Analysis on ALICE data

LNF Alice Weekly Meeting
January 31st, 2017

Silvia Pisano

Potential physics topics

Light Flavor

- Production of light hadrons (π, κ, p) *vs.* centrality

    1. analysis of p_T distributions and yields
    2. comparison to models (hydrodynamics, thermal…)

Phys. Rev. C 88 (2013) 044910 and available at arXiv:1303.0737
http://prc.aps.org/abstract/PRC/v88/i4/e044910

- Multi-strange baryon production

    1. study the emergence of collective phenomena
    2. enhancements (yields per partecipant nucleon relative to those in pp collisions)
    3. p_T spectra, rapidity, centrality

Phys. Lett. B 728 (2014) 216–227
http://www.sciencedirect.com/science/article/pii/S0370269313009544

Jets

- Analysis of the hadronization process: how do we get from the QCD Lagrangian degrees of freedom (i.e., quarks and gluons) to the «real» object that we observe in our detectors

- hadron formation in the «simple» pp system or in the nuclear medium produced through $PbPb$ collisions

- Jet is not a real object, but an algorithm ⇒ any assumption beyond its defition has to be taken under control, with the associated systematics

The real challenge in Alice is to have a good control of the Jet neutral component (EMCal physics)

Phys.Lett. B722 (2013) 262-272
http://inspirehep.net/record/1210881/

How to get into the business

1) Start with simple tasks to get familiar with technical aspects

   - Alice software environement
   - data: localization, typologies, access
   - Run Alice analysis code…and develop yours inside the framework

2) Preliminary studies:

   - LIGHT FLAVOR: reconstruction of a clear signal through tracks: $\varphi \rightarrow \kappa\kappa$
   - JET: analysis of calorimeter resolutions by analyzing the neutral pion invariant mass reconstructed through two photons: $\pi^0 \rightarrow \gamma\gamma$

Potential physics topics
…and how to get there

Installing Alice software environment

1. For Linux users: CentOS 7 (Alice software specialist's suggestion)

2. As a first thing, install AliBuild, AliPhysics and AliRoot. All the details can be found at the link https://alisw.github.io/alibuild/

3. Once the installation is done, check the libraries from the different analysis working groups through the command

   cd $HOME/alice/sw/BUILD/AliPhysics-latest/AliPhysics
   ctest

4. Start «AliEn», the ALIce software Environment through the command

```
[pisanos@lnf1 ~]$ alienv load AliPhysics/latest
Use alienv list to list loaded modules.
[pisanos@lnf1 ~]$
[pisanos@lnf1 ~]$ alienv list
Currently Loaded Modulefiles:
  1) BASE/1.0                    5) boost/v1.59.0-1              9) GEANT4/v4.10.01.p03-1      13) AliRoot/0-1
  2) AliEn-Runtime/v2-19-le-1    6) cgal/v4.6.3-1                10) vgm/4.3-1                 14) AliPhysics/latest
  3) GSL/v1.16-1                 7) fastjet/v3.2.1_1.024-alice1-1 11) GEANT4_VMC/v3-2-p1-1
  4) ROOT/v5-34-30-alice7-1      8) GEANT3/v2-2-1                12) Vc/1.2.0-1
[pisanos@lnf1 ~]$
```

