Milano Bicocca
University

UPGRADE BANDWIDTH STUDIES

Summer Student Project

August 30, 2016

Supervisors: Mark Whitehead
Conor Fitzpatrick

Simone Meloni
s.meloni1@campus.unimib.it

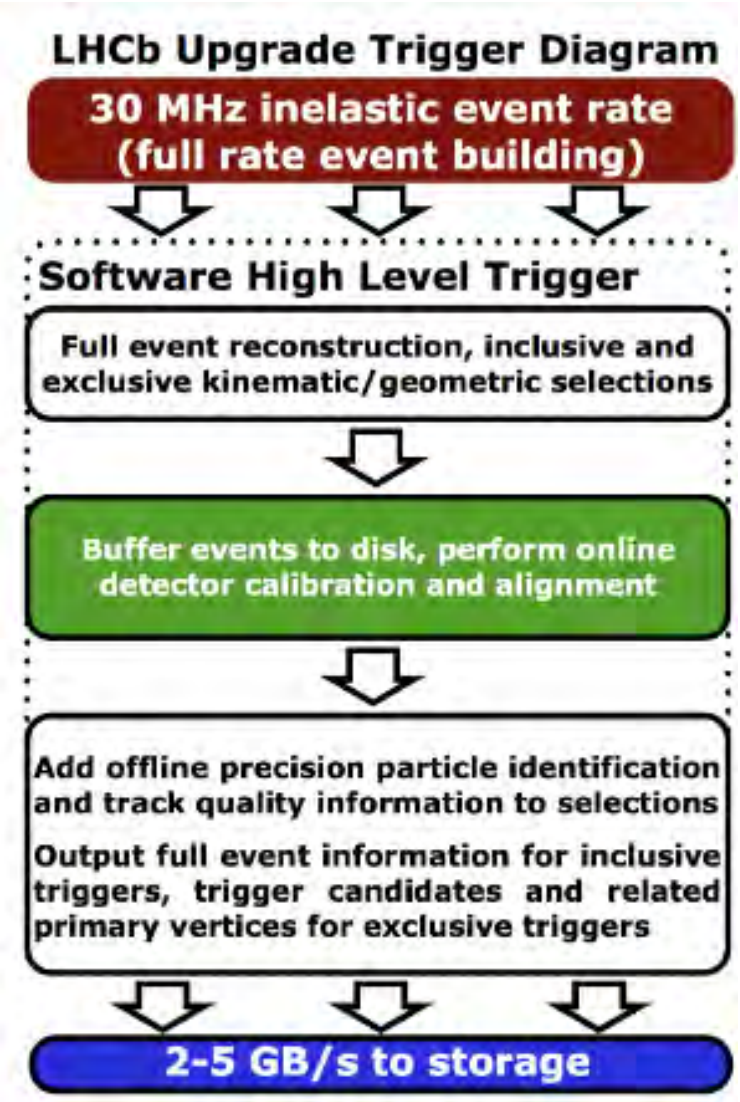
About Myself

- Simone Meloni
  - Born in Italy, Milan
  - 2015: Bachelor's degree in Physics at the Milano Bicocca University
  - Pursuing my studies in experimental particle physics at the Milano Bicocca University

About My Project

- Studies for the Upgrade of the trigger system (LHC Run 3)

Trigger Upgrade



- Run3 configuration :
  - $L = 2x10^{33} cm^{-2} s^{-1}$
  - Online Calibration
  - Full Software Trigger
- The Full Software trigger will process the full inelastic collision rate.
- **Challenge:** How do we distribute the limited bandwidth to disk across the LHCb physics programme?

Bandwidth division in the charm sector

- **My work**: Develop a framework to allow trigger selections to be adjusted in order to distribute the available bandwidth
  - Study of the physics potential as a function of the output bandwidth.
- Focus on the **charm sector**, and in particular:

$D^+ \to K^+K^-\pi^+$

$D^0 \to K^+K^-$

$D^0 \to K^+K^-\pi^+\pi^-$

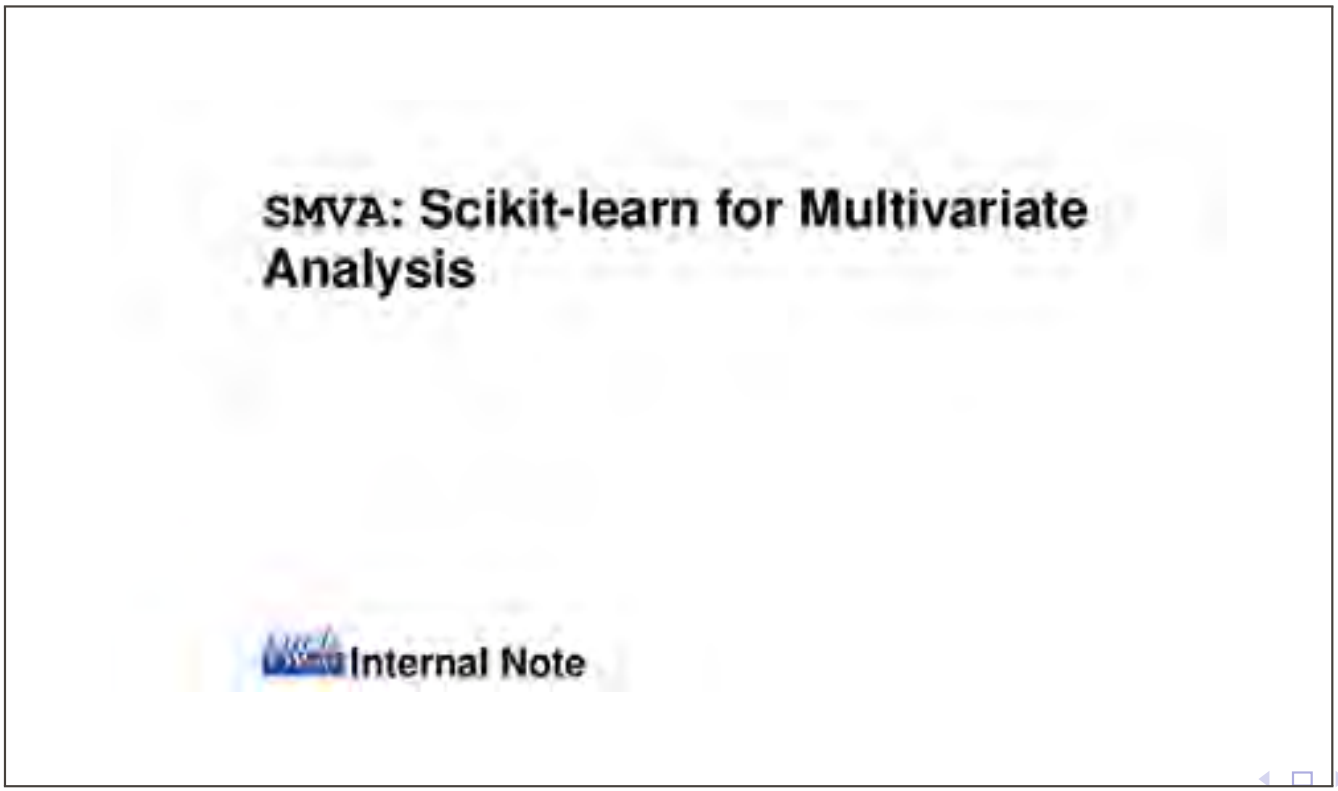
$D^0 \to K^0_S\pi^+\pi^-$

where the D^0 comes from $D^{+*} \to D^0\pi^+$

- Using latest Upgrade MC signal and minbias samples with the upgrade reconstruction sequence
  - A big thank to Mark Whitehead and Agnieszka Dziurda

Idea

- Single dial for each channel (or analysis) that tunes the bandwidth used!
- **For example**$\longrightarrow$ output of a continuous classifier
- First part of the project: Train an MVA for each channel
  - I have written a `Python` code that performs binary classification
  - Analysts can use this code to train their own MVA



Code and Documentation available at `https://gitlab.com/simeloni/SMVA`

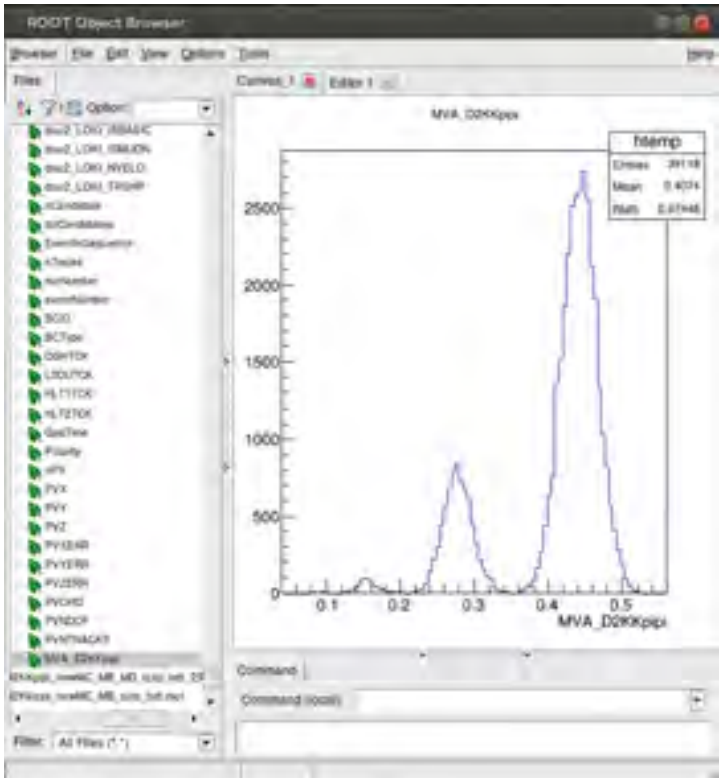
SMVA

- Performs binary classification (i.e. signal vs background separation)
- Produces *TMVA-Like* Plots in an automized way. Simple to use:

`inputfile.py`

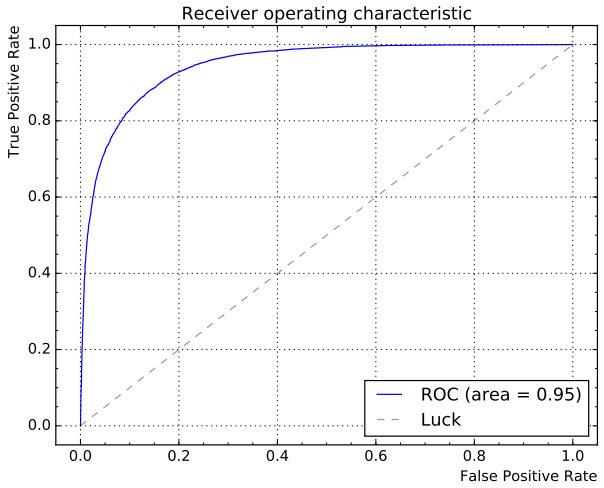
- file paths
- Preselections
- MVA and training var's

$\Longrightarrow$

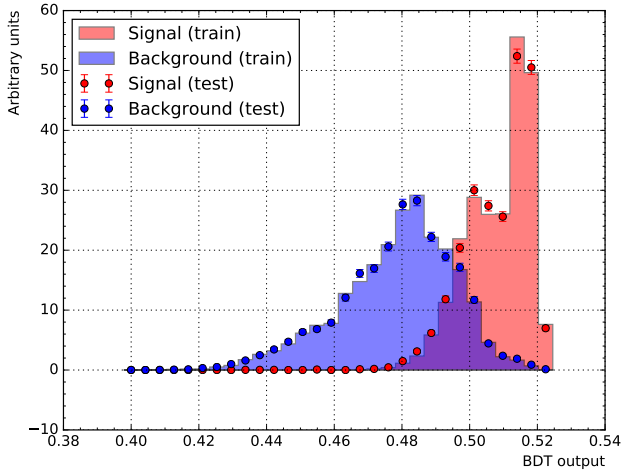


- Options available to enable specific Plots and features

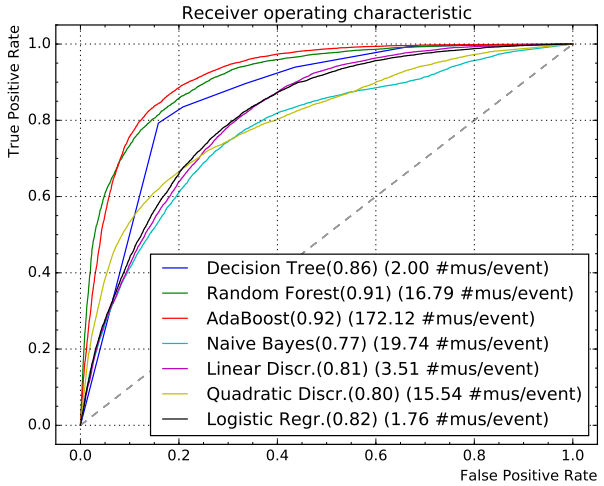
--MakeROC



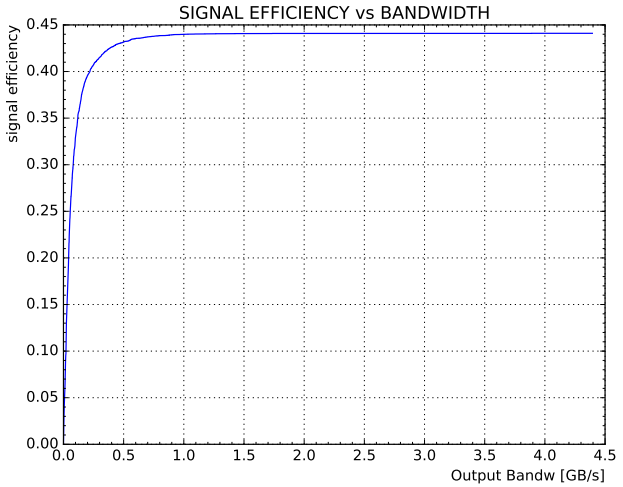
--CompareTrainTest



--ComparePerformances



--MakePlots



Bandwidth division algorithm

- **Idea:** use the single dial to find the right MVA selection for all the channels to fit into the Bandwidth limit maximizing the total signal efficiency
- **Algorithm:** minimize this function...

$$\chi^2 = \sum_{channels} \left(1 - \frac{\epsilon_s}{\epsilon_{max}}\right)^2 \qquad \epsilon_s = \frac{N(truth, presel., MVAsel.)}{N(truth)}$$

ϵ_{max} = maximum efficiency affordable for a single channel providing it with the maximum bandwidth.

...multiplying the efficiency by a penalty factor whenever $BW > BW_{limit}$

$$\text{penalty} = \frac{BW_{limit}}{BW}$$

Bandwidth division algorithm

- **first iteration:** For each channel independently minimize this function

$$\tilde{\chi}^2 = (1 - \epsilon_{max})^2$$

the cut found by the minimizer gives you ϵ_{max}

- **second iteration:** use the ϵ_{max} found to construct the χ^2 and minimize it.

$$\chi^2 = \sum_{channels} \left(1 - \frac{\epsilon}{\epsilon_{max}}\right)^2$$

- **OUTPUT:** A set of cuts with which we can evaluate the signal efficiencies, output Bandwidth and Rate for each of the channels

Bandwidth division output

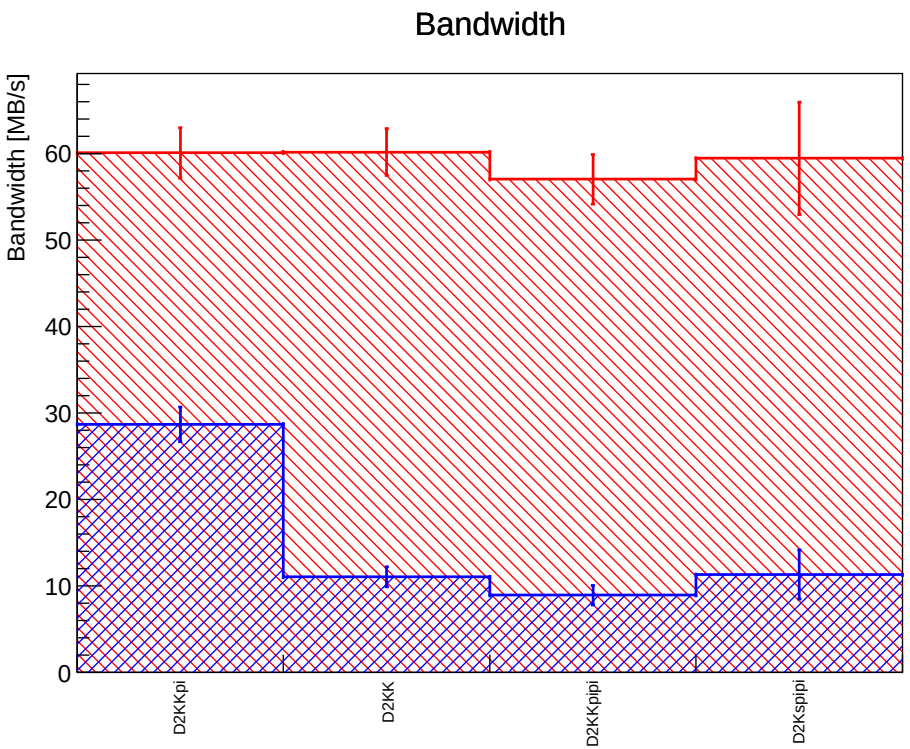
- Example of an output graph for the bandwidth division tool

At the first iteration, minimizing

$$\tilde{\chi}^2 = (1 - \epsilon_{max})^2$$

At the second iteration, minimizing

$$\chi^2 = \sum_{channels} \left(1 - \frac{\epsilon_s}{\epsilon_{max}}\right)^2$$



