

# How to choose the right inference approach for your problem

Sebastian Funk (@sbfnk)

Centre for the Mathematical Modelling of Infectious Diseases  
London School of Hygiene & Tropical Medicine



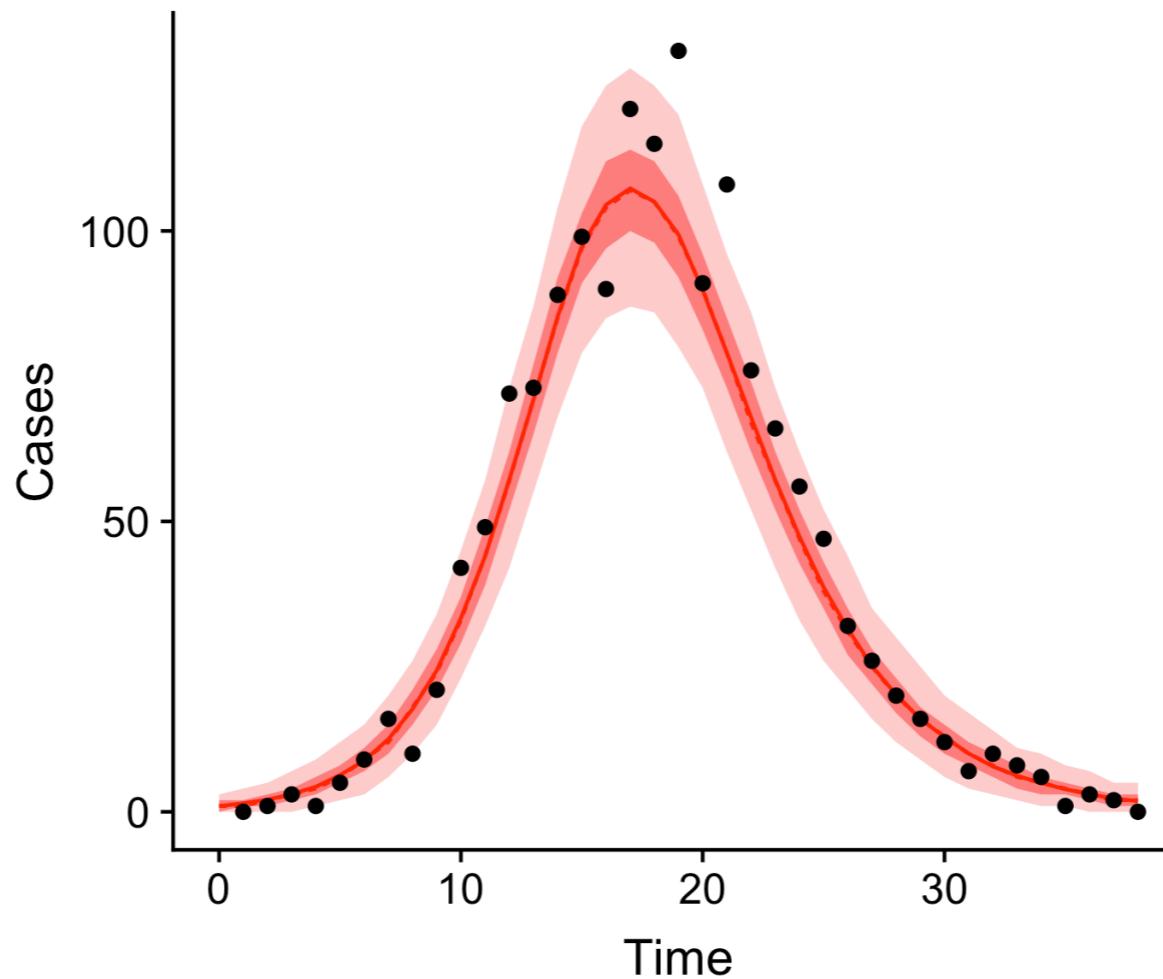
**centre for the  
mathematical  
modelling of  
infectious diseases**

LONDON  
SCHOOL *of*  
HYGIENE  
& TROPICAL  
MEDICINE



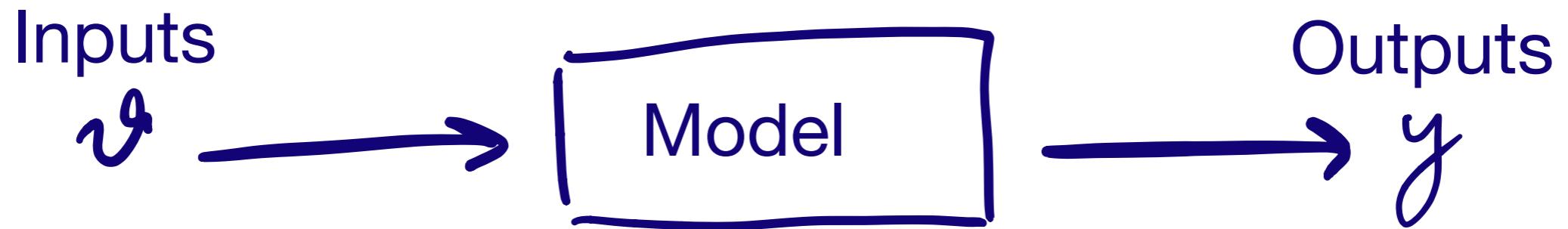


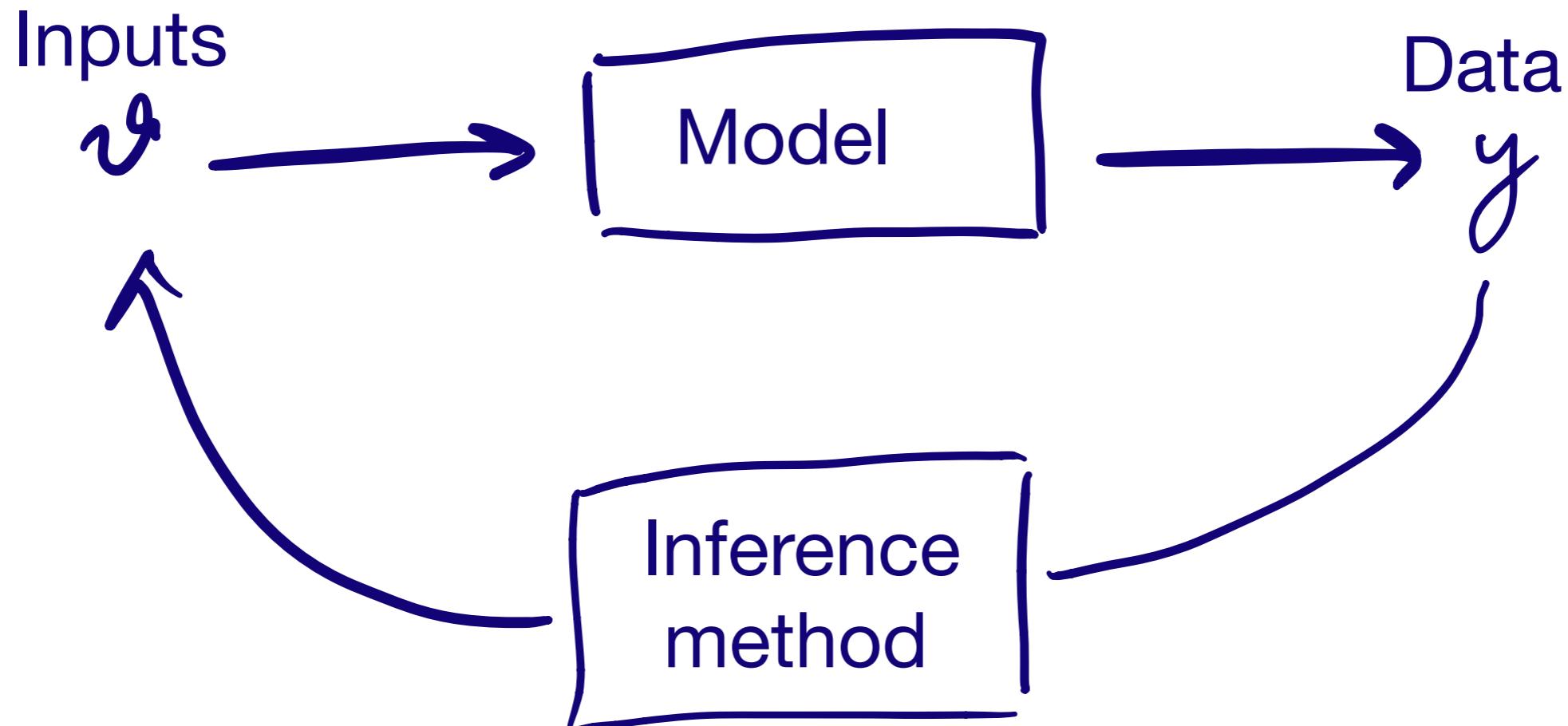
# Simulation-based inference



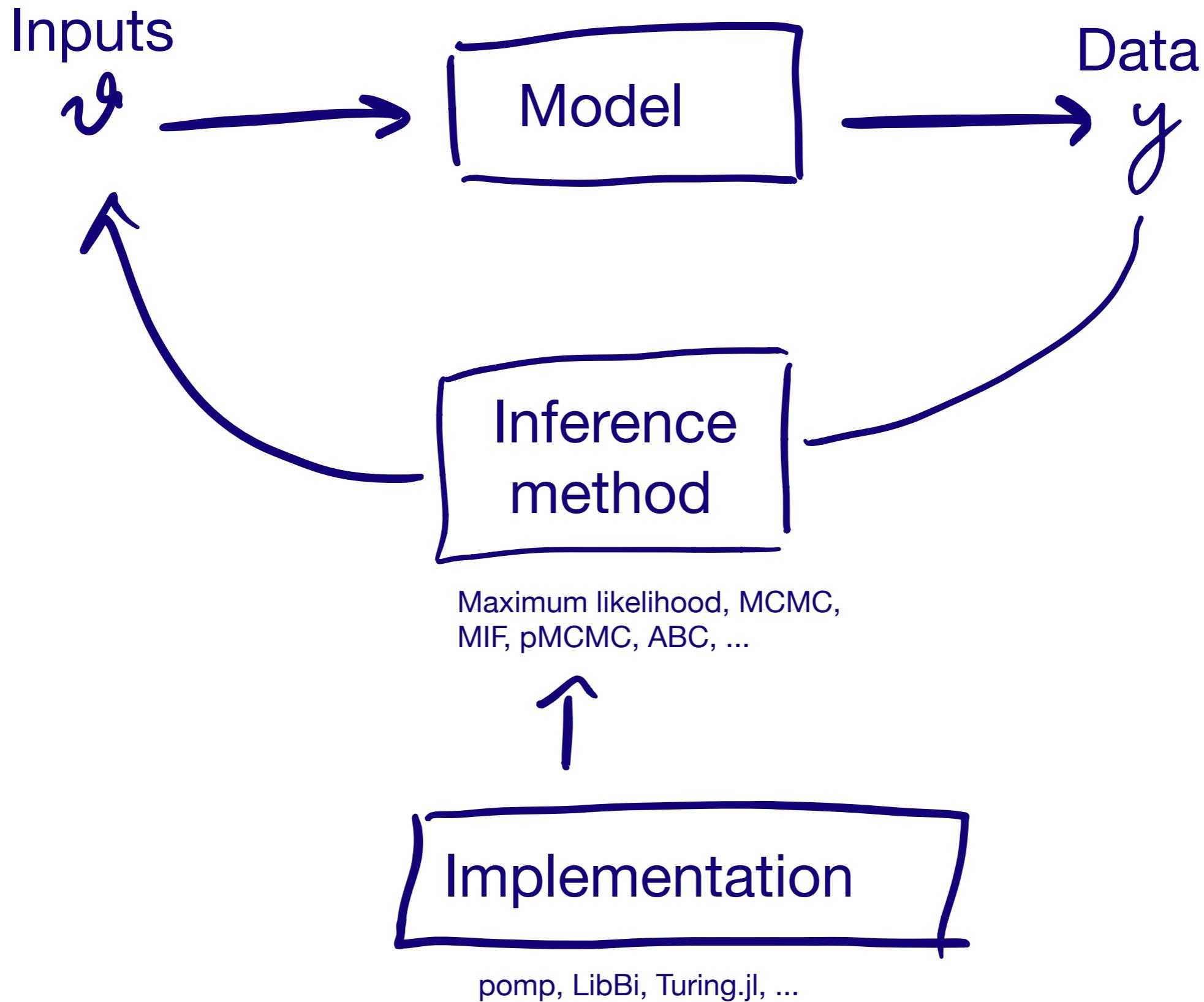
Combine mechanistic and statistical approach for

