# Review of MC event generator tuning tools and strategies

Andy Buckley
University of Edinburgh

MPI@LHC, Glasgow, 2010-12-01

# MC generators and MPI modelling

Event generators are a key tool in collider physics: extremely complex processes and detectors require very differential modelling. Event simulation that actually looks like real events $\Rightarrow$ design & test experiments

Most "generic" MPI modelling used by the LHC experiments is for minimum bias (pile-up) and and underlying event (associated QCD in signal processes).

Generators are the means by which MPI (and many other) physics ideas are realised. MPI/beam remnant interactions are a complex system, not amenable to analytic study.

# MC tuning

MPI physics is both non-perturbative and very dynamical: no factorisation theorem, *ab initio* modelling currently not an option. Pheno models used instead ⇒ parameters such as $p_T$ cutoff, energy evolution, colour-reconnection, ...

Tuning of soft QCD parameters required to constrain physics models *and* describe data!

Important to keep in mind a dichotomy: tuning is both for

- understanding/exploring the physics of soft QCD
- data mimicking for best experimental unfolding

Different MC programs have different philosophies of what can be tuned. Agreement on hadronisation and MPI, not so much on e.g. parton showers. But it's all linked.

# Tuning

- Particularly for PYTHIA, now have tens of tunes: maybe 5–10 currently viable ones. And that's just one model. **What to do with all these tunes?! And how to make more?**
- Lots of tunes = choice! But not all are equal, and lots of tunes implies lack of predictivity.
- **Tools now exist to construct and validate tunes and systematic variations.** E.g. multiple PDFs, error PDFs, scale variations, …
- Tuning is in its industrial revolution…expansion of possibilities, but still requires plenty of effort!
- Be picky: don't tune away physics, don't assume / insist on "silver bullet" generators/models

# Tuning methods

Lots of correlated parameters, 200k–10M events per run (kin. binning): tuning is non-trivial. Too slow for serial MCMC sampling approaches to be useful: **MC runs are "very expensive functions".**


You can do it manually, with intuition ⇒ Rick, I'm looking at you! But naturally limited to small datasets and numbers of params.

What I'm mainly going to talk about is a more automated approach, which widens our tuning horizons a little further.

**Rivet & Professor are the current standard tools for validation and tuning.** A pleasure to give this talk 2 years after first MPI@LHC, and see how quality, impact and acceptance have all increased! Also PROFFIT?

# Tools: Rivet

MC analysis system operating on HepMC events. Intentionally ignorant of what generator produced the events it sees.

Emphasis on not messing with the MC implementation details: actually reconstruct bosons, don't trace back partons, etc. Life is (eventually) simpler this way!

Lots of standard analyses built in, including key ones for pQCD and MPI model testing. New analyses can be picked up at runtime: nice API with lots of tools to make this as simple and pleasant as we can.

Please write Rivet analyses of your experiment's analyses and contribute them. ⇒ models tested and improved. So far only ATLAS has provided LHC analyses – but we'd love to have the ALICE/CMS MB and UE, and the LHCb/ALICE ID ratios.

**Latest version is 1.3.0.** Version 1.4.0 next week.

# Tools: Professor

**Professor** is a statistical tool which parameterises observable responses to changes in MC params. Allows rapid/accurate tuning because weeks of generation time can be simulated analytically (if the response is sufficiently smooth).

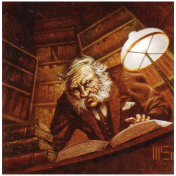
Can also play with “MC” interactively, cook up other fun things. . . I’ll mention much more of this!

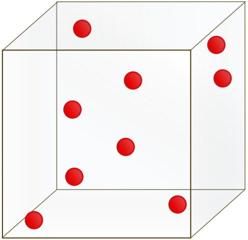
**Latest version is 1.0.0.** Version 1.0.1 next week.

Neither Rivet nor Professor are magic! Garbage in ⇒ garbage out.

