Vista as a tool for dataset validation

Stephen Mrenna
CD/FNAL

Vista is a tool developed to perform a global analysis of high pT data.

It was developed with an eye to discovering new phenomena.

In the process we have found it useful to “debug” the implementation of the Standard Model.

We have also found it useful in debugging generator level output by treating one Monte Carlo sample as “the data” and compareing to another sample.

Vista can be used to validate datasets!

First, I will get a brief overview of the pieces of Vista that are relevant to dataset validation.

Second, I will describe progress on implementing tools into CMSSW.

Vista addresses the question:

Do we understand the entirety of high-pT (collider) data?

Implement the Standard Model

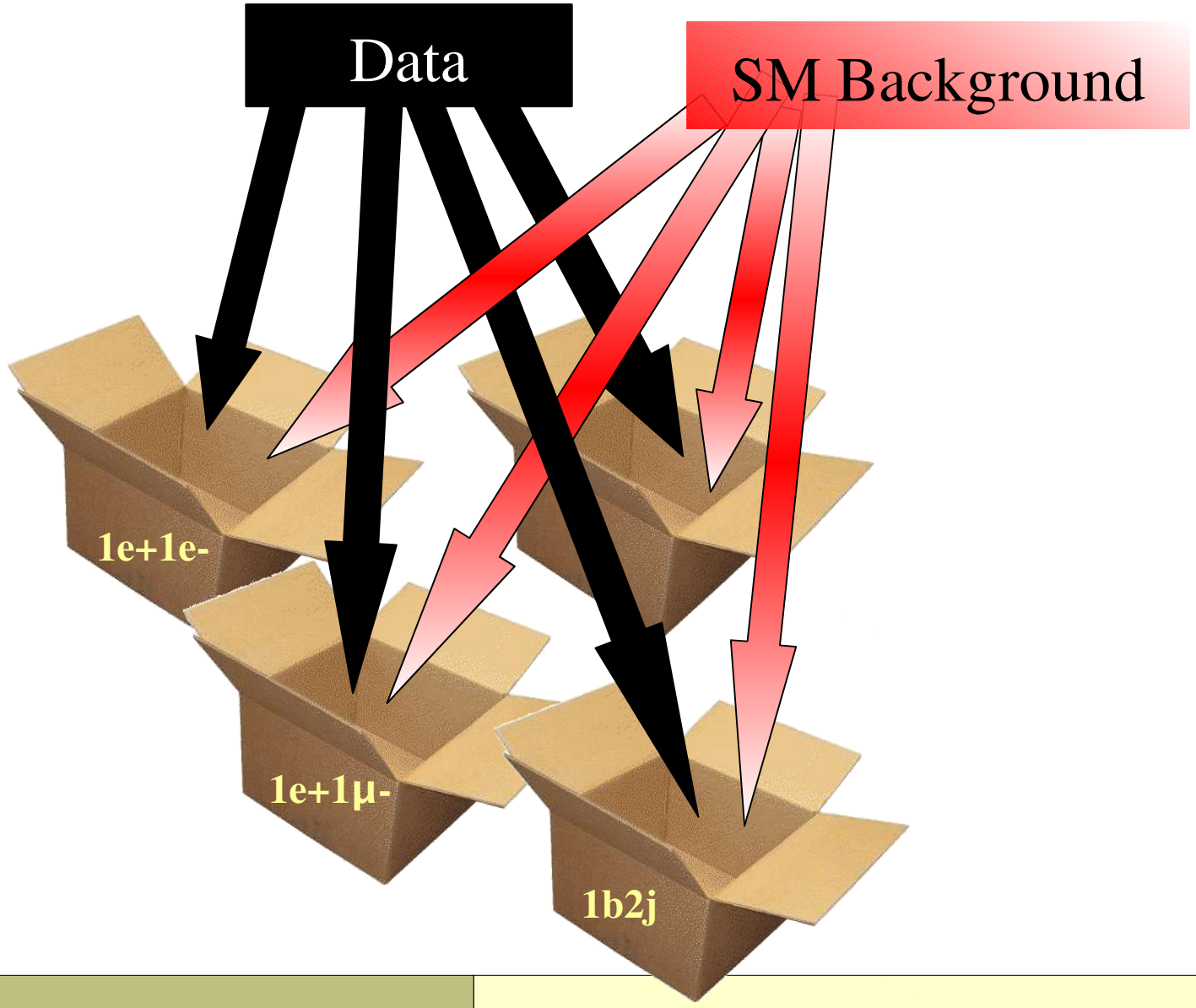
Look in many final states

Final State	Chi2	data	bkg	
1b3j1pmiss_sumPt400+ [73]	9.0	451	374.5 +- 18	(pyth_jj_200 =
2b1e+2j [-]	8.0	15	6.5 +- 1.9	(ttcp0z = 2.3
2j_sumPt0-400 [161]	6.0	69704	67013.6 +- 1171.2	(pyth_jj_018 =
