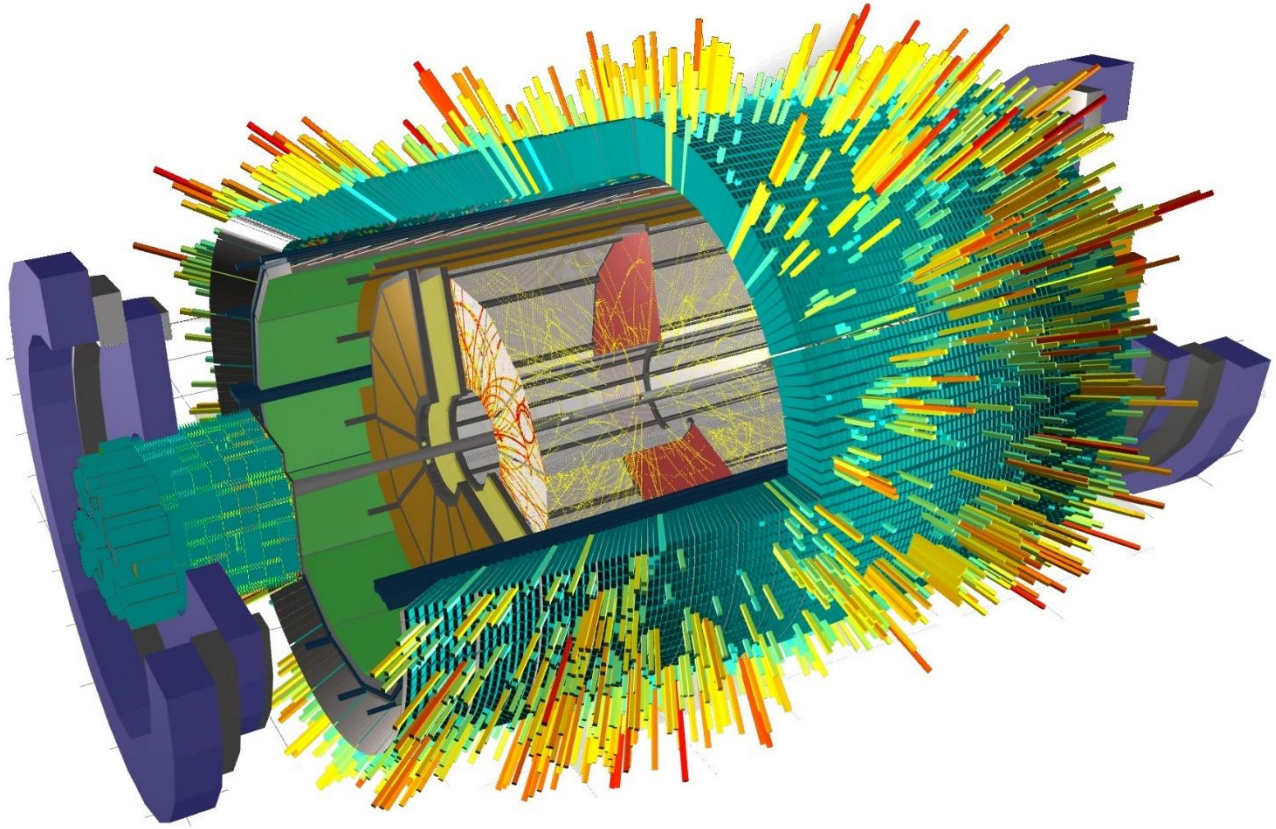
Implementation of ACTS Tracking with PID ML in MPD TPC

HNATIC Slavomir

on behalf of the

MPD Software Development & Computing Team

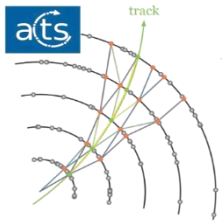


OUTLINE

- ACTS in MPD
- ACTS with PID ML (preliminary)
- Ionization Losses
- Required Upgrades & New Developments
  - MLEM for Fast Clustering
  - Multi-pass tracking
  - Occupancy-dependent Vertexing
  - Ambiguity Resolution upgrades
  - Refitting options
  - TOF matching upgrades