- Parameter estimation
- Prediction

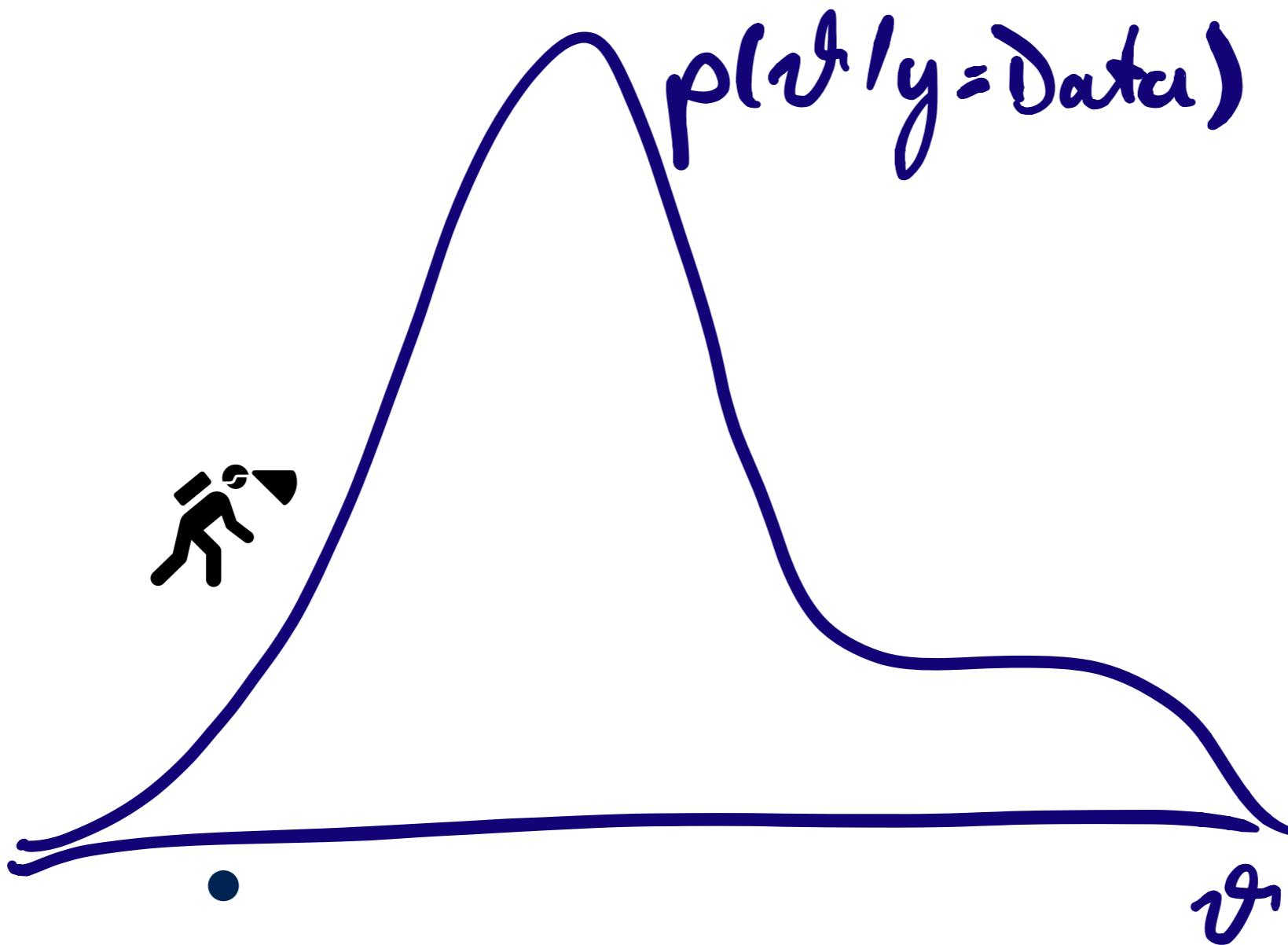




Maximum likelihood, MCMC,  
MIF, pMCMC, ABC, ...

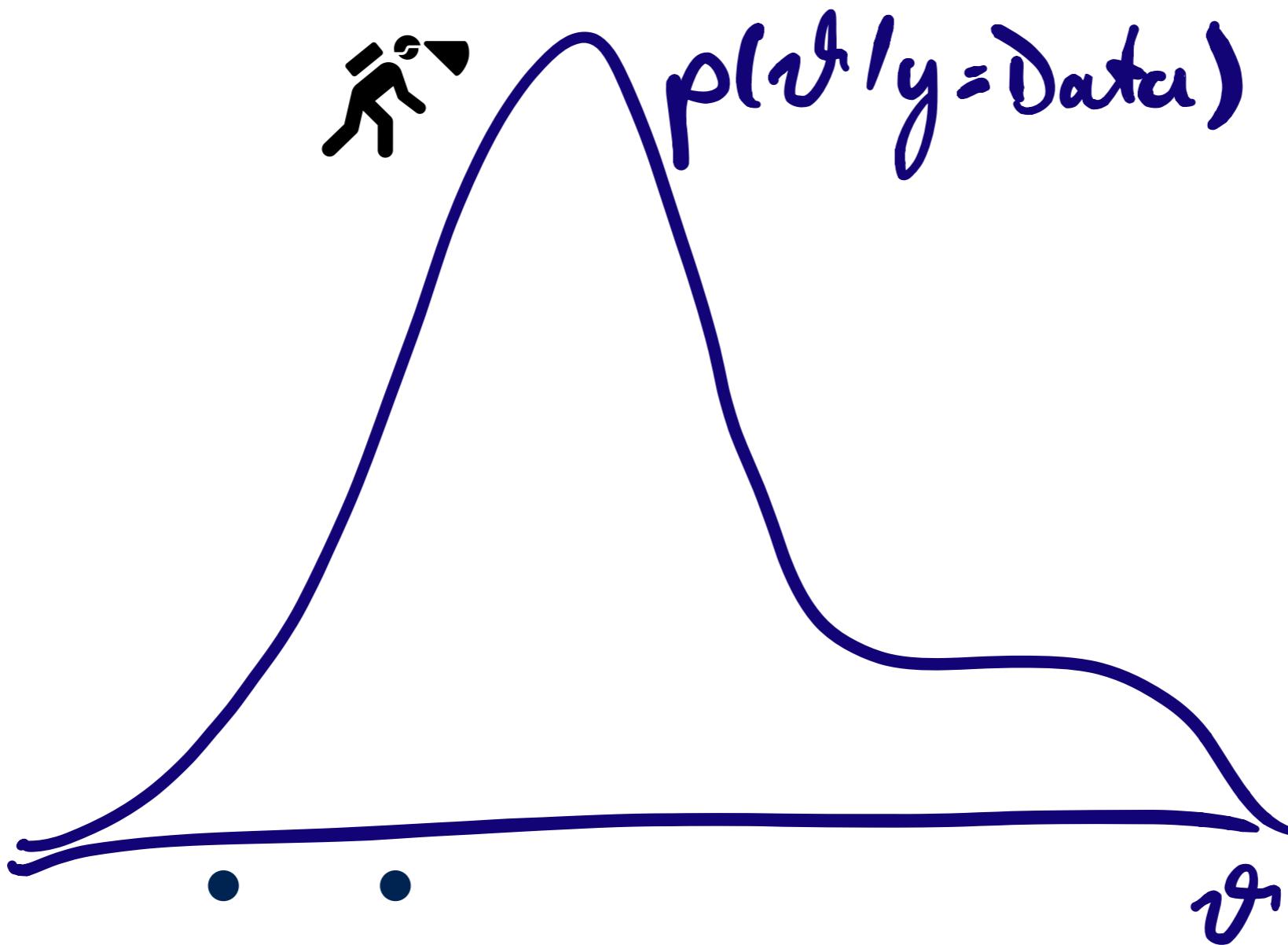


# Markov-chain Monte Carlo



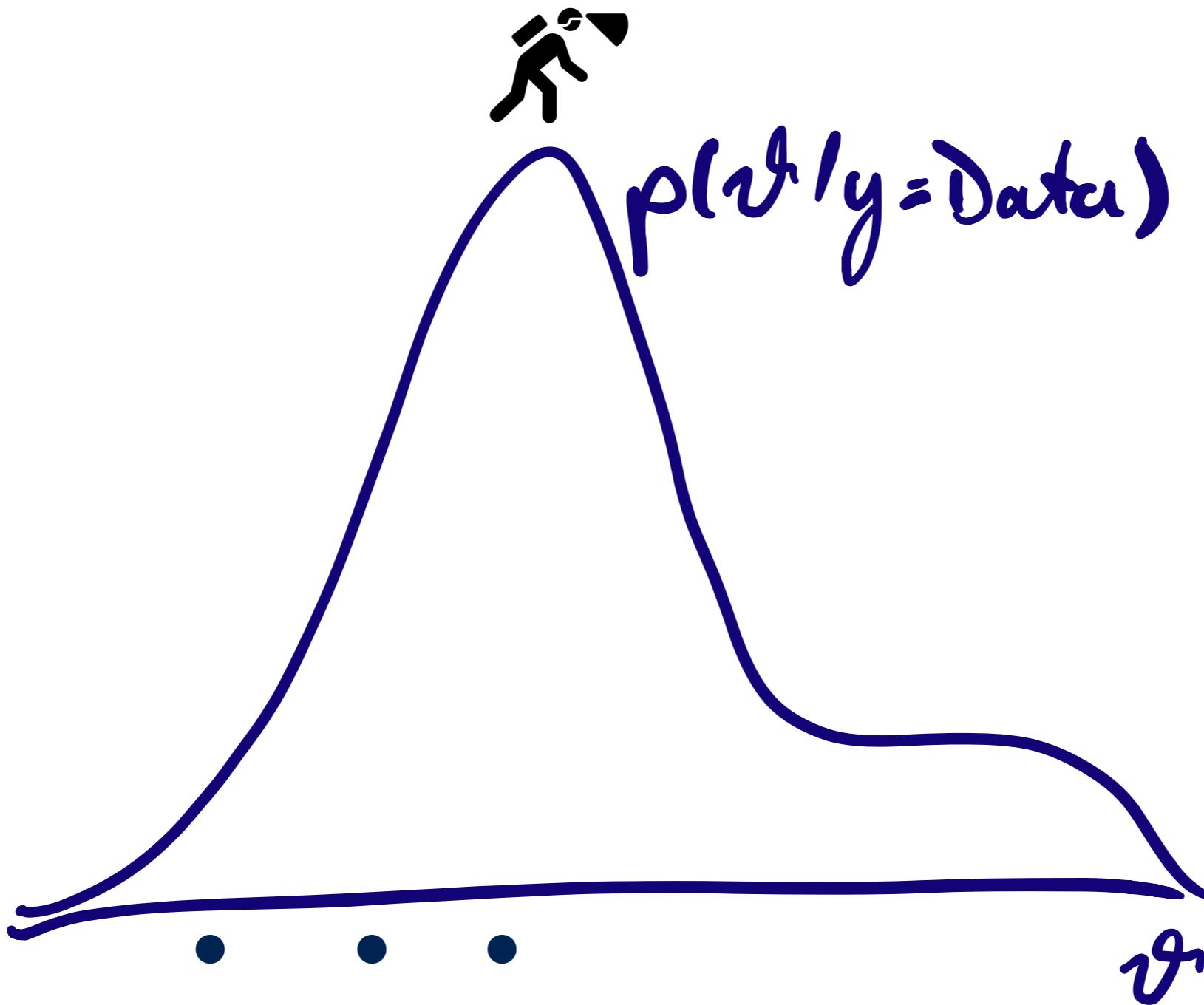
Sample from  $p(\theta | \text{Data})$ .

