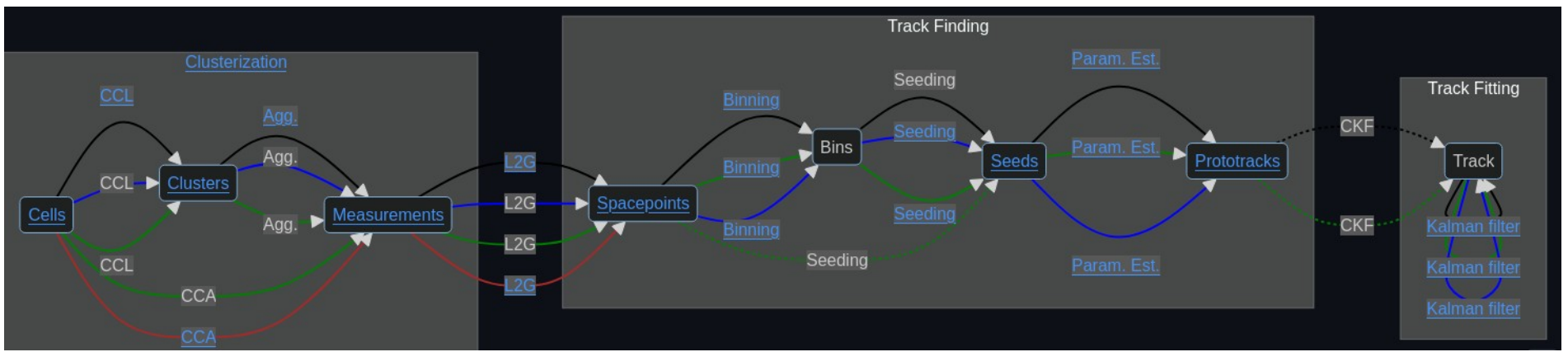
Acts Traccc Plugin

Frederik Verdoner Barba

Traccc

- Track reconstruction on CPU and GPU
- Detray: tacking geometry description
- Covfie: vector fields



Traccc to Acts EDM

- Cells
- Measurements
- Spacepoints
- Seeds
- Tracks

Core edm conversion functionality can be found in **Plugins/Traccc/**

Traccc Chain Example

ActsExamples/Traccc

- CPU Chain example
- CUDA Chain example

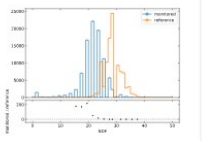
```
chainConfig = acts.examples.TracccChainConfig()

addTracccChain(
    s,
    recoGeometry,
    field,
    digiConfigFile=oddDigiConfig,
    inputCells="cells",
    outputDirRoot=outputDir,
    chainConfig=chainConfig,
    logLevel=acts.logging.INFO,
    platform=args.platform,
    enableAmbiguityResolution=args.ar,
)
```

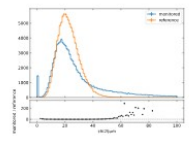
Results (full chain)