ACTS: A COMMON TRACKING SYSTEM

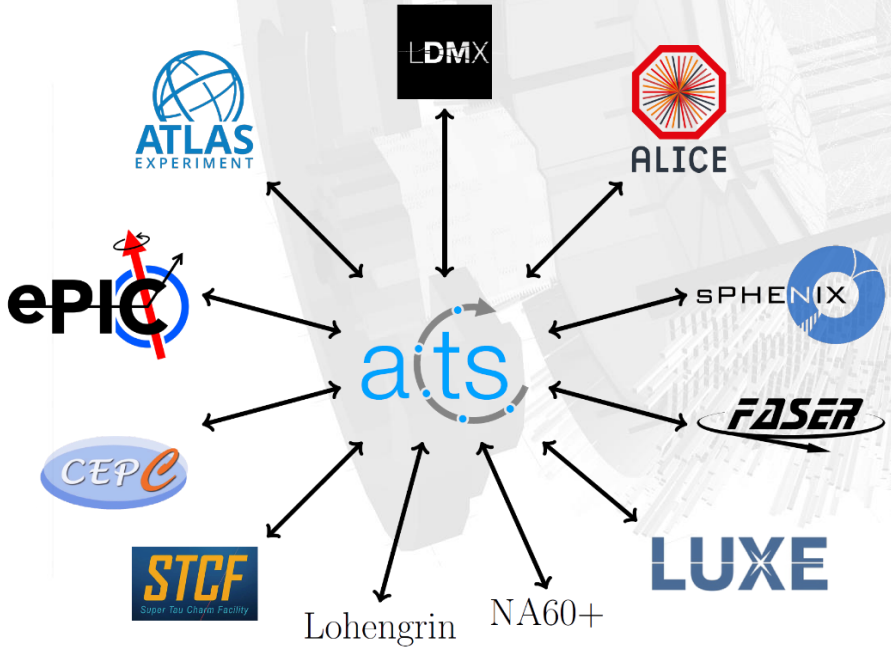


Track Finding with Combinatorial Kalman Filter

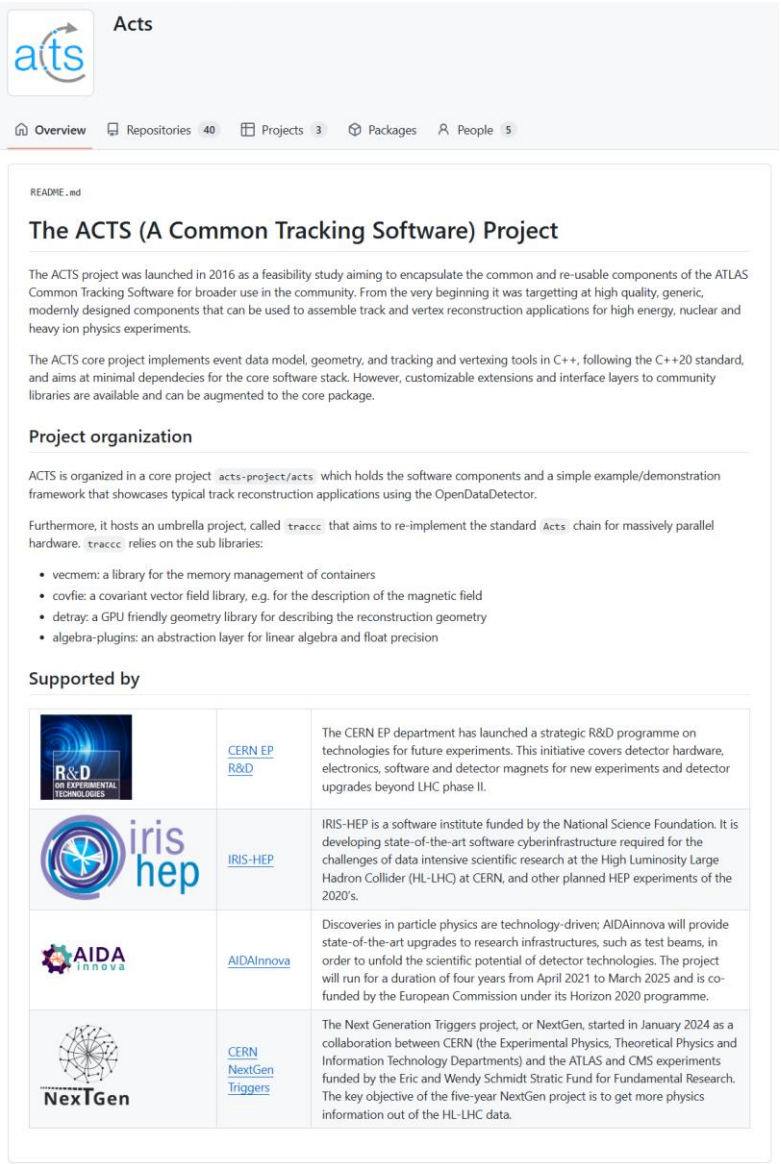
- uses KF on multiple branches

Vertex Finding

State-of-art development technologies



"The ACTS Project"
A. Salzburger et al.
ACTS for Nuclear Physics, Berkeley, 2025



Acts

Overview | Repositories 40 | Projects 3 | Packages | People 5

README.md

The ACTS (A Common Tracking Software) Project

The ACTS project was launched in 2016 as a feasibility study aiming to encapsulate the common and re-usable components of the ATLAS Common Tracking Software for broader use in the community. From the very beginning it was targetting at high quality, generic, modernly designed components that can be used to assemble track and vertex reconstruction applications for high energy, nuclear and heavy ion physics experiments.

The ACTS core project implements event data model, geometry, and tracking and vertexing tools in C++, following the C++20 standard, and aims at minimal dependecies for the core software stack. However, customizable extensions and interface layers to community libraries are available and can be augmented to the core package.

Project organization

ACTS is organized in a core project `acts-project/acts` which holds the software components and a simple example/demonstration framework that showcases typical track reconstruction applications using the OpenDataDetector.

Furthermore, it hosts an umbrella project, called `traccc` that aims to re-implement the standard `Acts` chain for massively parallel hardware. `traccc` relies on the sub libraries:

- vecmem: a library for the memory management of containers
- covfie: a covariant vector field library, e.g. for the description of the magnetic field
- detray: a GPU friendly geometry library for describing the reconstruction geometry
- algebra-plugins: an abstraction layer for linear algebra and float precision

Supported by

R&D on Experimental Technologies	CERN EP R&D	The CERN EP department has launched a strategic R&D programme on technologies for future experiments. This initiative covers detector hardware, electronics, software and detector magnets for new experiments and detector upgrades beyond LHC phase II.
iris hep	IRIS-HEP	IRIS-HEP is a software institute funded by the National Science Foundation. It is developing state-of-the-art software cyberinfrastructure required for the challenges of data intensive scientific research at the High Luminosity Large Hadron Collider (HL-LHC) at CERN, and other planned HEP experiments of the 2020's.
AIDA innova	AIDAInnova	Discoveries in particle physics are technology-driven; AIDAinnova will provide state-of-the-art upgrades to research infrastructures, such as test beams, in order to unfold the scientific potential of detector technologies. The project will run for a duration of four years from April 2021 to March 2025 and is co-funded by the European Commission under its Horizon 2020 programme.
NexTGen	CERN NextGen Triggers	The Next Generation Triggers project, or NextGen, started in January 2024 as a collaboration between CERN (the Experimental Physics, Theoretical Physics and Information Technology Departments) and the ATLAS and CMS experiments funded by the Eric and Wendy Schmidt Stratic Fund for Fundamental Research. The key objective of the five-year NextGen project is to get more physics information out of the HL-LHC data.

