Refactoring of the PndPidCorrelator

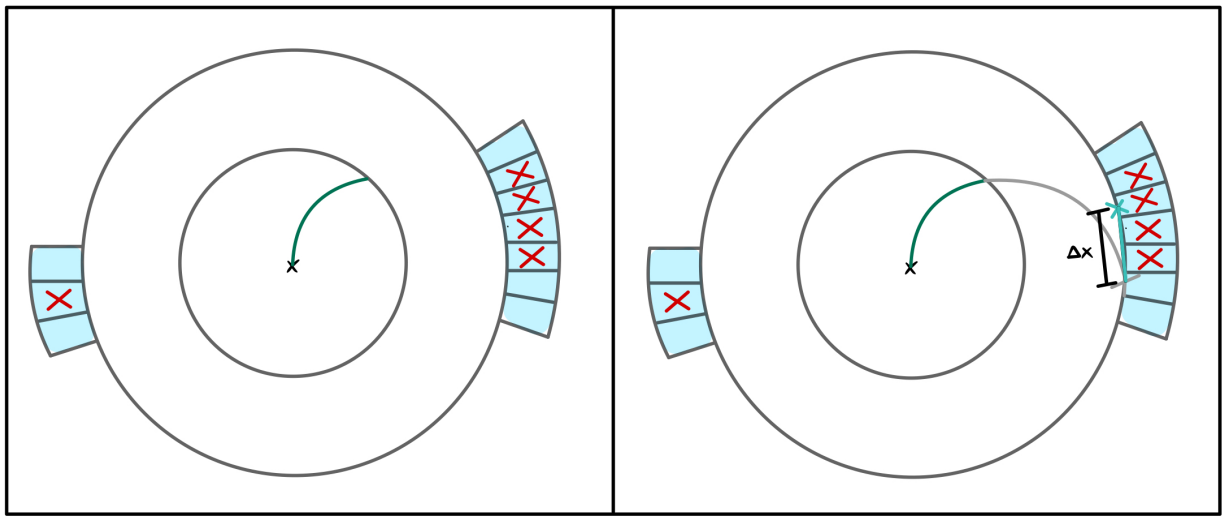
Sarah Gaiser

March 9, 2021

