Model tuning with Professor

Holger Schulz (IPPP Durham)

Lund, 3 July 2017

rivet.hepforge.org professor.hepforge.org













Event generators and Rivet

Data analyses



- arXiv:1407.0891[hep-ex]
- Analysis of ttbar events with ATLAS
- ► Fully corrected (unfolded) → compare with Truth level MC
- Data analysis is expensive in terms of computing and man-power
- How to compare the data with another model prediction?
- How to ensure we can do that also in 10 years?

Anatomy of a hadron collider event



Image: Marek Schönherr

HepMC

- HepMC::Version 2.06.09 HepMC::IO GenEvent-START EVENT LISTING E 0 -1 -1 0000000000000000 8 871071040596e 02 7 818608287725e 03 0 0 1 10001 10002 0 5 3 416101592648e+03 7 483422617301e - 08 3 416101592648e+03 1 20000000000e+01 0 N 5 "0" "1" "2" "3" "4" LLGEV MM C 2 846751327207e+02 2 846751327207e+02 E 2 21 4 008893662122e - 01 3 729543078303e - 02 9 782054163953e+02 1 532449558314e - 01 2 594332904812e+00 0 0 V = 1000022240P 10001 2212 0 0 3 999999889956e+03 4 0000000000e+03 9 382719993929e-01 2 0 0 -1 0 P 10002 2212 0.0 - 3 999999889956e+03 4 00000000000e+03 9 382719993929e - 01 2 0.0 - 1.0 11 P 10003 52 2 152701458984e+02 - 4 008098606740e+01 1 224843865257e+02 2 510978815160e+02 1 00000000000e+01 1 0 P 10004 - 52 - 1859496611203e+022683906048726e+022974023320540e+024417679711860e+02100000000000e+0110 P 10005 - 211 9.890420965096e - 03 - 7.232191998081e - 02 - 4.585734642526e - 01 4.848687322907e - 01 1.395700000000e -011P 10006 211 - 4.875521999232e - 02 - 6.391682129595e - 01 - 3.244942277751e+00 3.310595603025e+00 1.395700000000e P 10007 -211 -5.419747849145e -01 1.109603099151e+00 -5.358114740881e+00 5.500348085938e+00 1.39570000000e-01 P 10008 211 - 2.371586367548e - 01 9.913161177003e - 02 - 7.510349282885e - 01 8.059804860224e - 01 1.39570000000e - 01 1 P 10009 211 1.081239345290e+00 - 1.119198919259e+00 3.588261440166e-01 1.603097230116e+00 1.395700000000e-01 1.0 P 10010 - 2212 - 6.650888213379e - 02 7.369297556192e - 01 3.366876238277e+00 3.572631921424e+00 9.382720000000e P 10011 - 211 8.982907262553e+00 - 1.076204480402e+01 5.447163443754e+00 1.504012302540e+01 1.395699999998e-01 1 P 10012 211 1.093319138358e+00 - 1.292226110997e+00 6.619801755435e - 01 1.822880302695e+00 1.395700000000e - 01 1.0 P 10013 211 2 896119078340e - 02 - 2 408931921491e - 01 - 1 202631135574e+00 1 234775167287e+00 1 39570000000e P 10014 211 - 3.568764257915e - 01 - 2.361806061069e - 01 - 3.470102086736e+00 3.499175665676e+00 1.395700000000e - 01 1 P 10015 2112 - 2.003637330934e+01 - 2.382464016977e+01 1.759645813122e+02 1.786994303160e+02 9.395659999907e-01 1 P 10016 22 8.357442917026e - 01 - 2.581824372676e+00 1.269227052127e+01 1.297913774476e+01 - 2.384185791016e - 07 1.0
 - Particle 4 vectors
 - Vertices and genealogy
 - Common format for most event generators
 - Typically MB per event

Another look at an event graph A Pythia8 $t\bar{t}$ event



Most of this is not standardised: Other MCs generating the same physics look *very* different here. But final states and decay chains have to have equivalent meaning.

