

Fission2019-Unified-Analysis-Framework

- A global unified analysis framework to analyze experiment Fission2019 in RIBLL1 in LANZHOU
- This framework is designed for Fission2019 data Analysis
Contributor: Fenhai Guan(1)
(1) gfh16@mails.tsinghua.edu.cn
- Personal information could be found at the link:
<http://inspirehep.net/author/profile/Fen.Hai.Guan.1>
- The whole framework is available at the link:
<https://github.com/gfh16/Fission2019-Unified-Analysis-Framework>

目录

- Step1. 数据转换
 - 1.1 将原始数据二进制文件转换成.root文件
 - 1.2 将RawRoot数据转换成MapRoot数据
- Step2. 数据质检
 - 2.1 SetBranchAddress方法
 - 2.2 TTreeReader方法
- Step3. PPAC数据处理
- Step4. SSD数据处理
 - 4.1 SSD Energy Calibration
 - 4.2 Hit Multiplicity
 - 4.3 CsI Energy Calibration
 - 4.4 Hit Pixellation
 - 4.5 Particle Identification
- Step5. 物理分析

Step1. 数据转换

1.1 将原始数据二进制文件转换成.root文件

- 修改文件: /RIBLLVMEDAQ/Raw2ROOT.cpp
- 目的: 将原始数据(二进制文件)转换成.root文件

- **操作:** 批量转换

- (1) 输入文件(原始文件)都在 /vmedata/文件夹下
- (2) 在 /vmedata/中 添加 listfilename 文件, 将需要格式转换的原始文件名一一写出, 每个文件名占一行
- (3) 修改 /RIBLLVMEDAQ/下的 Raw2ROOT.cpp文件: 目的是修改输出文件的位置输出文件(.root文件)都在 /rootdata/文件夹下 (原因: 个人更倾向于将输出文件放在/rootdata/文件夹下)
- (4) 编译成功后, 执行 ./raw2roo.sh listfilename

```
$ make
```

```
$ ./raw2roo.sh listfilename
```

1.2 将RawRoot数据转换成MapRoot数据

- **修改文件:** /RIBLLVMEDAQ/ReadRootFile2D.cpp
- **目的:** 第一步得到的.root文件数据以 T103000 等命名, 第二步需要利用探测器Map将每个插件对应的探测器还原出来, 以PPAC1_T 等命名
- **操作**
 - (1) 修改/RIBLLVMEDAQ/ReadRootFile2D.cpp 文件
 - (2) 假定输入文件都在 /rootdata/中, 在/rootdata/下添加listfilename 文件,将需要转换的文件名称一一列出, 每个文件名占一行
 - (3) 为避免与原始的.root文件混淆, 转换后的.root文件需要另起名称, 且最好输出到不同的文件夹下
 - (4) 执行: 编译成功后, ./ReadRootFile2D listfilename

```
$ make
```

```
$ ./ReadRootFile2D listfilename
```

Step2. 数据质检

2.1 SetBranchAddress方法

2.2 TTreeReader方法

- QC_BranchAdress.C, QC_ReadTree.C
- 定义、填充直方图
- 存储所有的Hist, 输出pdf到文件, 方便肉眼进行质检
- 存储所有的Hist,写入.root文件. 这一步是为了:一旦pdf文件中的谱有问题, 马上可以查看.root文件中对应的直方图

- 编译 QC_ReadTree.C 后执行, ./QC_ReadTree listfilename

```
$ make
```

```
$ ./QC_ReadTree listfilename
```

Step3. PPAC数据处理

Step4. SSD数据处理

4.1 SSD Energy Calibration

(1) FindPedestal

- Pedestal是探测系统的零点道，是系统没有能量输入情况下，ADC中记录的道址. 理论上，ADC中探测到的所有能量信号都应该在对应的Pedestal以上. 因此Pedestal可以作为ADC能量的Cut值
- 写了一个手动选取拟合范围的程序. 基本操作是：单击鼠标中间键(滚轮)来取点，单击两次选择拟合范围，最后将拟合结果保存到pdf中，并将拟合参数保存到.dat文件中

```
$ root -l FindPedestals.C
```

(2) PulserCali_AutoFindPeaks

(3) PulserCali_LinearFit

- 写了一个自动寻峰的程序. 使用ROOT中TSpectrum类中的Search()方法实现自动寻峰.
- 将自动寻峰得到的每个峰的Ch作为Y值，每个峰对应的输入的pulser的相对幅度作为X值，画出一系列点
- 对这些pulser点进行用 $y = a * x + b$ 进行线性拟合，将拟合参数以及数据点保存到.dat文件中
- 将拟合结果保存成pdf，以便检查

```
$ root -l PulserCali_L1_AutoFindPeaksAndFit.C
```

```
$ root -l PulserCali_L2_AutoFindPeaksAndFit.C
```

(4) AlphaCali_FindPeak

- 使用TChain方法合并刻度文件：本次实验获得了多个alpha刻度文件，合并文件是为了增加统计量
- 利用从MSU拷贝回来的EnergyLossModule计算alpha穿过2um镀铝Mylar膜后的能量. 三组分alpha源的三个峰分别来源于239Pu, 241Am, 244Cm, 将三者发射alpha粒子

的加权平均能量作为alpha的出射能量

- 写了一个手动寻峰的程序，手动选取拟合范围，对三组分alpha源的三个alpha峰分别进行了拟合
- 将三个alpha峰的拟合结果保存到.dat文件中

```
$ root -l AlphaCali_MergeFiles.C
```

```
$ root -l AlphaCali_FindPeaks.C
```

```
$ root -l AlphaCali_CalEnergy.C
```

(5) Energy_Calibration

- 待续。。。。

```
$ root -l SiEnergyCali.C
```

4.2 Hit Multiplicity

4.3 CsI Energy Calibration

4.4 Hit Pixellation

4.5 Particle Identification

Step5. 物理分析