LuckyDoll, an analysis package for the AIDA detector

Vi Ho Phong

A preliminary manual
November 5, 2016

Contents

1 Changelog

1.1 November 5, 2016

Change the help menu in ./aidafull For ./aidafull and ./aida: fix bug that force users to enter threshold and calibration file even if they don't want to.

1.2 October 28, 2016

A faster clustering function has been added. About 25% improvement in the overall perfomance.

1.3 October 26, 2016

A almost final version released

2 Introduction

2.1 What This is About?

LuckyDoll is a data analysis package which is used for analyzing the data from AIDA detector.

The AIDA detector is a new generation implantation detector which is specificly designed for decay experiment at fast RI beam facilities. More information about AIDA can be found at [2]

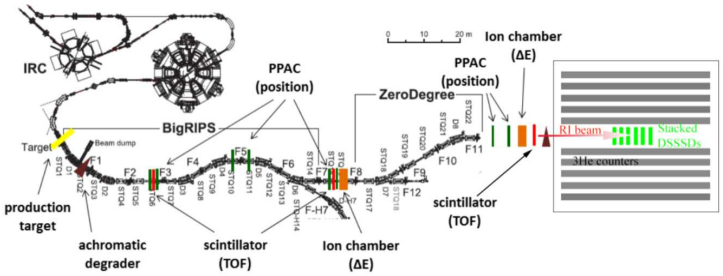

Figure 1: A schematic view of the BRIKEN setup

The developement of the LuckyDoll package is gorverned by a number of principles:

- Capatable to analyze a large amount of data from both implantation and decay events within a reasonable computing time so that it can be used for semi-online analysis during the experiment
- Comprehensive data structure
- Easy to understand, easy to use
- Produce output data with a mimized the number of the background and unwanted events.

2.2 LuckyDoll main classes and what are they good for

LuckyDoll package is divded into several classes, each is programed to perform a particluar task

AIDAUnpacker class handles the decoding of the MIDAS data and convert it into the "hits".

BuildAIDAEvent class is used to reconstruct the decay and implantation events from the energy deposition in the strips of the sillicon detector. The reconstruction is based on timing and spatial correlation. The implementation of the class is as follow:

- Open files, initialize the Event Builder object

```
TFile* ofile = new TFile(OutFile,"recreate");
 ofile->cd();
```

```
//! Book tree and histograms
TTree* treeion=new TTree("ion","tree ion");
TTree* treebeta=new TTree("beta","tree beta");
TTree* treepulser=new TTree("pulser","tree pulser");


BuildAIDAEvents* evts=new BuildAIDAEvents;
      evts->SetVerbose(Verbose);
      evts->SetFillData(true);
      evts->BookTree(treeion,treebeta,treepulser);
      evts->SetMappingFile(MappingFile); //must be set first
      evts->SetThresholdFile(ThresholdFile);//must be set first
      evts->SetCalibFile(CalibrationFile);
      evts->SetAIDATransientTime(TransientTime);
      evts->SetEventWindowION(WindowIon);
      evts->SetEventWindowBETA(WindowBeta);
evts->Init(R10_0);
```

- Event loop (similiar with analoop)

```
while(evts->GetNextEvent()){
            if(evts->IsBETA()) evts->GetAIDABeta()->GetCluster(0)->GetHitPos
            else evts->GetAIDAIon()->GetCluster(0)->GetHitPositionX();
evts->GetAIDAIon()->GetCluster(0)->GetEnergy();
evts->GetAIDAIon()->GetCluster(0)->GetTimestamp();
}
```

- Write tree and close files

```
      treeion->Write();
      treebeta->Write();
      treepulser->Write();
      ofile->Close();
}
```

2.3 LuckyDoll main programs

The Lucky main programs are builted on top of the main classes to perform some paticular analysis jobs.

- **./aida** : Unpack, calibrate, build aida events and produce a root file with minimum information which later can be used for the BRIKEN merger program.
- **./aidafull** : Unpack, calibrate, build aida events and produce a root file with all information needed.
- **./aida2tree** : Unpack the raw data and produce a root file containing infomation about hits in FEE channels (uncalibrated)
- **./aida2histbkg** : Make histograms of adc value versus strip number for the purpuse of determining the individual threshold for each AIDA strip.

- **./aida2histcalib** : Make histograms of adc value versus strip number considering the pulser events (high multiplicity) or low energy hits events for the purpuse of energy calibration using pulser or radioactive source.

3 LuckyDoll basic: How to run it and what is the data structure?

3.1 Installation

....

Prerequites

- **gcc**
- **cmake** version ≥ 6
- CERN **root** version ≥ 5.34

Installation Steps

```
git clone https://github.com/vihophong/LuckyDoll.git
cd LuckyDoll
cd ../
mkdir build_LuckyDoll
cd build_LuckyDoll
cmake ../LuckyDoll
make
```

3.2 Convert AIDA full data

```
$ ./aidafull
AIDA event builder
use ./aidafull with following flags:
        [-a    <char*    >: AIDA input list of files]
        [-o    <char*    >: output file]
        [-wi   <long long>: Ion event building window]
        [-wb   <long long>: Beta event building window]
        [-wd   <long long>: Fast Discriminator Scan window]
        [-v    <int      >: verbose level]
        [-map  <char*    >: mapping file (default: FEE_table.txt)]
        [-cal  <char*    >: calibration file]
        [-thr  <char*    >: threshold file]
        [-f    <int      >: fill data or not: 1 fill data 0 no fill (default: f
        [-tt   <long long>: aida transient time (20000?)]
        [-ecut <char*    >: specify energy cut file]
```

Note:

- The event building window, fast discriminator scan window and transient time have been set by default (see above). Therefore, there is no need to input the -wi -wb -wd -tt if user don't want to change those value.

- If there is no calibration table specified by user the energy calibration will not be made, which is not good for the event correlation, especially the clustering algorithm.
- If there is no calibration table specified by user, the program will consider getting all information from AIDA (threshold = -10000) for all the channels.

3.3 Convert AIDA partial data

```
$ ./aida
AIDA event builder
use ./aida with following flags:
        [-a    <char*    >: AIDA input list of files]
        [-o    <char*    >: output file]
        [-wi   <long long>: Ion event building window (default: 2500*10ns)]
        [-wb   <long long>: Beta event building window (default: 2500*10ns)]
        [-wd   <long long>: Fast Discriminator Scan window (default: 0 i.e no s
        [-v    <int      >: verbose level]
        [-map  <char*    >: mapping file]
        [-cal  <char*    >: calibration file]
        [-thr  <char*    >: threshold file]
        [-f    <int      >: fill data or not: 1 fill data 0 no fill (default: f
        [-tt   <long long>: aida transient time (default: 20000*10ns)]
        [-ecut <char*    >: specify energy cut file]
```

4 Advance topics: Understanding LuckyDoll

4.1 How to read the timestamp in AIDA

It's complicated...

4.2 How the clustering algorithm works

It's complicated...

4.3 How to handle the transient time after the implantation event

It's complicated...

References

[1] A. Tarifeño-Saldivia et al.: arXiv:1606.05544.

[2] http://www2.ph.ed.ac.uk/ td/AIDA/Information/information.html