

# Fitting mathematical models to time series of infectious disease cases

(“Choosing the right inference approach for your problem”)

Sebastian Funk (@sbfnk)

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



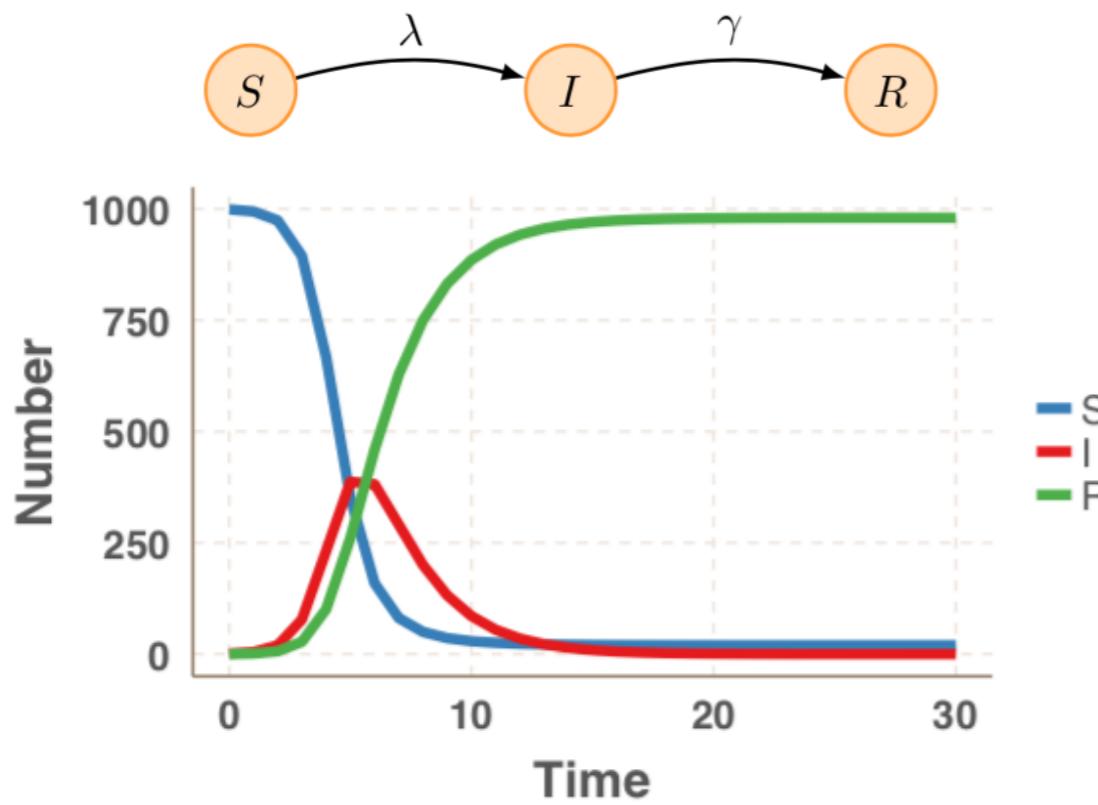
**centre for the  
mathematical  
modelling of  
infectious diseases**

LONDON  
SCHOOL *of*  
HYGIENE  
& TROPICAL  
MEDICINE



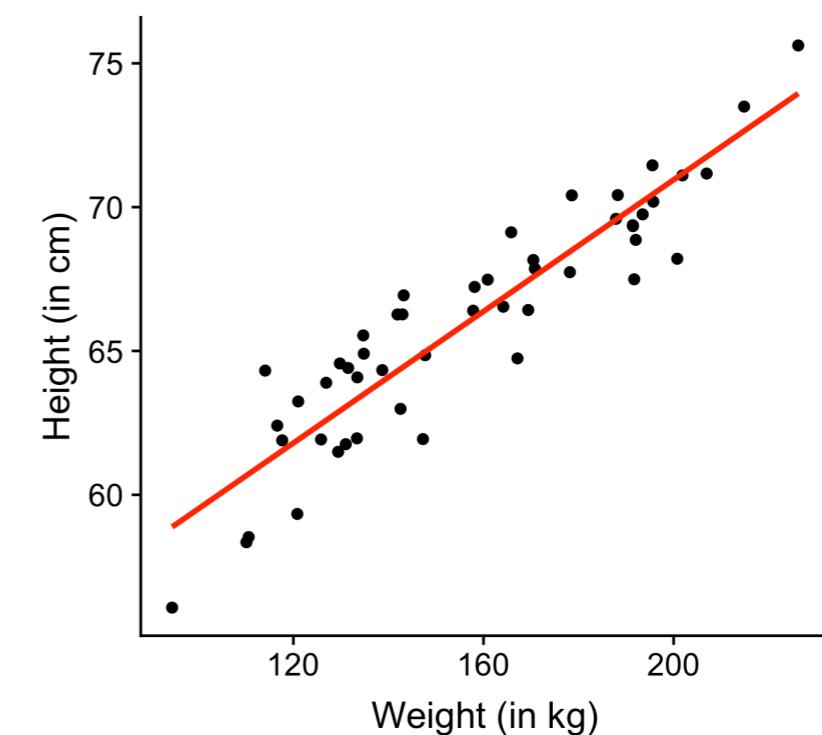
# Mechanistic approach

- Design model from first principles
- Focus on model behaviour under different scenarios / parameter ranges

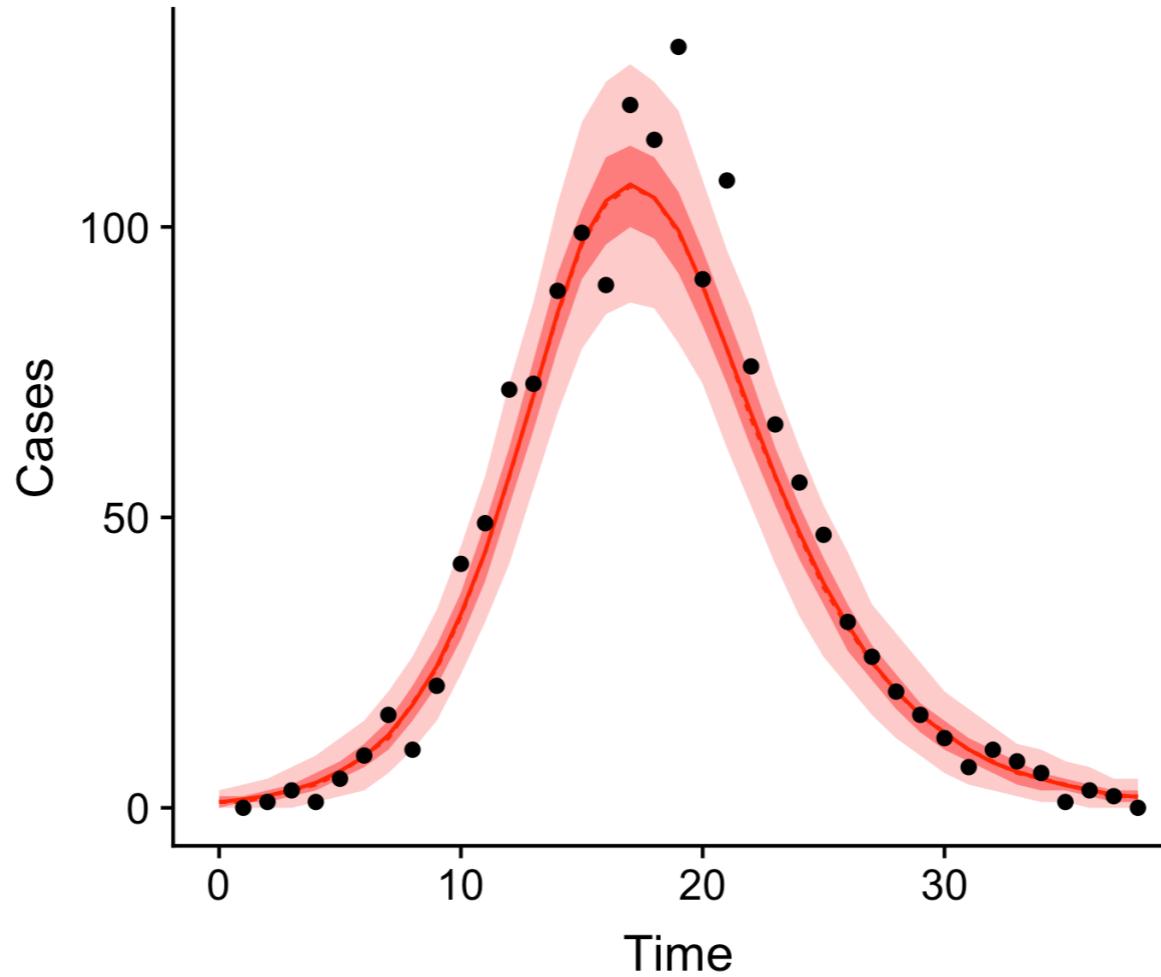


# Statistical approach

- Models as a tool to investigate data
- Model choice driven by data and hypotheses about relationships between variables



# Fitting mathematical models



Combine mechanistic and statistical approach for

- Parameter estimation
- Prediction

**Particlefilter**

HMCPMCMC MIFC MCMC ABC Likelihoodfree SMCPMC

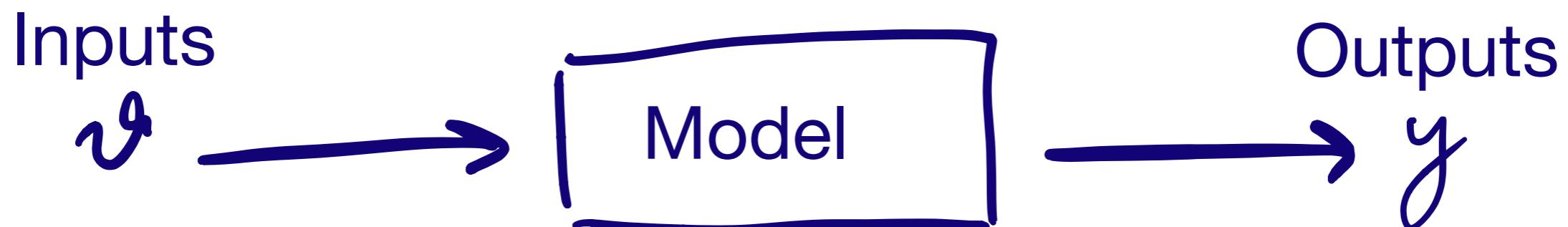
SMC pMCMC

bompmcmc

# CAVEAT

Subjective view of methods/approaches  
I am (somewhat) familiar with

A model takes input values  $\theta$  (parameters) and returns output values  $y$  (observations, data).

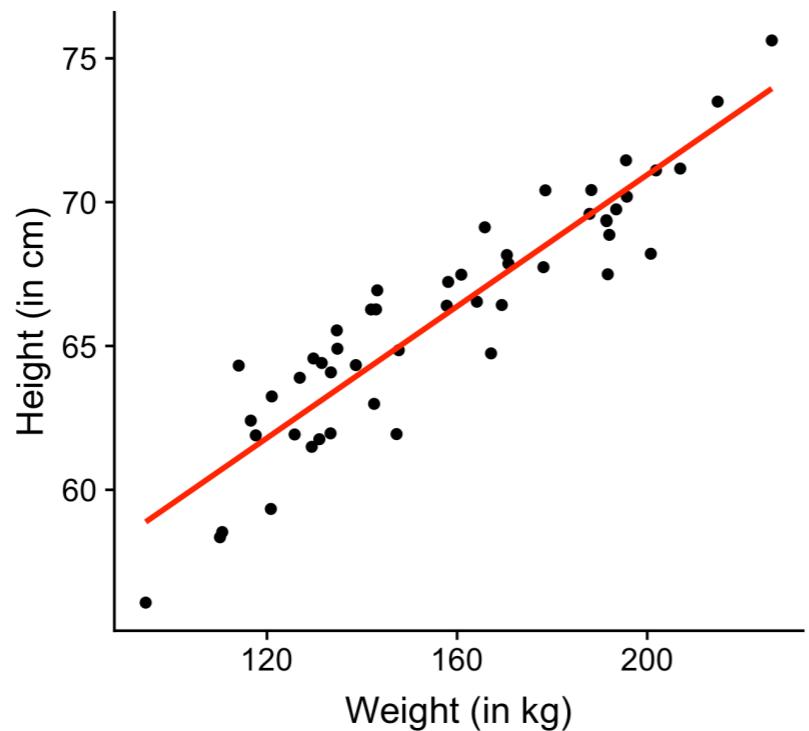


probability  $p(y|\theta)$  that  $y$  is the output, given inputs  $\theta$ .