# Markov-chain Monte Carlo



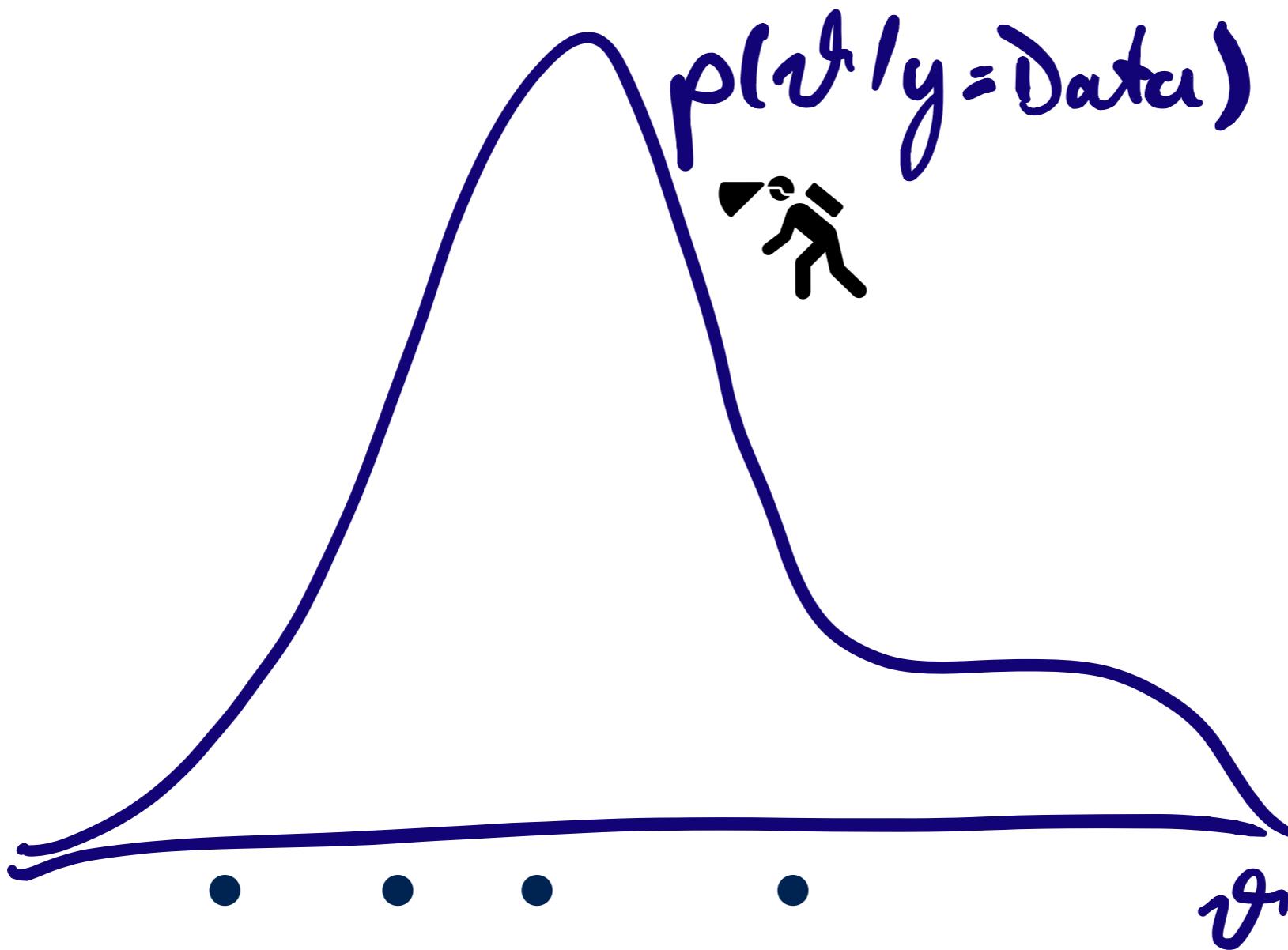
Sample from  $p(\theta | \text{Data})$ .

# Markov-chain Monte Carlo



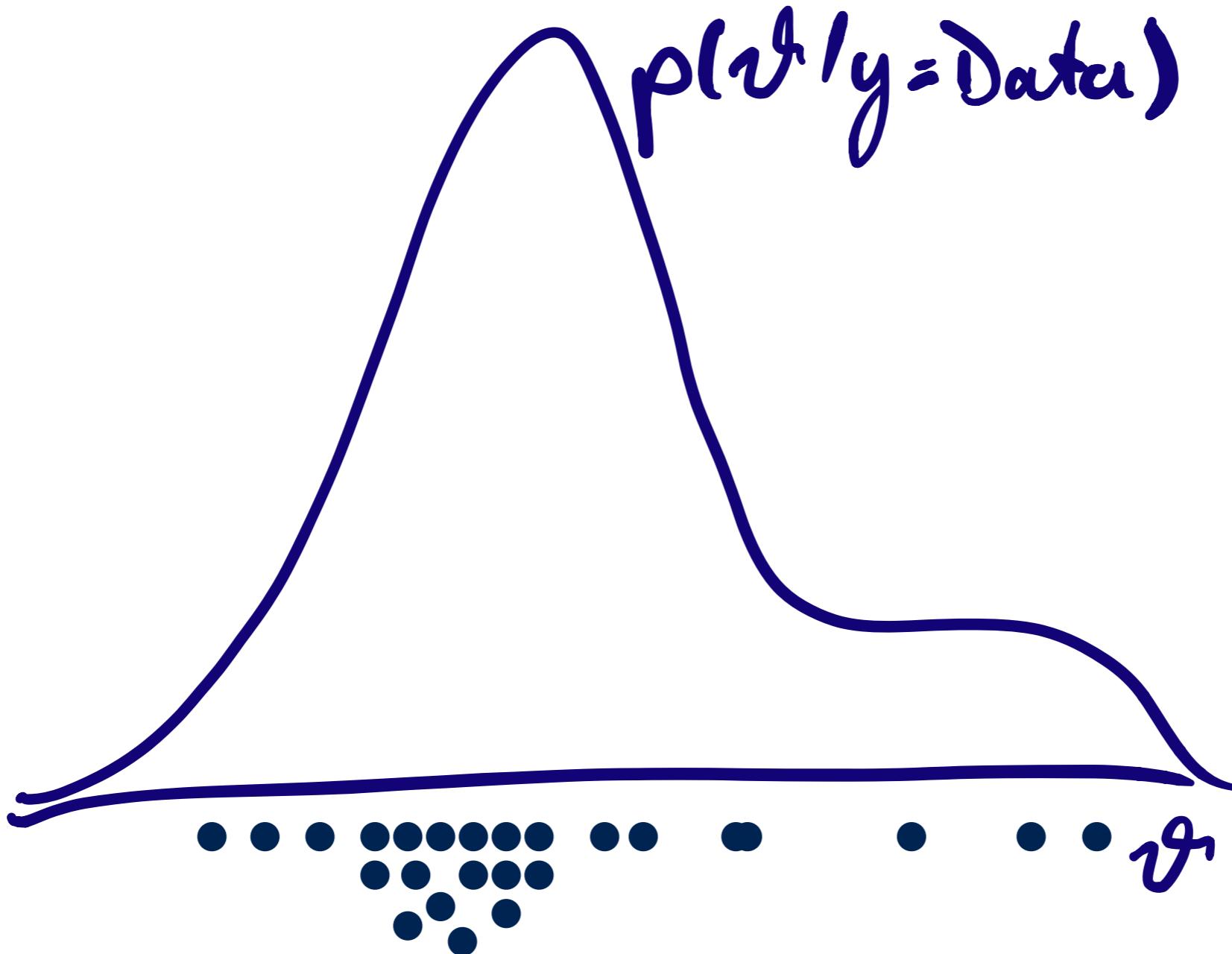
Sample from  $p(\theta | \text{Data})$ .

# Markov-chain Monte Carlo



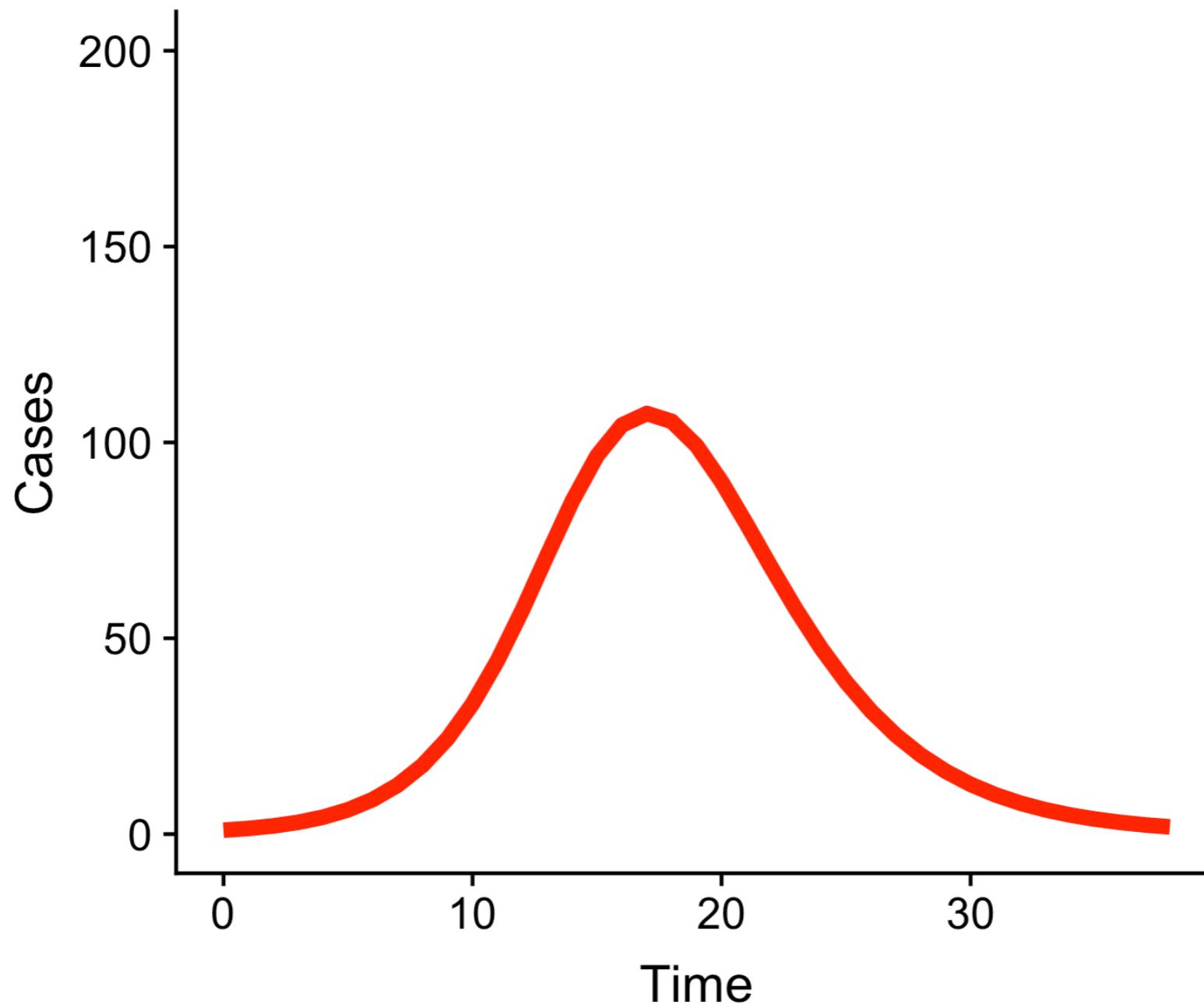
Sample from  $p(\theta | \text{Data})$ .

