Using Insightful Miner Trees for Glast Analysis

Toby Burnett

Analysis Meeting

2 June 2003

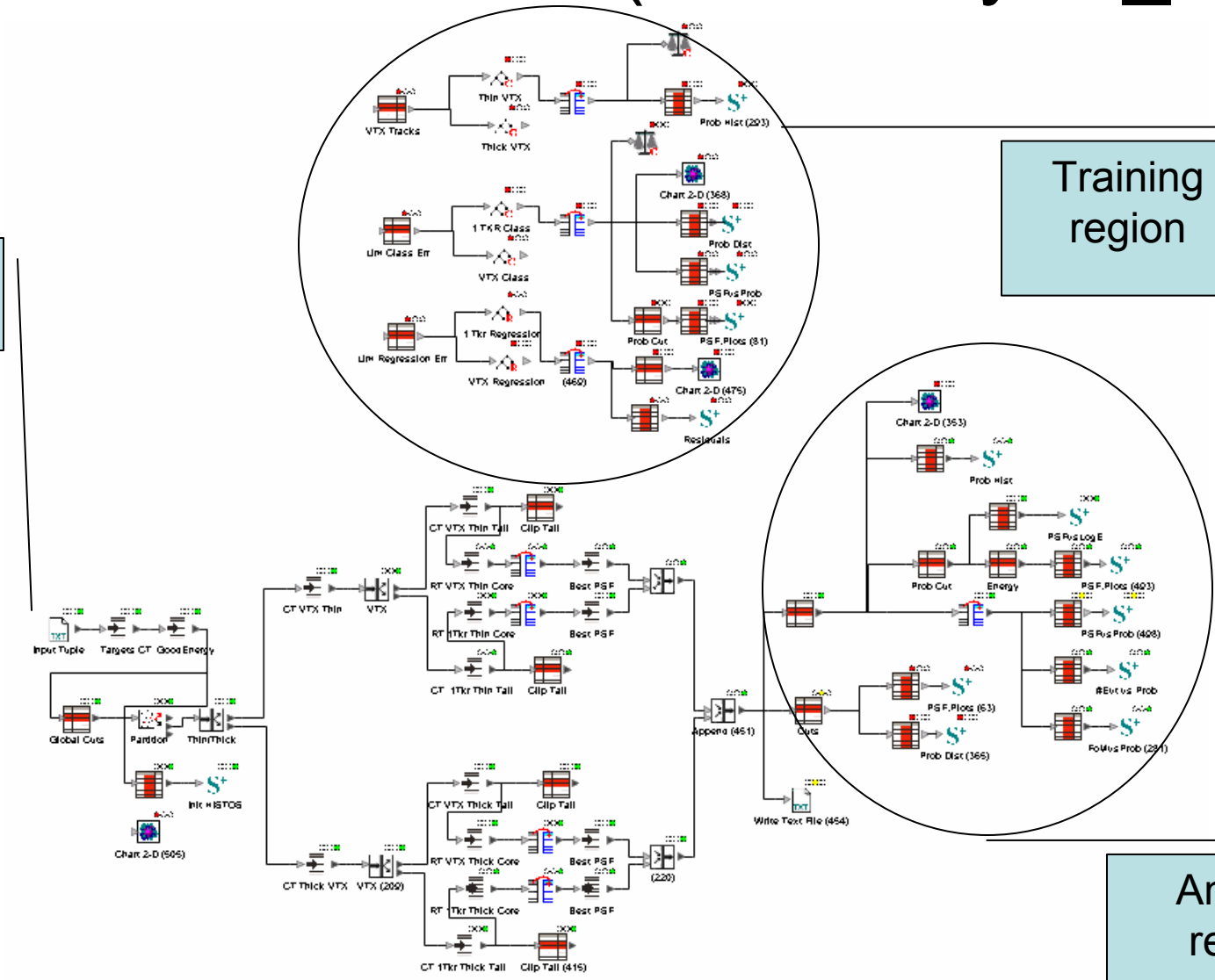
The problem

- Bill is using IM classification and regression tree analysis to achieve very good PSF results
- IM is proprietary, and very expensive

