Aalto University

ELFI: Engine for Likelihood Free Inference

Antti Kangasrääsiö

Aalto University, Probabilistic Machine Learning Research Group

Joint work with: Jarno Lintusaari (Aalto), Henri Vuollekoski (Aalto), Kusti Skytén (Aalto), Marko Järvenpää (Aalto), Michael Gutmann (University of Edinburgh), Aki Vehtari (Aalto), Jukka Corander (University of Oslo), Samuel Kaski (Aalto)

European Meeting of Statisticians 2017 Helsinki

Slides and the demo are available at: `github.com/elfi-dev/zoo` (branch ems2017)

July 25th, 2017

Overview

Approximate Bayesian Computation

- What is ABC
- Visual Demonstrations

ELFI Python Library

- What is ELFI
- Demonstration

Likelihood function

Likelihood function is central in model-based inference

In many cases, inference might consist of just finding the maximum of the likelihood function (ML point estimate)

Likelihood function

Likelihood function is central in model-based inference

In many cases, inference might consist of just finding the maximum of the likelihood function (ML point estimate)

However, with many complex models either:

- No explicit likelihood function exists
- Evaluating the likelihood function is very expensive

Likelihood function

Likelihood function is central in model-based inference

In many cases, inference might consist of just finding the maximum of the likelihood function (ML point estimate)

However, with many complex models either:

- No explicit likelihood function exists
- Evaluating the likelihood function is very expensive

Examples include:

- Climate models
- Biological models
- Cognitive models

What is Approximate Bayesian Computation

ABC is a set of inference methods that bypass the evaluation of the likelihood function

The core idea in ABC is to repeatedly simulate predictions from the model and evaluate their discrepancy with the observation data

What is Approximate Bayesian Computation

ABC is a set of inference methods that bypass the evaluation of the likelihood function

The core idea in ABC is to repeatedly simulate predictions from the model and evaluate their discrepancy with the observation data

Intuition: If parameters lead to predictions that match with the observation data, then these parameters have high likelihood

What is Approximate Bayesian Computation

ABC is a set of inference methods that bypass the evaluation of the likelihood function

The core idea in ABC is to repeatedly simulate predictions from the model and evaluate their discrepancy with the observation data

Intuition: If parameters lead to predictions that match with the observation data, then these parameters have high likelihood

ABC methods are based on Bayesian statistics, and with some assumptions we can closely approximate the true likelihood

ABC Math Background

Bayes' Rule says: $P(\theta|D) \propto P(D|\theta)P(\theta)$

Thus, we can always estimate $P(\theta|D)$ with the rejection scheme:

- Draw $\theta \sim P(\theta)$ and $D \sim P(D|\theta)$
- If $D = D_{obs}$, accept θ as a sample from $P(\theta|D_{obs})$

ABC Math Background

Bayes' Rule says: $P(\theta|D) \propto P(D|\theta)P(\theta)$

Thus, we can always estimate $P(\theta|D)$ with the rejection scheme:

- Draw $\theta \sim P(\theta)$ and $D \sim P(D|\theta)$
- If $D = D_{obs}$, accept θ as a sample from $P(\theta|D_{obs})$

Problem: Is most cases our acceptance rate is nearly 0 as we require that $D = D_{obs}$ precisely

ABC Math Background

Bayes' Rule says: $P(\theta|D) \propto P(D|\theta)P(\theta)$

Thus, we can always estimate $P(\theta|D)$ with the rejection scheme:

- Draw $\theta \sim P(\theta)$ and $D \sim P(D|\theta)$
- If $D = D_{obs}$, accept θ as a sample from $P(\theta|D_{obs})$

Problem: Is most cases our acceptance rate is nearly 0 as we require that $D = D_{obs}$ precisely

Approx: Accept samples when discrepancy $\delta(D, D_{obs}) < \varepsilon$, thus yielding samples from $P(\theta|\delta(D, D_{obs}) < \varepsilon)$

ABC Math Background

Bayes' Rule says: $P(\theta|D) \propto P(D|\theta)P(\theta)$

Thus, we can always estimate $P(\theta|D)$ with the rejection scheme:

- Draw $\theta \sim P(\theta)$ and $D \sim P(D|\theta)$
- If $D = D_{obs}$, accept θ as a sample from $P(\theta|D_{obs})$

