CMS Pixel Upgrade

Optimizing DAC Parameters

Habib/Petrukhin/Pitzl

First Test : [Will change as wel learn more about the chip]

Set of parameters to optimize = {Vsf, Vana, VhldDel}. [steerable]

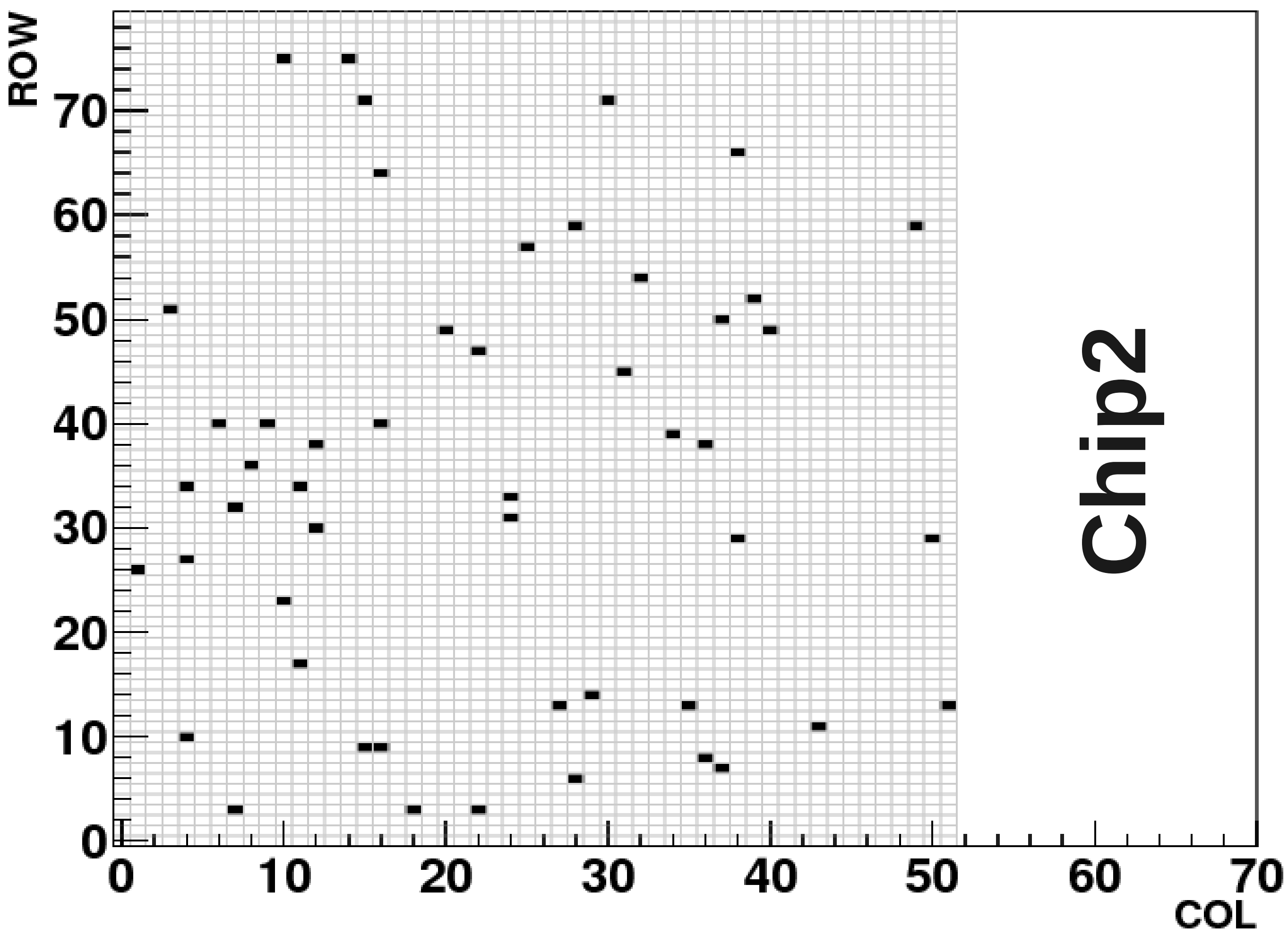
```
//Stuff the user will do

Calor *cal = new Calor();                   //Instance of Calibrator
cal->SetChipDir((Char_t*)"chip2");          //Set chip to calibrate
cal->MakePixelList(50,new TRandom3());      //Generate Pixel test list
cal->MakeROCRidge(3);                       //1 Ridge for Vsf
cal->MakeROCRidge(10);                      //2 Ridge for VhldDel
cal->MakeROCLinearity(3);                   //3 Linearity Scan for Vsf
cal->MakeROCLinearity(10);                  //4 Linearity Scan for VhldDel
cal->MakeROCLinearity(2);                   //5 Linearity Scan for Vana
cal->MakeROCLinearity(3);                   //6 Linearity Scan for Vsf
cal->MakeROCLinearity(10);                  //7 Linearity Scan for VhldDel
cal->MakeROCLinearity(2);                   //8 Linearity Scan for Vana
```

First Test : [Will change as we learn more about the chip]

Generate a set of pixels to test which will characterize the entire ROC.

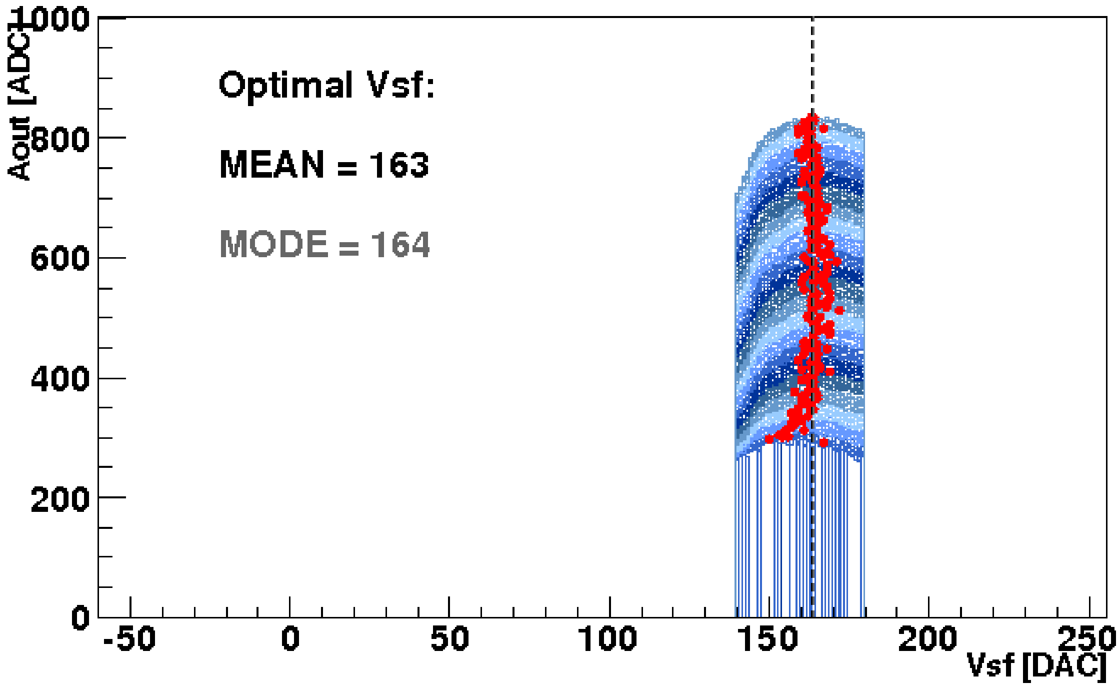
For this first test I used 50 random pixels.

