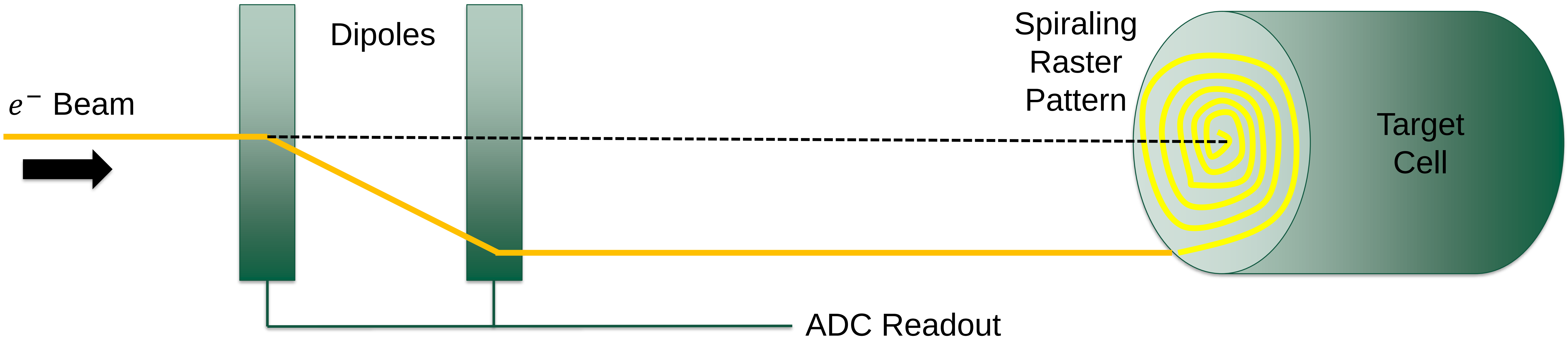
Raster Calibration Via Multi-Track Vertex Reconstruction

Derek Holmberg
3-12-24

WILLIAM & MARY
CHARTERED 1693

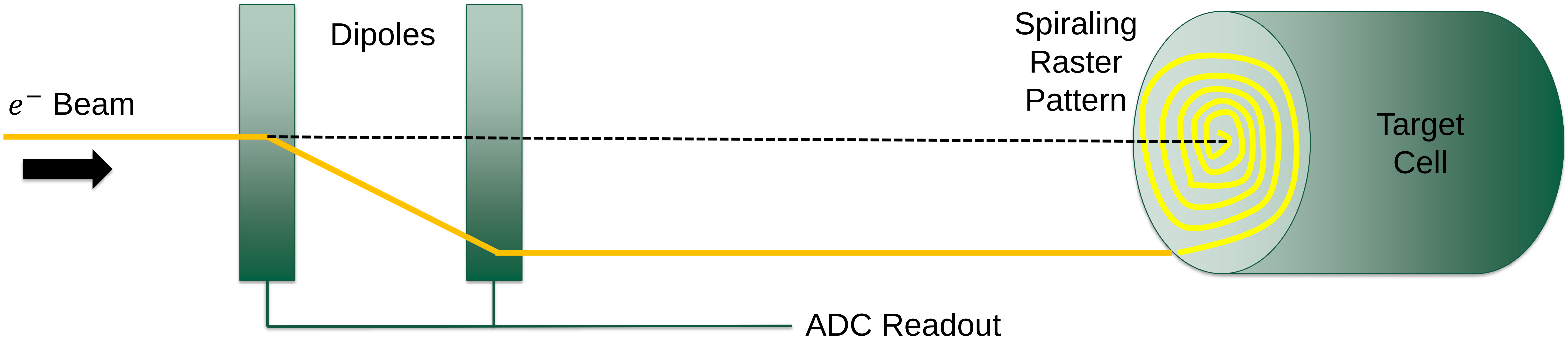
Raster Calibration (RG-C)

- **Raster System:** Two dipoles deflect the electron beam in a spiral pattern to evenly distribute radiation dose over the target
- **Calibration:** Find conversion factor from the dipole current ADC signal to beam position
- Use multi-track vertex reconstruction to relate ADC signal to reconstructed particle vertices (detached vertex tracking)



Raster Calibration (RG-C)

- **Detached Vertex Tracking**: Process by which two coincidental particles (e^- and π^+) are extrapolated back to a common scattering vertex
- Offline analysis: performed on runs that have already been cooked
- CLARA micro-service package developed by Veronique Ziegler

