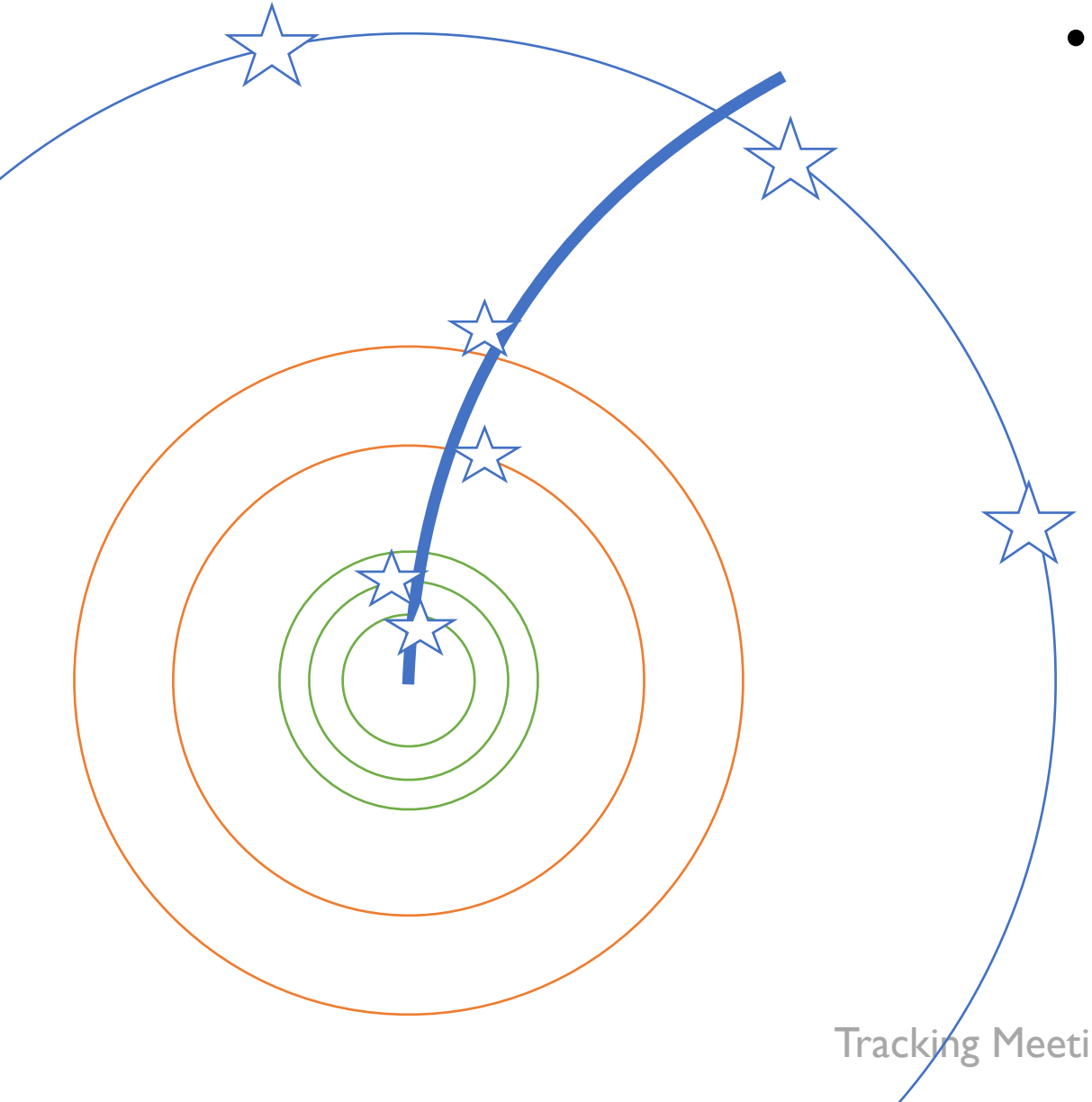
SiCalo Tracking meeting

Jaein Hwang (Korea Univ.)

July 10 2025

New update on the Si-Calo framework



- Software framework
  1) SiliconSeeding -> Si-track

  2) Extrapolate to EMCal Surface

  3) Calculate EMCal η based on vertex
  (If vertex object is existed. If not,
  use origin to calculate the η

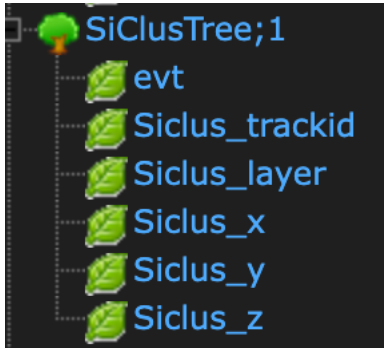
  4) Find clusters track pair which has minimum

$$\Delta R = \sqrt{(\eta_{track} - \eta_{emc})^2 + (\phi_{track} - \phi_{emc})^2}$$