# Linear model

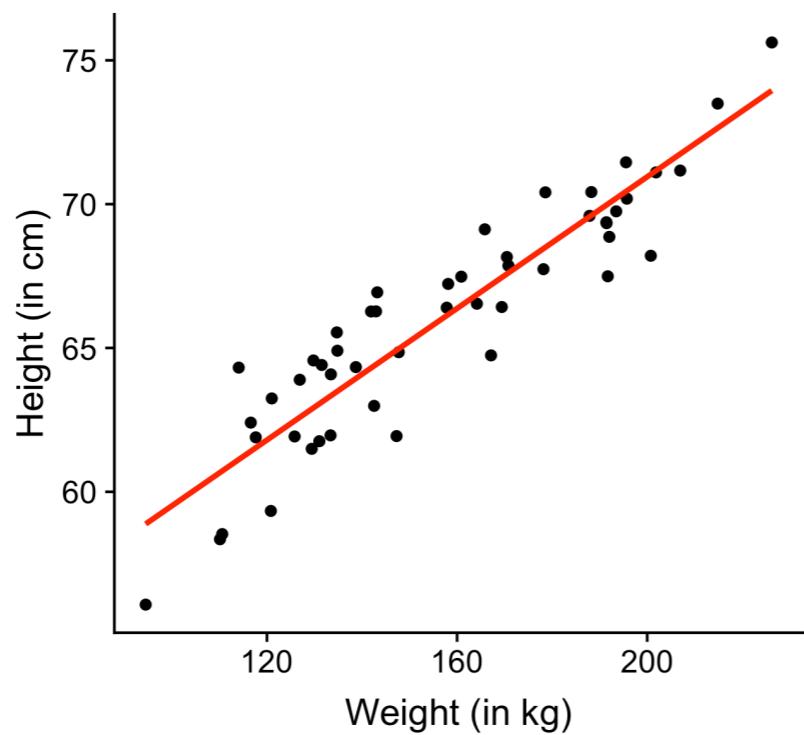
$$y = \beta_0 + \beta_1 x + \epsilon$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$



# Linear model

$$y \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2)$$

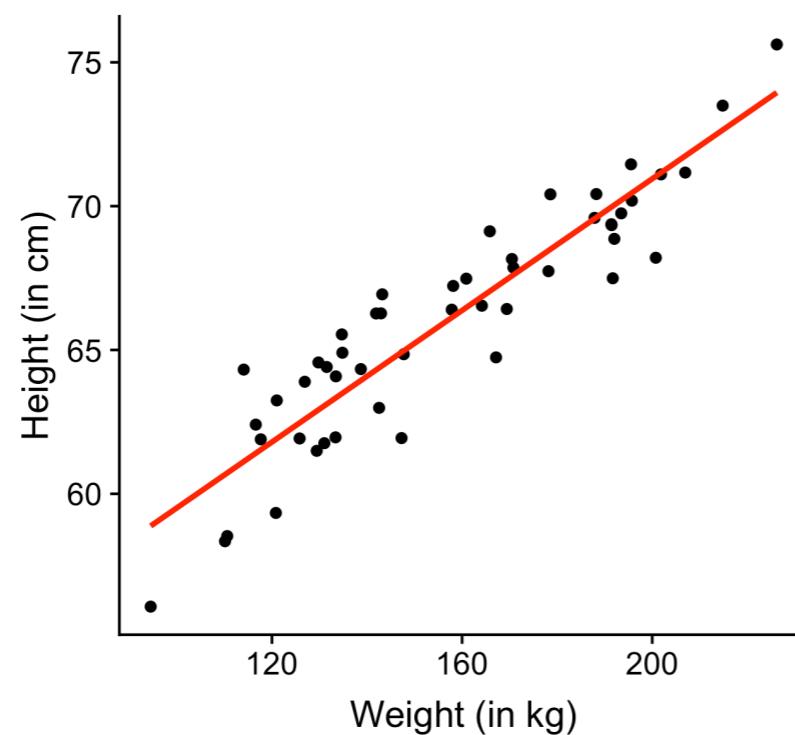


Normal  
probability density

## Linear model



$$p(y|\theta = \{\beta_0, \beta_1, \sigma\}) = f(y|\beta_0 + \beta_1 x, \sigma^2)$$



Normal  
probability density

## Linear model



$$p(y|\theta = \{\beta_0, \beta_1, \sigma\}) = f(y|\beta_0 + \beta_1 x, \sigma^2)$$

Given  $n$  data points  $(x_i, y_i)$ ,  $i=1..n$

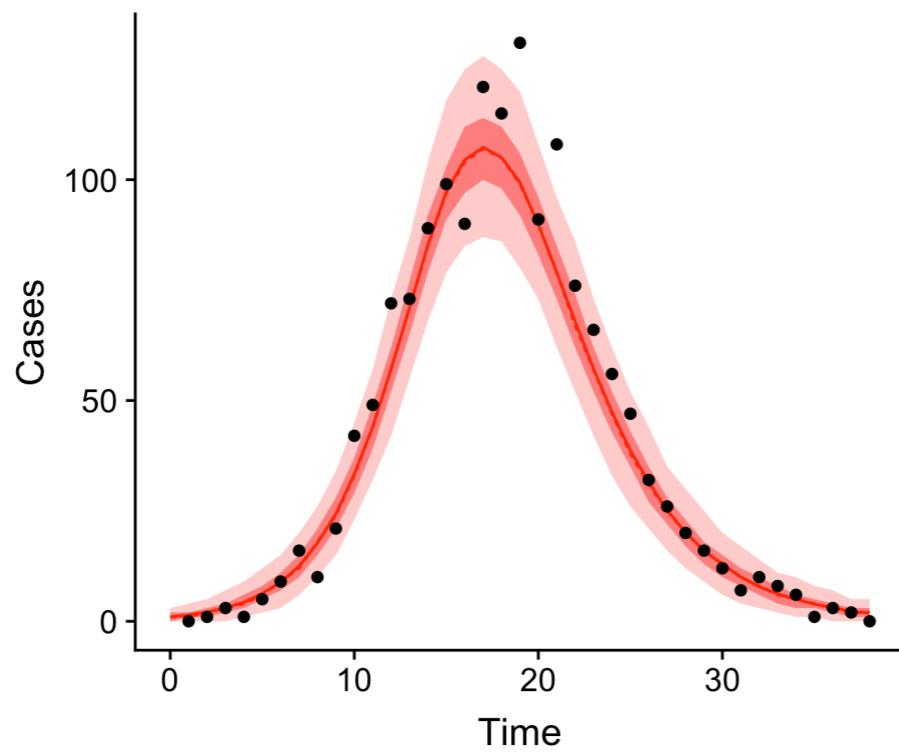
$$\begin{aligned} p(y_i|\theta) &= \prod_{i=1}^n p((y_i, x_i) | \beta_0, \beta_1, \sigma) \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} \prod_{i=1}^n e^{-\frac{y_i - (\beta_0 + \beta_1 x_i)}{2\sigma^2}} \end{aligned}$$

