An Introduction to BAT

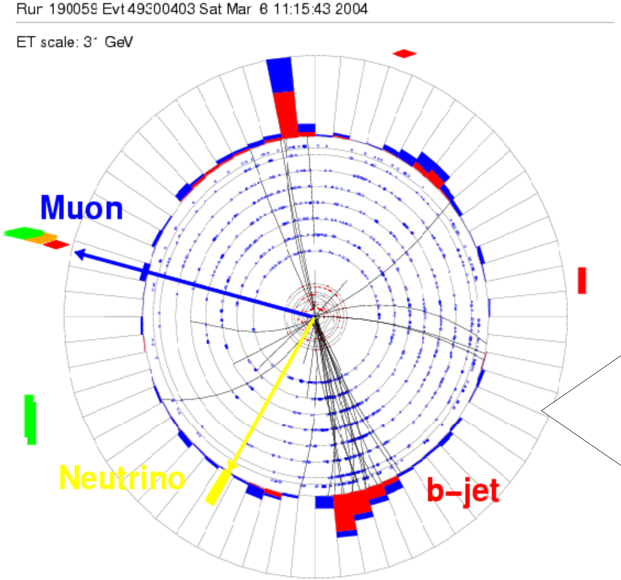
BAT Workshop Bologna 2011

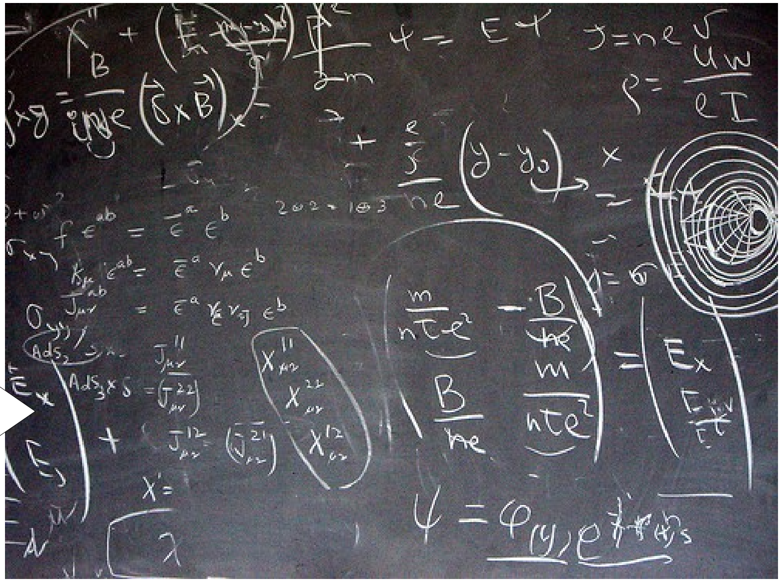
February 24th - 25th 2011

Kevin Kröninger
University of Göttingen

for the  Frederik Beaujean, Allen Caldwell, Daniel Kollar, Kevin Kröninger, Shabnaz Pashapour, Arnulf Quadt

Motivation ◦ BAT overview ◦ MCMC ◦ A working example ◦ this course ◦ summary





Experiment　　　Data analysis　　　Theory

Questions in data analysis:

- What does the data tell us about our model? — Parameter estimation
- Which model is favored by the data? — Model comparison
- Is the model compatible with the data? — Goodness-of-fit test

Need methods and tools to extract information:

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

BAYESIAN ANALYSIS TOOLKIT

Requirements

- Allow to phrase arbitrary models and data sets
- Interface to HEP software
- Estimate parameters (point estimates)
- Find probability densities (interval estimates)
- Propagate uncertainties
- Compare models
- Test validity of model against the data

Implementation:

- C++ library based on ROOT.
- Models are implemented as base classes and need to be defined by the user, or
- A set of of pre-defined models can be used.
- A set of algorithms can used to perform the actual analysis

Requirements

- Allow to phrase arbitrary models and data sets
- Interface to HEP software
- Estimate parameters (point estimates)
- Find probability densities (interval estimates)
- Propagate uncertainties
- Compare models
- Test validity of model against the data

Implementation:

- Minimization can be done via a Minuit interface or via Simulated Annealing.
- Marginalization and uncertainty estimation can be done via Markov Chain Monte Carlo (MCMC).
- Propagation of uncertainties (without Gaussian assumptions) can also be done via MCMC

Requirements

- Allow to phrase arbitrary models and data sets
- Interface to HEP software
- Estimate parameters (point estimates)
- Find probability densities (interval estimates)
- Propagate uncertainties
- Compare models
- Test validity of model against the data

Implementation:

- Direct comparison of model probabilities (Bayes factors)
- Integration methods from Cuba library linked
- Possibilities to do p-value tests

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

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 (anything)

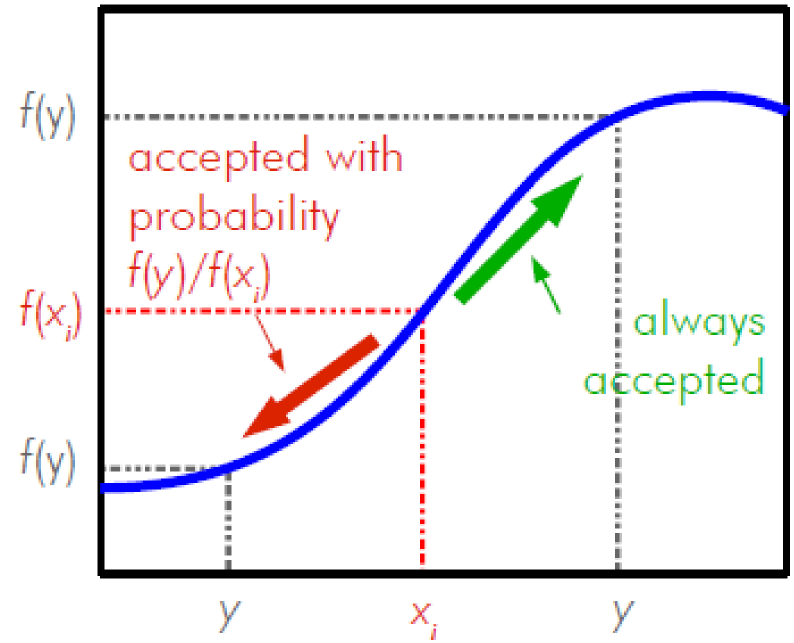
$$p(\vec{\lambda}\,|\,\vec{D}) = \frac{p(\vec{D}\,|\,\vec{\lambda})\; p_0(\vec{\lambda})}{\int p(\vec{D}\,|\,\vec{\lambda})\; p_0(\vec{\lambda})\; d\vec{\lambda}}$$

Tools:

- Point estimates:
  - Minuit
  - Simulated Annealing
  - MCMC
  - simple Monte Carlo
- Marginalization:
  - MCMC
  - simple Monte Carlo
- Integration:
  - sampled mean
  - importance sampling
  - CUBA (Vega, Suave, Divonne, Cuhre)
- Sampling:
  - simple Monte Carlo
  - MCMC

How does MCMC work?

- Output of Bayesian analyses are posterior probability densities, i.e., functions of an arbitrary number of parameters (dimensions).
- Sampling large dimensional functions is difficult.
- Idea: use random walk heading towards region of larger values (probabilities)
- Metropolis algorithm:

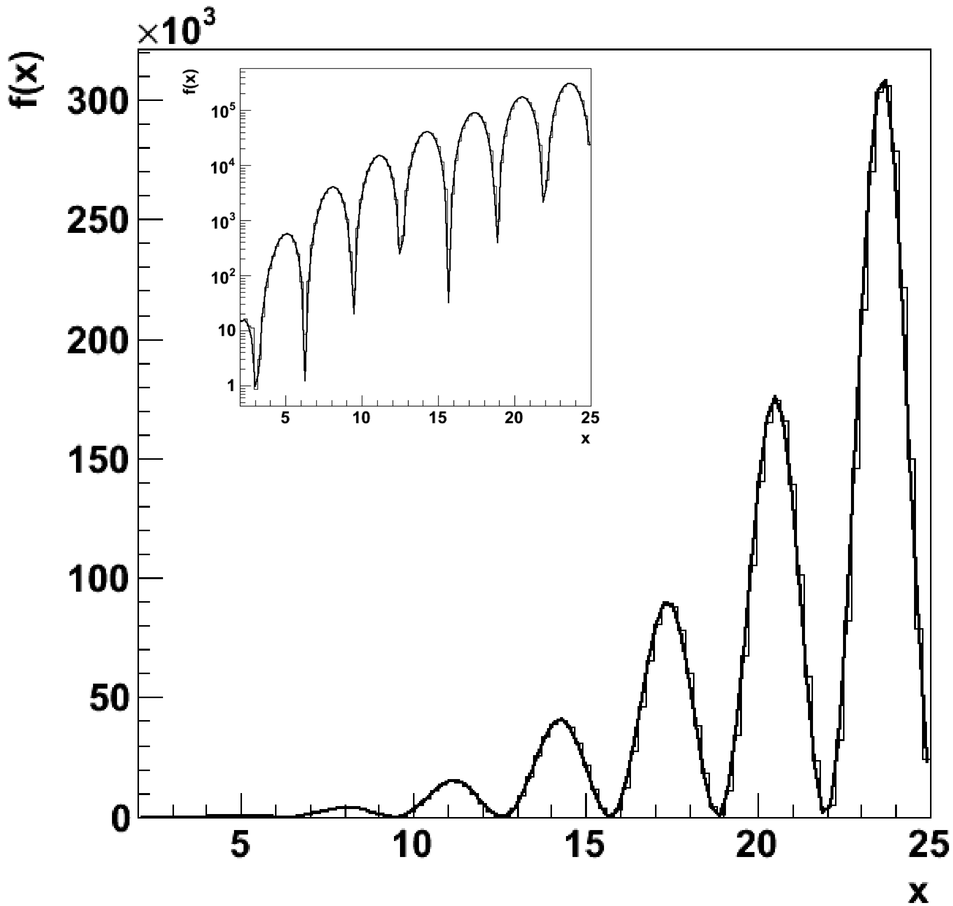


- Start at some randomly chosen x_i
- Randomly generate y around x_i
- If $f(y) \geq f(x_i)$, set $x_{i+1} = y$
- If $f(y) < f(x_i)$, set $x_{i+1} = y$ with probability $p = \frac{f(y)}{f(x_i)}$
- If y not accepted, stay where you are, i.e., set $x_{i+1} = x_i$
- Start over

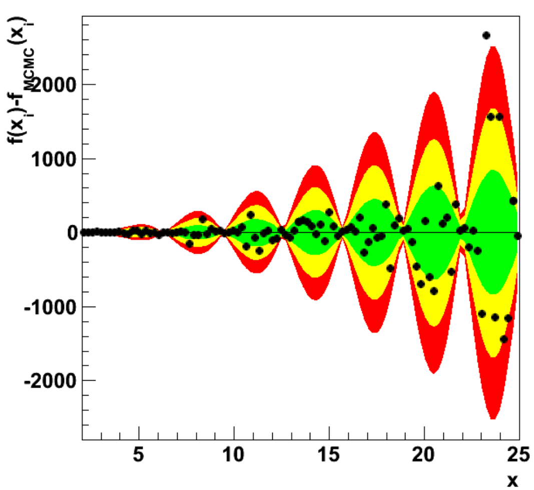
N. Metropolis et al., J. Chem. Phys. 21 (1953) 1087.