  5) Save in TrackTree

Output Tree for analysis update

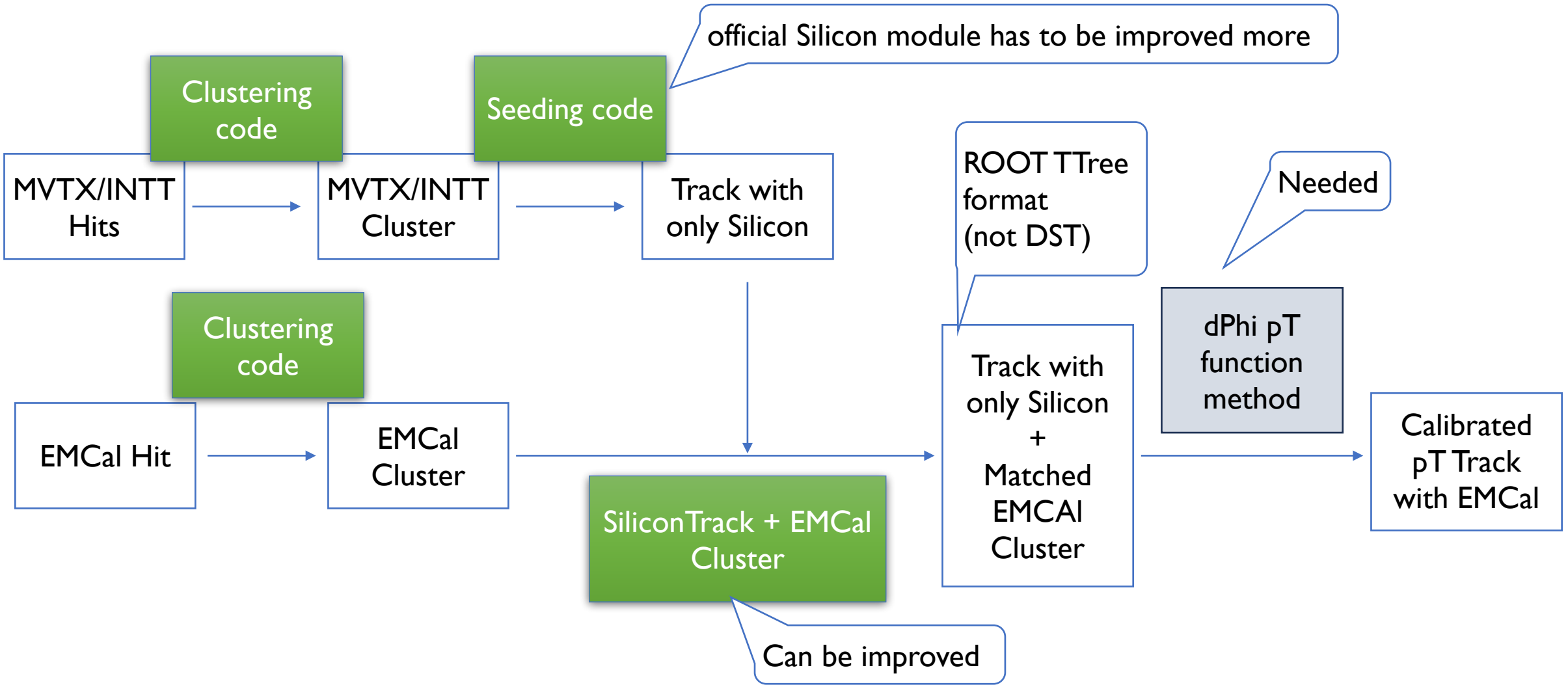
OLD



X0,Y0,Z0… track parameters at the point closest at origin

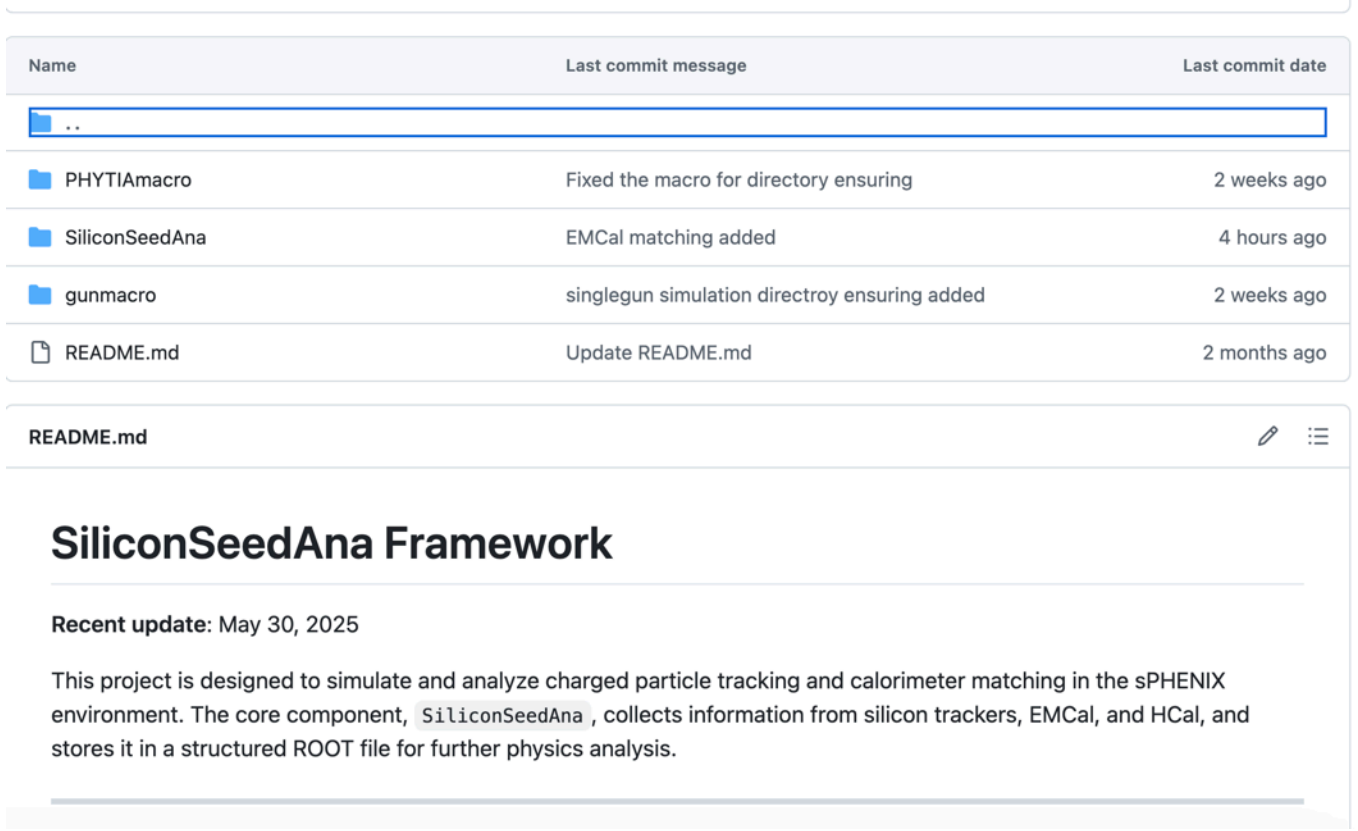
Matched calo hits saved in trackTree together for user convenience
calo_dR is calculated ΔR

Macro / code framework update



Instruction document(1)

- Test with beginner of this study(Mahiro), successfully run through the current instruction in GitHub
  GitHub Readme cannot include every information



Name	Last commit message	Last commit date
..		
PHYTIAmacro	Fixed the macro for directory ensuring	2 weeks ago
SiliconSeedAna	EMCal matching added	4 hours ago
gunmacro	singlegun simulation directroy ensuring added	2 weeks ago
README.md	Update README.md	2 months ago

README.md

SiliconSeedAna Framework

Recent update: May 30, 2025

This project is designed to simulate and analyze charged particle tracking and calorimeter matching in the sPHENIX environment. The core component, `SiliconSeedAna`, collects information from silicon trackers, EMCal, and HCal, and stores it in a structured ROOT file for further physics analysis.

Instruction document(2)

- You can find much more details on wiki

INTermediate Tracker (INTT)

Contents [hide]

1 General Info
2 Runs
3 Online
4 Offline
5 Hardware
6 Internal Info

General Info [edit | edit source]

- Overview
- Members
- Documents/Publications
- Talks

Runs [edit | edit source]

- Beam Test
- Run23
- Run24
- Run25

Online [edit | edit source]

- Overview
- DAQ Operation
- Online Monitoring
- Online Software

Offline [edit | edit source]

- Overview
- Offline QA
- Analysis Software
- Simulation

INTT Analysis Software

Back to INTT top page

Contents [hide]

1 General
2 Fun4All INTT packages
3 DST information
4 INTT+Calo Tracking
5 INTT Software tips
6 sPHENIX Software Tips

General [edit | edit source]

- INTT Code Cleanup

Fun4All INTT packages [edit | edit source]

DST information [edit | edit source]

- DST INTT Contents

INTT+Calo Tracking [edit | edit source]

- Mattermost Channel
- Meetings
- Takuya's workshop analysis note
- Jingyu's instruction on EMcal-related codes and G4Hit
- Jaein's Silicon + EMCal TTree generator (July 9 2025)

TrackSeed + EMCal TTree generator

Back to INTT Analysis Software page

Here is a basic understanding of the EMCal-related code and G4Hit in sPHENIX.

Contents [hide]

1 How to run the code
2 Macro Setup and details explanation
  2.1 Example macros
3 Analysis modules
  3.1 PHActsSiliconSeeding
  3.2 PHSiliconSeedMerger
  3.3 TrackSeedTrackMapConverter
  3.4 PHActsTrackProjection
  3.5 SiliconSeedsAna
4 SiliconSeedAna
  4.1 TrackTree
  4.2 SiClusTree
  4.3 caloTree
  4.4 truthTree (optional for MC)
  4.5 How to get projected value at EMCal?
  4.6 How to get cluster position used for track building from SVTX track object?

Instruction document(3)

- Basic instruction for how to run(will be detailed by Mahiro)
- Example Macros available, test has been done (good enough for analysis)
- Macro can cover single gun/PHYTIA/data.(But user must know what you want to do!)

TrackSeed + EMCal TTree generator

Back to INTT Analysis Software page

Here is a basic understanding of the EMCal-related code and G4Hit in sPHENIX.

Contents [hide]

1 How to run the code
2 Macro Setup and details explanation
2.1 Example macros
3 Analysis modules
3.1 PHActsSiliconSeeding
3.2 PHSiliconSeedMerger
3.3 TrackSeedTrackMapConverter
3.4 PHActsTrackProjection
3.5 SiliconSeedsAna
4 SiliconSeedAna
4.1 TrackTree
4.2 SiClusTree
4.3 caloTree
4.4 truthTree (optional for MC)
4.5 How to get projected value at EMCal?
4.6 How to get cluster position used for track building from SVTX track object?

How to run the code [edit | edit source]

Main Github link : github

1) git clone the code from the above link.

2) Compile the code in SiliconSeedAna directory

3) Set up SiliconSeedAna library in your setup

Macro requires several libraries built in coresoftware; please ensure you don't have your private coresoftware build in your library list.

Macro Setup and details explanation [edit | edit source]

This section details the analysis modules that you need to do Silicon+Calo study.

Example macros [edit | edit source]

Please refer to the example macros in GitHub under INTT repository.

For single gun simulation : gunmacro

For PHYTIA or data : phytia/datamacro

For a description of the analysis module commonly used in both macros, please refer to the Analysis Modules section below.

Instruction document(4)

- Include every analysis module required by example macros
  most of code are in coresoftware, but you may want to know what is what, where it is.

TrackSeed + EMCal TTree generator

Back to INTT Analysis Software page

Here is a basic understanding of the EMCal-related code and G4Hit in sPHENIX.

Contents [hide]

1 How to run the code
2 Macro Setup and details explanation
  2.1 Example macros
3 Analysis modules
  3.1 PHActsSiliconSeeding
  3.2 PHSiliconSeedMerger
  3.3 TrackSeedTrackMapConverter
  3.4 PHActsTrackProjection
  3.5 SiliconSeedsAna
4 SiliconSeedAna
  4.1 TrackTree
  4.2 SiClusTree
  4.3 caloTree
  4.4 truthTree (optional for MC)
  4.5 How to get projected value at EMCal?
  4.6 How to get cluster position used for track building from SVTX track object?

Analysis modules [edit | edit source]

This section details analysis modules for the macro in current framework. These are highly expected to be used in our study.

