Pythia Status and Plans

Philip Ilten
on behalf of the Pythia authors

University of Birmingham

November 26, 2018

Physics Event Generator Computing Workshop

History

- 1978: JetSet from the Lund theory group
- 1997: merged into Fortran based Pythia 6
- 2004: rewrite into C++ began
- 2007: first release of C++ based Pythia 8.1
- 2014: mature Pythia 8.2 released
- today: version 8.235 released in March
- future: release imminent ...

Documentation

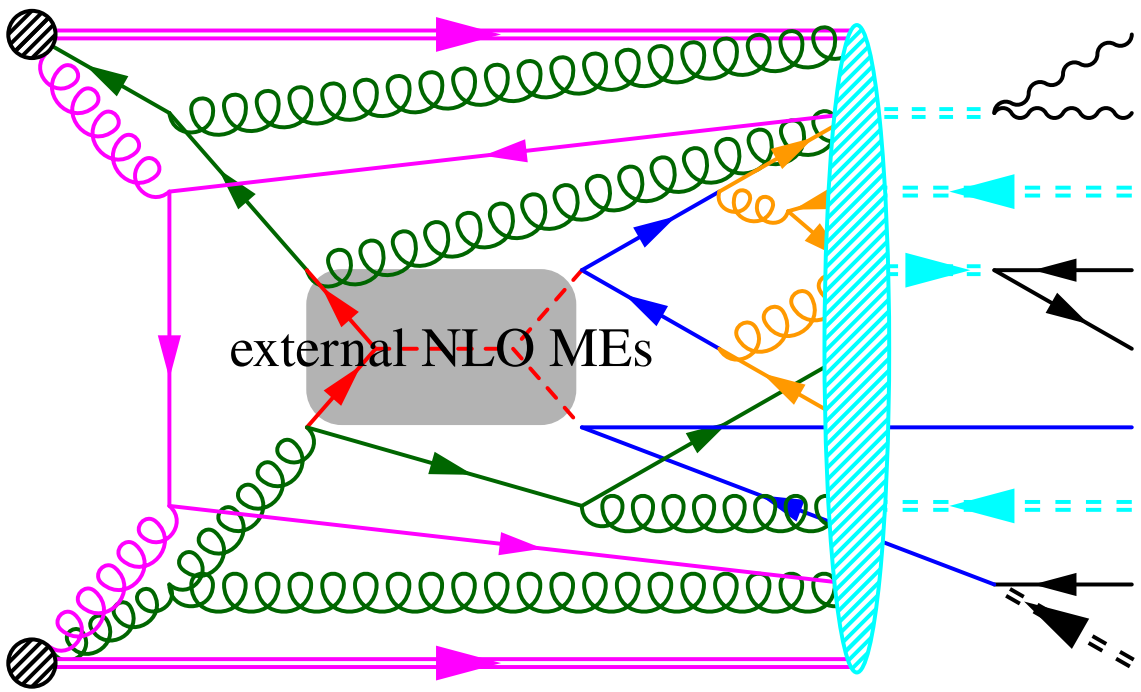
- *An Introduction to PYTHIA 8.2*
  Comput. Phys. Commun. **191**, 159 (2015)
- *PYTHIA 6.4 Physics and Manual*
  JHEP **0605**, 026 (2006)
- any of the original research cited from the manuals

- *Pythia 8 online manual* (also provided with source)
  `http://home.thep.lu.se/~torbjorn/pythia82html/Welcome.html`
- Doxygen documentation
  `http://home.thep.lu.se/~torbjorn/doxygen/annotated.html`
- Python documentation via `help`
- `mainXY.cc` and `mainXY.py` in `examples` directory

- email the author list or the authors directly
  `http://home.thep.lu.se/~torbjorn/pythiaaux/contact.html`

Philosophy

- **user-friendly and self-contained package**
- **designed to describe LHC events . . .**
  - does *not* provide automated matrix element (ME) generation
  - interfaces with large number of external ME providers
  - focus on everything *but* automatic ME generation



Core Interface

- both C++ and PYTHON interfaces
- PYTHIA is thread-safe except for adaptive maximum

```
// This configuration is not thread-safe.
pythia.readString("PhaseSpace:increaseMaximum = on")
```

- specialised constructor for multiple instances

```
// Initialize with XML database.
Pythia pythia0;
pythia0.init();
// Initialize with settings and particle data.
Pythia pythia1(pythia0.settings, pythia0.particleData);
Pythia pythia2(pythia0.settings, pythia0.particleData);
```