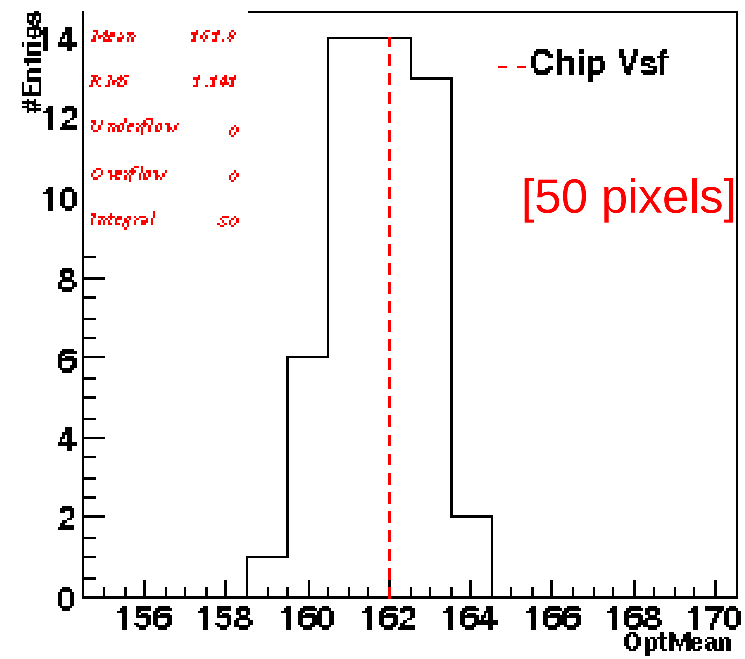


First Test : [Will change as wel learn more about the chip]

1. Get Ridge for Vsf

[1 out of the 50 pixels]





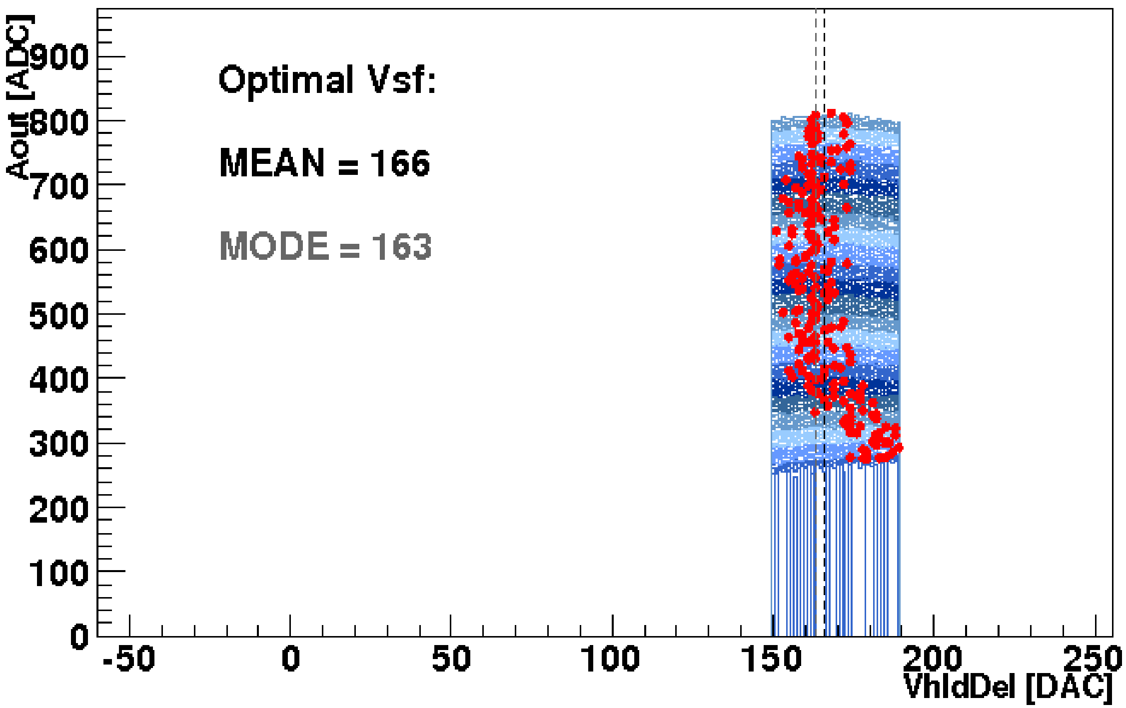

```
RidgeValueOfROC                    162
FractionOfPixelsWithin2            0.98
FractionOfPixelsWithin2_Error      0.019799
```

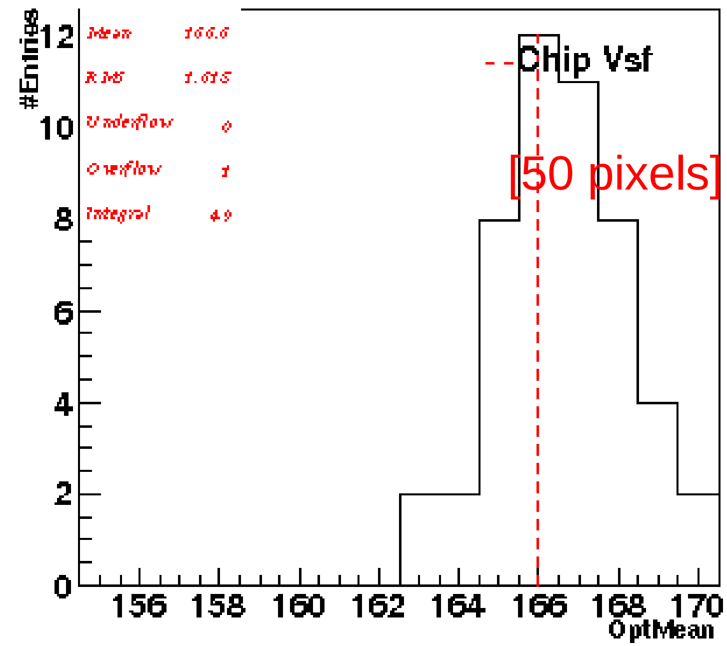
```
Vsf  170 -> 162
```

First Test : [Will change as wel learn more about the chip]

2. Get Ridge for VhldDel

[1 out of the 50 pixels]



