Further modeling approaches – a brief overview

The 2016 Summer Institute in Statistics and Modeling of Infectious Diseases
Infectious Diseases, Immunology and Within-Host Models

Author: Andreas Handel, Department of Epidemiology and Biostatistics, University of Georgia
ahandel@uga.edu

Types of mechanistic, dynamical models

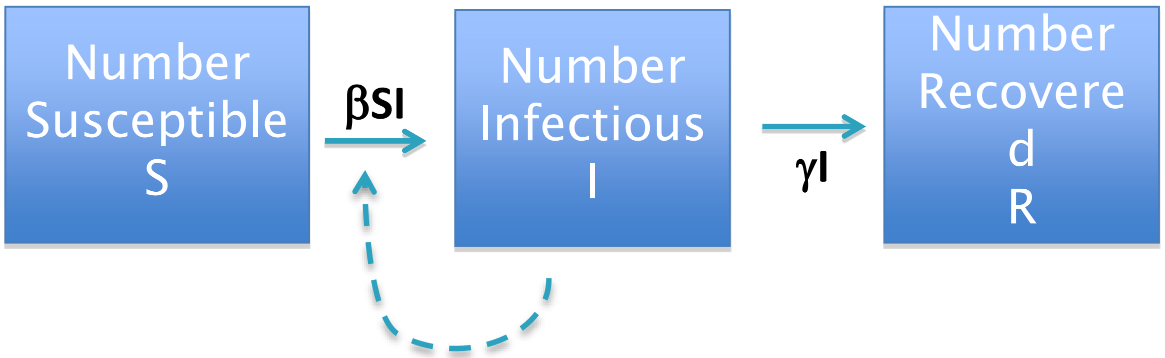
- Because infection processes describe changes in time, **dynamical, mechanistic models** are widely used.
- Such models can be formulated in many different ways. A few possible classifications are:
  - Compartmental ↔ Agent-based
  - Discrete time ↔ continuous time
  - Deterministic ↔ Stochastic
  - Space-less (homogeneous) ↔ Spatial
  - Memory-less (Markov) ↔ with memory
  - Small ↔ Big
  - Data-free ↔ With data

Compartmental versus Agent/Individual based models

Compartmental models

- The components of the model are treated as homogeneous groups (compartments), one only tracks population numbers/sizes
- The simplest type of model, sometimes mathematically tractable, easy to implement on a computer
- Good model for fitting data
- The assumption that populations are homogeneous and “well mixed” is always wrong (but sometimes it is a good enough approximation)
- Often the best starting point

Compartmental models - Examples



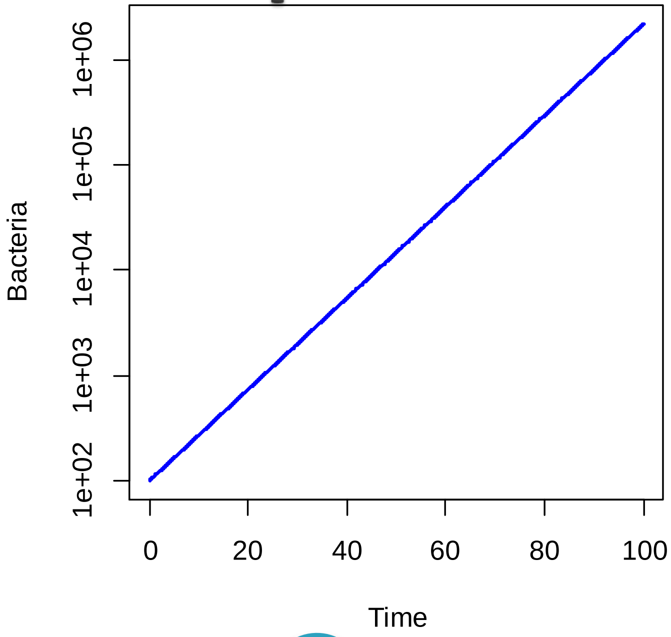
Number Susceptible S

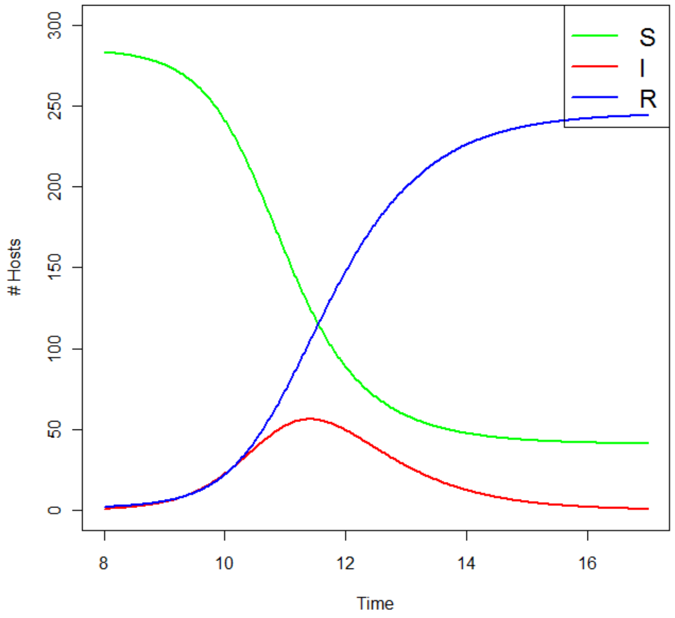
βSI

Number Infectious I

γI

Number Recovered R





$$\dot{S} = -\beta SI$$

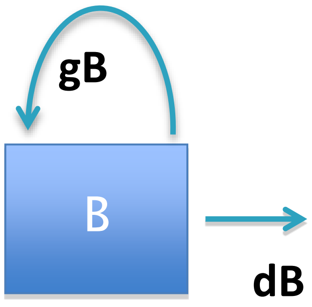
$$\dot{I} = \beta SI - \gamma I$$

$$\dot{R} = \gamma I$$

$$S_{t+\tau} = S_t - (\beta S_t I_t)\tau$$

$$I_{t+\tau} = I_t + (\beta S_t I_t - \gamma I_t)\tau$$

$$R_{t+\tau} = R_t + (\gamma I_t)\tau$$



gB

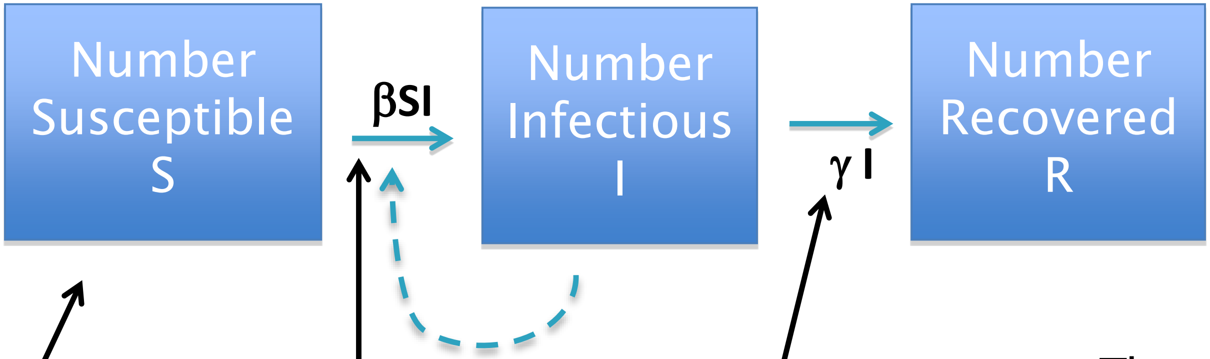
B

dB

$$\dot{B} = gB - dB$$

$$B_{t+\tau} = B_t + (gB - dB)\tau$$

(Compartmental) models - Terminology



Number Susceptible S

βSI

Number Infectious I

γ I

Number Recovered R

Flow (solid)

Interaction (dashed)

Parameter (mostly lower-case letters, often Greek)

Compartments or variables or stocks (mostly upper-case Latin letters)

Time. Technically a dependent variable, usually not called either variable or parameter. Symbol is t for general time, often a for age (sine infection), τ for time-steps

$$\dot{S} = -\beta SI$$

$$\dot{I} = \beta SI - \gamma I$$

$$\dot{R} = \gamma I$$

$$S_{t+\tau} = S_t - (\beta S_t I_t)\tau$$

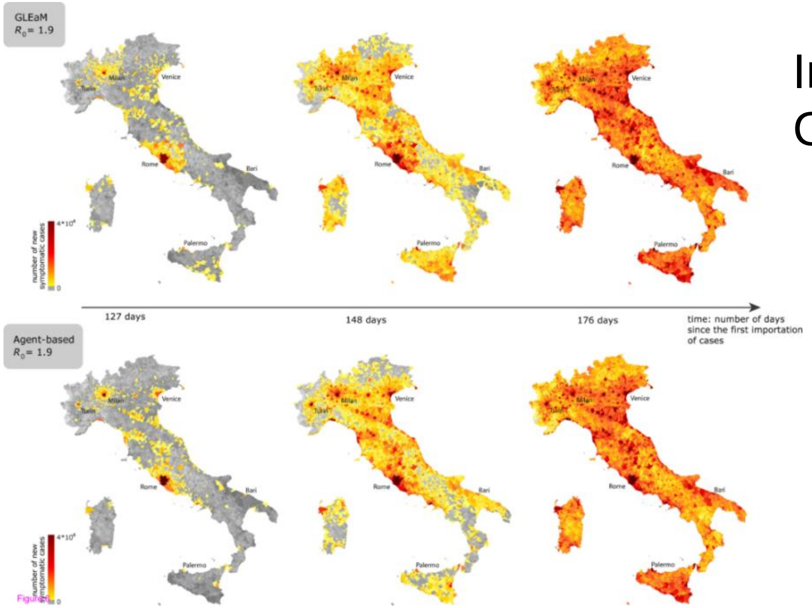
$$I_{t+\tau} = I_t + (\beta S_t I_t - \gamma I_t)\tau$$

$$R_{t+\tau} = R_t + (\gamma I_t)\tau$$

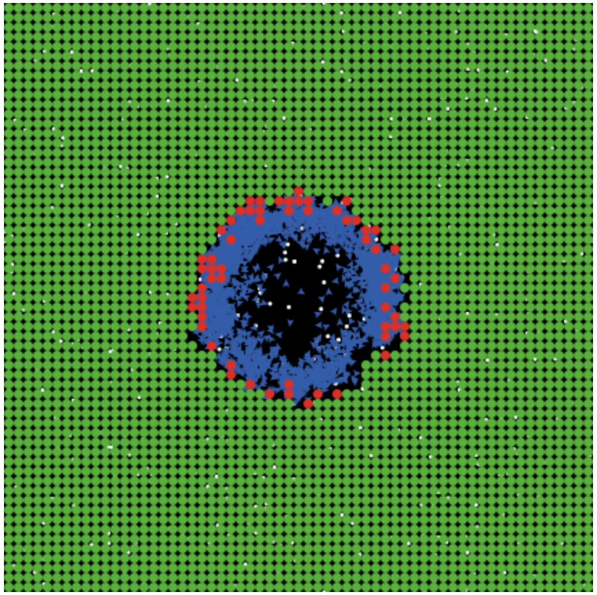
Individual/agent based models

- Every unit/host/individual is modeled/tracked. Called agent-based or individual-based models (ABM/IBM).
- Mostly computational, (almost) no mathematical analysis is possible.
- One can’t easily write down a set of equations, though one can specify a set of rules.
- ABM usually have many parameters.
- ABM take long to run
- ABM are difficult to fit to data
- ABM are potentially most detailed and realistic

Individual based models - examples

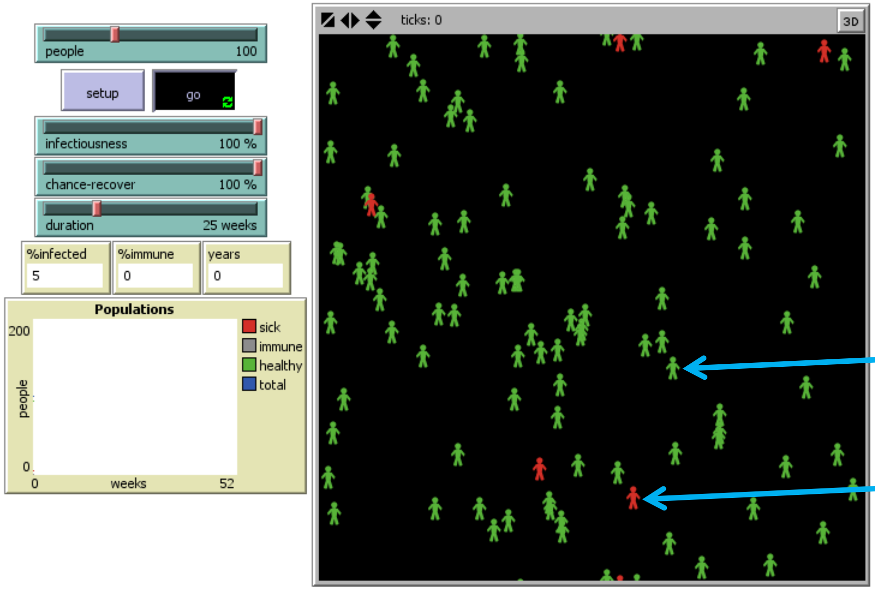


Influenza in Italy:
Gleamviz.org



Influenza infection inside a host

NetLogo outbreak example



people 100

infectiousness 100 %

chance-recover 100 %

duration 25 weeks

%infected 5

%immune 0

years 0

Susceptible

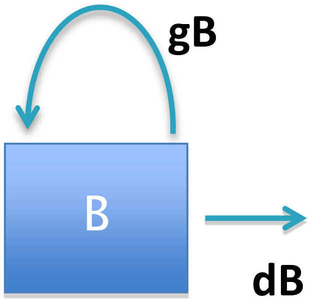
Infectious

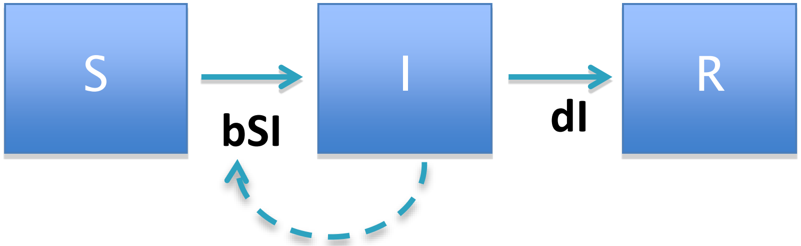
Discrete time versus continuous time models

Discrete time models

- The system is updated in discrete time-steps.
- Good for systems where there is a “natural” time step
  - Example: Some pathogens have a more-or-less fixed replication cycle (e.g. ≈24h for Plasmodium falciparum).
  - Example: For some animals, births occur during a small period in spring. Modeling the long-term dynamics of an ID in such a population might lend itself to a model that is updated annually.
- Complex models, such as Agent-based simulations are almost always discrete-time (for computational reasons).
- Discrete-time compartmental models are often formulated as difference equations.
- If the time-step becomes small, a discrete-time model approaches a continuous-time model.

Discrete time models - Examples



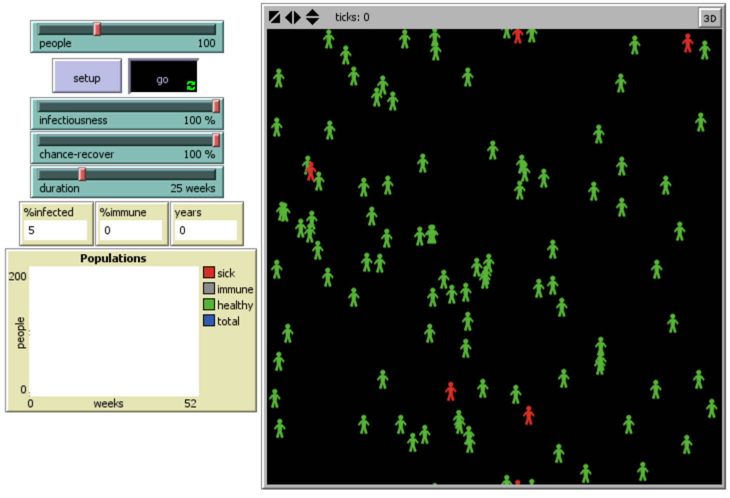


$$B_{t+\tau} = B_t + (gB - dB)\tau$$

$$S_{t+\tau} = S_t - (bS_t I_t)\tau$$

$$I_{t+\tau} = I_t + (bS_t I_t - dI_t)\tau$$

$$R_{t+\tau} = R_t + (dI_t)\tau$$



people 100

infectiousness 100 %

chance-recover 100 %

duration 25 weeks

%infected 5

%immune 0

years 0

ticks: 0

Continuous time models

- The system is updated continuously.
- Best for systems where changes occur continuously and concurrently.
  - Example: To model births and deaths of bacteria in a large population, with new birth and recoveries occurring continuously and concurrently, a continuous-time model might be best.
  - Example: To model an outbreak of flu (or some other ID) in a large population, with new infections and recoveries occurring continuously and concurrently, a continuous-time model might be best.
- Continuous-time models are usually described by differential equations.
- Ordinary differential equation (ODE) models are the most common and simplest one.

Ordinary Differential Equations (ODE)

- ODEs can be derived as the continuum limit of discrete models:

$$B_{t+\tau} = B_t + (gB - dB)\tau \xrightarrow[\text{rewrite}]{} \frac{B_{t+\tau} - B_t}{\tau} = (gB - dB)$$

$$\frac{B_{t+\tau} - B_t}{\tau} = (gB - dB) \xrightarrow[\tau \to 0]{} \frac{dB(t)}{dt} = gB(t) - dB(t)$$

or

$$\dot{B} = gB - dB$$

Ordinary Differential Equations (ODE)

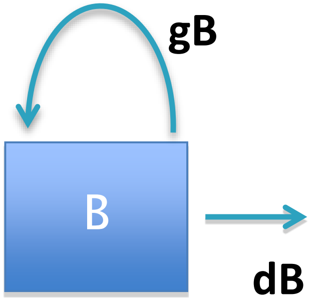
- Often, ODEs are derived as the continuum limit of discrete models:

The computer uses this

$$B_{t+\tau} = B_t + (gB - dB)\tau$$

We often can/could do math with this

$$\dot{B} = gB - dB$$



Same flow diagram for both models

Ordinary Differential Equations (ODE)

- ODEs can be derived as the continuum limit of discrete models:

$$S_{t+\tau} = S_t - (\beta S_t I_t)\tau$$
$$I_{t+\tau} = I_t + (\beta S_t I_t - \gamma I_t)\tau$$
$$R_{t+\tau} = R_t + (\gamma I_t)\tau$$

rewrite

$$\frac{S_{t+\tau} - S_t}{\tau} = -\beta S_t I_t$$
$$\frac{I_{t+\tau} - I_t}{\tau} = \beta S_t I_t - \gamma I_t$$
$$\frac{R_{t+\tau} - R_t}{\tau} = \gamma I_t$$

rename

$$\frac{S_{t+dt} - S_t}{dt} = -\beta S_t I_t$$
$$\frac{I_{t+dt} - I_t}{dt} = \beta S_t I_t - \gamma I_t$$
$$\frac{R_{t+dt} - R_t}{dt} = \gamma I_t$$

Reduce dt → 0

$$\frac{dS}{dt} = -\beta SI$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

Ordinary Differential Equations (ODE)

- Often, ODEs are derived as the continuum limit of discrete models:

The computer uses this

$$S_{t+\tau} = S_t - (\beta S_t I_t)\tau$$

$$I_{t+\tau} = I_t + (\beta S_t I_t - \gamma I_t)\tau$$

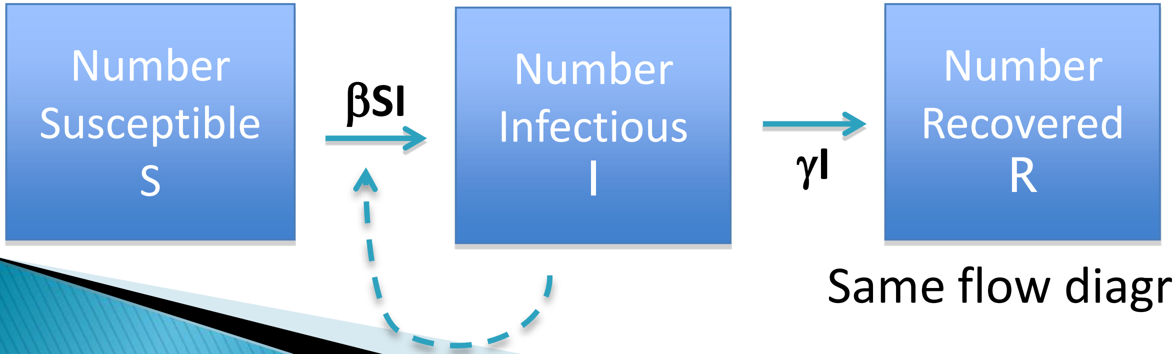
$$R_{t+\tau} = R_t + (\gamma I_t)\tau$$

We often can/could do math with this

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$



Same flow diagram for both models

Continuous time models - Notation

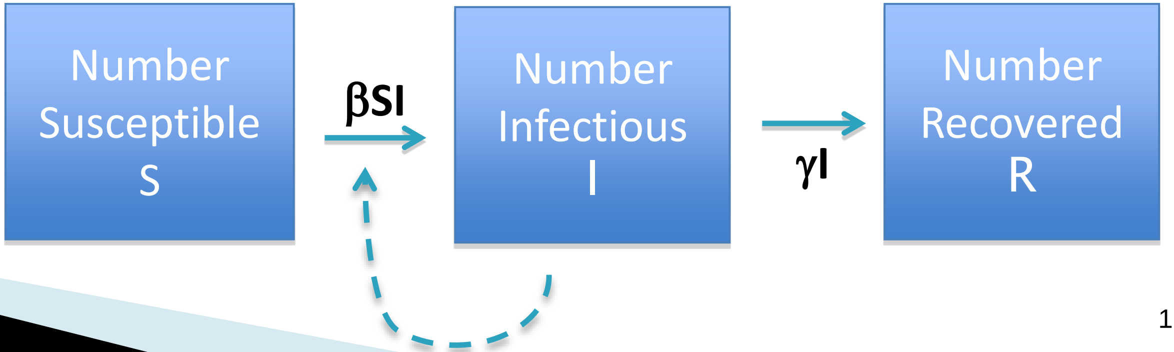
$$\frac{dS(t)}{dt} = -bS(t)I(t)$$
$$\frac{dI(t)}{dt} = bS(t)I(t) - dI(t)$$
$$\frac{dR(t)}{dt} = dI(t)$$

$$\frac{dS}{dt} = -\beta SI$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

$$\dot{S} = -\beta SI$$
$$\dot{I} = \beta SI - \gamma I$$
$$\dot{R} = \gamma I$$

LHS is the (instantaneous) change in a variable/compartment

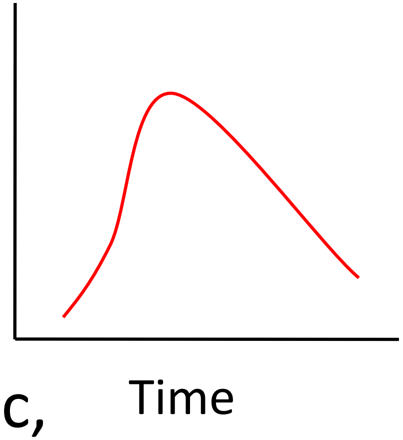
RHS explains how the variable on the LHS increases (positive terms) or decreases (negative terms)



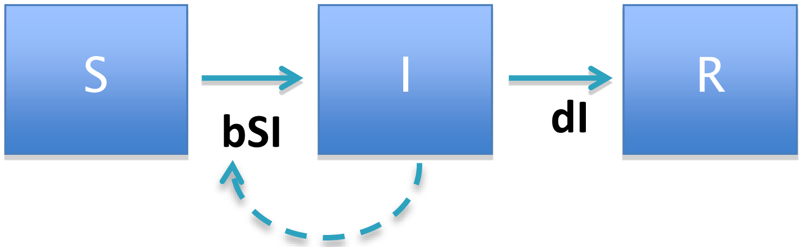
Deterministic versus stochastic models

Deterministic models

- For given parameters and initial conditions, the model always produces the same result
- Simple, easy to implement on a computer
- Sometimes one can do analytical calculations
- Real biological systems are never deterministic, but sometimes approximately so
- When large numbers are involved, deterministic models tend to be good. They break down when only few entities (e.g. few hosts) are involved



Deterministic models - examples



$$S_{t+\tau} = S_t - (bS_tI_t)\tau$$

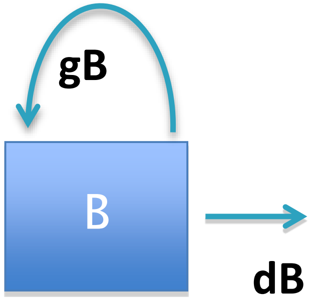
$$I_{t+\tau} = I_t + (bS_tI_t - dI_t)\tau$$

$$R_{t+\tau} = R_t + (dI_t)\tau$$

$$\dot{S} = -bSI$$

$$\dot{I} = bSI - dI$$

$$\dot{R} = dI$$

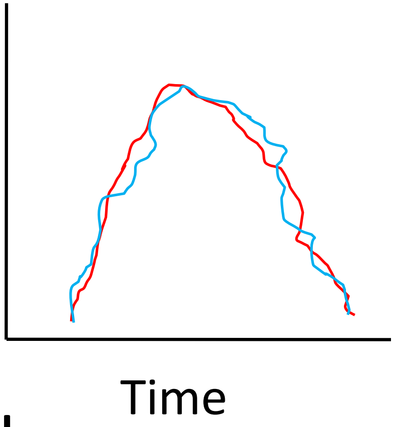


$$\dot{B} = gB - dB$$

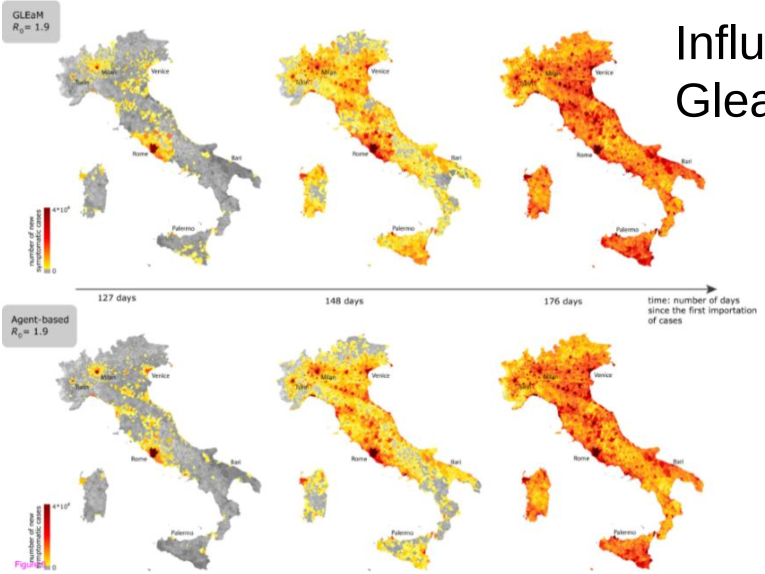
$$B_{t+\tau} = B_t + (gB - dB)\tau$$

Stochastic models

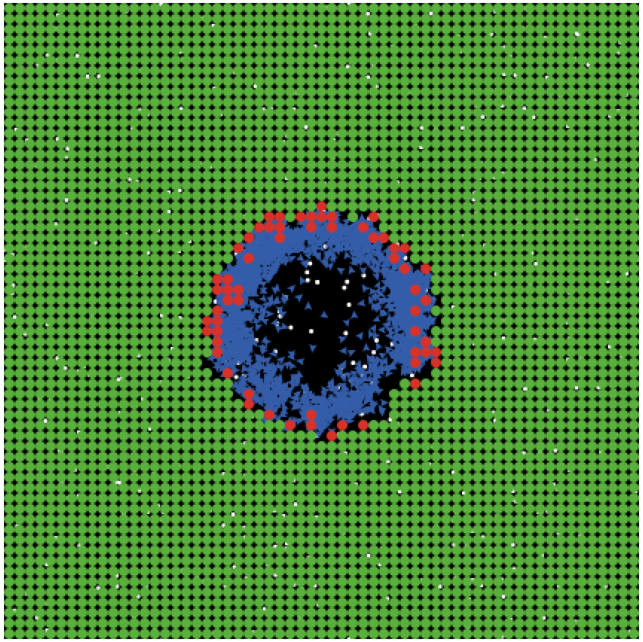
- Results differ between simulations, even for the same model conditions.
- More difficult to implement on a computer, takes longer to run.
- The math is more difficult.
- Closer to the “real” system.
- Stochastic effects are important at low numbers.
- The same model implemented as deterministic or stochastic can lead to different results!



Stochastic models - examples



Influenza in Italy:
Gleamviz.org



$$S_{t+\tau} = S_t - (\beta S_t I_t)\tau$$

$$I_{t+\tau} = I_t + (\beta S_t I_t - \gamma I_t)\tau + Noise$$

$$R_{t+\tau} = R_t + (\gamma I_t)\tau$$

Influenza infection inside a host

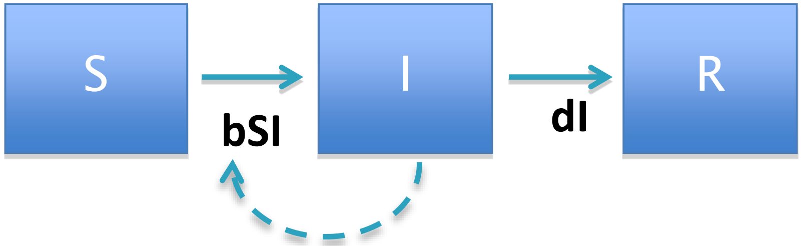
$$B_{t+\tau} = B_t + (gB - dB)\tau + Noise$$

Homogeneous/space-less versus spatial models

Homogeneous/space-less models

- There is no explicit notion of space. Entities (e.g. hosts) are assumed to exist in a homogenous space.
- Entities are assumed to be well-mixed and randomly bump into each other.
- Most compartmental models make this assumption.

Space-less models - examples



$$S_{t+\tau} = S_t - (bS_t I_t)\tau$$

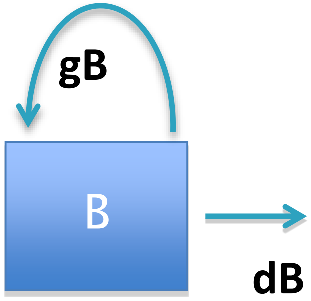
$$I_{t+\tau} = I_t + (bS_t I_t - dI_t)\tau$$

$$R_{t+\tau} = R_t + (dI_t)\tau$$

$$\dot{S} = -bSI$$

$$\dot{I} = bSI - dI$$

$$\dot{R} = dI$$

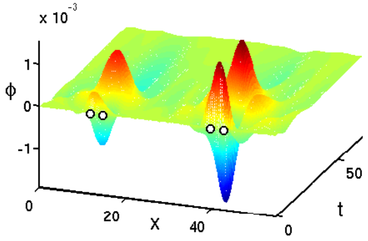


$$\dot{B} = gB - dB$$

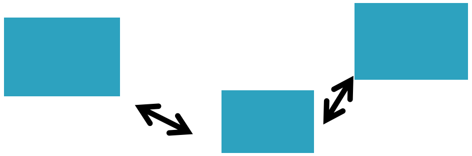
$$B_{t+\tau} = B_t + (gB - dB)\tau$$

Spatial models

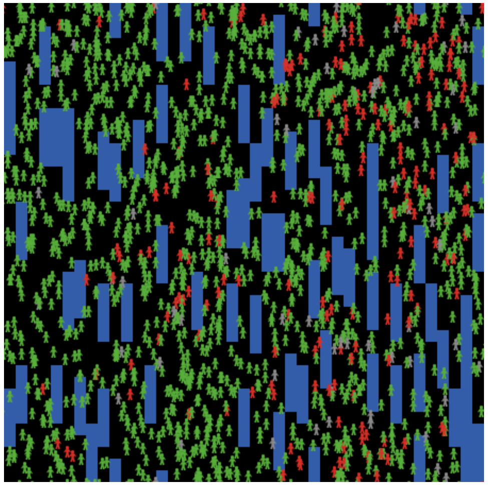
- Some notion of space is explicitly included.
- Different types of models can be used:
  - Partial Differential equations.



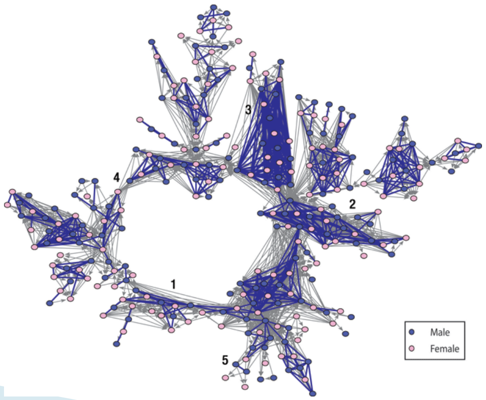
  - Patch/Meta-population models. Usually coupling of multiple compartmental models.



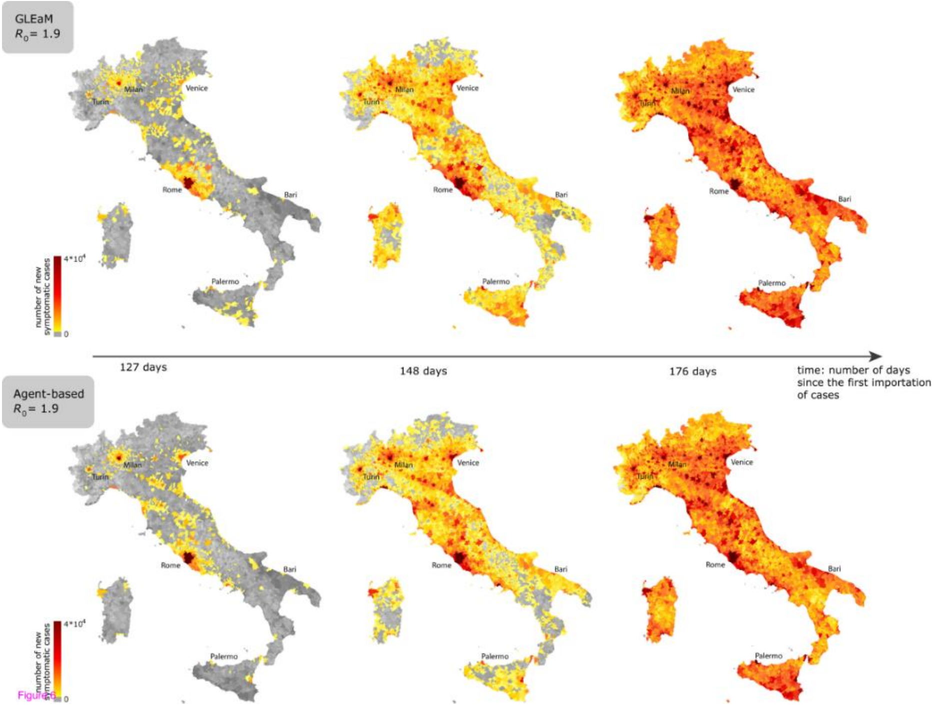
  - Agent-based models.

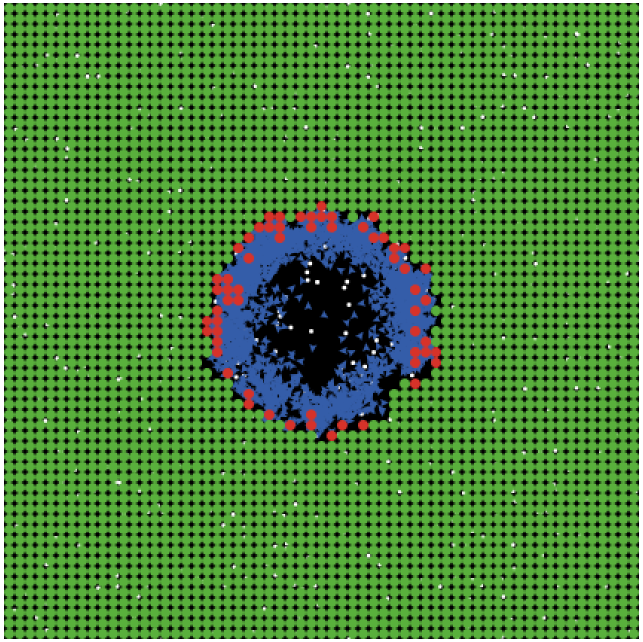


  - Network models.



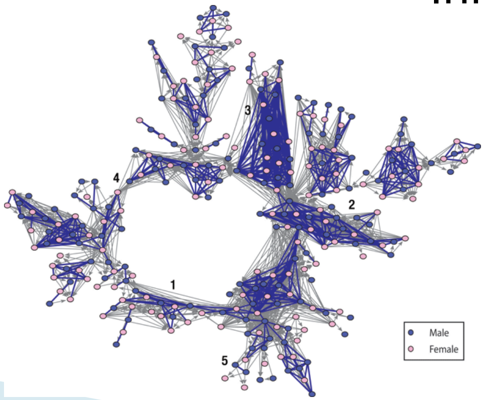
Spatial models - Examples





Influenza infection inside a host

Influenza in Italy:
Gleamviz.org

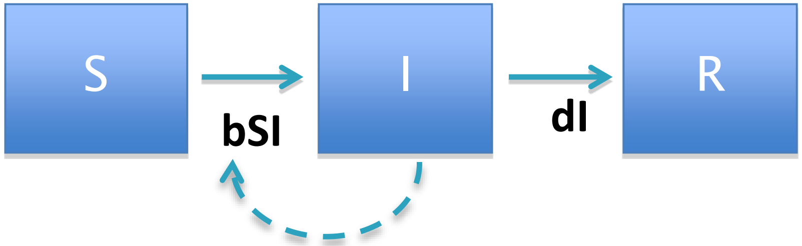


Memory-less (Markov) models versus models with memory

Memory-less models

- Many models (e.g. those based on ordinary differential equations, ODE) are memory-less (markovian). That means what happens next in the system only depends on the **current** state of the system, not on the past.
- That means for instance that an infected individual has an equal chance to recover at any time, no matter how long ago the infection occurred.
- This approximation is sometimes, but not always acceptable.

Memory-less models - examples



$$S_{t+\tau} = S_t - (bS_t I_t)\tau$$

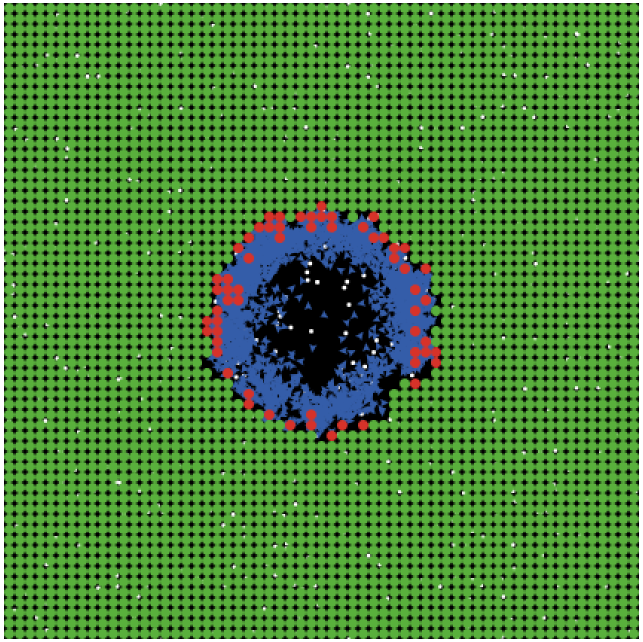
$$I_{t+\tau} = I_t + (bS_t I_t - dI_t)\tau$$

$$R_{t+\tau} = R_t + (dI_t)\tau$$

$$\frac{dS(t)}{dt} = -bS(t)I(t)$$

$$\frac{dI(t)}{dt} = bS(t)I(t) - dI(t)$$

$$\frac{dR(t)}{dt} = dI(t)$$



Depends on how the ABM is implemented.

Models with memory

- If we want to keep track of the past, e.g. if we want to let the chance of recovery depend on the time since infection, we can’t use ODE models.
- We need models that keep track of the past, i.e. that are non-markovian.
- Possible models:
  - Partial differential equations
  - Delay differential equations
  - Agent-based models
  - ODE models with “dummy compartments”

Models with memory - examples

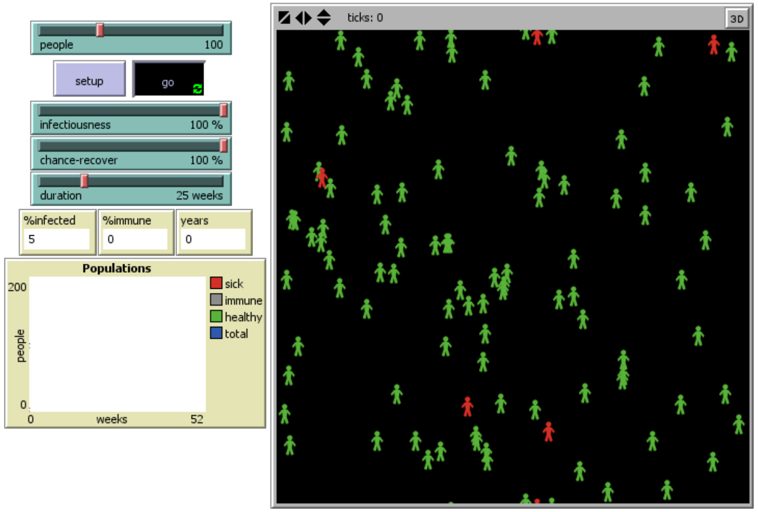
Delay differential equation (DDE)

$$\frac{dS(t)}{dt} = -\beta S(t) I(t-\tau)$$

$$\frac{dI(t)}{dt} = \beta S(t) I(t-\tau) - \gamma I(t)$$

$$\frac{dR(t)}{dt} = \gamma I(t)$$

delay



people 100

infectiousness 100 %

chance-recover 100 %

duration 25 weeks

%infected 5

%immune 0

years 0

ticks: 0

Depends on how the ABM is implemented.

