MC usage in ALICE

Redmer Alexander Bertens (UTK) on behalf of the ALICE collaboration

17th MCnet meeting

April 11, 2018, CERN, Switzerland

what we will cover

available generators and usage

MC productions and integration in the ALICE software framework

MC generators and RIVET

Generators in the ALICE software framework

(red) indicates main use case, presented in no preferred order

- Pythia (pp)
  - Various versions (6.x, 8.x)
  - Various tunes (Perugia, Monash, etc)
- PHOJET / DPMJET (pp)
- EPOS LHC / EPOS 3 (pp, p-Pb, Pb-Pb)
- Herwig / Herwig++ (pp)
- Sherpa (pp)

Generators in the ALICE software framework

- POWHEG (interfacing)
- STARLIGHT (ultra peripheral collisions)
- HIJING (Pb-Pb)
- AMPT (Pb-Pb)
- Therminator2 (Pb-Pb)
- JEWEL (Pb-Pb)

Analyzers **can** run generators **not** included in the software framework but for reasons of **validation** and efficiency this is **discouraged**

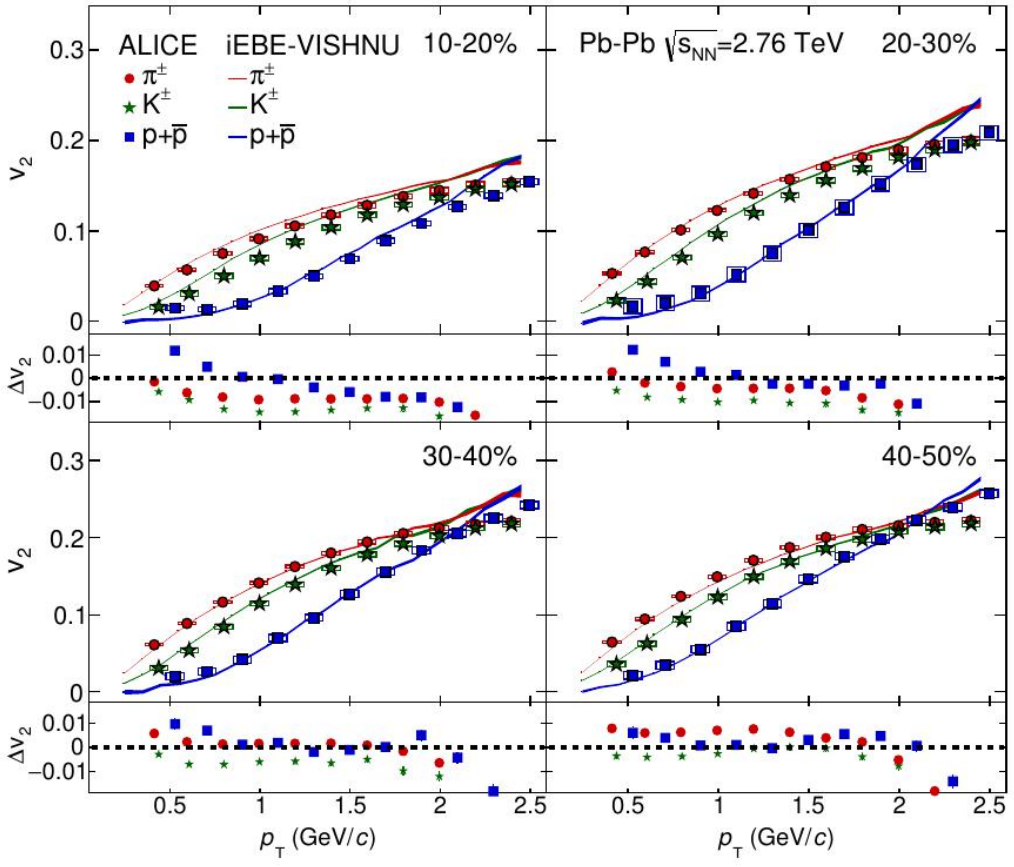
usage of generators

hypothesis testing

corrections and modeling

Usage: hypothesis testing

JHEP 1609 (2016) 164



Generator relies on
- Ansatz / hypothesis
- (Tunable) parameters

Comparison between MC and data
- Confirm / refute hypothesis
- Constrain parameters