Does it work for difficult functions?

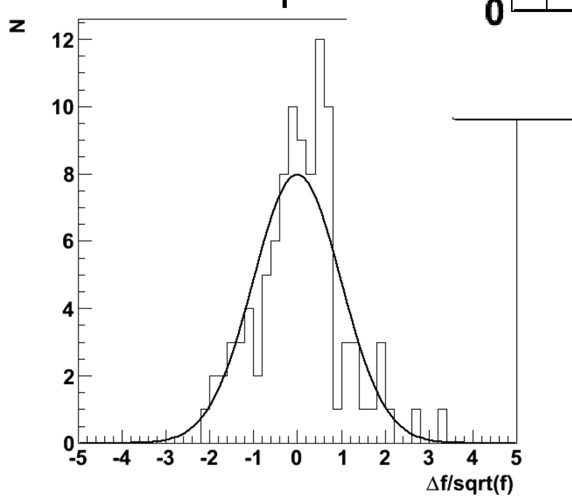
- Test MCMC on a function:
  $$f(x) = x^4 \cdot \sin(x^2)$$
- Compare MCMC distribution to analytic function
- Several minima/maxima are no problem.
- Different orders of magnitude are no problem.



chi2 vs. x



pull



For more examples, see our test suite on the BAT web page.

How does MCMC help in Bayesian inference?

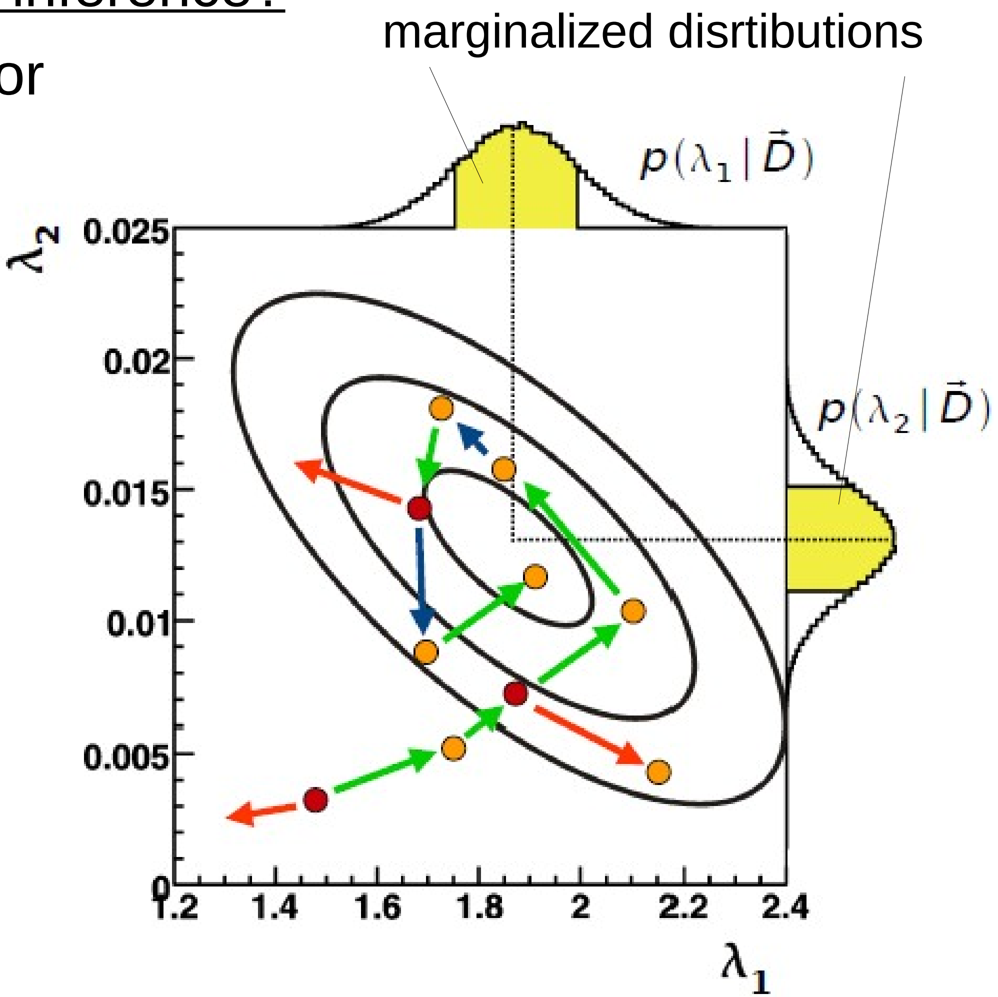
- Use MCMC to sample the posterior probability, i.e.

$$f(\vec{\lambda}) = p(\vec{D} \mid \vec{\lambda})\, p_0(\vec{\lambda})$$

- Marginalization of posterior:

$$p(\lambda_i \mid \vec{D}) = \int p(\vec{D} \mid \vec{\lambda})\, p_0(\vec{\lambda})\, d\vec{\lambda}_{j \neq i}$$

- Fill a histogram with just one coordinate while sampling
- Error propagation: calculate any function of the parameters while sampling
- Point estimate: find mode while sampling



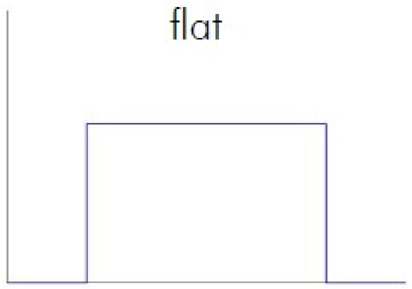
Metropolis is ~3 lines of code, fairly easy, but ...

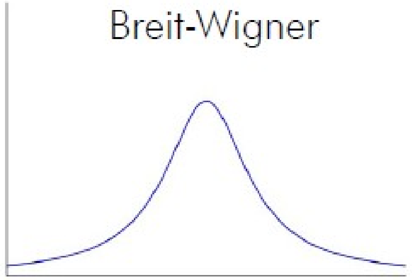
Technical details:

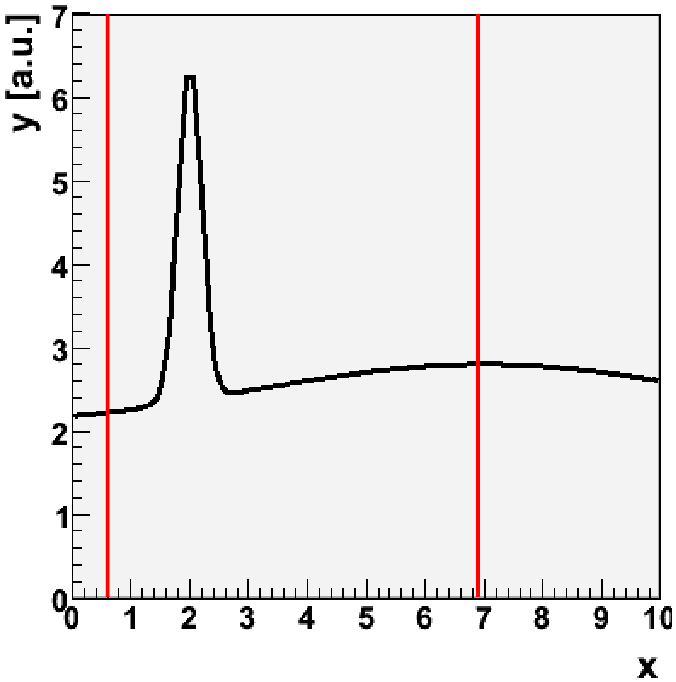
• How are the new points generated?	Proposal function
• How many points can I afford to throw away?	Efficiency
• How many iterations do we need?	Convergence criterion
• How correlated are the points?	Auto-correlation/lag

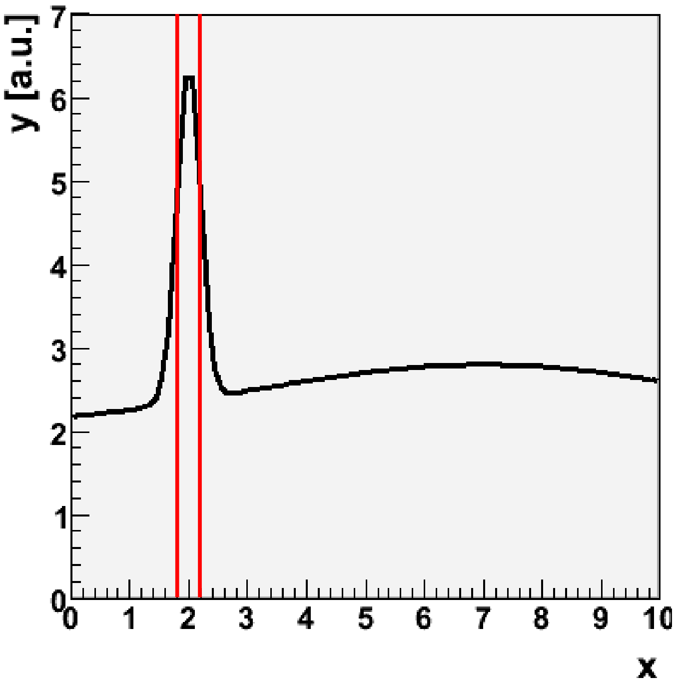
How are the new points generated?