Partial differential equation (PDE)

$$\frac{\partial I(t,a)}{\partial t} = -\gamma(a) I(t,a) - \frac{\partial I(t,a)}{\partial a}$$

Recovery depends on age since infection

Age since infection

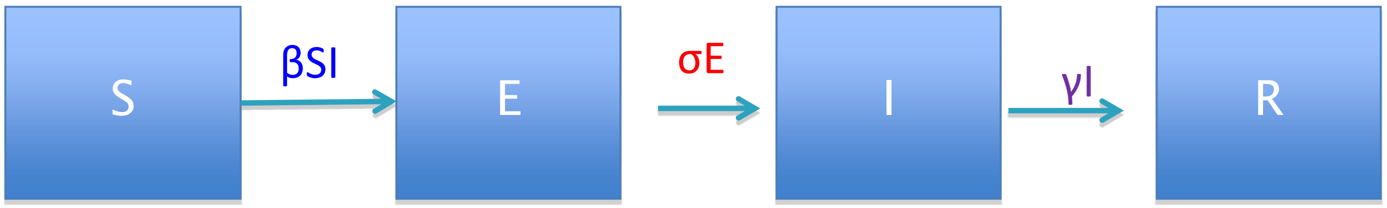
“aging” of infecteds

Small/simple versus large/complex models

Small models

- Start with a very simple model, try to capture the most important aspects of the known dynamics of the system.
- Analyze model to gain conceptual insights. It should be relatively easy to understand what is going on.
- It’s often possible to fit the model to data. Model rejection (e.g. poor agreement with data) is helpful, it taught us something.
- The model might not include crucial known biology and therefore the insights/results might be of limited use (or completely useless).

Small models - examples

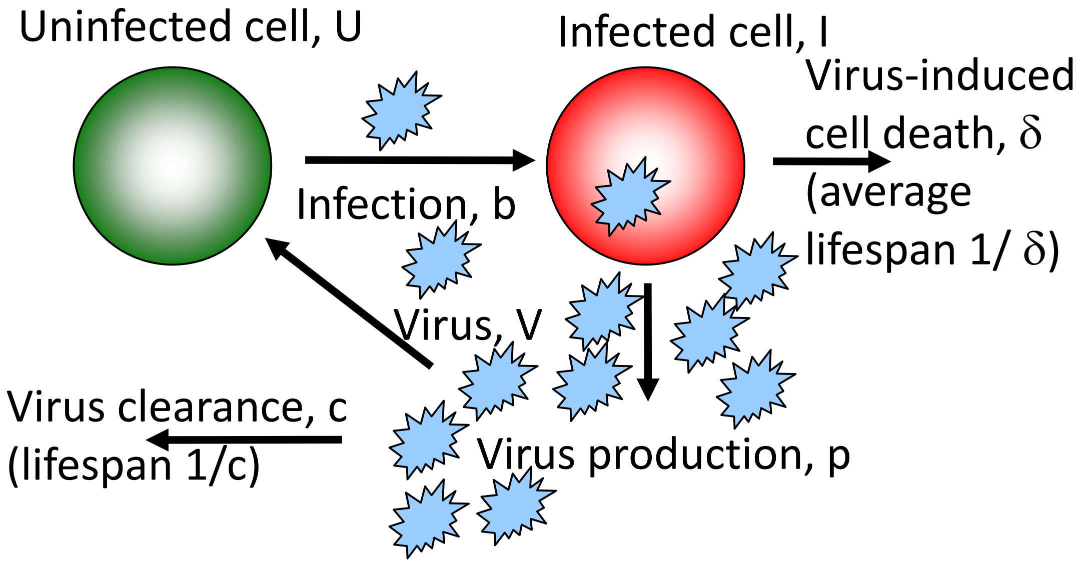


$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dE}{dt} = \beta SI - \sigma E$$

$$\frac{dI}{dt} = \sigma E - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$


Uninfected cell, U
Infected cell, I
Virus-induced cell death, δ (average lifespan 1/ δ)
Infection, b
Virus, V
Virus clearance, c (lifespan 1/c)
Virus production, p

$$\dot{U} = -bUV \qquad \text{(uninfected cells)}$$

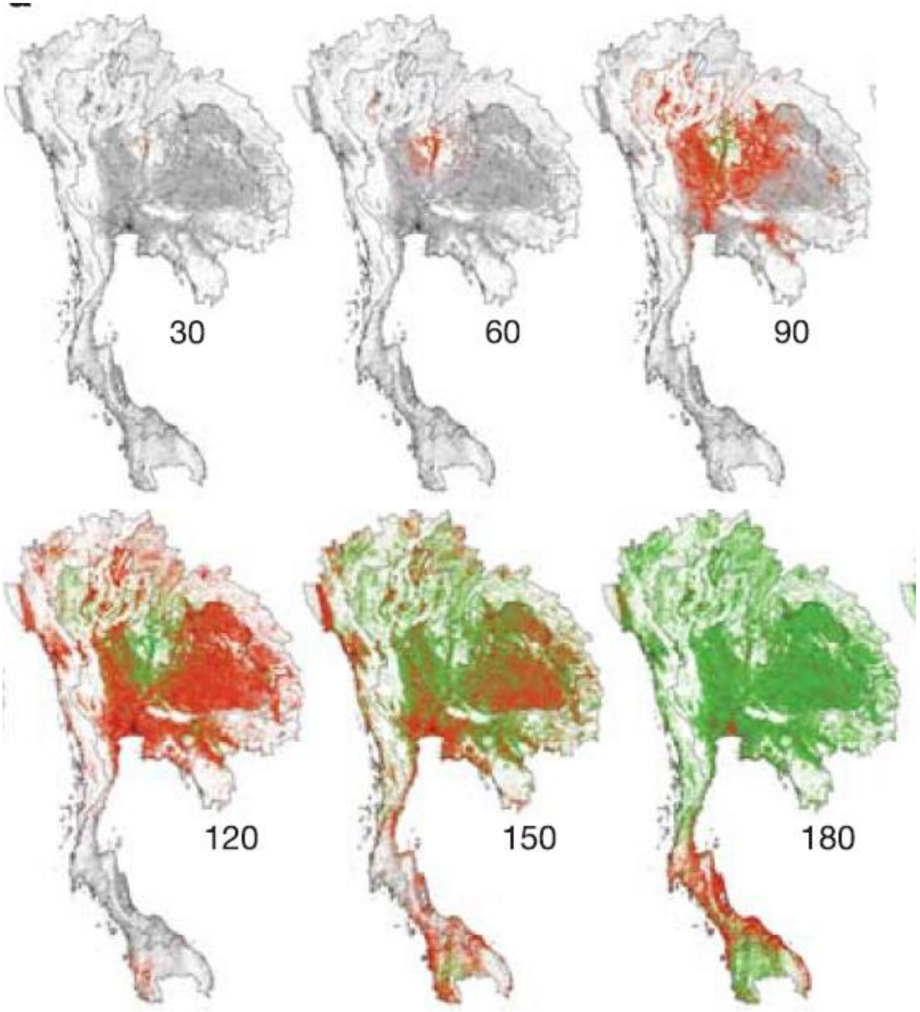
$$\dot{I} = bUV - \delta I \qquad \text{(infected cells)}$$

$$\dot{V} = pI - cV \qquad \text{(free virus)}$$

Big models

- Build a comprehensive model, try to include a lot of detail.
- Run the model on the computer, investigate results. It’s often hard to understand how the different components of the model influence the result. Careful analysis is needed.
- Model results that are at odds with known biology can suggest needed model modifications. But big models can reproduce a lot of observed phenomena, even if the model is wrong.
- Data fitting is usually not possible, seldom is enough data available. Without fitting, model can rarely be formally rejected.
- Comprehensive and accurate models can be used to make detailed, quantitative predictions.

Big models - examples



$$
\begin{aligned}
\lambda_{ii} &= \beta_{\mathrm{u,n}} I_{\mathrm{u,n}} + \beta_{\mathrm{u,p}} I_{\mathrm{u,p}} + \beta_{\mathrm{t,n}} I_{\mathrm{t,n}} + \beta_{\mathrm{t,p}} I_{\mathrm{t,p}} \\
\lambda_{bi} &= \alpha_{\mathrm{u,u}} B_{\mathrm{u,u}} + \alpha_{\mathrm{u,t}} B_{\mathrm{u,t}} + \alpha_{\mathrm{t,u}} B_{\mathrm{t,u}} + \alpha_{\mathrm{t,t}} B_{\mathrm{t,t}} \\
\lambda_1 &= \lambda_{ii} + \lambda_{bi} \\
\lambda_{bb} &= \gamma_{\mathrm{u,u}} B_{\mathrm{u,u}} + \gamma_{\mathrm{u,t}} B_{\mathrm{u,t}} + \gamma_{\mathrm{t,u}} B_{\mathrm{t,u}} + \gamma_{\mathrm{t,t}} B_{\mathrm{t,t}} \\
\lambda_{ib} &= \kappa_{\mathrm{u,n}} I_{\mathrm{u,n}} + \kappa_{\mathrm{u,p}} I_{\mathrm{u,p}} + \kappa_{\mathrm{t,n}} I_{\mathrm{t,n}} + \kappa_{\mathrm{t,p}} I_{\mathrm{t,p}} \\
\lambda_2 &= \lambda_{bb} + \lambda_{ib} \\
\dot{S} &= -\lambda_1 (1 - e_p f_p) S \\
\dot{I}_{\mathrm{u,n}} &= (1 - g_{\mathrm{p}})(1 - f_{\mathrm{t}})(1 - f_{\mathrm{p}})\lambda_1 S - v_{\mathrm{u,n}} I_{\mathrm{u,n}} - k_{\mathrm{u,n}} \lambda_2 I_{\mathrm{u,n}} \\
\dot{I}_{u,\mathrm{p}} &= g_{\mathrm{p}}(1 - f_{\mathrm{t}})(1 - f_{\mathrm{p}})\lambda_1 S - v_{\mathrm{u,p}} I_{\mathrm{u,p}} - k_{\mathrm{u,p}} \lambda_2 I_{\mathrm{u,p}} \\
\dot{I}_{\mathrm{t,n}} &= (1 - g_{\mathrm{p}})(f_{\mathrm{t}}(1 - f_{\mathrm{p}}) + f_{\mathrm{p}}(1 - e_{\mathrm{p}}))\lambda_1 S - v_{\mathrm{t,n}} I_{\mathrm{t,n}} - k_{\mathrm{t,n}} \lambda_2 I_{\mathrm{t,n}} \\
\dot{I}_{\mathrm{t,p}} &= g_{\mathrm{p}}(f_{\mathrm{t}}(1 - f_{\mathrm{p}}) + f_{\mathrm{p}}(1 - e_{\mathrm{p}}))\lambda_1 S - v_{\mathrm{t,p}} I_{\mathrm{t,p}} - k_{\mathrm{t,p}} \lambda_2 I_{\mathrm{t,p}} \\
\dot{B}_{\mathrm{u,u}} &= (1 - f_t)(1 - g_t)(v_{u,n} c_{u,n} + k_{u,n} \lambda_2) I_{\mathrm{u,n}} - \delta_{u,u} B_{u,u} \\
\dot{B}_{\mathrm{u,t}} &= (1 - f_t) g_t (v_{\mathrm{u,n}} c_{\mathrm{u,n}} + k_{\mathrm{u,n}} \lambda_2) I_{u,n} \\
&\quad + (1 - f_t)(v_{\mathrm{u,p}} c_{\mathrm{u,p}} + k_{\mathrm{u,p}} \lambda_2) I_{\mathrm{u,p}} - \delta_{\mathrm{u,t}} B_{\mathrm{u,t}} \\
\dot{B}_{\mathrm{t,u}} &= f_t (1 - g_t)(v_{\mathrm{u,n}} c_{\mathrm{u,n}} + k_{\mathrm{u,n}} \lambda_2) I_{\mathrm{u,n}} \\
&\quad + (1 - g_t)(v_{\mathrm{t,n}} c_{\mathrm{t,n}} + k_{\mathrm{t,n}} \lambda_2) I_{\mathrm{t,n}} - \delta_{\mathrm{t,u}} B_{t,u} \\
\dot{B}_{\mathrm{t,t}} &= f_t g_t (v_{\mathrm{u,n}} c_{\mathrm{u,n}} + k_{\mathrm{u,n}} \lambda_2) I_{\mathrm{u,n}} + f_{\mathrm{t}}(v_{\mathrm{u,p}} c_{\mathrm{u,p}} + k_{\mathrm{u,p}} \lambda_2) I_{\mathrm{u,p}} \\
&\quad + g_{\mathrm{t}}(v_{\mathrm{t,n}} c_{\mathrm{t,n}} + k_{\mathrm{t,n}} \lambda_2) I_{\mathrm{t,n}} + (v_{\mathrm{t,p}} c_{\mathrm{t,p}} + k_{\mathrm{t,p}} \lambda_2) I_{\mathrm{t,p}} - \delta_{\mathrm{t,t}} B_{\mathrm{t,t}}
\end{aligned}
$$

Ferguson et al, 2005 Nature

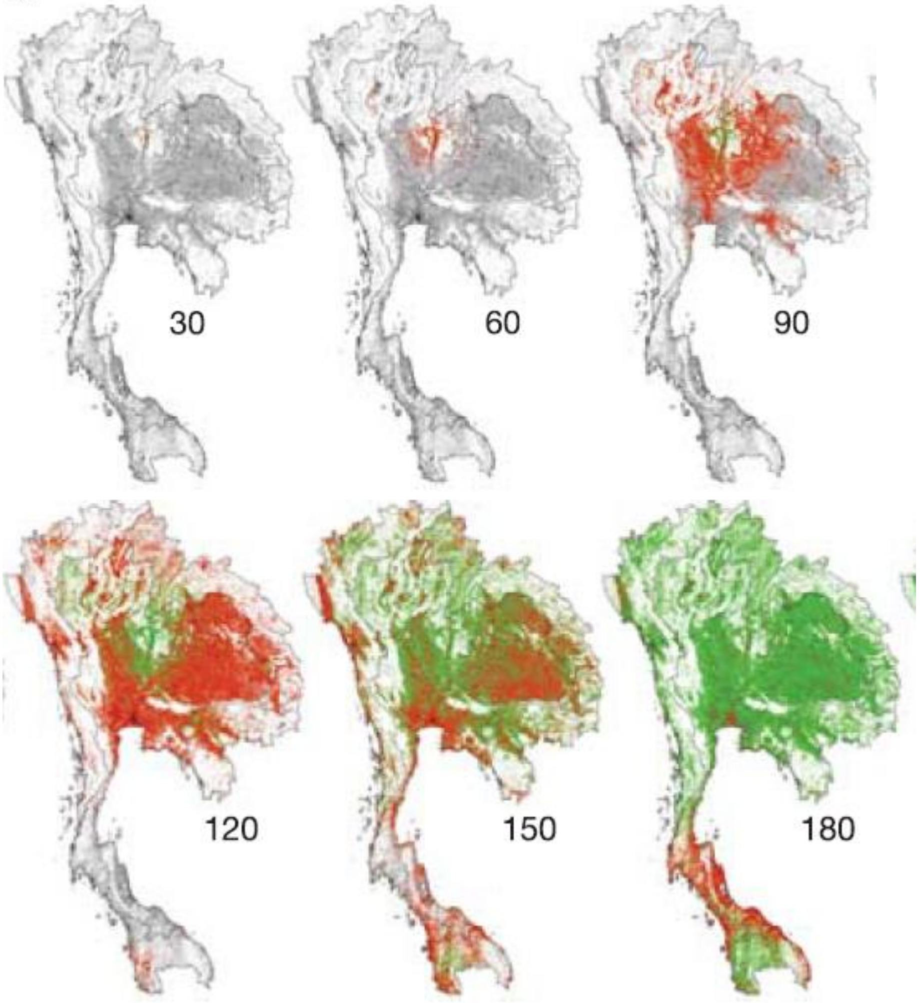
Handel et al, 2009 Epidemics

Data-free models
versus
models with data

Data-free

- Model is formulated and model parameters are chosen based on known biology.
- Model should be “data-driven”, i.e. based on what is known about the system.
- Model is analyzed “by itself”, i.e. without trying to perform inference and rigorously fit it to data.
- Relatively easy to do. Can produce useful insights even if only limited data are available.

Data-free – examples



Ferguson et al, 2005 Nature

$$
\begin{aligned}
\lambda_{ii} &= \beta_{\mathrm{u,n}} I_{\mathrm{u,n}} + \beta_{\mathrm{u,p}} I_{\mathrm{u,p}} + \beta_{\mathrm{t,n}} I_{\mathrm{t,n}} + \beta_{\mathrm{t,p}} I_{\mathrm{t,p}} \\
\lambda_{bi} &= \alpha_{\mathrm{u,u}} B_{\mathrm{u,u}} + \alpha_{\mathrm{u,t}} B_{\mathrm{u,t}} + \alpha_{\mathrm{t,u}} B_{\mathrm{t,u}} + \alpha_{\mathrm{t,t}} B_{\mathrm{t,t}} \\
\lambda_1 &= \lambda_{ii} + \lambda_{bi} \\
\lambda_{bb} &= \gamma_{\mathrm{u,u}} B_{\mathrm{u,u}} + \gamma_{\mathrm{u,t}} B_{\mathrm{u,t}} + \gamma_{\mathrm{t,u}} B_{\mathrm{t,u}} + \gamma_{\mathrm{t,t}} B_{\mathrm{t,t}} \\
\lambda_{ib} &= \kappa_{\mathrm{u,n}} I_{\mathrm{u,n}} + \kappa_{\mathrm{u,p}} I_{\mathrm{u,p}} + \kappa_{\mathrm{t,n}} I_{\mathrm{t,n}} + \kappa_{\mathrm{t,p}} I_{\mathrm{t,p}} \\
\lambda_2 &= \lambda_{bb} + \lambda_{ib} \\
\dot{S} &= -\lambda_1 (1 - e_p f_p) S \\
\dot{I}_{\mathrm{u,n}} &= (1 - g_{\mathrm{p}})(1 - f_{\mathrm{t}})(1 - f_{\mathrm{p}}) \lambda_1 S - v_{\mathrm{u,n}} I_{\mathrm{u,n}} - k_{\mathrm{u,n}} \lambda_2 I_{\mathrm{u,n}} \\
\dot{I}_{u,p} &= g_{\mathrm{p}} (1 - f_{\mathrm{t}})(1 - f_{\mathrm{p}}) \lambda_1 S - v_{\mathrm{u,p}} I_{\mathrm{u,p}} - k_{\mathrm{u,p}} \lambda_2 I_{\mathrm{u,p}} \\
\dot{I}_{\mathrm{t,n}} &= (1 - g_{\mathrm{p}})(f_{\mathrm{t}}(1 - f_{\mathrm{p}}) + f_{\mathrm{p}}(1 - e_{\mathrm{p}})) \lambda_1 S - v_{\mathrm{t,n}} I_{\mathrm{t,n}} - k_{\mathrm{t,n}} \lambda_2 I_{\mathrm{t,n}} \\
\dot{I}_{\mathrm{t,p}} &= g_{\mathrm{p}} (f_{\mathrm{t}}(1 - f_{\mathrm{p}}) + f_{\mathrm{p}}(1 - e_{\mathrm{p}})) \lambda_1 S - v_{\mathrm{t,p}} I_{\mathrm{t,p}} - k_{\mathrm{t,p}} \lambda_2 I_{\mathrm{t,p}} \\
\dot{B}_{\mathrm{u,u}} &= (1 - f_t)(1 - g_t)(v_{u,n} c_{u,n} + k_{u,n} \lambda_2) I_{\mathrm{u,n}} - \delta_{u,u} B_{u,u} \\
\dot{B}_{\mathrm{u,t}} &= (1 - f_t) g_t (v_{\mathrm{u,n}} c_{\mathrm{u,n}} + k_{\mathrm{u,n}} \lambda_2) I_{u,n} \\
&\quad + (1 - f_t)(v_{\mathrm{u,p}} c_{\mathrm{u,p}} + k_{\mathrm{u,p}} \lambda_2) I_{\mathrm{u,p}} - \delta_{\mathrm{u,t}} B_{\mathrm{u,t}} \\
\dot{B}_{\mathrm{t,u}} &= f_t (1 - g_t)(v_{\mathrm{u,n}} c_{\mathrm{u,n}} + k_{\mathrm{u,n}} \lambda_2) I_{\mathrm{u,n}} \\
&\quad + (1 - g_t)(v_{\mathrm{t,n}} c_{\mathrm{t,n}} + k_{\mathrm{t,n}} \lambda_2) I_{\mathrm{t,n}} - \delta_{\mathrm{t,u}} B_{t,u} \\
\dot{B}_{\mathrm{t,t}} &= f_t g_t (v_{\mathrm{u,n}} c_{\mathrm{u,n}} + k_{\mathrm{u,n}} \lambda_2) I_{\mathrm{u,n}} + f_{\mathrm{t}} (v_{\mathrm{u,p}} c_{\mathrm{u,p}} + k_{\mathrm{u,p}} \lambda_2) I_{\mathrm{u,p}} \\
&\quad + g_{\mathrm{t}} (v_{\mathrm{t,n}} c_{\mathrm{t,n}} + k_{\mathrm{t,n}} \lambda_2) I_{\mathrm{t,n}} + (v_{\mathrm{t,p}} c_{\mathrm{t,p}} + k_{\mathrm{t,p}} \lambda_2) I_{\mathrm{t,p}} - \delta_{\mathrm{t,t}} B_{\mathrm{t,t}}
\end{aligned}
$$

