Light nuclei analysis wagon for the MpdRoot

V. Kireyeu

PWG-2

17.09.2024

MPD Cross-PWG Meeting

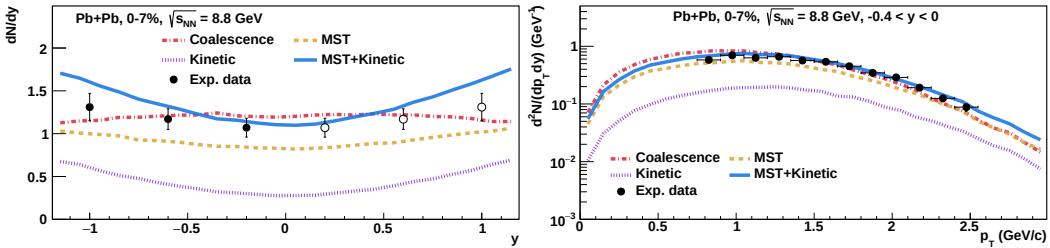
Introduction

The new "wagon" for the light nuclei (d, He) analysis is implemented within the MpdRoot "train"-like analysis chain. The main purpose of the nuclei wagon is to provide:

- The **phase-space distributions** for the particles of interest.
- The **efficiencies and contaminations** corrections for the for the final results.
- The double-differential p_T **spectra**.
- The **rapidity distributions**.
- The **coalescence parameters B_2 and B_3 spectra**.

PHQMD

Request 29: General-purpose, 20M PHQMD BiBi@9.2 was used for this analysis.



More on the PHQMD transport approach:

Aichelin, J. et al. (2020). "Parton-hadron-quantum-molecular dynamics: A novel microscopic *n* -body transport approach for heavy-ion collisions, dynamical cluster formation, and hypernuclei production". In: *Phys. Rev. C* 101.4, p. 044905.

Why it's important to look at high y and low p_T for light nuclei:

Kireyeu, V. et al. (2024). "Cluster formation near midrapidity: How the production mechanisms can be identified experimentally". In: *Phys. Rev. C* 109.4, p. 044906.

Technical details

Wagon structure and logic

Any analysis wagon must be:

- **Highly configurable** to avoid the unnecessary recompiling.
- **Highly automated** for the same reasons.
- **Well documented**.

The **light nuclei** analysis was developed taking into account the requirements above and it consists of two big separate parts:

- The MpdRoot **analysis wagon** (C++) – provides the phase space and efficiencies 2D histograms.
- The **post processing macro** (Python) – makes the final analysis.

Wagon structure and logic

- Initialization: settings are read from the JSON-file, histograms are booked for the each:
  - Particle.
  - Centrality bin.
  - PID type: true Monte Carlo (MC) or reconstructed PID (RC PID).
  - PID mode for the RC PID case:
    - MpdPid class.
    - evPID wagon.
- Event processing:
  - Events are checked for the "event quality" cuts.
  - Tracks are checked for the "track quality" cuts.
  - Centrality bin is selected.
  - Particles are identified (MC, PID).
  - The phase space histograms for the identified particles are filled.
- Helper subroutines for the configuration reading, "quality" checks, centrality and particles selection etc.

Configuration: global, event cuts

Every parameter of the "nuclei" wagon can be set in the JSON-formatted settings file.

```
"Verbose": "1",
"N_MPD_PID_Particles": "8",
"make_MC": "1",
"make_Efficiency": "1",
"PID_mode": "2",
"DCA_mode": "0",
"TOF_mode": "0",
"use_pt_corrections": "0",
"pt_corrections_file": "pt_corrections.root",
"Events": {
    "PrimaryVertexZ": "100",
    "Centrality": [[0, 20], [20, 40], [40, 80]]
},
```

Configuration: track quality, PID

```
"Tracks": {
    "NHits": "20",
    "NSigmaDCAx": "3",
    "NSigmaDCAy": "3",
    "NSigmaDCAz": "3",
    "LowPtCut": "0.05"
},
"PID": {
    "TPCSigma": "2",
    "TOFSigma": "2",
    "TOFDphiSigma": "3",
    "TOFDzSigma": "3"
},
```

```
"MpdPid": {
    "Energy": "9.2",
    "Coef": "1.0",
    "Generator": "PHQMD",
    "Tracking": "CFHM",
    "IniString": "pikaprdetrhe3he4",
    "Add_dedx_only": "1"
},
"evPID": {
    "VetoTPC": "3",
    "VetoToF": "3"
},
```