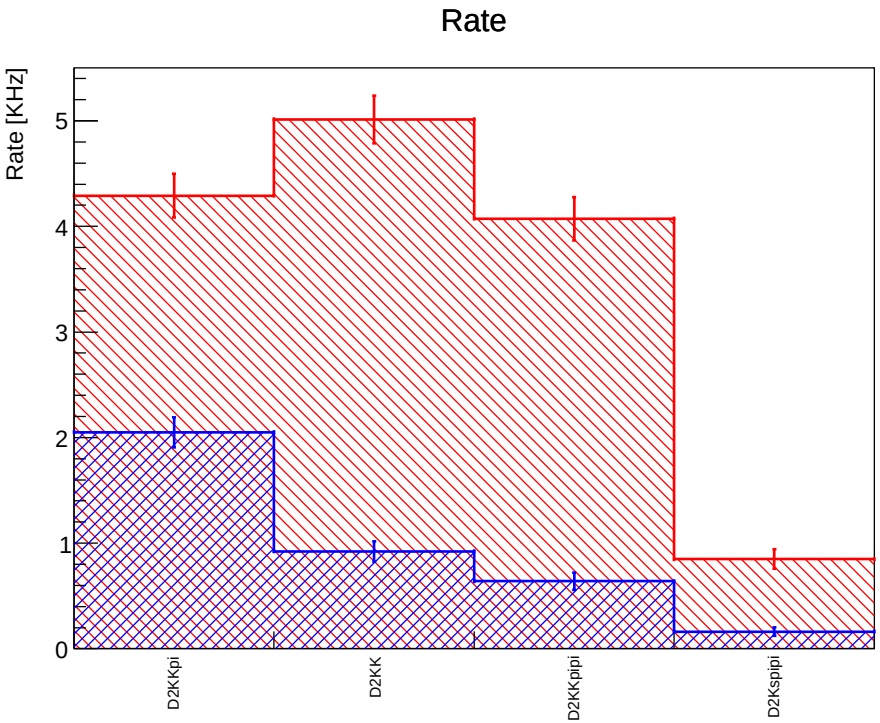
- Assuming a **Bandwidth limit of 60 MB/s** and an event size for each of the channels:

	event Size [kB/evt]
$D^+ \to K^+K^-\pi^+$	14
$D^0 \to K^+K^-$	12
$D^0 \to K^+K^-\pi^+\pi^-$	14
$D^0 \to K^0_S\pi^+\pi^-$	70

Event Size in the bandwidth division

- The way in which the events are saved affects the bandwidth division:

$D^+ \to K^+K^-\pi^+$	$D^+ \to K^+K^-\pi^+$	$D^0 \to K^+K^-$	$D^0 \to K^0_S\pi^+\pi^-$
14 kB/evt.	12 kB/evt.	14 kB/evt.	70 kB/evt.

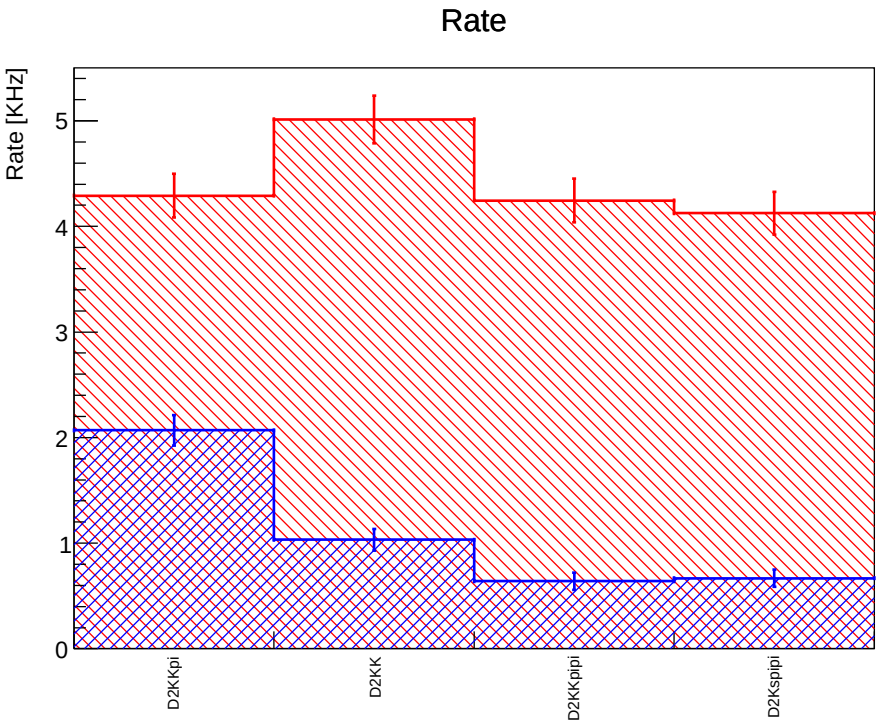


- The bigger the event size (for a given bandwidth), the harder the MVA cut
  - Channels with bigger event size get less rate because the MVA has to cut harder in order to fit them inside the Bandwith limit

Event Size in the bandwidth division

- The way in which the events are saved affects the bandwidth division:

$D^+ \to K^+K^-\pi^+$	$D^+ \to K^+K^-\pi^+$	$D^0 \to K^+K^-$	$D^0 \to K^0_S\pi^+\pi^-$
14 kB/evt.	12 kB/evt.	14 kB/evt.	14 kB/evt.

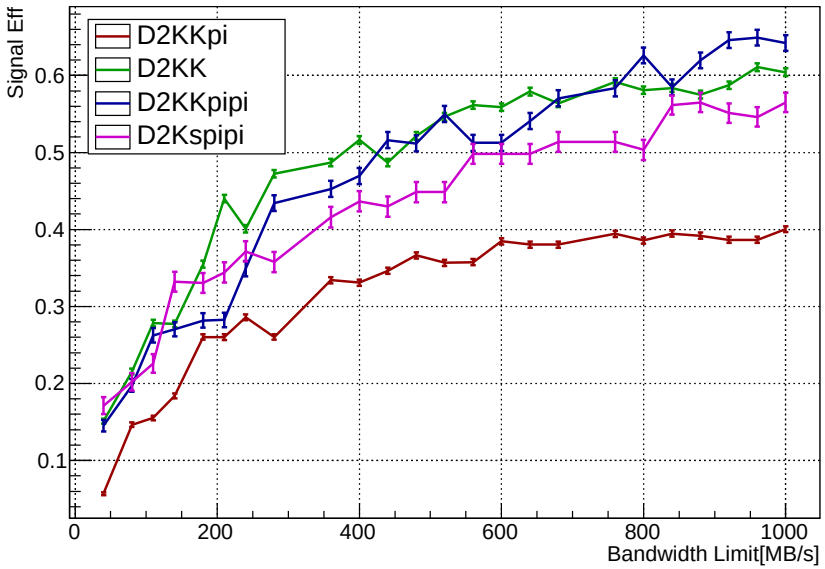


- The bigger the event size (for a given bandwidth), the harder the MVA cut
  - Channels with bigger event size get less rate because the MVA has to cut harder in order to fit them inside the Bandwith limit

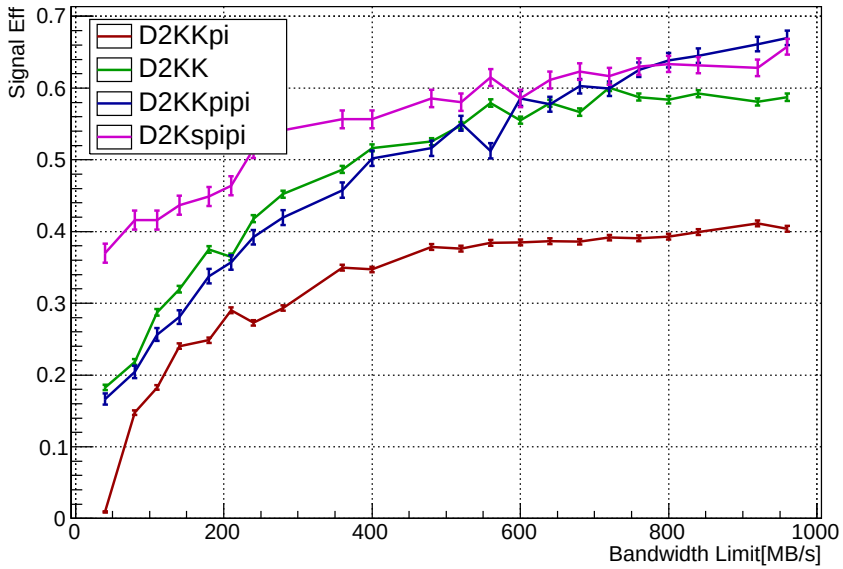
Bandwidth Scans

- How does the division vary with the available bandwidth?
- Bandwidth scans can be obtained running the same algorithm for different limits scenarios

$D^0 \to K^0_S \pi^+ \pi^- (70$ kB/evt.)



