

How to choose the right inference approach for your problem (or: the answer to all your questions)

Model fitting and inference for infectious disease dynamics

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



How to choose the right inference approach for your problem (NOT the answer to all your questions)

Model fitting and inference for infectious disease dynamics

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

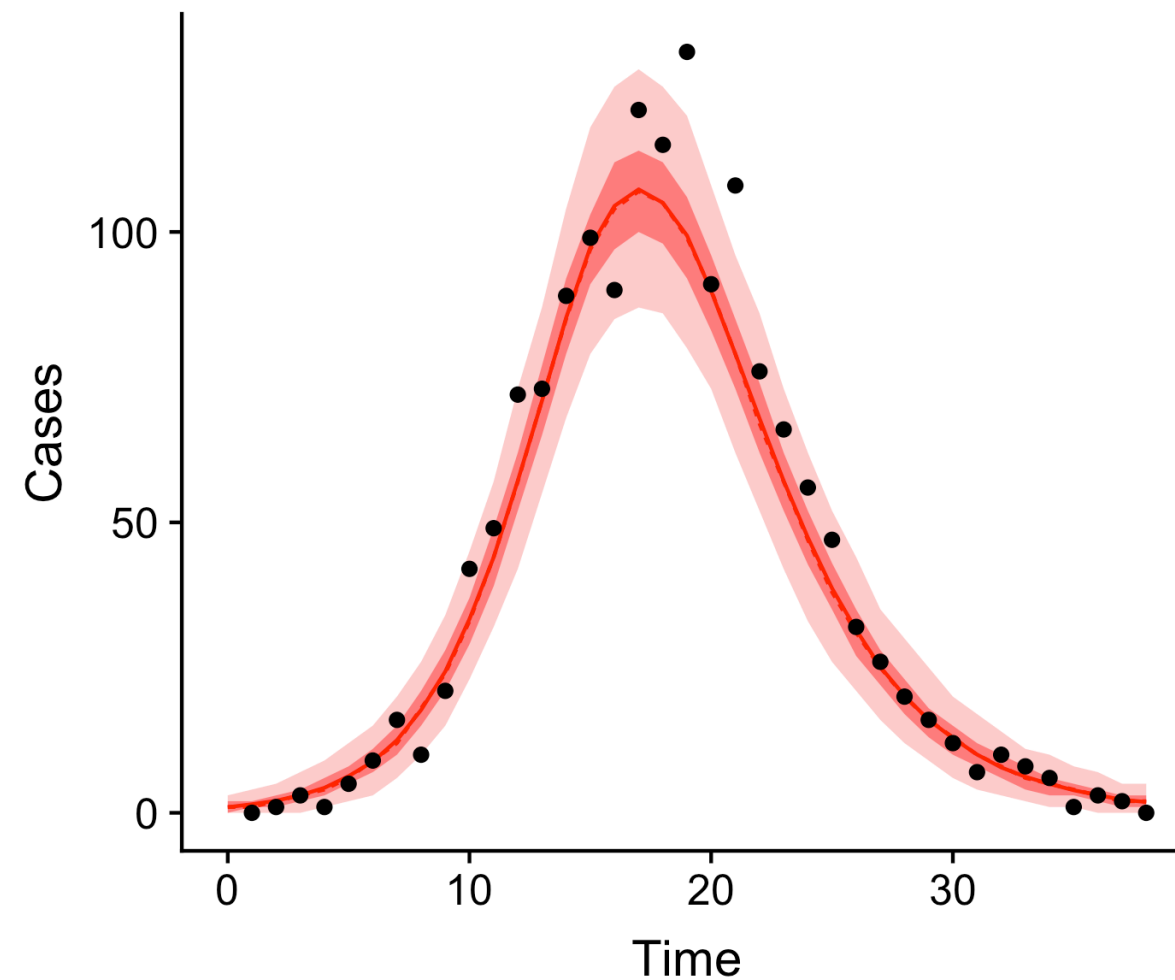


centre *for the*
mathematical
modelling *of*
infectious diseases

LONDON
SCHOOL *of*
HYGIENE
& TROPICAL
MEDICINE



Simulation-based inference

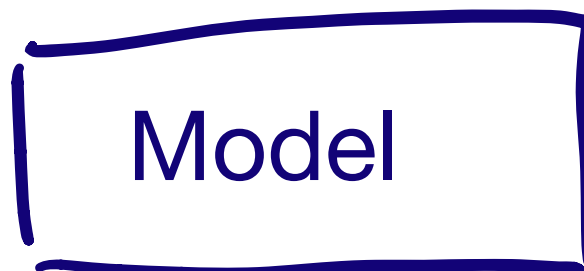


Combine mechanistic and statistical approach for

- Parameter estimation
- Prediction

Inputs

x

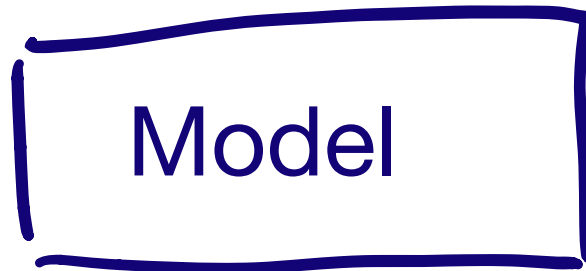


Outputs

y

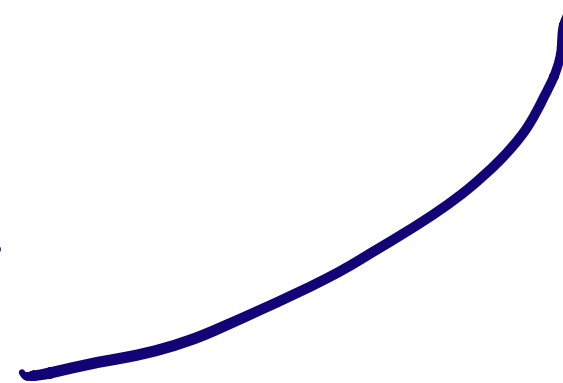
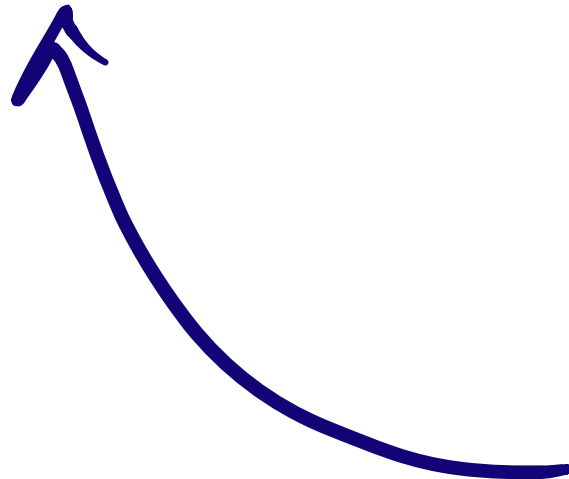
Inputs

θ



Data

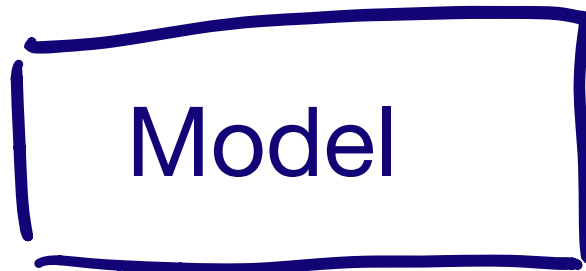
y



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

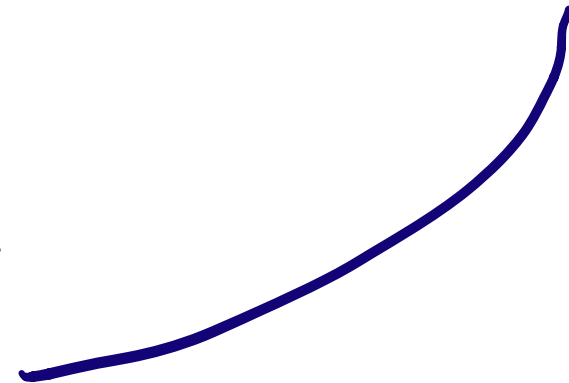
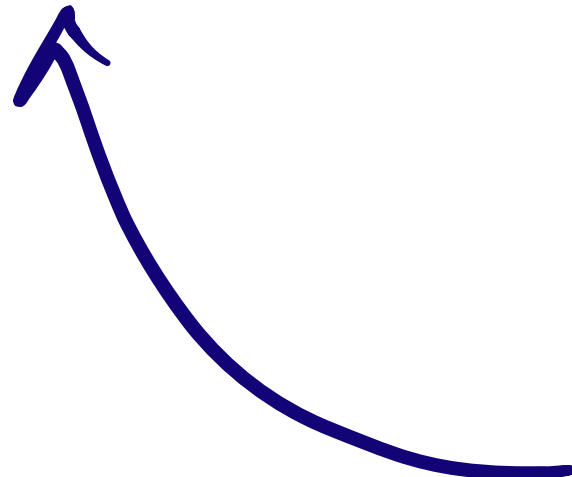
Inputs

θ



Data

y

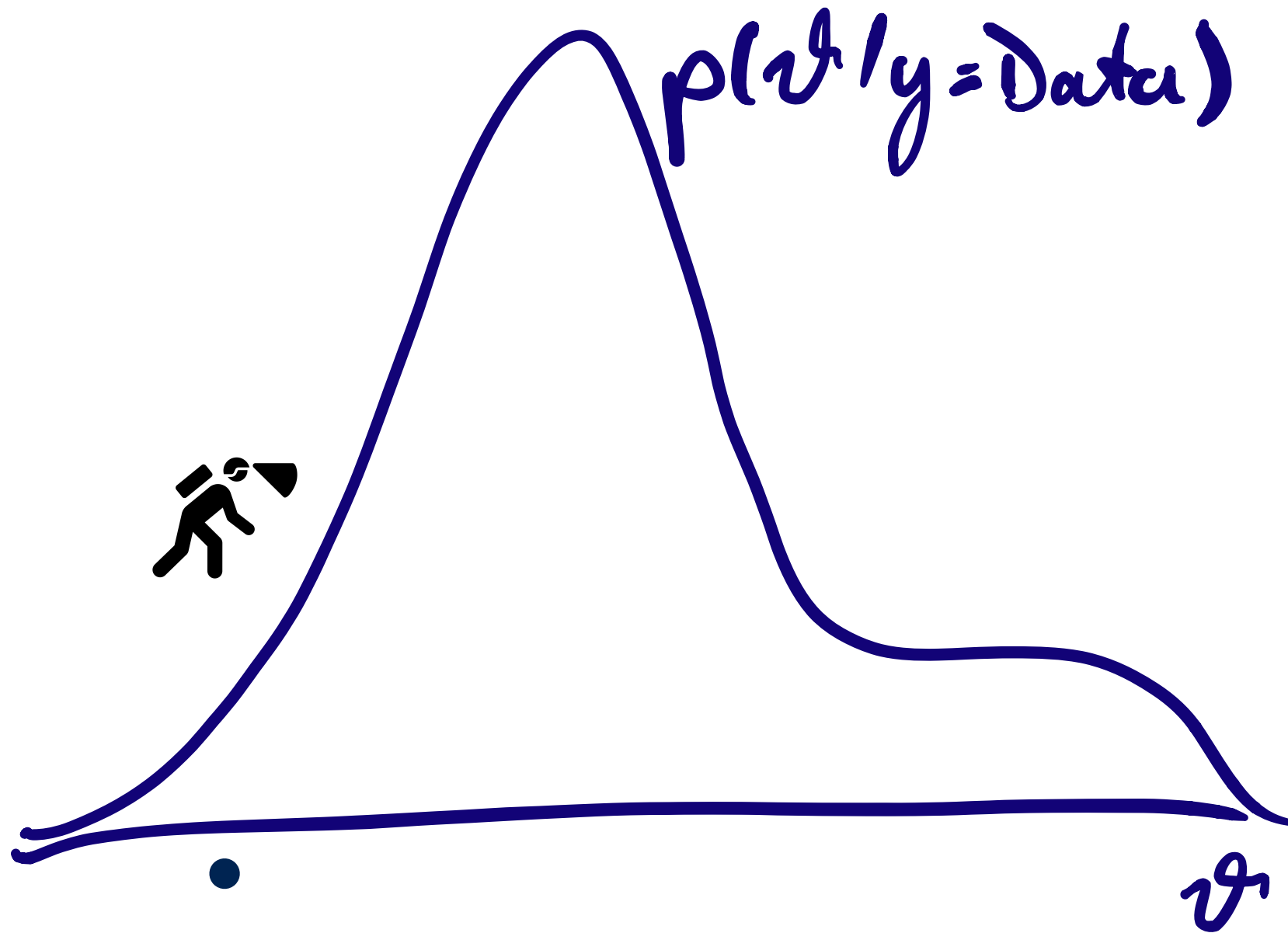


Maximum likelihood, MCMC,
MIF, pMCMC, ABC, ...