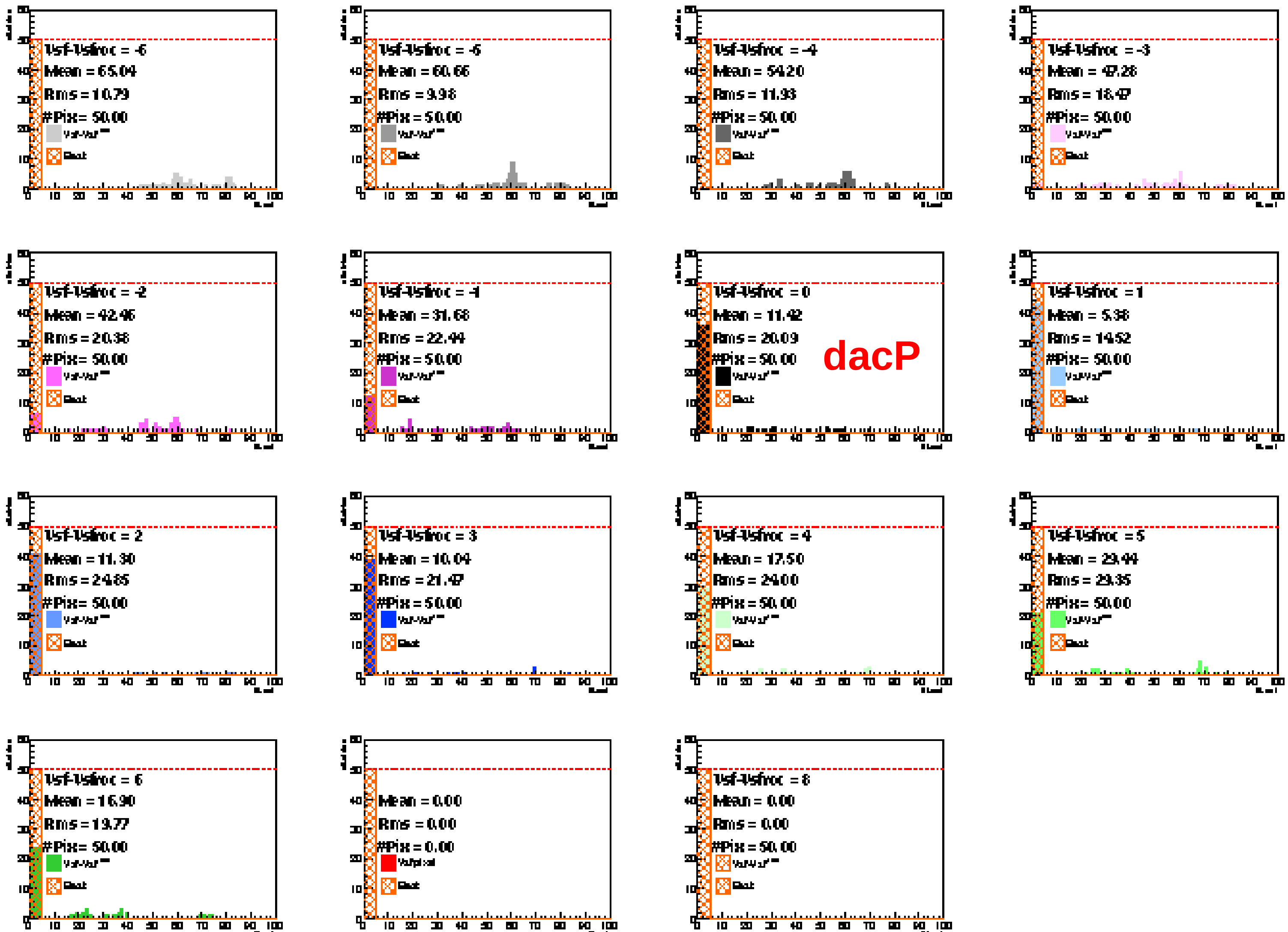



```
RidgeValueOfROC                    166
FractionOfPixelsWithin2            0.82
FractionOfPixelsWithin2_Error 0.0543323

VhldDel 150 -> 166
```

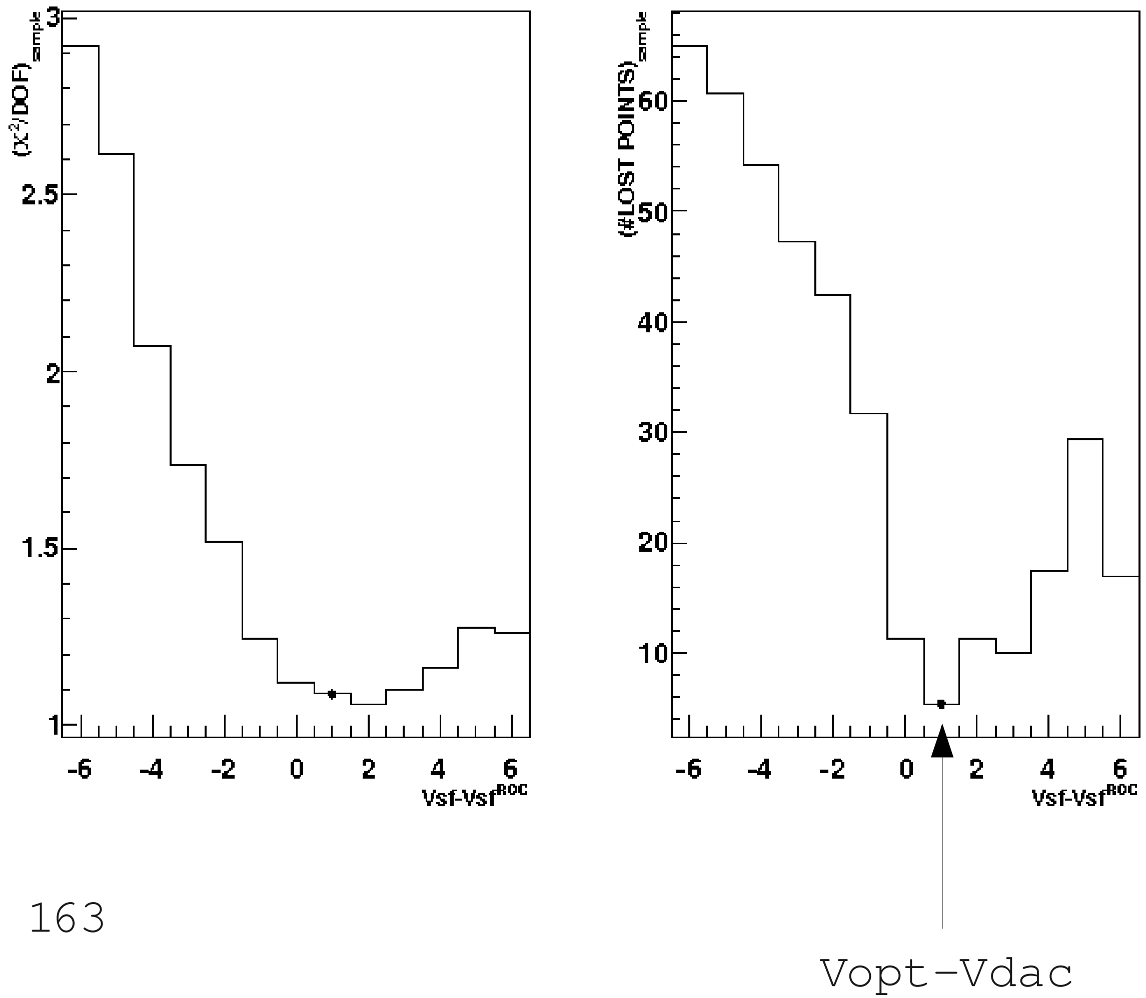
First Test : [Will change as wel learn more about the chip]