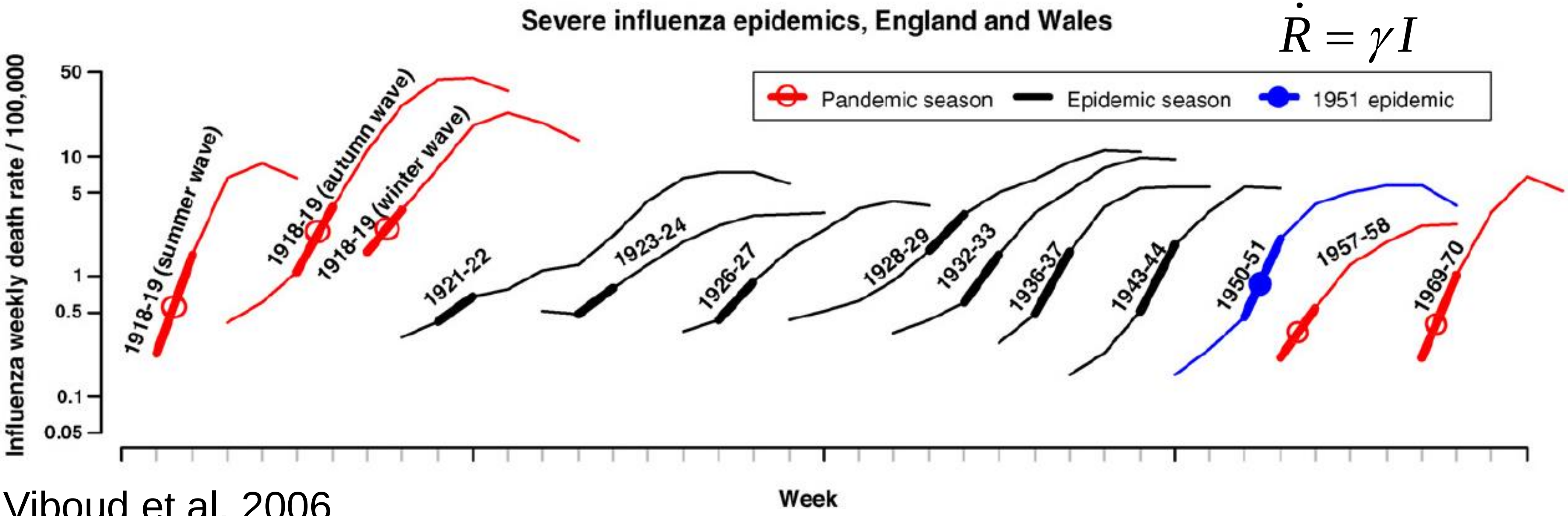
Handel et al, 2009
Epidemics

With data

- Models are being fitted to data (inference).
- Rigorous comparison of models with data.
- Used to discriminate hypotheses, determine parameters.
- Less flexibility in model formulation, often constrained/determined by available data.

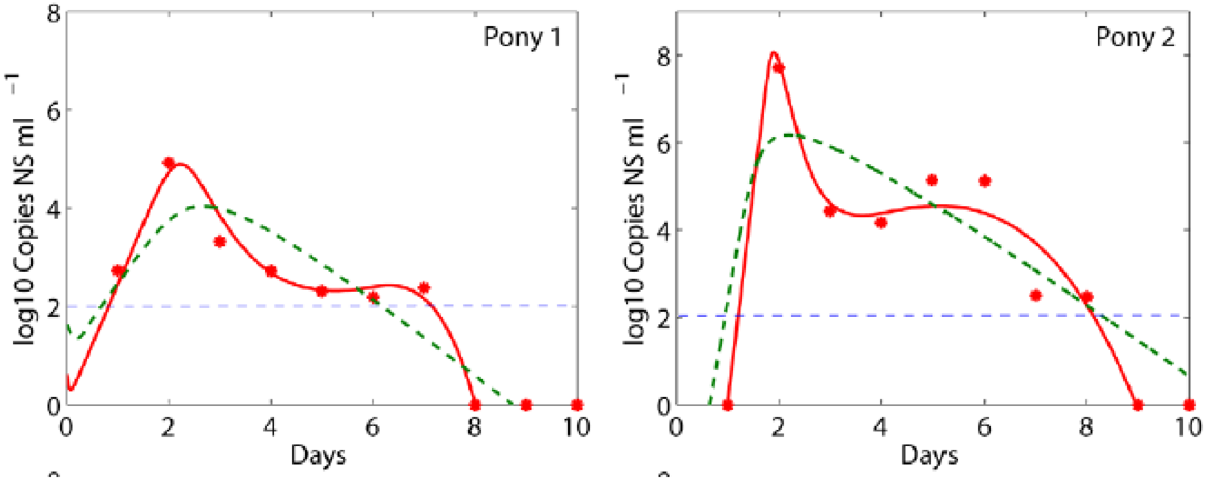
With data – examples

$$\dot{S} = -\beta SI$$
$$\dot{I} = \beta SI - \gamma I$$
$$\dot{R} = \gamma I$$



Viboud et al, 2006 Vaccine

$$\dot{U} = -bUV$$
$$\dot{I} = bUV - \delta I$$
$$\dot{V} = pI - cV$$



Pawelek et al, 2012 PLoS Comp Bio

Types of models

- Models have several of the different characteristics just described. Examples:
  - Deterministic, compartmental, continuous time, no-memory, no-space, small model – i.e. a set of ODEs (very common).
  - Deterministic, agent-based, no-space, no-memory (very rare)
- Ideal approach: Choose the model that is most suitable for the question you try to answer.
- In reality: Model selection is based on a mix of
  - Question one wants to answer
  - Expertise
  - Feasibility (CPU time, model complexity)
  - “Environment” (what approaches do others use)
  - “Marketing” (what kind of models “sell”)

How to build a model

- Figure out what the question/hypothesis/problem is you want to address.
- Decide what kind of model will best help you to answer your question (and if a model is useful at all)!
- Design, implement and test the model.
- Use the model to answer your question.

Stochastic Models

Limitations of deterministic models

- With deterministic models, we can not address questions such as
  - How likely/probable is an outbreak/infection to occur?
  - How likely is it that a pathogen goes extinct? (applies to both the within-host or between-host levels)
  - What variability should we expect when looking at real data?
- Any question that requires an answer in the form of a probabilistic statement needs stochastic methods.

Quick detour 1 – random numbers

- To do any kind of stochastic/probabilistic simulation, we need to produce random numbers
- But computers are deterministic machines…
- Solution: pseudo-random numbers **(reproducible!)**
  - One needs to set a seed, otherwise the computer produces RN depending on the current system time and results won’t be reproducible.
- Back when: Quite a few random number generators (RNG) were bad - the numbers they produced were not “random enough”. Many a published simulation study was wrong because of bad RNG.
- Today: Almost all RNG that come with programs such as R and Matlab are very good (the numbers are “really random”). Current “favorite” RNG of most folks is the Mersenne Twister.
- R can generate RN not only from uniform distributions but all kinds of other distributions (runif, rnorm, rpois, rbinom, …).
- We have already used RN without really discussing them. When?

Stochasticity – Observational error

- Observational error
  - The “true” dynamics of the virus is described by V, but you can only measure/observe O(V)
  - For instance you could sample people/animals and count virus. This might lead to observed values that are (log)normally distributed around the true value of V: O=N(V,σ)
  - Observational error does not affect the dynamics of the system

Observational noise

True system dynamics

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

What you measure

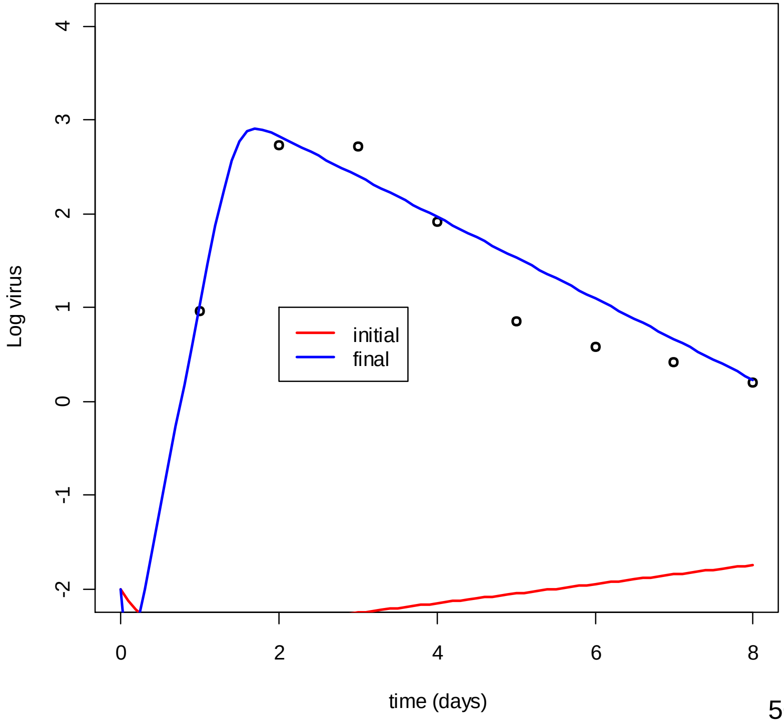
$$\hat{U} = ?$$

$$\hat{I} = ?$$

$$\hat{V} = F(V)$$

Observational noise

- We assumed that only this kind of noise was present (or at least was dominant) when we did fitting.
- Specifically, we assumed that the dynamics was properly described by our ODE model, with (log)-normally distributed noise due to measurement error.
- Other assumptions about error lead to different objective functions (Maximum Likelihood).



Stochasticity – process noise

- Process noise (sometimes called process error)
  - The dynamics of the system is affected by error/noise.
  - External noise (e.g. fluctuations in weather/metabolism) can be added to the model equations.
  - Internal noise (fluctuations in parameters) can be included by sampling a parameter (e.g. infection rate) from a distribution at every time step.
  - Process error/noise propagates through the system.

Process noise

$$\dot{U} = -bUV + N$$

$$\dot{I} = bUV - \delta I + N$$

$$\dot{V} = pI - cV + N$$

$$\dot{U} = -N(b)UV$$

$$\dot{I} = N(b)UV - \delta I$$

$$\dot{V} = pI - cV$$

Stochastic differential equations (SDE)

- One type of stochastic models are stochastic differential equations (SDE).
- The math behind stochastic processes and properly implementing SDE on a computer is tricky.
- If you are interested:
  - "An algorithmic introduction to numerical simulation of stochastic differential equations", D.J. Higham, SIAM Review, Education Section, 43, 2001 (available on his webpage at http://fox.maths.strath.ac.uk/~aas96106/)
  - Kloeden, P.E., Platen, E., 1992. Numerical Solution of Stochastic Differential Equations. Springer, Berlin
  - Publications by Des Higham and Kevin Burrage
- Since SDE are tricky, we will use discrete-time models to play around with noise.

Discrete stochastic model – R example

- The program SISMID-U9-noise.r runs the equations shown below twice.
- Add noise to the 2nd set of equations in any way you want (see previous slides for options).
- Check:
  - How does random number generation and (not) using a seed affect the output?
  - How do different types of noise (additive, multiplicative, etc.) affect the result?
  - How does the magnitude of noise affect the results?

$$U_{t+\tau} = U_t - (bUV)\tau$$

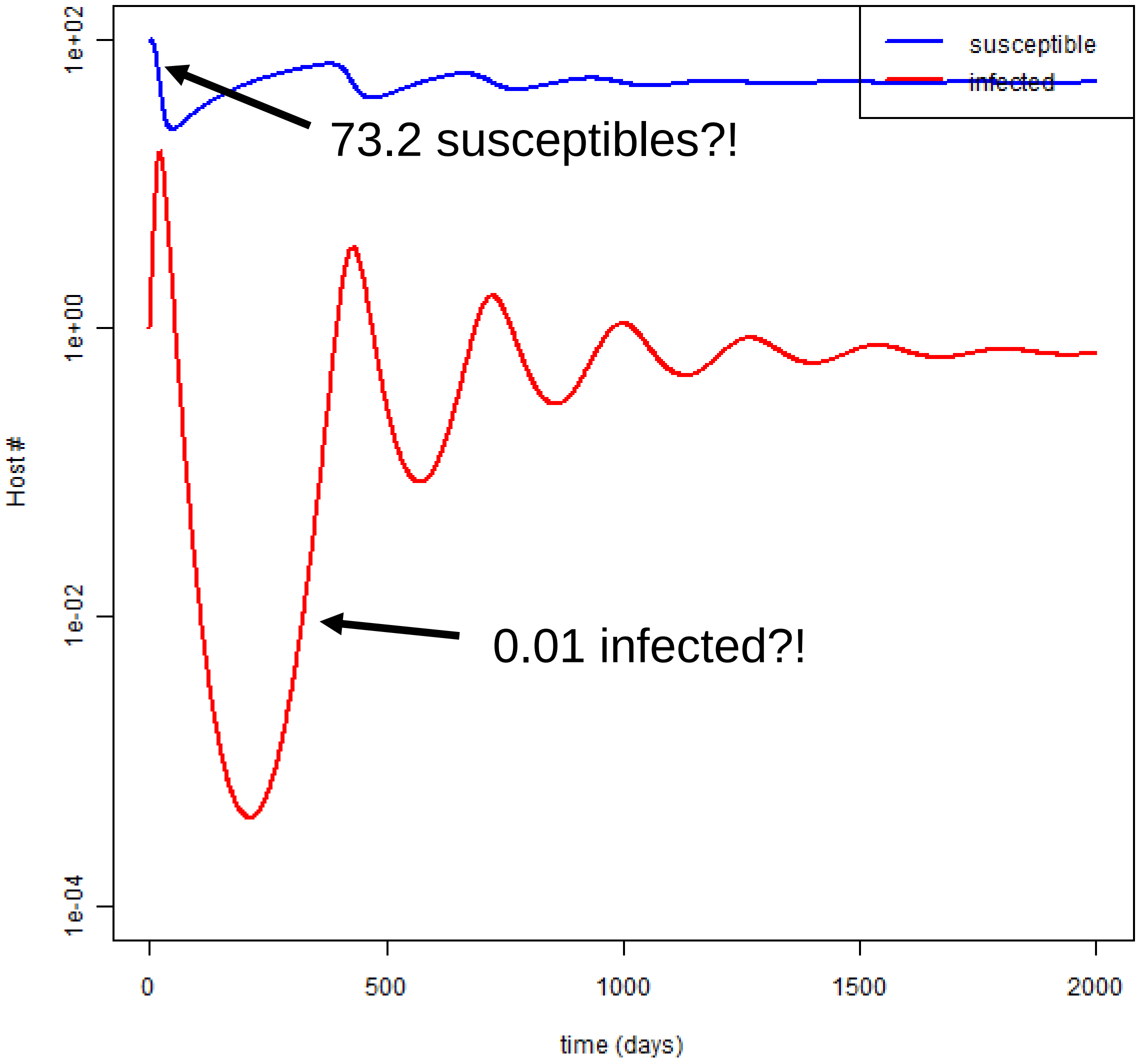
$$I_{t+\tau} = I_t + (bUV - \delta I)\tau$$

$$V_{t+\tau} = V_t + (pI - cV)\tau$$

Stochasticity so far

- Can be added to equations, easy for difference equations, trickier for differential equations.
- Computationally relatively fast.
- Very flexible, noise can be added in many ways.
- Setting noise to zero brings us back to deterministic models -> easy comparison.
- Real systems have inherent demographic stochasticity, even if there is “no noise”.

Demographic stochasticity



73.2 susceptibles?!

0.01 infected?!

Demographic stochasticity

- Discrete events (e.g. birth/death) happen randomly at random times.
- This can be implemented using event-driven approaches.
- The most common used approach is called the Gillespie algorithm (sometimes also referred to as Stochastic Simulation Algorithm, SSA)
  - Gillespie introduced his method(s) in 1977
  - Not much used initially, since it requires fast computers
  - Lots of development in recent years

Event-driven model

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

Transition (reaction/event)	Propensity
$U \to U-1, I \to I+1$	bUV
$I \to I-1$	δI
$V \to V+1$	pI
$V \to V-1$	cV

The propensity (rate) multiplied with the time step gives the probability that a given event occurs

Gillespie's direct method

- Compute all propensities and the sum of all propensities, P_{tot}
- The random time at which the next event occurs is $t=-\text{Log}(\text{RND})/P_{tot}$
- The event that occurs is randomly chosen, with probability proportional to its propensity
- Perform event, update time, return to step 1

Transition (reaction/event)	Propensity
$U \to U-1, I \to I+1$	bUV
$I \to I-1$	δI
$V \to V+1$	pI
$V \to V-1$	cV

Event-driven model: some terminology

Reactions/Events/Transitions

$$R1: \quad U \to U-1, I \to I+1$$

$$R2: \quad I \to I-1$$

$$R3: \quad V \to V+1$$

$$R4: \quad V \to V-1$$

State-change matrix

$$\begin{array}{ccccc} & R1 & R2 & R3 & R4 \\ U & -1 & 0 & 0 & 0 \\ I & +1 & -1 & 0 & 0 \\ V & 0 & 0 & +1 & -1 \end{array}$$

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

Gillespie's method

- The original method is very slow. As soon as numbers get large, it's not feasible anymore.
- There are several ways of speeding up computation
  - Fixed time steps. Easy/fast for computer, but approximation. We need to make sure that only few events occur during the time step. Done by pretty much all Agent-based models.
  - Smart approximations to Gillespie algorithm. Can potentially speed up code by a lot.
  - Switch to Fortran or C (painful).

Evolutionary dynamics

Modeling evolution

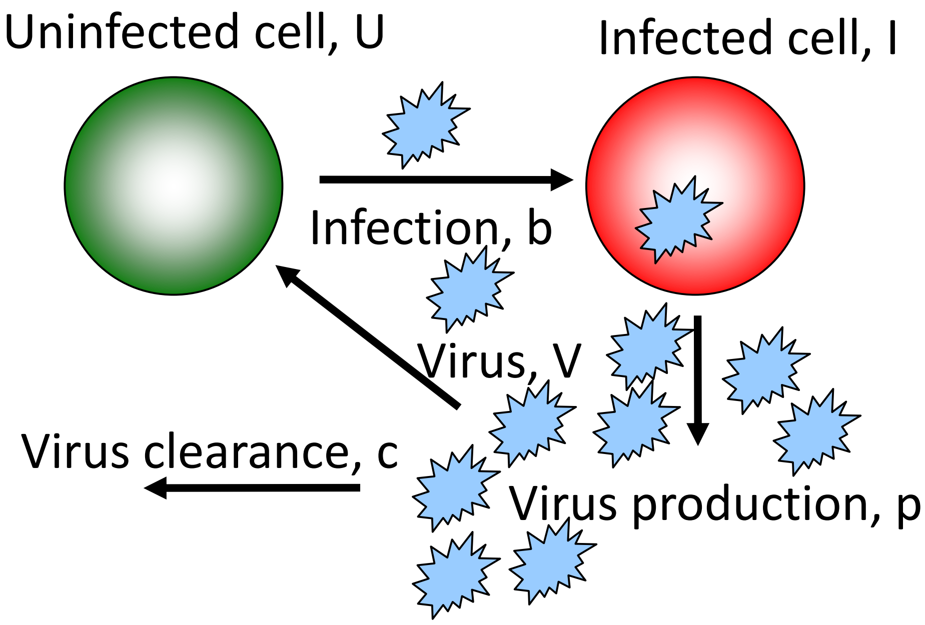
- During evolution, new phenotypes/genotypes usually occur at low frequency initially.
- Often we are interested in probabilities of fixation/extinction.
- This requires a stochastic approach.
- We can apply the previously discussed approaches to some simple evolutionary dynamics questions.

A “real” example: emergence of drug resistance

- Consider an acute virus infection (influenza).
- The host/patient receives drug treatment.
- There is a chance that during the infection a resistant mutant is generated.
- Resistance generation is an unlikely event, initially the resistant mutant starts with low numbers (1).
- Stochastic models are needed/appropriate.
- This is basically “within-host evolution”.