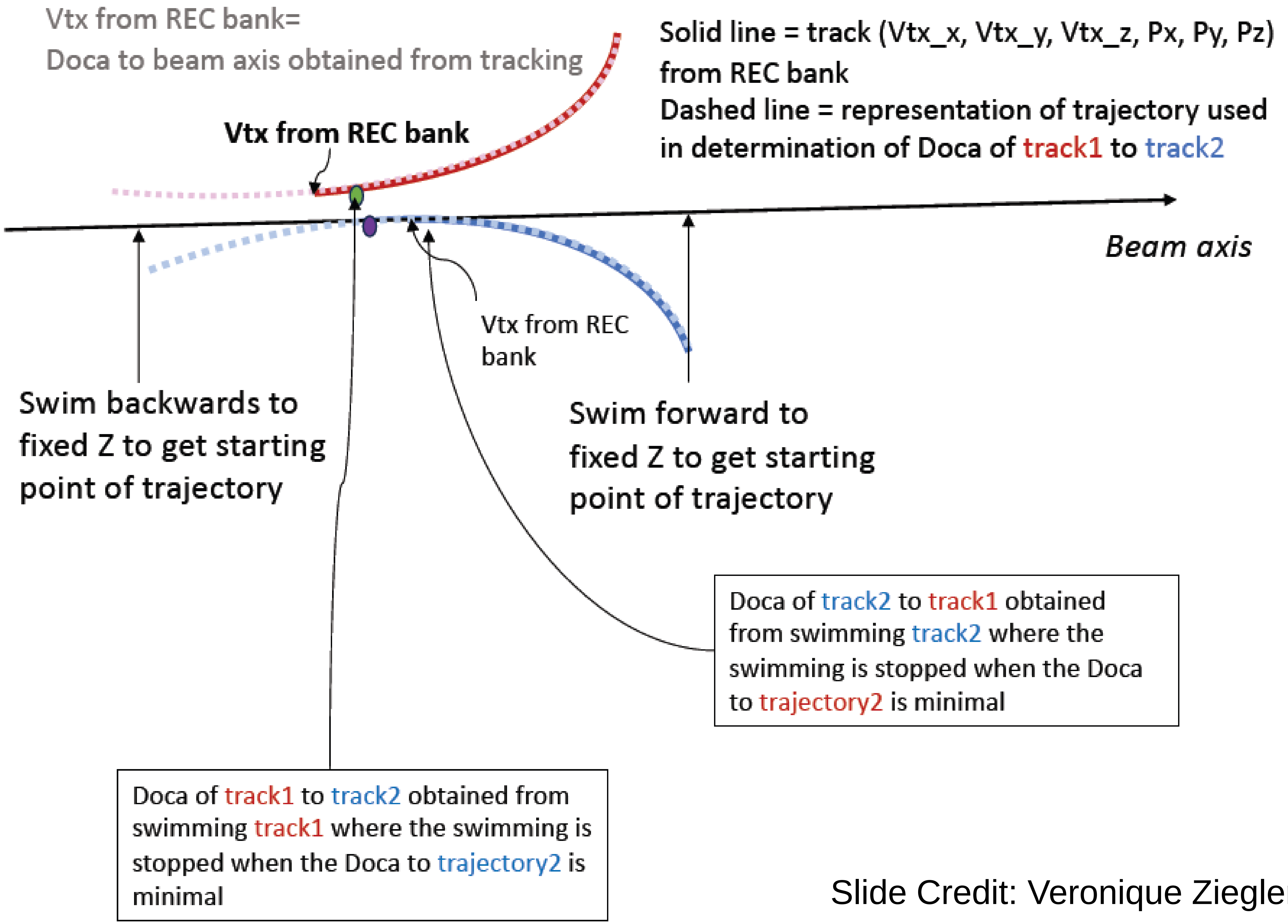


Detached Vertex Finder Algorithm

1. For each track
   1. Swim backwards to fixed Z to get starting point of trajectory
   2. Swim forward to fixed Z to get starting point of trajectory

2. Compute Doca of track2(1) to track1(2) obtained from swimming track2(1) where the swimming is stopped when the Doca to trajectory2(1) is minimal

