aDefectFinder Status

~ aDefectFinder-00-02-02 ~

outline

- ★ Review the Algorithm
- ★ User Inputs & Outputs
- ★ Upcoming Features
- ★ Conclusions

Giulia Casarosa

INFN - Sezione di Pisa

8th VXD Workshop @ Trieste ~ September, 10th 2015

The Automatic Offline Analysis

➡ The goal is to provide a common tool to find (*defect-list generator*) and classify (*defect classifier*) the problematic strips found in the electrical tests performed with APVDAQ during the SVD assembly

➡ The *deliverable* of the automatic defect finder (aDefectFinder) is:

- the list of defective channels
- a *proposed* pre-assigned classification of the detected defects in the list above
- the relevant plots for each of the defect to assist the operator in the final decision
- the cumulative plots of the good strips

➡ The analysis is performed in two steps:

1. selection of the defective channels
2. classification of the defective channels
    - based on the features of the single channel under study; informations from the adjacent channels are not used (their utilisation can be implemented if needed)

Step 1: defects selection

➡ A strip is automatically listed as *problematic* if any of the following applies:

<table>
<tr><th>selection criteria</th><th>motivation</th></tr>
<tr><td>Noise > 8 ADC</td><td>most of the defects show an abnormal higher noise</td></tr>
<tr><td>CalAmp < 50 ADC || CalTmax > 200 ADC</td><td rowspan="2">some defects show an abnormal APV response in terms of gain and peaking time</td></tr>
<tr><td>CalTmax < 100 ns || CalTmax > 200 ns</td></tr>
<tr><td>it is recognised as a pinhole (see next slide)</td><td>pinholes may have a normal noise, gain and peaking time at Vsep = 0 V</td></tr>
<tr><td>LaserResponse < 0.5</td><td>some defects show an abnormal response to radiation</td></tr>
</table>

➡ The selection criteria have been chosen studying SBW and SFW modules:

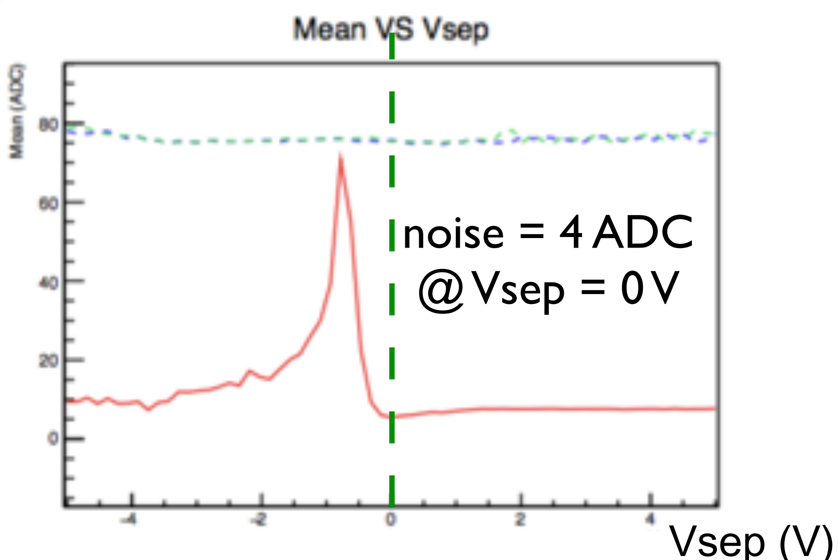
- modules in different layers and/or positions may need different cuts (including additional selection - or classification - criteria!)
- the cuts can be tuned by the operator changing the values in the file: `aDefectFinder-00-02-02/default_config/selection.config`

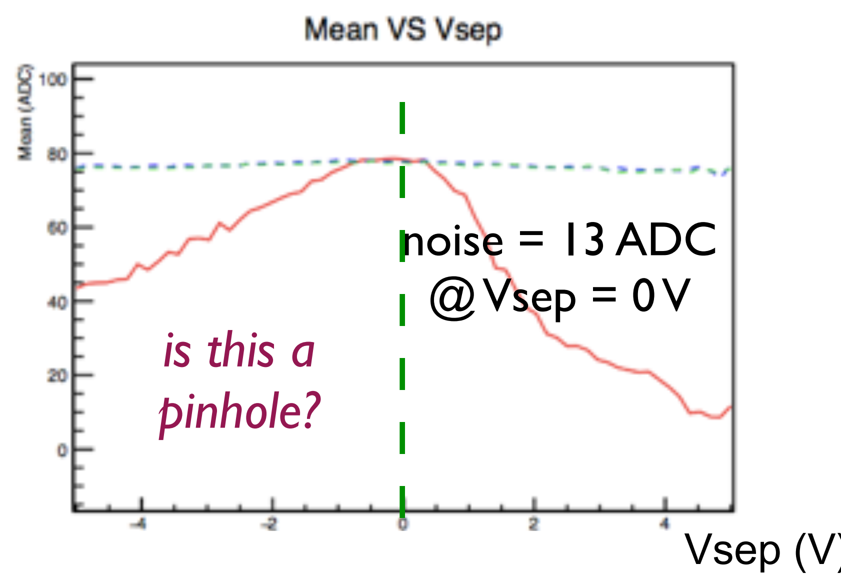
Pinholes Fingerprints

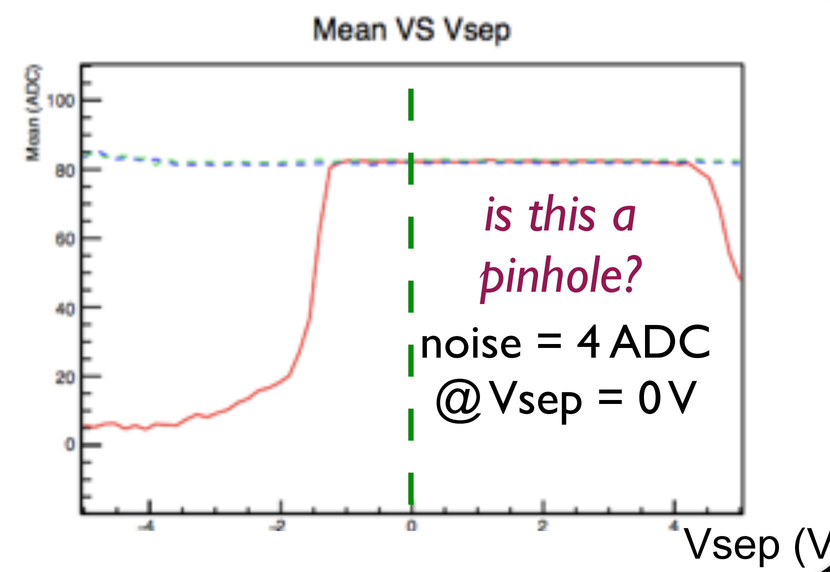
➡ Different pinholes "types" have been discovered during SFW and SBW tests:

SFW

the plots below report the Mean vs Vsep from the Vsep Scan APVDAQ run:

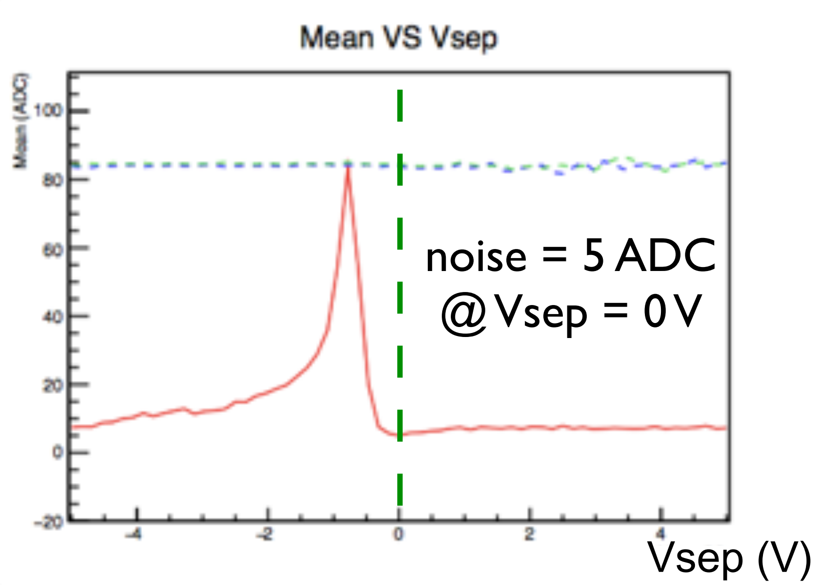


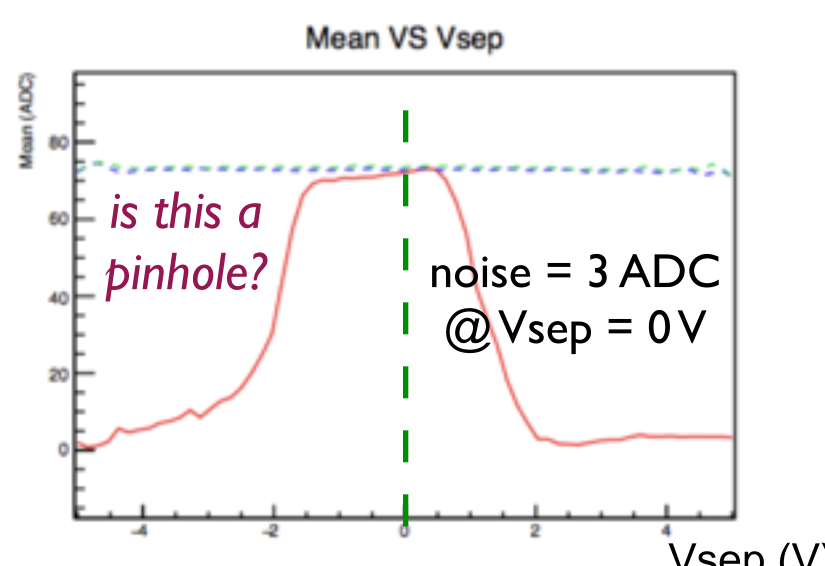


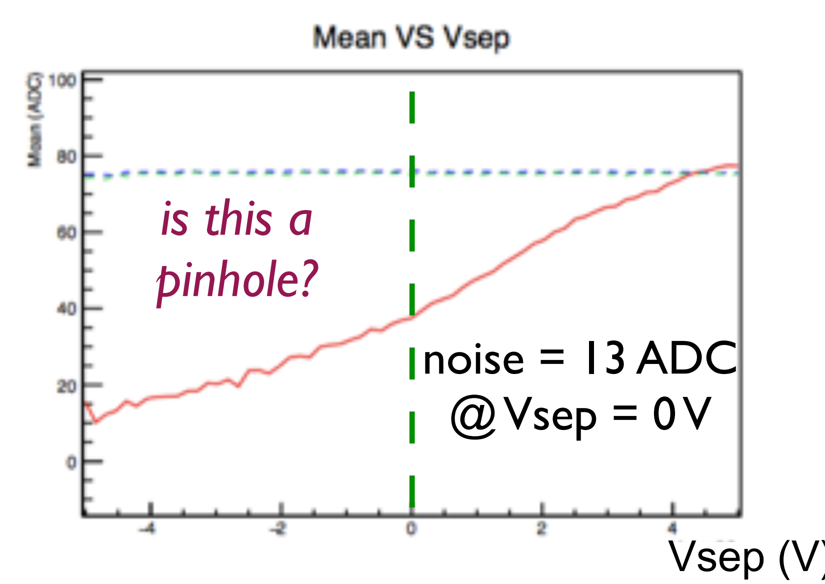


SBW

the plots below report the Mean vs Vsep from the Vsep Scan APVDAQ run:

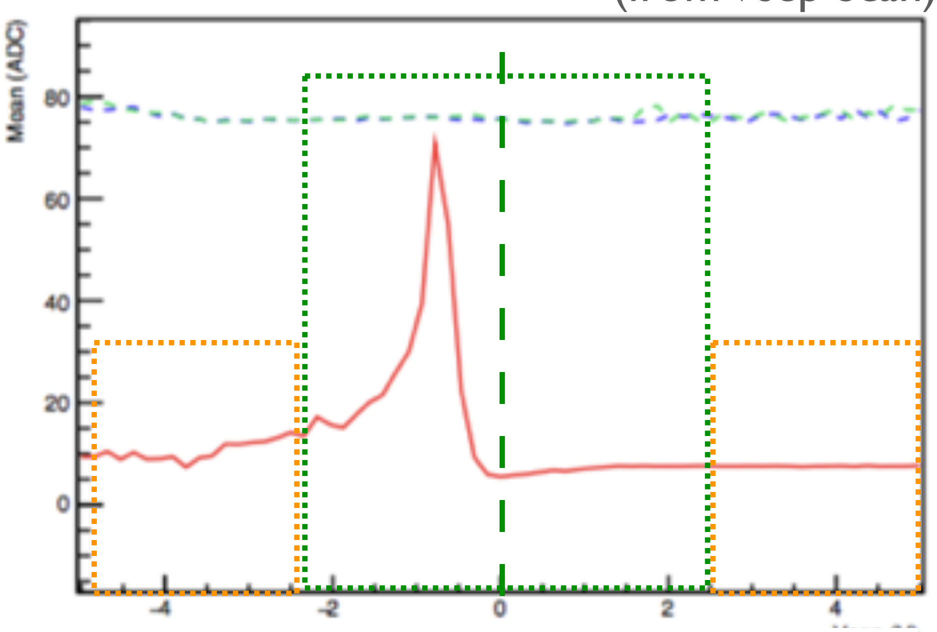






is-a-pinhole Criteria

Mean vs Vsep (from Vsep Scan)



Vsep (V)

1. compare the maximum of Mean in the central region of Vsep with the average of Mean for |Vsep| > 2.5V:

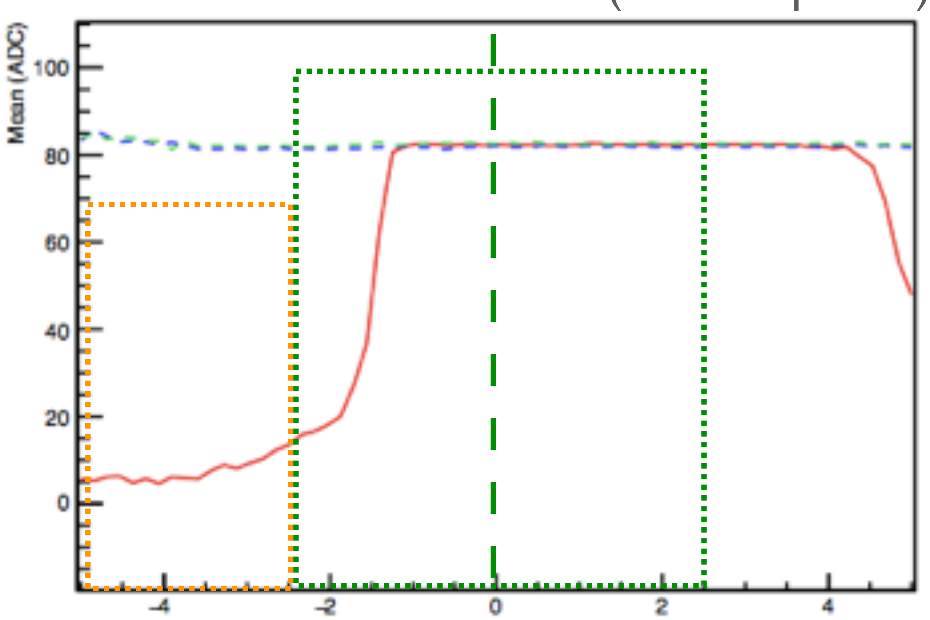
if $|average_{LR}(Mean) - max_C(Mean)| > 20 \rightarrow$ Pinhole

average of Mean for:
- *-5 < Vsep < -2.5 and*
- *2.5 < Vsep < 5*

max of Mean for:
- *|Vsep| < 2.5 V*

~ 5 x RMS(Mean)

Mean vs Vsep (from Vsep Scan)



Vsep (V)

2. compare the maximum of Mean in the central region of Vsep with the average of Mean for Vsep < – 2.5V:

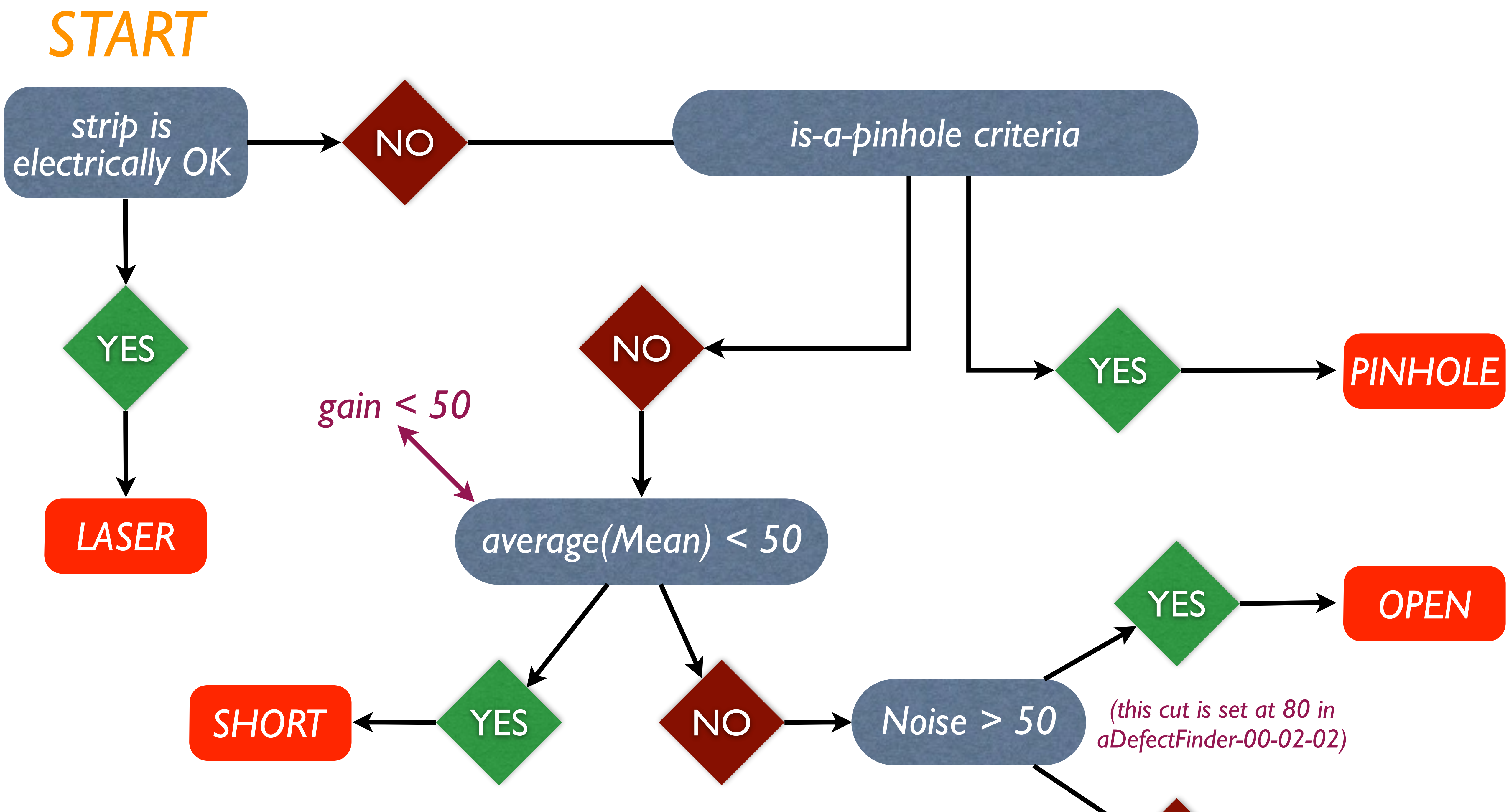
if $|average_{L}(Mean) - max_C(Mean)| > 20 \rightarrow$ Pinhole

average of Mean for:
- *-5 < Vsep < -2.5*

max of Mean for:
- *|Vsep| < 2.5 V*

~ 5 x RMS(Mean

Step 2: defects classification

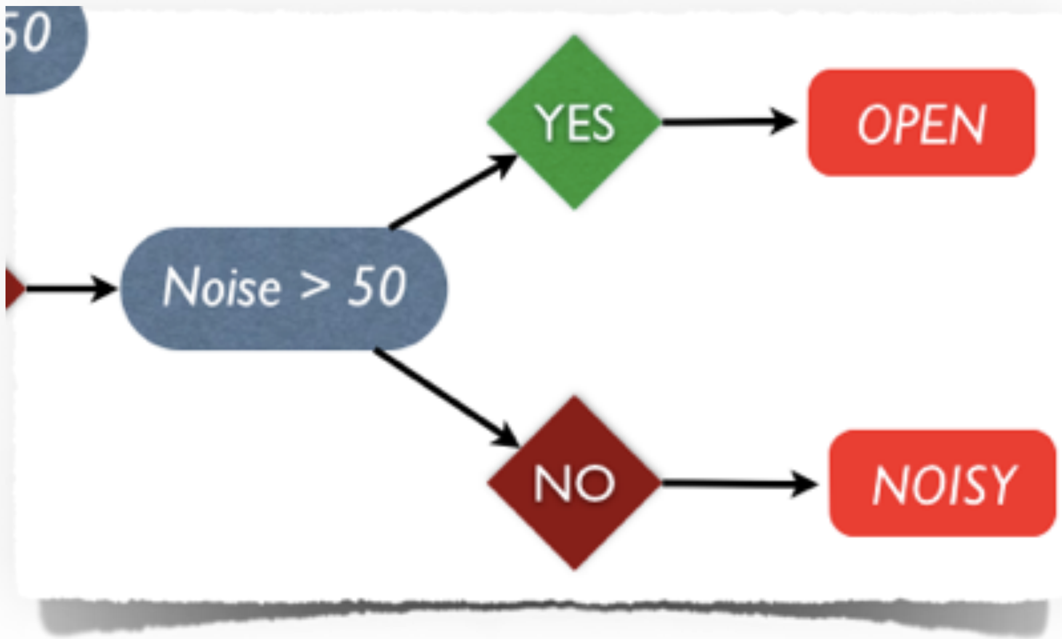


➡ The numerical cut values have been chosen studying SBW and SFW modules:

- modules in different layers and/or positions may need different cuts
- the cuts can be tuned by the operator changing the values in the file `aDefectFinder-00-02-02/default_config/classification.config`

