Heavy Ion Jet INteraction Generator

C++ version

XNWLX

18 08 2022

Xin-Nian Wang
Miklós Gyulassy
Ben-Wei Zhang

Gergely Gábor Barnaföldi
Gábor Bíró
Balázs Majoros
Gábor Papp
Péter Lévai

Former collaborators:
Szilveszter Harangozó
Bálint Csurgai
Guoyang Ma

arXiv:1905.11272
arXiv:1901.04220
arXiv:1811.02131
arXiv:1805.02635
arXiv:1707.09973
arXiv:1701.08496

”Modern day HEP requires high performance computing, relying on Monte Carlo simulations”

— Alberto Di Meglio

The CERN Quantum Technology Initiative, 20.10.2020.

1996, Budapest, Workshop on Strangeness in Hadronic Matter:

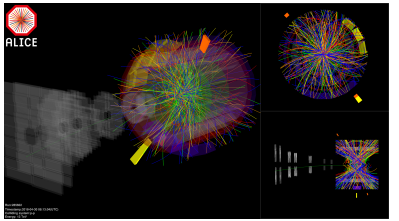
MC event generators

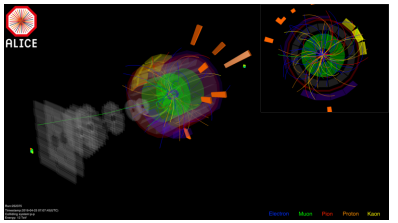
Simulation of one proton-proton collision event: complicated...

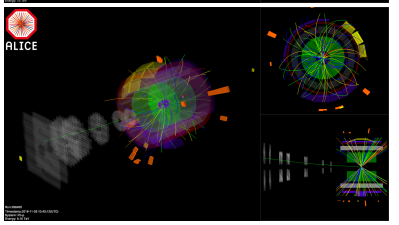
1) Perturbative QCD calculations

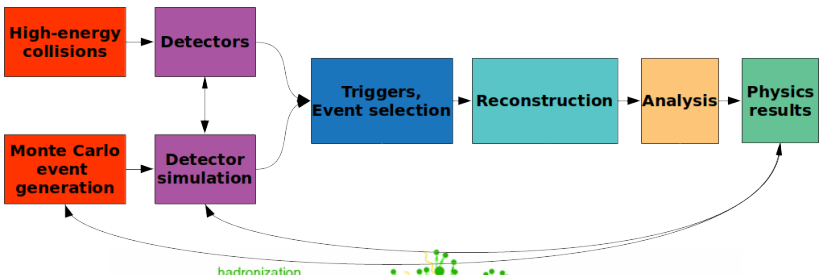
$$\frac{d^2\sigma^{lP\to hX}}{dxdQ^2} = \sum_{i=q,\bar{q},g} \int_x^1 \frac{dz}{z} f_i(z,\mu) d\hat{\sigma}_{il\to iX} \left(\frac{x}{z}, \frac{Q}{\mu}\right) D_i^h(z)$$

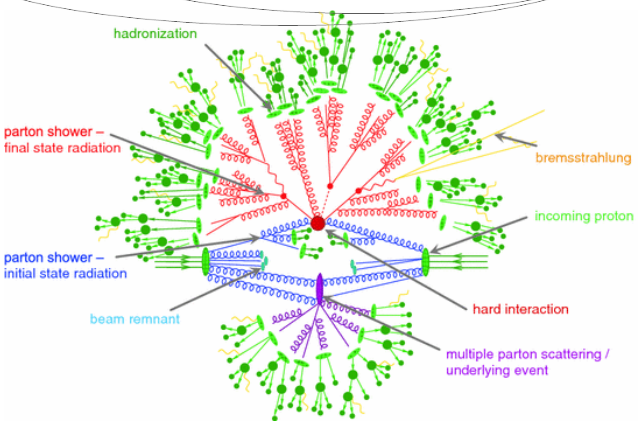
2) Additional phenomenological processes: MPI, colour reconnection, hadronization scheme...
3) Compromise: computational time ←→ precision
   - Tons of random numbers
4) Empirical parameters: need to be tuned



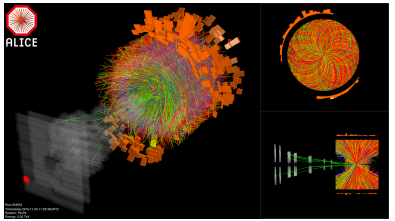


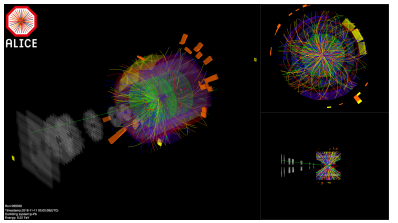


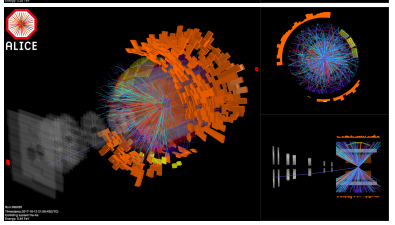




MC event generators





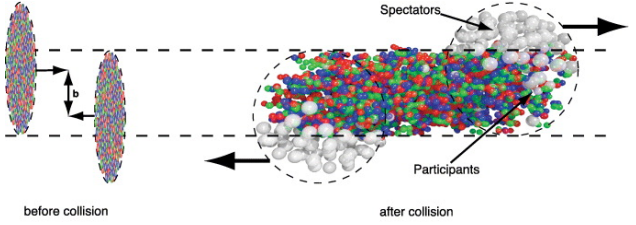


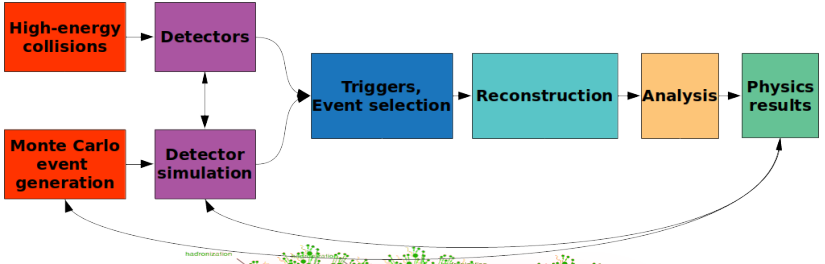
Simulation of one **heavy-ion** collision event: **even more** complicated...

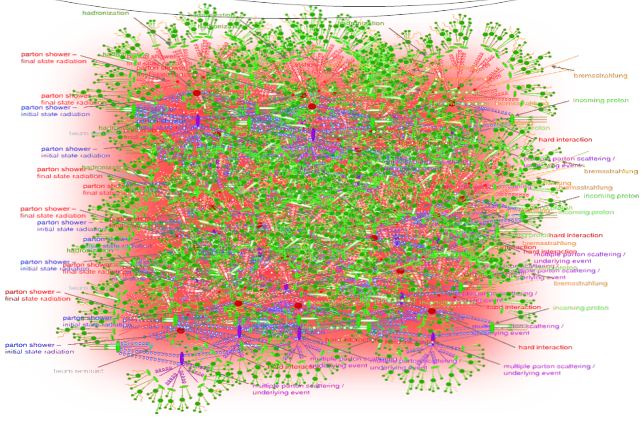
1) Perturbative QCD calculations

$$\frac{d^2\sigma^{lP\to hX}}{dxdQ^2} = \sum_{i=q,\bar{q},g} \int_x^1 \frac{dz}{z} f_i(z,\mu) d\hat{\sigma}_{il\to iX}\left(\frac{x}{z},\frac{Q}{\mu}\right) D_i^h(z)$$

2) Additional phenomenological processes: MPI, colour reconnection, hadronization scheme...
3) Compromise: computational time ←→ precision
   - **Tons** of random numbers
4) Empirical parameters: need to be tuned
5) Multiple nucleon-nucleon interactions
6) Additional nuclear effects: jet quenching, Cronin enhancement,
7) shadowing..