- Proposal function: probability density for taking one step during the random walk
- Should be independent of the underlying distribution, i.e., the same everywhere
- Shape is important (default: Breit-Wigner)
- Width defines efficiency = fraction of accepted points









- Small width = large efficiency
- Large width = small efficiency
- Trade off: efficiency ~25%

How many iterations do we need?

- MCMC distribution should converge asymptotically to underlying function.
- In practice: need to stop the chain at some point. Need criteria.
- Two strategies:
  - Single chain convergence
  - Multi-chain convergence
- Single chain convergence:
  - Could monitor auto-correlation
  - Very CPU-time intensive
  - Could be done offline
- Multi-chain convergence:
  - Test convergence of multiple chains to each other
  - Use Gelman&Rubin criterion

Gelman & Rubin convergence:

- Calculate average variance of all chains

$$W = \frac{1}{m}\frac{1}{n-1}\sum_{j=1}^{m}\sum_{i=1}^{n}(x_i - \bar{x}_j)^2$$

- Estimate variance of target distribution

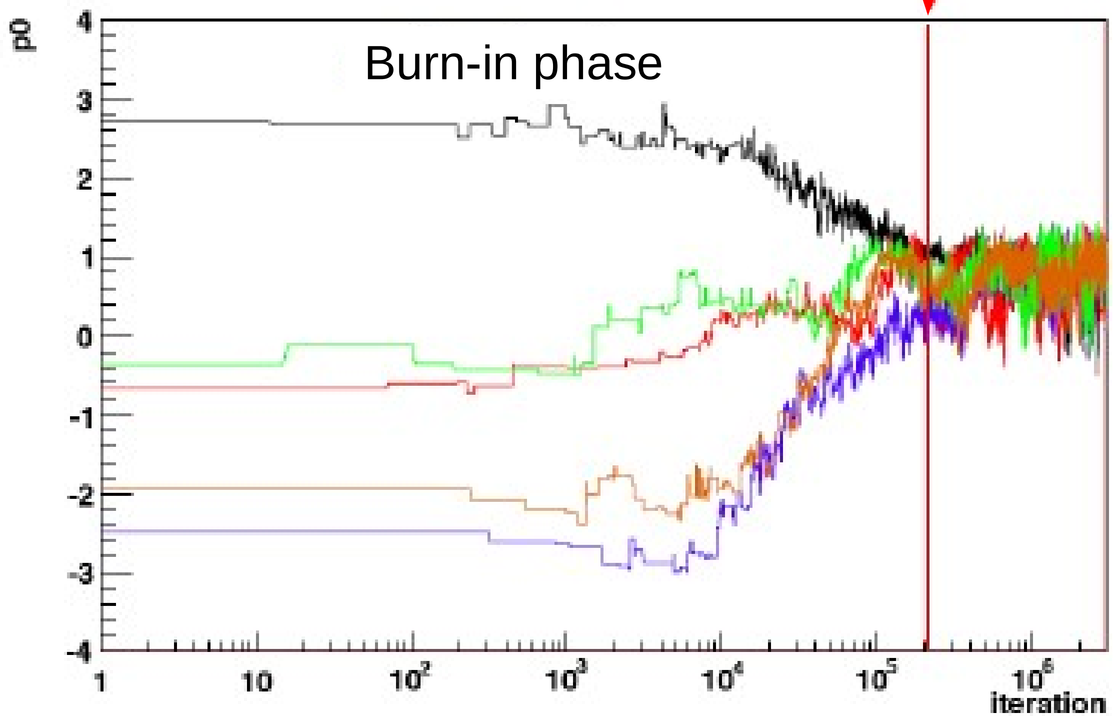
$$\hat{V} = (1-\frac{1}{n})W + \frac{1}{m-1}\sum_{j=1}^{m}(\bar{x}_j - \bar{x})^2$$

- Calculate ratio and compare with stopping criterion (relaxed version):

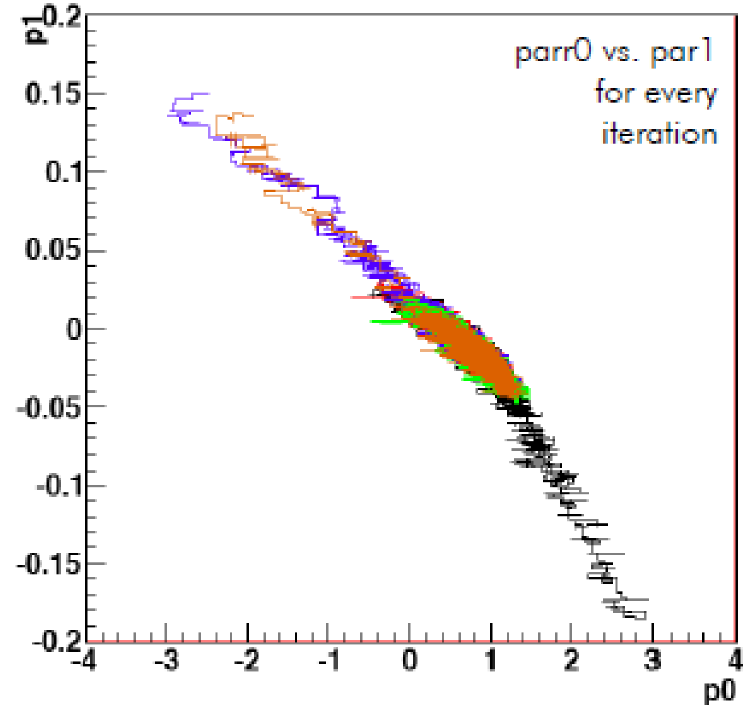
$$r = \sqrt{\frac{\hat{V}}{W}}$$ < 1.x (x = 0.1 default)

Gelman&Rubin, StatSci 7, 1992

Convergence a la Gelman & Rubin



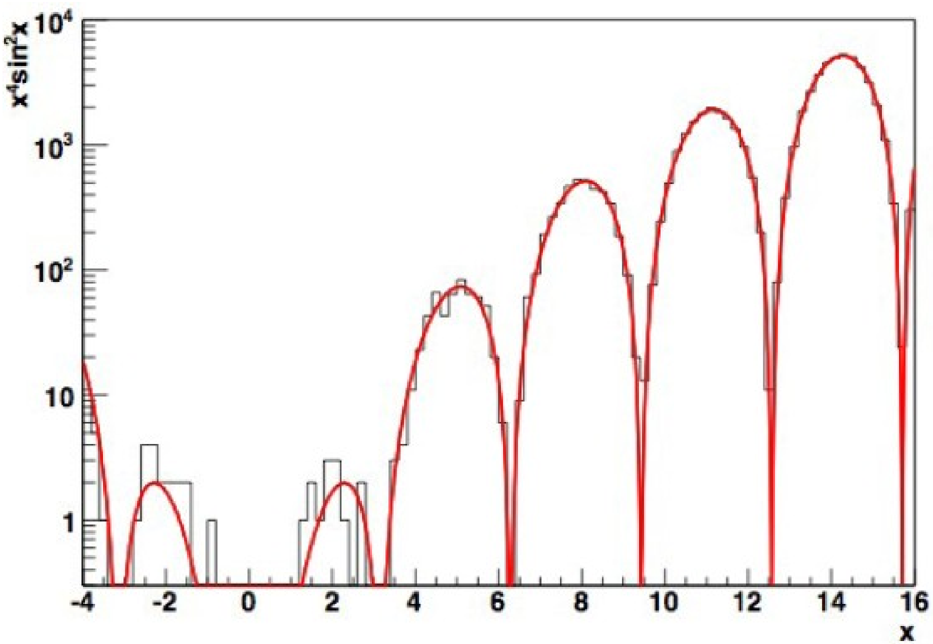
Parameter value vs. iteration

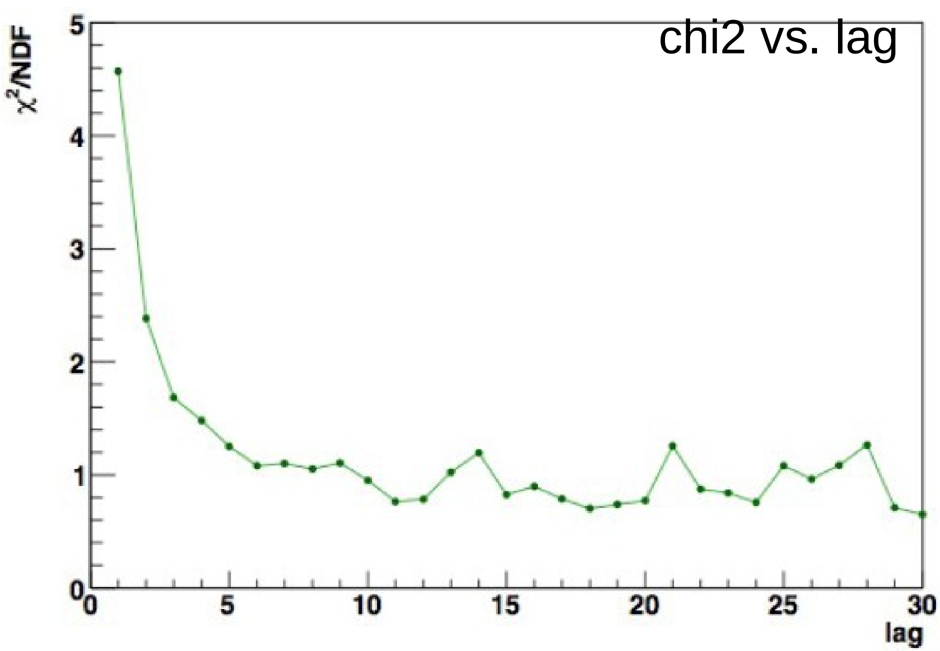


Parameter 1 vs parameter 0

How correlated are the points?

- True Monte Carlo and random walk create sets of points without (auto-correlation) while MCMC algorithm can cause auto-correlation, e.g., when rejecting a point (since the old one is taken again)
- Size of the correlation depends on the underlying posterior and the proposal function
- Can thin the MCMC sample by introducing a lag, i.e., take only every n^{th} point to calculate the marginalized distributions
- Cost: need to run a factor of *n* longer to get the same stat. precision





What exactly is being done in BAT?

