YETI Contur (+Rivet) tutorial

David Yallup

Durham Jan 2019

Environment setup

- Based on Docker, intended to run on local machine for ease, should be fairly standard obtainable via package manager,
  - https://docs.docker.com/install (docker-ce package on Ubuntu & Fedora)
    Maybe necessary:
    > sudo groupadd docker
    > sudo usermod -aG docker $USER
- Install Tutorial docker image
  - > docker pull dyallup/contur
  - Needs ~2GB of space
- Helpful commands
  - docker run: run an image
    docker ps: list active containers
    docker cp: copy files between host and container
    docker image ls: list available images/apps
    docker help: useful!
  - Install new system packages with dnf on the images (e.g. if you like emacs, > dnf install emacs within the image. Exit with Ctrl-d; detach with Ctrl-p Ctrl-q, reattach with "docker attach"

Environment setup

- You can work with docker however you like, these are my steps
- Make a directory in your filesystem, so we can mount this to easily get things in and out (in my case ~/YETITutorial/runArea)
- > docker run -v ~/YETITutorial/runArea:/runArea -it dyallup/contur bash
  - This runs the container and mounts the directory inside the container at the path /runArea
  - >dnf install xyz can be used (as one would use any other package manager), I already put emacs, vi and nano in
  - Ctrl+p ctrl+q to detach, >docker ps to list containers, >docker attach xyz to reattach
  - You should start in the contur directory, to test things are working run the commands below
  - > . setupContur.sh ; > make ; > Herwig --version

```
[root@5366b3e66366 contur]# ls
AnalysisTools  Makefile  Models  README.md  herwigPath.sh  modified_analyses  setupContur.sh
[root@5366b3e66366 contur]# . setupContur.sh
[root@5366b3e66366 contur]# make
rm -f AnalysisTools/contur/contur/TestingFunctions/analyses.db
sqlite3 AnalysisTools/contur/contur/TestingFunctions/analyses.db < AnalysisTools/contur/contur/TestingFunctions/analyses.sql
modified_analyses/Analyses/buildrivet.sh
/usr/bin/g++ -o "Rivet-ConturOverload.so" -shared -fPIC  -I/usr/local/include -I/usr/local/include -I/usr/local/include -I/usr/local/include   -pedantic -Wall -Wno-long-long -Wno-format -Wer
ror=uninitialized -Werror=delete-non-virtual-dtor -fopenmp -O2   -Wl,--no-as-needed  -L/usr/local/lib -L/usr/local/lib -L/usr/local/lib -Wl,-rpath,/usr/local/lib -lm -L/usr/local/lib -lfastj
ettools -lfastjet -lfastjetplugins -lsiscone_spherical -lsiscone  -lRivet    CMS_2013_I1256943.cc  --std=c++11
make[1]: Entering directory '/contur/modified_analyses/Analyses'
make[1]: Leaving directory '/contur/modified_analyses/Analyses'
[root@5366b3e66366 contur]# Herwig --version
Herwig 7.1.4
ThePEG 2.1.4
```

Container

Environment setup

- Now switch to the directory we mounted
- > cd ../runArea
- Clone the git repository (maybe preferable to do this from your host pc) into the linked runArea folder
- https://bitbucket.org/dyallup/conturtutorial/src/master/
  - > git clone https://bitbucket.org/dyallup/conturtutorial.git runArea
- Check you can see the files and open them in your desired console text editor inside the container

Exercise 1 - Messing around with generators

- Inside the Exercise_1 folder have a couple of files, let's just run first to make sure things work (Inside the container you should be able to see the following)
  - > Herwig read Dijets.in
  - > Herwig run Dijets.run -N 10 (-N requests N events to be generated)

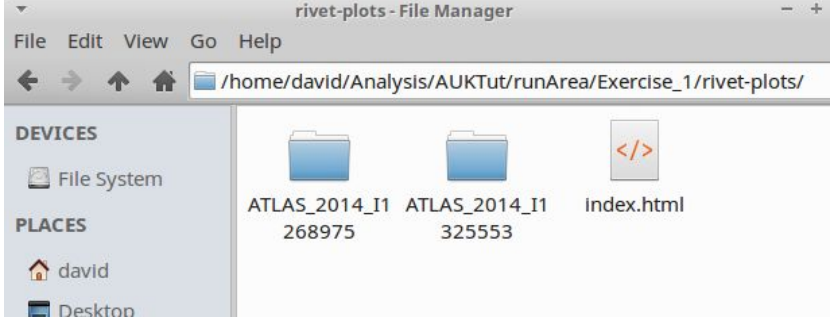
```
[root@ed3829d8c850 Exercise_1]# ls
Dijets.in  Rivet.ana
[root@ed3829d8c850 Exercise_1]# Herwig read Dijets.in
[root@ed3829d8c850 Exercise_1]# Herwig run Dijets.run -N 10
event>       10       10

BasicConsistency: maximum 4-momentum violation: 0.00991477 MeV

HepMCFile: generated HepMC output.

Dijets.hepmc  Dijets.in  Dijets.log  Dijets.out  Dijets.run  Dijets.tex  Rivet.ana
[root@ed3829d8c850 Exercise_1]# ls -lth
-rw-r--r-- 1 root root 1.6M Jan  3 18:08 Dijets.hepmc
-rw-rw-r-- 1 1000 1000 1.7K Jan  3 18:07 Dijets.in
-rw-r--r-- 1 root root 1.1M Jan  3 18:08 Dijets.log
-rw-r--r-- 1 root root  33K Jan  3 18:08 Dijets.out
-rw-r--r-- 1 root root 6.7M Jan  3 18:08 Dijets.run
-rw-r--r-- 1 root root 7.0K Jan  3 18:08 Dijets.tex
-rw-rw-r-- 1 1000 1000  103 Jan  3 18:07 Rivet.ana
```