2j2mu+1pmiss [-]	-5.0	2	12.2 +- 3	(mad_mu+mu-jj
1b2e+2j [-]	5.0	9	3.9 +- 1.5	(mrenna_e+e-jj
1j1ph1pmiss [5]	4.0	2591	2470.1 +- 37.7	(pyth_pj_045 =
2j1mu+1ph [-]	4.0	11	11.2 +- 2.2	(mrenna_mu+mu-
1e+1j1mu+ [-]	4.0	13	6.6 +- 2.1	(ztcp5i = 3.4
1e+2j1ph [-]	4.0	31	20.9 +- 2.7	(mad_aajj = 6.
3j2mu+ [-]	4.0	34	23.2 +- 2.7	(mrenna_mu+mu-
2b2j1pmiss_sumPt400+ [-]	-3.0	17	30.4 +- 4.2	(pyth_jj_200 =
1b2j_sumPt400+ [229]	3.0	4669	4518.6 +- 72.7	(pyth_jj_200 =
4j_sumPt0-400 [253]	-3.0	2611	2736.9 +- 42.3	(pyth_jj_040 =
2b1j1ph1pmiss [-]	3.0	6	2.7 +- 1.5	(pyth_jj_200 =
1b1j1mu+ [-]	3.0	67	53.8 +- 4.3	(pyth_jj_018 =
1j1ph [277]	3.0	31738	31149.8 +- 352.1	(pyth_pj_045 =
1e+1mu+ [-]	3.0	66	53.5 +- 3.2	(ztcp5i = 38.8
4j1mu+ [-]	3.0	73	61.3 +- 2.6	(pyth_jj_040 =
5j [269]	3.0	448	406 +- 14.5	(pyth_jj_040 =
1b5j [-]	3.0	8	8.9 +- 1.7	(pyth_jj_060 =
1b1j1pmiss_sumPt0-400 [-]	2.0	120	104 +- 7.2	(pyth_jj_040 =
2j1pmiss_sumPt0-400 [37]	2.0	2381	2281.2 +- 73.9	(pyth_jj_018 =

...