Tagging Opens

➡ It's very important to find the opens during the first electrical tests since they may be repaired



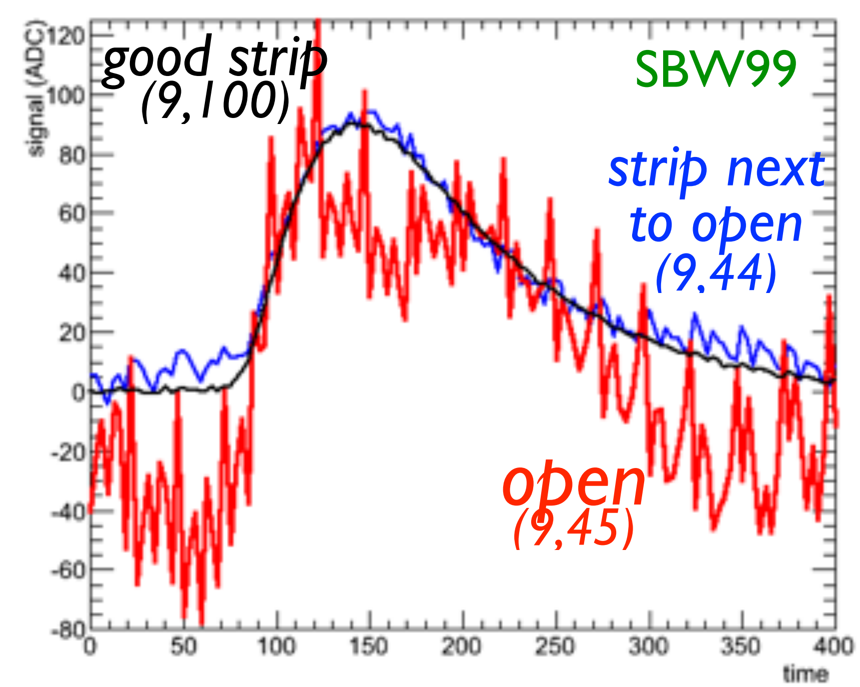
➡ At present, the discrimination between a noisy strip and an open is a cut on the Noise of the channel

➡ The value is lowered form 80 to 50 ADC in order to flag all opens as opens

➡ If a noisy strip is flagged as open it can be corrected by the **operator**

three adjacent strips with the central one with higher noise is a clear indication of an open

Response Curve



Configuration File

```
[module]
name = SBWtest
tag = bw
```

name of the object (L4.001, SB3.001, …)

position in the ladder = {bw, -z, ce, +z ,fw}

```
[input files]
calibration = ./default_trees/default_cal_tree.root
vsep = ./default_trees/default_cvs_tree.root
laserP = ./default_trees/default_laserP_tree.root
laserN = ./default_trees/default_laserN_tree.root
```

default trees exist in case you have no laser/radiation run

```
[output files]
rootfile = ../results/SBWtest/SBWtest_mergedTree.root
csv_defects = ../results/SBWtest/SBWtest_defects.csv
pdf_summary = ../results/SBWtest/SBWtest_summary.pdf

[Average Laser Response Cuts]
count_min = 0
count_max = 3000

[Defect Finding Cuts]
include ./default_config/selection.config
```

change this file to change the selection cuts

```
[Defect Classification Cuts]
include ./default_config/classification.config
```

change this file to change the classification cuts

```
[Electrical Defects Analysis]
output = ../results/SBWtest/SBWtest_electrical_defects.csv
include ./default_config/electrical_defects_without_sensor.config

[Package Version]
include ./default_config/package_version.config
```

- ➡ paths are relative to where you run the program
  - use aDefectFinder location and use relative paths to it (as in the example)
  - use another folder (e.g. data folder) and use absolute paths in the config file
- ➡ you can modify the selection and classification cuts
- ➡ check the screen printout right at the beginning of the execution to check if:
  - included files have been found
  - cuts are the ones you expect
  - the module has been correctly recognised (# strips,…)

in the near future a tool to create the configuration file will be provided and will automatically set the path and include the proper selection/classification files

The Output Files (1)

summary of cuts, input/output files

SBW006_bw Offline Analysis Results

[input files]

calibration = ../data/2015_08_03/SBW006_cal_20150803_1525.root

vsep scan = ../data/2015_08_03/SBW006_cvs_20150803_1528.root

laser P-side = ../data/2015_08_03/SBW00620150803_1549_001.root

laser N-side = ../data/2015_08_03/SBW00620150803_1604_002.root

[selection criteria]

Noise > 8.0
CalAmp < 50.0 || CalAmp > 150.0
CalTmax < 100.0 || CalTmax > 200.0
abs(LaserResponse - 1) > 0.5
is-a-pinhole criteria

[classification criteria]

Pinhole: - abs(average_LR - max_C) > 20.0
- abs(average_L - max_C) > 20.0

Short: average(Mean) < 50.0

Open: Noise > 80.0

[output files]

root file = ../results/SBW006/SBW006_mergedTree.root

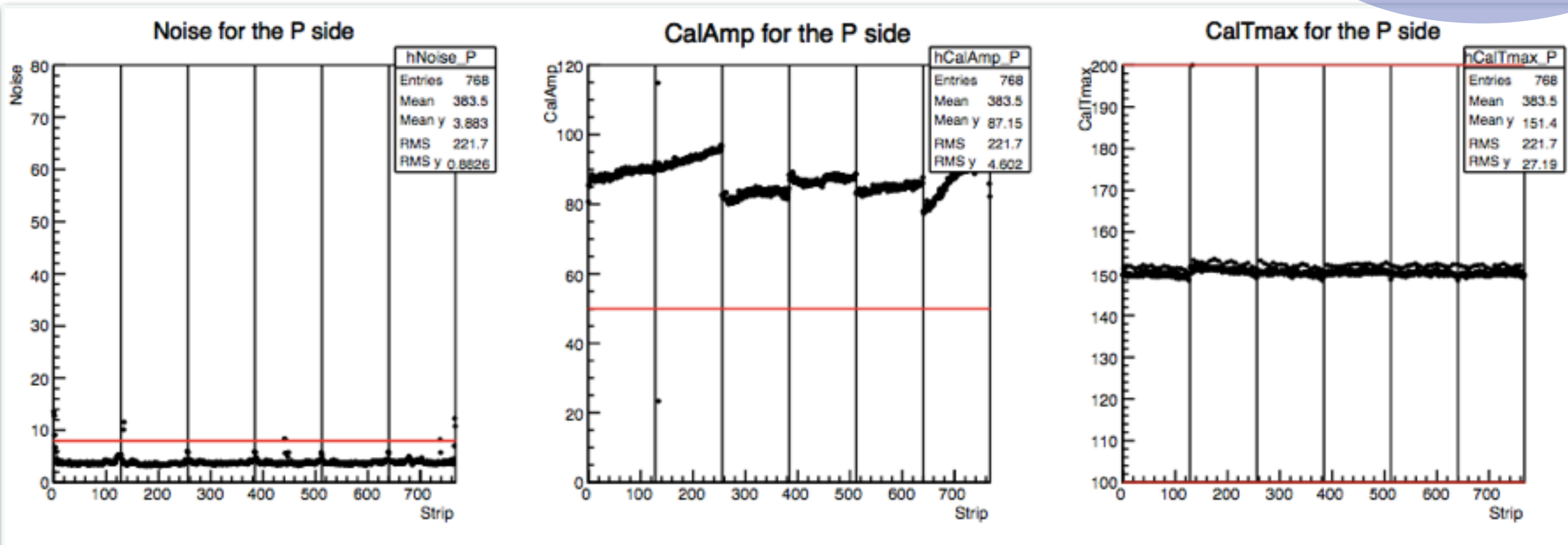
csv file = ../results/SBW006/SBW006_defects.csv

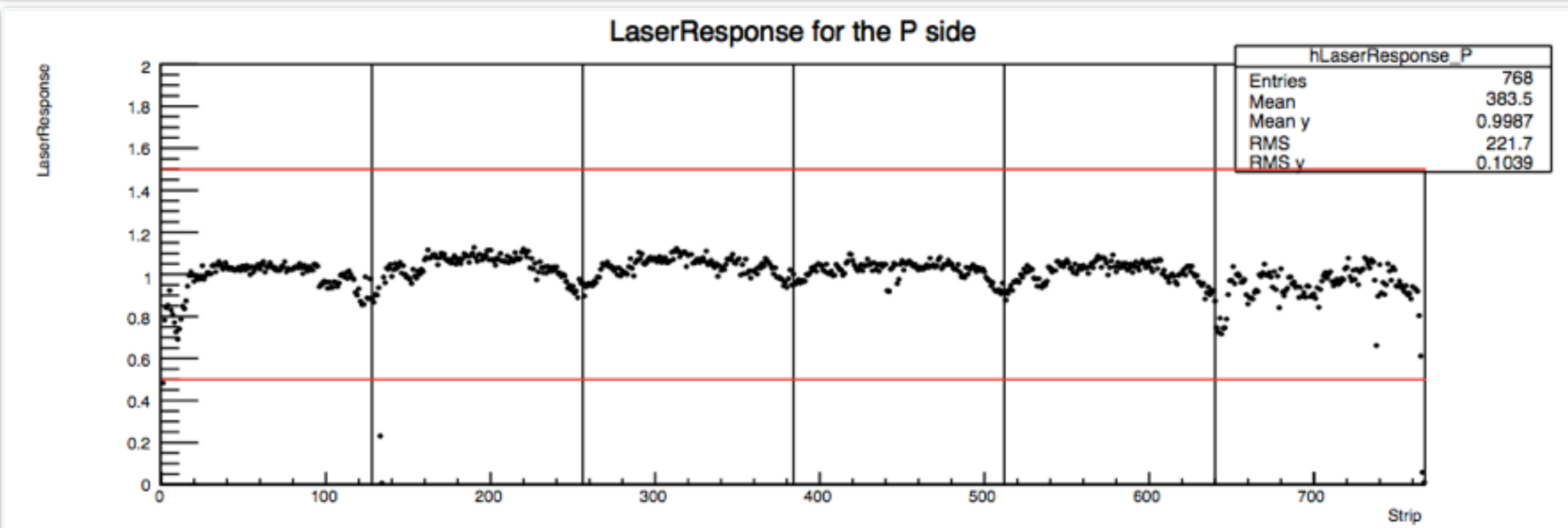
pdf file = ../results/SBW006/SBW006_summary.pdf

package version: aDefectFinder-00-02-02

The Output Files (2)