Configuration: particles of interest

```
"Particles": {
    "p": {
        "PDG": "2212",
        "Mass": "0.938",
        "Charge": "1",
        "Enum": "3",
        "pt_bins": [60, 0.0, 6.0],
        "rapidity_bins": [60, -3.0, 3.0]
    },
```

```
"d": {
    "PDG": "1000010020",
    "Mass": "1.876",
    "Charge": "1",
    "Enum": "4",
    "pt_bins": [30, 0.0, 6.0],
    "rapidity_bins": [60, -3.0, 3.0]
},
```

PID modes

In this wagon, particle can be identified by one of the two PID modes: **"evPID"** and **MpdPid**.

The PID mode can be defined in the configuration file – no need to recompile!

MpdPid combined mode (the standard for the MpdRoot, please, read the MpdPid class documentation):

- `PID -> FillProbs(p*charge, dedx, m2, charge)`
- `PID -> GetMaxProb()`

evPID mode: separate spectra for the particles identified by the dE/dx and m^2 information: GetTPCNSigma() and GetTOFNSigma() ($+ 3\sigma$ vetoing).

DCA modes

DCA cut can be applied by two methods:

- "Classic": $\sqrt{dca_x^2 + dca_y^2 + dca_z^2} < 2$ cm – track accepted.
- "N-Sigma": $|N\sigma_{x,y,z}| < 2$ – track accepted.

ToF Matching modes

ToF Matching can be checked by three methods:

- "Classic": by checking the ToF flags 2 and 6.
- "N-Sigma": by checking the GetTofDphiSigma() and GetTofDzSigma() from the MpdTrack.
- "MpdTofMatchingData": by using the MpdTofMatchingData branch.

Postprocessing highlights

• The post processing macro also uses the settings file used for the nuclei wagon run and works in (almost) fully **automatic mode**: the centrality bins, the particles of interest, the PID mode will be read from the settings files and corresponding subroutines will be utilized automatically.

• The fitting procedure is time consumig, so the **multiprocessing calculations are implemented**: several fits will be applied in parallel to reduce the overall computational time. The blocking serialisation and I/O issue #16184 was found due to this activity – fixed 2 days ago by ROOT devellopers in the "master".

• In the case of the bad fit result that fit procedure will start again with the new initial parameters.

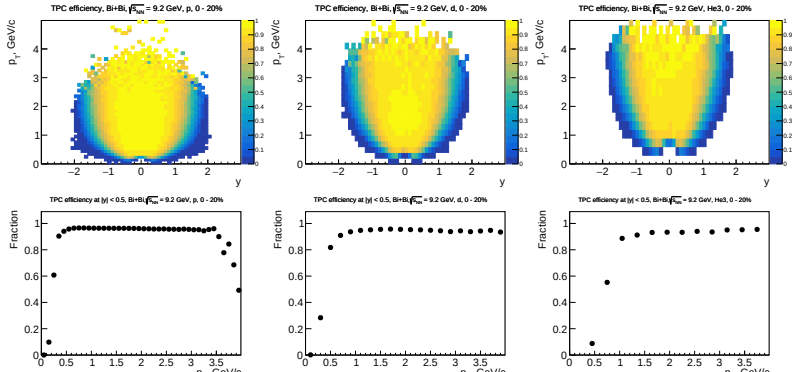
• All results are saved as the plots (pdf or png) and at the same time as the ROOT objects (histograms, fits) in a separate ROOT-file, so they can be used lated within the different drawing macro.

