EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-PH-EP-To be specified
To be specified

# Documentation for ALICE Master Classes 2012 ($R_{AA}$)

R. Averbeck[2)], F. Bock[3)], B. Doenigus[2)], Y. Foka[1)], P. Luettig[4)], S. Masciocchi[2)], K. Schwarz[2)], R. Simon[1)], J. Thaeder[1)] for the ALICE Collaboration [5)]

## Contents



[1]ExtreMe Matter Institute EMMI, GSI Helmholtzzemtrum für Schwerionenforschung, Darmstadt, Germany
[2]GSI, Darmstadt, Germany
[3]Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany
[4]Institut für Kernphysik, Goethe University, Frankfurt
[5]ALICE, CERN, Geneva

# 1 Physics Motivation

## 1.1 The ALICE-Detector System

ALICE (A Large Ion Collider Experiment), one of the four large experiments at the CERN Large Hadron Collider, has been designed to study heavy-ion collisions. It also studies proton proton-proton (pp) collisions, which primarily provide reference data for the heavy ion collisions. In addition, the pp-collision data allow for a number of genuine proton-proton physics studies.
The ALICE detector has been designed to cope with the highest particle multiplicities anticipated for collisions of lead nuclei at the extreme energies of the LHC. It is composed out of different sub-dectors. In the Central Barrel this are the Inner Tracking System (ITS), consisting out of three detectors having each 2 layers, the Time Projection Chamber (TPC), the Transition Radiation Detector (TRD), the Time Of Flight detector (TOF), the High Momentum Particle Identifikation Detector (HMPID), the two electomagnetic calorimeters EMCAL and PHOS and ACCORDE, the dedicated cosmic ray detector. In the following analysis mainly the ITS, TPC, TRD and TOF will be used.

## 1.2 Nuclear Modification Factor $R_{AA}$

In 2010, the first heavy ion collisions at the LHC have been recorded by the experiments. The ALICE experiment, which is well suited for the measurement of the properties of particles in high particle density environments, has published the nuclear modification factor $R_{AA}$ for unidentified charged particles. $R_{AA}$ is a measure for the difference in particle production in pp and PbPb collisions, assuming, that a collision of two Pb ions is the superposition of $N$ pp collisions. $R_{AA}$ is defined as:

$$R_{AA} = \frac{Y(PbPb)}{N_{coll}Y(pp)} \tag{1}$$

with Y(PbPb) and Y(pp) being the yield in PbPb / pp collisions and $N_{coll}$ the number of pp collisions, which have taken place in the collision of two lead ions.
If this nuclear modification factor is unity, the production of particles in pp and PbPb collisions is at least similar.

## 1.3 Centrality

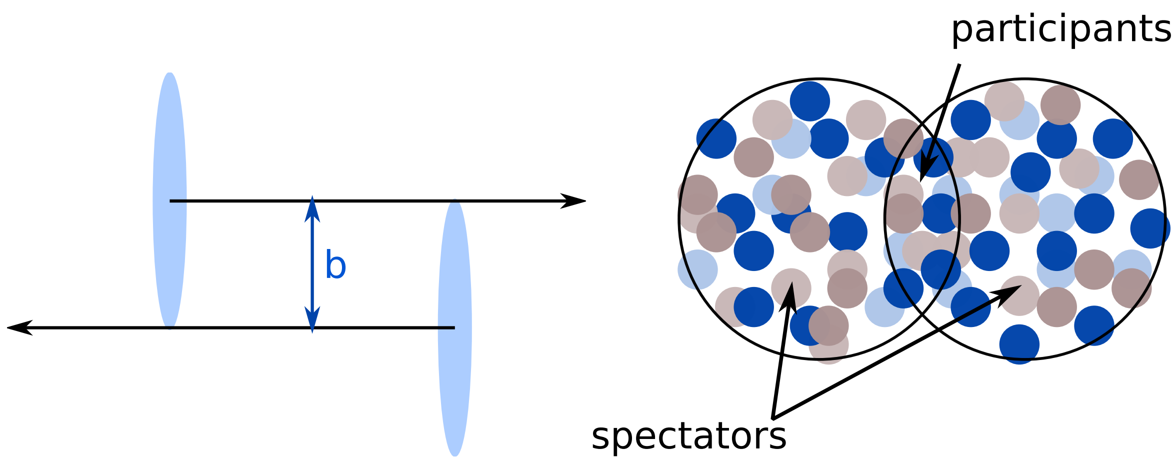


**Fig. 1:** Illustrations of the two colliding nuclei, indicating the impact parameter b (left), the participating nucleons, as well as the spectators (right).

The huge size of lead ion, compared to a proton, leads to the fact, that the type of collision differ. In pp collisions, there are only collisions (not taking into account ultraperipheral collisions), while in the collision of PbPb it needs to be differentiated between different sizes of the overlap region. If the overlap region is maximal, then highest energy densities can be reached and a new state of matter, the quark gluon plasma, is suggested to be created. If the overlap region of the two ions gets smaller, the energy density reduces as well.