ACTS: A COMMON TRACKING SYSTEM

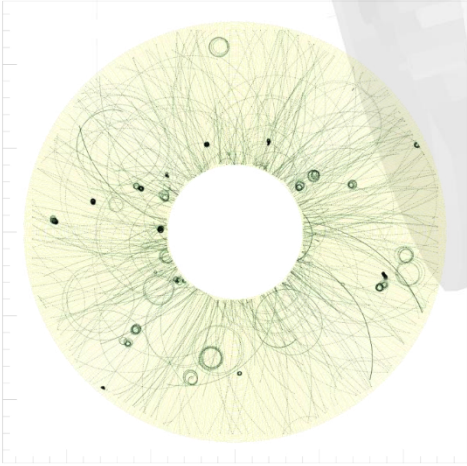
ACTS CKF STRATEGY
maximize efficiency
minimize fakes
many duplicates

CLEANUP & OPTIMIZATION
identify best tracks
discard duplicates
improve track quality

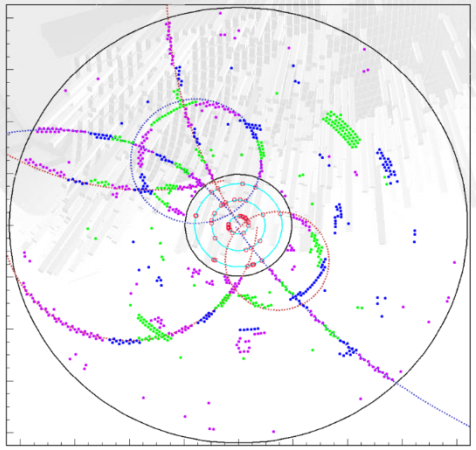
INPUT DATA
simulation,
experiment

KEY FACTOR – EVENT MULTIPLICITY

OUTPUT DATA
physical
analysis



MPD 100-1000

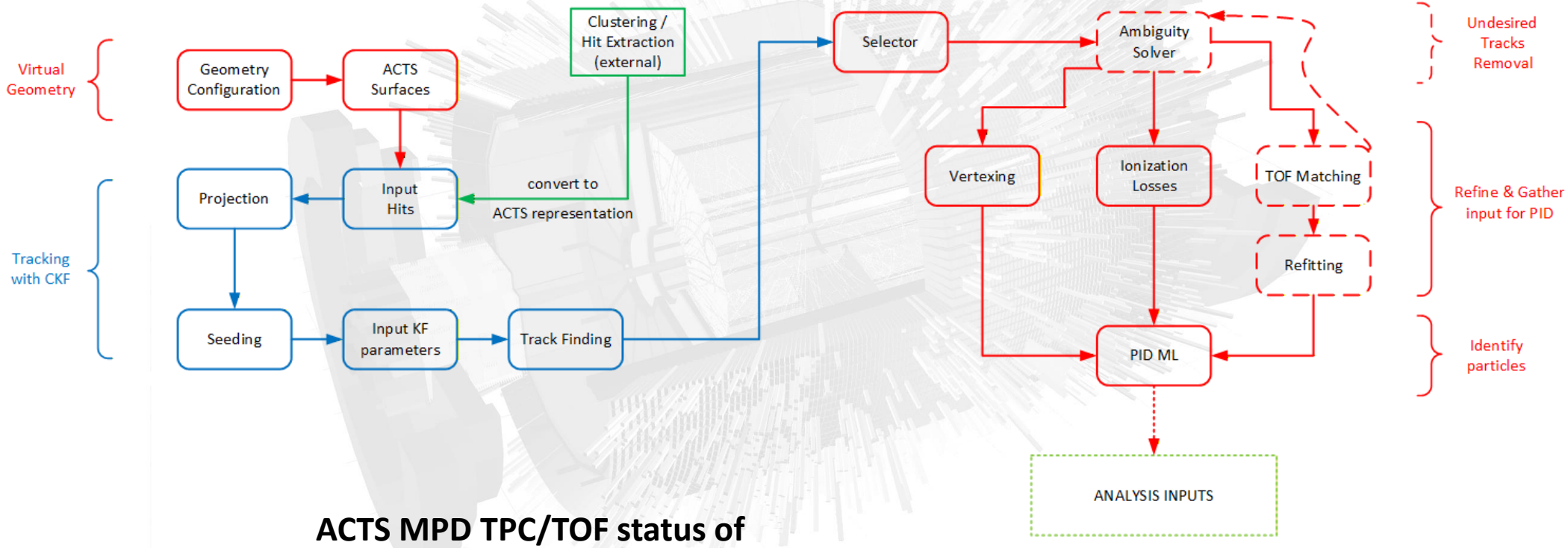


STCF 1-10

Algorithm encapsulation
naturally designed for testing and long term development

```
class SomeAlgorithm: public IAlgorithm {
public:
 struct Config {

   1. input/output data collections definition
   2. algorithm settings

 };

 SomeAlgorithm(config, log_level);
 ReturnStatus execute(context) const;

 const Config &config { return m_config; } const;

private:
 Config m_config;

 ReadDataHandle<inputDataType_i> inputData_i;
 WriteDataHandle<outputDataType_j> outputData_j;

};
```