Run test for checking libraries

```
[pisanos@lnf1 AliPhysics]$ ctest
Test project
/afs/cern.ch/user/p/pisanos/work/alice/analysis/sw/BUILD/3bafc5de58c9bffc43df2b0dcef67085bbd8
c1f1/AliPhysics
      Start  1: load_library_CORRFW
 1/94 Test  #1: load_library_CORRFW .....................   Passed    6.57 sec
      Start 69: load_library_PWGLFrsnextra
69/94 Test #69: load_library_PWGLFrsnextra .............   Passed    0.42 sec
      Start 70: load_library_PWGLFspectra
70/94 Test #70: load_library_PWGLFspectra ..............   Passed    0.64 sec
      Start 71: load_library_PWGLFspectraJET
71/94 Test #71: load_library_PWGLFspectraJET ...........   Passed    0.74 sec
      Start 72: load_library_PWGLFSTRANGENESS
72/94 Test #72: load_library_PWGLFSTRANGENESS ..........   Passed    0.48 sec
      Start 73: load_library_PWGLFthermalfits
73/94 Test #73: load_library_PWGLFthermalfits ..........   Passed    0.35 sec

94/94 Test #94: histmgr_fill_grouped ...................   Passed    0.17 sec

100% tests passed, 0 tests failed out of 94

Total Test time (real) =  71.28 sec
[pisanos@lnf1 AliPhysics]$
```

Installing AliEn client on your pc - 1

```
[pisanos@lnf1 ~]$ wget http://alien.cern.ch/alien-installer
--2017-01-30 16:17:53--  http://alien.cern.ch/alien-installer
Resolving alien.cern.ch (alien.cern.ch)... 137.138.99.142, 2001:1458:201:b49f::100:8
Connecting to alien.cern.ch (alien.cern.ch)|137.138.99.142|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 59563 (58K)
Saving to: 'alien-installer'

100%[==========================================================
============================================================
============================================================
============>] 59,563      --.-K/s   in 0s

2017-01-30 16:17:53 (190 MB/s) - 'alien-installer' saved [59563/59563]

[pisanos@lnf1 ~]$ chmod +x alien-installer
```

Installing AliEn client on your pc - 2

```
[pisanos@lnf1 ~]$ ./alien-installer
   Installer type is user or compile, we need to check for a valid certificate
   Found certificate: /afs/cern.ch/user/p/pisanos/.globus/usercert.pem
   Certificate subject: subject= /DC=ch/DC=cern/OU=Organic
Units/OU=Users/CN=pisanos/CN=671712/CN=Silvia Pisano
   Certificate issuer: issuer= /DC=ch/DC=cern/CN=CERN Grid Certification Authority
   Certificate valid. Moving forward with AliEn installation.
   ########################## Automated AliEn compile installer
##################################
   Installing in the default directory: /afs/cern.ch/user/p/pisanos/alien
   If you wish to install in a specific directory do: ./alien-installer -install-dir <path>
   Platform set as Linux: x86_64-unknown-linux-gnu
   Waiting 10 seconds. Press 'Ctrl+c' to exit the installation.....    Done
   Compiling packages : xrootd xrootd-xalienfs igtf-preinstalled-bundle-classic into
/afs/cern.ch/user/p/pisanos/alien.v2-19.192
   Building package xrootd


   Building package xrootd-xalienfs
   Building package igtf-preinstalled-bundle-classic
   Relocating the compile installation.   Done
   Installation finished!
[pisanos@lnf1 ~]$
```

Installing AliEn client on your pc - 3

```
[pisanos@lnf1 ~]$ export PATH=$PATH:~/bin
[pisanos@lnf1 ~]$ alien-token-init pisanos
------------------------------------------------------------
Setting central config:
============================================================
export
alien_API_SERVER_LIST="pcapiserv03.cern.ch:10000|pcapiserv08.cern.ch:10000|"
export TERMINFO=/usr/share/terminfo
============================================================
------------------------------------------------------------
Setting closest site to:
============================================================
Using X509_CERT_DIR=/afs/cern.ch/user/p/pisanos/work/alice/analysis/sw/slc7_x86-
64/AliEn-Runtime/v2-19-le-1/globus/share/certificates
=> Trying to connect to Server [1] root://pcapiserv08.cern.ch:10000 as User pisanos
/alice/cern.ch/user/m/mspyrop/
/alice/cern.ch/user/p/pisanos/
Your identity: pisanos
Creating token ..... Done
Your token is valid until: Tue Jan 31 16:30:23 2017
[pisanos@lnf1 ~]$
```