# Markov-chain Monte Carlo

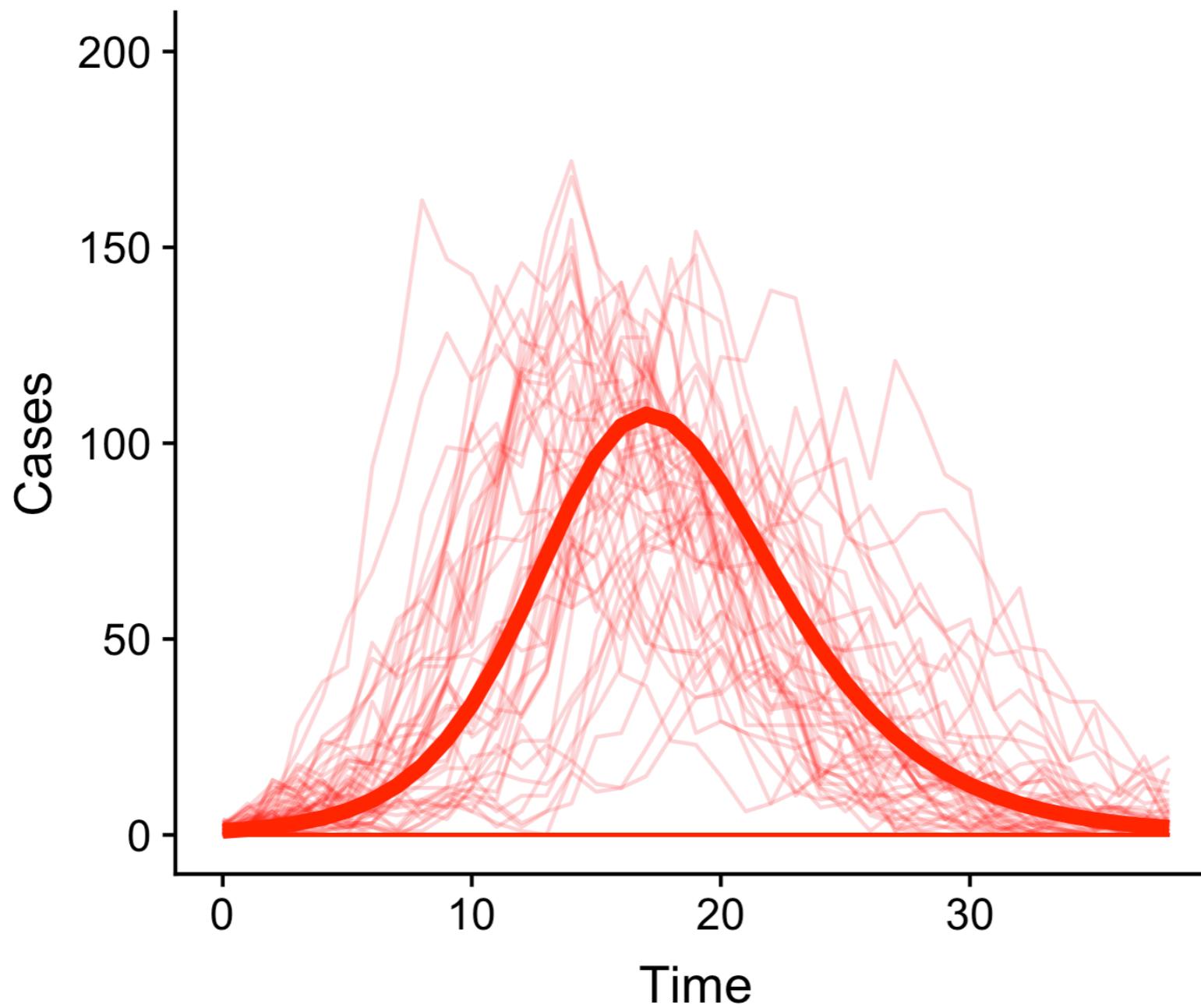


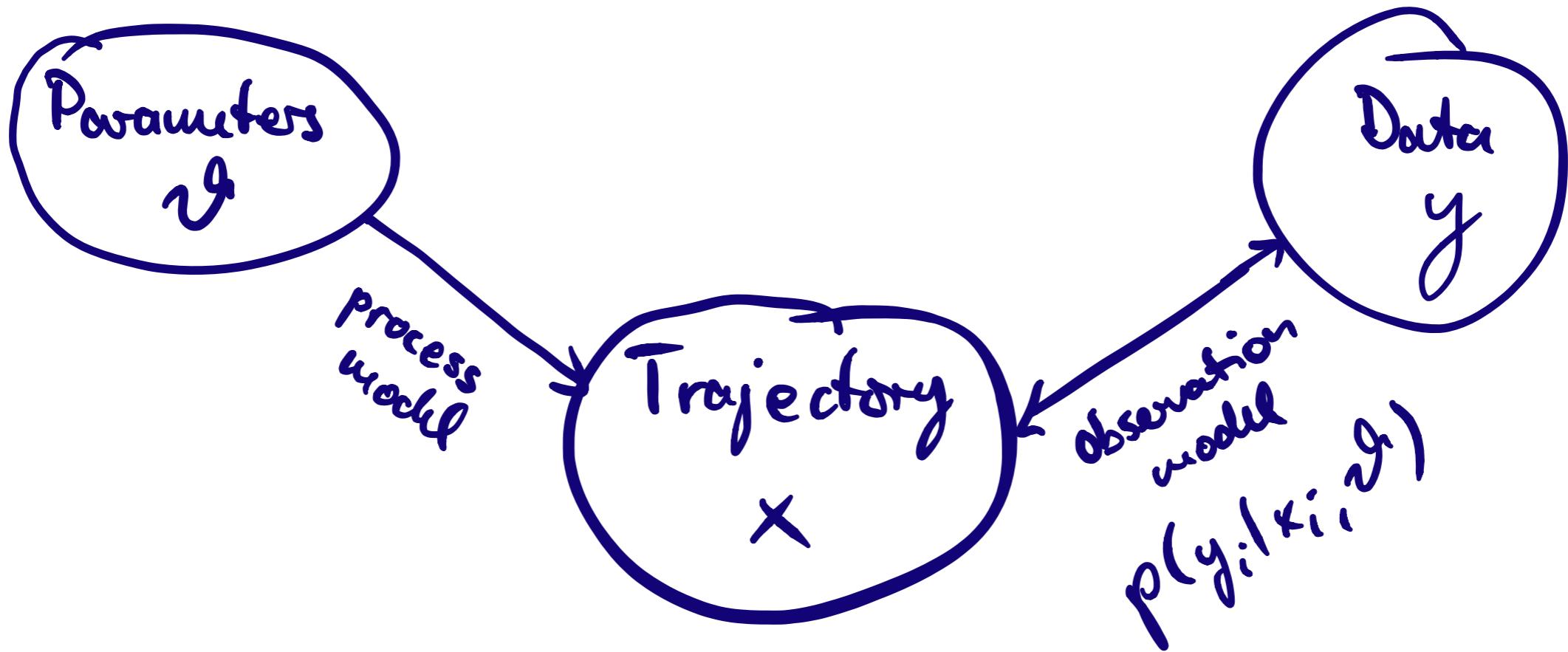
Sample from  $p(\theta|\text{Data})$ .

# Fitting deterministic vs. stochastic models

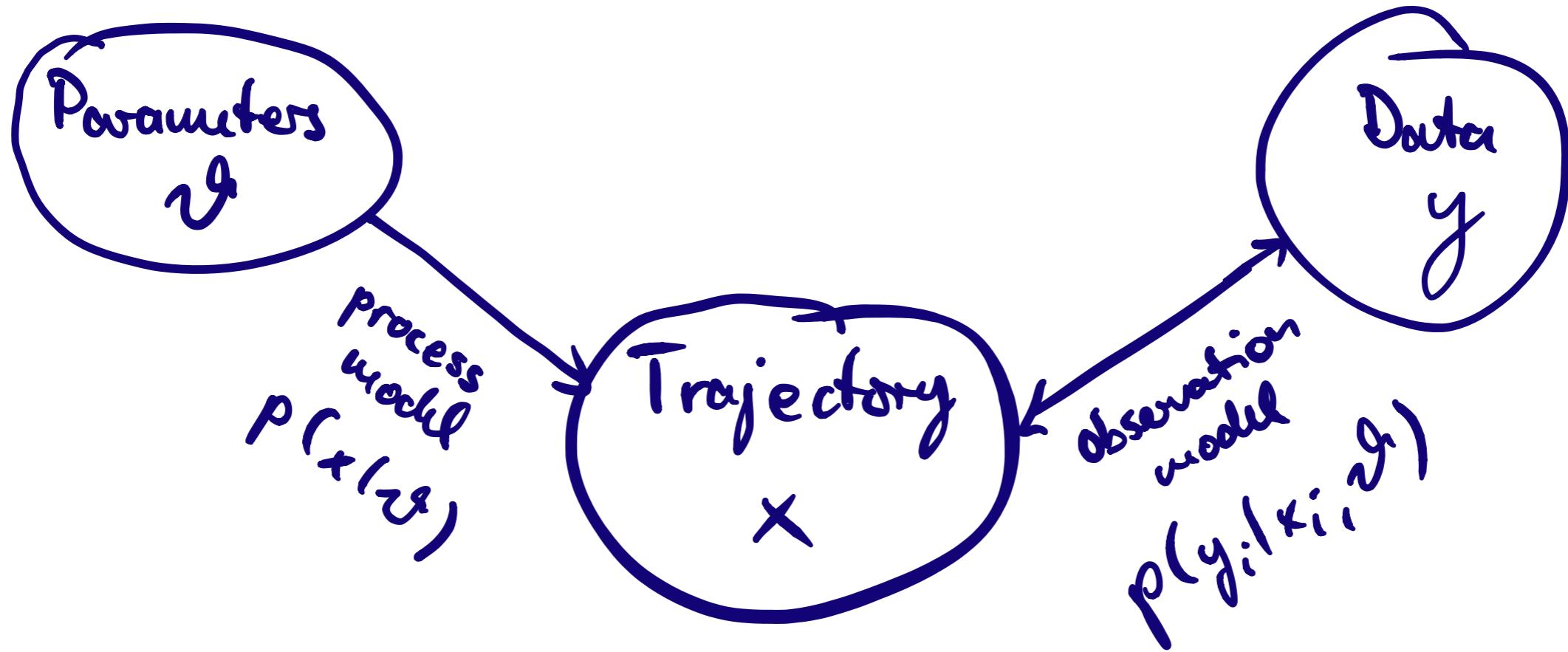


# Fitting deterministic vs. stochastic models





Can compute  $p(y|\theta)$  for each  $\theta$ .



Cannot compute  $p(y|\theta)$  for given  $\theta$ .