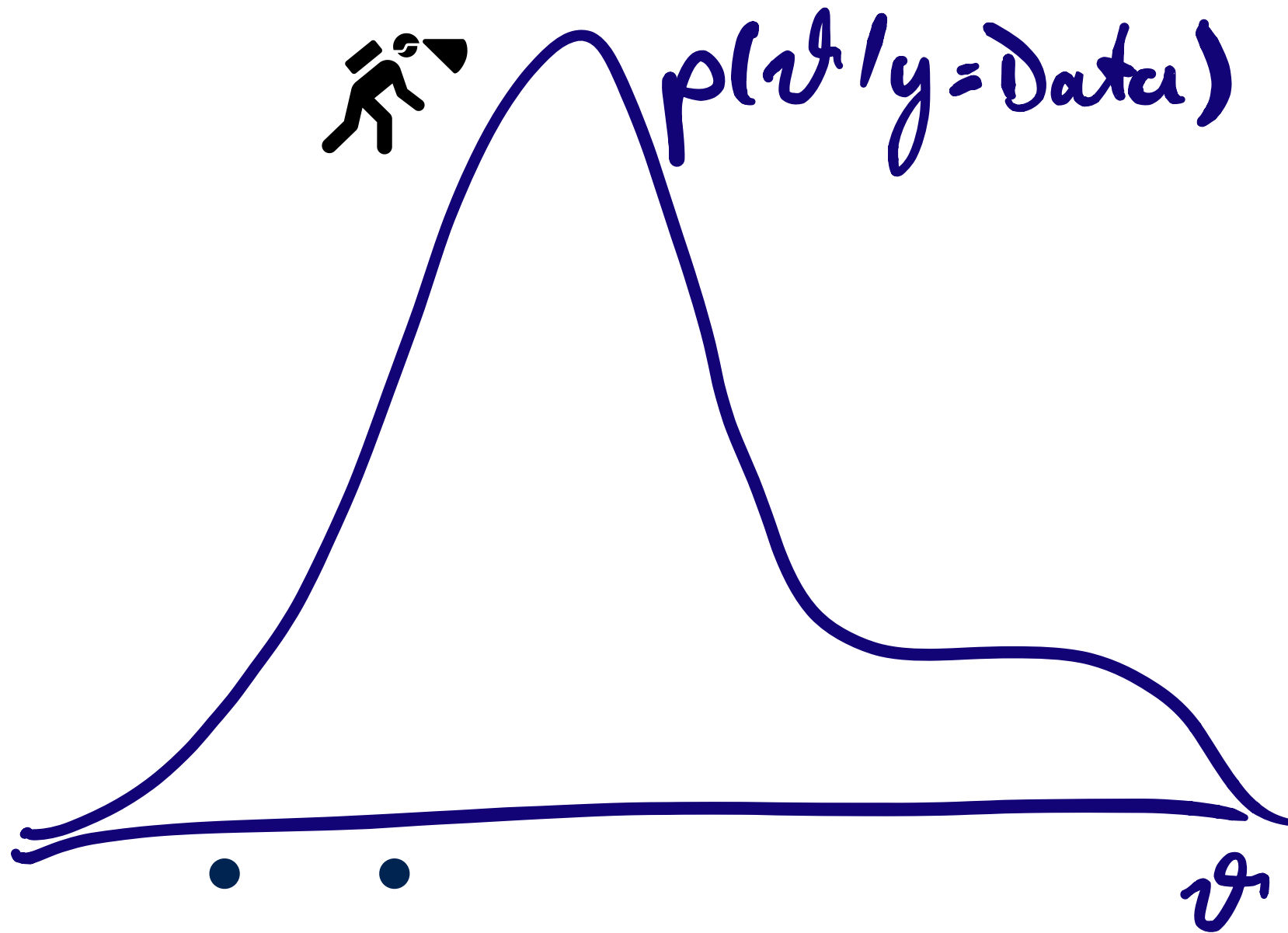
pomp, LibBi, Turing.jl, ...

Markov-chain Monte Carlo



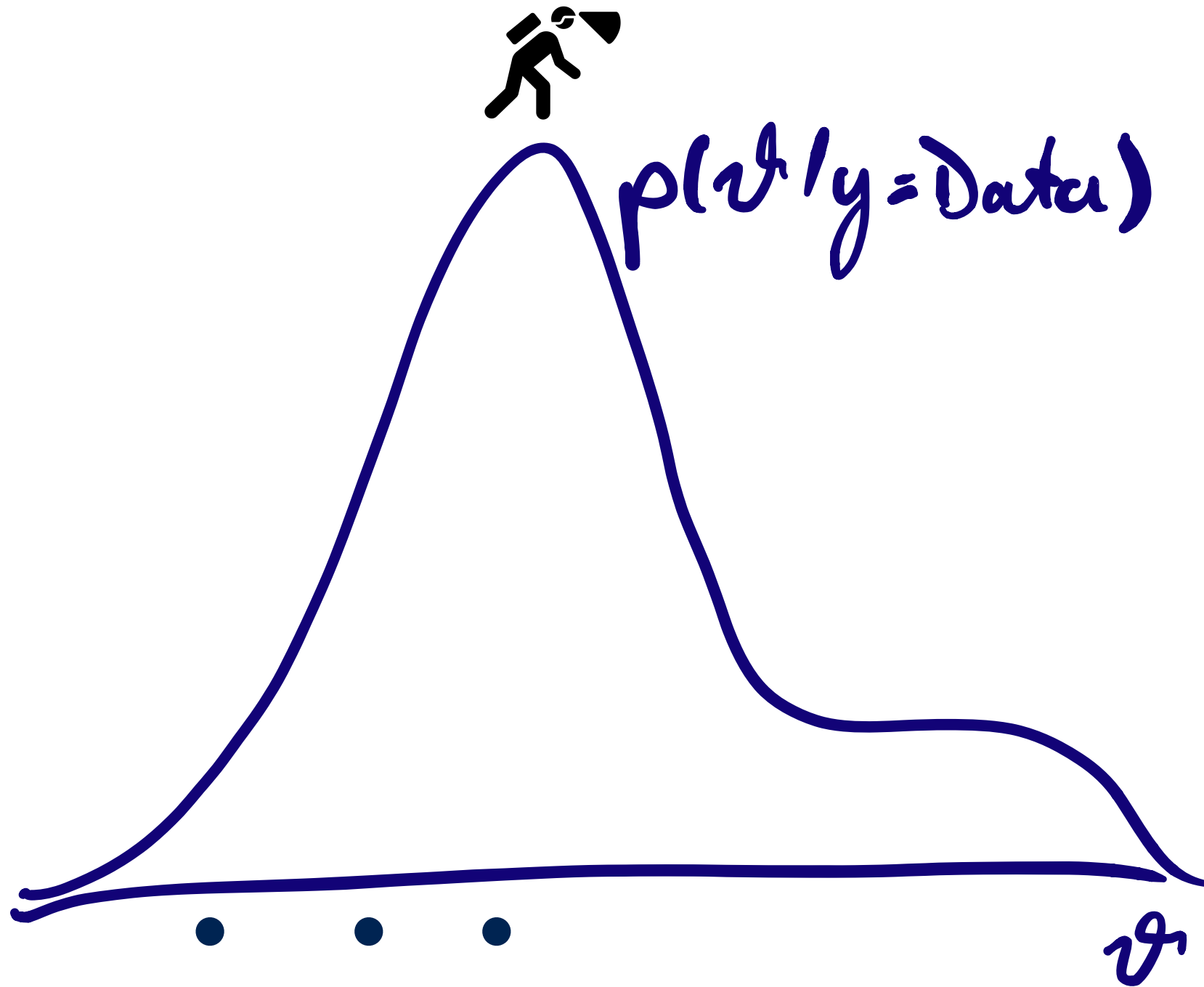
Sample from $p(\theta | \text{Data})$.

Markov-chain Monte Carlo



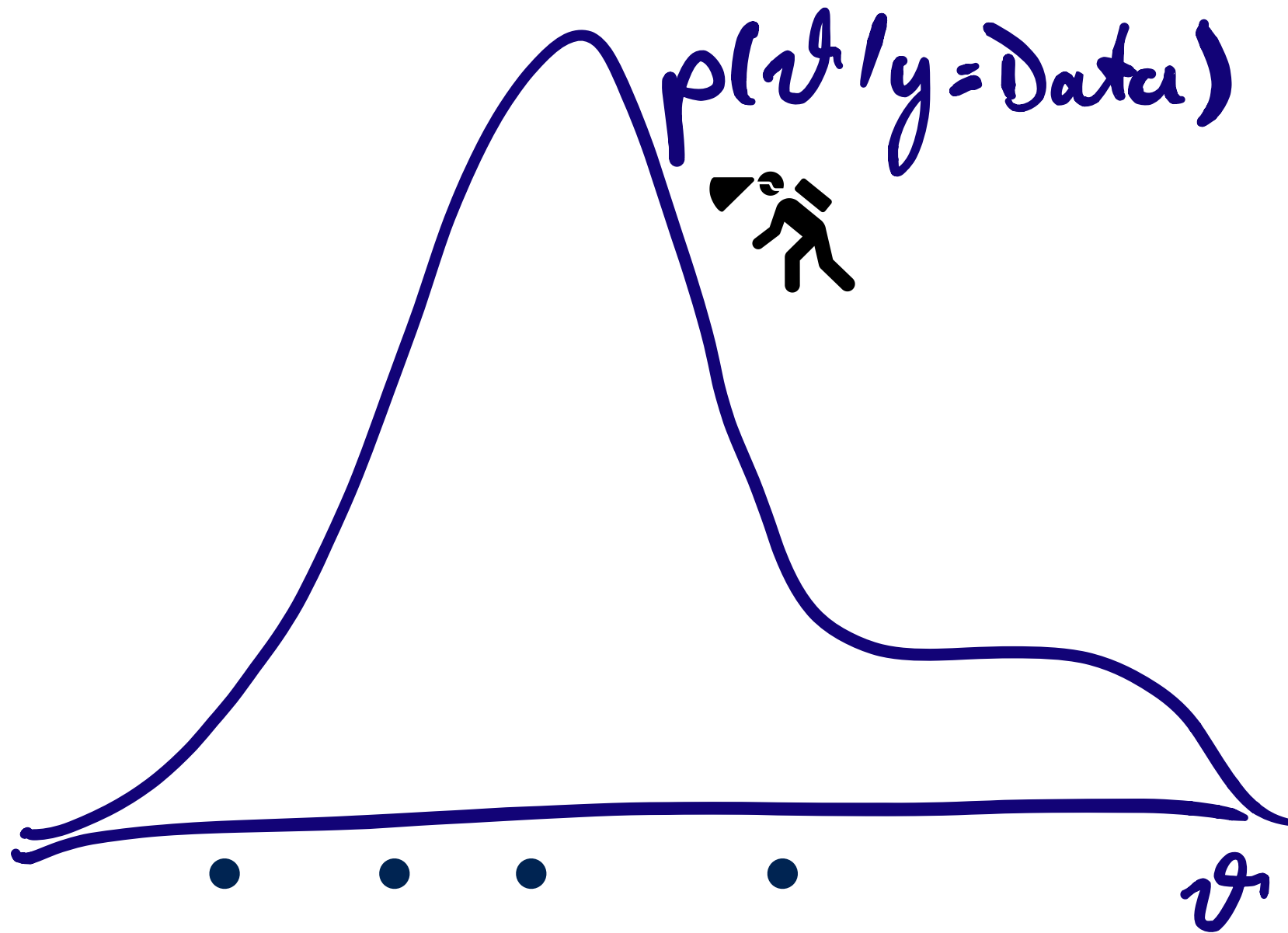
Sample from $p(\theta | \text{Data})$.