A NEW GENERATION OF HEAVY-ION MONTE CARLO

Heavy
Ion
Jet
INteraction
Generator
(C++ version)

核易经

[Hé –yì –jīng]

”Nuclear change theory”; Book of Changes, ”Originally a divination manual in the Western Zhou period (1000–750 BC)”

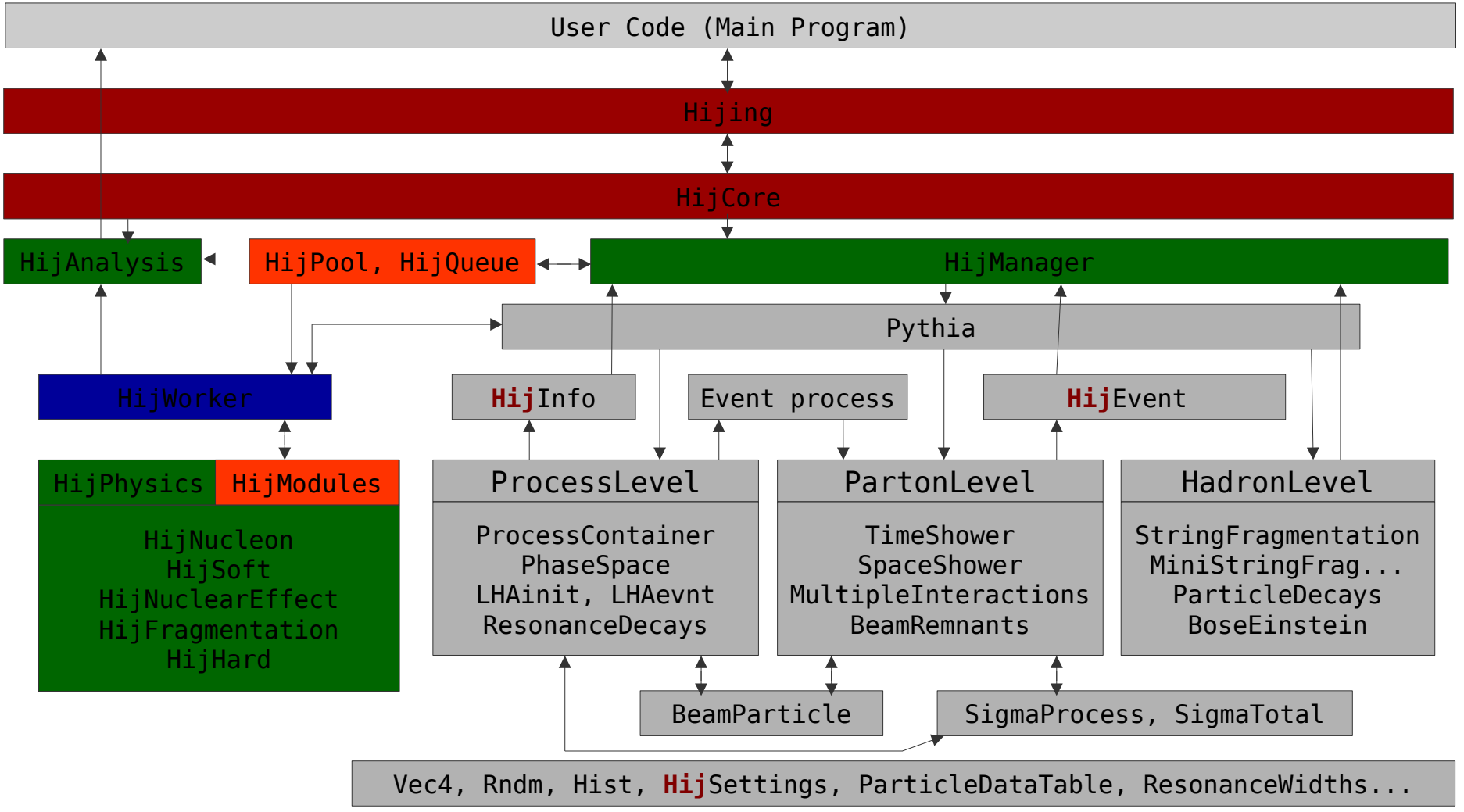
First, FORTRAN version: 1991, X.N. Wang, M. Gyulassy, **Phys. Rev. D 44, (1991) 3501.**

Computational challenge: more than 600 million collision in each second → HiLumiLHC: even more

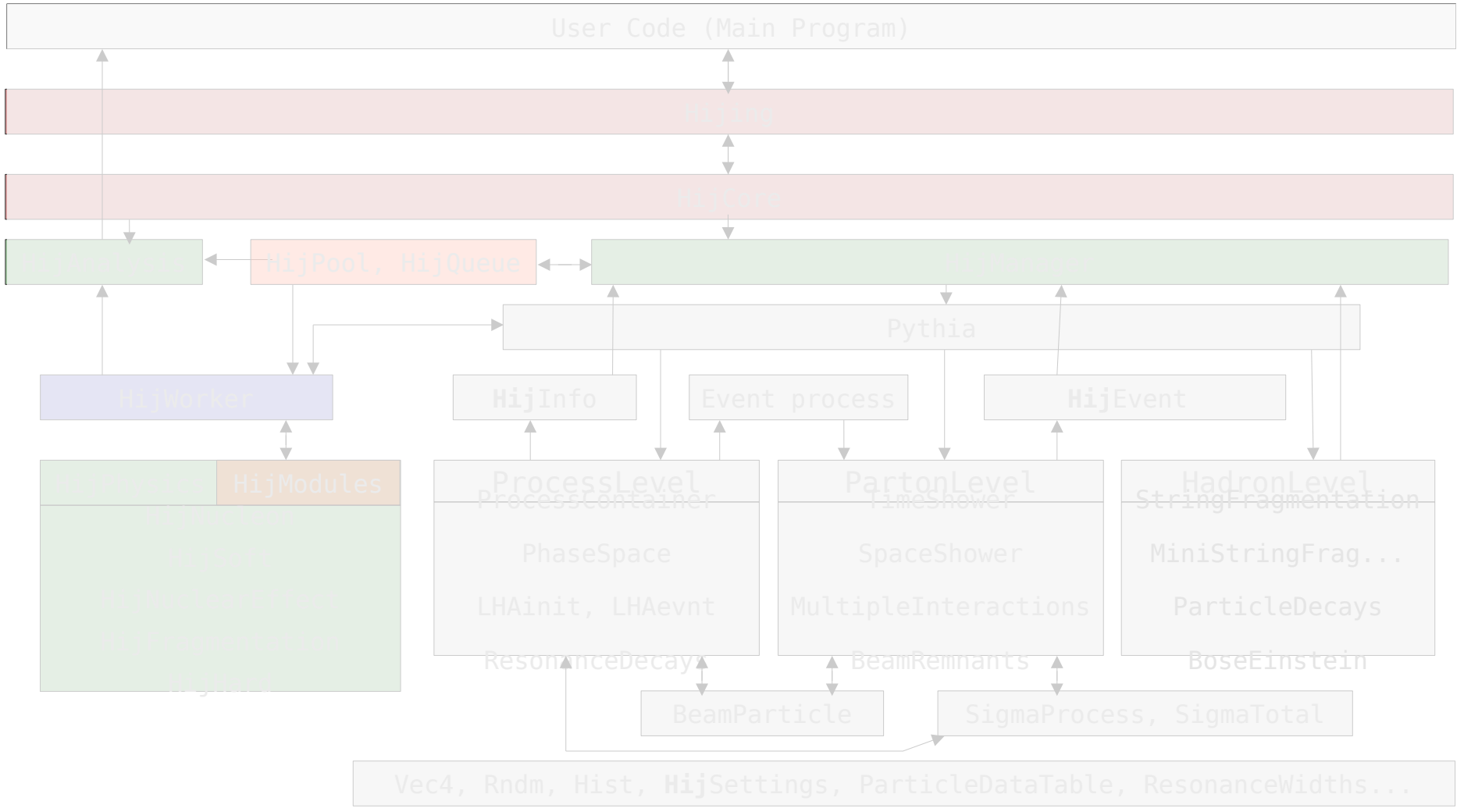
Requirements for a new version: multithreaded mode, maintainability, intuitive usage

	FORTRAN HIJING	HIJING++ v3.0	HIJING++ v3.1
Precision	simple	double	double
Pythia version	5.3	8.2	8.2+
(n)PDF	GRV98lo	LHAPDF6.2	LHAPDF6.2+
Jet quenching	(✓)	(✓)	(✓)
Multithreading	✗	✗	✓
Analysis interface	✗	✗	✓
Module management	✗	✗	✓
Dependencies, build system	Makefile	Makefile	CMake