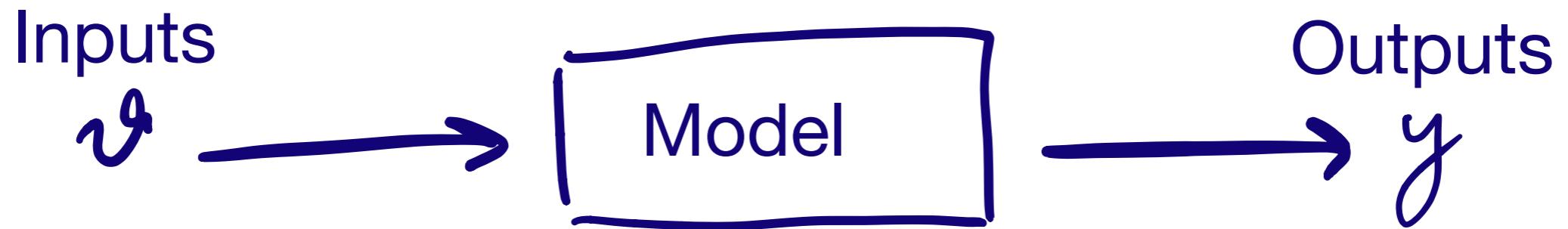
# Likelihood

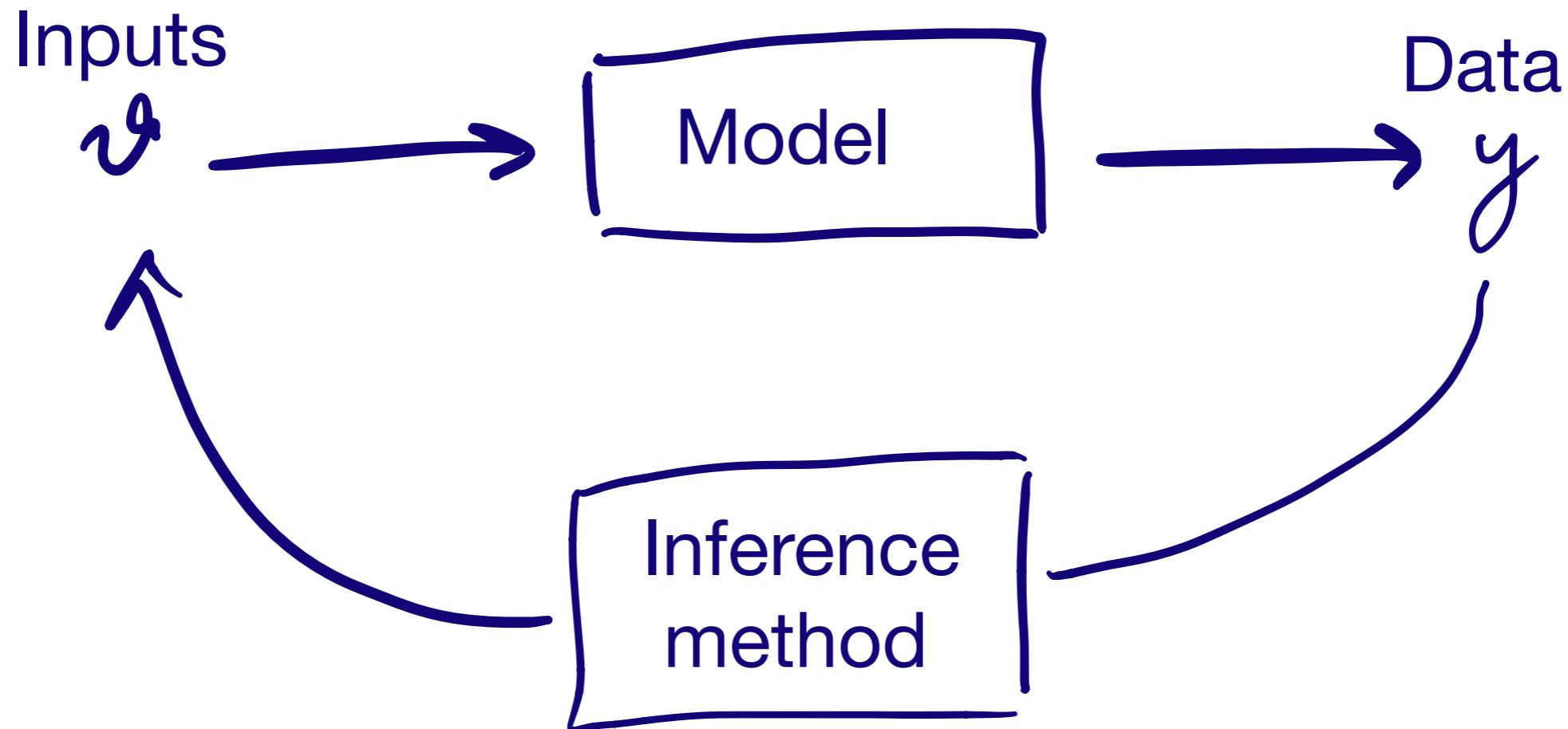
$$p(y = \text{Data} | \theta)$$

# Infectious disease model

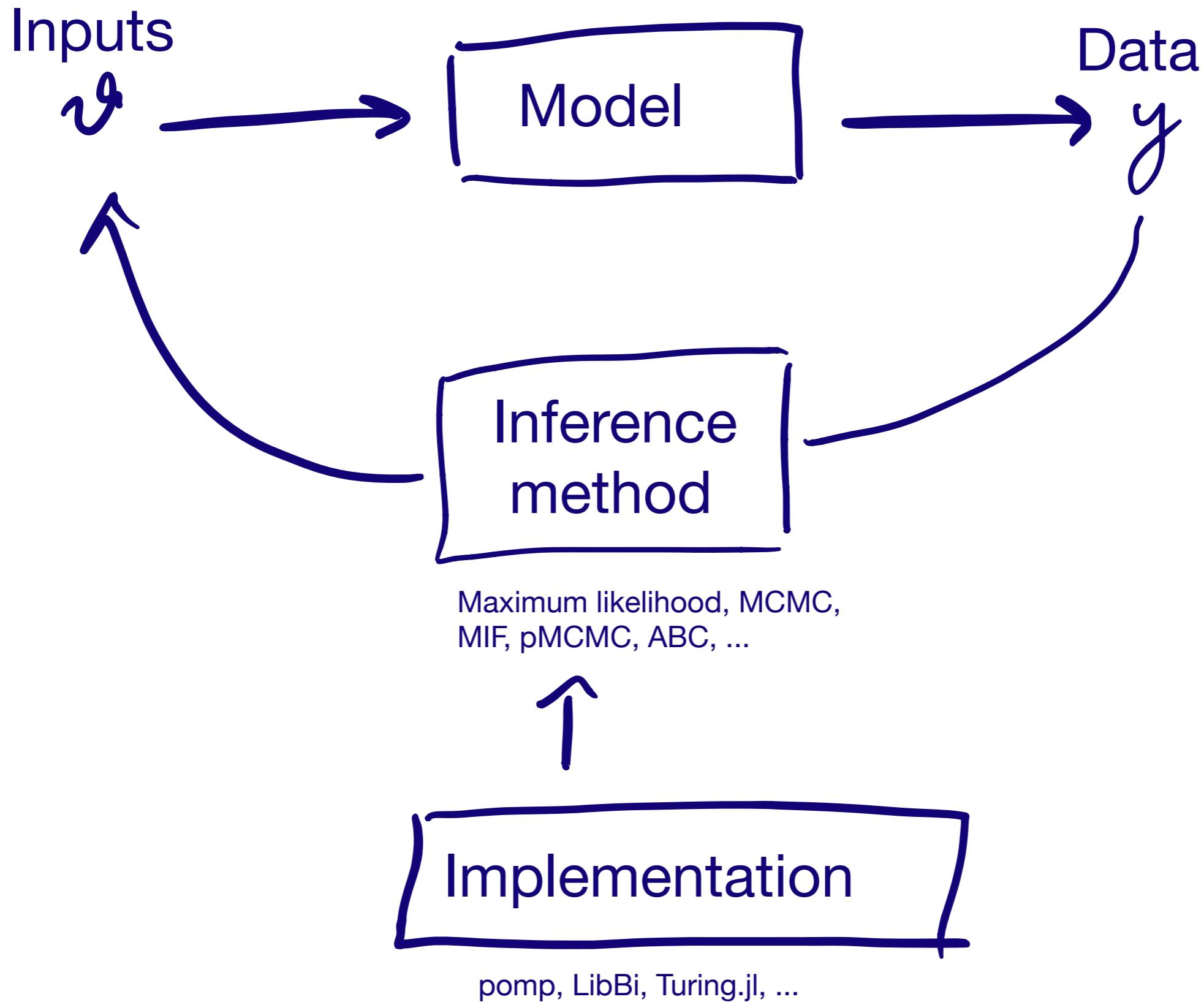
$$p(y|\theta) = ???$$







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





## Bayesian vs Frequentist



- Quantity of interest:  
posterior probability

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

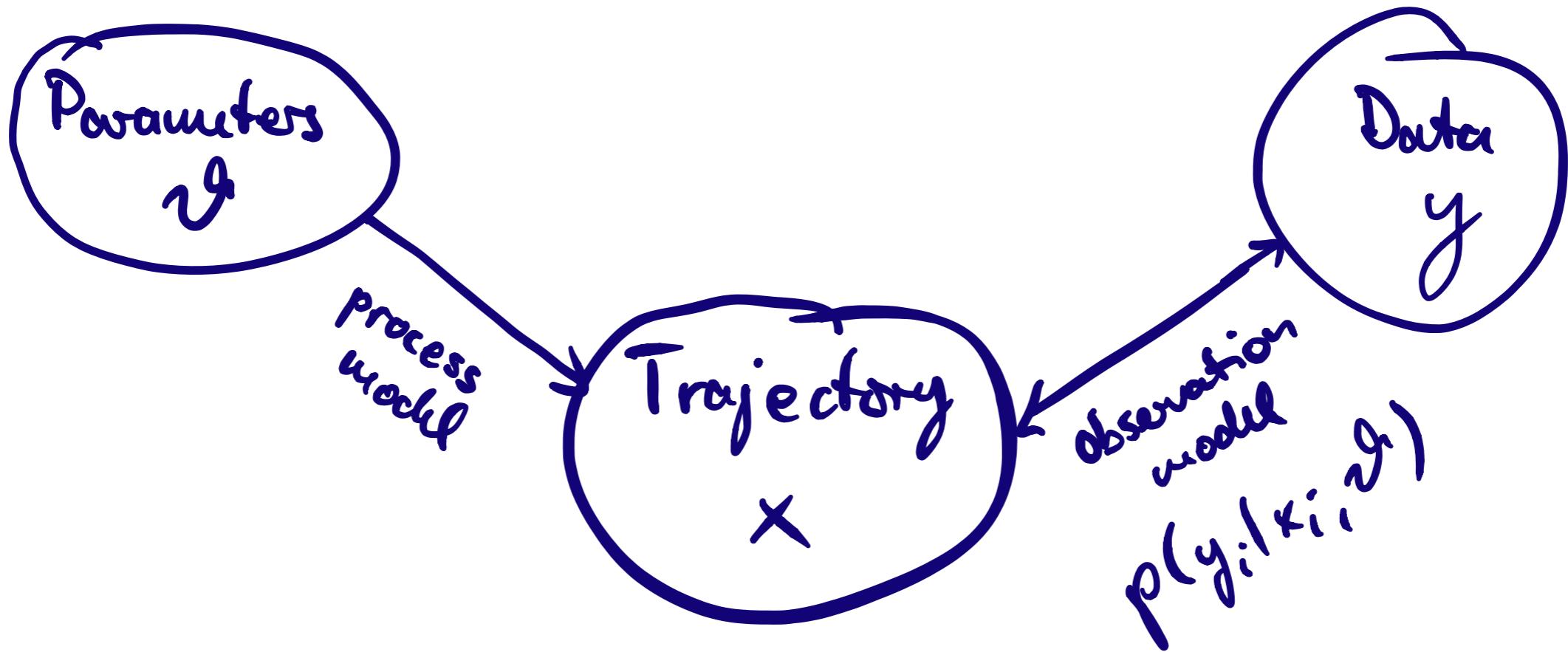
- $p(\theta)$ : prior belief
- $p(\theta|y)$ : updated belief after confronting model with data

- Quantity of interest:  
value of  $\theta$  that  
maximises likelihood

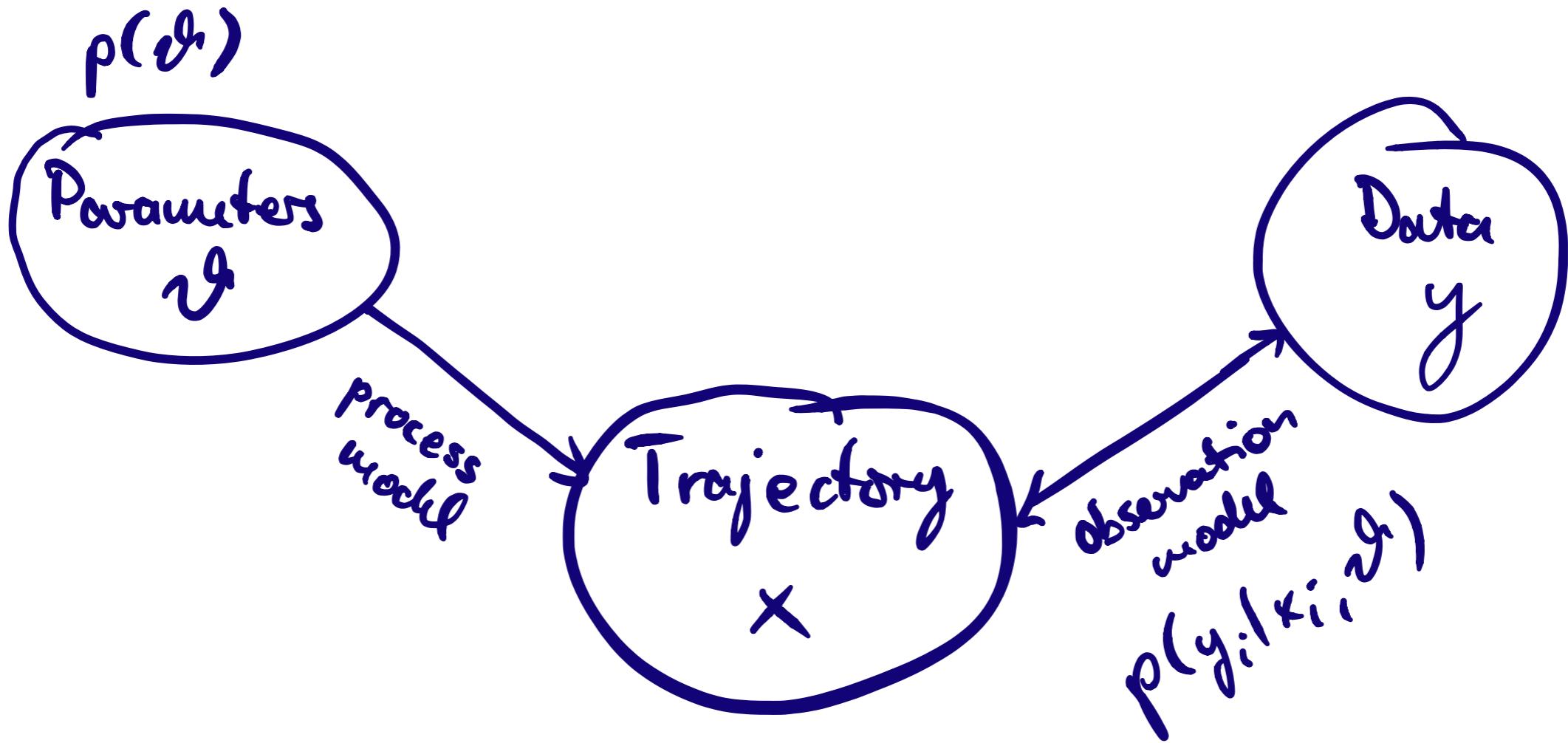
$$L(\theta) = p(y|\theta)$$

# Methods for model fitting

## 1. Deterministic models

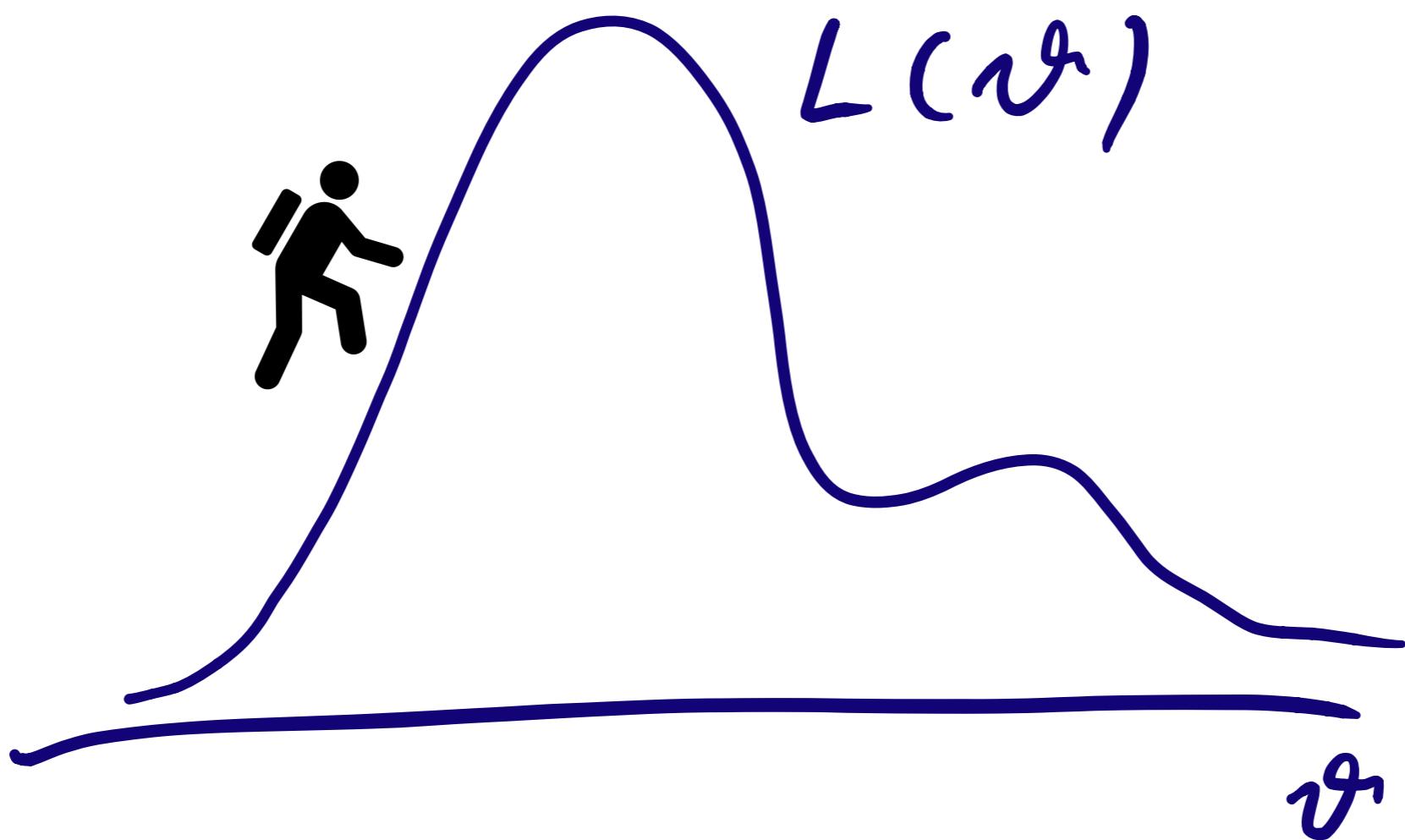


Can compute  $p(\text{Data}|\theta)$  for each  $\theta$ .



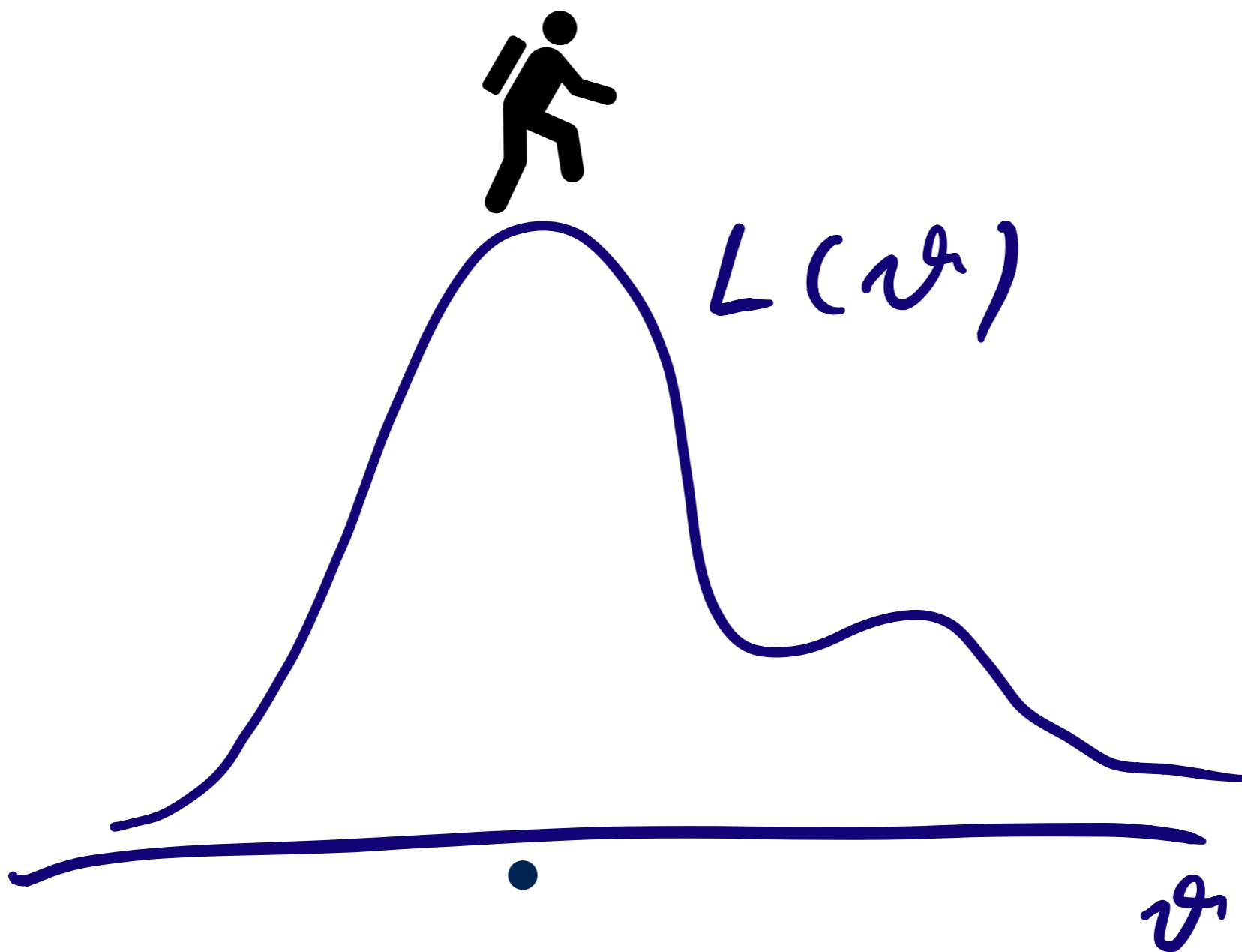
Can compute  $p(\theta|\text{Data})$  for each  $\theta$ .