Markov-chain Monte Carlo



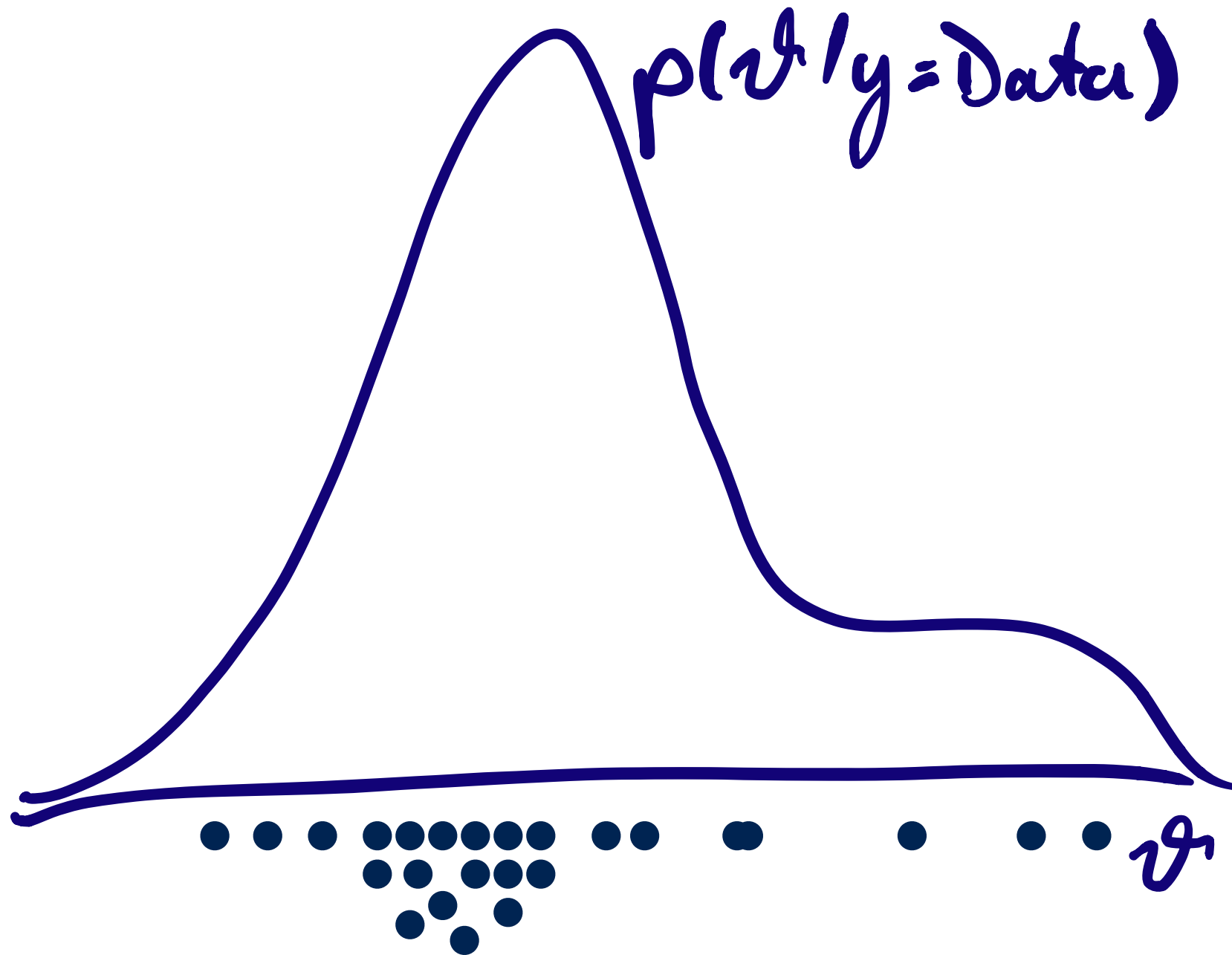
Sample from $p(\theta | \text{Data})$.

Markov-chain Monte Carlo



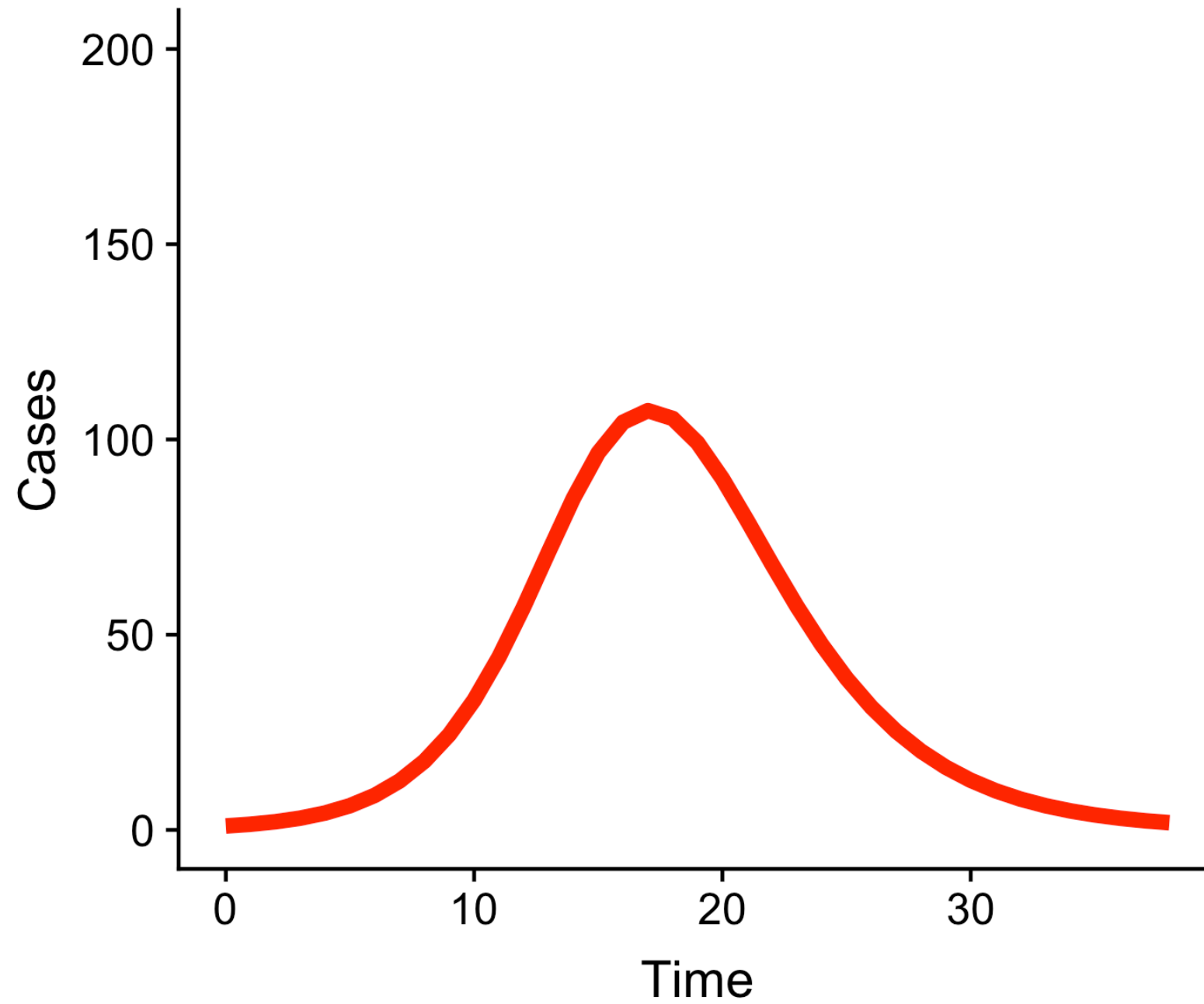
Sample from $p(\theta | \text{Data})$.