Bill's IM worksheet (PSFAnalysis_14)

Input tuple

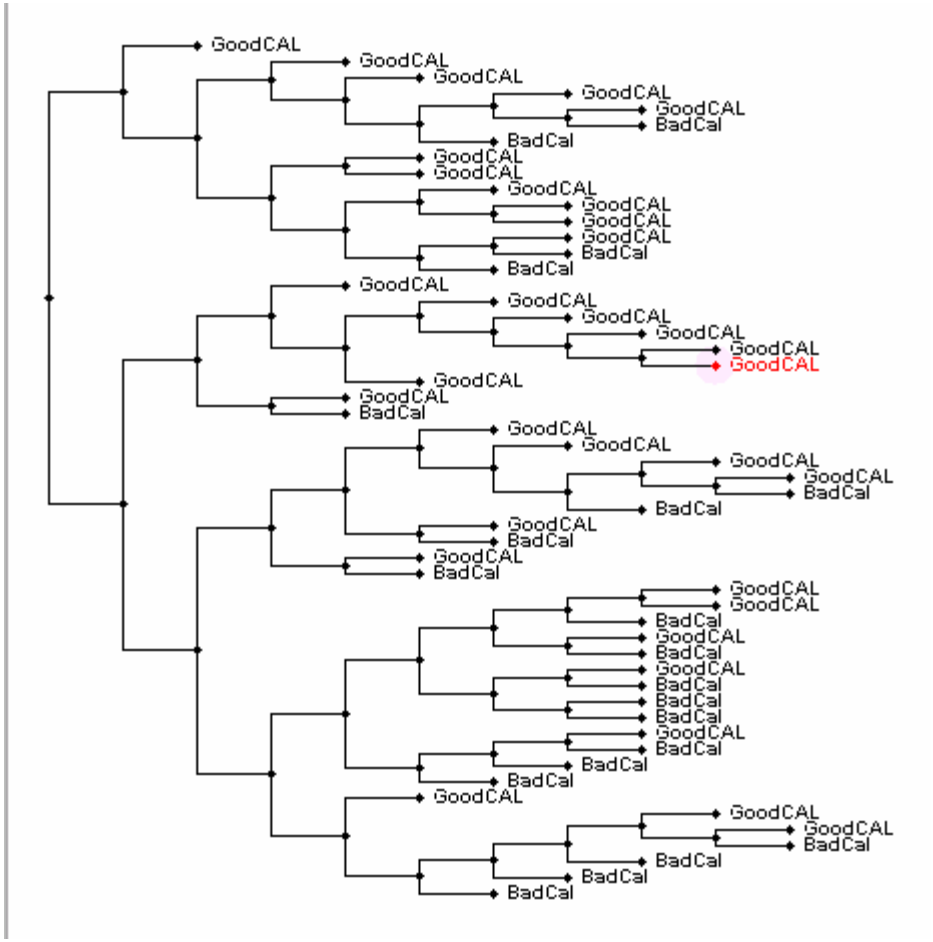
Training region



Analyze results

The Trees: calculate 4 values with 11 nodes

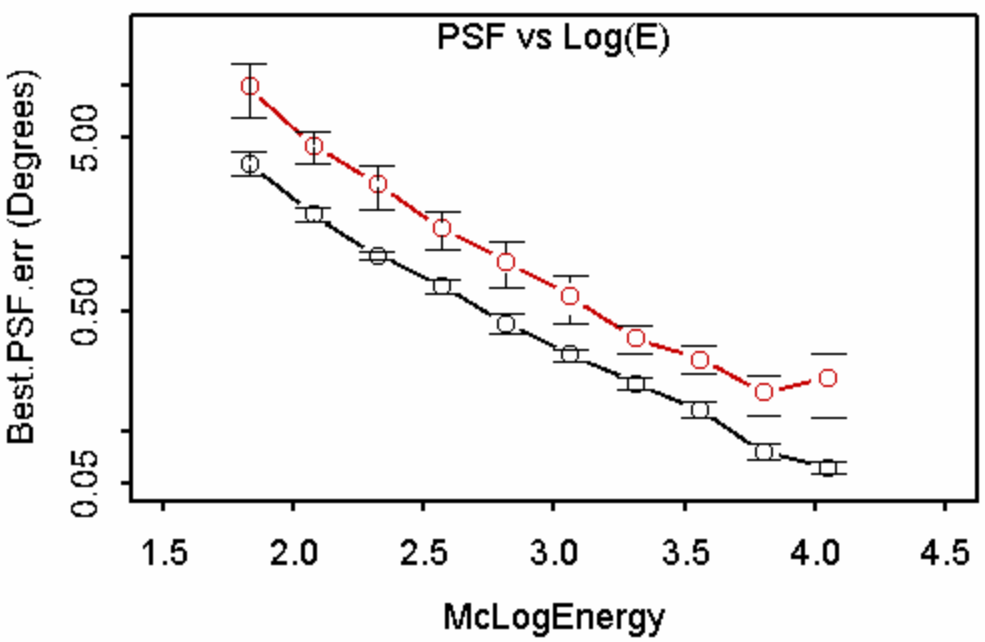
- Good calorimeter measurement [1 node]
- vertex vs. 1 track (thin and thick) [2 nodes]
- Core vs tail (thin/thick and vtx/1 trk) [4 nodes]