PHActsSiliconSeeding [edit | edit source]

The PHActsSiliconSeeding github module searches for silicon (MVTX + INTT) clusters and identifies a list of clusters, which is usually referred to as a seed.

```
auto silicon_Seeding = new PHActsSiliconSeeding;
// silicon_Seeding->set_track_map_name("SvtxTrackSeedContainer");
silicon_Seeding->Verbosity(0); // Verbosity level (0 = less printing)
silicon_Seeding->setinttRPhiSearchWindow(0.2); // INTT r-phi seed search window
silicon_Seeding->setinttZSearchWindow(1.0); // INTT Z-seracrh window
silicon_Seeding->seedAnalysis(false); // Some auto-analysis with seed object. Better to
write your code for own code.
se->registerSubsystem(silicon_Seeding); // Register at Fun4AllServer; (usually se is :
auto se = Fun4AllServer::instance(); )
```

PHSiliconSeedMerger [edit | edit source]

SiliconSeedMerger github scans the SeedContainer and removes the redundant seeds from the container.

```
auto merger = new PHSiliconSeedMerger;
// merger->trackMapName("SvtxTrackSeedContainer");
merger->Verbosity(0);
se->registerSubsystem(merger);
```

TrackSeedTrackMapConverter [edit | edit source]

TrackSeedTrackMapConverter github converts Tracks Seeds into SVTX Tracks. The SVTX TrackMap is the main analysis container that holds kinematic variables such as pT, x, y, z, and charge. This module performs a simple circle fit using the clusters from the SiliconSeed, extracts the track parameters, and stores them in the SVTX TrackMap container. The user can specify a particular seed container using the setTrackSeedName function, depending on the intended purpose. If the seed container name is not explicitly changed, the default will be one of the following:

Instruction document(5)

- Include every analysis module required by example macros
  most of code are in coresoftware, but you may want to know what is what, where it is.

TrackSeed + EMCal TTree generator

Back to INTT Analysis Software page

Here is a basic understanding of the EMCal-related code and G4Hit in sPHENIX.

Contents [hide]

1 How to run the code
2 Macro Setup and details explanation
  2.1 Example macros
3 Analysis modules
  3.1 PHActsSiliconSeeding
  3.2 PHSiliconSeedMerger
  3.3 TrackSeedTrackMapConverter
  3.4 PHActsTrackProjection
  3.5 SiliconSeedsAna
4 SiliconSeedAna
  4.1 TrackTree
  4.2 SiClusTree
  4.3 caloTree
  4.4 truthTree (optional for MC)
  4.5 How to get projected value at EMCal?
  4.6 How to get cluster position used for track building from SVTX track object?

SiliconSeedAna [edit | edit source]

This section much details on SiliconSeedAna part. Basically SiliconSeedAna makes 3 TTree

TrackTree [edit | edit source]

```
trackTree = new TTree("trackTree", "Track Data");
trackTree->Branch("evt", &evt, "evt/I"); // int: event index

trackTree->Branch("track_id", &track_id); // vector<unsigned int>: track index per event (multiple tracks possible)

trackTree->Branch("x0", &track_x); // vector<float>: track x position at origin
trackTree->Branch("y0", &track_y); // vector<float>: track y position at origin
trackTree->Branch("z0", &track_z); // vector<float>: track z position at origin

trackTree->Branch("px0", &track_px); // vector<float>: x component of momentum at origin
trackTree->Branch("py0", &track_py); // vector<float>: y component of momentum at origin
trackTree->Branch("pz0", &track_pz); // vector<float>: z component of momentum at origin

trackTree->Branch("eta0", &track_eta); // vector<float>: pseudorapidity at origin
```

truthTree (optional for MC) [edit | edit source]

```
truthTree = new TTree("truthTree", "Truth Particle Data");

truthTree->Branch("truth_pid", &truth_pid);   // vector<int>: PDG ID of the truth particle
truthTree->Branch("truth_px", &truth_px);     // vector<float>: x component of momentum
truthTree->Branch("truth_py", &truth_py);     // vector<float>: y component of momentum
truthTree->Branch("truth_pz", &truth_pz);     // vector<float>: z component of momentum
truthTree->Branch("truth_e", &truth_e);       // vector<float>: energy of the truth particle

truthTree->Branch("truth_pt", &truth_pt);     // vector<float>: transverse momentum (pT)
truthTree->Branch("truth_eta", &truth_eta);   // vector<float>: pseudorapidity
truthTree->Branch("truth_phi", &truth_phi);   // vector<float>: azimuthal angle

truthTree->SetBasketSize("*", 50000); // Set a larger basket size for better performance
```

- Truth information of primary particles

Instruction document(6)

- Last two parts are expected question from user..
  1) How to get project track value at EMCal
  2) How to get cluster position used for track builing from SVTX track object?
  Let me know if you have additional question on this framework. I can add more

How to get projected value at EMCal? [edit | edit source]

SvtxTrackState module projects the track into a specific radius. **You MUST put PHActsTrackProjection module in your macro to use SvtxTrackState class.**

See the example below from the SiliconSeedAna module.

```
SvtxTrackState *emcalState = track->get_state(_caloRadiusEMCal);
    fillEMCalState(emcalState);
```

```
void SiliconSeedsAna::fillEMCalState(SvtxTrackState* state) {
  track_x_emc.push_back(state ? state->get_x() : NAN);
  track_y_emc.push_back(state ? state->get_y() : NAN);
  track_z_emc.push_back(state ? state->get_z() : NAN);
  track_px_emc.push_back(state ? state->get_px() : NAN);
  track_py_emc.push_back(state ? state->get_py() : NAN);
  track_pz_emc.push_back(state ? state->get_pz() : NAN);
  track_eta_emc.push_back(state ? state->get_eta() : NAN);
  track_phi_emc.push_back(state ? state->get_phi() : NAN);
  track_pt_emc.push_back(state ? state->get_pt() : NAN);
}
```

How to get cluster position used for track building from SVTX track object? [edit | edit source]

See the example below from SiliconSeedAna module.

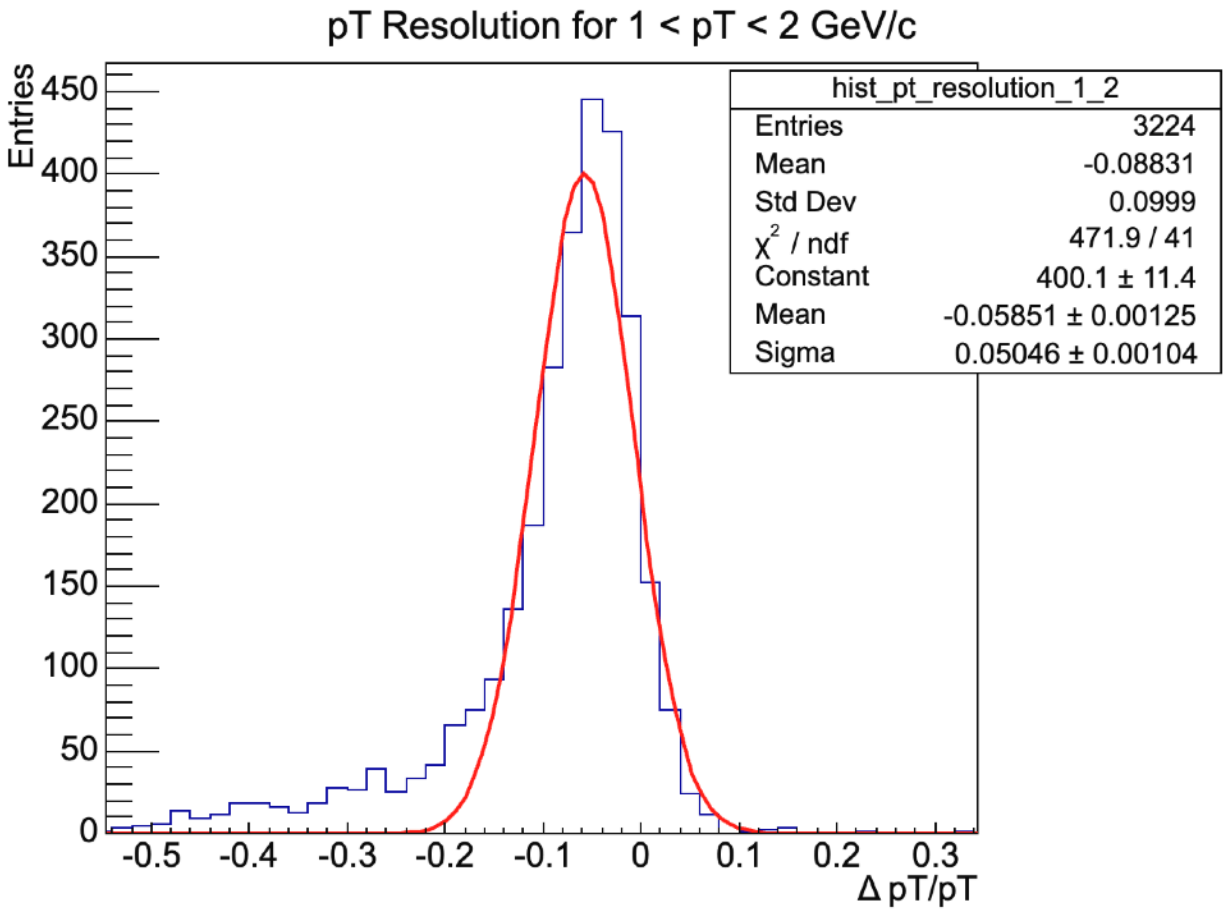
```
//Here si_seed is extracted from
// si_seed = track->get_silicon_seed(); .. si_seed is TrackSeed class and
track is SvtxTrack class


 for (auto key_iter = si_seed->begin_cluster_keys(); key_iter != si_seed-
>end_cluster_keys(); ++key_iter)
      {
        const auto &cluster_key = *key_iter;
        trkrCluster = clustermap->findCluster(cluster_key);
        if (!trkrCluster)
        {
          continue;
        }
        if (TrkrDefs::getTrkrId(cluster_key) == TrkrDefs::TrkrId::mvtxId)
        {
          n_mvtx_clusters++;
        }
```