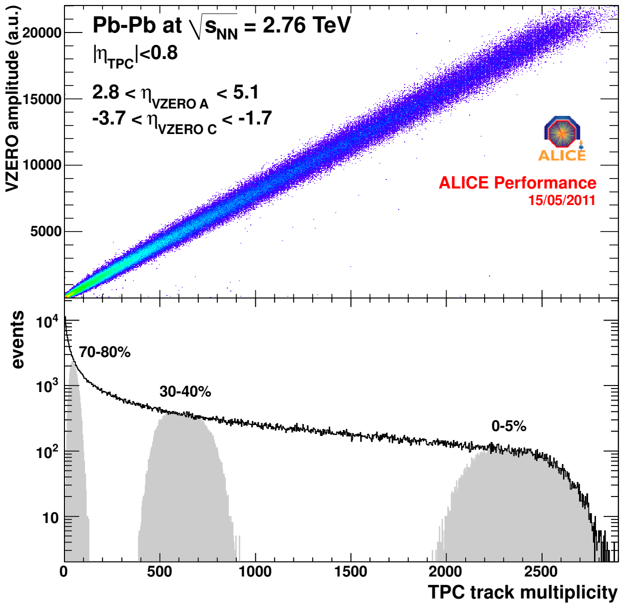


**Fig. 2:** a) Correlation between VZERO amplitude and the Time Projection Chamber (TPC) track multiplicity measured by the ALICE collaboration. b) Minimum bias distribution of the TPC track multiplicity with indicated bins for 70-80% and 0-5% centrality of the Pb-Pb collision at $\sqrt{s_{\rm NN}} = 2.76$ TeV. [**?**]

To see the difference between pp and Pb-Pb collisions one first has to define event classes for heavy-ion collisions. One of the criteria to define such classes is the collision centrality: an event selection related to the impact parameter $b$ (the distance between the colliding nuclei perpendicular to the beam axis). However this parameter is not directly measurable, but it can be determined via multiplicity measurements and model fits to these distributions. In the most central events $(0-5\%)$ two ions collide head-on. Peripheral events ($> 70\%$ centrality), on the other hand, should behave like pp collisions. Quantitative estimates of the collision centrality are given by the number of participating nucleons $N_{part}$, binary nucleon-nucleon collisions $N_{coll}$ or spectators $N_{spec} = 2A - N_{part}$ (where $A$ is the mass number of the initial nuclei), or by the forward hadronic energy $E_{ZDC}$. These quantities can be related to the impact parameter via Glauber Model calculations. ALICE is capable to measure the centrality in four different ways: via the energy deposition in the ZDC or via the multiplicities measured in the SPD, VZERO or TPC detectors. The correlation of two of these measurements is shown in Fig. 2 in the upper panel, in the lower panel the distributions of the uncorrected TPC track multiplicity is shown for minimum bias events. Additionally the most central and most peripheral bins are indicated. The determination of $R_{AA}$ as function of the centrality gives insight to the properties of the medium, since the production of particles in a medium is compared to the production of particles without a medium.

### 1.4 $R_{AA}$ as function of the transverse momentum

Not only the integrated $R_{AA}$ gives insight to the properties of the medium, which the particles are traversing, but the transverse momentum dependece of the nuclear modification factor.
The transverse momentum ($p_t$) is the momentum in the xy-plane perpendicular to the beam axis. It can be calculated by the following formular:

$$p_t = \sqrt{p_x^2 + p_y^2}$$

In a collision of the two nuclei, there might not only be a hot and dense medium created, but there can also be hard interactions between two partons. These two partons will traverse through the medium. Since they can interact strongly, they might loose energy to the medium.
If a particle is fast enough, has a high transverse momentum $p_T$, than a so-called punch-through might happen. High $p_T$ particles are that fast, that they nearly do not interact with the medium and reach the surface of the medium with nearly the same momentum.

| Classification | approximate centrality class | nColl |
|---|---|---|
| peripheral | 80 - 90 % | 6.32 |
| semicentral | 20 - 40 % | 438.80 |
| central | 0 - 5 % | 1686.87 |

**Table 1:** Classification of the PbPb events for the visual analysis and their corresponding number of binary collisions according to the Glauber model (nColl).

For the most central events, a strong transverse momentum dependence is observed, since the size of the fireball is largest in this configuration. The most peripheral events only show a weak $p_T$ dependence, since the medium is much smaller or maybe not existent.