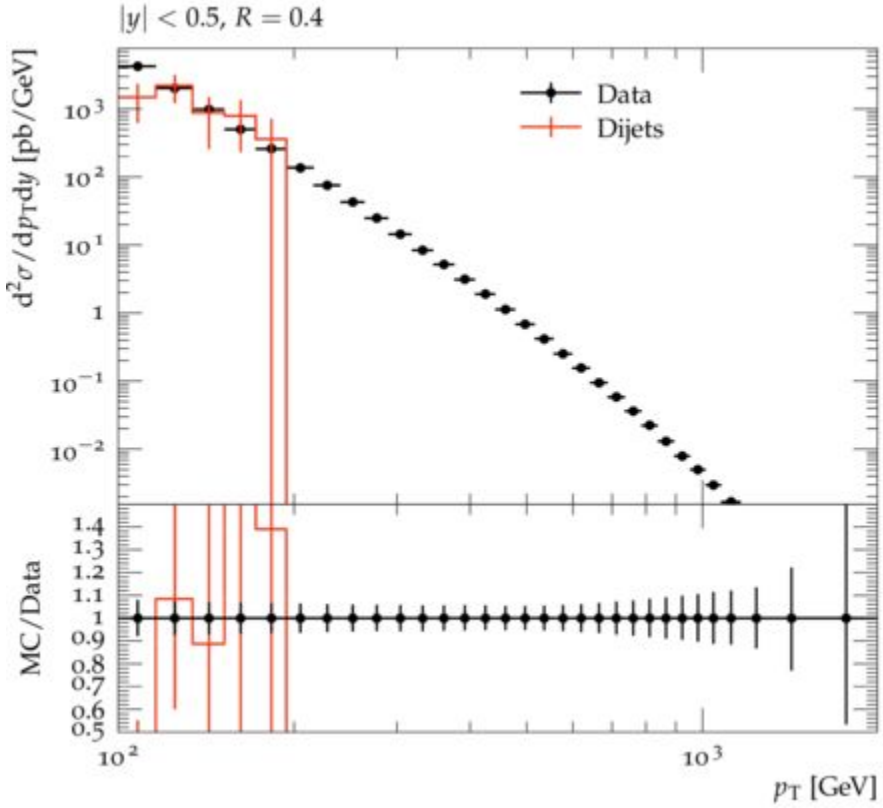
HepMC makes large files (this is the agreed LHC Event generator language for events). Best to pipe them directly to analysis software (Rivet as we are about to do)

Exercise 1 - Messing around with generators

- Edit the Dijets.in file
  - Comment out the HepMC block and uncomment the rivet block
  - Repeat the previous read/run steps but this time request 1000 events (we no longer have to worry about disk space as we won't save the hepmc), so previously we ran 10 only
  - > Herwig run Dijets.run -N 1000
- Now we output a Dijets.yoda file
  - > rivet-mkhtml Dijets.yoda --mc-errs
    - Default plotting script: rivet-mkhtml
    - --mc-errs requests stat errors be drawn on the plots
- Navigate to your linked directory on the host PC and open index.html

```
[root@ed3829d8c850 Exercise_1]# Herwig read Dijets.in
[root@ed3829d8c850 Exercise_1]# Herwig run Dijets.run -N 1000
event>     1000     1000

BasicConsistency: maximum 4-momentum violation: 0.0463695 MeV

[root@ed3829d8c850 Exercise_1]# rivet-mkhtml Dijets.yoda --mc-errs
Making 24 plots
Plotting ./rivet-plots/ATLAS_2014_I1325553/d01-x01-y01.dat (24/24 remaining)
Plotting ./rivet-plots/ATLAS_2014_I1325553/d02-x01-y01.dat (23/24 remaining)
Plotting ./rivet-plots/ATLAS_2014_I1325553/d03-x01-y01.dat (22/24 remaining)
Plotting ./rivet-plots/ATLAS_2014_I1325553/d04-x01-y01.dat (21/24 remaining)
```



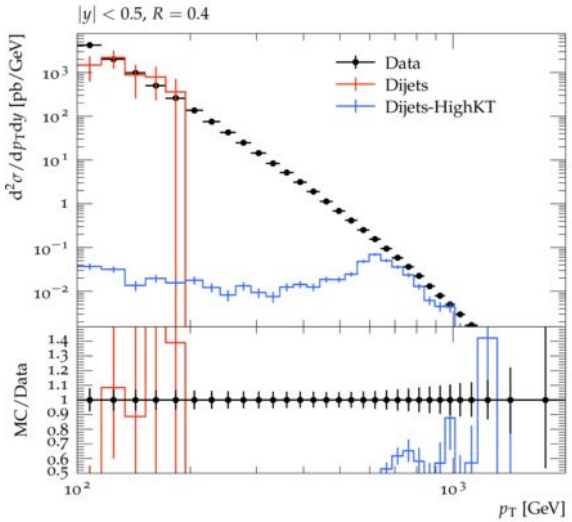

$|y| < 0.5$, $R = 0.4$
Data
Dijets
$d^2\sigma/dp_T dy$ [pb/GeV]
MC/Data
p_T [GeV]