Another look at an event graph A Pythia8 $t\bar{t}$ event



Most of this is not standardised: Other MCs generating the same physics look *very* different here. But final states and decay chains have to have equivalent meaning. Rivet

- Analysis tool for MC events, generator agnostic via HepMC
- Provides most relevant methods for multi particle final states:
 - Cuts
 - Jets, jet shapes
 - Resonance finders
 - Event shapes
 - ...
- Writes out histogram (YODA format)
- External infrastructure: HepData, inspire
- Implementation and validation of new (data) analyses now largely provided by experiments
- Easy to write new analyses for signal and background estimates from MC



Rivet

- Recent big changes in LHC experiment/theory interaction
 - ⇒ more direct collaboration to improve methods and modelling, starting from SM & QCD, now also Top, Higgs, and BSM
- Rivet analysis library is part of this: a lightweight way to exchanging analysis details and ideas
- Implementing a Rivet analysis to complement the data analysis is increasingly expected of LHC analyses. Everyone benefits!
- One dedicated contact person in CMS and ATLAS



Rivet

Rivet is an analysis system for MC events, and *lots* of analyses

470 built-in, at today's count! 54 are pure MC, and some double/triple-counting

- Generator-agnostic for physics & pragmatics
- A quick, easy and powerful way to get physics plots from lots of MC gens
 - Only requirement: use HepMC event record
 - Usually via ASCII, but in-memory exchange is faster
- Rivet has become the LHC standard for archiving LHC data analyses
 - Focus on *unfolded* measurements, esp. QCD and EW+QCD, rather than searches
 - But there are BSM studies using it! And detector simulation now possible
 - Key input to MC validation and tuning increasingly comprehensive coverage
 - Also "recasting" of SM and BSM data results on to new/more general BSM model spaces
 - Add your analyses, too!



Analysis naming scheme



- ATLAS_2014_I1304688
- EXPERIMENT_YEAR_INSPIREKEY
- http: //inspire-hep.net/record/1304688
- This particular histogram is ATLAS_2014_I1304688/d02-x03-y01
- http://hepdata.cedar.ac.uk/view/ ins1304688, table 2, row 3

Design philosophy / pragmatics

Rivet operates on HepMC events, intentionally unaware of who made them...so don't "look inside" the event graph. ⇒ reconstruct resonances, dress leptons, avoid partons, etc.

This "hard work" way is actually simpler – fewer gotchas. Makes you think about physics & helps find analysis bugs/ambiguities

Tech stuff:

- C++ library with Python interface & scripts
- ► "Plugins" ⇒ write your analyses without needing to rebuild Rivet Trivial from user / analysis author point of view
- Tools to make "doing things properly" easy and default
- Computation caching for efficiency
- Histogram syncing: keep code clean and clear

+ helpful developers! New contributors always welcome

Basic principle



Basic principle



Basic principle



Output example

1	BEGIN YODA_HISTO1D /CMS_2015_I1384119/d01-x01-y01 IsBaf-1
3	$Path=/CMS_2015_11384119/d01 - x01 - v01$
	ScaledBy=1.00000000000000000 = -04
5	Title=
	Type=Histo1D
7	XLabel=
	YLabel=
9	# Mean: 2.047907e-03
	# Area: 2.128100e+01
11	# ID ID sumw sumw2 sumwx sumwx2 numEntries
	Total Total 2.128100e+01 2.128100e-03 4.358152e-02 2.926641e+01 212810
13	Underflow Underflow 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0
	Overflow Overflow 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0
15	# xlow xhigh sumw sumw2 sumwx sumwx2 numEntries
	-2.000000e+00 -1.800000e+00 1.105200e+00 1.105200e-04 -2.100061e+00 3.994160e+00 11052
17	-1.800000e+00 -1.600000e+00 1.111800e+00 1.111800e-04 -1.889834e+00 3.216089e+00 11118
	-1.600000e+00 -1.400000e+00 1.090400e+00 1.090400e -04 -1.636227e+00 2.458917e+00 10904
19	-1.400000e+00 -1.200000e+00 1.100500e+00 1.100500e-04 -1.430107e+00 1.862041e+00 11005
~ 1	-1.200000e+00 -1.000000e+00 1.0/4200e+00 1.0/4200e -04 -1.181831e+00 1.303/98e+00 10/42
21	-1.0000000+00 -8.0000000-011.063/000+001.063/000-04 -9.5/80329-018.6598950-011063/
22	-8.0000000e-01-6.000000e-011.048100e+001.048100e-04-7.341084e-013.176310e-0110481
23	- 0.0000000 - 01 - 4.0000000 - 01 1.0371000 - 00 - 03.1924200 - 01 2.0343070 - 01 10371
25	
2.5	$-2.000000 \pm 012000000 \pm 01100000 \pm 013.5100000 \pm 013.5100000 \pm 013.5100000 \pm 0123.547000000000000000000000000000000000000$
27	2.000000 = -01.4.0000000 = -01.9.9170000 = -01.9.9170000 = -05.2.981036 = -01.9.293782 = -02.9917
- /	4.00000e - 01 6.00000e - 01 1.020100e+00 1.020100e - 04 5.094963e - 01 2.578570e - 01 10201
29	6.000000e - 01 8.000000e - 01 1.045300e+00 1.045300e - 04 7.323970e - 01 5.166559e - 01 10453
	8.000000e-01 1.000000e+00 1.068300e+00 1.068300e-04 9.613896e-01 8.687205e-01 10683
31	1.000000e+00 1.200000e+00 1.076400e+00 1.076400e-04 1.184727e+00 1.307530e+00 10764
	1.200000e+00 1.400000e+00 1.093400e+00 1.093400e-04 1.422497e+00 1.854267e+00 10934
33	1.400000e+00 1.600000e+00 1.111300e+00 1.111300e-04 1.665988e+00 2.501220e+00 11113
1	1.600000e+00 1.800000e+00 1.121100e+00 1.121100e-04 1.904675e+00 3.239614e+00 11211
35	1.800000e+00 2.000000e+00 1.114100e+00 1.114100e-04 2.116399e+00 4.024084e+00 11141
	END YODA_HISTO1D

