Simply Simplifying Simple Selections

Some experience and tools for quick and comprehensive analysis

C. Fitzpatrick (University of Edinburgh)

LHCb UK Student Meeting

C. Fitzpatrick

November 23, 2010

Outline

C. Fitzpatrick

November 23, 2010

Introduction

- This talk will present some of my experiences in developing selections for early data, specifically for the $\mathrm{D}_\mathrm{s}^+ \to \phi\pi^+$ analysis
- It will also be a shameless plug for the `Simpletools` package
- Both the talk and the tools presented are very general: They may not be ideal for your analysis
  - But they may help you starting out
  - and the code is freely modifiable
- I will include links to the code and a self-documented tutorial at the end of the talk

C. Fitzpatrick

November 23, 2010

Why select candidates?

- LHCb is an experiment on a hadron collider: Massive backgrounds!
- Most of our work deals with low BR suppressed or rare decays
- If you want signal in your ntuple, you'll need to improve the S/B
- Every analysis requires a selection
  - Usually 3 selections: Hlt, Stripping, Offline
- But how do you make these selections?

C. Fitzpatrick

November 23, 2010

What is a selection?

- At a basic level, a selection is an ensemble of cuts that enhance signal at the expense of background
  - Selections can be simple: A couple of cuts based on common sense and some kinematic considerations
  - They can also be crazy multivariate monstrosities using all the fancy bells and whistles
- The complexity of a selection really depends on what you want and resource constraints:
  - Hlt/Stripping: Low CPU time (so keep it simple)
  - Stripping: Retention requirements might be tight: Low rate, low CPU
  - Offline: Do you really want to manipulate a 20GB ntuple on your laptop?
  - Bias/Correlations: Can you get easily account for an acceptance in your fit?

C. Fitzpatrick

November 23, 2010

Starting out

- The selection is usually the starting point of your analysis
- Changing it later can be a lot of work
- Before starting out, it makes a lot of sense to think about your requirements
- You'll need to justify/defend your choice of selection countless times: Your supervisor, working group, the collaboration, editors will all want to know why you used that cut!
- You should think about what variables you might want to avoid cutting on
- And will probably find unexpected ones along the way...
- In my experience starting simple and increasing the complexity over time works best
  - Do you really need a tactical nuclear warhead to crack a nut?

C. Fitzpatrick

MC or Data?

- In order to develop a selection you need to know what signal/background look like
- This is achieved using a Discriminating variable
- In MC this is just the truth: BKGCAT==0 is signal, BKGCAT!=0 is background
- in Data choose an uncorrelated variable you can easily model, eg: mass distributions

MC

- Even if it isn't exactly like detector data, it's often close enough. DC'06 selections are still in use now
- It gives you a good feel for (obvious) correlations and what variables you might want to cut on
- A lot of selections are completely defined/optimised on MC

Data

- If you don't have any signal MC, have reason not to trust MC or are feeling brave you can use Data
- You need to understand your signal and background beforehand though: Peaking backgrounds could lead to problems in a selection optimisation
- Such a technique is generally harder to justify, but can boost your efficiency by bypassing MC/Data differences

C. Fitzpatrick

November 23, 2010

A general selection strategy

1. Work out what your requirements are
2. Look at MC/Data with a very loose by-eye preselection
   - ensure you've not cut too hard already
   - or on variables you shouldn't be cutting on
3. From this preselection determine what variables have good separation power
4. Check these variables for correlations to anything you want to fit to
5. Remove any variables you think are potentially dangerous
6. Subject the remaining variables to your favourite optimisation technique
7. Look closely at the results, make sure they're reproducible/stable

C. Fitzpatrick

November 23, 2010

Choosing a preselection

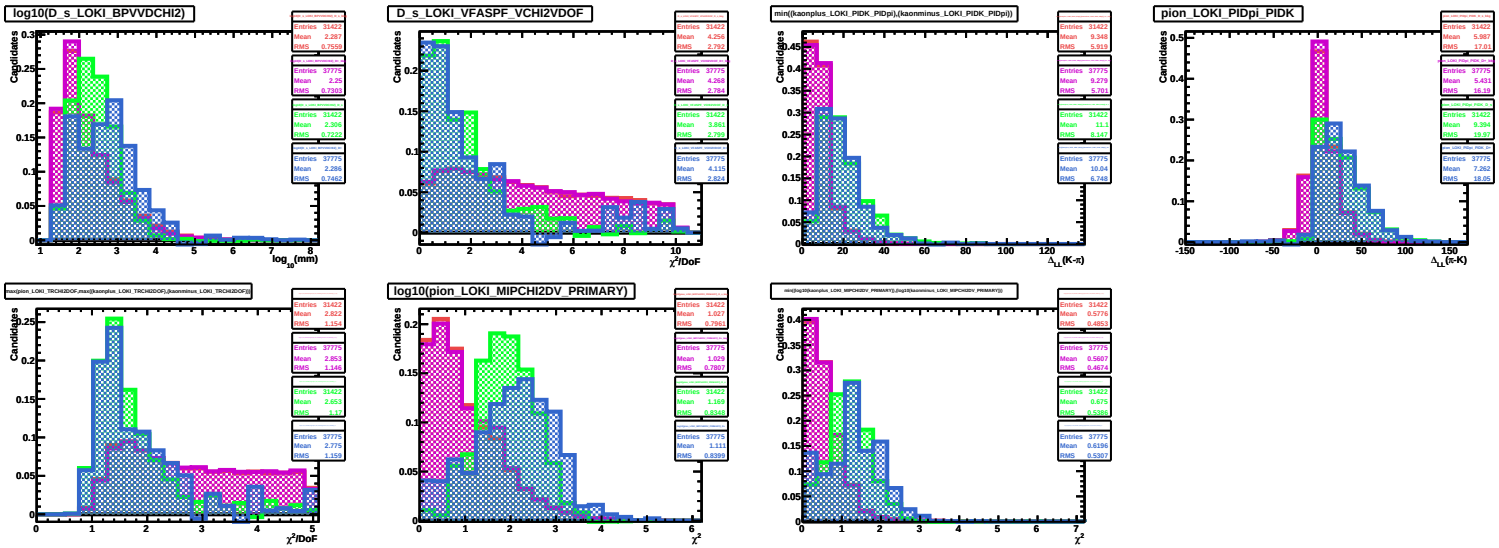
- This is usually the trickiest part:
  - If you make a mistake here it means re-running on the grid :(
  - You're all physicists, use your physics intuition and understanding
  - Track quality cuts, clone rejection, vertex isolation and quality are good "general" cuts
  - Mass windows: Make sure they're much larger than 5σ so you can use sidebands
  - Look at particle lifetimes eg: K^0_S have a large $c\tau$ so use vertex distance cuts (but be careful of lifetime bias!)
  - Make sure you've got the right PV using DIRA, etc
- The key is to cut loose at preselection- give your selection room to be optimised over