old-style plots of the relevant variables

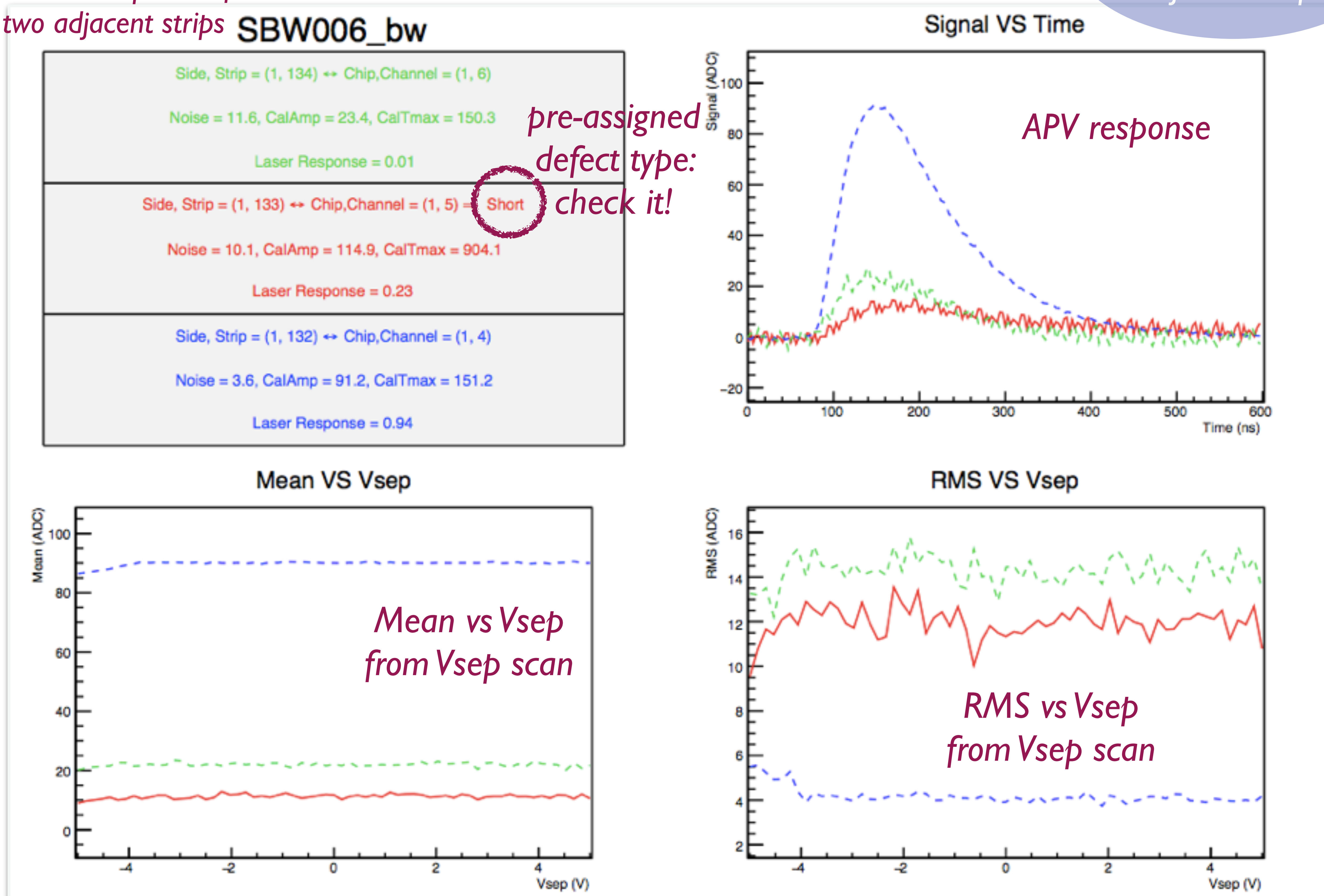




The Output Files (3)

summary of the characteristics of the problematic strip and of the two adjacent strips

relevant plots for each defects and the adjacent strips



pre-assigned defect type: check it!

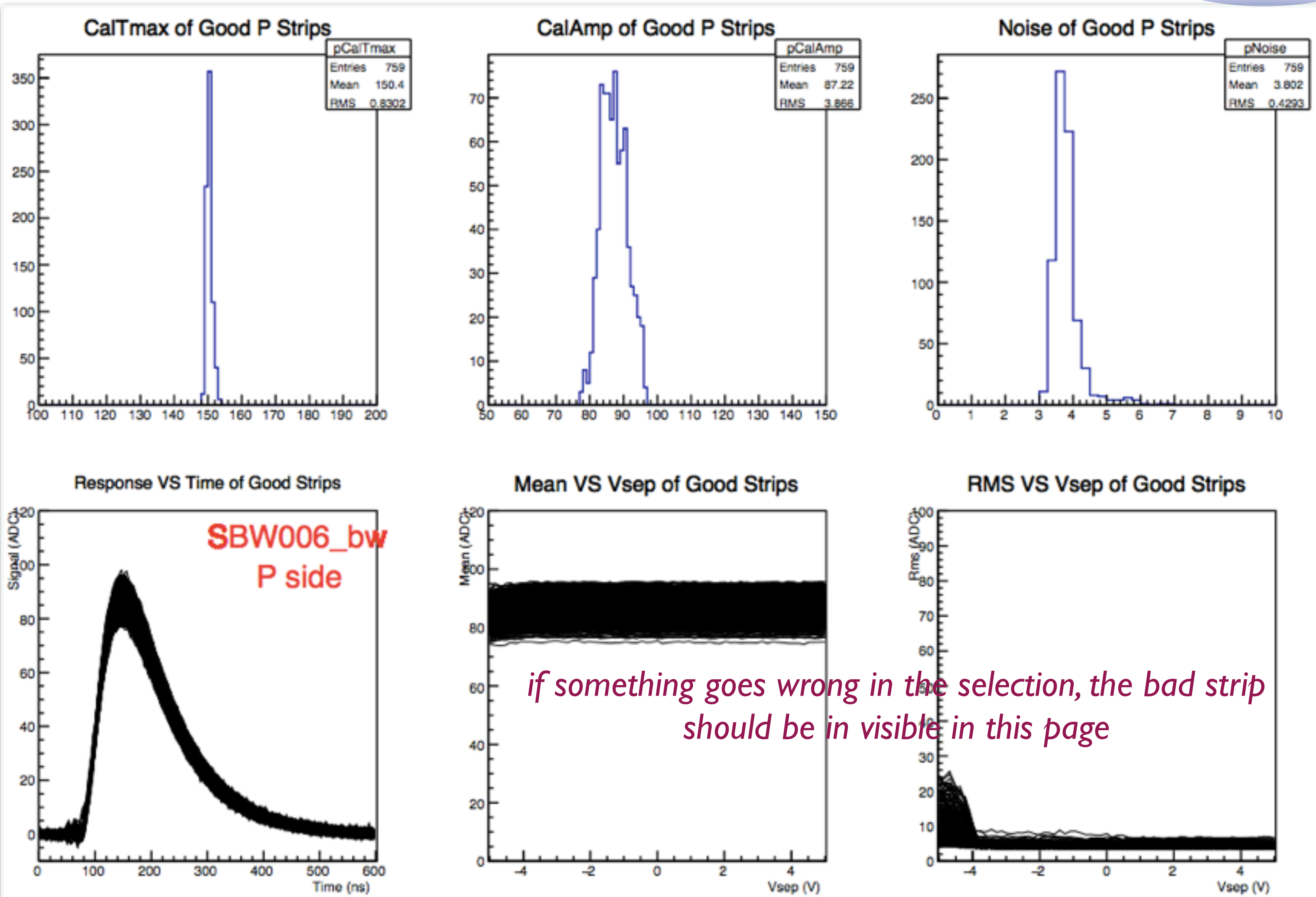
APV response

Mean vs Vsep from Vsep scan

RMS vs Vsep from Vsep scan

The Output Files (4)

cumulative plots of the good strips



if something goes wrong in the selection, the bad strip should be in visible in this page

The Features Under Development

1. Automatic Configuration File
   - a tool that automatically creates the configuration file is under development by a student in HEPHY (Daniel Lukic)
2. Automatic comparison of the defects found during the electrical test and the defects declared on the sensor
   - minimum deliverable is the list of matched defects and the list of channels that are found only in one test (sensor or electrical).
   - format of the files to be compared is under finalisation
   - format of the output file is under discussion:
     - it depends on the level of comparison that we want
     - it depends on what we want to do with it (e.g.: count the matched defects, build a statistics, …)
   - input and output files should be uploaded/downloaded from the database

Conclusions

- ➡ aDefectFinder-00-02-02 is available at:
  - https://belle2.cc.kek.jp/svn/groups/svd/aDefectFinder-tags/aDefectFinder-00-02-02/
  - more information on the installation, compilation and usage in the README file; more information in a Twiki page to be created very soon.
- ➡ aDefectFinder is still under development but it already provides useful informations for APVDAQ users
- ➡ Feedbacks from the users is fundamental to improve the performances of the algorithm (giulia.casarosa@pi.infn.it)
- ➡ Additional features will be implemented soon in order to simplify the user life.

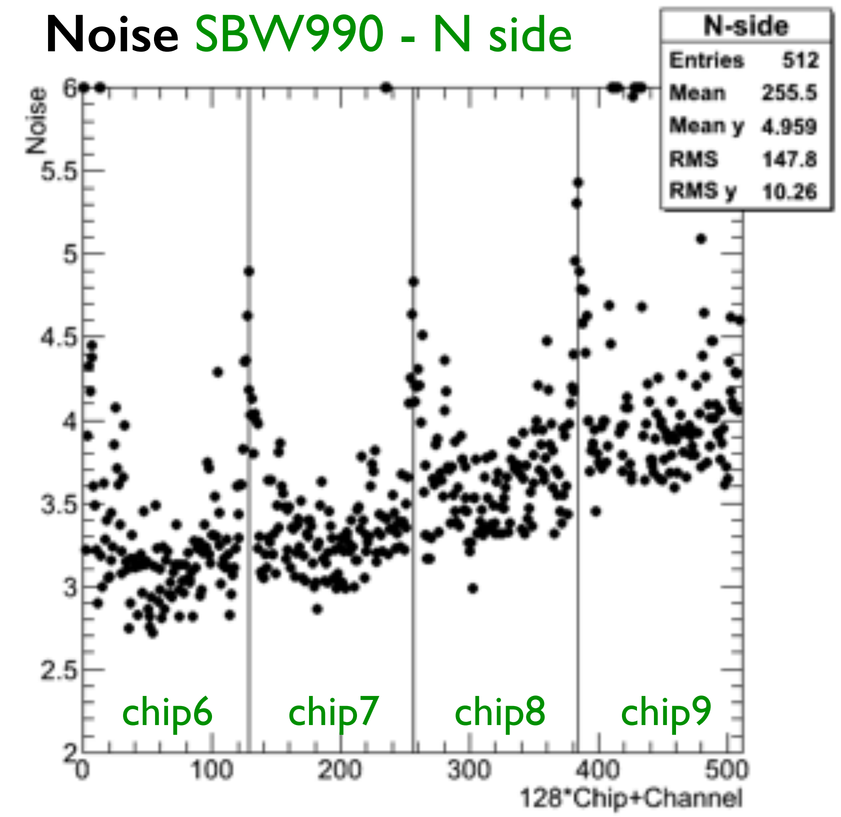
Thank You!