- settings passed by files, strings, or directly

```
pythia.readString(line)              // Read a line.
pythia.readFile(file)                // Read a file.
pythia.settings.flag(key, value)     // Set a flag.
pythia.settings.mode(key, value)     // Set a mode.
// Set parm, word, and vector versions of setting types.
```

- particle data configured via file, strings, or directly

```
pythia.particleData.m0(id, m0);
// Set any relevant particle data.
```

External Pointers (1)

- random number generator

```
// Possibility to pass in pointer for external random number generation→
    .
bool setRndmEnginePtr(rndmEnginePtr);
```

- hard process generation

```
// Possibility to pass in pointer to external LHA-interfaced generator.
bool setLHAupPtr(lhaUpPtr);

// Possibility to pass in pointer(s) for external cross section,
// with option to include external phase-space generator(s).
bool setSigmaPtr(sigmaPtr, phaseSpacePtr);
```

- particle and resonance decays

```
// Possibility to pass in pointer for external handling of some decays.
bool setDecayPtr(decayHandlePtr, handledParticles);

// Possibility to pass in pointer(s) for external resonance.
bool setResonancePtr(resonancePtr);
```

External Pointers (2)

- beams and PDFs

```
// Possibility to pass in pointers to PDF's.
bool setPDFPtr(pdfAPtr, pdfBPtr, ...);

// Set photon fluxes externally, "PDF:lepton2gammaSet = 2".
bool setPhotonFluxPtr(photonFluxA, photonFluxB);

// Possibility to pass in pointer for beam shape.
bool setBeamShapePtr(beamShapePtr);
```

- parton showers

```
// Possibility to pass in pointer for full merging class.
bool setMergingPtr(mergingPtr);

// Possibility to pass in pointer for merging hooks.
bool setMergingHooksPtr(mergingHooksPtr);

// Possibility to pass in pointer for external showers.
bool setShowerPtr(timesDecPtr, timesPtr, spacePtr);
```

- heavy ions

```
// Possibility to pass in pointer for heavy ion collisions.
bool setHeavyIonsPtr(heavyIonsPtr);

// Modify the behavior of the heavy ion modelling.
bool setHIHooks(hiHooksPtr);
```

External Pointers (3)

- user hooks

```
// Possibility to pass in pointer for user hooks.
bool setUserHooksPtr(UserHooks* userHooksPtrIn);
bool addUserHooksPtr(UserHooks* userHooksPtrIn);
```

1. access event between process → parton and parton → hadron
2. interrupt downwards evolution at a given scale
3. veto event after ISR/FSR or MPI emissions
4. veto ISR or FSR emission
5. modify trial hard process
6. veto resonance decay sequence
7. set scale of shower evolution
8. perform colour reconnection
9. enhance shower splittings
10. veto individual hadrons

- space-time vertices

```
// Possibility to pass in pointer for setting of parton space-time →
    vertices.
bool setPartonVertexPtr( PartonVertex* partonVertexPtrIn);
```

External Interfaces

- HepMC 2 and HepMC 3 interfaces for writing events
- read in AlpGen, LHEF, and SLHA files
- interface to PowhegBox (`main33`)

```
PowhegProcs procs(&pythia, "hvq");
```

- interface to MadGraph (`main34`)

```
LHAupMadgraph madgraph(&pythia, true, "madgraphrun", exe);
madgraph.readString("generate p p > mu+ mu-");
```

- interface to HelacOnia (`main35`)

```
LHAupHelaconia helaconia(&pythia, "helaconiarun", exe);
helaconia.readString("generate g g > cc~(3S11) g");
```

- interface to EvtGen with signal weighting (`main48`)

```
EvtGenDecays evtgen(&pythia, dec, pdl);
```