3. Compute r as the distance between the so-obtained doca points of each track



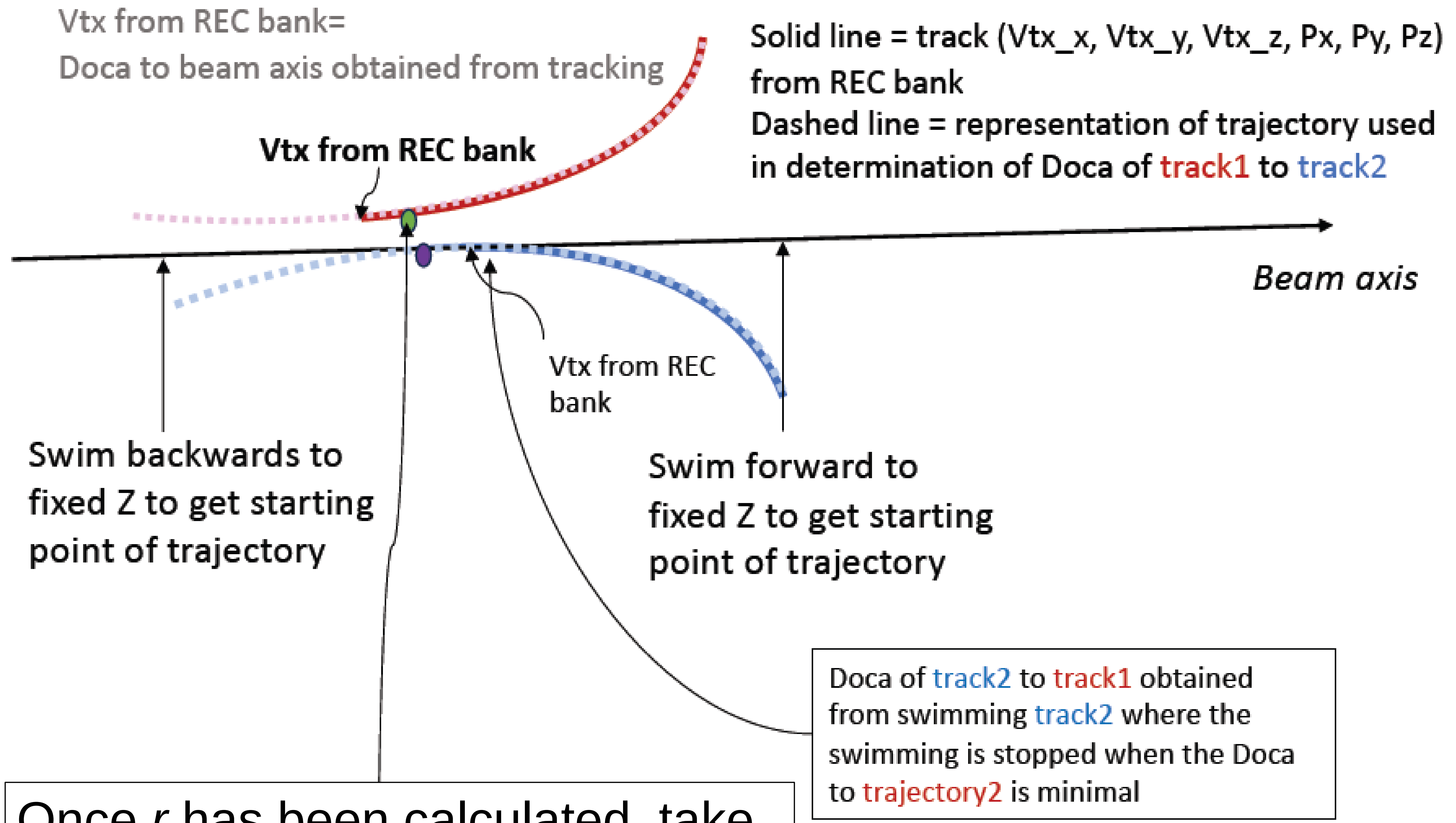
Slide Credit: Veronique Ziegler

Detached Vertex Finder Algorithm

1. For each track
   1. Swim backwards to fixed Z to get starting point of trajectory
   2. Swim forward to fixed Z to get starting point of trajectory

2. Compute Doca of track2(1) to track1(2) obtained from swimming track2(1) where the swimming is stopped when the Doca to trajectory2(1) is minimal