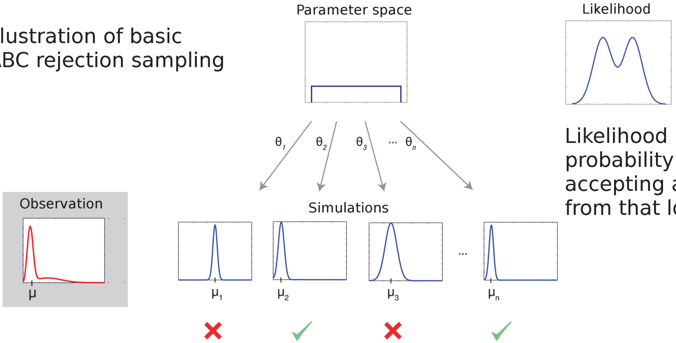
Problem: Is most cases our acceptance rate is nearly 0 as we require that $D = D_{obs}$ precisely

Approx: Accept samples when discrepancy $\delta(D, D_{obs}) < \varepsilon$, thus yielding samples from $P(\theta|\delta(D, D_{obs}) < \varepsilon)$

In general, this approach makes sense when evaluating the generative model is easier than evaluating the likelihood

ABC Visual Demonstration 1

Illustration of basic
ABC rejection sampling



Likelihood is the
probability of
accepting a sample
from that location

Accept samples that are close
enough to observation data

Figure adapted from: Sunnåker et al., 2013, `https://doi.org/10.1371/journal.pcbi.1002803`

ABC Design Parameters

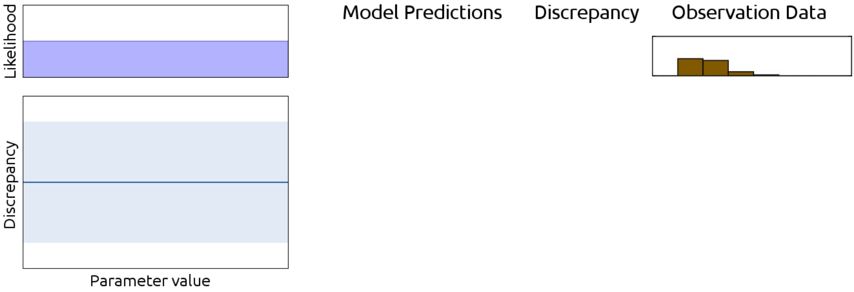
What one needs to choose to use ABC:

- Discrepancy function: $\delta(D_{sim}, D_{obs}) \rightarrow [0, \infty)$
- Discrepancy threshold: $\varepsilon \in [0, \infty)$

Choosing the discrepancy function is analogous to choosing a loss function: no perfect choice but some are more justified than others

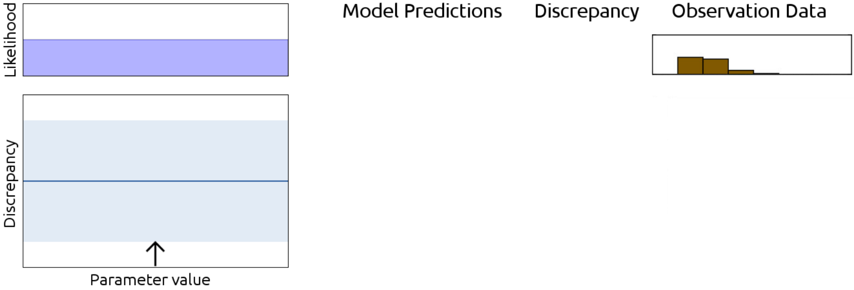
Threshold can even be chosen post-sampling if all samples can be stored in memory

ABC Visual Demonstration 2



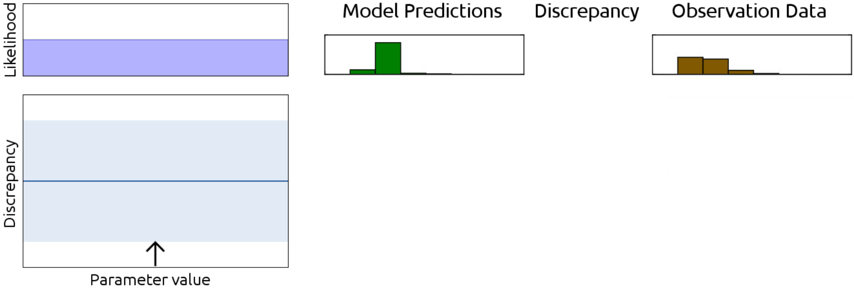
From (slides): Kangasräsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