C. Fitzpatrick

November 23, 2010

Finding cut variables to use

- Once you have a preselection you need to work out how best to "feed" your optimiser
- The human brain is a great pattern-recognition tool. Plotting your signal and background distributions normalised to the same area on a layered histogram works well
- For distributions with long tails, taking the log of the *y* axis is helpful



- Digging through dozens of distributions looking for variables that have separating power is time consuming. Is there an easier way?

C. Fitzpatrick

November 23, 2010

Separation Power

- While by-eye will find you the variables, there is an analytical technique
- Define the separation power $\delta(v)$ as:

Separation Power

$$\delta(v) = \frac{1}{2}\int \frac{(f_s(v) - f_b(v))^2}{f_s(v) + f_b(v)} dv$$

- Where f_s, f_b are the signal and background distributions of variable v
- For distributions that are well separated, $\delta(v)$ is large. For similar distributions, $\delta(v)$ is close to 0.
- Sorting by descending power and taking the top variables gets you the ones that have the most to offer

C. Fitzpatrick

November 23, 2010

Correlations and bias

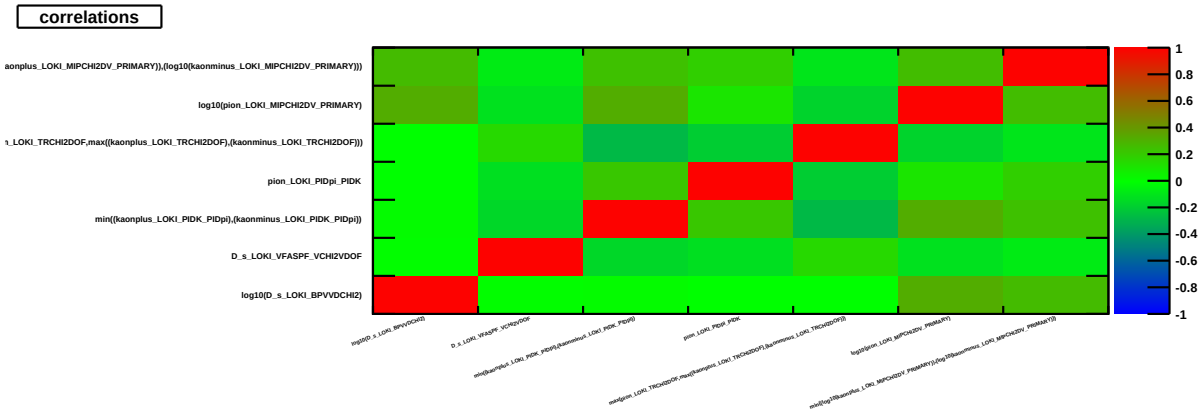
- Now we have a list of variables that when cut on remove more background than signal
- Feeding these to your optimiser would give you a lovely clean sample. . .
- but your signal might be badly distorted from bias
- you need to look at correlations between your cut variables and the distributions you want to fit to
- note: "Correlation" usually means a linear deviation in one variable as a function of the other
  - We usually abuse the term in HEP to mean "**any** deviation in one variable as a function of the other"
  - Two variables with 0% linear correlation may still be entangled in some nonlinear fashion
  - Just because someone states a correlation is "small" in percent, it doesn't mean you are safe to use this variable!

