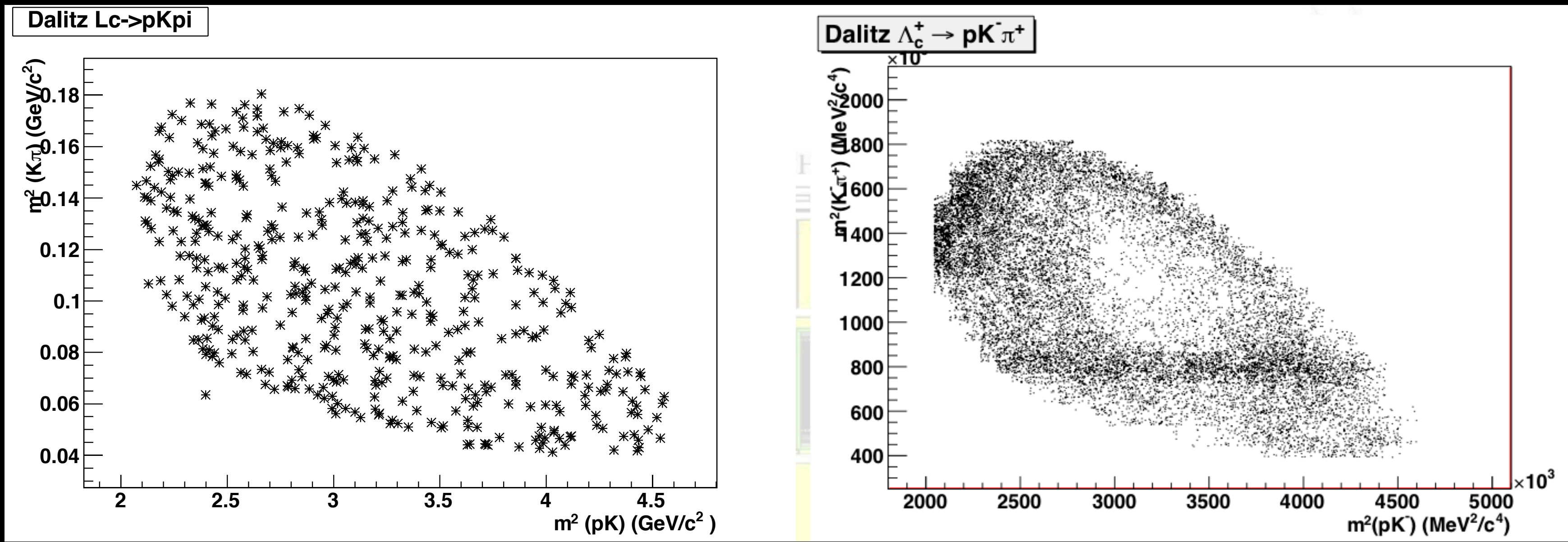
Lambda_c+ TMVA Selection Update

16/12/11

Lc->pKpi psuedo-resonant model

- Have had some very strange errors stopping me getting high stats in Dalitz plot. Approx. 500 entries but not enough to see if the MC is functioning properly.



MC

Data

- Now have the machinery in place to generate Dalitz plots directly from Gauss gen… if problem with large ntuples can be solved…

TMVA Selection Recap

- We had a ROC curve which looked suspect…
- Evidence TMVA was incorrectly configured.

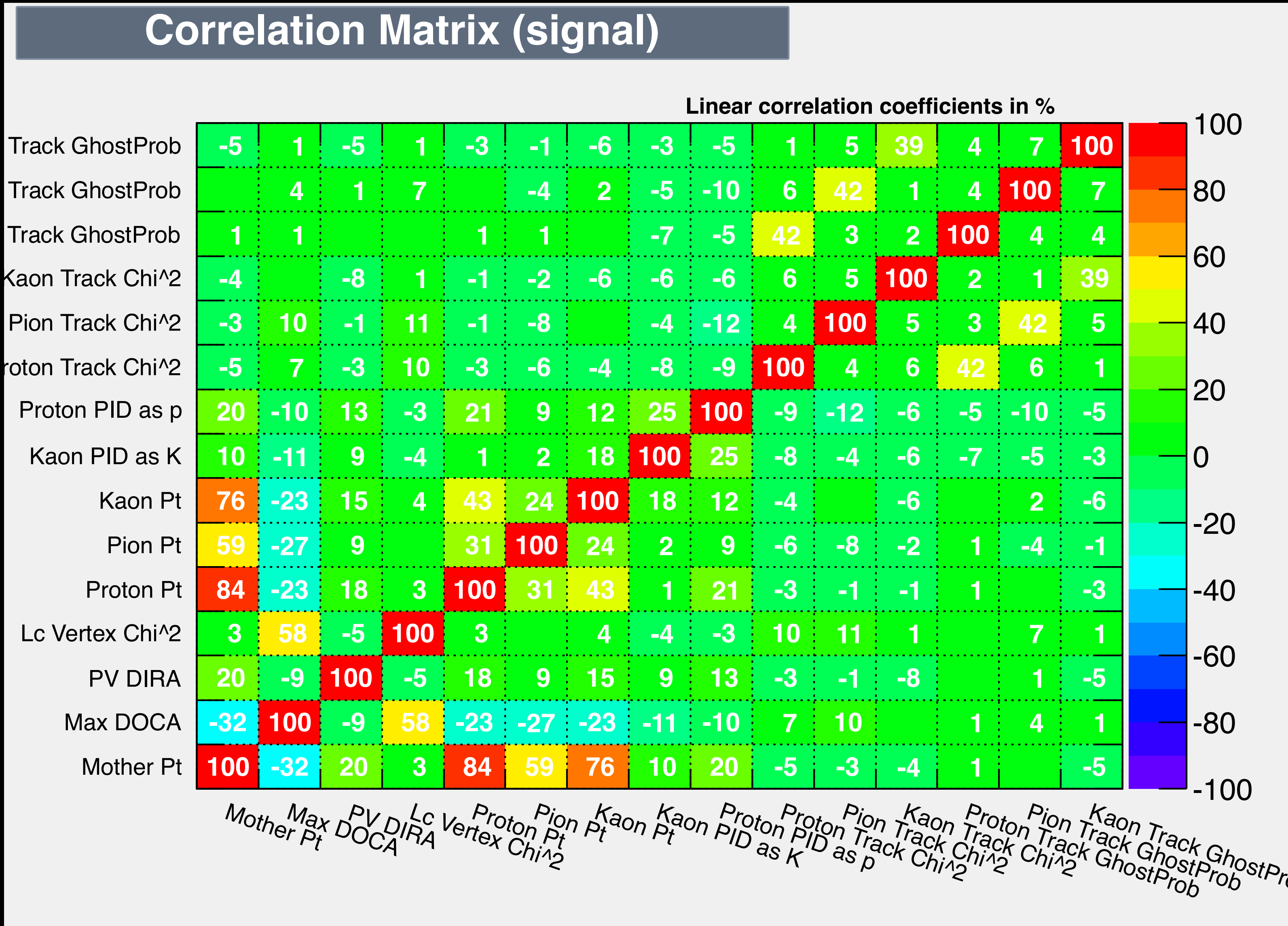
- Since then, looked at TMVA's behaviour in a variety of situations.