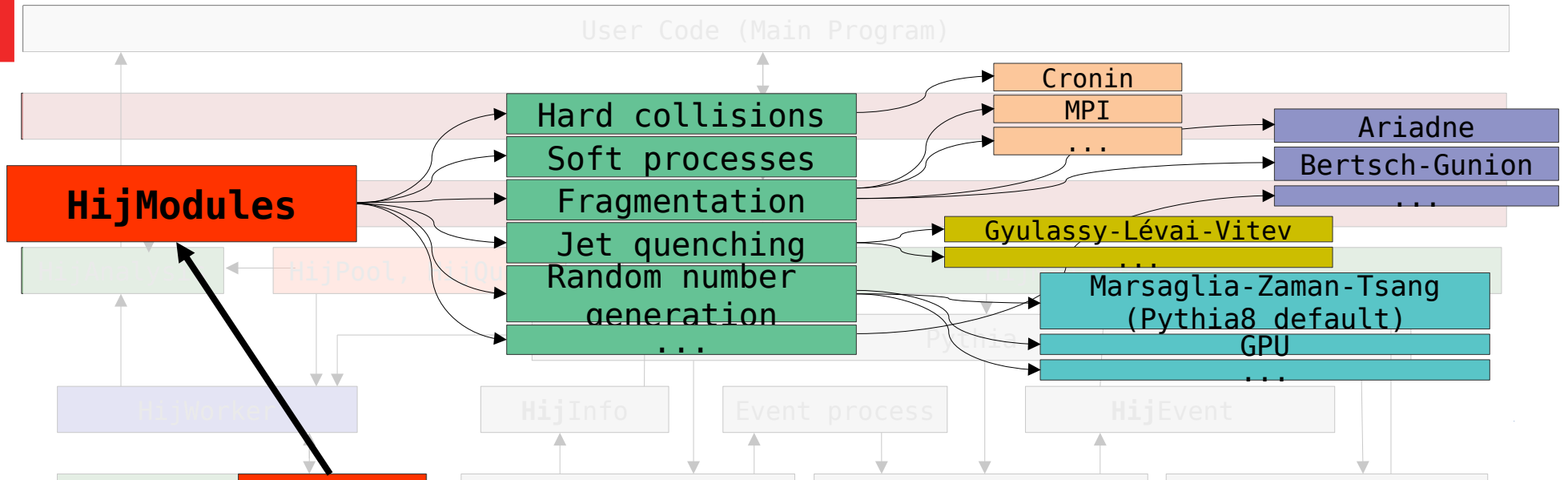
HIJING++ structure



HIJING++ structure

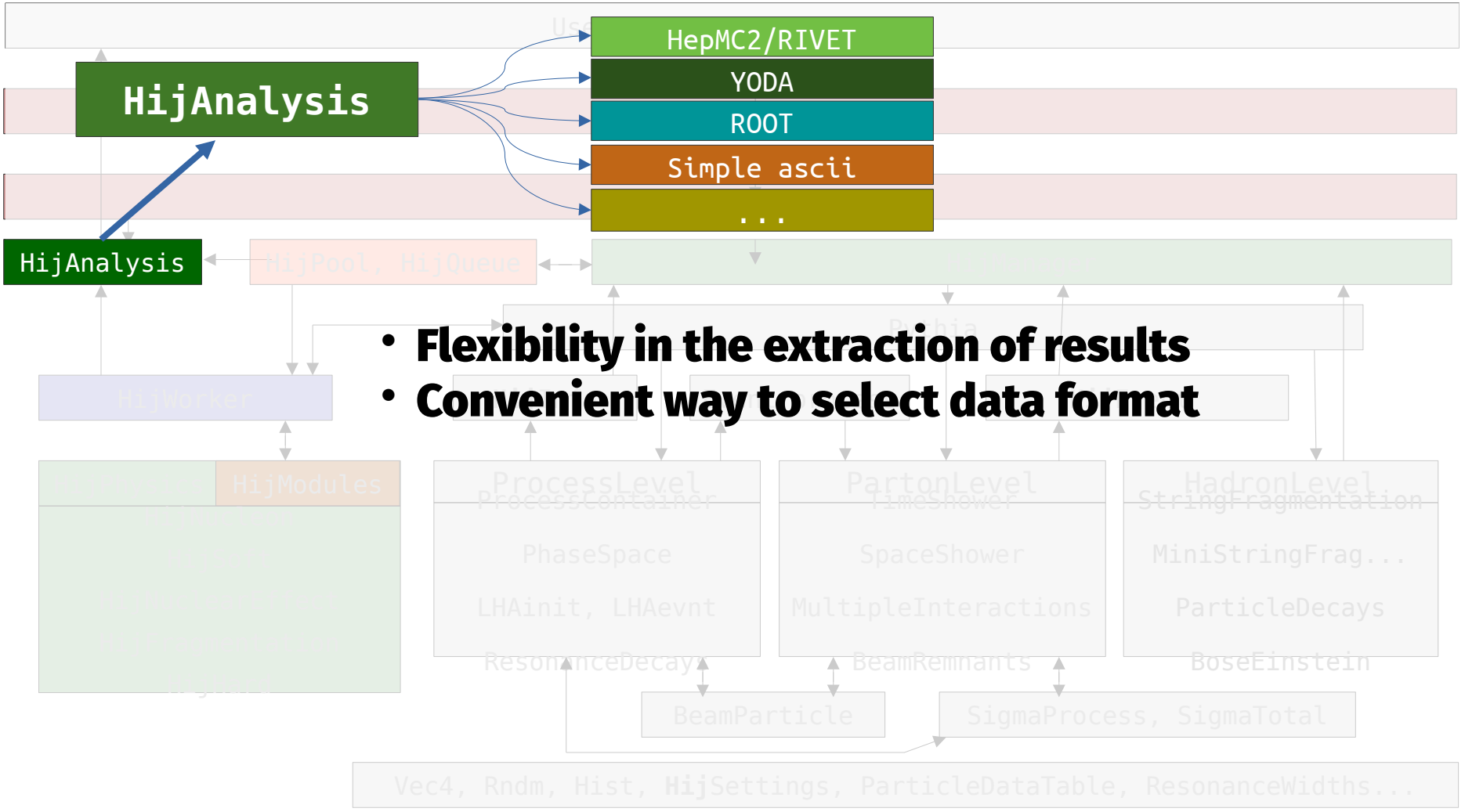


HIJING++ structure



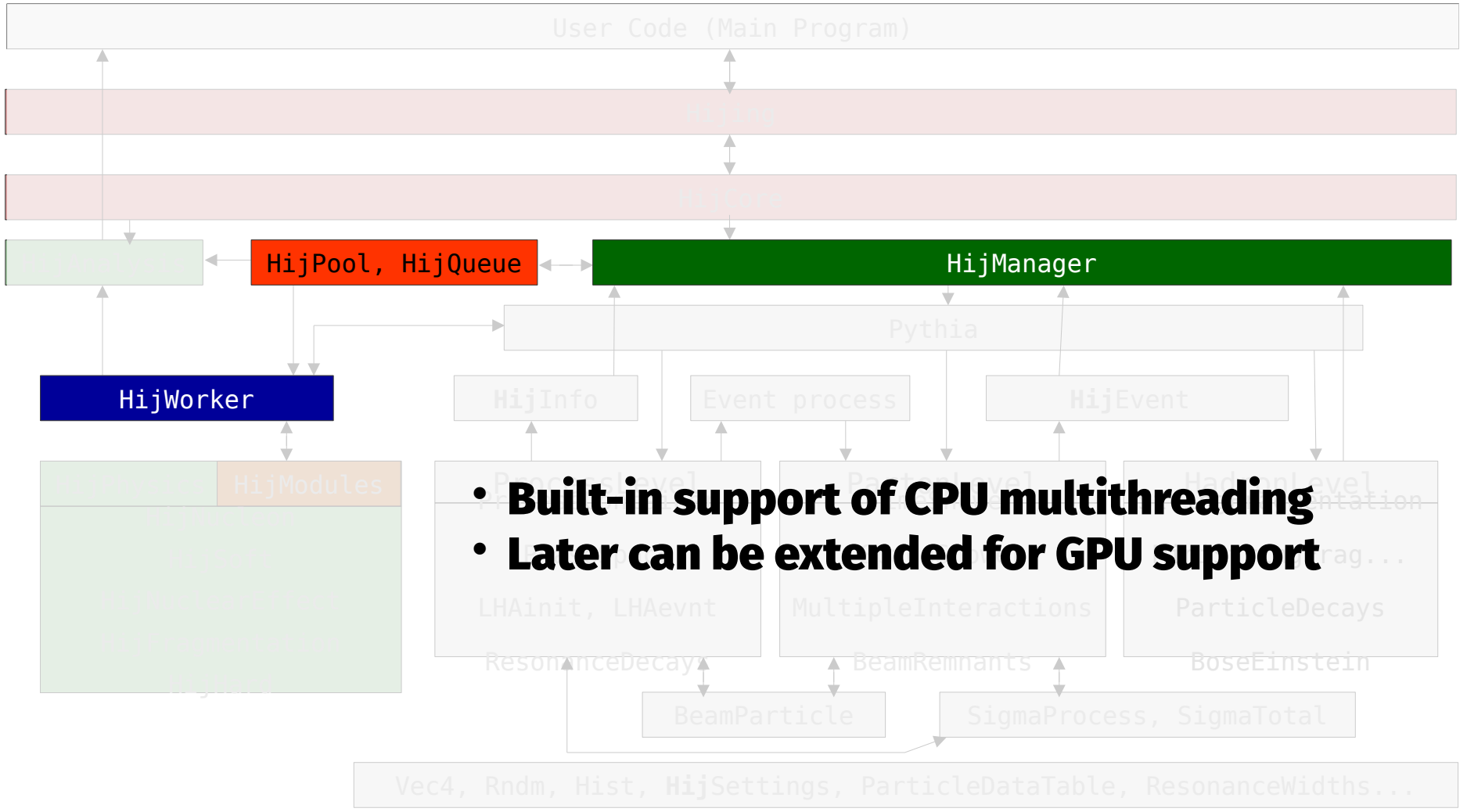
- **Possibility to add custom modules (e.g. hydro afterburner or coalescence hadronization)**
- **Given set of modules: scheme (can be used similarly as a tune)**