3. Compute r as the distance between the so-obtained doca points of each track

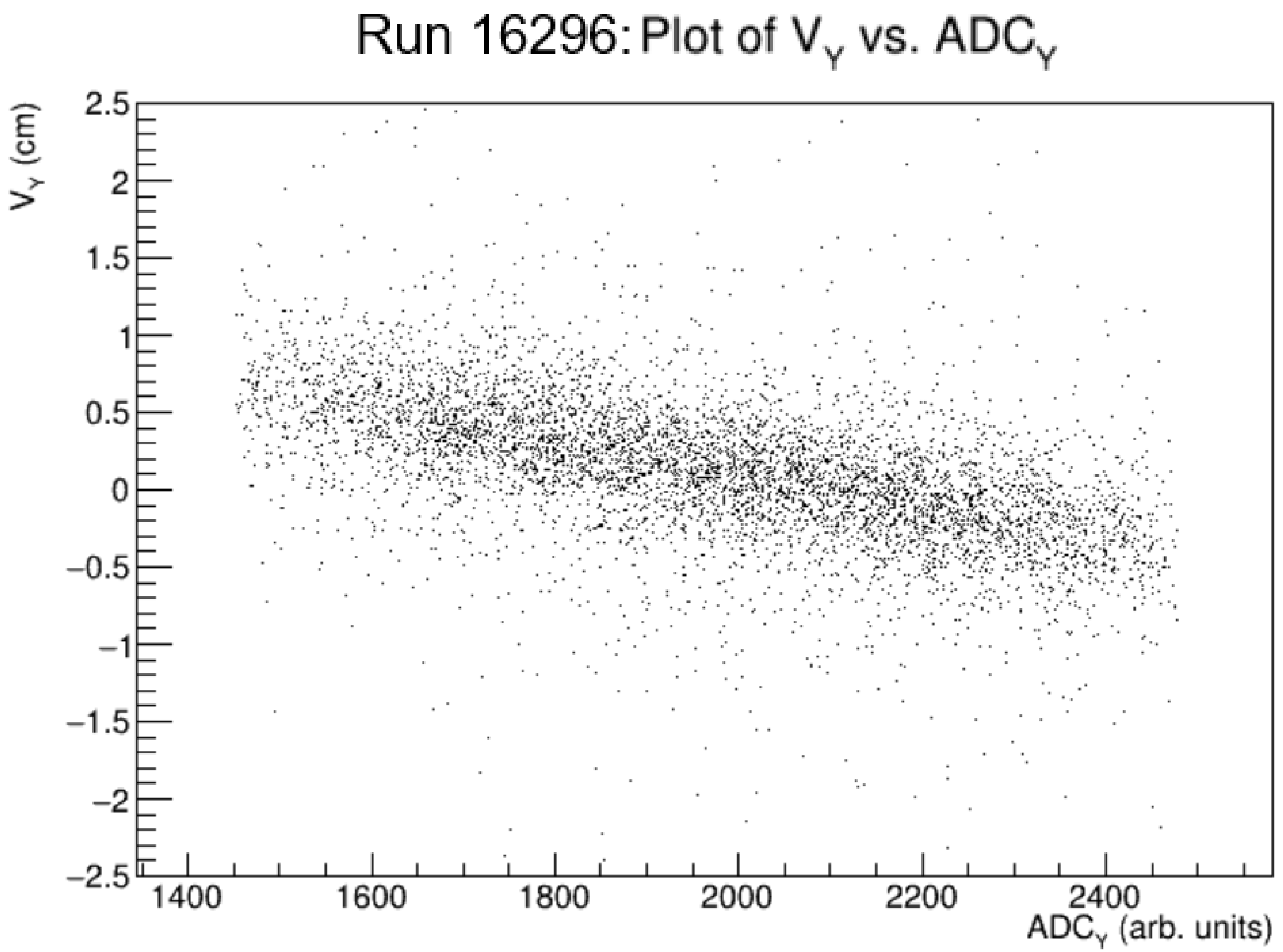


Once *r* has been calculated, take the average of the detached vertices as the new scattering vertex used in analysis

Slide Credit: Veronique Ziegler

Extracting Calibration Constants

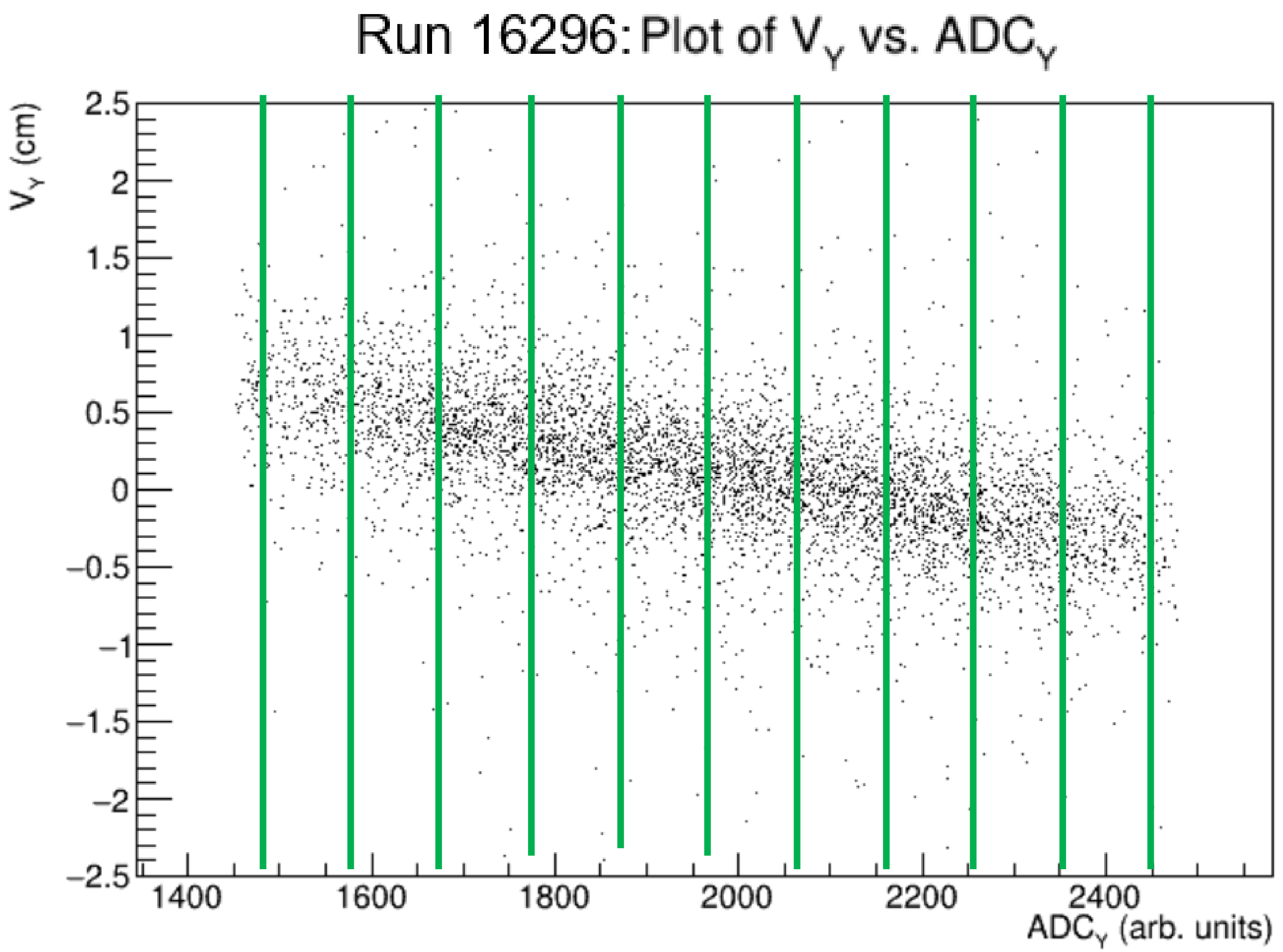
- **Calibration Constants:**
  - Plot extracted x, y vertices vs. $ADC_{x,y}$ signals and fit data directly to a straight line
  - $V_{x,y} = P_0 + P_1 * ADC_{x,y}$
  - Converts raster signal (ADC) to vertex position
  - Two constants each for x, y (four total for each data set)