## Marginal likelihood

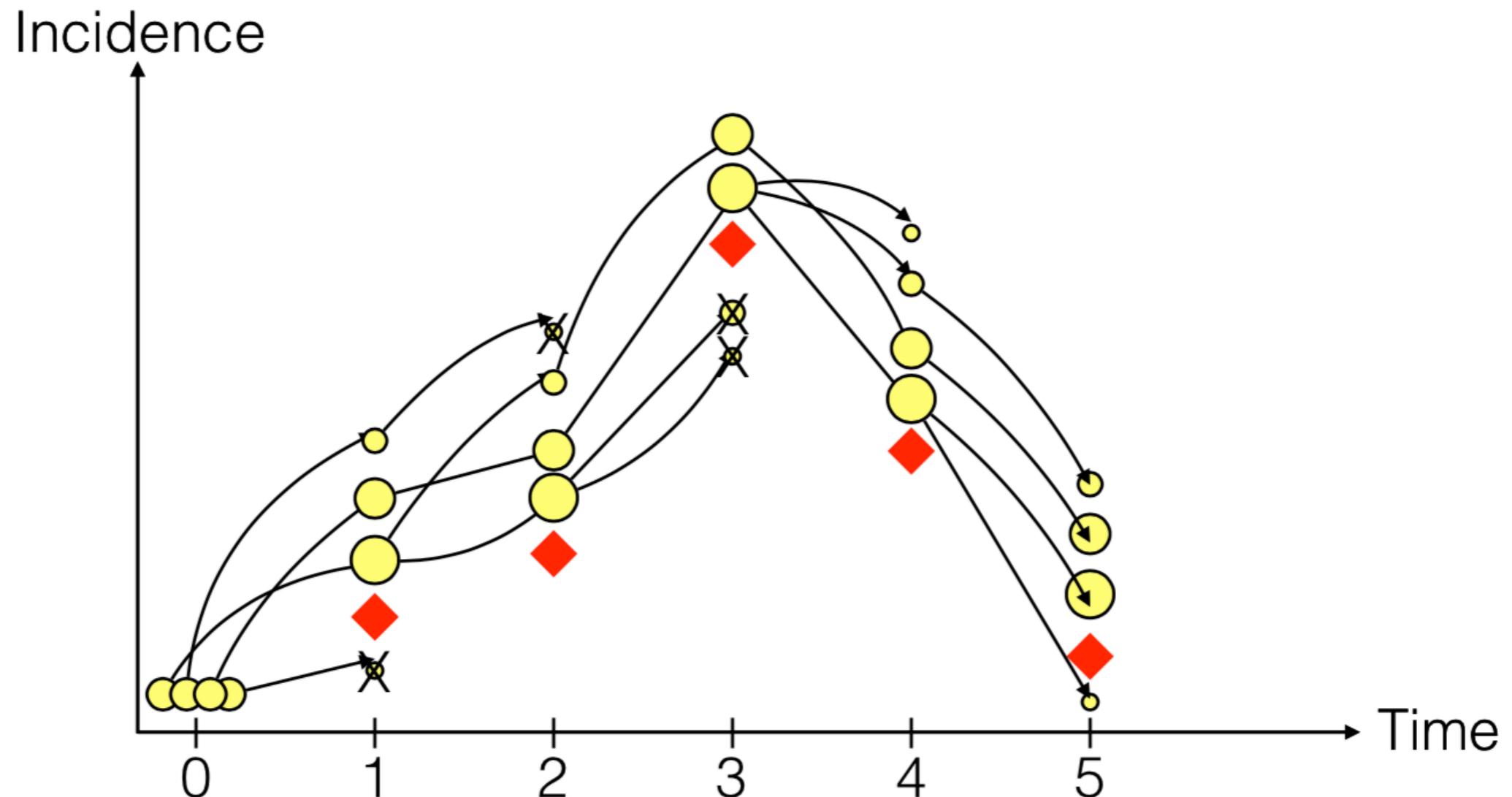
$$p(y|\theta) = \sum p(y|x, \theta)p(x|\theta)$$

$x$



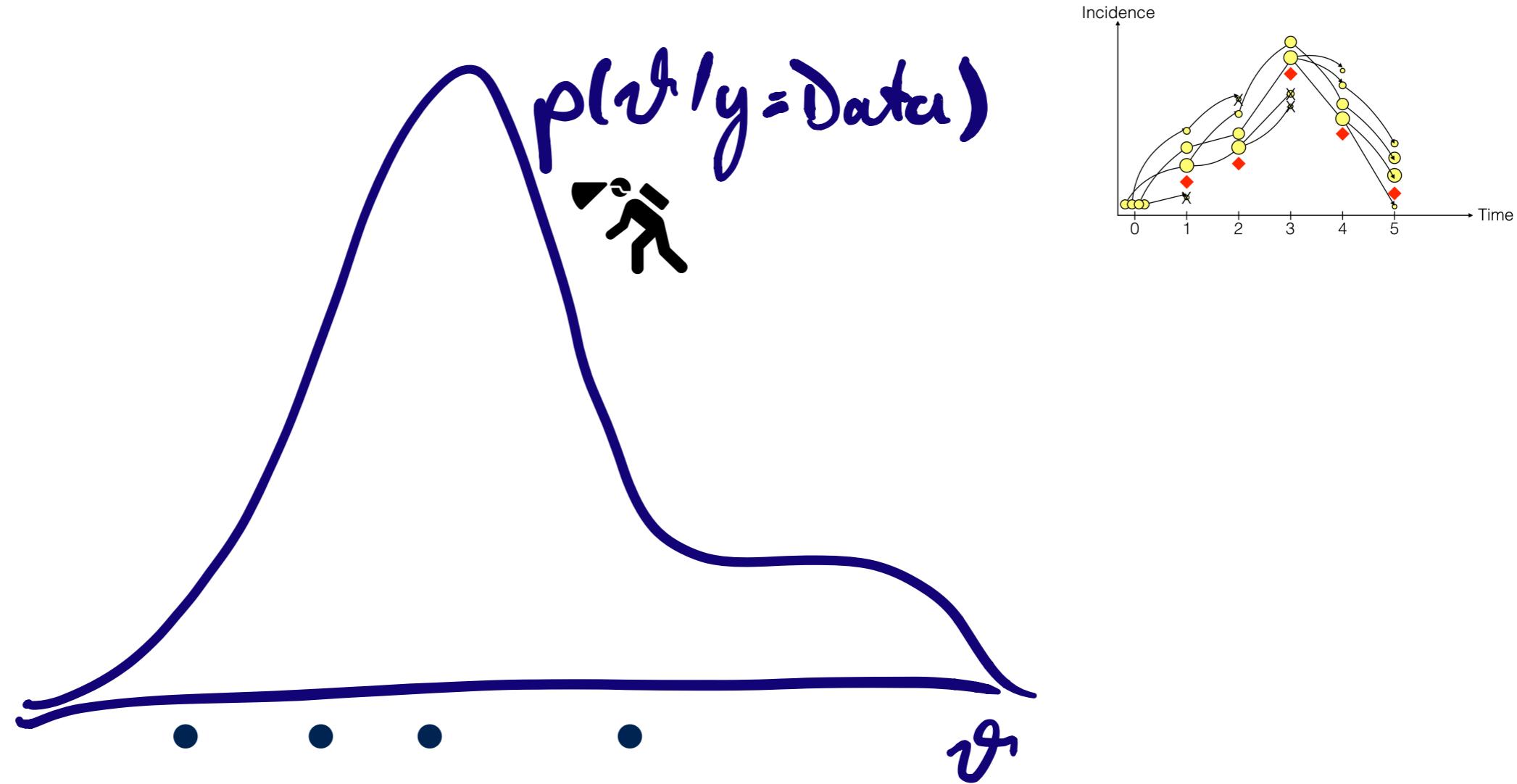
All possible trajectories, given  $\theta$

# The particle filter



Estimate  $p(\text{Data}|\theta)$  for given  $\theta$ .

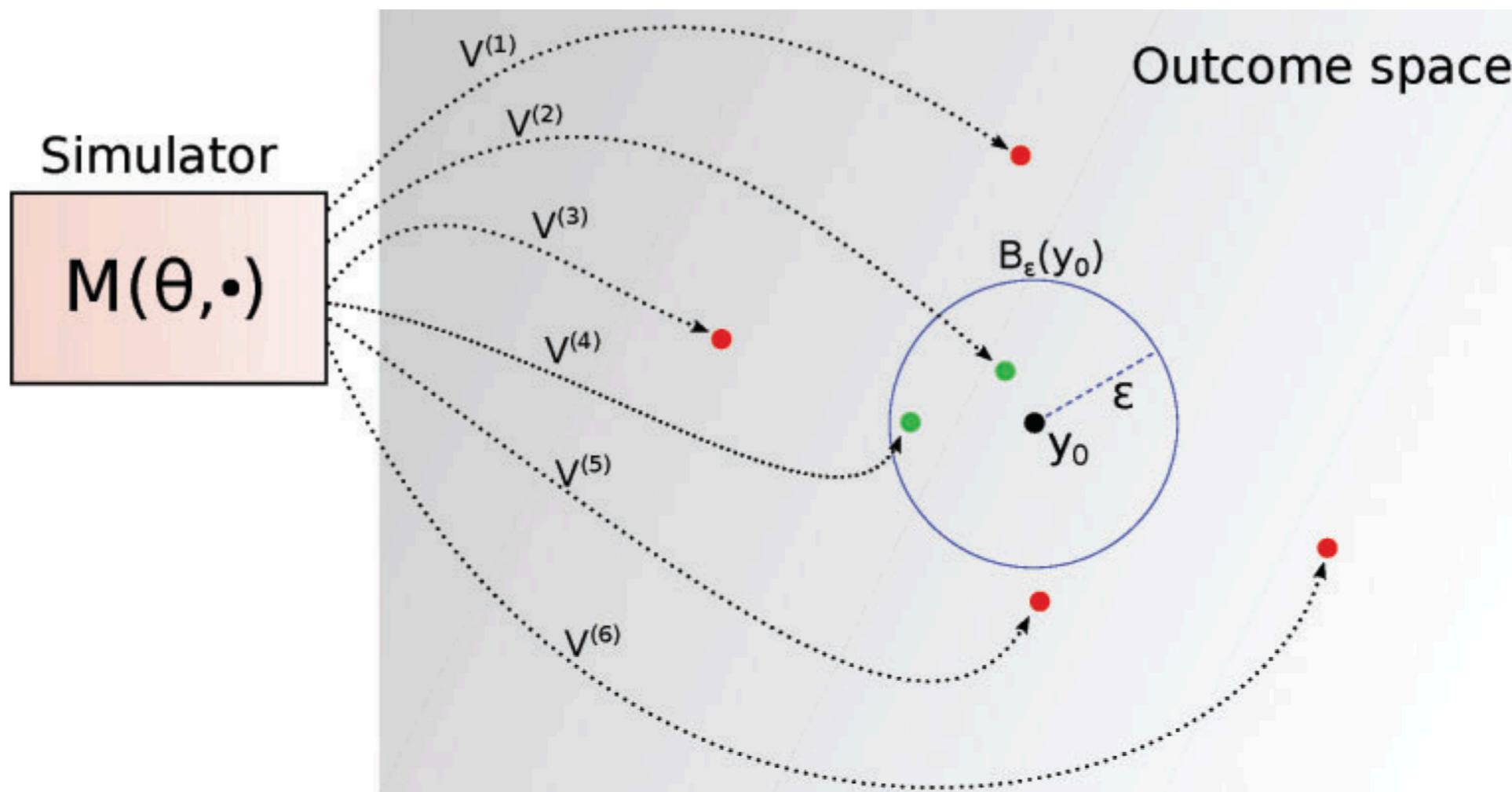
# Particle Markov-chain Monte Carlo (pMCMC)