Markov-chain Monte Carlo

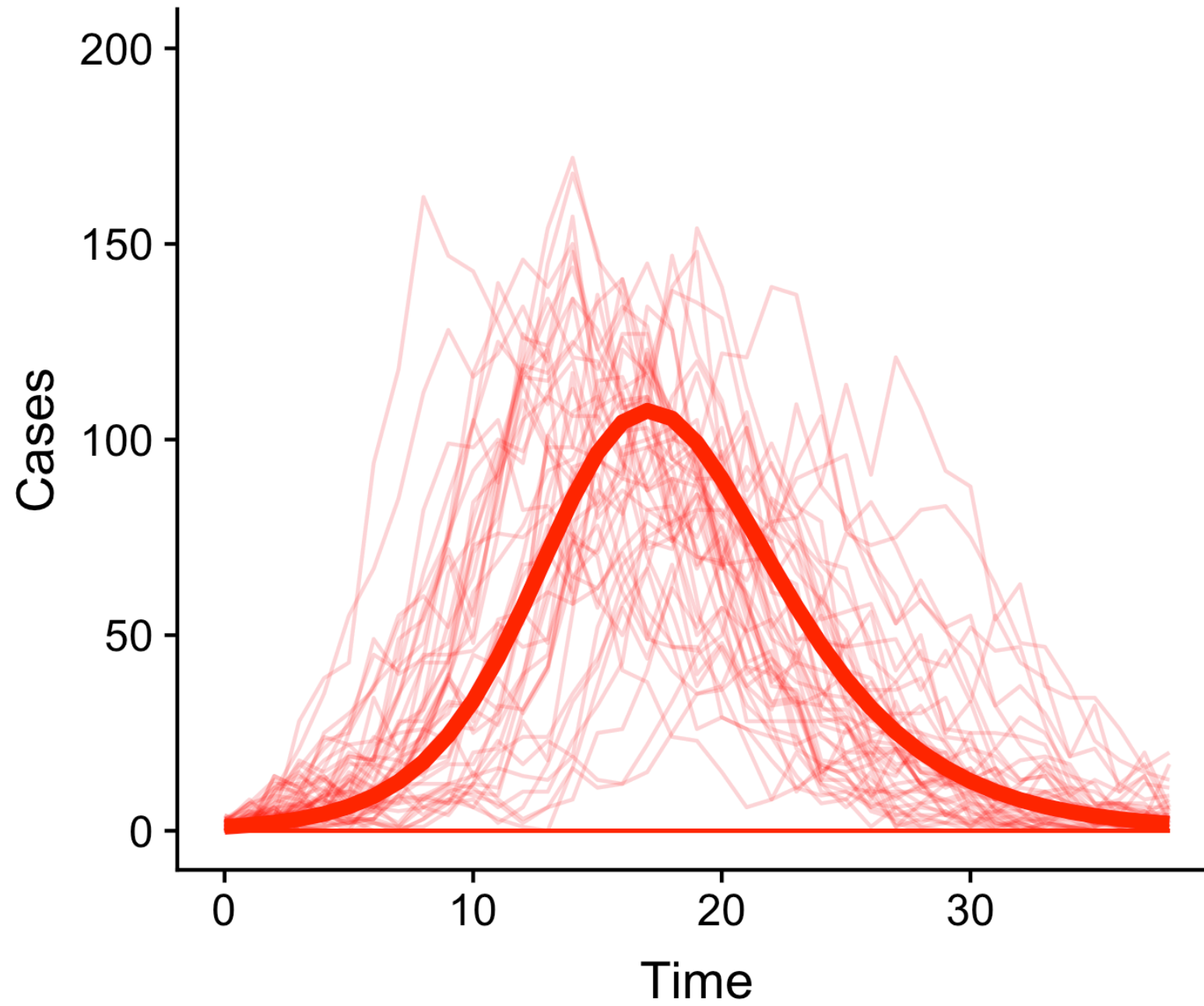


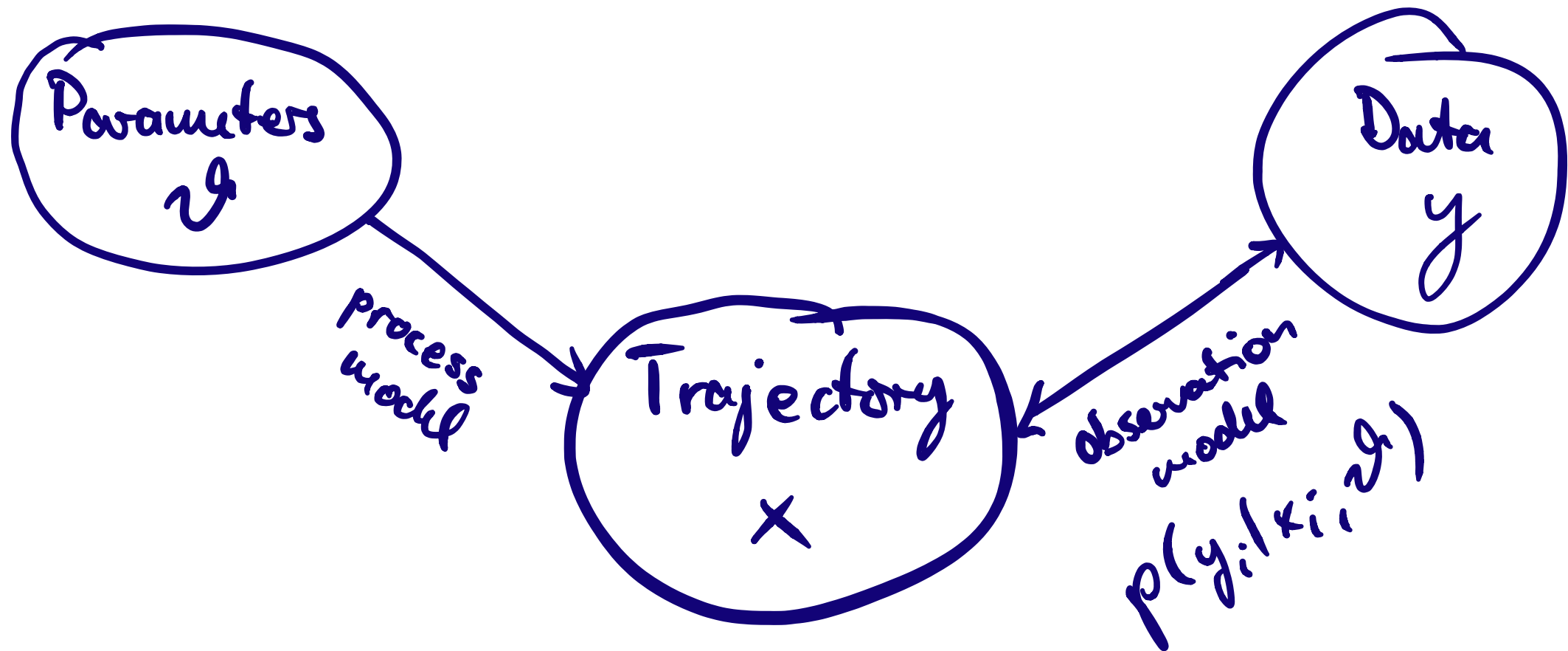
Sample from $p(\theta | \text{Data})$.

Fitting deterministic vs. stochastic models



Fitting deterministic vs. stochastic models

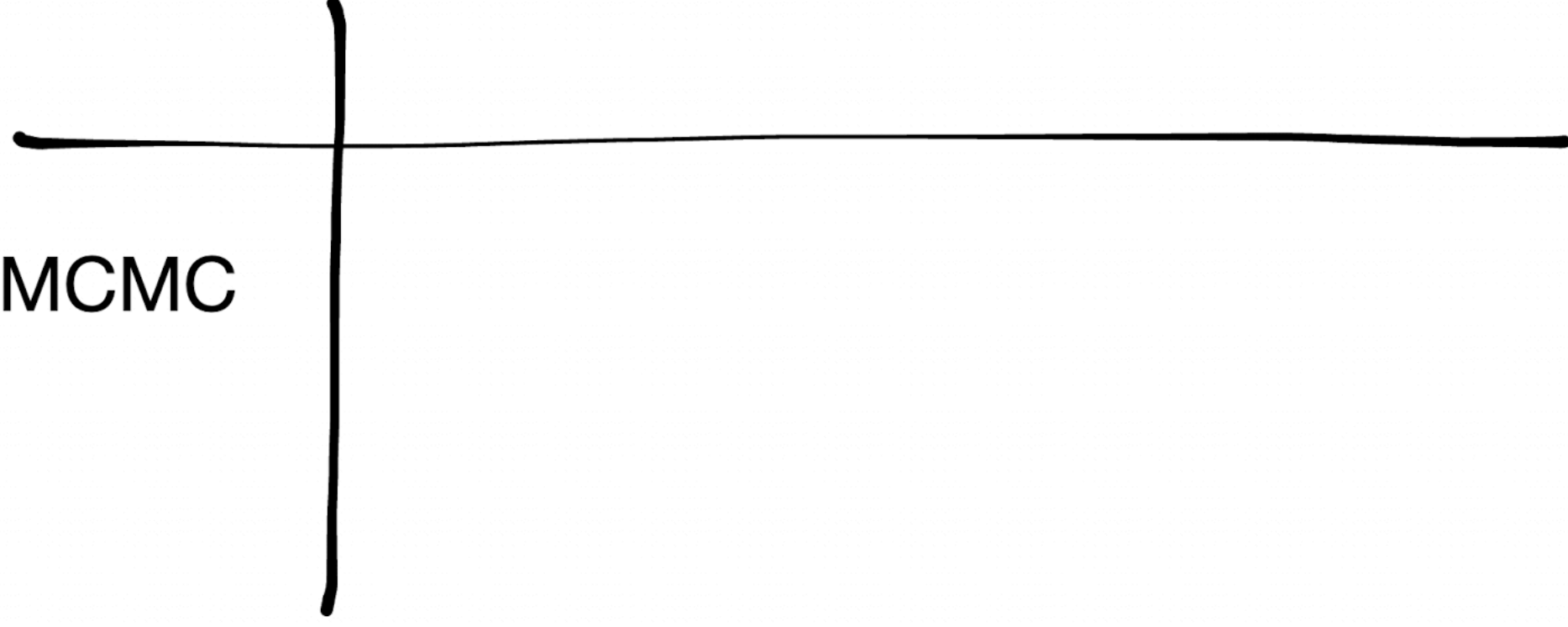


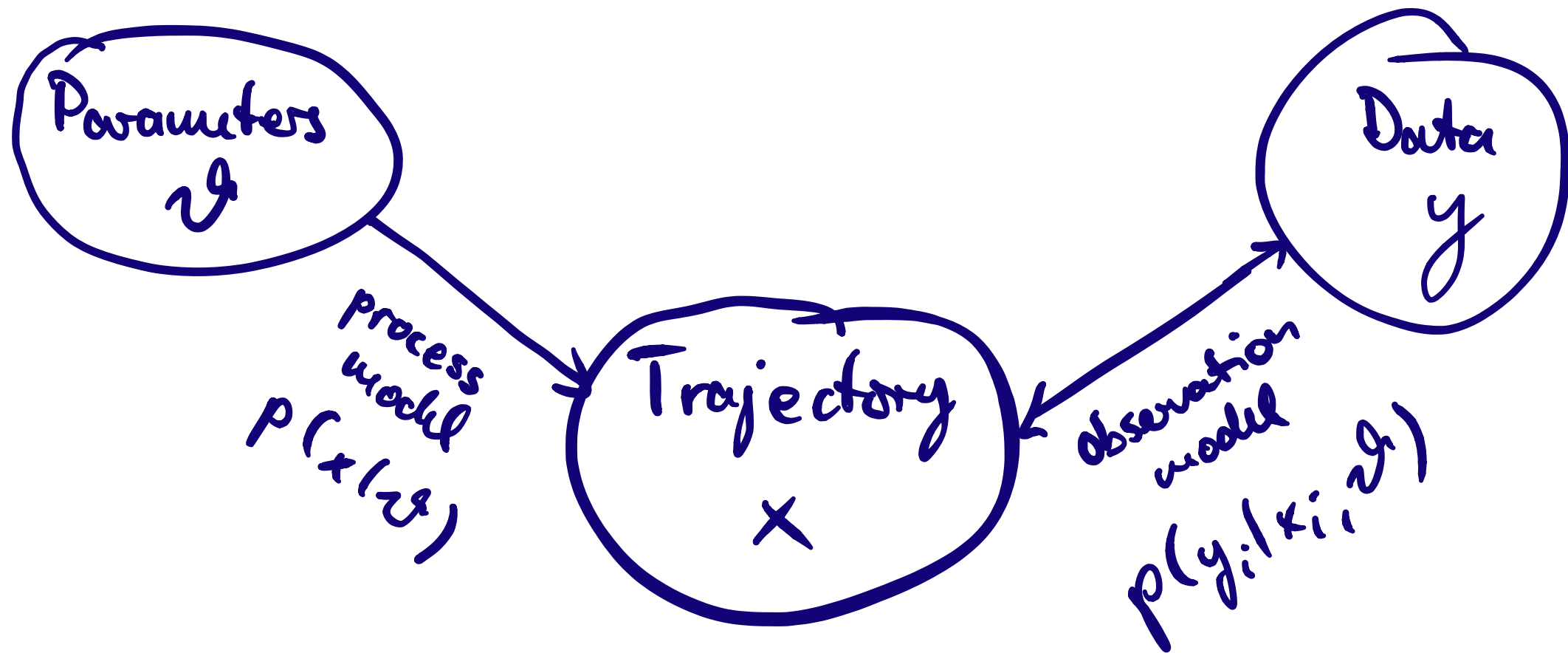


Can compute $p(y|\theta)$ for each θ .