Summary of Developments

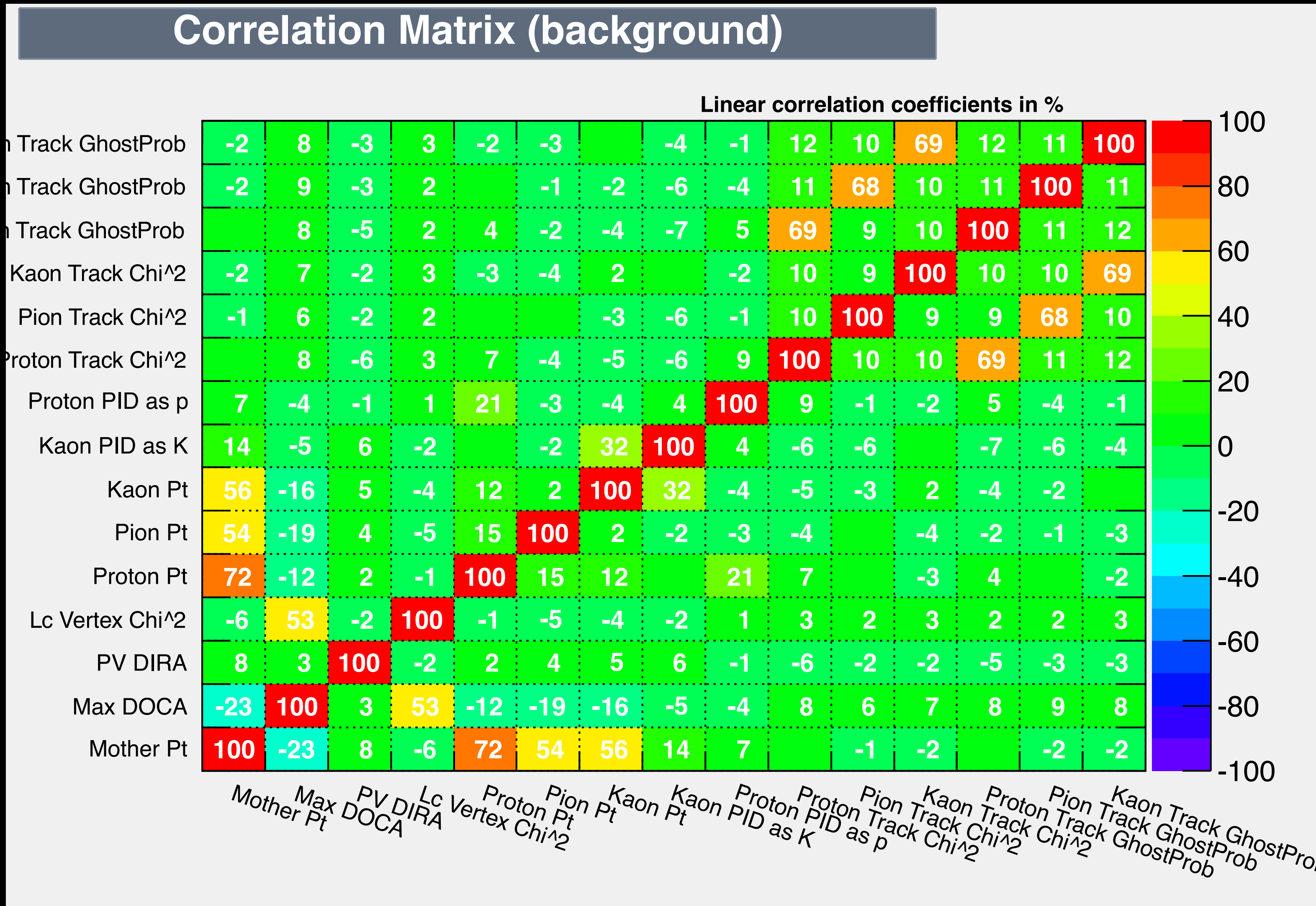
- Have since learned from Paul/Eduardo that things in TMVA are not so simple…
- Can only acquire values for cuts with the “Cuts” method, which was not found to be working last week. All MVA methods would have to be implemented manually in stripping.
- Unfortunate since MVA methods are found to be more discriminating.
- But with some prompting the “Cuts” method doesn’t spout gibberish…
- Trick seems to be in consideration of correlations and which variables can be trimmed.
- Last week made the mistake of thinking TMVA was smarter than it is…

Signal Correlations

- Was bugged last week, fixed now.



