igraph)

##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
library(deSolve)
library(rJava)
```

## Modelo SIR básico

Crie uma função com as equações diferenciais de um modelo SIR.

```
sir <- function(time, state, parameters) {

  with(as.list(c(state, parameters)), {

    dS <- -beta * S * I
    dI <-  beta * S * I - gamma * I
    dR <-                gamma * I

    return(list(c(dS, dI, dR)))
  })
}
```

Defina os parâmetros da função.

Comece pela proporção em cada compartimento: Suscetíveis = S, Infectados = I, Recuperados = R.

```
init <- c(S = 0.999, I = 0.001, R = 0.000)
```

Defina as taxas: beta: taxa de infecção; gamma: taxa de recuperação.

```
parameters <- c(beta = 1.025, gamma = 0.2)
```

Defina a escala de tempo da simulação.

```
times <- seq(0, 100, by = 1)
```

Resolva as equações diferenciais usando a função ode (General Solver for Ordinary Differential Equations).

```
out <- ode(y = init, times = times, func = sir, parms = parameters)
```

Converta o resultado em um data frame.

```
out <- as.data.frame(out)
```

Delete a variável de tempo.

```
out$time <- NULL
```

Inspecione o resultado.

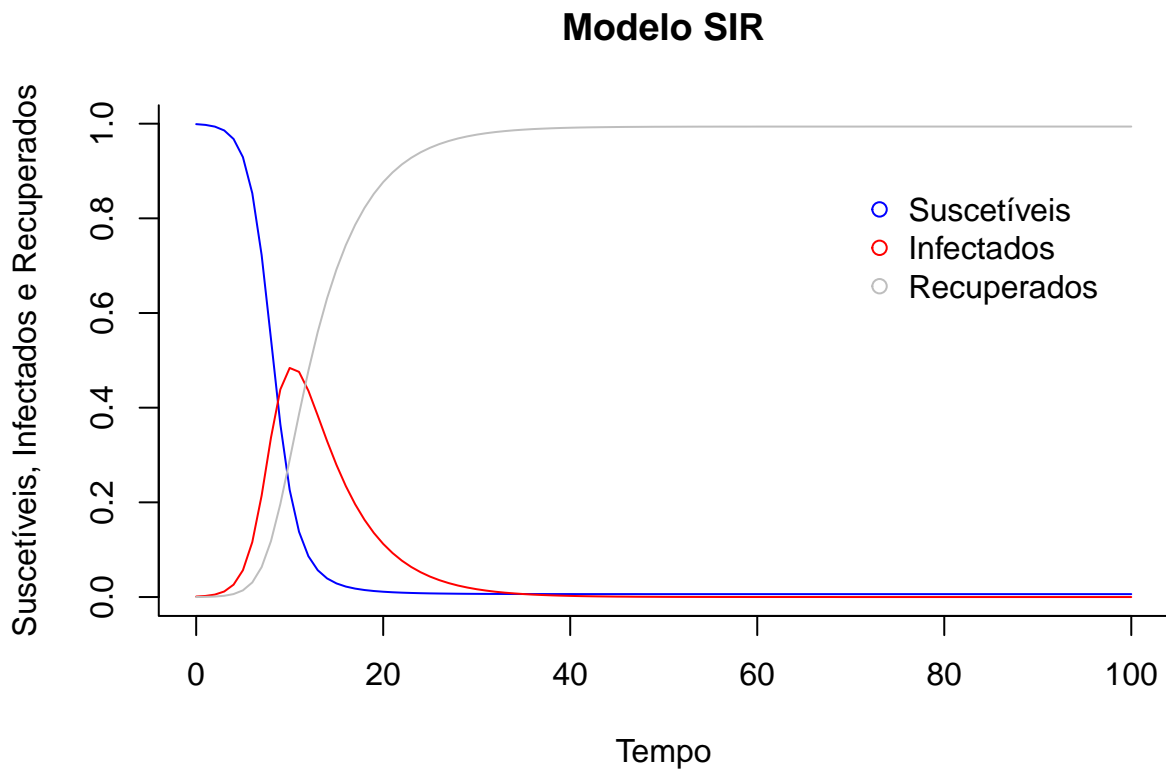
```
head(out)
```

```
##           S           I           R
## 1 0.9990000 0.001000000 0.000000000
## 2 0.9974092 0.002279865 0.0003109669
## 3 0.9937991 0.005182385 0.0010184875
## 4 0.9856709 0.011708131 0.0026209671
## 5 0.9676644 0.026116989 0.0062186047
## 6 0.9292083 0.056660091 0.0141315674
```

Plote as curvas SIR.