Getting Alice data files

- start ***aliensh*** through

```
[pisanos@lnf1 ~]$ aliensh
 [ aliensh 1.0.140x (C)
ARDA/Alice: Andreas.Joachim.Peters@cern.ch/Derek.Feichtinger@cern.ch]
aliensh:[alice] [1] /alice/cern.ch/user/p/pisanos/ >
```

- …and copy the file you need thorugh the syntax

```
aliensh:[alice] [13] /alice/cern.ch/user/p/pisanos/ >cp
/alice/data/2011/LHC11h_2/000167813/ESDs/pass2/AOD115/0002/AliAOD.root
file:/afs/cern.ch/user/p/pisanos/work/alice/analysis/tutorials/.
[xrootd] Total 277.17 MB        |====================|  100.00 % [43.6 MB/s]
aliensh:[alice] [14] /alice/cern.ch/user/p/pisanos/ >
```

Data file path:
/alice/data/2011/LHC11h_2/000167813/ESDs/pass2/AOD115/0002/AliAOD.root

data_taking: changes at any detector configuration	run number (restart with any acquisition – e.g., after a beam dump)	reconstruction pass	number of sub-jobs on the GRID

Alice data files

There are two typologies of data files in Alice

ESD FILES

These files contains the information coming from all the subdetectors.

E.g., they contain information from the trackins subsystems (ITS, TPC) that are combined to produce a list of tracks for any event

AOD FILES
(analysis object data)

These are the actual files on which the analysis is performed.