Extracting Calibration Constants

- **Calibration Constants:**
  - Plot extracted x, y vertices vs. $ADC_{x,y}$ signals and fit data directly to a straight line
  - $V_{x,y} = P_0 + P_1 * ADC_{x,y}$
  - Converts raster signal (ADC) to vertex position
  - Two constants each for x, y (four total for each data set)

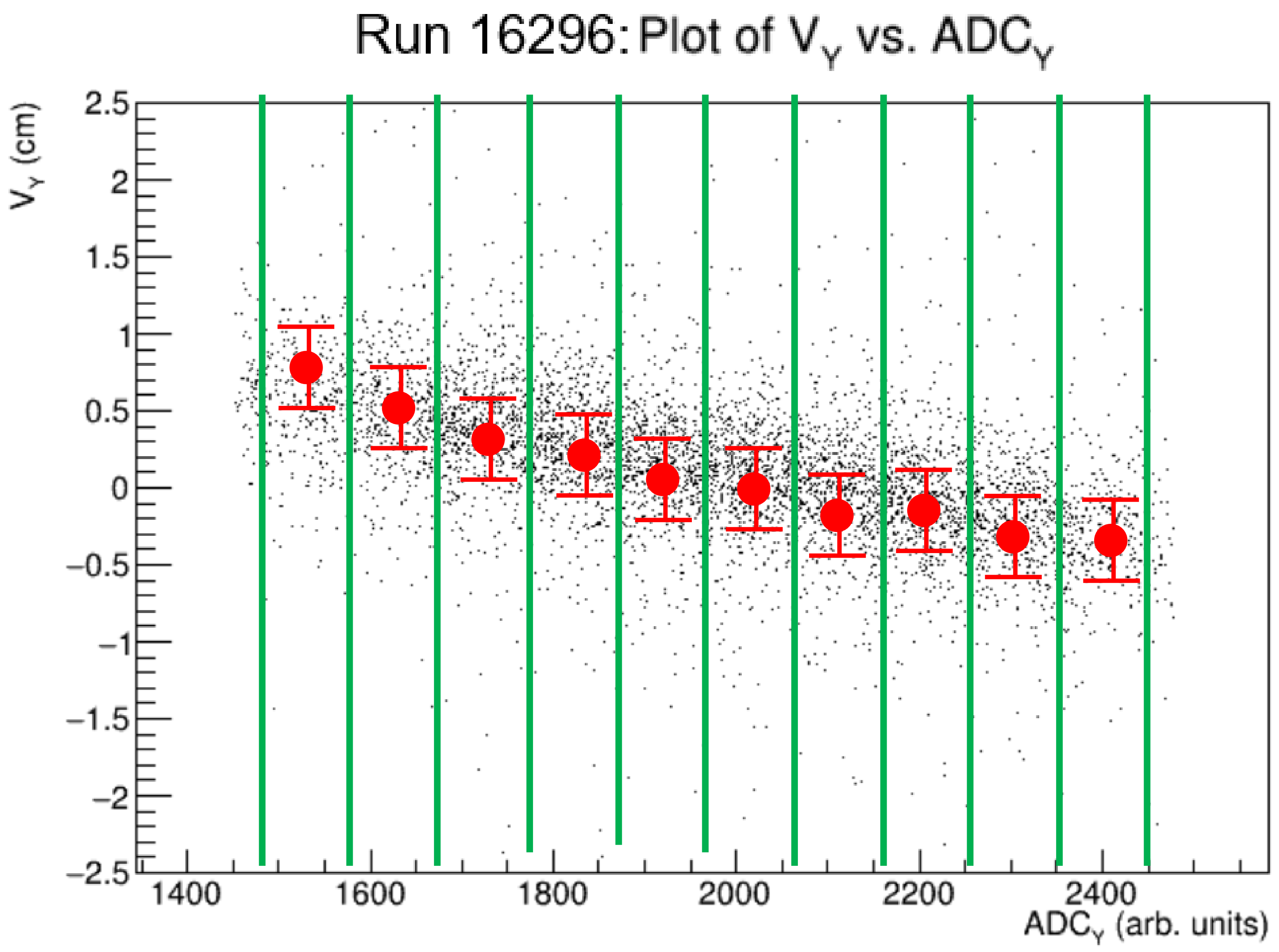
- **Profile fit:**
  - Break data set up into 10 separate ADC bins (between green lines) to account for non-gaussian distributions
  - Fit each of the bins to a double gaussian to find average vertex for the ADC signal bin
  - Fit results using $V_{x,y} = P_0 + P_1 * ADC_{x,y}$