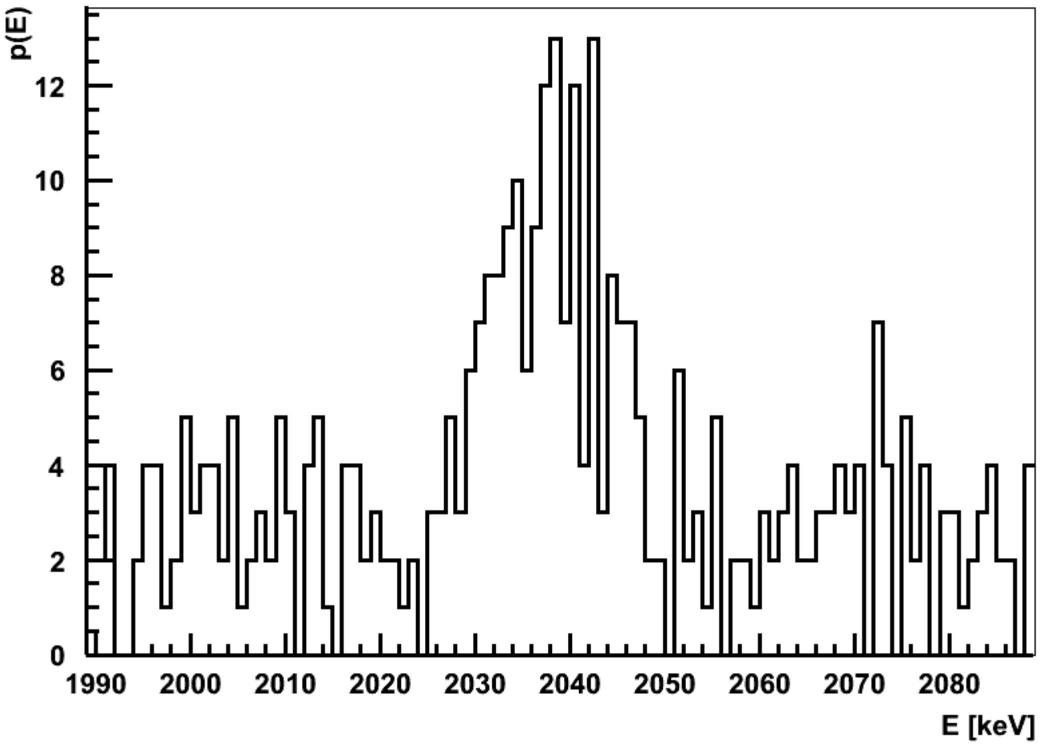
- Step 1: Starting points
  - Random within parameter space (default)
  - Center or user defined
- Step 2: Burn-in
  - Use multiple chains (default: 5)
  - Run until convergence is reached and chains are efficient
  - Or run until the maximum number of iterations is reached
  - Chains are efficient if the efficiency is between 15% and 50%
  - Run in sequences to adjust the width of the proposal functions:
    - If efficiency > 50%: increase the width
    - If efficiency < 15%: decrease the width
- Step 3: Main run
  - Use width obtained from efficiency optimization and convergence
  - Store information (next slide)

What is done in each step?

- Marginalization:
  - Fill 1-D and 2-D histograms
  - Large number: N·(N+1)/2, e.g., for N=50 there are 1275 histograms
  - Individual histograms can be switched on/off
- Optimization:
  - Search for maximum of posterior
  - Not precise, but helpful as starting point for other algorithms
- Error propagation:
  - Calculate arbitrary (user-defined) functions from parameters
- Misc:
  - Write points to ROOT tree for offline analysis
  - Perform any user-defined analysis, histogram filling, etc.

Phrasing the problem:

- Estimate signal strength of Gaussian signal on top of flat background
- Data generated with the following settings:
  - Gaussian signal:
    - position μ = 2039 keV
    - width σ = 5 keV
    - strength <S> = 100
  - Flat background:
    - strength <B> = 3/keV
- Number of events per bin fluctuate with Poisson distribution



Statistical model:

- Gaussian signal on top of flat background
- 4 (+2) fit parameters: Gauss (3) and flat (1)
  (+2 nuisance parameters for efficiency)
- Prior knowledge:
  - Background: 300 +- 173 in 100 keV (e.g., from sideband analysis)
  - Signal strength: exponentially decreasing (e.g., theoretical intuition)
  - Signal position: flat (e.g., no idea about the mass of a resonance)
  - Signal width: 5 +- 1 keV (detector resolution)
  - Signal and background efficiency fixed to 1 (in this example)
- Statistical model:
  - Bin data
  - Assume independent Poisson fluctuations in each bin

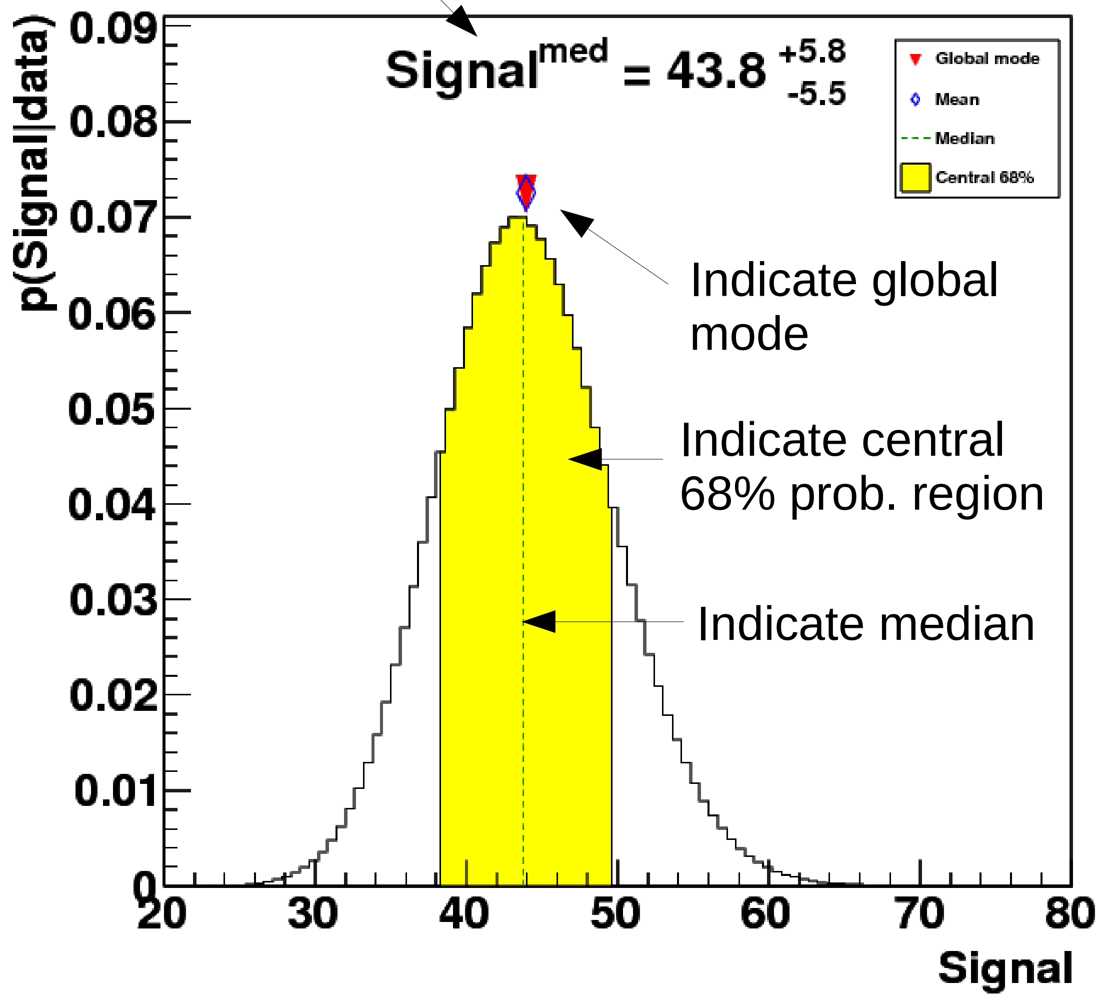
$$p(D|S,\mu,\sigma,B)=\prod_{i=1}^{N_{\text{bins}}}\frac{\lambda_i^{n_i}}{n_i!}e^{-\lambda_i}$$

$$\lambda_i=\int_{\Delta_i}\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(x-\mu)^2}{2\sigma^2}}dx+\frac{B}{\Delta_i}$$

Marginalized distributions:

- Project posterior onto one parameter axis, i.e., integrate over all other parameters
- Global mode and mode of marginalized distribution do not have to coincide
- Full (correlated) information in Markov Chain
- Default output:
    - Mean +- std. deviation
    - Median and central int.
    - Mode and smallest int.
- All 1-D and 2-D distributions are written out during main run