Sampling from the posterior

MCMC





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

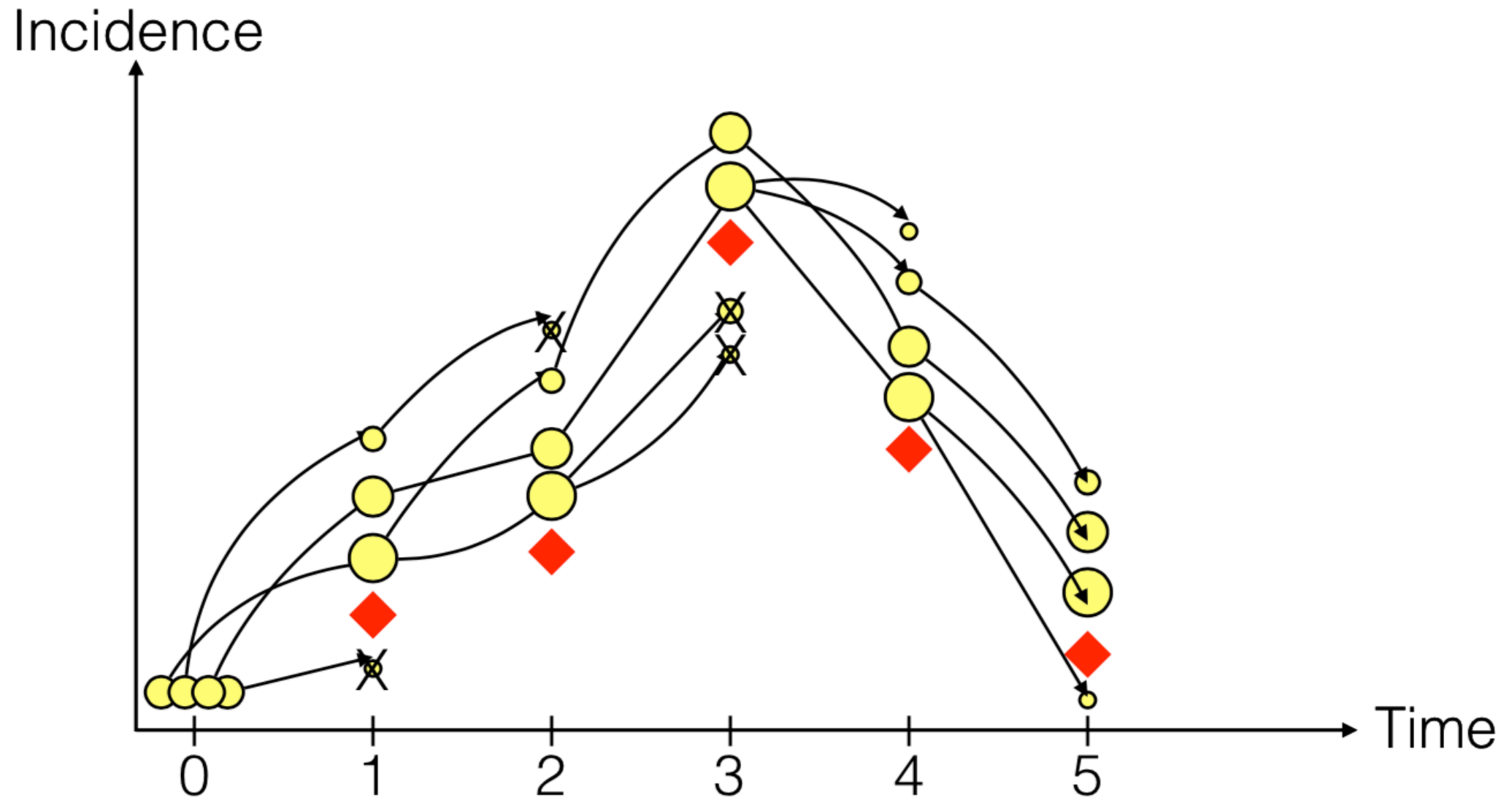
Marginal likelihood

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



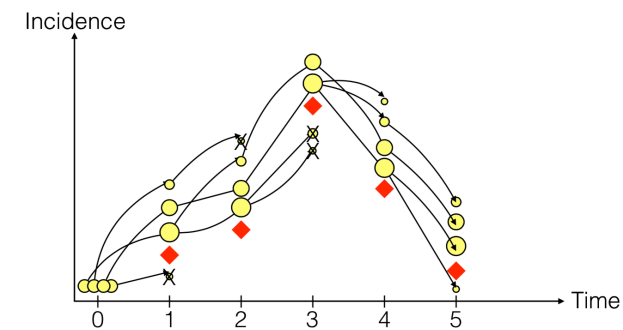
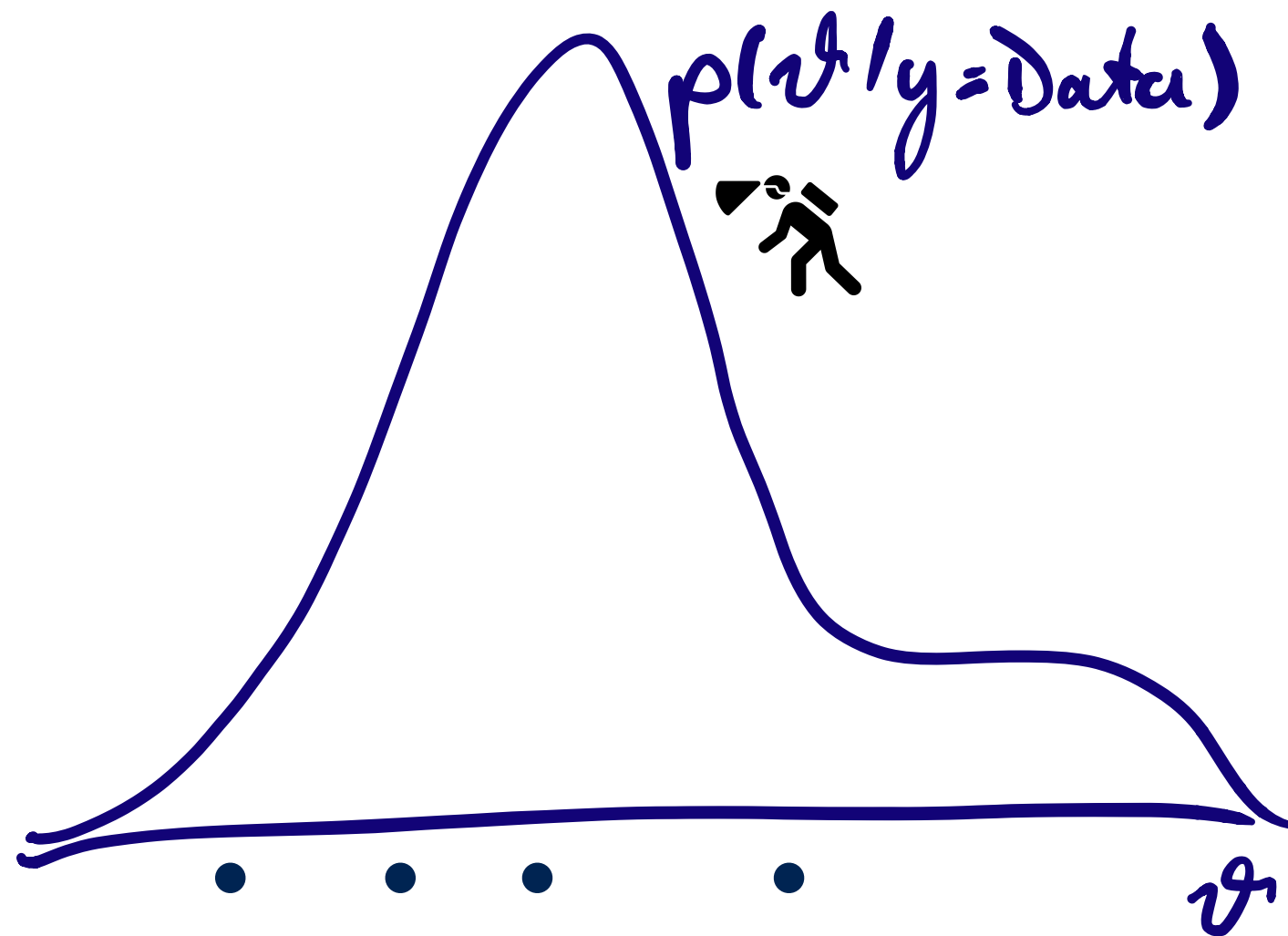
All possible trajectories, given θ

The particle filter



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

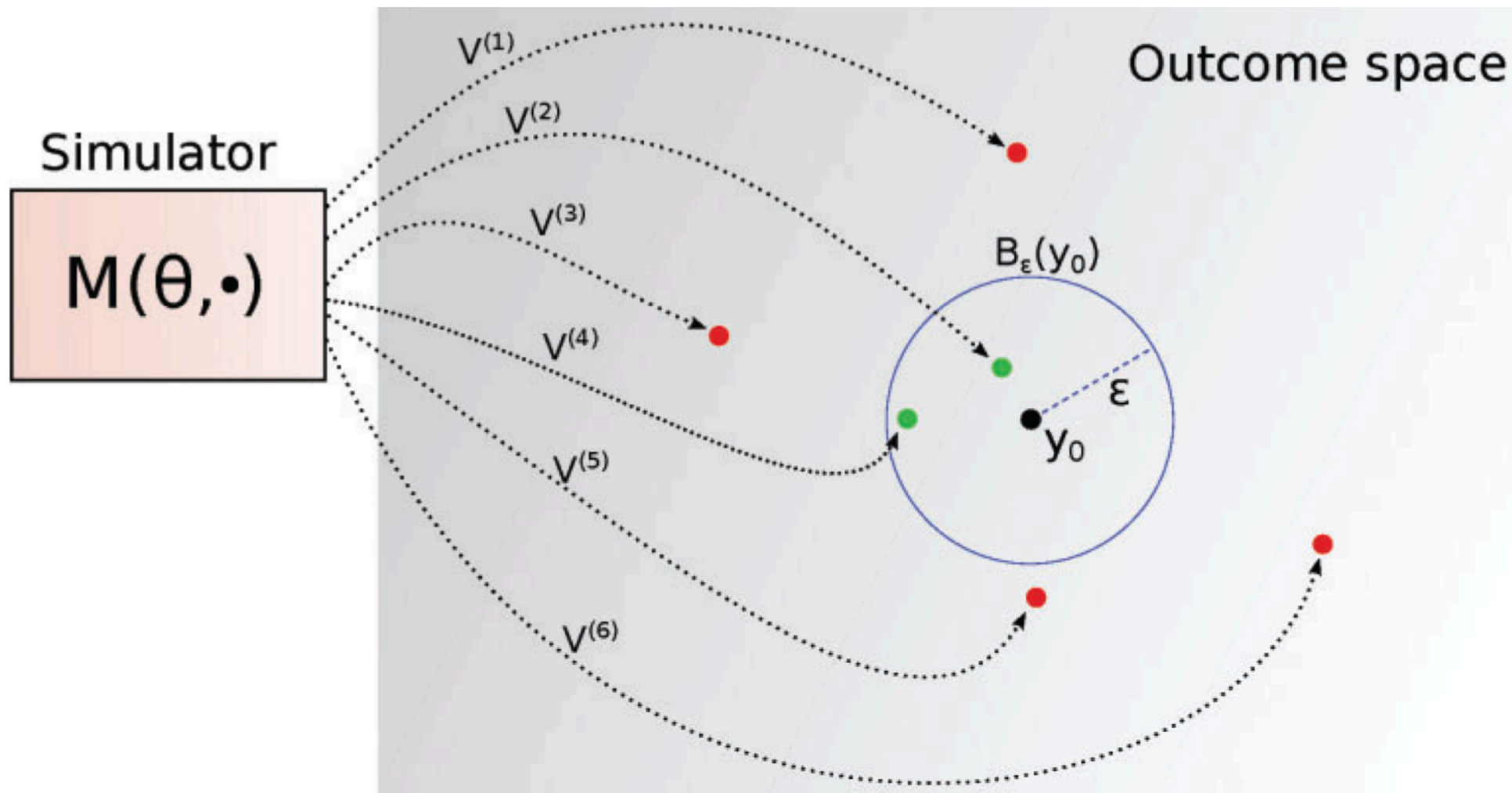
Particle Markov-chain Monte Carlo (pMCMC)



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

Approximate Bayesian Computation (ABC)

