



Facultad de
Medicina



pyR0compute: A Python Library for Automated Symbolic Computation of the Basic Reproduction Number ($\mathcal{R}_0$) via the Next-Generation Matrix

Dr. Yvan Baldera-Moreno

Facultad de Medicina, Universidad Católica del Maule, Chile
ybaldera@ucm.cl

MS3: Inmunología Matemática en enfermedades infecto-contagiosas
XIV Congreso de la Sociedad Latinoamericana de Biología Matemática, SOLABIMA 2026, Paraguay
"Modelando los Secretos de la Vida"

¿Qué es $\mathcal{R}_0$?

El número esperado de nuevas unidades infecciosas generadas por una unidad infecciosa durante su período infeccioso, cuando el sistema se encuentra en un estado completamente susceptible.

$$\mathcal{R}_0 < 1$$

la infección tiende a desaparecer

$$\mathcal{R}_0 > 1$$

la infección puede propagarse

Método de la matriz de próxima generación

Con los m compartimentos infectados primero, $x = (x_1, \dots, x_m, \dots, x_n)$:

$$\dot{x}_i = \underbrace{\mathcal{F}_i(x)}_{\text{nuevas infecciones}} - \underbrace{\mathcal{V}_i(x)}_{\text{otras transiciones}}, \quad i = 1, \dots, n$$

Linealizando en el equilibrio libre de enfermedad x_0 :

$$F = \left[\frac{\partial \mathcal{F}_i}{\partial x_j}(x_0) \right], \quad V = \left[\frac{\partial \mathcal{V}_i}{\partial x_j}(x_0) \right], \quad 1 \leq i, j \leq m$$

$$\mathcal{R}_0 = \rho(FV^{-1})$$

FV^{-1} : matriz de próxima generación

ρ : radio espectral

Teorema 2 (van den Driessche & Watmough, 2002)

Hipótesis (A1)-(A5)

- (A1) Todas las tasas de transferencia son no negativas.
- (A2) No hay salidas desde un compartimento vacío.
- (A3) Los compartimentos no infectados no reciben nuevas infecciones.
- (A4) Sin infectados, la población permanece libre de enfermedad.
- (A5) Si $\mathcal{F} = 0$, el equilibrio libre de enfermedad es estable.

Conclusión

x_0 es **localmente asintóticamente estable** si $\mathcal{R}_0 < 1$ e **inestable** si $\mathcal{R}_0 > 1$.

Ejemplo 1: Modelo SIR clásico

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta SI}{N} \\ \frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases}$$

Población cerrada: $N = S + I + R$ constante

Compartimento infectado: I

Ejemplo 1.1: vectores $\mathcal{F}$, $\mathcal{V}$ y equilibrio x_0

Se ordena primero el infectado: $x = (I, S, R)^T$, $m = 1$

$$\mathcal{F}(x) = \begin{bmatrix} \frac{\beta SI}{N} \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{V}(x) = \begin{bmatrix} \gamma I \\ \frac{\beta SI}{N} \\ -\gamma I \end{bmatrix}, \quad \dot{x} = \mathcal{F}(x) - \mathcal{V}(x)$$

Con $I = 0$ todas las derivadas se anulan: cualquier $(0, S, R)$ con $S + R = N$ es equilibrio. El DFE **no es único**; se toma la población completamente susceptible.

$$x_0 = (0, N, 0)$$

Ejemplo 1.2: matrices F , V y $\mathcal{R}_0$

$$F = \frac{\partial \mathcal{F}_1}{\partial I}(x_0) = \frac{\beta S_0}{N} = \beta$$

$$FV^{-1} = \frac{\beta}{\gamma}$$

$$V = \frac{\partial \mathcal{V}_1}{\partial I}(x_0) = \gamma$$

Matriz 1×1 : su único valor propio es β/γ .

$$\mathcal{R}_0 = \rho(FV^{-1}) = \frac{\beta}{\gamma}$$

Tasa de contagio β $\times$ duración media de la infección $1/\gamma$

Ejemplo 2: Modelo SEIR

$$\begin{aligned}\frac{dS}{dt} &= \Lambda - \beta SI - \mu S \\ \frac{dE}{dt} &= \beta SI - (\sigma + \mu)E \\ \frac{dI}{dt} &= \sigma E - (\gamma + \mu)I \\ \frac{dR}{dt} &= \gamma I - \mu R\end{aligned}$$