They have predefined cuts for particle identification ⇒ no flexibility

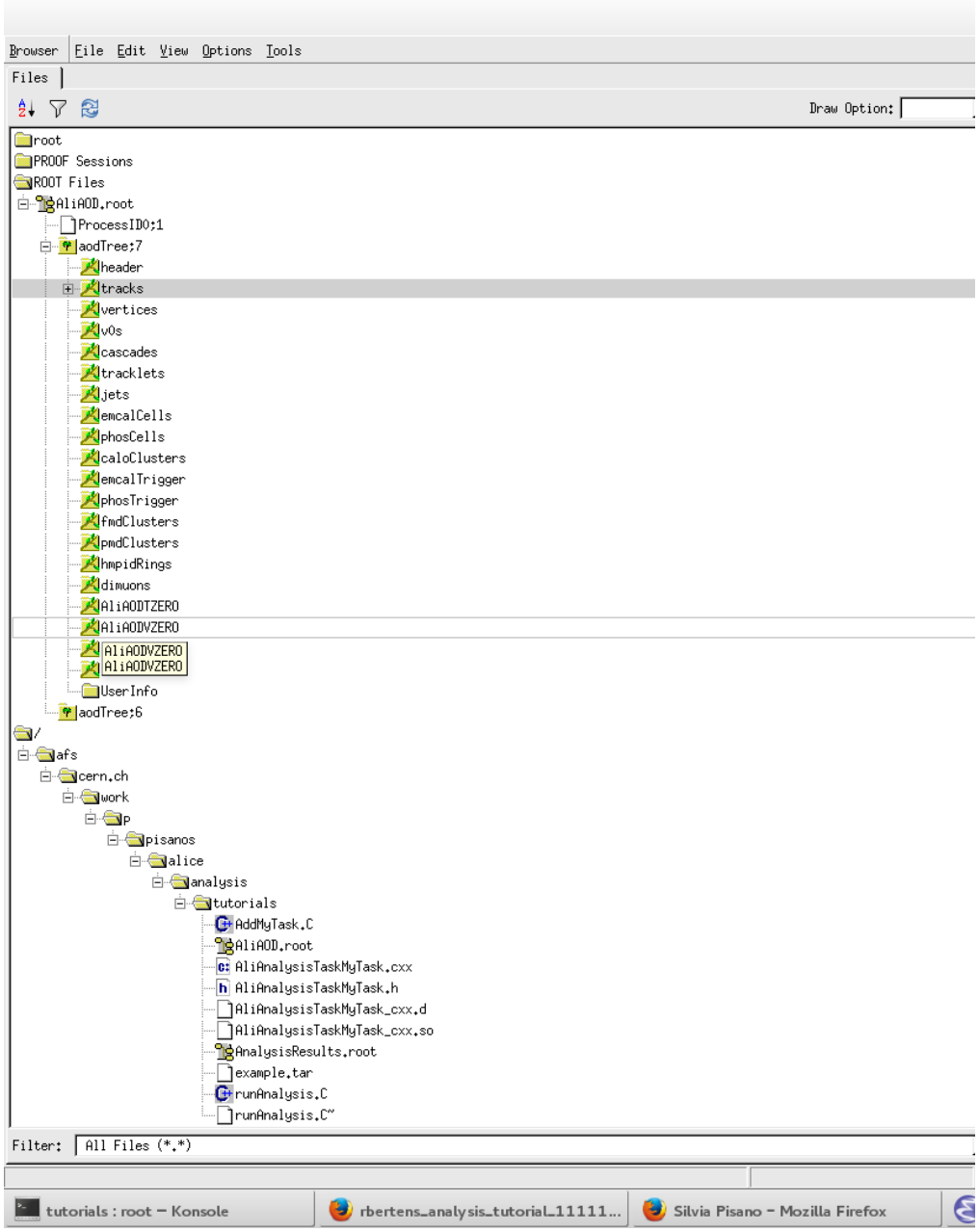
Skimmed data for higher-level analysis

If, e.g., one needs to check how a spectrum is affected by the reconstruction criteria, he/she has to rely on ESD files.

Alice AOD data structure

Let's start by looking at the «tracks» in a given event.

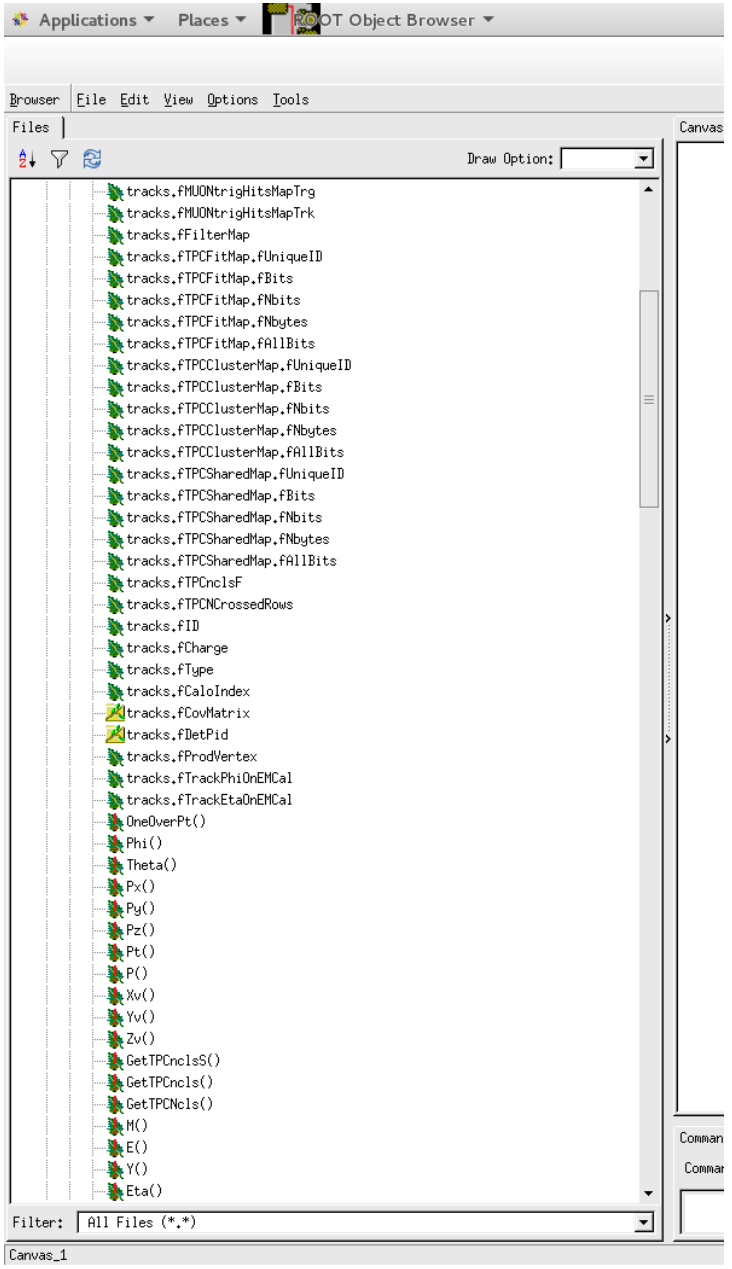
They are a collection in the event rootfile, and contain all the variables associated to a track object: kinematical properties, detector information etc.



