Hands on introduction to BAT

Statistics Tools School
7 Apr 2011

*Julia Grebenyuk for the BAT team**

**Allen Caldwell, Daniel Kollar, Kevin Kroeninger, Shabnaz Pashapour*
Special thanks to Frederik Beaujean and Fabian Kohn

Tutorial on radioactive decay rate

Goal: learn how to use BAT on a simple example

- Let's consider measuring the *decay rate of a radioactive isotope* in presence of background
- Two measurements:
  - One without radioactive source to measure background
  - One with the source
  - Duration: T = 100s each
    - N_1 = 100 – number of background counts only
    - N_2 = 110 – number of counts including the source

Data Set	Run Time	Events
1	100	100
2	100	110

Decay rate

Decay rate of the isotope: $N = N_0 e^{-t/\tau}$ $\frac{dN}{dt} = -\frac{N}{\tau}$

Total rate = signal rate + background rate: $R=R_S+R_B$

Measured for the time T, observed N_1 and N_2 events

Assume $R=R_S+\mathbf{R}_B$ constant

Learn about probable values of R_S Bayes' Theorem

Bayes' Theorem

Posterior ~ Likelihood x Prior

Number of events N, in a time T follows a Poisson distribution → the probability of the data (likelihood) is:

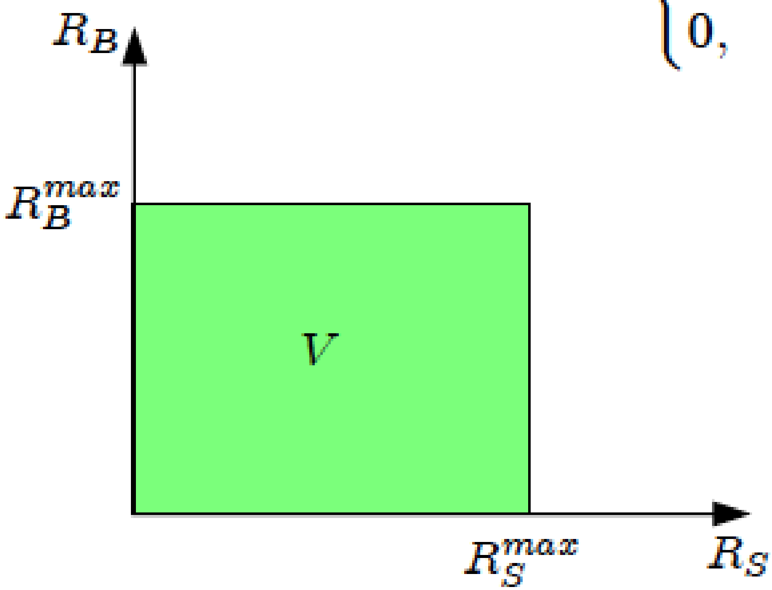
$$P\left(N_1|R_B\right) = \frac{e^{-n_B} \cdot (n_B)^{N_1}}{N_1!} \qquad n_B = R_B \cdot T$$

$$P\left(N_2|R_B, R_S\right) = \frac{e^{-n_{S+B}} \cdot (n_{S+B})^{N_2}}{N_2!} \qquad n_{S+B} = (R_S + R_B) \cdot T$$

Prior

Simplest choice: flat prior in a box $V \equiv [0, R_S^{max}] \times [0, R_B^{max}]$

$$
\begin{aligned}
P(R_S, R_B) &= P(R_S)\,P(R_B) \\
&= \begin{cases} \frac{1}{R_S^{max}} \cdot \frac{1}{R_B^{max}}, & (R_S, R_B) \in V \\ 0, & \text{else} \end{cases}
\end{aligned}
$$



Combining the measurements

1. Estimate R_B using the first measurement, then add second measurement and estimate R_s and R_B

$$P\left(R_B|N_1\right) \propto P\left(N_1|R_B\right) \cdot P\left(R_B\right)$$

$$P\left(R_B, R_S|N_1, N_2\right) \propto P\left(N_2|R_B, R_S\right) \cdot P\left(R_S\right) P\left(R_B|N_1\right)$$