Postprocessing macro usage

```
$ ./postprocess.py --help
usage: postprocess [-h] [-i INPUT] [-e EFFICIENCIES] [-s SETTINGS] [-o OUTPUT] [-d DIR] [-r REPORT] [-p PTCORRECTIONS] [-t TABLE]
                   [--qa] [--png] [--skip] [--clear] [--nodedx]

MpdNuclei wagon data processing program

options:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        Input ROOT file
  -e EFFICIENCIES, --eff EFFICIENCIES
                        Efficiencies ROOT file
  -s SETTINGS, --settings SETTINGS
                        Settings JSON file
  -o OUTPUT, --output OUTPUT
                        Output spectra ROOT-file
  -d DIR, --dir DIR     Output directory for histograms
  -r REPORT, --report REPORT
                        Output report file
  -p PTCORRECTIONS, --ptcorr PTCORRECTIONS
                        Make file for pt corrections
  -t TABLE, --table TABLE
                        Output file with dndy info for fits etc (JSON)
  --qa                  Produce extended QA plots
  --png                 Produce PNG-plots onstead of PDF
  --skip                Skip already triggered functions
  --clear               Clear the output directory before populating
  --nodedx              Do not produce histograms for the case PID = dE/dx only

If the EFFICIENCIES file is not set, then the INPUT will be used instead.
```

Corrections

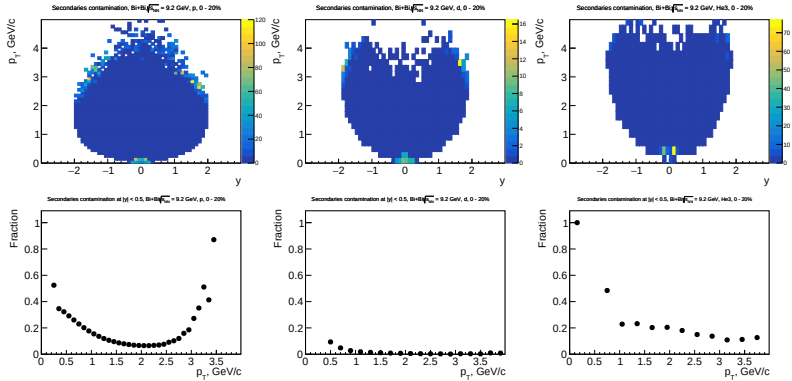
TPC efficiency



$$\varepsilon_{TPC} = \frac{[\text{Primary(MC)} = \text{TRUE}]\ [\text{Identified(MC)}]\ [\text{Track quality}]}{[\text{Primary(MC)} = \text{TRUE}]\ [\text{Identified(MC)}]}$$

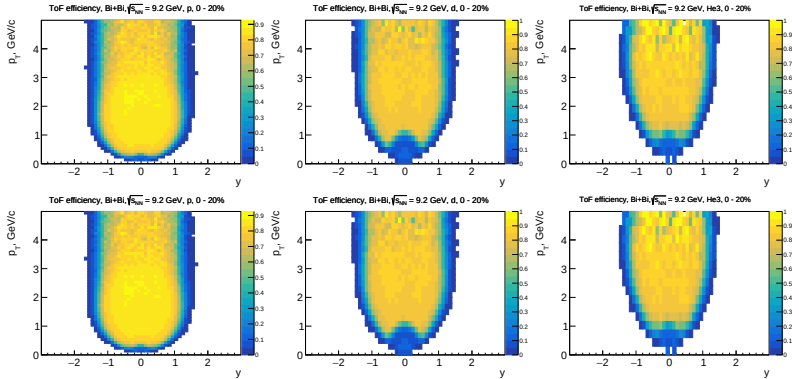
Numerator: primary tracks which satisfy the reconstruction 'Track quality' conditions: $DCA < 3$ cm, $N_{hits} > 20$. Denominator: all tracks. The split track is rejected.

Secondaries contamination



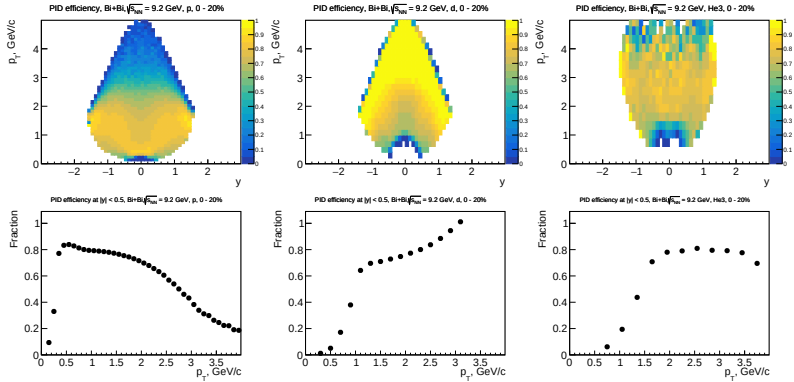
$$C_{sec.} = \frac{[\text{Identified(MC)}]\ [\text{Track quality}]\ [\text{Primary(MC)} = \text{FALSE}]}{[\text{Identified(MC)}]\ [\text{Track quality}]}$$

