Introduction to the Bayesian Analysis Toolkit - BAT

BAT Tutorial

Rome – April 22nd 2010

Frederik Beaujean[1], Allen Caldwell[1], Daniel Kollár[2], Kevin Kröninger[3]

[1]Max Planck Institute for Physics, Munich

[2]CERN

[3]University of Göttingen

Bayes' formula for parameter estimation

$$p(\vec{\lambda}\,|\,\vec{D}) = \frac{p(\vec{D}\,|\,\vec{\lambda})\; p_0(\vec{\lambda})}{\int p(\vec{D}\,|\,\vec{\lambda})\; p_0(\vec{\lambda})\; d\vec{\lambda}}$$

BAT Software package for solving of statistical problems using Bayesian approach:

- collects common tools in Bayesian analyses in a single modular framework
- C++ based, interfaced to ROOT

USER DEFINED

- create model
- read in data

MODEL INDEPENDENT (common tools)

- normalize
- find mode / fit
- test the fit
- marginalize wrt one or two parameters
- compare models
- provide nice output
  - plots
  - numbers

Define MODEL

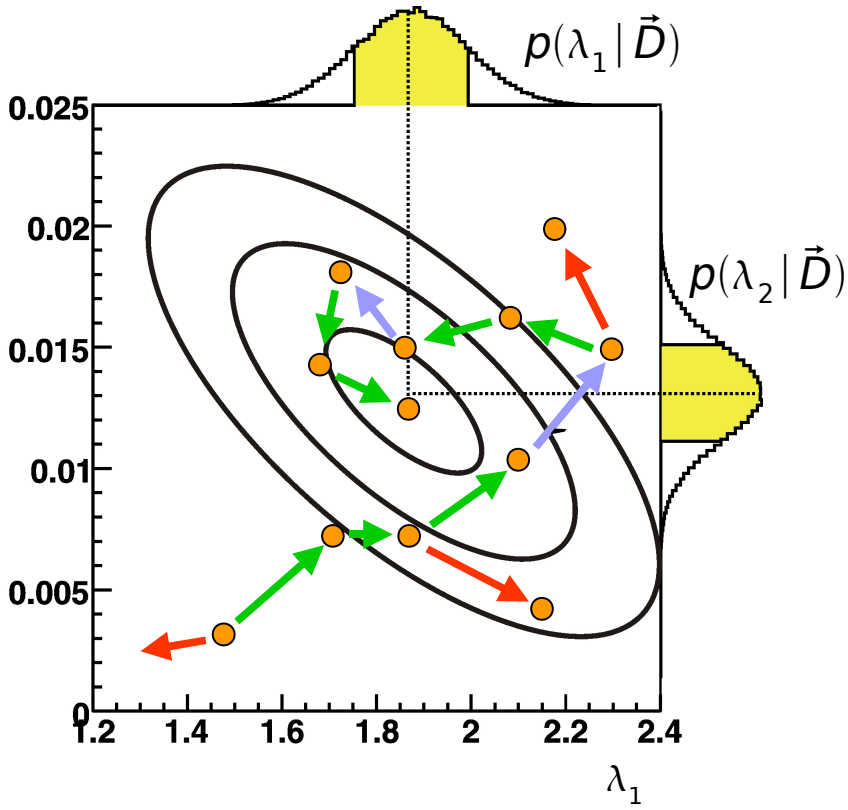
- define parameters $\vec{\lambda}$
- define likelihood $p(\vec{D}\,|\,\vec{\lambda})$
- define priors $p_0(\vec{\lambda})$

