PHYSICS AT THE TERA SCALE
Helmholtz Alliance

— Bayesian Analysis Toolkit

Frederik Beaujean[1], Allen Caldwell[1], Daniel Kollàr[2], Kevin Kröninger[3], Shabnaz Pashapour[3], Arnulf Quadt[3]

[1]Max-Planck-Institut für Physik, [2]CERN, [3]Georg-August-Universität Göttingen

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Introduction

- Primary aims of data analysis:
  - Compare data with model
  - Assess the validity of the model
  - Find model parameters
- Bayesian data analysis
  - comprehensive statistical interpretation
  - not trivial to implement
  - need for accessible common tools
- The idea behind BAT is to
  - Provide all the common parts of Bayesian analysis in a software package
  - Create a flexible environment to phrase arbitrary problems
  - Develop a set of well-tested/tuned numerical algorithms and tools

Overview

- BAT:
  - Software package to solve statistical problems using Bayesian approach

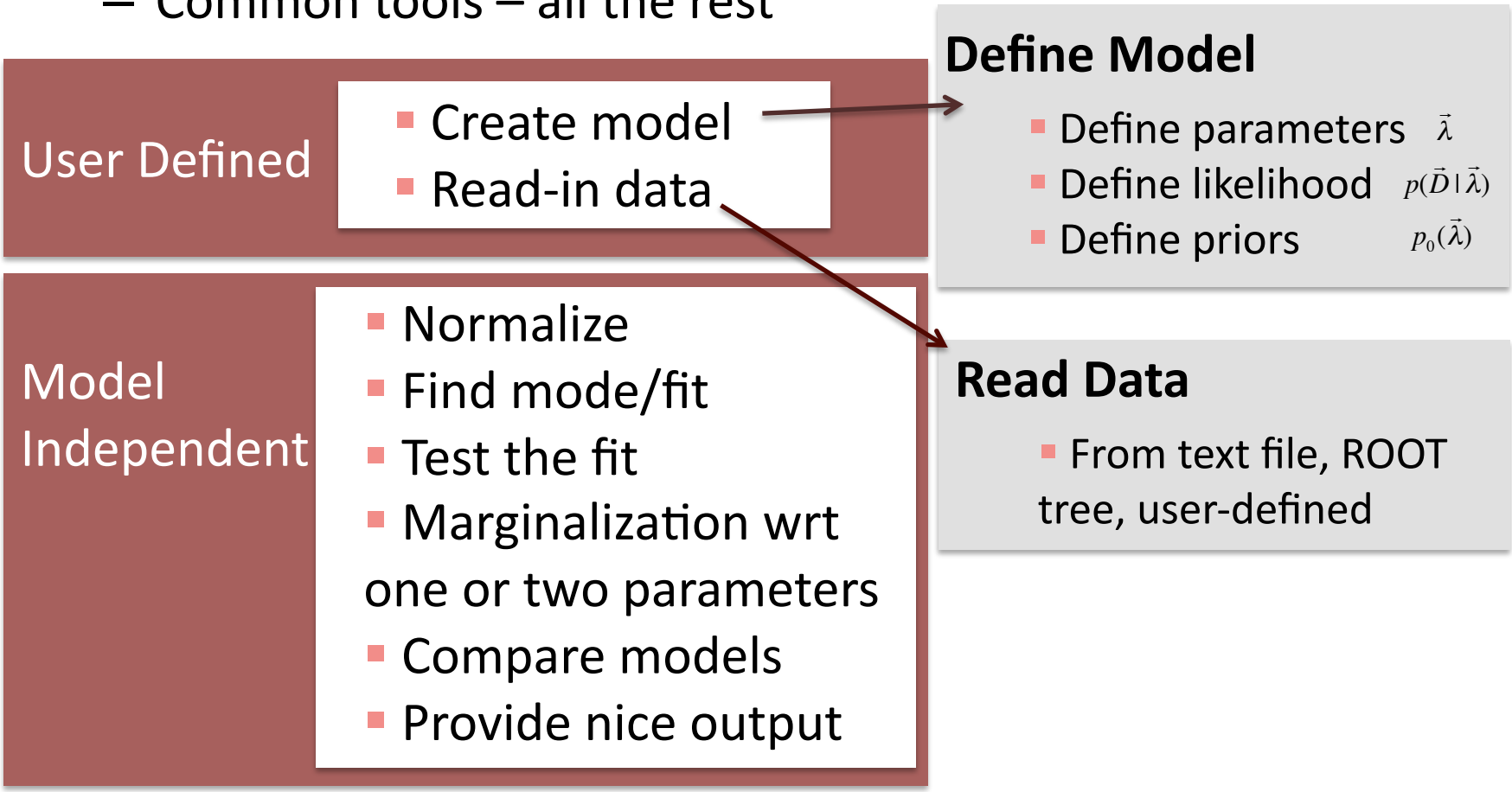
$$p(\vec{\lambda} \mid \vec{D}) = \frac{p(\vec{D} \mid \vec{\lambda}) p_0(\vec{\lambda})}{\int p(\vec{D} \mid \vec{\lambda}) p_0(\vec{\lambda}) d\vec{\lambda}}$$