HIJING++ structure



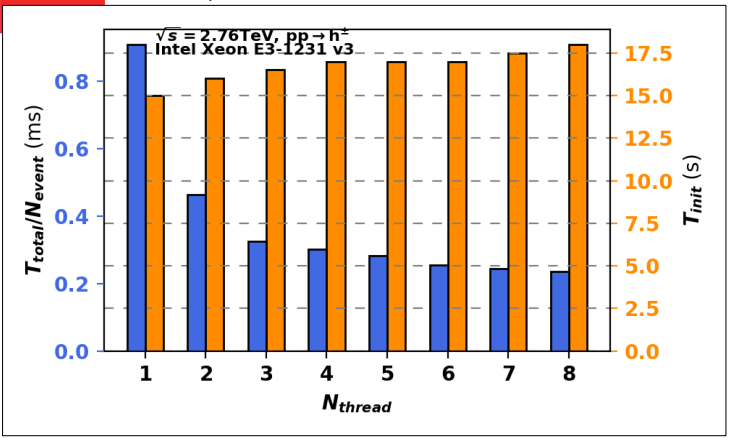
- **Flexibility in the extraction of results**
- **Convenient way to select data format**

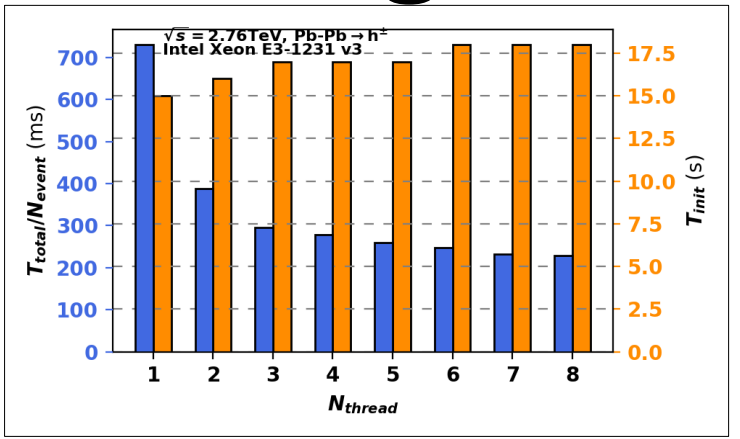
HIJING++ structure

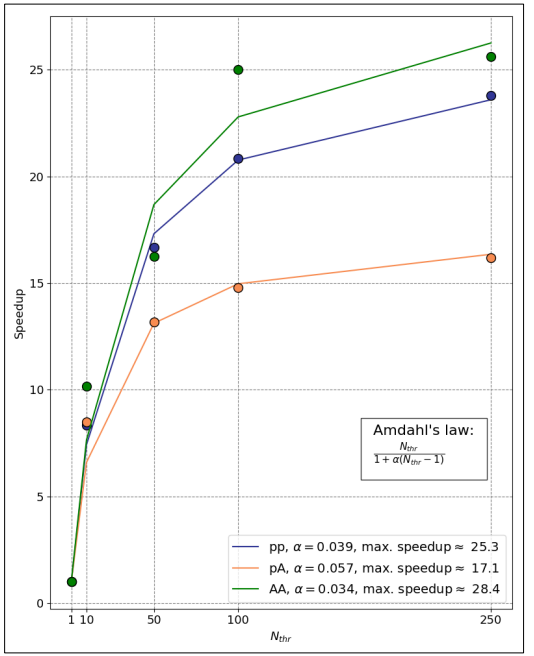


- **Built-in support of CPU multithreading**
- **Later can be extended for GPU support**

HIJING++ multithreading





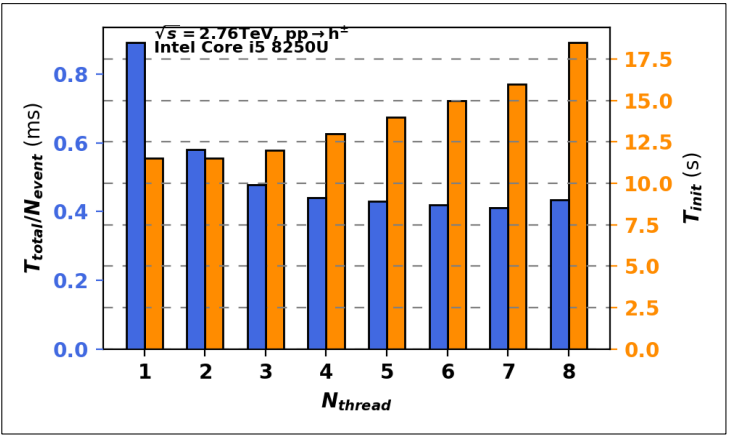


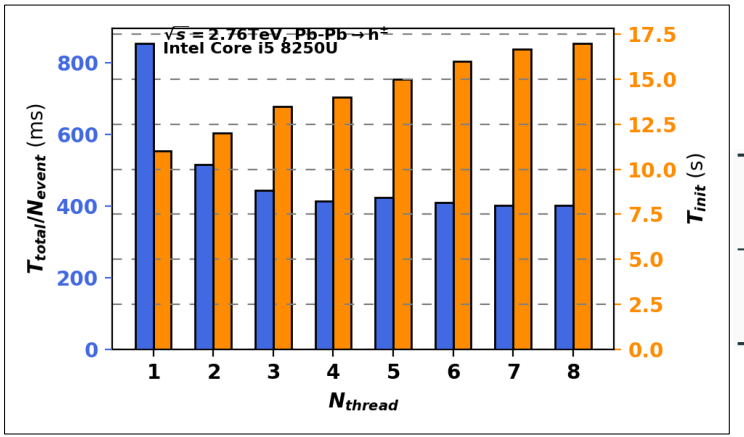
Data center: Intel Xeon E3 312 (4 core, 8 thread)

Laptop: Intel Core i5 8250U (4 core, 8 thread)

Proton-proton

Pb-Pb





CPU	Speedup		
	pp	p-Pb	Pb-Pb
Intel Core i5-8250U	2.6x	2.7x	2.6x
Intel Xeon E3-1231 v3	6.4x	6.6x	4.5x

Installation

Introduction

These are the setup instructions.

Prerequisites

- git
- cmake (min. v3.2)
- LHAPDF6 (v6.2.0 or newer)
- Pythia8 (v8219 or newer)
- c++ compiler with c++14 support (gcc 5 or later)

LHAPDF6

```
wget http://www.hepforge.org/archive/lhapdf/LHAPDF-6.X.Y.tar.gz
tar -xvf LHAPDF-6.X.Y.tar.gz
cd LHAPDF-6.X.Y
./configure --prefix=/where/to/install
make -jN
sudo make install
```

Install (nuclear) pdf sets

The pdf set *GRV98lo* is included in the downloaded package. It is mainly used during the development, since it is an unvalidated, "unofficial" set. However, if you wish

1. copy the *GRV98lo* folder (you can found it in `misc`) into `/path/to/install/LHAPDF6/share/LHAPDF`
2. insert into the file `pdfsets.index` at the correct line number (i.e. between 80000 and 80111) the following: *80060 GRV98lo 1*:

```
sed -i '/80000\ METAv10LHC\ 2/a 80060 GRV98lo 1' /path/to/install/LHAPDF6/share/LHAPDF/pdfsets.index
```

If you wish to use other npdf sets, visit http://lhapdf.hepforge.org/pdfsets.html and repeat the first step.

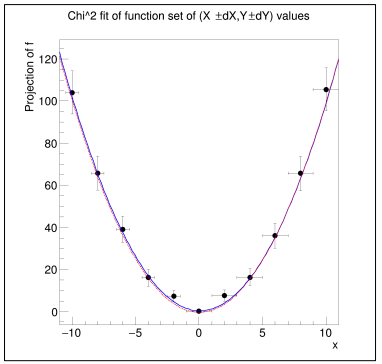
Pythia8

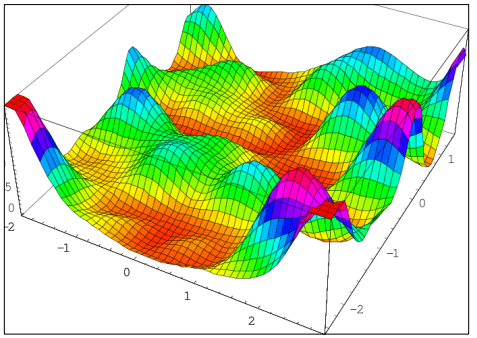
Download and install the latest version from the official webpage:

HIJING++ analysis interface

```cpp
#include "Hijing.hpp"

using namespace Hijing3;

int main(int argc, char* argv[])
{
	Hijing hijing;

	// collision energy, beams, #threads, event number...
	hijing.readFile("testSettings.cmnd");

	hijing.init();
	hijing.newAnalysis("root", "EventEnd", "histo_id1", 50, 0.0, 20.0);
	hijing.newAnalysis("ascii","EventEnd", "eta_charged_ascii", 20, -5.0, 5.0);
	// ...
	hijing.newAnalysis("yoda", "EventEnd", "ALICE_2010_I880049/d01-x01-y01", binnum_cent, edges_cent);
	hijing.newAnalysis("hepmc2", "ascii", "EventEnd", "output_file");

	hijing.analysisCustomCode(90001, [&](HijEvent &hijevent, pair<double, double> &val) {
		int cent = getMultClass(hijevent.b(), hijevent.Nbin(), hijevent.Npart());
		val.first = edges_cent[cent] + 0.1;
		double mult = 0;
		Event &event = hijevent(EventType::mainEvent);
		for (int iE = 0; iE < event.size(); iE++) {
			if (event[iE].isFinal() && abs(event[iE].y()) < 0.5 && event[iE].isCharged())
				mult++;
			else
				continue;
		}
		val.second = mult;});

	hijing.analysisProperties("histo_id1", "final", "pT", "yw-0.5to0.5", "ID211", "ID-211");
	hijing.analysisProperties("ALICE_2010_I880049/d01-x01-y01", "CC#90001", "nonorm");
	// ...
	hijing.start();
```

HIJING++ tuning (WIP)

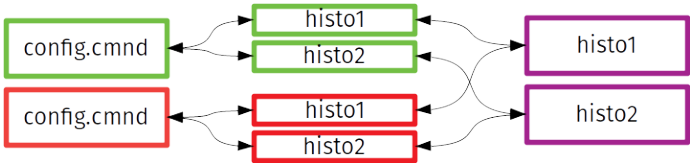


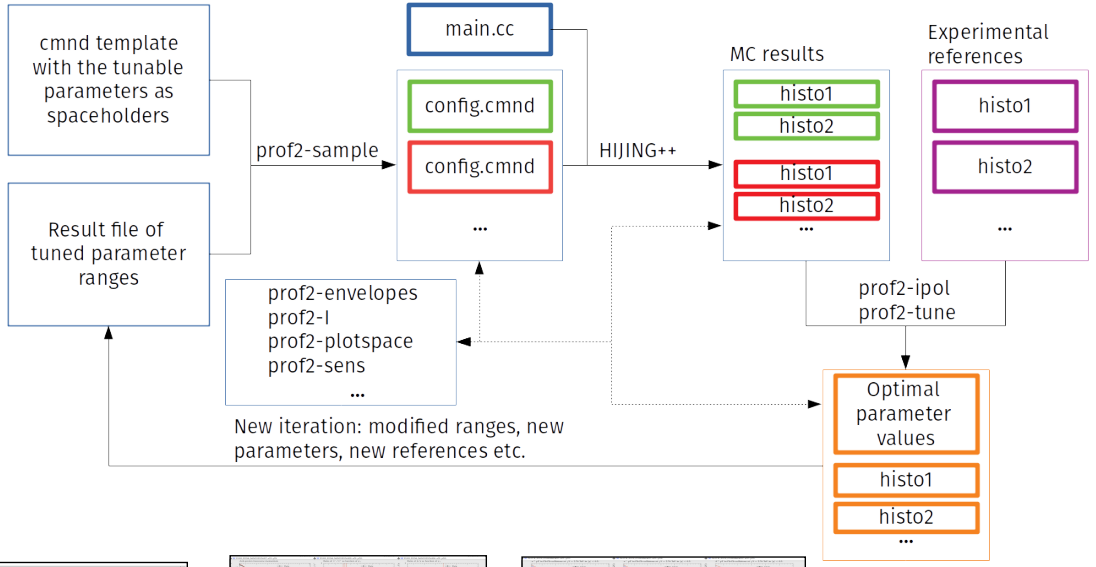


Tuning: set the empirical parameters to fit the experimental data → basically ~~„just"~~ an iterative χ^2 minimization → **very serious business**

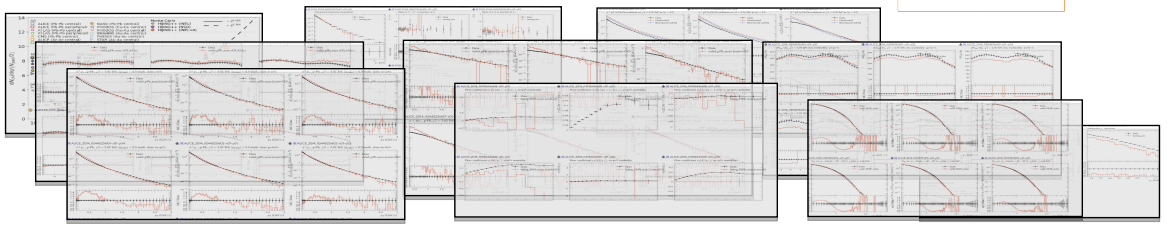
$$\chi^2 = \sum_i \left[\frac{y_i - f(x_i)}{\sigma_i} \right]^2$$

sample→*calculate*→minimize→repeat

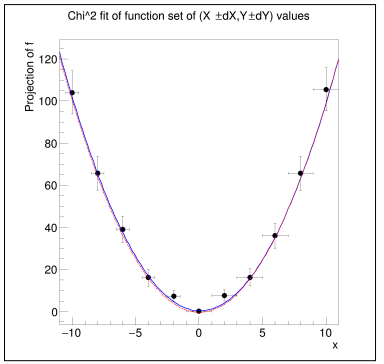


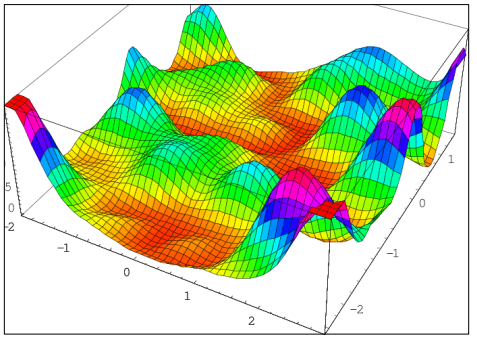


`YODA` (Yet more Objects for Data Analysis)
`Rivet` (Robust Independent Validation of Experiment and Theory)
`Professor` (Tuning tool for Monte Carlo event generators)
`MCNNUTNES` (A machine learning based optimization tool)



HIJING++ tuning (WIP)

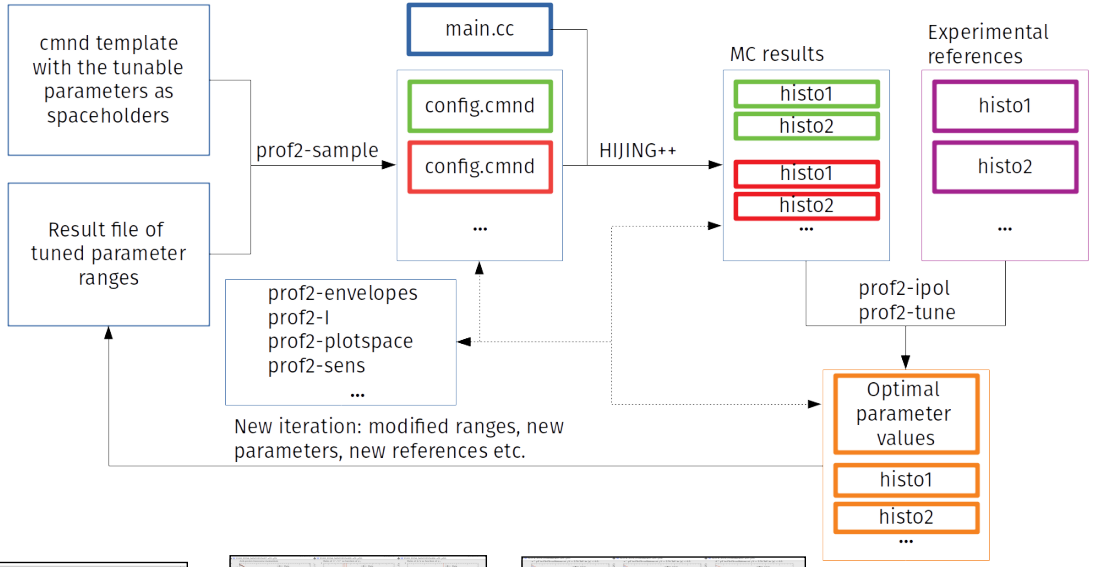




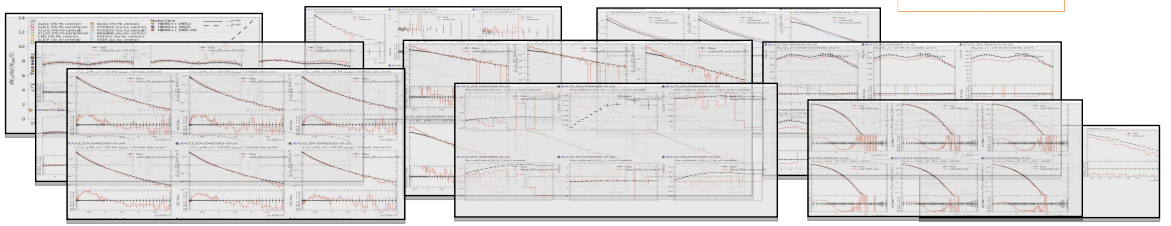
Tuning: set the empirical parameters to fit the experimental data → basically ~~„just"~~ an iterative χ^2 minimization → **very serious business**