C. Fitzpatrick

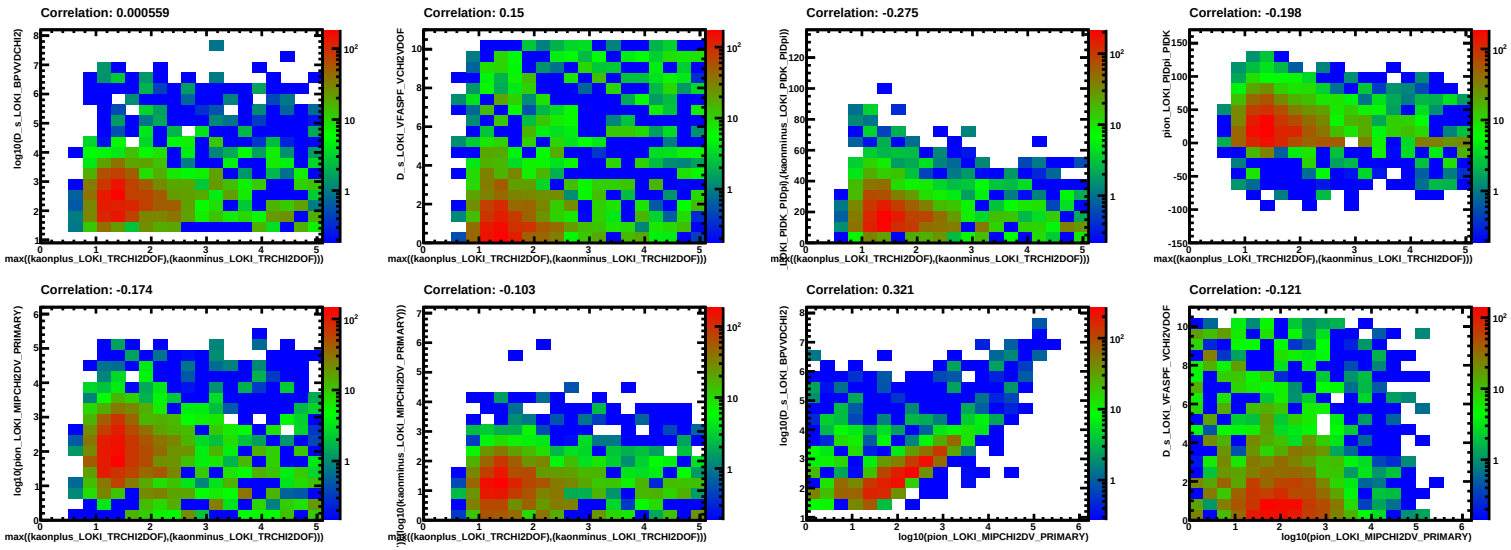
November 23, 2010

Spotting correlations

- As an obvious check, look at the linear correlations between variables first:



- The most powerful technique is to plot the 2D scatter between variables:



C. Fitzpatrick

November 23, 2010

Optimisation techniques

- In general, some form of selection optimisation is always performed:

By-eye tuning

- Cuts are modified by hand, looking at how your discriminating variable changes
- Hard to justify on anything but the cleanest modes or most basic analyses
- Unlikely that you will arrive at an "optimal" ensemble by-eye, and takes time

Rectangular cut optimisation (Semi-multivariate)

- Cuts are moved to maximise a "Figure of Merit" (FoM)
- Automatic, Fast, difficult to overtrain
- Easy to understand: The result is a list of optimal cuts

Full-blown Multivariate techniques (MVA)

- Almost always more powerful (sometimes MUCH more powerful) than simple rectangular cuts
- Not always easy to understand, easy to overtrain
- Difficult to explain/justify to the "old farts" of the collaboration

C. Fitzpatrick

November 23, 2010

Choosing a FoM

- For any optimisation procedure you need to have a target to optimise towards
- This can be something simple, like "As much signal as possible"
- It can also be something much more complicated, like "minimise the error on a fit parameter"
- These need to be encapsulated into some form of Figure of Merit (FoM)
- Typical FoMs are:
  - Most popular, the "Signal Significance": $S/\sqrt{S+B}$ is an estimate of the cleanliness of signal accounting for signal and background statistics.
  - For rare decays or where ($S << B$), $S/\sqrt{B}$ is sometimes used
  - In bandwidth limited situations (trigger, stripping), it is more common to ask "Give me as much signal as possible in X kHz" (If your signal is high-rate the cuts will be tight and the result pure, if your signal is low rate you get loose cuts and higher backgrounds)
- From an analysis perspective the best bet is to try and minimise the error on your final result. If you can derive the formula for the error in terms of calculable parameters, invert it and feed it to your optimiser. Often, this is very similar to $S/\sqrt{S+B}$.

C. Fitzpatrick

November 23, 2010

What tools are out there?

- Selection optimisation isn't new: The Tevatron and LEP experiments, BaBar, and many others have had to do this too
- It means there's already a lot of packages out there which allow you to investigate and optimise cuts.
- Some popular ones:
- TMVA
  - Does everything: Allows you to test many MVA techniques to pick the best
  - Reasonably easy to use, comes bundled with ROOT
  - Suffers from bloat: Does everything OK instead of one thing well
  - Slow, and can't handle large datasets or lots of cuts
- neurobayes
  - Neural networks with powerful preprocessing
  - Looks really promising, ask our resident NeuroShill U. Kerzel
  - Non-free, closed source. You need a license (CERN has some)
  - Not so easy to go from ROOT Ntuple to NeuroBayes input
- Because these are experiment-agnostic and very general, getting results from them quickly and efficiently isn't all that easy. There's also a learning curve associated with both.
- Is there an easier way to get a selection quickly?

C. Fitzpatrick

What is CROP?

- CROP is the Cut Recursive OPtimiser. It aims to:
  - Optimise an arbitrarily large set of cuts
  - on arbitrarily many signal/background datasets of any size
  - in a way that anyone can understand
  - in the easiest, fastest way possible
- CROP falls into our second category of optimisation techniques
- It recursively optimises an ensemble of rectangular cuts maximising a Figure of Merit
- The FoM can be supplied by the user. By default this is $S/\sqrt{S+B}$
- While running, it produces useful diagnostics and plots in real-time
- CROP works with both MC and Data: to do your analysis MC-free you just need a simple fit.
- C++ compiled binaries: Any computer with ROOT installed can run it
- Fully modular and object-oriented code: easy to examine and modify

C. Fitzpatrick

November 23, 2010

How does it work?