Summary

- Framework updated
- Documentation updated and framework has been tested with several setup and checked by beginner through instruction
- Feel free to add more sentences and comments on current documentation
- You can also make a change and commit on the GitHub(no longer my own private job)
- Would be great if other people also start to use this framework and merge our individual effort to make more keepable framework.

pT resolution Estimation



Fitting range fix around the peak region only

5% of pT resolution achieved.(Almost reach to goal)
pT = [1 GeV/c,2 GeV/c]
Genki's dphi resolution
at 1 GeV/c ~ 3%, at 2GeV/c ~ 3.76%
But here, dphi resolution is estimated with fixed eta

2nd report: Summary

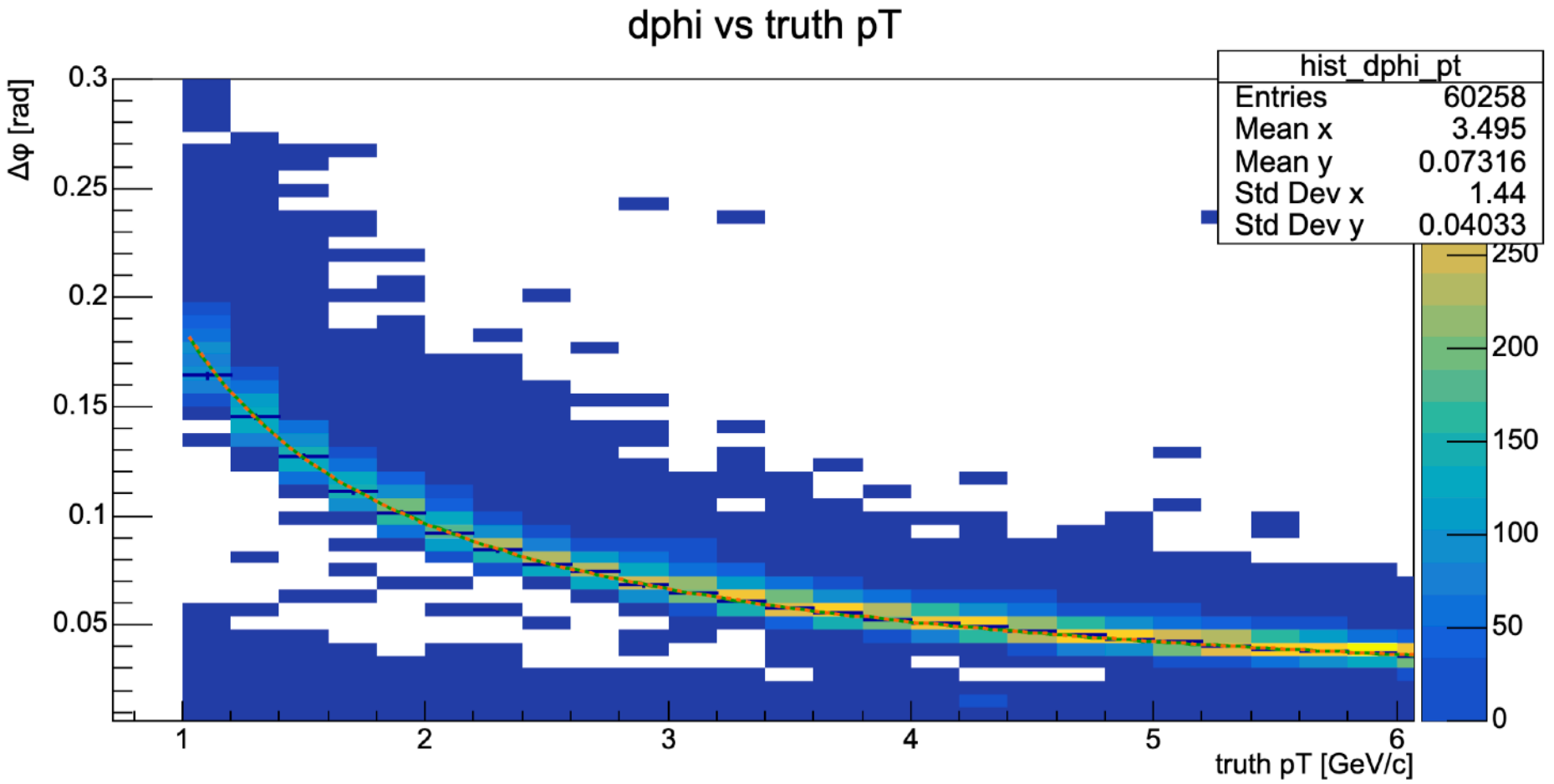
Run	#event	Beam	Momentum (GeV/c)	ϕ dist (rad)	η dist	vertex (cm)	with EMCal hits asso. with beam Δϕ mean (mrad)	Δϕ std. dev. (mrad)	Δϕ resolution (%)
4	5.4E+04	μ⁻	0.5	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-385.3	8.7	2.25%
1	5.0E+04	μ⁻	1	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-186.0	3.9	2.07%
2	4.9E+04	μ⁻	2	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-91.9	1.9	2.10%
3	5.0E+04	μ⁻	4	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-45.7	2.1	4.54%
9	5.6E+04	e⁻	0.5	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-392.1	12.3	3.14%
5	5.1E+04	e⁻	1	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-189.4	5.8	3.07%
6	2.9E+04	e⁻	2	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-94.3	3.5	3.76%
7	4.1E+04	e⁻	4	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-50.6	4.7	9.27%
8	4.5E+04	e⁻	8	[-π, π]	fixed at 0	fixed at (0, 0, 0)	-27.4	4.4	16.01%