$$\chi^2 = \sum_i \left[\frac{y_i - f(x_i)}{\sigma_i} \right]^2$$

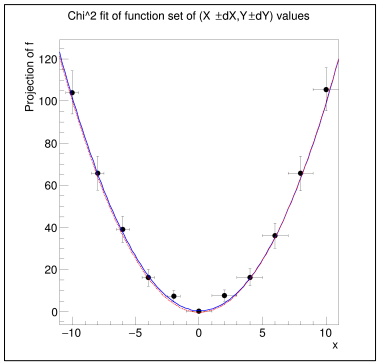
sample→*calculate*→minimize→repeat

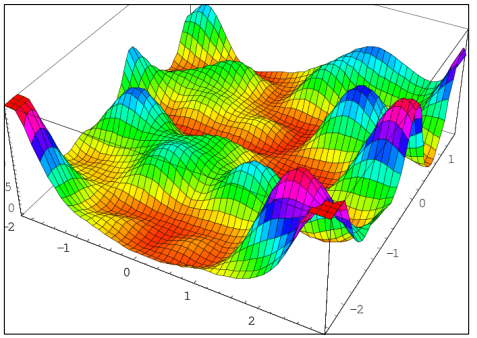


`YODA` (Yet more Objects for Data Analysis)
`Rivet` (Robust Independent Validation of Experiment and Theory)
`Professor` (Tuning tool for Monte Carlo event generators)
`MCNNUTNES` (A machine learning based optimization tool)



HIJING++ tuning (WIP)

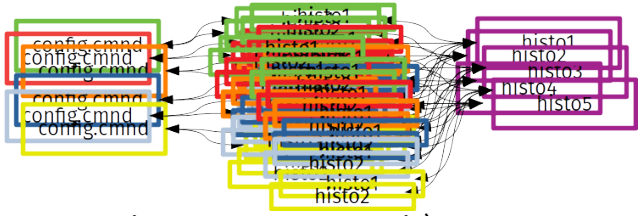




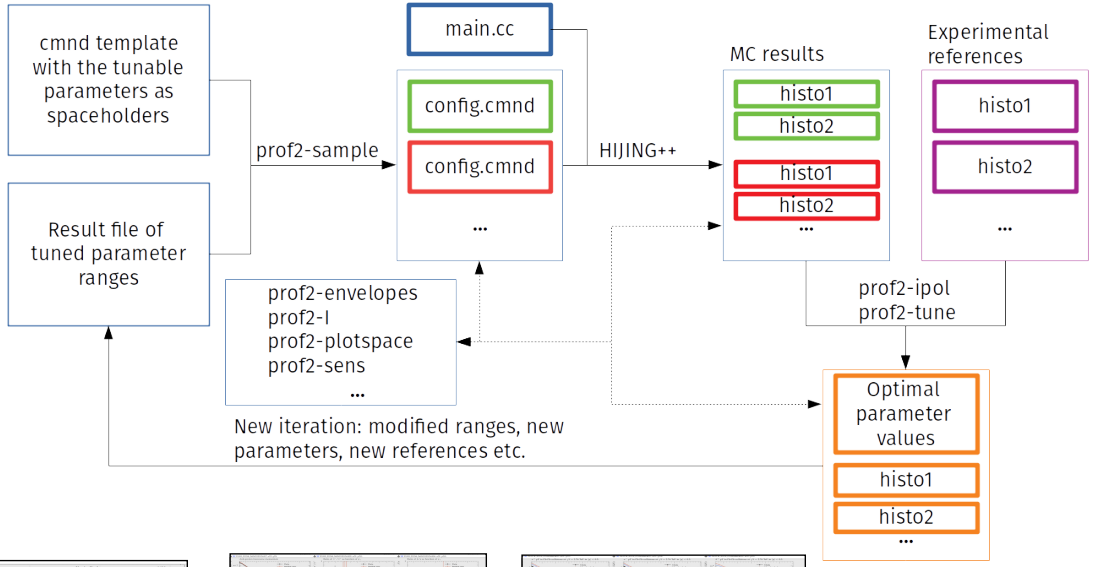
Tuning: set the empirical parameters to fit the experimental data → basically ~~„just"~~ an iterative χ^2 minimization → **very serious business**

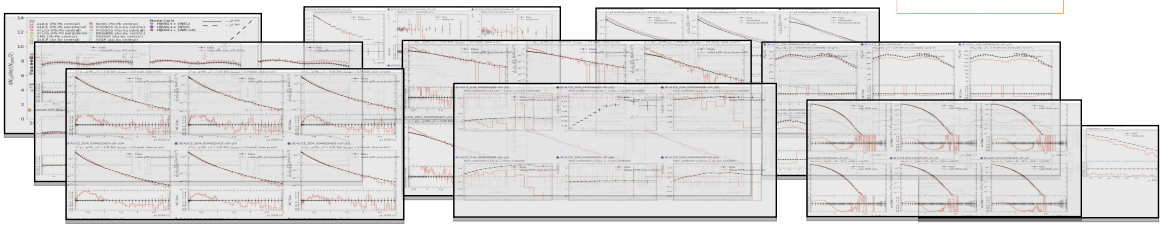
$$\chi^2 = \sum_i \left[\frac{y_i - f(x_i)}{\sigma_i} \right]^2$$