- Prediction of recon direction error [4 nodes]



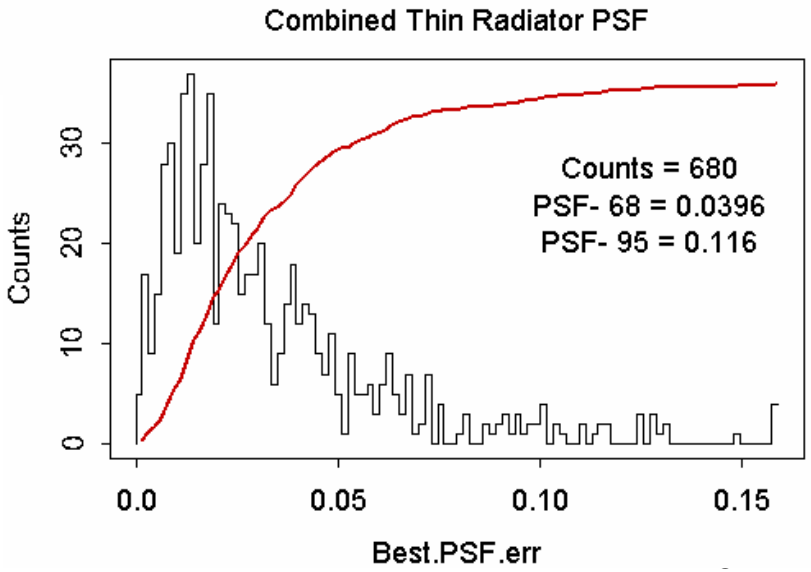
Example: A Good CAL/Bad Cal prediction node

CalTwrEdge<48.48,
CalTrackDoca<10.27,
CalTwrEdge>=26.58,
CalTwrEdge<34.81,
CalXtalRatio<0.82,
CalTransRms>3,611.48,
CalTrackDoca>3.96,
CalXtalRatio<0.46,
CalTotSumCorr>1.76

