

# 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](mailto: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

- 数据转换
  - 将原始数据二进制文件转换成.root文件
  - 将RawRoot数据转换成MapRoot数据
- 数据质检
  - SetBranchAddress方法
  - TTreeReader方法
- PPAC数据处理
- SSD数据处理
  - 数据刻度
    - Energy Calibration
    - Hit Multiplicity
    - Csl Energy Calibration
    - Hit Pixellation
    - Particle Identification
  - 物理分析

## 数据转换

## 将原始数据二进制文件转换成.root文件

1. 修改文件：/RIBLLVMEDAQ/Raw2ROOT.cpp
2. 目的：将原始数据(二进制文件)转换成.root文件
3. 操作：批量转换
  - 3.1 输入文件(原始文件)都在 /vmedata/文件夹下
  - 3.2 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
  - 3.3 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. 执行：编译成功后, ./raw2roo.sh listfilename

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

## 将RawRoot数据转换成MapRoot数据

1. 修改文件：/RIBLLVMEDAQ/ReadRootFile2D.cpp
2. 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
3. 操作
  - 3.1 修改/RIBLLVMEDAQ/ReadRootFile2D.cpp 文件
  - 3.2 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
  - 3.3 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
  - 3.4 执行: 编译成功后, ./ReadRootFile2D listfilename

```
$ make  
$ ./ReadRootFile2D listfilename
```

## 数据质检

### SetBranchAdress方法

### TTreeReader方法

1. QC\_BranchAdress.cpp, QC\_ReadTree.cpp
2. 定义、填充直方图

3. 存储所有的Hist, 输出pdf到文件, 方便肉眼进行质检
4. 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
5. 编译 QC\_ReadTree.cpp 后执行, ./QC\_ReadTree listfilename

```
$ g++ QC_ReadTree.cpp -o QC_ReadTree  
$ ./QC_ReadTree listfilename
```

## PPAC数据处理

## SSD数据处理

## 数据刻度

### Energy Calibration

#### 1. FindPedestal

1.1 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

1.2 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.cpp
```

#### 2. PulserCali\_AutoFindPeaks

#### 3. PulserCali\_LinearFit

3.1 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.

3.2 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

3.3 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

3.4 将拟合结果保存成pdf, 以便检查

```
$ root -l PulserCali_L1_AutoFindPeaksAndFit.cpp  
$ root -l PulserCali_L2_AutoFindPeaksAndFit.cpp
```

#### 4. AlphaCali\_FindPeak

4.1 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

4.2 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 $^{239}\text{Pu}$ ,  $^{241}\text{Am}$ ,  $^{244}\text{Cm}$ , 将三者发射alpha粒子的加权平均能量作为alpha的出射能量

4.3 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

4.4 将三个alpha峰的拟合结果保存到.dat文件中

```
$ root -l AlphaCali_MergeFiles.cpp
```

```
$ root -l AlphaCali_FindPeaks.cpp
```

```
$ root -l AlphaCali_CalEnergy.cpp
```

#### 5. Energy\_Calibration

- 待续。。。

```
$ root -l SiEnergyCali.cpp
```

## Hit Multiplicity

## CsI Energy Calibration

## Hit Pixellation

## Particle Identification

## 物理分析