sample→*calculate*→minimize→repeat

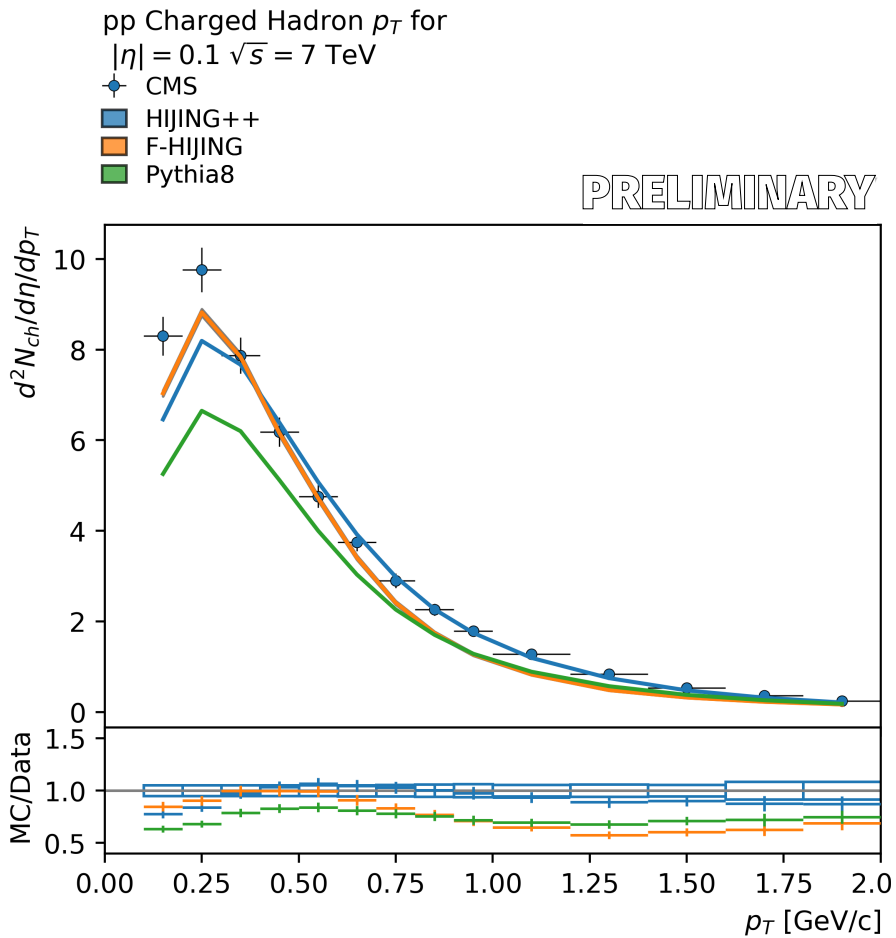


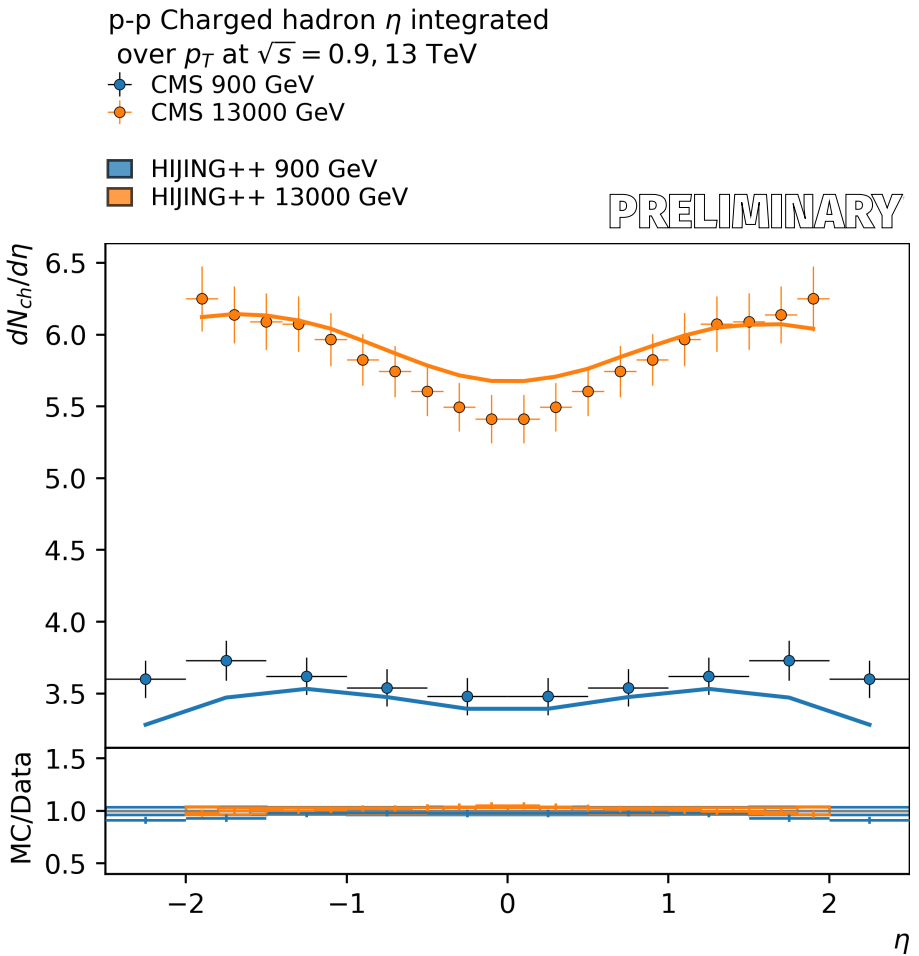
`YODA` (Yet more Objects for Data Analysis)
`Rivet` (Robust Independent Validation of Experiment and Theory)
`Professor` (Tuning tool for Monte Carlo event generators)
`MCNNUTNES` (A machine learning based optimization tool)



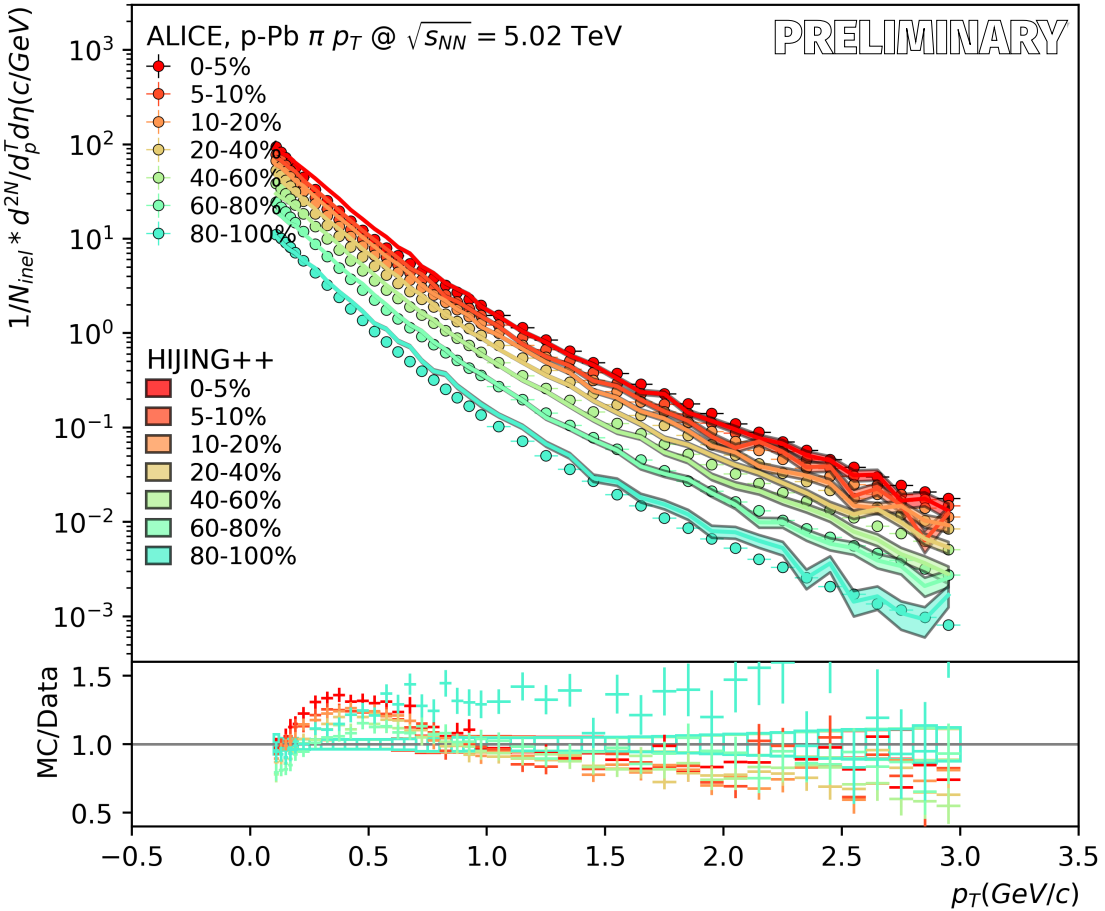


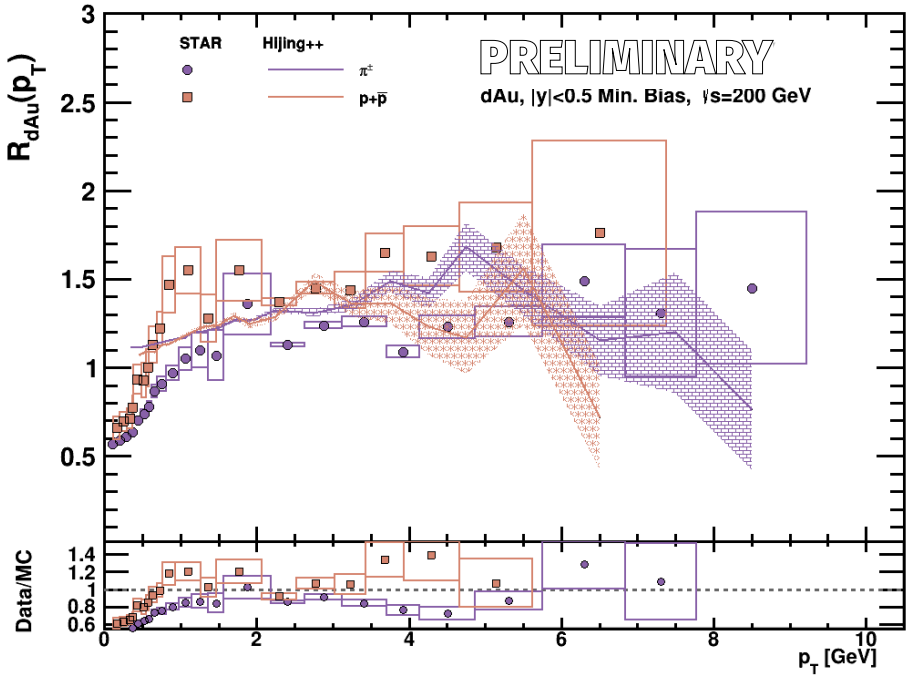
Results


pp Charged Hadron p_T for
$|\eta| = 0.1$ $\sqrt{s} = 7$ TeV
CMS
HIJING++
F-HIJING
Pythia8
PRELIMINARY
$d^2N_{ch}/d\eta/dp_T$
MC/Data
p_T [GeV/c]