ODD, geometric config, 500 events, precision=float

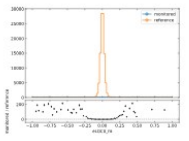
NDF



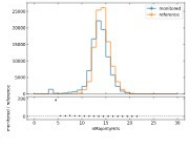
chi2Sum



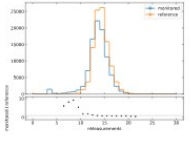
eLOC0_fit



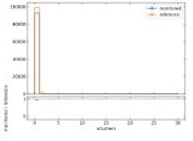
nMajorityHits



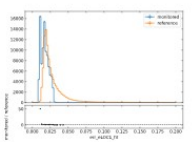
nMeasurements



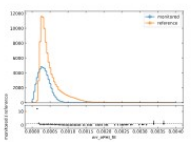
nOutliers



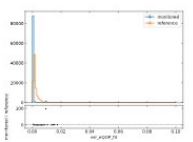
err_eLOC1_fit



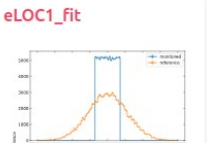
err_ePHI_fit



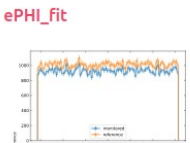
err_eQOP_fit



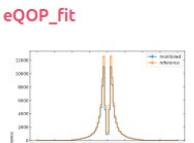
eLOC1_fit



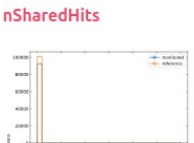
ePHI_fit



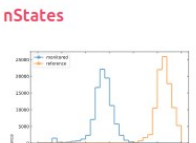
eQOP_fit



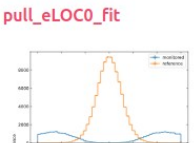
nSharedHits



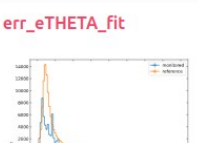
nStates



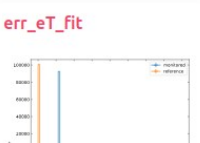
pull_eLOC0_fit



err_eTHETA_fit



err_eT_fit



measurementChi2

eTHETA_fit

eT_fit

err_eLOC0_fit

pull_eLOC1_fit

pull_ePHI_fit

pull_eQOP_fit

measurementLayer

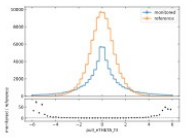
measurementVolume

nHoles

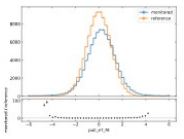
Results (full chain)

Continued

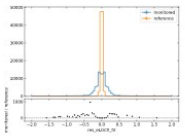
pull_eTHETA_fit



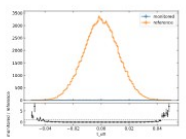
pull_eT_fit



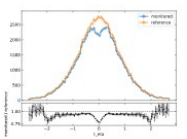
res_eLOC0_fit



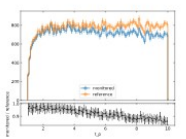
t_d0



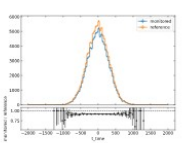
t_eta



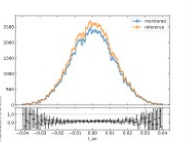
t_p



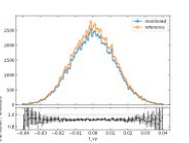
t_time



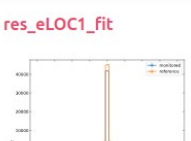
t_vx



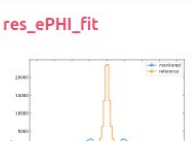
t_vy



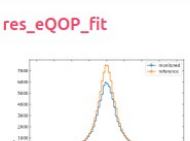
res_eLOC1_fit



res_ePHI_fit

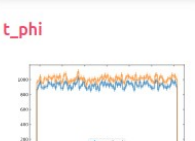


res_eQOP_fit

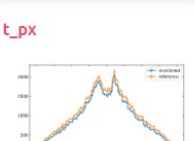


t_pT

t_phi

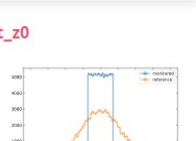


t_px



t_vz

t_z0



trackClassification

res_eTHETA_fit

res_eT_fit

t_charge

t_py

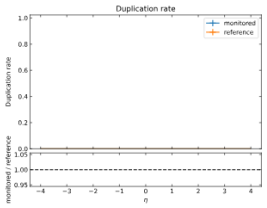
t_pz

t_theta

Results (seeding)