# 1. Maximum likelihood via numerical optimisation



Find  $\theta$  that maximises  $L(\theta)=p(\text{Data}|\theta)$ .

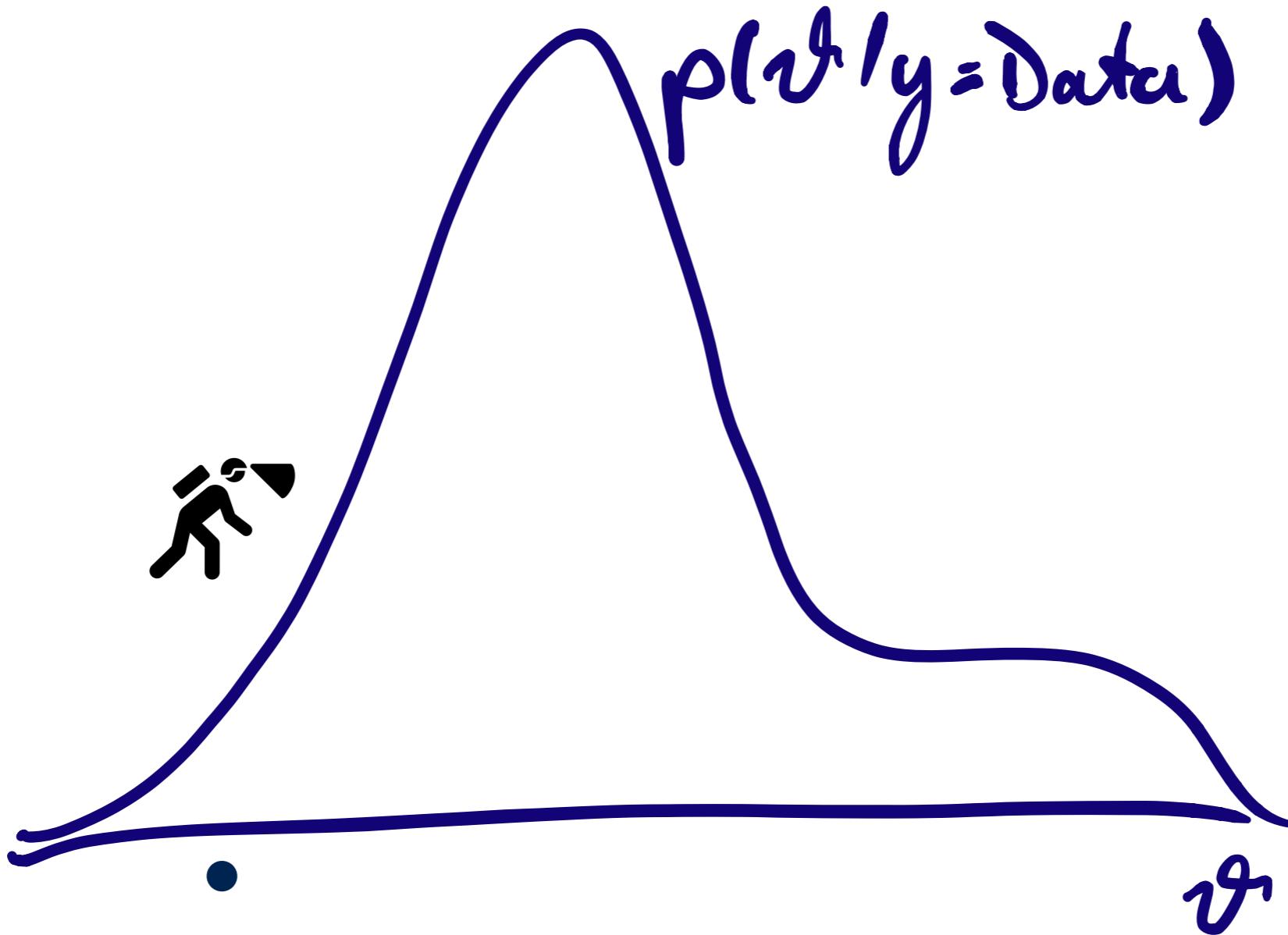
# 1. Maximum likelihood via numerical optimisation



Find  $\theta$  that maximises  $L(\theta) = p(\text{Data}|\theta)$ .