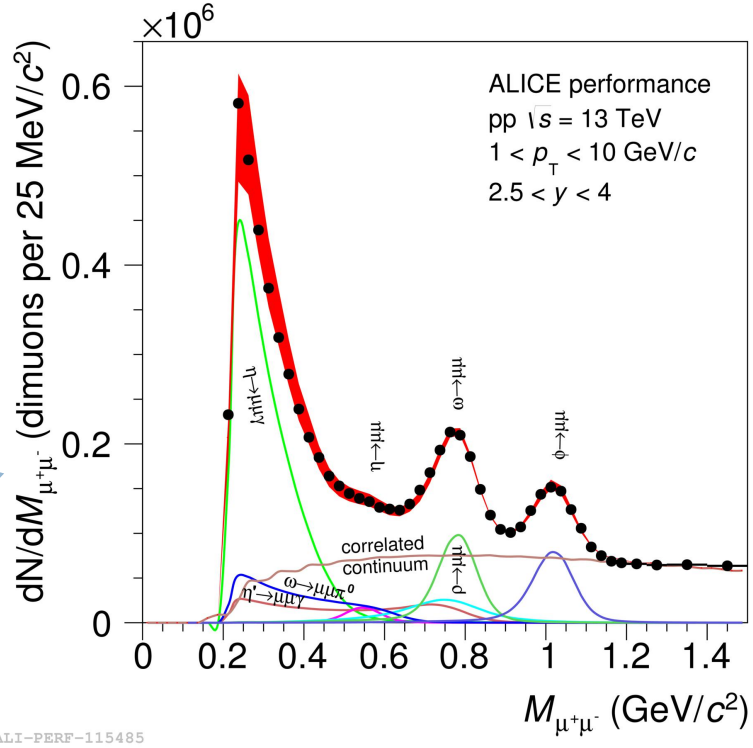
Experimental data should be corrected to particle level, or vice versa

Also important for **tuning** and **validation** of MC generators

Usage: corrections and unfolding

Some measurements need to be **corrected** for 'unknown' signals

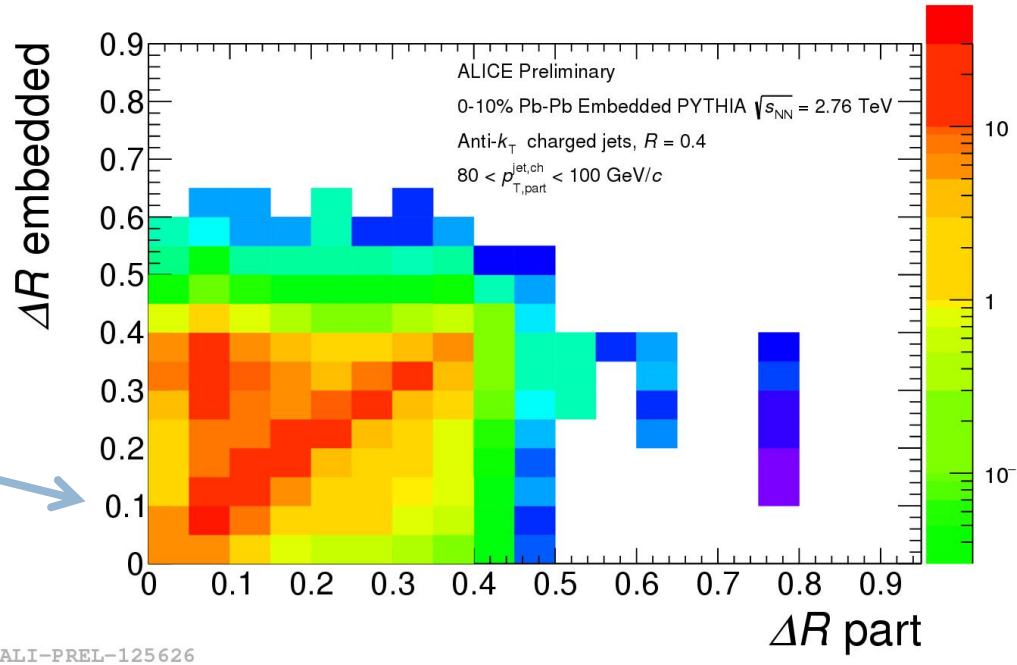
- MC generators necessary to quantify signals
- Prerequisite is **full confidence** in validity of generator output



Measurement can be biased by

- **Detector** resolution
- Processes **uncorrelated** to the measurement

Embedding MC processes into real data (or even MC) allows to quantify these effects



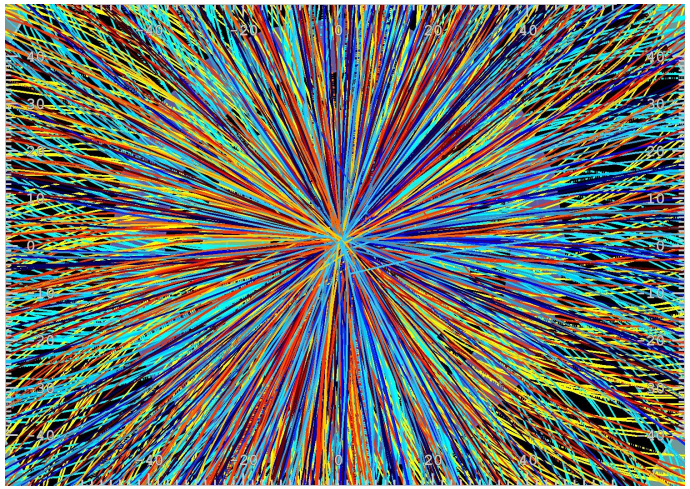
MC productions and integration in the ALICE software framework