Development

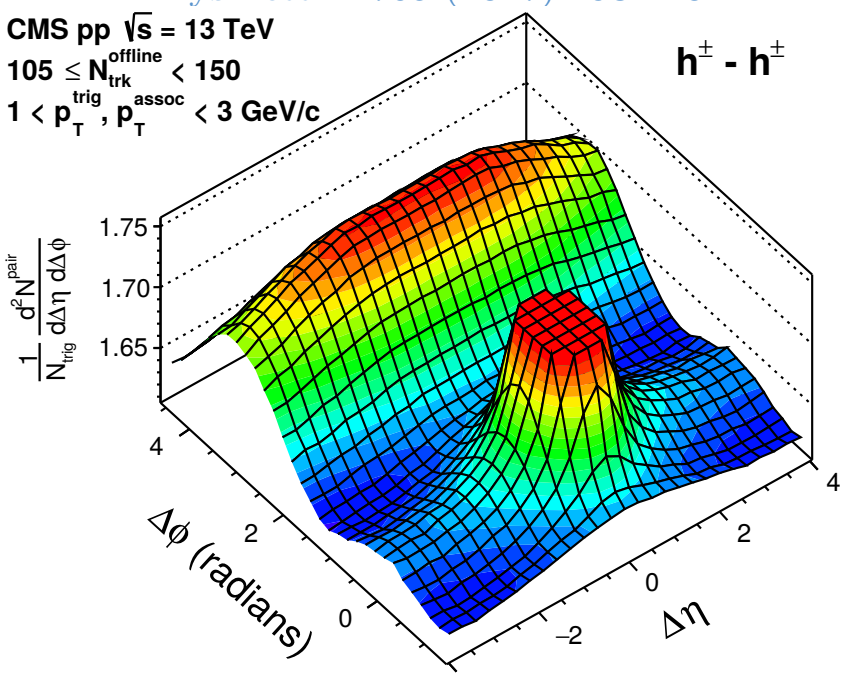
New and Upcoming

- Angantyr heavy-ion collisions model
  - introduction of collective effects, including for proton-proton
  - alternative models for string fragmentation
- full integration of Dire and Vincia parton showers
- combination of shower variations with merging
- space-time vertices available in hadronisation and decays
- new models for total, elastic, and diffractive cross-sections
- photoproduction, including diffraction and ultraperipheral collisions
- new dark matter models
- deuteron production
- extended quarkonia production including in showers

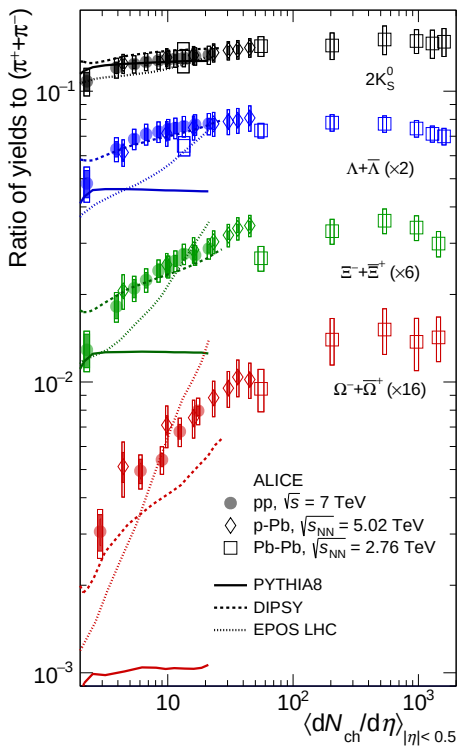
Heavy Ions (JHEP 1810 (2018) 134)

- the LHC doesn't just collide protons with protons ...
- heavy ion and high energy communities getting closer
  - ridge effect in AA, pA, and pp
  - enhanced strangeness
- are collective effects from a thermal origin?

Phys.Lett. B765 (2017) 193-220

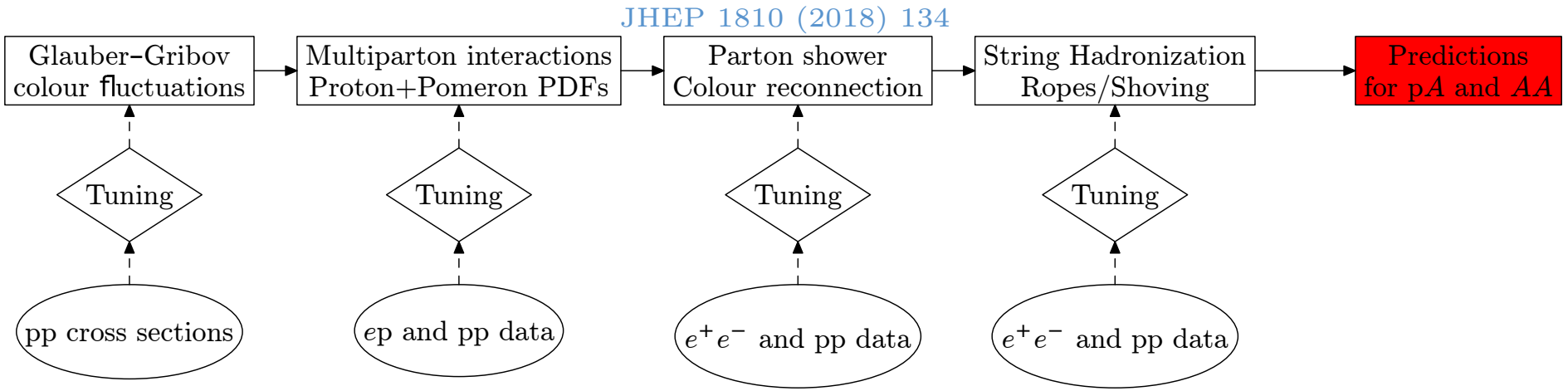


Nature Phys. 13 (2017) 535-539



Heavy Ions (JHEP 1810 (2018) 134)

