TNSmooth: Root Multi-dimensional PDFs

Bill Murray
CCLRC RAL
w.murray@rl.ac.uk
home.cern.ch/~murray

ATLAS UK Higgs meeting
RAL
16 June, 2006

TNSmooth class
TNSData class
Adaptive kernel
What error to allow?

TNSmooth Class

```
TNSmooth(const char *name, const char *title,Int_t nDim, TAxis ** axes,TNSData * dataSource=0);
 Int_t Fill(Int_t nEvents=-1);
 TH1D * FillFunction(...);
 TH1D * project1D(...);
 TH2D * project2D(...);
 TH2D * slice2D(...);
 Int_t SmearD2(..);
 void normalize(Double_t total=1);
 Int_t add(TNSmooth * b);
 Int_t divide(TNSmooth * b);
 TNSmooth * createResolutionMap(Double_t errorRequired);
 Int_t equalizeBinContents(Int_t nEvents,Bool_t useOldLimits=true);
```

What does it do?

- Builds a smoothed version of an n-dimensional ntuple
  - Multi-dimensional binned distribution.
  - Operate on it (to calc. Likelihood)
  - Display projections/slices
- It does not yet know about the data source
  - Needs helper class: TNSData

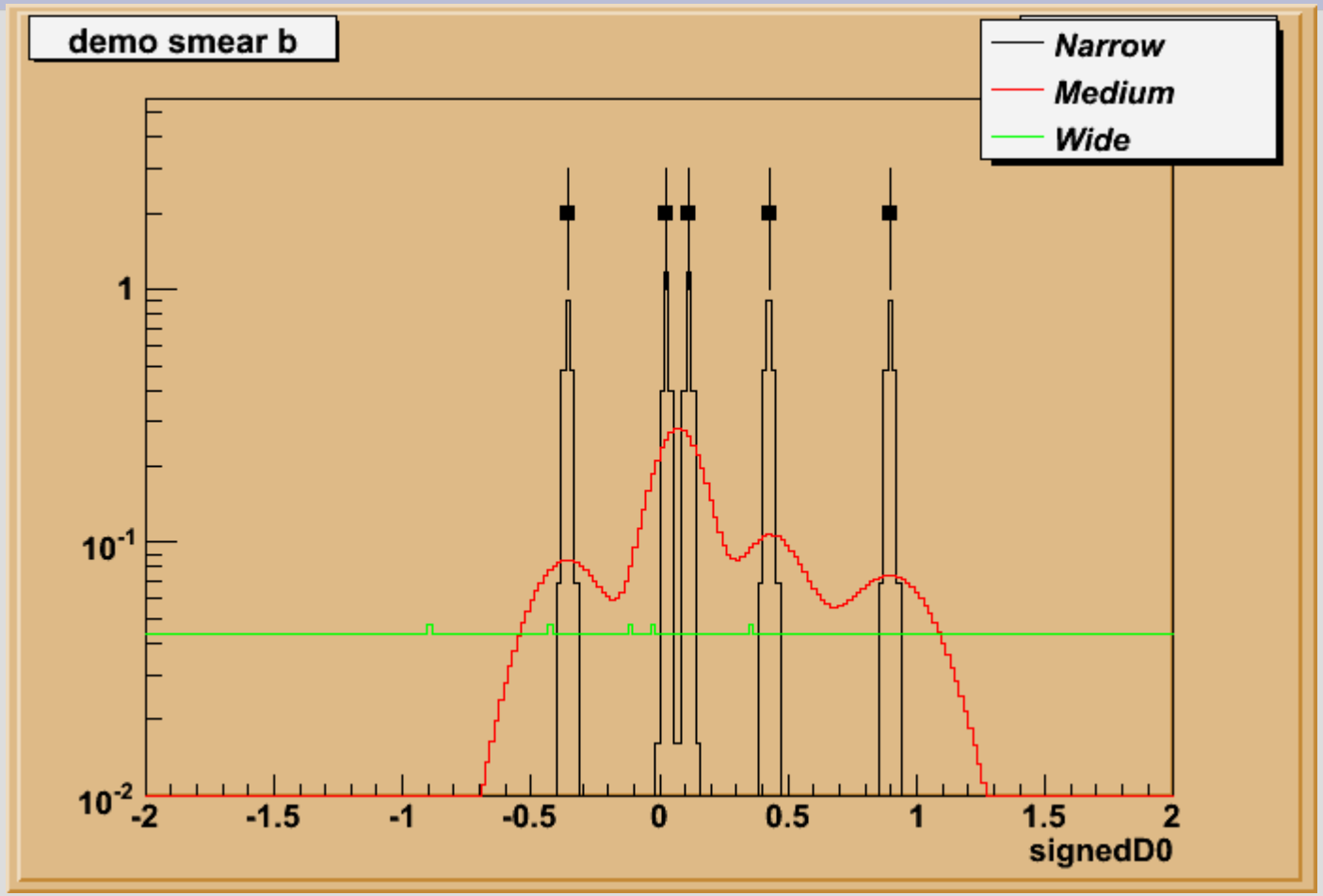
What is it for?