pT resolution Estimation

We have to try to reduce the effect of the width from eta effect / radius effect

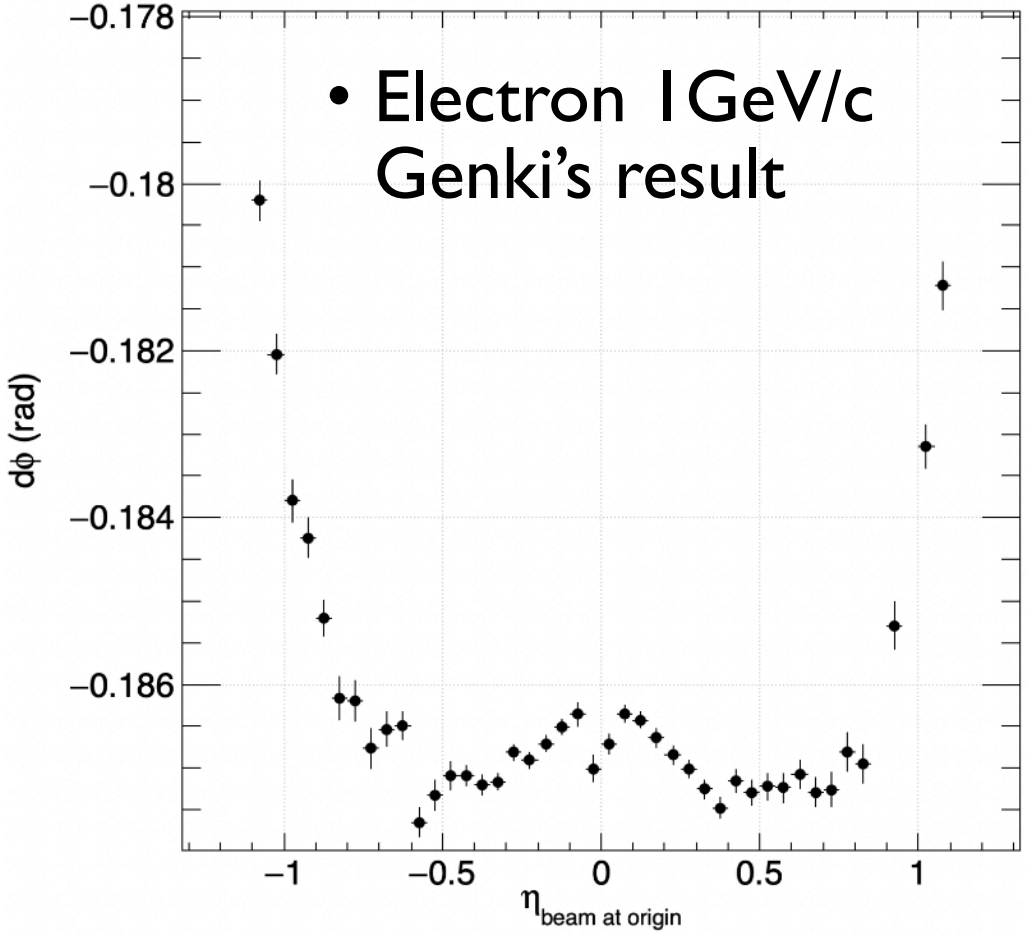
$$\Delta\phi = a + \frac{b*f(\eta)}{pT}$$

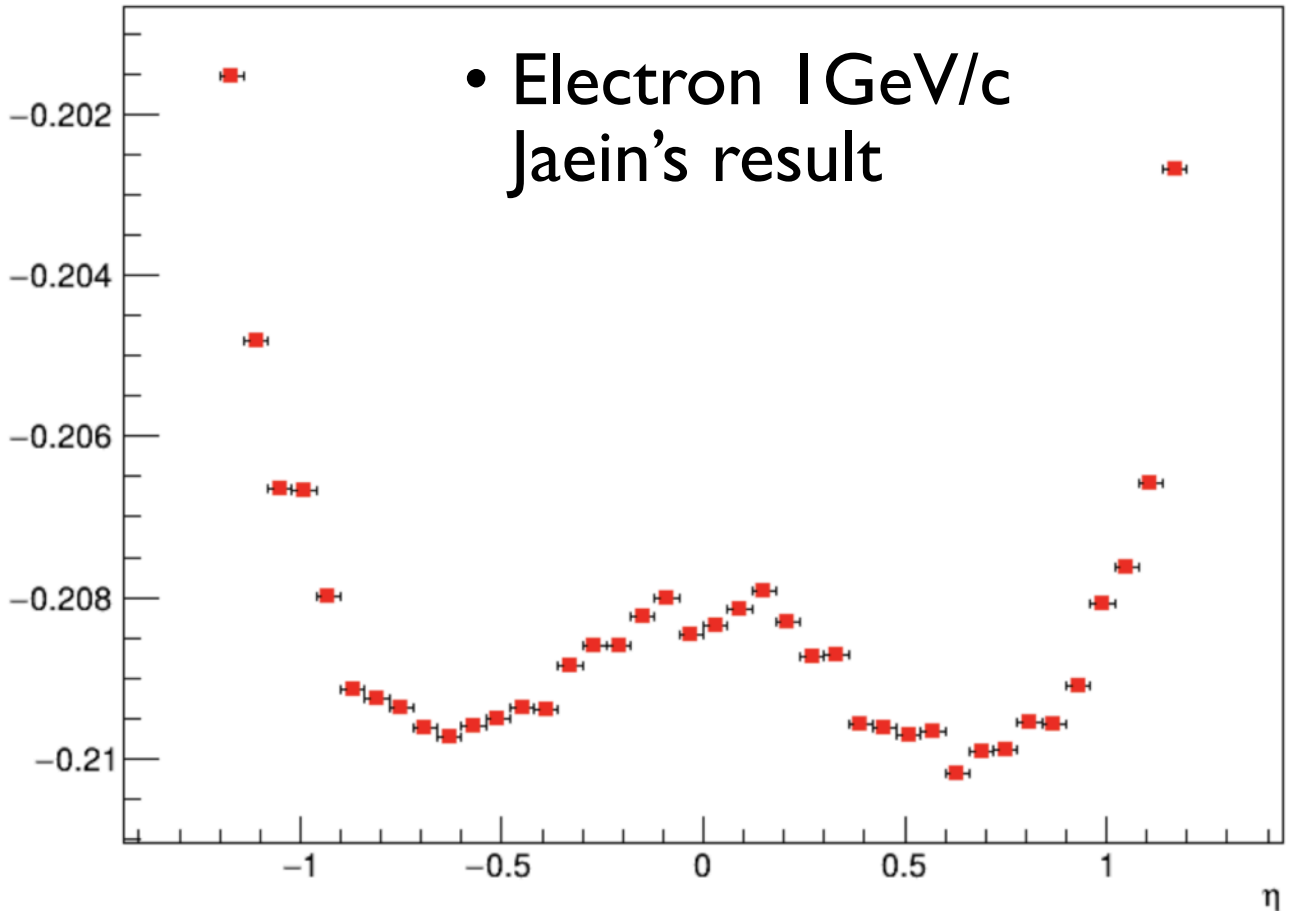
$$p_T = b * \frac{f(\eta)}{\Delta\phi - a}$$



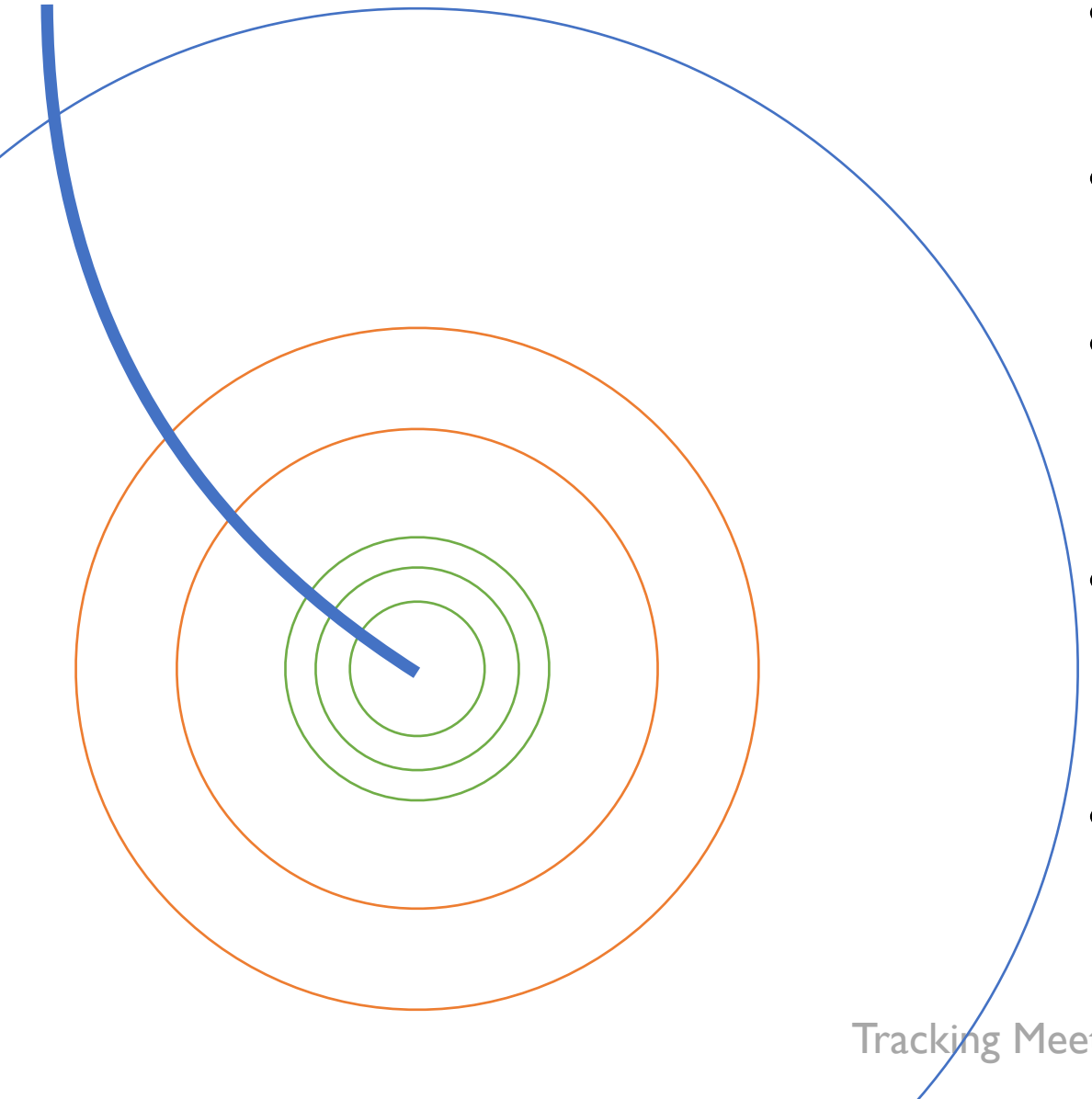
Estimate pT resolution based on dp

- dphi-eta shows waving shape from both of us.