# Professor

Method implemented as a Python package and set of scripts:



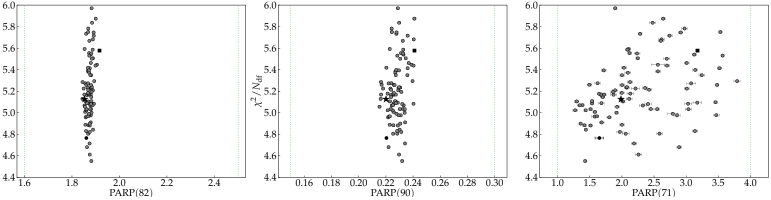


1. Sample $N$ random MC runs from $n$-param hypercube using e.g. Rivet
2. For each bin $b$ in each distribution, use the $N$ points to fit an interpolation function using a singular value decomposition.
3. Construct overall $\chi^2$ function and (numerically) minimise
4. Test optimised point by scanning around it in param and lin comb directions

Ask for details…or see the paper: arXiv:0907.2973

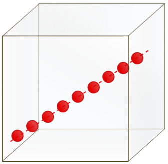
# Some tune param spreads

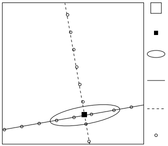
Oversampling required, but if we *really* oversample, then can make many combinations of input MC runs:

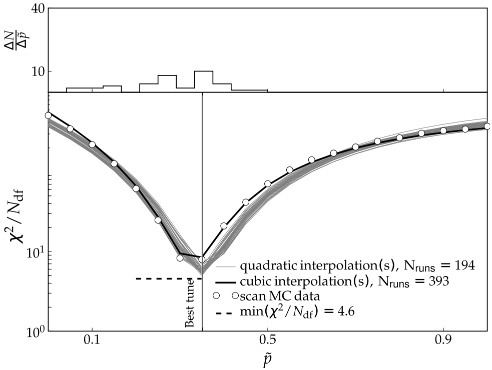



Gives an *informal* picture of how well-constrained (the projection of) a parameter is. Return to this later.

# Checking parameterisation: line-scans



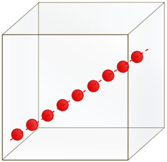


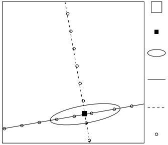


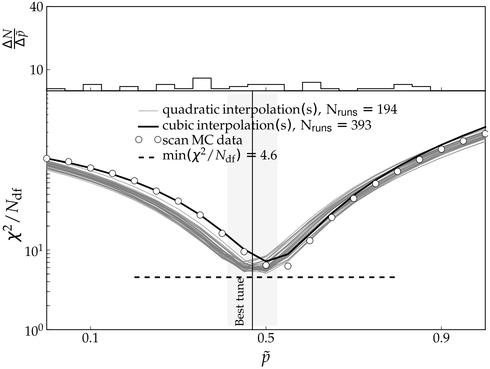

In practice, try to separate params into tractable semi-factorised blocks.

Difficult parameters/observables: Jacobian trf/metaparams possible. Extension idea: neural net parameterisation.

# Checking parameterisation: line-scans








In practice, try to separate params into tractable semi-factorised blocks.

Difficult parameters/observables: Jacobian trf/metaparams possible. Extension idea: neural net parameterisation.

# Interactivity

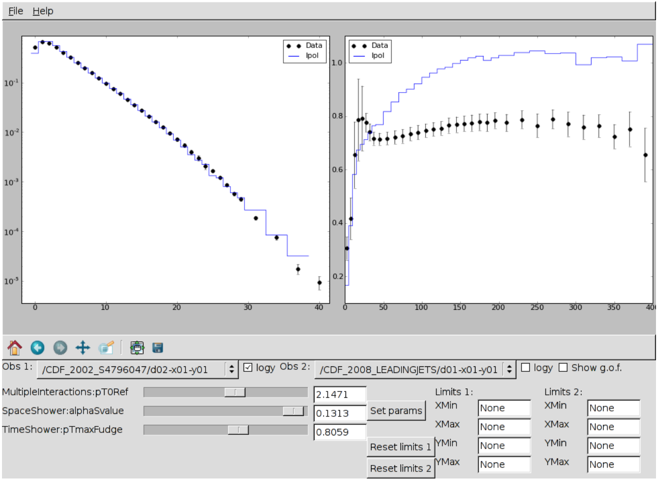
Two key features of Professor: **a)** we are parameterising a very expensive function, and **b)** the input to that parameterisation can be trivially parallelised.

[Prof parameterisation (for many, many run combinations) can also be parallelised, as can optimisation.]

So single-run MC produces a fast, analytic "pseudo-generator". Can get a good approximation of what a generator will do when run for many hours/days with particular params, in $< 1$ second!