$D^0 \to K^0_S \pi^+ \pi^- (14$ kB/evt.)



- Big enhancement in the Signal Efficiency for the line that turned to Turbo Stream
- If we turn one of the lines from Turbo to Non-Turbo stream, the other lines aren't badly affected!

Conclusions

With the tools I have developed analysts can provide HLT2 selections that can be adjusted in a controlled manner to fit within the upgrade Bandwidth. The tools consist of two stages:

- **First stage:** MVA training using `Scikit Learn`
  - Developed an agile `Python` script to automate the production of *TMVA-like* plots
- **Second stage:** Used the MVA output to perform the bandwidth division in the charm sector with the upgraded conditions
  - Implemented a quantitative way to divide the bandwidth amongst the physics channels
  - Performed some Projections in different output bandwidth scenarios
- **Next goal:** use this tool to extraploate to the entire charm programme and outline the expected bandwidth available for each channel in the upgrade conditions.

THANK YOU FOR YOUR ATTENTION!

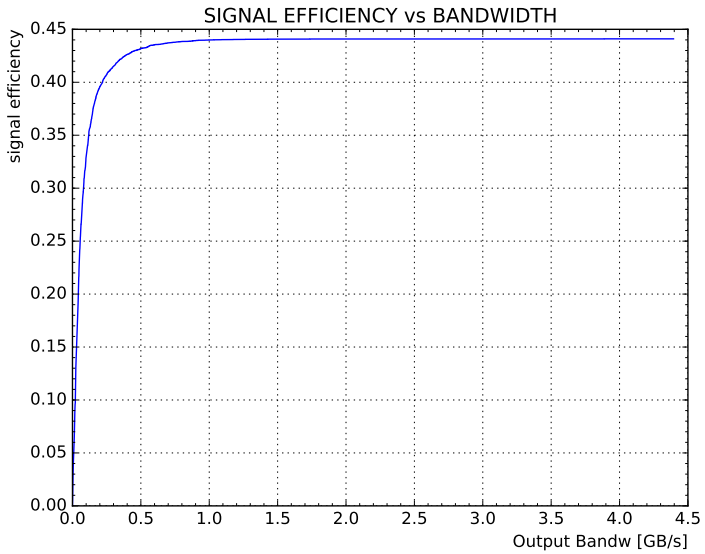
BACKUP

Bandwidth division

- The important plot for the bandwidth division tool is the signal efficiency versus the Output Bandwidth

$$\epsilon_s = \frac{N(truth, presel., MVAsel.)}{N(truth)}$$

$$Bandwidth = \epsilon_{minbias} \times R_{minbias} \times Evt.Size$$



- Preselection = (Loose) HLT2 selections (Thanks to Mark Williams)

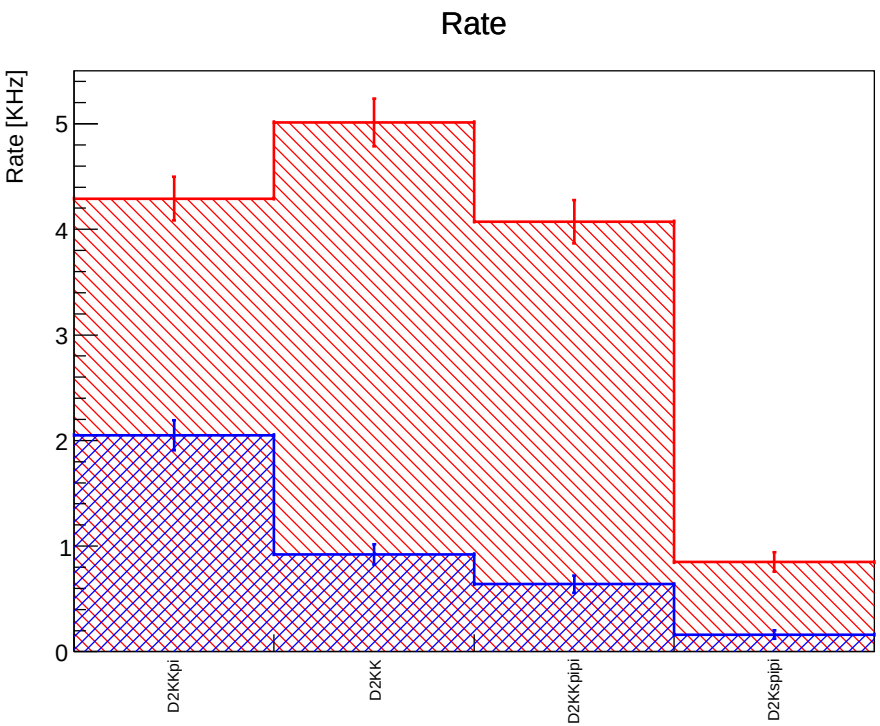
	D2KKpi	**D2KK**	**D2KKpipi**	**D2Kspipi**
Event Size	14 kB	12 kB	14 kB	70kB
Signal Eff.[1]	44.1(4)%	63.33(5)%	76.6(8)%	68.7(1)%
Background Eff.[2]	0.0656(4)%	0.279(3)%	1.690(6)%	3.621(9)%

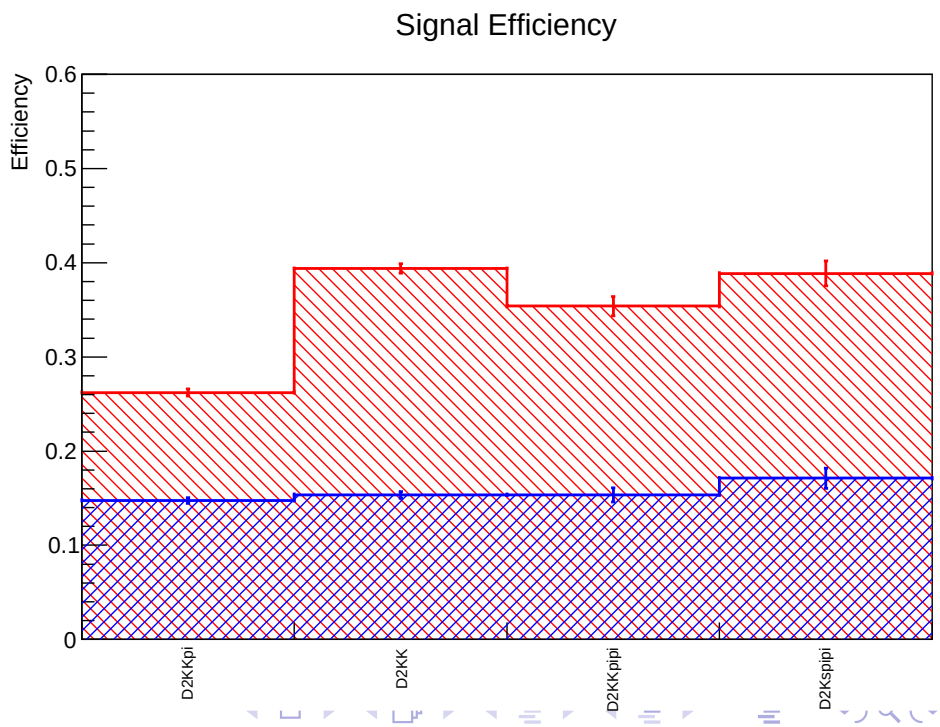
[1] wrt truth matched cand.
[2] wrt anti-truth matched cand.

Event Size in the bandwidth division

- The way in which the events are saved affects the bandwidth division:
  - The bigger the event size, the harder the MVA cut

$D^+ \to K^+K^-\pi^+$	$D^+ \to K^+K^-\pi^+$	$D^0 \to K^+K^-$	$D^0 \to K^0_S\pi^+\pi^-$
14 kB/evt.	12 kB/evt.	14 kB/evt.	70 kB/evt.