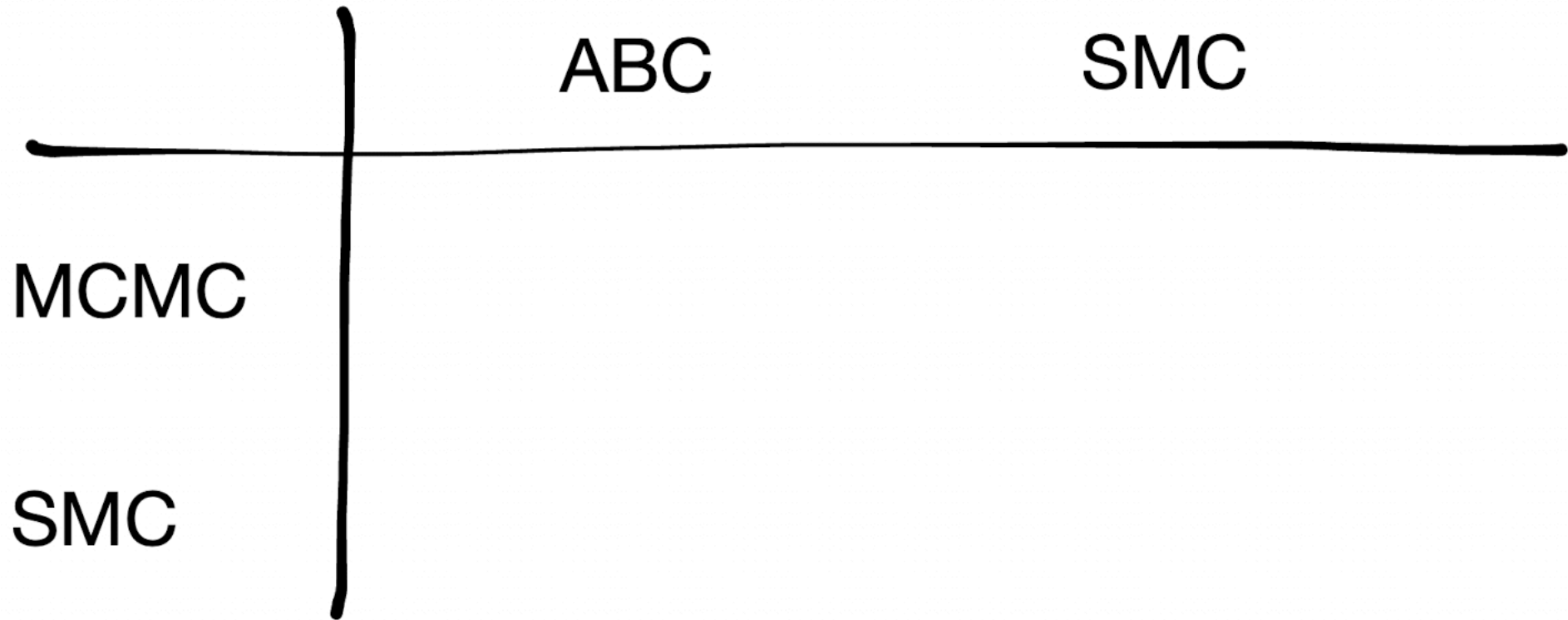
Lintusaari et al. (2016)



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

Sampling from
the posterior

Estimating the likelihood



Sampling from
the posterior

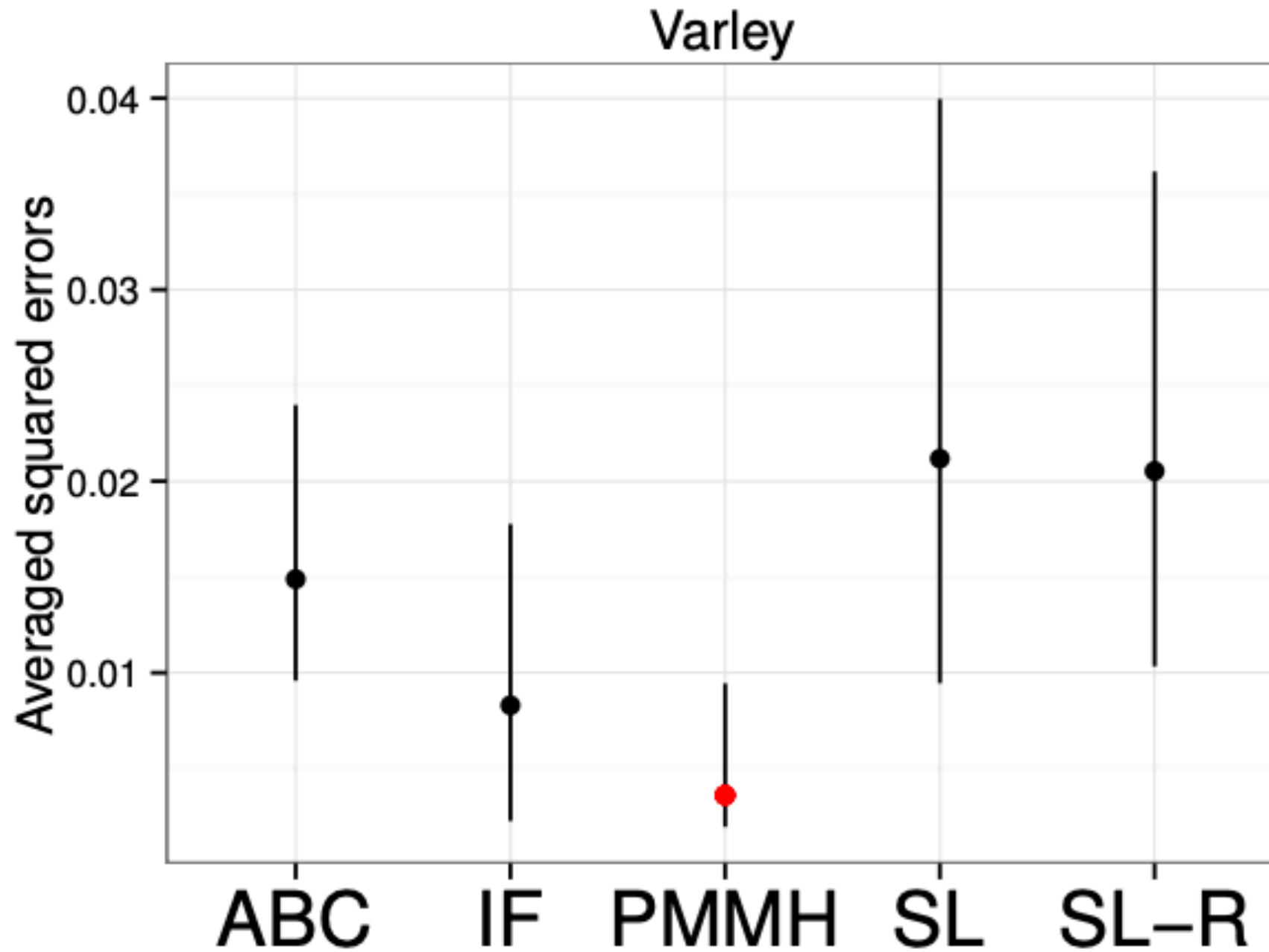
Estimating the likelihood

	ABC	SMC
MCMC	ABC-MCMC	PMCMC ✓
SMC	ABC-SMC ✓	SMC2

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

Aaron King, <https://kingaa.github.io/sbied/mif/mif.html>

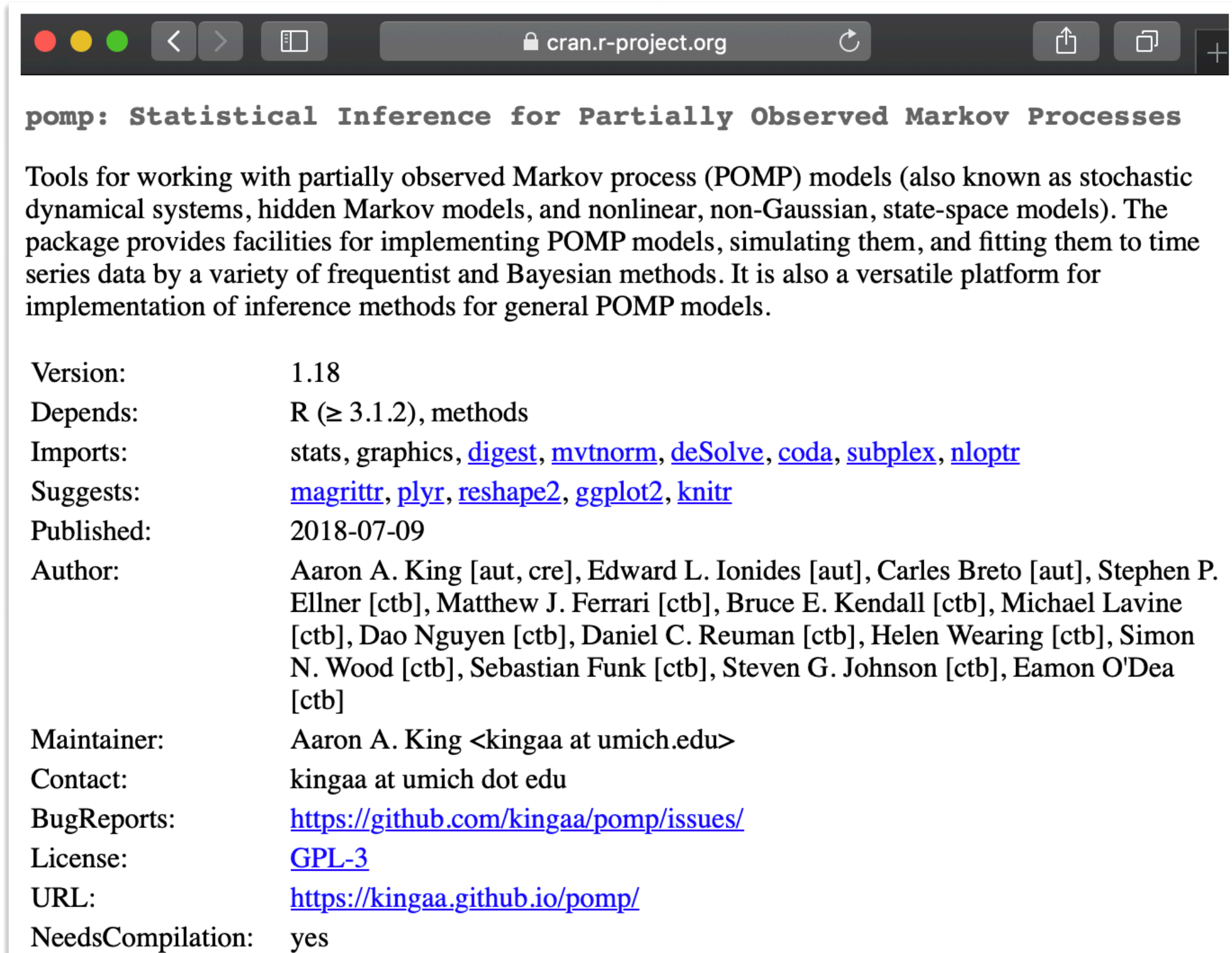
Computational efficiency vs Statistical efficiency



Fasiolo et al. (2015)

Computational efficiency vs Coding efficiency

pomp



pomp: Statistical Inference for Partially Observed Markov Processes

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.

Version: 1.18

Depends: R ($\geq 3.1.2$), methods

Imports: stats, graphics, [digest](#), [mvtnorm](#), [deSolve](#), [coda](#), [subplex](#), [nloptr](#)

Suggests: [magrittr](#), [plyr](#), [reshape2](#), [ggplot2](#), [knitr](#)

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 <kingaa at umich.edu>

Contact: kingaa at umich dot edu

BugReports: <https://github.com/kingaa/pomp/issues/>

License: [GPL-3](#)