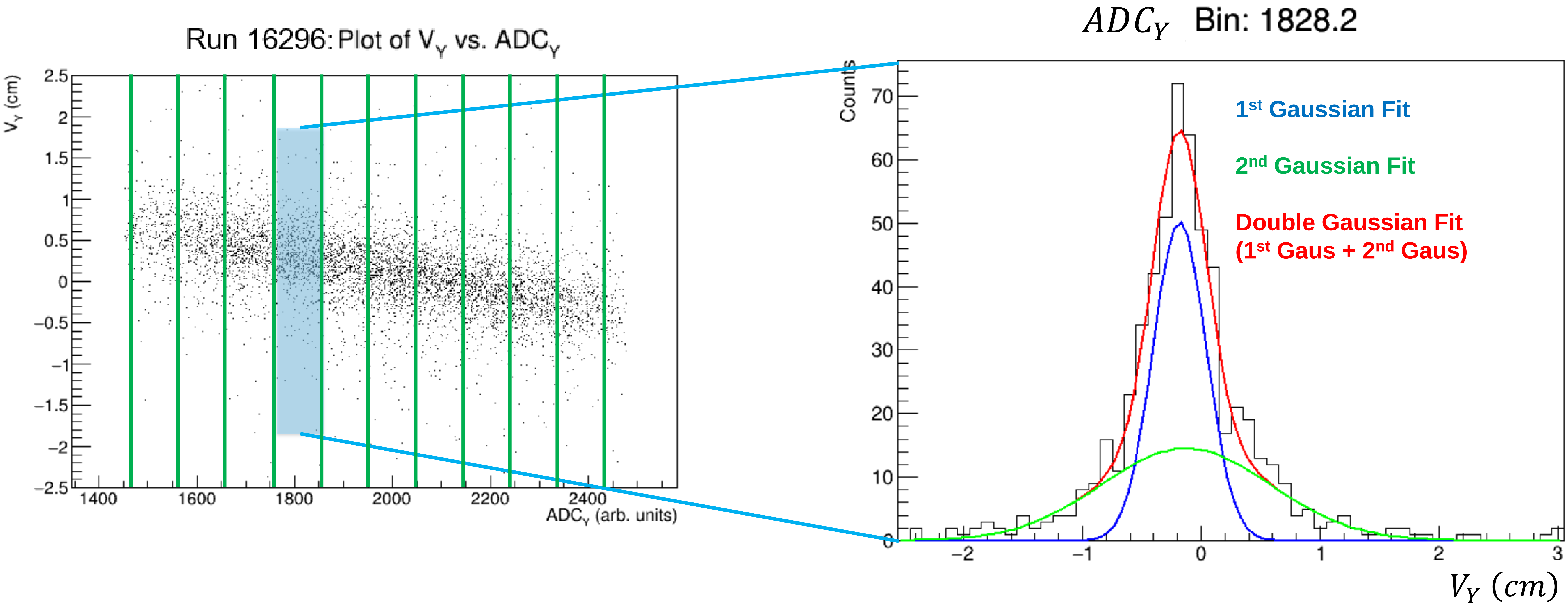
Extracting Calibration Constants

- **Calibration Constants:**
  - Plot extracted x, y vertices vs. $ADC_{x,y}$ signals and fit data directly to a straight line
  - $V_{x,y} = P_0 + P_1 * ADC_{x,y}$
  - Converts raster signal (ADC) to vertex position
  - Two constants each for x, y (four total for each data set)

- **Profile fit:**
  - Break data set up into 10 separate ADC bins (between green lines) to account for non-gaussian distributions
  - Fit each of the bins to a double gaussian to find average vertex for the ADC signal bin
  - Fit results using $V_{x,y} = P_0 + P_1 * ADC_{x,y}$