## 2 Visual Analysis

### 2.1 The Task

The goal of this visual analysis is to make you familar with the concept of:

- *clusters*: electronic signature left by a particle traversing a detector, with additional information about time or space
- *track*: Path of a praticle trough the detectorssystem, reconstructed on the basis of the space and time information of individual clusters. The path of the particle is straight in an environment without additional forces (i.e. no magnetic (B) or electric fields (E)). If a charged particle is traversing a magnetic field it will be influence by the magnetic field due to the Lorentz-force and the tracks will be curved.
- *primary vertex*: collision vertex selected for the analysis
- *sattelite collisions/pile-up vertices*: not selected collisions vertices seen in the detector due to high intensity of the LHC-beam
- *primary track*: track orginating in the selected collision vertex (for this analyis distance of closest approach $< 1$ cm)
- *secondary track*: track not orginating in the primary vertex but in a decay vertex (for example strange particle decay)

Furthermore your task will be to count for the 30 pp events at centre of mass energy $(\sqrt{s}) = 2.76$ TeV at a magnetic field of 0.5 T the number of tracks originating in the primary vertex (*multiplicity*) by clicking on each primary tracks. The tool will count for you the total number of tracks in an event and will publish it to a histogram, from which you than can read of the mean of the distribution after having analysed the 30 events. Additionally you should count the multiplicity in one peripheral, one semi-central and one central Pb-Pb event and calculate an integrated $R_{AA}$ for these events.
For the integrated $R_{AA}$ calculation you need to multiply the pp reference with 0.6 and devide the Pb-Pb samples by the number of binary collisions (nColl) given in the Table 1:

### 2.2 The Tool

For this purposes a software has been developed based on the root-environment (`http://root.cern.ch/`). It can visualize the clusters in the main detectors of the central barrel (ITS, TPC, TRD and TOF) and show the reconstructed tracks which passed some quality cuts. Furthermore it can show you the primary vertex and the primary tracks originating in this vertex and it will help you count them. The tool can be started from a terminal in the directory MasterClass2012/Part1 with the command:

```
root masterclass.C
```

Then window shown in Fig. 3 will pop up, in addition you will see a short version of the instructions. There are three things which you can do:

- Start - the analysis.
- Select the data-set which you would like to analyse (the selection field). Please ask your supervisor which data-set you should be taking.
- Exit - finish the analysis.

After having pressed the button “Start”, it will take a while, due to compiling, until the window shown in Fig. 4 will pop up with a shorter version of the full instructions for the visual analysis.
This window is devided in two main sections the left column with the options and the right bigger column with the event display. You can see at the top 3 different tabs “Viewer 1”, “Multi View” and “Event Characteristics”, the first two of them are event displays. “Viewer 1” allows you to see the event in 3D as a single view, while the “Multi View” already presents it to you in two projections ($r\phi$ projection -right upper corner, and rz projection -right lower corner) and in the 3D-view in the middle. In the event-display you can click on anything you want and can navigate either with the mouse by holding the right mouse botton (turning the 3D-view), or by just using the arrows on your keyboard. To zoom in or out please click in the corresponding view and use the “+” or “-” keys on your keyboard. By clicking on the tracks you can count them for you multiplicity distribution, if the counter is open. In addition several properties of the track will be displayed in the upper part of the counter. If you have clicked on a track it turns red. However if you change something in your display settings, like switching on and off the tracks it will be shown in grey/blue again.
For counting the primary tracks please select the “Show primary tracks only”-button first, it will make your life much easier. In case of Pb-Pb you should probalbly turn the clusters off as it will take a lot of time otherwise and you will not see anything anymore. In Fig. 5 you can see a semicentral Pb-Pb event with everything switched on. The multiplicities in the three Pb-Pb events need to be written down on the prepared sheet of paper, as they will not be stored anywhere.
In the “Event Characteristics” -tab (Fig. 6) you can see for histograms, these will be filled for pp at $\sqrt{s} = 2.76$ TeV only. The left column of the main window offers the following options:

- Instructions:
  This button will again display the shorter version of this instructions.
- Event navigation:
  You can navigate between the different events with “next” and “previous”, while the current event number will be displayed in the middle.