Handel et al. (2007) PLoS Computational Biology

2-strain model for influenza infection

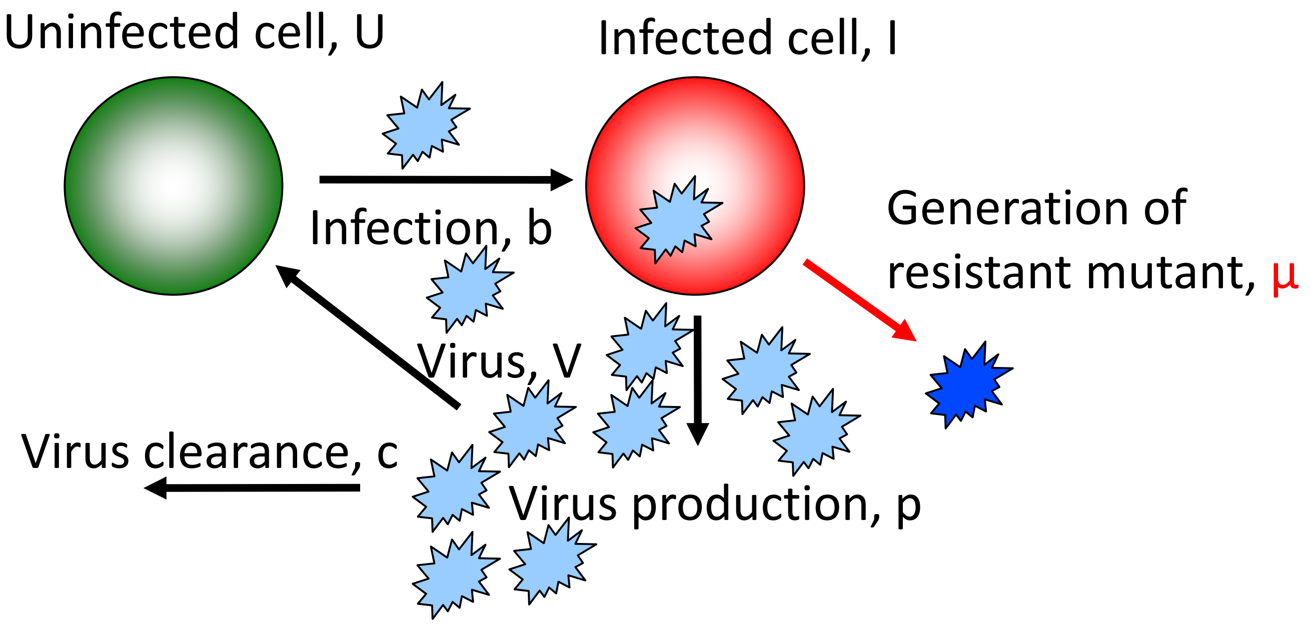


$$\dot{U} = -bUV_s \qquad \text{(uninfected cells)}$$

$$\dot{I}_s = bUV_s - \delta I_s \qquad \text{(drug sensitive infected cells)}$$

$$\dot{V}_s = pI_s - cV_s \qquad \text{(drug sensitive virus)}$$

2-strain model for influenza infection



Uninfected cell, U

Infected cell, I

Infection, b

Generation of resistant mutant, μ

Virus, V

Virus clearance, c

Virus production, p

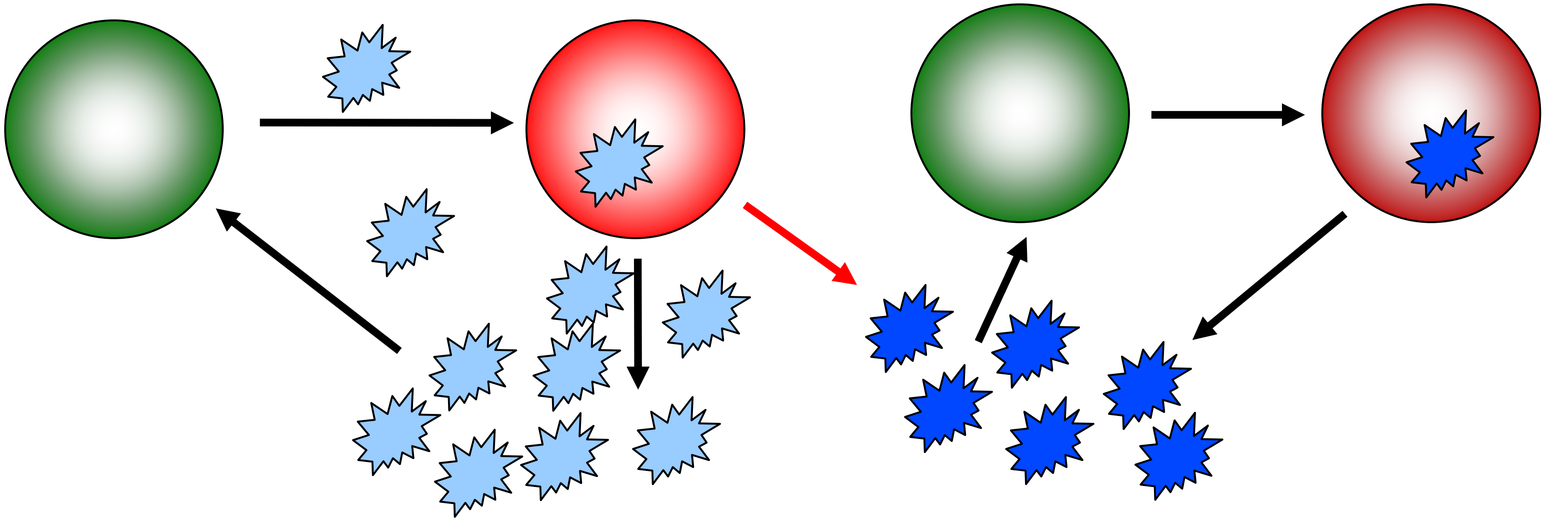
$$\dot{U} = -bUV_s \qquad \text{(uninfected cells)}$$

$$\dot{I}_s = bUV_s - \delta I_s \qquad \text{(drug sensitive infected cells)}$$

$$\dot{V}_s = (1-\mu)pI_s - cV_s \qquad \text{(drug sensitive virus)}$$

$$\dot{V}_r = \mu pI_s \qquad \text{(drug resistant virus)}$$

2-strain model for influenza infection



$$\dot{U} = -bUV_s - bUV_r$$ (uninfected cells)

$$\dot{I}_s = bUV_s - \delta I_s$$ (sensitive infected cells)

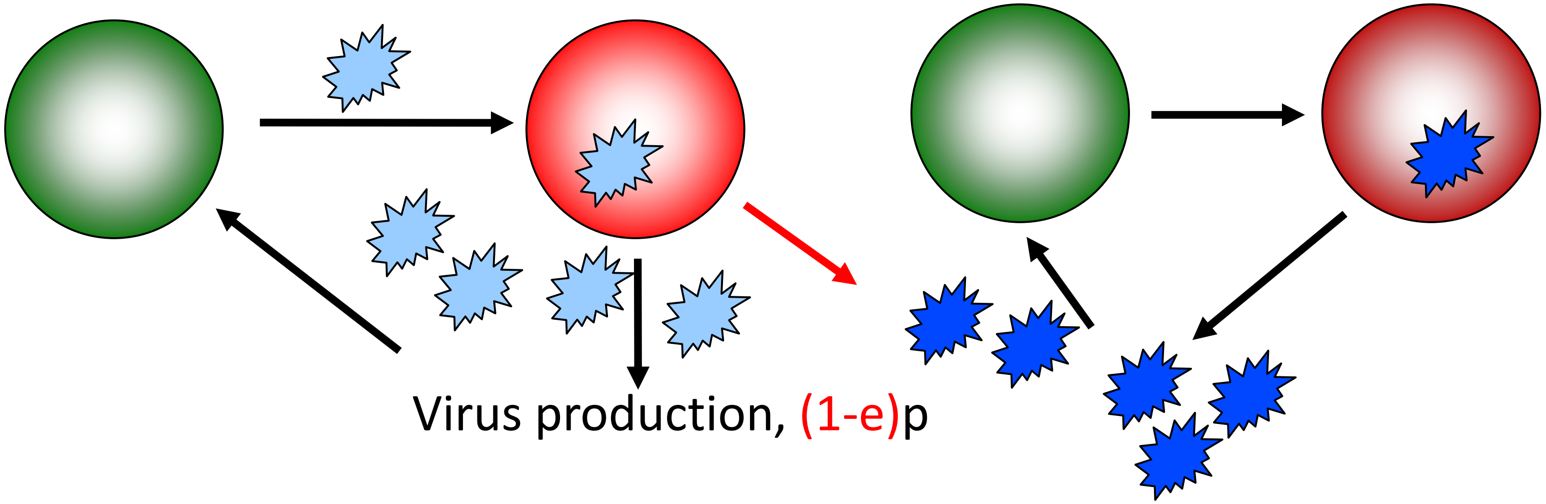
$$\dot{I}_r = bUV_r - \delta I_r$$ (resistant infected cells)

$$\dot{V}_s = (1-\mu)pI_s - cV_s$$ (sensitive virus)

$$\dot{V}_r = \mu pI_s + (1-f)pI_r - cV_r$$ (resistant virus)

Fitness cost

Including antiviral drug treatment



Virus production, (1-e)p

$$\dot{U} = -bUV_s - bUV_r$$ (uninfected cells)

$$\dot{I}_s = bUV_s - \delta I_s$$ (sensitive infected cells)

$$\dot{I}_r = bUV_r - \delta I_r$$ (resistant infected cells)

$$\dot{V}_s = (1-e)(1-\mu)pI_s - cV_s$$ (sensitive virus)

$$\dot{V}_r = (1-e)\mu pI_s + (1-f)pI_r - cV_r$$ (resistant virus)

Drug resistance for influenza

- Let’s try to answer a “real” question: “How does the probability that resistance emerges depend on a given level of treatment?”
- How would we go about answering this?

$$
\begin{aligned}
\dot{U} &= -bUV_s - bUV_r \\
\dot{I}_s &= bUV_s - \delta I_s \\
\dot{I}_r &= bUV_r - \delta I_r \\
\dot{V}_s &= (1-e)(1-\mu)pI_s - cV_s \\
\dot{V}_r &= (1-e)\mu pI_s + (1-f)pI_r - cV_r
\end{aligned}
$$

Drug resistance for influenza

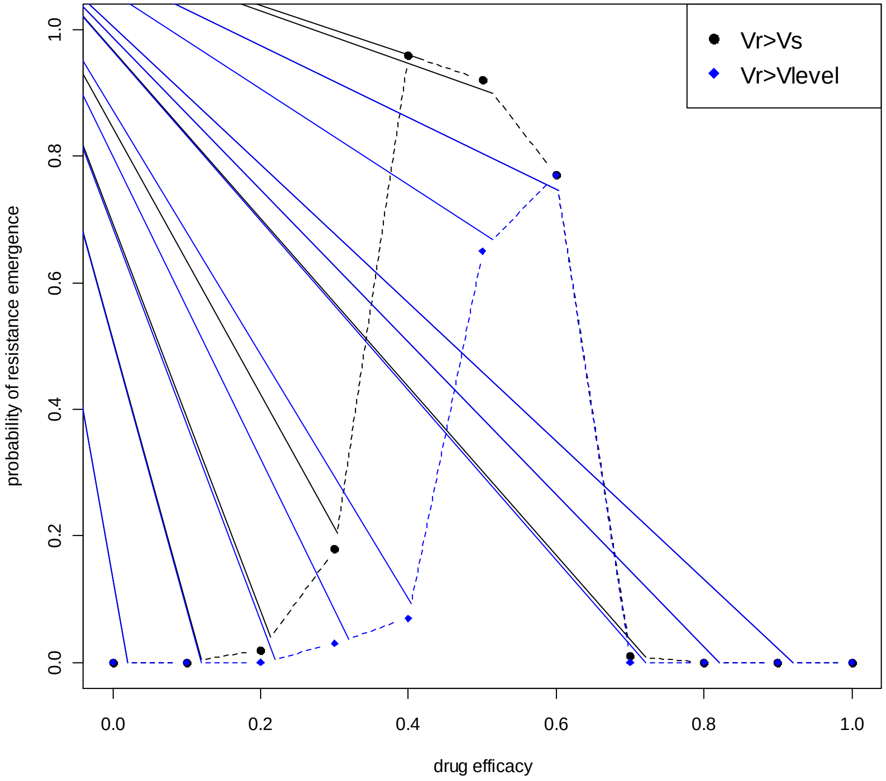
- Let's try to answer a "real" question: "How does the probability that resistance emerges depend on a given level of treatment?"
- How would we go about answering this?
  - 1. Set ***e*** to some value.
  - 2. Run the stochastic simulation N times.
  - 3. Record for how many of those N simulations the resistant strain emerged (we need to define this).
  - 4. Change ***e***.
  - 5. Go back to step 2 until we have done the simulation for a number of ***e*** values (e.g. between 0 and 1 in steps of 0.1).
  - 6. Plot result: ***e*** on the x-axis, fraction of simulations for which we had resistance emergence on y-axis.

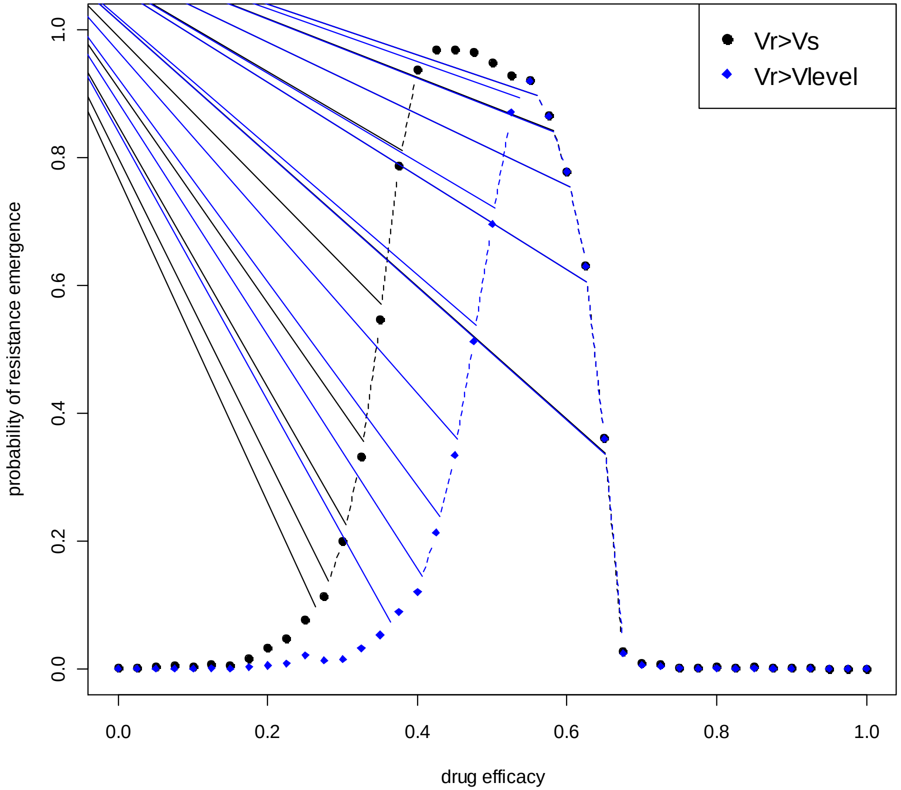
$$\dot{U} = -bUV_s - bUV_r$$
$$\dot{I}_s = bUV_s - \delta I_s$$
$$\dot{I}_r = bUV_r - \delta I_r$$
$$\dot{V}_s = (1-e)(1-\mu)pI_s - cV_s$$
$$\dot{V}_r = (1-e)\mu pI_s + (1-f)pI_r - cV_r$$

Drug resistance for influenza - R

- The “pure” Gillespie method is too slow for this project, even with unrealistically low numbers for cells/virus.
- A smart approximate method exists in the **adaptivetau** package in R.
- Install **adaptivetau** if you haven’t already.
- Open SISMID-U9-evolution.r. Read & understand the code.
- Run the model, make sure you understand the results.

Drug resistance for influenza - R





Stochasticity - Discussion

- Noise/randomness enters in many ways: Observation error, internal/external fluctuations, demographic stochasticity.
- Stochasticity can be implemented in models in different ways. It always makes the model somewhat more difficult and slower to run.
- If you have a question for which you think ODEs are a good approximation, start with those.
- If you have a system or question where stochasticity is important, you need to use some kind of stochastic approach (SDE, discrete model with noise, purely stochastic (Gillespie), Agent-based model…)

Stochastic models and data

- One can fit data to stochastic models.
- Most work so far has been done on the population/between-host/epidemiology level under the label of infectious disease inference.
- It's complicated, definitely beyond this module. If you want to learn more:
  - "Design and Analysis of Vaccine Studies" by Halloran, Longini and Struchiner (2009), Springer – currently most comprehensive book on the topic
  - "Analysis of Infectious Disease Data" by Niels Becker (1989) – good theory, somewhat outdated with regard to computational aspects
  - "Bayesian Analysis for Emerging Infectious Diseases" by Jewell et al. (2009) Bayesian Analysis
  - "Inference in Epidemic Models without Likelihoods" by McKinley et al. (2009) International Journal of Biostatistics
  - Some of the other SISMID modules

Further reading

- Bolker (2008) “Ecological Models and Data in R” (covers some fitting of stochastic models to data)
- Keeling and Rohani (2008) “Modeling Infectious Diseases”, chapter 6
- Gillespie algorithm: Gillespie (1977) Journal of Physical Chemistry
- GillespieSSA package: Pineda-Krch (2008) Journal of Statistical Software
- Some people/groups who work on state-of-the art stochastic/hybrid solvers: Linda Petzold, Daniel Gillespie, Yang Cao, Kevin Burrage, Yiannis Kaznessis,…

Models with memory

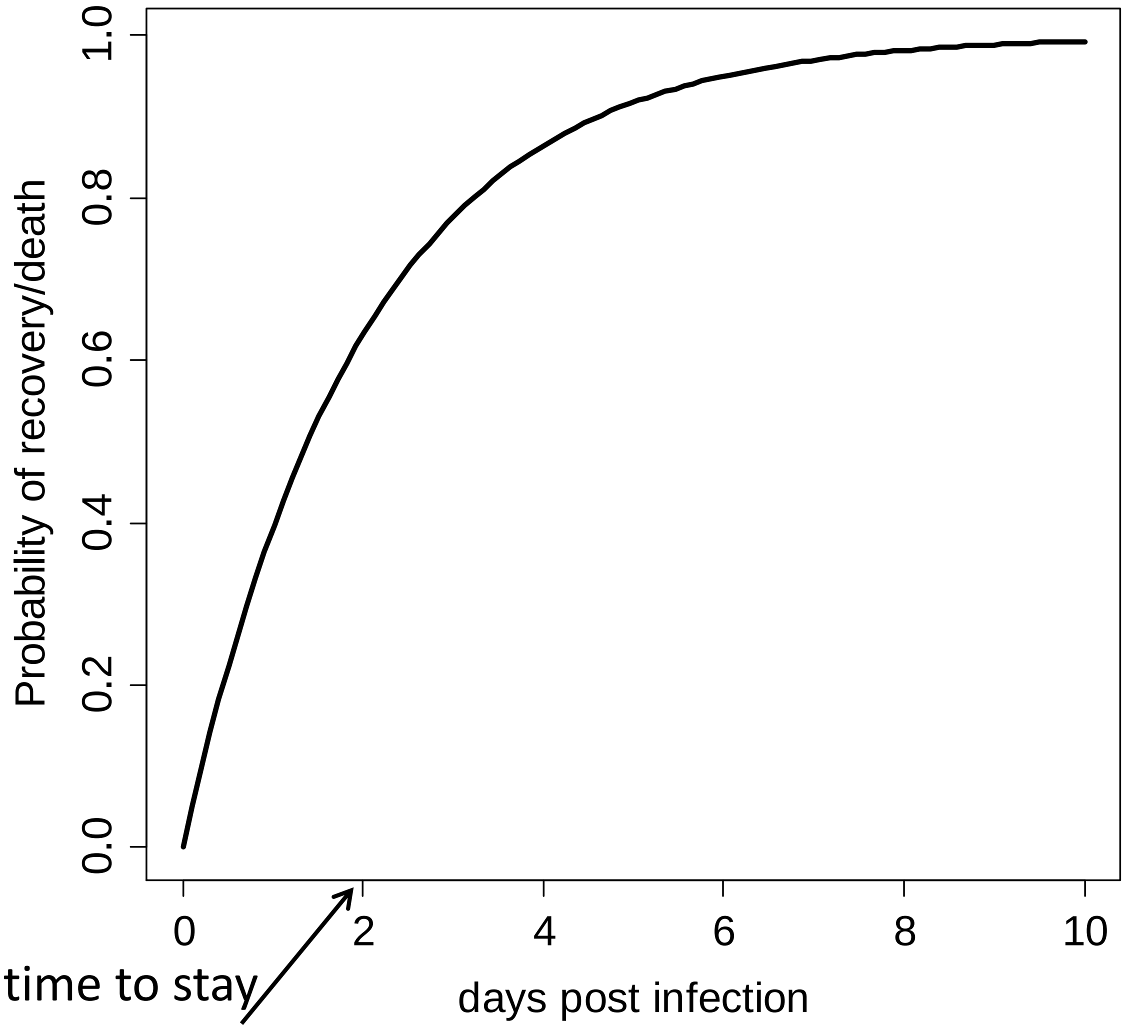
Limitations of standard ODE models

- Simple ODE models are “memory-less”, the dynamics of the system only depends on the **current** state of the system.
- Example: For the simple within-host model we have looked at, infected cells die at a constant rate, δ. A cell that was infected 10 seconds has the same chance of dying as a cell that was infected 10 hours ago.

$$\dot{I} = bUV - \delta I$$

Limitations of standard ODE models

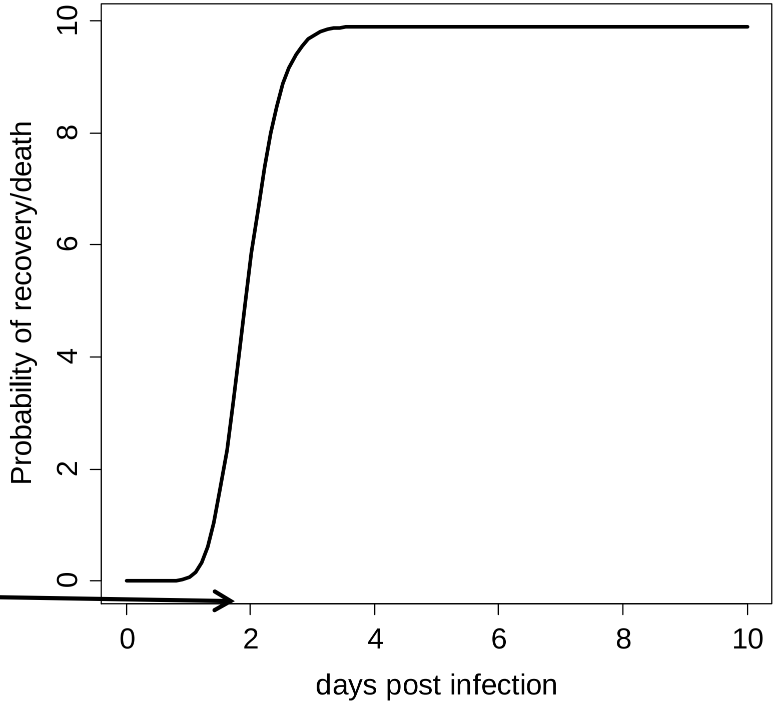
$$\dot{I} = bUV - \delta I$$



Average time to stay in compartment I

Realistic transitions

- It is more likely that an infected cell dies after a certain time, with some variation.
- To take this into account, we need to keep track of the time/age since infection, i.e. our model needs to include some form of memory.