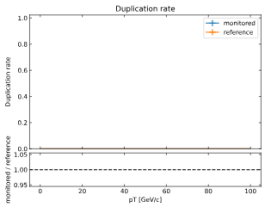
ODD, geometric config, 500 events, precision=float

duplicationRate_vs_eta



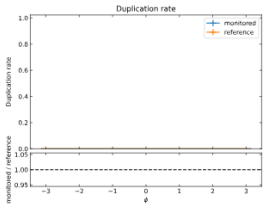
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

duplicationRate_vs_pT



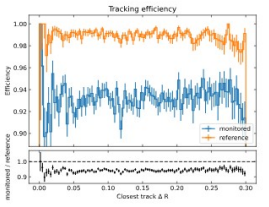
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

duplicationRate_vs_phi



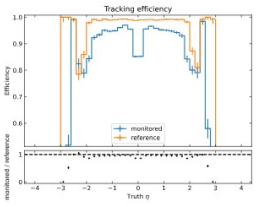
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

trackeff_vs_DeltaR



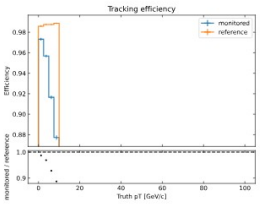
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

trackeff_vs_eta



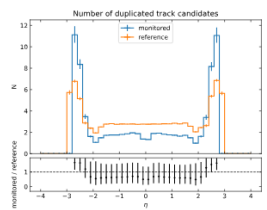
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

trackeff_vs_pT



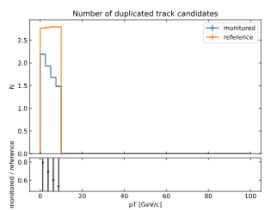
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

nDuplicated_vs_eta



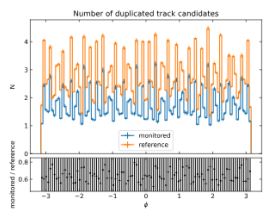
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

nDuplicated_vs_pT



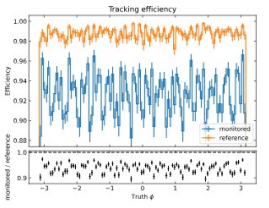
Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

nDuplicated_vs_phi



Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

trackeff_vs_phi

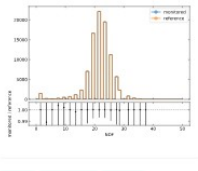


Chi2Test KolmogorovTest RatioCheck ResidualCheck IntegralTest

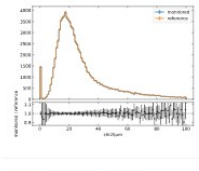
traccc float (orange) vs traccc double (blue)