**Fig. 3:** Startup-window for the ALICE master classes 2012

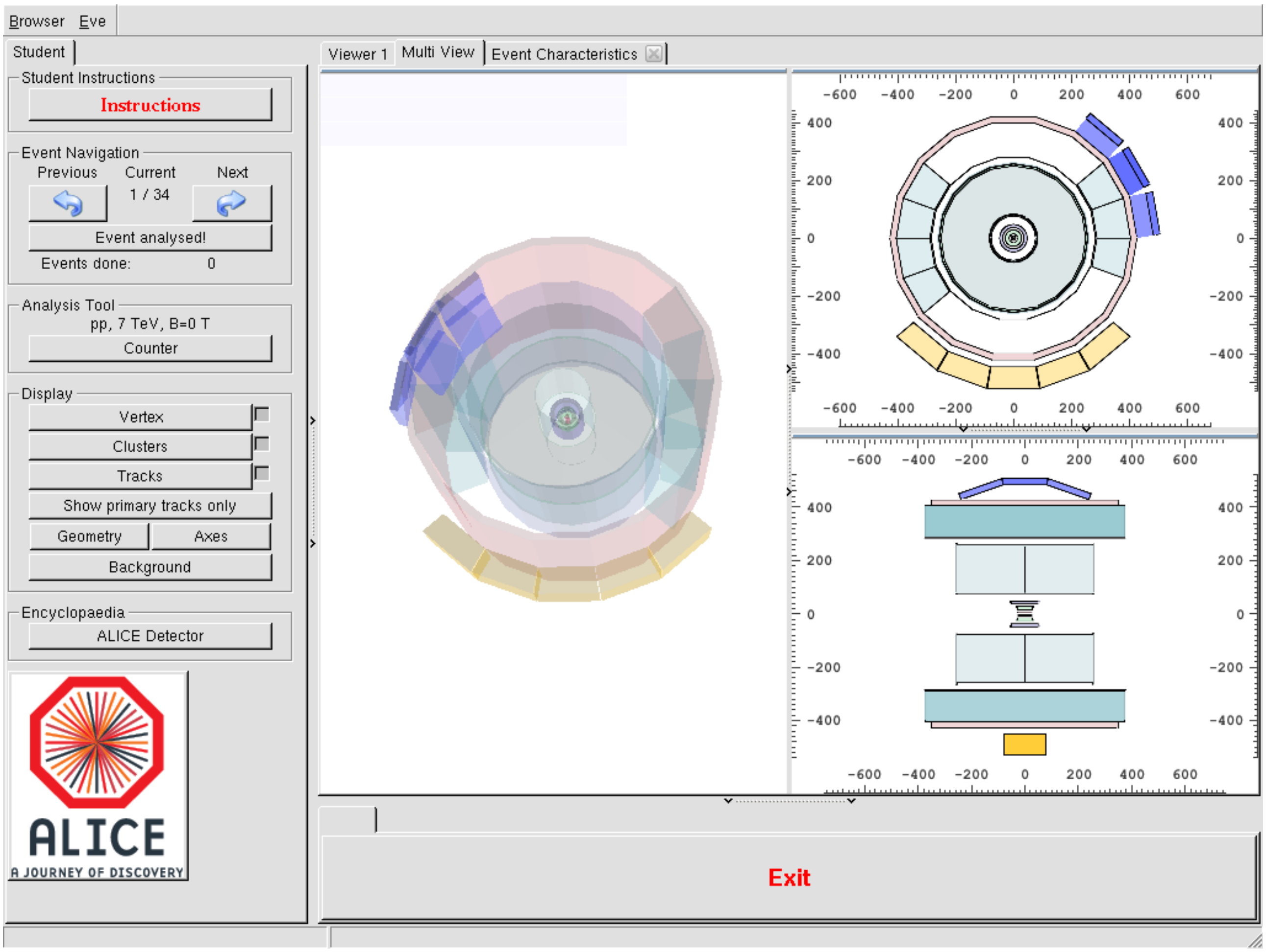


**Fig. 4:** Main window for the visual analysis directly after start-up

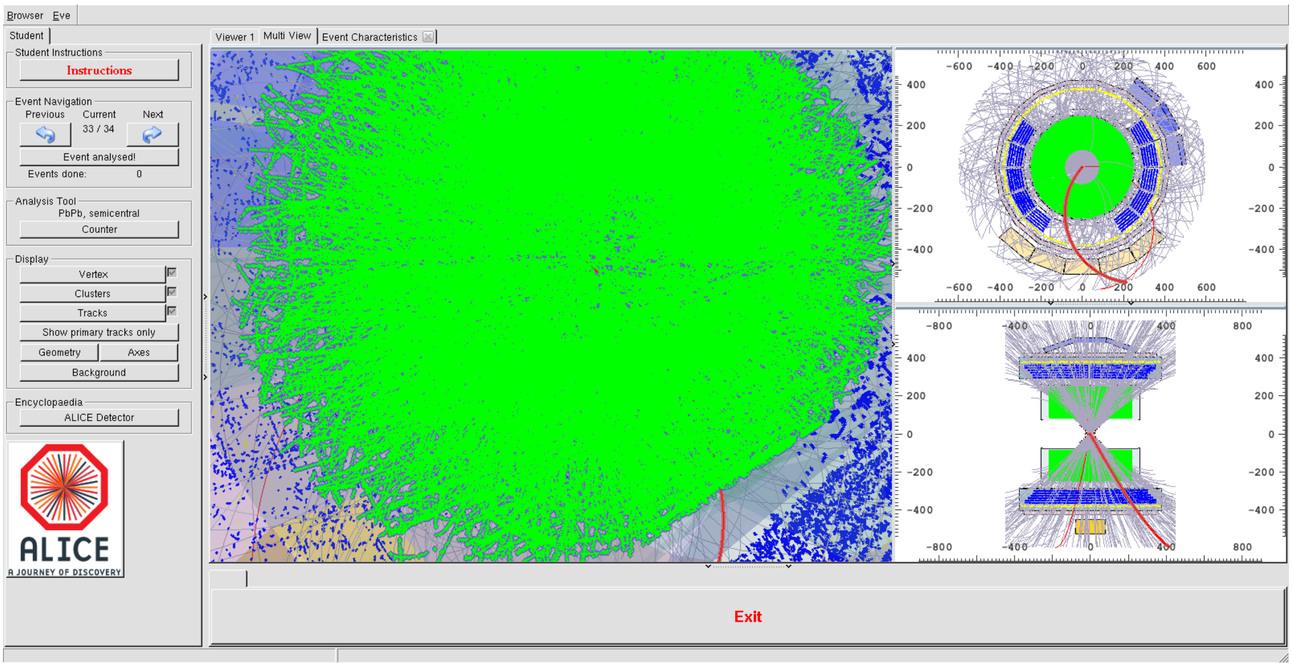


**Fig. 5:** PbPb- event seen in the Master Classes visualisation tool.

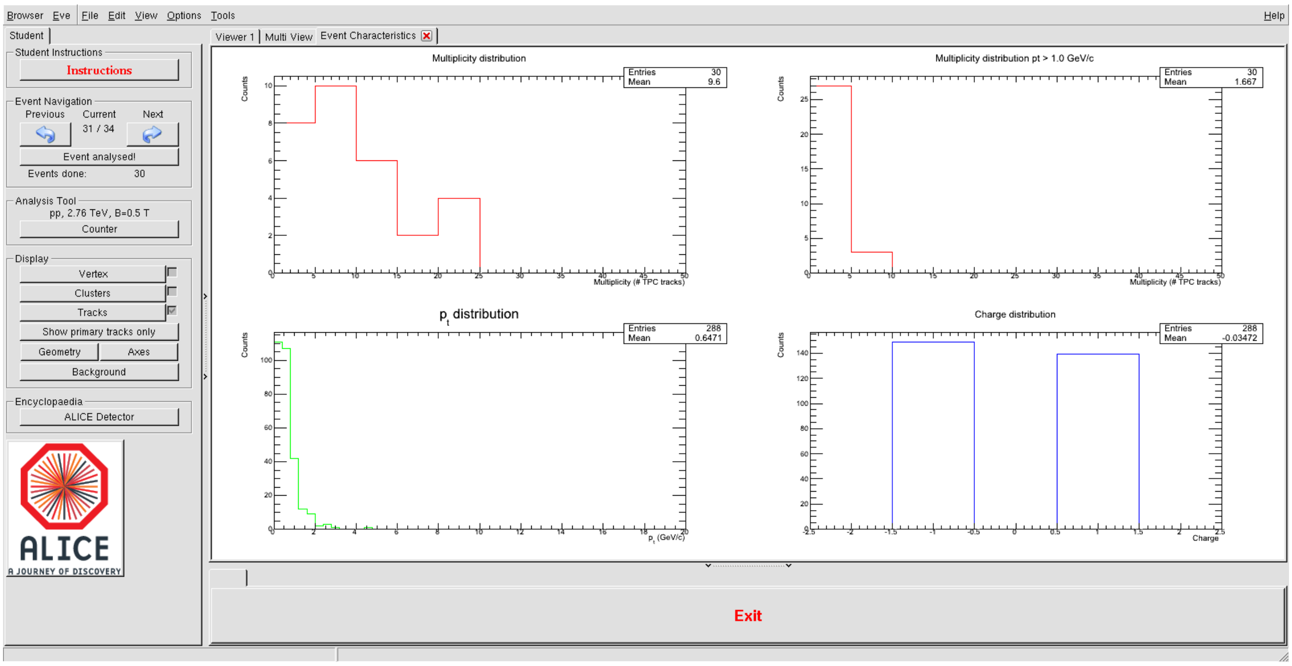


**Fig. 6:** Histograms for the event and track characteristics for pp events at $\sqrt{s} = 2.76$ TeV.

If you click on the “Event analysed”-button the multiplicity will be published to the corresponding histogram. (will be explained later) Additionally it will increase the number of analysed events in the row below the button.

– Analysis Tool:
  In this part you can start the Counting tool according to the event-type being analysed.
  There are 5 different types of the events:

  – 1 pp-event at $\sqrt{s} = 7$ TeV with no magnetic field - $B = 0$ T ($1^{st}$ event)
  – 30 pp-events at $\sqrt{s} = 2.76$ TeV with magnetic field of $B = 0.5$ T
  – 1 peripheral PbPb-event ($32^{nd}$ event)
  – 1 semi-central PbPb-event ($33^{rd}$ event)
  – 1 central PbPb-event ($34^{th}$ event)

  Please close the previous counter if the event-type changes and open a new one, by clicking on the button “Counter”. The two different types shown in figure 7 will show up according to the chosen collision conditions, for the first class of events ( $B = 0$ T) there exists no counting tool. In both counters some properties either of the tracks or the event in total can be displayed, like $p_x, p_y, p_z, p_t$,, charge for tracks or multiplicity for the event. The modes are slightly different for the different counters. In case of pp (2.76 TeV, B= 0.5 T) the $p_T$ and charge will be automatically published to the histograms in the tab “Event Characteristics”. The multiplicities in this case can be either published by the clicking the button “Event analysed” in the main window, or by pressing “Publish to Mult Histogram”. This should only be done if you are sure you counted really all primary tracks, as this is not reversable and will screw up your mean multiplicity otherwise. The “Clear” button resets all entries to 0, however all track-properties which have already been published will remain in the histograms. For the Pb-Pb case there exists the possibilty of automatically counting the primary tracks, please reduce the tracks to the primary tracks only before (Clicking button “Show primary tracks only” in the main window). For Pb-Pb there exists no automatic storring of the multiplicities, so please keep tracks of these yourself on a sheet of paper.

