Systematic MC tuning at LHC

Hendrik Hoeth
(Durham University)

Overview

Tuning . . . and then a miracle occurs

Miracles . . . and where we can't cheat

Plots . . . we haven't seen this week

Motivation for Tuning

All experimental measurements rely on Monte Carlo generators (to a certain degree).

All generators are based on phenomenological models: string fragmentation, dipole cascade, cluster hadronisation, MPI, ...

The models have free parameters which are a priori unknown and need to be tuned: flavour ratios, q_0^2, fragmentation parameters, intrinsic k_T, reabsorption/rescattering ...

Even parameters like α_s need to be optimised!

Physics Validation and Global Comparisons

We want the MC to have predictive power, so we need to fit as wide a range of available data as possible!

Systematic global validation is essential for testing models or developing general-purpose tunings.

Generator tuning also helps debugging code and models. Short turn-around is important.

Problems

The model parameters are highly correlated
$\Rightarrow$ can't be tuned one after the other.

Many parameters to be tuned ($\mathcal{O}(10)$).

Tuning all parameters at the same time puts us into a high dimensional parameter space.

Brute force approaches (usually) don't work: Running the MC generator takes too long for every point in the parameter space (= setting of parameters).

We use two toolkits: Rivet and Professor.

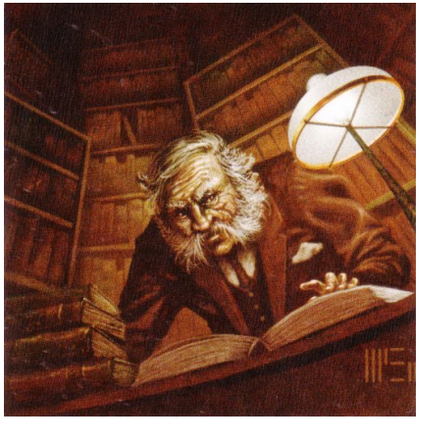
Rivet

Rivet is a library of tools (event shape calculators, jet algorithms, final state definitions, histogramming . . .) and so far about 90 analysis routines which use them.

- Can read HepMC events from any generator
- Histograms can be auto-booked from reference data files
- Observables are automatically cached
- User analyses loaded as plugins
- Reference data is included in the Rivet release
- Perfect place for archiving analysis knowledge

Documentation, download etc:
http://projects.hepforge.org/rivet/

Professor



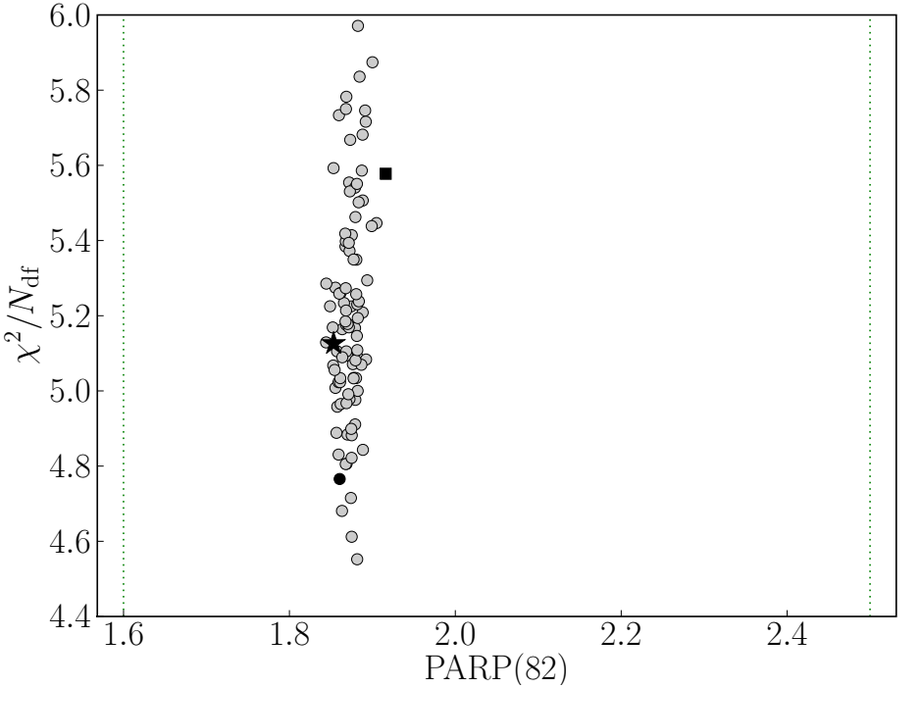
Professor is a tuning tool developed within MCnet, extending the strategy used by Tasso and Delphi, keeping the historic name.

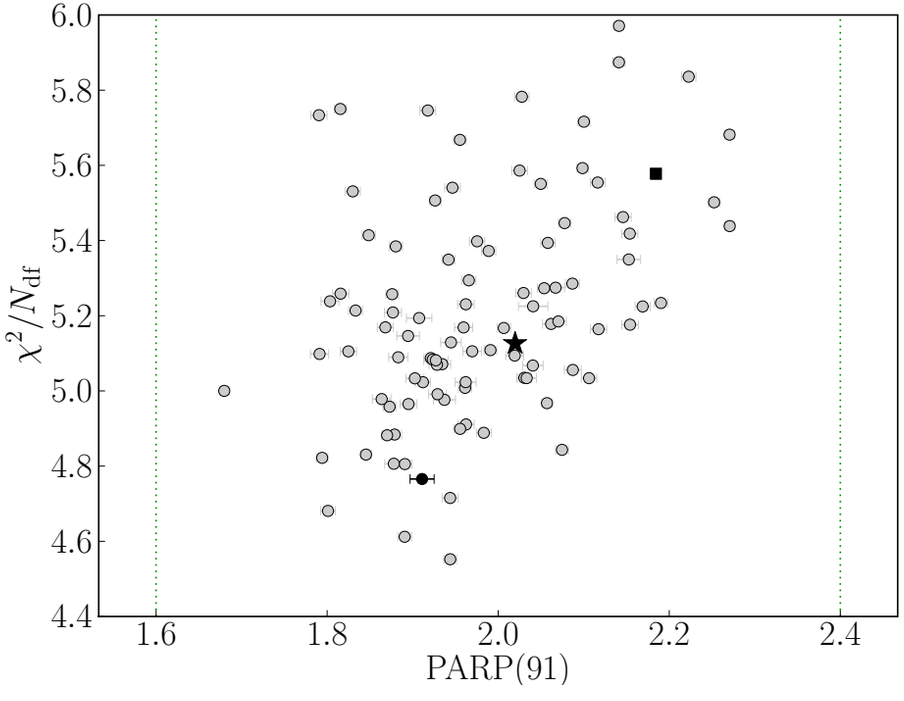
- Sample N random points in parameter space and run the generator with those settings
- For each bin of each distribution use the N MC runs to fit an interpolation function in order to parametrise the MC output
- Construct overall χ^2 comparing this parametrisation with data and minimise
- Use different combinations of observables, weights etc. to get a feeling for stability, systematics ...

http://projects.hepforge.org/professor/

Professor – Tools

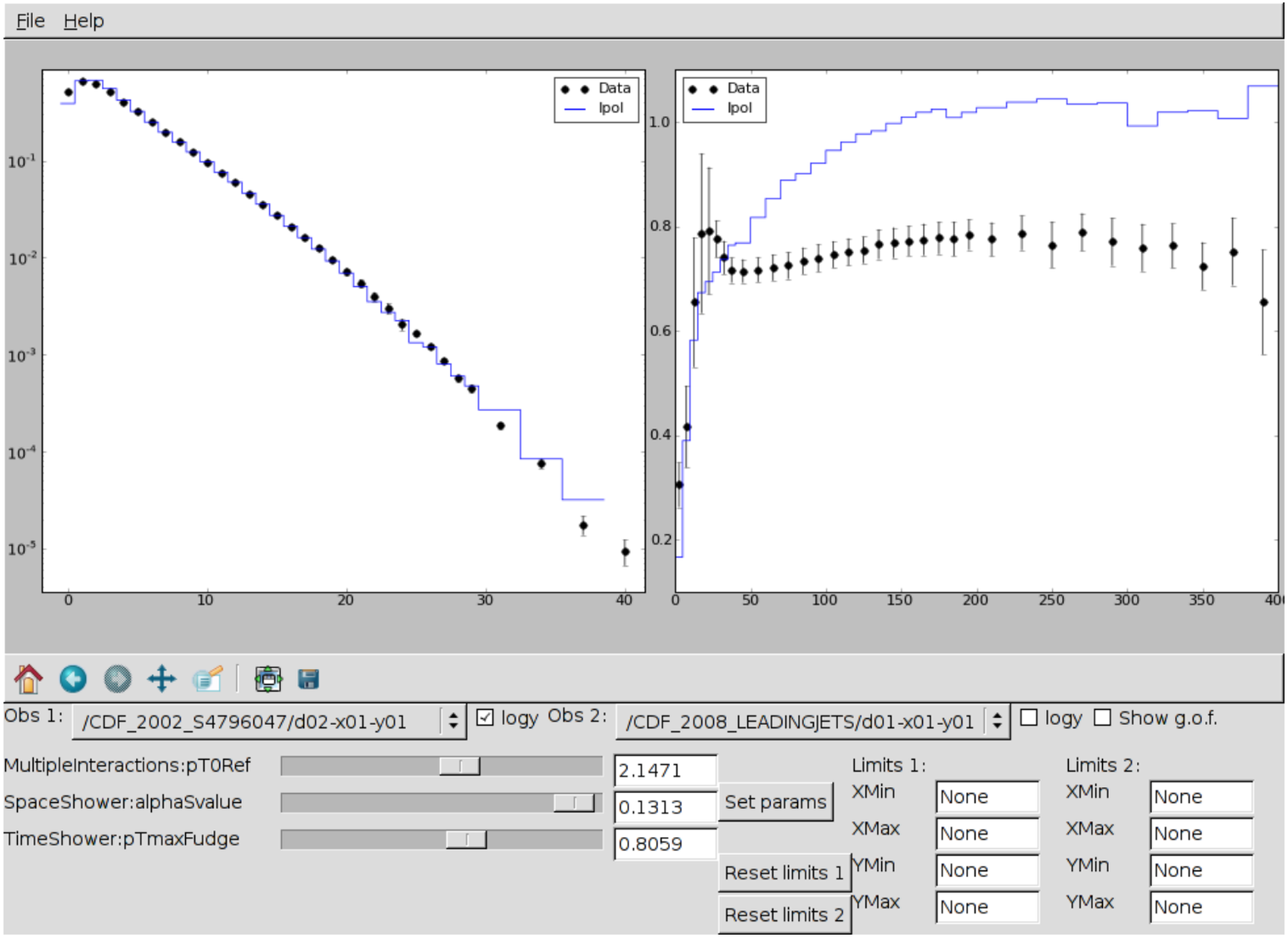
Check sensitivity / stability and estimate uncertainties:





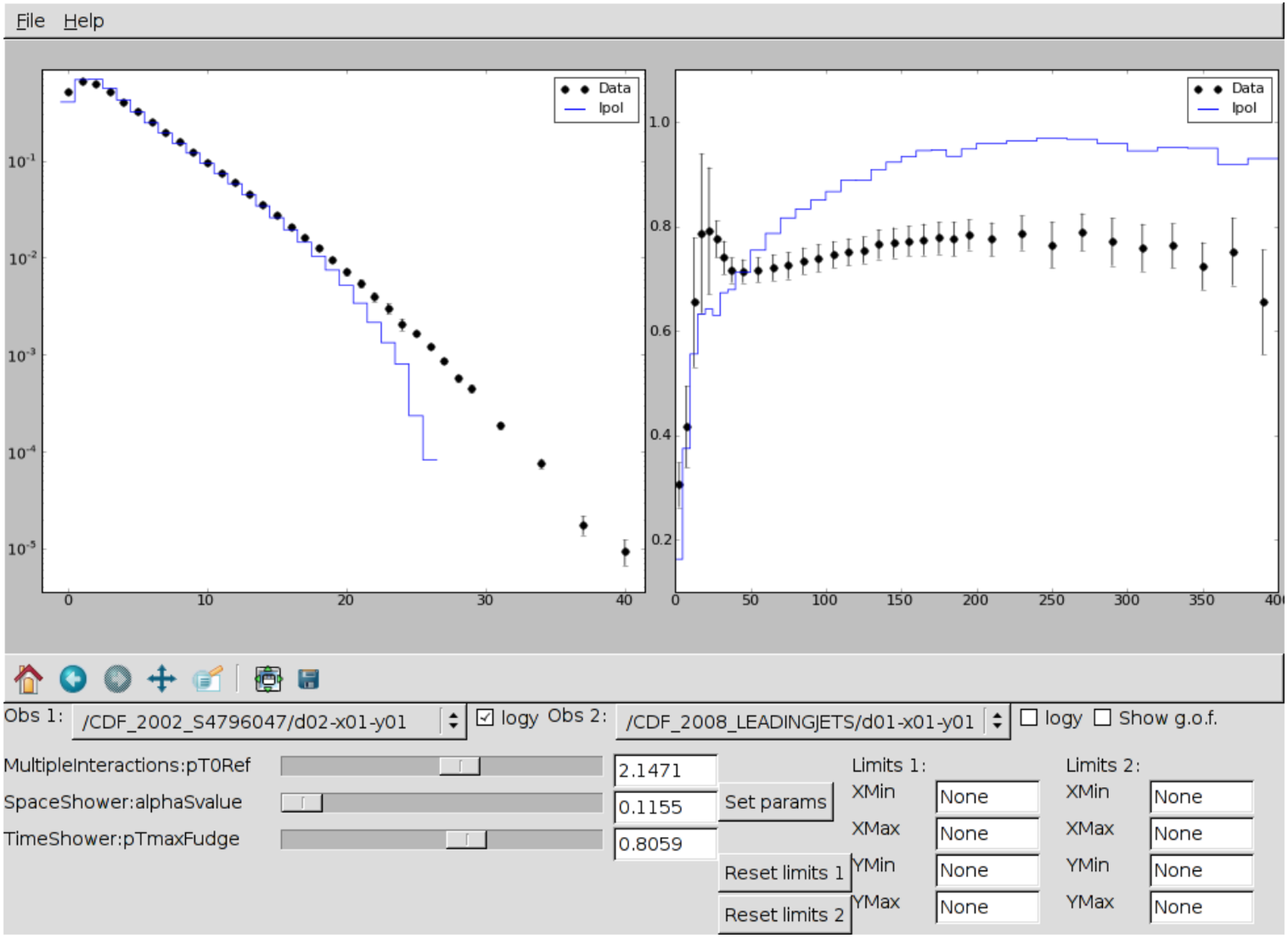
Professor – Tools

Play interactively with parameters:



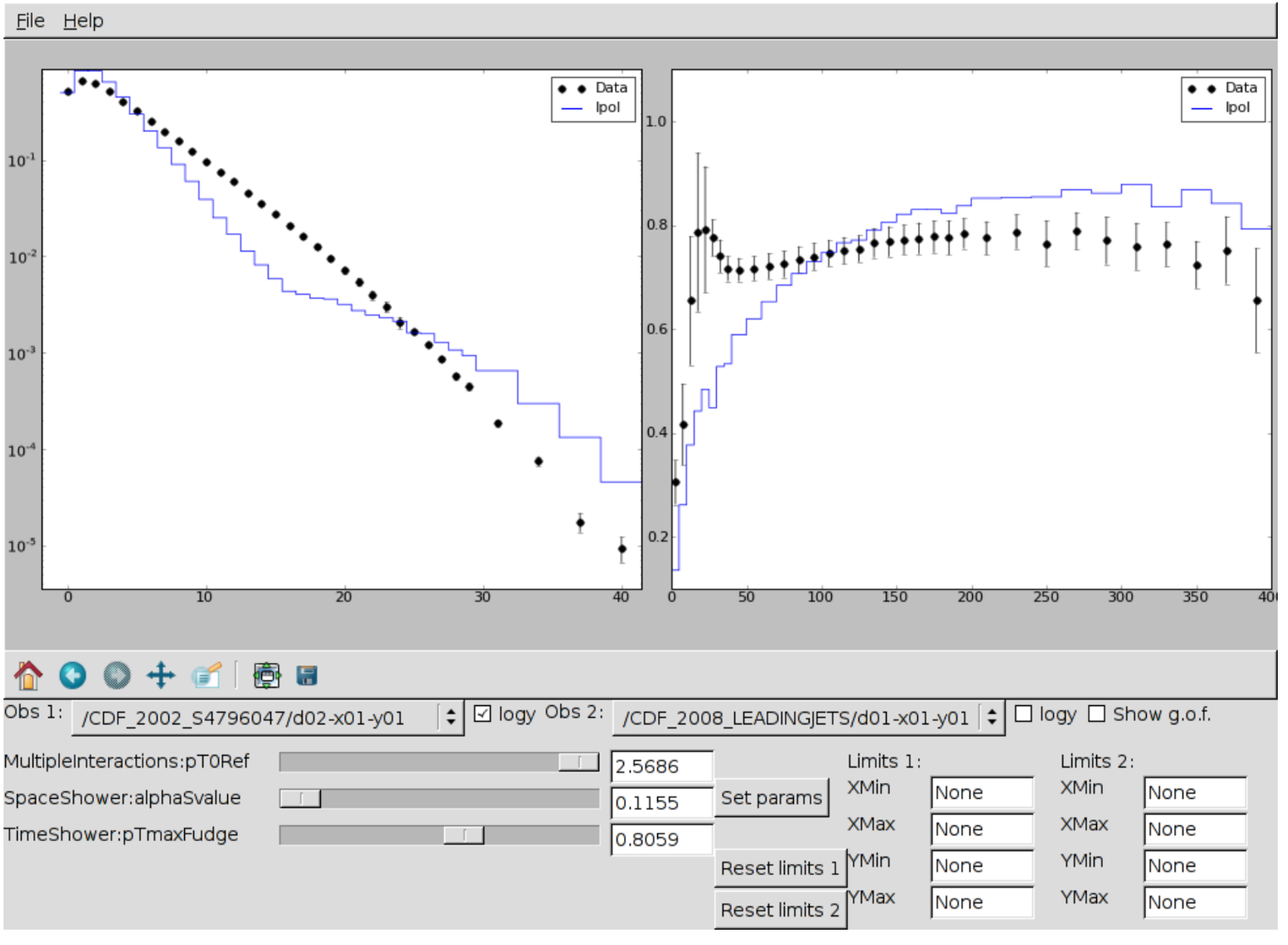
Professor – Tools

Play interactively with parameters:



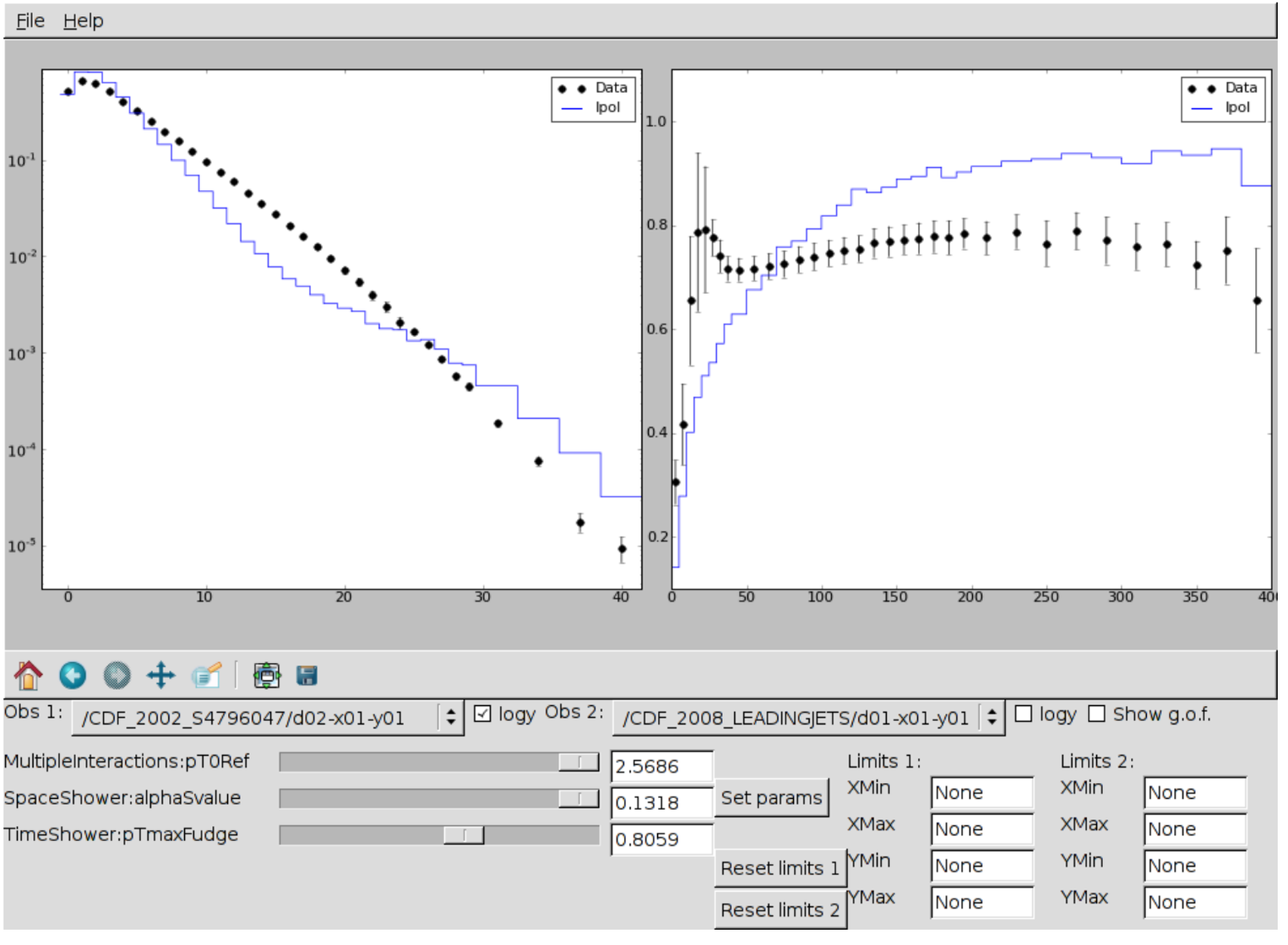
Professor – Tools

Play interactively with parameters:



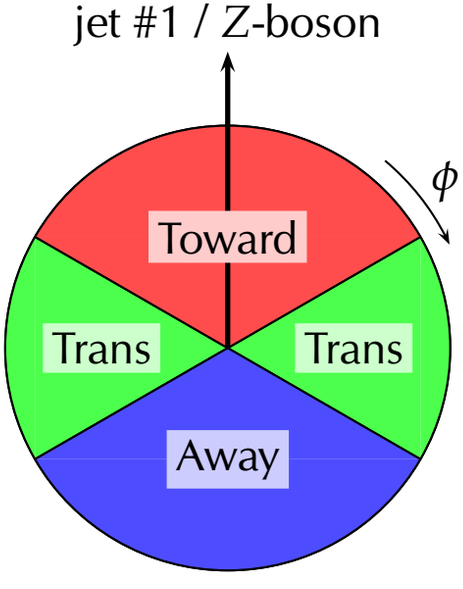
Professor – Tools

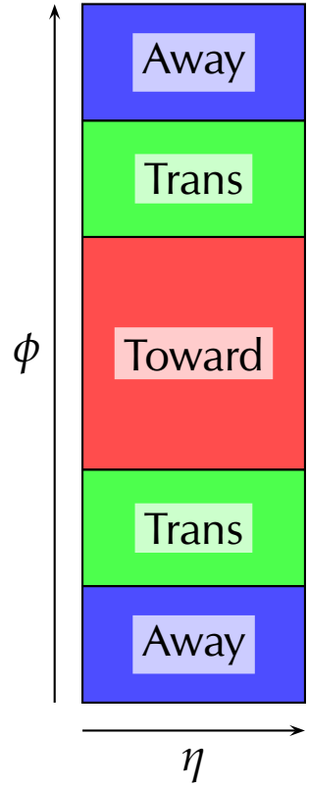
Play interactively with parameters:

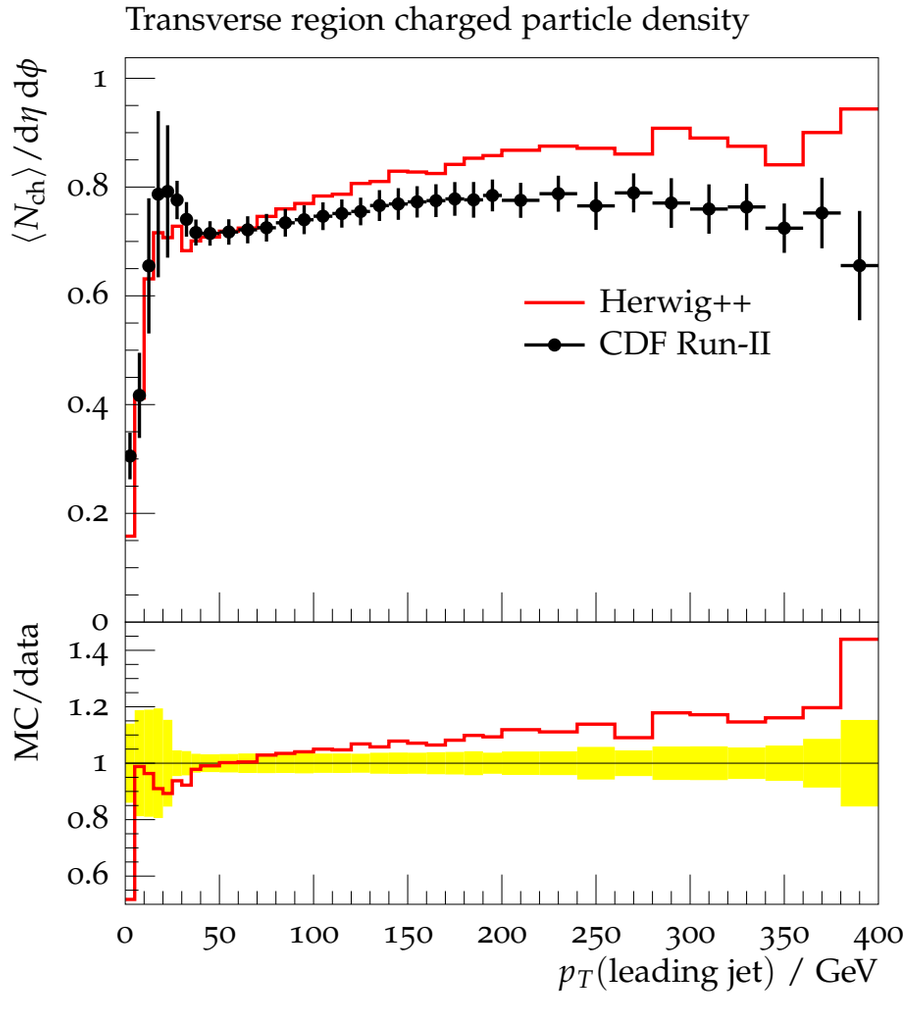


Herwig++ UE – the Problem

Herwig++ has too much stuff flying around in the transverse region.