Getting Rivet

Rivet is readily available on the MCnet2017 virtual machine.

Easy to install using our *bootstrap script*:

wget http://rivet.hepforge.org/hg/bootstrap/raw-file/2.5.4/rivet-bootstrap bash rivet-bootstrap

Latest version is 2.5.4 Requires C++11

Docker image available: docker pull hepstore/rivet:2.5.4 http://rivet.hepforge.org/trac/wiki/Docker cvmFs installations on lxplus.

Getting Rivet

- rivet command line tool to query available analyses
- Can be used as a library (e.g. in big experiment software frameworks)
- Can also be used from the command line to read HepMC ASCII files/pipes: very convenient
- Helper scripts like rivet-mkanalysis, rivet-buildplugin
- Histogram comparisons, plot web albums, etc. very easy



Docs online at http://rivet.hepforge.org – PDF manual, HTML list of existing analyses, and Doxygen. Entries in HEPdata point to existing rivet analyses.

Writing an analysis

Writing an analysis is of course more involved. But the C++ interface is pretty friendly: most analyses are short, simple, and readable – details handled in the library + expressive API functions.

A single C++ file is sufficient. Rivet comes with scripts that generate analysis templates and compile the new code into a shared library (plugin).

Mostly "normal":

- Typical init/exec/fin structure
- Histogram titles, labels, etc.: use .plot file
- Rivet's own Particle, Jet and FourMomentum classes: some nice things like abseta() and abspid(), sorting and filtering

Multiple parton interactions (MPI)

Hadron collisions

 Colliding hadrons extended objects — only natural to expect more than one parton-parton collision in event



Images by Leif Lönnblad

Hadron collisions

 Colliding hadrons extended objects — only natural to expect more than one parton-parton collision in event



Images by Leif Lönnblad

MPI motivation



- Partonic cross-section exceeds total cross-section
- Dependence on PDF choice, \sqrt{s} and p_{\perp} cut-off

MPI impact



- Jet-shape observables, measure energy deposition in cone-shells of jets
- Blue: MPI off
- Green: MPI on

MPI measurement



arXiv:1701.05390

MPI modelling Sjöstrand-Zijl (PhysRevD.36.2019, 1987)

- Basic idea: number of additional scatters is function of $\frac{\sigma_{2\rightarrow 2}}{\sigma_{ND}}$
- $\sigma_{2\rightarrow 2}$ is a function of $p_{\perp,0}$
- $p_{\perp,0}$ modelled as: $A \cdot (\frac{\sqrt{s}}{B})^C$
- In addition: impact parameter (b) dependence due to hadronic matter overlap Õ_{DG}(b)
- Also: colour reconnection mechanisms a handle on competing hardness and multiplicity of produces particles



Model tuning with Professor

Tuning

- Realistic events contain physics at low scales where perturbation breaks down
- Rely on model assumptions that introduce many parameters
- Need to find "meaningful" settings
- Can be done manually but hard to on reasonable time-scale