- developed by Leif Lönnblad and Christian Bierlich



1. determine nucleon positions of colliding nuclie
2. calculate number of interacting nucleons (Glauber model)
3. estimate contribution from each interacting nucleon (Fritiof based)
4. merge parton-level interactions (Pythia MPI)
5. hadronise (ropes and strings)

Heavy Ions (JHEP 1810 (2018) 134)

- examples `main111`, `main112`, `main113`

```
pythia.readString("Beams:idA = 2212");          // Proton.
pythia.readString("Beams:idB = 1000822080");    // Lead ion.
pythia.readString("Beams:eA = 4000");           // Proton energy.
pythia.readString("Beams:eB = 1570");           // Lead ion energy.
pythia.readString("Beams:frameType = 2");       // Asymmetric beams.
pythia.readString("HeavyIon:mode = 2");         // Force if proton beams.
```

- embedded hard process signal production can be specified

```
pythia.readString("WeakSingleBoson:ffbar2gmZ = on")
```

- initialisation can be slow but can also be cached

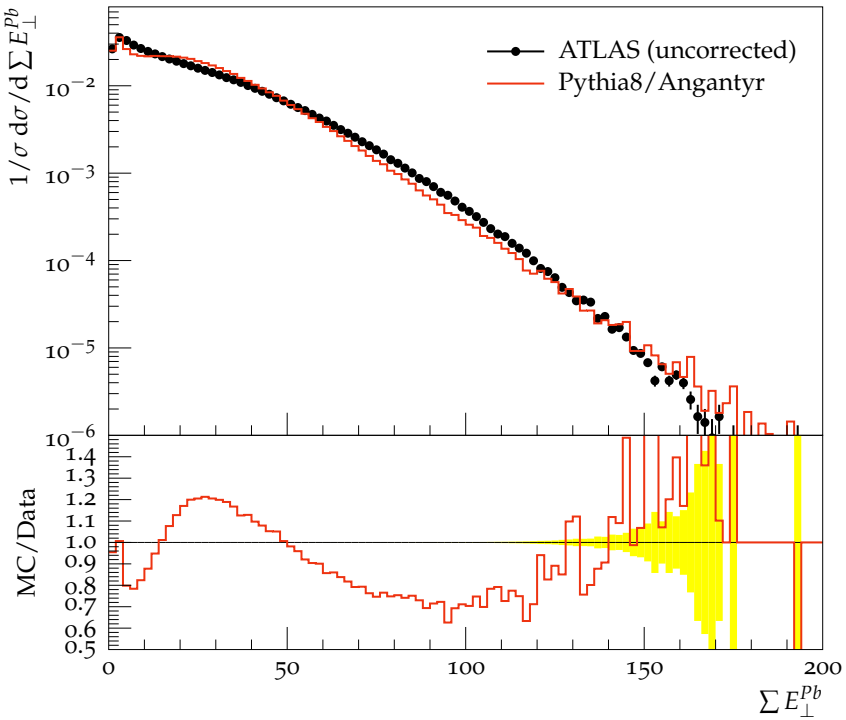
```
pythia.readString("HeavyIon:SigFitNGen = 0");
pythia.readString("HeavyIon:SigFitDefPar = →
    13.88,1.80,0.22,0.0,0.0,0.0,0.0,0.0");
```

- rope hadronisation can be used to describe interactions between overlapping strings
- validation with heavy ion model underway

Heavy Ions (JHEP 1810 (2018) 134)

