

文件格式说明

Mar 5, 2020

文件格式

我们将数据储存在[HDF5](#)文件中，其中有三个表：ParticleTruth每个事例中粒子的初始位置沉积能量和粒子种类；Waveform保存波形信息；PETruth保存每个光子的击中时间。每个表都有对应的事例编号或通道编号，如下所示。

ParticleTruth

EventID(int64)	x(float32)	y(float32)	z(float32)	E(float32)	Type(int16)
1	0	0	0	1	0
2	0	0	0	9	1
3	0	0	0	9	1
.....

Waveform

EventID(int64)	ChannelID(int16)	Waveform(int16[1029])
1	0	977,973,.....,972
1	1	973,974,.....,975
1	2	973,973,.....,974
.....

PETruth

EventID(int64)	ChannelID(int16)	PETime(int16)
1	0	267
1	0	285
1	0	288
.....

使用[HDFView](#)打开数据文件，可以查看文件的大致结构

playground_bak.h5

数据文件结构

另外我们也提供了ChannelID所对应的PMT的位置的txt文件，大致如下

PMT_Position.txt - 记事本			
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)			
File: './PMT_Position_1ton.xml'			
Size: 7002 Blocks: 16 IO Block: 2097152 regular file			
Device: c:\02b520h\3238180128d Inode: 144371452433455112 Links: 1			
Access: (0644/-rw-r--r--) Uid: (1662/wangzhe2) Gid: (100/ users)			
Access: 2020-02-29 03:18:06.000000000 +0800			
Modify: 2020-02-29 03:14:47.000000000 +0800			
Change: 2020-02-29 03:15:06.000000000 +0800			
Position data:			
0	428	139	700
8	0	451	700
16	-428	139	700
24	-265	-364	700
1	265	-364	700
9	796	0	245
17	644	467	245
25	245	756	245
2	-245	756	245
10	-644	467	245
18	-796	0	245
26	-644	-467	245
3	-245	-756	245
11	245	-756	245
19	644	-467	245
27	796	0	-245
4	644	467	-245
12	245	756	-245
20	-245	756	-245
28	-644	467	-245
5	-796	0	-245
13	-644	-467	-245
21	-245	-756	-245
29	245	-756	-245
6	644	-467	-245
14	428	139	-700
22	0	451	-700
23	-428	139	-700
7	-265	-364	-700
15	265	-364	-700

ChannelID所对应的PMT位置

其中每一列的具体含义如下

PMT Position

ChannelID	x(mm)	y(mm)	z(mm)
0	428	139	700
8	0	451	700
16	-428	139	700
.....	

读取文件示例

Python

利用Python的 `pytables` 模块

```
pip install tables
```

如果出现下载过慢的问题，可以更改pip配置，改为国内资源

[pip配置国内资源](#)