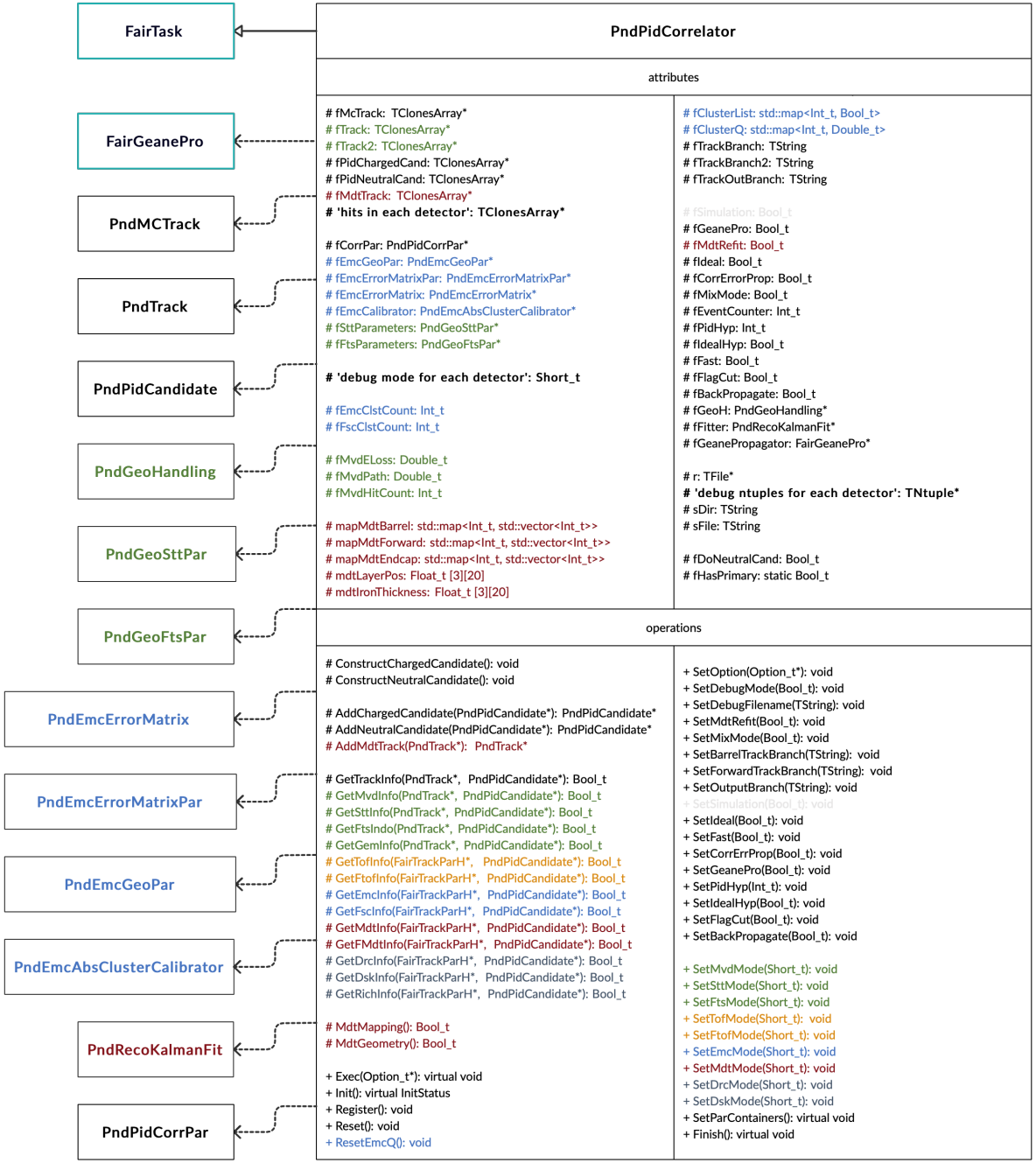
UNIVERSITÄT BONN

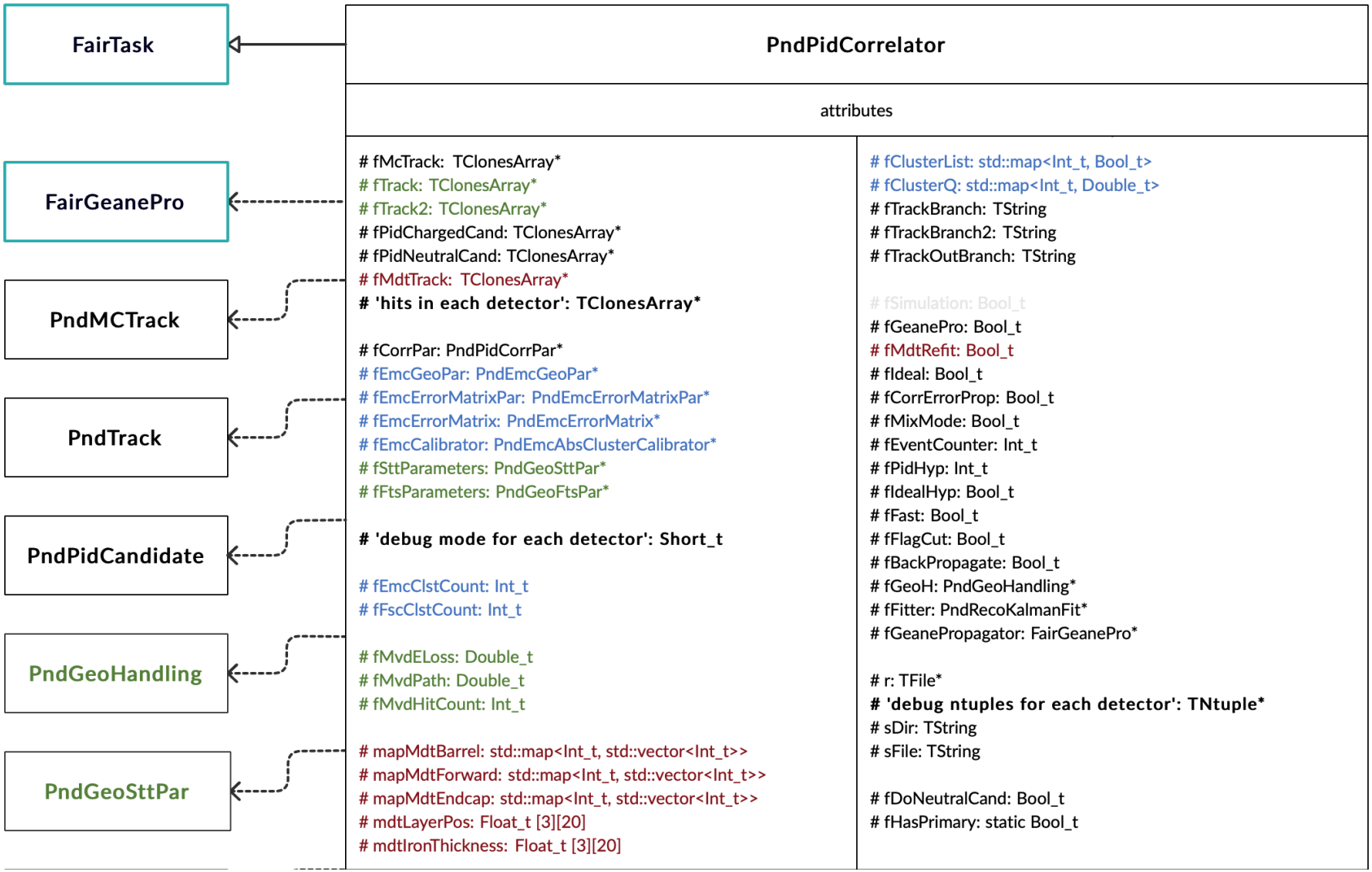
What is the PndPidCorrelator?