MCMC with  $p(y|\theta)$  replaced with particle filter estimate

# Approximate Bayesian Computation (ABC)

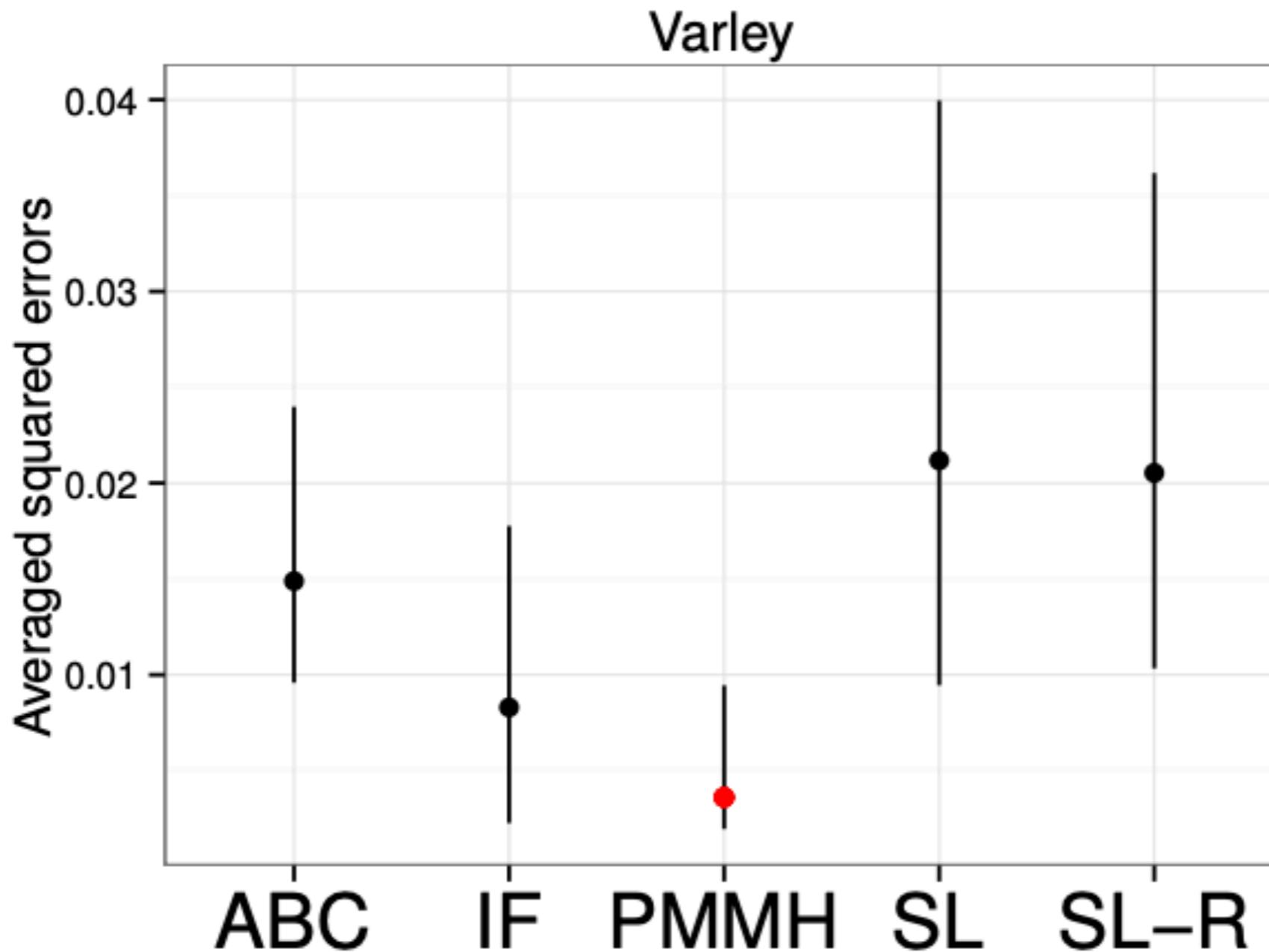
Lintusaari et al. (2016)



instead of calculating likelihood, compares summary statistics calculated on simulations with summary statistics calculated on data

		<b>Frequentist</b>	<b>Bayesian</b>
Plug-and-play	<b>Full-information</b>	iterated filtering	particle MCMC
		simulated moments	ABC
	<b>Feature-based</b>	synthetic likelihood (SL)	SL-based MCMC
		nonlinear forecasting	
Not plug-and-play	<b>Full-information</b>	EM algorithm	MCMC
		Kalman filter	
	<b>Feature-based</b>	Yule-Walker <sup>1</sup>	extended Kalman filter <sup>2</sup>
		extended Kalman filter <sup>2</sup>	

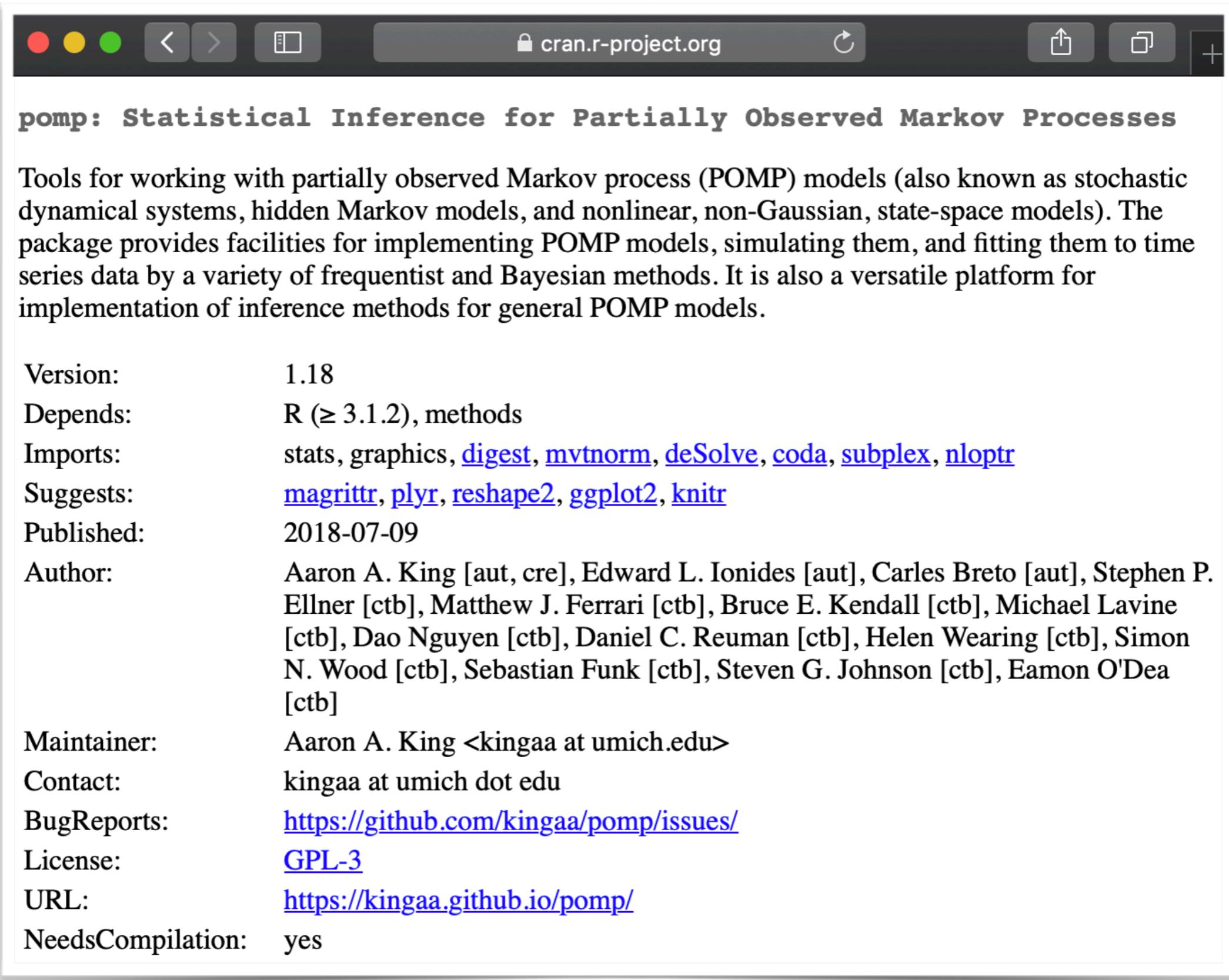
# Computational efficiency vs Statistical efficiency



Fasiolo et al. (2015)

# **Computational efficiency vs Coding efficiency**

# pomp



The screenshot shows a web browser window with the URL `cran.r-project.org` in the address bar. The page content is the CRAN package page for `pomp`. The title is **pomp: Statistical Inference for Partially Observed Markov Processes**. The package description states: "Tools for working with partially observed Markov process (POMP) models (also known as stochastic dynamical systems, hidden Markov models, and nonlinear, non-Gaussian, state-space models). The package provides facilities for implementing POMP models, simulating them, and fitting them to time series data by a variety of frequentist and Bayesian methods. It is also a versatile platform for implementation of inference methods for general POMP models." Below the description, there is a table of package details:

Version:	1.18
Depends:	R ( $\geq 3.1.2$ ), methods
Imports:	stats, graphics, <a href="#">digest</a> , <a href="#">mvtnorm</a> , <a href="#">deSolve</a> , <a href="#">coda</a> , <a href="#">subplex</a> , <a href="#">nloptr</a>
Suggests:	<a href="#">magrittr</a> , <a href="#">plyr</a> , <a href="#">reshape2</a> , <a href="#">ggplot2</a> , <a href="#">knitr</a>
Published:	2018-07-09
Author:	Aaron A. King [aut, cre], Edward L. Ionides [aut], Carles Breto [aut], Stephen P. Ellner [ctb], Matthew J. Ferrari [ctb], Bruce E. Kendall [ctb], Michael Lavine [ctb], Dao Nguyen [ctb], Daniel C. Reuman [ctb], Helen Wearing [ctb], Simon N. Wood [ctb], Sebastian Funk [ctb], Steven G. Johnson [ctb], Eamon O'Dea [ctb]
Maintainer:	Aaron A. King < <a href="mailto:kingaa@umich.edu">kingaa at umich.edu</a> >
Contact:	<a href="mailto:kingaa@umich.edu">kingaa at umich dot edu</a>
BugReports:	<a href="https://github.com/kingaa/pomp/issues/">https://github.com/kingaa/pomp/issues/</a>
License:	<a href="#">GPL-3</a>
URL:	<a href="https://kingaa.github.io/pomp/">https://kingaa.github.io/pomp/</a>
NeedsCompilation:	yes

# pomp

```
sir_step <- Csnippet("
  double dN_SI = rbinom(S,1-exp(-Beta*I/N*dt));
  double dN_IR = rbinom(I,1-exp(-gamma*dt));
  S -= dN_SI;
  I += dN_SI - dN_IR;
  R += dN_IR;
  H += dN_IR;
")

sir_init <- Csnippet("
  S = N-1;
  I = 1;
  R = 0;
  H = 0;
")

pomp(sir,rprocess=euler.sim(sir_step,delta.t=1/6),initializer=sir_init,
  paramnames=c("Beta","gamma","N"),statenames=c("S","I","R","H")) -> sir
```



kingaa.github.io

## User guides and tutorials

Statistical Inference for Partially Observed Markov Processes via the R Package <b>pomp</b> (a <i>Journal of Statistical Software</i> paper)	(PDF) (R)
Getting started with <b>pomp</b>	(HTML) (R)
<b>pomp</b> version 2 upgrade guide	(HTML)
Coding POMP models: R vs C snippets	(HTML) (R)
IF2 Pseudocode and Example	(HTML) (R)
Measles in large and small towns (He et al. 2010)	(HTML) (R) (data)
Simulation-based Inference for Epidemiological Dynamics (a short course)	(HTML)
Model-based Inference in Ecology and Epidemiology (a short course)	(HTML)
Two perspectives on the <b>pomp</b> project	(HTML)
<b>pomp2</b> package manual	(PDF)
<b>pomp</b> package manual	(PDF)
Frequently Asked Questions	(HTML)
More <b>pomp</b> examples	(HTML)
Pimp my pomp	(HTML)
Package NEWS	(HTML)

pomp



The bioRxiv logo features the word "bioRxiv" in a large, bold, sans-serif font. The letter "R" is stylized with a red Greek chi symbol (χ). Below the main title, the text "THE PREPRINT SERVER FOR BIOLOGY" is written in a smaller, all-caps, sans-serif font.

