MCMC Gibbs sampling for number of events under the source γ peaks

Mariam Ali

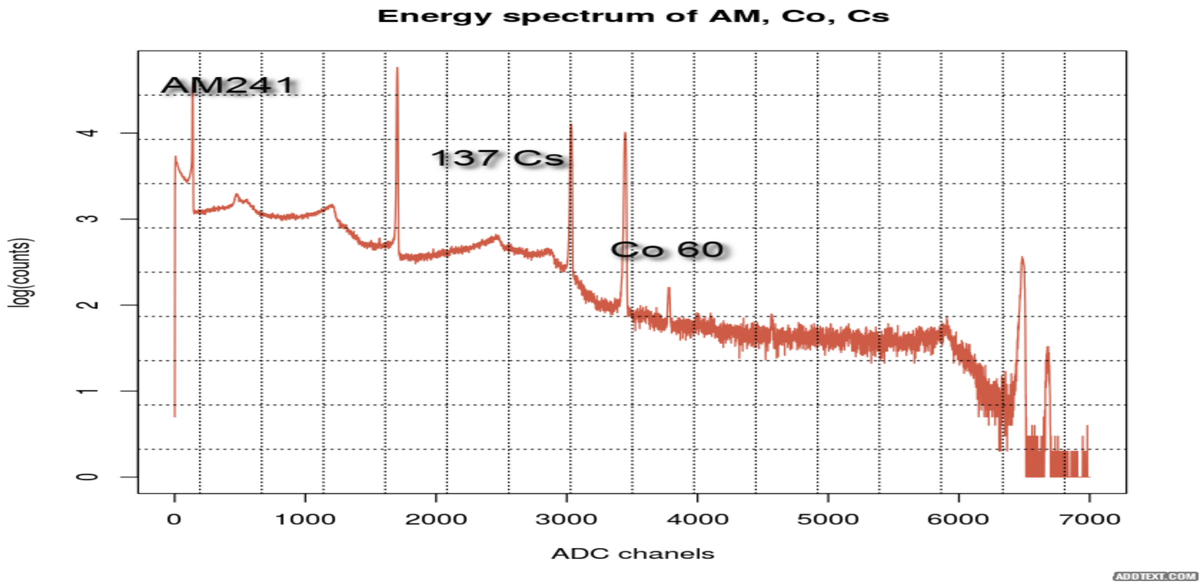
Advanced Statistics for physics Analysis

17/07/2019

Outline

The Data

The data used for this Inference problem is collected by a Germanium detector that's detecting the radiation of three chemical elements: Americium, Cesium and Cobalt .
The nuclides of each element release photons at different energies as shown in the following plot.



Problem and Solution

The problem of this project:
is to figure out the number of events occurring under a gamma peak, in other words how many photons release happen for each element.
The Solution:
Since spectral line shape describes the form of a feature, observed in spectroscopy, corresponding to an energy change in an atom, molecule or ion. Ideal line shapes include Lorentzian, Gaussian and Voigt functions. That's why I am solving this problem using bayesian inference with a normal distribution after the normalization of the data.

Methods

Markov Chain Monte Carlo is an inferring method that takes data input, prior and the likelihood that the data is most likely to follow $p(X|\theta)$ and samples the posterior of that likelihood and prior based on the so called metropolis ratio that's equal to 1 for symmetric distribution like the one I'm using in this Analysis.

$$p(\theta|X) \propto p(X|\theta) * p(\theta) \tag{1}$$

the following is the equation corresponding to the metropolis ratio:

$$\rho = \frac{p(s) * p(\theta|s)}{p(\theta) * p(s|\theta)} \tag{2}$$

Computational Methods

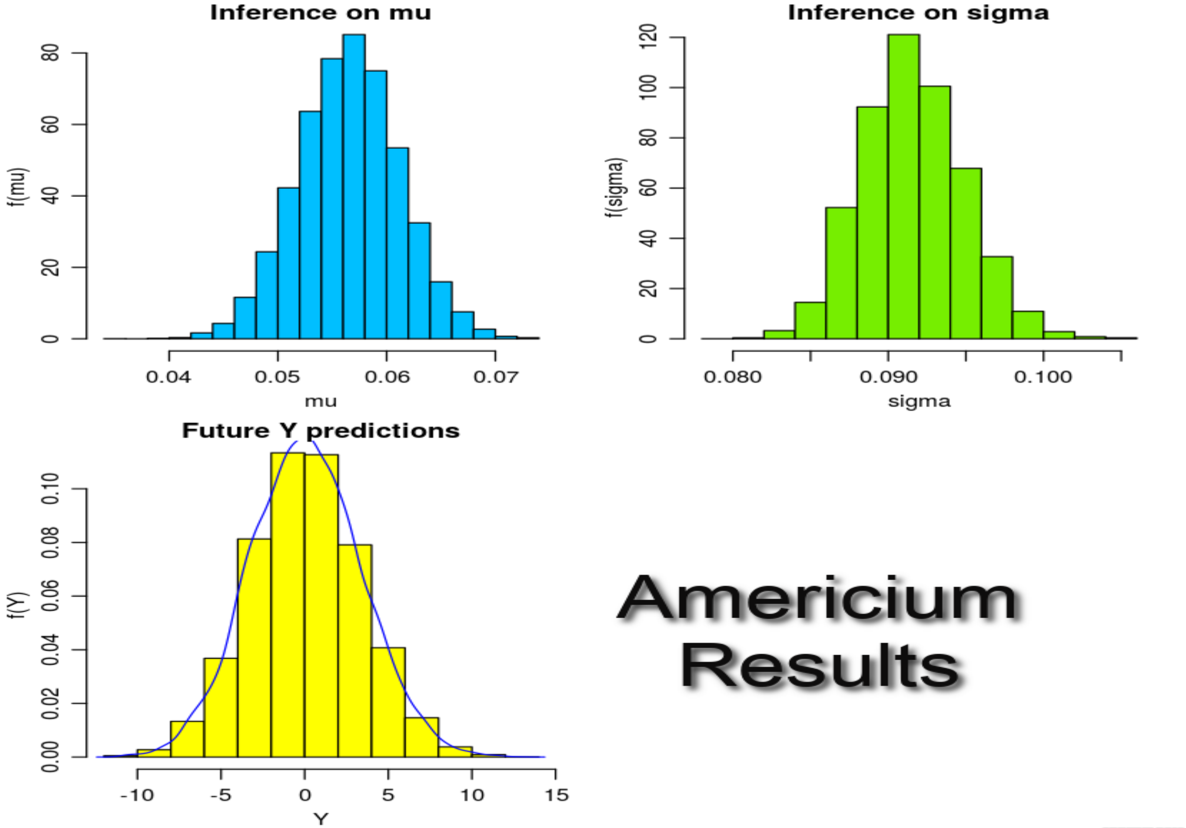
The computational Methods I used to conduct the sampling is the Marcov chain Monte Carlo with gibbs sampling which is similar method to the metropolis hasting but involves generating a multi-dimensional Markov chain by splitting the vector of random variables representing the model parameters into subvectors and sampling each subvector in turn, conditional on the most recent values of all other elements of the parameters. The library supporting this method on R is ('rjags') and the model for the likelihood would be a normal distribution.

Analysis steps

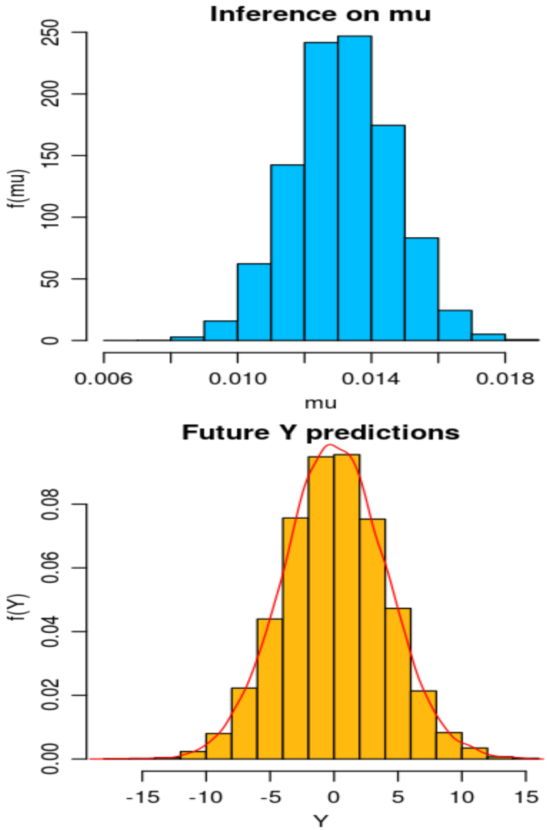
- Cut the data around the peak of each element.
- Normalize the data, since I will be using normal model having a distribution summing up to one, I prefer having the data normalized, i.e all points of the data are divided by the maximum point.
- define the observed data which is a normal distribution having the same mean and standard deviation of the real data.
- the model, define a model using BUGS (Bayesian Inference Using Gibbs Sampling): simple normal model
- Run the model and extract distribution parameters and predictions of the count.
- find density of of the predictions and calculate the number of events to be the area under the density curve.
- model the background.