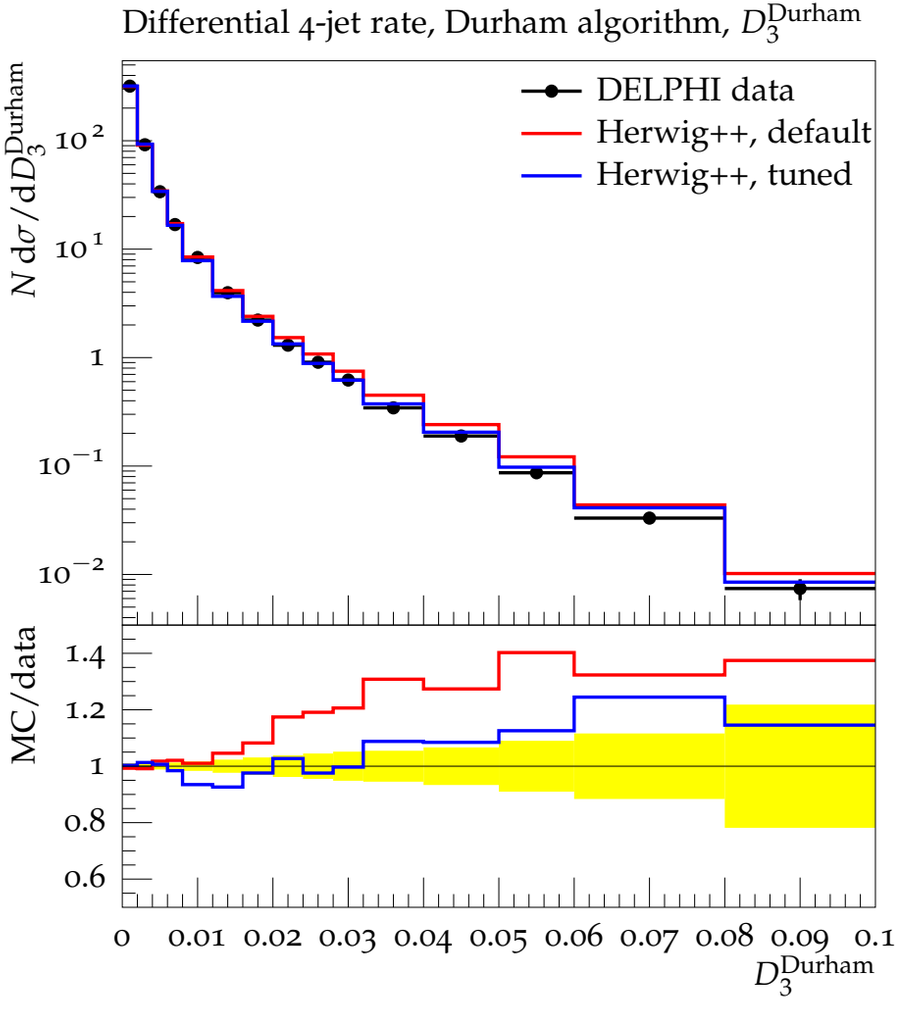


Herwig++ – LEP

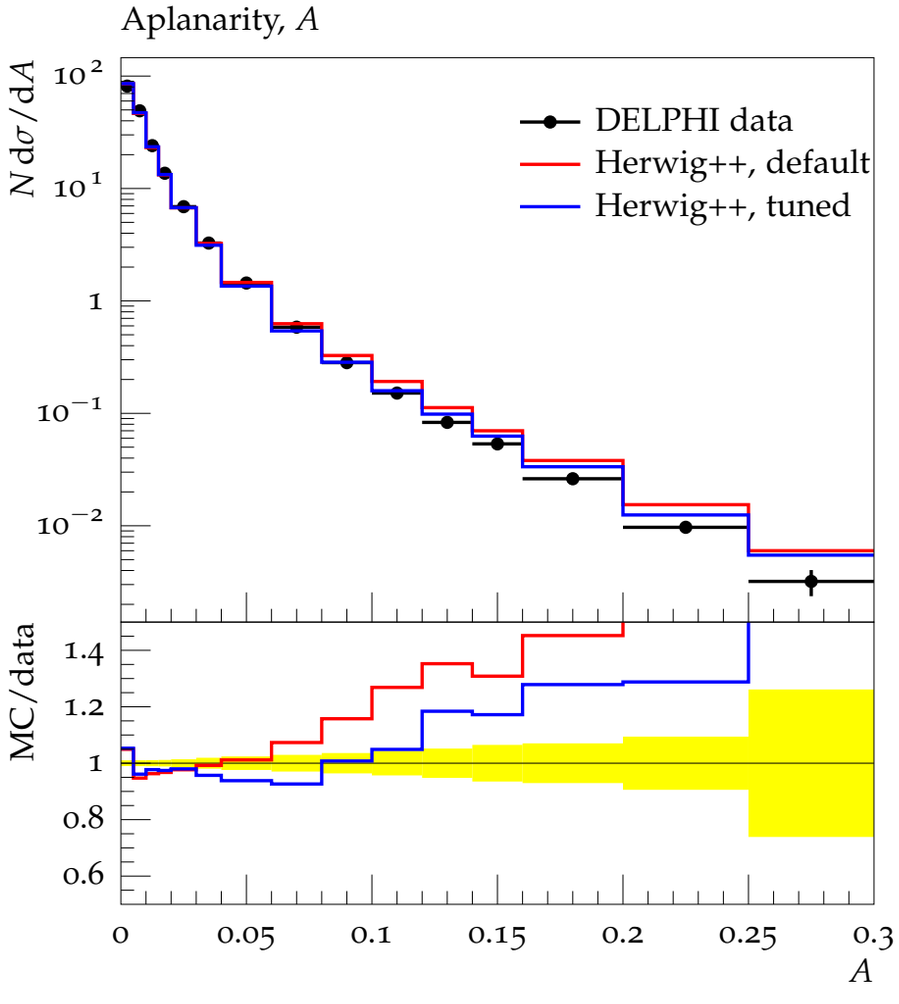
Looking at LEP observables, we see that the problem already exists here:

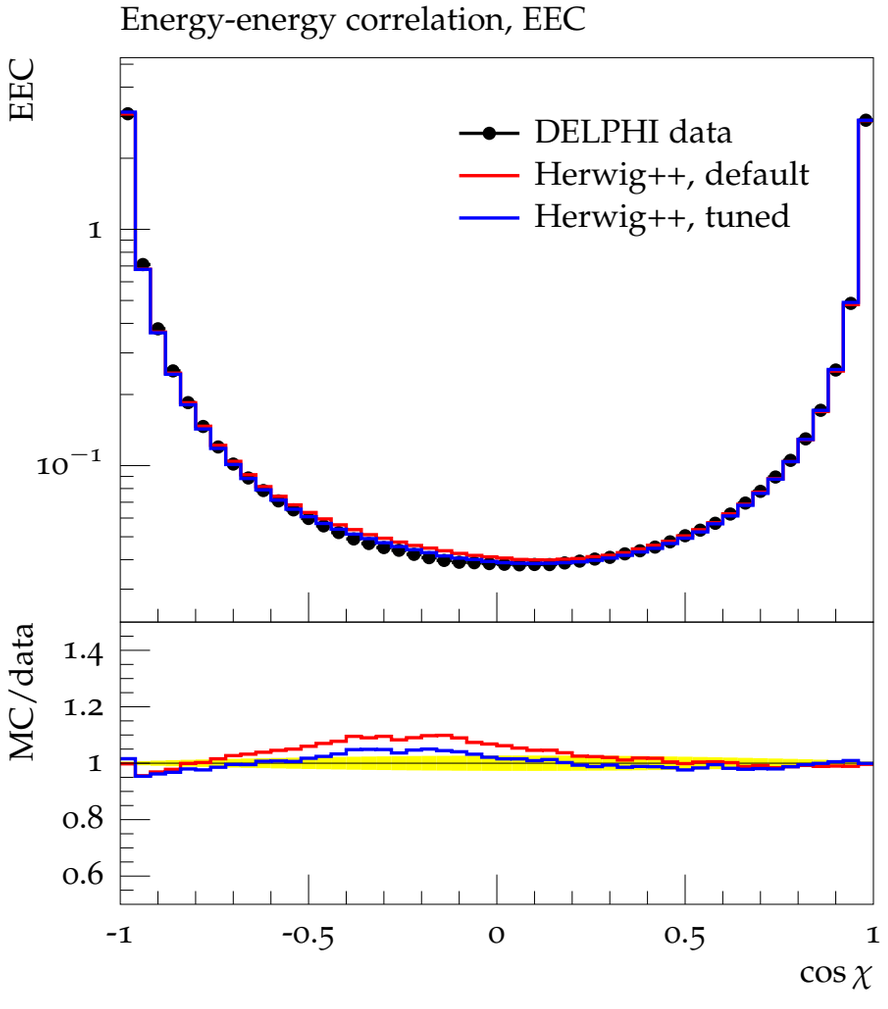
Multijet rates are high, events are to spherical, EEC shows a 3-jet excess.

$\Rightarrow$ The toward/away regions probably bleed into the transverse region, causing the slope.