Have a measure of agreement

Vista: partitioning into final states

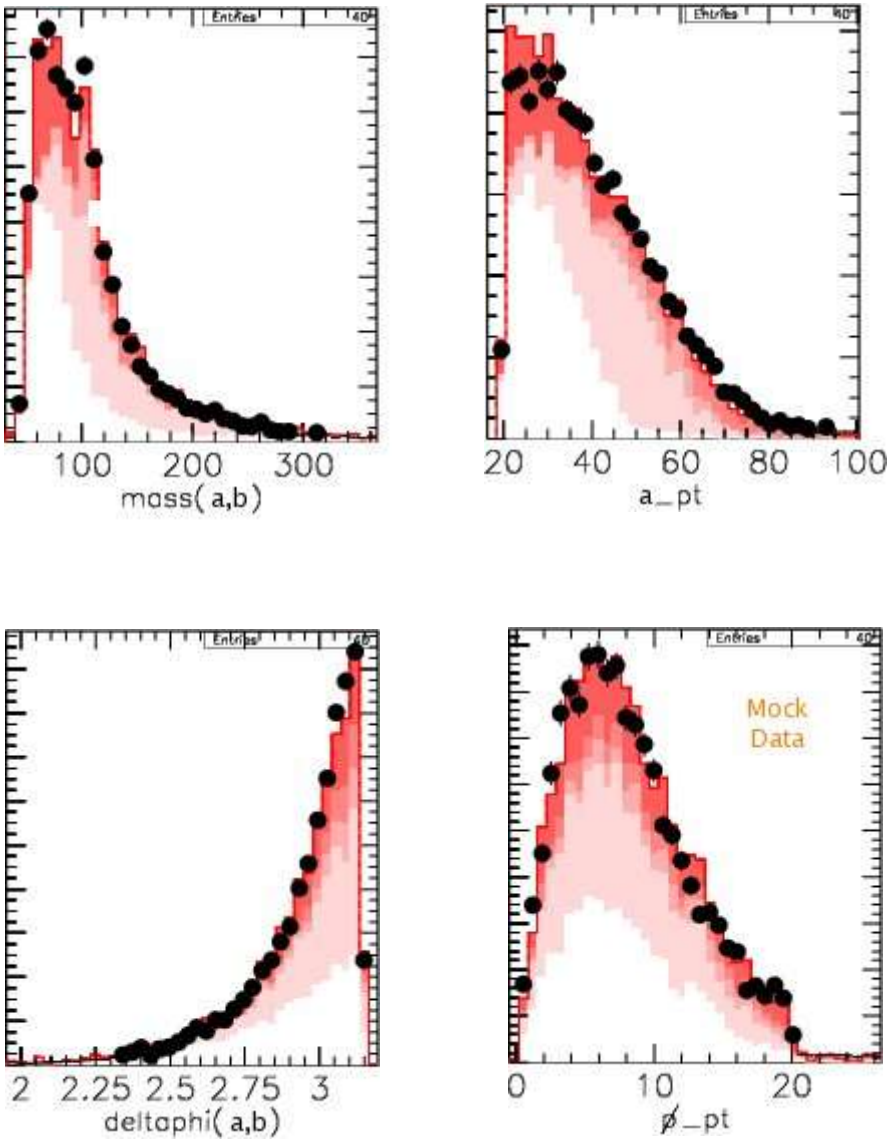


~350 boxes

partition rules for Vista

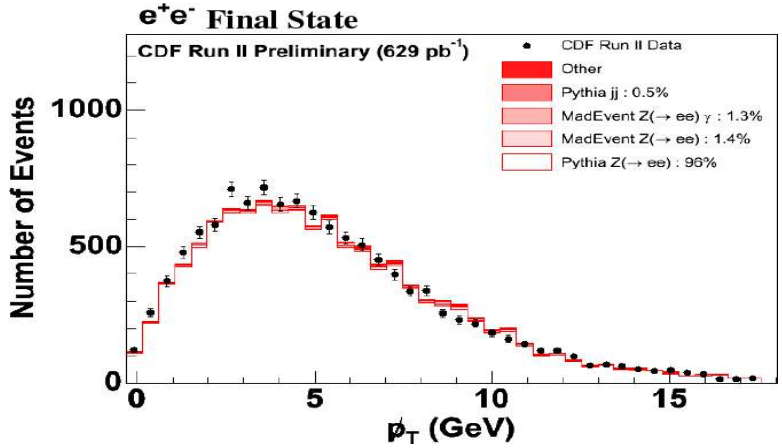
- Group events exclusively, according to what final reconstructed objects they contain. Objects are : e, γ, μ, τ, b, j, pTmiss
- Any object with pT > 17 GeV counts.
- μ,τ, b : $|\eta| < 1.0$
- e, j, γ : $|\eta| < 2.5$