3. Minimize #Lost Points for Vsf [*Scan within +/- 6 of current dac Parameter Value*]



First Test : [Will change as wel learn more about the chip]

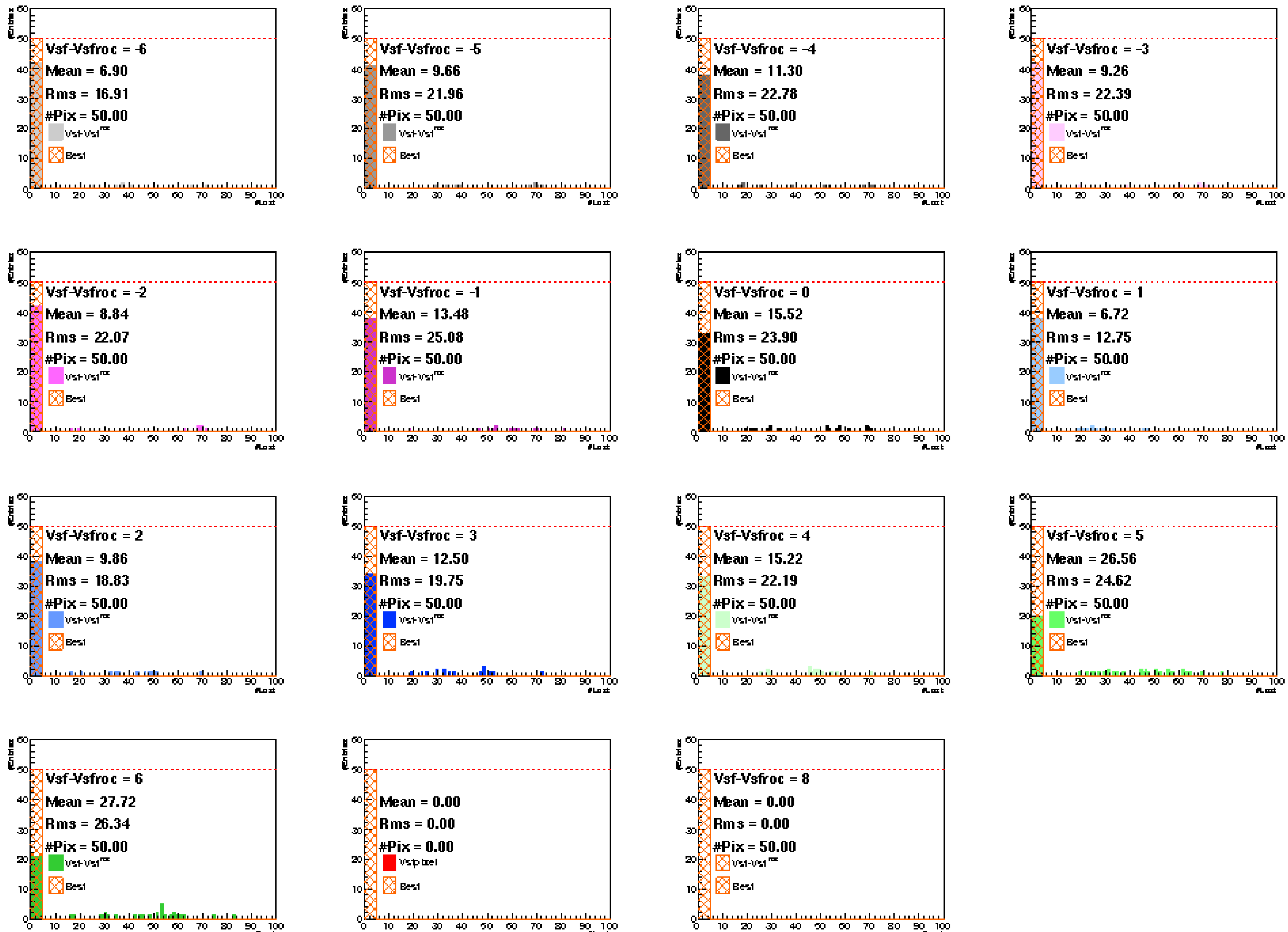
3. Minimize #Lost Points for Vsf [*Scan within +/- 6 of current dac Parameter Value*]