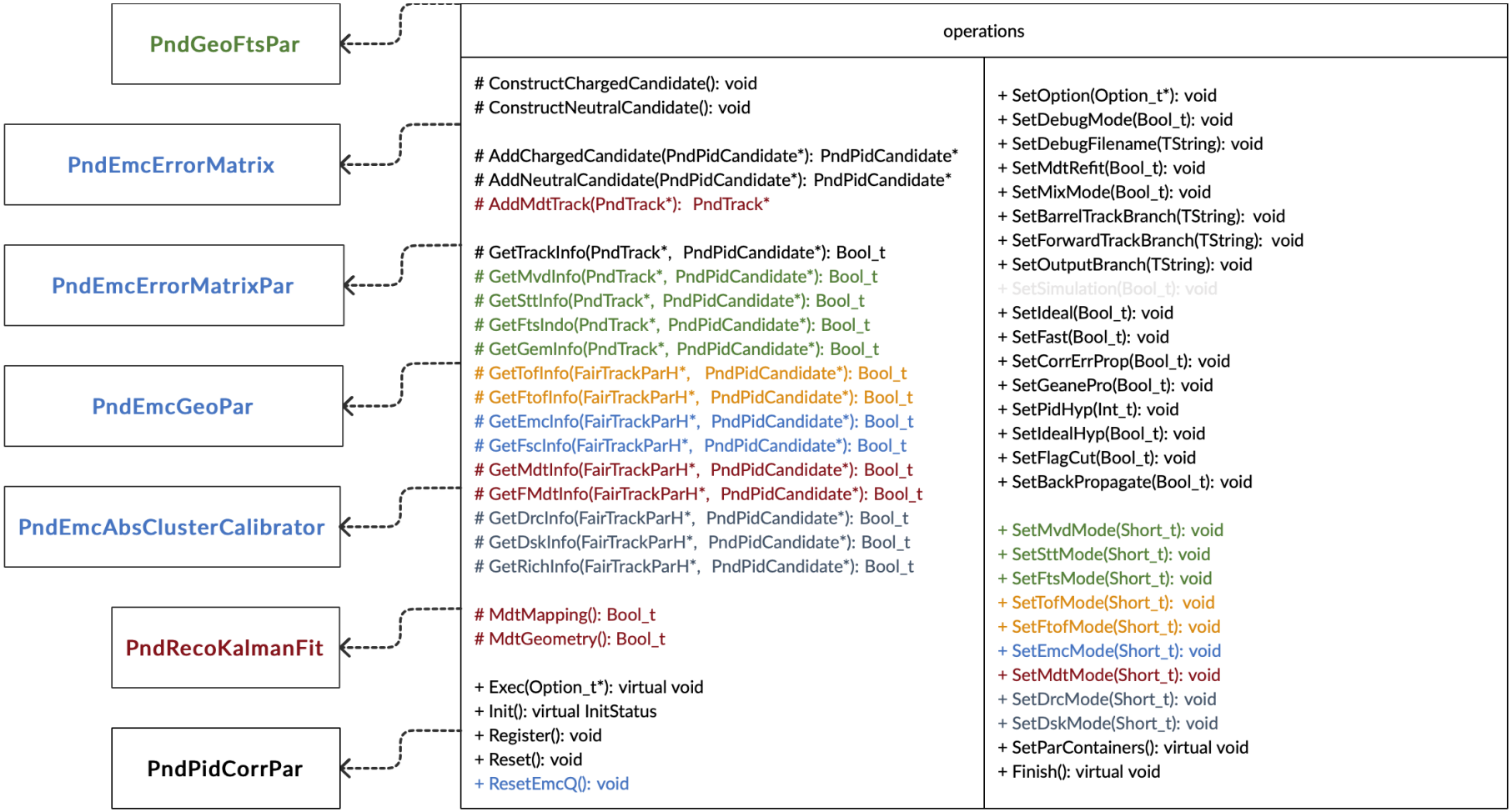
- Correlate tracks and detector hits
  - → Propagate track to detector
  - → Find closest match



- Fill PndPidCandidates with detector information
  - → Reconstructed tracking information
  - → Detector pid information, e.g. $\frac{dE}{dx}$, EMC cluster size, ...