Background Correlations

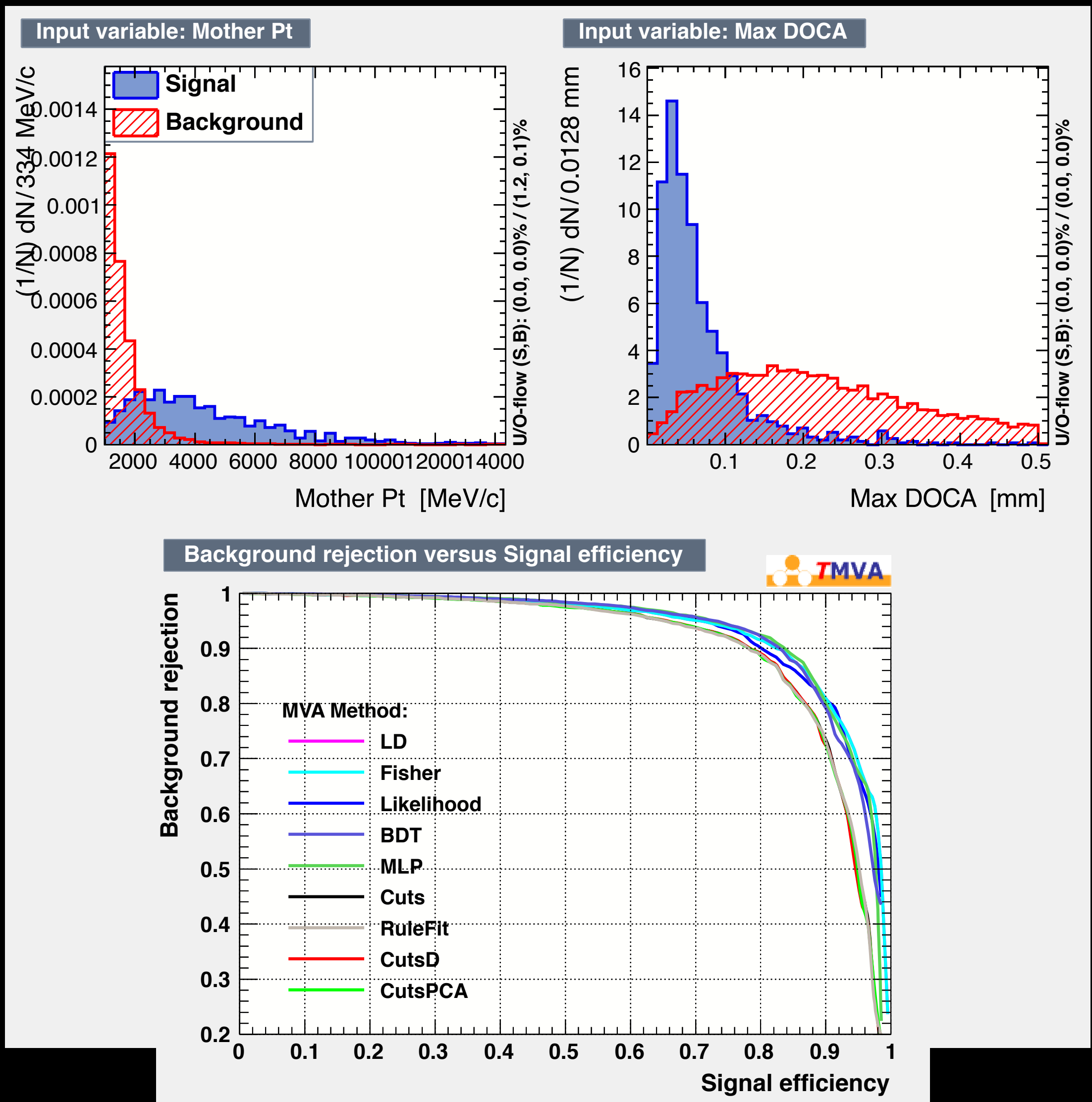


Correlation Differences

- Ghostprob and track chi^2 for daughters more strongly correlated in Bkg.
- Daughter Pts more correlated with each other in signal
- Daughter Pts more correlated with Mother Pt in signal.
- DOCA somewhat more correlated with Mother Pt in signal.
- Overall no big surprises except can't really explain the first point.

Simple 2 variable case:

- Should reduce to rectangular cuts from before.

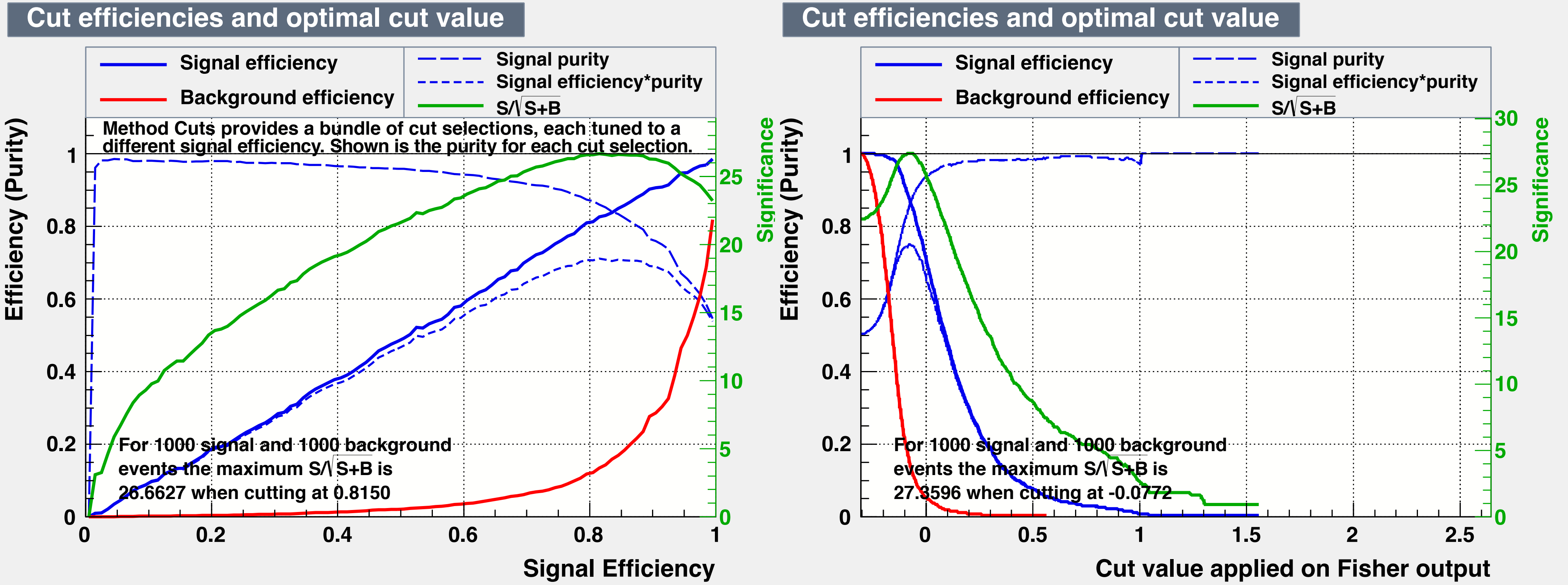


- Cuts competitive with MVA methods, but not quite as good.

```
: ------------------------------------------------------------------------
: Cut values for requested signal efficiency: 0.7
: Corresponding background efficiency       : 0.068548
: Transformation applied to input variables : None
: ------------------------------------------------------------------------
: Cut[ 0]:     2732.45 <         Lambda_cplus_PT <=       1e+30
: Cut[ 1]: 0.00182328 < Lambda_cplus_DOCAMAX <=       1e+30
: ------------------------------------------------------------------------
:
: ------------------------------------------------------------------------
: Cut values for requested signal efficiency: 0.8
: Corresponding background efficiency       : 0.126689
: Transformation applied to input variables : None
: ------------------------------------------------------------------------
: Cut[ 0]:     2273.09 <         Lambda_cplus_PT <=       1e+30
: Cut[ 1]: -0.00179459 < Lambda_cplus_DOCAMAX <=       1e+30
: ------------------------------------------------------------------------
:
: ------------------------------------------------------------------------
: Cut values for requested signal efficiency: 0.9
: Corresponding background efficiency       : 0.281297
: Transformation applied to input variables : None
: ------------------------------------------------------------------------
: Cut[ 0]:     1761.71 <         Lambda_cplus_PT <=       1e+30
: Cut[ 1]: -0.00201632 < Lambda_cplus_DOCAMAX <=       1e+30
: ------------------------------------------------------------------------
```