Extracting Calibration Constants



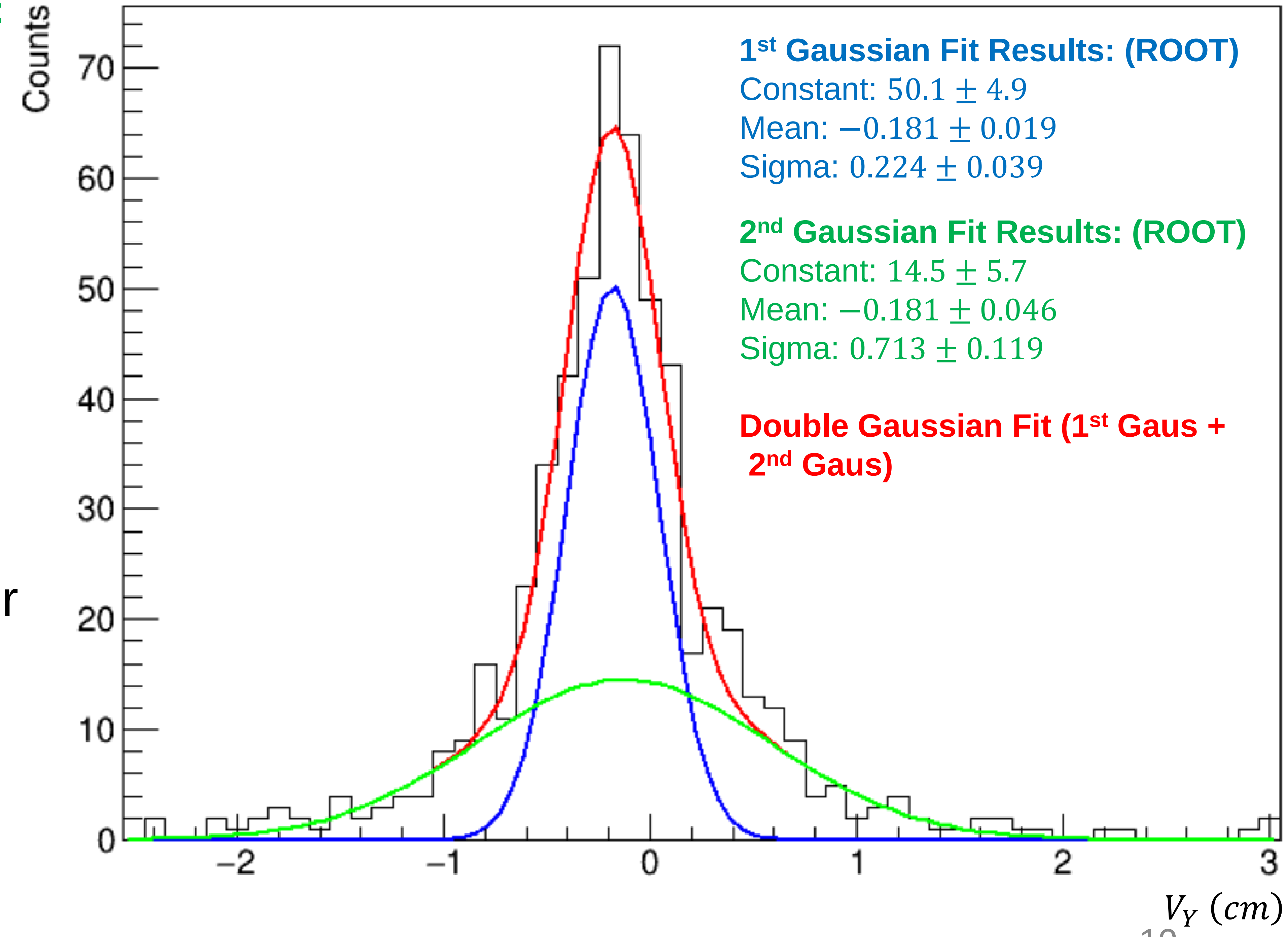
Extracting Calibration Constants

- Fit data with double gaussian
  - $G(V_i) = A_1 * \exp\left(\frac{\mu - V_i}{\sigma_1}\right)^2 + A_2 * \exp\left(\frac{\mu - V_i}{\sigma_2}\right)^2$
- Fitting Procedure:
  1. Fit distribution to a single gaussian
  2. Use parameters from that fit to initialize double gaussian
  3. Fit again with double gaussian

- Plot $\mu\ (cm)\ vs.\ ADC$ signal
- Bin ADC signal taken as average ADC value for the bin
- Error on mean is $\delta\mu = \sigma_1/\sqrt{N}$, N= number of counts in the histogram

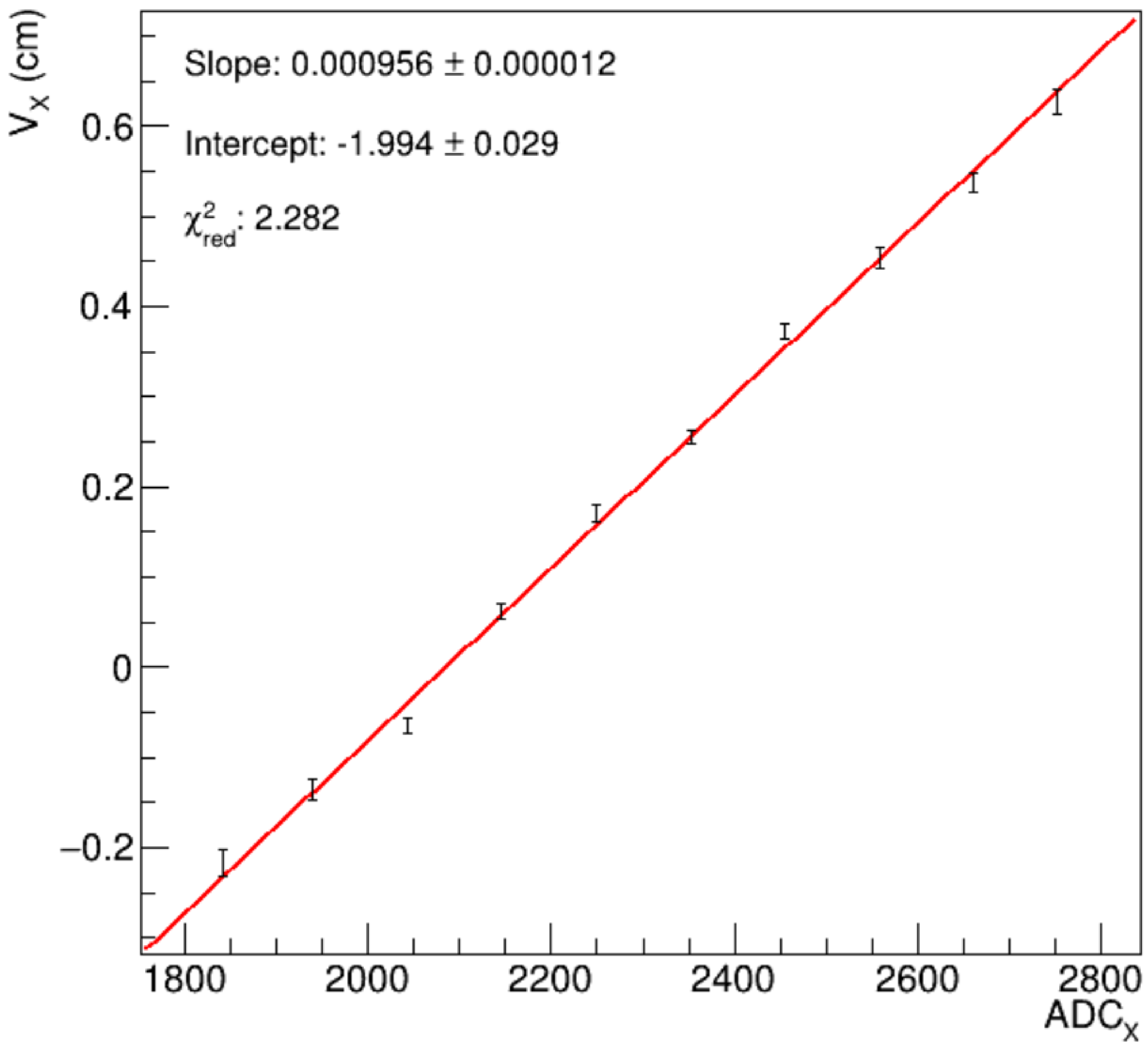
μ = reconstructed vertex for fixed ADC signal