Don't
PANIC!
backup slides

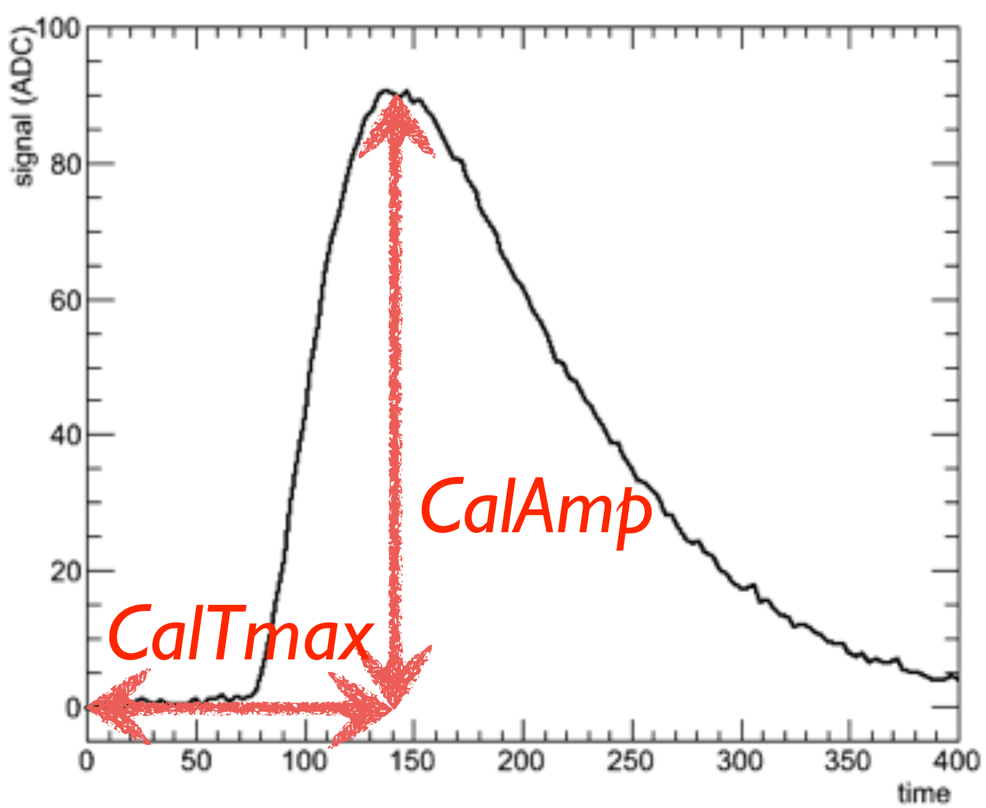
APVDAQ Calibration Run

- 600 events randomly triggered to evaluate Noise, RawNoise and Pedestal for each channel
- fixed ΔV injected on the capacitance of the APV injection circuit of all channels, sampling of the response curve of 16 strips at a time (8 groups, strips i+8j with j = 0to16 are in the i^{th} group)

Noise SBW990 - N side



Response Curve

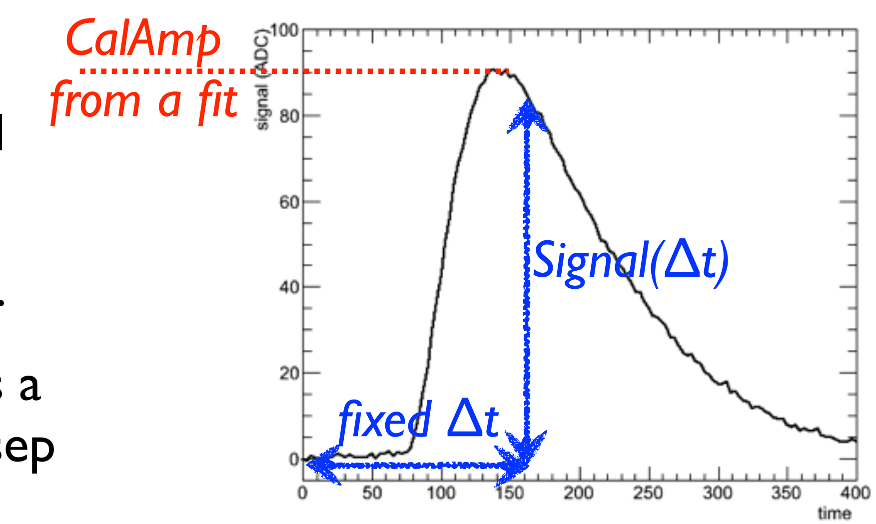


- The maximum amplitude (CalAmp) and the peaking time (CalTmax) are <u>extracted with a fit to the curve</u>
- WARNING: in channels with very high noise (>50 ADC) the fit can fail and return crazy values → look at the response curve for that channel
- *Temperature effect*: the hybrids heat up when DAQ is running (up to ~100°C). The performance decrease with temperature:
  - *decrease of CalAmp*
  - *increase of CalTmax*

APVDAQ Vsep Scan

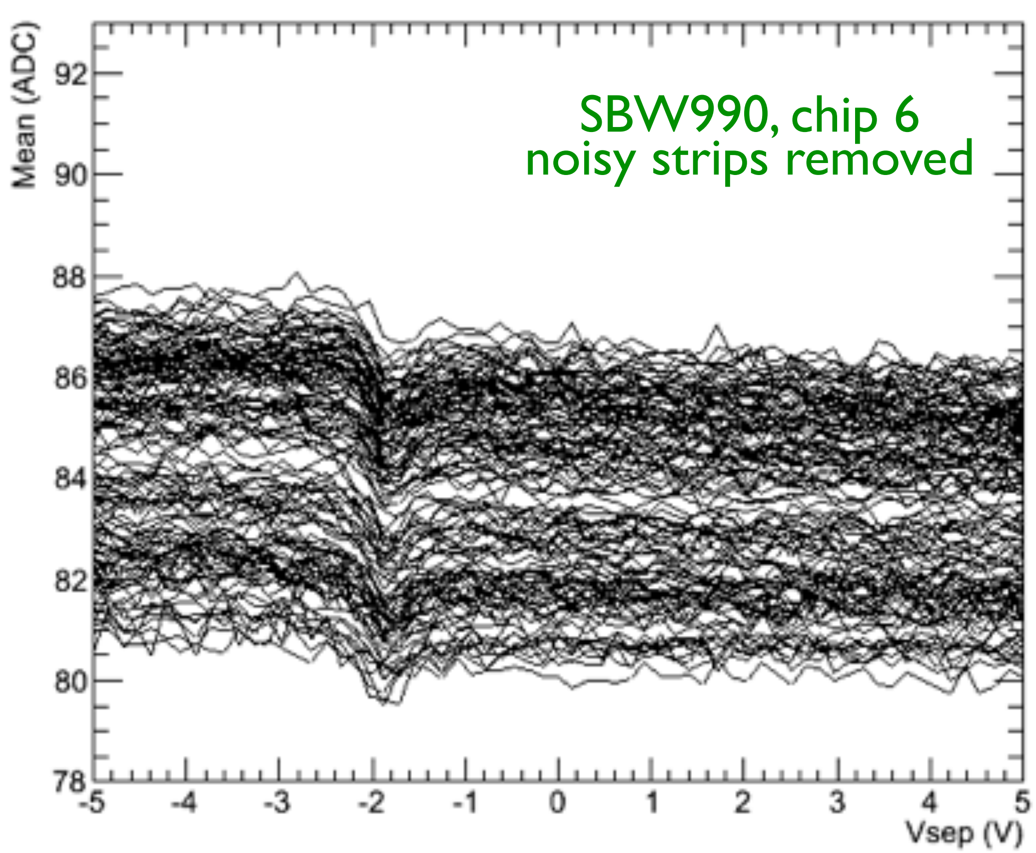
- 600 events randomly triggered to evaluate Noise, RawNoise and Pedestal for each channel
- APV response evaluated at a fixed Δt (no fit) for different values of Vsep.
  - average (Mean) and RMS of the distribution of Signal(Δt) are plotted as a function of Vsep

Response Curve



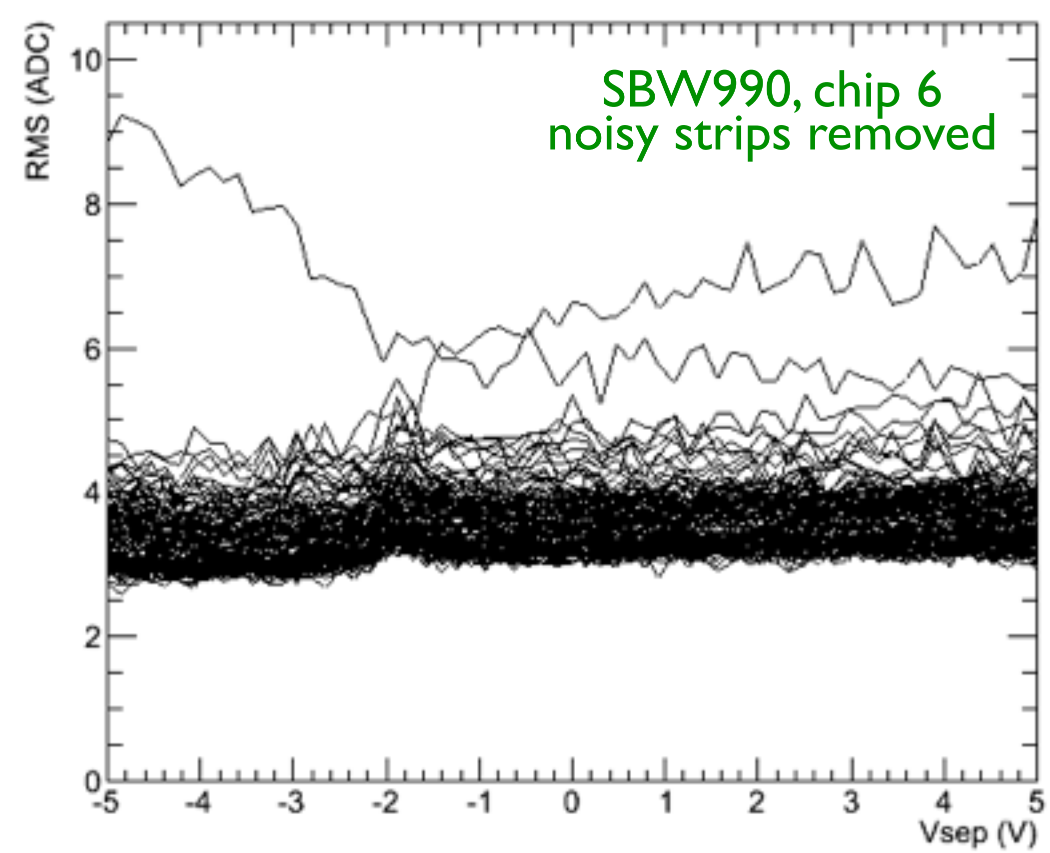
Mean vs Vsep

$$\text{Mean} = \frac{\Sigma_i^N Signal_i(\Delta t)}{N}$$



RMS vs Vsep

$$\text{RMS} = \sqrt{\frac{\Sigma_i^N (Signal_i(\Delta t) - Mean)^2}{N-1}}$$

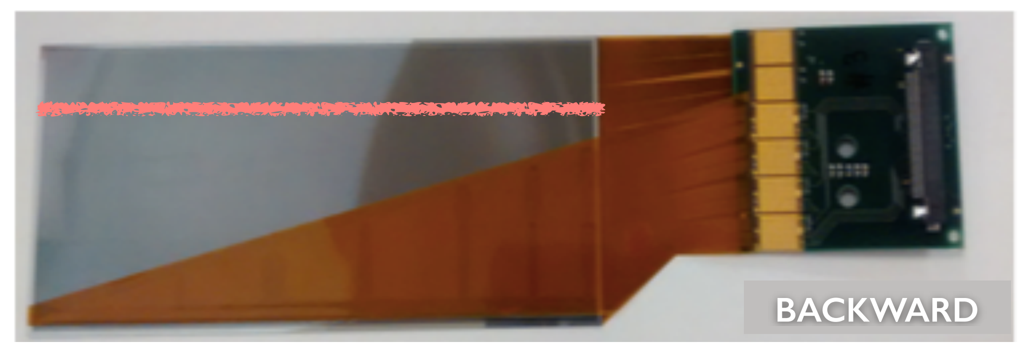


Laser Scan

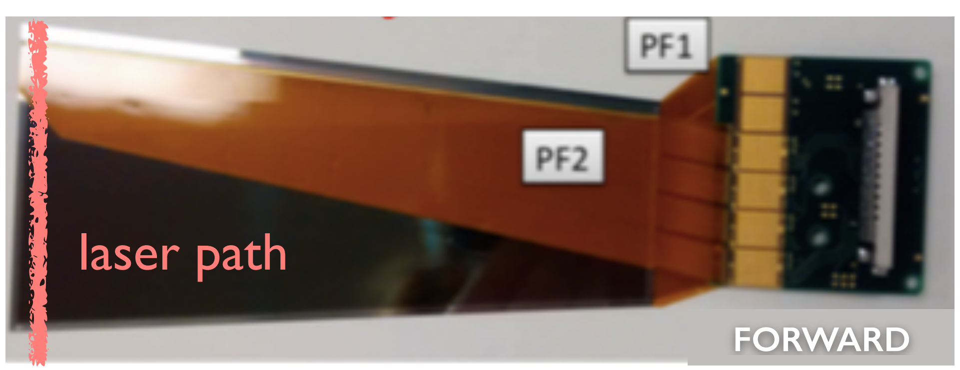
The subassembly (N-side up) is placed on the plexiglass support (fixed to the box) that provides a good alignment