ToF matching efficiency



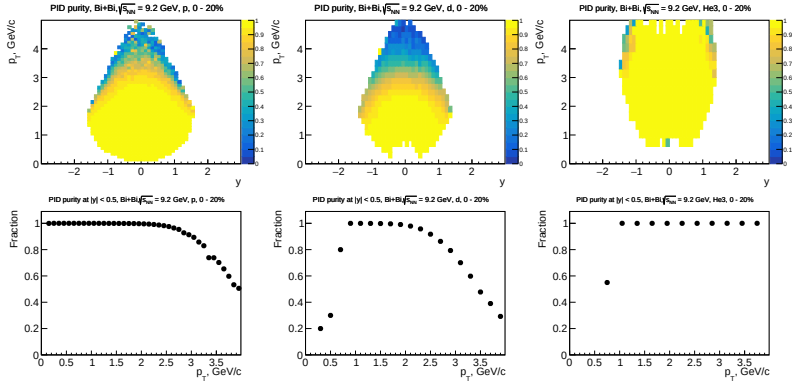
$$\varepsilon_{ToF} = \frac{\text{[Identified(MC)] [Track quality] [ToF flags]}}{\text{[Identified(MC)] [Track quality]}}$$

PID efficiency MpdPid mode



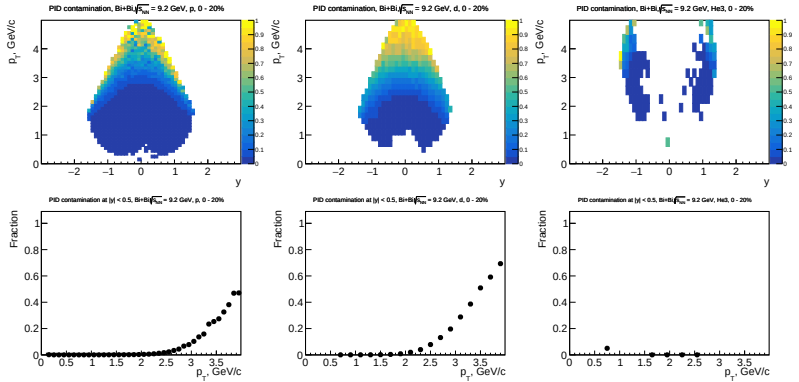
$$\varepsilon_{PID} = \frac{\text{[ToF flags] [Track quality] [Identified(PID)]}}{\text{[ToF flags] [Track quality] [Identified(MC)]}}$$

PID purity MpdPid mode



$$\varepsilon_{PID} = \frac{[\text{ToF flags}]\ [\text{Track quality}]\ [\text{Identified(PID)} = \text{TRUE}]}{[\text{ToF flags}]\ [\text{Track quality}]\ [\text{Identified(PID)}]}$$

PID contamination MpdPid mode

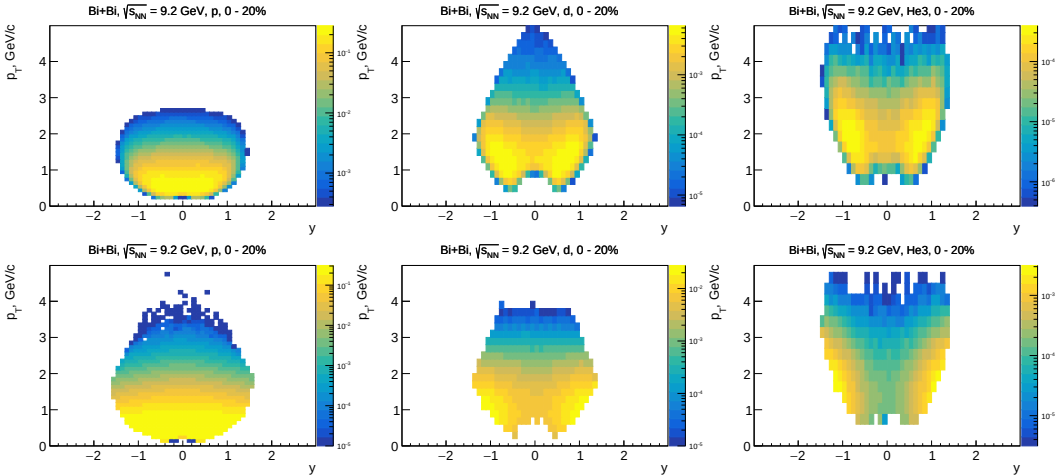


$$C_{PID} = \frac{[\text{ToF flags}]\ [\text{Track quality}]\ [\text{Identified(PID)} = \text{FALSE}]}{[\text{ToF flags}]\ [\text{Track quality}]\ [\text{Identified(PID)}]}$$