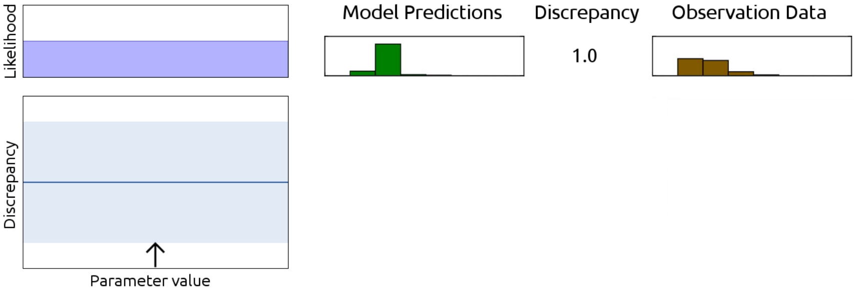
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



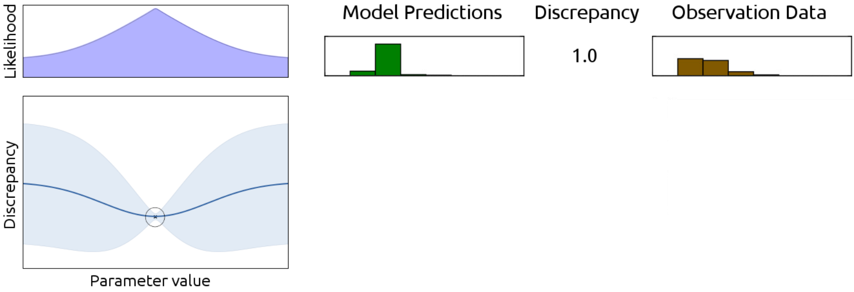
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



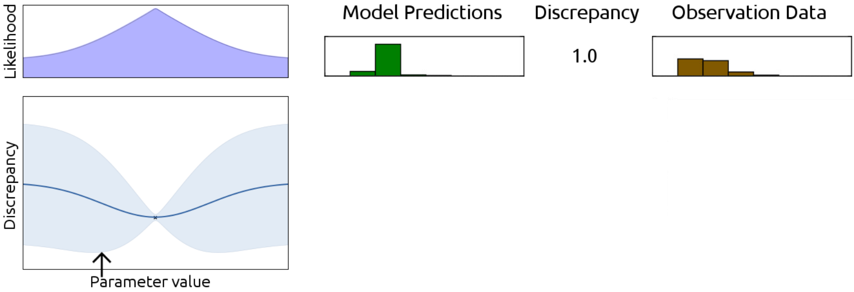
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



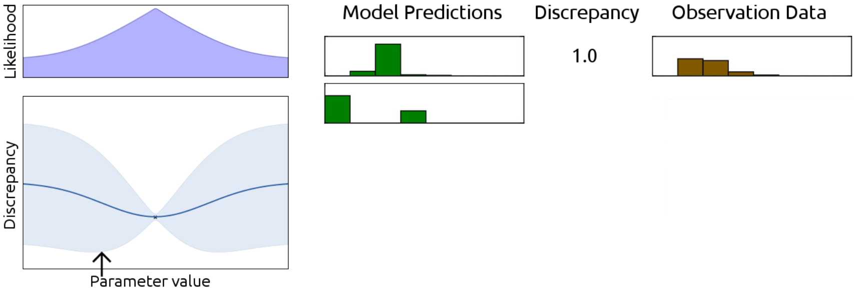
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



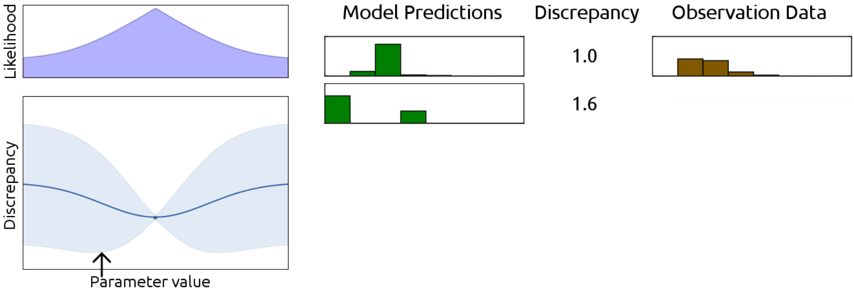
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