  - Based on C++ framework
  - Interfaced with ROOT, Cuba, Minuit and RooStats
  - Flexible to use user-defined functions and algorithms
  - Free software: tutorials, examples, all at http://mpp.mpg.de/bat/

- The key is the use of **Markov Chain Monte Carlo**
- BAT paper: Computer Physics Communications **180** (2009) 2197-2209

The Approach

- Separate the common parts from the rest
  - Case specific – the model and the data
  - Common tools – all the rest



Common Tools

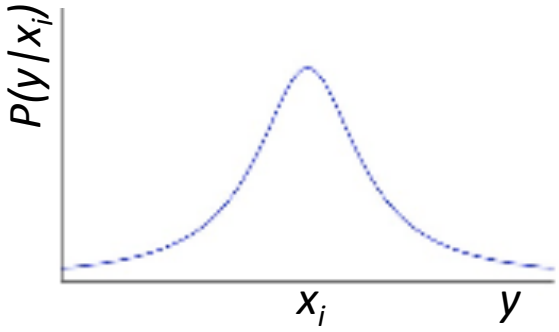
- Marginalization
  - Markov Chain Monte Carlo (MCMC) – Metropolis
  - A lot of emphasis on efficiencies, performance and validation
- Integration
  - Simple Monte Carlo algorithms
    - Sampled mean, importance sampling
  - Interface to CUBA (VEGAS)
- Optimization
  - Monte Carlo (hit & miss)
  - Interface to Minuit
  - Simulated annealing
- Error propagation
  - Calculate any function of the parameters during a run
- Goodness-of-fit
  - Ensemble testing and p-value

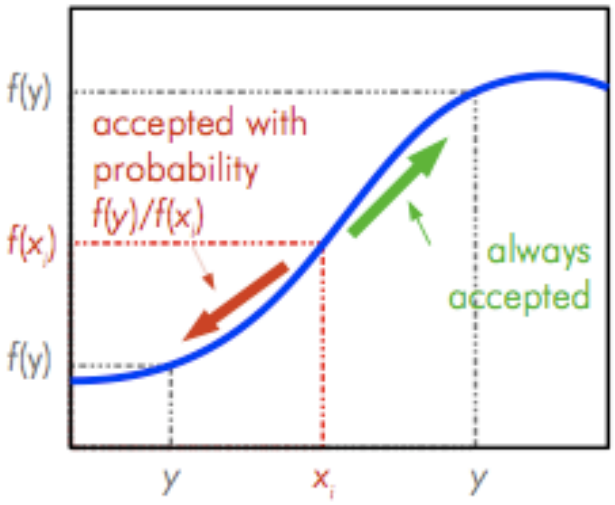
Markov Chain Monte Carlo (MCMC)

- Aim: mapping a positive function *f(x)* by taking a random walk to points with higher probabilities

- Metropolis algorithm in BAT
  - Starts at a random x_i
  - Generate a random point around x_i
    - If $f(y) \geq f(x_i)$, set $x_{i+1} = y$
    - If $f(y) < f(x_i)$, set $x_{i+1} = y$ *with probability* $r=f(y)/f(x_i)$
    - If *y* not accepted, stay where you are
    - Generate a new *y* around the new *x*
  - For an infinite number of steps
    - x_i distribution is guaranteed to converge to *f(x)*
  - For finite number of steps
    - have to check for convergence