我们可以方便地读取hdf5文件。

```
import tables
import matplotlib
import matplotlib.pyplot as plt

# Read hdf5 file
filename = "test.h5"
h5file = tables.open_file(filename, "r")

WaveformTable = h5file.root.Waveform
entry = 0
EventId = WaveformTable[entry]['EventID']
ChannelId = WaveformTable[entry]['ChannelID']
Waveform = WaveformTable[entry]['Waveform']
minpoint = min(Waveform)
maxpoint = max(Waveform)

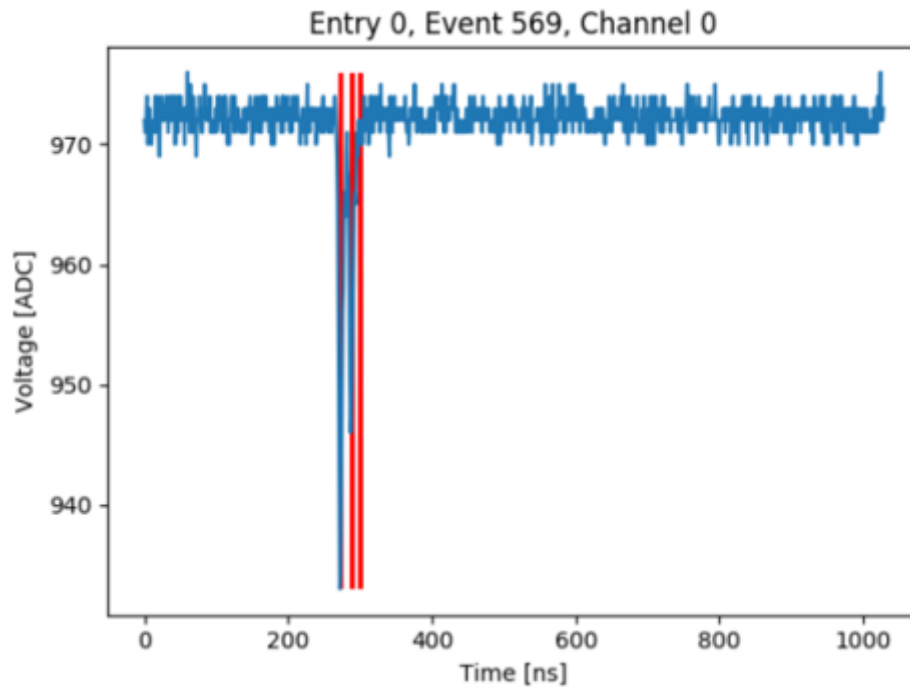
PETruthTable = h5file.root.PETruth
PETime = [x['PETime'] for x in PETruthTable.iterrows() if x['EventID'] ==
EventId and x['ChannelID']==ChannelId]
print(PETime)

plt.plot(Waveform)
plt.xlabel('Time [ns]')
plt.ylabel('Voltage [ADC]')
for time in PETime:
    plt.vlines(time, minpoint, maxpoint, 'r')

plt.title("Entry %d, Event %d, Channel %d" % (entry, EventId, ChannelId))
plt.show()

h5file.close()
```

这段程序将Waveform表中的第一个记录读取，并在PETTruth表中取出对应的光电子到达时间，将其画出。得到的图像为



波形和击中时间的示例图

C++

HDF5原生库提供了C/C++的支持。编译安装HDF5库后指定头文件目录和库即可编译下面的示例代码

```
g++ -std=c++11 -o test -I/usr/local/hdf5/include/ ReadExample.cpp -lhdf5 -lhdf5_hl
```

```
#include "hdf5.h"
#include "hdf5_hl.h"
#include <stdlib.h>
#include <iostream>
using namespace std;

constexpr size_t nWindowSize = 1029;
constexpr size_t nFields = 3;

struct WaveformData
{
    long long EventID;
    short ChannelID;
    short waveform[nWindowSize];
};

int main()
{
    WaveformData wf_buf;

    /* Calculate the size and the offsets of our struct members in memory */
    size_t dst_size = sizeof(WaveformData);
    size_t dst_offset[nFields] = { HOFFSET( WaveformData, EventID ),
                                   HOFFSET( WaveformData, ChannelID ),
                                   HOFFSET( WaveformData, waveform )
    };
    size_t dst_sizes[nFields] = { sizeof(wf_buf.EventID),
```

```

        sizeof(wf_buf.ChannelID),
        sizeof(wf_buf.Waveform)
    };

    hid_t file_id = H5Fopen( "test.h5", H5F_ACC_RDONLY, H5P_DEFAULT);
    hid_t group_id = H5Gopen(file_id, "/", H5P_DEFAULT);
    H5TBread_records( group_id, "Waveform", 0, 1, dst_size, dst_offset,
dst_sizes, &wf_buf);
    for(int i=0; i<nwindowSize; i++)
        cout<<wf_buf.Waveform[i]<<" ";
    cout<<endl;

    H5Fclose( file_id );
    return 0;
}

```

这段程序读取了Waveform表中的第一条记录，并把波形数值打印了出来。