Herwig++ – LEP

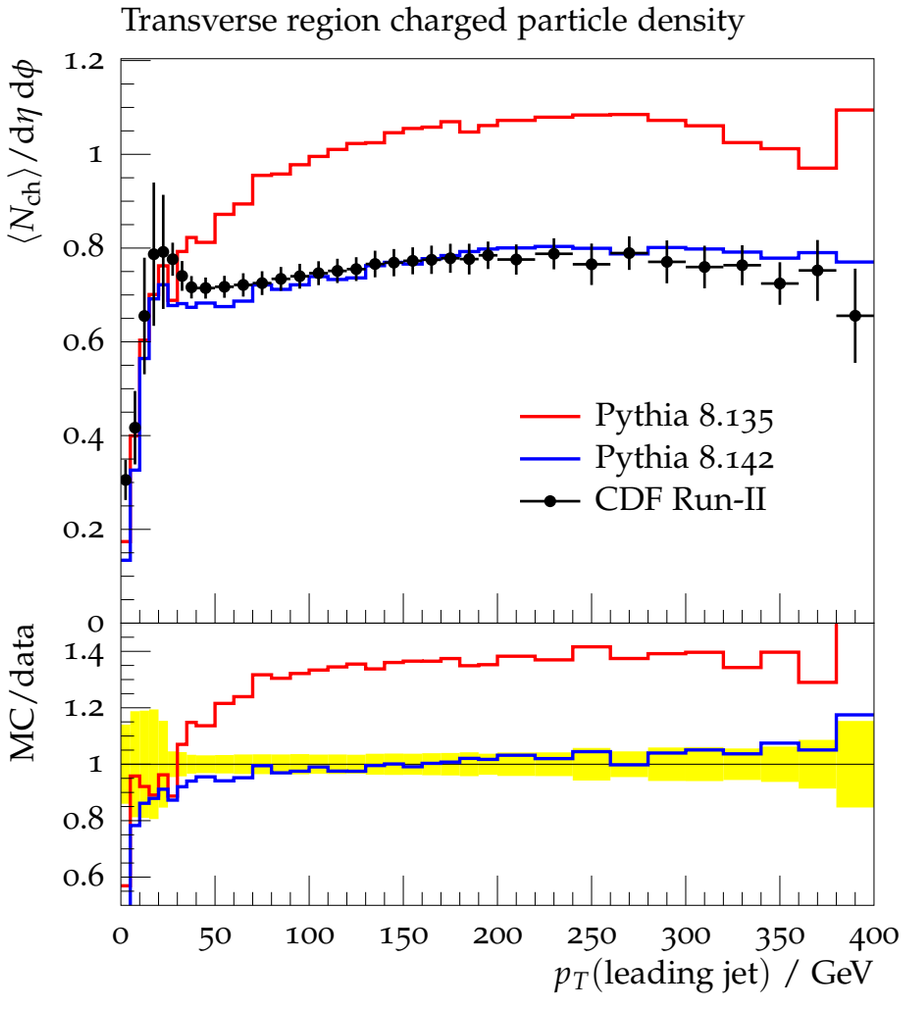


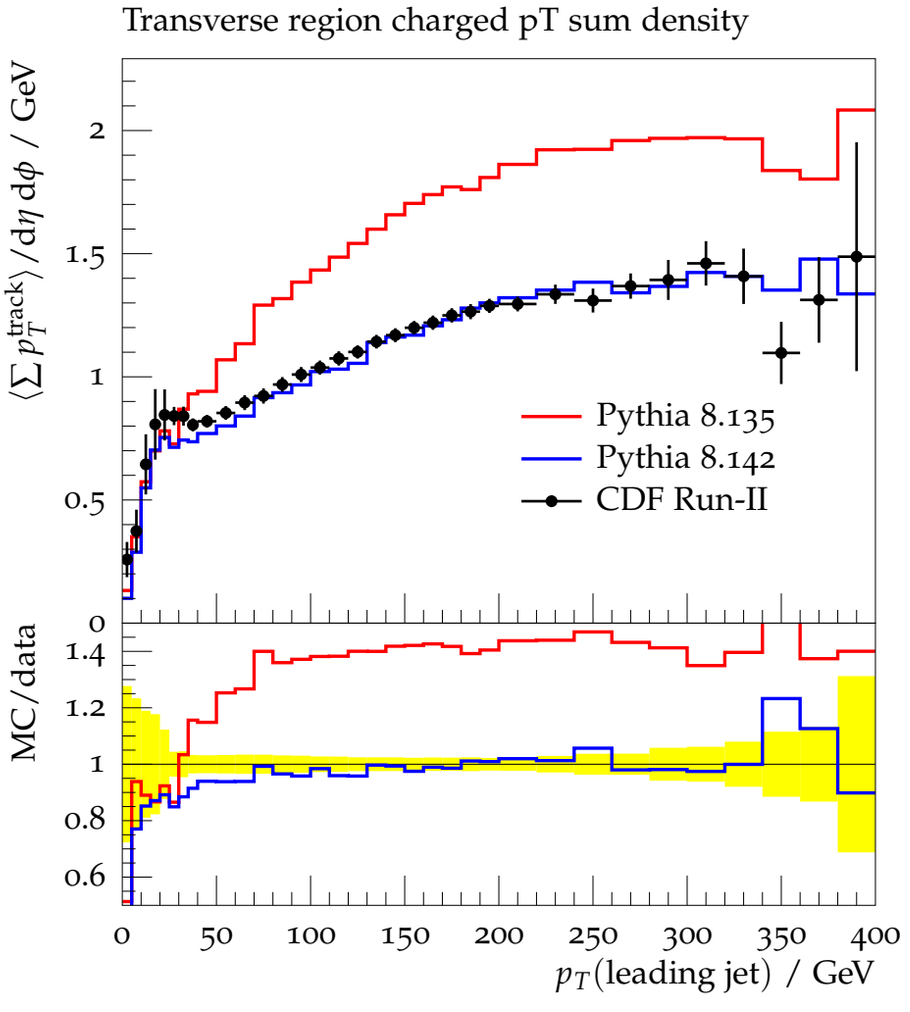


$\Rightarrow$ Problem in the shower, still under investigation.

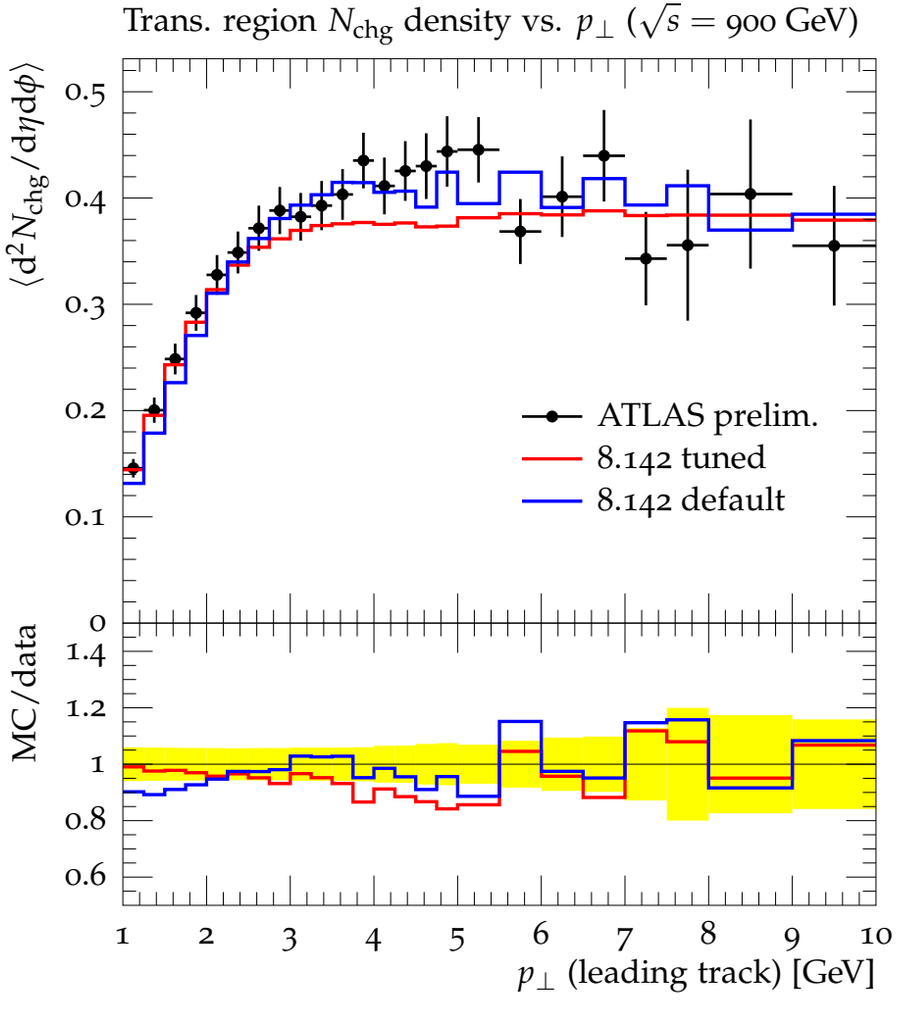
Pythia 8 – UE

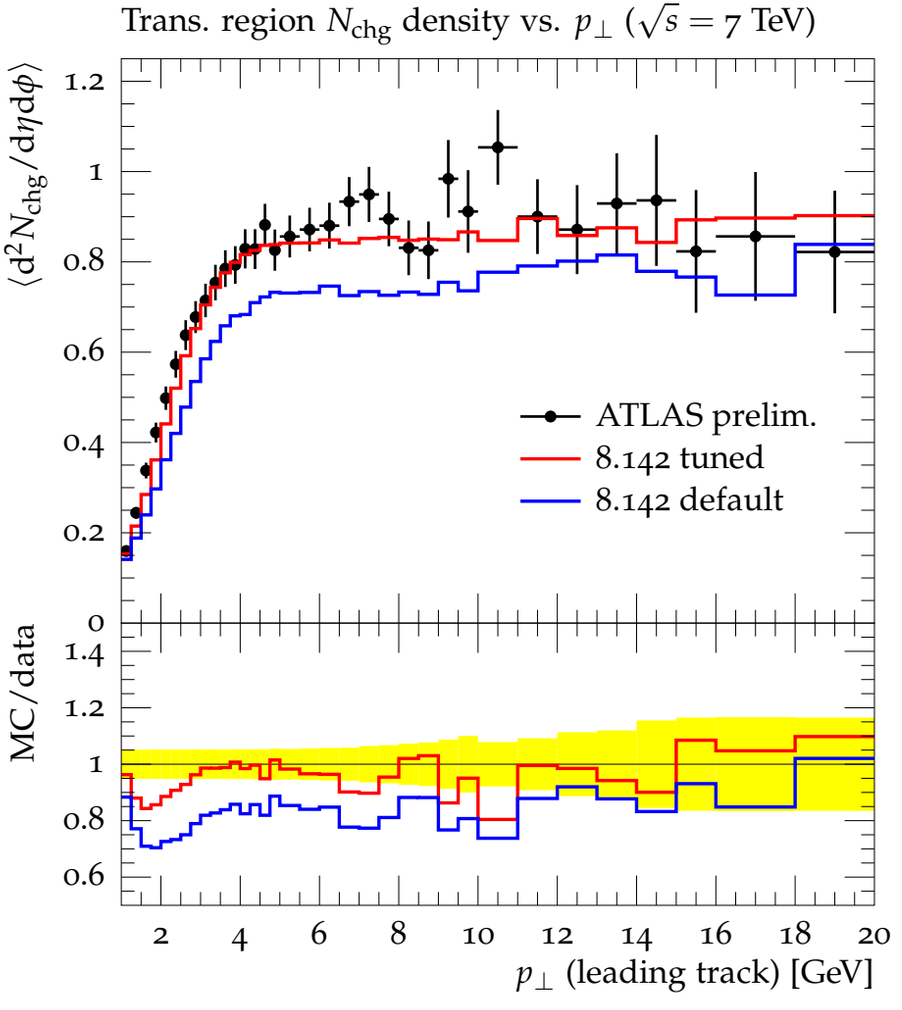
Bug in versions older than 8.140, affecting hard QCD events.