- Datasets are loaded from simple text-based weightfiles, where cuts and average or per-event weighting can be specified

```
#S/B    file                        ntuple  weight   cut  legend   fill    color
S       dsSweights_presel.root      merged  NSig_sw  1    D_s      3001    77
S       dplusSweights_presel.root   merged  NSig_sw  1    D+       3001    855
B       dsSweights_presel.root      merged  NBkg_sw  1    D_s_bkg  3001    207
B       dplusSweights_presel.root   merged  NBkg_sw  1    D+_bkg   3001    617
```

- the "Weight" column can be numeric to define an average weighting, or it can be a formula taken from ntuple columns to make a per-event weight
- Cuts to optimise are defined in a text-based cutfile, along with min, max, resolution for the cut

```
#cut                                                    min   max   res
log10(D_s_LOKI_BPVVDCHI2)>                              0.9   8.2   0.073
D_s_LOKI_VFASPF_VCHI2VDOF<                              0     11    1.0
min((kaonplus_LOKI_PIDK_PIDpi),(kaonminus_LOKI_PIDK_PIDpi))> 0     138   1.0
pion_LOKI_PIDpi_PIDK>                                   -150  170   1.0
max(pion_LOKI_TRCHI2DOF,max((kaonplus_LOKI_TRCHI2DOF),(ka... 0     5.1   0.1
log10(pion_LOKI_MIPCHI2DV_PRIMARY)>                     0     6.2   0.1
min((log10(kaonplus_LOKI_MIPCHI2DV_PRIMARY)),(log10(kaonm... 0     7.2   0.1
```

- The last character of your cut needs to be ">" or "<" to define what side to keep.
- these can be arbitrarily complicated: `cos(x)<`, `abs(x-y)>` etc.
- If you can use it as a `TCut` in ROOT, CROP will understand it.

C. Fitzpatrick

November 23, 2010

Running CROP

- CROP takes a few commandline arguments. For help, don't specify any:

```
conor@maxwell:~$ crop
  _________  ____  ___
 / ___/ _ \/ __ \/ _ \
/ /__/ , _/ /_/ / ___/
\___/_/|_|\____/_/
The Cut Recursive OPtimiser

crop: Part of the SimpleTools Package v1.0k
Author: Conor Fitzpatrick conor.fitzpatrick@cern.ch
Optimises a set of rectangular cuts
Usage:
crop <WeightFile> <cutfile> <steps> <OrderMethod> <output.root> [-b]
```

- the weightfile and cutfile you already know about
- `steps` tells CROP how long it should run for before bailing out
  - CROP will automatically finish if it converges on an optimal ensemble
  - This just tells it "If you've failed to converge after N steps, give up"
- `OrderMethod` defines how crop reorders the cuts at each step:
  - S: Orders in ascending signal efficiency
  - B: Orders in ascending background efficiency
  - SB: Orders as product of the above (usually fastest)
  - O: Orders as per the cutfile
  - R: Orders randomly
- `output.root` is the location to which diagnostic plots are saved
- `-b` is optional, and specifies batch mode
- If you call crop without it, you get real-time updated plots displayed on-screen

C. Fitzpatrick

November 23, 2010

Initialisation

- CROP loads and parses your weightfile, displaying how many entries and corresponding weighted entries there are in it

```
INFO: ========================================================================
INFO: D_s is a dataset of type Signal
INFO: ------------------------------------------------------------------------
INFO:   File Path: ../dsSweights_presel/dsSweights_presel.root
INFO:   Ntuple Path: merged
INFO:   Total Entries: 31422
INFO:   Using per-event weight of NSig_sw
INFO:   Total Weighted Entries: 2020.65+/-89.411
INFO: ========================================================================
INFO: D+ is a dataset of type Signal
INFO: ------------------------------------------------------------------------
INFO:   File Path: ../dplusSweights_presel/dplusSweights_presel.root
INFO:   Ntuple Path: merged
INFO:   Total Entries: 37775
INFO:   Using per-event weight of NSig_sw
INFO:   Total Weighted Entries: 1028.66+/-88.2992
INFO: ========================================================================
INFO: D_s_bkg is a dataset of type Background
INFO: ------------------------------------------------------------------------
INFO:   File Path: ../dplusSweights_presel/dplusSweights_presel.root
INFO:   Ntuple Path: merged
INFO:   Total Entries: 37775
INFO:   Using per-event weight of NBkg_sw
INFO:   Total Weighted Entries: 36746.3+/-208.601
INFO: ========================================================================
Type Entries    Proc.      Proc. Eff. Weighted Entries Proc. Weighted Entries ratio Name
------------------------------------------------------------------------
S    3.14e+04   3.14e+04   1+/-   0   2.02e+03+/-89.4  2.02e+03+/-89.4    1+/-   0     D_s
S    3.78e+04   3.78e+04   1+/-   0   1.03e+03+/-88.3  1.03e+03+/-88.3    1+/-   0     D+
S    6.92e+04   6.92e+04   1+/-   0   3.05e+03+/- 126  3.05e+03+/- 126    1+/-   0     TOTAL
B    3.14e+04   3.14e+04   1+/-   0   2.94e+04+/- 188  2.94e+04+/- 188    1+/-   0     D_s_bkg
B    3.78e+04   3.78e+04   1+/-   0   3.67e+04+/- 209  3.67e+04+/- 209    1+/-   0     D+_bkg
B    6.92e+04   6.92e+04   1+/-   0   6.61e+04+/- 281  6.61e+04+/- 281    1+/-   0     TOTAL
```