Check also consistency between shapes of object distributions in each final state

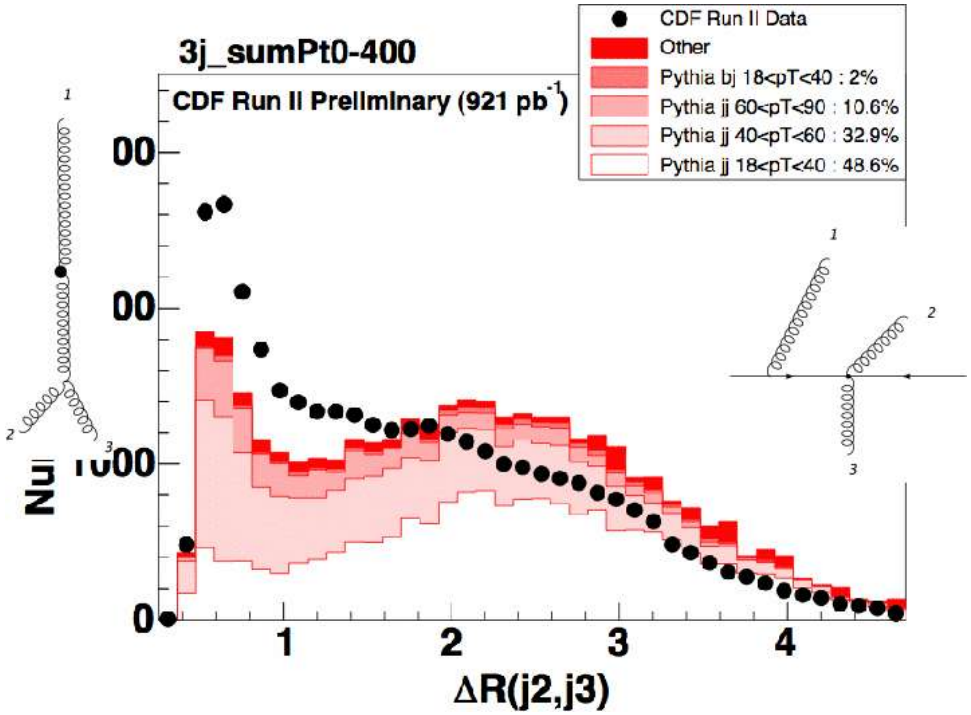


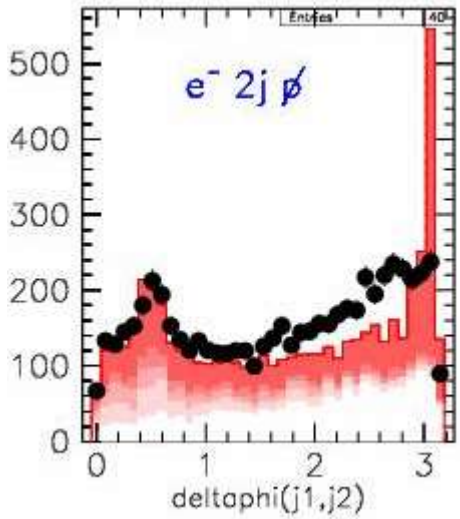
Successful "Debugging" of SM implementation

Large instrinsic kT



ISR too hard in Tune A





FSR for multiparton interactions

My aim is to make the same sort of comparisons, but using generator-level information for both “data” and “theory”.

I take the generator-level information and massage it into the equivalent of high-level “objects.” It doesn't matter exactly how I do this, as long as I treat “data” and “theory” the same way.

The goal is to catch implementation errors and inconsistencies.

Over the past few days, I have implemented a converter that reads MC generator datasets and extracts the particle level information.

Particles from the hard interaction and decay are extracted, and jets are formed from the rest.

As a proof of principle, I have been looping over a sample of tt0j events.

Dataset:

/tt0j/CMSSW_1_3_1-Spring07-1560/GEN-SIM-DIGI-RECO « more »

contains 41328 events, 89 files, 92.9GB.

