

# Lesson 1: Introduction to Simulation-based Inference for Epidemiological Dynamics

Aaron A. King, Edward L. Ionides and Qianying Lin

# Outline

- 1 Introduction
  - What makes epidemiological inference hard?
  - Course overview
- 2 Partially observed Markov processes
  - Mathematical definitions
  - From math to algorithms
- 3 The **pomp** package

# Objectives for this lesson

- To understand the motivations for simulation-based inference in the study of epidemiological and ecological systems.
- To introduce the class of partially observed Markov process (POMP) models.
- To introduce the **pomp** R package.

# Epidemiological and Ecological Dynamics

- Ecological systems are complex, open, nonlinear, and **nonstationary**.
- “Laws of Nature” are unavailable except in the most general form.
- It is useful to model them as **stochastic systems**.
- For any observable phenomenon, **multiple** competing explanations are possible.
- Central scientific goals:
  - **Which explanations are most favored** by the data?
  - Which kinds of data are most informative?
- Central applied goals:
  - How to design ecological or epidemiological intervention?
  - How to make accurate forecasts?
- Time series are particularly useful sources of data.

# Obstacles to inference


Obstacles for **ecological** modeling and inference via nonlinear mechanistic models enumerated by Bjørnstad and Grenfell (2001)

- 1 Combining measurement noise and process noise.
- 2 Including covariates in mechanistically plausible ways.
- 3 Using continuous-time models.
- 4 Modeling and estimating interactions in coupled systems.
- 5 Dealing with unobserved variables.
- 6 Modeling spatial-temporal dynamics.

The same issues arise for **epidemiological** modeling and inference via nonlinear mechanistic models.

The *partially observed Markov process* modeling framework we focus on in this course addresses most of these problems effectively.

# Course objectives

- ① To show how stochastic dynamical systems models can be used as scientific instruments.
- ② To teach statistically and computationally efficient approaches for performing scientific inference using POMP models.
- ③ To give students the ability to formulate models of their own.
- ④ To give students opportunities to work with such inference methods.
- ⑤ To familiarize students with the **pomp** package. 
- ⑥ To provide documented examples for adaptation and re-use.

# Questions and answers

- ① How to explain the resurgence of pertussis in countries with sustained high vaccine coverage?
- ② What roles are played by asymptomatic infection and waning immunity in cholera epidemics?
- ③ What explains the seasonality of measles?
- ④ Can serotype-specific immunity explain the strain dynamics of human enteroviruses?
- ⑤ Do subclinical infections of pertussis play an important epidemiological role?

# Questions and answers II

- ⑥ What is the contribution to the HIV epidemic of dynamic variation in sexual behavior of an individual over time? How does this compare to the role of heterogeneity between individuals?
- ⑦ What explains the interannual variability of malaria?
- ⑧ What will happen next in an Ebola outbreak?
- ⑨ Can hydrology explain the seasonality of cholera?
- ⑩ What is the contribution of adults to polio transmission?



# Partially observed Markov process (POMP) models

- Data  $y_1^*, \dots, y_N^*$  collected at times  $t_1 < \dots < t_N$  are modeled as noisy, incomplete, and indirect observations of a Markov process  $\{X(t), t \geq t_0\}$ .
- This is a **partially observed Markov process (POMP)** model, also known as a hidden Markov model or a state space model.
- $\{X(t)\}$  is Markov if the history of the process,  $\{X(s), s \leq t\}$ , is **uninformative** about the future of the process,  $\{X(s), s \geq t\}$ , given the current value of the process,  $X(t)$ .
- If all quantities important for the dynamics of the system are placed in the **state**,  $X(t)$ , then the Markov property holds by construction.

# Partially observed Markov process (POMP) models II

- Systems with delays can usually be rewritten as Markovian systems, at least approximately.
- An important special case: any system of differential equations  $dx/dt = f(x)$  is Markovian.
- POMP models can include all the features desired by Bjørnstad and Grenfell (2001).