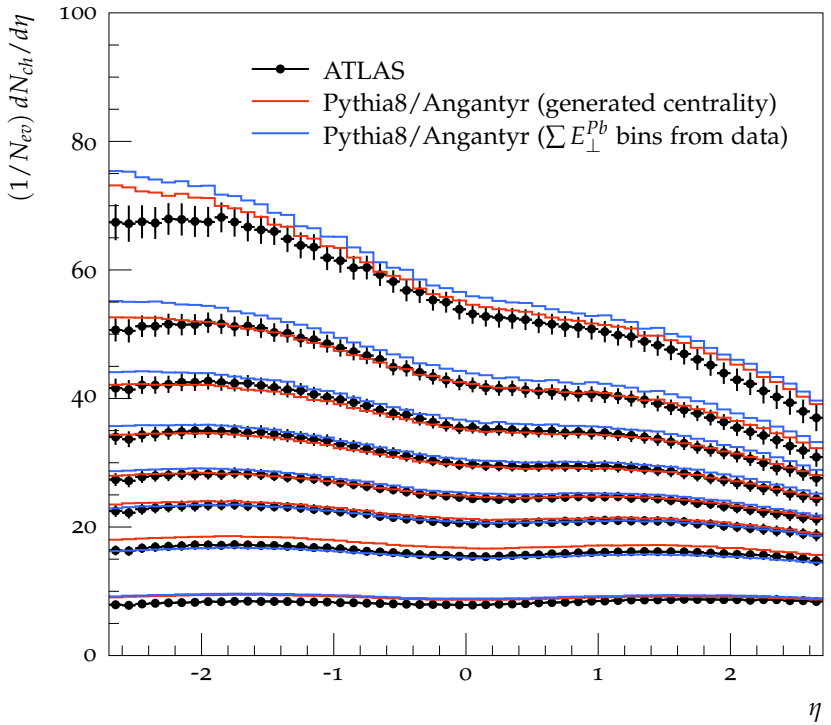
JHEP 1810 (2018) 134

Sum $E_{\perp}^{Pb}$ distribution, pPb, $\sqrt{S_{NN}} = 5$ TeV.



JHEP 1810 (2018) 134

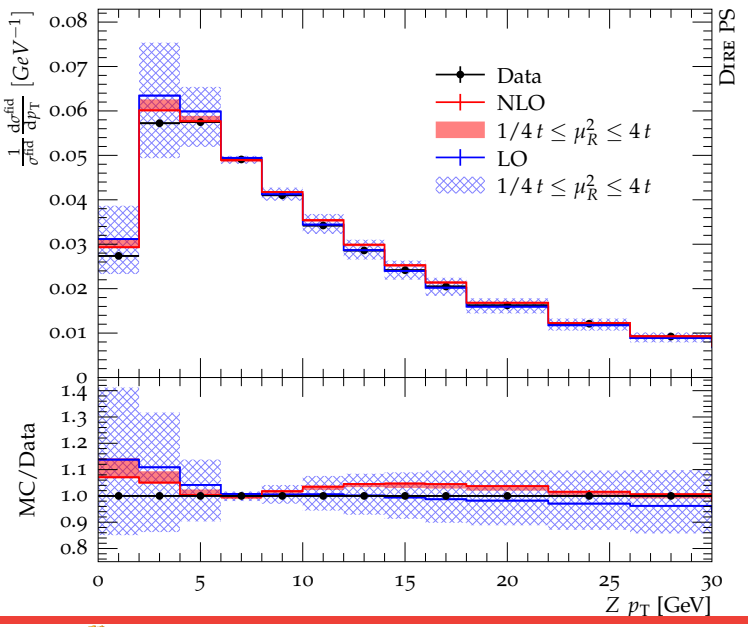
(a) Centrality-dependent η distribution, pPb, $\sqrt{S_{NN}} = 5$ TeV.



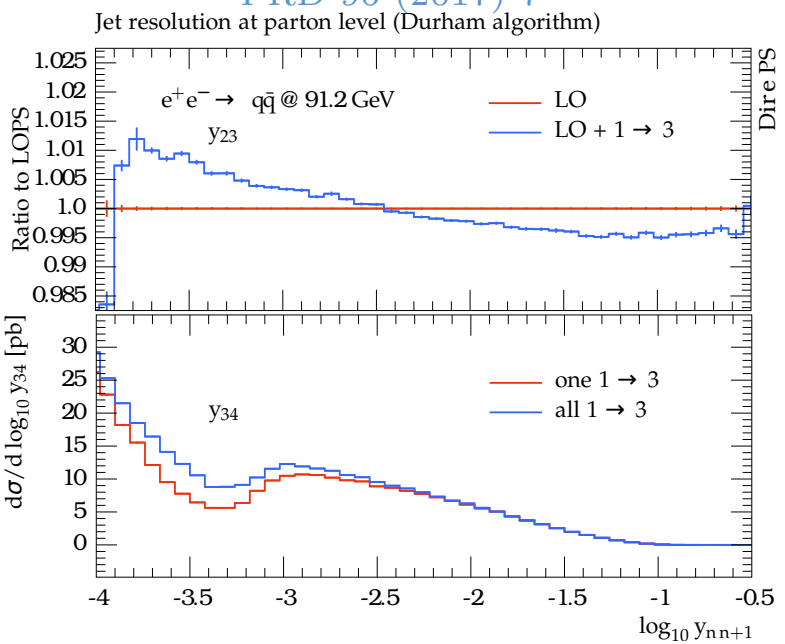
Dire (EPJC 75 (2015) 9)