Show: Summary Details

Location	Events	Files	size	LFNs
ccsrm.in2p3.fr	41328	89	92.9GB	cff plain
cmssrm.fnal.gov	41328	89	92.9GB	cff plain
srm.unl.edu	41328	89	92.9GB	cff plain

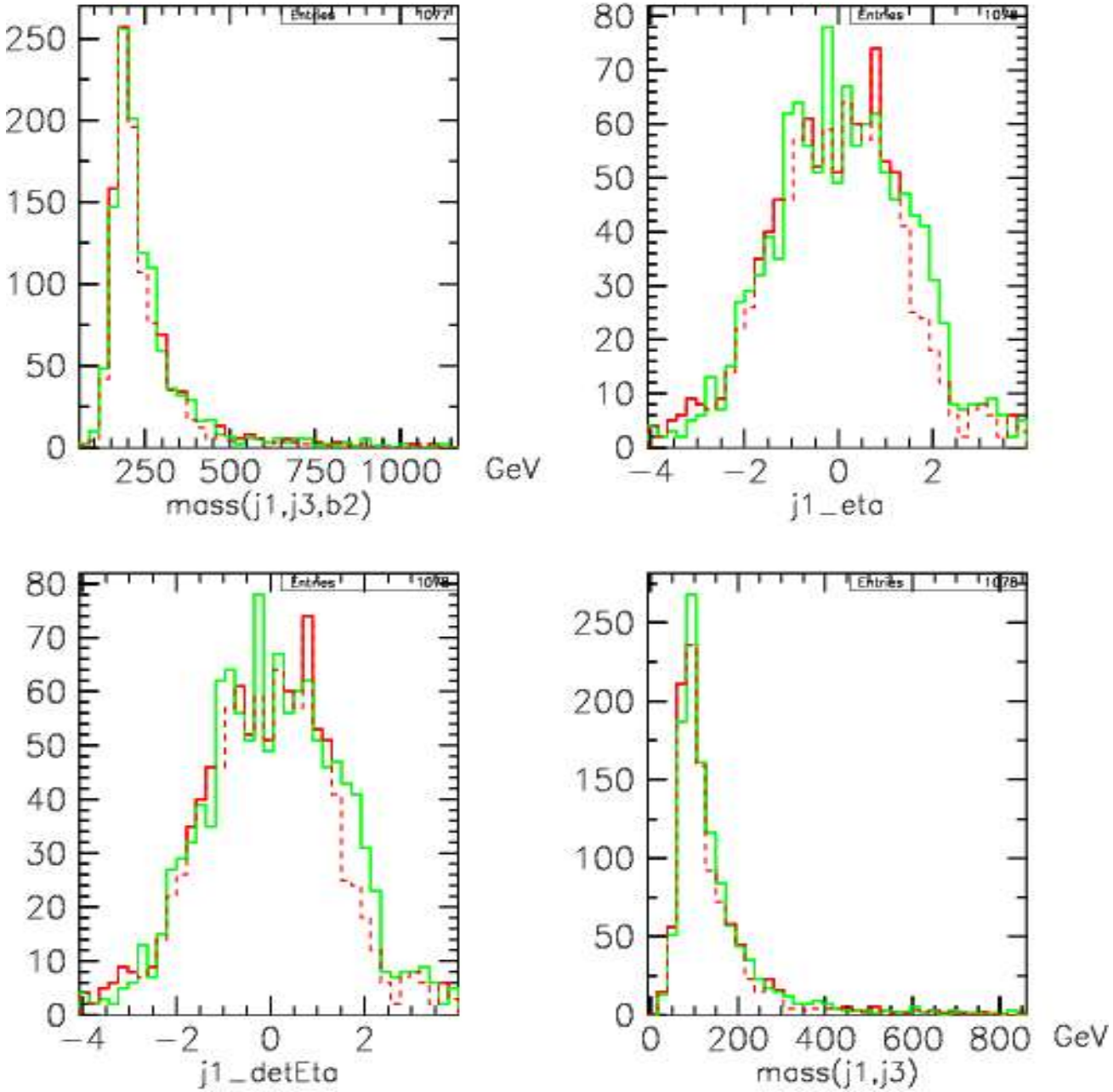
As an unexpected consequence, I discovered that about 1% of the events in a particular tt0j sample violate energy-conservation at a noticeable level.