# Schematic of the structure of a POMP

- Arrows in the following diagram show causal relations.
- A key perspective to keep in mind is that **the model is to be viewed as the process that generated the data.**
- That is: the data are viewed as one realization of the model's stochastic process.



# Notation for POMP models

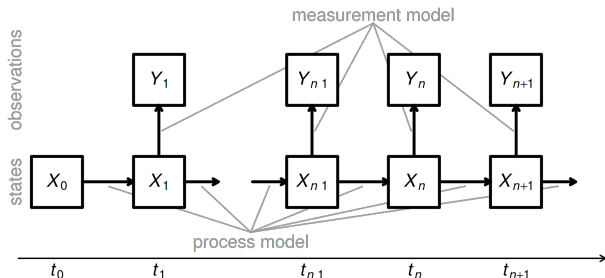
- Write  $X_n = X(t_n)$  and  $X_{0:N} = (X_0, \dots, X_N)$ . Let  $Y_n$  be a random variable modeling the observation at time  $t_n$ .
- The one-step transition density,  $f_{X_n|X_{n-1}}(x_n|x_{n-1}; \theta)$ , together with the measurement density,  $f_{Y_n|X_n}(y_n|x_n; \theta)$  and the initial density,  $f_{X_0}(x_0; \theta)$ , specify the entire POMP model.
- The joint density  $f_{X_{0:N}, Y_{1:N}}(x_{0:N}, y_{1:N}; \theta)$  can be written as

$$f_{X_0}(x_0; \theta) \prod_{n=1}^N f_{X_n|X_{n-1}}(x_n|x_{n-1}; \theta) f_{Y_n|X_n}(y_n|x_n; \theta)$$

- The marginal density for  $Y_{1:N}$  evaluated at the data,  $y_{1:N}^*$ , is

$$f_{Y_{1:N}}(y_{1:N}^*; \theta) = \int f_{X_{0:N}, Y_{1:N}}(x_{0:N}, y_{1:N}^*; \theta) dx_{0:N}$$

# Another POMP model schematic



- The state process,  $X_n$ , is Markovian, i.e.,

$$f_{X_n|X_{0:n-1}, Y_{1:n-1}}(x_n|x_{0:n-1}, y_{1:n-1}) = f_{X_n|X_{n-1}}(x_n|x_{n-1}).$$

- Moreover,  $Y_n$ , depends only on the state at that time:

$$f_{Y_n|X_{0:N}, Y_{1:n-1}}(y_n|x_{0:n}, y_{1:n-1}) = f_{Y_n|X_n}(y_n|x_n), \quad \text{for } n = 1, \dots, N.$$

# Moving from math to algorithms for POMP models


We specify some **basic model components** which can be used within algorithms:

- 'rprocess': a draw from  $f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$
- 'dprocess': evaluation of  $f_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$
- 'rmeasure': a draw from  $f_{Y_n|X_n}(y_n|x_n;\theta)$
- 'dmeasure': evaluation of  $f_{Y_n|X_n}(y_n|x_n;\theta)$
- 'rinit': a draw from  $f_{X_0}(x_0;\theta)$




These basic model components define the specific POMP model under consideration.

# What is a simulation-based method?

- **Simulating** random processes **is often much easier** than evaluating their transition probabilities. 
- In other words, we may be able to write `rprocess` but not `dprocess`.
- **Simulation-based** methods require the user to specify `rprocess` but not `dprocess`.
- **Plug-and-play**, **likelihood-free** and **equation-free** are alternative terms for “simulation-based” methods.
- Much development of simulation-based statistical methodology has occurred in the past decade.

# The **pomp** package for POMP models

- **pomp** is an R package for data analysis using partially observed Markov process (POMP) models (King *et al.*, 2016).
- Note the distinction: lower case **pomp** is a software package; upper case POMP is a class of models. 
- **pomp** builds methodology for POMP models in terms of arbitrary user-specified POMP models.
- **pomp** provides tools, documentation, and examples to help users **specify** POMP models.
- **pomp** provides a platform for modification and sharing of models, data-analysis workflows, and methodological development.