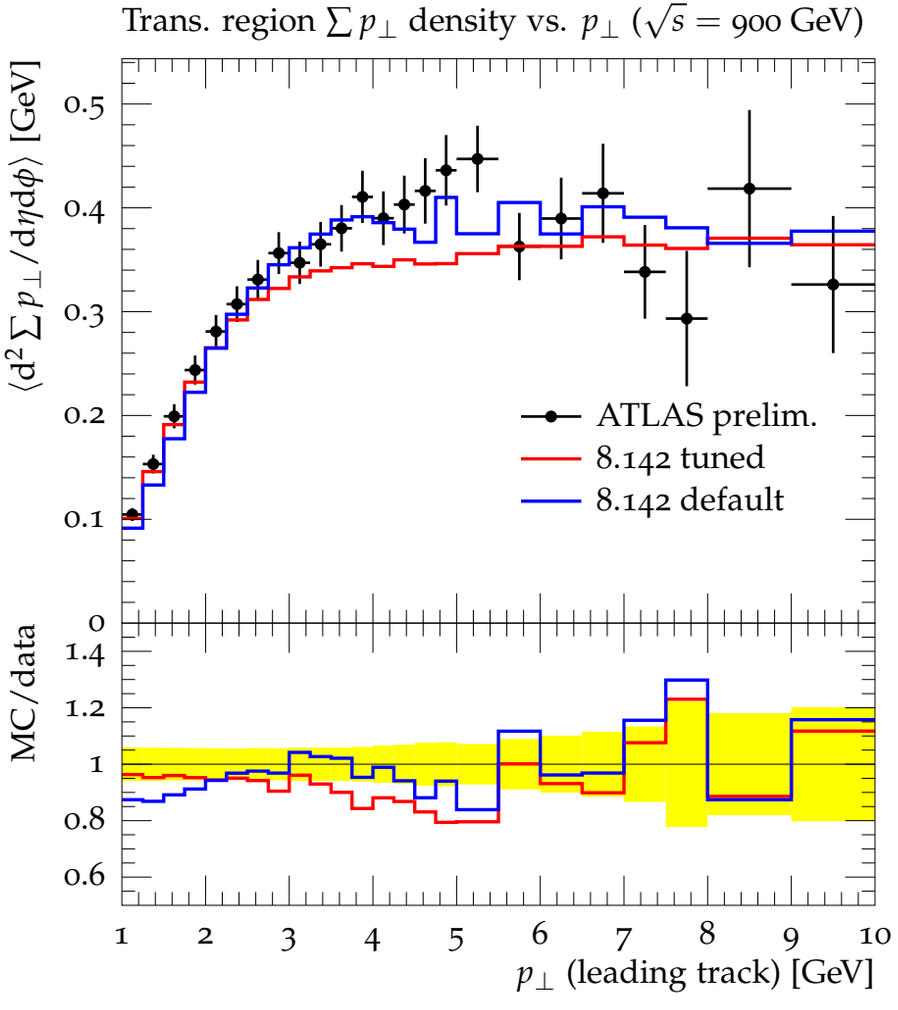


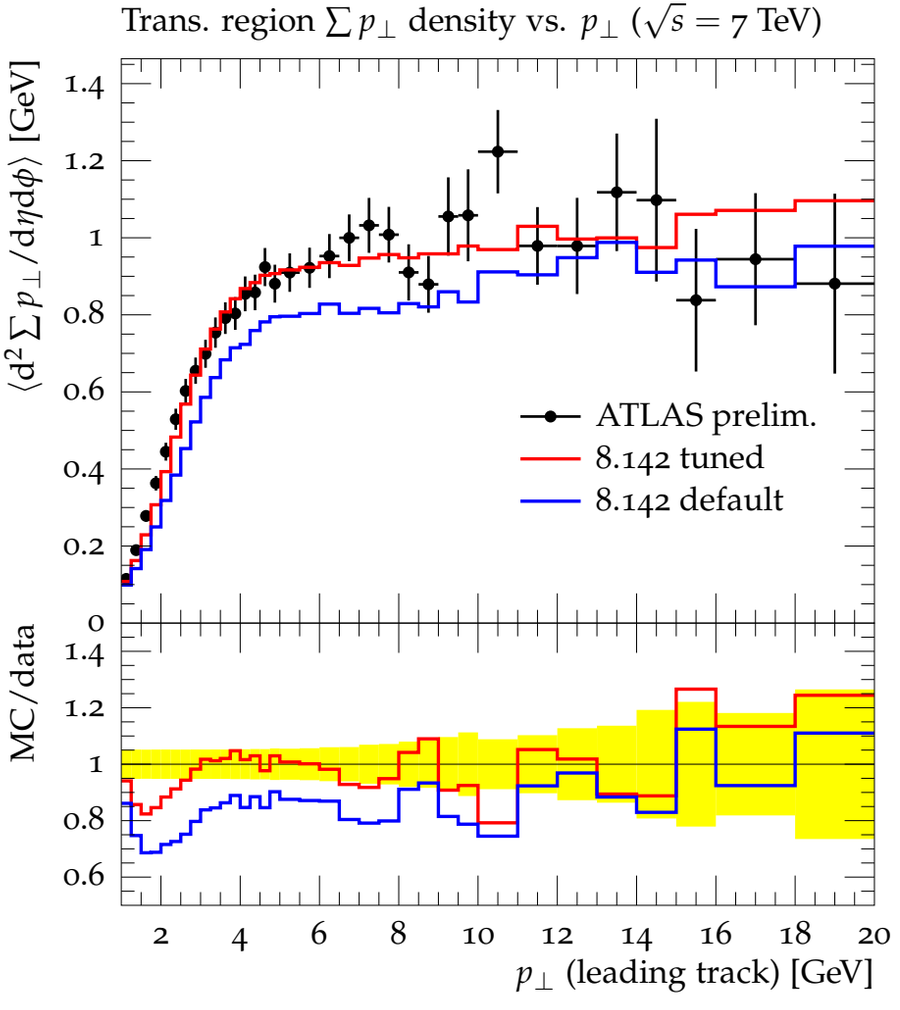
Pythia 8.142 – First Tuning Results



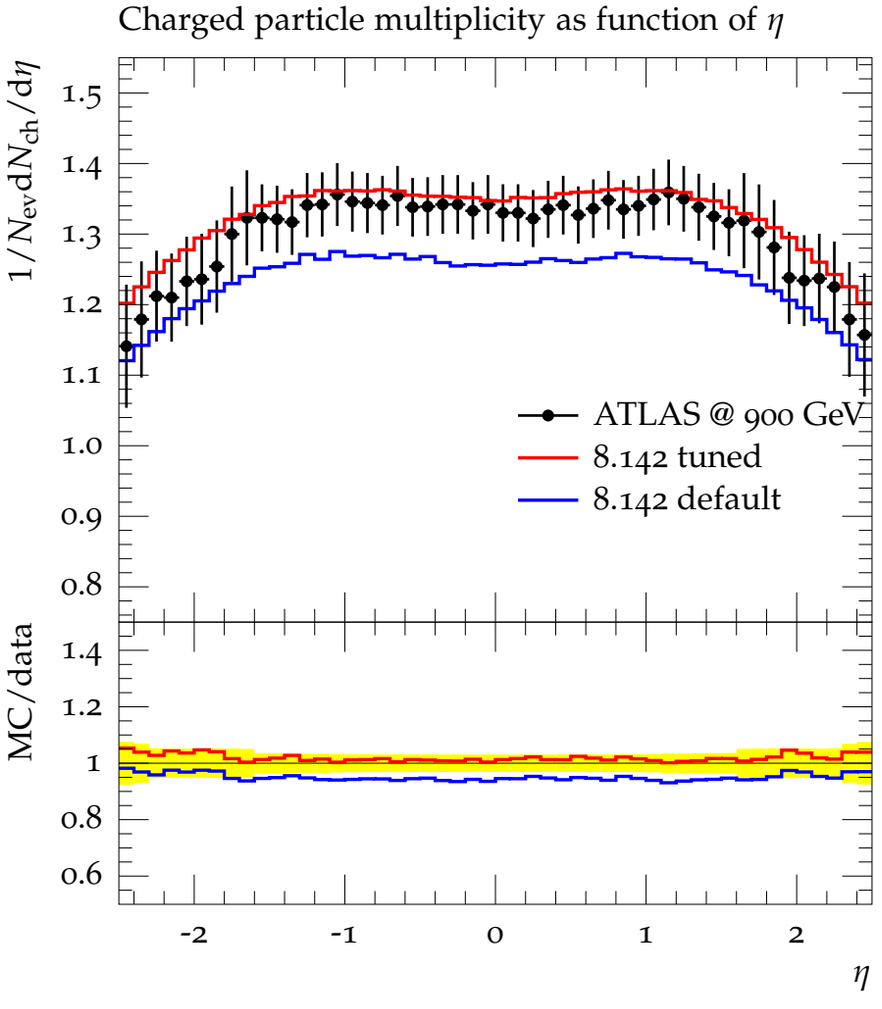


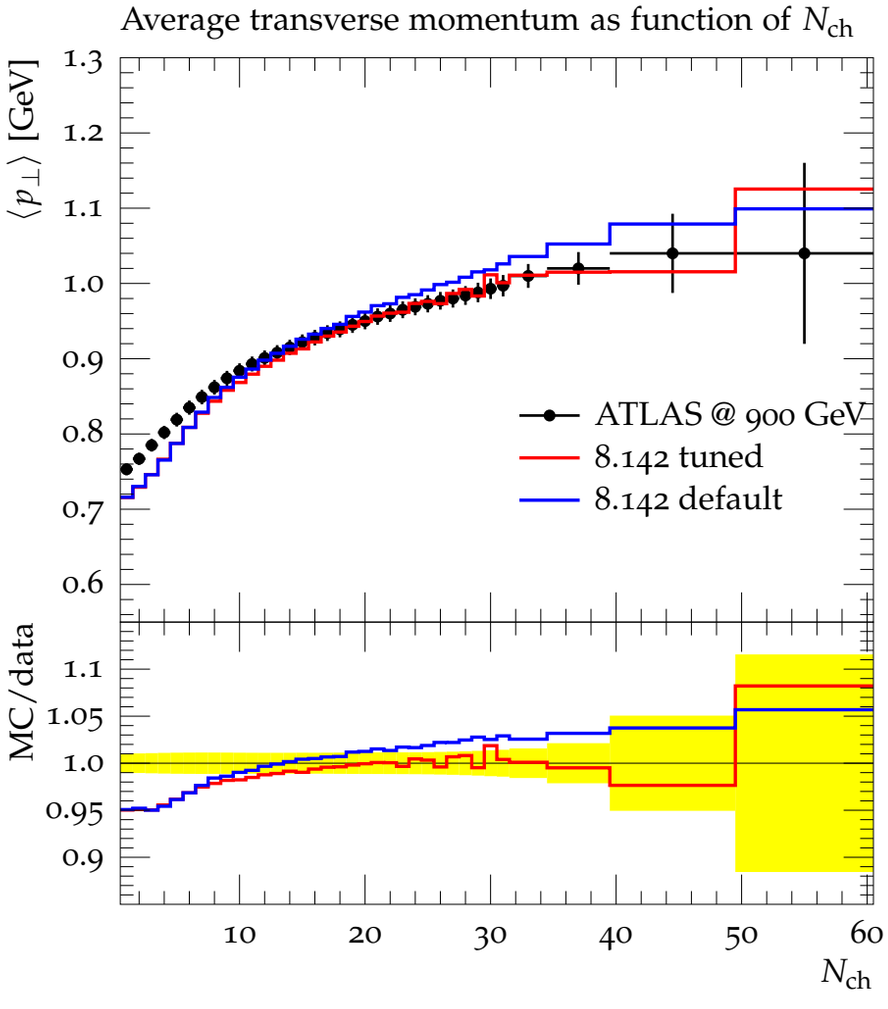
Pythia 8.142 – First Tuning Results





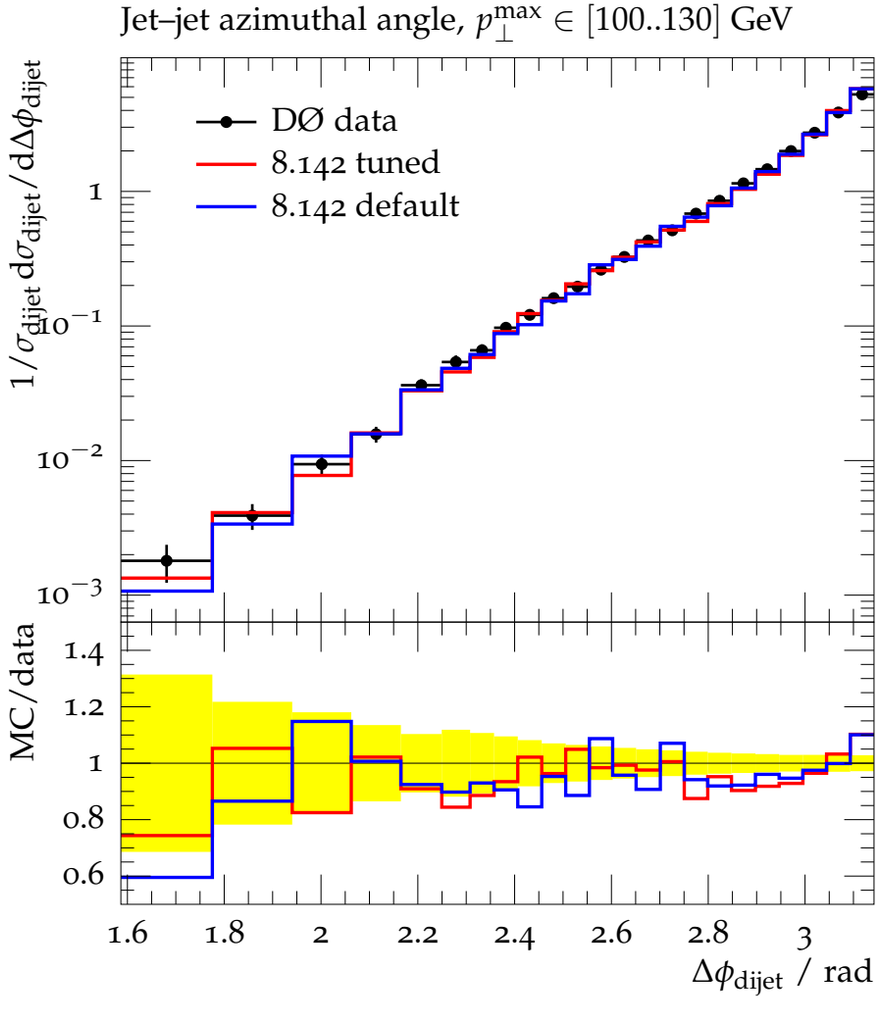