Vsf 162 -> 163

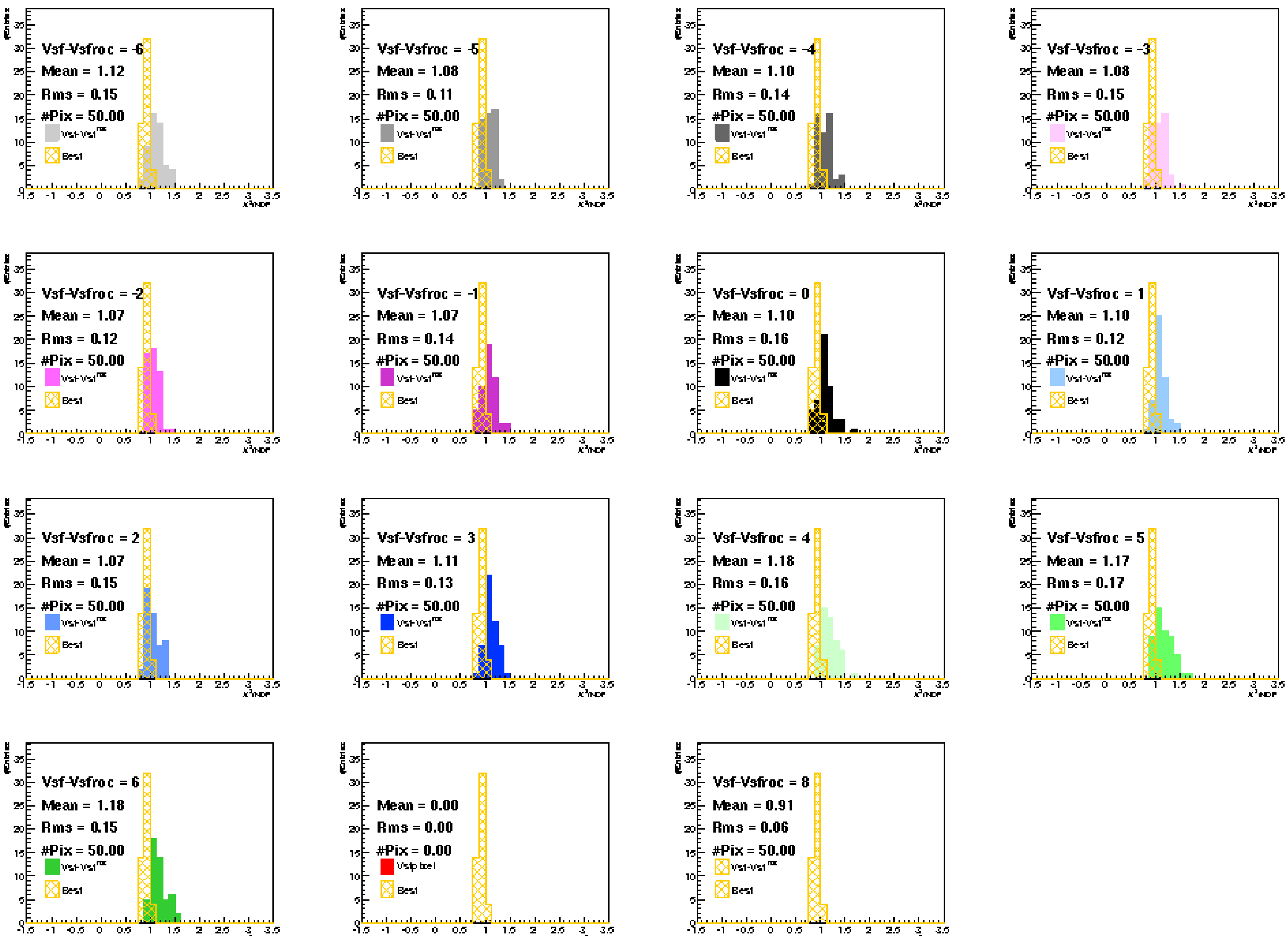
First Test : [Will change as wel learn more about the chip]

4. Minimize #Lost Points for VhldDel [*Scan within +/- 6 of current dac Parameter Value*]



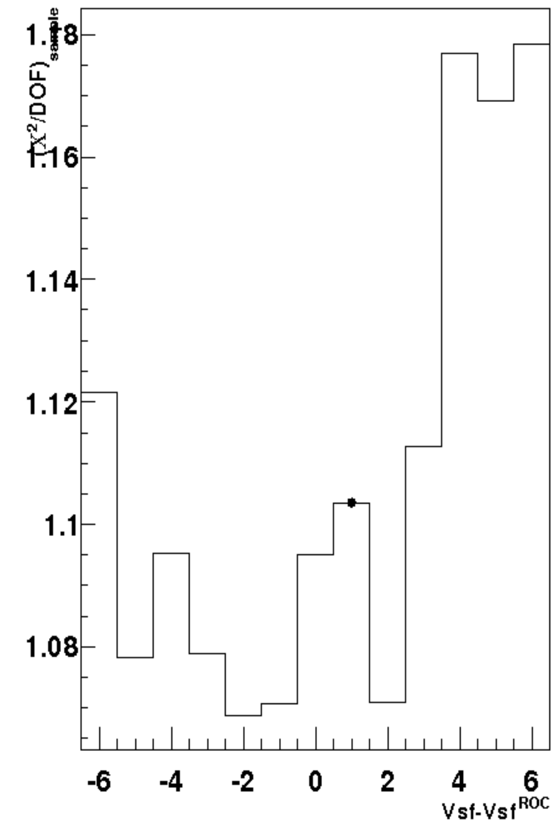
First Test : [Will change as wel learn more about the chip]

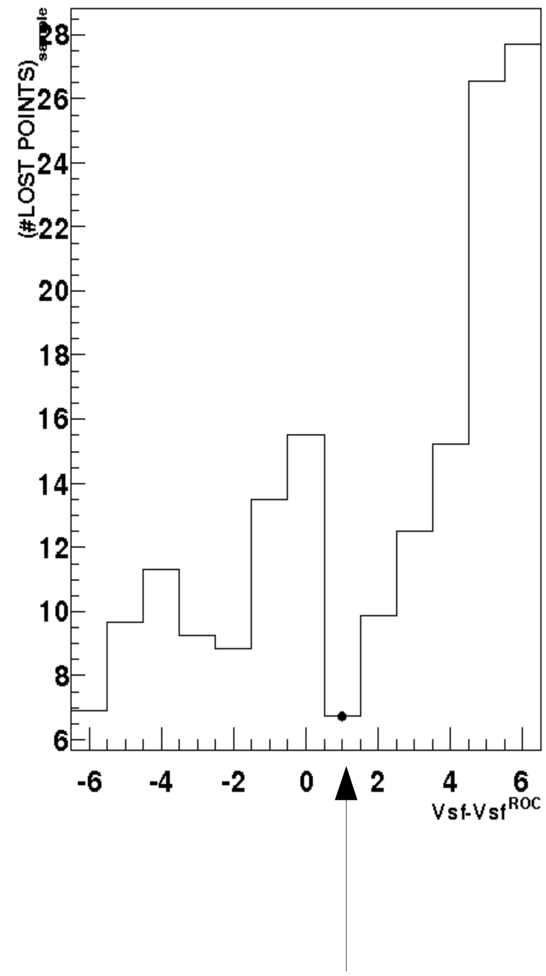
4. Minimize #Lost Points for VhldDel [*Scan within +/- 6 of current dac Parameter Value*]



First Test : [Will change as wel learn more about the chip]