FairTask
PndPidCorrelator
attributes
FairGeanePro
PndMCTrack
PndTrack
PndPidCandidate
PndGeoHandling
PndGeoSttPar
PndGeoFtsPar
PndEmcErrorMatrix
PndEmcErrorMatrixPar
PndEmcGeoPar
PndEmcAbsClusterCalibrator
PndRecoKalmanFit
PndPidCorrPar
fMcTrack: TClonesArray*
fTrack: TClonesArray*
fTrack2: TClonesArray*
fPidChargedCand: TClonesArray*
fPidNeutralCand: TClonesArray*
fMdtTrack: TClonesArray*
'hits in each detector': TClonesArray*
fCorrPar: PndPidCorrPar*
fEmcGeoPar: PndEmcGeoPar*
fEmcErrorMatrixPar: PndEmcErrorMatrixPar*
fEmcErrorMatrix: PndEmcErrorMatrix*
fEmcCalibrator: PndEmcAbsClusterCalibrator*
fSttParameters: PndGeoSttPar*
fFtsParameters: PndGeoFtsPar*
'debug mode for each detector': Short_t
fEmcClstCount: Int_t
fFscClstCount: Int_t
fMvdELoss: Double_t
fMvdPath: Double_t
fMvdHitCount: Int_t
mapMdtBarrel: std::map<Int_t, std::vector<Int_t>>
mapMdtForward: std::map<Int_t, std::vector<Int_t>>
mapMdtEndcap: std::map<Int_t, std::vector<Int_t>>
mdtLayerPos: Float_t [3][20]
mdtIronThickness: Float_t [3][20]
fClusterList: std::map<Int_t, Bool_t>
fClusterQ: std::map<Int_t, Double_t>
fTrackBranch: TString
fTrackBranch2: TString
fTrackOutBranch: TString
fSimulation: Bool_t
fGeanePro: Bool_t
fMdtRefit: Bool_t
fIdeal: Bool_t
fCorrErrorProp: Bool_t
fMixMode: Bool_t
fEventCounter: Int_t
fPidHyp: Int_t
fIdealHyp: Bool_t
fFast: Bool_t
fFlagCut: Bool_t
fBackPropagate: Bool_t
fGeoH: PndGeoHandling*
fFitter: PndRecoKalmanFit*
fGeanePropagator: FairGeanePro*
r: TFile*
'debug ntuples for each detector': TNtuple*
sDir: TString
sFile: TString
fDoNeutralCand: Bool_t
fHasPrimary: static Bool_t
operations
ConstructChargedCandidate(): void
ConstructNeutralCandidate(): void
AddChargedCandidate(PndPidCandidate*): PndPidCandidate*
AddNeutralCandidate(PndPidCandidate*): PndPidCandidate*
AddMdtTrack(PndTrack*): PndTrack*
GetTrackInfo(PndTrack*, PndPidCandidate*): Bool_t
GetMvdInfo(PndTrack*, PndPidCandidate*): Bool_t
GetSttInfo(PndTrack*, PndPidCandidate*): Bool_t
GetFtsIndo(PndTrack*, PndPidCandidate*): Bool_t
GetGemInfo(PndTrack*, PndPidCandidate*): Bool_t
GetTofInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetFtofInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetEmcInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetFscInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetMdtInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetFMdtInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetDrcInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetDskInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetRichInfo(FairTrackParH*, PndPidCandidate*): Bool_t
MdtMapping(): Bool_t
MdtGeometry(): Bool_t
+ Exec(Option_t*): virtual void
+ Init(): virtual InitStatus
+ Register(): void
+ Reset(): void
+ ResetEmcQ(): void
+ SetOption(Option_t*): void
+ SetDebugMode(Bool_t): void
+ SetDebugFilename(TString): void
+ SetMdtRefit(Bool_t): void
+ SetMixMode(Bool_t): void
+ SetBarrelTrackBranch(TString): void
+ SetForwardTrackBranch(TString): void
+ SetOutputBranch(TString): void
+ SetSimulation(Bool_t): void
+ SetIdeal(Bool_t): void
+ SetFast(Bool_t): void
+ SetCorrErrProp(Bool_t): void
+ SetGeanePro(Bool_t): void
+ SetPidHyp(Int_t): void
+ SetIdealHyp(Bool_t): void
+ SetFlagCut(Bool_t): void
+ SetBackPropagate(Bool_t): void
+ SetMvdMode(Short_t): void
+ SetSttMode(Short_t): void
+ SetFtsMode(Short_t): void
+ SetTofMode(Short_t): void
+ SetFtofMode(Short_t): void
+ SetEmcMode(Short_t): void
+ SetMdtMode(Short_t): void
+ SetDrcMode(Short_t): void
+ SetDskMode(Short_t): void
+ SetParContainers(): virtual void
+ Finish(): virtual void


FairTask
FairGeanePro
PndMCTrack
PndTrack
PndPidCandidate
PndGeoHandling
PndGeoSttPar
PndPidCorrelator
attributes
fMcTrack: TClonesArray*
fTrack: TClonesArray*
fTrack2: TClonesArray*
fPidChargedCand: TClonesArray*
fPidNeutralCand: TClonesArray*
fMdtTrack: TClonesArray*
'hits in each detector': TClonesArray*
fCorrPar: PndPidCorrPar*
fEmcGeoPar: PndEmcGeoPar*
fEmcErrorMatrixPar: PndEmcErrorMatrixPar*
fEmcErrorMatrix: PndEmcErrorMatrix*
fEmcCalibrator: PndEmcAbsClusterCalibrator*
fSttParameters: PndGeoSttPar*
fFtsParameters: PndGeoFtsPar*
'debug mode for each detector': Short_t
fEmcClstCount: Int_t
fFscClstCount: Int_t
fMvdELoss: Double_t
fMvdPath: Double_t
fMvdHitCount: Int_t
mapMdtBarrel: std::map<Int_t, std::vector<Int_t>>
mapMdtForward: std::map<Int_t, std::vector<Int_t>>
mapMdtEndcap: std::map<Int_t, std::vector<Int_t>>
mdtLayerPos: Float_t [3][20]
mdtIronThickness: Float_t [3][20]
fClusterList: std::map<Int_t, Bool_t>
fClusterQ: std::map<Int_t, Double_t>
fTrackBranch: TString
fTrackBranch2: TString
fTrackOutBranch: TString
fSimulation: Bool_t
fGeanePro: Bool_t
fMdtRefit: Bool_t
fIdeal: Bool_t
fCorrErrorProp: Bool_t
fMixMode: Bool_t
fEventCounter: Int_t
fPidHyp: Int_t
fIdealHyp: Bool_t
fFast: Bool_t
fFlagCut: Bool_t
fBackPropagate: Bool_t
fGeoH: PndGeoHandling*
fFitter: PndRecoKalmanFit*
fGeanePropagator: FairGeanePro*
r: TFile*
'debug ntuples for each detector': TNtuple*
sDir: TString
sFile: TString
fDoNeutralCand: Bool_t
fHasPrimary: static Bool_t