Bill's result*



100 MeV, with tail cuts and best estimate



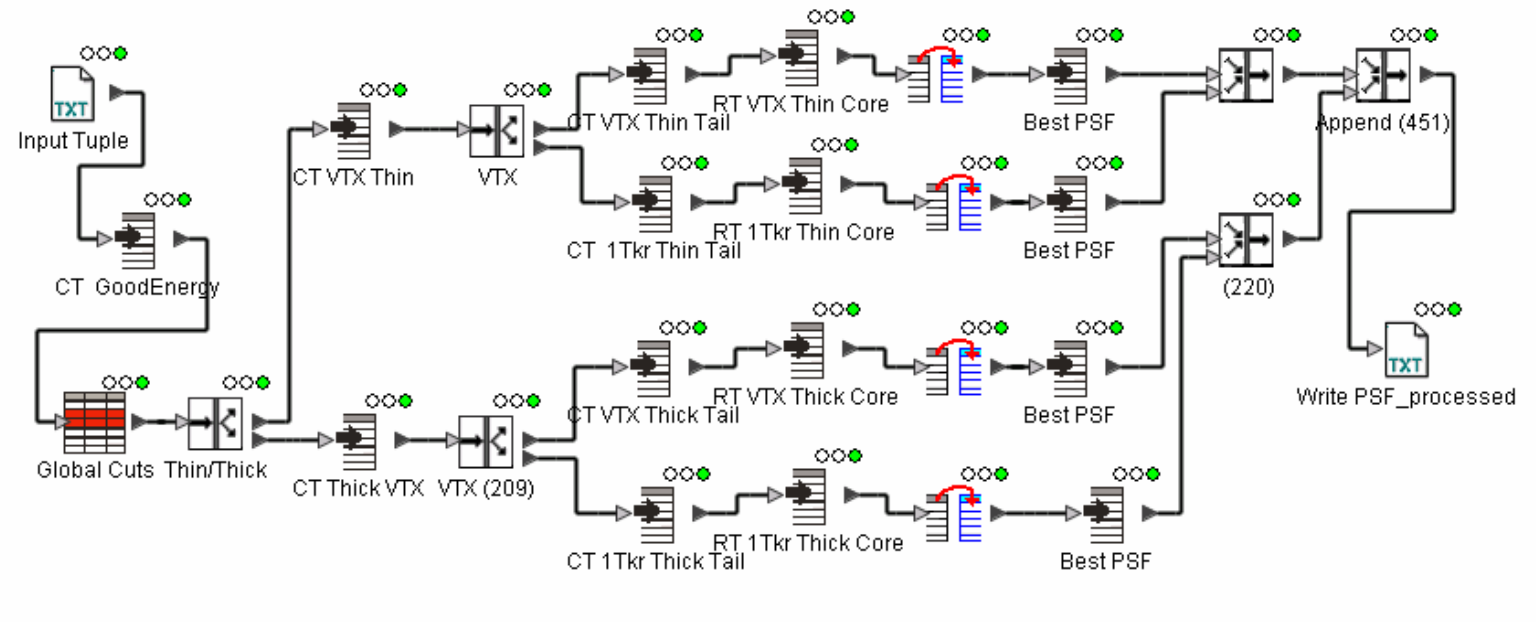
*Flawed by G4 problems

A Solution

- IM saves its results as XML files, which are easy to interpret
- A new package, “classification” defines a class *classification::Tree* that does the following:
  - accepts a “lookup” object to obtain a pointer to the value associated with named quantities
  - parses the XML file, creating a tree structure for each prediction tree found
  - for a given event, returns a value from each tree
- Merit creates and fills the new tuple variables, in a new class *ClassificationTree*.
  - duplicates the logic defining the 4 categories
  - evaluates each of the 4 variables