# Structure of the **pomp** package

It is useful to divide the **pomp** package functionality into different levels:

- Basic model components
- Workhorses
- Elementary POMP algorithms
- Inference algorithms

# Basic model components

Basic model components are user-specified procedures that perform the elementary computations that specify a POMP model. There are nine of these:

- 'rinit': simulator for the **initial-state distribution**, i.e., the distribution of the latent state at time  $t_0$ .
- 'rprocess' and 'dprocess': simulator and density evaluation procedure, respectively, for the process model.
- 'rmeasure' and 'dmeasure': simulator and density evaluation procedure, respectively, for the measurement model.
- 'rprior' and 'dprior': simulator and density evaluation procedure, respectively, for the prior distribution.
- 'skeleton': evaluation of a deterministic skeleton. 
- 'partrans': parameter transformations.

The scientist must specify whichever of these basic model components are required for the algorithms that the scientist uses.

# Workhorses

Workhorses are R functions, built into the package, that cause the basic model component procedures to be executed.

- Each basic model component has a corresponding workhorse.
- Effectively, the workhorse is a vectorized wrapper around the basic model component.
- For example, the `rprocess()` function uses code specified by the `rprocess` model component, constructed via the `rprocess` argument to `pomp()`.
- The `rprocess` model component specifies how a single trajectory evolves at a single moment of time. The `rprocess()` workhorse combines these computations for arbitrary collections of times and arbitrary numbers of replications.

# Elementary POMP algorithms

These are algorithms that interrogate the model or the model/data confrontation without attempting to estimate parameters. There are currently four of these:

- `simulate` performs simulations of the POMP model, i.e., it samples from the joint distribution of latent states and observables.
- `pfilter` runs a sequential Monte Carlo (particle filter) algorithm to compute the likelihood and (optionally) estimate the prediction and filtering distributions of the latent state process.
- `probe` computes one or more uni or multivariate summary statistics on both actual and simulated data.
- `spect` estimates the power spectral density functions for the actual and simulated data.

# POMP inference algorithms

These are procedures that build on the elementary algorithms and are used for estimation of parameters and other inferential tasks. There are currently ten of these:


- `abc`: approximate Bayesian computation
- `bsmc2`: Liu-West algorithm for Bayesian SMC
- `pmcmc`: a particle MCMC algorithm
- `mif2`: iterated filtering (IF2)
- `enkf`, `eakf` ensemble and ensemble adjusted Kalman filters
- `traj_objfun`: trajectory matching
- `spect_objfun`: power spectrum matching
- `probe_objfun`: probe matching
- `nlf_objfun`: nonlinear forecasting

*Objective function methods*: among the estimation algorithms just listed, four are methods that construct stateful objective functions that can be optimized using general-purpose numerical optimization algorithms such as `optim`, `subplex`, or the optimizers in the **`nloptr`** package.

# References

- Bjørnstad ON, Grenfell BT (2001). “Noisy clockwork: Time series analysis of population fluctuations in animals.” *Science*, **293**, 638–643. doi: [10.1126/science.1062226](https://doi.org/10.1126/science.1062226).
- King AA, Nguyen D, Ionides EL (2016). “Statistical Inference for Partially Observed Markov Processes via the R Package `pomp`.” *Journal of Statistical Software*, **69**(12), 1–43. doi: [10.18637/jss.v069.i12](https://doi.org/10.18637/jss.v069.i12).

# License, acknowledgments, and links

- This lesson is prepared for the [Simulation-based Inference for Epidemiological Dynamics](#) module at the 2020 Summer Institute in Statistics and Modeling in Infectious Diseases, [SISMID 2020](#).
- The materials build on [previous versions of this course and related courses](#).
- Licensed under the [Creative Commons Attribution-NonCommercial license](#). Please share and remix non-commercially, mentioning its origin. 
- Produced with R version 4.1.1 and **pomp** version 4.0.11.0.
- Compiled on December 4, 2021.

[Back to course homepage](#)  
[pomp homepage](#)