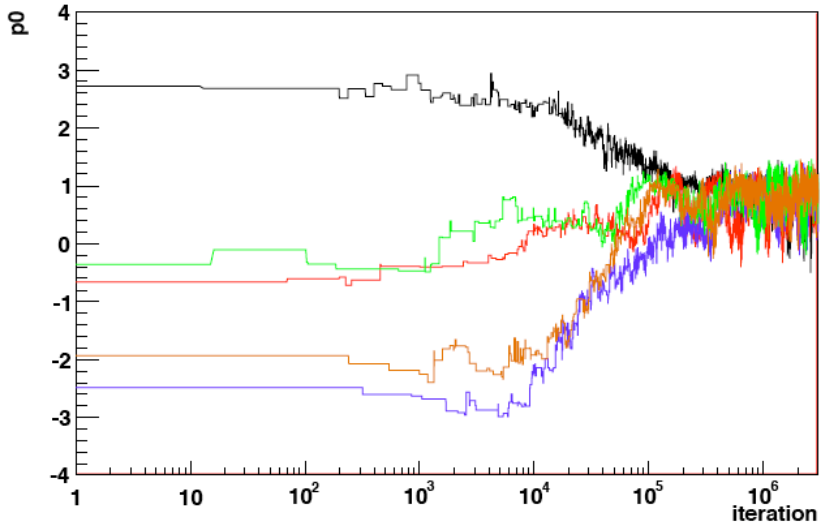
Proposal function plot





MCMC II

- Pre-run/burn-in phase
  - Use several chains/starting positions in parameter space
  - Update scales of proposal function to optimize performance
  - Monitor evolution of log-likelihood and individual parameters
- Convergence based on R-value[1]
  - A ratio of the mean of variances and the variance of the means of chains
  - Efficiency: 15%-50%
- Main run
  - All scales are fixed. Collect samples for posterior analysis
  - Get marginalized distributions
  - Save the chain as TTree.



[1] A. Gelman and D.B. Rubin, *Inference from Iterative Simulation Using Multiple Sequences, Statistical Science* ***7 (1992) 457-472***