1. apply the bias, V_{bias} = 100V, Vsep = -0.75V *in case of pinholes*, Vsep = 0 V otherwise
2. two Hardware Runs, APVDAQ (external trigger for the laser pulse and the APVDAQ)

- scan of the N strips (~10 minutes, 1500 hits per strip on average):
  - move the laser at a constant speed ~orthogonal to the N strips, away from PA if possible.



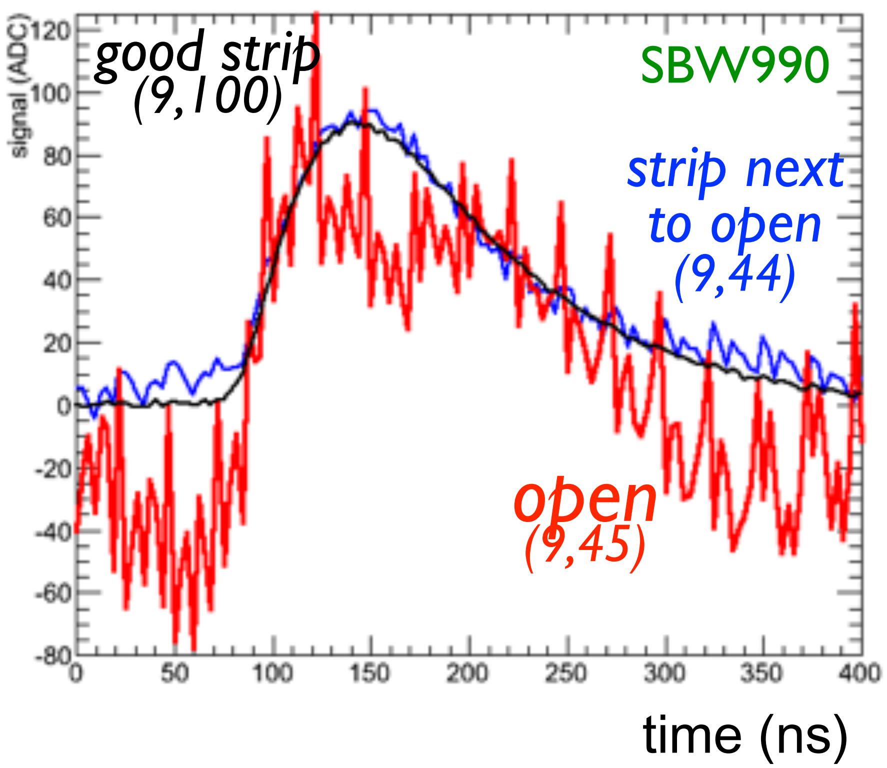
- scan of the P strips (~10 minutes, 1500 hits per strip on average):
  - move the laser at a constant speed ~orthogonal to the P strips, away from PA if possible.



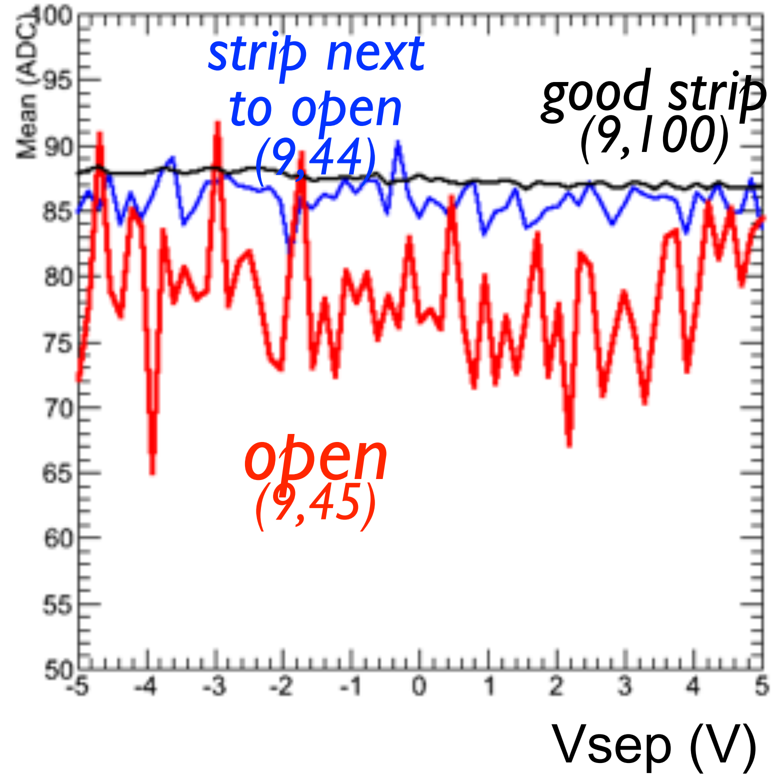
NOTE: before each scan we take 600 events randomly triggered to evaluate Noise, RawNoise and Pedestal for each channel

Opens Fingerprint

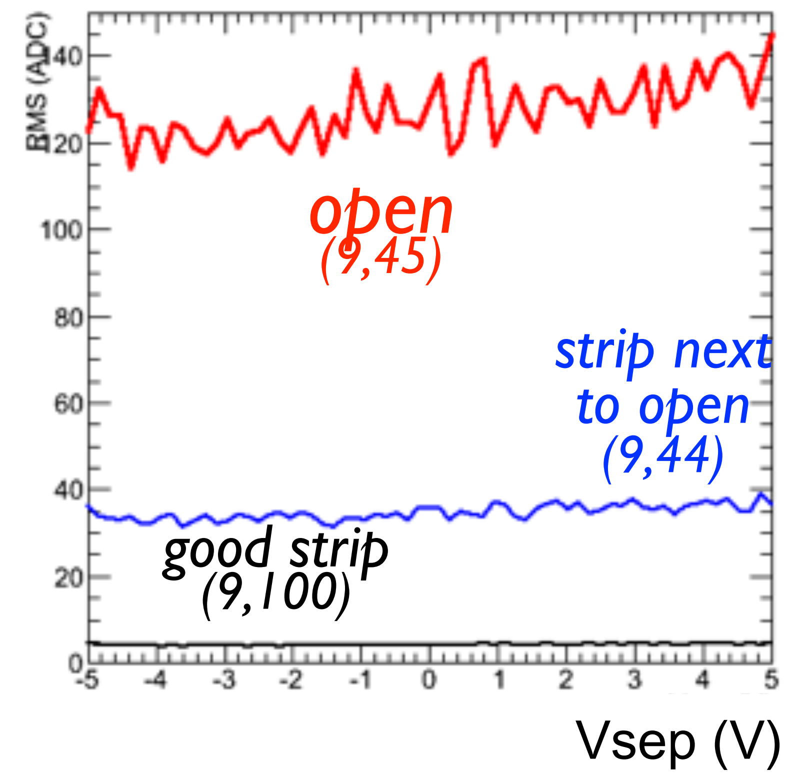
Response Curve



Mean VS Vsep



RMS vs Vsep



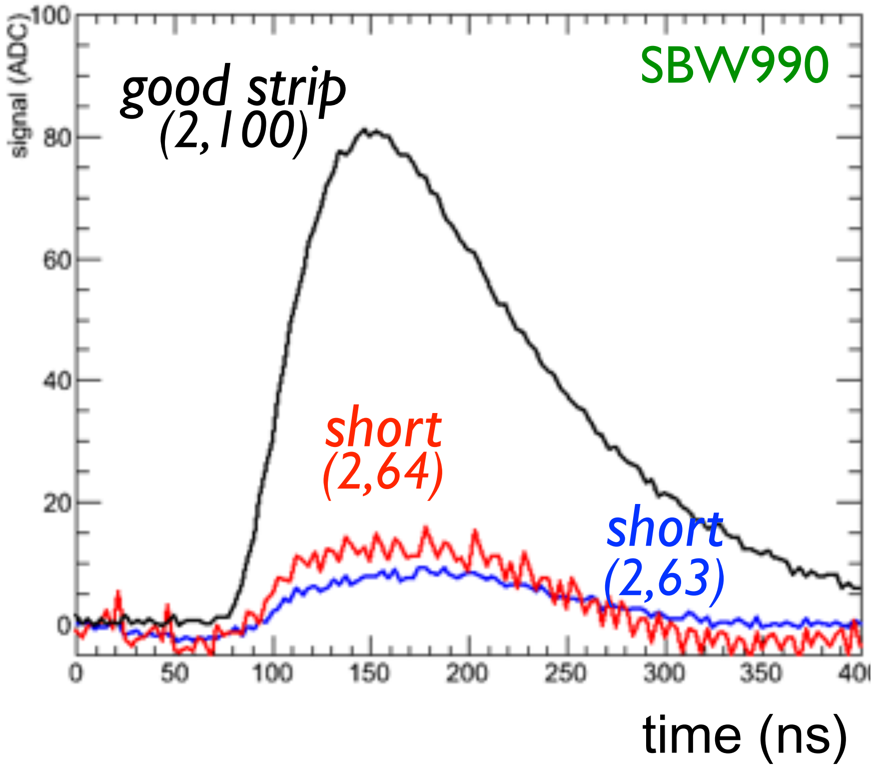
- same behaviour if the open is on the sensor or on the APV side
- very high Noise (high noise also on the 2+2 adjacent strips)
- Normal CalAmp and CalTmax but the fit to the response curve may converge to crazy values
  → look at the response curve to evaluate "by eye" if the gain is normal or low

- Laser Response:
  - affected by the high noise
  - the 2+2 nearby strips have a lower response to radiation because of their higher noise

