INTT software

Takashi Hachiya
Nara Women's University

Topics

- Status/proposal of HotDead map implementation with Fun4All framework
- A proposal INTT based tracking for p+p 2024 and peripheral Au+Au
- Survey geometry to GEANT
  - Not discussed in the slide. But need to think

Status HotDead map

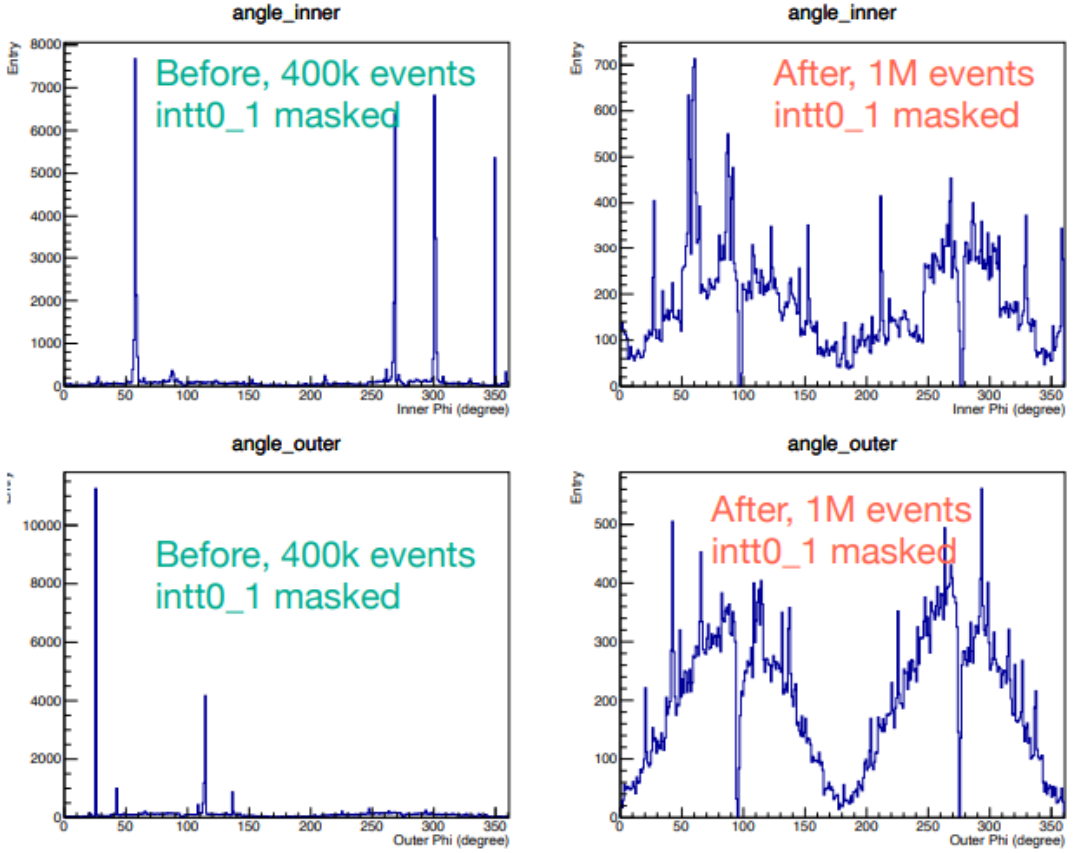
- Urgent issue: hotdead map for the cosmic dataset
  - Tracking group requested the hotdead map for the cosmic data at the last collaboration meeting. They plan to perform the geometry alignment with the cosmic data

- What needed for DST production
  1. Study hitmap (hit/ch) to determine hot/dead map
     - For Au+Au dataset, Jaein, Joseph, and Yuka did (doing) nice work. Jaein/Yuka's version is available
     - Cosmic ray : ChengWei showed some plots. WeiChe is working.
  2. Software module/structure to handle the hot/dead map in F4A
  3. F4A module to generate Hot/Dead map
     - Jaein is working this?

HitMap

Hot channel algorithm - cosmic rays

Cosmic run 25952



Data driven method, check the average of number of entry of each channel

The empty region (~95 degree, 275 degree) → gap b/w west & east halves, survey data included

- Au+Au method (hit/ch) can be applied but different threshold
- The map should be made for run by run.
  - If the map is stable for longer period, the common map should be OK.