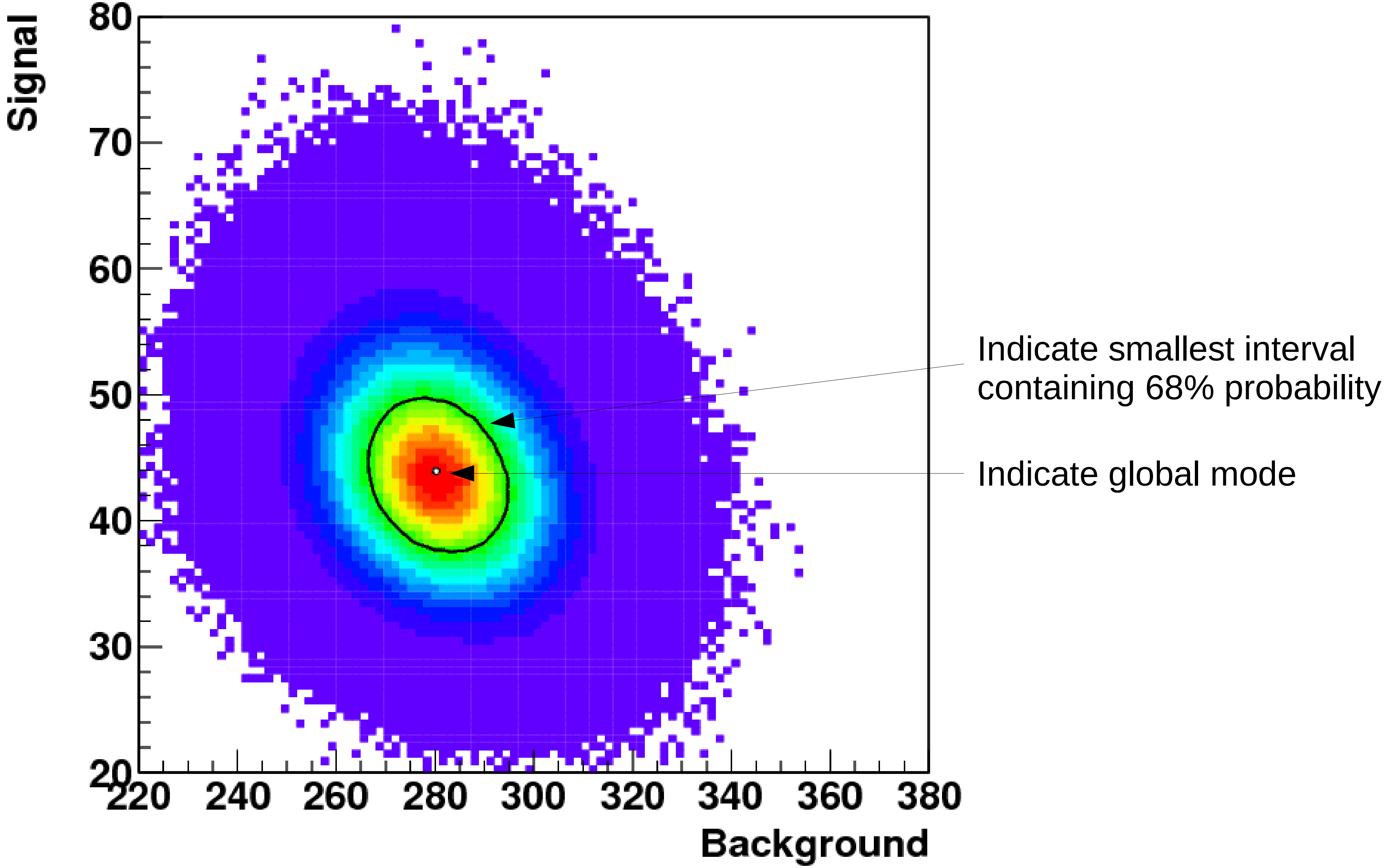
Quote median and central 68% prob. region

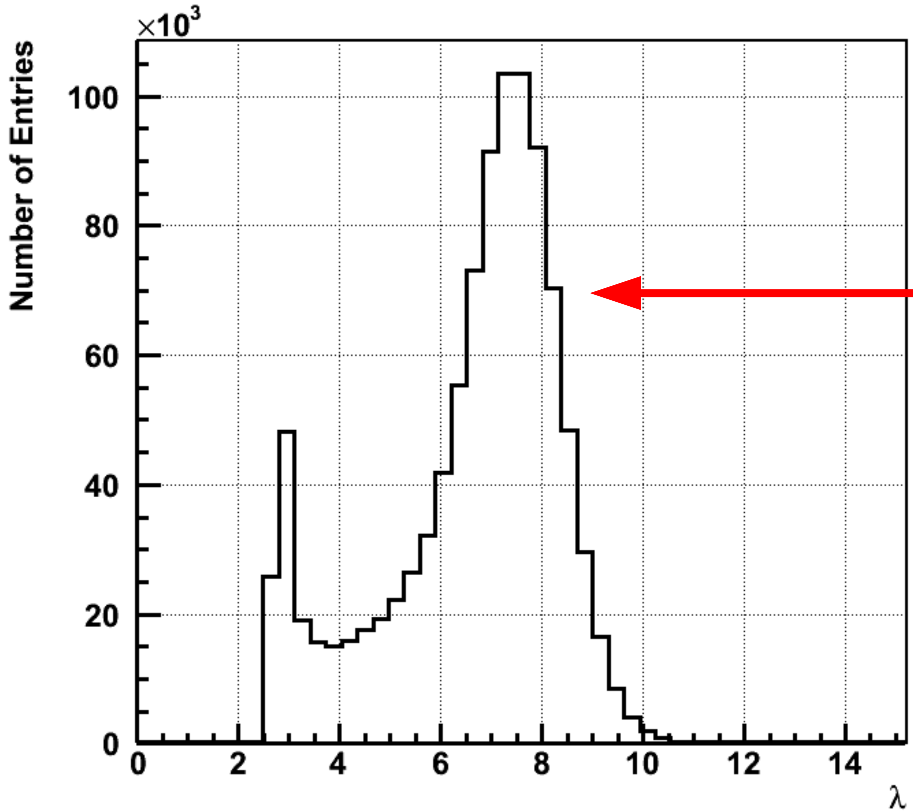


Indicate global mode

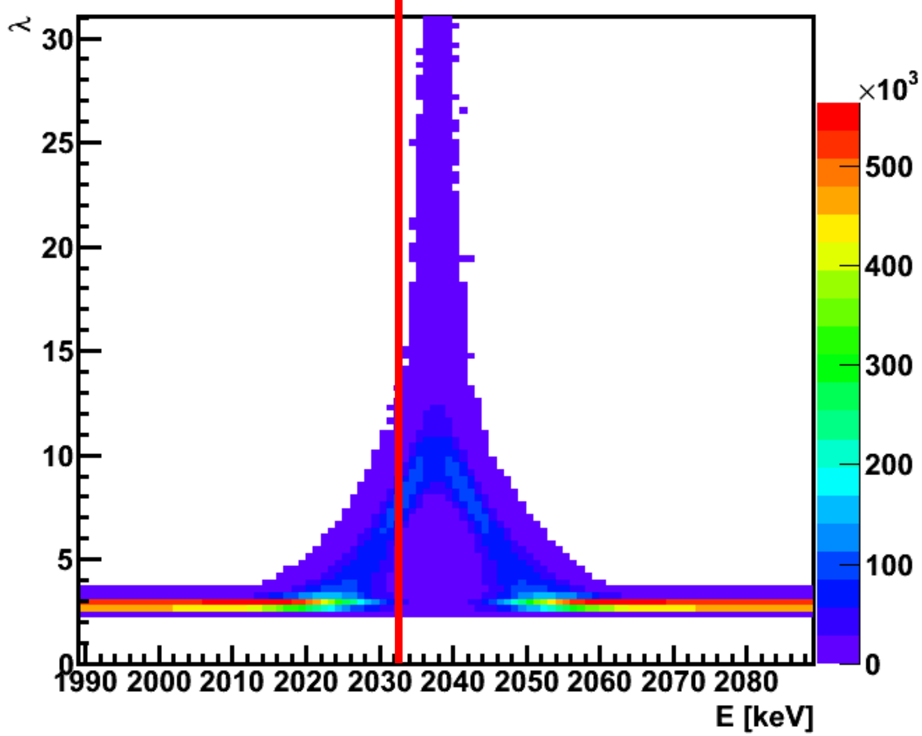
Indicate central 68% prob. region

Indicate median


Signal
80
70
60
50
40
30
20
220
240
260
280
300
320
340
360
380
Background
Indicate smallest interval containing 68% probability
Indicate global mode

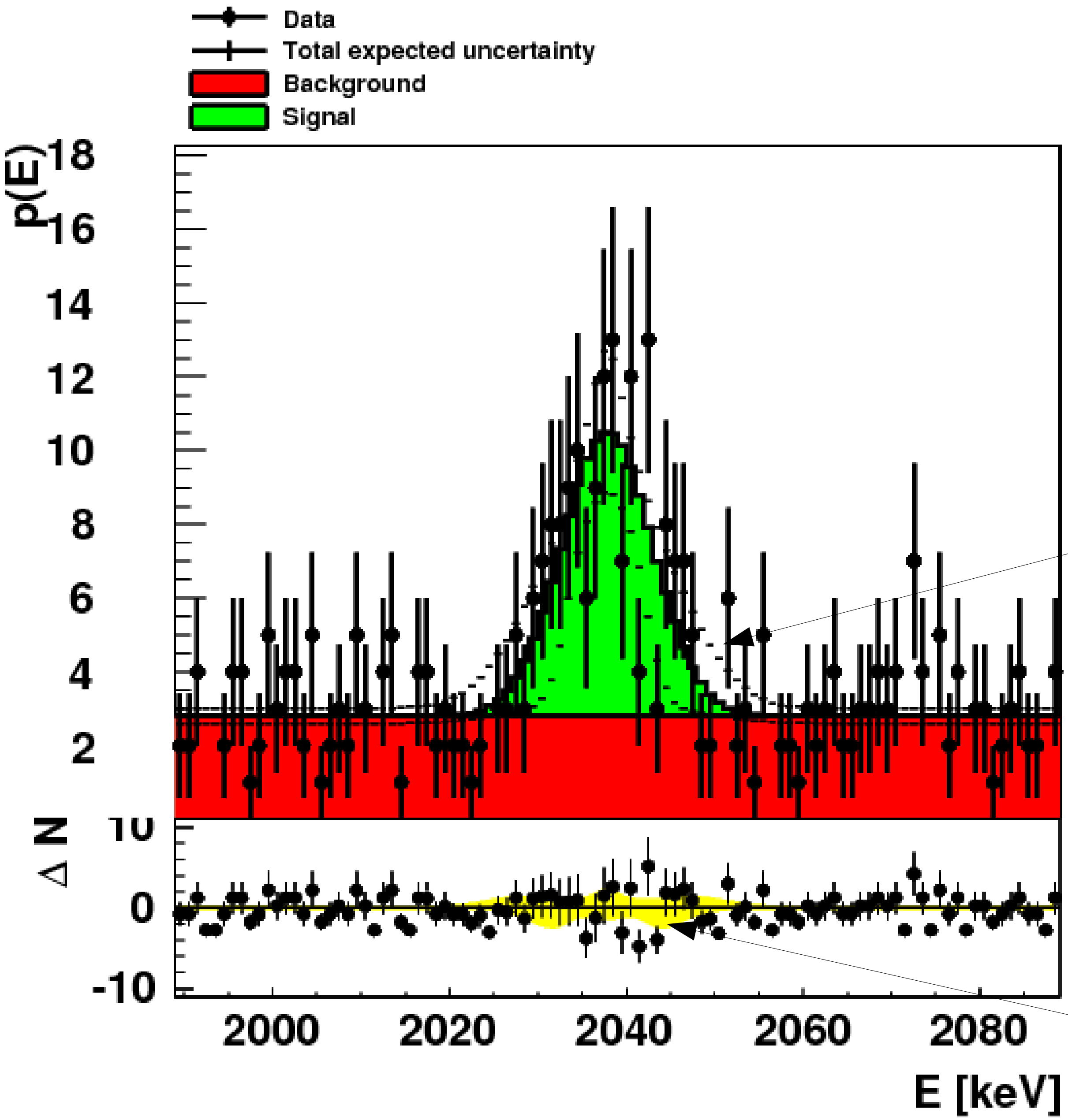


Posterior probability for the number of expected events with energy E=2032 keV



Sum of all possible fit functions weighted with posterior: calculate fit function at energy E for all parameter values