– Display:
  This part allows you to switch on and off several features of the event/ event display (like primary vertex - “Vertex”, clusters - “Clusters”, tracks - “Tracks”, geometry of the detectors -“Geometry”, coordinate

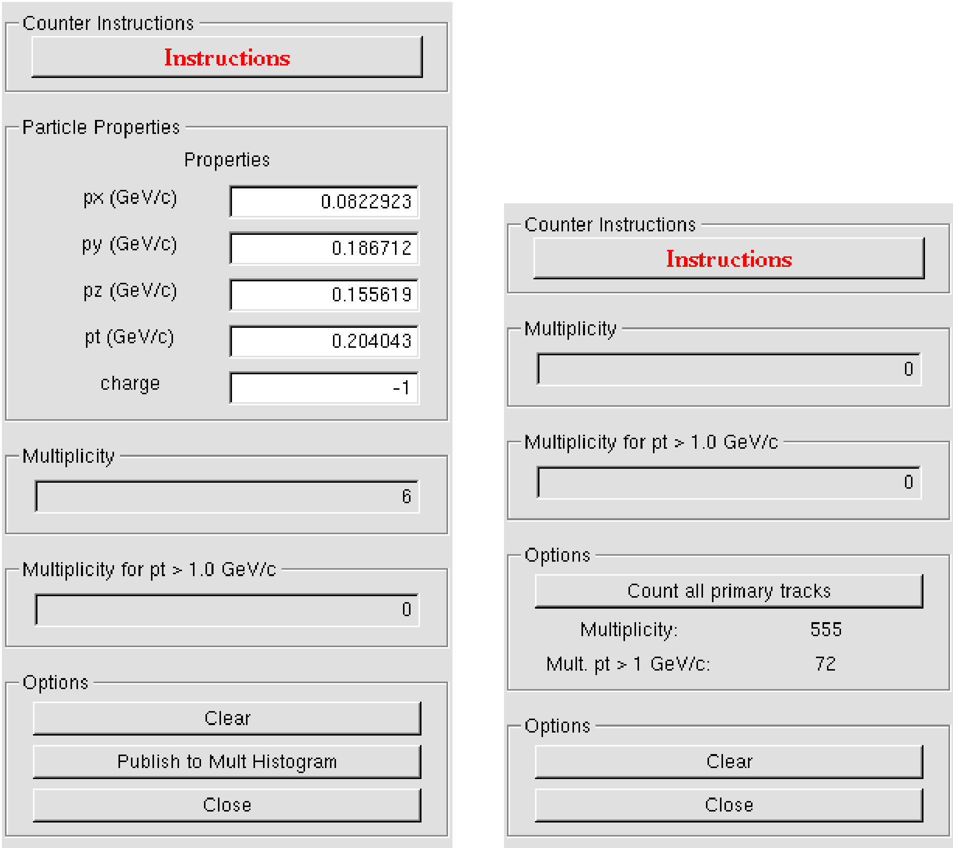


**Fig. 7:** Counting-tool for pp - 2.76 TeV, B= 0.5 T (left) and Pb-Pb -2.76 TeV, B= 0.5 T (right)

axis - “Axis” and change the background color “Background”). Furthermore it allows you to reduce the tracks to the tracks orignating in the primary vertex by pressing “Show primary tracks only”. If you press it twice it will return to the full number of good tracks in the event.

– Encyclopaedia:
In this section you can learn a little more about the ALICE detector, furthermore informations about the individual detectors will pop up by clicking at the detector volumes in the event display. If you want to close them again just click in the window.

## 3 The Large Scale Analysis

### 3.1 The Task

In this part of the ALICE master classes you shall be introduced to a large scale analysis based on real Pb-Pb data. Your task will be to implement the extraction of the PbPb- transverse momentum spectra in a given centrality class from a prepared data-sample (tree), which contains the centrality of the event, the track multiplicity and the transverse momentum of each track. Furthermore you are supposed to write a program which calculates the $R_{CP}$ and $R_{AA}$ and plots the transverse momentum spectra, the track multiplicity, the $R_{CP}$ and $R_{AA}$ for different centralities in the same plot.

### 3.2 Building the Transverse Momentum Spectra

For this part of the analysis you need to program approximately 10-15 lines of code in a prepared root-macro, the macro is called “AnalyseTreeForRAAStudents.C” and can be found in the folder MasterClasses2012/Part2. It can be started with the command in this directory:

```
root -x -q -b -l 'AnalyseTreeForRAAStudents.C++("MasterClassesTree_LHC10h_Run139036.root
    ","PbPb","kFALSE",0,5)'
```

The options after the root-call ("-x -q -b -l") are some settings for running root in quite mode and executing the macro which is following afterwards. The real function call follows, first there is the name of the macro "AnalyseTreeForRAAStudents.C", then the "++" stands for compiled mode and afterwards several options are given to the macro itself.