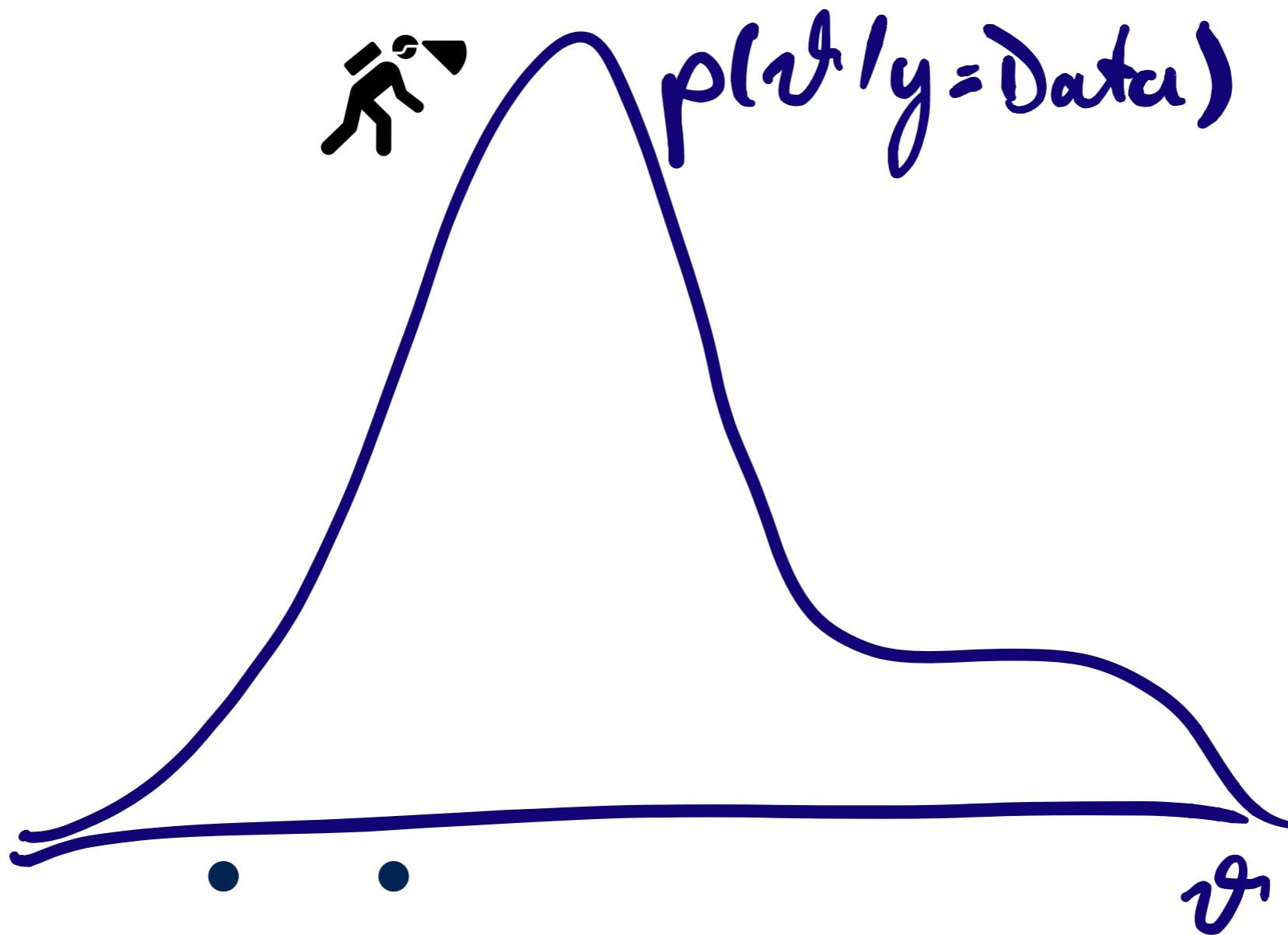
## 2. Markov-chain Monte Carlo



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



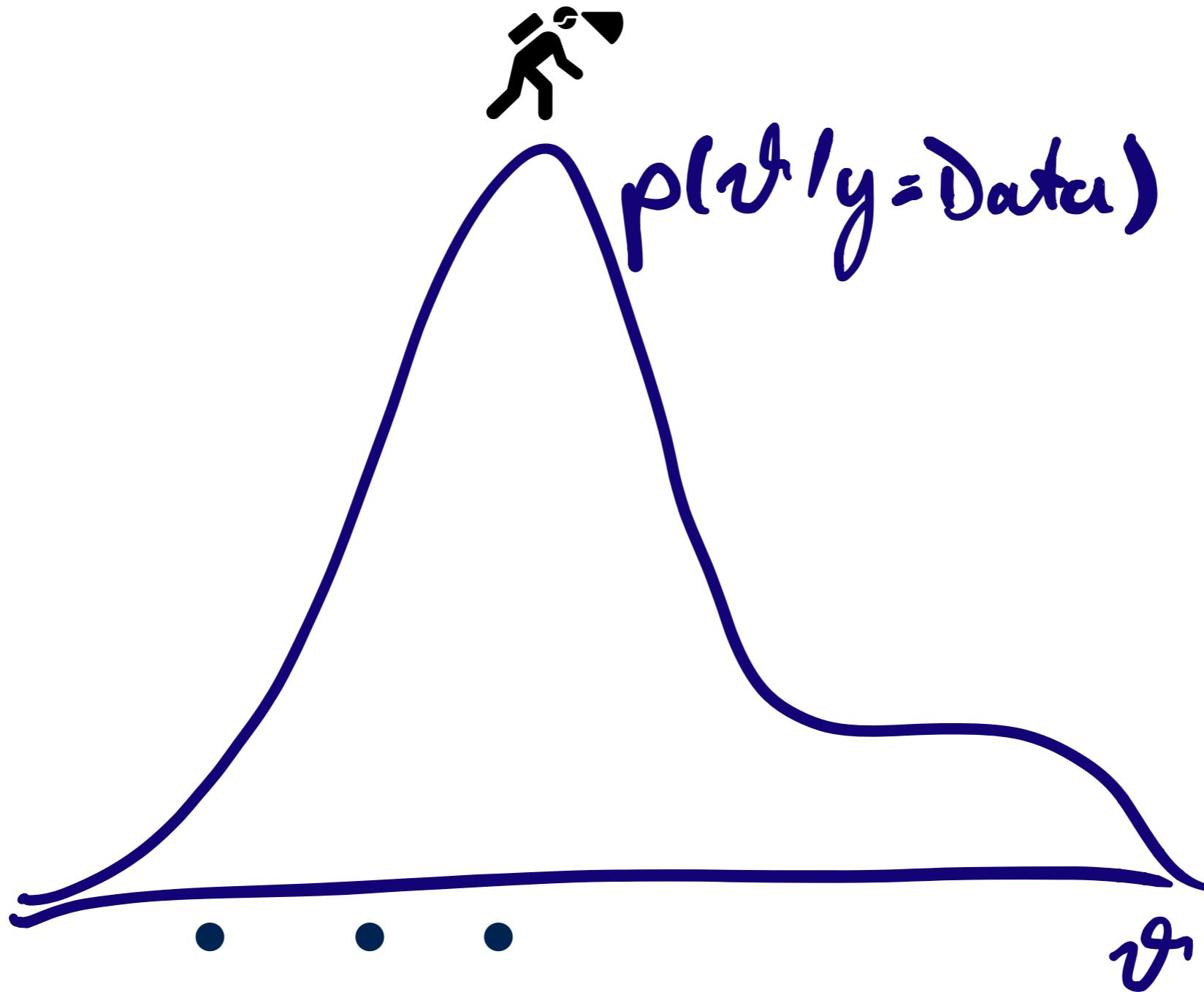
## 2. Markov-chain Monte Carlo



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



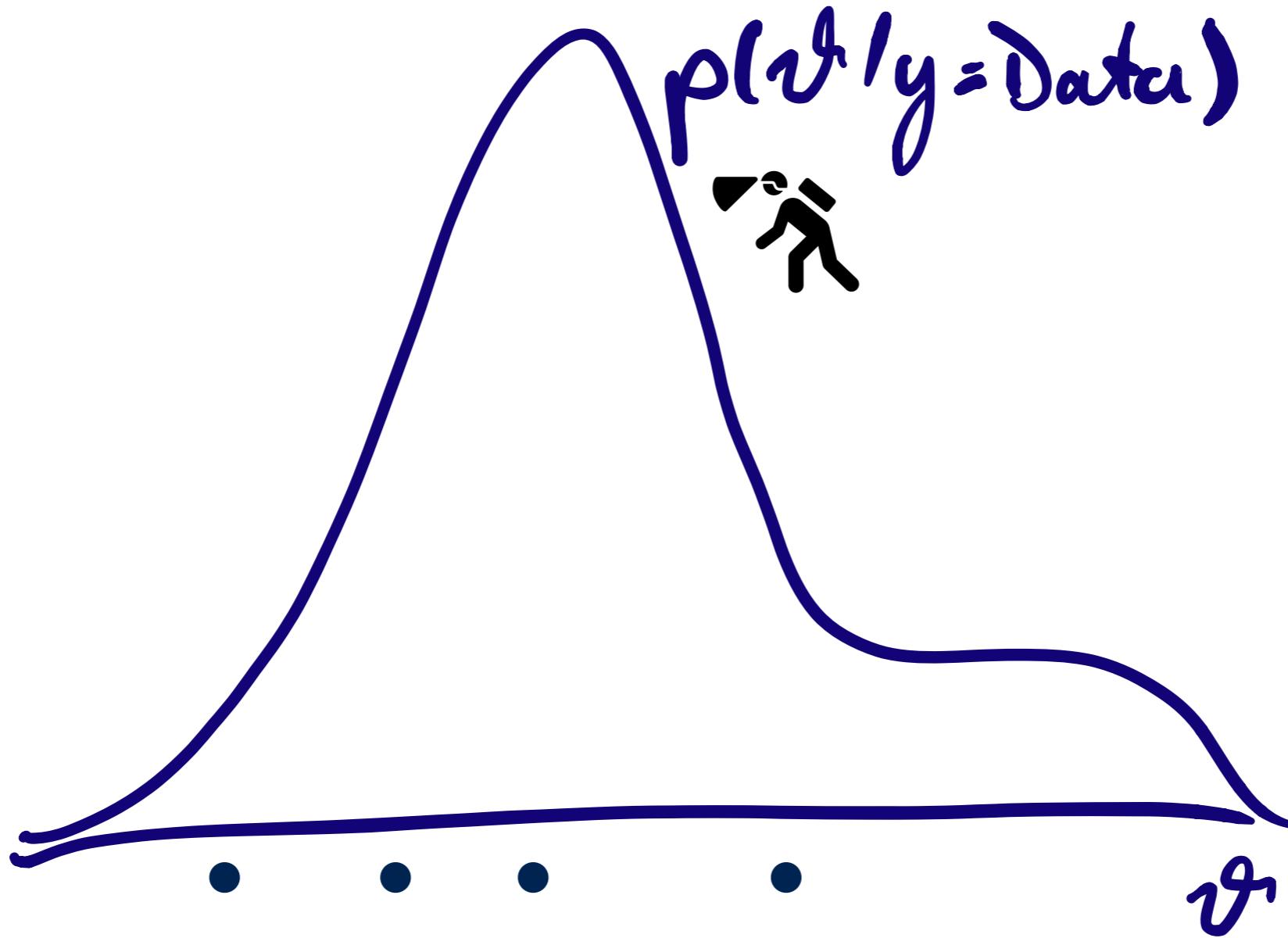
## 2. Markov-chain Monte Carlo



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



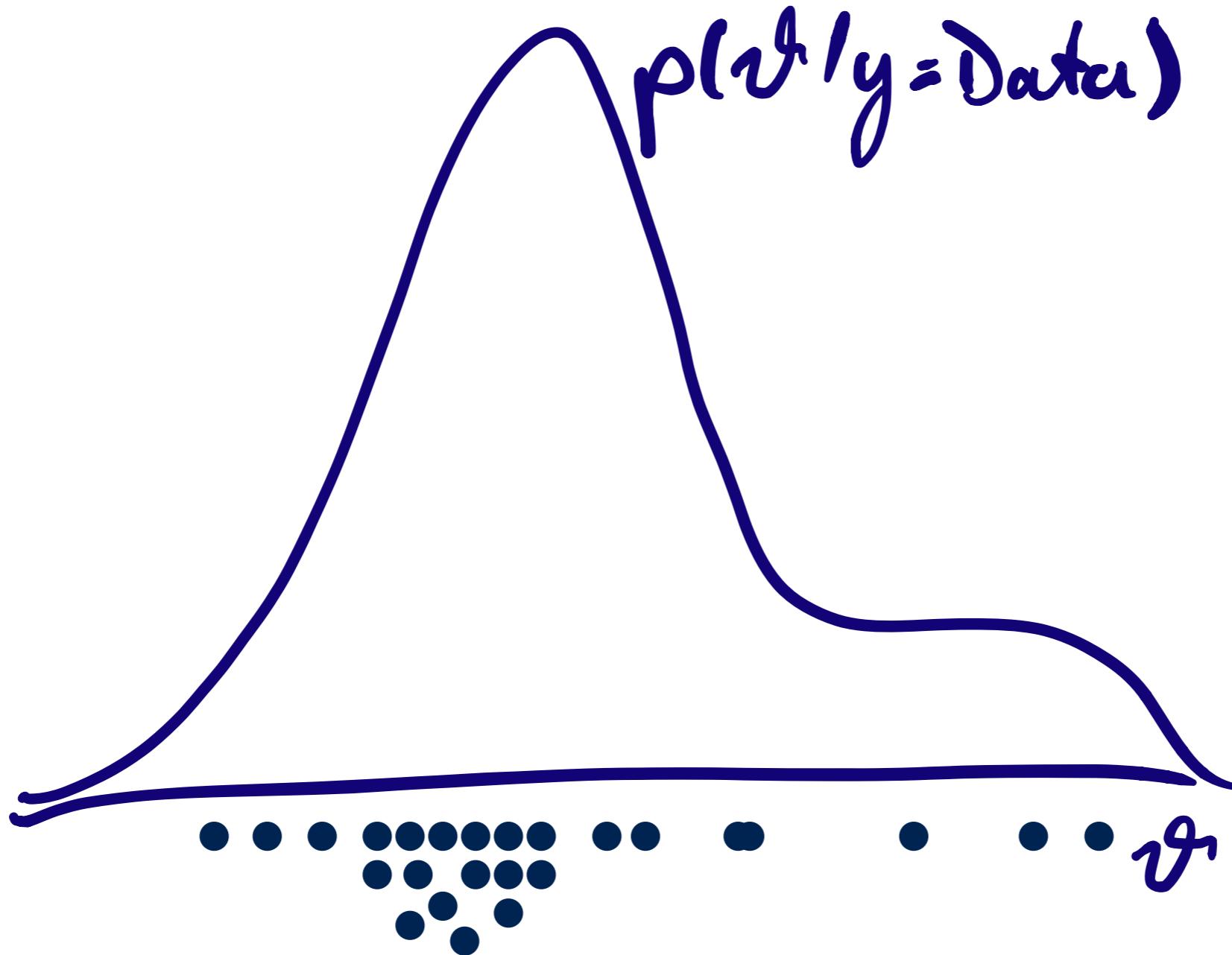
## 2. Markov-chain Monte Carlo



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

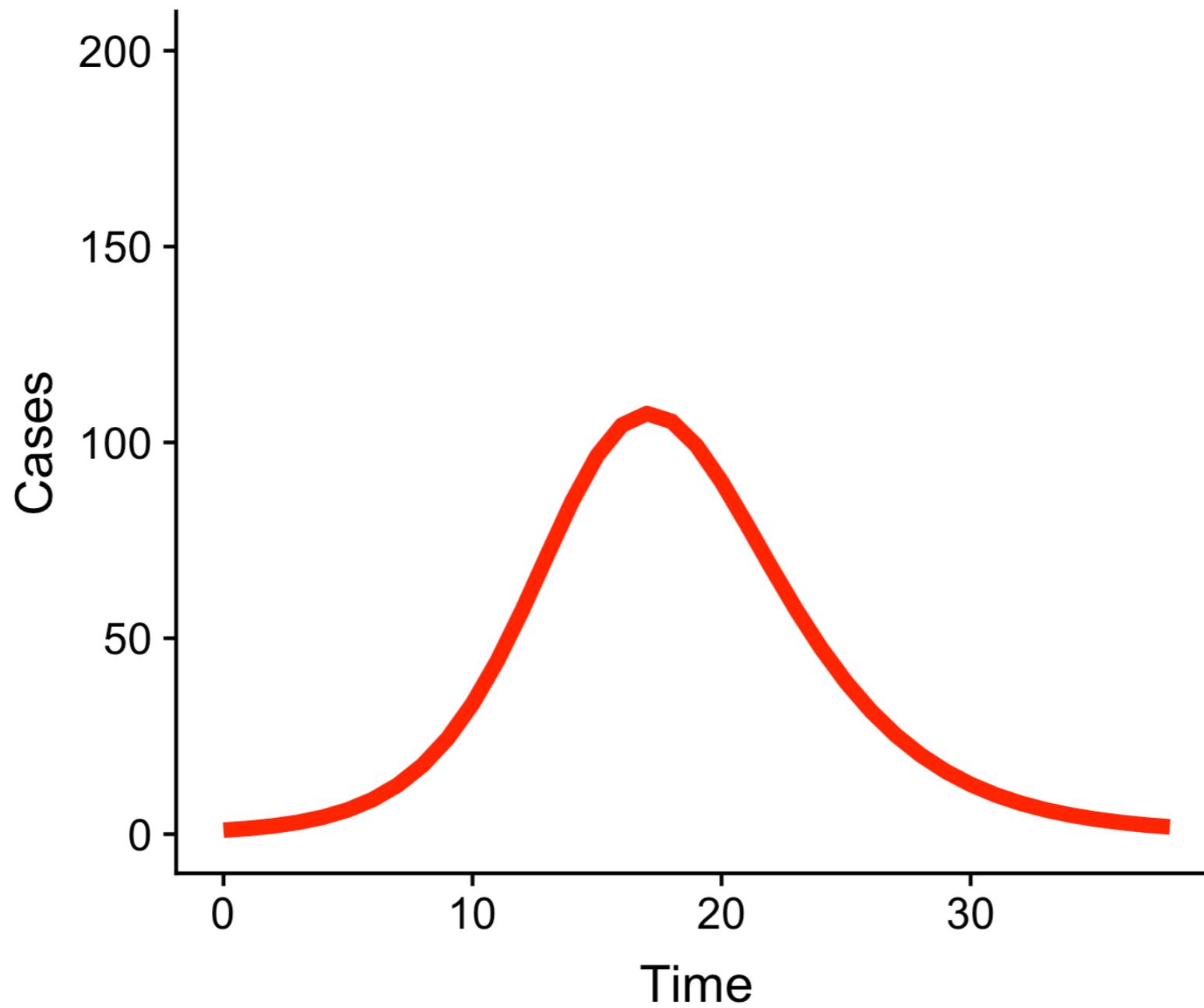


## 2. Markov-chain Monte Carlo

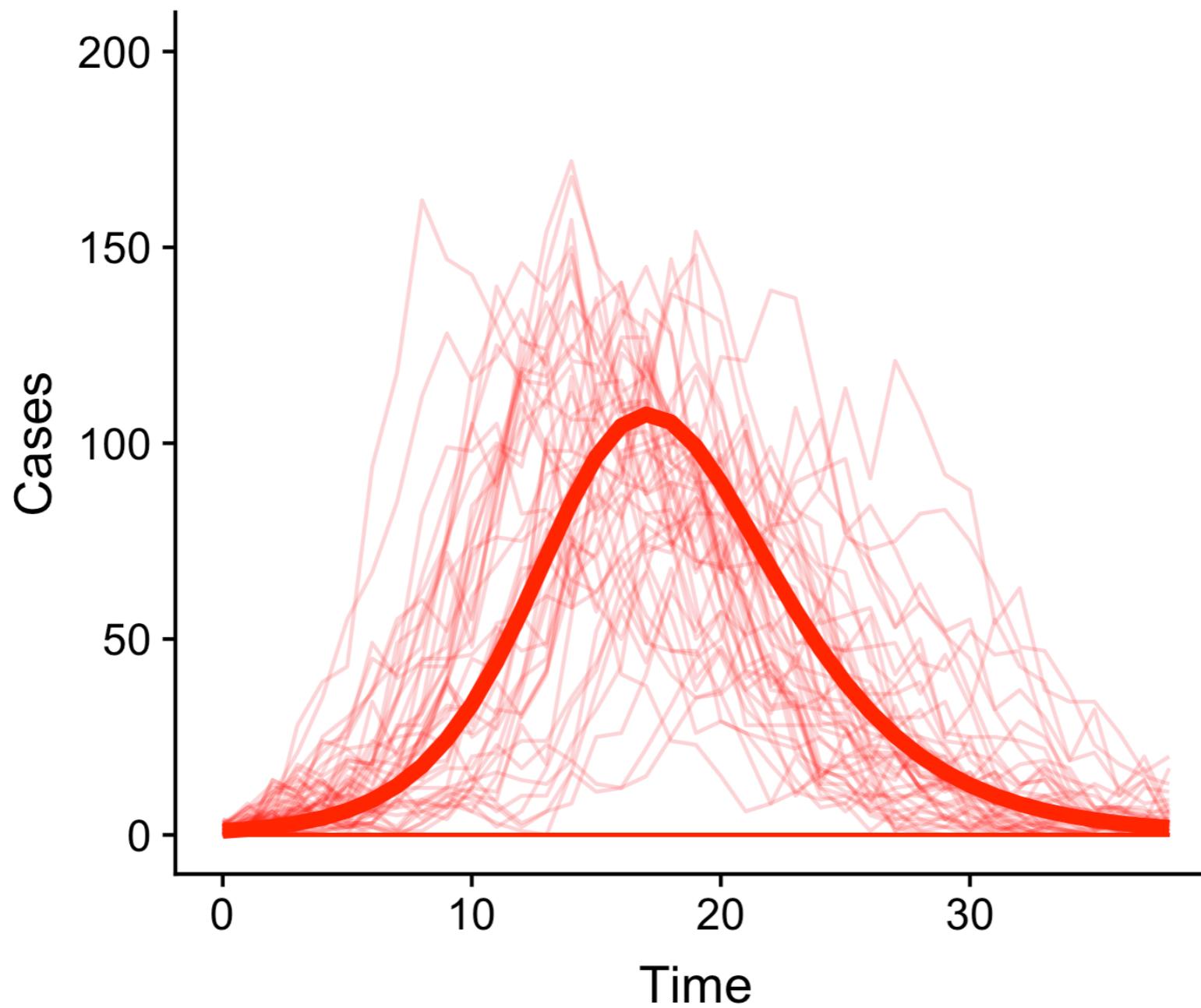


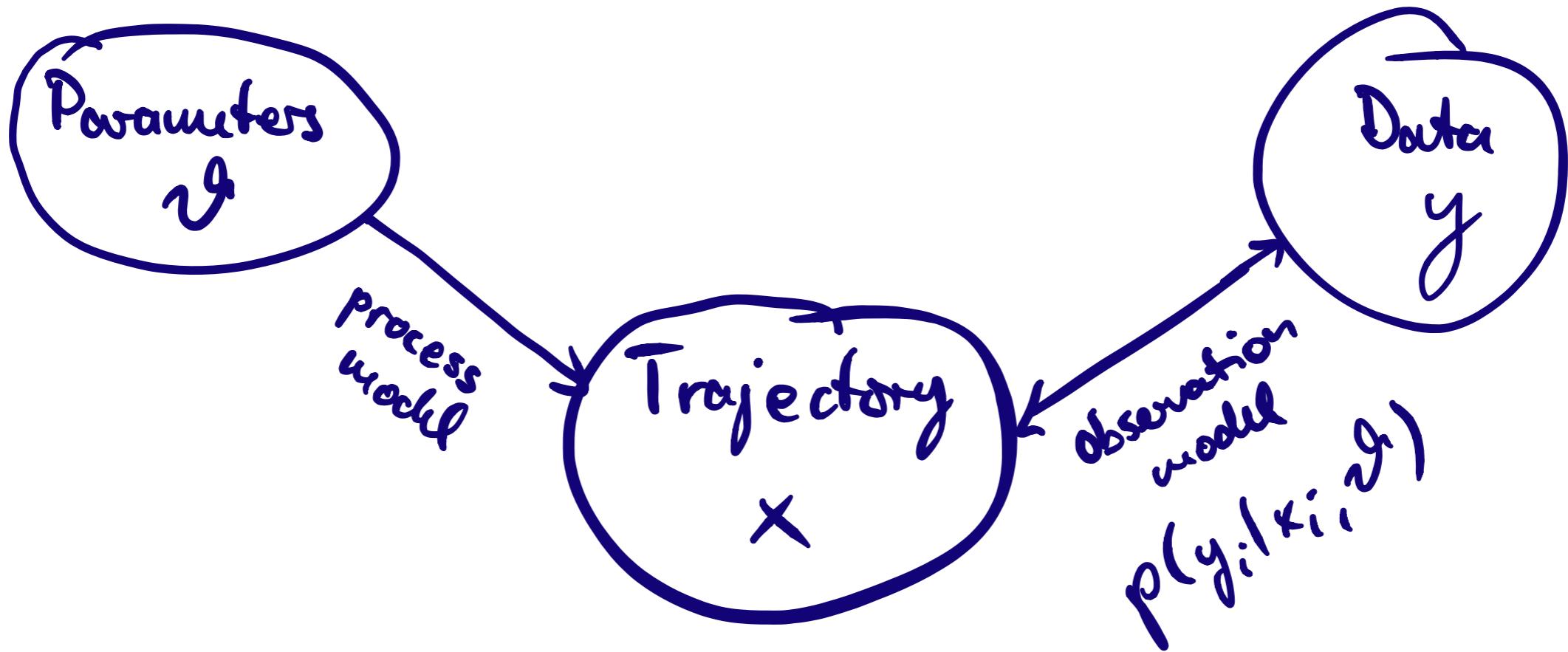
Sample from  $p(\theta | \text{Data})$ .