Tuning with Professor in a nutshell

- ▶ Random sampling: *N* parameter points in *n*-dimensional space
- ▶ Run generator and fill histograms (e.g. Rivet) trivial parallel
- For each bin:
 - Don't care about actual dependence on parameters
 - Polynomial approximation per bin
- Construct overall (now trivial) $\chi^2(\vec{p}) \approx \sum_{bins} \frac{(D_b I_b(\vec{p}))^2}{error^2}$
- and numerically *minimise* with iminuit





1 bin example, 2 parameters (x,y), 2nd order polynomial

$$\mathrm{MC}_{b}(\vec{p}) \approx \alpha_{0}^{(b)} + \sum \beta_{i}^{(b)} p_{i}' + \sum_{i \leq j} \gamma_{ij}^{(b)} p_{i}' p_{j}'$$



$$\begin{split} c^{(\vec{b})} &= (\alpha, \beta_x, \beta_y, \gamma_{xx}, \gamma_{xy}, \gamma_{yy}) \\ \tilde{p}_i &= (1, x_i, y_i, x_i^2, x_i y_i, y_i^2) \\ \mathrm{MC}_b(\vec{p}) &\approx \sum_{i=1}^{N_{\min}(P)} c_i^{(b)} \, \tilde{p}_i \end{split}$$

1 bin example, 2 parameters (x,y), 2nd order polynomial





1 bin example, 2 parameters (x,y), 2nd order polynomial



- $\mathcal{I}[\tilde{P}]$ is the pseudo-inverse of \tilde{P}
- ▶ $\mathcal{I}[\tilde{P}]$ is calculated using singular value decomposition (SVD)
- SVD is least-squares fit
- We need at least as many \tilde{p}_i as there are coefficients

• With $c_i^{(b)}$ calculated \rightarrow prediction

$$\mathrm{MC}_{b}(\vec{p}) \approx \sum_{i=1}^{N_{\mathrm{min}}(P)} c_{i}^{(b)} \, \tilde{p}_{i}$$

for any \vec{p} in *milliseconds*

Separate polynomials for central value and uncertainty of a bin

Professor technicalities

- ► C++ core functionality, python bindings for everything else
- In case of MC, input generation trivial to do in parallel (different points in parameter space)
- Result is fast analytic pseudo-generator
- Storage of coefficients as plain text file → can use parameterisation in other C++ codes



Example output

Plain text file

Meta information, parameter space, polynomial coefficients

DataDir: /home/hschulz/Sherpa/Tuning-NewHad-rel-2-2-4/2017-05-22-bla/scan2 ProfVersion: 2.2.2beta2 Date: 2017-06-01 13:10:01 DataFormat: binned 2 ParamNames: KT_O ALPHA_D ALPHA_L ALPHA_H GAMMA_D GAMMA_L GAMMA_H STRANGE_FRACTION BARYON_FRACTION Dimension: 9 MinParamVals: 0.000674 - 0.497223 - 0.499586 0.000757 0.000580 0.002653 0.504251 0.250288 0.250298 MaxParamVals: 2.248679 1.998837 1.997479 3.999912 1.998901 1.998744 7.999943 0.749977 0.749808 DoParamScaling: 1 10 NumInputs: 1484 /ALEPH 1991 S2435284/d01-x01-v01#0 1.00000e+00 3.00000e+00 val: 9.2 - 0.00360025 - 0.00100512.0.000544102.0.000296846.0.000213943 - 6.35394e - 0.6 - 0.000699236 - 0.00021037 0 000510953 0 000588548 0 00654176 - 0 000914022 - 0 000716218 0 000180158 - 0 00333635 0 00132221 - 4 3627e - 05 -0.000422647 -0.000267011 0.000627989 -0.000548664 0.00176204 0.000364814 -0.00053499 0.000451779 -0.000277334 -0.000844911 -0.00125288.0.0197997 -0.000322684.0.00048657.0.000335561.0.00748714.0.000241382 -0.000162675 -0.0185692 0.00166507 0.000238913 -0.000310697 -0.000593107 -0.000290523 -0.000245745 -3 22819e-05 0 000135187 -0 00107326 -0 000518321 0 00102011 0 000345511 -6 48719e-05 -0 00151553 -7.40607e-05 0.000112481 -0.00291144 0.000185875 2.37507e-05 err: 9.2.0.000181519 - 5.02472e - 05.8.08095e - 06.0.00013151.3.28828e - 05.1.61701e - 05.0.000153176.8.04158e - 05.1.87033e -05-7.41364e-05 6.47977e-05 0.000105081 -0.00018051 -0.00010014 -7.04912e-05 -5.71483e-05 -2.17563e -05 6 00403e-06 - 1 02219e-05 - 3 50495e-05 - 1 21488e-05 3 6774e-05 4 73355e-05 - 5 82478e-05 - 7 81474 e-05 -2.7055e-05 2.44168e-05 -0.000104596 -0.000124106 -4.96546e-05 -2.92521e-05 1.82974e-05 9.29349 e-05 9.0611e-05 7.48066e-05 -2.10497e-05 -4.00846e-05 -3.43163e-05 2.72302e-05 -8.66645e-05 5.17795 e-05 5.00373e-06 0.000117281 -1.30487e-05 4.24917e-05 0.000133045 -5.13495e-06 -0.000112468 -1.16532e -05 3 84505e -05 5 59006e -05 6 78926e -05 9 02897e -05 -1 57199e -05 -0 000171531