Results

Results MpdPid

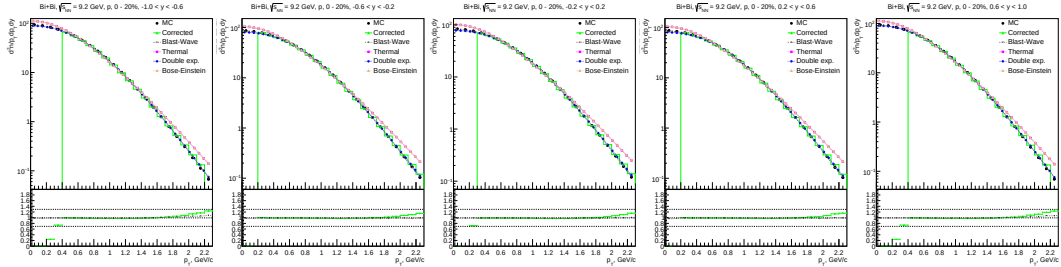
The uncorrected phase-spaces are on the top, corrected – on the bottom.



$$\text{Result} = \text{Uncorrected} \cdot \frac{(1 - C_{sec.}) \cdot (1 - C_{PID})}{\varepsilon_{TPC} \cdot \varepsilon_{ToF} \cdot \varepsilon_{PID}}$$

Transverse momentum spectra: protons, 0-20%, MpdPid

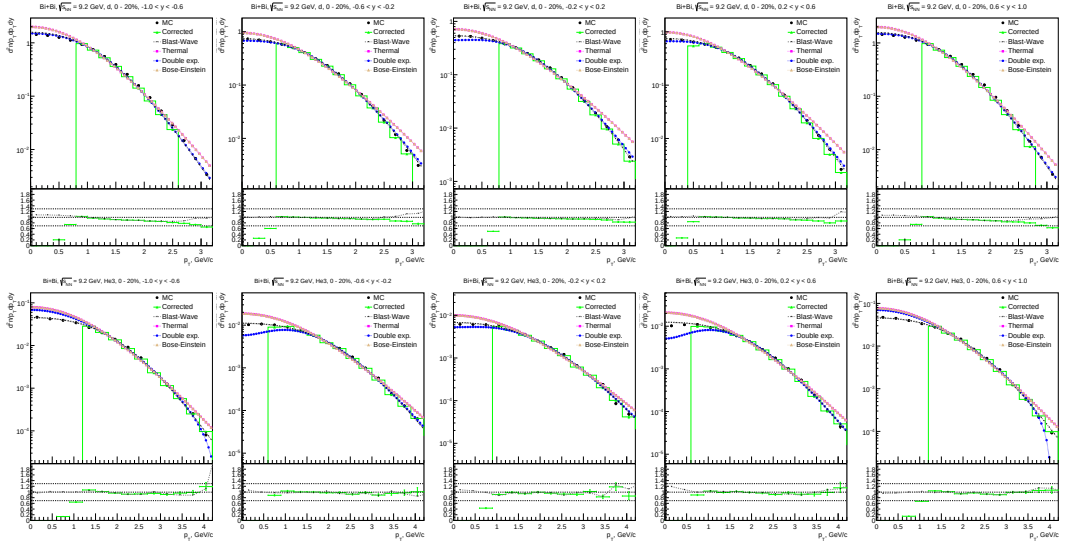
The "raw" output of the post processing macro.



Several fits are applied to the reconstructed points ("thermal", "Blast-Wave", "double-exponential", "Bose-Einstein") to study how well they describe the original Monte Carlo data.