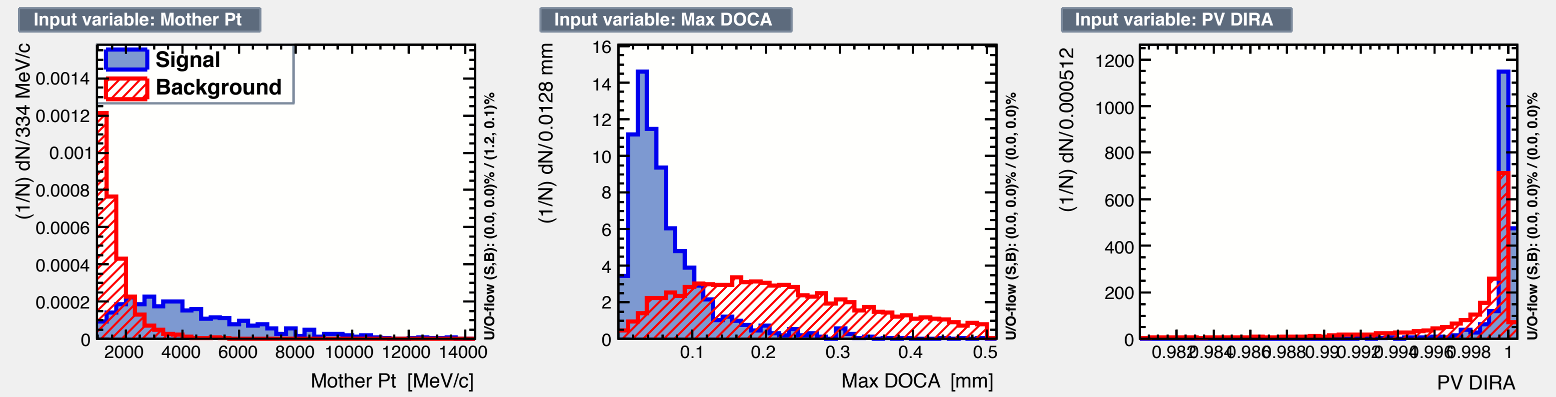
- Values less harsh than previous rect cuts, also –ve DOCA is suspect.

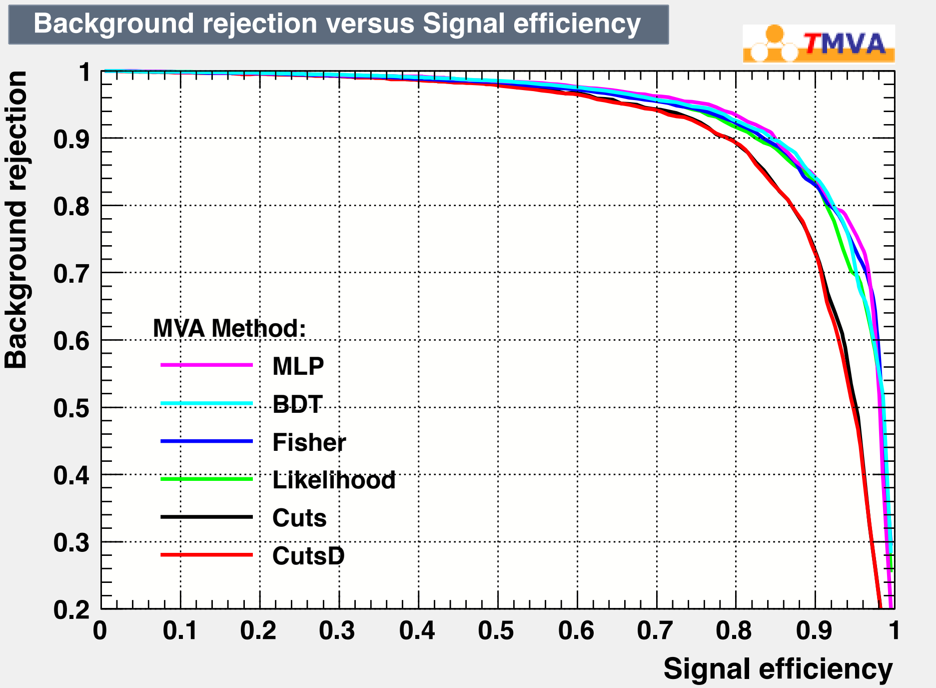
2 Vars -2



Cut efficiencies and optimal cut value
Signal efficiency
Background efficiency
Signal purity
Signal efficiency*purity
S/√S+B
Method Cuts provides a bundle of cut selections, each tuned to a different signal efficiency. Shown is the purity for each cut selection.
For 1000 signal and 1000 background events the maximum S/√S+B is 26.6627 when cutting at 0.8150
Efficiency (Purity)
Significance
Signal Efficiency
Cut efficiencies and optimal cut value
Signal efficiency
Background efficiency
Signal purity
Signal efficiency*purity
S/√S+B
For 1000 signal and 1000 background events the maximum S/√S+B is 27.3596 when cutting at -0.0772
Efficiency (Purity)
Significance
Cut value applied on Fisher output