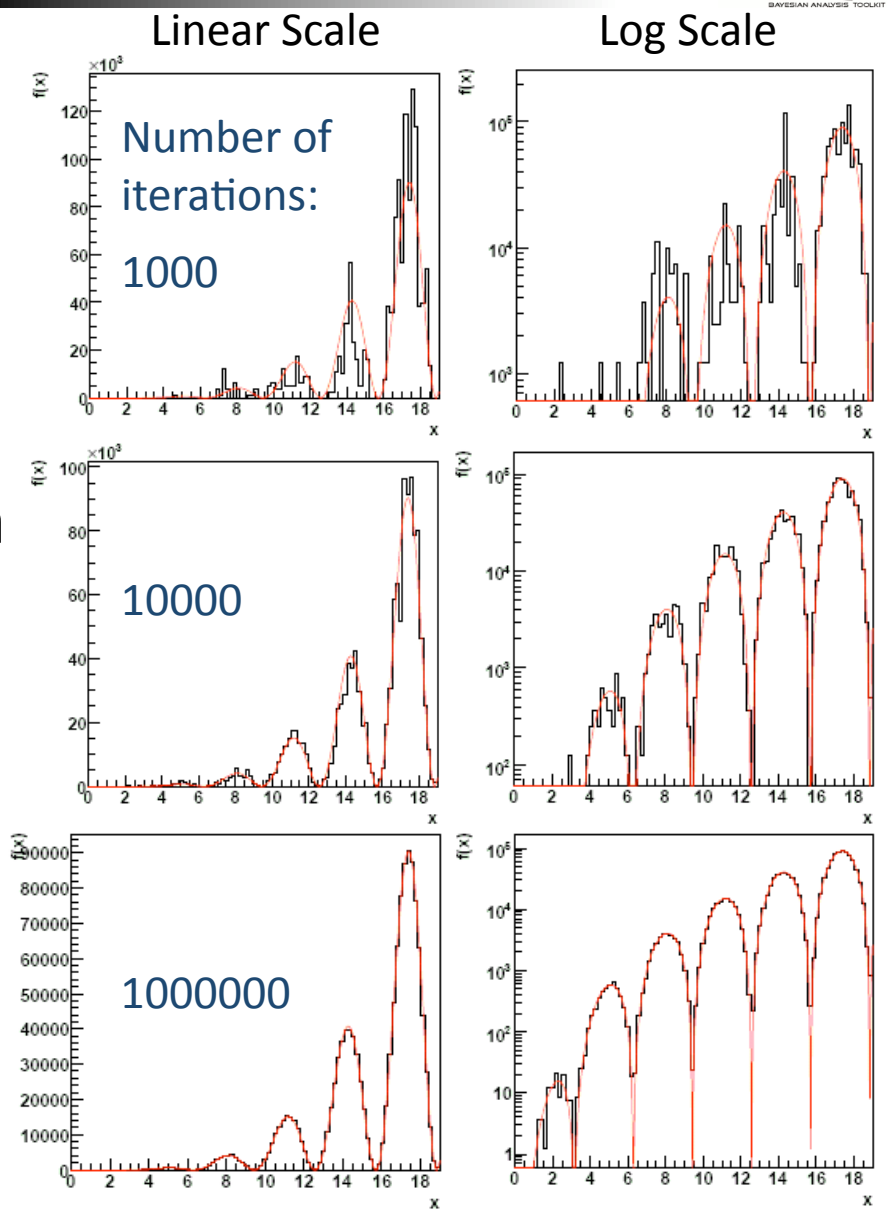
MCMC in Action

- Mapping an arbitrary function:

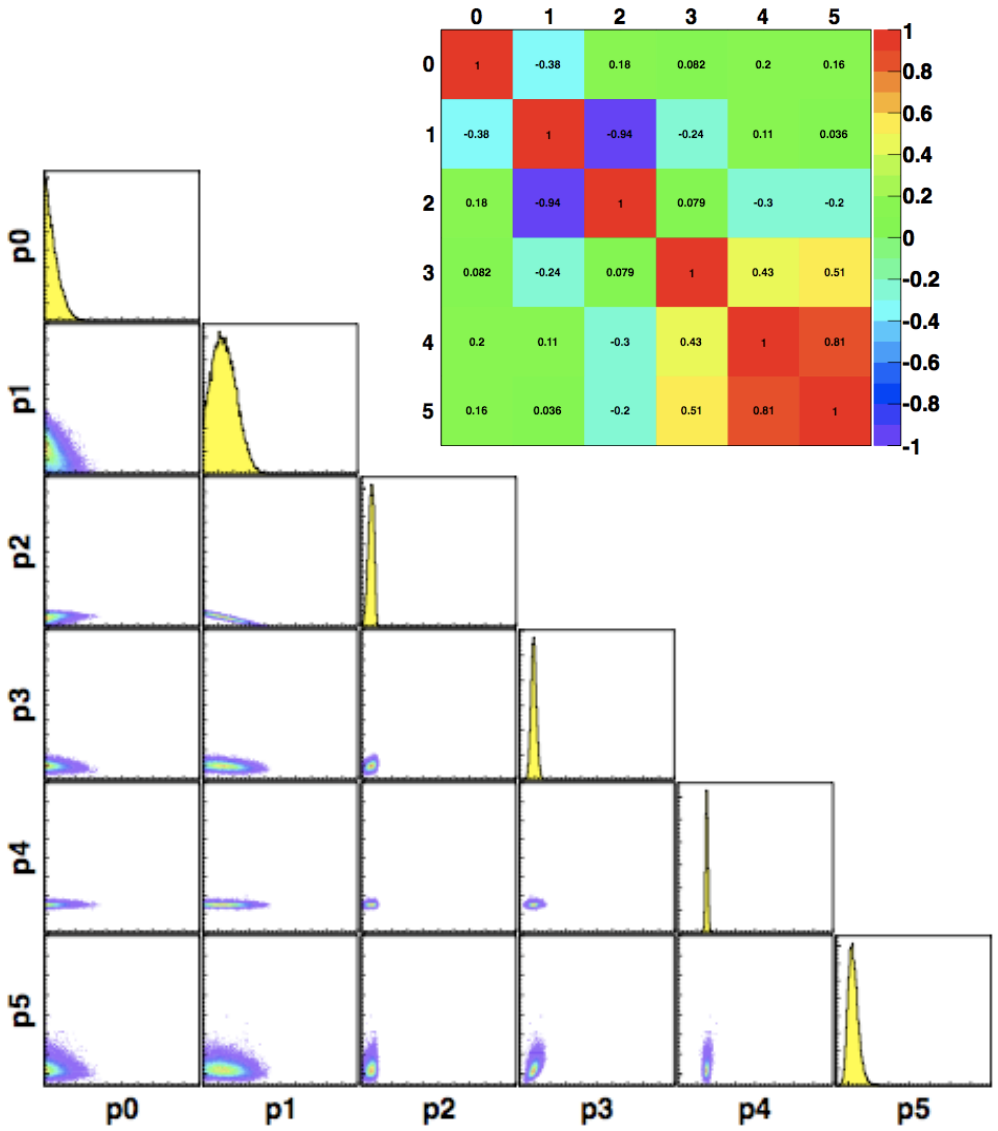
  $f(x) = x^4 \sin^2 x$

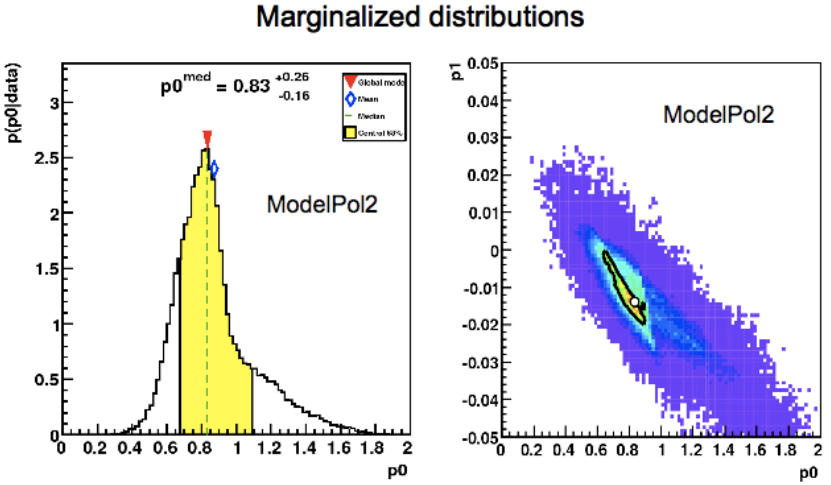
- MCMC sampled distribution quickly converges to the underlying distribution

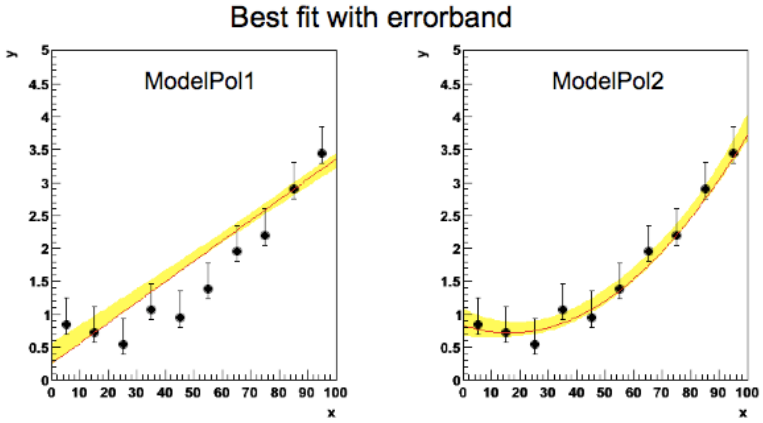
- Complicated shapes with multiple minima and maxima



Sample Outputs







Comments

- Easy to use
- Nice graphical output
- Extensive test suite for proper MCMC sampling
- MCMC output allows for flexible use of posterior
- Simplified error propagation
- Handling very complicated problems with a large number of parameters
- Doing all the hard numerical calculations for model selection and hypothesis testing

- Includes a broad array of sophisticated numerical packages for fitting, integration, …

Future Plans

- Continuously improve BAT performance
  - Speed
    - Simple code changes
    - More modularity
    - Possibility of parallel processing
  - Functionality
    - Addition of new algorithms
  - No ROOT dependence

Summary

- The Bayesian Analysis Toolkit have been introduced
- The philosophy behind it and some of its capabilities are presented
- MCMC implementation and performance in BAT is shown
- Very briefly ideas for future and reasons to use BAT are discussed