```
matplot(x = times, y = out, type = "l",
        xlab = "Tempo",
        ylab = "Suscetíveis, Infectados e Recuperados",
        main = "Modelo SIR",
        lwd = 1, lty = 1, bty = "l",
        col = c("blue", "red", "grey"))

legend(70, 0.9, c("Suscetíveis", "Infectados", "Recuperados"),
      pch = 1,
      col = c("blue", "red", "grey"),
      bty = "n")
```



## Modelo SIR com estrutura de rede

Primeiro, remova os objetos criados anteriormente.

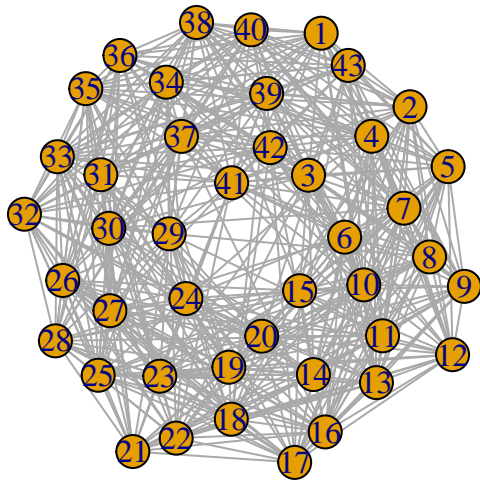
```
rm(list= ls())
```

Crie uma rede com estrutura de mundo pequeno (Watts-Strogatz), tipicamente observada em redes sociais.

```
g <- sample_smallworld(dim = 1,      #tamanho inicial da rede
                      size = 43,    #tamanho final da rede
                      nei = 12,     #grau médio
                      p = 0.05,     #probabilidade de reconexão
                      loops = F,    #a rede pode ter loops ou não?
                      multiple = F) #pode haver elos múltiplos ou não?
```

Plote o grafo para visualizar a rede criada.

```
plot(g)
```

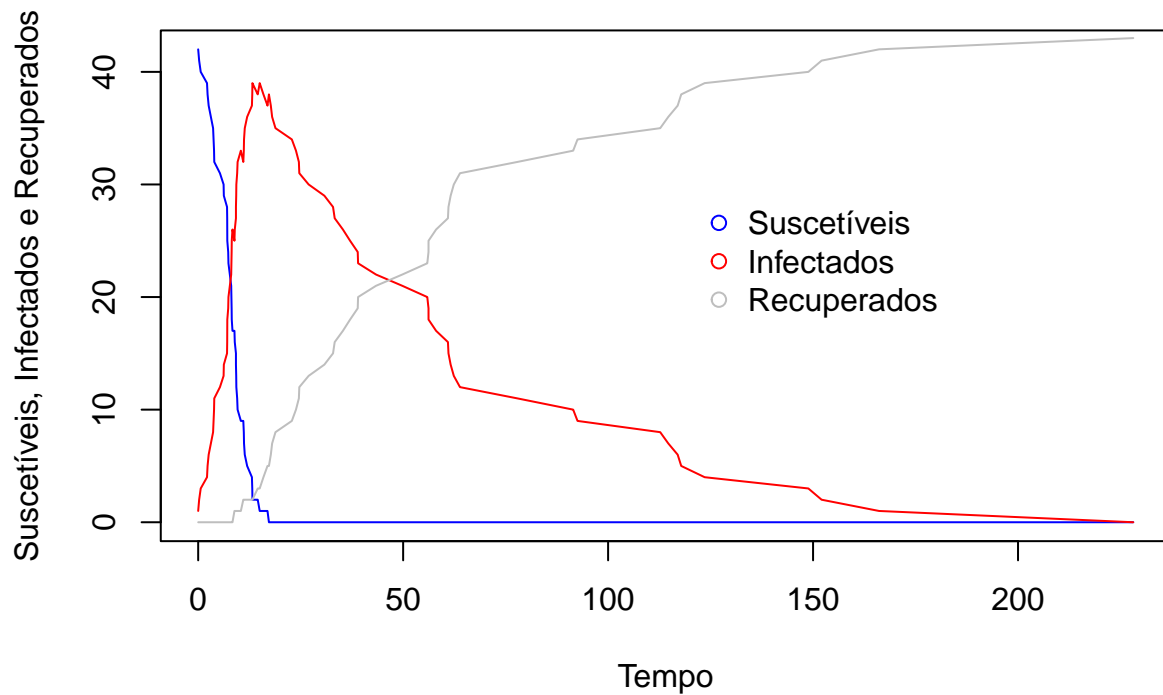


Rode um modelo SIR sobre a estrutura da rede criada.

```
sm <- sir(g,          #a rede que você criou
         beta = 0.02, #probabilidade de infecção
         gamma = 0.02, #probabilidade de recuperação
         no.sim = 1)  #número de rodadas de simulação
```

Plote as curvas SIR.

```
plot(sm[[1]]$NS~sm[[1]]$times,
     col = "blue",
     type = "l",
     xlab = "Tempo",
     ylab = "Suscetíveis, Infectados e Recuperados")
lines(sm[[1]]$NI~sm[[1]]$times,col="red")
lines(sm[[1]]$NR~sm[[1]]$times,col="grey")
legend(120, 30, c("Suscetíveis", "Infectados", "Recuperados"),
      pch = 1,
      col = c("blue", "red", "grey"),
      bty = "n")
```



### Para saber mais

1. O modelo SIR e o achatamento da curva
2. [http://rstudio-pubs-static.s3.amazonaws.com/6852\\_c59c5a2e8ea3456abbab017185de603e.html](http://rstudio-pubs-static.s3.amazonaws.com/6852_c59c5a2e8ea3456abbab017185de603e.html)
3. <https://archives.aidanfindlater.com/blog/2010/04/20/the-basic-sir-model-in-r/>
4. [https://igraph.org/c/doc/igraph-Generators.html#igraph\\_watts\\_strogatz\\_game](https://igraph.org/c/doc/igraph-Generators.html#igraph_watts_strogatz_game)
5. <https://igraph.org/r/html/latest/sir.html>