3 Variable Case - 1

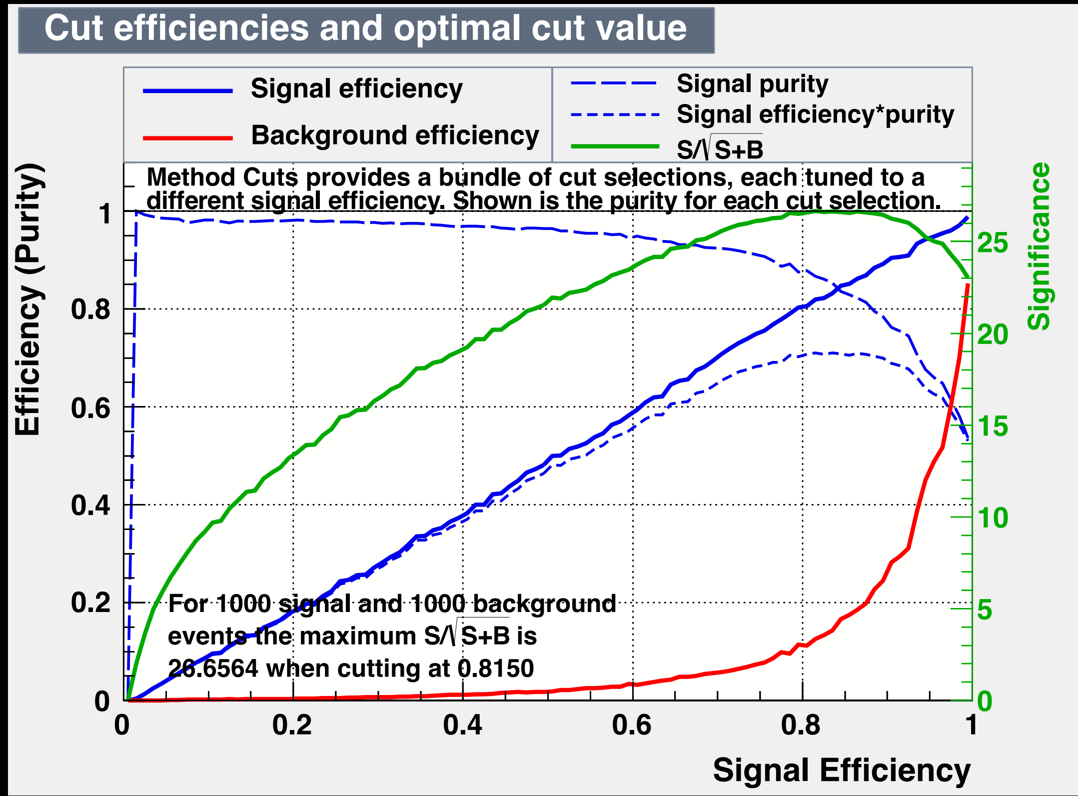




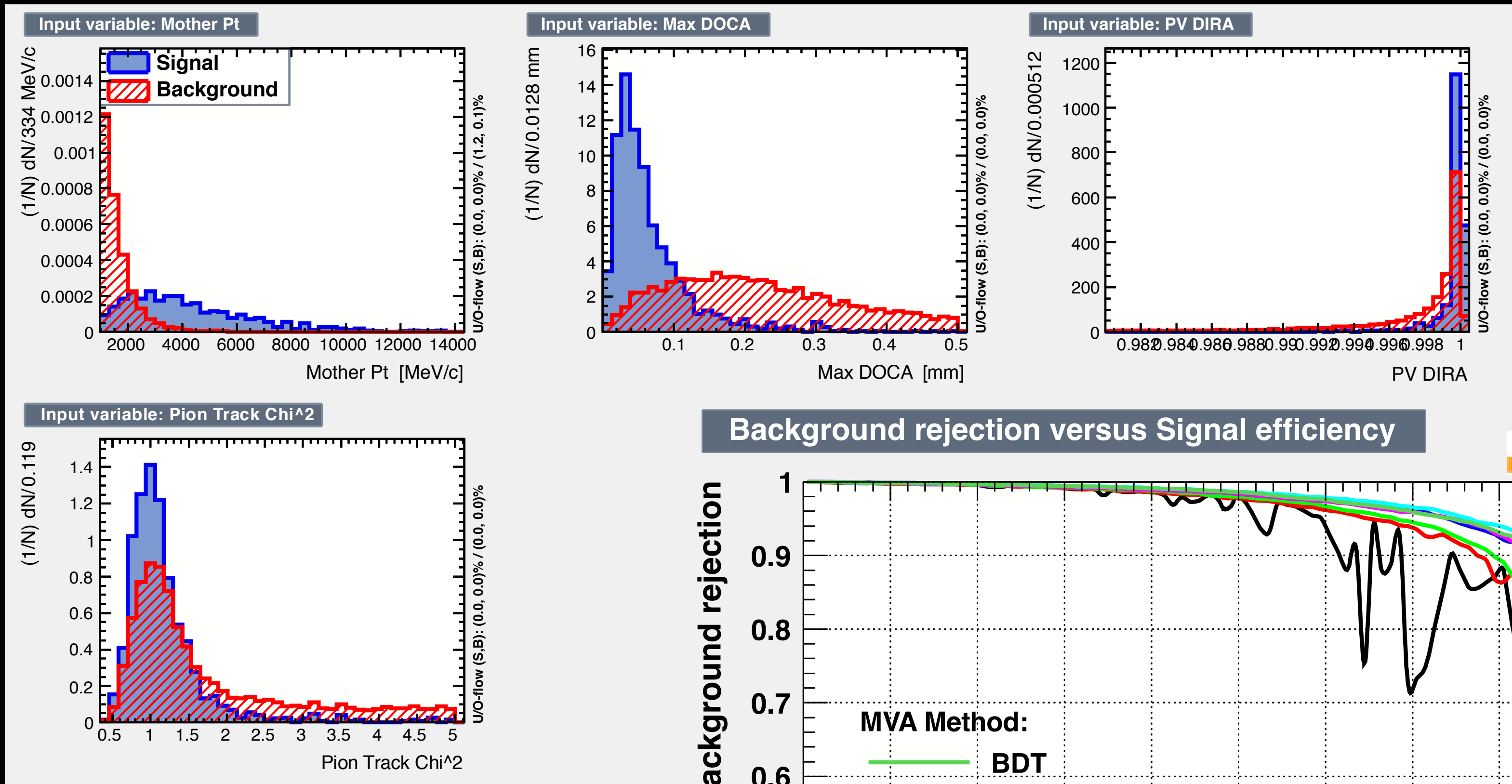
MVA staring to get better discrimination over cuts, cuts also well behaved here.

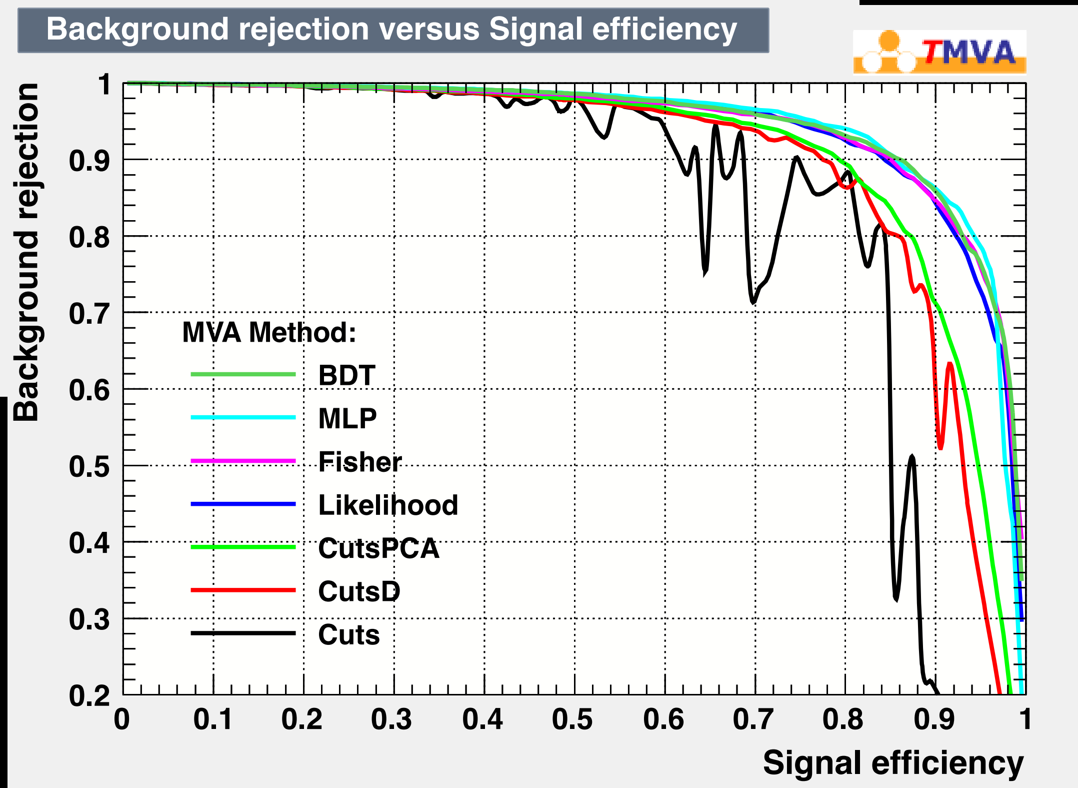
3 Variable Case

```
: ----------------------------------------------------------------
: Cut values for requested signal efficiency: 0.7
: Corresponding background efficiency       : 0.061443
: Transformation applied to input variables : None
: ----------------------------------------------------------------
: Cut[ 0]:     2674.7 <          Lambda_cplus_PT <=      1e+30
: Cut[ 1]: -0.000723353 <    Lambda_cplus_DOCAMAX <=      1e+30
: Cut[ 2]:   0.997771 < Lambda_cplus_DIRA_OWNPV <=      1e+30
: ----------------------------------------------------------------
: ----------------------------------------------------------------
: Cut values for requested signal efficiency: 0.8
: Corresponding background efficiency       : 0.117682
: Transformation applied to input variables : None
: ----------------------------------------------------------------
: Cut[ 0]:    2176.98 <          Lambda_cplus_PT <=      1e+30
: Cut[ 1]: 0.000466988 <    Lambda_cplus_DOCAMAX <=      1e+30
: Cut[ 2]:   0.997372 < Lambda_cplus_DIRA_OWNPV <=      1e+30
: ----------------------------------------------------------------
```