4. Minimize #Lost Points for VhldDel [*Scan within +/- 6 of current dac Parameter Value*]



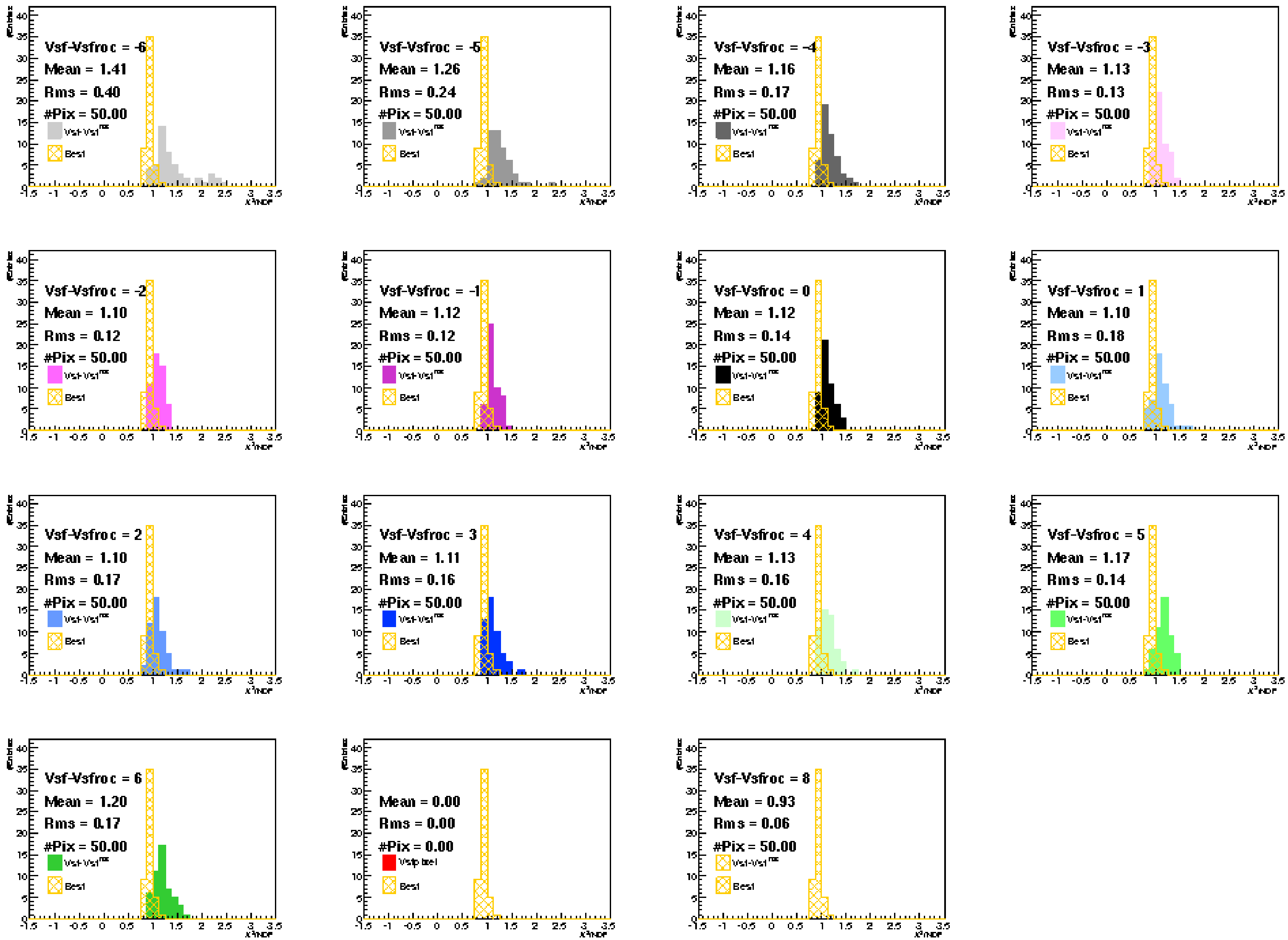


VhldDel 166 -> 167

Vopt-Vdac

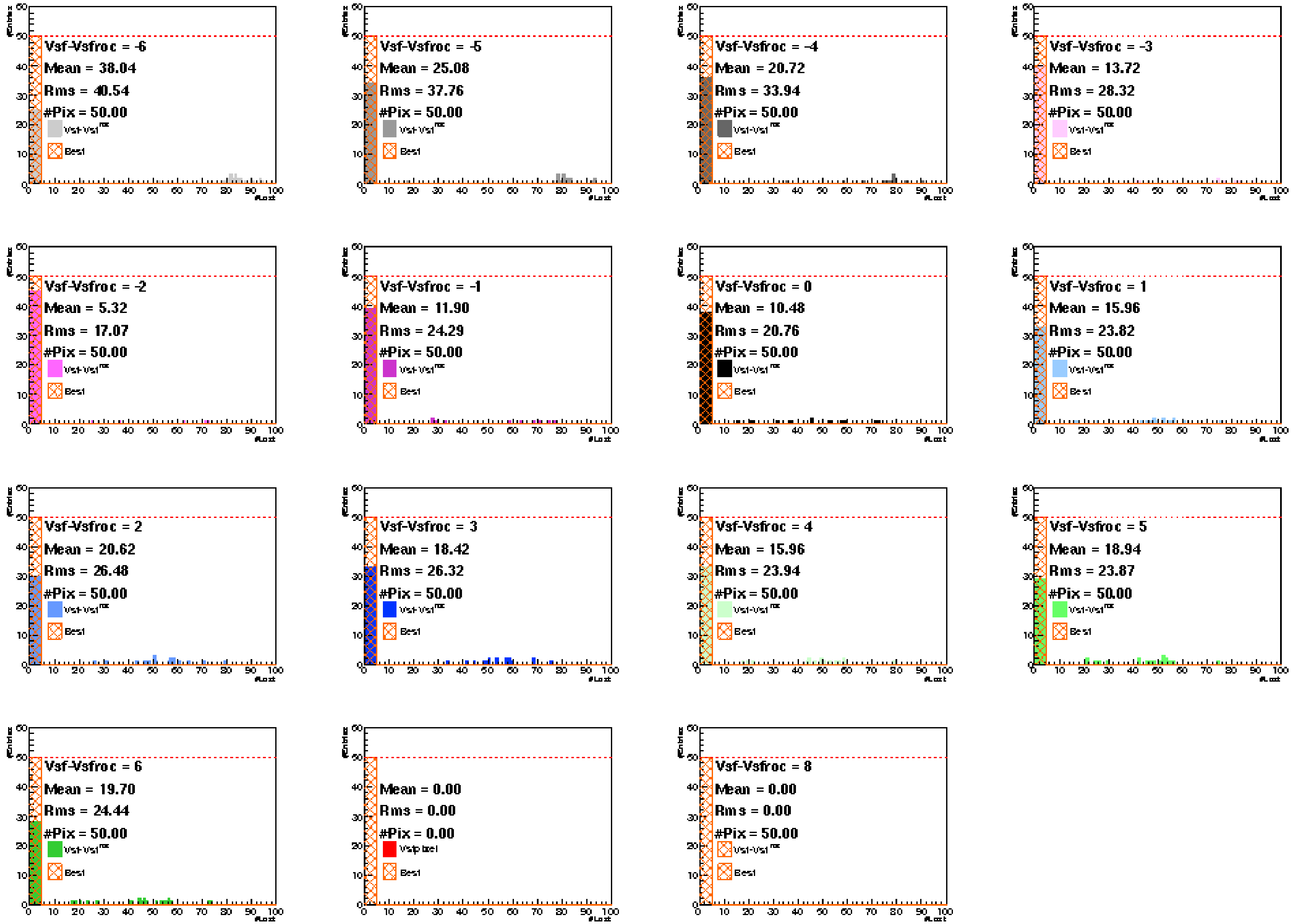
First Test : [Will change as wel learn more about the chip]