- Ratio of densities gives likelihood estimator
  - Most powerful way to cut
  - Or use to weight events
    - e.g. Weight tracks for b tagging
- Separate S, B density can be used for limits calculation
  - Not yet implemented.

TNSData

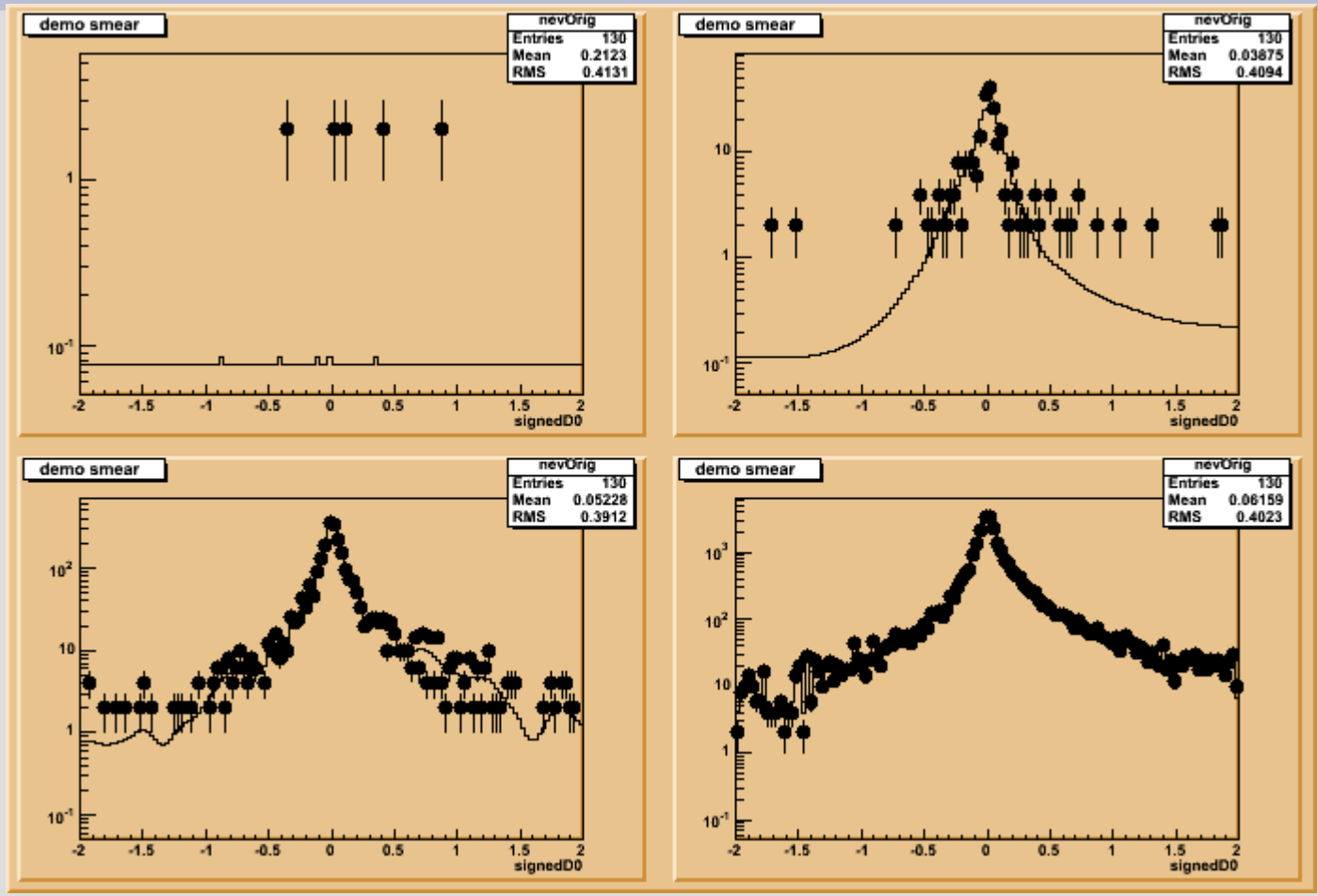
- Base version will work on a simple root tree
- Given the variable names it passes the values to TNSmooth
- A derived class TNSDataBtag is more specialised
  - Handles multi tracks per event
  - Calculates variables from AAN quantities
  - Specific to my AAN.
  - But an example of technique..