Use as error band



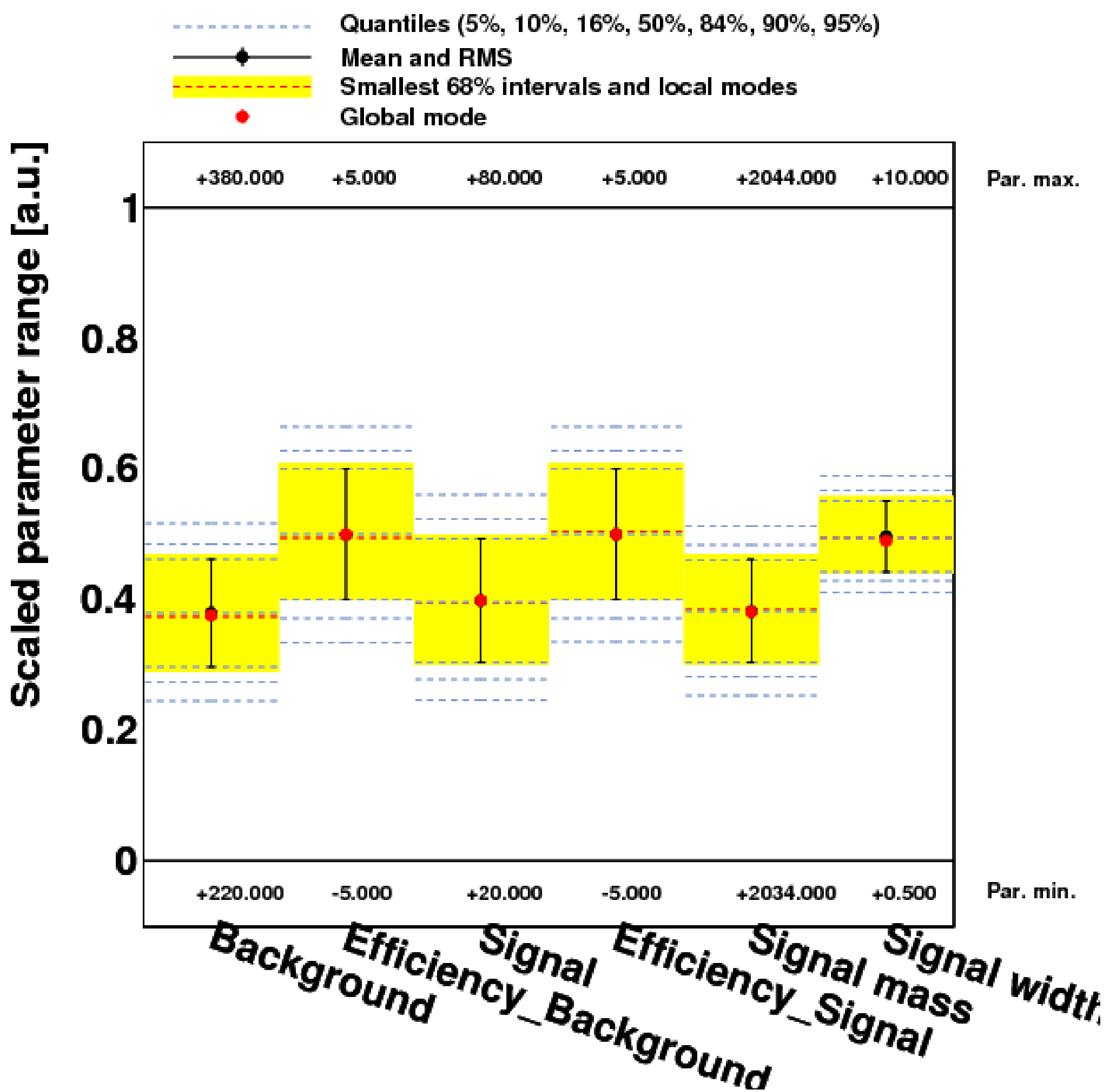
Uncertainty of prediction

Error band

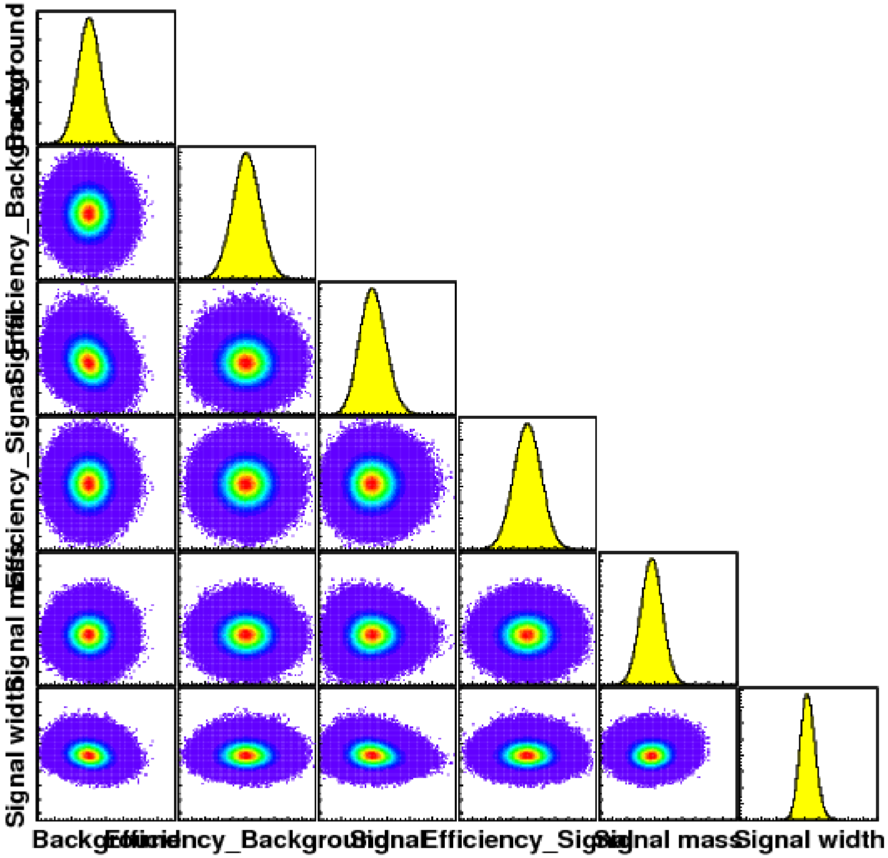
A working example: text output

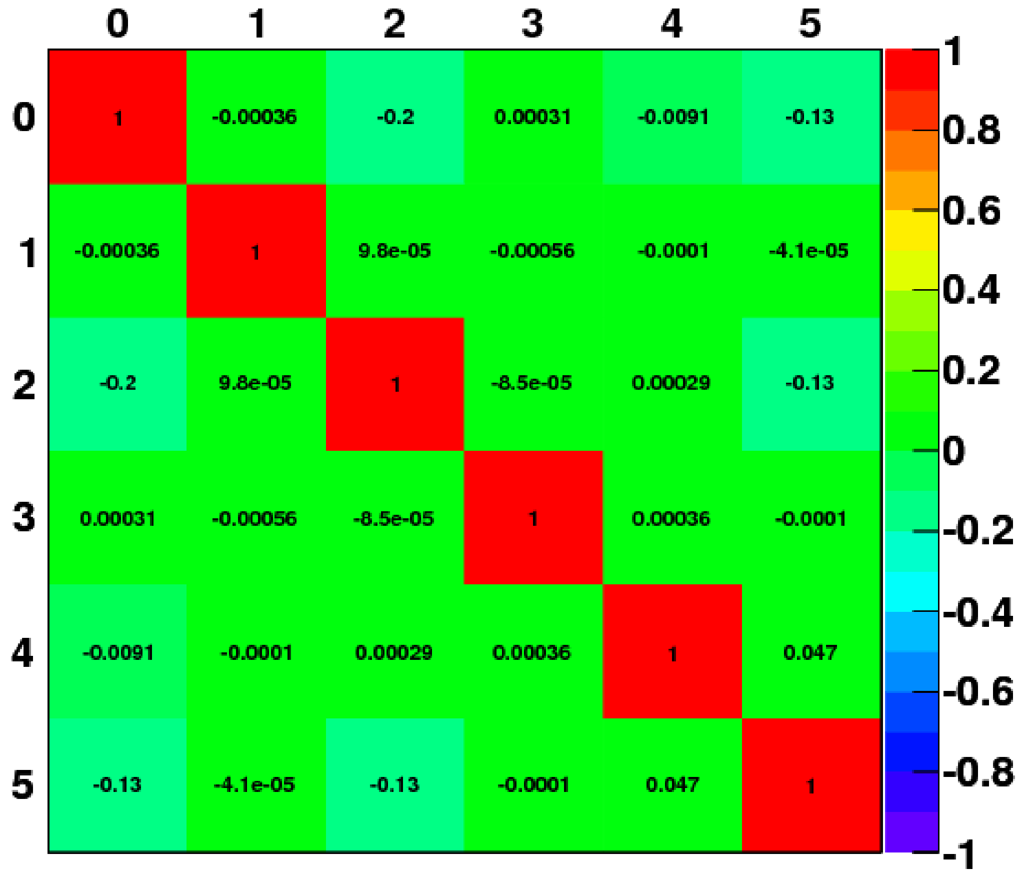
```
Results of the marginalization
===============================
List of parameters and properties of the marginalized
distributions:
 (0) Parameter "Background":
     Mean +- sqrt(V):                280.8 +- 13.16
     Median +- central 68% interval: 280.7 +  13.2 - 13.02
     (Marginalized) mode:            280
      5% quantile:                   259.2
     10% quantile:                   263.9
     16% quantile:                   267.7
     84% quantile:                   294.4
     90% quantile:                   297.7
     95% quantile:                   302.6
     Smallest interval(s) containing 68% and local modes:
      (266.4, 295.2) (local mode at 280 with rel. height 1; rel. area 0.6978)

 (2) Parameter "Signal":
     Mean +- sqrt(V):                43.94 +- 5.724
     Median +- central 68% interval: 43.78 +  5.849 - 5.532
     (Marginalized) mode:            43.7
      5% quantile:                   34.8
     10% quantile:                   36.71
     16% quantile:                   38.25
     84% quantile:                   49.88
     90% quantile:                   51.38
     95% quantile:                   53.62
     Smallest interval(s) containing 68% and local modes:
      (38, 50) (local mode at 43.7 with rel. height 1; rel. area 0.6821)

 (4) Parameter "Signal mass":
     Mean +- sqrt(V):                2038 +- 0.7871
     Median +- central 68% interval: 2038 +  0.7806 - 0.7781
     (Marginalized) mode:            2038
      5% quantile:                   2037
     10% quantile:                   2037
     16% quantile:                   2037
     84% quantile:                   2039
     90% quantile:                   2039
     95% quantile:                   2039
     Smallest interval(s) containing 68% and local modes:
      (2037, 2039) (local mode at 2038 with rel. height 1; rel. area 0.693)
...
```

```
Results of the optimization
============================
Optimization algorithm used:Metropolis MCMC
List of parameters and global mode:
 (0) Parameter "Background": 280.2 +- 13.08
 (2) Parameter "Signal": 43.94 +- 5.674
 (4) Parameter "Signal mass": 2038 +- 0.7652
 (5) Parameter "Signal width": 5.159 +- 0.5012

Status of the MCMC
==================
Convergence reached:                    yes
Number of iterations until convergence: 24000
Number of chains:                       10
Number of iterations per chain:         10000000
Average efficiencies:
 (0) Parameter "Background": 20.03%
 (2) Parameter "Signal": 17.35%
 (4) Parameter "Signal mass": 24.52%
 (5) Parameter "Signal width": 19.56%
```