# Fitting deterministic vs. stochastic models

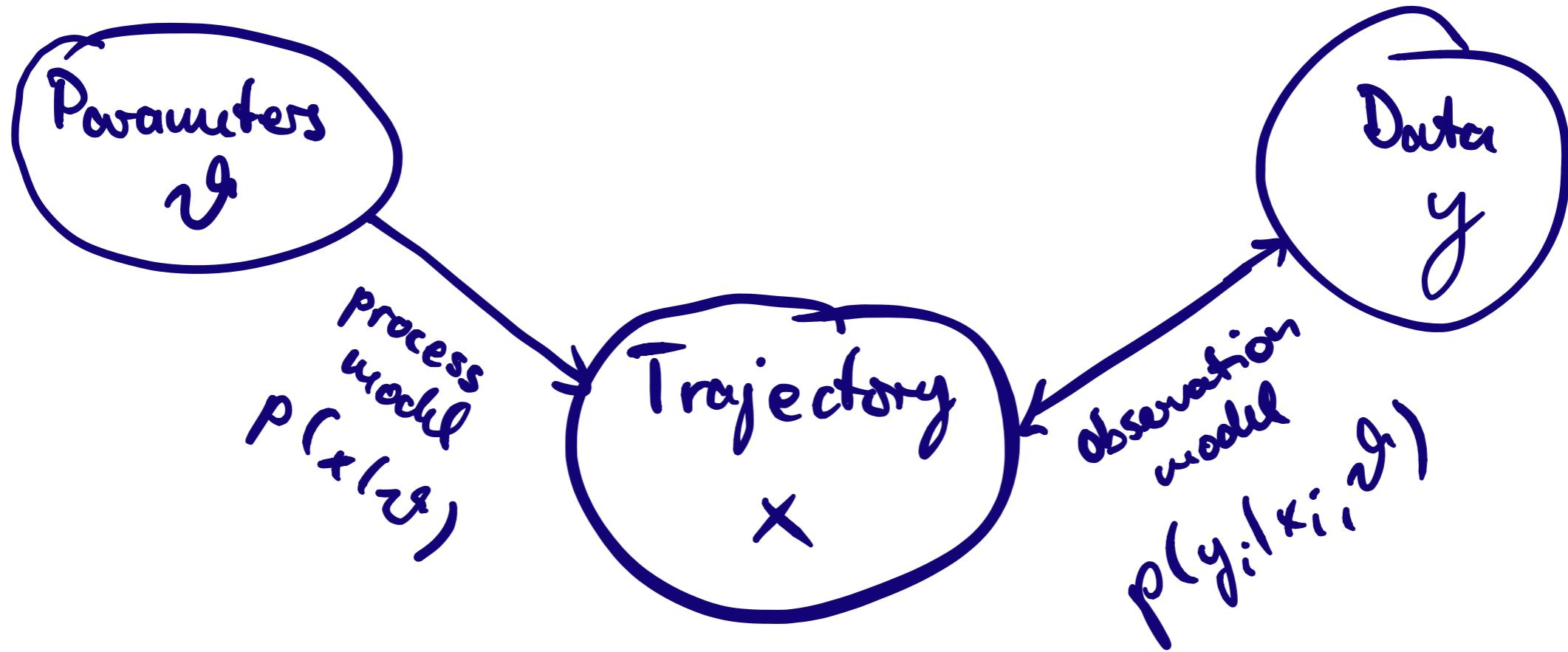


# Fitting deterministic vs. stochastic models





Can compute  $p(\text{Data}|\theta)$  for each  $\theta$ .



Cannot compute  $p(\text{Data}|\theta)$  for given  $\theta$ .

## Marginal likelihood

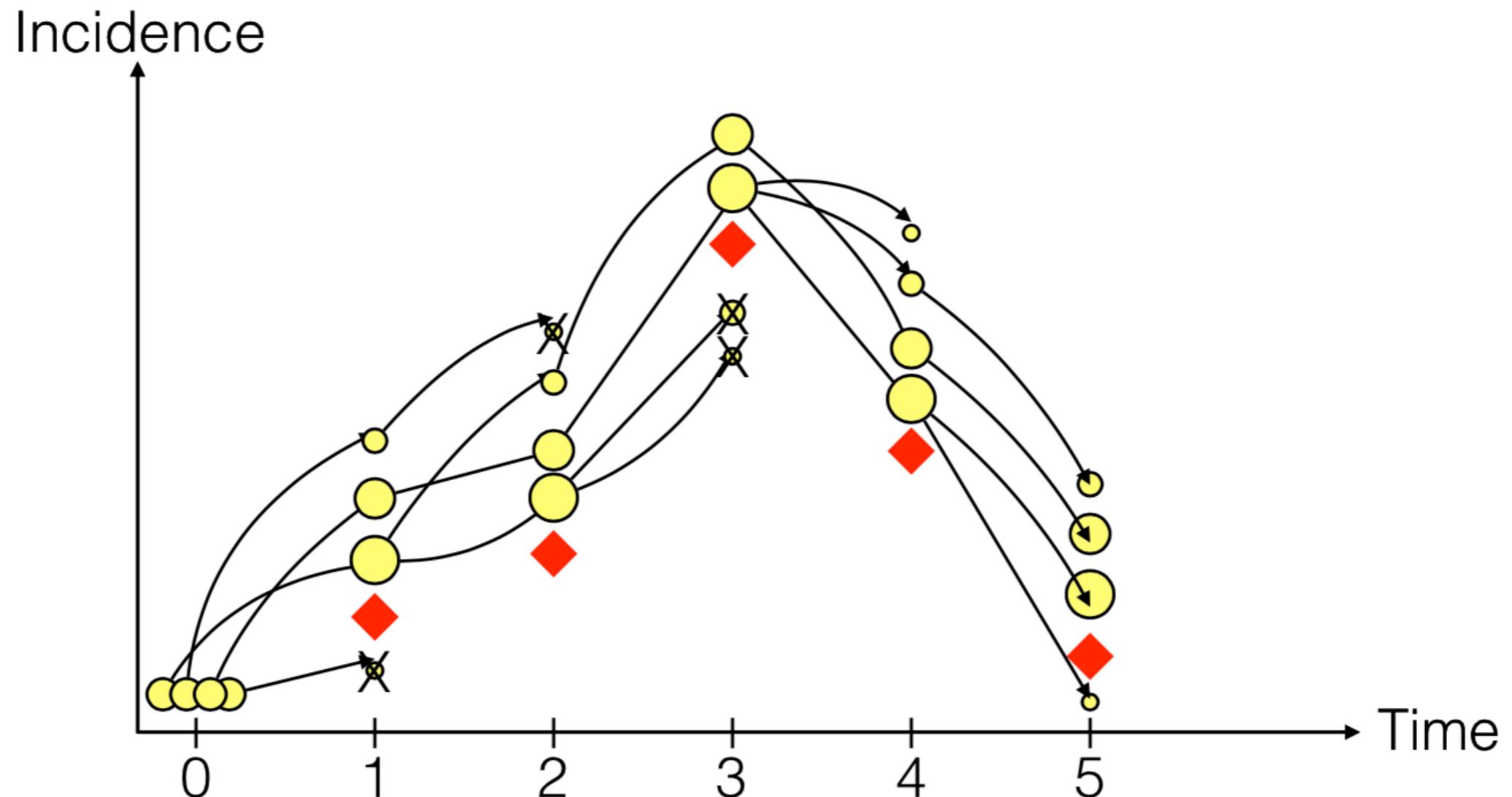
$$p(y|\theta) = \sum p(y|x, \theta)p(x|\theta)$$

$x$



All possible trajectories, given  $\theta$

# The particle filter

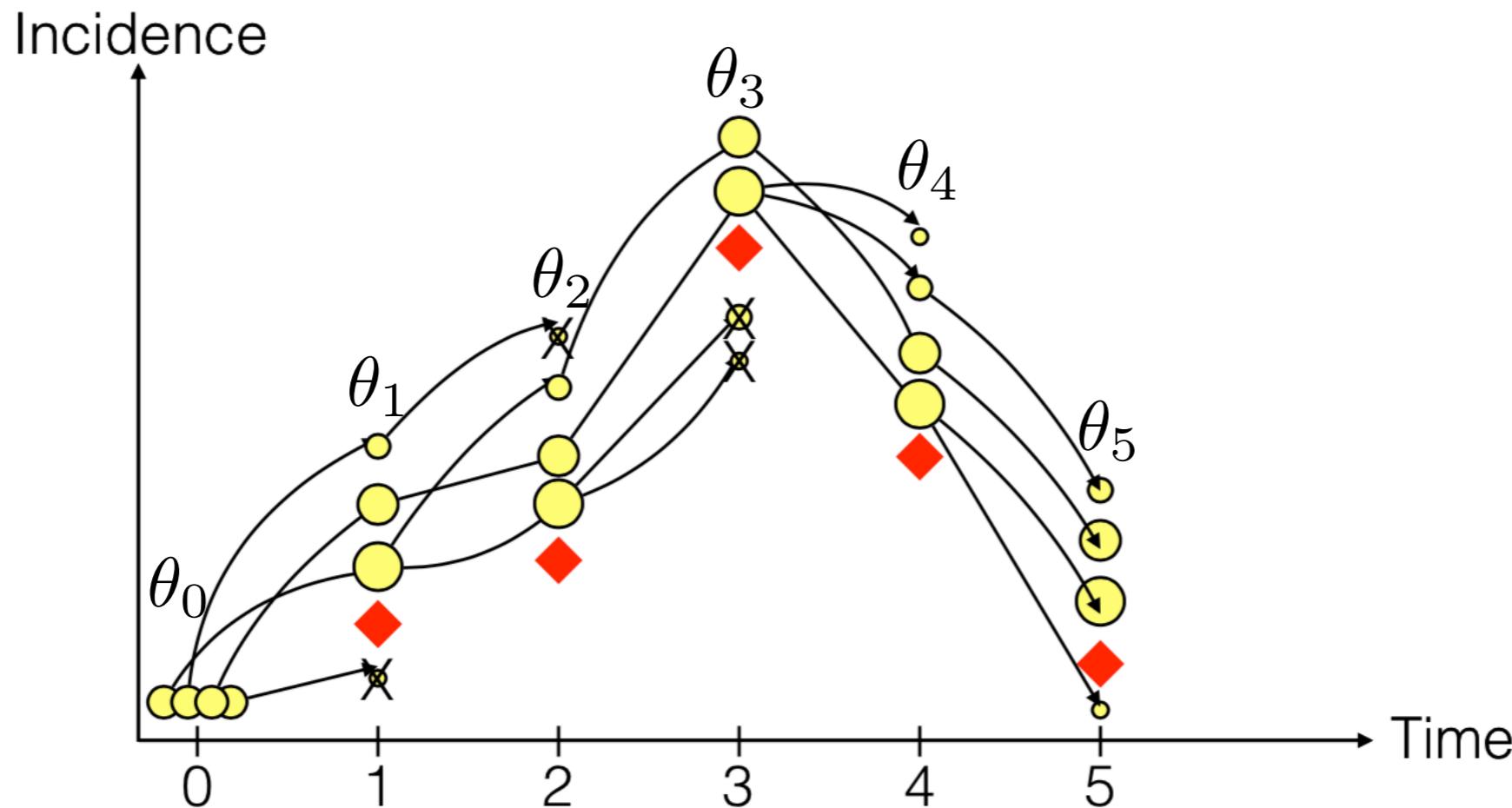


Estimate  $p(\text{Data}|\theta)$  for given  $\theta$ .