Alice AOD data structure

Let's start by looking at the «tracks» in a given event.

They are a collection in the event rootfile, and contain all the variables associated to a track object: kinematical properties, detector information etc.



Run MyAnalysisTask.C – the starting point for any Alice analysis

Download the example.tar file with the beginning C++ class

https://indico.cern.ch/event/586577/contributions/2363125/attachments/1369962/2077200/example.tar

Go to your working directory and launch alienv

[pisanos@lnf1 tutorials]$ alienv load AliPhysics/latest

Run runAnalysis.C:

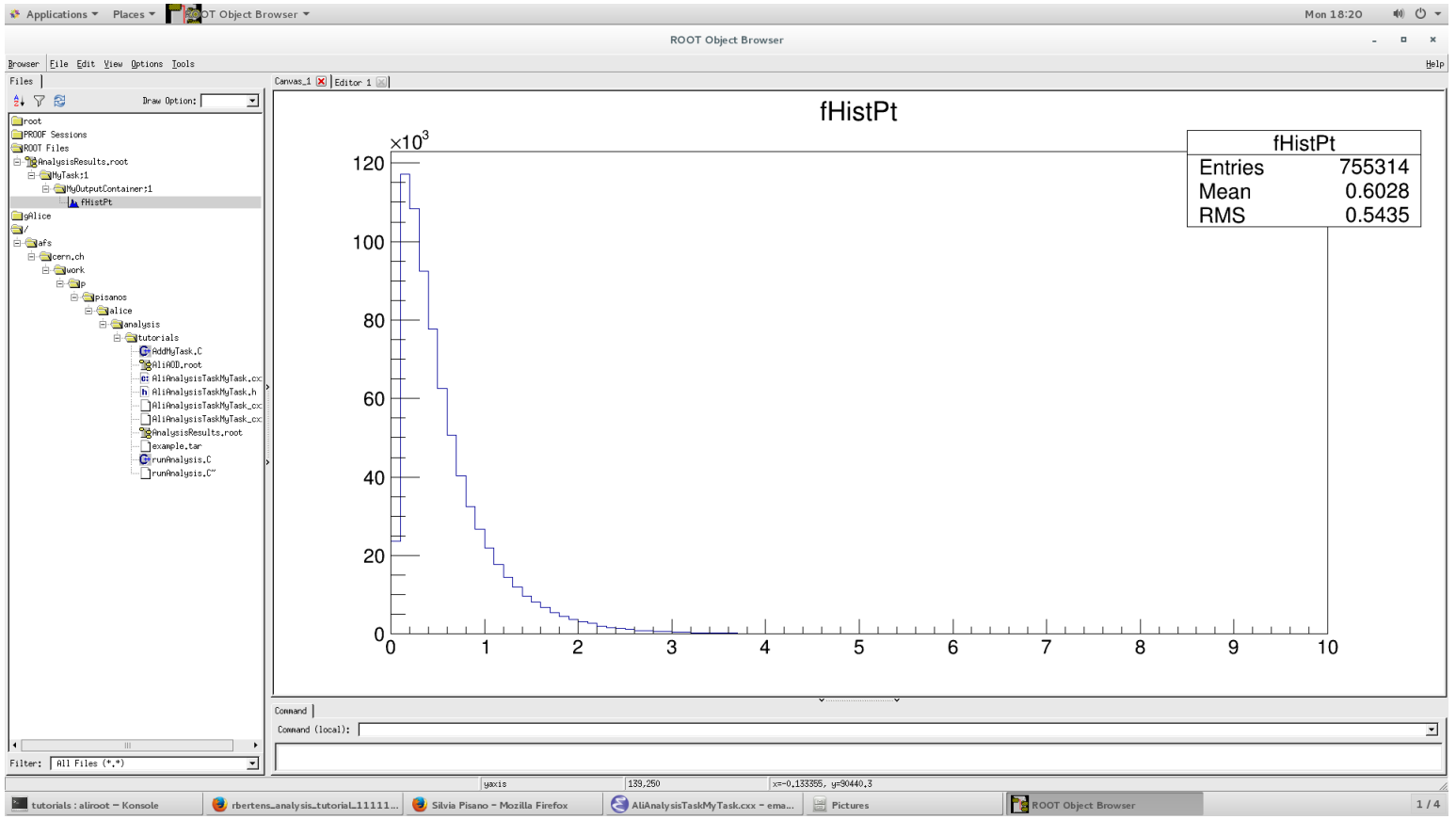
[pisanos@lnf1 tutorials]$ aliroot runAnalysis.C