PndGeoFtsPar
PndEmcErrorMatrix
PndEmcErrorMatrixPar
PndEmcGeoPar
PndEmcAbsClusterCalibrator
PndRecoKalmanFit
PndPidCorrPar
operations
ConstructChargedCandidate(): void
ConstructNeutralCandidate(): void
AddChargedCandidate(PndPidCandidate*): PndPidCandidate*
AddNeutralCandidate(PndPidCandidate*): PndPidCandidate*
AddMdtTrack(PndTrack*): PndTrack*
GetTrackInfo(PndTrack*, PndPidCandidate*): Bool_t
GetMvdInfo(PndTrack*, PndPidCandidate*): Bool_t
GetSttInfo(PndTrack*, PndPidCandidate*): Bool_t
GetFtsIndo(PndTrack*, PndPidCandidate*): Bool_t
GetGemInfo(PndTrack*, PndPidCandidate*): Bool_t
GetTofInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetFtofInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetEmcInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetFscInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetMdtInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetFMdtInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetDrcInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetDskInfo(FairTrackParH*, PndPidCandidate*): Bool_t
GetRichInfo(FairTrackParH*, PndPidCandidate*): Bool_t
MdtMapping(): Bool_t
MdtGeometry(): Bool_t
+ Exec(Option_t*): virtual void
+ Init(): virtual InitStatus
+ Register(): void
+ Reset(): void
+ ResetEmcQ(): void
+ SetOption(Option_t*): void
+ SetDebugMode(Bool_t): void
+ SetDebugFilename(TString): void
+ SetMdtRefit(Bool_t): void
+ SetMixMode(Bool_t): void
+ SetBarrelTrackBranch(TString): void
+ SetForwardTrackBranch(TString): void
+ SetOutputBranch(TString): void
+ SetSimulation(Bool_t): void
+ SetIdeal(Bool_t): void
+ SetFast(Bool_t): void
+ SetCorrErrProp(Bool_t): void
+ SetGeanePro(Bool_t): void
+ SetPidHyp(Int_t): void
+ SetIdealHyp(Bool_t): void
+ SetFlagCut(Bool_t): void
+ SetBackPropagate(Bool_t): void
+ SetMvdMode(Short_t): void
+ SetSttMode(Short_t): void
+ SetFtsMode(Short_t): void
+ SetTofMode(Short_t): void
+ SetFtofMode(Short_t): void
+ SetEmcMode(Short_t): void
+ SetMdtMode(Short_t): void
+ SetDrcMode(Short_t): void
+ SetDskMode(Short_t): void
+ SetParContainers(): virtual void
+ Finish(): virtual void

Problems

- Not modular
- Doubling of code
- Inefficient and slow
- Smaller bugs and errors

Problems

- Not modular
  - → Very complicated to adjust to new detectors (or changes in code)
  - → Mistake in any detector can easily break the pid correlation
- Doubling of code
- Inefficient and slow
- Smaller bugs and errors

Problems

- Not modular
  - → Very complicated to adjust to new detectors (or changes in code)
  - → Mistake in any detector can easily break the pid correlation
- Doubling of code
  - → Unnecessary increase of lines of code
  - → Complicated to debug
  - → Hard to distinguish if code just looks the same or does the same
- Inefficient and slow
- Smaller bugs and errors

Problems

- Not modular
  - → Very complicated to adjust to new detectors (or changes in code)
  - → Mistake in any detector can easily break the pid correlation
- Doubling of code
  - → Unnecessary increase of lines of code
  - → Complicated to debug
  - → Hard to distinguish if code just looks the same or does the same
- Inefficient and slow
  - → `FairGeanePro` used for propagation ⇒ slows down correlation
  - → For creation of neutral candidates matching of tracks and hits is repeated.
- Smaller bugs and errors