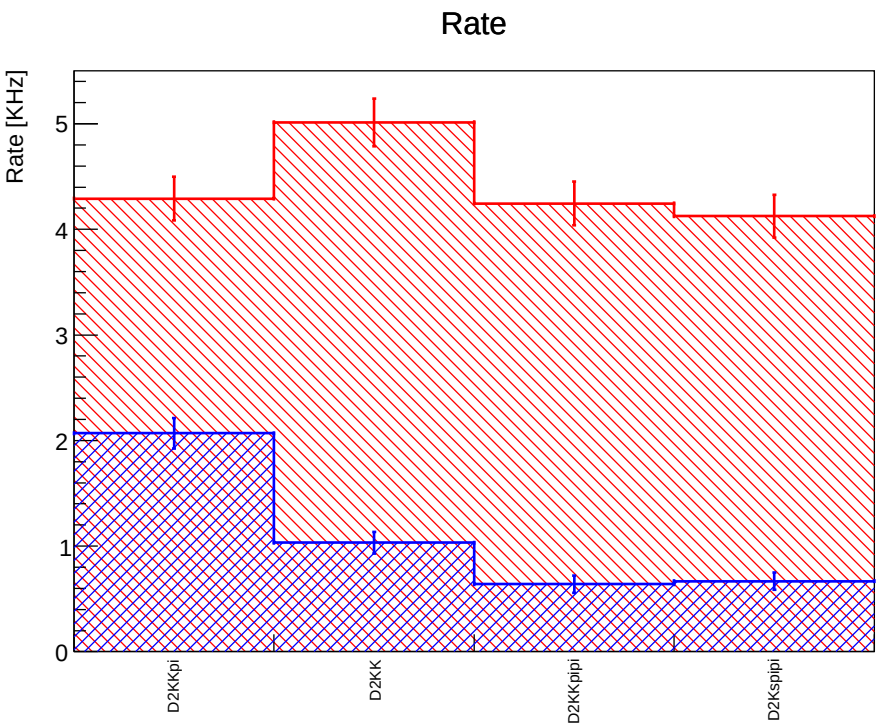


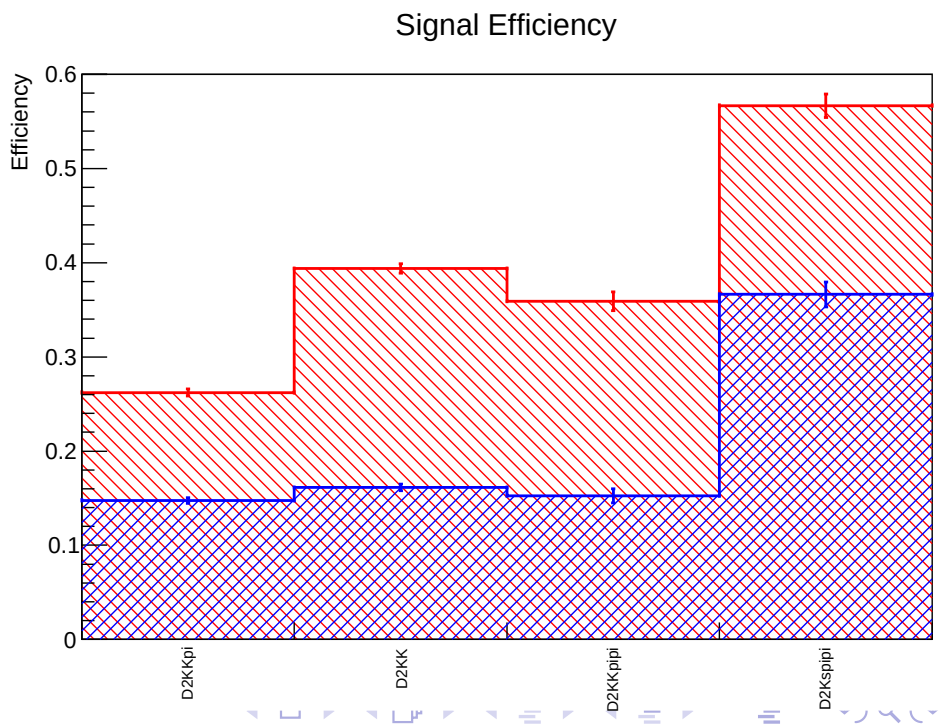


Event Size in the bandwidth division

- The way in which the events are saved affects the bandwidth division:
  - The bigger the event size, the harder the MVA cut

$D^+ \to K^+K^-\pi^+$	$D^+ \to K^+K^-\pi^+$	$D^0 \to K^+K^-$	$D^0 \to K^0_S\pi^+\pi^-$
14 kB/evt.	12 kB/evt.	14 kB/evt.	14 kB/evt.





Preselections

- Kinematical Cuts, Track quality, Vertex quality, Ghost Probability, and a loose Mass Window on the candidates

	D2KKpi	**D2KK**	**D2KKpipi**	**D2Kspipi**
Processed events	92704	103163	93581	102468
Reco. Candidates	2953517	231261	406361	345262
Truth matched cand.	13255	11130	2299	1174
Truth matched and preselected cand.	5887	7049	1762	806
Signal Efficiency [3]	44.1(4)%	63.33(5)%	76.6(8)%	68.7(1)%
Background Efficiency [4]	0.0656(4)%	0.279(3)%	1.690(6)%	3.621(9)%

- Big difference on the number of reco. candidates with respect to the number of events in the D2KKpi channel: there is no Δ_m cut (No D^* decay)

[3] wrt truth matched cand.

[4] wrt anti-truth matched cand.

PreSelection on the Minbias events

	D2KKpi	D2KK	D2KKpipi	D2Kspipi
Processed events	2984550	2963306	2954144	2963773
Reco. Candidates	45044857	2927035	4709605	4085688
Anti-Truth matched cand.	45039035	2926918	4709569	4085532
Anti-Truth matched and preselected cand.	29552	8190	79581	147950

Preselections

- **D2KKpi**

```
[    '(cand_M > 1700)',
   '(cand_M < 2100)',
   '((dau1_PT + dau2_PT + dau3_PT) > 3000)',
   '(TMath::ACos(cand_LOKI_BPVDIRA) < 0.010)',
   '(cand_LOKI_BPVLTIME > 0.0004)',
   '(dau1_PT > 250)',
   '(dau3_PT > 250)',
   '(dau1_TRACK_CHI2NDOF < 6.0)',
   '(dau2_TRACK_CHI2NDOF < 6.0)',
   '(dau3_TRACK_CHI2NDOF < 6.0)',
   '(dau1_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
   '(dau2_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
   '(dau1_TRACK_GhostProb<0.4)',
   '(dau2_TRACK_GhostProb<0.4)',
   '(dau3_TRACK_GhostProb<0.4)',
   '(cand_LOKI_VFASPF_VCHI2VDOF < 6.)']
```

Preselections

- **D2KK**

```
['(TMath::ACos(dau1_LOKI_BPVDIRA) < 0.0173)',
'(dau1_LOKI_BPVVDCHI2 > 25.0)',
'(dau1_PT > 1000.)',
'(dau1_LOKI_VFASPF_VCHI2VDOF < 10.)',
'(Ddau1_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
'(Ddau2_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
  '(Ddau1_PT > 500.)',
'(Ddau2_PT > 500.)',
'(Ddau1_P  > 5000.)',
'(Ddau2_P  > 5000.)',
'(Ddau1_TRACK_CHI2NDOF < 3.0)',
'(Ddau2_TRACK_CHI2NDOF < 3.0)',
        '(Ddau1_TRACK_GhostProb<0.4)',
'(Ddau2_TRACK_GhostProb<0.4)',
'(dau1_LOKI_VFASPF_VCHI2VDOF < 10.0)']
```

Preselections

- **D2KKpi**

```
[    '(cand_M > 1700)',
   '(cand_M < 2100)',
   '((dau1_PT + dau2_PT + dau3_PT) > 3000)',
   '(TMath::ACos(cand_LOKI_BPVDIRA) < 0.010)',
   '(cand_LOKI_BPVLTIME > 0.0004)',
   '(dau1_PT > 250)',
   '(dau3_PT > 250)',
   '(dau1_TRACK_CHI2NDOF < 6.0)',
   '(dau2_TRACK_CHI2NDOF < 6.0)',
   '(dau3_TRACK_CHI2NDOF < 6.0)',
   '(dau1_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
   '(dau2_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
   '(dau1_TRACK_GhostProb<0.4)',
   '(dau2_TRACK_GhostProb<0.4)',
   '(dau3_TRACK_GhostProb<0.4)',
   '(cand_LOKI_VFASPF_VCHI2VDOF < 6.)']
```

Preselections

- **D2KKpipi**

```
['(Ddau1_PT > 250.)',
   '(Ddau2_PT > 250.)',
   '(Ddau3_PT > 250.)',
   '(Ddau4_PT > 250.)',
            '((Ddau1_PT + Ddau2_PT + Ddau3_PT + Ddau4_PT ) > 1800.)',
   '(Ddau1_LOKI_MIPCHI2DV_PRIMARY > 3)',
   '(Ddau2_LOKI_MIPCHI2DV_PRIMARY > 3)',
   '(Ddau3_LOKI_MIPCHI2DV_PRIMARY > 3)',
                                    '(Ddau4_LOKI_MIPCHI2DV_PRIMARY > 3)',
   '(Ddau1_TRACK_GhostProb <0.4)',
   '(Ddau2_TRACK_GhostProb <0.4)',
   '(Ddau3_TRACK_GhostProb <0.4)',
   '(Ddau4_TRACK_GhostProb <0.4)',
   '(dau1_M > 1700.)',
   '(dau1_M < 2100.)',
   '(dau1_LOKI_VFASPF_VCHI2VDOF < 12.0)',
  '(TMath::ACos(dau1_LOKI_BPVDIRA < 0.20))',
   '(dau1_LOKI_BPVLTIME > 0.0001)',
   '(dau1_PT > 2000.)',
   '(dau1_LOKI_BPVVDCHI2 > 25.)']
```