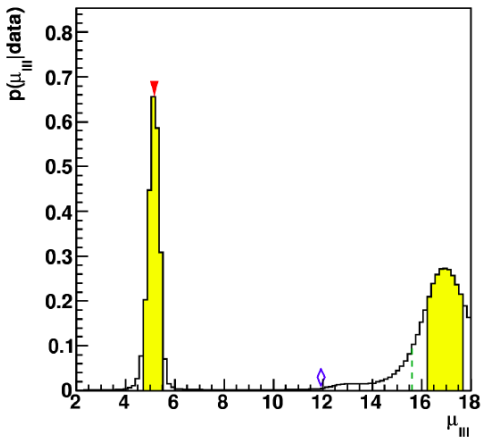
Read DATA

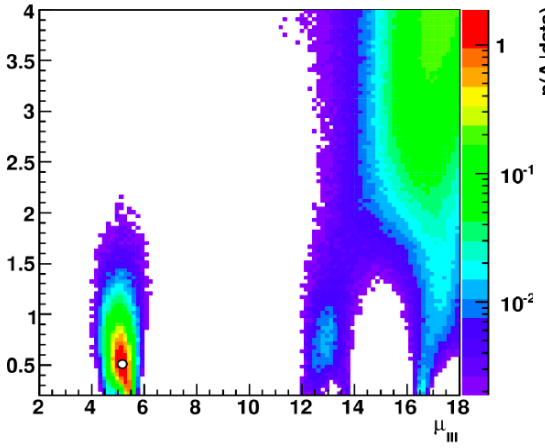
- from text file, ROOT tree, user defined
- by hand

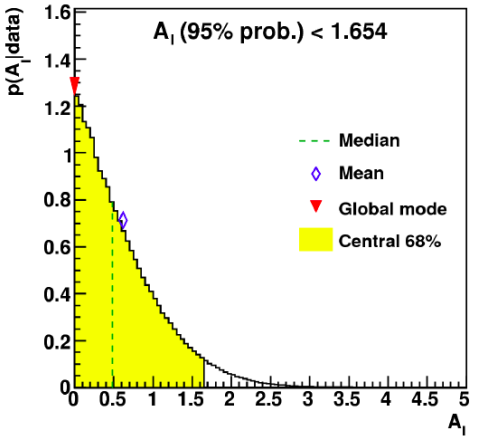
Key tool: Markov Chain Monte Carlo (MCMC)

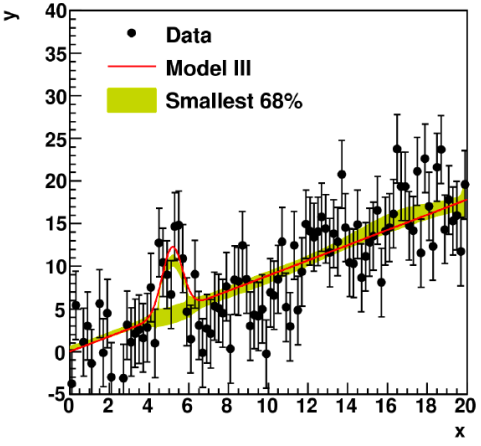
- Implemented Metropolis algorithm
  - Sample parameter space with random walk
  - Walk guided to regions of larger probability
- Calculate posterior probability for all parameters:
  - Calculate marginalized distributions wrt one or two parameters
  - Integrate out all other parameters
  - Calculate probability distribution of any function, i.e., full error propagation











Define a model class inheriting from a base model class BCModel

- add parameters:
- implement likelihood:
- implement prior:

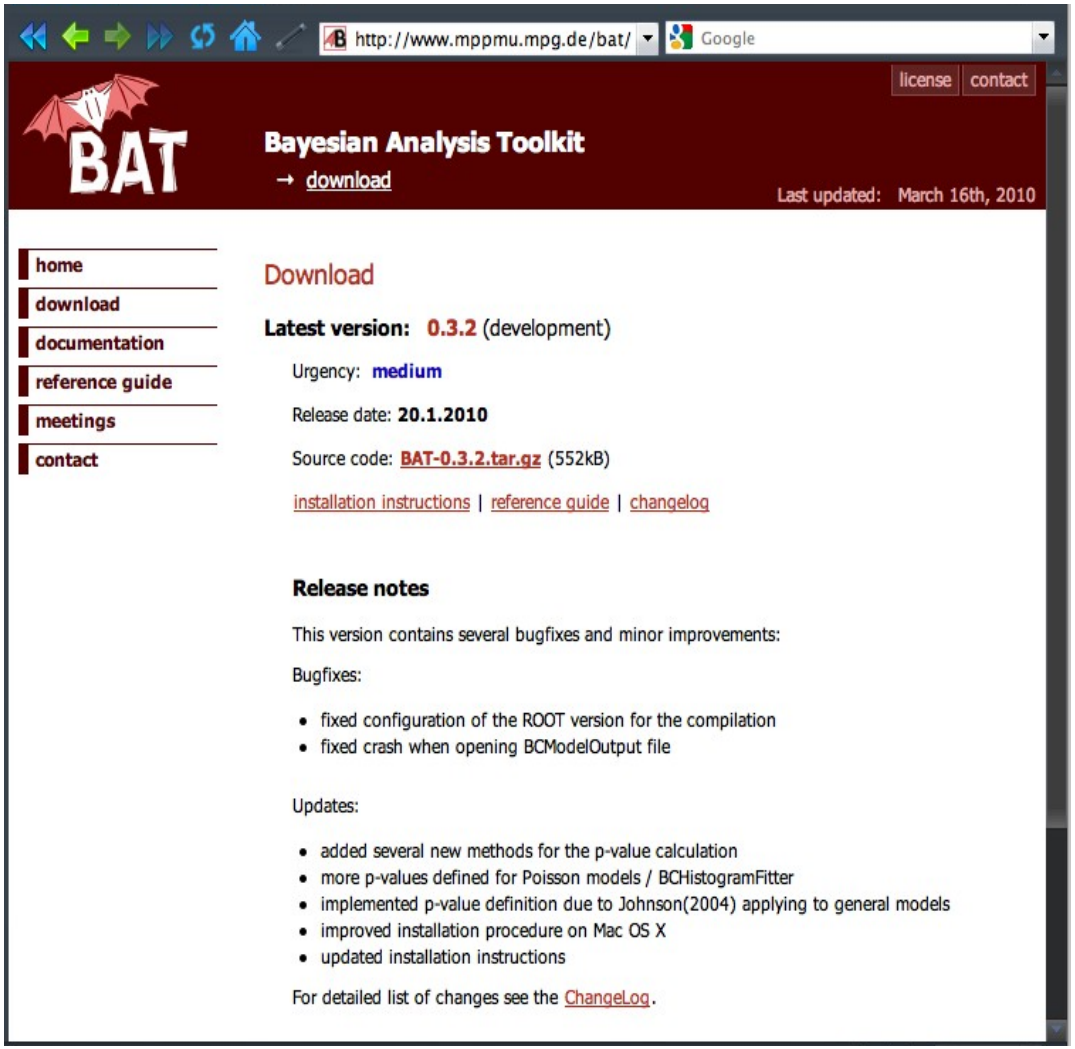
```
AddParameter("p0",-4,4);
double LogLikelihood(vector<double> params)
double LogAPrioriProbability(vector<double> params)
```

Main program/macro

- Create a model:
- read data:
- assign data to the model:
- do the analysis:

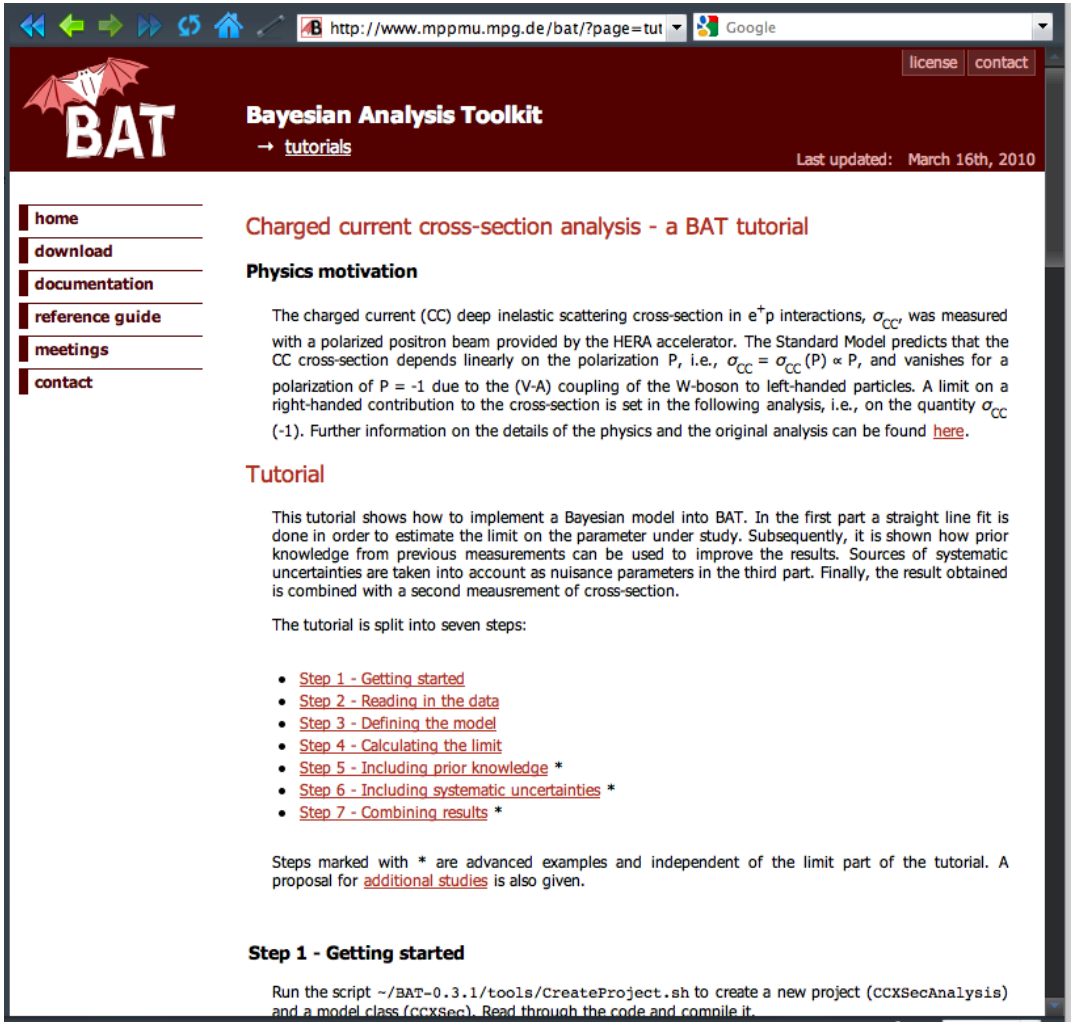
```
MyModel * mm = new MyModel("Model 0");
BCDataSet * data = new BCDataSet();
data->ReadFromFileTxt("file.txt",4);
mm->SetDataSet(data);
mm->Normalize();
mm->MarginalizeAll();
mm->FindModeMinuit( mm->GetBestFitParameters() );
mm->PrintAllMarginalized("distributions.ps");
etc.
```

- can be downloaded from: `http://www.mppmu.mpg.de/bat`
- BAT comes in form of shared library
- depends of the ROOT I/O functionality
- BAT contains at the moment 15 classes which provide:
  - main infrastructure
  - algorithms
  - output and logging
  - extension classes to solve specific (frequent) fitting problems
  - a set of well documented examples is included in BAT distribution
- good starting point “Introduction to BAT” document
- check the reference guide!



- BAT paper published: Computer Physics Communications **180** (2009) 2197-2209

- tutorials section was added to the BAT webpage
  - can be found under: `Documentation → Tutorials`
- four tutorials at the moment:
  - limit setting
  - hypothesis testing and template fitting
  - *efficiency fitting* ← today
  - *Poisson example* ←
- show basic information on how to
  - set up a model
  - calculate limits
  - define prior
  - include systematic uncertainties
  - etc.
- tutorials have form of exercises with solutions
- more tutorials will come



today