Exercise 1 - Messing around with generators

- Edit the Dijets.in file
  - This time edit the line in the cuts block
    - > set /Herwig/Cuts/JetKtCut:MinKT 600.
  - Repeat the previous read/run steps but this time tag the run with a different name so we don't overwrite our old output (-t CL argument)
  - > Herwig run Dijets.run -N 1000 -t HighKT
- Now we can plot both Yoda files
  - > rivet-mkhtml Dijets.yoda Dijets-HighKT.yoda --mc-errs
  - Outputs a rivet-plots folder, look at that from your host pc
    - Open the index.html file in your browser

This is motivating a bit how tricky it can be to get a good prediction over such a huge kinematic range

Hopefully this gives a little impression of how the generators work and some basic manipulation with Rivet



Exercise 1 - Messing around with generators

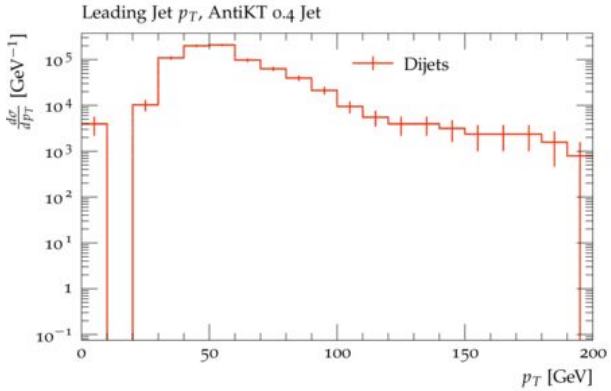
A couple of notes

- We used the Herwig native Rivet interface
  - Look at the Rivet.ana file, and the corresponding line in Dijets.in
- This is not necessary, can run on any static HepMC file or generator that outputs HepMC
  - rivet.hepforge.org/trac/wiki/FirstRivetRun
- Find other standard plugins that may be of interest and try including them if you want
  - rivet.hepforge.org/analyses.html
- These plugins can be filled with the best possible SM event generation and be seen to agree across the board, Find the ones we just ran on the Herwig page and compare
  - https://herwig.hepforge.org/plots/herwig7.0/Rivet-LHC-Jets/index.html

Exercise 2 - Custom plugins

Much of the same basic premise as Exercise 1, now we add a custom plugin

- > source sourceRivet.sh
  - Sets environment variables so rivet knows where our custom plugin is
- > rivet-buildplugin MC_TUTORIAL.cc
  - Build our custom plugin
- Herwig read/run Dijets.in as before, rivet-mkhtml Dijets.yoda as before, view plots as before



Exercise 2 - Custom plugins

Look inside the MC_TUTORIAL.cc source code, 3 main component blocks to this analysis class

- Init
  - Things we want to run once at the beginning of a run, declare projections (object definitions to cast out of events), book histograms to fill etc.
- Analyze
  - The main event loop, the generator feeds events in one by one, this block performs the operations requested on each event
- Finalize
  - Things we want to run once at the end, normalising the filled histos etc.

(plus a little extra bit of pointer declaration etc. sprinkled around)

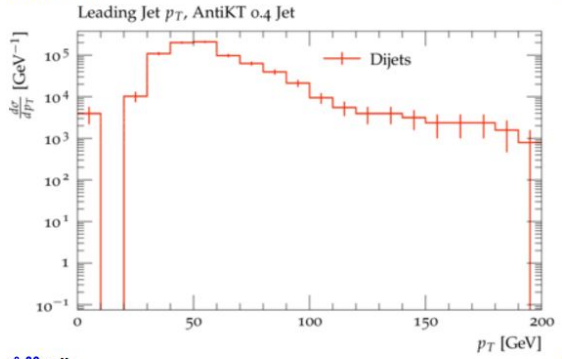
Exercise 2 - Custom plugins

1. Try and add a new variable to plot, to begin with lets say 2nd jet pt (can copy most of what is in there for plotting leading jet pt)
   a. Remember to build the plugin again if you modify the c++ source code and repeat the generator run
   b. Why does the plot style look different? Look at the .plot file and match the path to the .yoda file
2. Try a more complicated variable, say mjj
   a. We are playing with jets so the rivet code documentation on the members of the jet class is helpful here: rivet.hepforge.org/code/dev/classRivet_1_1Jet.html
   b. The library of existing plugins we were using before gives good inspiration of how to calculate things, for example we can look at the dijet mass routine we included before for inspiration: rivet.hepforge.org/analyses/ATLAS_2014_I1268975.html
3. Try and add a new projection to do a smeared version of the leading jet pt (approximate the detector and make a “reco level” version)
   a. Again I just take inspiration from an existing plugin that utilises such things, for example: rivet.hepforge.org/analyses/ATLAS_2016_CONF_2016_094.html

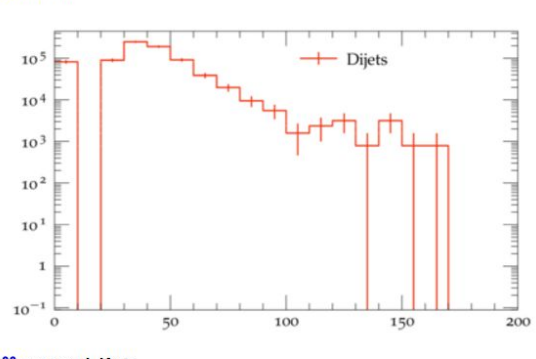
Exercise 2 - Custom plugins

- A worked example with this custom plugin dressed up is in the solutions sub directory, so refer to that for hints too!