Preselections

- **D2KKpi**

```
[    '(cand_M > 1700)',
   '(cand_M < 2100)',
   '((dau1_PT + dau2_PT + dau3_PT) > 3000)',
   '(TMath::ACos(cand_LOKI_BPVDIRA) < 0.010)',
   '(cand_LOKI_BPVLTIME > 0.0004)',
   '(dau1_PT > 250)',
   '(dau3_PT > 250)',
   '(dau1_TRACK_CHI2NDOF < 6.0)',
   '(dau2_TRACK_CHI2NDOF < 6.0)',
   '(dau3_TRACK_CHI2NDOF < 6.0)',
   '(dau1_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
   '(dau2_LOKI_MIPCHI2DV_PRIMARY > 4.0)',
   '(dau1_TRACK_GhostProb<0.4)',
   '(dau2_TRACK_GhostProb<0.4)',
   '(dau3_TRACK_GhostProb<0.4)',
   '(cand_LOKI_VFASPF_VCHI2VDOF < 6.)']
```

Preselections

- **D2Kspipi**

```
['(dau1_M > 1700)',
   '(dau1_M < 2100)',
   '((Ddau1_PT + Ddau2_PT + Ddau3_PT) > 1500)',
   '(dau1_PT > 1800. )',
   '(TMath::ACos(dau1_LOKI_BPVDIRA) < 0.0346)',
   '(dau1_LOKI_BPVVDCHI2 > 20.0)',
   '(dau1_LOKI_VFASPF_VCHI2VDOF < 20.)',
   '(dau1_LOKI_BPVLTIME > 0.0001 )', #ns
   '(Ddau1_PT > 250.)',
   '(Ddau2_PT > 250.)',
   '(Ddau3_PT > 250.)',
   '(Ddau1_LOKI_MIPCHI2DV_PRIMARY > 3.)',
          '(Ddau2_LOKI_MIPCHI2DV_PRIMARY > 3.)',
   '(Ddau3_LOKI_MIPCHI2DV_PRIMARY > 3.)',
   '(Ddau2_TRACK_GhostProb < 0.4)',
   '(Ddau3_TRACK_GhostProb < 0.4)']
```

Projection Plots: Bandwidth

Figure : D2KSpipi NON Turbo

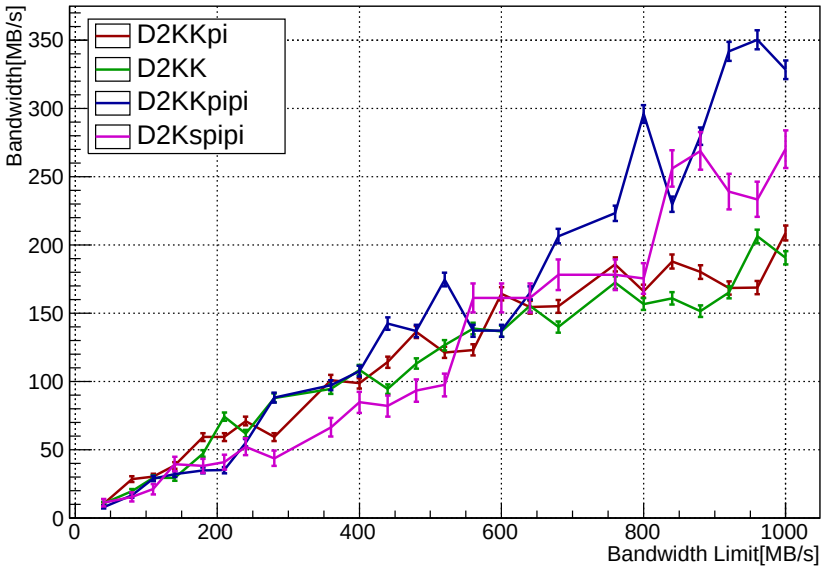
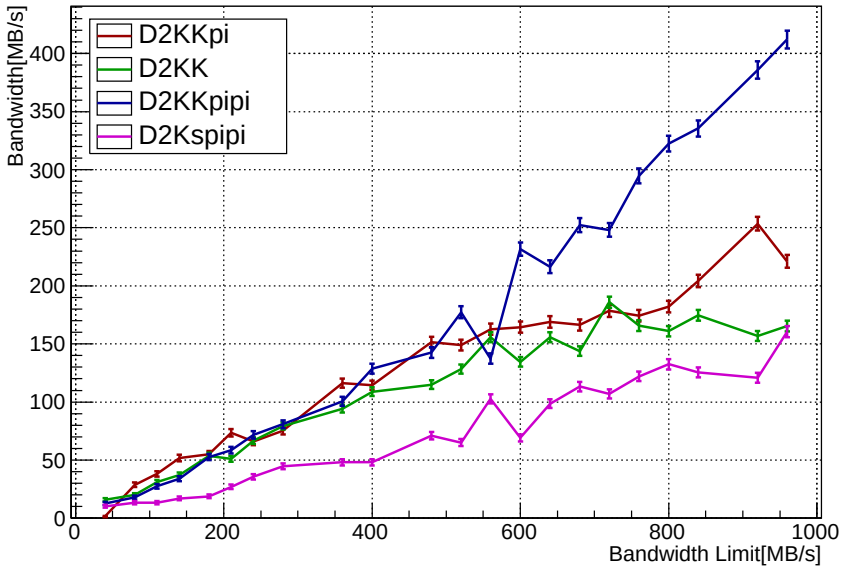


Figure : D2KSpipi Turbo



Projection Plots: max Bandwidth

Figure : D2KSpipi NON Turbo

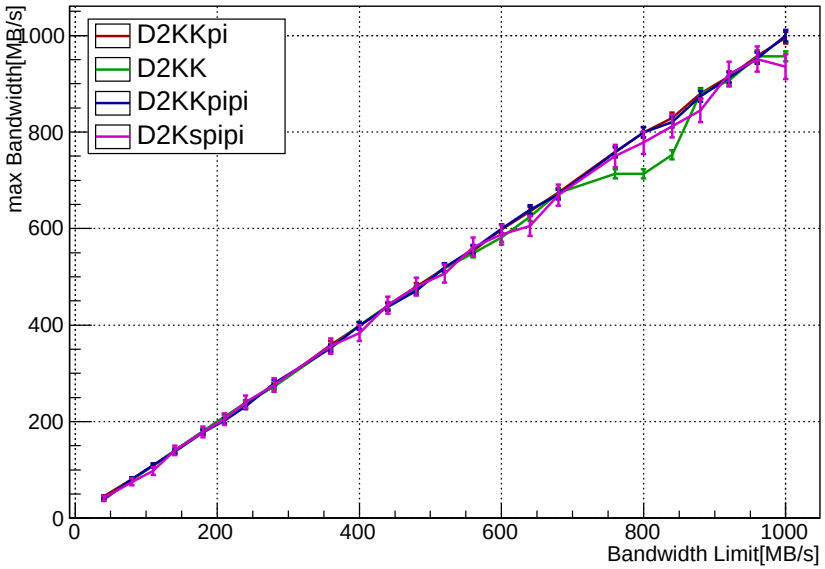
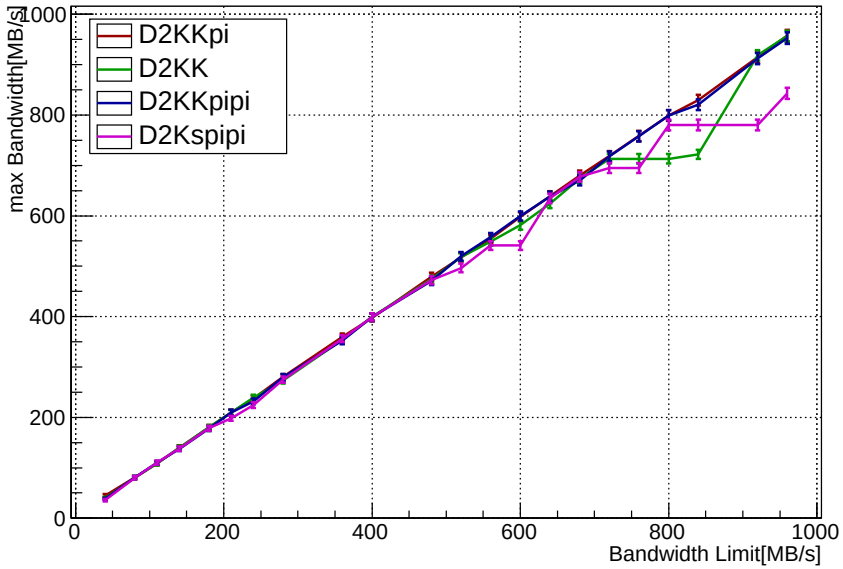