5. Minimize #Lost Points for Vana [*Scan within +/- 6 of current dac Parameter Value*]



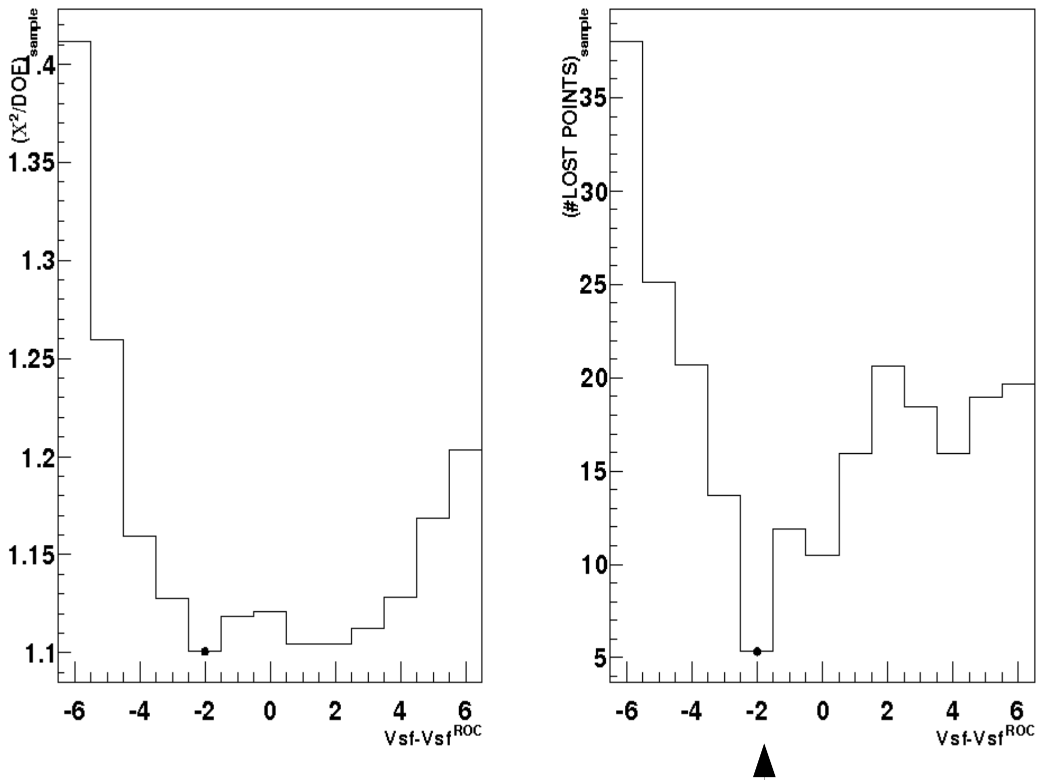
First Test : [Will change as wel learn more about the chip]

5. Minimize #Lost Points for Vana [*Scan within +/- 6 of current dac Parameter Value*]



First Test : [Will change as wel learn more about the chip]

5. Minimize #Lost Points for Vana [*Scan within +/- 6 of current dac Parameter Value*]

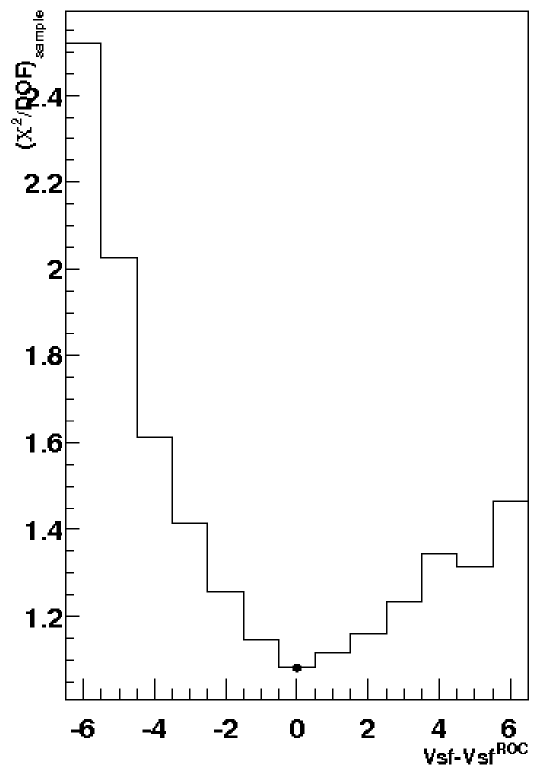


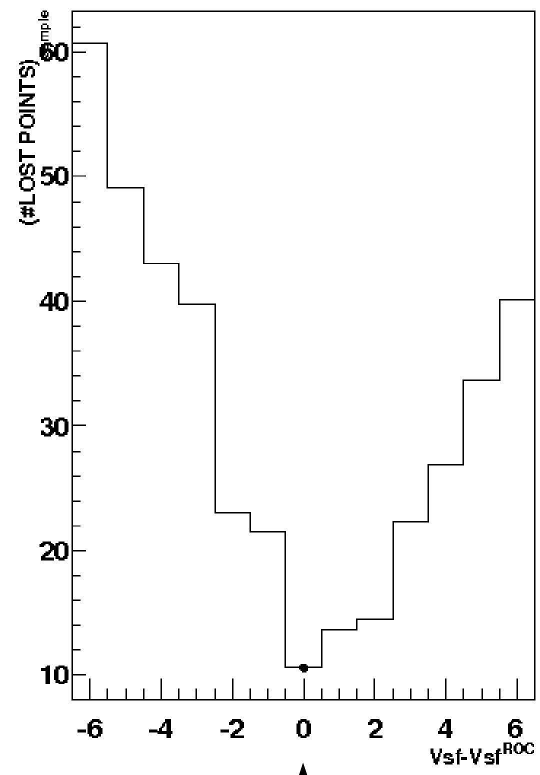
Vana 150 -> 148

Vopt−Vdac

First Test : [Will change as wel learn more about the chip]