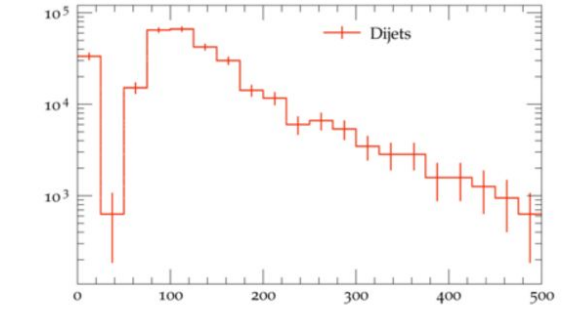
⚓⌘ **jet1pt:**



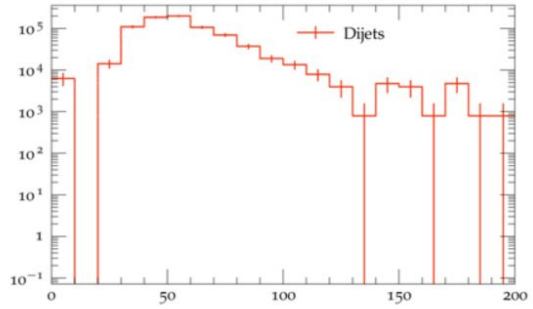
⚓⌘ **jet2pt:**



⚓⌘ **mjj:**

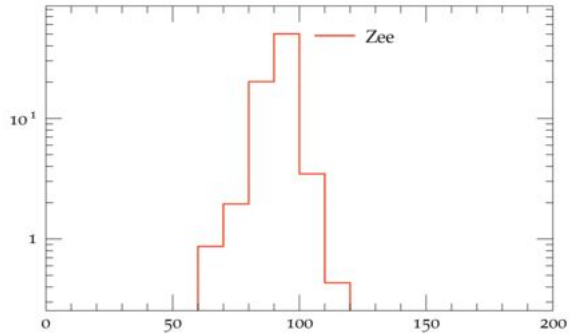


⚓⌘ **smeared_j1pt:**



Exercise 3 - Start from scratch

- The Exercise 3 folder contains a different generator, this time Zee
- Try building the whole set from scratch
  - > rivet-buildplugin MC_ZEE
  - This generates a boilerplate analysis template
- Use the previous examples, remembering bits like the sourceRivet step, try and make a plot of the Z Mass peak
- Again Doxygen (hint ZFinder class…, or use electrons by pT) or existing plugins
- Worked example again in solutions subfolder



Exercise 4 - Starting with Contur

- The Exercise 4 folder contains a UFO model file
  - http://feynrules.irmp.ucl.ac.be/wiki/ModelDatabaseMainPage
  - FR Tutorial at YETI 2016
    - Might have a more recent version, not covered here
- This is a generator independent BSM format, output from Feynrules, we have to read this into a language Herwig understands
  - > ufo2herwig DM_ vector_mediator_HF_UFO
  - > make

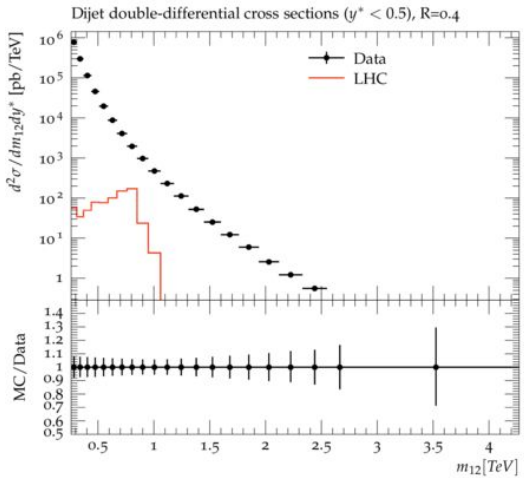
```
[root@d22b1de2392f Exercise_4]# ls
DM_vector_mediator_HF_UFO  LHC.in  Rivet.ana
[root@d22b1de2392f Exercise_4]# ufo2herwig DM_vector_mediator_HF_UFO/
============================================================
LENGTH 1
finished generating model:      FRModel
model directory:                DM_vector_mediator_HF_UFO/
generated:                      126 vertices
============================================================
library:                        FRModel.so
input file:                     LHC-FRModel.in
model file:                     FRModel.model
============================================================
To complete the installation, compile by typing "make".
An example input file is provided as LHC-FRModel.in,
you'll need to change the required particles in there.

DONE!
============================================================
[root@d22b1de2392f Exercise_4]# make
g++ -fPIC -I/usr/local/include -I/usr/local/include  -Wall -Wextra -pedantic -O2 -c FRModel.cc -o FRModel.o
g++ -fPIC -I/usr/local/include -I/usr/local/include  -Wall -Wextra -pedantic -O2 -c FRModel_Vertices_004.cc -o FRModel_Vertices_004.o
g++ -fPIC -I/usr/local/include -I/usr/local/include  -Wall -Wextra -pedantic -O2 -c FRModel_Vertices_005.cc -o FRModel_Vertices_005.o
g++ -fPIC -I/usr/local/include -I/usr/local/include  -Wall -Wextra -pedantic -O2 -c FRModel_Vertices_006.cc -o FRModel_Vertices_006.o
g++ -shared -fPIC -Wall -Wextra -pedantic -O2  FRModel.o FRModel_Vertices_004.o FRModel_Vertices_005.o FRModel_Vertices_006.o -o FRModel.so
[root@d22b1de2392f Exercise_4]#
```