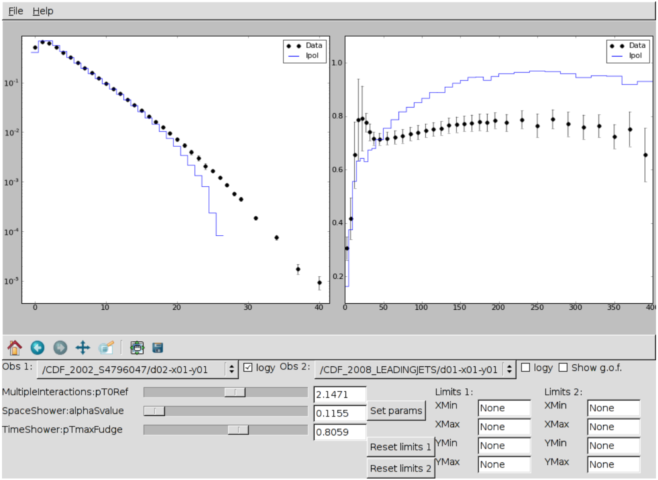
But these things are more general than optimising a tune: why not make an interactive MC simulator?

# prof-I

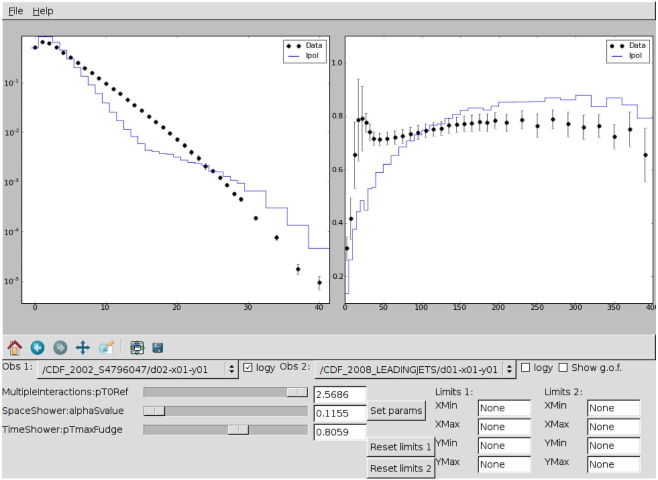


File Help
Data
Ipol
Data
Ipol
Obs 1: /CDF_2002_S4796047/d02-x01-y01
logy
Obs 2: /CDF_2008_LEADINGJETS/d01-x01-y01
logy
Show g.o.f.
MultipleInteractions:pT0Ref
2.1471
SpaceShower:alphaSvalue
0.1313
TimeShower:pTmaxFudge
0.8059
Set params
Reset limits 1
Reset limits 2
Limits 1:
XMin None
XMax None
YMin None
YMax None
Limits 2:
XMin None
XMax None
YMin None
YMax None

# prof-I


File Help
Data
Ipol
Data
Ipol
Obs 1: /CDF_2002_S4796047/d02-x01-y01
logy
Obs 2: /CDF_2008_LEADINGJETS/d01-x01-y01
logy
Show g.o.f.
MultipleInteractions:pT0Ref
2.1471
SpaceShower:alphaSvalue
0.1155
TimeShower:pTmaxFudge
0.8059
Set params
Reset limits 1
Reset limits 2
Limits 1:
XMin None
XMax None
YMin None
YMax None
Limits 2:
XMin None
XMax None
YMin None
YMax None

# prof-I


File Help
Data
Ipol
Data
Ipol
Obs 1: /CDF_2002_S4796047/d02-x01-y01
logy Obs 2: /CDF_2008_LEADINGJETS/d01-x01-y01
logy Show g.o.f.
MultipleInteractions:pT0Ref 2.5686
SpaceShower:alphaSvalue 0.1155
TimeShower:pTmaxFudge 0.8059
Set params
Reset limits 1
Reset limits 2
Limits 1:
XMin None
XMax None
YMin None
YMax None
Limits 2:
XMin None
XMax None
YMin None
YMax None

# Tuning of MPI

MPI tuning industry was started by Rick Field's manual CDF tunes of PYTHIA, from Tune A onwards. Still going, still PYTHIA!

Next developments were 2008 Professor tunes and Perugia tunes of PYTHIA from P. Skands.

Current state of the art (but I'm biased) is the Professor-based activity in ATLAS and MCnet: many generators in addition to PYTHIA, e.g. JIMMY, SHERPA, Herwig++, Pythia 8. I'll leave the MPI plots for the next talk.

# The standard tuning strategy

Our established strategy for tuning has been to assume that hadronisation is universal ⇒ tune it (and FSR) to LEP and other $e^+e^-$ data and take those params to hadron colliders. No *a priori* good ISR/MPI description required.