Average time to stay in compartment I

Tracking age - PDEs

- One way to explicitly specify an age since infection in the model leads to a partial differential equation (PDE)

$$\frac{\partial I(t,\tau)}{\partial t} = -\delta(\tau) I(t,\tau) - \frac{\partial I(t,\tau)}{\partial \tau}$$

Death rate depends on age since infection

Age since infection

“aging” of infecteds

$$\frac{dI(t)}{dt} = -\delta I(t)$$

Comments on PDEs

- Mathematically “elegant”, some analytics is often possible but more challenging than ODEs.
- More difficult to implement numerically. Only rudimentary support in R, e.g. solvers in deSolve package.
- There is another, simpler way to fix the problem with the infected cell life-span, based on introducing additional equations which represent dummy compartments.

Dummy compartments for a realistic life-span

$$\dot{U} = \lambda - dU - bUV$$

$$\dot{I}_1 = bUV - n\delta I_1$$

$$\dot{I}_2 = n\delta I_1 - n\delta I_2$$

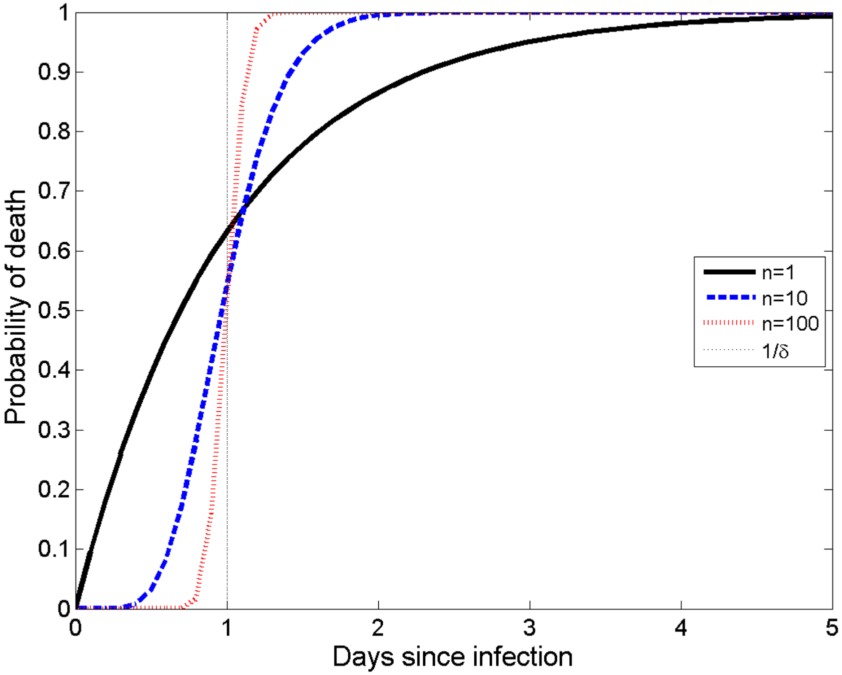
$$\dot{I}_3 = n\delta I_2 - n\delta I_3$$

$$\vdots$$

$$\dot{I}_n = n\delta I_{n-1} - n\delta I_n$$

$$\dot{V} = \sum_{i=1}^{n} p_i I_i - cV$$

n dummy compartments. Infected cells spend 1/**nδ** in each compartment, with a total mean duration of **n/nδ=1/δ**, as before. For n=1, we have the previous model. As **n** gets larger, the lifespan becomes more concentrated around the mean value.



Lloyd (2001a,b) Proc Soc B

Delay Differential equations (DDE)

- Sometimes it is useful to consider time lags.
  - Example: Production of new virions starts some time after a cell has become newly infected.
- Now the dynamics of the system depends on the **current and past** state of the system. Again, ODE’s can’t do that because they are “memory-less”. To keep track of past states, one needs delay differential equations (DDE).

$$\dot{U} = \lambda - dU - bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI(t-\tau) - cV$$

Production of virus proportional to number of infected cells time τ ago.

Using DDEs

- The most tedious way is to write your own DDE solver.
- A better way is to use an existing solver, for instance dede() in deSolve or the package PBSddesolve.
- Another option is to rewrite DDE using dummy compartments. This is the same trick as the one I mentioned for PDEs. It is sometimes called the “linear chain trick”.

Linear Chain trick for delays

$$\dot{U} = \lambda - dU - bUV$$

$$\dot{I}_1 = bUV - ngI_1$$

$$\dot{I}_2 = ngI_1 - ngI_2$$

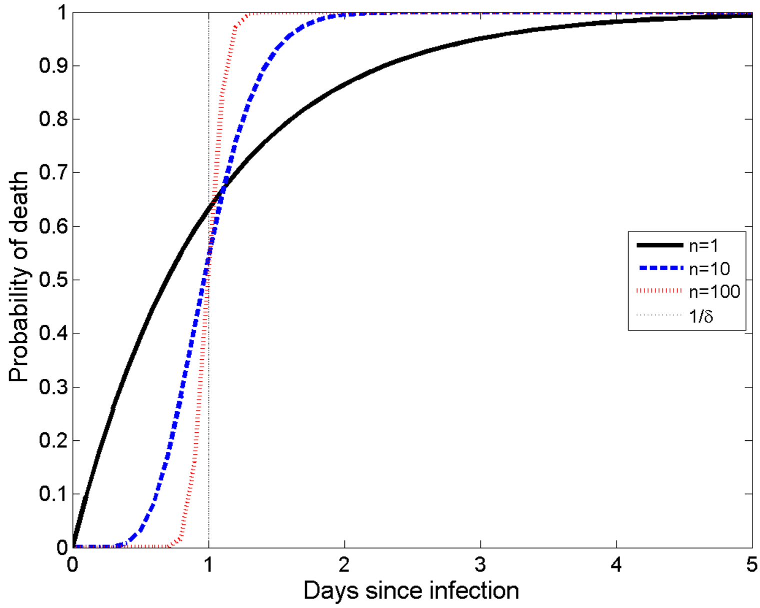
$$\dot{I}_3 = ngI_2 - ngI_3$$

$$\vdots$$

$$\dot{I}_f = ngI_n - \delta I_f$$

$$\dot{V} = pI_f - cV$$

- Only the last compartment produces virus.
- Since it takes a newly infected cell **n/ng=1/g** amount of time before it reaches the last compartment, we have implemented a delay between cells becoming infected and starting to produce virus.
- Similar to having a latent compartment with non-exponential transitions (see dummy compartment example above).



Models with memory – R example

- Open SISMID-U9-memory.r
- Try to understand the code. The situation/model is again the simple within-host virus model. The model is run twice, once with a DDE using the built-in DDE solver (dede) from the deSolve pacakge, once with dummy compartments.
- Run the model & see how different delays and dummy compartments affect the results.

$$\dot{U} = \lambda - dU - bUV$$
$$\dot{I} = bUV - \delta I$$
$$\dot{V} = pI(t-\tau) - cV$$

$$\dot{U} = \lambda - dU - bUV$$
$$\dot{I}_1 = bUV - ngI_1$$
$$\dot{I}_2 = ngI_1 - ngI_2$$
$$\vdots$$
$$\dot{I}_f = ngI_n - \delta I_f$$
$$\dot{V} = pI_f - cV$$

Comments on memory models

- The simple memory-less ODE model is unrealistic, however, sometimes it’s a decent approximation.
- The more realistic models don’t change steady states, but can change the dynamics.
- If used for data fitting, the simple model and the more realistic model can produce different results, for instance lead to different estimates for R_0.
- Further reading: *Alun Lloyd (2001a,b) Proc Soc B, Helen Wearing et al. (2005) PLoS Medicine*

Spatially explicit Models

So far – homogenous models

- Different entities, such as virus/cells or uninfected/infected hosts, were assumed to be well mixed and bump into each other randomly (mass-action) and “live” in a homogenous space.
- As we saw, mass-action was not ideal (recall the HIV example) but fixing some problems with mass-action by using saturating functions still assumed essentially homogeneous mixing.
- If we want to consider spatial structure, we need to use spatially explicit models.

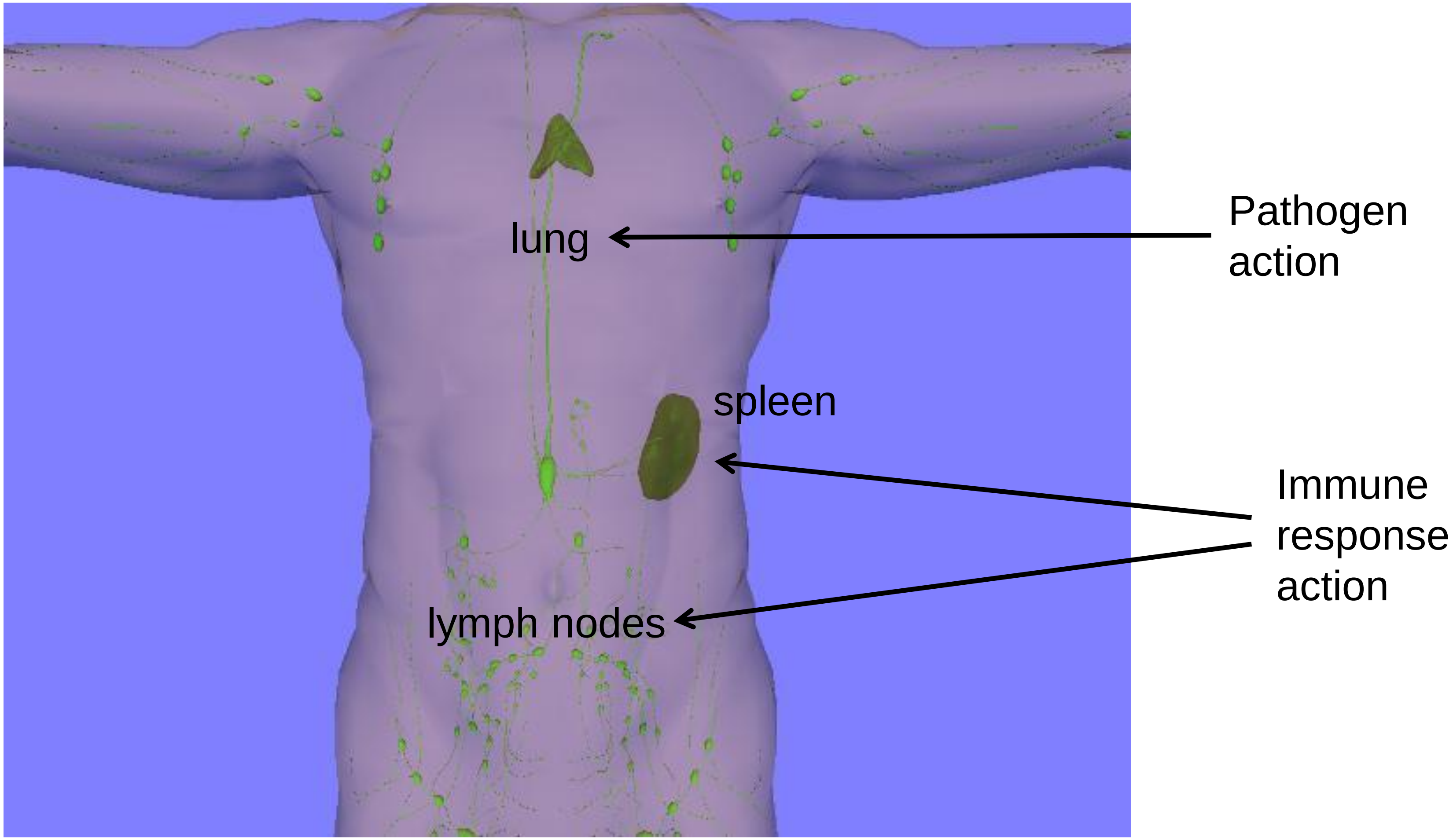
Spatial Models

- Metapopulation/patch models
  - Use ODEs or discrete time/stochastic compartmental models to simulate dynamics of populations in different distinct sites.
  - Migration/interaction terms couple the equations/sites.
  - Simple and straightforward extension of non-spatial compartmental models.
  - Still non-spatial within a given site.
- Partial differential equations
  - Space is an explicit dimension of the model.
  - One can sometimes do some analytics (but it’s not easy).
  - For certain situations realistic enough.
  - Potentially difficult to implement on a computer.
- Agent-based (individual-based) models
  - These models can be the most detailed/realistic.
  - Almost no analytics is possible, purely computational.
  - Needs sufficient computational resources, especially for large populations.

Metapopulation models

- On each patch/site, a dynamical process occurs. Sites/populations are coupled to form a metapopulation.
- Entities can move between sites
- Entities from different sites can interact

Modeling multiple "locations"



Modeling multiple “locations”

$$\dot{U} = -bUV_l$$

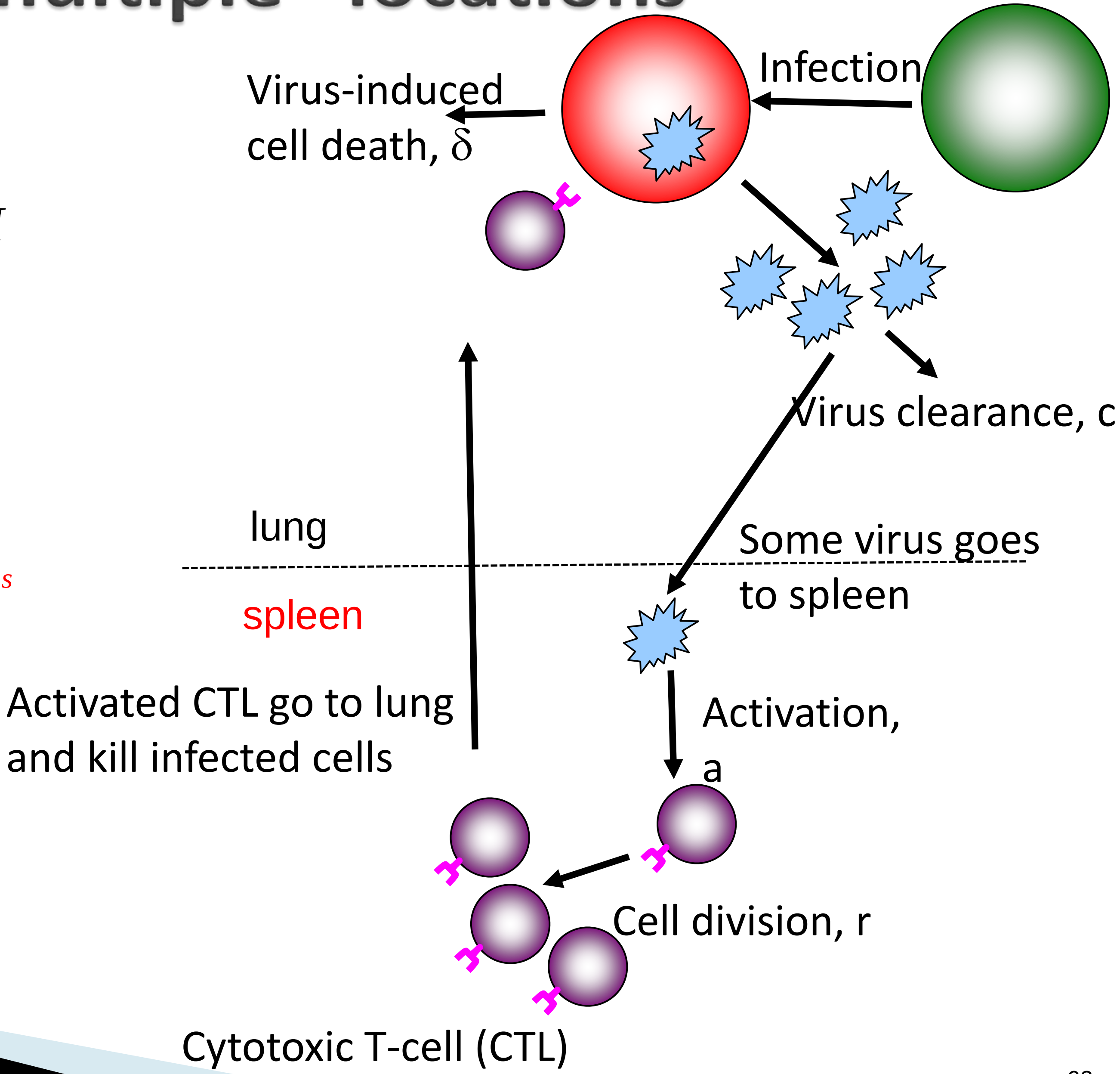
$$\dot{I} = bUV_l - \delta I - kX_l I$$

$$\dot{V}_l = pI - c_l V_l - mV_l$$

$$\dot{X}_l = fX_s - dX_l$$

$$\dot{V}_s = mV_l - c_s V_s$$

$$\dot{X}_s = aV_s + rX_s - fX_s$$


Infection
Virus-induced cell death, δ
Virus clearance, c
lung
spleen
Some virus goes to spleen
Activation, a
Activated CTL go to lung and kill infected cells
Cell division, r
Cytotoxic T-cell (CTL)

Agent-based Models

Agent-based models (ABM)

- For ABM, every individual/agent is modeled explicitly.
- ABM are very flexible, they can be very detailed and realistic.
- One can usually not write down equations, ABM are almost purely computational.
- Since ABM are complex, they have the usual drawbacks (many parameters, many unknowns).
- To run AMB on a computer usually requires lots of CPU power.