URL: <https://kingaa.github.io/pomp/>

NeedsCompilation: yes

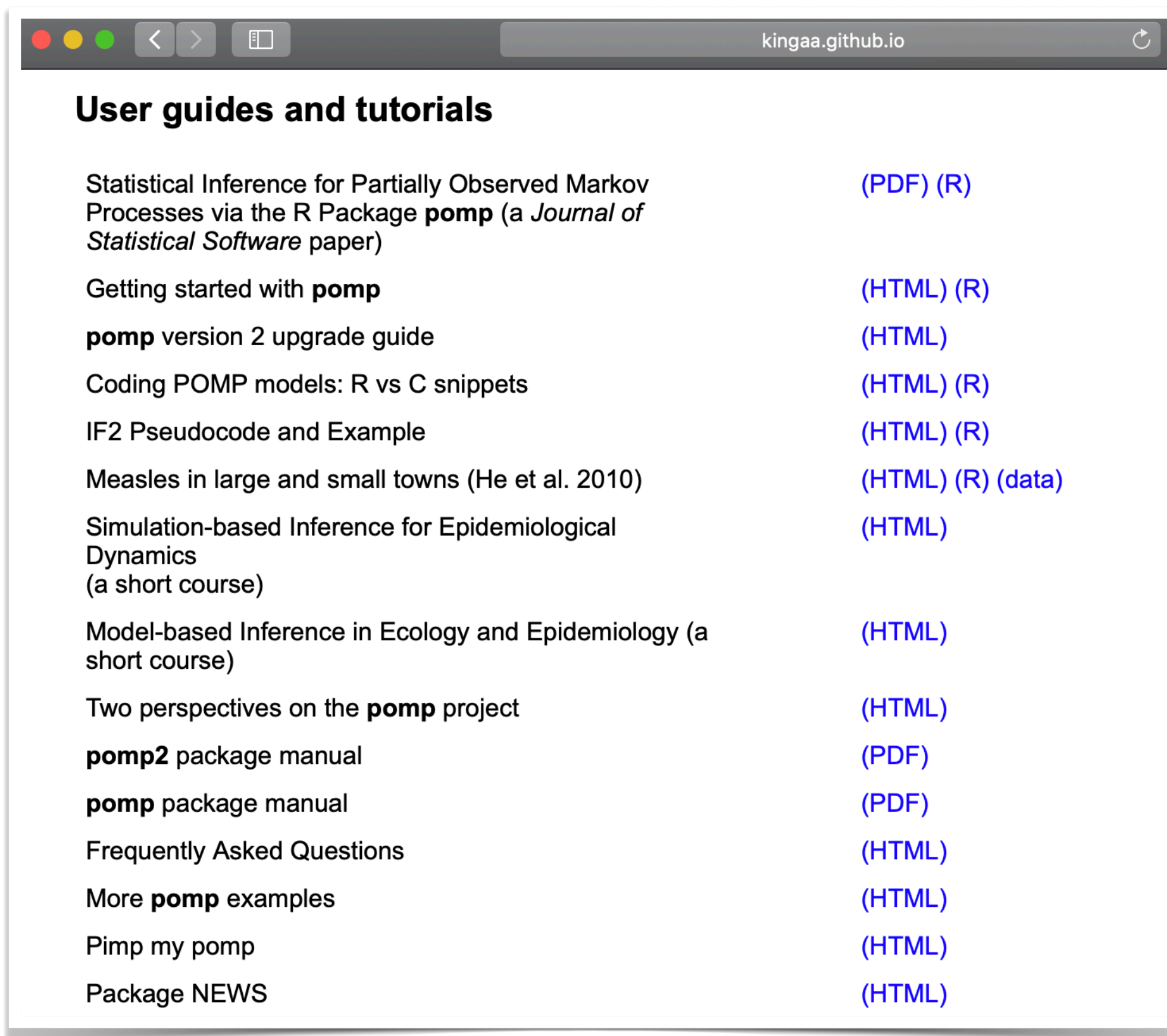
pomp

```
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



kingaa.github.io

User guides and tutorials

- 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



bioRxiv
THE PREPRINT SERVER FOR BIOLOGY

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?].

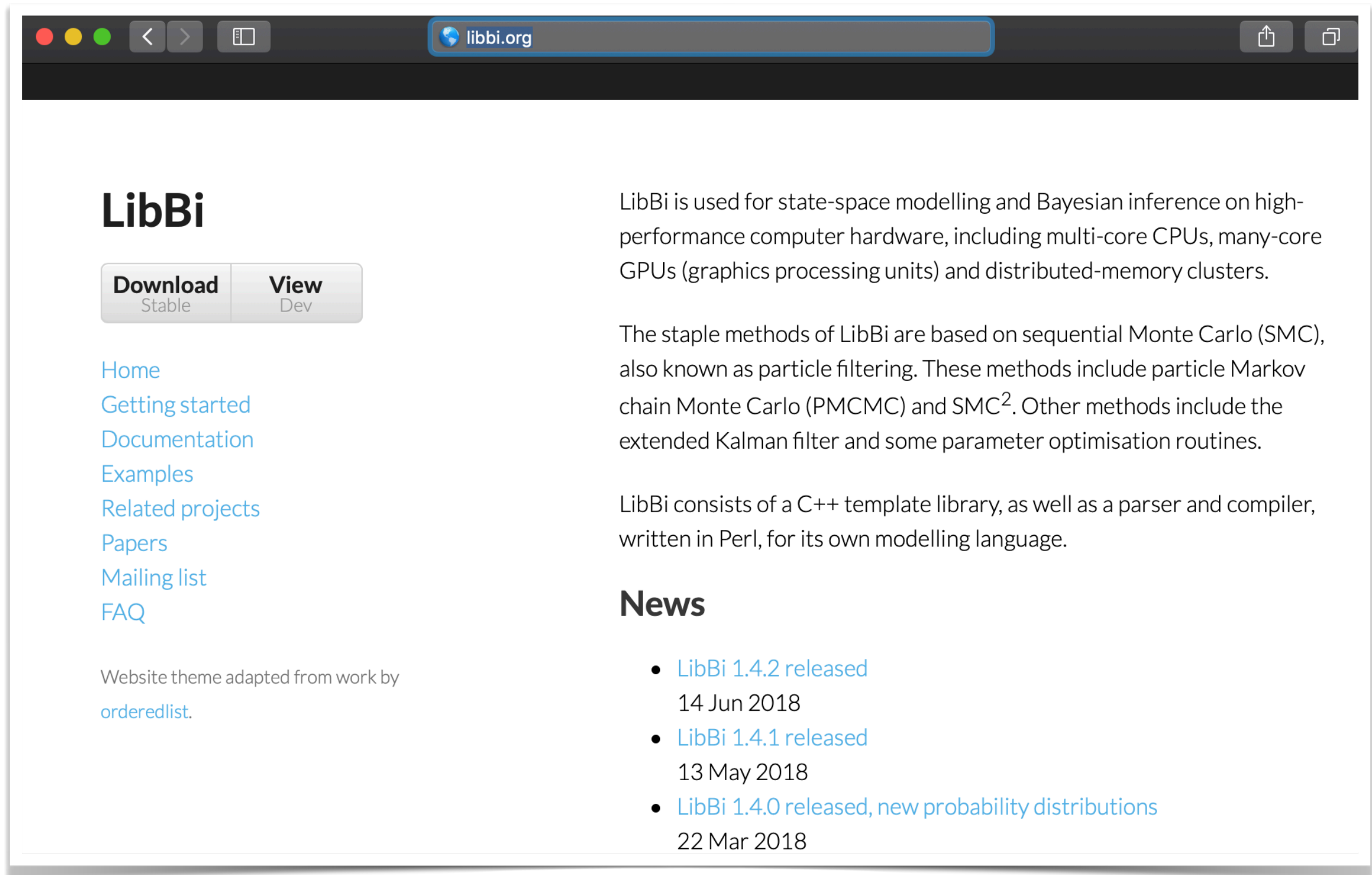
Abstract

Info/History

Metrics

 Preview PDF

LibBi



The image shows a browser window with the URL `libbi.org`. The page content includes a navigation menu on the left, a main heading, two buttons for downloading stable and development versions, a description of the library's use, a list of navigation links, a paragraph about the underlying methods, a paragraph about the library's components, and a news section with three entries.

LibBi

[Download Stable](#) [View Dev](#)

- [Home](#)
- [Getting started](#)
- [Documentation](#)
- [Examples](#)
- [Related projects](#)
- [Papers](#)
- [Mailing list](#)
- [FAQ](#)

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

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.

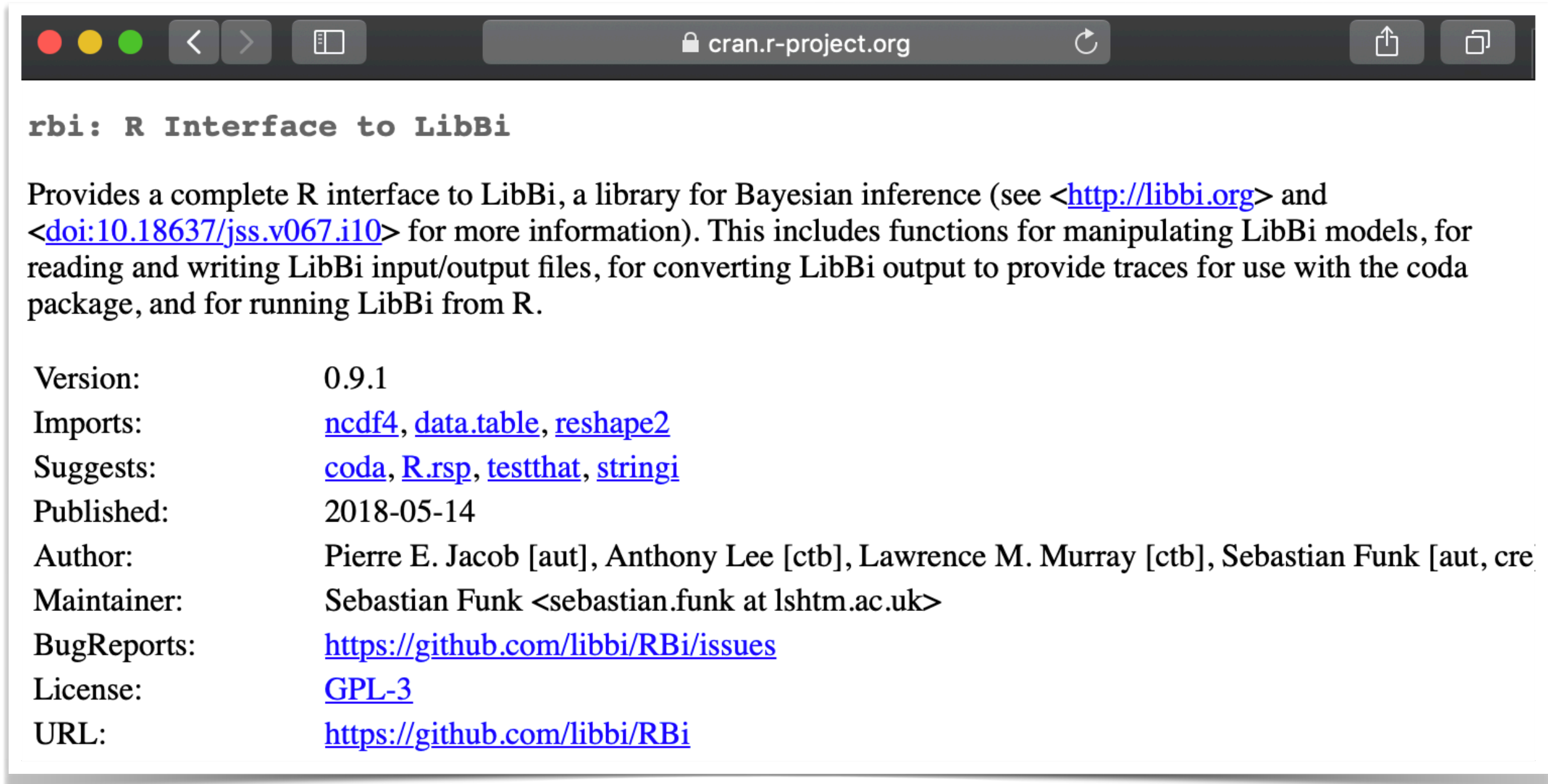
The staple methods of LibBi are based on sequential Monte Carlo (SMC), also known as particle filtering. These methods include particle Markov chain Monte Carlo (PMCMC) and SMC². Other methods include the extended Kalman filter and some parameter optimisation routines.

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

rbi



The image shows a browser window with the address bar displaying 'cran.r-project.org'. The page content is as follows:

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

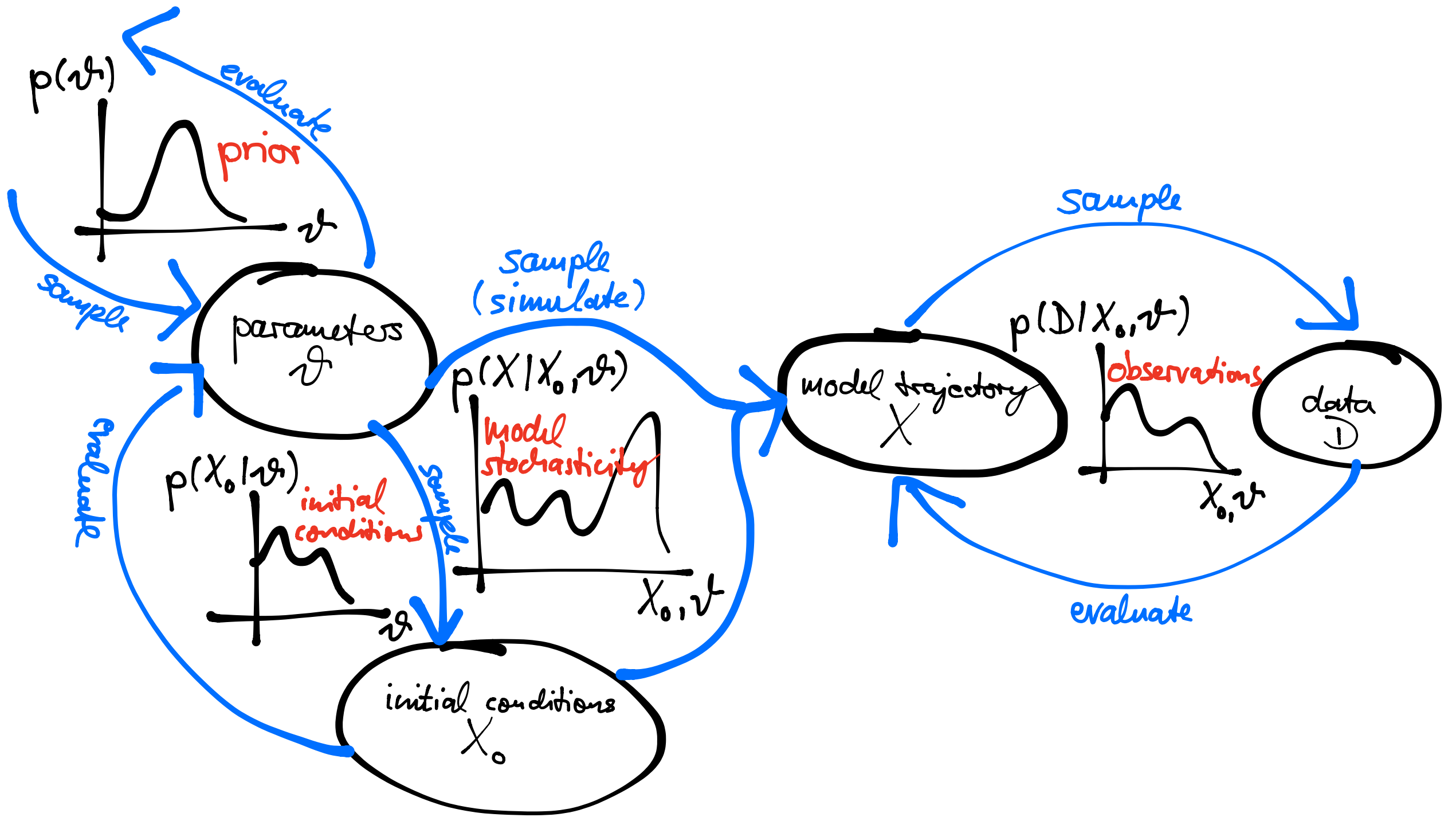
Author: Pierre E. Jacob [aut], Anthony Lee [ctb], Lawrence M. Murray [ctb], Sebastian Funk [aut, cre]

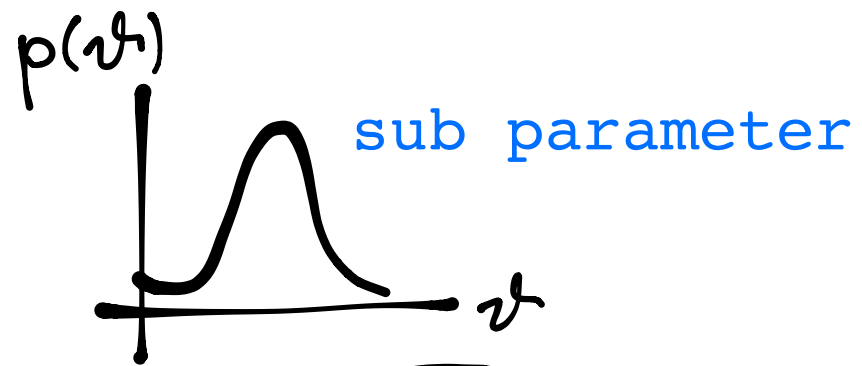
Maintainer: Sebastian Funk <sebastian.funk@lshtm.ac.uk>

BugReports: <https://github.com/libbi/RBi/issues>

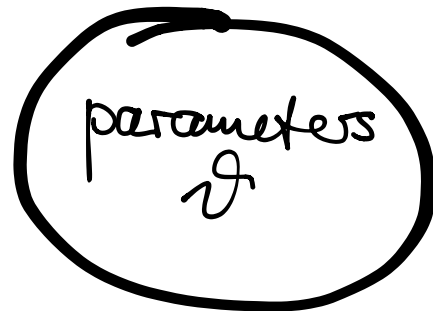
License: [GPL-3](#)

URL: <https://github.com/libbi/RBi>

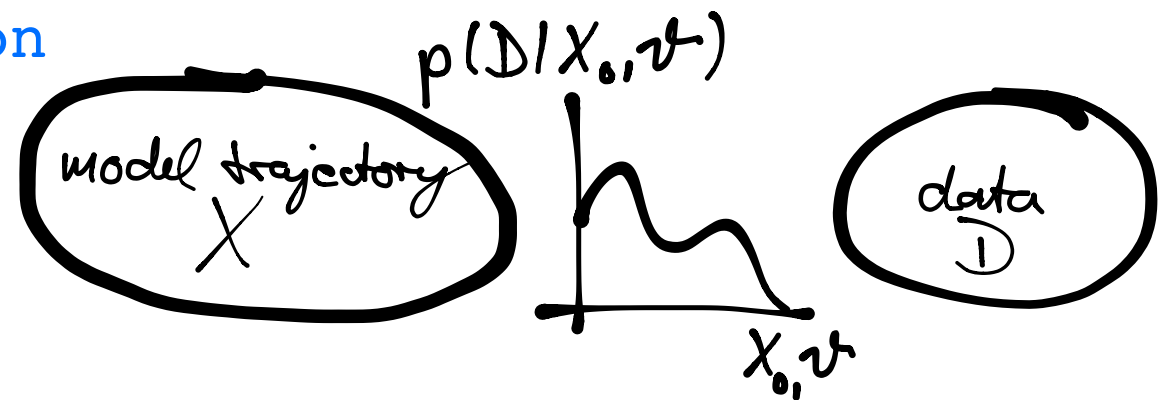
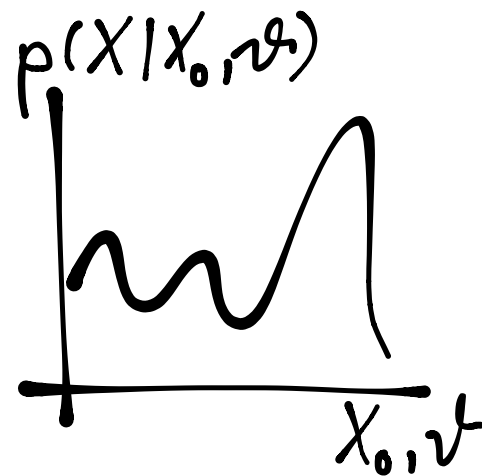




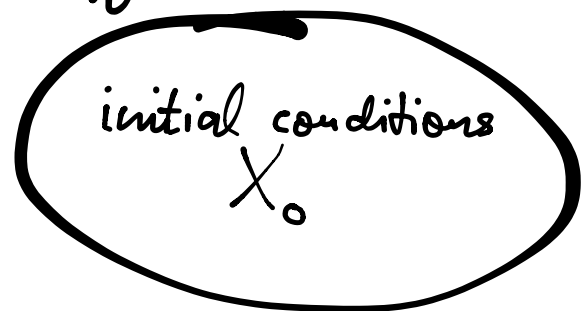
sub observation



sub transition



sub initial