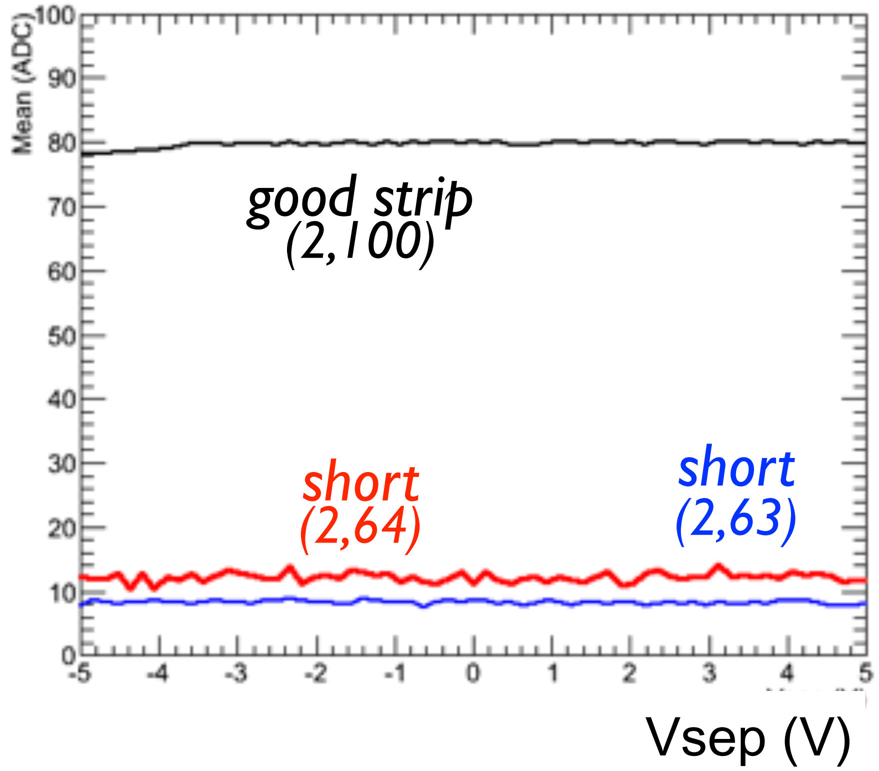
note: strips are indicated with the convention (APVchip, APVchannel)

Shorts Fingerprint

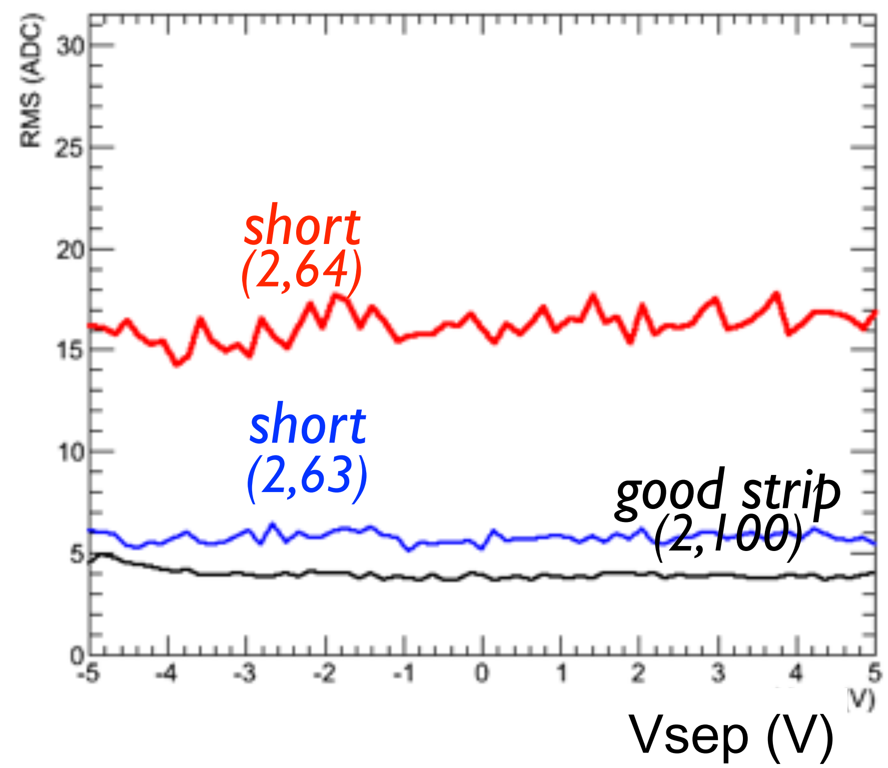
Response Curve



Mean vs Vsep



RMS vs Vsep

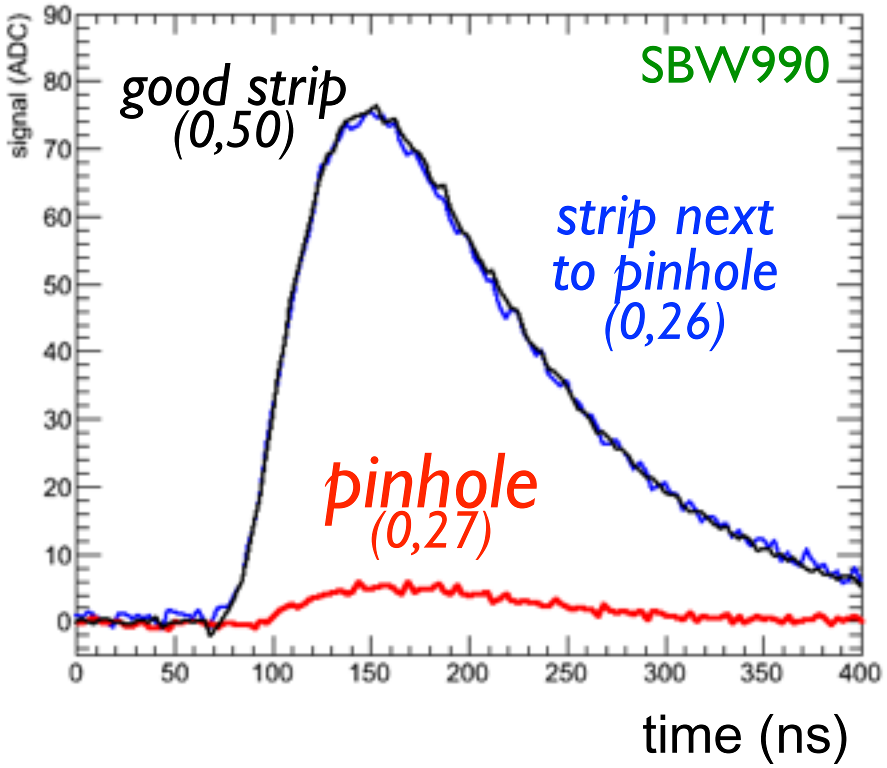


- shorts consist in at least two adjacent strips
- Lower CalAmp and longer CalTmax, usually the fit to the curve converges, but it's always better to check the values of CalAmp and CalTmax looking at the response curve
- high Noise

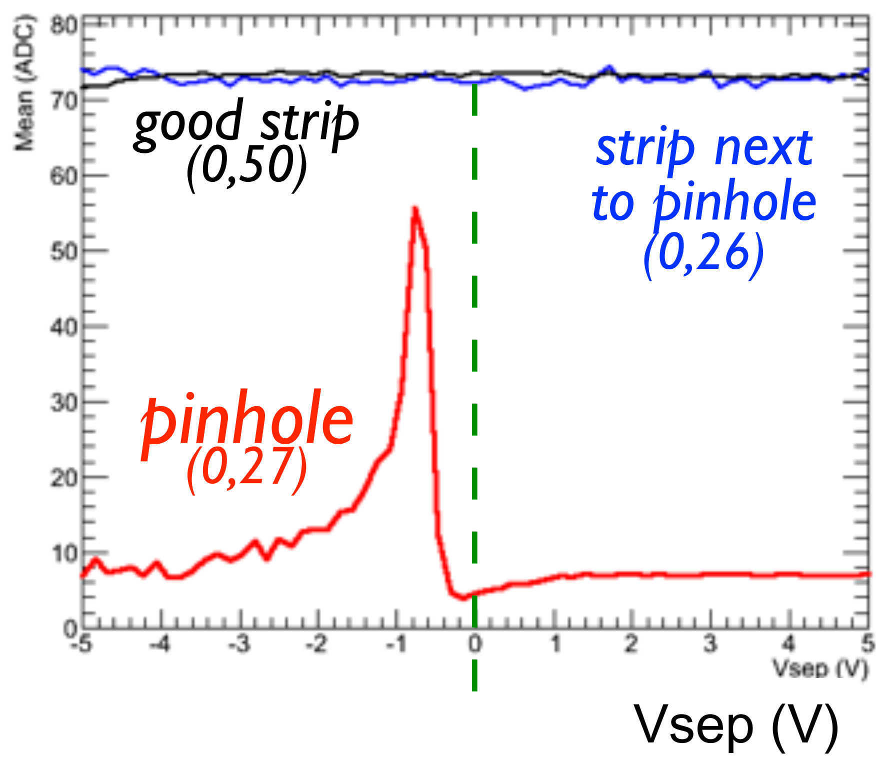
- Laser Response:
  - affected by the lower gain