Smearing 5 events, 1D



- The kernel width is crucial

Varying event nos.



- But more events always helps

How much to smear?

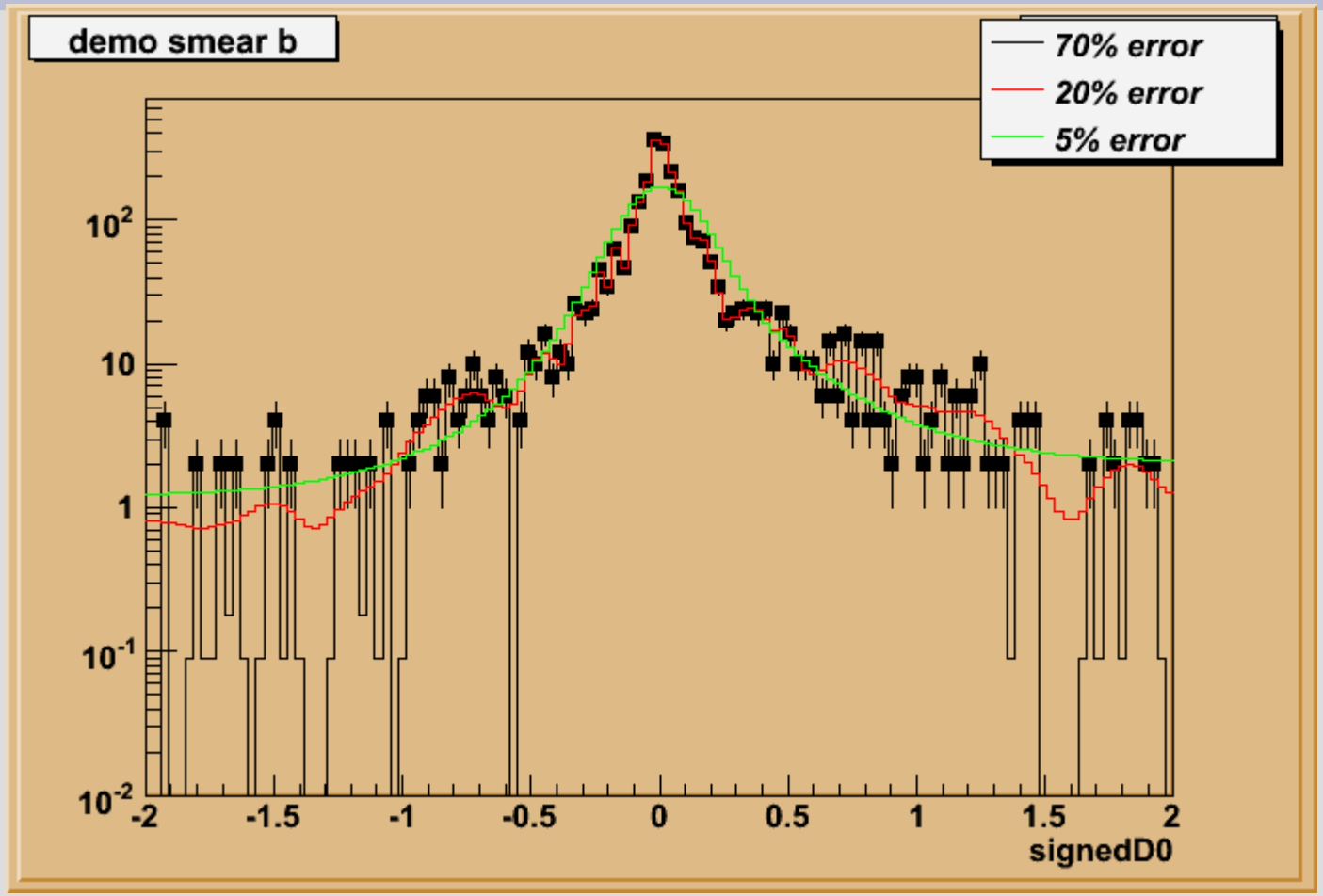
- Smear by a fixed resolution.
  - Simple to understand result
- Smear by a fixed error
  - Vary smearing resolution to give fixed stat. Error on result
  - Need to calculate map of smearing width which gives desired error.
  - More complex, (bugs?)
  - If dealing with multiple distributions (e.g. Signal, background) should use same map
  - I much prefer this

Smearing metric

- What space to smear in?
  - Frequently discussed problem for multi-dimensions
  - Not necessarily trivial for 1D
- Choice:

  The binning of the final function
- This hands the problem to the user.
- A tool exists to choose a binning uniform in event rate
- But the number of bins on each axis is user decision

How to pick the error?