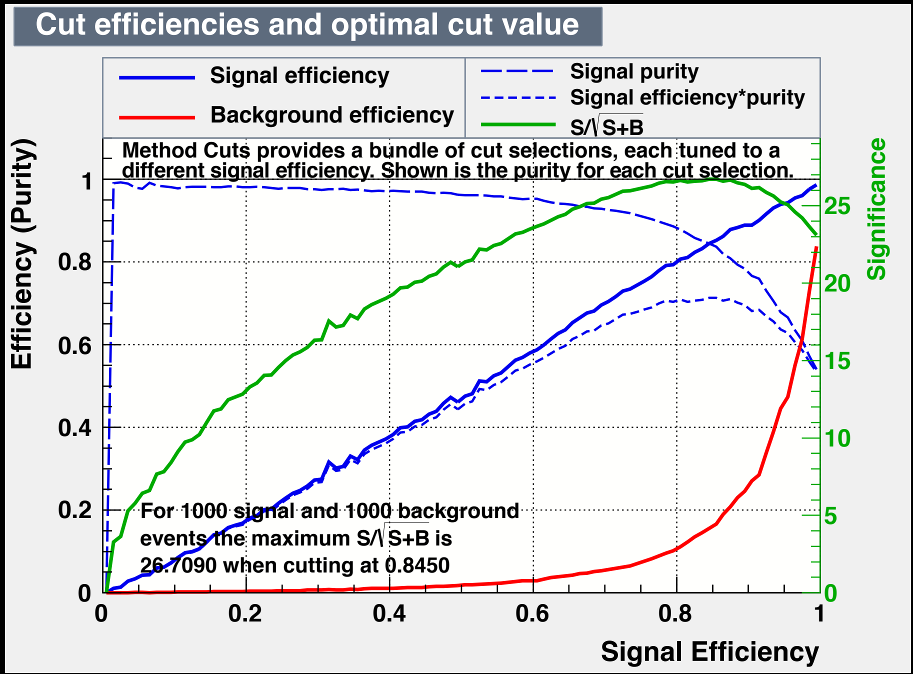
4 Variables

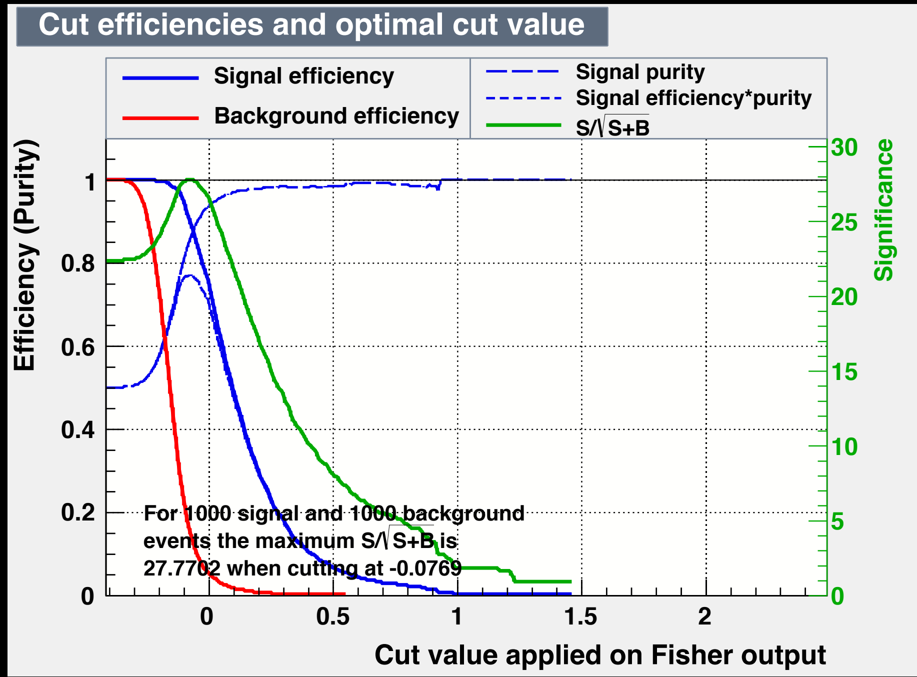




Begins to demonstrate benefits of PCA (principal component analysis).