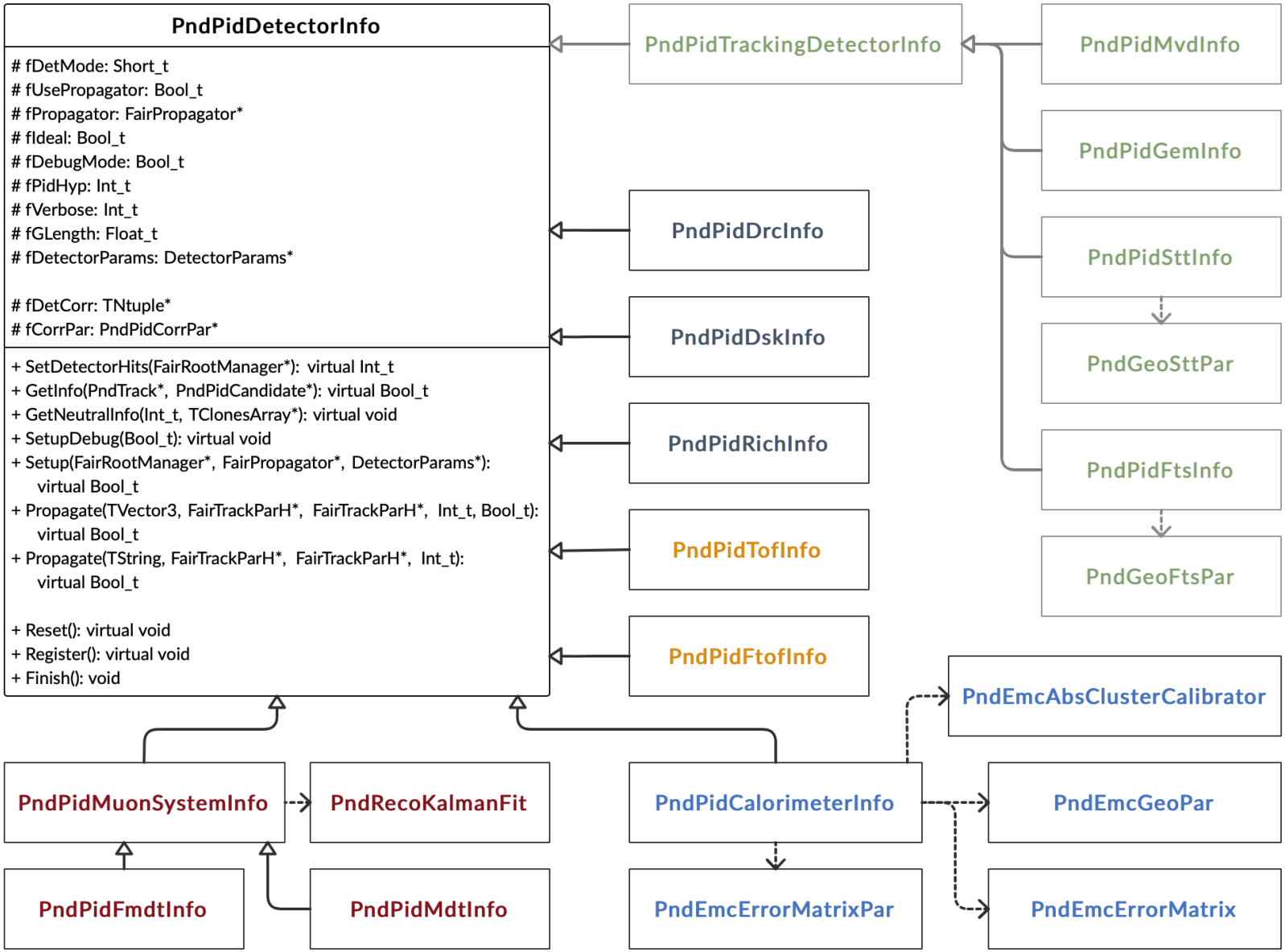
Problems

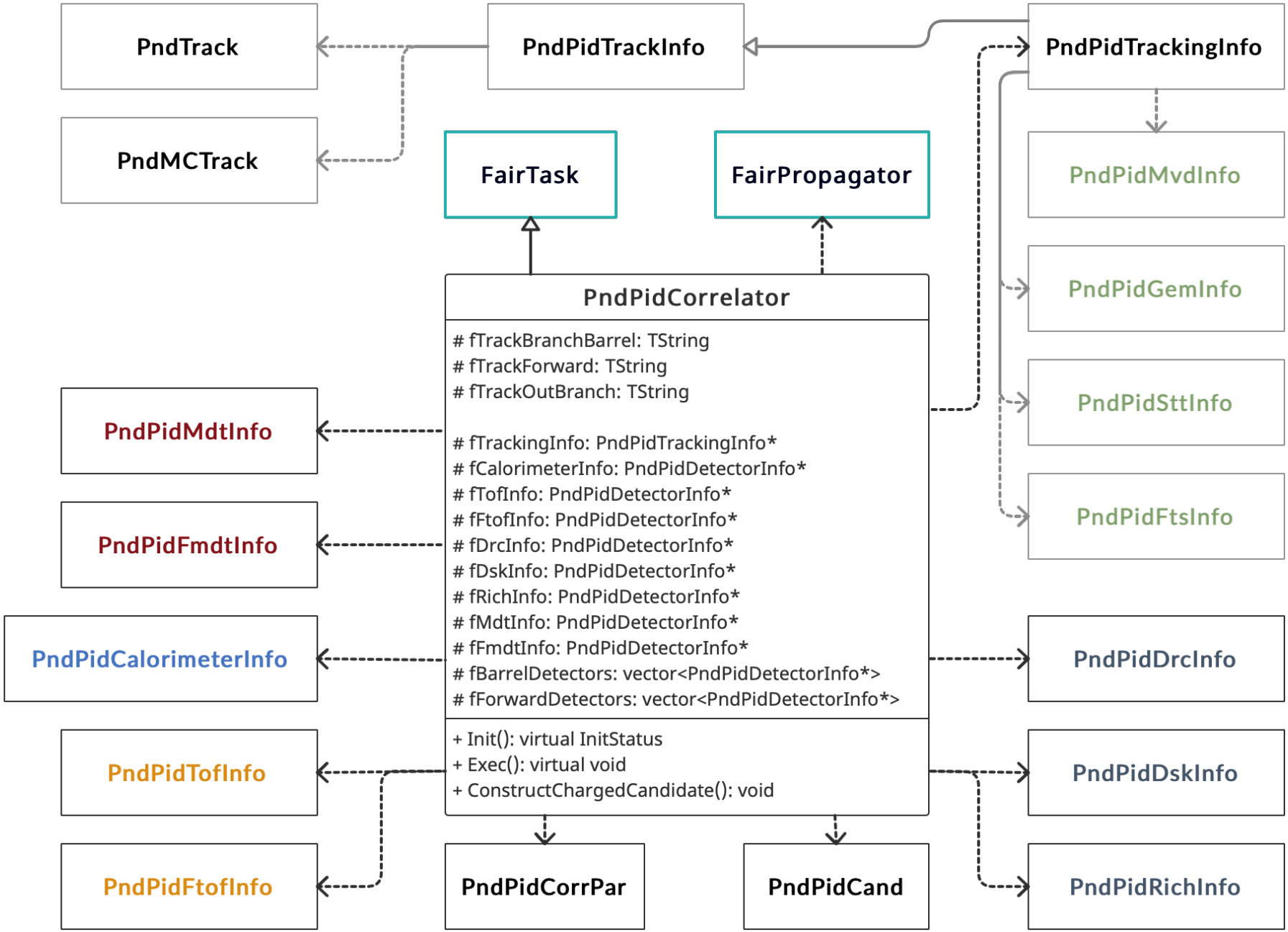
- Not modular
  - → Very complicated to adjust to new detectors (or changes in code)
  - → Mistake in any detector can easily break the pid correlation
- Doubling of code
  - → Unnecessary increase of lines of code
  - → Complicated to debug
  - → Hard to distinguish if code just looks the same or does the same
- Inefficient and slow
  - → `FairGeanePro` used for propagation ⇒ slows down correlation
  - → For creation of neutral candidates matching of tracks and hits is repeated.
- Smaller bugs and errors
  - → `fGeanePro = kFALSE` ⇒ PndPidCorrelator throws fatal error
  - → `fPidHyp` set at wrong point in code
  - → Charged candidates created only if barrel track present ⇒ fixed

How to refactor existing code

- Before start of refactoring
  - QA tests
    - → Create complete set of QA tests which probe overall behaviour of system
    - → Test if mean, standard deviation and number of entries in histograms of PndPidCandidate stays approx. the same
    - → Helps to familiarize with code, get overview
  - Clearify goals of refactoring
    - → Modular structure
    - → Follow SOLID principles of OOD
    - → Create better overview (structure, comments, ...)
- Throughout refactoring
  - → QA tests indicate if anything is broken
  - → Check and modify initial goals if necessary