More detailed comparisons are forthcoming.

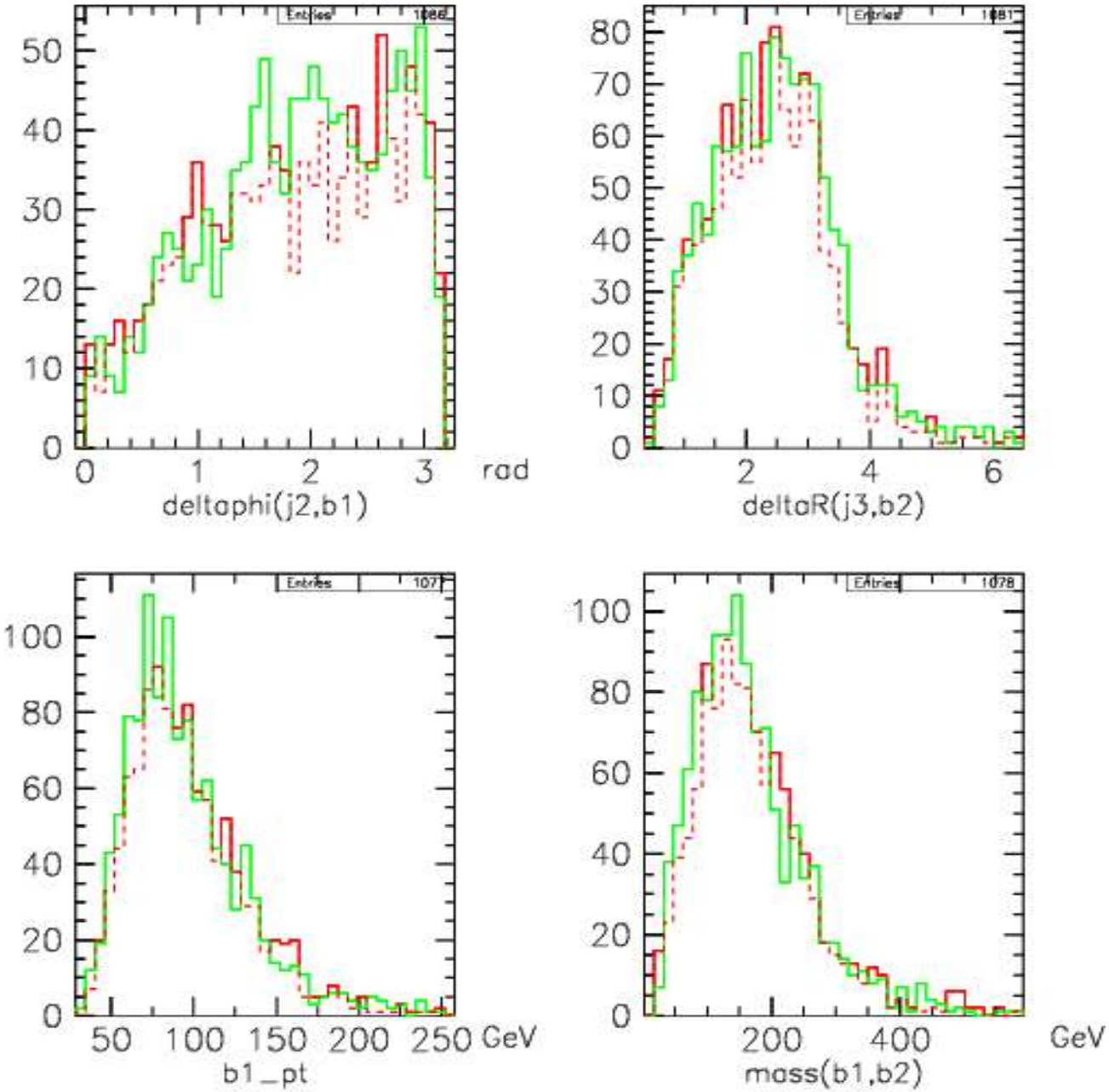
```
~/beamer ] more shape.txt
Wed Sep 19 08:19:29 CDT 2007

With 9172 distributions considered in 552 final states,

Dissimilarities follow:

    Final state                                          tt0j_run1       tt0j_run0

    2b1e-1j1mu+1pmiss [-]        +0.5                      20          3   +-    2.7    ( sig = 3 )
    2b1e-1pmiss1tau+ [-]          0                        37         15   +-    4.9    ( sig = 15 )
    2b1e-3j1pmiss [221]           0                       135         92   +-   10.6    ( sig = 92 )
    1b2j [81]                     0                        65         37   +-    7.1    ( sig = 37 )
    1b2j1pmiss [77]               0                       209        271   +-   17.5    ( sig = 271 )
    2b1mu+1pmiss [293]            0                        91         59   +-    8.7    ( sig = 59 )
    1b3j1tau+ [-]                 0                        14          4   +-    3      ( sig = 4 )
    2b7j1pmiss1tau+ [-]           0                         5          0   +-    1      ( )
    2b1e+3j [229]                 0                         8         23   +-    5.8    ( sig = 23 )
    1b2j1mu- [-]                  0                        18          7   +-    3.6    ( sig = 7 )
    1b1pmiss1tau- [57]            0                         8         22   +-    5.7    ( sig = 22 )
    1b1e-1j1pmiss1tau+ [-]        0                        16          6   +-    3.4    ( sig = 6 )
    2b4j [401]                    0                      1142       1233   +-   36.1    ( sig = 1233 )
    1b3j1mu+1pmiss1tau- [-]       0                         4          0   +-    1      ( )
    1e+2j1pmiss [-]               0                         8          2   +-    2.4    ( sig = 2 )