Pythia 8.142 – First Tuning Results





Minbias at 900 GeV, $n_{ch} \geq 2$

Pythia 8.142 – First Tuning Results



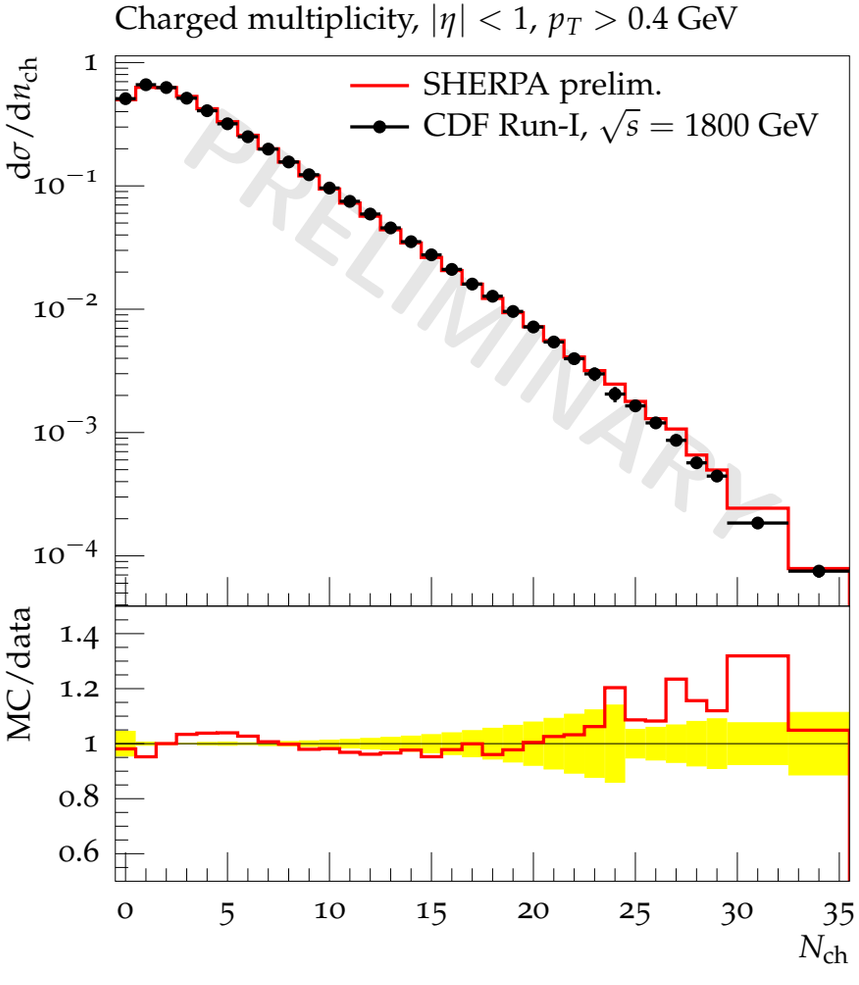
Pythia 8 is now also usable for UE.

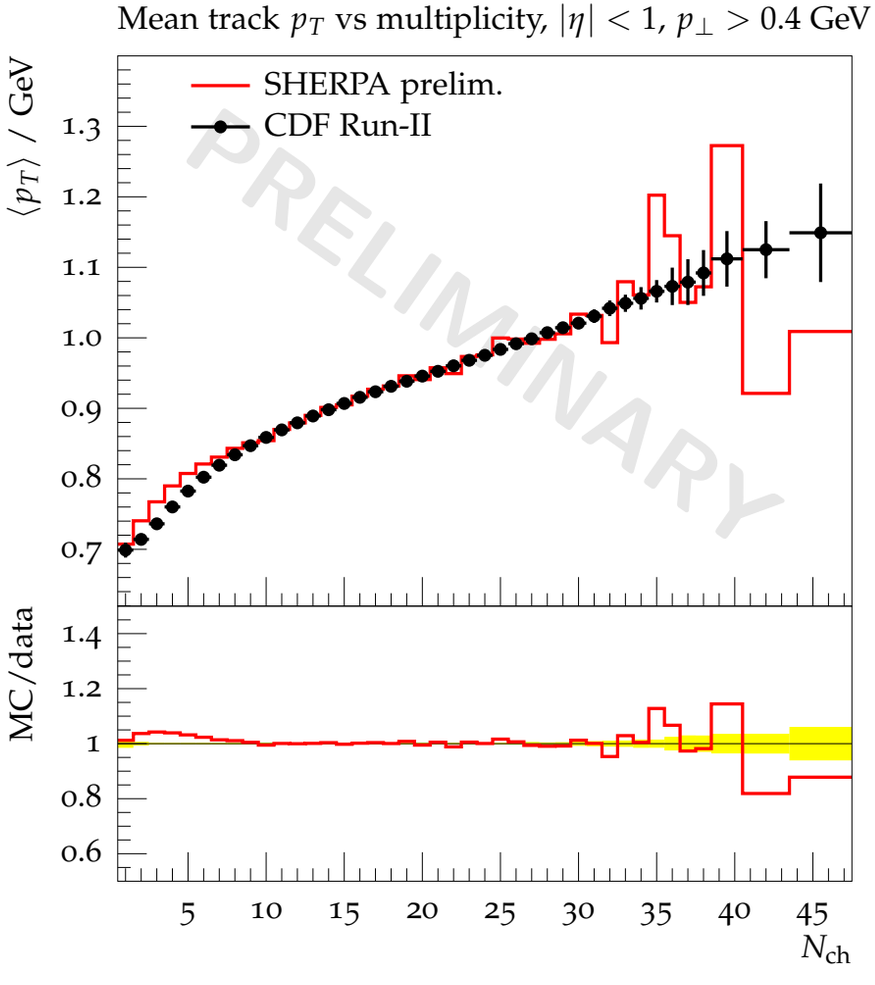
Expect the next release to be tuned.