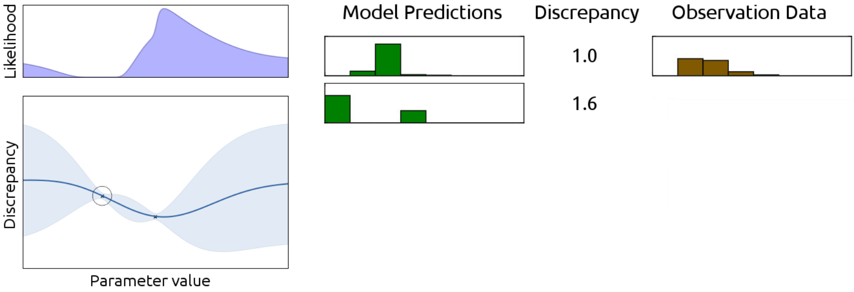
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



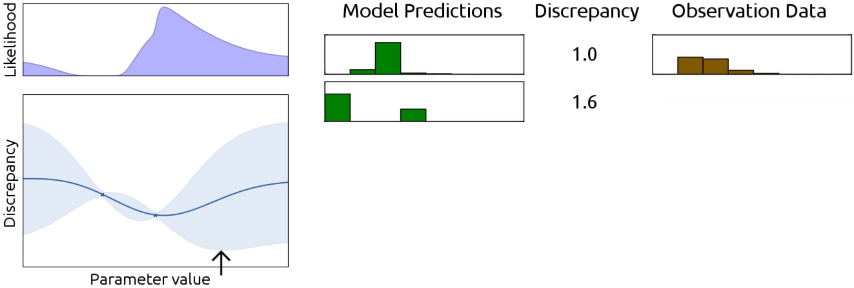
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



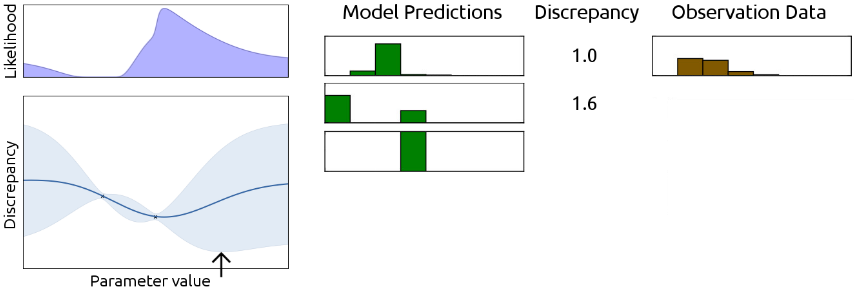
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



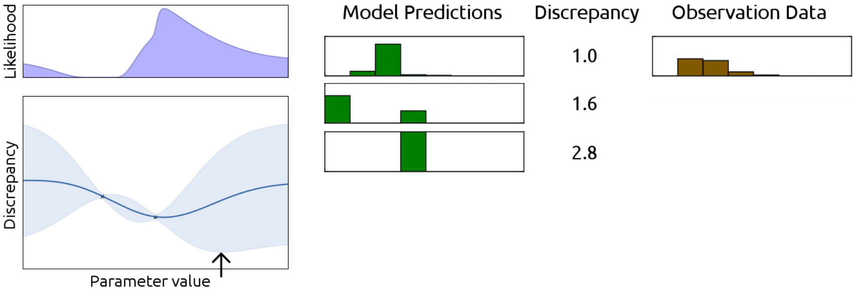
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



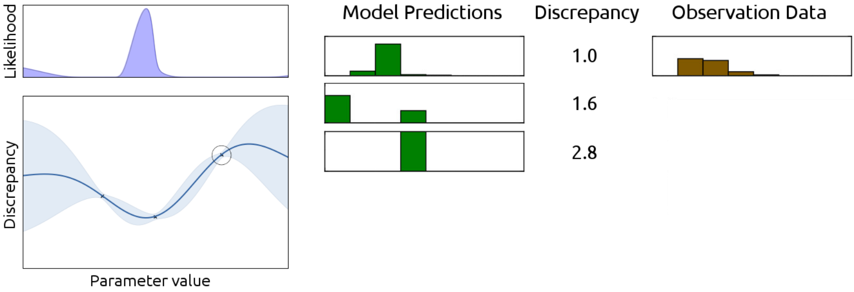
From (slides): Kangasräăsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