Exercise 4 - Starting with Contur

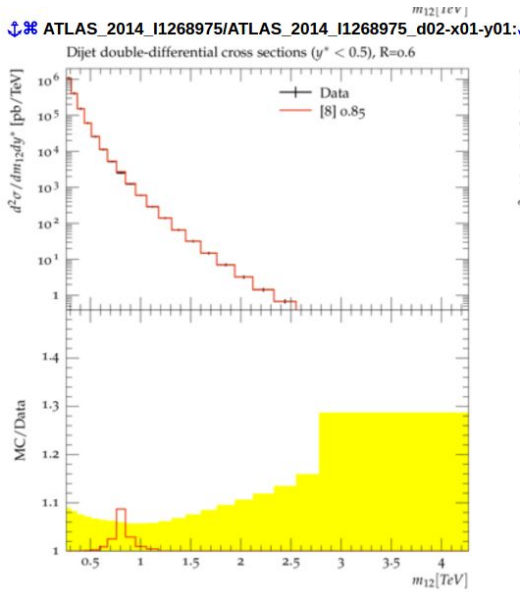
- Let’s just run again and reverse engineer what happened
- > Herwig read LHC.in
- > Herwig run LHC.run -N 1000
- > rivet-mkhtml LHC.yoda
- > contur LHC.yoda
- > contur-mkhtml LHC.yoda

The two mkhtml commands output respective x-plots folders, open the index.html file in there from your host pc browser

Rivet output



Contur output



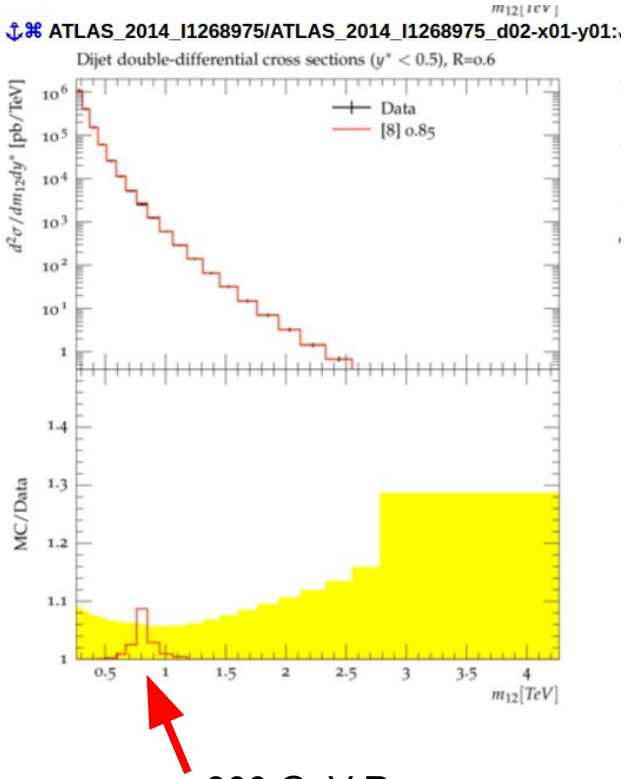
Exercise 4 - Starting with Contur

- Let’s edit the LHC.in file now
- First block where we access the parameters (These names correspond to those in the ufo files [browse the UFO directory here if you want to see])
  - This was a simplified dark matter model, with a vector mediating boson Y1, a DM candidate Xm, then couplings of the mediator to quarks gYq and to the DM gYXm