Sherpa – Soft Physics Outlook

- KMR model being implemented for minbias / UE
- multi-channel eikonal approach
- BFKL-inspired interpretation: exchange of “ladders” (cut pomerons)
- Ladders described by evolution equations in rapidity y, with starting conditions at $y = \pm\infty$

Sherpa – Soft Physics Outlook





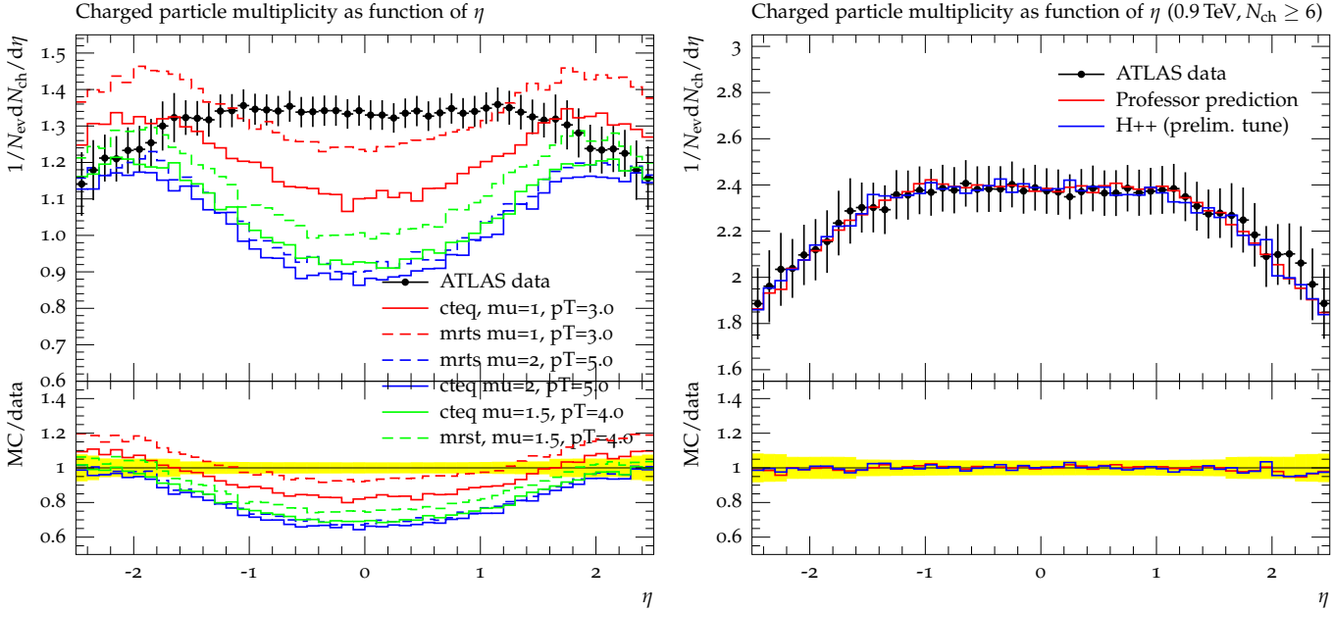
Herwig++ – New Developments for MB/UE

Shown by Andrzej Siódmok at the UE/MB working group meeting this month:

MinBias ATLAS 900 GeV

Preliminary results

(space for improvement)

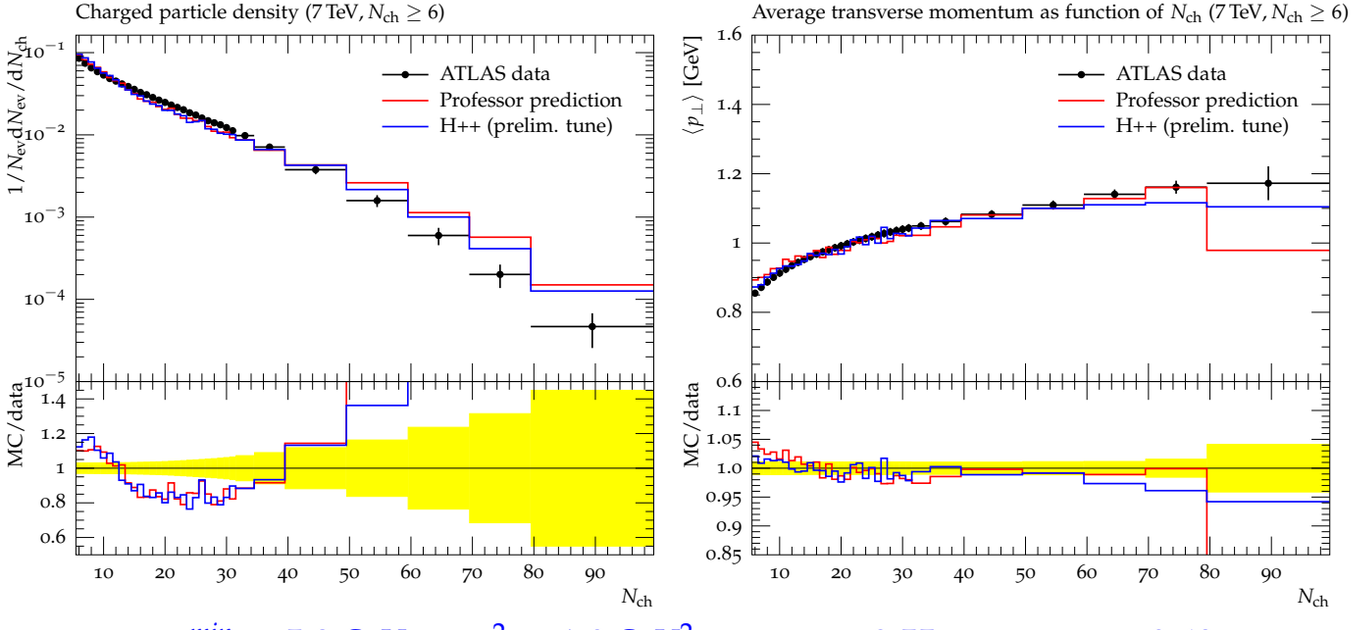


Herwig++ – New Developments for MB/UE

Shown by Andrzej Siódmok at the UE/MB working group meeting this month:

MinBias ATLAS 7000 GeV

Very preliminary results



$p_t^{min} = 5.2\,GeV\,, \quad \mu^2 = 1.8\,GeV^2, \quad p_{\text{reco}} = 0.55, \quad p_{\text{disrupt}} = 0.68$

Many thanks to the Professor team for help and hints how to use their program!
(Especially to Holger Schulz and Eike von Seggern)

Conclusions

- Monte Carlo is crucial for most analyses and we will need good MC if we want to find BSM physics.
- Systematic tuning also helps us to find bugs and improve the models
- There are lots of interesting LHC analyses for MC tuning around, but results are often uncorrected or not available in numeric format – we'll have to be patient.
- Tunings for the major MC generators are ongoing right now and will be reiterated as new data becomes available.

Backup slides

1. Choosing parameters

Pick the parameters you want to tune:

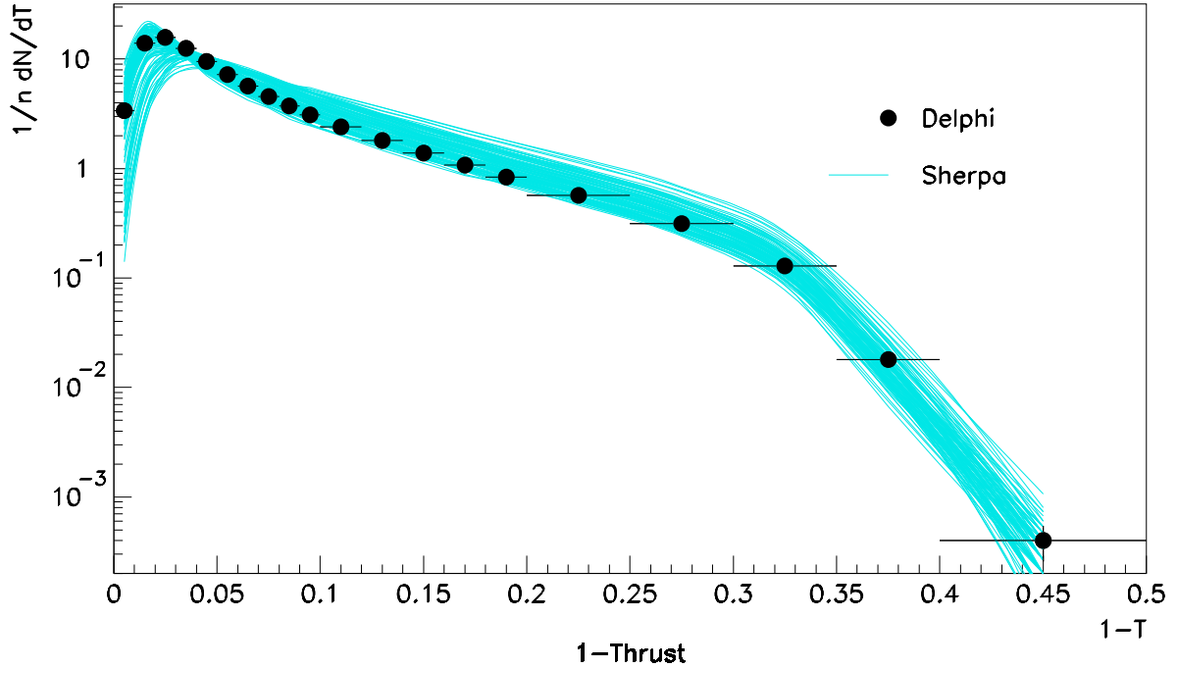
- Tune everything that is important.
- But remember: Each additional parameter adds one dimension to the parameter space.

Define parameter intervals:

- Make the interval large enough so that the result will not be outside.
- But remember: Cutting down 10 intervals by 10 % shrinks the volume of the parameter space by 2/3.

Now pick random points in parameter space and run the generator for each setting.

Calculating observables yields plots like this:



Every line corresponds to a certain parameter setting.

2. Predict the Monte Carlo

Get a bin by bin prediction for the MC response as function of the parameter set $\vec{p} = (p_1, p_2, \ldots, p_n)$.

Interpolate between the parameter points using a order polynomial:

$$MC^{(b)}(\vec{p}) \approx f^{(b)}(\vec{p}) = \alpha_0^{(b)} + \sum_i \beta_i^{(b)} p_i + \sum_{i \leq j} \gamma_{ij}^{(b)} p_i p_j$$

This takes the correlations between the parameters into account.

3. Fit the prediction to data

Using the interpolation we can predict the MC output for any set of parameters very fast. This prediction can be fitted to data, minimising the χ^2:

$$\chi^2(\vec{p}) = \sum_{\text{observables}} \sum_{\text{bins}} \frac{\left(X_{\text{data}} - X_{\text{MC}}(\vec{p})\right)^2}{\sigma^2_{\text{data}} + \sigma^2_{\text{MC}}}$$

Include all the relevant data distributions in the fit!

This fit only takes seconds or minutes (as compared to weeks or months for a brute force approach).

4.+5. Use different data sets, pick nicest tune

Now we approach the artistic part:

- Use different combinations of observables.
- Put different weights on the observables.
- Learn something about correlations and stability of the tuning.
- Interpret the results in the model's context.
- Maybe adjust/fix parameters by hand.
- Pick the nicest result.