ALICE main goal: heavy-ion physics

Heavy-ion MC generators generally **resource hungry**

- Multiple collision phases (time consuming)
  - Initial state
  - Medium evolution (*x, y, z, t*)
  - Rescattering and freeze-out (hadronization)

Large particle multiplicities (large data volumes)



Generator-only vs. generator+GEANT

- Simulation of detector effects often necessary but CPU intensive
- Per event GEANT CPU time order(s) of magnitude larger than generator-only CPU time

In practice

- **Large-scale** productions have to be **approved** by the collaboration
- Parallel production of events on **GRID**, mandatory validation
- Productions are **kept** on disk **until obsolete** (limited on-the-fly analysis)

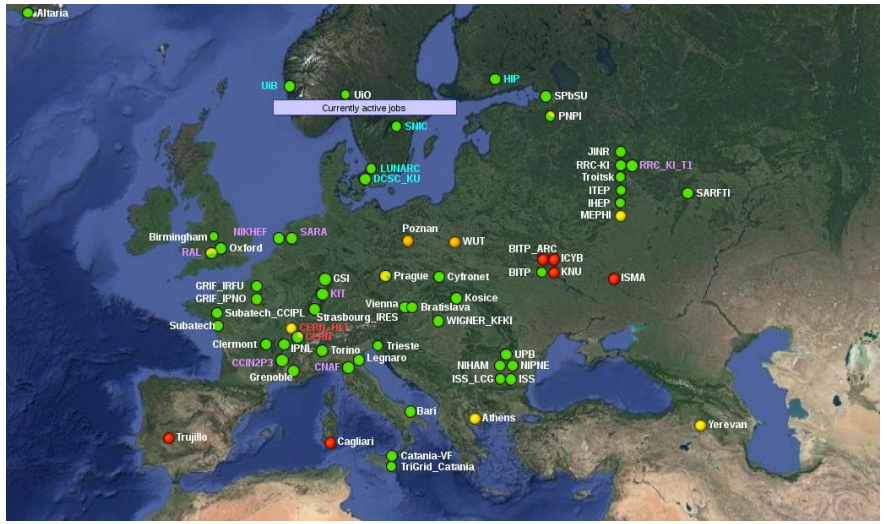
MC production on GRID

Productions are proposed by **P**hysics **W**orking **G**roups

- Request is motivated and quantified (how many events are needed? what will we learn? how many analyses could benefit?)
- Estimates of CPU years & required disk space are made

'**Generator level**' output, as well as '**detector level**' output (propagation through GEANT, **anchored** to specific detector conditions) are generated

Upon approval, production runs
in parallel on GRID
(max ~150000 CPUs)
output is distributed
but available from all nodes



MC production on GRID

Output **stored** in standard ALICE AOD (**A**nalysis **O**bject **D**ata) format

- User (data) analysis can **run** on production **without modifications**
- Full **MC info** is also still available (and accessible in user analysis)

name and description

status

events

resources

used software

Production	Description		Status	Run range	Event Count	Requested	Comment	Known issues	Running time	Saving time	Output size	AliRoot version(s)	AliPhysics version(s)	
LHC18b10b_calo	p-p, 5.02 TeV - Pythia GJ events with gammas in DCal+PHOS anchored to 17p/q, calo triggers, ALIROOT-7692		Quality check 10%	282146-282342	844,600		RAW OCDB		2y 298d	9d 19:47	407.2 GB	v5-09-20-1	v5-09-20-01-1	
LHC18b10b_calofast	p-p, 5.02 TeV - Pythia GJ events with gammas in DCal+PHOS anchored to 17p/q, calofast triggers, ALIROOT-7692		Quality check 10%	282306-282342			B		4y 163d	15d 22:52	636.9 GB	v5-09-20-1	v5-09-20-01-1	
LHC18b8_fast	p-p, 5.02 TeV - Pythia8+JetJet production anchored to LHC17p/q, fast, ALIROOT-7689		Running	282341-282343			B		49y 166d	76d 17:11	5.303 TB	v5-09-20-1	v5-09-20-01-1	
LHC18b8_cent_woSDD	p-p, 5.02 TeV - Pythia8+JetJet production anchored to LHC17p/q, cent_woSDD, ALIROOT-7689		Running	282341-282343	8,183,200		RAW OCDB		27y 168d	40d 22:48	2.934 TB	v5-09-20-1	v5-09-20-01-1	
LHC17h7a_extra	p-p, 13 TeV, Pythia6_Perugia2011 production anchored to LHC17j, extra statistics, ALIROOT-7432		Completed	274593-274595	54,628,400		RAW OCDB		160y 30d	235d 0:23	21.28 TB	v5-09-14-1	v5-09-14-01-1	
LHC18b11c	Pb-Pb, 5.02 TeV - jet-jet Pythia events embedded in HI MC events anchored to LHC15o, cent 50-90 %, ALIROOT-7693		Completed	244918-246994	241,160		RAW OCDB				8			
LHC18b11b	Pb-Pb, 5.02 TeV - jet-jet Pythia events embedded in HI MC events anchored to LHC15o, cent 10-50 %, ALIROOT-7693		Completed	244918-246994	238,710		RAW OCDB			52d 5:22	5.606 TE	-13q-p5-1	v5-08-13q-p5-1	
					6,113,138,864				28518y 249d	312y 223d	6.183 PB			