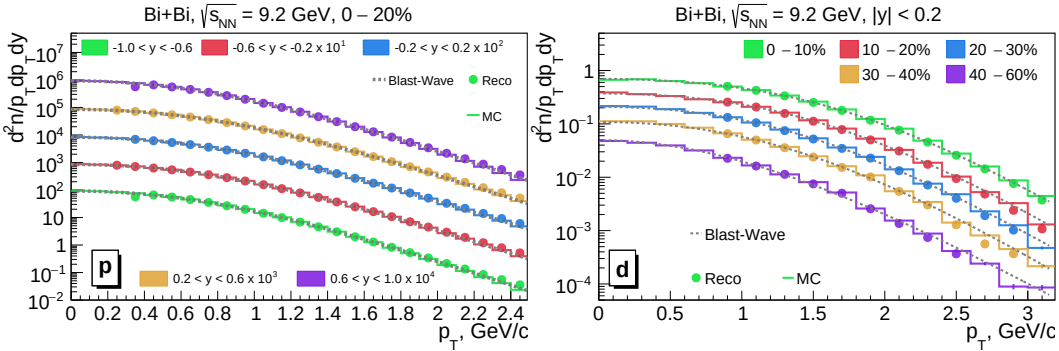
Transverse momentum spectra: d and ^3He, 0-20%, MpdPid

The "raw" output of the post processing macro.

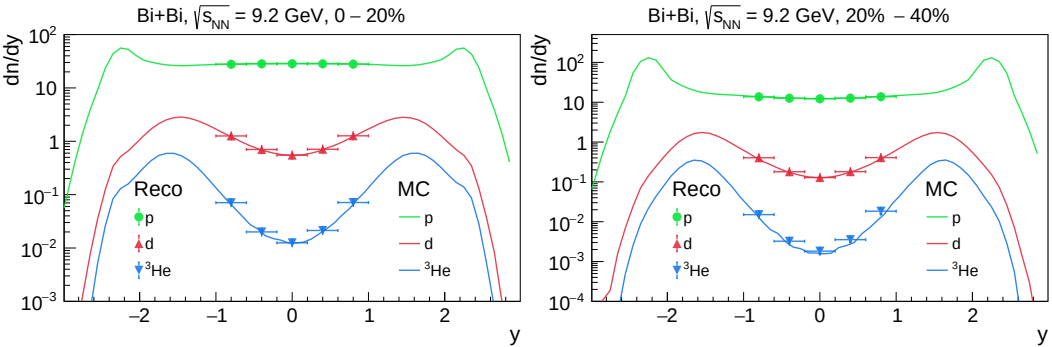


Transverse momentum spectra: protons, deuterons, MpdPid

The separate drawing macro can be applied after the post processing for the "publication-ready" plots.

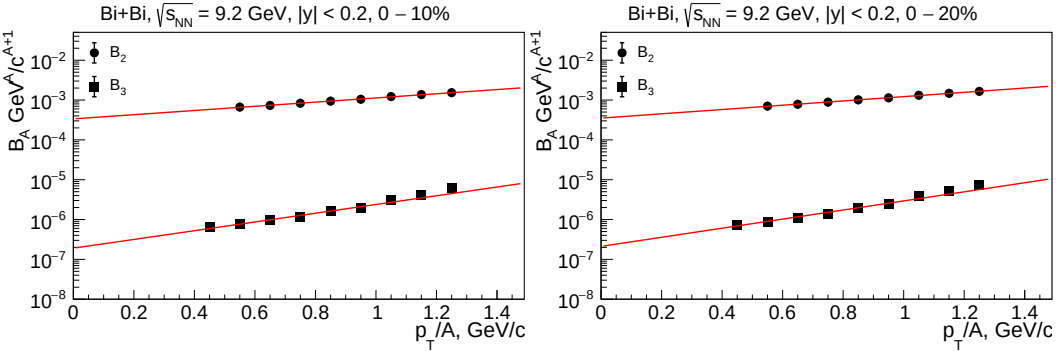


Rapidity spectra: protons, deuterons, ^3He, MpdPid



The rapidity spectra are obtained from the double-differential $d^2N/dp_T dy$ spectra by combining the reconstructed points and the interpolation of the Blast-Wave fits in the low and high p_T regions.

Coalescence parameters B_2 and $B_3(^3He)$, MpdPid



The coalescence parameters B_2 and $B_3(^3He)$ spectra are obtained from the double-differential d^2N/dp_Tdy distributions. Red lines are the exponential fits.