Pinholes Fingerprint

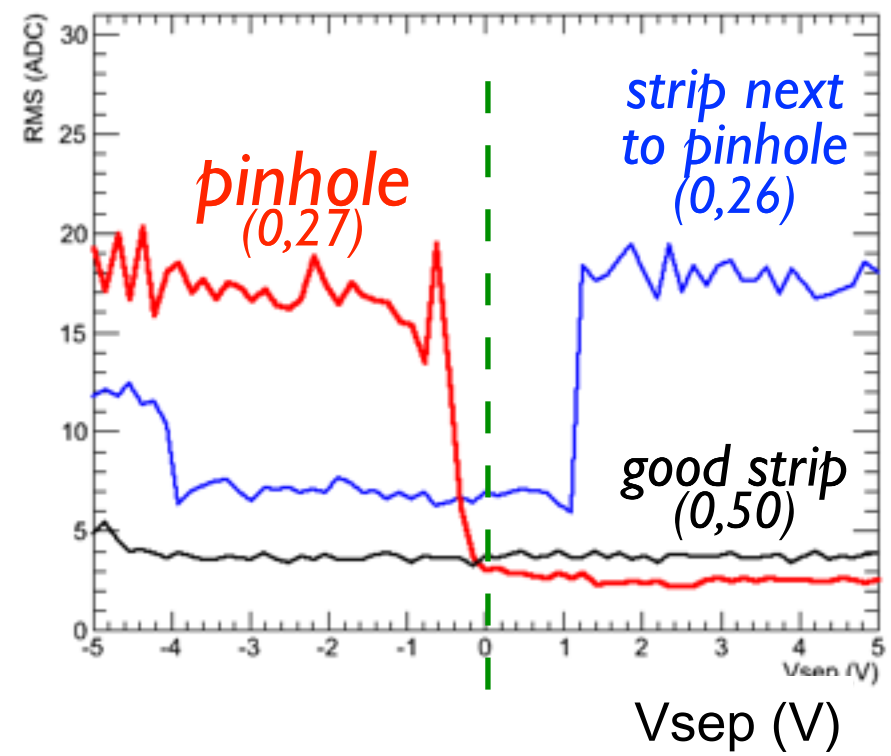
Response Curve



Mean vs Vsep



RMS vs Vsep



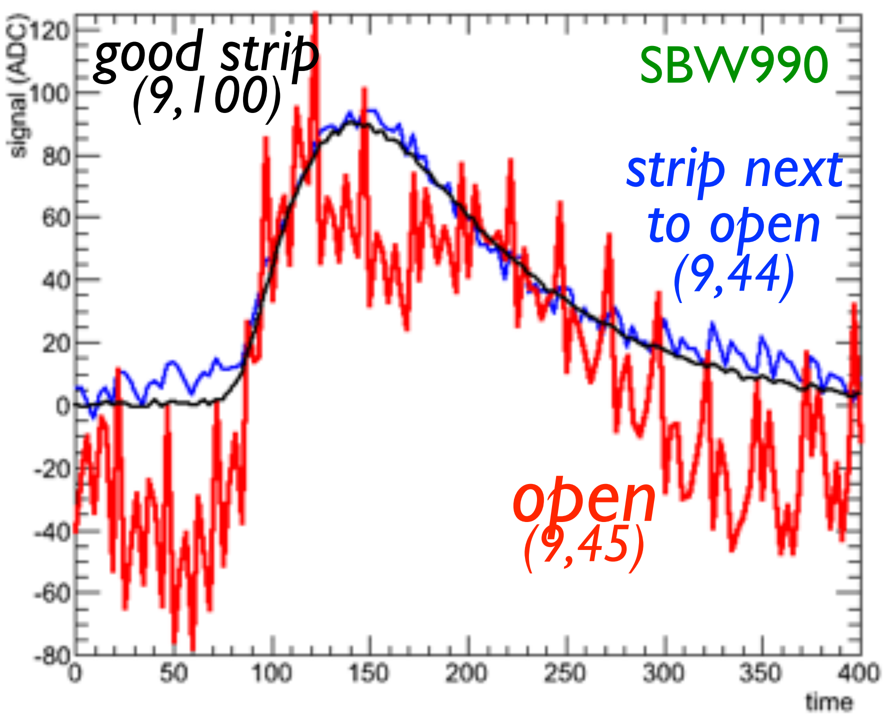
- Lower CalAmp and longer CalTmax at Vsep = 0V
- Gain (partially) recovered at Vsep = –0.75V
- in some cases slightly higher Noise at Vsep = 0V, higher noise for Vsep<0.75V

- Laser Response:
  - affected by the lower gain

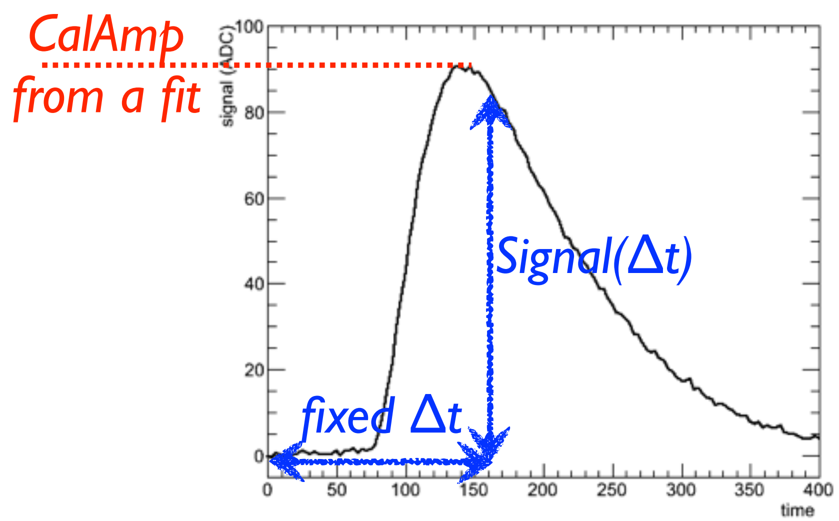
```
note: some pinholes may have a
 normal behaviour at Vsep = 0V
```

Gain Evaluation

Response Curve

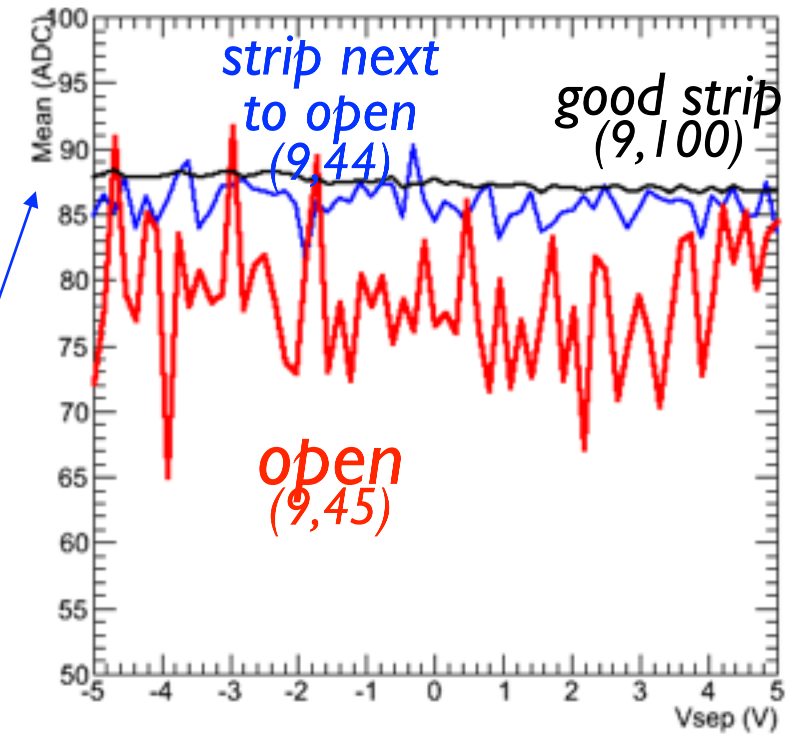


Response Curve



$$\text{Mean} = \frac{\Sigma_i^N Signal_i(\Delta t)}{N}$$

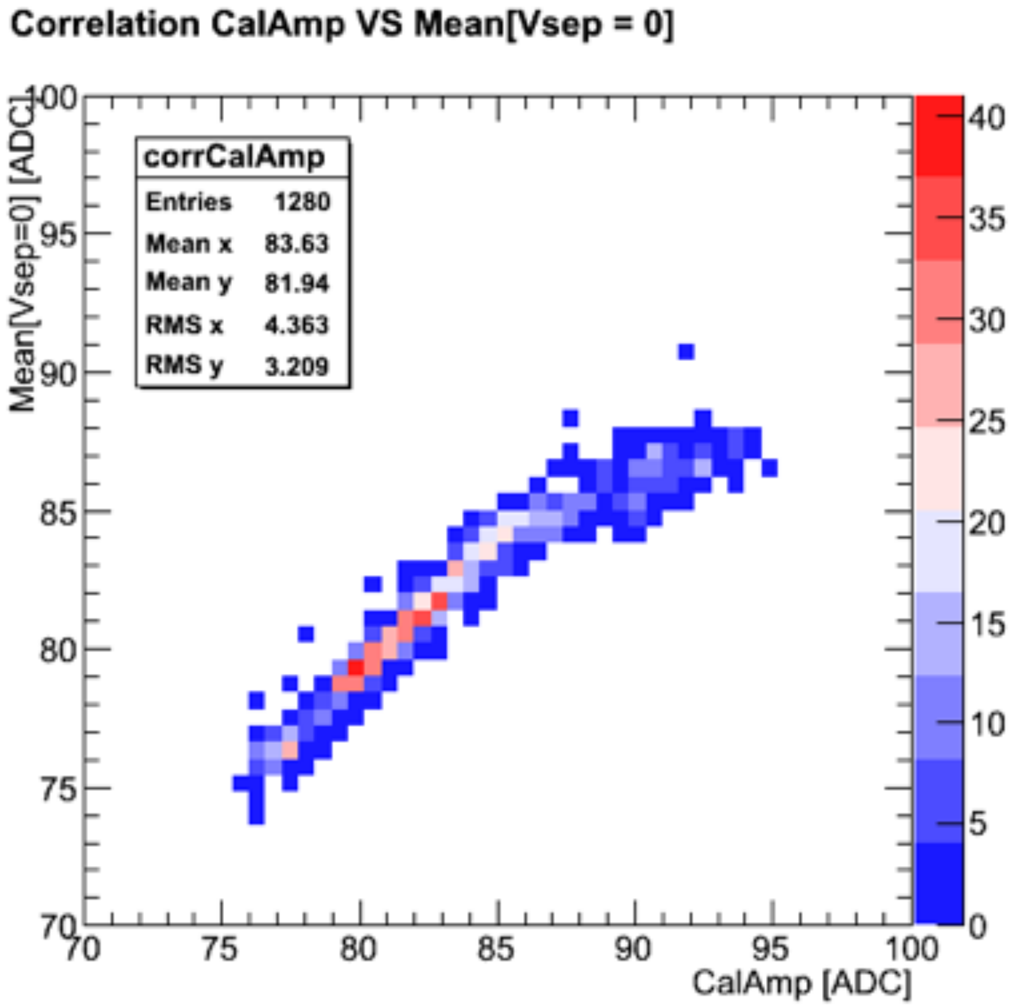
Mean VS Vsep



- For strips with very high noise, the CalAmp value is not always reliable since the fit to the response curve may converge to crazy values → the CalAmp value can not be used to classify the defect
- Let's use the average of Mean over the different Vsep to estimate CalAmp:

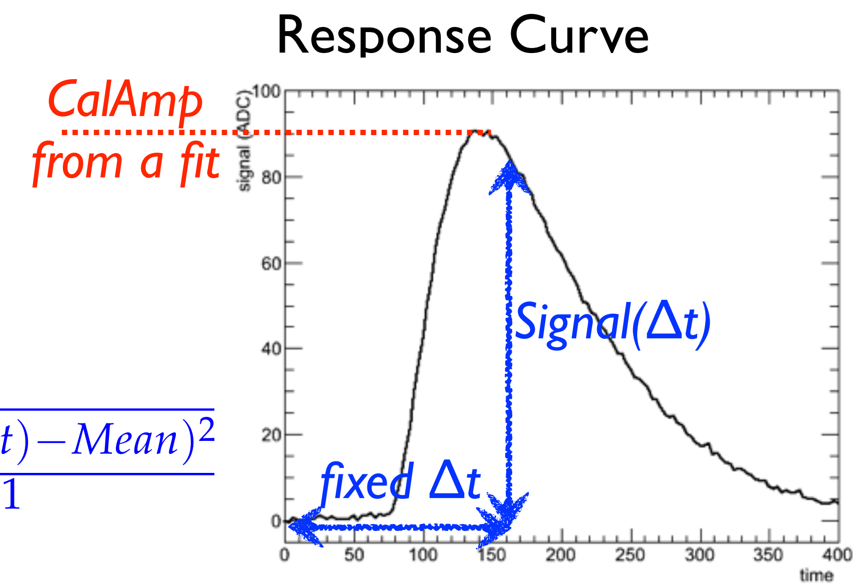
$$average(Mean) = \frac{\Sigma_{Vsep}^{N'} Mean(Vsep)}{N'}$$

Mean and RMS from Vsep Scan



$$\text{Mean} = \frac{\Sigma_i^N Signal_i(\Delta t)}{N}$$

$$\text{RMS} = \sqrt{\frac{\Sigma_i^N (Signal_i(\Delta t) - Mean)^2}{N-1}}$$



➡ The Mean[Vsep=0] is correlated with CalAmp

- Mean[Vsep=0] is a good estimate for CalAmp in case the fit to the response curve does not converge

➡ The RMS[Vsep=0] is correlated with the Noise