Figure : D2KSpipi Turbo



Projection Plots: Retention

Figure : D2KSpipi NON Turbo

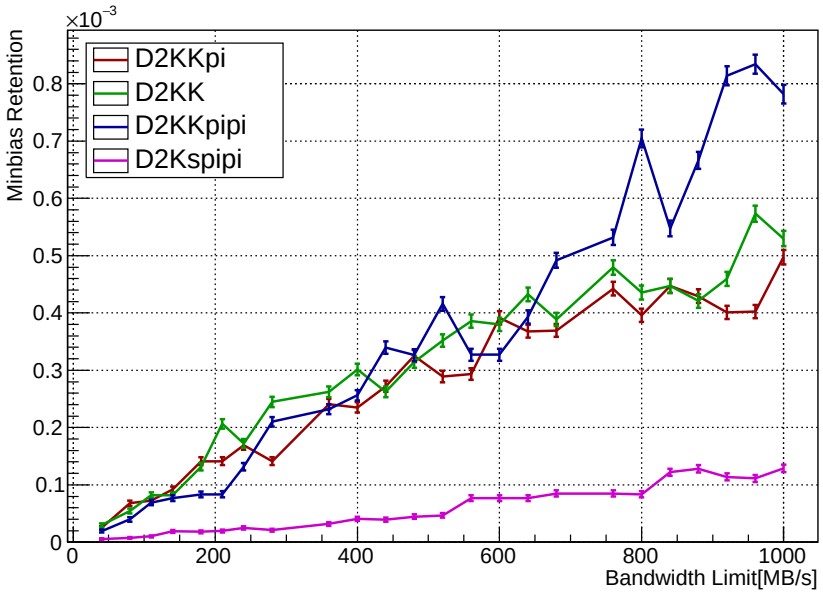
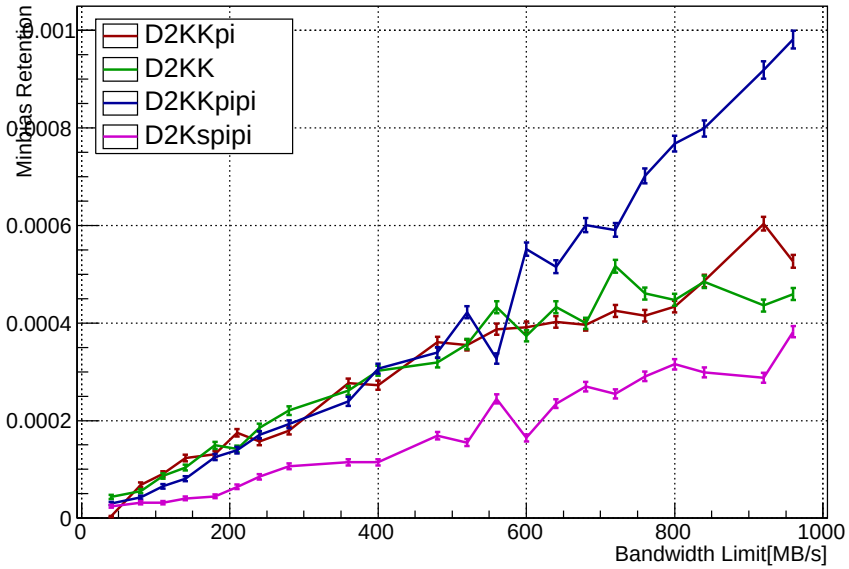


Figure : D2KSpipi Turbo



Projection Plots: max Retention

Figure : D2KSpipi NON Turbo

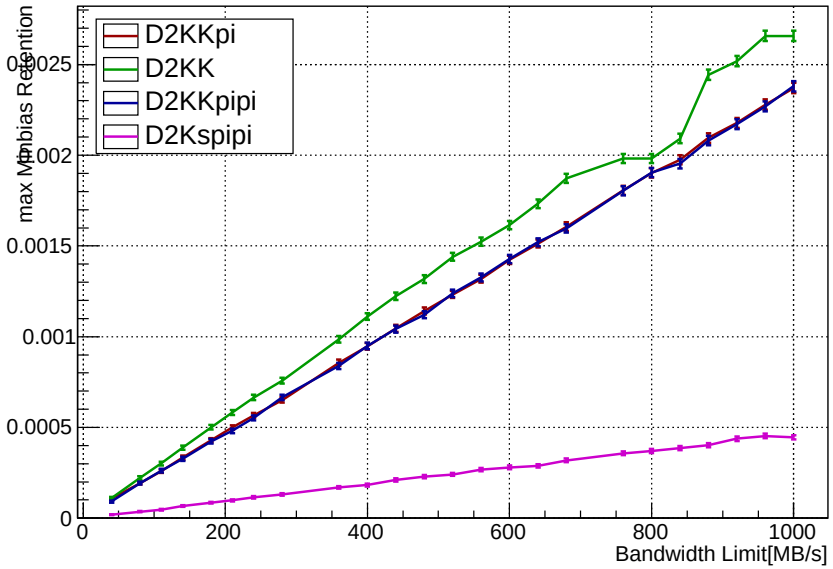
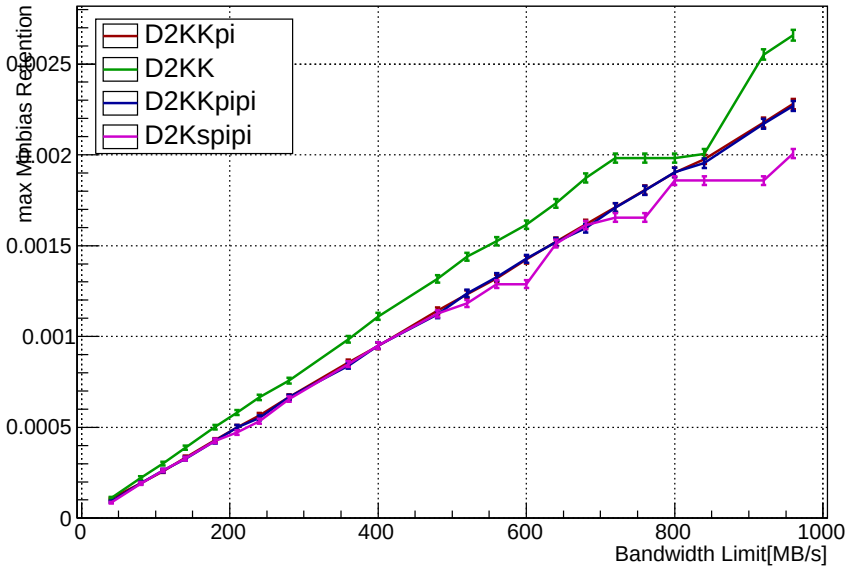


Figure : D2KSpipi Turbo



Projection Plots: max Rate

Figure : D2KSpipi NON Turbo

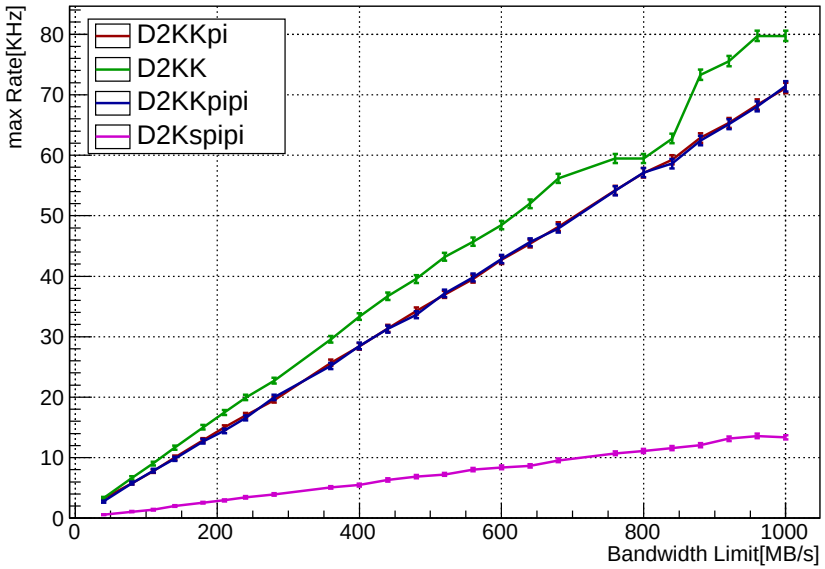
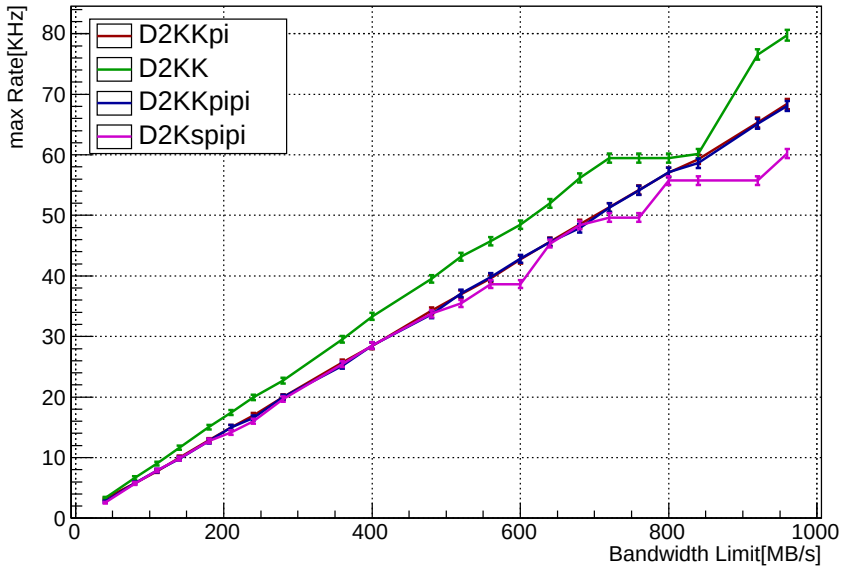