ACTS IN MPDROOT



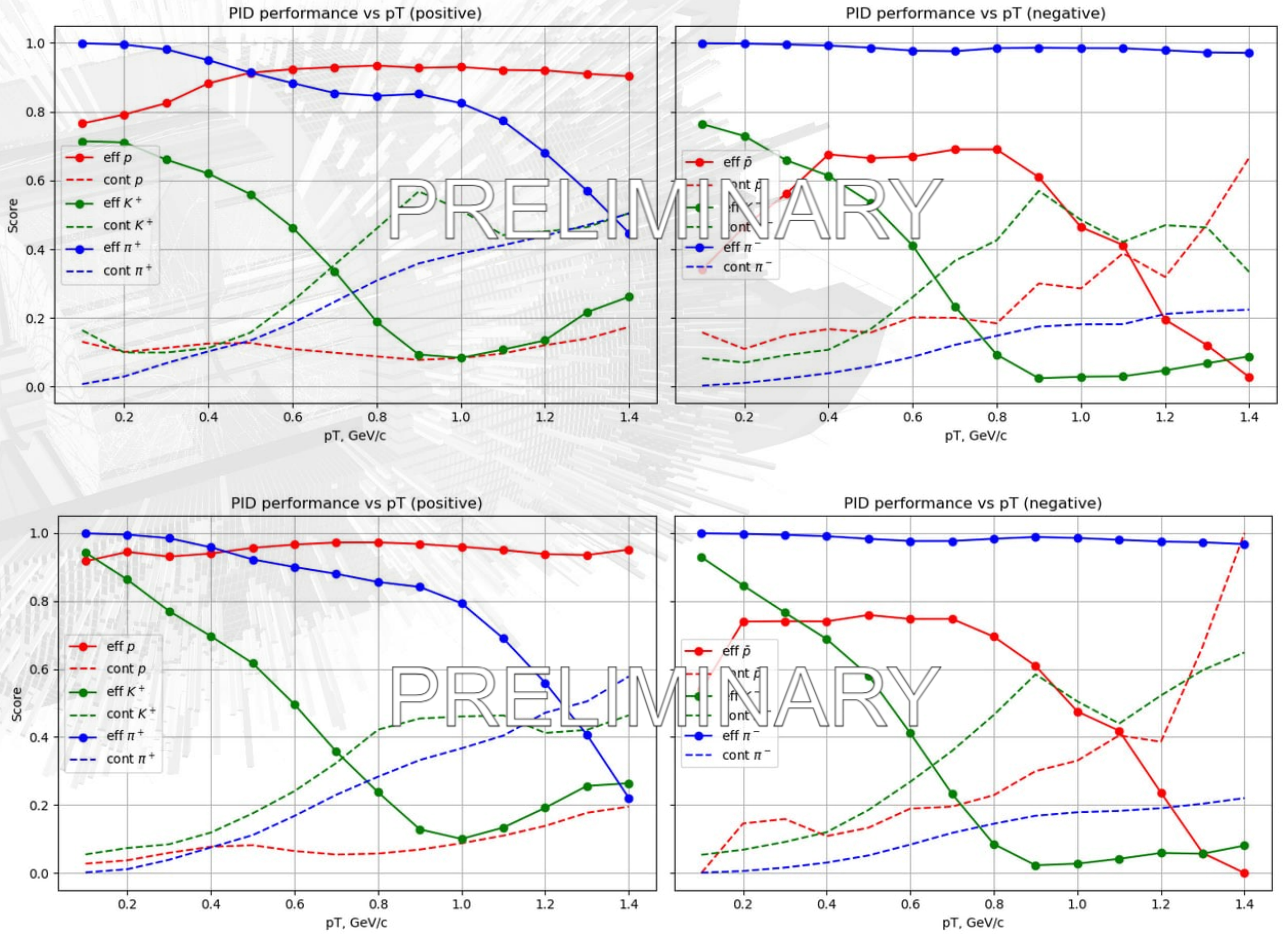
ACTS MPD TPC/TOF status of reconstruction backend architecture

ACTS WITH PID ML

PID ML in MPDRoot

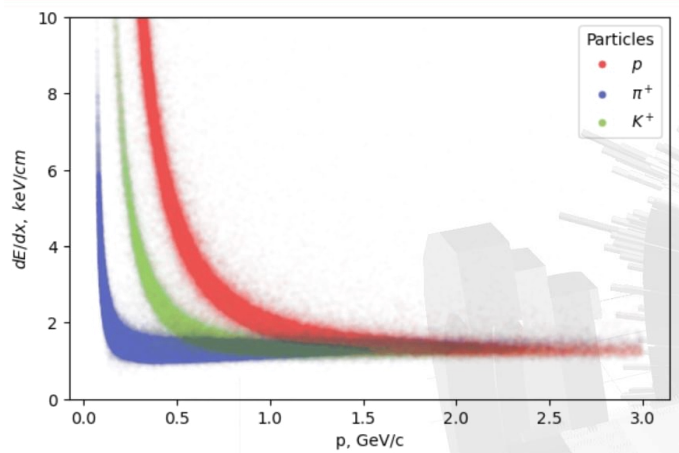
- integrated in future_main branch
- author V.V.Papoyan
- use any trained model in MPDRoot
- understands how many input features and list of particle types it can predict
- returns predicted particle classes (probabilities) with PDG for analysis