Recap



- SvtxTrack class includes various kinematic variables
- Track->get_x() get_phi(), get_z()…. Default returns the track info at the R=0
- Need projection to EMCal surface and extract correct information for Si-Calo Matching
- Software is available (not easy to use unless you know how to use it..)
- Preparing macro so people can use it..

Check eta dependence of pT-dphi relation

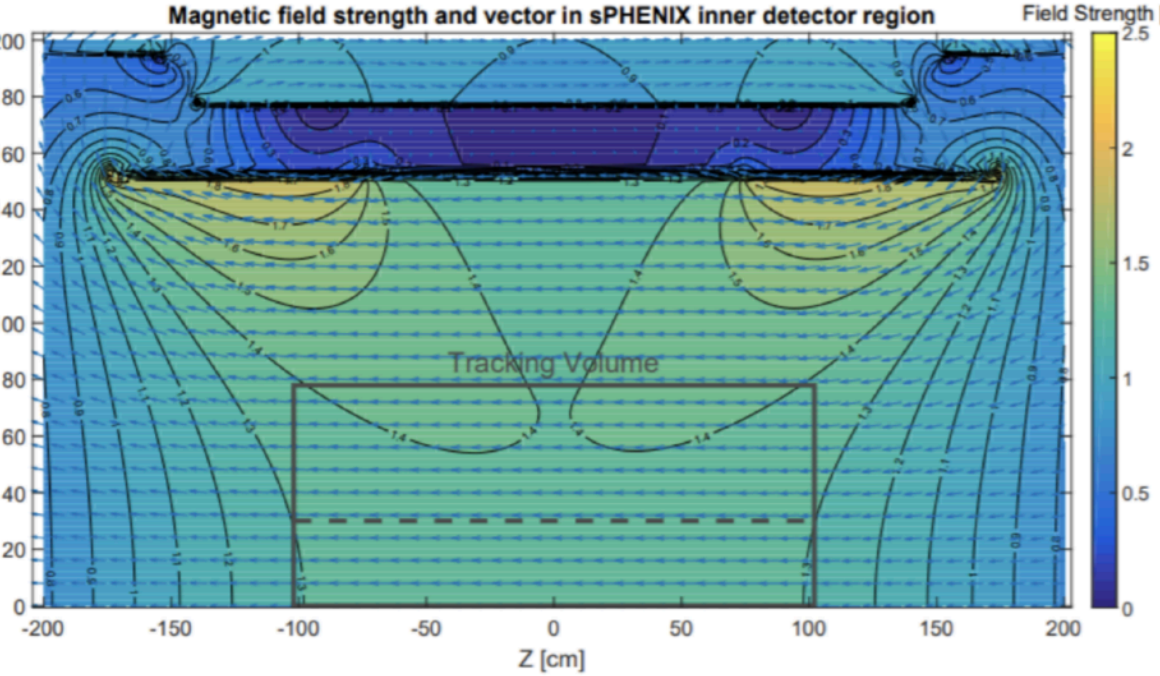
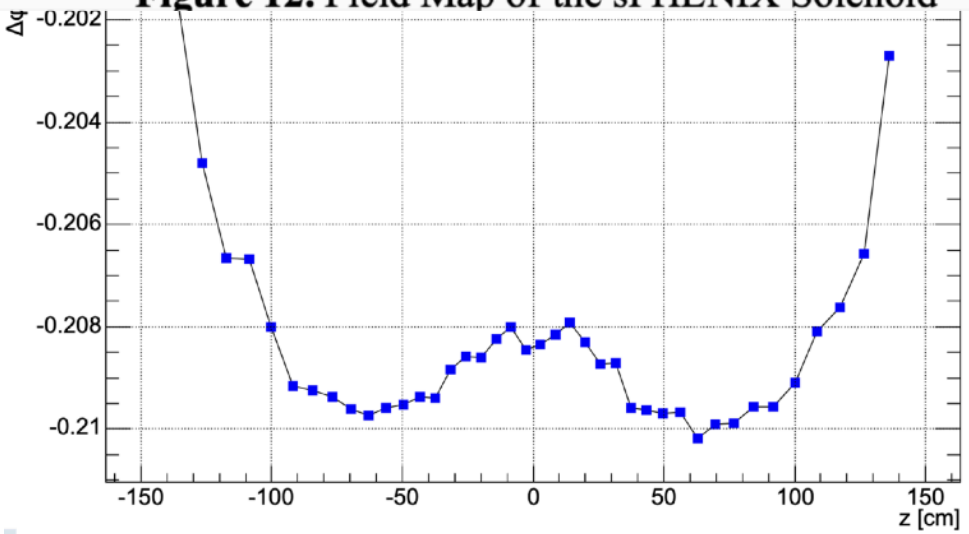
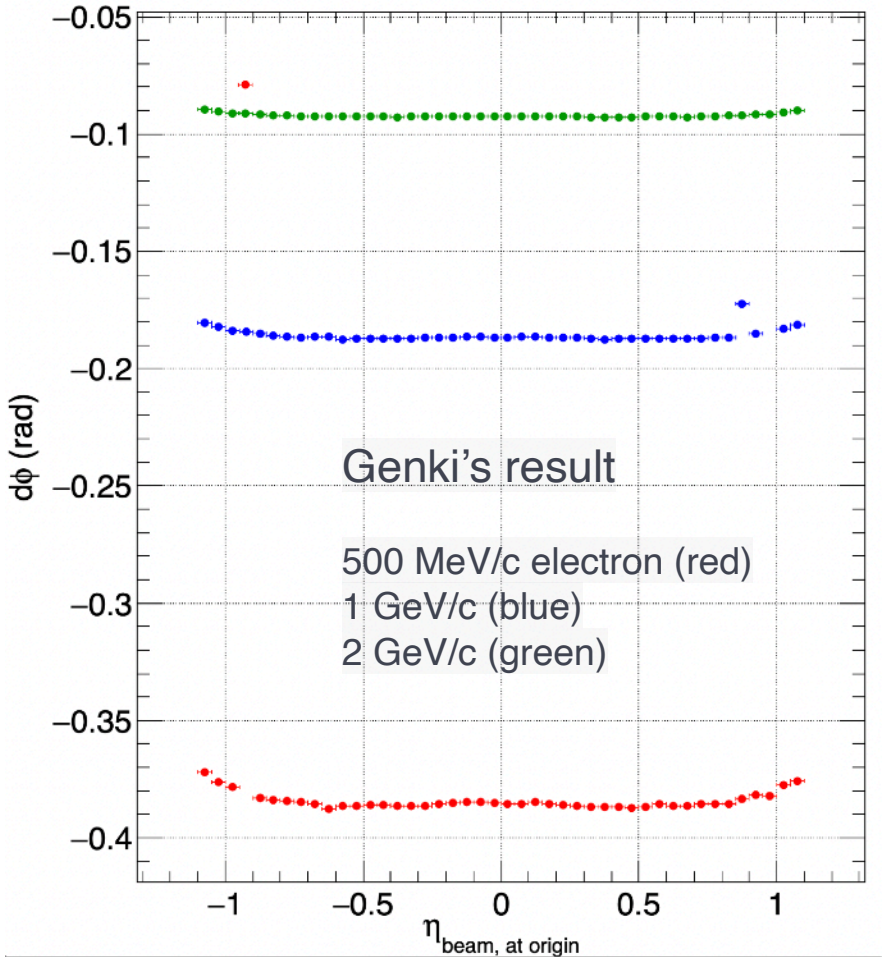