1 filename = "MasterClassesTree_LHC10h_Run139036.root"
This is the name of the file you would like to analyse. This contains the data elements seen in Figure **??**.

2 collision system = "PbPb"
This is the selected collision system which you will be analysing, in your case this will always be "PbPb".

3 test mode = "kFALSE"
If you set this variable to "kTRUE" it will be run in test mode and will only analyse 1000 events and 1000 tracks, which makes it much faster.

4 start of centrality class = 0
This variable determines the starting value of your centrality class.

5 end of centrality class = 5
This variable determines the end value of your centrality class. In this example the most central bin from 0-5 % will be analysed.

The macro is structured as follows:
First there are two functions to make the plots look nicer

```
void StyleSettings()
void HistoSetMarkerAndColor( TH1* histo1, Style_t markerStyle, Size_t markerSize, Color_t
    markerColor, Color_t lineColor )
```

These two you should not modify however you should read the comments above them to get an idea how to use them. Comments in root/C++ are starting with "//" or "/* —- text —-*/, the statements following this expression, or in between for the second case, will never be executed.
Afterwards the main-functions is starting, it has the same name as the macro itself just without the ".C" in the end, the code belonging to a function is always within these brackets { }, the same accounts for loops or conditional statements.

```
void AnalyseTreeForRAAStudents(TString filename = "MasterClassesTree_LHC10h_Run139036.
    root", TString optionCollSystem = "PbPb", TString optionTest= "kFALSE", Int_t
    startCentrality = 0., Int_t endCentrality = 100.)
```

The options for the main-function are already explained above, here just the variable names and the standard settings are given in addition.
In this main function the first task is to attach the file and set the correct variable names

```
//****************************************************************************
//Declaration of leaves types for track tree
//****************************************************************************
Float_t          trackCentrality;  // variable for the centrality in the track tree
Double_t         trackPt;        // variable for the transverse momentum of the track

//****************************************************************************
//Declaration of leaves types for event tree
//****************************************************************************
Float_t          eventCentrality;  // variable for the centrality in the event tree
Int_t            eventMult;       // variable for the multiplicity in the event
```

Afterwards these are related to the quantities in the corresponding parts of the input tree (Line 142 - 155) and the number of entries in each tree is evaluated (Line 161 - 183). Then the binning in transverse momentum is determined and the histograms are created.

```
    //*************************************************************************
186 // Definition of bins in transverse momentum pt, due to steply falling spectrum
    // and not enough statistics at high pt
188 //*************************************************************************
    Int_t fNBinsPt = 54;
190 Double_t fBinsPt[55] =  {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45,
                    0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95,
192                 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9,
                    2, 2.2, 2.4, 2.6, 2.8, 3, 3.2, 3.4, 3.6, 3.8,
194                 4, 4.5, 5, 5.5, 6, 6.5, 7, 8, 9, 10,
                    11, 12, 13, 14, 15};
196

198
    //*************************************************************************
200 // Defintion of histograms:
    // * to be filled with trackpt and number of charged tracks in the TPC (TH1*)
202 // * correlation between centrality an nTracks TPC (TH2F)
    //*************************************************************************
204
    TH1D *htrackPt = new TH1D("htrackPt","track pt",fNBinsPt,fBinsPt);
206 TH1F *hNTPC = new TH1F("hNTPC","Number of TPC tracks ",200,0,2000);
    TH2F *hNTPCvsCent = new TH2F("hNTPCvsCent","Number of TPC tracks ",200,0,2000, 100, 0,
        100);
208
    //*************************************************************************
210 // Definition of correction histogram in order to normalise for the binwidth in
    // Pt
212 //*************************************************************************
    TH1D* fDeltaPt = new TH1D("deltaPt","",fNBinsPt,fBinsPt);
214 for(Int_t iPt=1;iPt<fNBinsPt+1;iPt++){
      fDeltaPt->SetBinContent(iPt,fBinsPt[iPt]-fBinsPt[iPt-1]);
216   fDeltaPt->SetBinError(iPt,0);
    }
```

Then the event quantities are read from the event-tree and filled in the corresponding histograms. Therefore a loop over all entries in this tree has to be created (line 230) and the each event has to be checked whether it belongs to the correct centrality class (line 232). The lines 233 and 234 fill the 1 dimensional and two dimensional histograms for the number of tracks (versus centrality).

```
      //*************************************************************************
220   // * Reading the entries form the event tree (extract nTracksTPC) and filling
      //   it in the multiplicity histograms (hNTPC,hNTPCvsCent)
222   // * Distinction between PbPb and pp as for PbPb you need to fill them with
      //   restriction in centrality
224   // * nEntriesPerCent will give you the normalization value for the different
      //     centralities
226   //*************************************************************************

228   ULong_t nEntriesPerCent= 0;
      ULong64_t nbytes2 = 0;
230   for (ULong_t i=0; i<nEntriesEvent;i++) {
        nbytes2 += Event->GetEvent(i);
232     if (eventCentrality > startCentrality && eventCentrality < endCentrality){
          hNTPC->Fill(eventMult);
234       hNTPCvsCent->Fill(eventMult,eventCentrality);
          nEntriesPerCent++;
236     }
        // give an output for every 10Mio events processed to see that is working
```

```
    if ( i%10000000 == 0 ) {
      cout << i/10000000 << " * 10^7 events have been processed" << endl;
    }
  }
```