Reported by ChengWei at the software meeting

HotDead map software

- I think Joseph already is making this. I would like to see what the other subsystems make for
  - Good to know “sPHENIX way” to handle the calibration parameter

- Handling calibration parameter in F4A
  - CDB : Conditional Database for calibration parameter
    - Parameter condition is stored in DB
      - Ex: run2023
    - Calibration parameters are stored in the ROOTFILE (not DB)

Handling calibration parameter in F4A

- CDBInterface
  - utility to find the calibration file in the internet
  - not sure how this module access to the DB. It is under investigation

- 2 default container classes for calibration parameter in F4A
  - CDBTTree : TTree style
    - std::map<int, std::map<std::string, double>> m_DoubleEntryMap;
      - Int channel,
      - String variable name,
      - double parameter
    - Not only double, but other type (float, int, int64)
  - CDBHisto: Histogram style
  - Parameters are kept in ROOTFILE

Example : Calorimeter (RecoTowerCalib)

InitRun in RecoTowerCalib (subsysreco module for F4A)

```
else if (m_dettype == CaloTowerDefs::HCALOUT)
 {
   m_detector = "HCALOUT";
   m_DETECTOR = TowerInfoContainer::HCAL;

   if (!m_overrideCalibName)
   {
     m_calibName = "ohcal_abscalib_cosmic";
   }
   if (!m_overrideFieldName)
   {
     m_fieldname = "ohcal_abscalib_mip";
   }
   std::string calibdir = CDBInterface::instance()->getUrl(m_calibName);
   if (!calibdir.empty())
   {
     cdbttree = new CDBTTree(calibdir);
   }
   else
   {
     std::cout << "CaloTowerCalib::::InitRun No calibration file for domain " << m_calibName << " found" << std::endl;
     exit(1);
   }
```

- Interface
  - Get the directory name of the calibration file with DB
- New CDBTTree with dir-name
  - Get tree

Example : Calorimeter (RecoTowerCalib)

```
int CaloTowerCalib::process_event(PHCompositeNode *topNode)
{
  TowerInfoContainer *_raw_towers
            = findNode::getClass<TowerInfoContainer>(topNode, RawTowerNodeName);
  TowerInfoContainer *_calib_towers
            = findNode::getClass<TowerInfoContainer>(topNode, CalibTowerNodeName);
  unsigned int ntowers = _raw_towers->size();

  for (unsigned int channel = 0; channel < ntowers; channel++)
  {
    unsigned   int key      = _raw_towers->encode_key(channel);
    TowerInfo *caloinfo_raw = _raw_towers->get_tower_at_channel(channel);
    _calib_towers->get_tower_at_channel(channel)->copy_tower(caloinfo_raw);

    float raw_amplitude = caloinfo_raw->get_energy();
    float calibconst = cdbttree->GetFloatValue(key, m_fieldname);
    _calib_towers->get_tower_at_channel(channel)->set_energy(raw_amplitude * calibconst);
  }
  return Fun4AllReturnCodes::EVENT_OK;
}
```