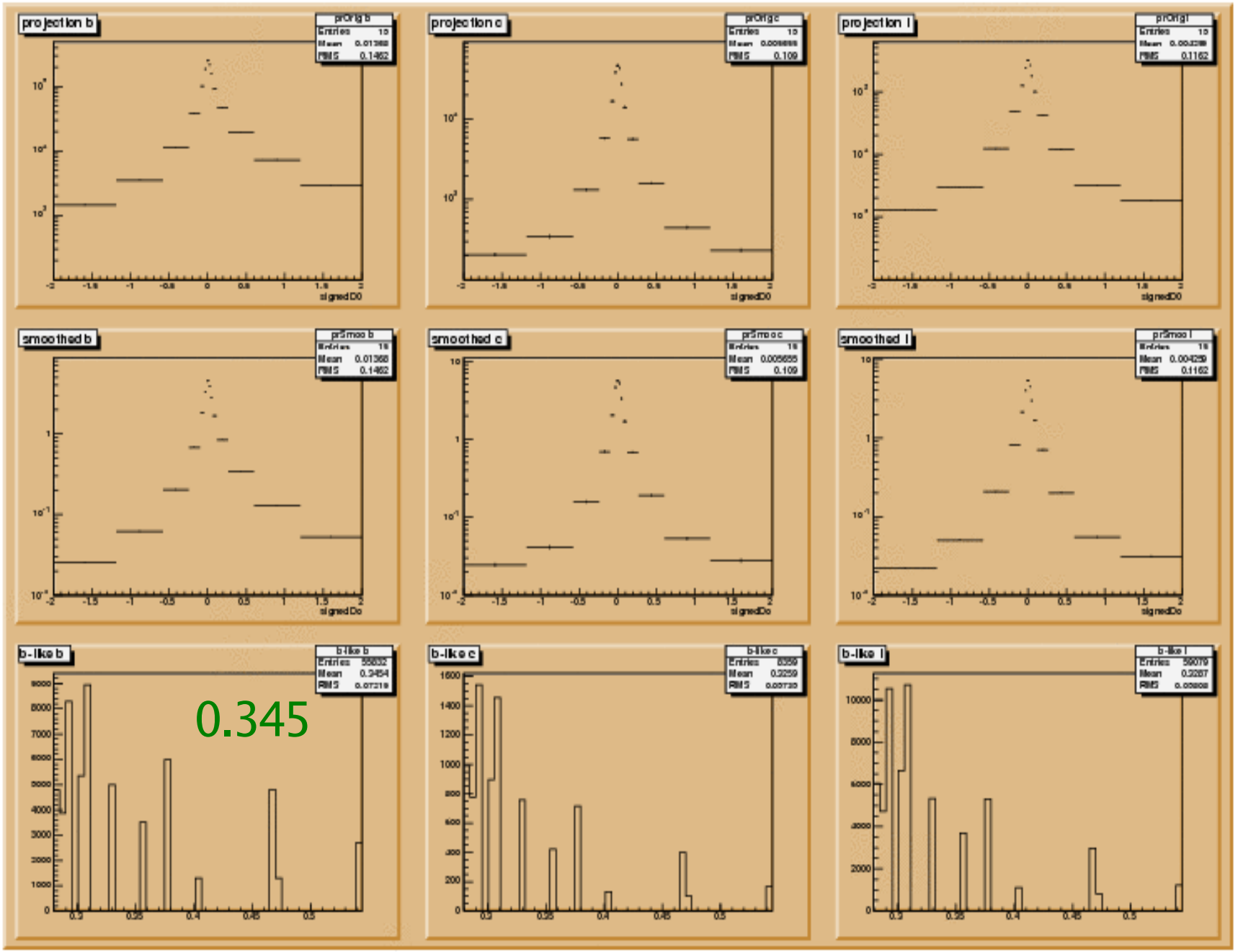


- I often use 20% But it depends what it is for!