Figure 12. Field Map of the sPHENIX Solenoid



- Magnetic field map effect



Recap

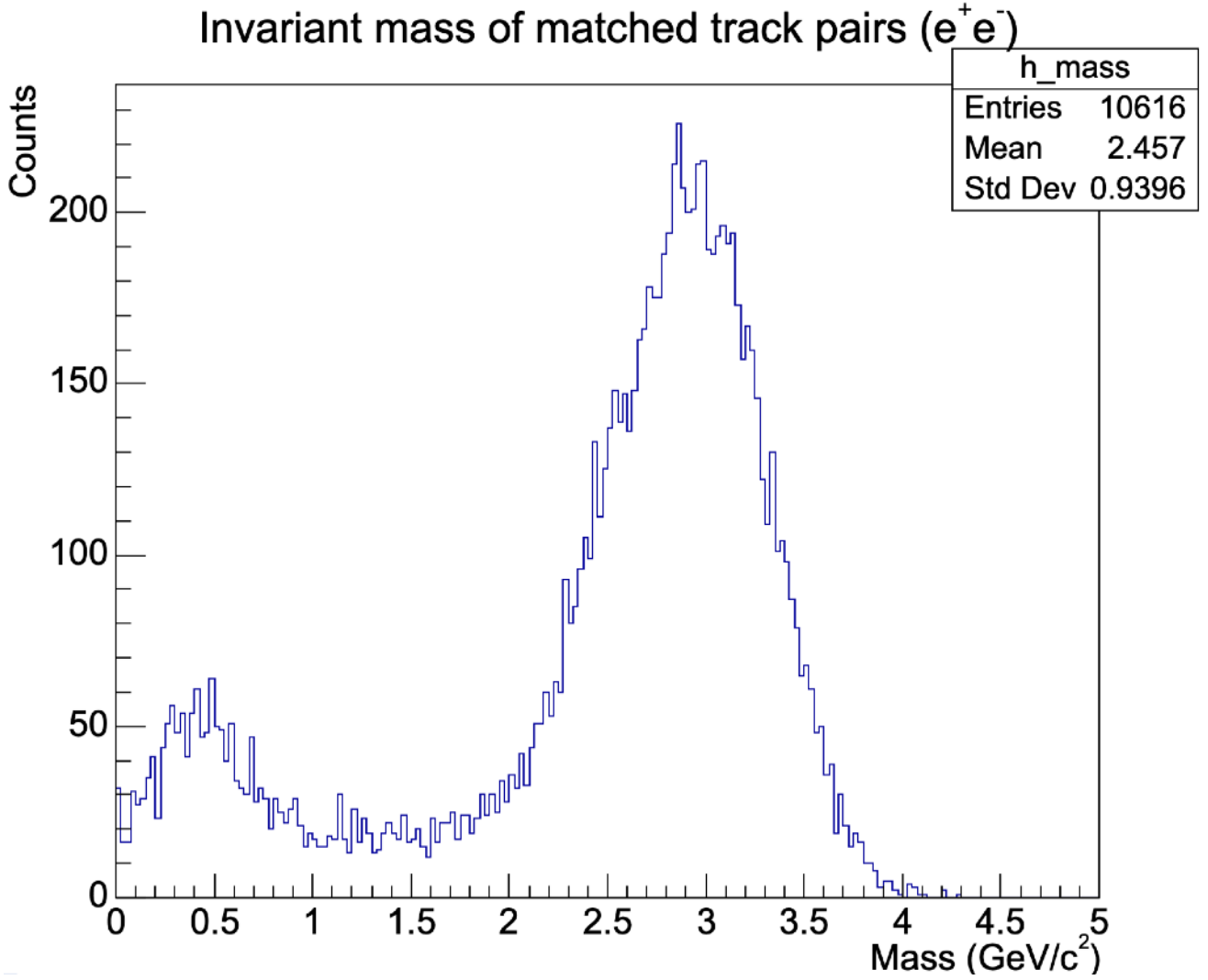
Basic Matching criteria

Minimum $\Delta R = \sqrt{(\Delta z)^2 + (\Delta \phi)^2}$

(Planning to change to $\Delta R = \sqrt{(\Delta \eta)^2 + (\Delta \phi)^2}$

Additional cut

0.8 < E/p <1.2
dz < 4 cm
pt > 0.5 GeV
nINTT > 1 && nMVTX > 2
Chi2/ndf < 4
opposite sign