…you will find the output rootfile AnalysisResults.root

Your first piece of code for Alice analysis

```
void AliAnalysisTaskMyTask::UserExec(Option_t *)
{
   // user exec
   // this function is called once for each event
   // the manager will take care of reading the events from file, and with the static function InputEvent()
you
   // have access to the current event.
   // once you return from the UserExec function, the manager will retrieve the next event from the chain
   fAOD = dynamic_cast<AliAODEvent*>(InputEvent());    // get an event (called fAOD) from the input
file
                                        // there's another event format (ESD) which works in a similar wya
                                        // but is more cpu/memory unfriendly. for now, we'll stick with aod's
   if(!fAOD) return;                            // if the pointer to the event is empty (getting it failed) skip this
event
      // example part: i'll show how to loop over the tracks in an event
      // and extract some information from them which we'll store in a histogram
   Int_t iTracks(fAOD->GetNumberOfTracks());        // see how many tracks there are in the event
   for(Int_t i(0); i < iTracks; i++) {              // loop ove rall these tracks
      AliAODTrack* track = static_cast<AliAODTrack*>(fAOD->GetTrack(i));       // get a track (type
AliAODTrack) from the event
      if(!track) continue;                      // if we failed, skip this track
      fHistPt->Fill(track->Pt());                  // plot the pt value of the track in a histogram
   }                                    // continue until all the tracks are processed
   PostData(1, fOutputList);                        // stream the results the analysis of this event to
                                        // the output manager which will take care of writing
                                        // it to a file
```

Output of your first analysis task



Next steps

- Set up the «intensive» GRID usage: submitting jobs, having them running and, *possibly*, not crashing
- Start with the simple reconstruction of the afore-mentioned channels: $\varphi \rightarrow \kappa\kappa$ and $\pi^0 \rightarrow \gamma\gamma$ through the invariant mass technique
- Fit the previous signals through a suitable function ⇒ understanding how to model the background
- Extract the relevant kinematical dependences (p_T, η)
- …and study their variation with different cuts at the reconstruction level (access to the ESD files)