- Input
- Output
- Gain parameter from CDBTTree
- Access by key

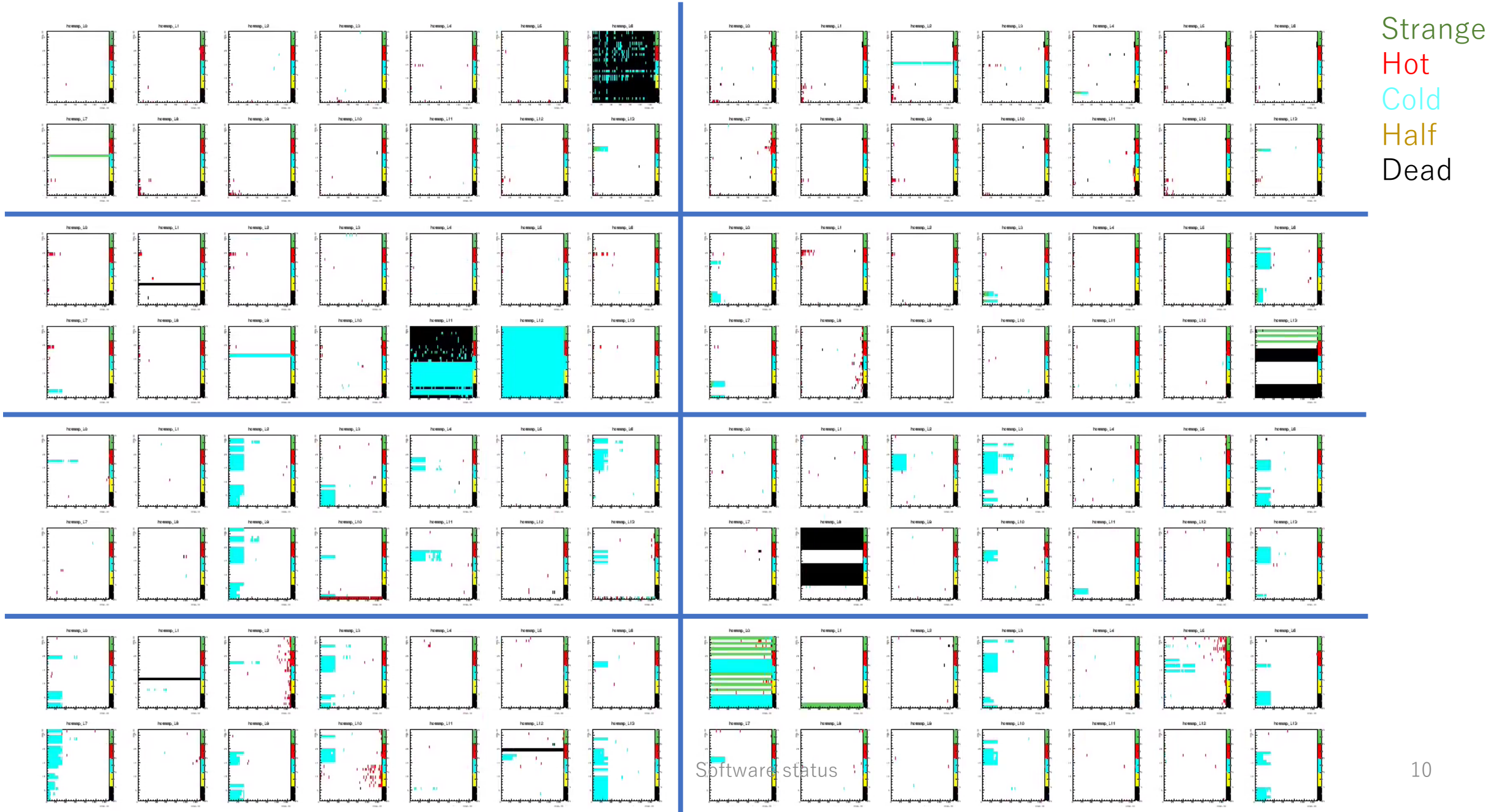
INTT hot/dead map class

Hot/Dead Status	ID
Good	0
Hot	8
Cold	4
Half	2
Dead	1

- Propose to make new HodDeadMap class
  - Having same I/F with InttDeadMap (for sim) and replace it
- Adding structure
  - CDBTTree
    - Use std::map<int, std::map<std::string, int>> m_IntEntryMap;
      - Int channel = encoded ladder/chip/channel ch
      - String variable name = InttHotDeadMap
      - double parameter = status
  - Should have
    - isBad(online chan-id) or offline chan-id

```
//------------------
// remove hotchannel
if(hotmap_.isHot(rawdata.felix_server,
                 rawdata.felix_channel,
                 rawdata.chip+1,
                 rawdata.channel))
{
   //cout<<" hotchanel : "<<rawdata.feli
   //                     <<rawdata.chip
   continue;
}
```

HotDead Map (by Yuka)



Strange
Hot
Cold
Half
Dead

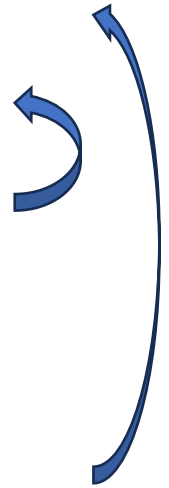
Proposal INTT based tracking

- INTT based track is useful for high luminosity p+p collision
  - INTT is good to identify the collision from single beam crossing.
  - Because of small multiplicity in p+p, INTT based tracking can provide good S/N with pT measurement

- I would like to propose INTT based tracking project and should complete first p+p run in 2024 (May 1st)
  - Use this to both B-OFF and B-ON data
  - We have a lot of experience to generate INTT tracklet for the vertex measurement and so on. This effort can extend to the INTT based tracking
  - Many students makes their own tracking algorithm. It is GOOD to cross check the algorithm. Now good to unite their efforts

INTT based tracking workflow

1. INTT track reconstruction
    1. Make INTT tracklet (a pair of INTT hits at inner and outer layer) with MBD z-vertex and Beam center
    2. Reco INTT z-vertex
    3. Momentum reconstruction using 2 hits and vertex



2. Extend INTT track into the MVTX(inner) and EMCal (outer)
    - Confirm INTT track
    - Better Z- resolution with at least one MVTX point
    - Better pT resolusion with more points with larger R (EMCal)

- Repeat these steps to refine vertex position and tracks with better tracks at each stage.
- To do this iteration procedure, the data format is important
- I will bring detailed framework next week

Summary

- Hot/dead map
  - Urgent for the cosmic dataset
  - Machinary both determination and handling the map is necessary
- INTT based tracking
  - Good for p+p running

- Online Channel
  - FELIX: 0-7, Lad:0-13, Chip: 1-26, Chan: 0-127
    - 3bit + 4bit + 5bit + 7bit = 19 bit
- Offline Channel
  - Lay: 3-6, Lad:0-15 (11), Sens: 0-3, Chip: 0-7 (5), Chan: 0-255
    - 3bit + 4bit + 2bit + 3bit + 8bit = 20 bit

INTT calibration parameter