Documentation

Documentation

Documentation is a **fundamental and crucial** aspect of the software development, ensuring **reliability, usability and longevity** of (scientific) software.

- Ensures that the **understanding of the software's functionality**, design and usage is shared among current and future team members.
- **Facilitates onboarding of new members** by providing them with a comprehensive guide to the codebase.
- Clear the gap between different teams making it easier to **understand each other's work and share best practices**.
- **Aids in maintaining and updating** the software by providing a clear roadmap of the code's structure and logic.
- **Reduces the likelihood of errors** by providing guidelines and explanations for the software's intended use and limitations.
- **And MUCH more!**

Documentation

We all know how good the **ROOT Reference Documentation** (`https://root.cern/doc/master/`) is. **Why not to do it for our routines?**

Where to look if I start to write an analysis "wagon"?!

- **What are the common practices?**
- **Which methods should I use?**
- **Branches, classes, common cuts?**
- **etc and so on...**

Documentation

The "light nuclei wagon" is using the specially-formatted comments within the code processed by the documentation generator Doxygen.
Current documentation status:

- A "Readme" file with the common wagon description is provided with the source code.
- The wagon settings are documented.
- All subroutines are documented.
- The output histograms are described.
- The post-processing macro and its subroutines are documented.

Documentation

- MpdNuclei.cxx: 649 lines of code, ∼460 lines of the documentation and comments.
- MpdNuclei.h: 144 lines of code, ∼380 lines of the documentation and comments.

Documentation and comments are ∼105% of the code (∼55% of the total "wagon" lines)!

- postprocess.py: 1558 lines of code, ∼1013 lines of the documentation and comments.

Documentation and comments are ∼65% of the code (∼47% of the total lines in the macro)!

Documentation

```
/** This subroutine performs the event quality checks.
  \param event MpdAnalysisEvent to analyse
  \return
    - true (1) if checks are not passed ("bad" event -- skip)
    - false (0) if checks are passed ("good" event -- analyze)

  The event can be skipped if:
    - The event vertex is not filled
    - The centrality is not defined
    - The event vertex is not reconstructed
    - The event vertex is beyond the limits set by MpdNuclei::s__ev_PrimaryVertexZ
*/
bool MpdNuclei::bad_event(MpdAnalysisEvent &event){
  if(!event.fVertex)                                    return true; // The event vertex is not filled -- skip event
  if(event.getCentrTPC() < 0 || event.getCentrTPC() >= 100) return true; // The centrality is not defined -- skip event
  MpdVertex *vertex = (MpdVertex *)event.fVertex -> First();          // Get the event primary vertex
  vertex -> Position(mPrimaryVertex);                                 // Access its position
  if(mPrimaryVertex.Z() == 0)                           return true; // The event vertex is not reconstructed -- skip event
  if(fabs(mPrimaryVertex.Z()) > s__ev_PrimaryVertexZ)   return true; // The event vertex is beyond the limits -- skip event
  return false;
}
```

Documentation

◆ bad_event()

bool MpdNuclei::bad_event (**MpdAnalysisEvent** & event) private

This subroutine performs the event quality checks.

Parameters

event MpdAnalysisEvent to analyse

Returns

- true (1) if checks are not passed ("bad" event – skip)
- false (0) if checks are passed ("good" event – analyze)

The event can be skipped if:

- The event vertex is not filled
- The centrality is not defined
- The event vertex is not reconstructed
- The event vertex is beyond the limits set by **MpdNuclei::s_ev_PrimaryVertexZ**

Definition at line **452** of file **MpdNuclei.cxx**.