Sampling

- Define parameter space with input textfile
- Can bias sampling to avoid say unphysical parameter space
- Convenient template instantiation for e.g. generator steering cards
- Random uniform or Sobol sampling
- So far no clear preference



Residuals

- SVD is a least squares fit through points \vec{p}
- For each of those we know the exact value $MC(\vec{p})$
- ► Define residual as distance between the two: res = $[I(\vec{p}) - MC(\vec{p})] / I(\vec{p})$
- ▶ Put all res into histogram, expect something symmetrical around 0



Jackknifing

- Separate input data into training and test sample
- Run parameterisation for *training* sample
- Calculate pull

$$\langle \text{pull} \rangle = \frac{1}{N_{\text{testpoints}}} \sum_{\vec{p} \in \text{testsample}} \frac{I(\vec{p}) - \text{MC}(\vec{p})}{I(\vec{p})}$$

- Shows which polynomial order is best suited
- Also shows limitation of polynomials when exceeding machine precision



Left: Input data generated from 3rd order polynomial Right: Input data generated from 9th order polynomial

Weights and choice of observables

- > Tuning: reasonable measure between model prediction and data
- ▶ By default, ad-hoc "chi square" inspired goodness-of-fit

$$\chi^2(\vec{p}) = \sum_{\mathcal{O}} \sum_{b \in \mathcal{O}} w_b \cdot \frac{(f^{(b)}(\vec{p}) - \mathcal{R}_b)^2}{\Delta_b^2(\vec{p})}$$

- ▶ Weights *w*^{*b*} neccessary because models not perfect:
 - Exclude regions of observables (e.g. bad coverage, breakdown of polynomial approximation)
 - Force good description of certain observables (at the cost of others)



Sensitivities

- With polynomials at hand, gradient always known
- Can be used to calculate a measure of sensitivity
- Helps identify most important observables/parameters



Interactive explorer

- ▶ GTK application to interactively "play" with parameterisation
- One slider per parameter, moving them redraws histograms
- Good for intuition building
- Running the event generator would require a few hours wait



Interactive explorer

- ▶ GTK application to interactively "play" with parameterisation
- One slider per parameter, moving them redraws histograms
- Good for intuition building
- Running the event generator would require a few hours wait



Interactive explorer

- ▶ GTK application to interactively "play" with parameterisation
- One slider per parameter, moving them redraws histograms
- Good for intuition building
- Running the event generator would require a few hours wait



Minimisation

- ► Tuning is numerical minimisation of goodness-of-fit measure
- ▶ We use iminuit as it is a flexible python wrapper for Minuit
- Input to tuning stage is parameterisation file, text file with weights, directory with data files

1	/ATLAS_2010_S8918562/d03-x01-y011 # Set weight to 1 for each bin of this histo
	/ATLAS_2010_S8918562/d05-x01-y01100 # Set weight to 100 for each bin of this histo
3	/ATLAS_2010_S8918562/d07-x01-y01@0:20 10 # Set weight to 10 for bins with binEDGES in [0,20)
	/ATLAS_2010_S8918562/d07 - x01 - y01@20:40 50 # Set weight to 10 for bins with binEDGES in [20,50)
5	/TOTEM_2012_I1115294/d01 - x01 - y01#0:20 10 # Set weight to 10 for bins with binINCIDES in [0,20]
	/TOTEM_2012_I1115294/d01-x01-y01#20:40 50 # Set weight to 10 for bins with binINDICES in [20,50)