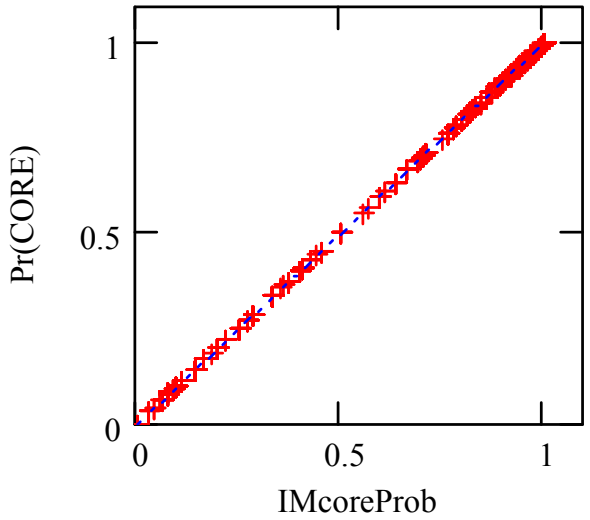
Current Procedure

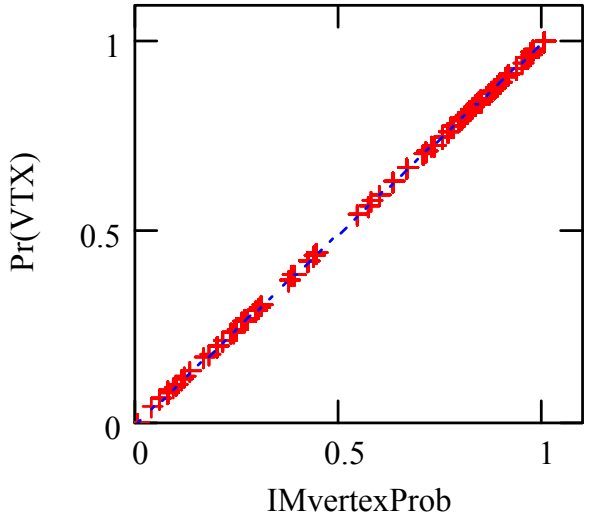
- Bill releases an IM file.
- I strip it down, removing nodes not required for analysis
  - size reduced by 1/2, to 500 Kb.
- Rename it, and check it in to cvs as *classification/xml/PSF_Analysis.xml*
- Create a tuple with merit, containing the new tuple quantities
- Feed that tuple to this IM worksheet, which writes a new tuple with both versions of the same variables



Results: the good

- The comparisons were with 10000 generated 100 MeV normal
- The vertex classification (used to select vertex vs. 1 Track direction estimate) is perfect, as is the core vs. tail