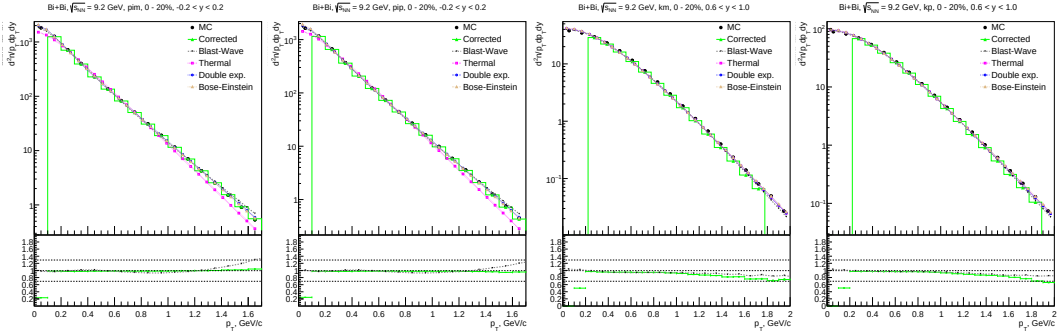
Summary

- **The "Light nuclei" wagon is implemented**.
- The wagon is **highly configurable** and **well documented**.
- Several methods for the PID, DCA and ToF matching are implemented as the parameters in the settings file.
- Several instances of the wagon with different settings can be running within the single analysis 'train'.
- **The post-processing macro is provided**. It can create and draw:
  - The phase-space distributions.
  - The double-differential p_T spectra.
  - The dN/dy rapidity distributions extracted from the p_T spectra with the interpolation to the low anf high p_T regions.
  - The coalesscence parameter B_2 and $B_3(^3He)$spectra.
- The post-processing macro stores all plots in the output ROOT-file and in the directory with the pdf or png files. It can automatically create the single report file (pdf) with all generated plots.

Thank you for your attention!

This presentation was prepared using LaTeX.

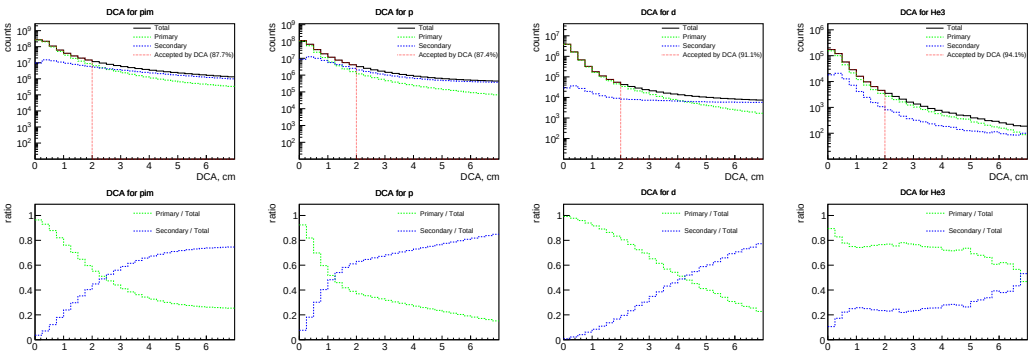
Nuclei, but mesons too



DCA

DCA cut can be applied by two methods:

- "Classic": $\sqrt{dca_x^2 + dca_y^2 + dca_z^2} < 2$ cm – track accepted.

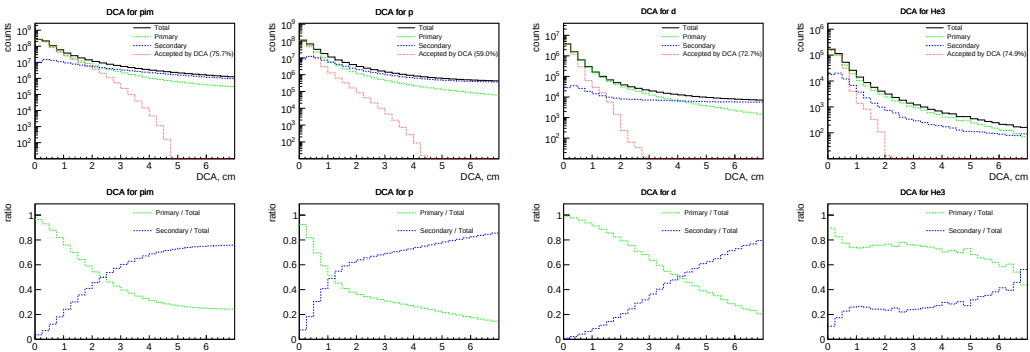


Here DCA $= \sqrt{dca_x^2 + dca_y^2 + dca_z^2}$ (in cm) is plotted.

DCA

DCA cut can be applied by two methods:

- "N-Sigma": $|N\sigma_{x,y,z}| < 2$ – track accepted.



Here DCA $= \sqrt{dca_x^2 + dca_y^2 + dca_z^2}$ (in cm) is plotted, but the DCA cut is in σ.