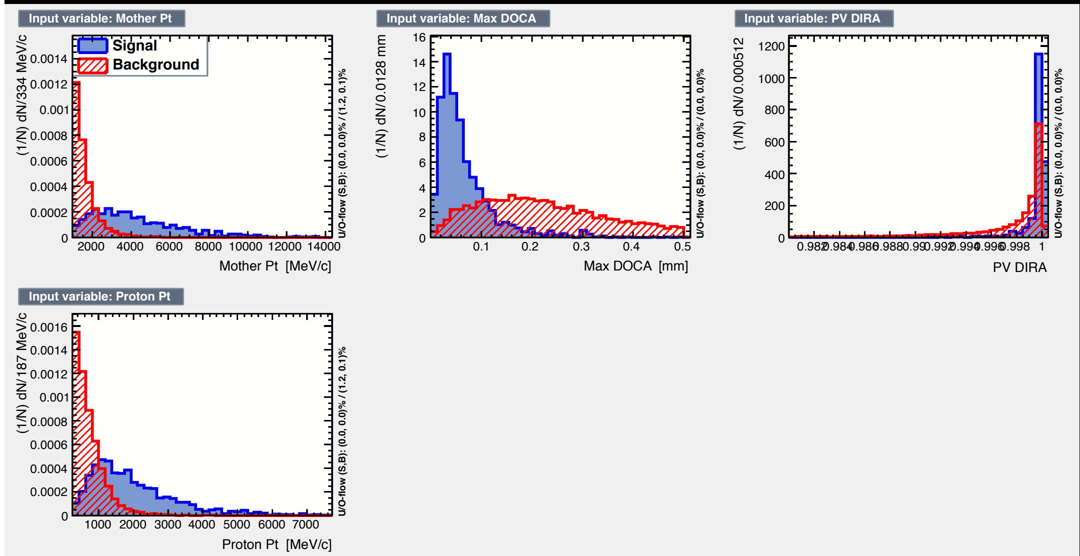
4 Variables 2

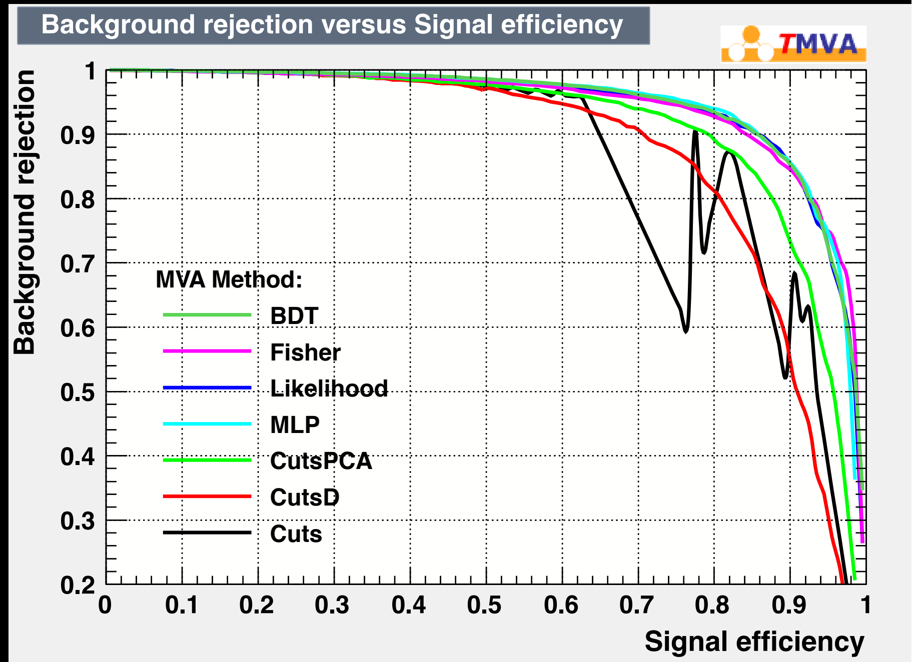


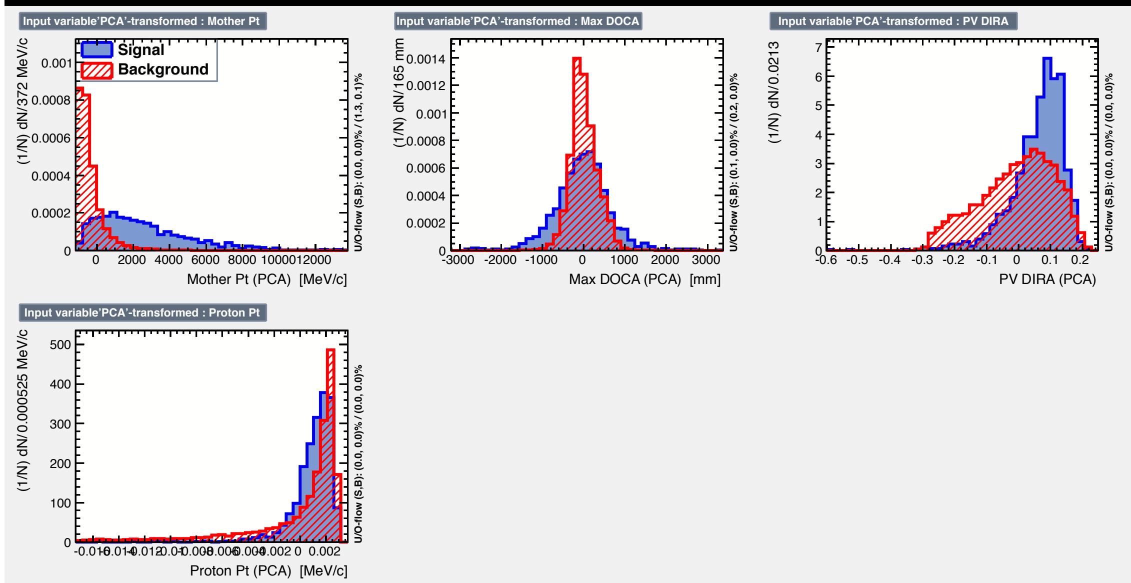


Still fairly competitive with MVA methods… but how easy is it to implement PCA-transfomred variables?

4 Variables with 1 strong correlation



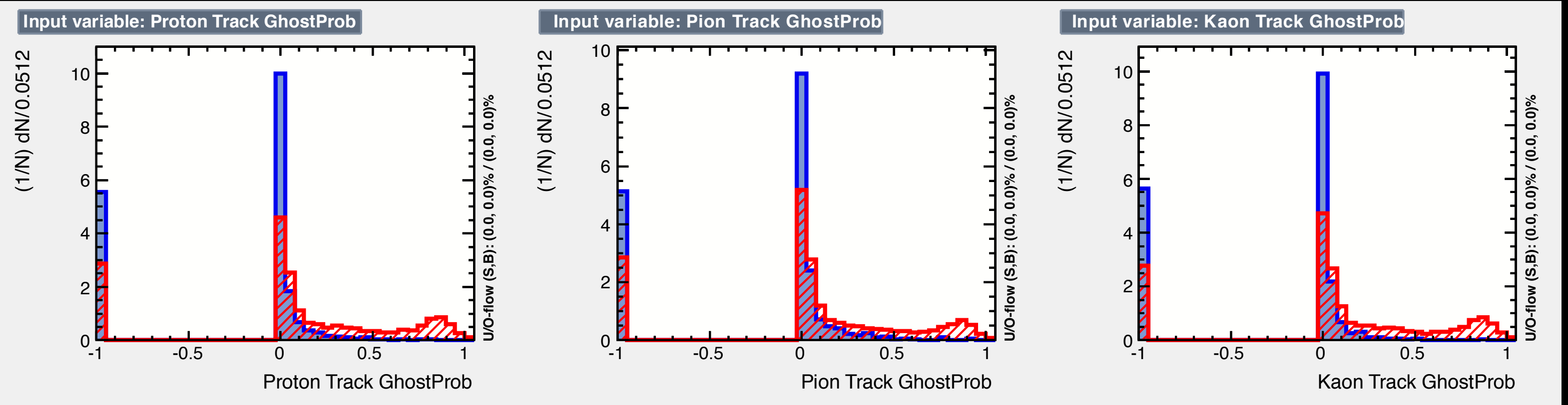




Replace pion track quality with Proton Pt, which is 84% correlated with Mother Pt, and behaviour is different. CutsD is actually better than before…

