

提交格式说明

Mar 4, 2020

提交格式

选手提交HDF5格式的文件，在文件中写入一个极为简单的表格，一列为EventID，一列为Alpha的概率。将此表命名为Answer（不需要建立group）

Answer

EventID(int64)	Alpha (float32)
1	0.0
2	0.5
3	1.0
.....

Alpha用于鉴别粒子，Alpha=1 表示该 Event 有 100% 几率为Alpha粒子。

进一步推广，如果分析算法认为在该 Event 有 50% 的几率是Alpha粒子，则 可以把该行的 Alpha 写为 0.5。

写入文件示例

在下面的示例中，我们向文件中写入上表中的三条示例数据。

Python

```
# An example of writing a file.
import tables

# Define the database columns
class AnswerData(tables.IsDescription):
    EventID    = tables.Int64Col(pos=0)
    Alpha      = tables.Float32Col(pos=1)

# Create the output file and the group
h5file = tables.open_file("MyAnswer.h5", mode="w", title="OneTonDetector")

# Create tables
AnswerTable = h5file.create_table("/", "Answer", AnswerData, "Answer")
answer = AnswerTable.row

# Write data
answer['EventID'] = 1
answer['Alpha'] = 0.3
answer.append()
answer['EventID'] = 1
```

```

answer['Alpha'] = 0.5
answer.append()
answer['EventID'] = 1
answer['Alpha'] = 1.0
answer.append()

# Flush into the output file
AnswerTable.flush()

h5file.close()

```

C++

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

```

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

constexpr size_t nFields = 2;
constexpr size_t nRecord = 3;

struct AnswerData
{
    long long EventID;
    float Alpha;
};

int main() {

    AnswerData dst_buf;

    // Calculate the size and the offsets of our struct members in memory
    size_t dst_size = sizeof( AnswerData );
    size_t dst_offset[nFields] = { HOFFSET( AnswerData, EventID ),
        HOFFSET( AnswerData, Alpha )
    };

    size_t dst_sizes[nFields] = { sizeof( dst_buf.EventID),
        sizeof( dst_buf.Alpha)
    };

    // Define an array of data
    AnswerData p_data[nRecord] = {
        {1, 0.3},
        {2, 0.5},
        {3, 1.0}
    };

    // Define field information
    const char *field_names[nFields] =
    { "EventID", "Alpha" };
    hid_t      field_type[nFields];
    hid_t      file_id;

```

```

hsize_t    chunk_size = 10;
int         *fill_data = NULL;
int         compress   = 0;
int         i;

// Initialize field_type
field_type[0] = H5T_NATIVE_LLONG;
field_type[1] = H5T_NATIVE_FLOAT;

// Create a new file using default properties.
file_id = H5Fcreate( "MyAnswer_cpp.h5", H5F_ACC_TRUNC, H5P_DEFAULT,
H5P_DEFAULT );

// Make the table.
H5TBmake_table( "Table Title", file_id, "Answer", nFields, nRecord,
               dst_size, field_names, dst_offset, field_type,
               chunk_size, fill_data, compress, p_data );

// Close the file.
H5Fclose( file_id );

return 0;
}

```