A similar loop has to be created for the track-tree as the "/// To do: " in line 254 indicates. Afterwards the filled histograms need to be plotted and saved. For the plotting an example for 2D and 1D histogram is given in lines 269 - 300 and a similar plot should be produced for the transverse momentum as indicated in line 312. Please don't forget to uncomment the lines 315-319, otherwise the scaling will no be done correctly and the line 341 for saving the histogram.

### 3.3 Building the $R_{AA}$ and $R_{CP}$

For this part of the analysis you need to program approximately the same amount of code as in the exercis before in a prepared root-macro, the macro is called "BuildRAAStudents.C" and can be found in the same directory as the previous one. It can be started by running the following command in this directory:

```
root -x -q -b -l 'BuildRAAStudents.C++("RAABaseOutput.root")'
```

As already described in the previous section the first commands are just some root settings and the the macro-name follows, which should be executed in compiled mode as the "++" indicates. The only option which you can had over is the file name of the input file for the Pb-Pb- results which are provided by the previous exercise. This file will contain the results for different centralities, you just need to find out how to read them from the file.
The first part in the macro "BuildRAAStudents.C" are again the functions for the styling of the plots. Then follows the main-routine:

```
void BuildRAAStudents(TString filename = "RAABaseOutput.root")
```

In this function the first few lines are for general setting and then follows the definition of the variable for the number of collisions in each centrality - *nColl_0_5* -, where the first number indicates the start centrality and the second one the end of the centrality bin (lines 111-123).
Afterwards the Pb-Pb and p-p spectra are read from the corresponding files and scaled to the number of collisions, here you need to add the other centrality classes as indicated in line 132 and 140. The pp spectrum gives you the baseline for the $R_{AA}$.

```
    //****************************************************************************
126 //Attaching & reading the input-file
    //****************************************************************************
128 TFile fileInput(filename.Data());
    //____ reading the number of TPC tracks for pp events & 0-5% PbPb events__________
130 TH1D *hNTracksTPCPbPb_0_5 =      (TH1D*)fileInput.Get(Form("nTracksTPC_PbPb_%i-%i",0,5))
        ;
    TH1D *hNTracksTPCPbPb_70_80 =      (TH1D*)fileInput.Get(Form("nTracksTPC_PbPb_%i-%i"
        ,70,80));
132 /// To do: Do the same for the other centralities

134 //_____reading the pt-spectrum for 0-5% & 70-80% PbPb events, scaling it__________
    //_____by corresponding number of Collisions (nColl)_______________________________
136 TH1D *hTrackPtPbPb_0_5 =      (TH1D*)fileInput.Get(Form("trackPt_PbPb_%i-%i",0,5));
    hTrackPtPbPb_0_5->Scale(1./nColl_0_5);
138 TH1D *hTrackPtPbPb_70_80 =      (TH1D*)fileInput.Get(Form("trackPt_PbPb_%i-%i",70,80));
    hTrackPtPbPb_70_80->Scale(1./nColl_70_80);
140 /// To do: Do the same for the other centralities

142 //****************************************************************************
```

```
    // Attaching and Reading pp-reference file
144 //****************************************************************************
    TFile* fileInputPP =    new TFile("PP_2760GeV_BaseLine.root");
146 TH1F *hNTracksTPCpp =     (TH1F*)fileInputPP->Get("nTracksTPC_pp");
    TH1D* hTrackPtpp =      (TH1D*)fileInputPP->Get("trackPt_pp");
```

Then the $R_{CP}$ is build from the most central (0-5%) and the most peripheral bin (70-80%), therefore you need to divide most central spectrum scaled to nColl by the most peripheral spectrum scaled to the corresponding nColl. As seen in lines 149 ff, this should be done in a similar manner for the other centralities and the $R_{AA}$ where the pp reference is used as divisor (“To do” in line 161).

```
    //****************************************************************************
150 // Building the RCP
    //****************************************************************************
152 TH1D* hRCP_0_5 = (TH1D*)hTrackPtPbPb_0_5->Clone("RCP_vs_Pt_0-5");
    hRCP_0_5->Sumw2();
154 hRCP_0_5->Divide(hTrackPtPbPb_70_80);
    /// To do: Do the same for the other centralities
```

The lines 166 to the end then just show the routines for plotting as already explained in the previous section, here you should add the other centralities as well as the plot for the $R_{AA}$. Don’t forget to put the axis-labels as well as the legends an example for the legend is given in line 189ff.