- ► Output:
 - Text file with minimisation result, covariance matrix etc.
 - File with histograms calculated from parameterisation at this minimum
- Quick turnaround, minimisation seconds to minutes, plots comparing with data seconds
- Usually iterative procedure, look at plots, adjust weights

Uncertainties



- Can exploit minimiser covariance of parameters, C_{xy}
- Get principal directions, similar to PDF fit uncertainties
- "Eigentunes"
- But: what Δ goodness-of-fit since not really $\chi 2$?

Goodness-of-fit distribution

- Need to obtain distribution of goodness-of fit measure to find e.g. 68% percentile
- ► We use the parameterisation of the input uncertainties and central values to smear inputs MC(p)
- ► Say 1000 smeared clones of the input data → 1000 slightly different parameterisations
- ▶ Do 1000 minimisations using the *same* weights (+ smear data)
- Histogram resulting goodness-of-fit values $\rightarrow \Delta gof$
- ▶ Plots: 3 parameters, 20 bins \rightarrow clearly no χ^2 ! (less than 10 minutes)



Error propagation

- The Eigentune points allow us to determine a modified covariance matrix C
 xy
- Gradients of polynomials $\vec{\nabla} I(\vec{p})$ always known in Professor

• Can do error propagation of \tilde{C}_{xy} :

 $C_{ij} = \vec{\nabla} I(\vec{p}_0) \; \tilde{C}_{xy} \; (\vec{\nabla} I(\vec{p}_0))^{\mathrm{T}}$

• Resulting in uncertainty for bin *i* as $\sqrt{C_{ii}}$

Example uncertainties





- Eigentunes for all three principal directions
- Contributions differ over observable range
- Propagated uncertainties shown as well

Getting Professor

Beta version of Prof 2.2.2 readily available on MCnet2017 VM

- ▶ Prerequisits: Eigen3 headers, C++ 11 compiler, Python 2.7
- professor.hepforge.org
- Docker image: docker pull iamholger/professor:2.2.1

Running Professor

Each stage of the tuning problem has its own script:

- prof2-ncoeffs lists number of coefficients required
- prof2-sample sample a parameter space, write run templates
- prof2-envelopes make envelope plots of all MC inputs
- prof2-ipol run the parameterisation, write to file
- prof2-1s lists contents of parameterisation file
- prof2-residuals tests parameterisation against its own inputs
- prof2-sens calculate sensitivity plots
- prof2-I interactive parameter explorer
- prof2-tune the minimisation stage

Note: we typically use **rivet** for plotting the output of the minimisation, i.e. **rivet-mkhtml**.

Professor beyond tuning

- Instead of fiddling with say hadronisation model parameters, explore BSM parameter space
- Lots of experience can be transferred from tuning to BSM



Professor beyond collider physics

- Started collaboration with neutrino MC community, Genie
- Triggered containerisation of Professor with Docker
- Similarly, Dark Matter direct detection codes: Professor in likelihood evaluation (MultiNest)



Next design goals for Professor

- Automatic checks of validity of polynomial approximation.
 - Partitioning of parameter space in case parameter space too big
 - Avoid overfitting
 - Need to allow to drop inputs in case of vanishing cross-sections (avoid discontinuities in polynomial fit)
 - Move away from histogram picture in storage
- Usage of other parameterisations, e.g. Gaussian Processes for $\frac{1}{x}$, *Pade'* approximation
- Finalise work on uncertainties, usage of pymultinest

Summary

Rivet:

- User-friendly MC analysis system for prototyping and preserving data analyses
- Also a very useful cross-check: quite a few ATLAS analysis bugs have been found via Rivet!
- Integrated with ATLAS and CMS software
- Now supports detector simulation for BSM search preservation
- Professor:
 - Parametrisation of computationally expensive functions
 - Inputs can always be parallelised in a trivial way
 - Seamless integration into numerical tools iminuit, pymultinest through python bindings → tuning and BSM applications
 - Robust estimation and propagation of tuning uncertainties available very soon