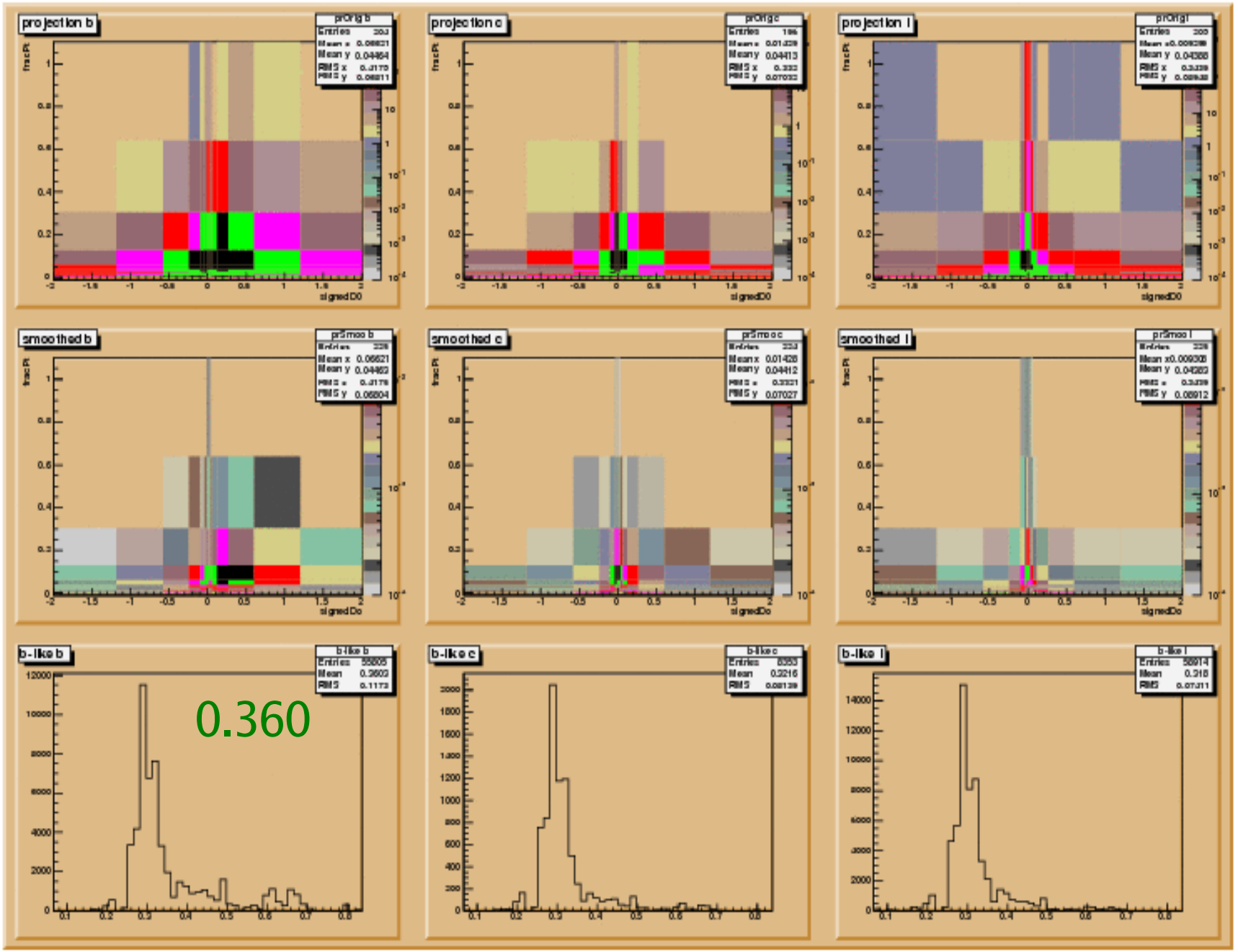
How many dimensions?

- Code is written for arbitrary number
  - Arbitrary limit of 10 imposed, one number to change
- Limited by numbers of events
  - Add a dimension and the smearing width will rise in the other dimensions to maintain error
  - So need lots of MC for big dimensionality
- Limited by CPU
  - Smearing each bin into surrounding ones slow
  - Currently 2hrs for 4 dim, 15 bins of each.
  - Might improve, not infinitely.