ODD, geometric config, 500 events

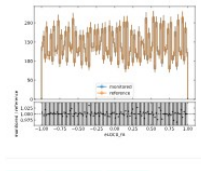
NDF



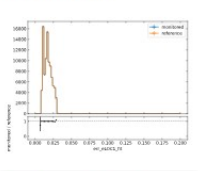
chi2Sum



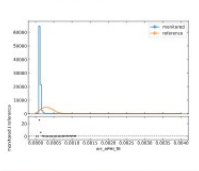
eLOC0_fit



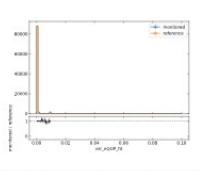
err_eLOC1_fit



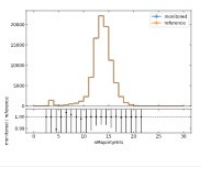
err_ePHI_fit



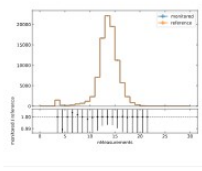
err_eQOP_fit



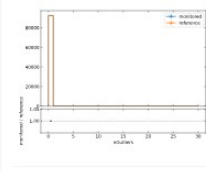
nMajorityHits



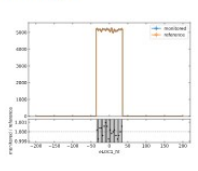
nMeasurements



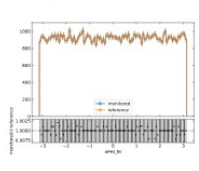
nOutliers



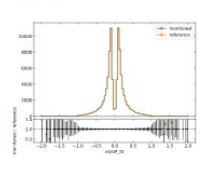
eLOC1_fit



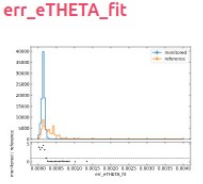
ePHI_fit



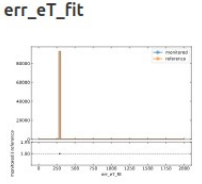
eQOP_fit



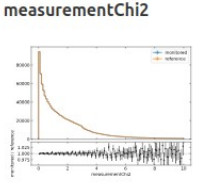
err_eTHETA_fit



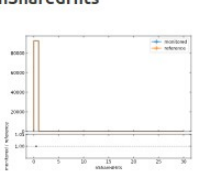
err_eT_fit



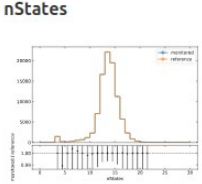
measurementChi2



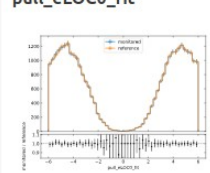
nSharedHits



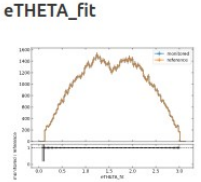
nStates



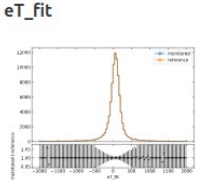
pull_eLOC0_fit



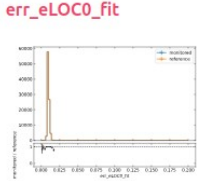
eTHETA_fit



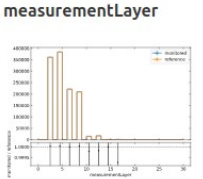
eT_fit



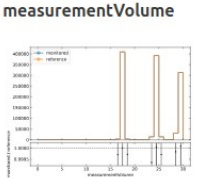
err_eLOC0_fit



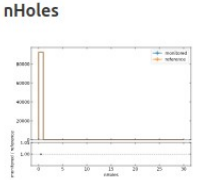
measurementLayer



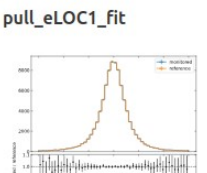
measurementVolume



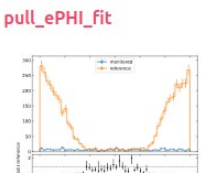
nHoles



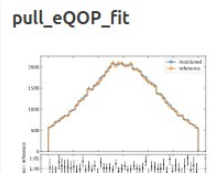
pull_eLOC1_fit



pull_ePHI_fit



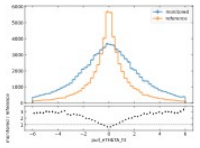
pull_eQOP_fit



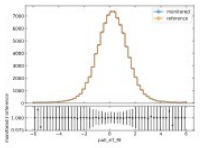
traccc float (orange) vs traccc double (blue)