# Methods for model fitting

## 2. Stochastic models

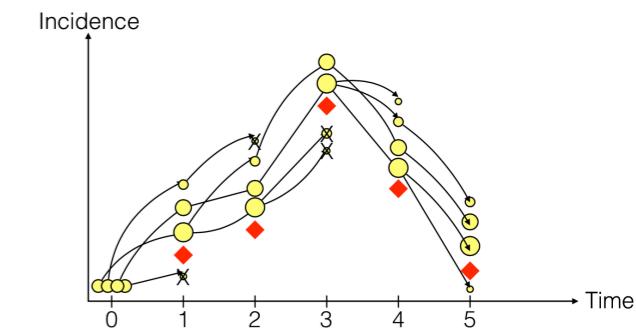
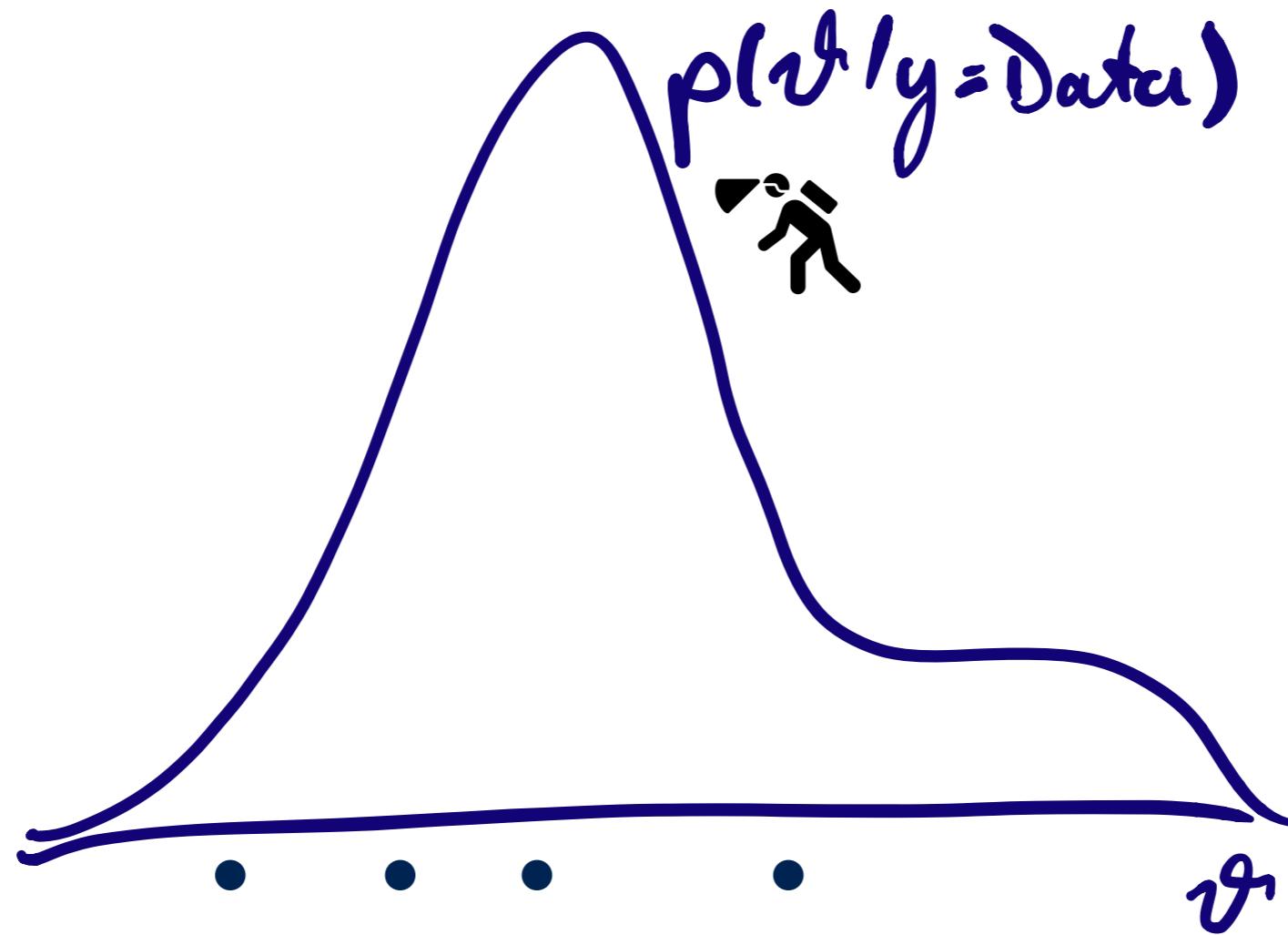
### 3. Maximum likelihood via iterated filtering (MIF)



- particle filter with random walk on  $\theta$
- intensity of random walk decreases at every iteration
- converges to maximum likelihood estimate of  $\theta$



## 4. Particle Markov-chain Monte Carlo (pMCMC)

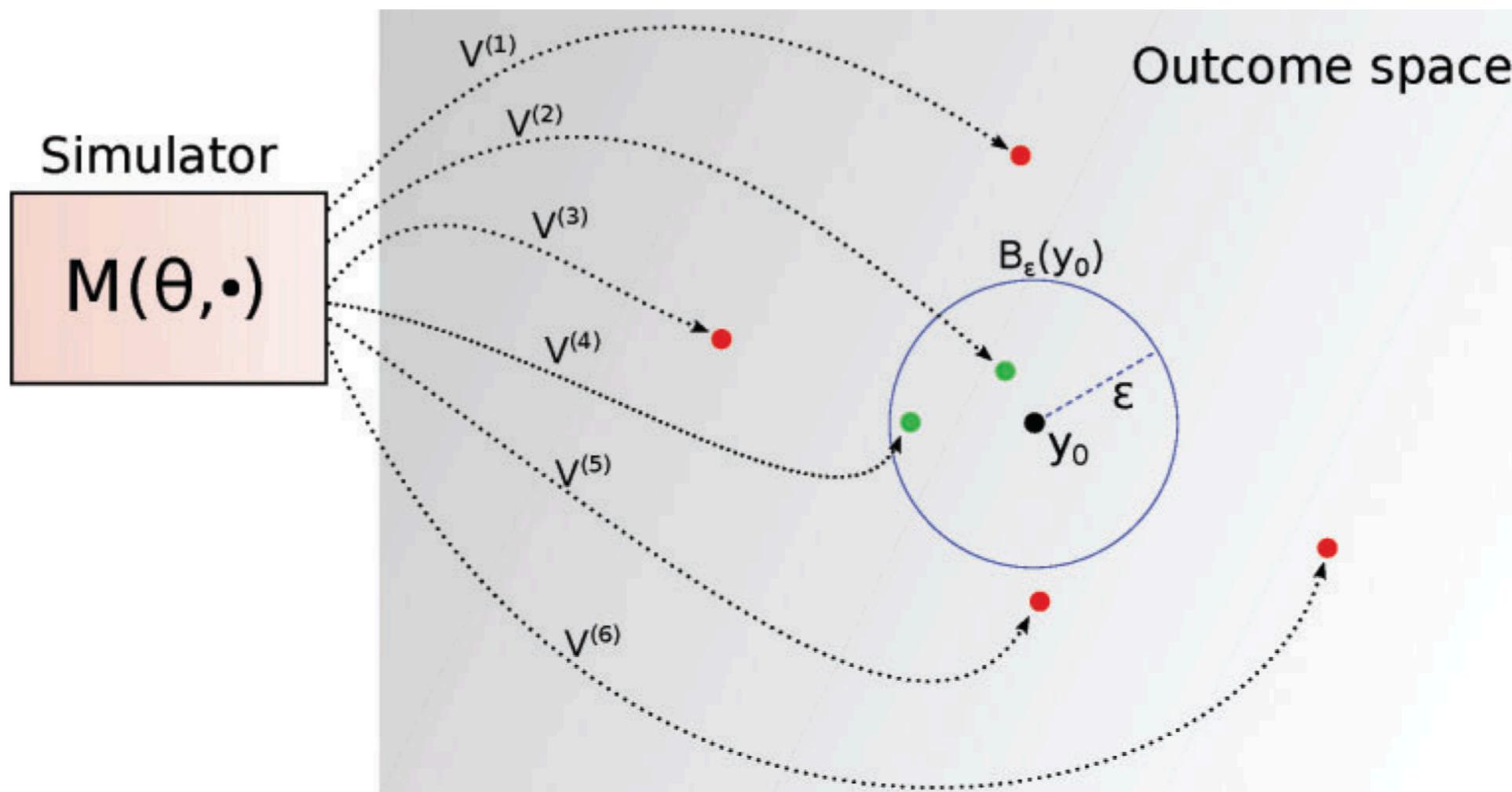


MCMC with  $p(\text{Data}|\theta)$  replaced with particle filter estimate



## 5. Approximate Bayesian Computation (ABC)

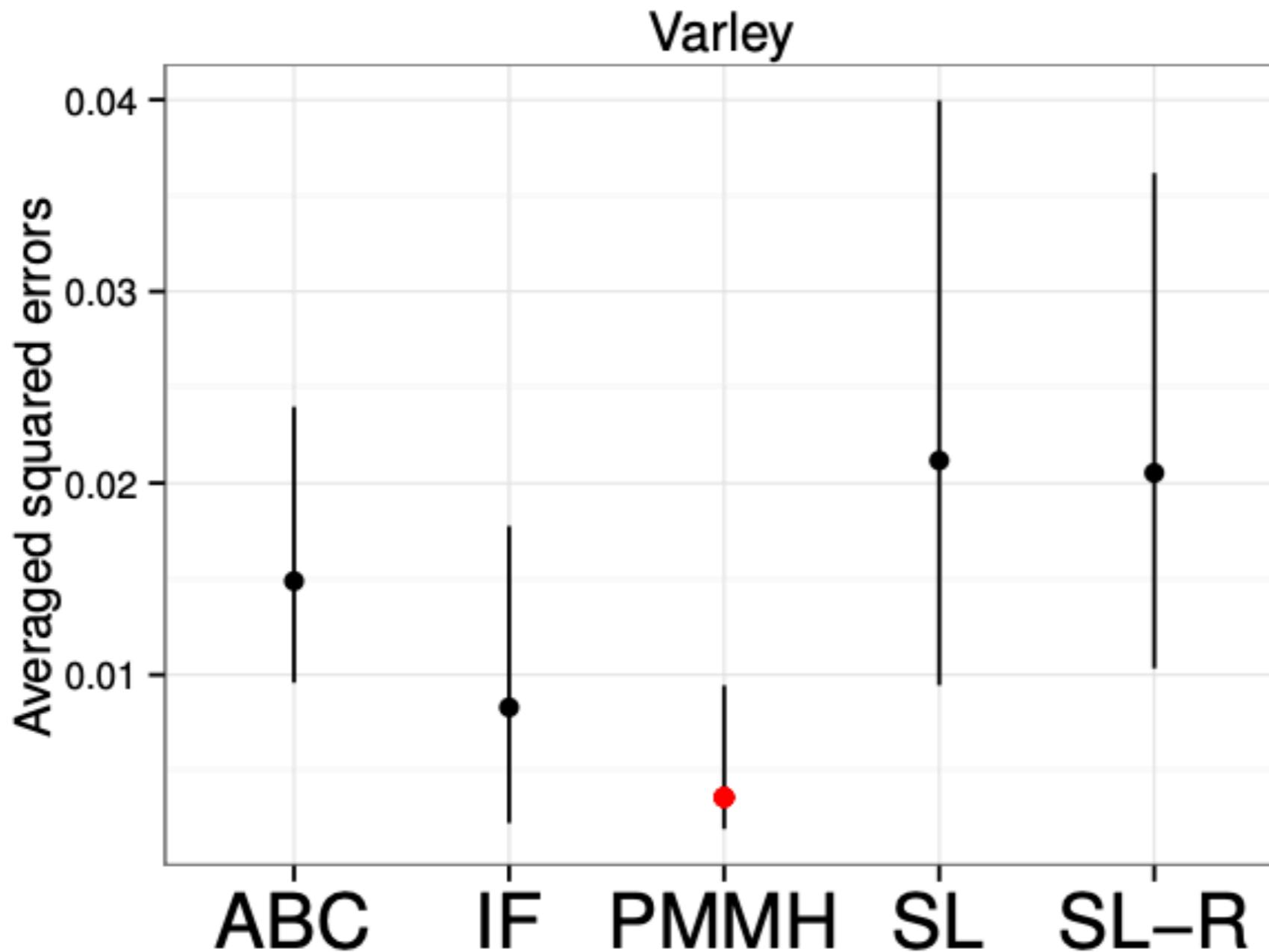
Lintusaari et al. (2016)



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)

**BIG MODELS**

**FOR**

**BIG DATA**

**?**

OPEN ACCESS

PEER-REVIEWED

RESEARCH ARTICLE

# Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda

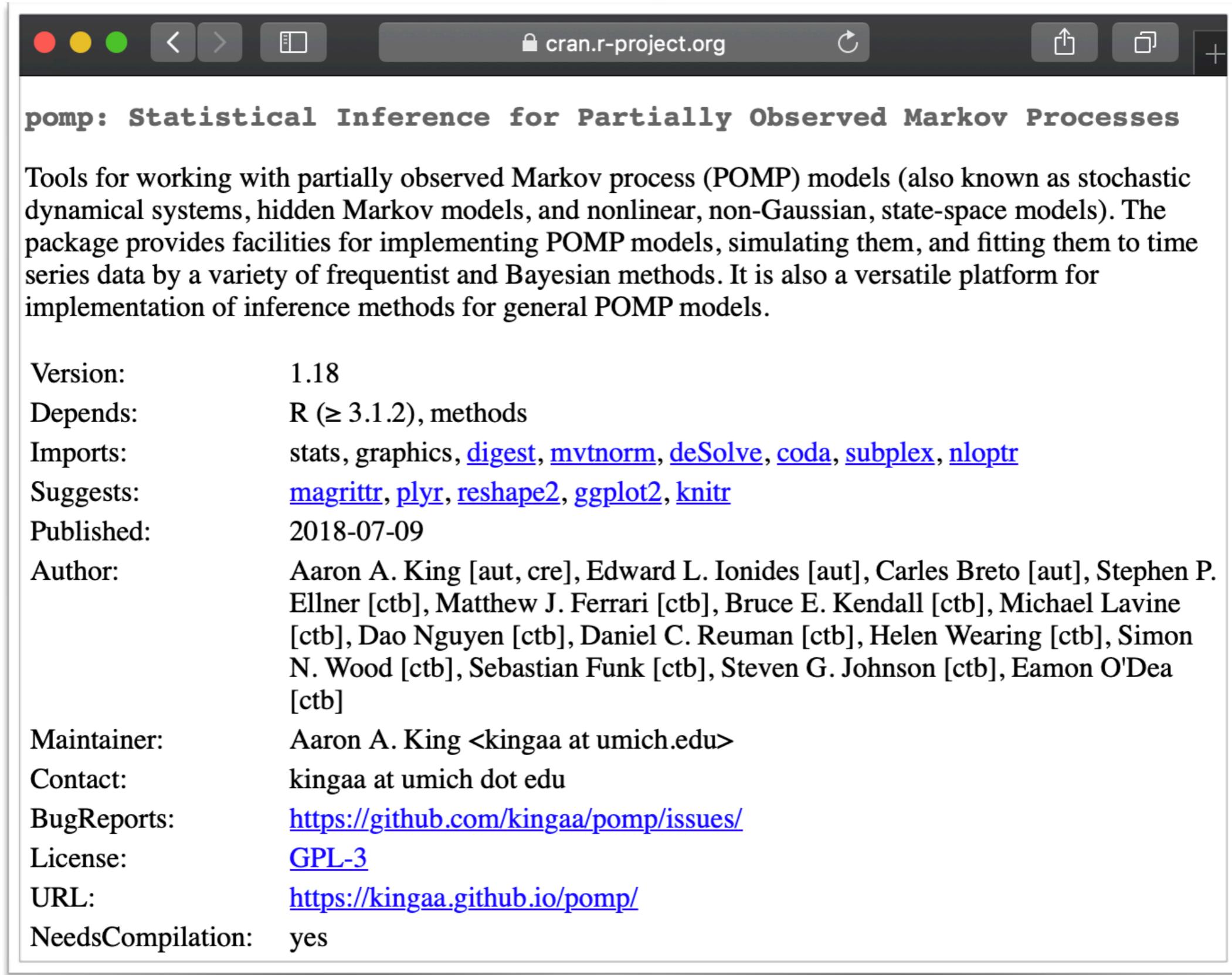
Ioannis Andrianakis , Ian R. Vernon, Nicky McCreesh, Trevelyan J. McKinley, Jeremy E. Oakley, Rebecca N. Nsubuga, Michael Goldstein, Richard G. White