C. Fitzpatrick

November 23, 2010

First step

- Once the datasets and cuts are loaded, CROP finds an initial working point
- This is done by calculating the maximum figure of merit for each cut "autistically" (in the **absence** of any other cut)

```
Using Purity Ordering
INFO: Initialising: Finding initial working point...
order  number  Sig. Eff        Bkg. Rej        FoM              cut
--------------------------------------------------------------------------------
0       0       0.863+/-0.00623 0.377+/-0.00188 12.6+/-0.494    log10(D_s_LOKI_BPVVDCHI2)>1.85
1       1       0.825+/-0.00688 0.605+/-0.0019  14.9+/-0.522    D_s_LOKI_VFASPF_VCHI2VDOF<3...
2       2       0.606+/-0.00885 0.891+/-0.00121 19.4+/-0.631    min((kaonplus_LOKI_PIDK_PID...
3       3       0.355+/-0.00866 0.945+/-0.000885 15.8+/-0.658    pion_LOKI_PIDpi_PIDK>33
4       4       0.695+/-0.00834 0.726+/-0.00173 14.9+/-0.539    max(pion_LOKI_TRCHI2DOF,max...
5       5       0.645+/-0.00867 0.82+/-0.00149  16.7+/-0.574    log10(pion_LOKI_MIPCHI2DV_P...
6       6       0.583+/-0.00893 0.893+/-0.0012  18.9+/-0.63     min((log10(kaonplus_LOKI_MI...
```

- The cuts are then sorted according to the ordering method described earlier, and all but the first one are applied to the datasets
- This has the effect of overshooting the optimal FoM, but puts us "in the neighbourhood" by reducing the proximity to other local maxima

C. Fitzpatrick

November 23, 2010

Optimisation

- The highest-ordered cut is now reoptimised in the **presence** of all other cuts, and applied at this value
- The next-highest ordered cut this step gets the same treatment with all other cuts plus the one just reoptimised
- This continues until the lowest ordered cut is reached, and the cuts are reordered again at this point
- This concludes one step of the procedure

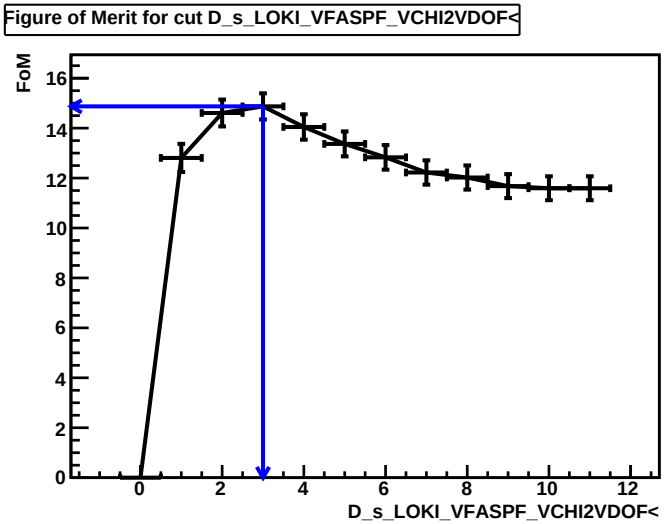
```
INFO: Starting Optimisation....
INFO: STEP NUMBER: 0 OF 10:
convergency? order  number  Sig. Eff        Bkg. Rej        FoM              cut
-------------------------------------------------------------------------------------
0            0      3       0.993+/-0.00369 0.115+/-0.0289  21.2+/-1.11     pion_LOKI_PIDpi_...
0            1      6       0.926+/-0.0093  0.48+/-0.0257   24.1+/-1.08     min((log10(kaonp...
0            2      2       0.934+/-0.00755 0.307+/-0.0195  27.1+/-1.03     min((kaonplus_LO...
0            3      5       0.966+/-0.00504 0.401+/-0.0169  29.9+/-1.03     log10(pion_LOKI_...
0            4      4       0.93+/-0.00622  0.346+/-0.0134  32+/-0.987      max(pion_LOKI_TR...
0            5      1       0.898+/-0.00689 0.467+/-0.0114  33+/-0.969      D_s_LOKI_VFASPF_...
1            6      0       0.918+/-0.00631 0.382+/-0.012   33+/-0.969      log10(D_s_LOKI_B...
```