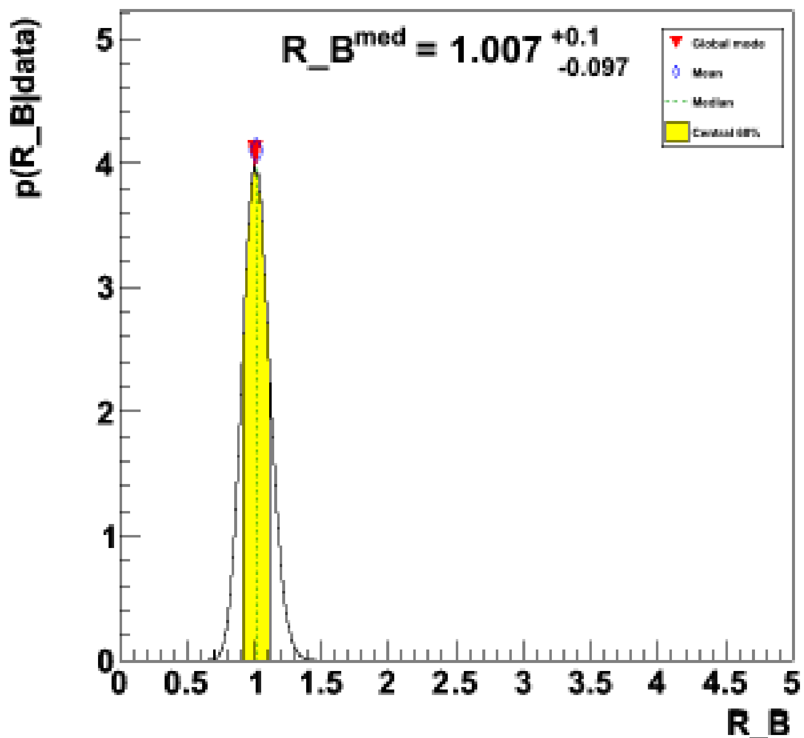
2. Use both measurements together

$$P\left(R_B, R_S|N_1, N_2\right) \propto P\left(N_2|R_B, R_S\right) P\left(N_1|R_B\right) \cdot P\left(R_S\right) P\left(R_B\right)$$

Results using the first measurement

R_B obtained using background measurement only

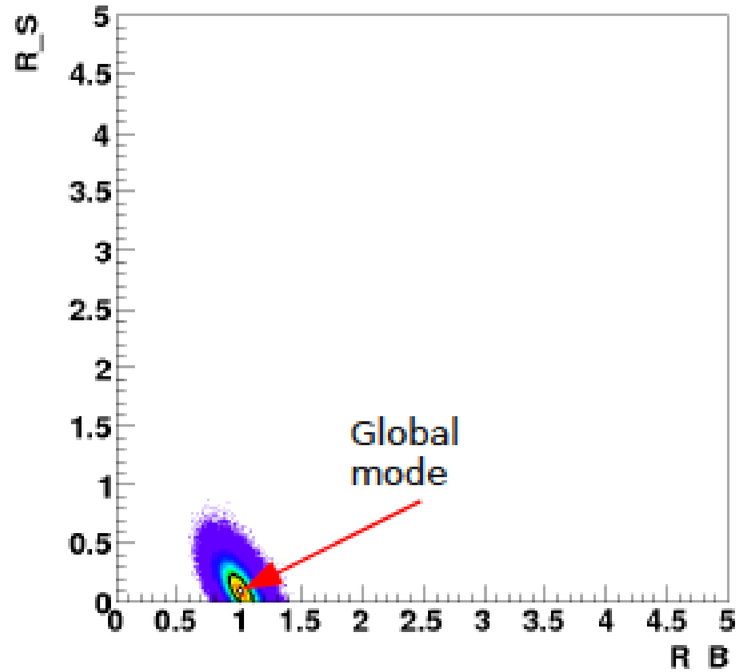
$$P\left(R_B|N_1\right) \propto P\left(N_1|R_B\right) \cdot P\left(R_B\right)$$



Results using two measurements

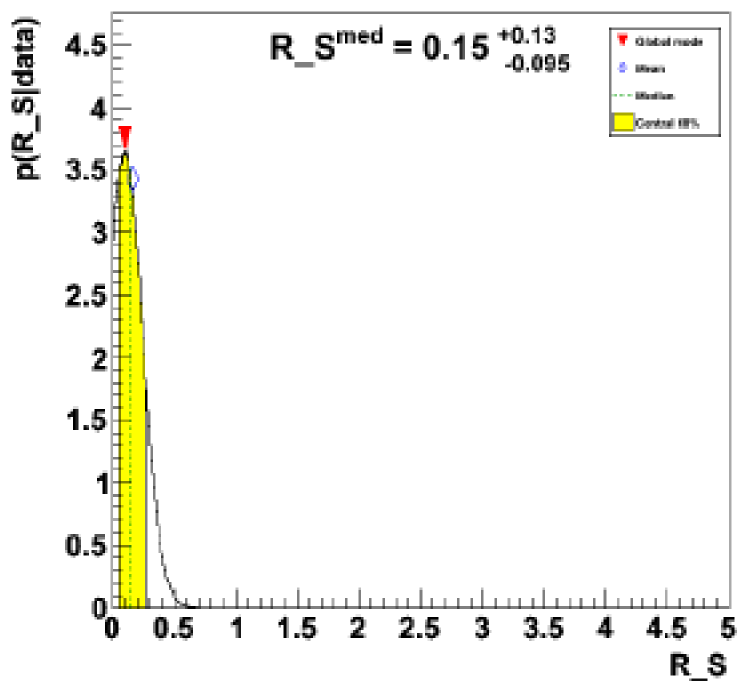
R_B and R_s obtained using two measurements