Published: January 8, 2015 • <https://doi.org/10.1371/journal.pcbi.1003968>

Andrianakis et al., 2015

# **Computational efficiency vs Coding efficiency**

# pomp



The screenshot shows a web browser window displaying the CRAN package page for 'pomp'. The title bar reads 'cran.r-project.org'. The main content area has a dark header with the package name 'pomp' and its subtitle 'Statistical Inference for Partially Observed Markov Processes'. Below this, a large paragraph describes the package's purpose: '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.' A table below lists various 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
```

# 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
```

# pomp

The screenshot shows a web browser window with a dark grey header bar. On the left of the header are standard window control icons (red, yellow, green circles, and arrows). In the center is the URL 'kingaa.github.io'. On the right of the URL are a refresh icon and a search bar placeholder 'Search...'. The main content area has a white background and a dark grey border. At the top of this area, the title 'User guides and tutorials' is displayed in a large, bold, black font. Below the title is a list of items, each consisting of a title in black text and a pair of blue hyperlinks in parentheses to its right. The items are:

- Statistical Inference for Partially Observed Markov Processes via the R Package **pomp** (a *Journal of Statistical Software* paper)  
(PDF) (R)
- Getting started with **pomp**  
(HTML) (R)
- pomp** 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 **pomp** project  
(HTML)
- pomp2** package manual  
(PDF)
- pomp** package manual  
(PDF)
- Frequently Asked Questions  
(HTML)
- More **pomp** 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 "R" and "x" are in red, while the other letters are in black. Below it, the text "THE PREPRINT SERVER FOR BIOLOGY" is written in a smaller, all-caps sans-serif font.

HOME | AB

Search

New Results

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

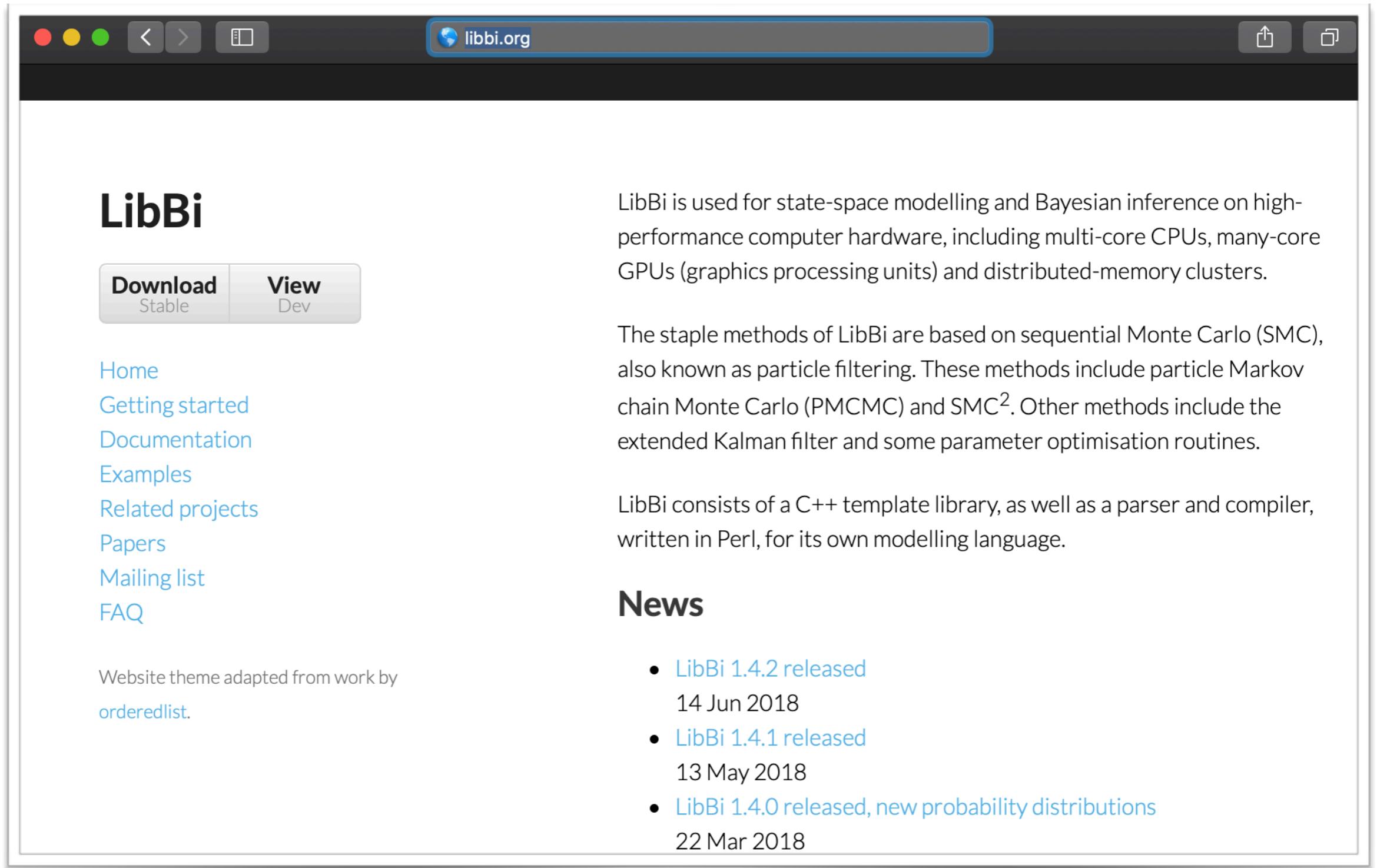
 Theresa Stocks, Tom Britton,  Michael Höhle

**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 about LibBi's use cases. Further down, there are sections for "News" (listing recent releases) and a sidebar with links to "Home", "Getting started", "Documentation", "Examples", "Related projects", "Papers", "Mailing list", and "FAQ". At the bottom left, there is a note about 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.

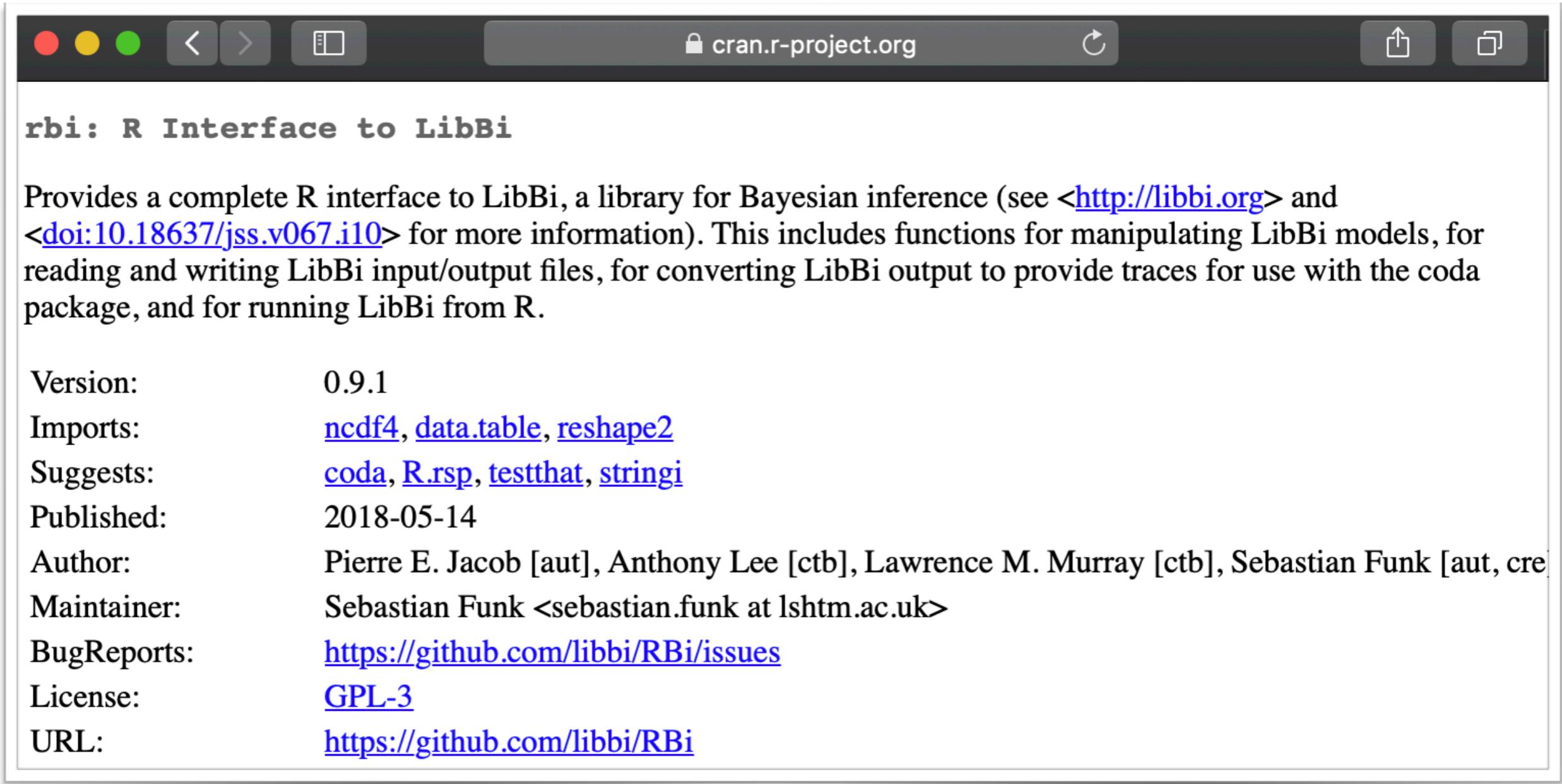
LibBi consists of a C++ template library, as well as a parser and compiler, written in Perl, for its own modelling language.

## 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)
    }

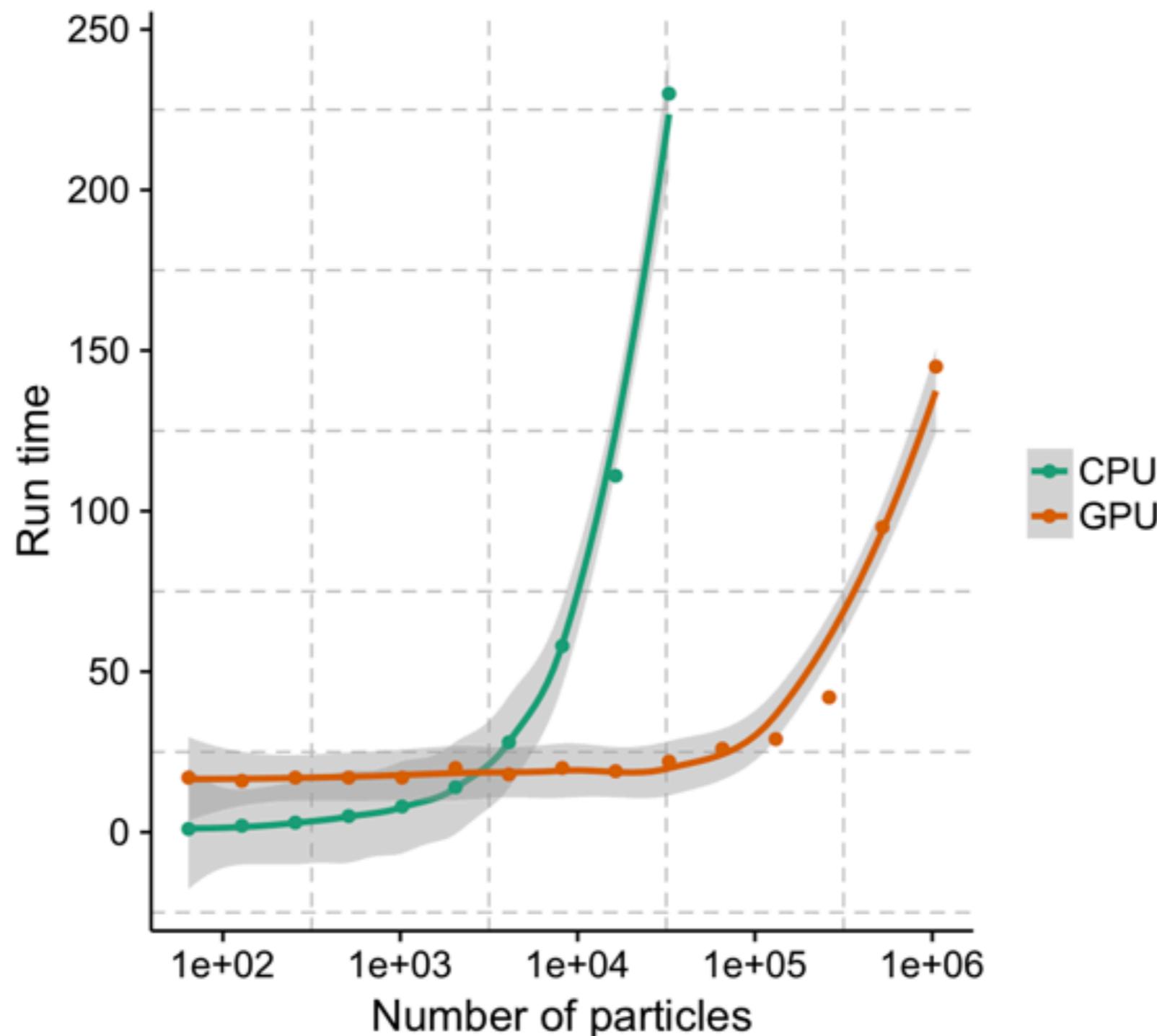
    sub initial {
        S <- N - 1
        I <- 1
        R <- 0
    }

    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**

1. Frequentist vs Bayesian
2. Deterministic vs stochastic
3. Computational vs statistical efficiency
4. Computational vs coding efficiency