HOME | ABSTRACT

Search

New Results

## pomp-Astic Inference For Epidemic Models: Simple Vs. Complex

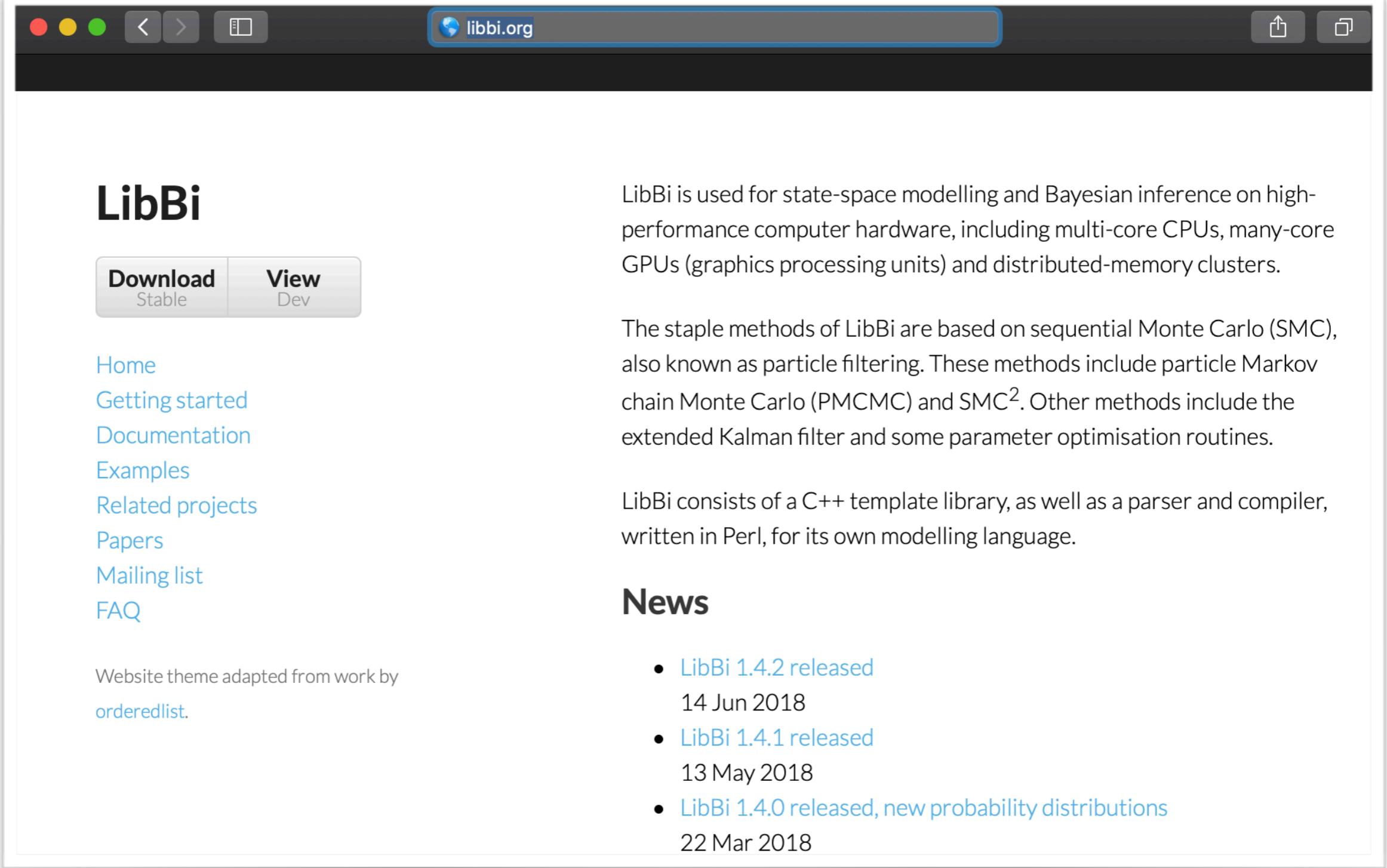
id Theresa Stocks, Tom Britton, id Michael Höhle

doi: <https://doi.org/10.1101/125880>

This article is a preprint and has not been peer-reviewed [what does this mean?].

**Abstract** Info/History Metrics Preview PDF

# LibBi



The screenshot shows a Mac OS X desktop environment with a window for the LibBi website. The window title bar says "libbi.org". The main content area of the website features the LibBi logo at the top left. Below it are two buttons: "Download Stable" and "View Dev". To the right of these buttons is a large text block describing LibBi's use for state-space modelling and Bayesian inference on various hardware. Further down, there are sections for "News" (listing recent releases) and "Documentation" (with links to "Home", "Getting started", "Examples", "Related projects", "Papers", "Mailing list", and "FAQ"). A copyright notice at the bottom credits "orderedlist" for the website theme.

LibBi is used for state-space modelling and Bayesian inference on high-performance computer hardware, including multi-core CPUs, many-core GPUs (graphics processing units) and distributed-memory clusters.

The staple methods of LibBi are based on sequential Monte Carlo (SMC), also known as particle filtering. These methods include particle Markov chain Monte Carlo (PMCMC) and SMC<sup>2</sup>. Other methods include the extended Kalman filter and some parameter optimisation routines.

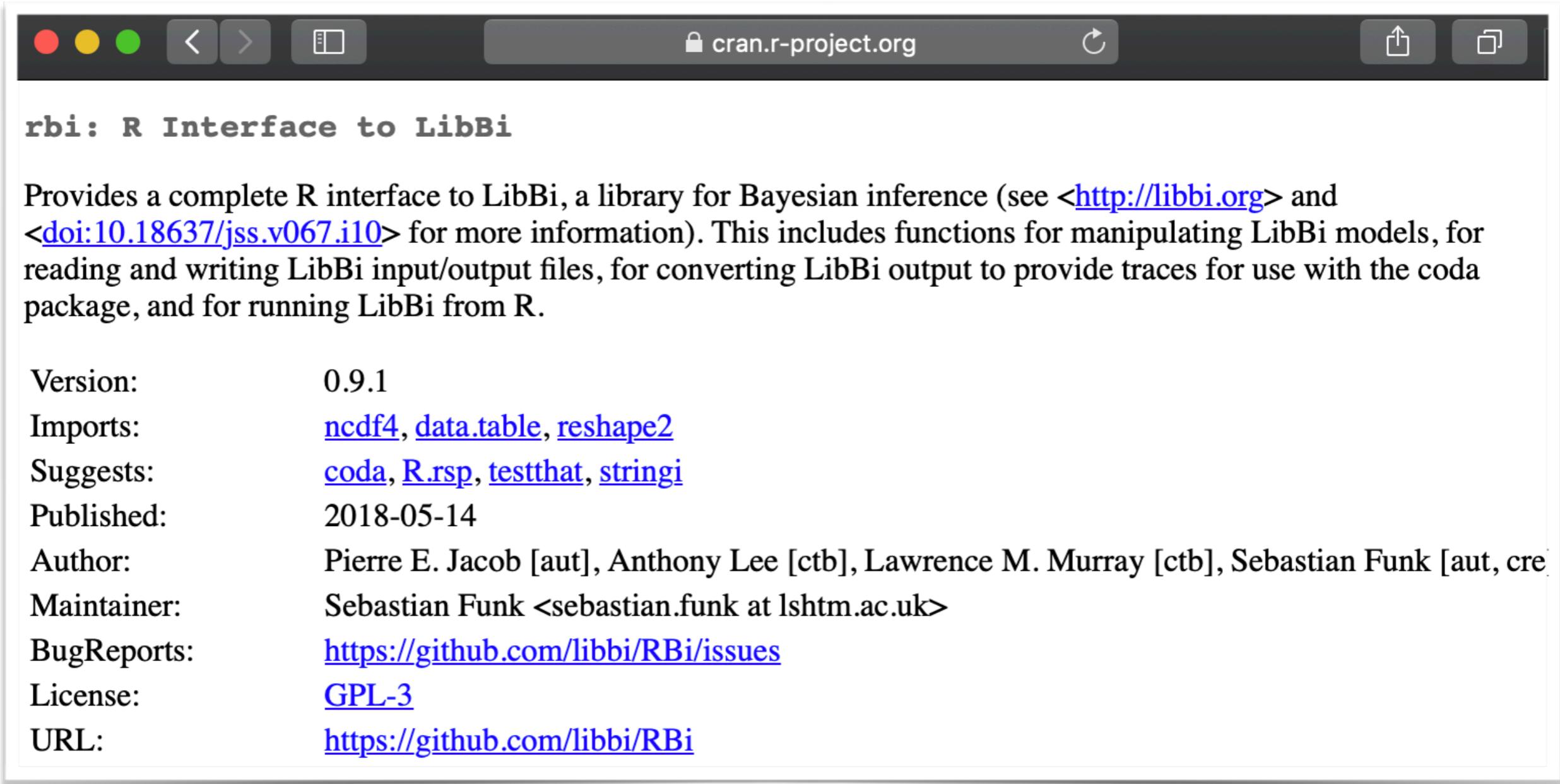
LibBi consists of a C++ template library, as well as a parser and compiler, written in Perl, for its own modelling language.

## News

- [LibBi 1.4.2 released](#)  
14 Jun 2018
- [LibBi 1.4.1 released](#)  
13 May 2018
- [LibBi 1.4.0 released, new probability distributions](#)  
22 Mar 2018

Website theme adapted from work by  
[orderedlist](#).