TPC only - UrQMD, BiBi 9.2 GeV



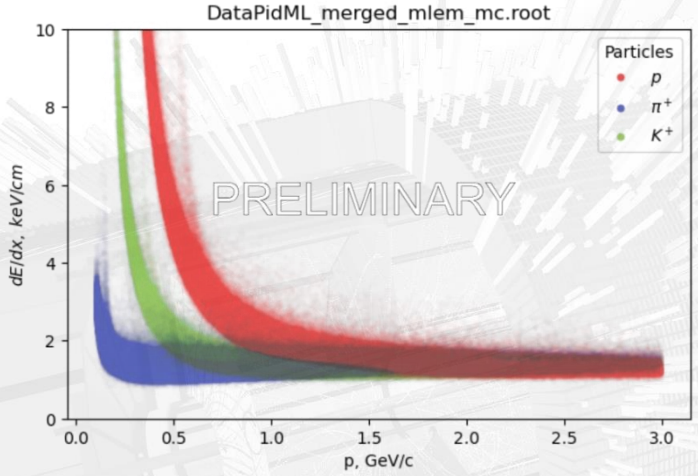
IONIZATION LOSSES

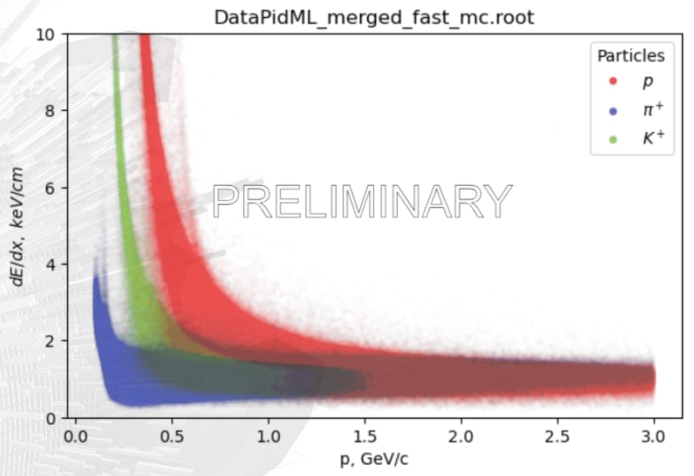
Default tracker

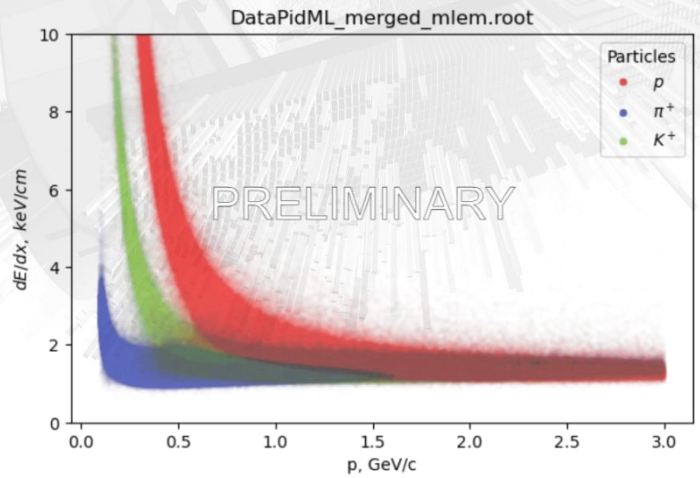


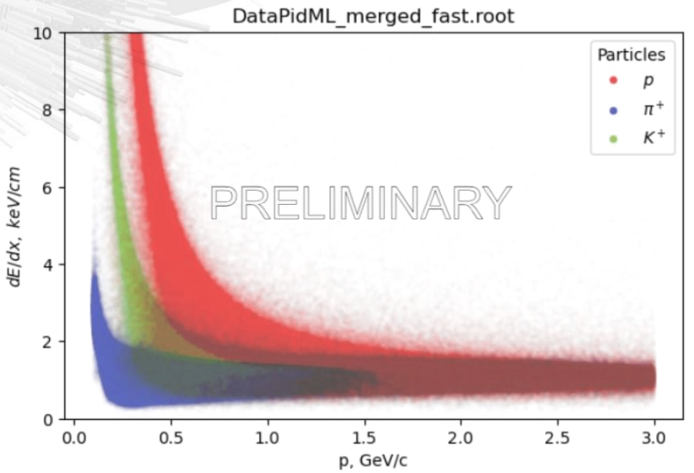
Noise coming from

- Signal reconstruction not addressed properly yet
- Not the best tracks selected from CKF output
- Corrections for energy losses at low p present in default tracker (uses some MC truth)









FAST CLUSTERING

How to run:

toolbox enter a9-nica-dev

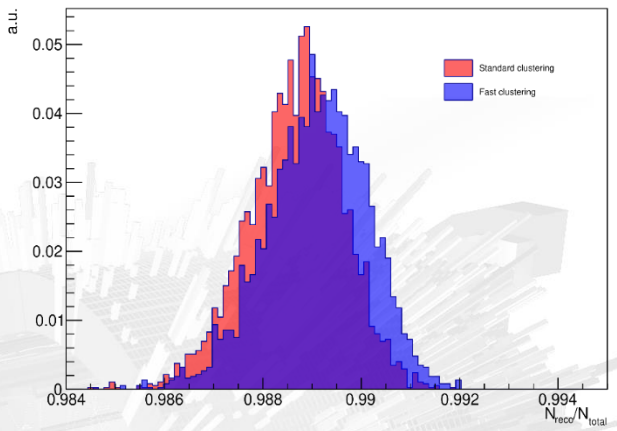
module add mpddev ACTS

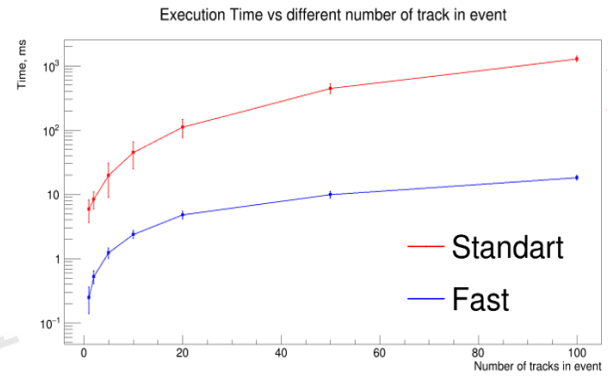
build mpdroot's dev branch

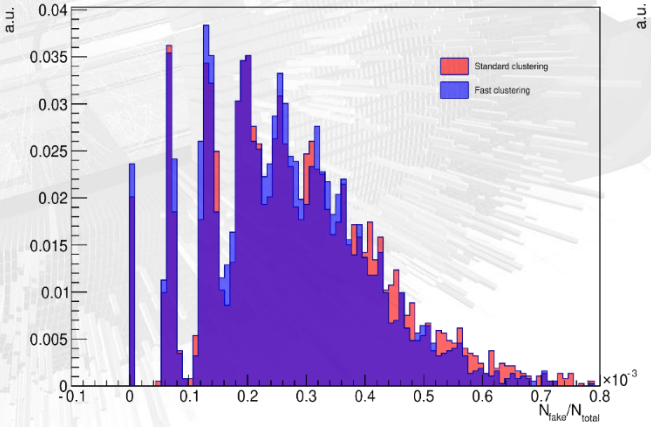
execute runReco.C with

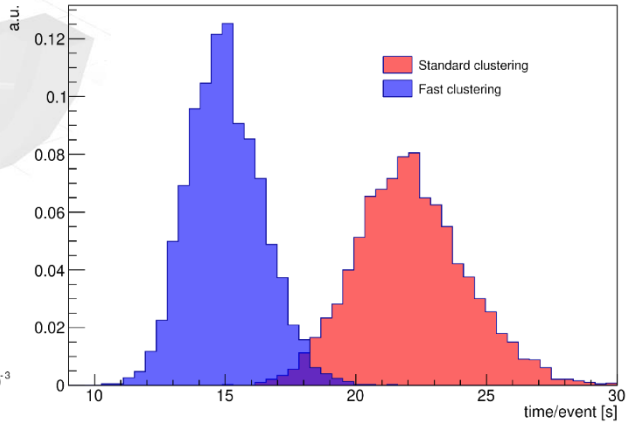
ETpcClustering::FAST, ETpcTracking::ACTS