- developed by Stefan[2] (Höche and Prestel)
- dipole shower with careful treatment of collinear enhancements
  - NLO DGLAP evolution (JHEP 1710 (2017) 093)
  - $1 \to 3$ splitting kernels (PRD 96 (2017) 7)
- independent implementations for Pythia and Sherpa
  - version 2.002 for Pythia at `dire.gitlab.io`
  - will be more closely integrated with Pythia in next release

JHEP 1710 (2017) 093



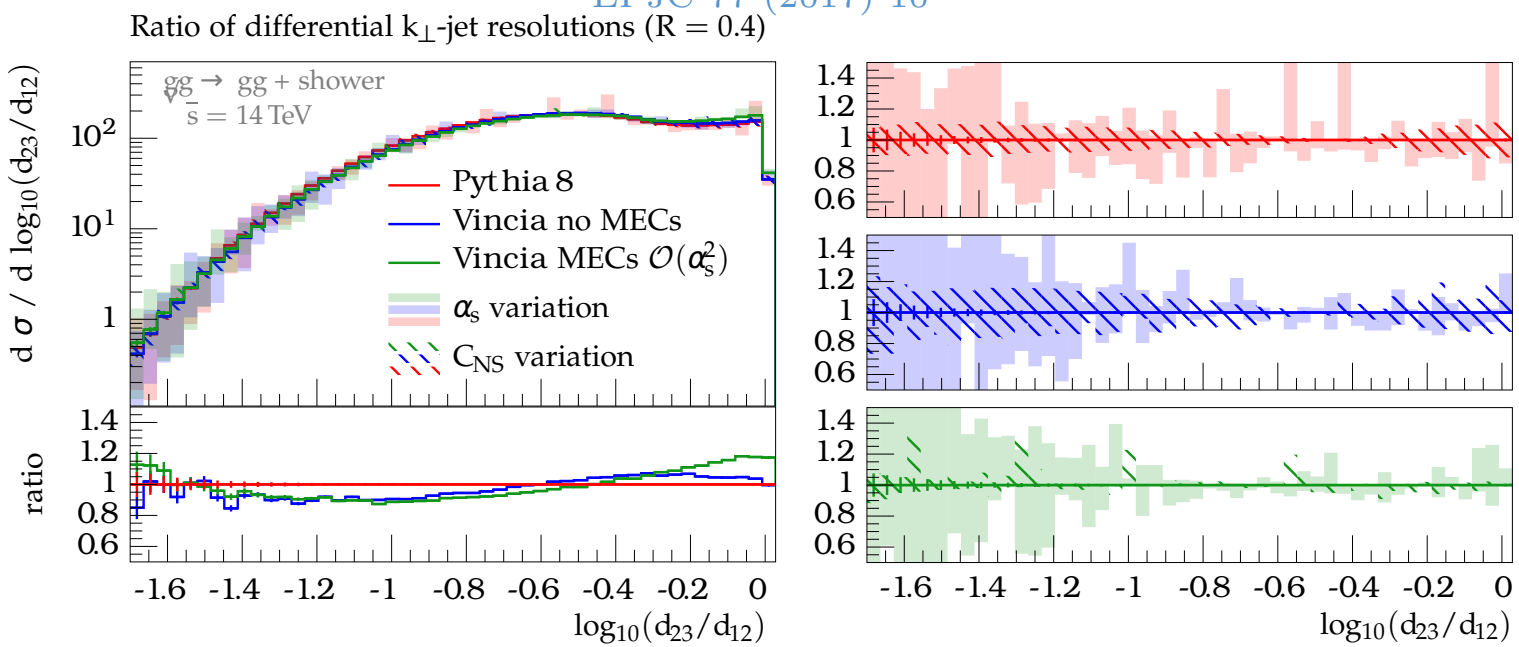
PRD 96 (2017) 7



Vincia (EPJC 76 (2016) 11)