Compartimentos infectados: E (expuestos) e I (infecciosos)

Ejemplo 2.1: vectores $\mathcal{F}$, $\mathcal{V}$ y equilibrio x_0

Se ordenan primero los infectados: $x = (E, I, S, R)^T$, $m = 2$

$$\mathcal{F}(x) = \begin{bmatrix} \beta SI \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{V}(x) = \begin{bmatrix} (\sigma + \mu)E \\ -\sigma E + (\gamma + \mu)I \\ \beta SI + \mu S - \Lambda \\ -\gamma I + \mu R \end{bmatrix}, \quad \dot{x} = \mathcal{F}(x) - \mathcal{V}(x)$$

Equilibrio libre de enfermedad ($E = I = 0$):

$$\begin{aligned} \Lambda - \mu S &= 0 \Rightarrow S = \frac{\Lambda}{\mu} \\ -\mu R &= 0 \Rightarrow R = 0 \end{aligned}$$

$$x_0 = (E^*, I^*, S^*, R^*) = \left(0, 0, \frac{\Lambda}{\mu}, 0\right)$$

Ejemplo 2.2: matrices F , V y $\mathcal{R}_0$

$$F = \left[\frac{\partial \mathcal{F}_i}{\partial x_j}(x_0) \right]_{i,j \leq 2} = \begin{bmatrix} 0 & \frac{\beta \Lambda}{\mu} \\ 0 & 0 \end{bmatrix}$$

$$V = \left[\frac{\partial \mathcal{V}_i}{\partial x_j}(x_0) \right]_{i,j \leq 2} = \begin{bmatrix} \sigma + \mu & 0 \\ -\sigma & \gamma + \mu \end{bmatrix}$$

$$V^{-1} = \begin{bmatrix} \frac{1}{\sigma + \mu} & 0 \\ \frac{\sigma}{(\sigma + \mu)(\gamma + \mu)} & \frac{1}{\gamma + \mu} \end{bmatrix}$$

$$FV^{-1} = \begin{bmatrix} \frac{\Lambda \beta \sigma}{\mu(\sigma + \mu)(\gamma + \mu)} & \frac{\Lambda \beta}{\mu(\gamma + \mu)} \\ 0 & 0 \end{bmatrix}$$

Matriz triangular: valores propios

$$\lambda_1 = \frac{\Lambda \beta \sigma}{\mu(\sigma + \mu)(\gamma + \mu)}, \quad \lambda_2 = 0$$

$$\mathcal{R}_0 = \rho(FV^{-1}) = \frac{\Lambda \beta \sigma}{\mu(\sigma + \mu)(\gamma + \mu)}$$

Desafío: ¿Cuál es $\mathcal{R}_0$ del modelo huésped-vector?.

$$\dot{S}_h = \Delta_h - \delta\alpha_{hv}\frac{S_h I_v}{N_h} - \beta\alpha_{hh}\frac{S_h I_h}{N_h} - (\theta + \mu_h)S_h$$

$$\dot{E}_h = \delta\alpha_{hv}\frac{S_h I_v}{N_h} + \beta\alpha_{hh}\frac{S_h I_h}{N_h} - (\sigma + \mu_h)E_h$$

$$\dot{I}_h = \sigma E_h - (\gamma + \mu_h)I_h$$

$$\dot{R}_h = \gamma I_h - \mu_h R_h + \theta S_h$$

$$\dot{M}_v = \phi_v M_v \left(1 - \frac{M_v}{\kappa_v}\right) - (\omega + \mu_m)M_v$$

$$\dot{S}_v = \omega M_v - \delta\alpha_{vh}\frac{S_v I_h}{N_h} - (\mu_v + \zeta)S_v$$

$$\dot{E}_v = \delta\alpha_{vh}\frac{S_v I_h}{N_h} - (\psi + \mu_v + \zeta)E_v$$

$$\dot{I}_v = \psi E_v - (\mu_v + \zeta)I_v$$

Compartimentos infectados: E_h , I_h , E_v , I_v .