BiBi 9.2 GeV
UrQMD
3000 x 100 events
Min hits 9,12









	FAST	STANDARD
Efficiency	μ = 98.9099 % σ = 0.1022 %	μ = 98.8663 % σ = 0.0968 %
Fakes	μ = 0.02645 % σ = 0.01329 %	μ = 0.02806 % σ = 0.01438 %
Speed	μ = 15 s / event σ = 1.42 s / event	μ = 22.12 s / event σ = 2.11 s / event

Experiment

- SAMPA data
- detector calibration parameters
- noise information
- no mc branch

FAST CLUSTERING - MLEM UPGRADE

Maximum Likelihood Expectation Maximization

- MLEM also used in gamma-ray astronomy to achieve high detection precision - distinguishing 2 close cosmic sources

- Used to distinguish 2 close tracks in TPC
- Current COCG hit extraction sufficient only for simple clusters: single maximum, no charge overflow, not at edge

- All other clusters must be processed in a more sophisticated way
- ***Point Spread Function*** *(single electron cluster shape)* – describes how electron cluster blurs as it travels through TPC (drift time, dip angle, crossing angle)
  From it we can get whether large blob travelled long distance or is caused by 2 sources (particles)
- ***Coupling Matrix –*** geometrical charge connection between TPC volume point and detector sensors
- ***Mlem procedure*** uses those to iteratively compare predictions with reality. Usually ran twice: original cluster and created sub-pixel (bin) grid is created for better resolution. Output local maxima are checked for large enough dips to make sure no incorrect hit divides are made.

Other corrections

- Dip angle corrections (shape adjustment, Z-vertex estimate)
- Edge bias corrections (signal distortion)
- Visibility corrections from PSF (missing charge adjustment)

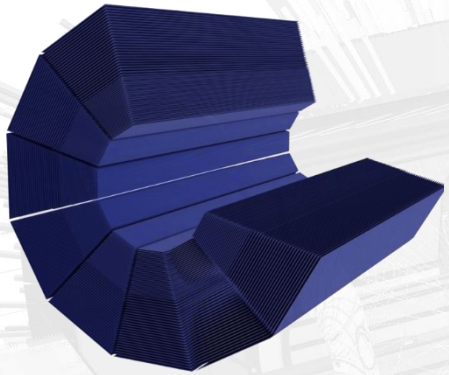
Without MLEM a complex cluster treated by COCG gives a ghost position with off-the-wall charge value

MLEM
MC track number:418, P: 0.379592, Pt: 0.276917, Eta: -0.836533
DeDx: 1.4417 nMeasurements: 40 nOutliers: 2
P: 0.382864 Phi: 0.428273 Theta: 2.31971 Pt: 0.280422 Eta: -0.83068

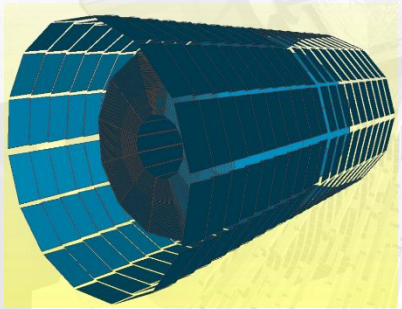
FAST
MC track number:418, P: 0.379592, Pt: 0.276917, Eta: -0.836533
DeDx: 1.33656 nMeasurements: 38 nOutliers: 6
P: 0.385218 Phi: 0.425803 Theta: 2.32119 Pt: 0.281756 Eta: -0.832709

ACTS GEOMETRY

Native MPD TPC geometry
- 24 Sectors & 53 Padrows
- no conversion:
  Volumes & Surfaces are created and glued together directly



TOF detector added on top of TPC
- ACTS CKF acting in a single space where both TPC and TOF are located
- TOF matching



PRESENT STATUS

- fast, lightweight
- zero projection errors
NO ISSUES

FUTURE

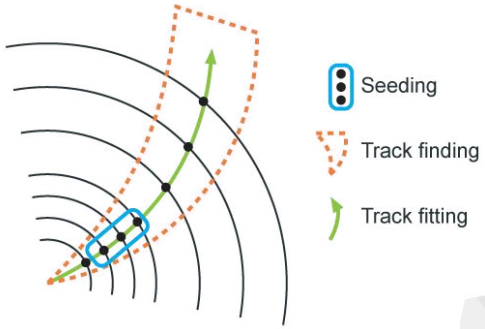
- real detector geometry

TO DO

- *currently, there are unrealistic constant variances during Measurement (CKF inputs) creation (some default values to make code run)*
- *pessimistic for close hits, optimistic for far hits, CKF creating track wiggles*

To be estimated and additionally tuned
- *drift distance/diffusion based for TPC hits*
- *static hardware based for TOF hits*

SEEDING



ATLAS – like seeding inwards to outwards

- Current configuration settings very “soft”
- Combinatorial explosion at high occupancy events
- Additional tune-up inevitable for experiment

Investigate lower values for:

maxSeedsPerSpM = 3 (too many seeds now)
ImpactMax = 30 (too many secondaries now)

Investigate various values of deltaR ranges,
deltaZMax = 200mm probably too big

TO DO: Multi-pass Seeding / Reconstruction for high occupancy events

Pass 1: Find easy, high quality tracks
- Consecutive padrow strategy (Pt > 500MeV, eta < 1.2)
- Outward to inward seeding not native for ACTS, but it can be tricked to do so. Condition – they should not deviate too much from estimated vertex (0,0,estimatedZ)
- Filter duplicates

Pass 2: High eta tracks (do not reach TOF)
- inward to outward seeding

Pass 3: Low momentum curved tracks
- loose curvature constraints for seeds
- inward to outward seeding

Secondaries (pass 4)

Issue – how to mask used hits between passes?
Try different than winner-takes-all approach, weight based system for shared hits