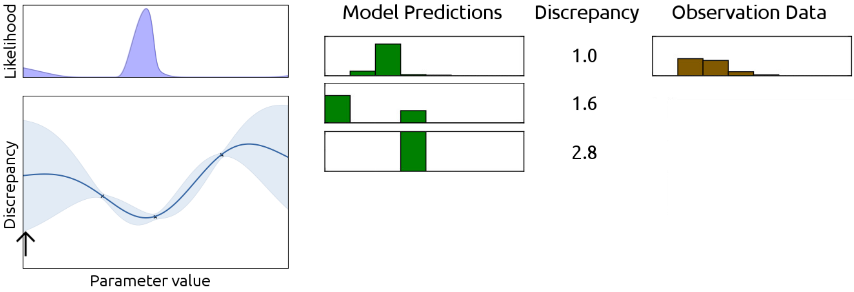
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



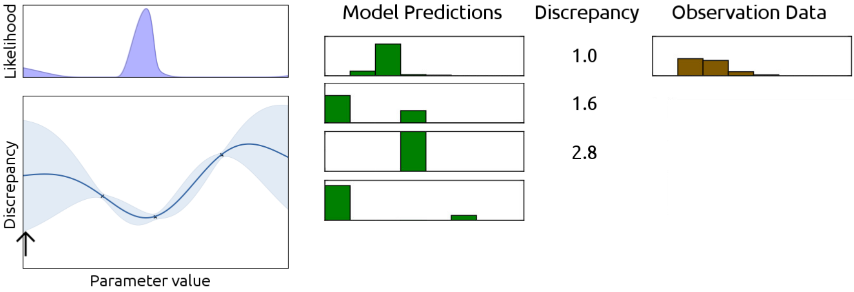
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



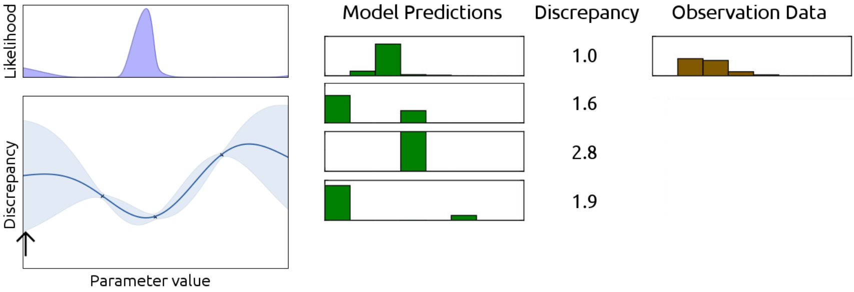
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



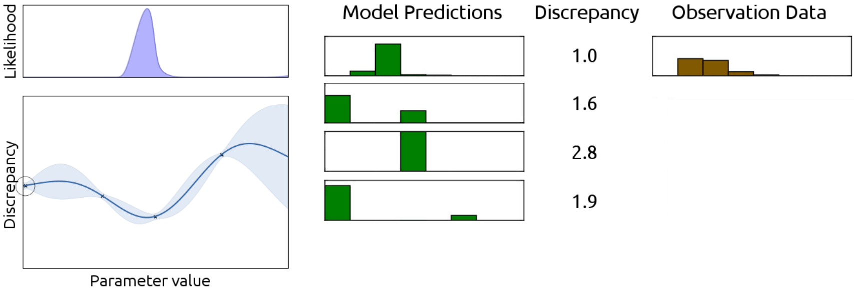
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



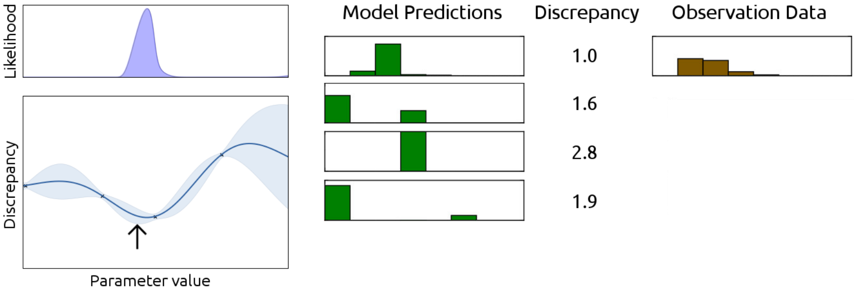
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