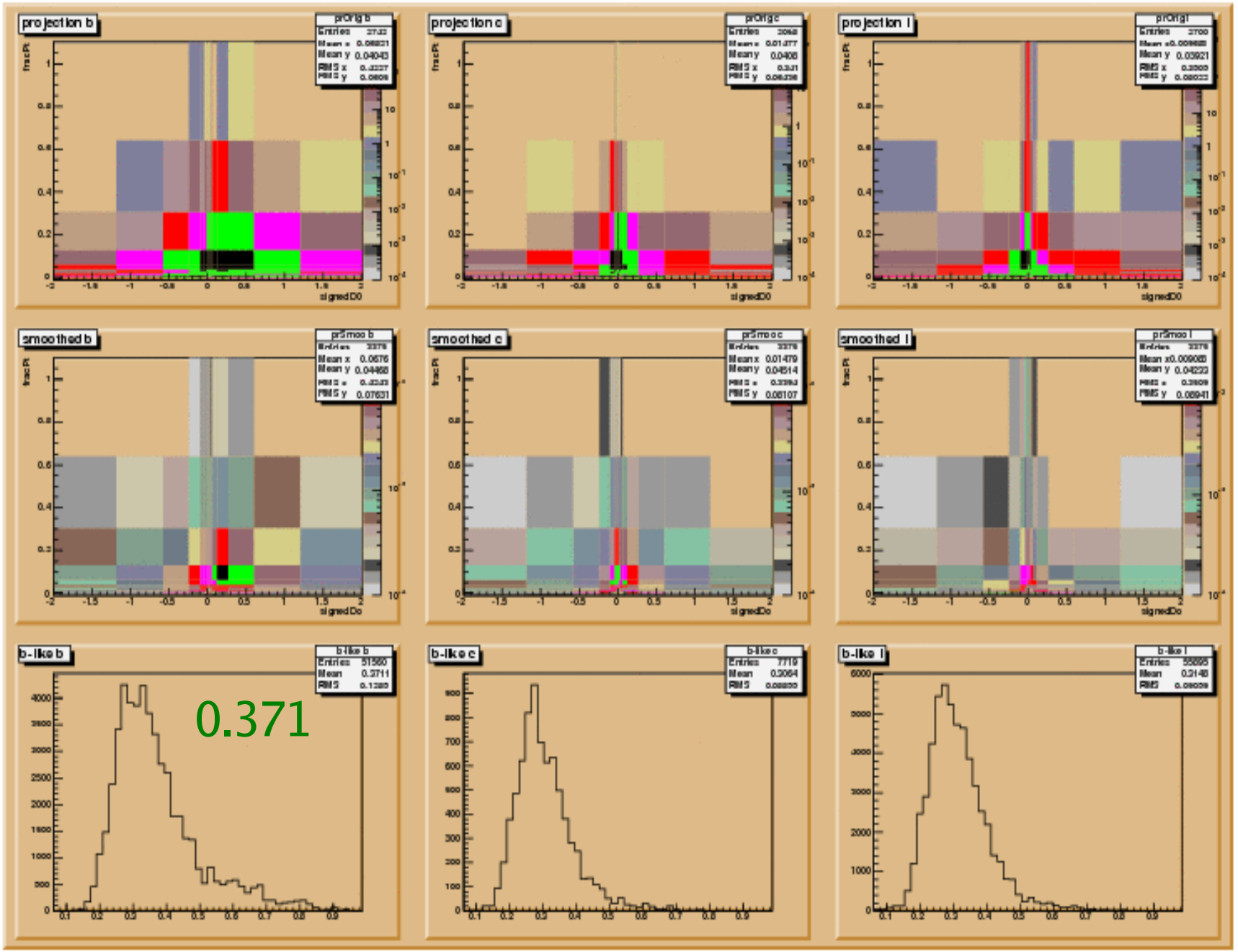
1D


projection b
projection c
projection l
smoothed b
smoothed c
smoothed l
b-like b
b-like c
b-like l
0.345