'anchor'

total CPU time

disk usage

'On-the-fly' productions

MC generators can be selected as 'data sets' in the analysis framework

- **User** sets generator **configuration** and desired **number of events via** web interface
- **Generator** runs in **parallel** with analysis and sends output to a named pipe (fifo)
- Analysis framework converts fifo contents to AODs **on-the-fly**

Allows for efficient analysis of MC events with little IO

- Only suited for **fast** generators

Name	Owner	Dependencies	AMPT_ v22...	DIPSY _v2...	DPMJE T	EPOS_ 3_d...	EPOSL HC_...	EPOSL HC_...	Herwi g	JEWEL	Pythi a_6...	Pythi a_8...	Pythi a_8...	Therm ina...	Last test	Last run	
Group Default			Enable Disable	Enable Disable	Enable Disable	Enable Disable	Enable Disable	Enable Disable	Enable Disable	Enable Di...ble	Enable Disable	Enable Disable	Enable Disable	Enable Disable	59	46	
FlowOnMC	rbertens														46	46	
FlowOnMC_kaon	rbertens														46	46	
FlowOnMC_pion	rbertens														46	46	
FlowOnMC_proton	rbertens														46	46	

MC generators

data analysis classes

Integration with ALICE software

Initial approach

- **Include** generator code as part of ALICE framework, load shared object at framework run-time
- Generation and AOD production handled by dedicated **wrapper class** for each generator

Pro: full **integration and control** over generator

Con: not **flexible** (e.g. updating, RIVET use)

```
class TAmpt : public TGenerator {
public:

  ...

  virtual void        GenerateEvent();
  virtual Int_t       ImportParticles(TClonesArray *particles, Option_t *option="");

  // common HIJJET1 access routines - read-only common:
  virtual Int_t       GetNPJ(Int_t key) const;
  virtual Int_t       GetKFPJ(Int_t key1, Int_t key2) const;
  virtual Float_t     GetPJPX(Int_t key1, Int_t key2) const;
```

example: AMPT wrapper class in AliROOT

includes e.g. functions to generate event and import particles to analysis format

functions in turn access fortran common blocks

Integration with ALICE software

Approach for **new** generators / updates overseen by **dedicated new working** group (PWGMM)

- Distribute generators as **stand-alone** executables (on GRID via CernVM File System CVMFS)
  - **Full integration** of generators in ALICE **build system**
  - Requires that generators can be **properly built** (CMake, makefiles, etc)
- Run stand-alone generator as **background process** during production/analysis
- Generator writes to **fifo**, analysis framework reads from fifo

HepMC is **fully integated** with the ALICE software framework

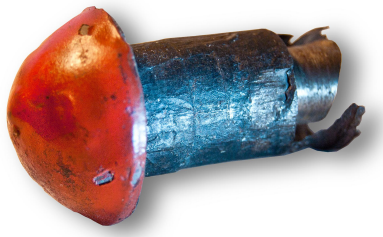
- Non-HepMC generator output is either **parsed** to HepMC or handled by a dedicated 'MC-to-AOD' parser

Caveats of HepMC

- Mandatory **graph structure** incompatible with some HI generators
  - e.g. only producing particles at freeze-out (no **connection** to beam particles)

MC generators and RIVET

RIVET and ALICE



ALICE 'RIVETizes' only a fraction of analyses

- p-Pb and Pb-Pb analyses need some features that RIVET misses

Using MC generators as **stand-alone** executables (slide 14) well suited for (and aimed at) RIVET and MCPlots project

- **Same** generator suite can be used for **RIVET** analysis and **ALICE MC analysis**
- Ensures well-defined **generator configuration** for RIVET and ALICE MC analyses
- Ability to run generators with a **well-defined configuration**, be it **in** or **out** of the ALICE analysis framework, is generally **crucial**

Goal

- **Encourage** RIVETization of analyses
- Actively contribute to **tuning** of generators using RIVET plugins

~ *fin* ~

questions & discussion