Then tune ISR parameters to hadron collider jet shapes, dijet decorrelation, etc. Take care re. $2 \to 2$ matrix elements. Minimise senstivity of observables to MPI / or *iterate*

Tune MPI params to MB and UE observables: final stage after the perturbative parts are well- and cleanly-constrained. Can we currently describe MB and UE at the same time? ATLAS data suggests not. **Can test by tuning to them separately and seeing how different the tunes are…or a cleverer method?**

# Tuning strategies for hadron collider hadronisation

LHCb/ALICE (and other) hadron collider identified particle distributions require breaking this simplified strategy. **Iteration needed ⇒ pain.** Tune the kinematics for hadron colliders as described, then return to baryon/antibaryon/strangeness params…and again and again? Can we semi-automate the iteration?

Can LHCb use central MPI tunes? Do MPI params influence ID particle ratios? Can use sensitivities from parameterisations to find out: $\mathrm{d}\ln \mathrm{MC}_b / \mathrm{d}\ln p_i$.

LHCb may require different tunes from other expts, to emphasise quality of $B$ and other meson simulation in forward rapidity. Forward MPI data will be very interesting: please submit it to Rivet!

Variation tunes, different PDFs require a degree of meta-systematisation: next stage for MC tuning?

# Assessing MC systematics

One good thing about all these tunes should be the potential for assessing MC systematics. But we're not quite yet systematic enough about systematics.

- What is the stat interpretation of difference between two discrete models?
- How far to change a factorization/renormalization scale to represent 68%?
- And which models to include, which are borderline reasonable, and which are just rubbish?

Tools like Professor can also help us to apply a bit more computational weight and statistical robustness to seeing how much variation is reasonable in a single model/tune.

# Covariance matrices for fun and profit

Having made a tune, there are corresponding errors provided by Minuit. However, these are just projected errors: varying tune params one at a time will be sub-optimal.

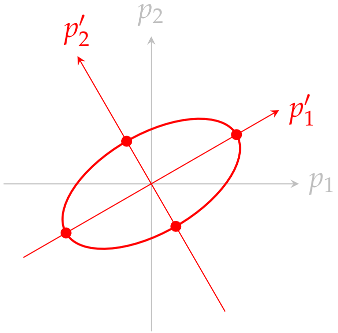
Should take account of parameter correlations, via covariance matrix.

The next few slides show a few ways which this can be done now, either just by using the covariance matrix to make alternative tunes, or by using it in conjunction with Professor fast MC parameterisation.

Need to develop re-weighting methods for tune variations cf. PDF errors?

# Eigentunes

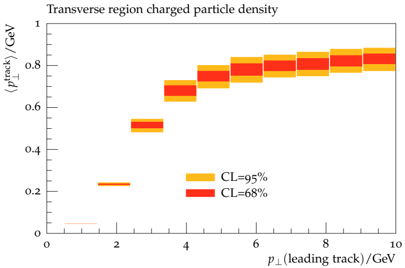
Pick the extremal points of the $\chi^2$ contour hyper-ellipsoid as representative tunes, cf. Hessian PDF errors.

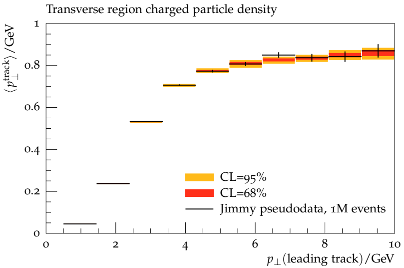



Such variations are only within the tuning "block diagonal" blocks. But sufficient for shower and hadronisation (and MPI) independently.

# Statistically-driven tune error bands

## Errors from run-combination sampling




Sample thousands of times via parameterised MC. Cf. NNPDF replicas.

Most complete procedure for full systematics in Les Houches proceedings (arXiv:1003.1643).

# Summary / some questions

LHC MC tuning is well-underway: new tools make it faster and more systematic. Über-systematisation requires logistics and planning.

Lots of potential still untapped: good models + good tunes means better control of experimental systematics. Should deprecate and abandon bad tunes:

- ALICE: PYTHIA ATLAS CSC tune!!
- CMS: PYTHIA D6T?
- LHCb: systematically studying parameter sensitivities of ID particle observables is needed to understand results.

Need to be careful *and* ruthless: not everything should be tuned! **Good tuning exposes model limitations:** systematic methods can give real confidence that model limitations are responsible for data/MC disagreement rather than a lack of tuning imagination. LHC data will drive model improvements!