# rbi

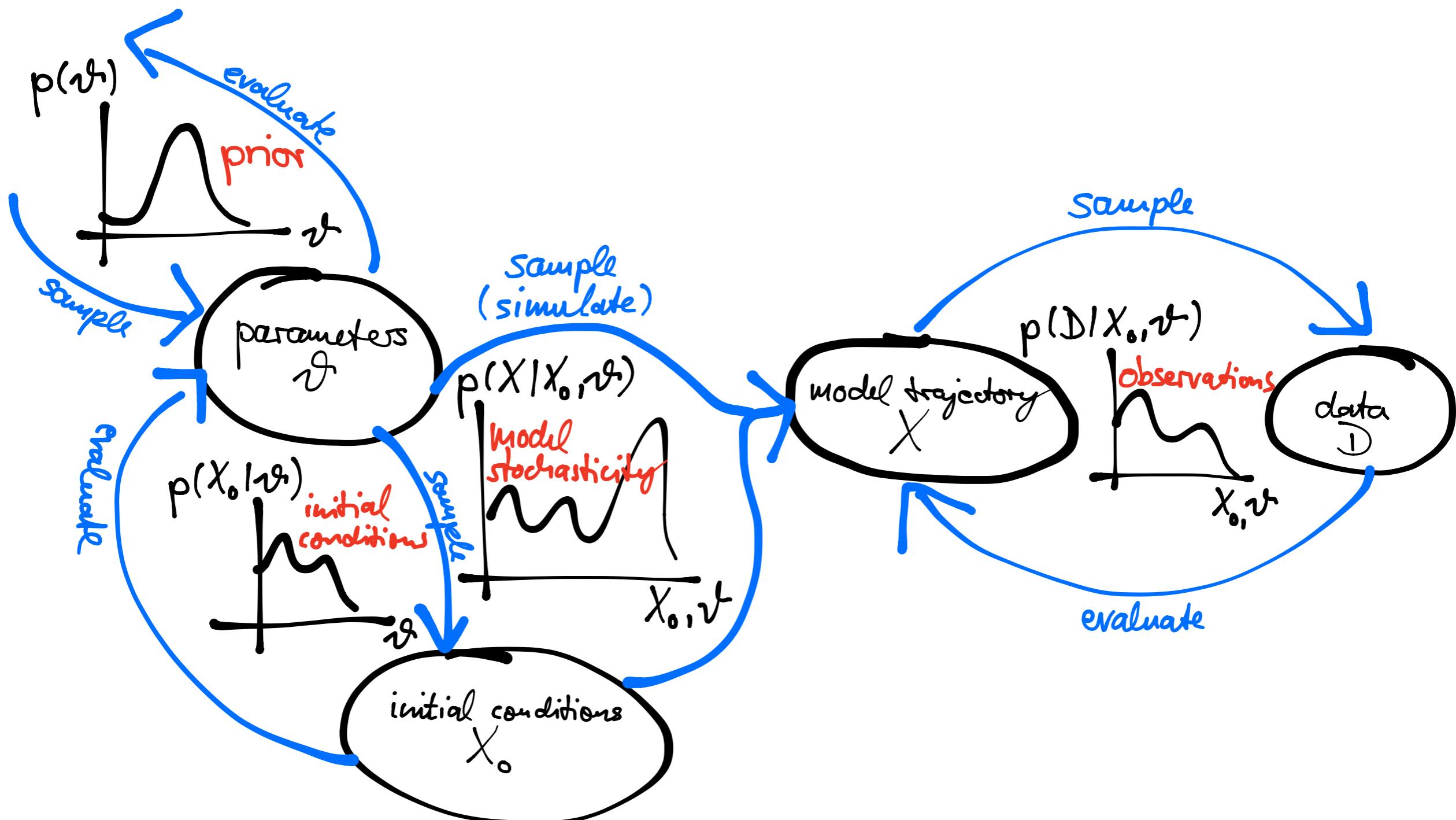


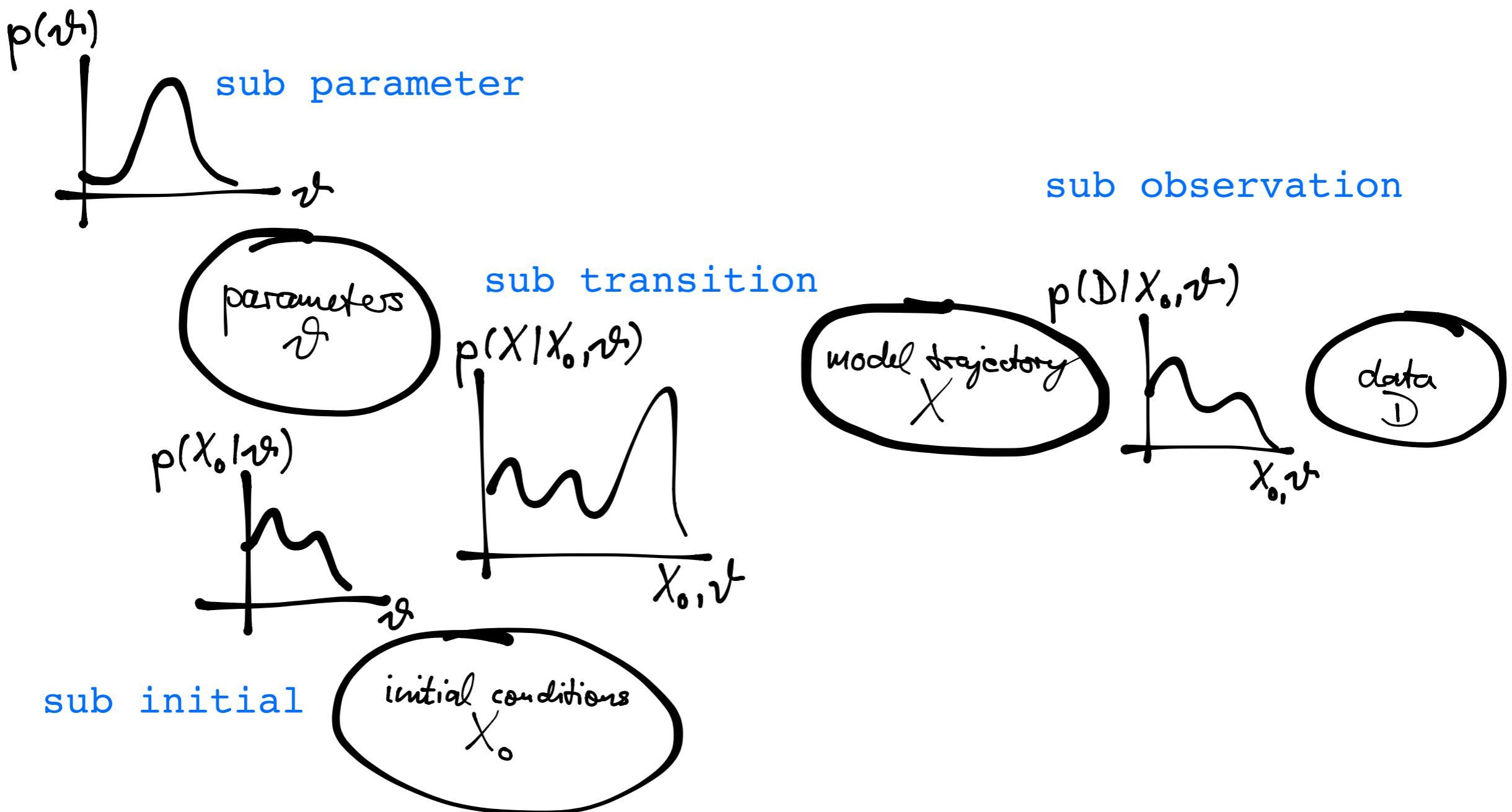
The screenshot shows a web browser window with the URL [cran.r-project.org](https://cran.r-project.org) in the address bar. The page content is about the **rbi** package, which provides an R interface to LibBi. The package version is 0.9.1, published on 2018-05-14, and is maintained by Sebastian Funk. It imports `ncdf4`, `data.table`, and `reshape2`, and suggests `coda`, `R.rsp`, `testthat`, and `stringi`. The package author is Pierre E. Jacob, and it is licensed under GPL-3. The URL for the GitHub repository is <https://github.com/libbi/RBi>.

**rbi: R Interface to LibBi**

Provides a complete R interface to LibBi, a library for Bayesian inference (see <<http://libbi.org>> and <[doi:10.18637/jss.v067.i10](https://doi.org/10.18637/jss.v067.i10)> for more information). This includes functions for manipulating LibBi models, for reading and writing LibBi input/output files, for converting LibBi output to provide traces for use with the `coda` package, and for running LibBi from R.

Version: 0.9.1  
Imports: [ncdf4](#), [data.table](#), [reshape2](#)  
Suggests: [coda](#), [R.rsp](#), [testthat](#), [stringi](#)  
Published: 2018-05-14  
Author: Pierre E. Jacob [aut], Anthony Lee [ctb], Lawrence M. Murray [ctb], Sebastian Funk [aut, cre]  
Maintainer: Sebastian Funk <[sebastian.funk at lshtm.ac.uk](mailto:sebastian.funk@lshtm.ac.uk)>  
BugReports: <https://github.com/libbi/RBi/issues>  
License: [GPL-3](#)  
URL: <https://github.com/libbi/RBi>





```

model SIR_deterministic {
  const N = 1000; // population size
  const d_infection = 14; // duration of infection: 2 weeks

  state S, I, R; // susceptible, infectious, recovered

  obs Prevalence; // observations

  param R0; // basic reproduction number

  sub parameter {
    R0 ~ uniform(1, 3)
  }

  sub initial {
    S <- N - 1
    I <- 1
    R <- 0
  }

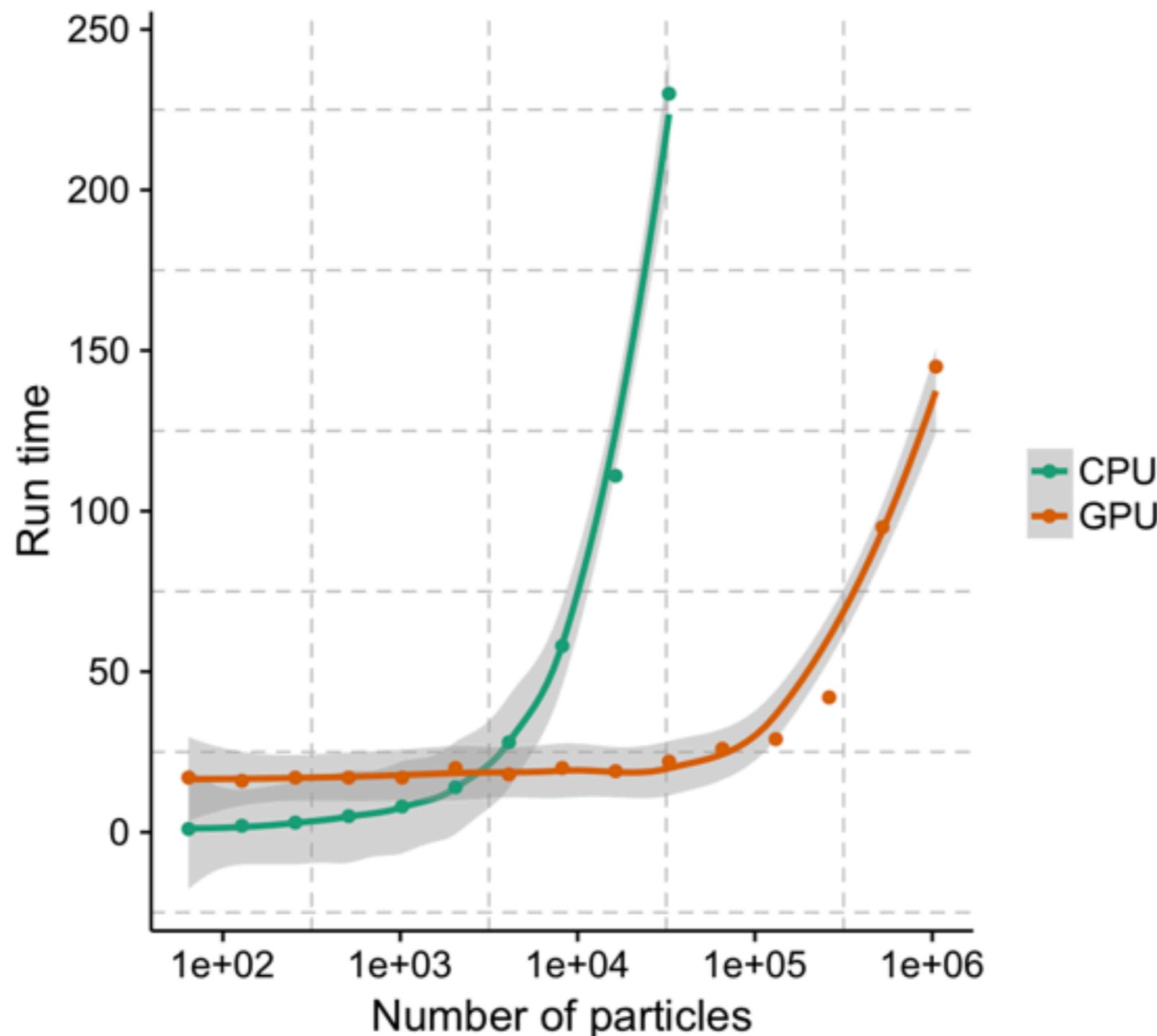
  sub transition { // daily time step
    inline beta = R0 / d_infection
    inline gamma = 1 / d_infection
    ode {
      dS/dt = - beta * S * I / N
      dI/dt = beta * S * I / N - gamma * I
      dR/dt = gamma * I
    }
  }

  sub observation {
    Prevalence ~ poisson(I)
  }
}

```

<http://sbfnk.github.io/rbi/rbi.html>

# pMCMC on GPUs with rbi/LibBi



# **Summary: choosing an inference approach**

Computational efficiency vs.

1. Biological / statistical model adequacy  
(having a “good” model)
2. Statistical efficiency  
(getting the “right” answer)
3. Coding efficiency  
(time investment / learning curve)