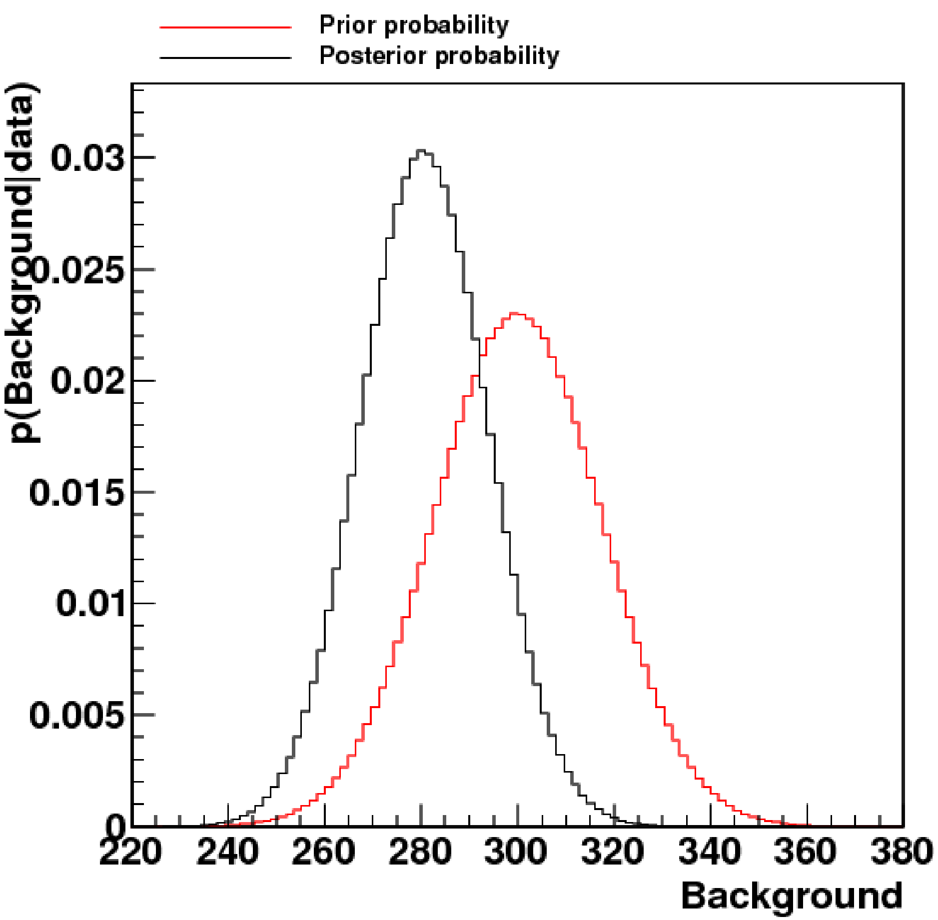


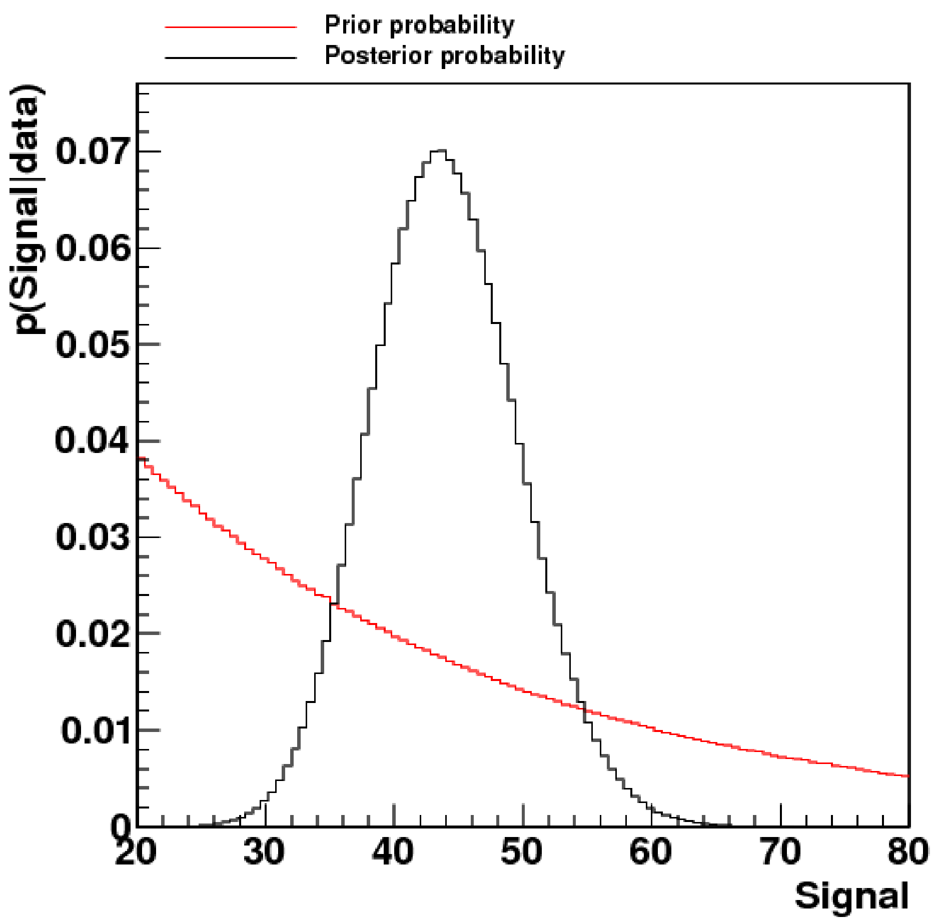
Quantiles (5%, 10%, 16%, 50%, 84%, 90%, 95%)
Mean and RMS
Smallest 68% intervals and local modes
Global mode
Scaled parameter range [a.u.]
+380.000
+5.000
+80.000
+5.000
+2044.000
+10.000
Par. max.
1
0.8
0.6
0.4
0.2
0
+220.000
-5.000
+20.000
-5.000
+2034.000
+0.500
Par. min.
Background
Efficiency_Background
Signal
Efficiency_Signal
Signal mass
Signal width