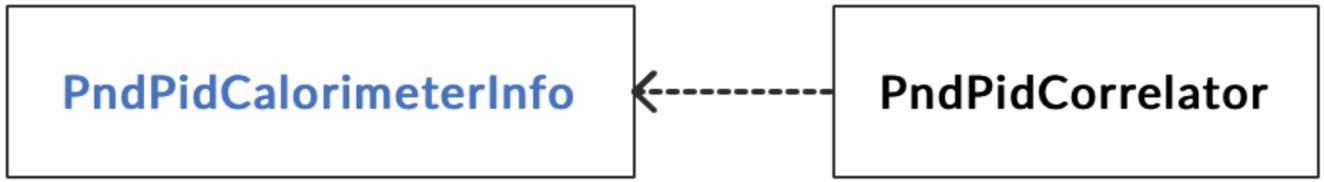
First step - PndPidDetectorInfo class




PndTrack
PndPidTrackInfo
PndPidTrackingInfo
PndMCTrack
FairTask
FairPropagator
PndPidMvdInfo
PndPidCorrelator
PndPidGemInfo
fTrackBranchBarrel: TString
fTrackForward: TString
fTrackOutBranch: TString
PndPidSttInfo
PndPidMdtInfo
fTrackingInfo: PndPidTrackingInfo*
fCalorimeterInfo: PndPidDetectorInfo*
fTofInfo: PndPidDetectorInfo*
PndPidFtsInfo
PndPidFmdtInfo
fFtofInfo: PndPidDetectorInfo*
fDrcInfo: PndPidDetectorInfo*
fDskInfo: PndPidDetectorInfo*
fRichInfo: PndPidDetectorInfo*
fMdtInfo: PndPidDetectorInfo*
PndPidCalorimeterInfo
fFmdtInfo: PndPidDetectorInfo*
PndPidDrcInfo
fBarrelDetectors: vector<PndPidDetectorInfo*>
fForwardDetectors: vector<PndPidDetectorInfo*>
PndPidTofInfo
+ Init(): virtual InitStatus
+ Exec(): virtual void
PndPidDskInfo
+ ConstructChargedCandidate(): void
PndPidFtofInfo
PndPidCorrPar
PndPidCand
PndPidRichInfo

New structure of PndPidCorrelator

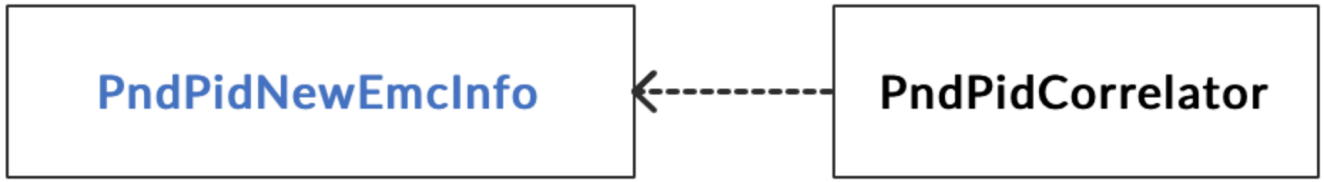
- Modular structure
  - → DetectorInfo classes easily replaceable
  - → PndPidCorrelator: organize detectors and pass general settings
  - → Less dependencies on detector specific code in PndPidCorrelator



- Documentation
  - → Comments and descriptions in code
  - → QA and unit test: illustrate use of PndPidCorrelator

New structure of PndPidCorrelator

- Modular structure
  - → DetectorInfo classes easily replaceable
  - → PndPidCorrelator: organize detectors and pass general settings
  - → Less dependencies on detector specific code in PndPidCorrelator



- Documentation
  - → Comments and descriptions in code
  - → QA and unit test: illustrate use of PndPidCorrelator

Improvements

	old version	refactored version	why important?
number of classes	1	18	single responsibility principle
easily extendable?	no	yes	open-closed principle
dependence on detector implementations	strong	only in DetectorInfo sub-classes	dependency-inversion principle
number of lines	1000	410	better overview
runtime	normal: 87 s ideal: 37 s	normal: 81 s ideal: 30 s	decrease in runtime

Testing

- New version passes QA tests
  - → Some expected deviations (bugs in old version)
- Surprising benefit: PidCorrelator QA tests are sensitive to changes in other components of PandaRoot.
  - → Missing MC truth propagation in EMC ⇒ fixed
- QA tests are functional (black box) tests: run program, check return
- What about structural (white box) tests?
  - → Unit tests: test single unit of code, e.g. if a method is working as intended
  - → For details see `https://panda-wiki.gsi.de/foswiki/bin/view/Computing/MinutesComputingMeeting17-02-2021`

Summary

- PndPidCorrelator is now
  - → Well structured and modular
  - → Open to extensions and changes in detectors
  - → Open to changes in propagation method
  - → Well documented
- Outlook:
  - → New version of PndPidCorrelator on dev branch
  - → Possibilty to improve efficiency and speed by implementing new propagation method.
  - → Next step: analyze quality of track-hit matching, include helix propagation from end of track to EMC

Performance - Runtime

particle	normal			ideal		
	old	new	diff	old	new	diff
electron	118 s	104 s	-14 s	45 s	33 s	-12 s
kaon	110 s	103 s	-7 s	45 s	37 s	-8 s
muon	93 s	88 s	-5 s	47 s	42 s	-5 s
photon	13 s	13 s	-0 s	9 s	7 s	-2 s
pion	106 s	98 s	-8 s	41 s	33 s	-8 s
proton	84 s	80 s	-4 s	35 s	30 s	-5 s

Table 1: Runtime of PndPidCorrelator for 1000 events of given particle type

$\Rightarrow$ Slight decrease in runtime for new version of PndPidCorrelator

Performance - Max memory

	normal			ideal		
particle	old	new	diff	old	new	diff
electron	660 MB	665 MB	+5 MB	671 MB	675 MB	+4 MB
kaon	623 MB	622 MB	-1 MB	633 MB	638 MB	+5 MB
muon	604 MB	611 MB	+7 MB	614 MB	621 MB	+7 MB
photon	627 MB	634 MB	+7 MB	639 MB	646 MB	+7 MB
pion	640 MB	646 MB	+6 MB	651 MB	657 MB	+6 MB
proton	608 MB	615 MB	+7 MB	619 MB	626 MB	+7 MB

Table 2: Max memory of PndPidCorrelator for 1000 events of given particle type

$\Rightarrow$ Very slight increase in memory for new version of PndPidCorrelator