$$P\left(R_B, R_S | N_1, N_2\right)$$



...
List of parameters and global mode:
(0) Parameter "R_B": 1 +- 0.09998
(1) Parameter "R_S": 0.1 +- 0.1439
...

$$P\left(R_S | N_1, N_2\right) = \int \mathrm{d}R_B \; P\left(R_S, R_B | N_1, N_2\right)$$



...
Parameter "R_S"
Mean +- sqrt(V): 0.1609 +- 0.1071
Median +- central 68% interval: 0.1456 + 0.1251 - 0.09516
(Marginalized) mode: 0.1175

Where to find the tutorial

BAT homepage:

http://www.mppmu.mpg.de/bat/

Navigate to:

Documentation → Tutorials → Counting experiment

Direct link:

http://www.mppmu.mpg.de/bat/?page=tutorials&name=counting_experiment

Steps

Step 1 - Compiling your first BAT program

Step 2 - Fitting the background-only model

Step 3 - Including the signal contribution

Step 4 – Further steps

Step 1: create the project

- → On your Virtual Machine go to: `/statistics-school/BAT-0.4.2/`
- → Navigate to `tools` subdirectory
- → Run the script `CreateProject.sh` to create a project named `CountingExp`
- → Have a look at the generated C++ classes and compile the code with `make`

Information about the data sets and details of the run goes in `runCountingExp.cxx`

Information about the model and prior goes in `CountingExp.cxx`

Step 1: create the project

- → On your Virtual Machine go to: `/statistics-school/BAT-0.4.2/`
- → Navigate to `tools` subdirectory
- → Run the script `CreateProject.sh` to create a project named `CountingExp`
- → Have a look at the generated C++ classes and compile the code with `make`

Information about the data sets and details of the run goes in `runCountingExp.cxx`

Information about the model and prior goes in `CountingExp.cxx`

Step 2: Fitting background-only model

- Create a data point, add it to a data set and register the data set with the model
- Define the parameter R_B and add it to the model
- Define the log likelihood for the Poisson process with parameter R_B. The natural logarithm of the factorial is provided by BCMath::LogFact(int n). One can also use the approximation provided by BCMath::ApproxLogFact(int n) which is much faster for large numbers.
- Use a flat prior for R_B
- Start to sample from the posterior using the Markov chain
- Find the mode of the posterior
- Save the results of the fit in text form and create a plot of the (marginal) posterior distribution

Step 3: Include the second measurement

- → Add a second data point, N_2, to the data set
- → Include the second parameter R_S with flat prior in the model
- → Update the likelihood to incorporate N_2 and R_S
- → Plot the marginal distributions and compare the values of mean, median and mode for the individual parameters. What is the correlation between R_B and R_S?
- → Extract the 95% limit on R_S and save the plot

Step 4: Further steps

→ Redo step 2 and 3. Save $P(R_B|N_1)$ and $P(R_B|N_1,N_2)$ as a ROOT TH1D histogram. Limit R_B to the range [0,2] and use more bins (500 instead of the default 100) to store the marginalized distribution.

→ Normalize and plot the two histograms.

→ Measure the time it takes to run the program.

→ Modify LogAPrioriProbability to do nothing else than returning zero. This amounts to setting the prior to 1. Compare execution time.

→ Multiplying the likelihood by a constant just affects the normalization, but not the values of mode, mean... Thus remove all terms that are added to LogLikelihood and which are independent of R_B, R_S. You should observe that running the program takes only about a quarter of the time compared to 3.

→ Redo step 2, but now use the Reference prior (=Jeffrey's prior here) for R_B which reads $P(R_B) \propto 1/\sqrt{R_B}$. Does the posterior $P(R_B|N_1)$ change significantly?