Cálculo manual del $\mathcal{R}_0$

Modelo SIR

$$\begin{aligned} \dot{S} &= \Lambda - \beta \frac{S}{N} I - \mu S \\ \dot{I} &= \beta \frac{S}{N} I - (\gamma + \mu) I \\ \dot{R} &= \gamma I - \mu R \end{aligned}$$

Matriz Jacobiana

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

Matriz Jacobiana en el estado estable

$$J = \begin{pmatrix} \Lambda - \beta \frac{S}{N} I - \mu S & -\beta \frac{S}{N} I & -\mu S \\ \beta \frac{S}{N} I & \beta \frac{S}{N} I - (\gamma + \mu) I & -(\gamma + \mu) I \\ \gamma I & -\mu R & \gamma I - \mu R \end{pmatrix}$$

pyR0compute

```
pip install pyR0compute
```

pyR0compute

```
pip install pyR0compute
```

```
from pyR0compute import R0Model

seir = R0Model("""
    dS/dt = Lambda - beta*S*I - mu*S
    dE/dt = beta*S*I - (sigma + mu)*E
    dI/dt = sigma*E - (gamma + mu)*I
    dR/dt = gamma*I - mu*R
""", infected=["E", "I"])

seir.R0
```

pyR0compute

```
pip install pyR0compute
```

```
from pyR0compute import R0Model

seir = R0Model("""
    dS/dt = Lambda - beta*S*I - mu*S
    dE/dt = beta*S*I - (sigma + mu)*E
    dI/dt = sigma*E - (gamma + mu)*I
    dR/dt = gamma*I - mu*R
    """, infected=["E", "I"])

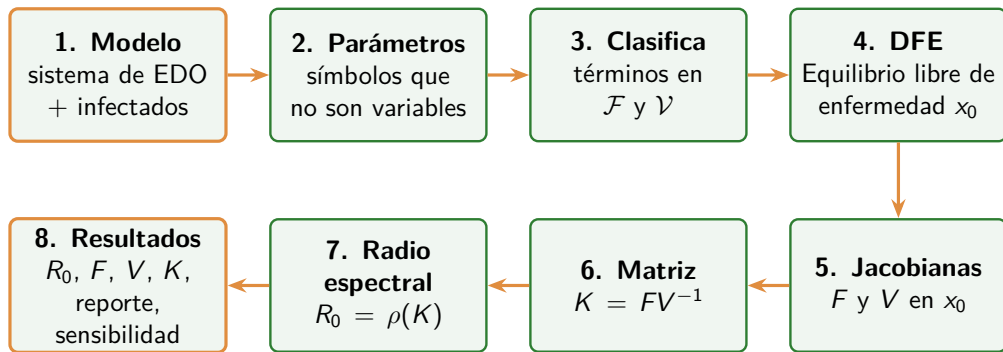
seir.R0
```

Define tu modelo e indica los **compartimentos infectados**;
pyR0compute identifica automáticamente los *parámetros* y calcula $\mathcal{R}_0$.

La clase R0Model(equations, infected, dfe=None, ...)

Nombre	Descripción	Tipo / salida
Atributos		
equations	sistema de EDO (texto, diccionario o lista)	str dict list
infected	compartimentos infectados	list
variables	variables de estado (solo si equations es lista)	list, None
parameters	parámetros declarados, para verificar los detectados	list, None
new_infections	términos $\mathcal{F}_i$ fijados manualmente	dict list, None
dfe	equilibrio libre de infección x_0 (total o parcial)	dict, None
Salidas		
R0	número reproductivo básico $\mathcal{R}_0 = \rho(K)$	expresión SymPy
parameters	parámetros detectados automáticamente	tuple de símbolos
new_infections, transitions	términos $\mathcal{F}_i$ y $\mathcal{V}_i$	dict
dfe, dfe_candidates	equilibrio usado y todos los no negativos	dict, list
F, V, K	matrices F , V y $K = FV^{-1}$	sympy.Matrix
eigenvalues	valores propios de K	list
Métodos		
latex()	expresión de $\mathcal{R}_0$ en LaTeX	str
report_latex()	cálculo paso a paso (LaTeX o Markdown)	str
R0_numeric(values)	evaluación numérica de $\rho(K)$	float
sensitivity_indices()	índices de sensibilidad de $\mathcal{R}_0$	dict

¿Qué hace pyR0compute internamente?



El usuario solo entrega el paso 1; los pasos 2 a 7 son automáticos.

Continuemos en Google Colab

Calculemos $\mathcal{R}_0$ con **pyR0compute**

```
pip install pyR0compute
```

Dr. Yvan Baldera-Moreno

ybaldera@ucm.cl



Escanea para abrir el notebook
`pyR0compute_ejemplos.ipynb`