2D

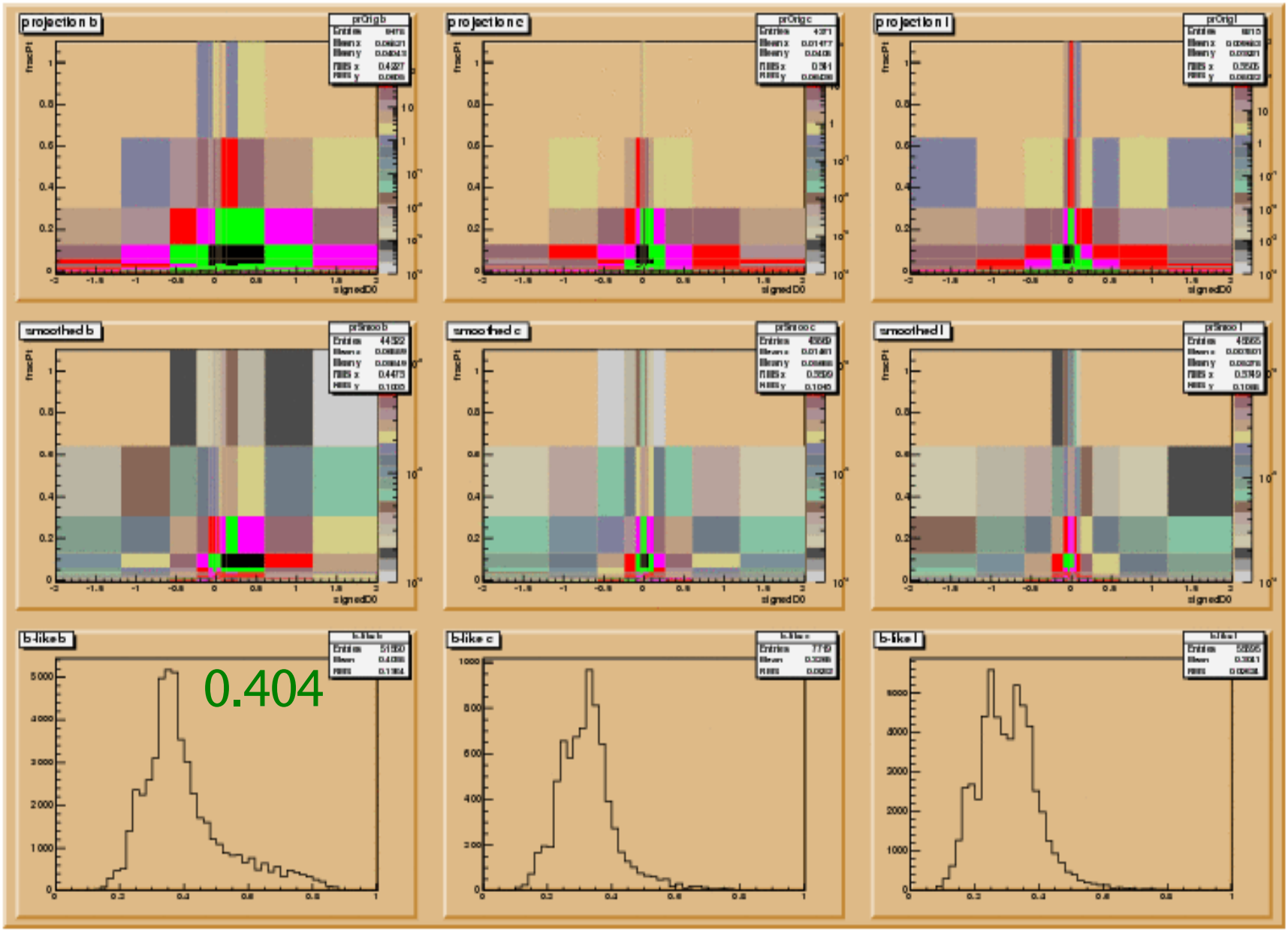

projection b
projection c
projection l
smoothed b
smoothed c
smoothed l
b-like b
b-like c
b-like l
0.360

3D


projection b
projection c
projection l
smoothed b
smoothed c
smoothed l
b-like b
b-like c
b-like l
0.371

4D

Overtaining?



How to use this?

- Works in Root
- Load library and new classes available.
- Create, fill, project, on fly (small .C)
- Can use root persistency

- I would help anyone..

Conclusions

- TNSmooth is just about functional
  - Still bugs being ironed out
  - Some functionality missing
    - e.g.: Put Likelihood back in AAN? (tricky)
  - A code cleaning should be done
- Any users would be welcome
- Likelihood calculations working
- Limits to be done