COMBINATORIAL KALMAN FILTER

TO DO: TUNE SETTINGS

Pass 1 (HQ tracks)
- branching: low, chi2cutoff: low
- minMeasurements: high
- How much can CKF look ahead: low
- Maximum missed rows: low
- BranchStopper configured to ensure long clean tracks

Pass 2 (forward tracks)
- branching: medium, chi2cutoff: medium
- minMeasurements: low
- How much can CKF look ahead: high
- Maximum missed rows: high
- BranchStopper configured to keep short tracks

Pass 3 (Low momentum tracks / secondaries)
- branching: high, chi2cutoff: high
- minMeasurements: medium
- How much can CKF look ahead: low
- Maximum missed rows: low
- BranchStopper configured to allow spirals

TRACK QUALITY

- *CKF Seed Deduplication* often discards best quality tracks, while keeping poor duplicates
- should be disabled and custom seed deduplicator written
- executed after track finding

VERTEXING

VERTEXING

- Provides **global constraint** for CKF
- best to run after CKF pass 1 stage before Ambiguity Resolution
- ACTS AMVF implemented, gives ok results (high multiplicity events)

VERTEX Z-ESTIMATE

- Statistical estimate from hits obtained before seeding
- Insufficient quality for low occupancy events

TODO

Low occupancy events

- Iterative Vertex Finder is a better choice

High occupancy events

- Run AMVF on pass 1 (HQ) tracks, use result to constrain pass 2, pass 3 tracks
- Use Gaussian Grid Seeder (better noise resistance) instead of default Adaptive Grid Seeder in AMVF

General

- Vertex constraints (d_0, estimated Z)
- Extrapolate tracks to found Vertex, refit

IONIZATION LOSSES – PART 2

IONIZATION LOSSES (dE/dx)

- calculated from clustering input
- formula from old tracker rewritten for ACTS that takes into account spatial and drift effects, removes delta electrons contribution (Landau tail)
- critical information for PID in TPC for tracks that do not reach TOF

TO DO

- use to filter tracks with inconsistent dE/dx profile (single stitched track from 2 particles flying close together)
- use in Ambiguity Resolution to filter duplicates (better track having better dE/dx profile due to picking less “bad” hits)

AMBIGUITY RESOLUTION

All available algorithms in ACTS currently **do not work** for MPD TPC

We have currently our own:

Coverage based – effective, ranks tracks based on quality, then identifies duplicates based on covered regions

Shared Hits Based – identifies duplicates based on shared hits

Low quality tracks filter – self explanatory

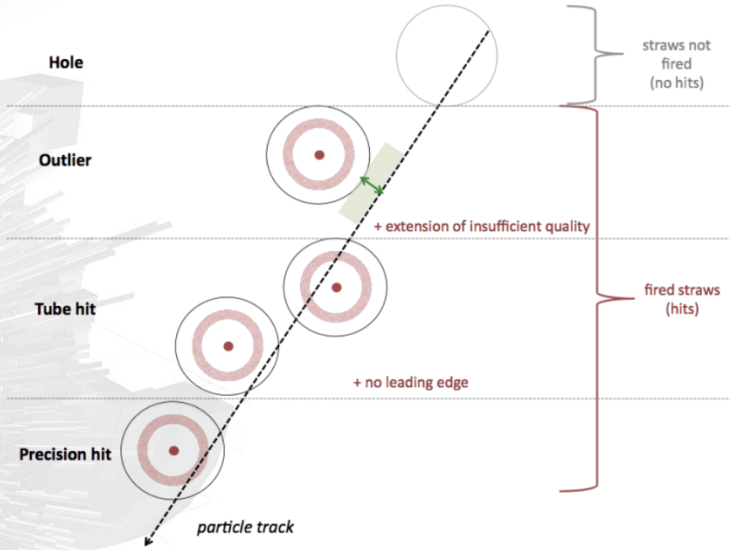
- work well in low occupancy events

High occupancy events

- current status: ~15% duplicates at ~96% efficiency, 1/10000 fakes
- will improve in multi-pass reconstruction
- some ACTS ambiguity algos likely to work in separate passes with separate settings (for example greedy algorithm in pass 1)

Technically most resource consuming and challenging task

- *in real experiment ambiguity must be removed at various reconstruction stages*



TO DO

- Seed track quality based Deduplication
- TOF shared hit filter
- Ionization losses filter

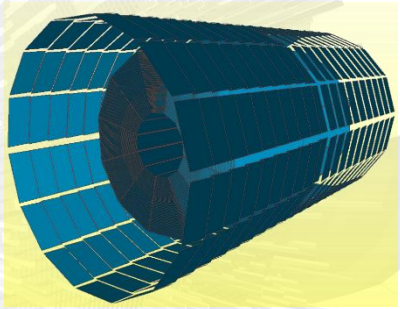
TOF

TOF matching is at the end of reconstruction chain,
very sensitive to track quality and therefore highly affected
by the current crude tracks and their cumulative error

CURRENT STATUS

- TOF geometry & and extrapolate tracks methods ready
- tracks that cannot reach TOF (~30%) and must be identified by dE/dx only
- tracks matched by CKF (~20%)
- tracks matched by propagating to RTOF and **(naïve approach)** looking for nearest neighbor (~30%)
- bad matches (bad quality tracks, secondary TOF hits)

We gave up on adapting ACTS TPC output to MpdTpcKalmanTrack class (no resources to deal with the old messy code having no solid API) to use existing TOF matching.