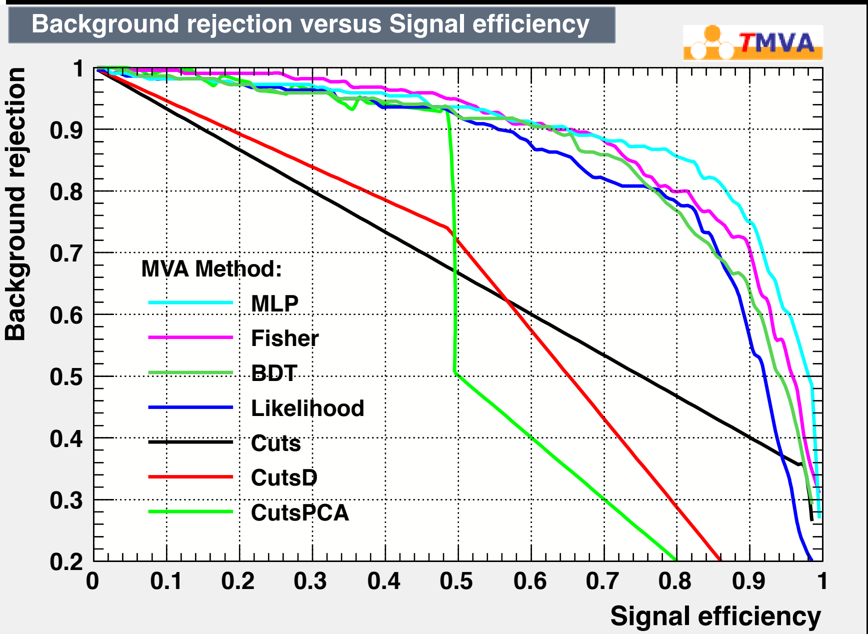
Trimming variables…

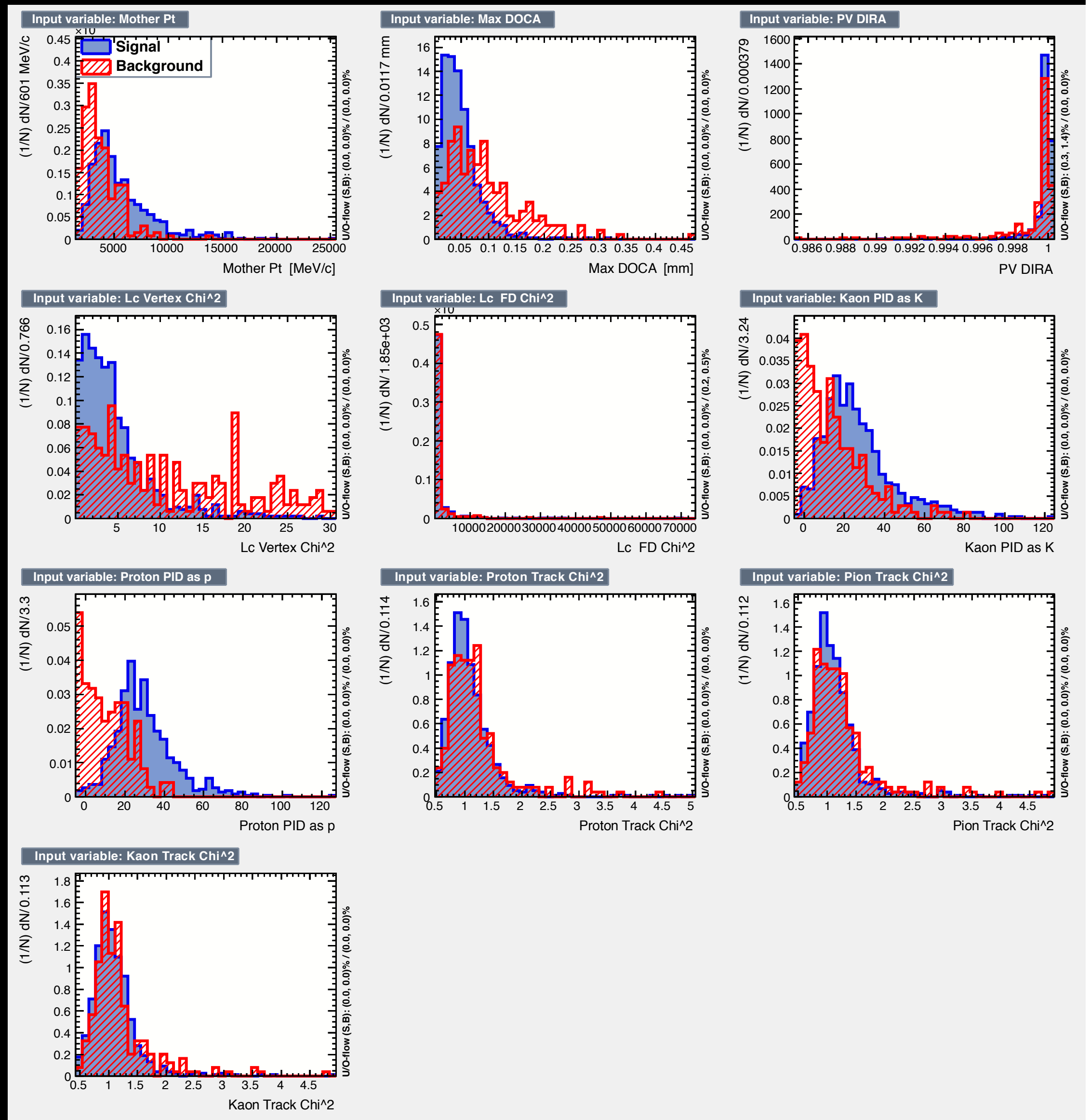
- Want to eliminate highly correlated variables and also to reduce dimensionality to imrpove Cuts-based methods efficiencies.
- Apply global cuts on ghostprob and daughter pt "by eye".
- All ghostprob <0.5 & Proton Pt > 1GeV & pi/K Pt >500MeV



But cuts gibberish with 10 variables… and MVA methods less effective…

Need to further trim which variables used.





Conclusions we can draw…

- Cuts based methods can be almost as good as MVA, IF we can easily implement decorrelated variables, I have no idea.
- To use any MVA method in stripping or trigger = a lot of work
- Can keep trying to get a sensible set of cuts using the most powerful discriminants with ~4 variables and apply global cuts using our previous optimisation knowledge, think this may be the best solution.