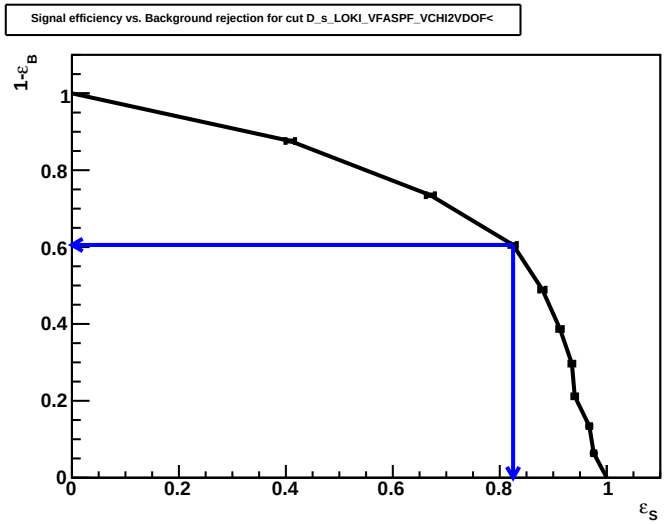

C. Fitzpatrick

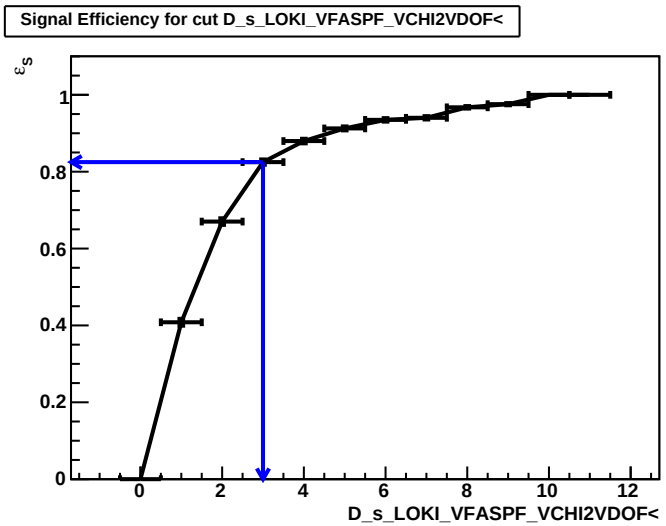
November 23, 2010

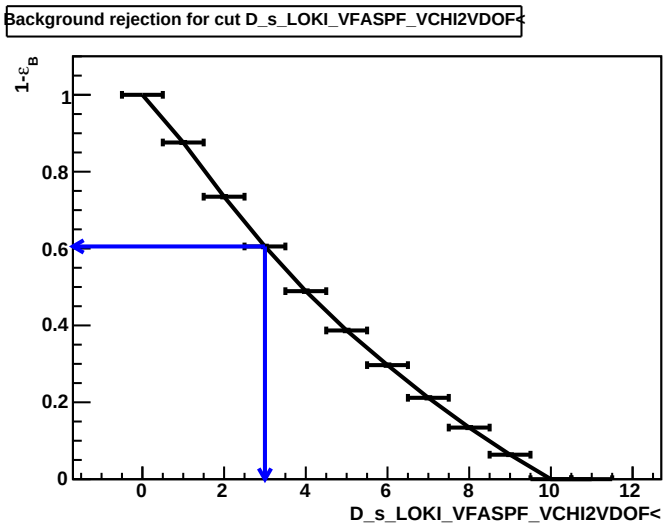
Realtime plots

- if you've not called crop with the "-b" option, plots are produced and displayed at each iteration:









C. Fitzpatrick

November 23, 2010

Convergency

- As the procedure continues, the cuts stabilise, gradually moving to values that maximise the ensemble FoM
- The `convergency` column indicates which cuts have not changed since the last step

```
INFO: STEP NUMBER: 5 OF 10:
convergency? order  number  Sig. Eff        Bkg. Rej        FoM              cut
-------------------------------------------------------------------------------------
1            0      6       0.88+/-0.00731  0.641+/-0.00959 33.8+/-0.991    min((log10(kaonp...
1            1      5       0.95+/-0.00511  0.53+/-0.0114   33.8+/-0.991    log10(pion_LOKI_...
1            2      1       0.904+/-0.00674 0.465+/-0.0122  33.8+/-0.991    D_s_LOKI_VFASPF_...
1            3      2       0.921+/-0.0062  0.422+/-0.0125  33.8+/-0.991    min((kaonplus_LO...
1            4      4       0.936+/-0.00569 0.34+/-0.0128   33.8+/-0.991    max(pion_LOKI_TR...
1            5      0       0.977+/-0.00359 0.203+/-0.012   33.8+/-0.991    log10(D_s_LOKI_B...
1            6      3       0.99+/-0.00239  0.143+/-0.0108  33.8+/-0.991    pion_LOKI_PIDpi_...
```

- You can see here that a result of the ordering procedure is to move less effective cuts down the list
- This is intentional- the cuts at the top of the list free up more events when opened, allowing the ones lower down to benefit from higher stats
- This greatly speeds up the convergency: Using random ordering takes on average twice as many steps

C. Fitzpatrick

November 23, 2010

Finalisation

- When CROP has converged, it prints the inclusive and exclusive efficiencies of each cut as applied to each input dataset, as well as totals and time taken
- This is ordered by background rejection, so useless cuts appear last