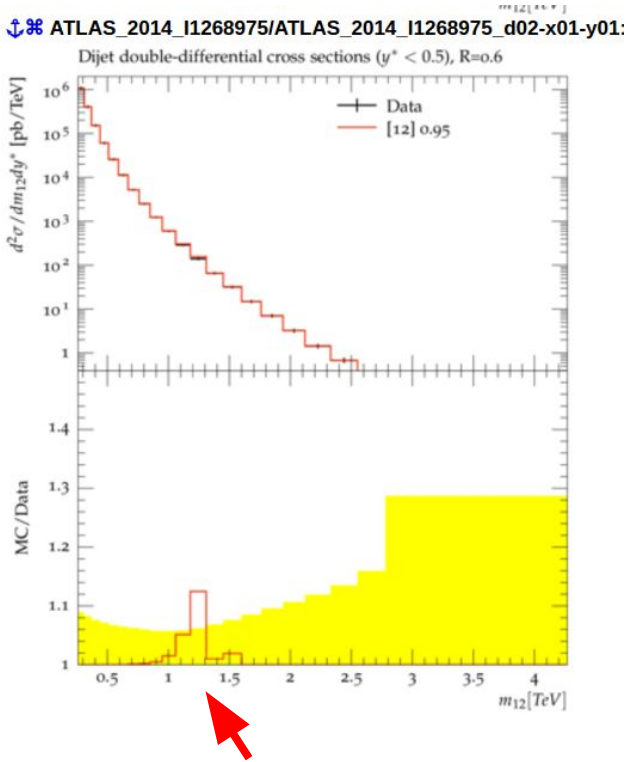
```
read FRModel.model
set /Herwig/FRModel/Particles/Y1:NominalMass 800.0
set /Herwig/FRModel/Particles/Xm:NominalMass 500.0
set /Herwig/FRModel/FRModel:gYq 0.5
set /Herwig/FRModel/FRModel:gYXm 1.0
```

- Change the Y1 Nominal Mass, lets say to 1200GeV and compare to the previous run
  - Can change any of these parameters, does what you see make sense
  - Remember, repeat the Herwig read/run steps (give the run step a -t HighMass command line option to not overwrite, then repeat the contur xyz.yoda, contur-mkhtml xyz.yoda

Exercise 4 - Starting with Contur



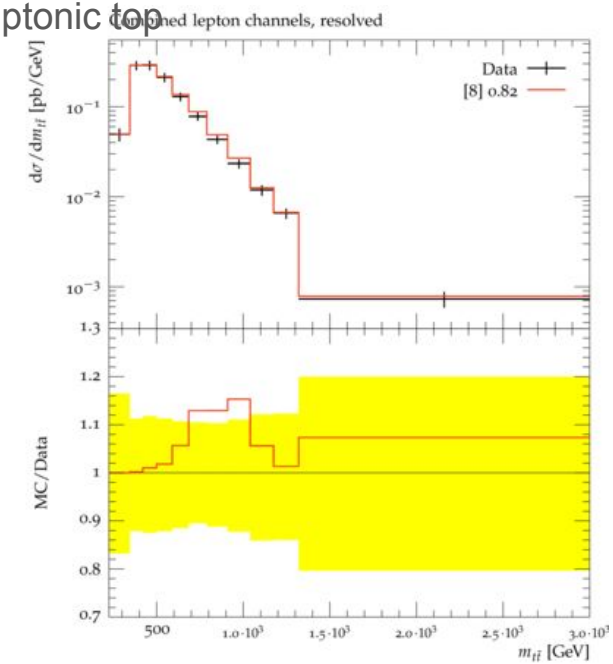
800 GeV Run



1.2 TeV Run

Exercise 4 - Starting with Contur

- Try and change up the run to the following
  - 13TeV Beam energy (set in the LHC.in file)
  - Swap the rivet routines to:
    - insert Rivet:Analyses 0 CMS_2018_I1662081 # semileptonic top
    - insert Rivet:Analyses 0 CMS_2018_I1663958 # semileptonic top
    - insert Rivet:Analyses 0 ATLAS_2017_I1614149
  - May need to restrict the decay modes, comment out
- What happens now?
  - Does it make sense? Hint, look at the LHC.out files
  - Try taking the Y1 mass below 2*Mtop
  - Does this make sense?
- Worked example in solutions directory



Exercise 5 - Contur on a grid

- Got an idea of looking at 1 dimension of a model, mostly want to evaluate multiple dependencies at once
- Exercise_5 directory:
  - Template LHC.in file (look the top parameter block, the expressions in curly braces will be string matched and values substituted in
  - Template param_file.dat (this is the contur grid mode steering file, the expressions in double square braces match those in the template generator file)
    - 4 Run modes, LIN (linear, shown in example file), LOG (logarithmic, setup as linear is with start and stop values), CONST (constant, shown in file), REL (relative, evaluates algebraic expression for other parameters, commented out but shown in file)
  - GridPack, contains model file, compile as we did in Exercise 4, ufo2herwig etc.
    - Note the template Rivet include files here, divided up by beam energy and some final state, the one choses matches that set in the LHC.in template

Exercise 5 - Contur on a grid

- Try running command in this directory (-s tells it not to run the event generator but just setup the templates)
  - Again batch-submit --help contains info, the default will pick up what is already in the dir
- > batch-submit -s
- Makes a myscanXY directory
  - N subfolders based on number of LOG/LIN scanned parameters.
    - Do the parameters in the subfolder LHC.in and params.dat files make sense
- Try and use the REL mode for the parameter by swapping in the commented out example in the top level param_card.dat
- Build the model in the reference gridpack as in Ex4, (ufo2herwig DM_… && make)
- Try running the individual point in the subfolder
- > source runpoint_WXYZ.sh

Exercise 5 - Contur on a grid

- Try running command in this directory (-s tells it not to run the event generator but just setup the templates)
  - Again batch-submit --help contains info, the default will pick up what is already in the dir