    2b5j1mu- [-]                  0                         8          2   +-    2.4    ( sig = 2 )
    2b3j [377]                    0                      1163       1086   +-   34      ( sig = 1086 )
    1b1e+2j [-]                   0                         8         19   +-    5.4    ( sig = 19 )
    2b1e-1pmiss [197]             0                        73         55   +-    8.4    ( sig = 55 )
    1b1e+7j1pmiss [-]             0                         3          0   +-    1      ( )
    2b1e+5j [-]                   0                         3          0   +-    1      ( )
    2b1j1mu-1ph1pmiss [-]         0                         3          0   +-    1      ( )
    2b5j1tau- [-]                 0                         3          0   +-    1      ( )
    1b1e+1pmiss1tau- [-]          0                         9          3   +-    2.7    ( sig = 3 )
    1b5j1pmiss1tau+ [-]           0                         9         19   +-    5.4    ( sig = 19 )
    1b5j [137]                    0                       355        317   +-   18.8    ( sig = 317 )
    2b6j1mu-1pmiss [-]            0                         7         16   +-    5      ( sig = 16 )
    2b2j1pmiss [337]              0                       913        853   +-   30.2    ( sig = 853 )
    1b1e+5j1pmiss [-]             0                        17          9   +-    4      ( sig = 9 )
--More--(5%)
```

lhc mc

2b 3j



lhc mc 2b 3j



Conclusions:

1. I am advocating a straightforward method for validating MC datasets at the generator level based on Vista tools developed by Knuteson and collaborators.

2. I have implemented it in a rudimentary way using text files. It will take about a week to implement the relevant Vista software in a cleaner way.

3. If this methodology seems promising, then we should make the testing a standard part of dataset generation.