Figure : D2KSpipi Turbo



Projection Plots: max Signal Efficiency

Figure : D2KSpipi NON Turbo

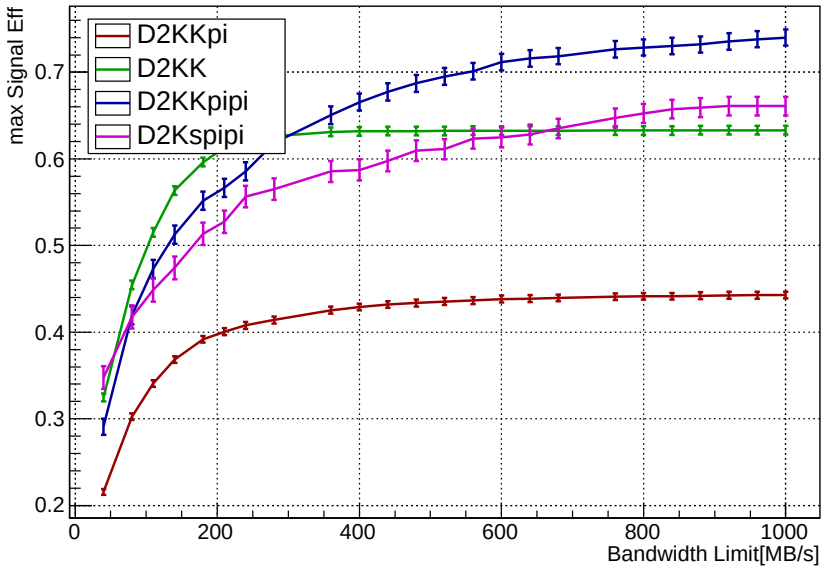
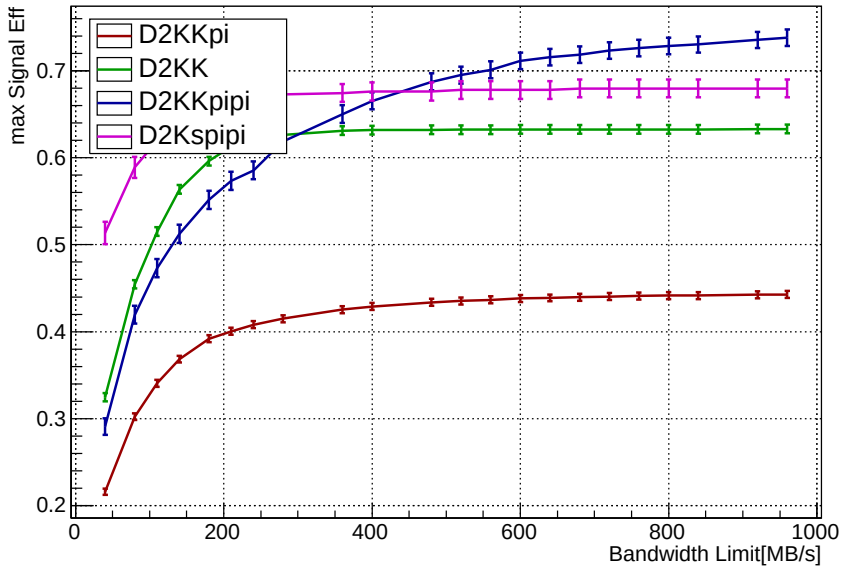


Figure : D2KSpipi Turbo



Projection Plots: MVA Signal Efficiency

Figure : D2KSpipi NON Turbo

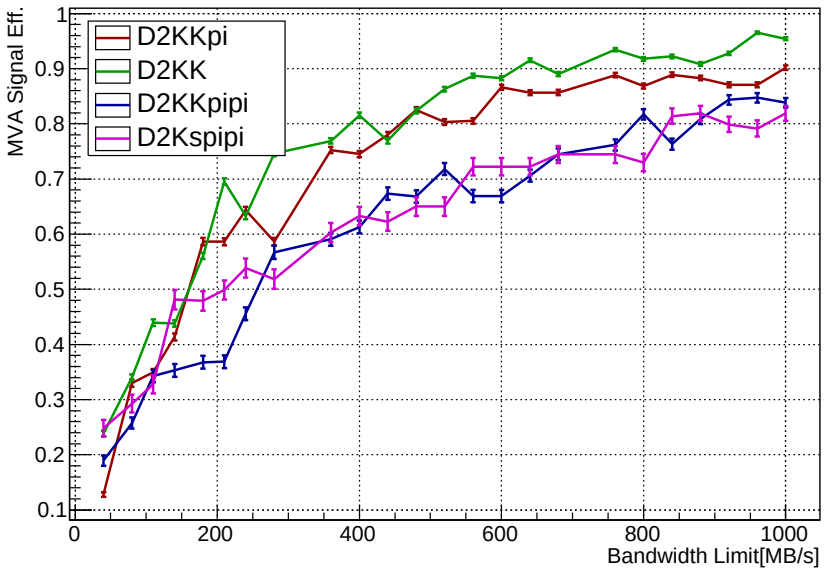
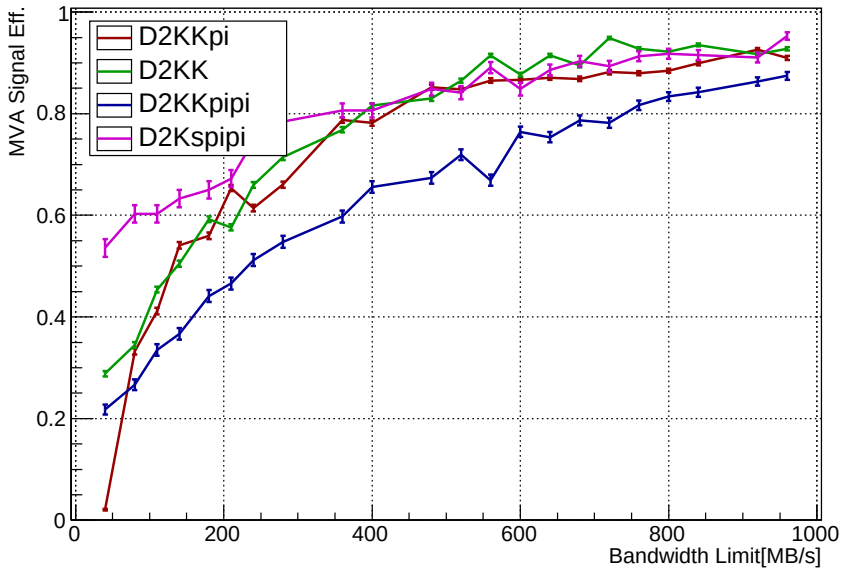


Figure : D2KSpipi Turbo



Projection Plots: max MVA Signal Efficiency

Figure : D2KSpipi NON Turbo

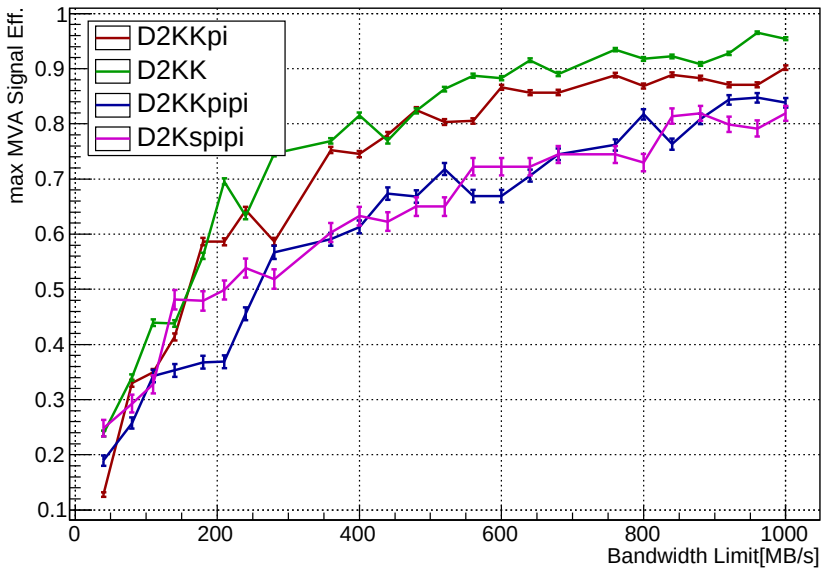


Figure : D2KSpipi Turbo