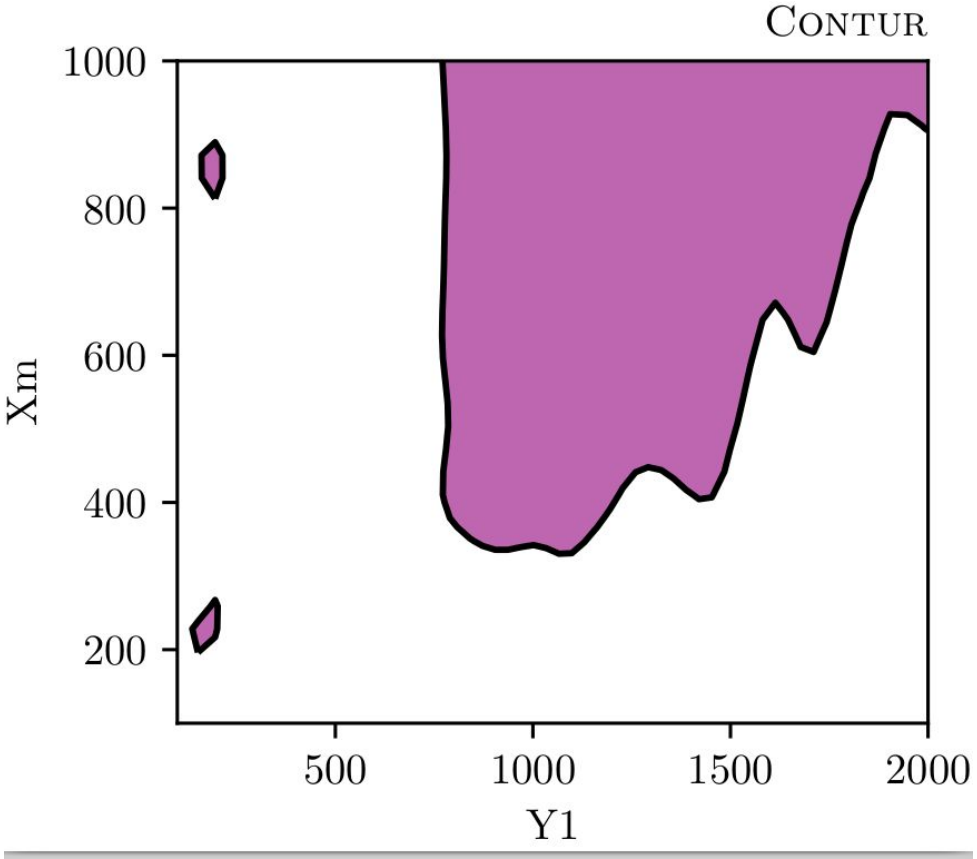
```
[root@2155368261ae Exercise_5]# cd myscan01/
[root@2155368261ae myscan01]# ls
0000  0007  0014  0021  0028  0035  0042  0049  0056  0063  0070  0077  0084  0091  0098  0105  0112  0119  0126  0133  0140  0147  0154  0161  0168  0175  0182  0189  0196
0001  0008  0015  0022  0029  0036  0043  0050  0057  0064  0071  0078  0085  0092  0099  0106  0113  0120  0127  0134  0141  0148  0155  0162  0169  0176  0183  0190  0197
0002  0009  0016  0023  0030  0037  0044  0051  0058  0065  0072  0079  0086  0093  0100  0107  0114  0121  0128  0135  0142  0149  0156  0163  0170  0177  0184  0191  0198
0003  0010  0017  0024  0031  0038  0045  0052  0059  0066  0073  0080  0087  0094  0101  0108  0115  0122  0129  0136  0143  0150  0157  0164  0171  0178  0185  0192  0199
0004  0011  0018  0025  0032  0039  0046  0053  0060  0067  0074  0081  0088  0095  0102  0109  0116  0123  0130  0137  0144  0151  0158  0165  0172  0179  0186  0193  sampled_points.dat
0005  0012  0019  0026  0033  0040  0047  0054  0061  0068  0075  0082  0089  0096  0103  0110  0117  0124  0131  0138  0145  0152  0159  0166  0173  0180  0187  0194
0006  0013  0020  0027  0034  0041  0048  0055  0062  0069  0076  0083  0090  0097  0104  0111  0118  0125  0132  0139  0146  0153  0160  0167  0174  0181  0188  0195
[root@2155368261ae myscan01]# cd 0000/
[root@2155368261ae 0000]# ls
LHC.in  params.dat  runpoint_0000.sh
[root@2155368261ae 0000]# . runpoint_0000.sh
Rivet.AnalysisLoader: WARN  Ignoring duplicate plugin analysis called 'CMS_2013_I1256943'
Forked child 0, PID 1075
Waiting for forked jobs.
Forked child 1, PID 1076
event>      200    15000
```