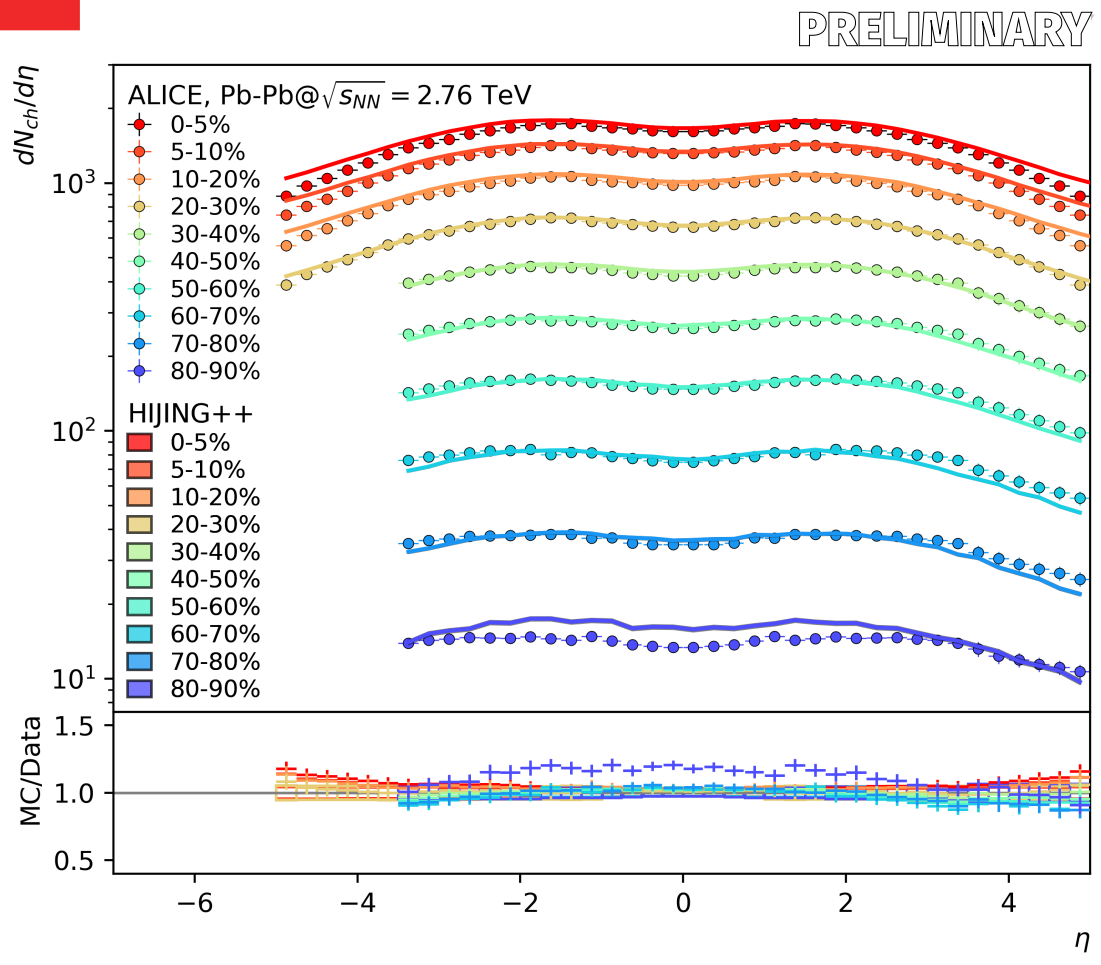

p-p Charged hadron η integrated
over p_T at $\sqrt{s} = 0.9, 13$ TeV
CMS 900 GeV
CMS 13000 GeV
HIJING++ 900 GeV
HIJING++ 13000 GeV
PRELIMINARY
$dN_{ch}/d\eta$
MC/Data
η

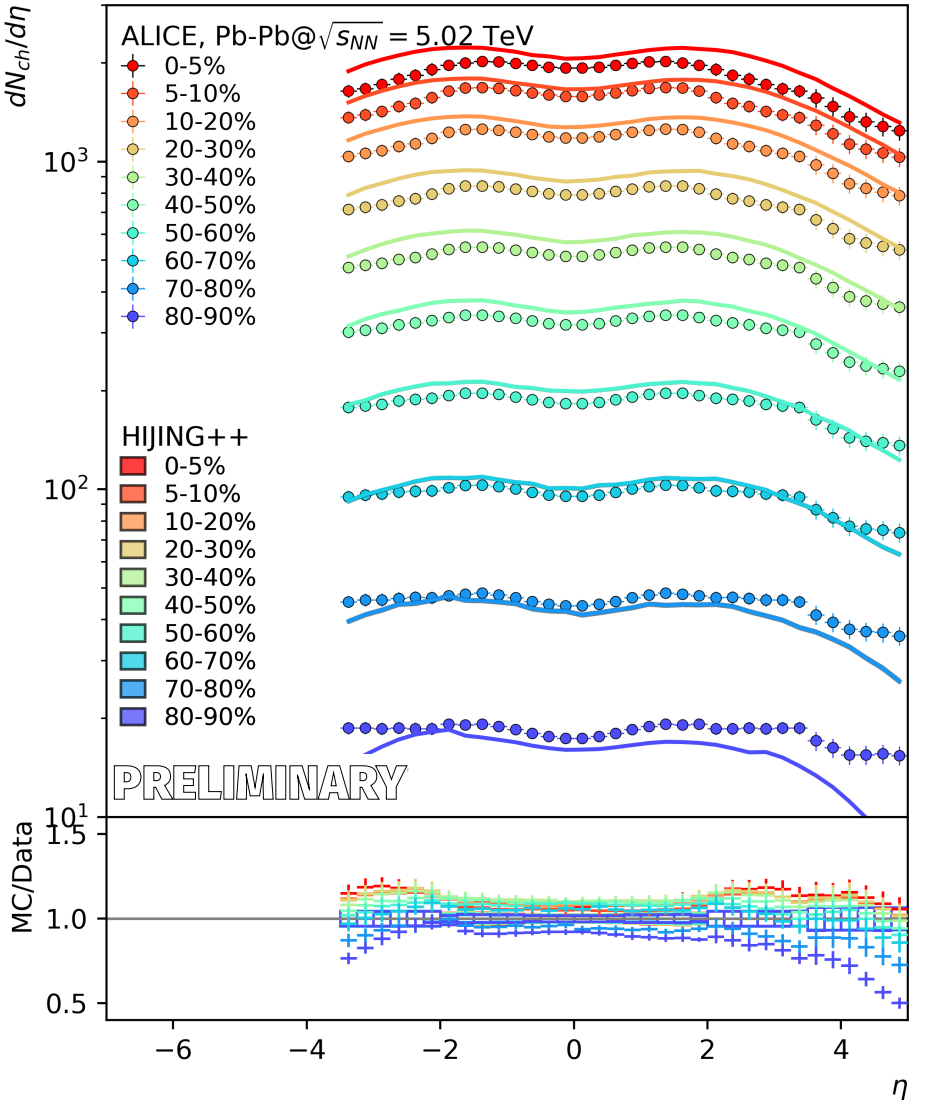
Results


ALICE, p-Pb π p_T @ $\sqrt{s_{NN}}$ = 5.02 TeV
PRELIMINARY
0-5%
5-10%
10-20%
20-40%
40-60%
60-80%
80-100%
HIJING++
0-5%
5-10%
10-20%
20-40%
40-60%
60-80%
80-100%
$1/N_{inel} * d^{2N}/d_p^T d\eta (c/GeV)$
MC/Data
$p_T (GeV/c)$


STAR
HIJING++
$\pi^{\pm}$
p+$\bar{p}$
PRELIMINARY
dAu, |y|<0.5 Min. Bias, $\sqrt{s}$=200 GeV
$R_{dAu}(p_T)$
Data/MC
p_T [GeV]

Results


PRELIMINARY
ALICE, Pb-Pb@$\sqrt{s_{NN}}$ = 2.76 TeV
0-5%
5-10%
10-20%
20-30%
30-40%
40-50%
50-60%
60-70%
70-80%
80-90%
HIJING++
0-5%
5-10%
10-20%
20-30%
30-40%
40-50%
50-60%
60-70%
70-80%
80-90%
$dN_{ch}/d\eta$
10^3
10^2
10^1
MC/Data
1.5
1.0
0.5
−6
−4
−2
0
2
4
η


ALICE, Pb-Pb@$\sqrt{s_{NN}}$ = 5.02 TeV
0-5%
5-10%
10-20%
20-30%
30-40%
40-50%
50-60%
60-70%
70-80%
80-90%
HIJING++
0-5%
5-10%
10-20%
20-30%
30-40%
40-50%
50-60%
60-70%
70-80%
80-90%
PRELIMINARY
$dN_{ch}/d\eta$
10^3
10^2
10^1
MC/Data
1.5
1.0
0.5
−6
−4
−2
0
2
4
η

Summary

Monte Carlo event generators **were/are/will be** crucial in high-energy physics

- Computationally very demanding (both to operate and to develop)
- **HIJING++**: the next generation of high-energy heavy-ion simulations
- Multithreaded, modular, intuitive
- Needs to be tuned → time consuming
- Room for future improvements → compatible with other popular frameworks, e.g. **JetScape**
- **Future:** support of Machine Learning-based modules

See related works: Wigner Scientific Computing Laboratory (The former Wigner GPU Laboratory): https://wigner.hu/en/wsclab

Cutting-edge technologies and infrastructure

Several partners both from academy and industry

Open opportunities for collaborations!