- developed by Peter Skands and previously Nadine Fischer
- dipole-antenna shower
  - parameterless shower merging hierarchy (EPJC 77 (2017) 9)
  - helicity antenna showers (EPJC 77 (2017) 10)
- plugin for Pythia
  - version 2.2.02 for Pythia at `vincia.hepforge.org`
  - will be more closely integrated with Pythia in next release

EPJC 77 (2017) 10



Shower Variations (PRD 94 (2016) 7)

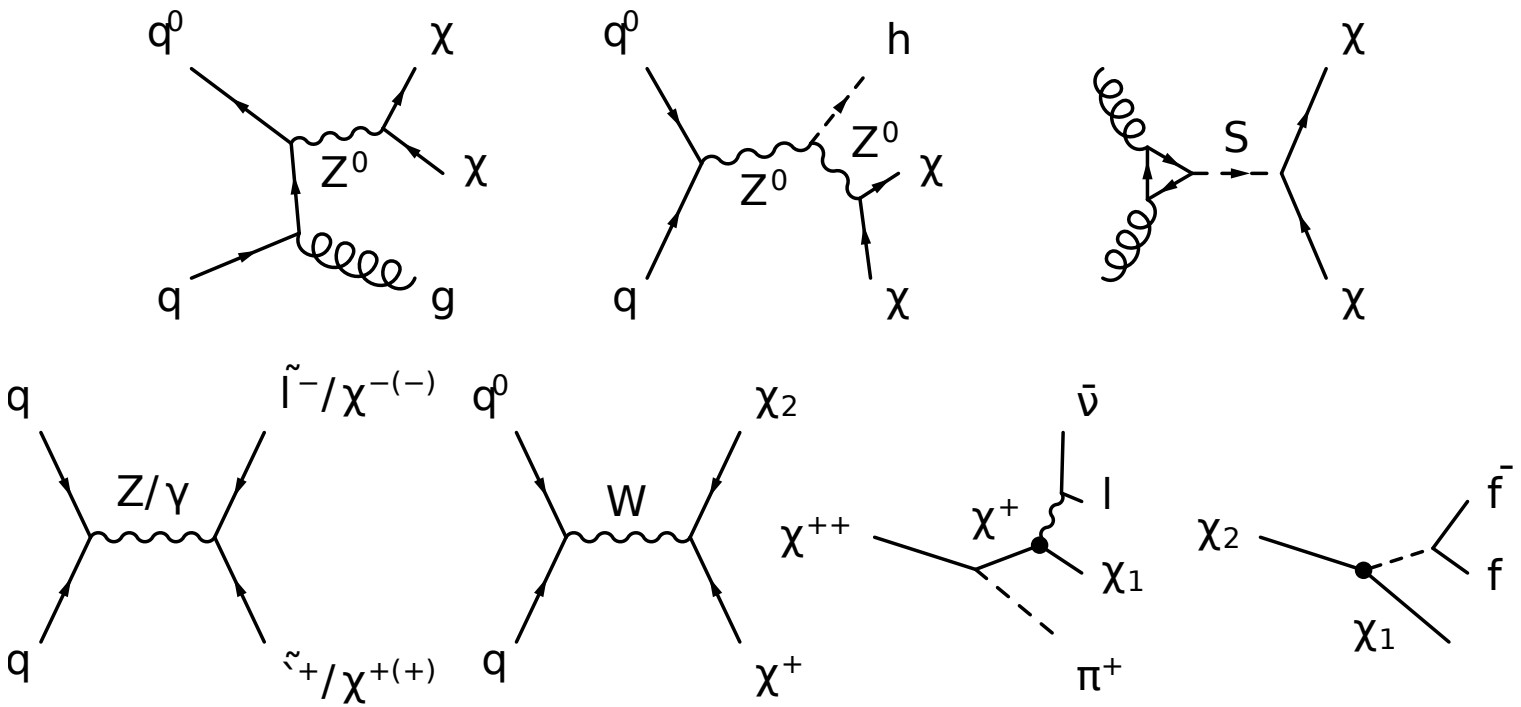
- developed by Stephen Mrenna and Peter Skands
- ongoing work to fully include variations in merging
  - FSR QCD renormalisation scale
  - ISR QCD renormalisation scale
  - non-singular terms in QCD FSR
  - non-singular terms in QCD ISR
  - PDF members

```
# Switch on shower uncertainty estimates
UncertaintyBands:doVariations = on

# Renormalisation and PDF uncertainty.
UncertaintyBands:List = {
  scale_fsr_lo fsr:muRfac=0.5,
  scale_fsr_hi fsr:muRfac=2.0,
  scale_isr_lo isr:muRfac=0.5,
  scale_isr_hi isr:muRfac=2.0,
  scale_pdf_lo isr:pdf:minus=0.5,
  scale_pdf_hi isr:pdf:plus=2.0
}
```