Writing ABM

- In principle, any programming language can be used.
- If speed is crucial, use Fortran/C (but the programming can be **very** tedious).
- R, Matlab and similar languages make programming somewhat easier, but the code is still much more involved compared to simple compartmental models.
- Specialized ABM languages exist: NetLogo, Swarm, Repast, MASON,…

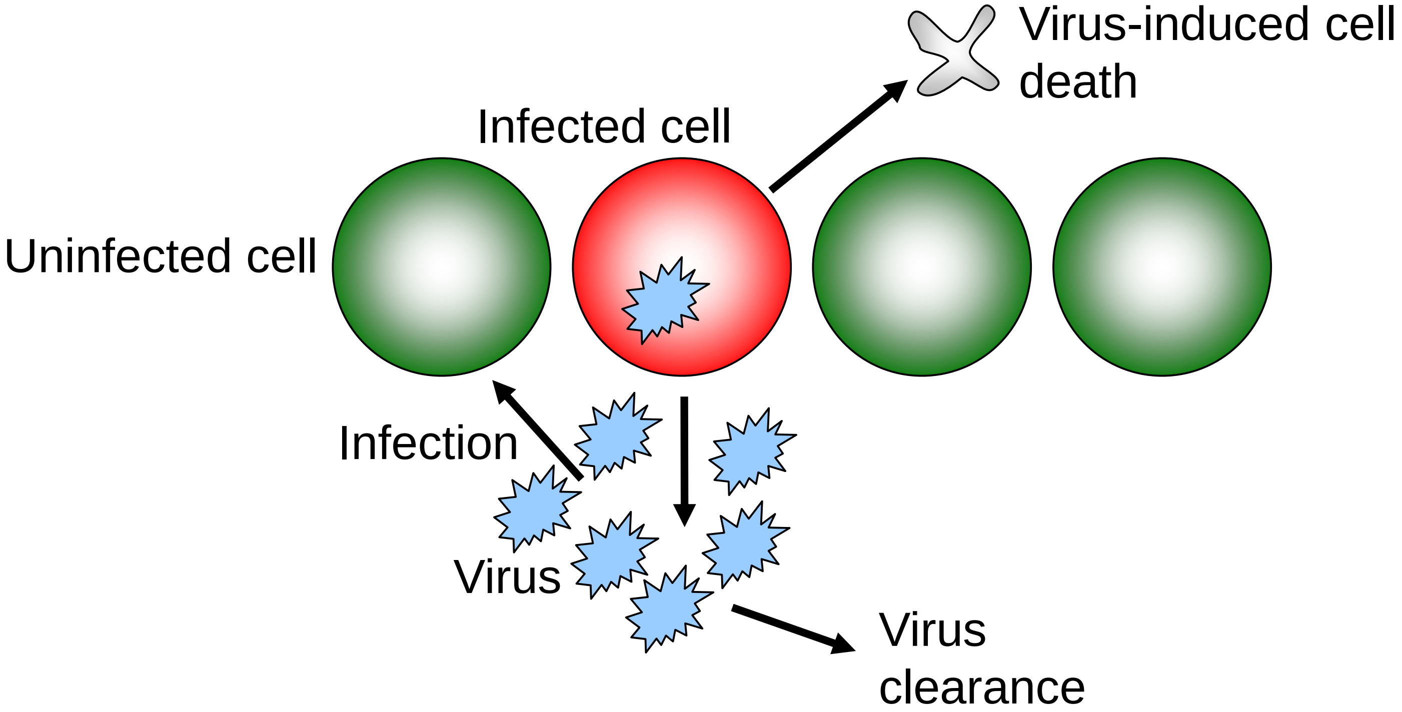
NetLogo

- Based on the Logo programming language
- Mainly meant as teaching tool
- Very user-friendly, easy to program
- Free! (http://ccl.northwestern.edu/netlogo/)
- Powerful enough to do some science/research with it
- Many model examples
- Models are mostly stochastic, but can be deterministic. Time step is **fixed** – not event-driven like Gillespie algorithm (ABM with Gillespie are usually way too CPU intensive).

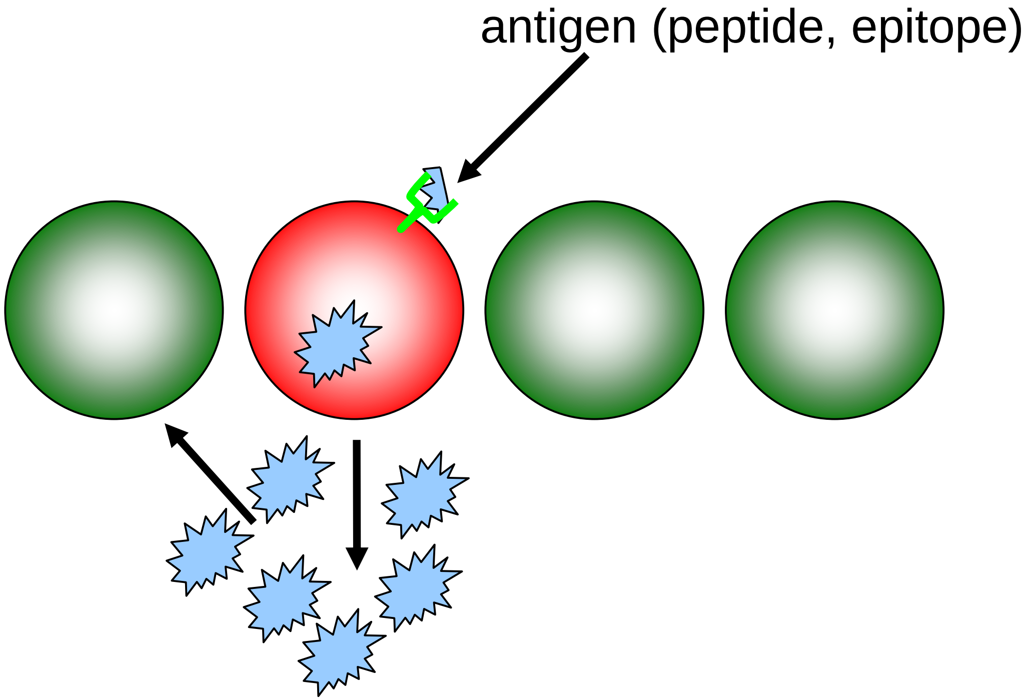
NetLogo and research – an example

- Based on: “Sharing the burden: Antigen transport and firebreaks in immune responses” A. Handel, A. Yates, SS. Pilyugin, R. Antia (2009), Journal of the Royal Society Interface

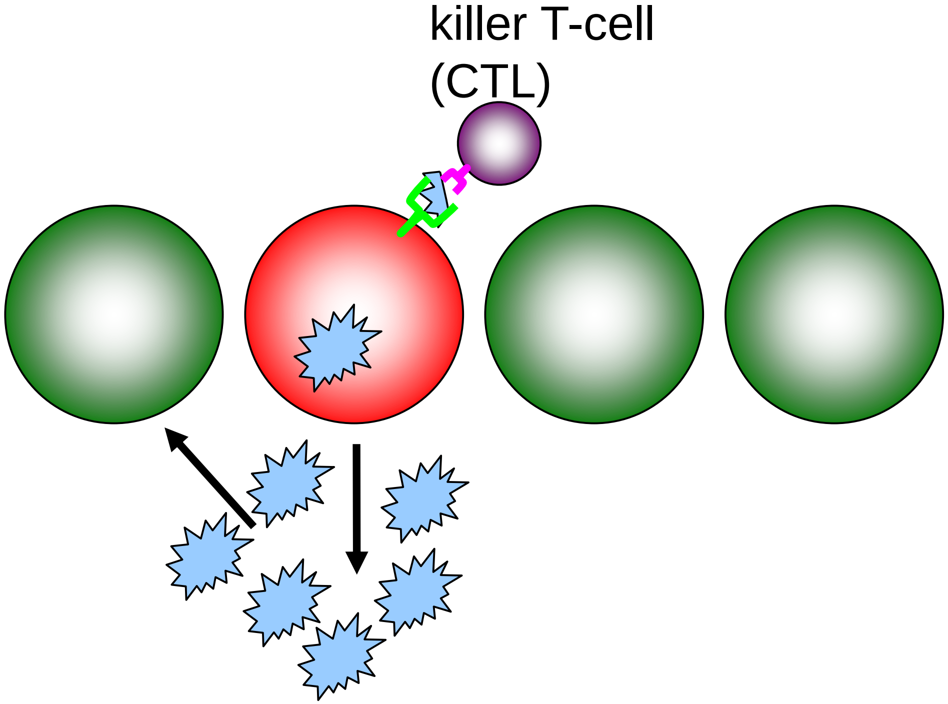
A virology/immunology primer



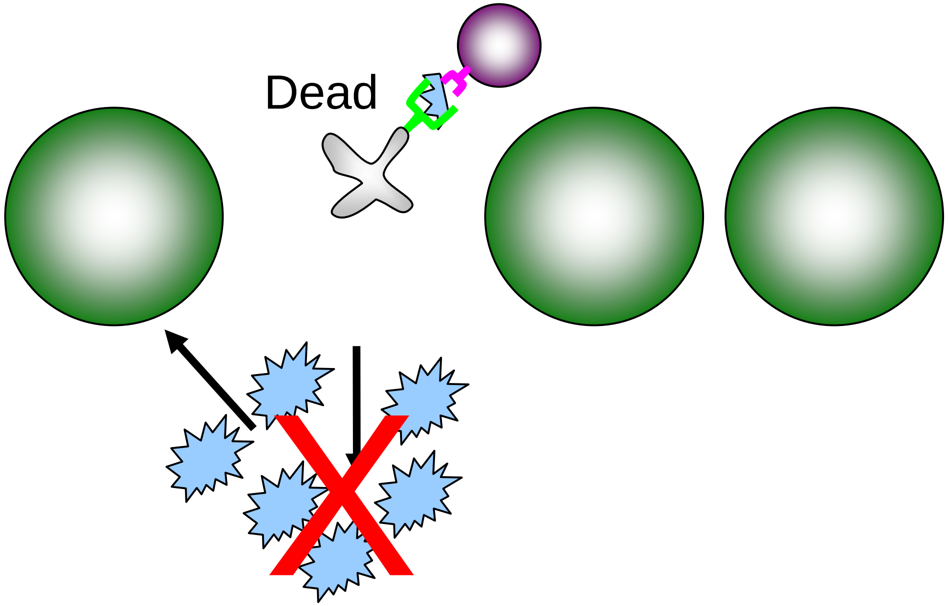
A virology/immunology primer



A virology/immunology primer

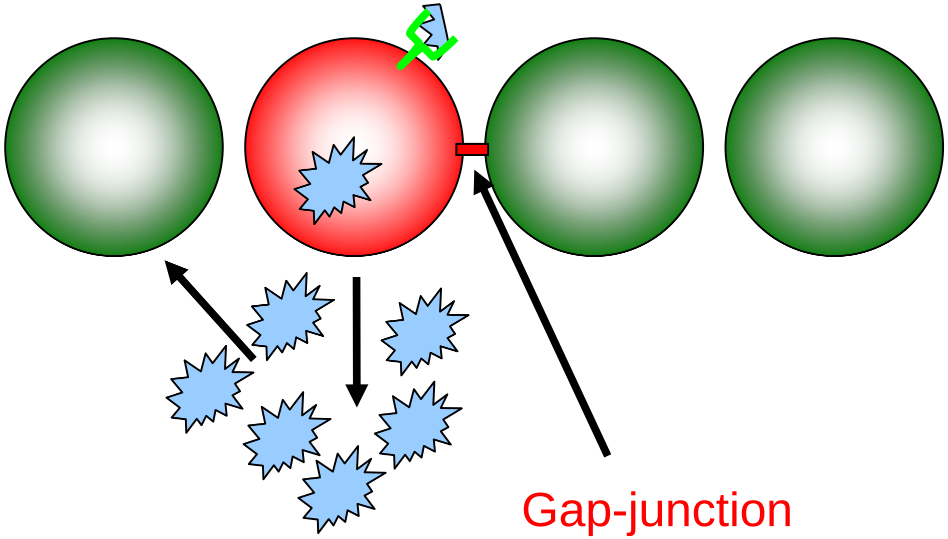


A virology/immunology primer

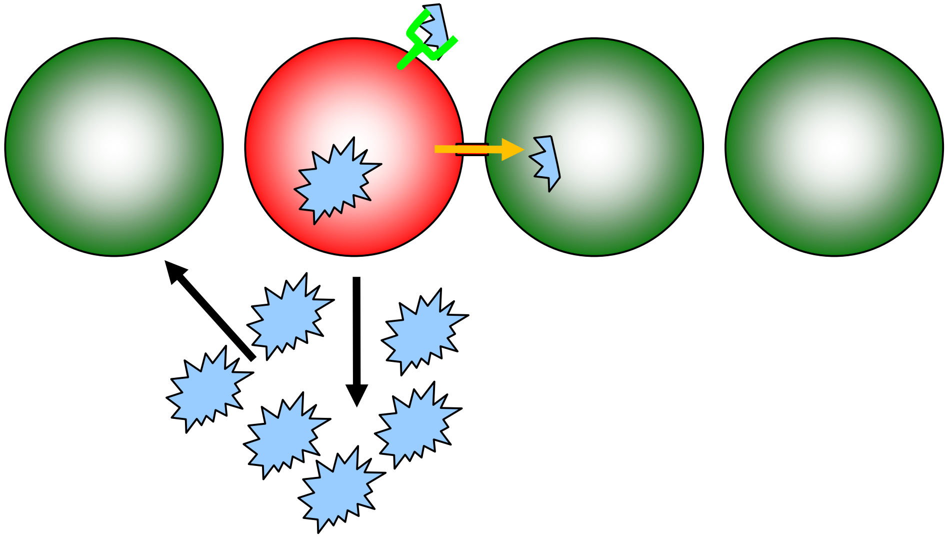


The good: Virus production interrupted
The bad: A dead cell (immunopathology)

Gap-junctions during viral infection

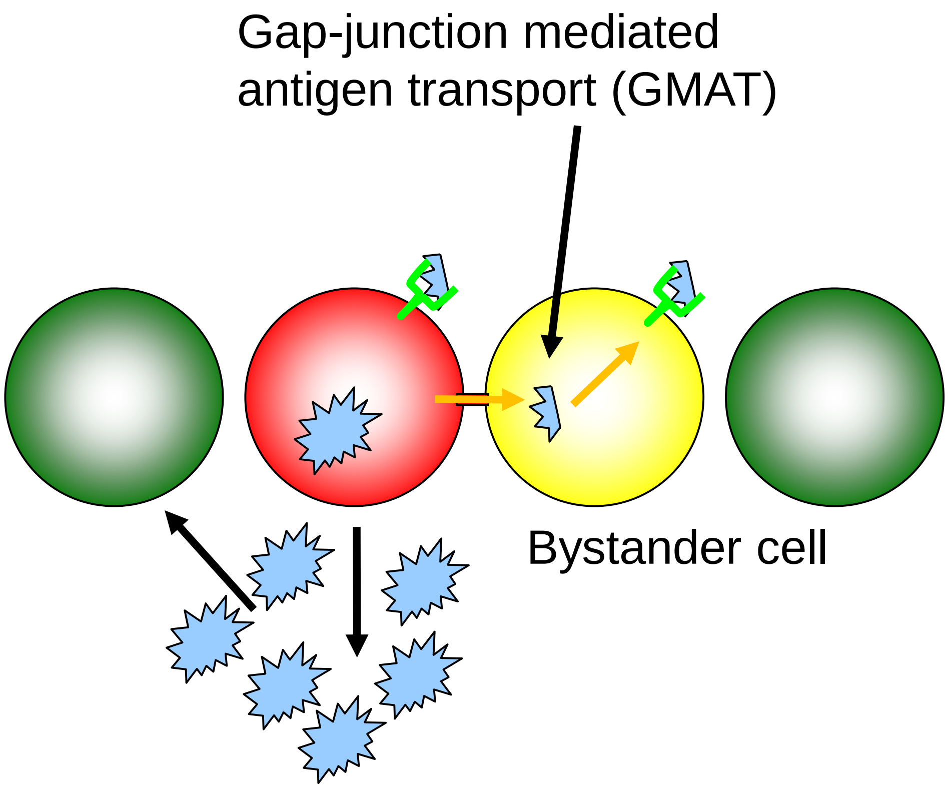


Gap-junctions during viral infection



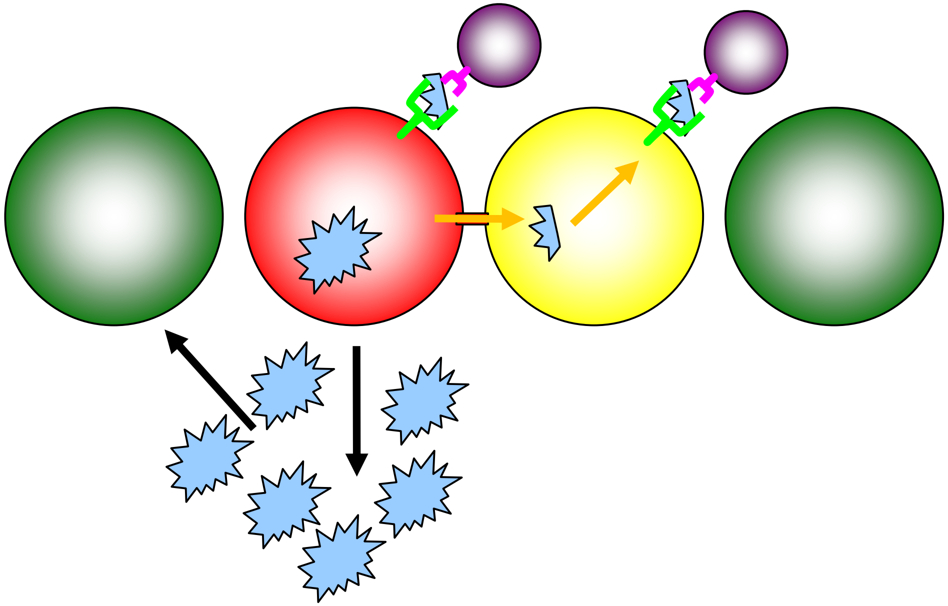
Neijssen et al. (2005) Nature

Gap-junctions during viral infection

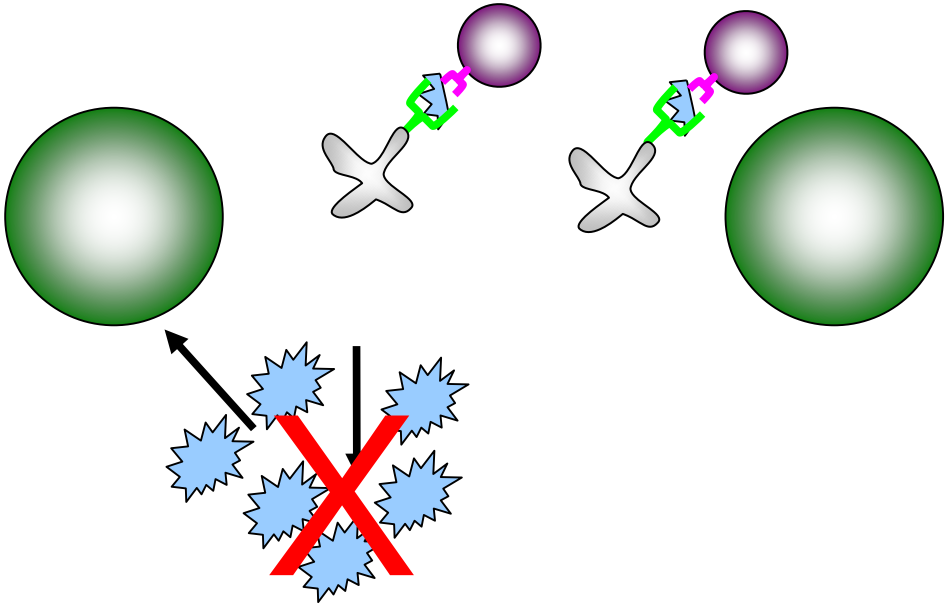


Neijssen et al. (2005) Nature

Gap-junctions during viral infection



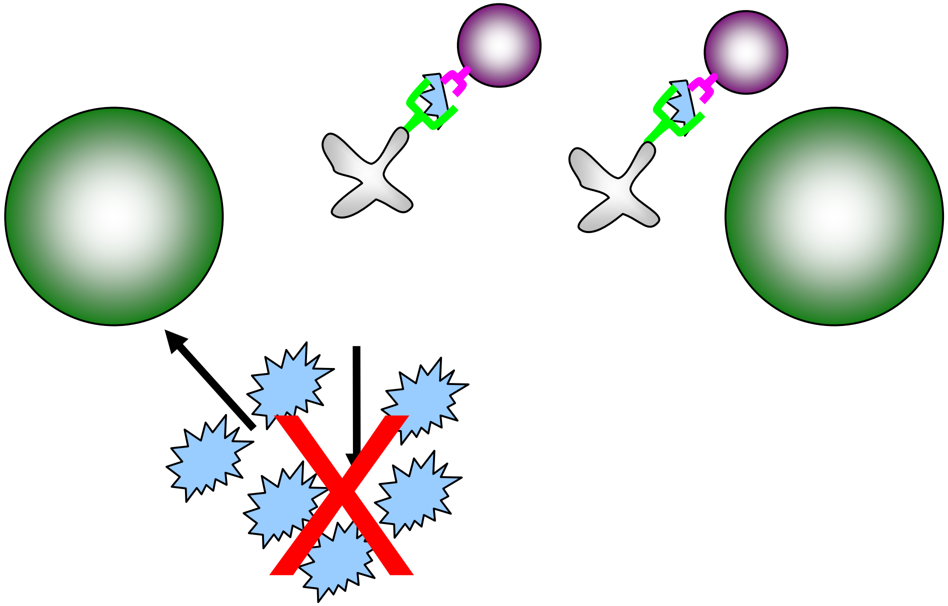
Gap-junctions during viral infection



The good: Virus production interrupted before it even started

The bad: 1) More dead cells.
2) CTL that are busy killing bystander cells can’t kill infected cells

Gap-junctions during viral infection



Question: Are gap-junctions potentially useful for the host?