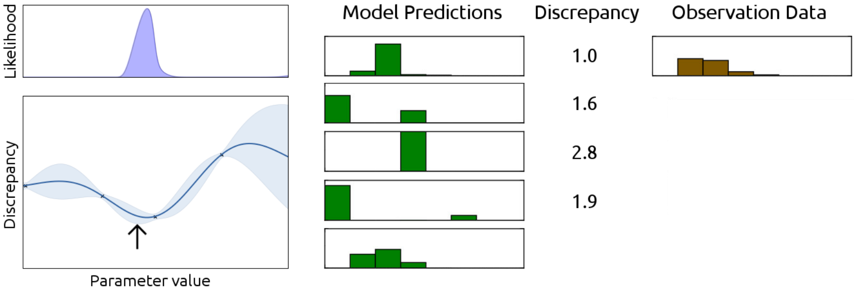
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



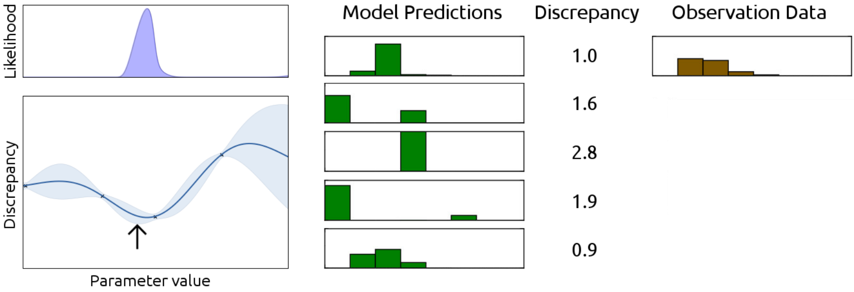
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



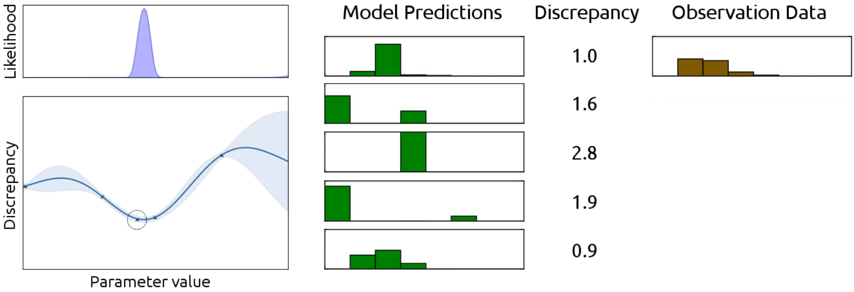
From (slides): Kangasräasiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



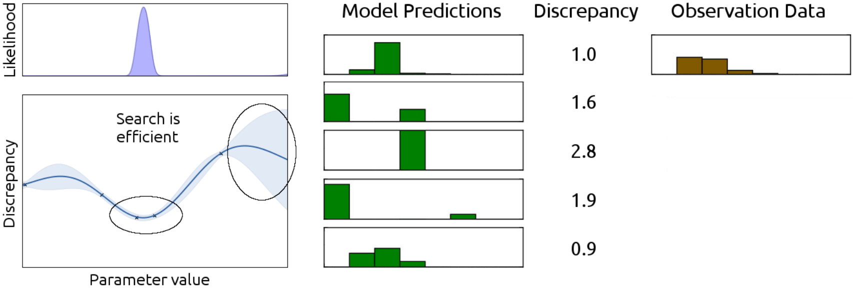
From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

ABC Visual Demonstration 2



From (slides): Kangasrääsiö et al., 2017, `http://dl.acm.org/citation.cfm?id=3025576`
Algorithm (BOLFI): Gutmann & Corander, 2016, `http://jmlr.org/papers/v17/15-017.html`

Engine for Likelihood Free Inference

ELFI is an open-source Python library that aims to make it easy for **practicioners** to apply ABC

Main features of the library:

- Implementations of ABC algorithms
- Easy syntax for defining model structure
- Storage and re-use of results
- Reporting and diagnostics tools

ELFI for Developers

ELFI aims to also become a platform where implementations of new methods can be easily published

For **developers** ELFI offers:

- Modular design of the library
- Well-defined API for implementing own modules
- Development and maintenance of the library (by Aalto PML)
- Growing user community

Demonstration

Live demonstration of the main features of ELFI

Slides and the demo are available at:
`github.com/elfi-dev/zoo` (branch **ems2017**)