Dark Matter Models (arXiv:1807.04240)

- developed by Nishita Desai
- four new dark matter models added
  - vector s-channel mediator
  - scalar s-channel mediator
  - scalar t-channel mediator
  - SU(2) n-plet mixed with singlet

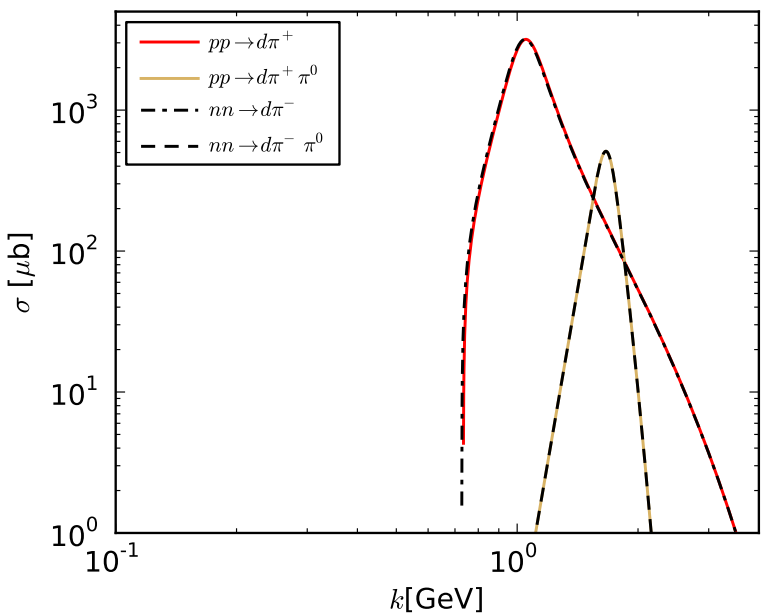


Deuteron Production

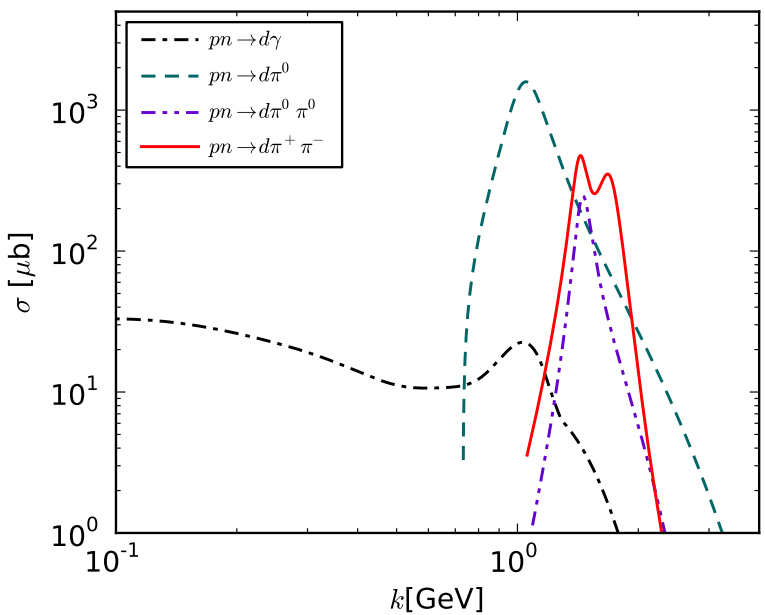
- developed by Phil Ilten
- generalised implementation of PRD 62 (2000) by Donato, Forengo, and Salati

$$pn \to \gamma d \quad pn \to \pi^0 d \quad pn \to \pi^- \pi^+ d \quad pn \to \pi^0 \pi^0 d$$
$$pp \to \pi^+ d \quad nn \to \pi^- d \quad pp \to \pi^+ \pi^0 d \quad nn \to \pi^- \pi^0 d$$

PRD 62 (2000)



PRD 62 (2000)



Conclusions

Outlook

- Pythia is ready for multi-threaded environments
- designed to allow interfacing of external tools

- new parton showers are being more closely interfaced
- heavy ions are now available
- significant work on photoproduction and diffraction
- quarkonia, dark matter, deuterons, and more ...

Thanks!