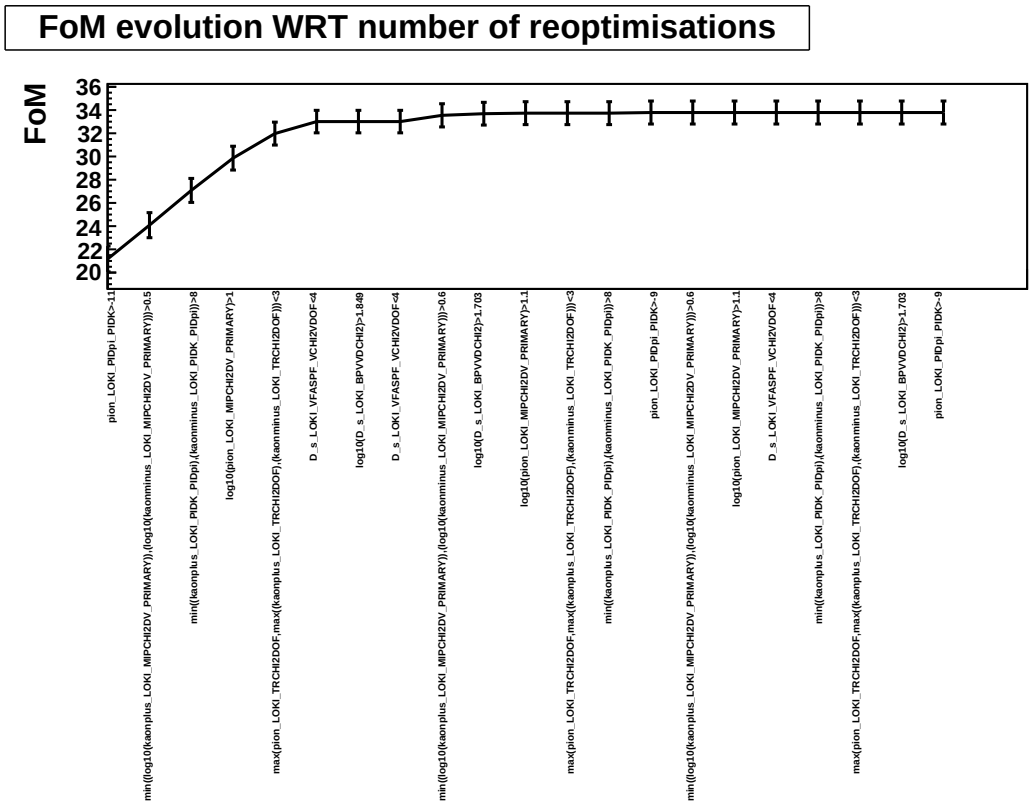
```
min((log10(kaonplus_LOKI_MIPCHI2DV_PRIMARY)),(log10(kaonminus_LOKI_MIPCHI2DV_PRIMARY)))>0.6:
Type    Excl. Cands.    Excl. Eff.      Incl. Cands.    Incl. Eff.      Name
S       1.72e+03+/-64.3 0.852+/-0.0079  1.19e+03+/-38.1 0.875+/-0.009   D_s
S        849+/-58.7     0.825+/-0.0118   548+/-27.3     0.893+/-0.0125  D+
S       2.57e+03+/-  87 0.843+/-0.00659 1.73e+03+/-46.9 0.88+/-0.00731  TOTAL
B       1.11e+04+/- 116 0.378+/-0.00283  415+/-26.1     0.363+/-0.0142  D_s_bkg
B       1.36e+04+/- 127 0.369+/-0.00252  484+/-26.1     0.355+/-0.013   D+_bkg
B       2.47e+04+/- 172 0.373+/-0.00188  899+/-36.9     0.359+/-0.00959 TOTAL

log10(pion_LOKI_MIPCHI2DV_PRIMARY)>1.1:
Type    Excl. Cands.    Excl. Eff.      Incl. Cands.    Incl. Eff.      Name
S       1.83e+03+/-64.6 0.906+/-0.00649 1.19e+03+/-38.1 0.949+/-0.00625 D_s
S        851+/-59.4     0.827+/-0.0118   548+/-27.3     0.953+/-0.00885 D+
S       2.68e+03+/-87.7 0.879+/-0.0059  1.73e+03+/-46.9 0.95+/-0.00511  TOTAL
B       1.07e+04+/- 114 0.365+/-0.00281  415+/-26.1     0.497+/-0.0173  D_s_bkg
B       1.39e+04+/- 129 0.377+/-0.00253  484+/-26.1     0.449+/-0.0151  D+_bkg
B       2.46e+04+/- 172 0.372+/-0.00188  899+/-36.9     0.47+/-0.0114   TOTAL
.
.
Optimal cutstring:
min((log10(kaonplus_LOKI_MIPCHI2DV_PRIMARY)),(log10(kaonminus_LOKI_MIPCHI2DV_
PRIMARY)))>0.6&&log10(pion_LOKI_MIPCHI2DV_PRIMARY)>1.1&&D_s_LOKI_VFASPF_VCHI2
VDOF<4&&min((kaonplus_LOKI_PIDK_PIDpi),(kaonminus_LOKI_PIDK_PIDpi))>8&&max(pi
on_LOKI_TRCHI2DOF,max((kaonplus_LOKI_TRCHI2DOF),(kaonminus_LOKI_TRCHI2DOF)))<
3&&log10(D_s_LOKI_BPVVDCHI2)>1.703&&pion_LOKI_PIDpi_PIDK>-9:
Type    Excl. Cands.    Excl. Eff.        Incl. Cands.    Incl. Eff.        Name
S       1.19e+03+/-38.1 0.587+/-0.011     1.19e+03+/-38.1 0.587+/-0.011     D_s
S       548+/-27.3      0.533+/-0.0156    548+/-27.3      0.533+/-0.0156    D+
S       1.73e+03+/-46.9 0.568+/-0.00897   1.73e+03+/-46.9 0.568+/-0.00897   TOTAL
B       415+/-26.1      0.0141+/-0.000688 415+/-26.1      0.0141+/-0.000688 D_s_bkg
B       484+/-26.1      0.0132+/-0.000595 484+/-26.1      0.0132+/-0.000595 D+_bkg
B       899+/-36.9      0.0136+/-0.00045  899+/-36.9      0.0136+/-0.00045  TOTAL
Real time 0:14:08, CP time 840.710
```


C. Fitzpatrick

November 23, 2010

Results

- The first "autistic" and final inclusive plots are saved to the output ntuple for comparison
- Also saved to this file is the FoM evolution plot:

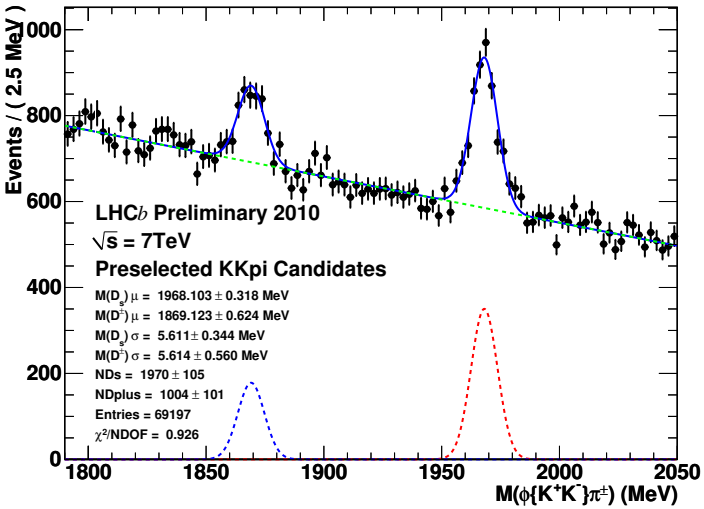


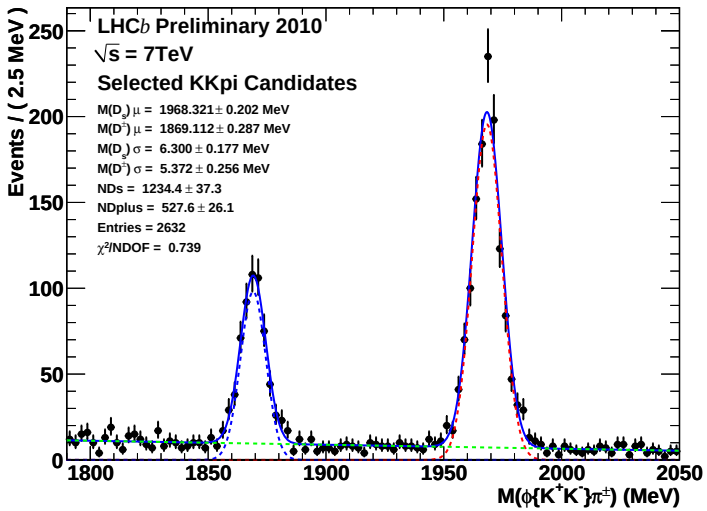
- The FoM is added to the plot at each reoptimisation: This should always have a positive gradient and becomes flat when converged

C. Fitzpatrick

November 23, 2010

Using the result





- The optimal cutstring printed at the end of running can be cut and pasted verbatim into root macros
- A number of additional tools are provided in the `Simpletools` package which can make use of the cut
- The cuts can also be translated into LOKI cuts for application to the trigger, stripping or selection

C. Fitzpatrick

November 23, 2010

Automating the whole procedure

- Throughout this talk I've shown a number of steps that should be taken to determine your selection
- Sifting large ntuples looking at plots is tiring work...
- Wouldn't it be great if all of this was somehow automatable?
- It is! The `Simpletools` package takes the effort out of selections.

C. Fitzpatrick

November 23, 2010

What's in `Simpletools`?

- `Simpletools` is a group of complimentary binary executables including CROP that share a common codebase
- each tool does something different, but many input arguments/files are the same
  - `stacker` makes publication-ready plots of layered/stacked datasets, multiple plots per page, log/linear scaling, per-event weighting included
  - `corr` takes `stacker` input variables and makes 2D scatterplots in addition to the linear correlation matrix
  - `eff` divides one dataset by another in each distribution specified, useful for determining cut acceptances
  - `sepper` calculates the separation power of a set of variables and sorts them according to this power
  - `stackergen` takes a crop dataset and generates a list of variables common to each ntuple and their ranges compatible with the previous tools
  - `varstocuts` turns a `stackergen` variable list into a `CROP` cut list as the final step
  - Many more! `tuplesampler`, `cutapplier`, etc...
- Starting with only a weightfile as shown earlier, the entire analysis can be automated in a single shell script that calls each of these tools sequentially
- You can go and have a coffee, come back and look at the results

C. Fitzpatrick

November 23, 2010

What's the catch?

- Computers have no brain. You need to provide your own
- While automating the procedure gets a result, you should look at how it got there
- You will probably optimise your selection several times, adjusting variable ranges, preprocessing cuts, etc along the way
- While these tools take the effort out of it, they don't take the thought out

C. Fitzpatrick

Getting and using Simpletools

- The latest version of Simpletools and tutorials can always be found in my AFS space:
- /afs/cern.ch/user/c/cofitzpa/public/simpletools/
- Or you can download it from here and the tutorial from here
- on LXplus you may need to SetupProject Gaudi ROOT to get a recent root version
- untar the latest version and run make
- Add the following lines to your .bashrc (substitute for the location of the simpletools directory)

```
SIMPLETOOLS=$HOME/simpletools_2.0a
export LD_LIBRARY_PATH=$SIMPLETOOLS/lib:$LD_LIBRARY_PATH
export PATH=$SIMPLETOOLS/bin:$PATH
```

- You should now have access to all the Simpletools
- The tutorial is a self contained selection procedure using real LHCb data from the charm cross-section analysis
- read and run tutorial.sh to see what Simpletools can do!

C. Fitzpatrick

November 23, 2010

Conclusions

- Selections are usually the starting point for your analysis
- They can be tricky to get right, and can be hard to make sense of
- There are many different ways to optimise a selection
  - "Horses for Courses": Not all methods are as powerful
  - but power isn't the only metric (workload, robustness)
- The `Simpletools` package aims to make choosing a selection easier

C. Fitzpatrick

November 23, 2010