```

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
  }

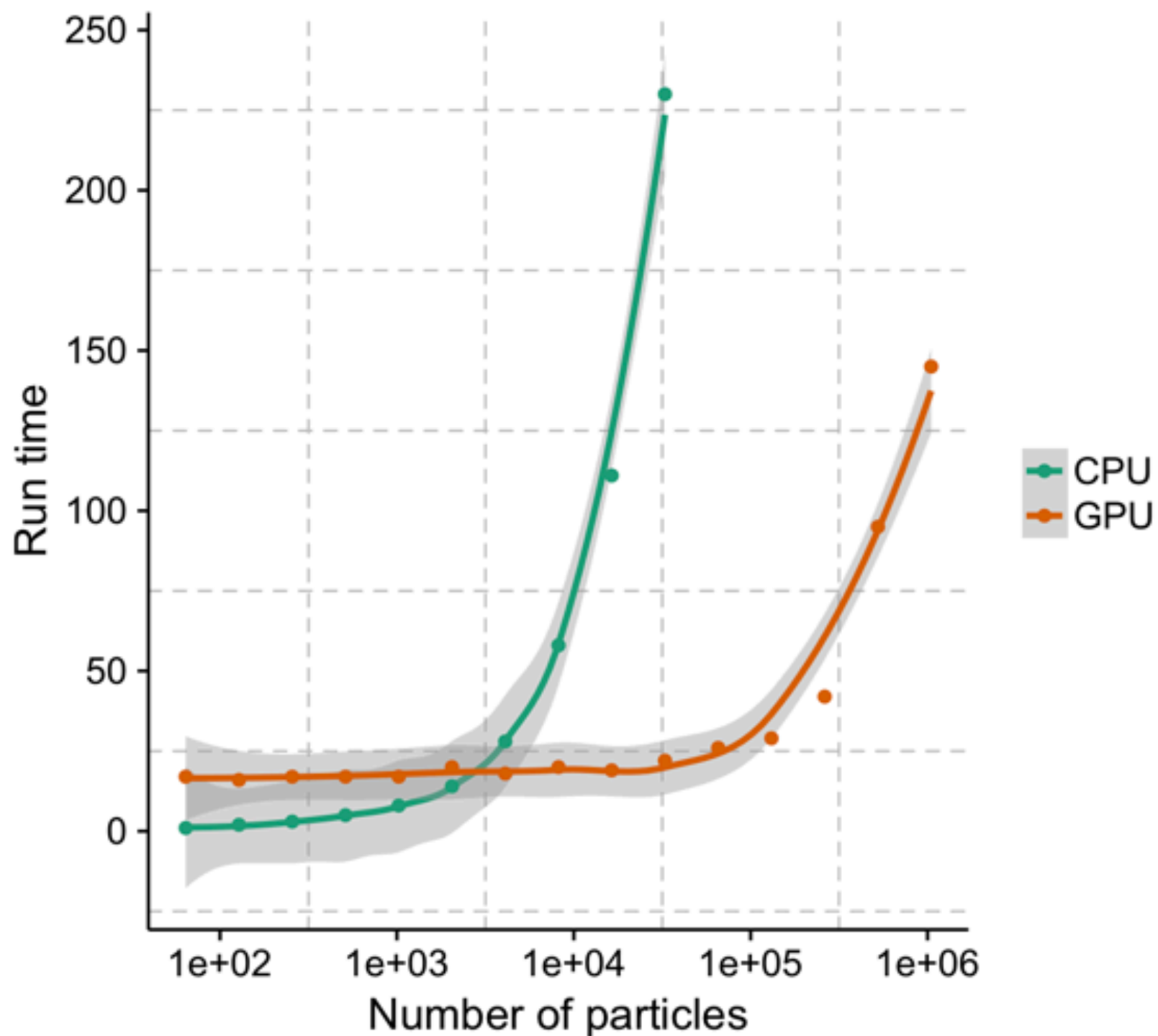
  sub transition { // daily time step
    inline beta = R0 / d_infection
    inline gamma = 1 / d_infection
    ode {
      dS/dt = - beta * S * I / N
      dI/dt = beta * S * I / N - gamma * I
      dR/dt = gamma * I
    }
  }

  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)