TOF & TPC
in the same space

TODO:

Reduce cumulative error first:

- improve overall track quality (multi-pass)
- reduce duplicates
- replace nearest neighbor by adaptive (T) **global** optimization TOF hit search strategy

REFITTING OPTIONS

Pass 1 (HQ tracks)

- fix TPC track to PV
- smooth in outward direction (KF refit)
- propagate to TOF surface, then TOF match
- global chi2 refit

Pass 2 (Forward tracks)

- might use Vertex-seeded CKF to find track
- run the smoother inwards (KF refit)
- if no PV found from pass 1, attempt to find PV

  after pass 2, if PV found – fix & refit inwards
- if no PV found at all (ultra peripheral event) –

  vertex blind refit (POCA, perigee refit)

Pass 3 (Low p_t tracks)

- might use Vertex-seeded CKF to find track
- run the smoother inwards (KF refit)
- remove loopers
- find TOF match for non-loopers
- KF refit with smoother inwards

TO DO

- KalmanFitter (Global chi2 ready)
- kill low p_t seeds & write looper-finding

  aborter ($\Delta p_r < 0$) for CKF

NEAR FUTURE

MAIN TASK

- Tracking for real experiment
- Clustering with MLEM / proper Ionization losses
- Multi-pass seeding, CKF, AR, Vertexing, Refit
- TOF matching algorithmic upgrade
- PID ML with/without TOF data

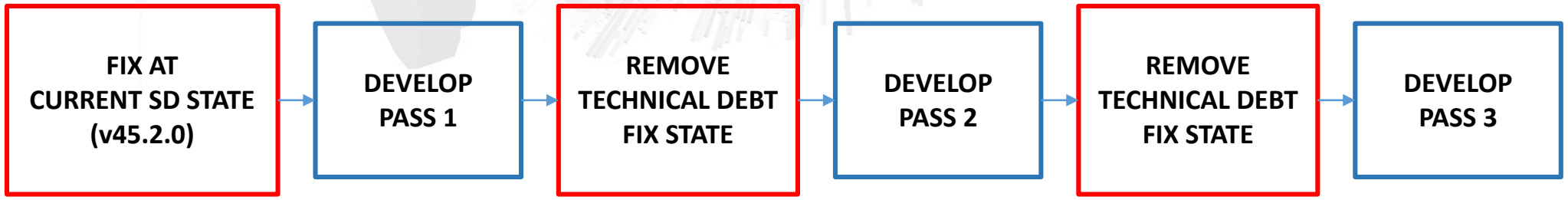
TECHNICAL DEBT

- Migration to future_main
- Upgrade to latest ACTS (v46.3.0)

SOFTWARE DEVELOPMENT

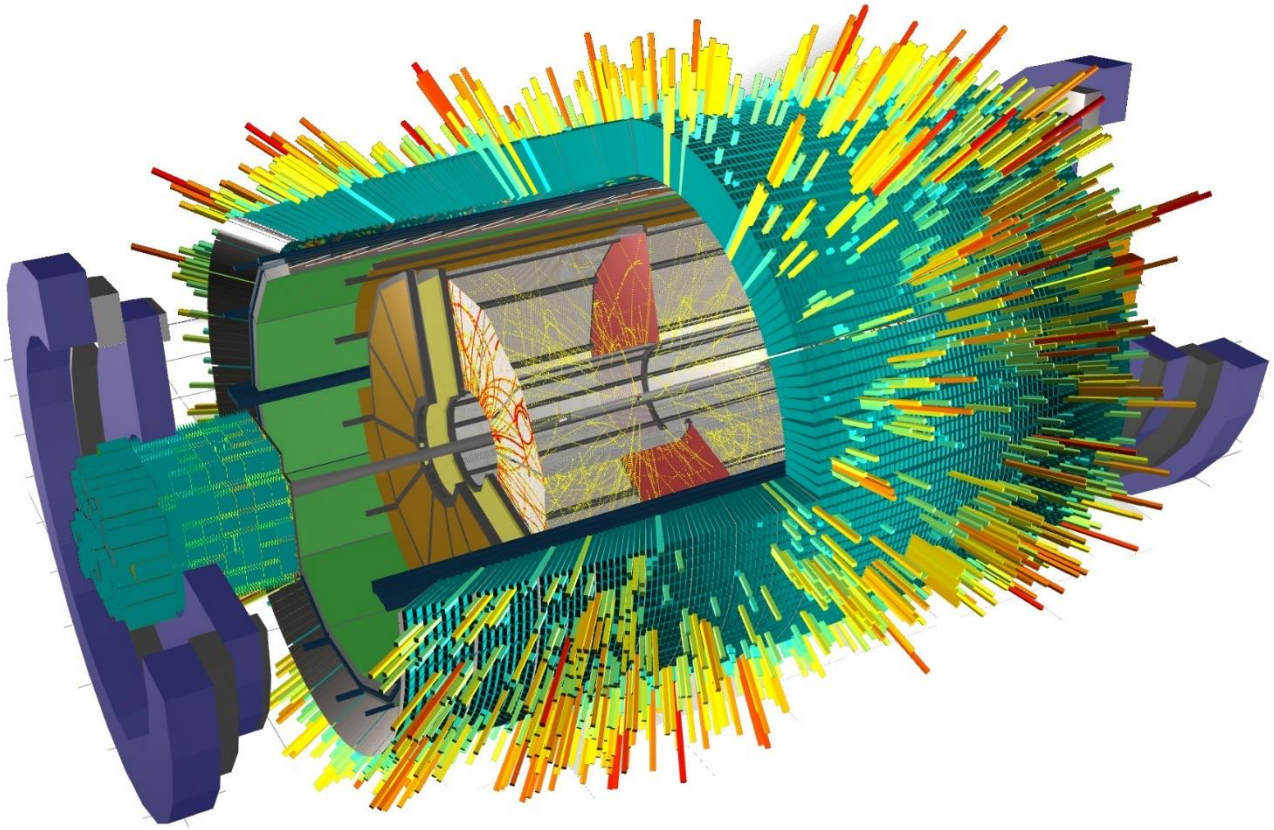
- up to date with latest packages (Acts, ROOT, …)
- regular release schedule
- automated tests
- cleanup
- refactoring

DEVELOPMENT ROADMAP



Thank You !

Q & A



MPD Software Development & Computing Team

Rogachevsky O.	Coordinator
Krylov V.	Clustering, Microservices
Krylov A.	MPD Event Display
Bychkov A.	Detector Simulation
Kuzmin V.	Detector Alignment
Moshkin A., Pelevanyuk I.	Distributed Computing
Alexandrov E., Alexandrov I.	Databases
Papoyan V.	PID ML
Busa J.	Build System
Hnatic S.	Architecture, Tracking

Acknowledgements to MLIT JINR, VBLHEP JINR