A mathematical model of the system

$$\dot{U} = -bUV - \boxed{g}UI \qquad \text{(uninfected cells)}$$

$$\dot{I} = bUV + bBV - dI - kXI \qquad \text{(infected cells)}$$

$$\dot{B} = \boxed{g}UI - bBV - kXB \qquad \text{(bystander cells)}$$

$$\dot{C}_I = kXI - \delta C_I \qquad \text{(CTL - infected cell complex)}$$

$$\dot{C}_B = kXB - \delta C_B \qquad \text{(CTL - bystander cell complex)}$$

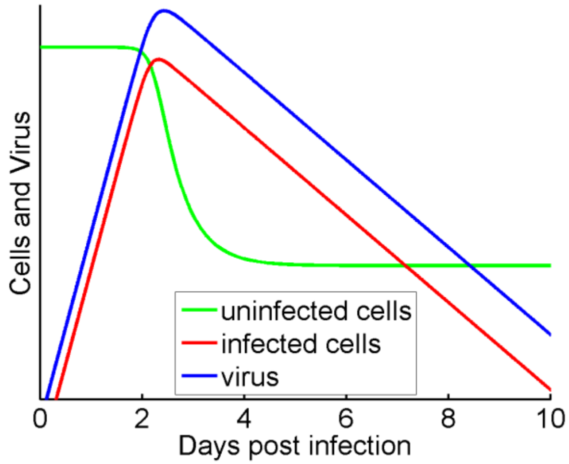
$$\dot{V} = p(I + C_I) - cV \qquad \text{(free virus)}$$

$$\dot{T} = rT \qquad \text{(total CTL)}$$

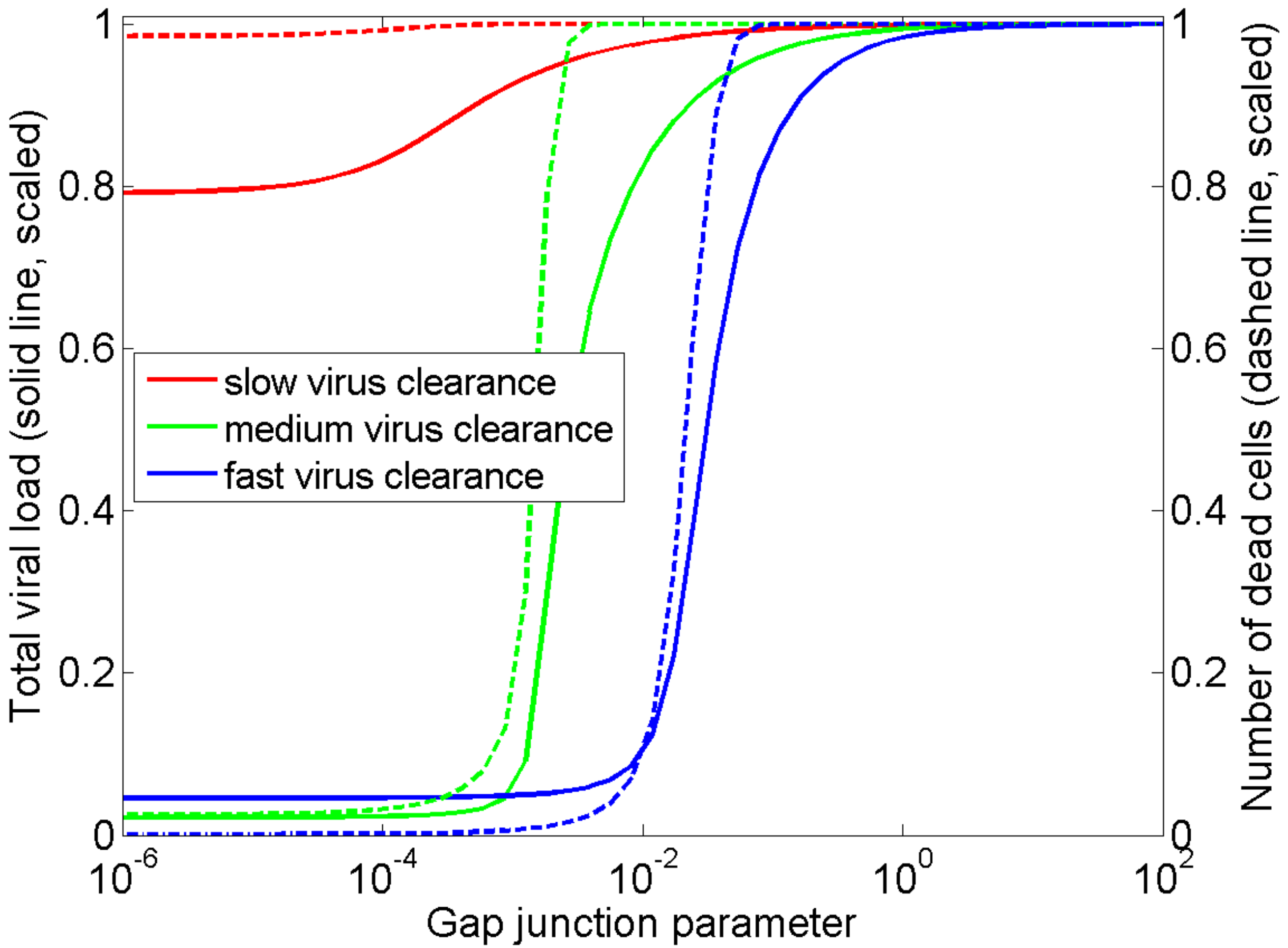
$$X = T - C_I - C_B \qquad \text{(free CTL)}$$

Analysis

- Run the model/simulation
- Record total virus load (area under curve) and total number of dead cells
- Do that for different values of the gap-junction parameter g



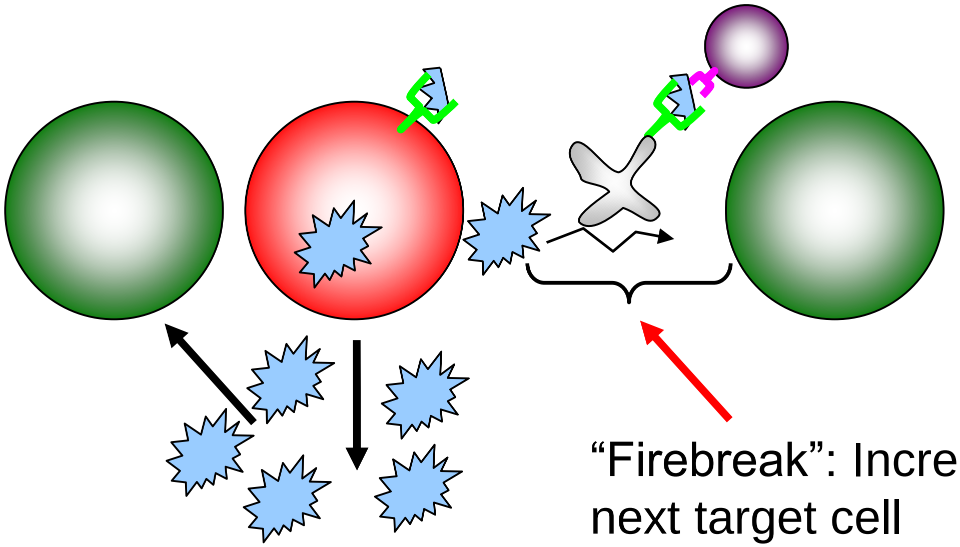
Results


slow virus clearance
medium virus clearance
fast virus clearance
Total viral load (solid line, scaled)
Number of dead cells (dashed line, scaled)
Gap junction parameter
0
0.2
0.4
0.6
0.8
1
10^{-6}
10^{-4}
10^{-2}
10^{0}
10^{2}

So far, so bad

- The model suggests that virus load and immunopathology increase with increased gap junction-mediated antigen transport (GMAT).
- BUT: We have not yet considered spatial effects.

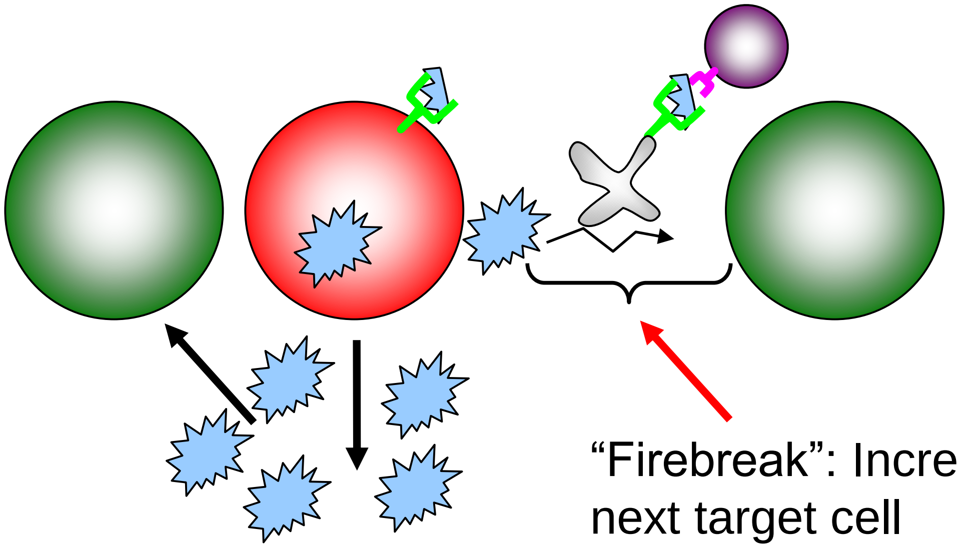
Gap-junctions as firebreaks



“Firebreak”: Increased distance to next target cell

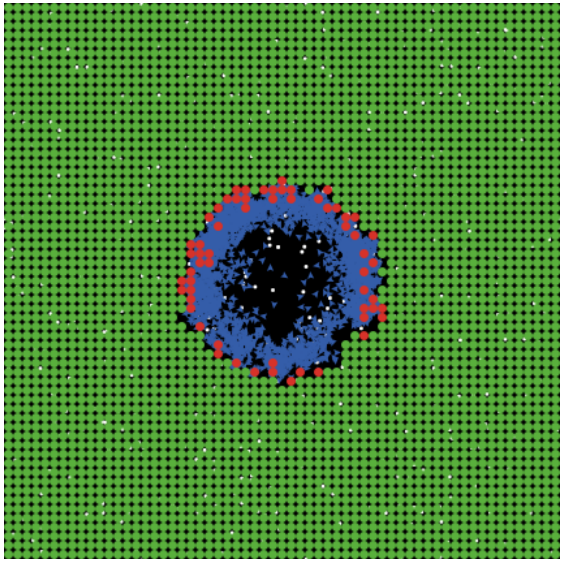
Gap-junctions as firebreaks

- To study firebreaks, an ODE model does not work well. We need a model that includes space.



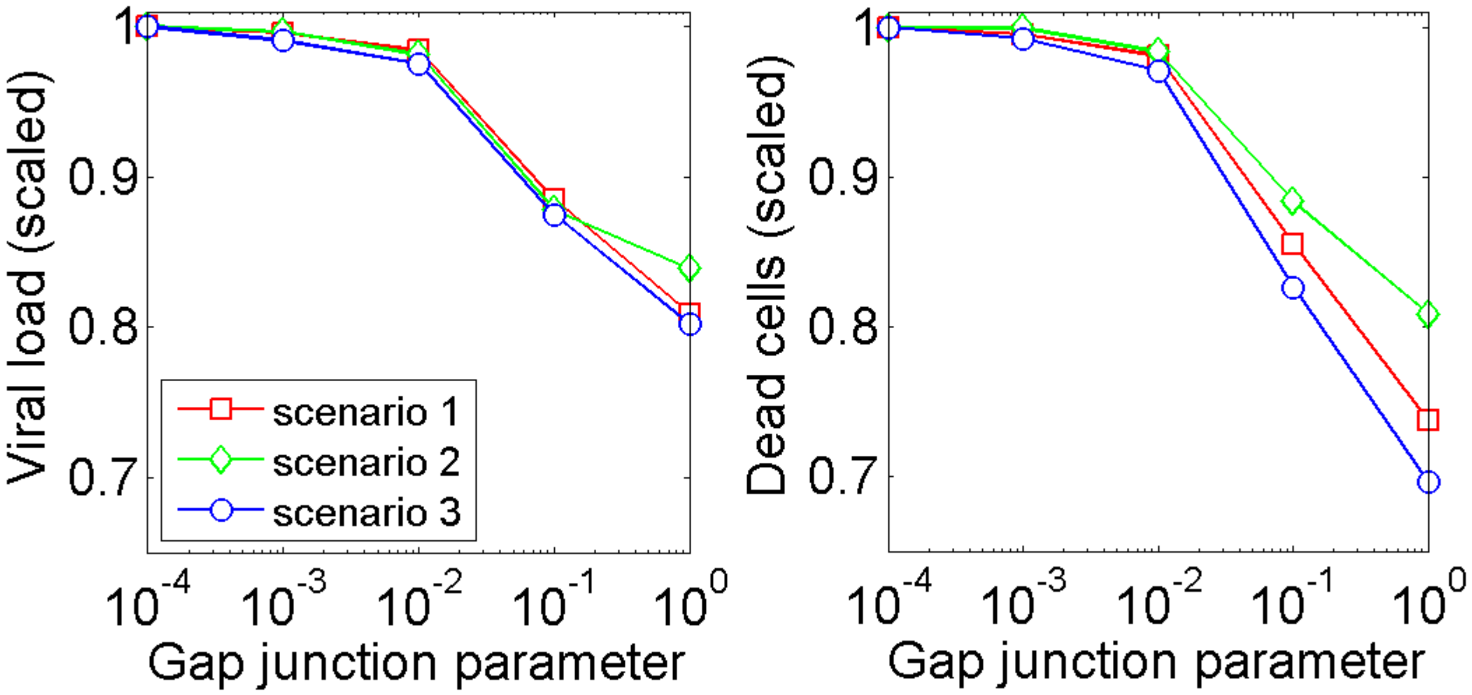
“Firebreak”: Increased distance to next target cell

An agent-based model for gap-junctions



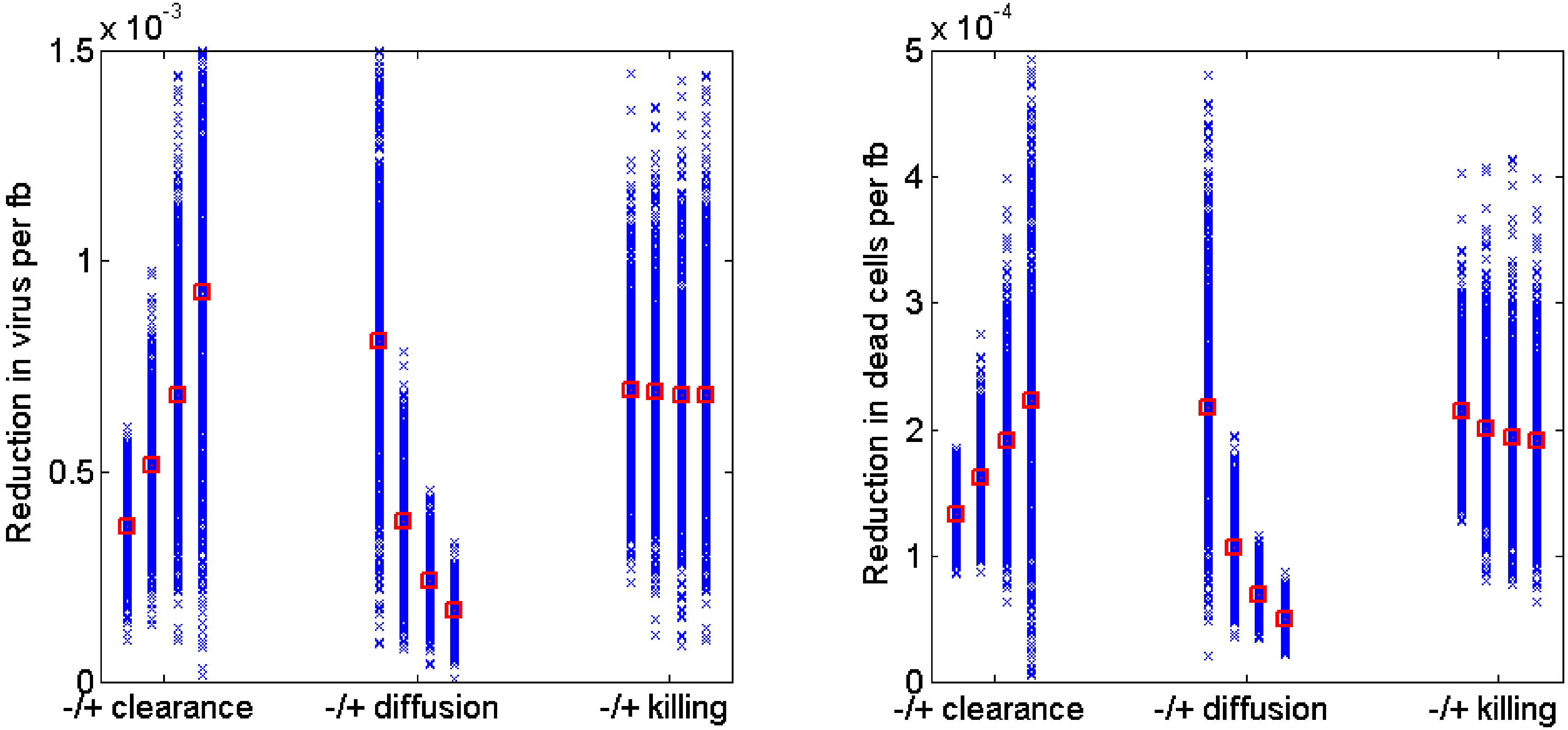
- The players
  - Virus, Target (epithelial) cells, CTL
- The place
  - A square grid filled with (fixed) target cells, representing a patch of epithelial tissue
- The action
  - Infection starts at middle of grid
  - Virions diffuse around; are cleared at a fixed rate; bind to and enter cells and thereby infect uninfected cells
  - Uninfected cells become infected, produce virus, die
  - CTL slowly enter the “scene”, move around, can find and kill infected cells
  - Infected cells produce bystander cells, which can be killed by CTL

Spatial results



Shown are averages of 200 simulations.
The 3 different scenarios have different values for parameters such as diffusion speed of virions and CTL, probability of virus death and killing of infected/bystander cells by CTL per time-step, etc.

Spatial results



GMAT/ the firebreak has a greater effect if virus clearance is fast and diffusion slow. Speed of CTL killing has little effect.

A firebreak (FB) is a bystander cell killed by CTL.

Spatial results

- The spatial model suggests that GMAT might be a beneficial mechanism for the host immune response to reduce both virus load and immunopathology.
- The non-spatial differential equation model could not capture this result. → It is important to choose the right modeling framework for the question at hand.

Discussion of ABM

- ABM are the most detailed (and potentially realistic) we have seen so far, they allow one to address questions that can't be addressed with simple compartmental models.
- Model needs to be carefully tailored to the question and data.
- More complexity means less general/conceptual insights, more reliance on simulations. Almost no math/analytics is possible.
- Many parameters, usually more than in ODE models. Becomes a problem if these parameters are unknown or poorly known.
- Potentially more accurate.
- Speed can become a serious issue.
- The models are often stochastic, but don't have to be.
- Data fitting becomes hard.

Further reading – ABM

- Durrett (1999) SIAM Reviews (mathematical review of spatial model approaches)
- Grimm and Railsback (2005) “Individual-based Modeling and Ecology” Princeton U Press (focus on ecology, but approaches can also apply to within-host modeling)
- Railsback et al. (2006) Simulation (reviews several common software packages)
- Chavali et al. (2008) Trends in Immunology (review of ABM in immunology)
- Keeling and Rohani (2008) “Modeling Infectious Diseases” Princeton U. Press (mostly between-host, only a bit of spatial, but good advice)
- Bauer et al. (2009) Information Science “Agent-based modeling of host–pathogen systems: The successes and challenges”

On your own – Exploring NetLogo

- Start Netlogo
- In “Files -> Models Library” under “Biology” open the model called “Virus”
- Press “Setup”, then “Go”
- You might have to adjust the simulation speed with the speed slider
- Change around the other sliders and see how that affects the dynamics
- Pressing “Go” again stops simulation, then you can restart with “Setup/Go”
- Go to the “Information” tab to learn more about the simulation
- Go to the “Procedures” tab to see the actual code
- Explore the many more interesting models in the models library.

Summary

- Mechanistic, dynamical models are well suited for studying the dynamics of infectious diseases.
- Different types of models exist, the model choice should be driven by the question/system.
- Models always make simplifying assumptions. The applicability of the models and the conclusions drawn are only valid if the approximations made for the model are fulfilled.
- This is never the case, but often the error we make in translating (and thereby simplifying) a complex biological process to a mathematical model is small enough to make even simple models useful.
- Models are always wrong but sometimes surprisingly useful. (Also applies to all experimental model systems).

Summary

- There are many approaches to building a model
- For a given approach, there are many variants of implementing specific mechanisms (recall HIV models)
- Ideal approach:
  - Choose the model that is most suitable for the question you try to answer.
  - If you can, maybe try a few model variants.
- In reality: Model selection is based on a mix of
  - Question one wants to answer
  - Expertise
  - Feasibility (CPU time, model complexity)
  - Time to graduation/end of grant/tenure review
  - “Environment” (what approaches do others use)
  - “Marketing” (what kind of models “sell”)