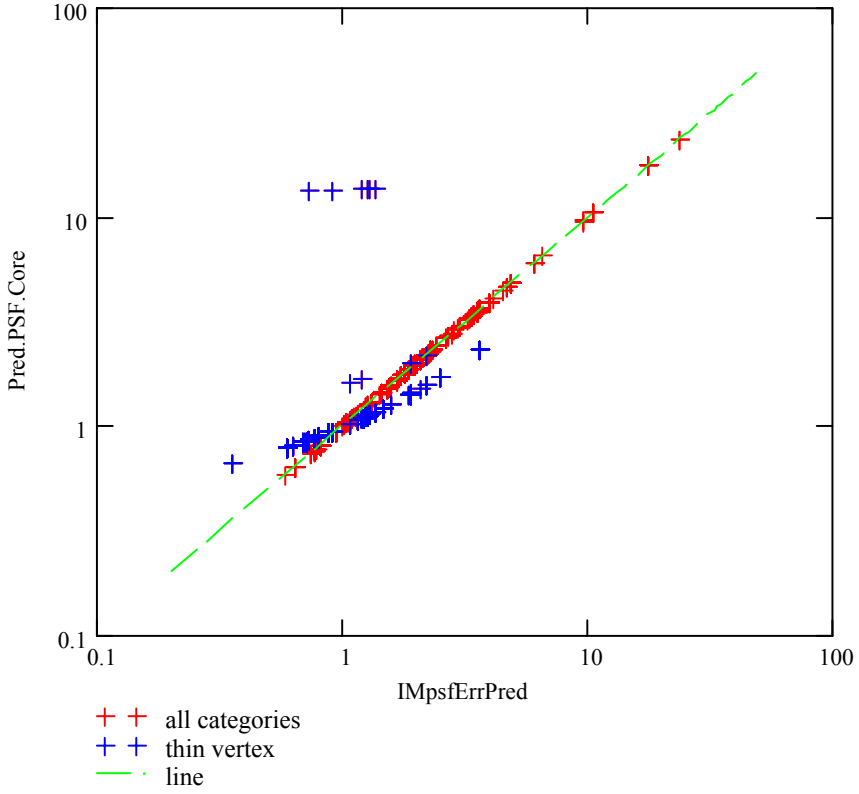


Results: the bad

- The results of the “regression tree” to predict the psf error has two populations!
- The agreement is rather poor for the “thin vertex” category; otherwise perfect.

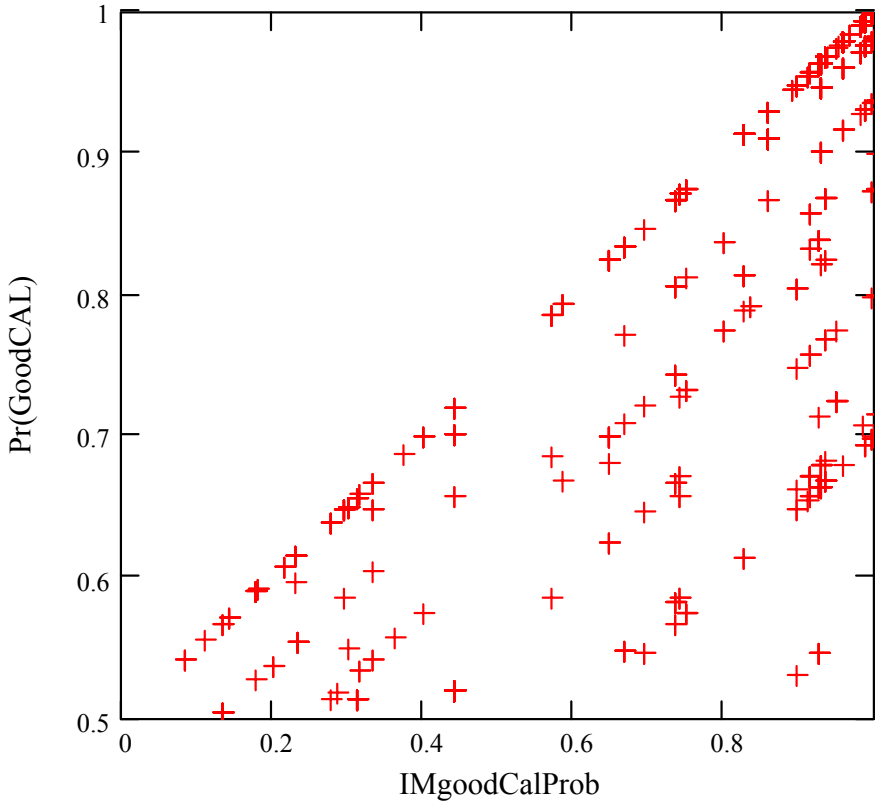
An explanation: Bill generated two different trees from different data sets, of 1000, and 243 events. (The latter has only two nodes and can only generate 3 values.)

  - The merit evaluation is only the first tree
  - The IM evaluation uses an average of the two trees.
  - Note that there are three branches.



Results: the ugly

- This is the comparison of the prediction for good energy measurement
- Again, Bill created two trees, which are apparently being averaged.



Observations, plans?

- Two possibilities to fix the “disagreement”
  - Bill: train only one tree
  - me: average all the trees
- Using IM to train the classification or regression trees
  - The current procedure is exploratory
  - If we decide to use these trees in the final analysis, they must be trained systematically
  - Another possibility (idea from Tracy): use the classification/regression analysis in S-PLUS, which manages tree objects.

100 MeV analysis w/ merit analysis

- Example only: G4 5.0 is too flawed to take seriously
- Tail cuts are clearly effective