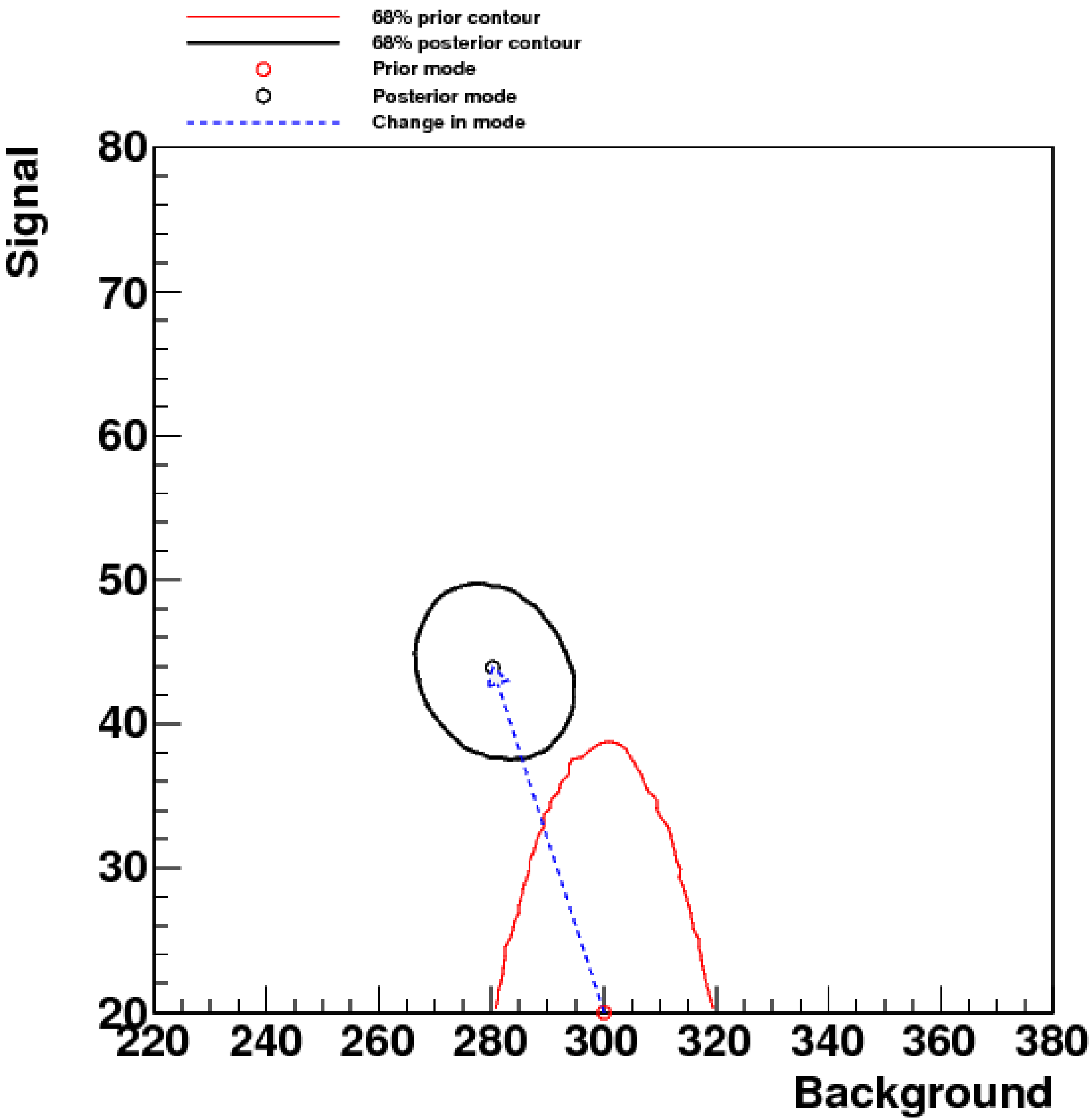


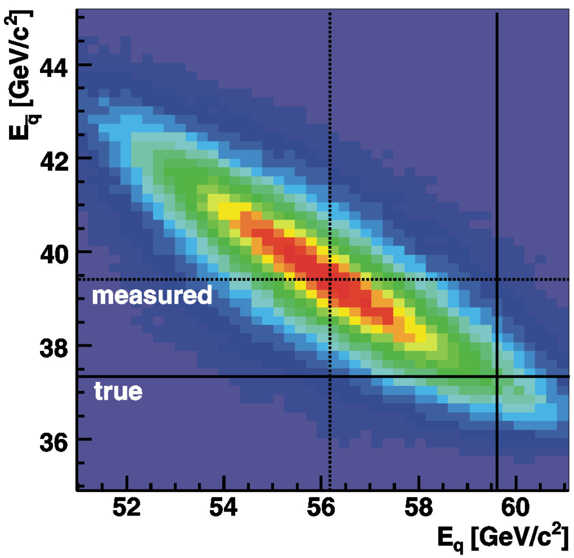

0 1 2 3 4 5
0: 1, -0.00036, -0.2, 0.00031, -0.0091, -0.13
1: -0.00036, 1, 9.8e-05, -0.00056, -0.0001, -4.1e-05
2: -0.2, 9.8e-05, 1, -8.5e-05, 0.00029, -0.13
3: 0.00031, -0.00056, -8.5e-05, 1, 0.00036, -0.0001
4: -0.0091, -0.0001, 0.00029, 0.00036, 1, 0.047
5: -0.13, -4.1e-05, -0.13, -0.0001, 0.047, 1
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1

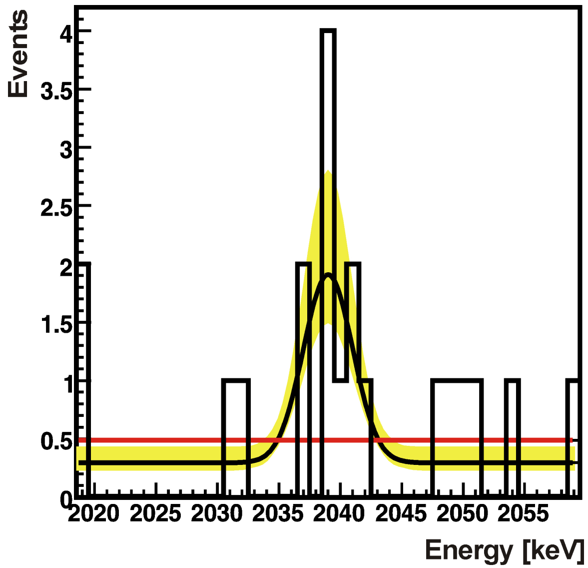
A working example: knowledge update

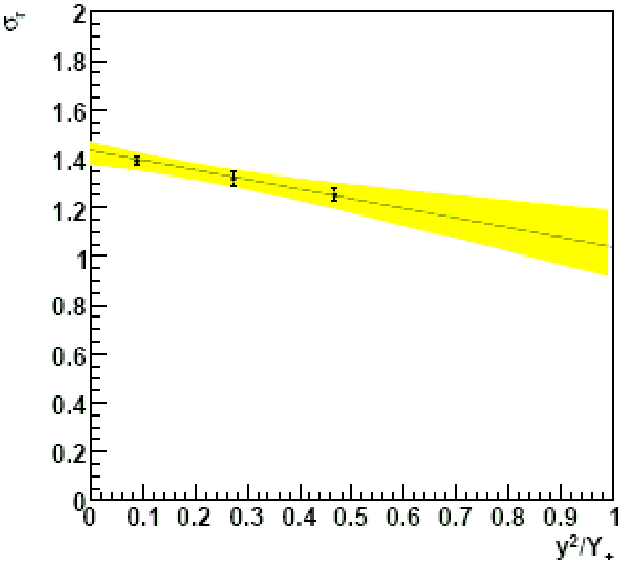





68% prior contour
68% posterior contour
Prior mode
Posterior mode
Change in mode
Signal
80
70
60
50
40
30
20
220
240
260
280
300
320
340
360
380
Background







ATLAS:
Example of kinematic fitting in top quark decays

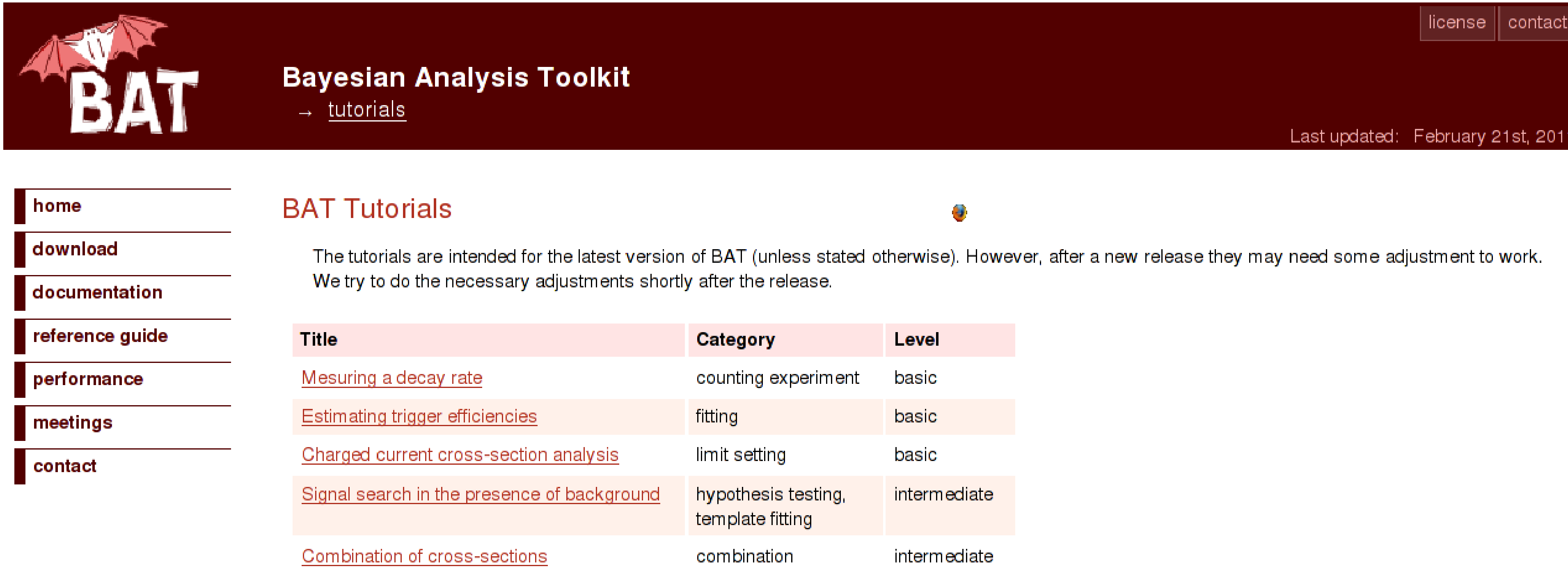
GERDA:
Fitting signal on top of a background

ZEUS:
Extraction of the longitudinal structure function

Contact:

- Web page: http://www.mppmu.mpg.de/bat/
- Contact: bat@mppmu.mpg.de
- Paper on BAT:

  A. Caldwell, D. Kollar, K. Kröninger, BAT - The Bayesian Analysis Toolkit
  Comp. Phys. Comm. 180 (2009) 2197-2209 [arXiv:0808.2552].



license contact

Bayesian Analysis Toolkit

→ tutorials

Last updated: February 21st, 2011

- home
- download
- documentation
- reference guide
- performance
- meetings
- contact

BAT Tutorials

The tutorials are intended for the latest version of BAT (unless stated otherwise). However, after a new release they may need some adjustment to work. We try to do the necessary adjustments shortly after the release.

Title	Category	Level
Mesuring a decay rate	counting experiment	basic
Estimating trigger efficiencies	fitting	basic
Charged current cross-section analysis	limit setting	basic
Signal search in the presence of background	hypothesis testing, template fitting	intermediate
Combination of cross-sections	combination	intermediate

Tutorials:

- Quite a few on the web
- Our program here:
  - Couting experiment
  - Charged-current cross-section analysis
  - Using BAT for searches

Summary:

- Bayesian inference requires some computational effort (e.g., nuisance parameters)
- Markov Chain Monte Carlo is the key tool to solve these issues
- BAT is a tool to combine Bayesian inference with MCMC
- Toolbox with more algorithms (integration, optimization, etc.)
- C++ library, modular, easy to use
- Informative output with predefined plots, numbers, etc.
- Did not talk about:
  - Hypothesis testing and goodness-of-fit
  - p-values
  - Bayes factors, information criteria
  - ...
- Upgrade of BAT ongoing, more to come
- Participation and feedback are always welcome