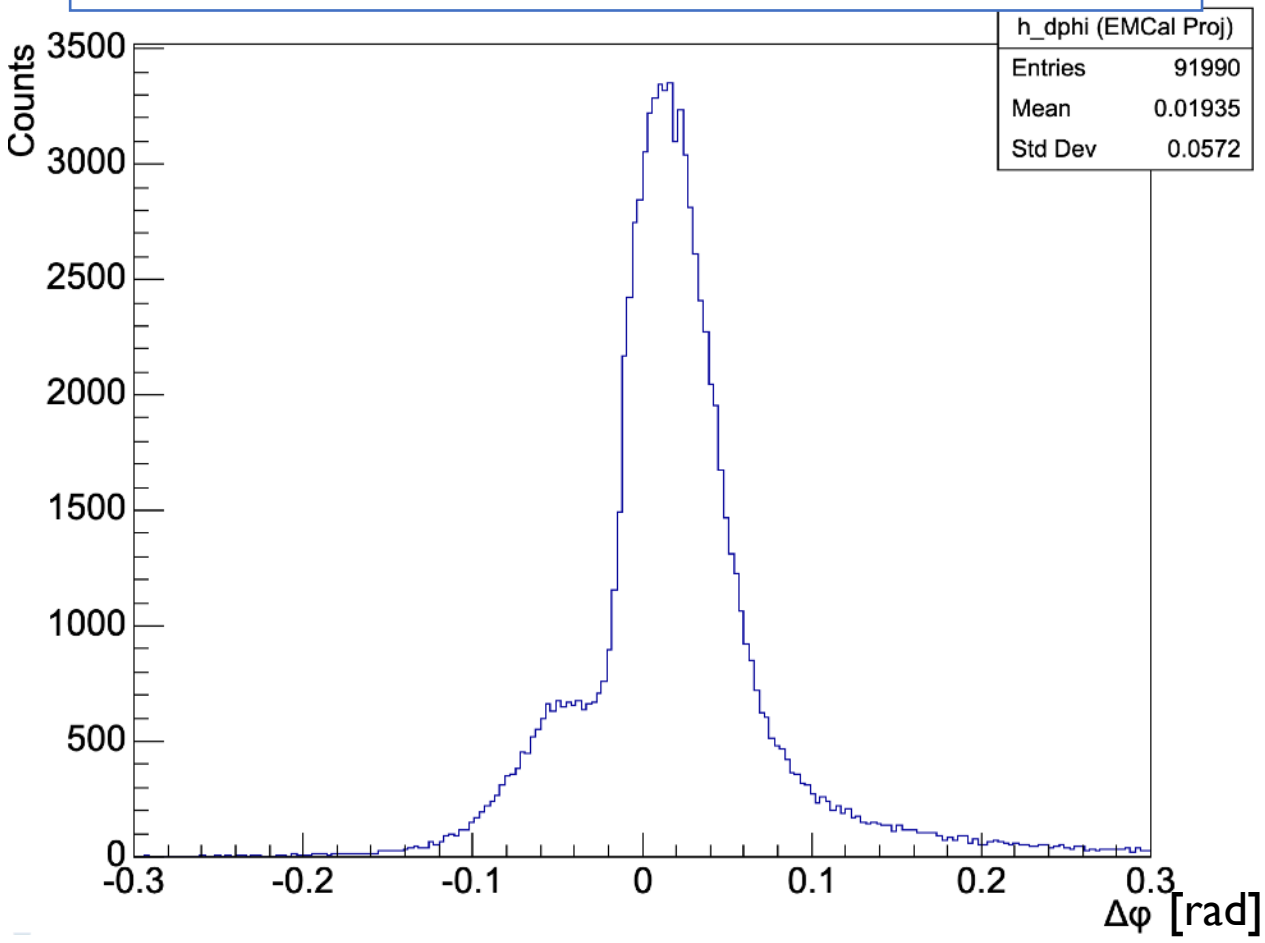


THIS IS NOT Physics yet! Just for fun and testing the algorithm

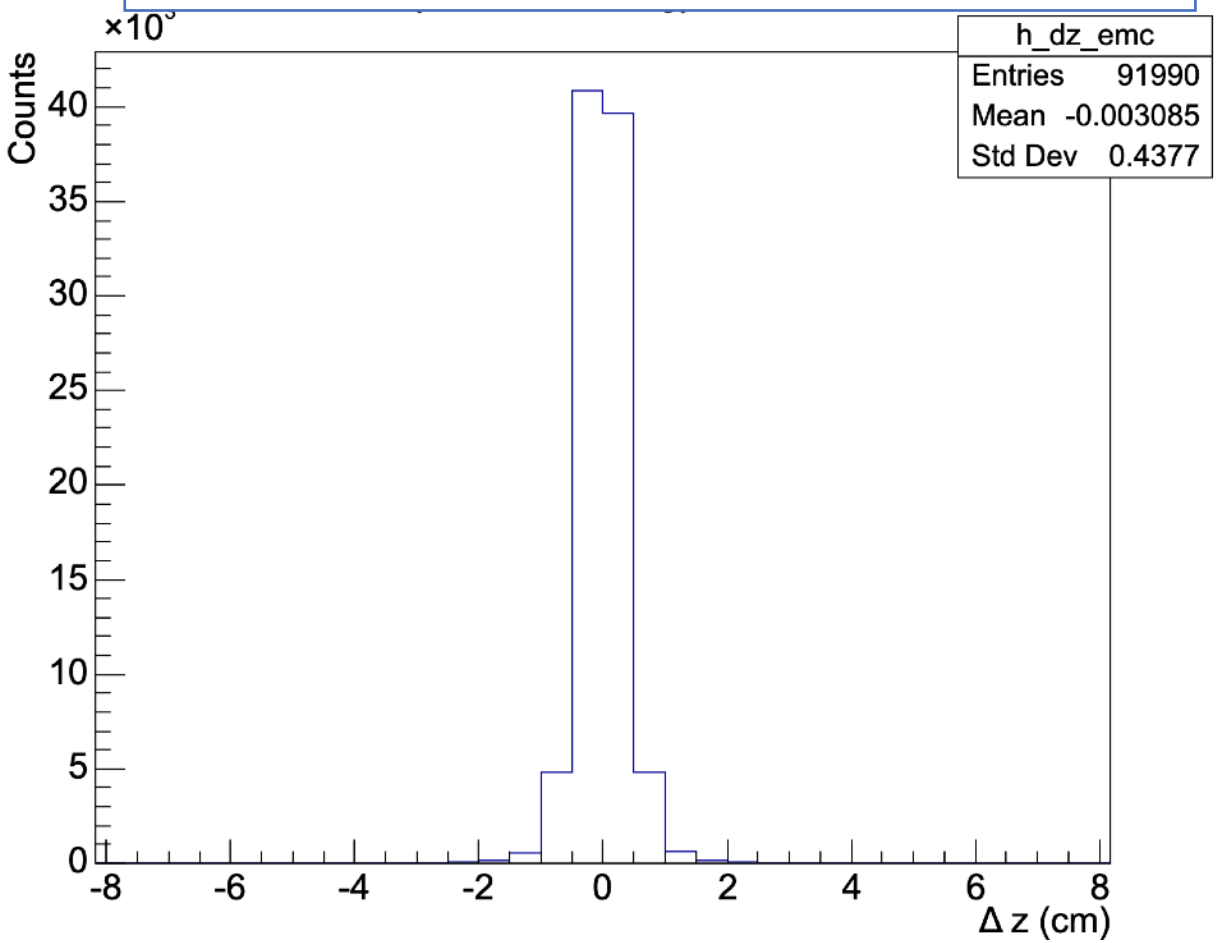
Projection test with electron gun

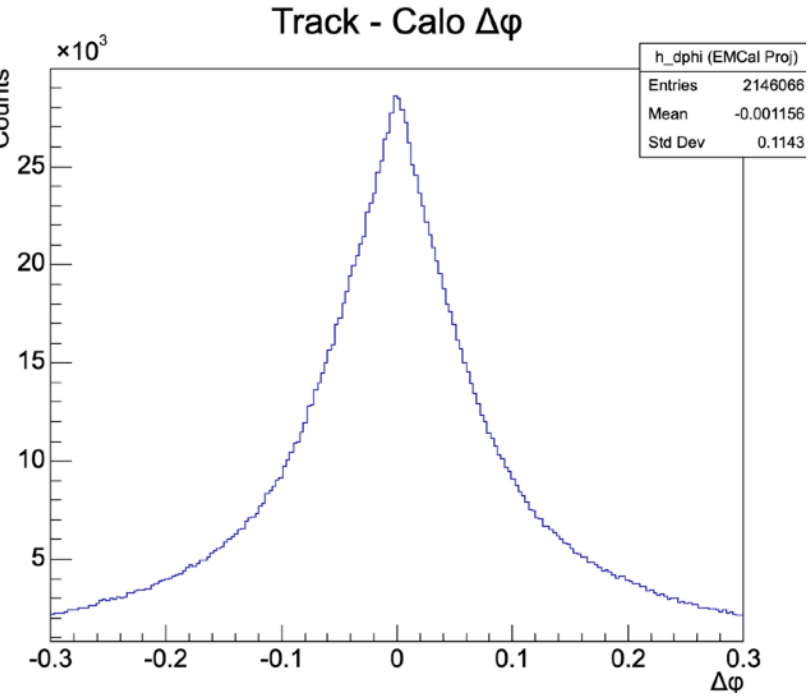
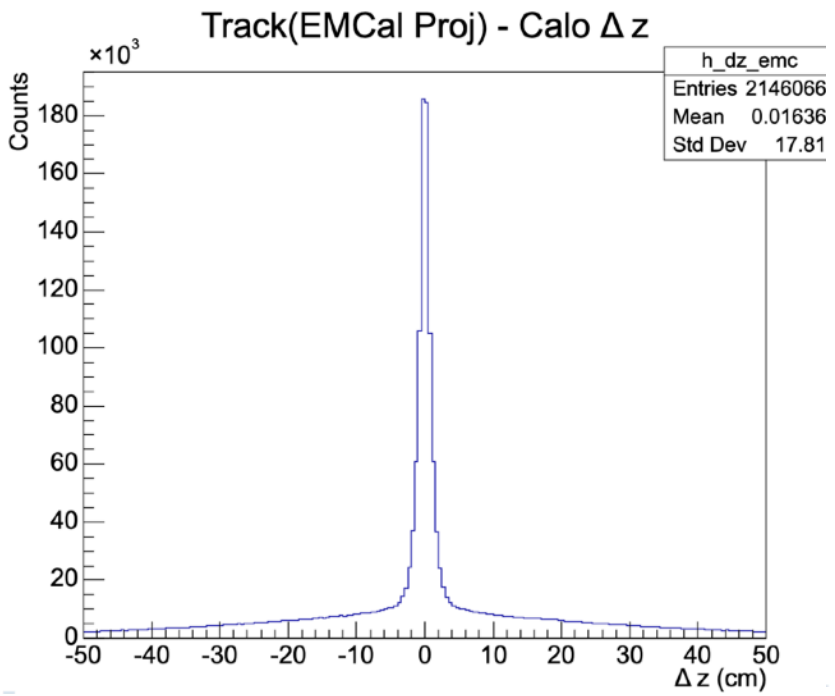
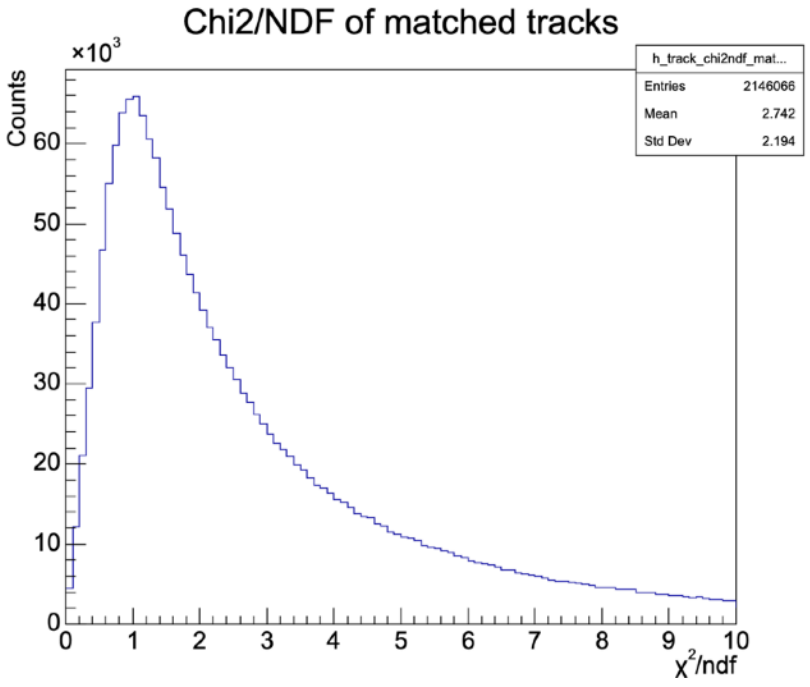
https://github.com/gwd213/INTT/tree/main/general_codes/Jaein/SiliconSeeding

[Track phi at EMCal surface] - [Calo phi]



[Track z at EMCal surface] - [Calo z]



Trial for Si-Calo matching

Try to Si-Calo matching .. electron gun might not good enough..
Try J/ψ reconstruction with J/ψ gun! -> more than 1 track!
also Good test tool for E/p cut ; rejection for di-muon / hadronic decay