Continued

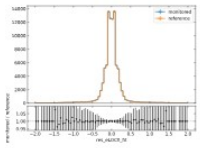
pull_eTHETA_fit



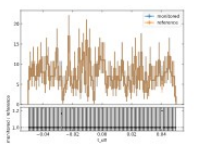
pull_eT_fit



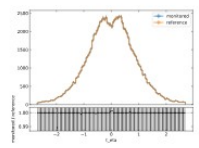
res_eLOC0_fit



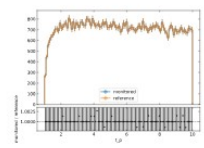
t_d0



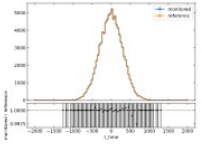
t_eta



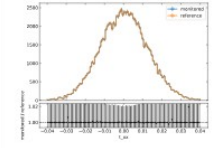
t_p



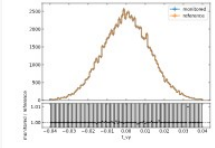
t_time



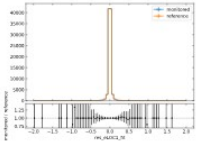
t_vx



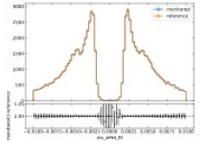
t_vy



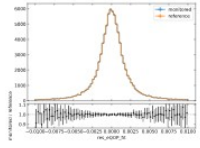
res_eLOC1_fit



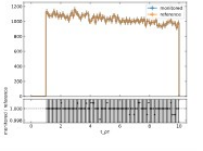
res_ePHI_fit



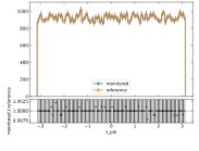
res_eQOP_fit



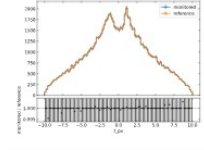
t_pT



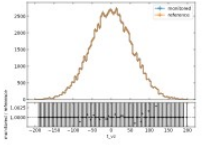
t_phi



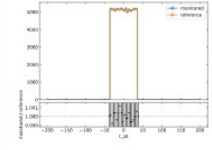
t_px



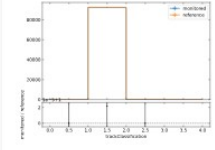
t_vz



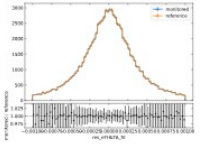
t_z0



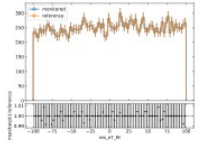
trackClassification



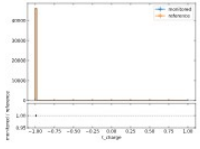
res_eTHETA_fit



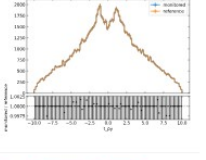
res_eT_fit



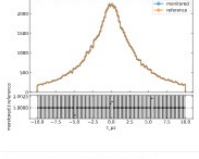
t_charge



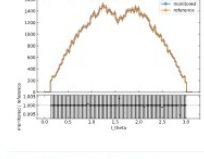
t_py



t_pz



t_theta



Acts Covfie Plugin

Covfie Plugin

- Functions for converting ACTS magnetic fields to covfie fields:

  ConstantMagneticField

  InterpolatedMagneticField

  MagneticFieldProvider

```python
def test_constant_field_conversion():
    from acts import covfie

    v = acts.Vector3(1, 2, 3)
    af = acts.ConstantBField(v)
    cf = covfie.covfieField(af)
    view = covfie.newView(cf)
    points = [(0, 0, 1), (1, 1, 1), (1, 0, 2)]
    for x, y, z in points:
        assert view.at(x, y, z) == [1, 2, 3]
```

```cpp
/// @brief Creates a covfie field from an interpolated magnetic field.
/// @param magneticField The acts interpolated magnetic field.
/// @return An affine linear strided covfie field.
InterpolatedField covfieField(
    const Acts::InterpolatedMagneticField& magneticField);

/// @brief Creates a covfie field from a constant B field.
/// @param magneticField The acts constant magnetic field.
/// @return A constant covfie field.
ConstantField covfieField(const Acts::ConstantBField& magneticField);

/// @brief Creates a covfie field from a magnetic field provider by sampling it.
/// The field must be defined within min (inclusive) and max (inclusive).
/// @param magneticField The acts magnetic field provider.
/// @param cache The acts cache.
/// @param nBins 3D array of containing the number of bins for each axis.
/// @param min (min_x, min_y, min_z)
/// @param max (max_x, max_y, max_z)
/// @return An affine linear strided covfie field.
InterpolatedField covfieField(const Acts::MagneticFieldProvider& magneticField,
                              Acts::MagneticFieldProvider::Cache& cache,
                              const std::vector<std::size_t>& nBins,
                              const std::vector<double>& min,
                              const std::vector<double>& max);
```

Plugins/Covfie/FieldConversion.hpp

Future work

- Test CUDA Chain example
- Obtain detray detector using detray plugin (currently it reads from a file)