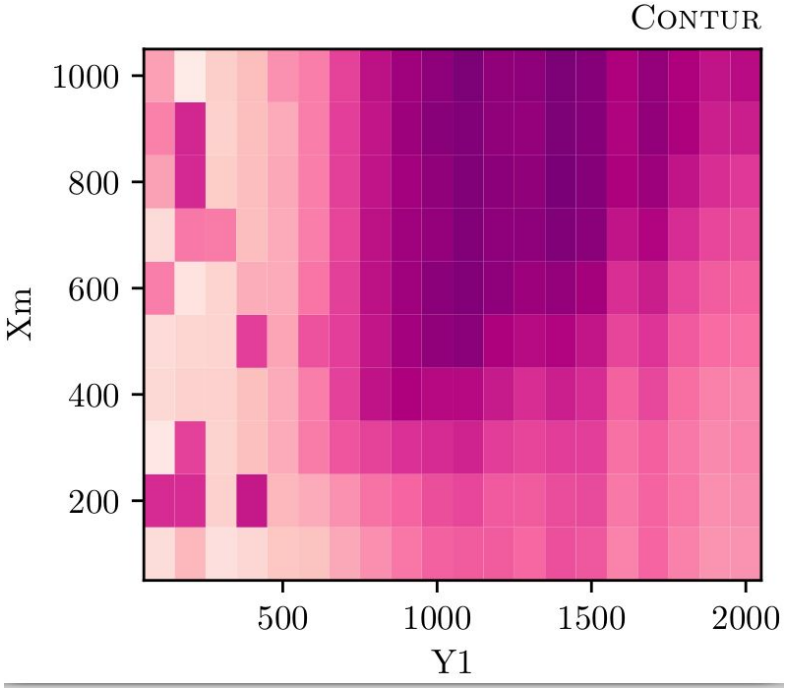
- Build the model in the reference gridpack as in Ex4, (ufo2herwig DM_… && make)
- Try running the individual point in the subfolder
- > source runpoint_WXYZ.sh

Exercise 5 - Contur on a grid

- We use -s to not run all of these jobs, can see that these 2D scans quickly require a lot of MC generated
- In the preMadeScan subfolder there is a tarball with an example grid already ran
- > tar -xvzf preMadeScan.tar.gz
  - Format should be familiar as per previous part of Ex5
  - Here I have ran all of the event generator jobs, look in the subfolders, yoda files!
- > contur -g myscan00
  - -g to specify a grid output in the contur grid structure
- Outputs (default name, can be changed, again --help on any terminal command should help!) a folder ANALYSIS
- Inside here is a contur.map file (this contains all the grid information)
  - To plot this, > contur-plot contur.map Y1 Xm
    - We specify which parameter names we want to draw, these match the input template file again, again this outputs to a conturPlots folder in the directory you run by default

Exercise 5 - Contur on a grid





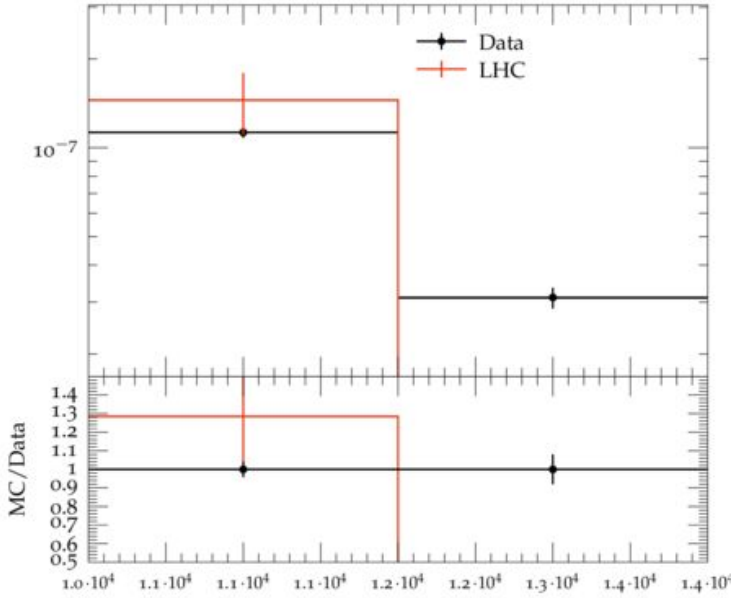
The plotting contains more options to dress this output, but for now we just note that darker colour = more disfavoured by data. Theres no fancy index.html just yet here unfortunately!

Exercise 6 - Bringing it all together

You could start this from scratch (we just combine Ex 2 and Ex 4), the directory contains a worked example

- In Ex2 we made a custom rivet plugin
  - Make a version with an mjj histogram with a 10-12 TeV bin and a 12-14 TeV bin
  - Fill with the Dijets generator (use a high jet cut!) and a new Beam energy for your new collider
  - Rename the output so Rivet recognises it as reference data
    - > source dressRef.sh
- Now use the same procedure as in Ex4 and edit the LHC.in file (match the beam energies), compile the model as before.
  - Ufo2herwig can be temperamental, might need to move the MC_TUTORIAL.cc plugin to a subfolder temporarily
  - Set the LHC.in model parameters to interesting values (hint mediator mass ~ dijet mass)
- Run the generator and rivet-mkhtml the yoda output

Exercise 6 - Bringing it all together



Exercise 6 - Bringing it all together

- We could think now to include this output in the ‘limit’ setting framework (a further exercise…)
- We have to be very careful here though
  - Do we trust the SM prediction we use as the null hypothesis?
  - What statement would we be making by assuming the data taken is equivalent to the theoretical simulation
    - This is different to assuming the theoretical simulation matches the data taken….
  - Detector effects at this beam energy?
  - Fiducial volume of the hypothesised detector?
  - Extrapolation of LHC tunes/pdfs to new phase space
  - Does the model still make sense at this energy?

This is just a toy example but these are the kinds of things we have to think of when extrapolating to future colliders!