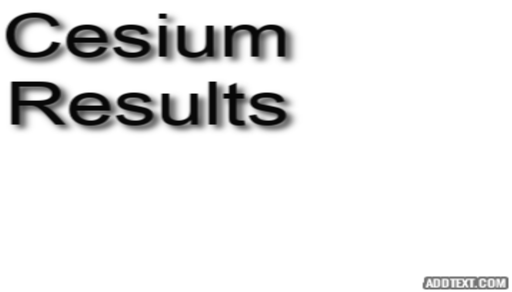
Results

Americium

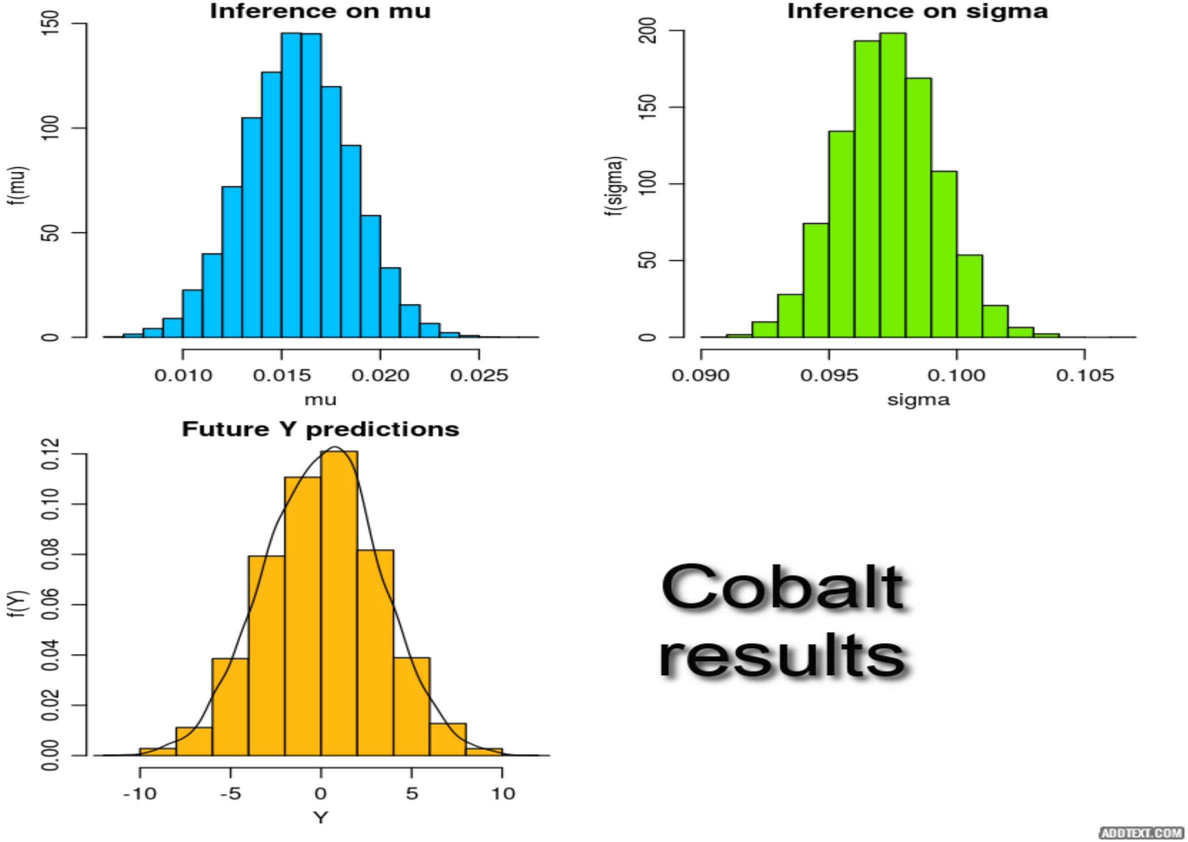


Cesium


Inference on mu
f(mu)
0
50
100
150
200
250
0.006
0.010
0.014
0.018
mu
Inference on sigma
f(sigma)
0
100
200
300
0.060
0.062
0.064
0.066
sigma
Future Y predictions
f(Y)
0.00
0.02
0.04
0.06
0.08
-15
-10
-5
0
5
10
15
Y


Cesium
Results
ADDTEXT.COM

Cobalt



Inference on mu
f(mu)
0
50
100
150
0.010
0.015
0.020
0.025
mu
Inference on sigma
f(sigma)
0
50
100
150
200
0.090
0.095
0.100
0.105
sigma
Future Y predictions
f(Y)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
-10
-5
0
5
10
Y
Cobalt results
ADDTEXT.COM

Number of events of each element

Element	Number of events
Am	33263.47
Cs	10071.73
Co	9025.953

Of course these value could change with each run due to the randomness of the generation method of the output.

Importance

In my opinion there is two major results out of this analysis.

- A way of finding the number of counts of for a given element energy spectrum, knowing the values of the ADC.
- for an anonymous dataset we can know the corresponding energy for each peak, the corresponding number of events, and the elements detected if the number of peaks is limited enough.

Further Work

- Infer the number of events using the model without the need of integrating with an inference problem of four parameters not three.
- Apply the code on an unknown dataset, apply a more complex code to test for all the available peaks and figure out if you can decide what are the elements detected in the dataset.