6. Minimize #Lost Points for Vsf [*Scan within +/- 6 of current dac Parameter Value*]



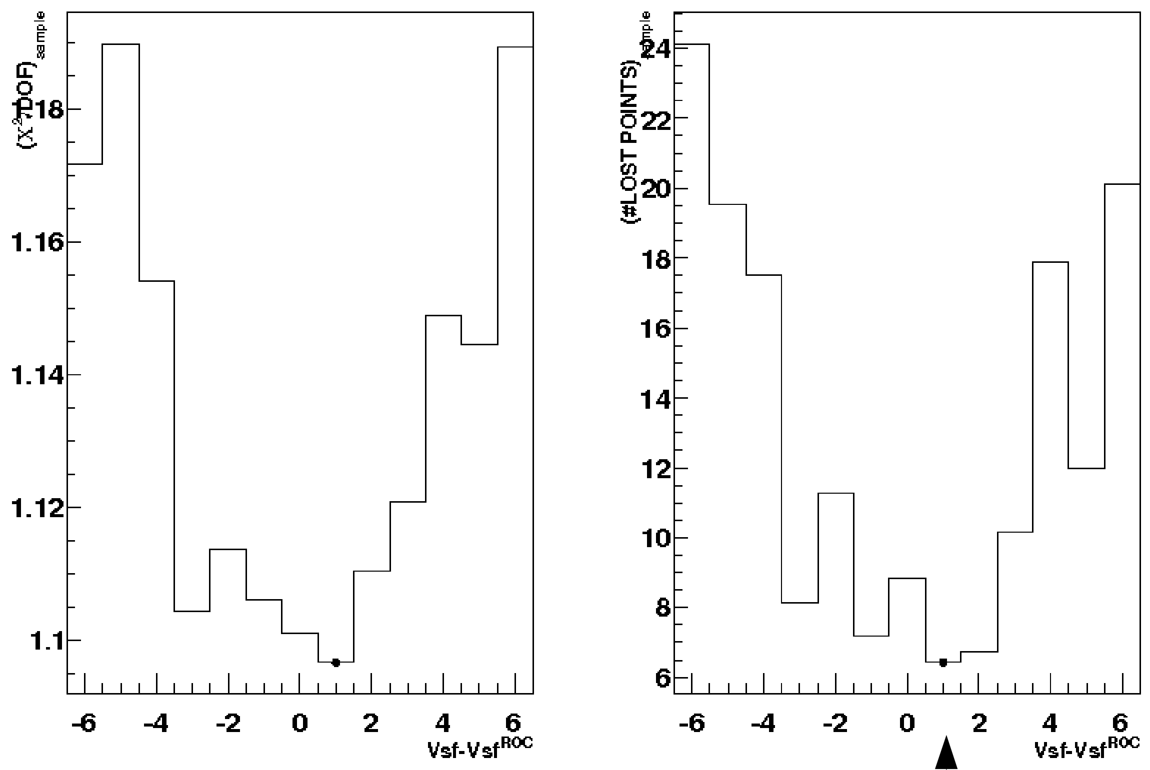


Vsf 163 -> 163

Vopt-Vdac

First Test : [Will change as wel learn more about the chip]

7. Minimize #Lost Points for VhldDel [*Scan within +/- 6 of current dac Parameter Value*]

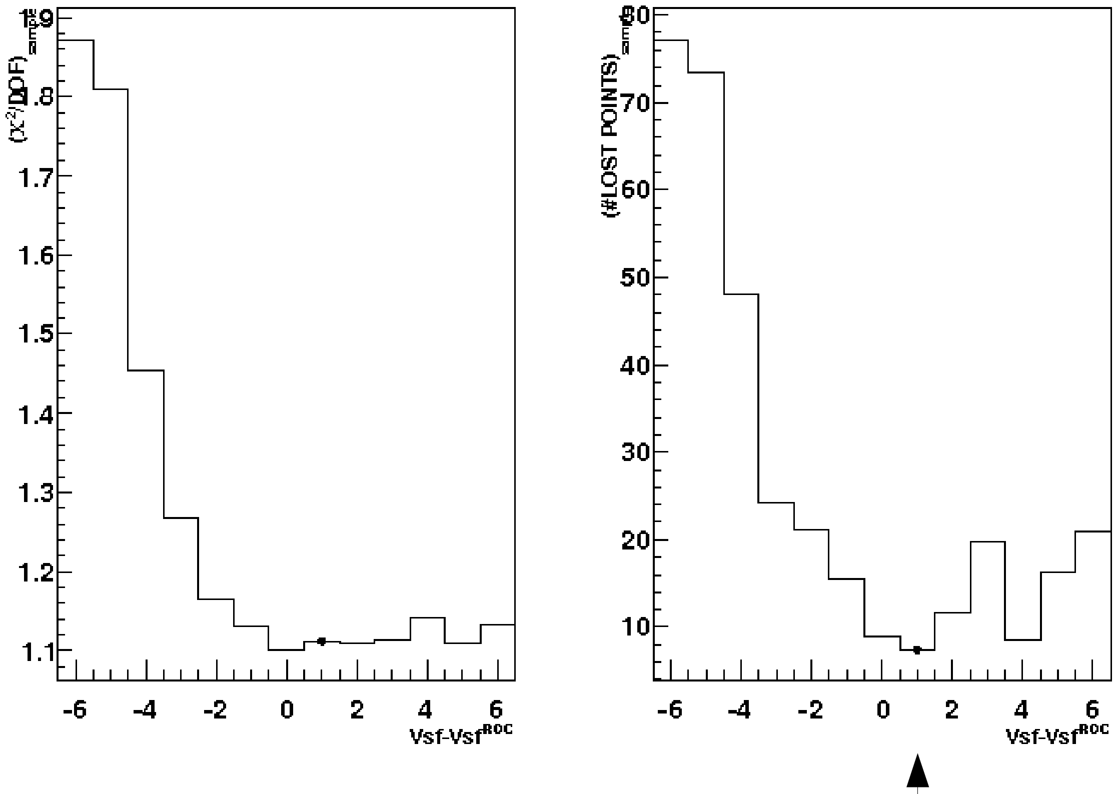


VhldDel 167 - > 168

Vopt-Vdac

First Test : [Will change as wel learn more about the chip]

8. Minimize #Lost Points for Vana [*Scan within +/- 6 of current dac Parameter Value*]



Vana 148 -> 149

Vopt-Vdac

Calibration Summary

Step			Vsf	VhldDel	Vana	chi2/ndf	**#lost**
0	svn dacParameters		170	150	150	1.46000	49.4
1	Vsf	ridge	162				
2	VhldDel	ridge		166			
3	Vsf	linearity	163			1.08561	5.38
4	VhldDel	linearity		167		1.10356	6.72
5	Vana	linearity			148	1.10046	5.32
6	Vsf	linearity	163			1.08067	10.5
7	VhldDel	linearity		168		1.09655	6.42
8	Vana	linearity			149	1.11222	7.38

After Linearity Scan [Step 5] {Vsf,VhldDel,Vana} = {163,167,148}
After Linearity Scan [Step 8] {Vsf,VhldDel,Vana} = {163,168,149}
Converged.

Here we take 8 steps. We could just decide to STOP if #lost is around 7 and chi2/dof around 1.1. Maybe takes 3 steps!

Need time to get experience.

Calibration Summary

Step	Opt. Type	Time	Duration/min
0	--	10:32	0
1	Ridge	10:57	29
2	Ridge	11:21	24
3	Linearity	11:42	21
4	Linearity	12:02	20
5	Linearity	12:23	21
6	Linearity	12:43	20
7	Linearity	13:03	20
8	Linearity	13:24	21

<u>TOTAL TIME</u>:

50 pix x 8 steps = 2:52 i.e. 3/4 hr 100pixel^{-1} step^{-1}

Reduce by (maybe) factor x5 going from 64- to 32-bit cpu

Conclusions

Calibration code does a good job in choosing dac parameters to optimize linearity.

Convergence occurs after 8 steps [maybe 5 (3?) with experience]

Need to test more pixels and chips.

Time needed estimated at 45 min per 100pixel per step [64-bit]; much better for 32-bit.

<u>OUTLOOK</u>:

* Like to look at more testing.
* Do the actual calibration Aout vs Vcal!
* Time Walk