ADC_Y Bin: 1828.2

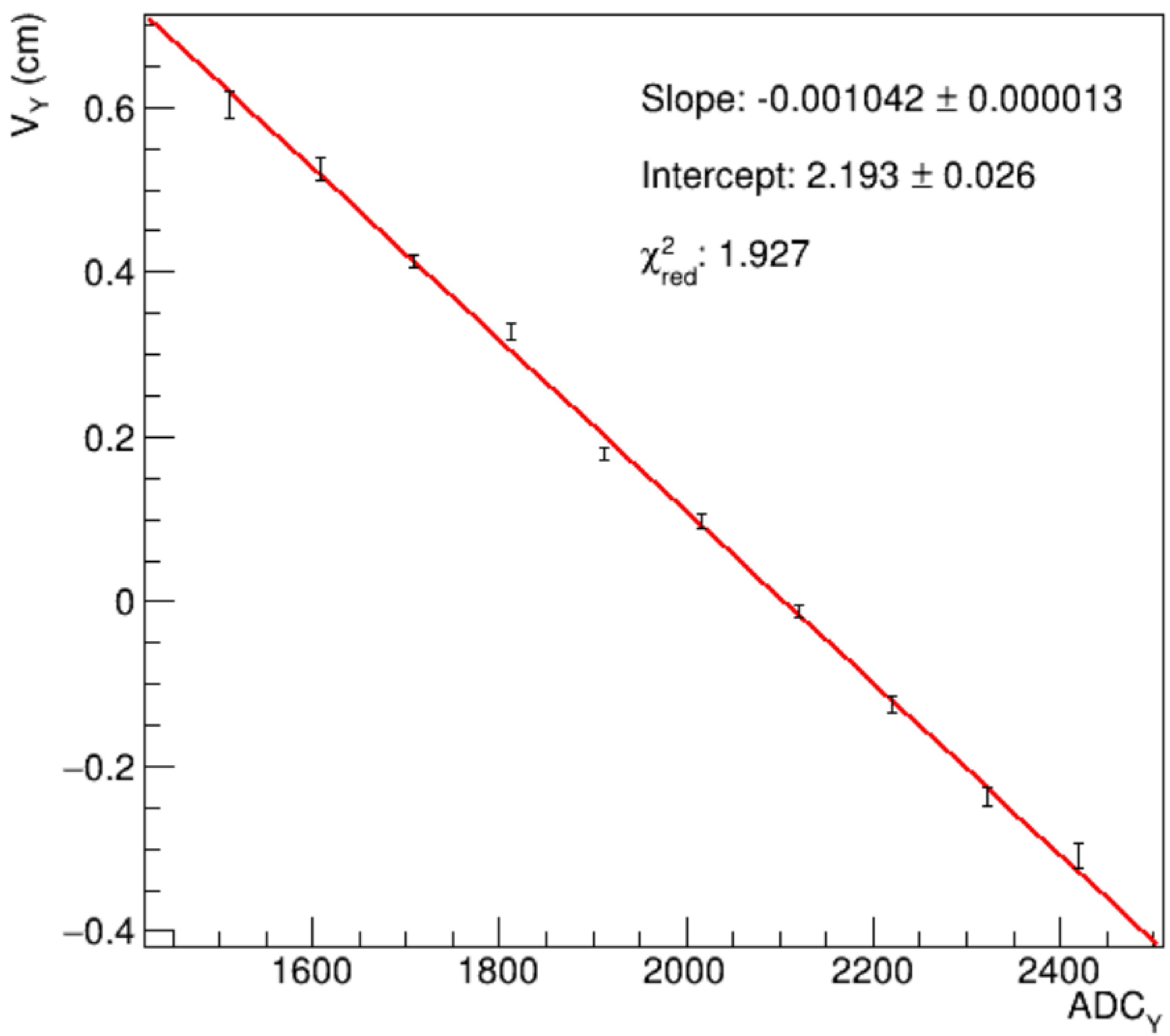


Extracting Calibration Constants

Run 16296: Plot of V_X vs. ADC_X



Run 16296: Plot of V_Y vs. ADC_Y



Extracting Calibration Constants

- Cuts to the Data:
  - e^- seen in DC, π^+ seen in CVT
  - $\chi^2_{pid} < 3$ for e^-, π^+
  - $-8\ cm < V_z < 2\ cm,\ |V_{x,y}| < 2.05\ cm$ for all data points
  - Separation distance $0\ cm < \Delta\mathrm{R} < 1.0\ cm$

- All the data came from the first 20 HIPO files of the runs, and no additional kinematic cuts were made (i.e. no SIDIS skims, DVCS skims, etc.)

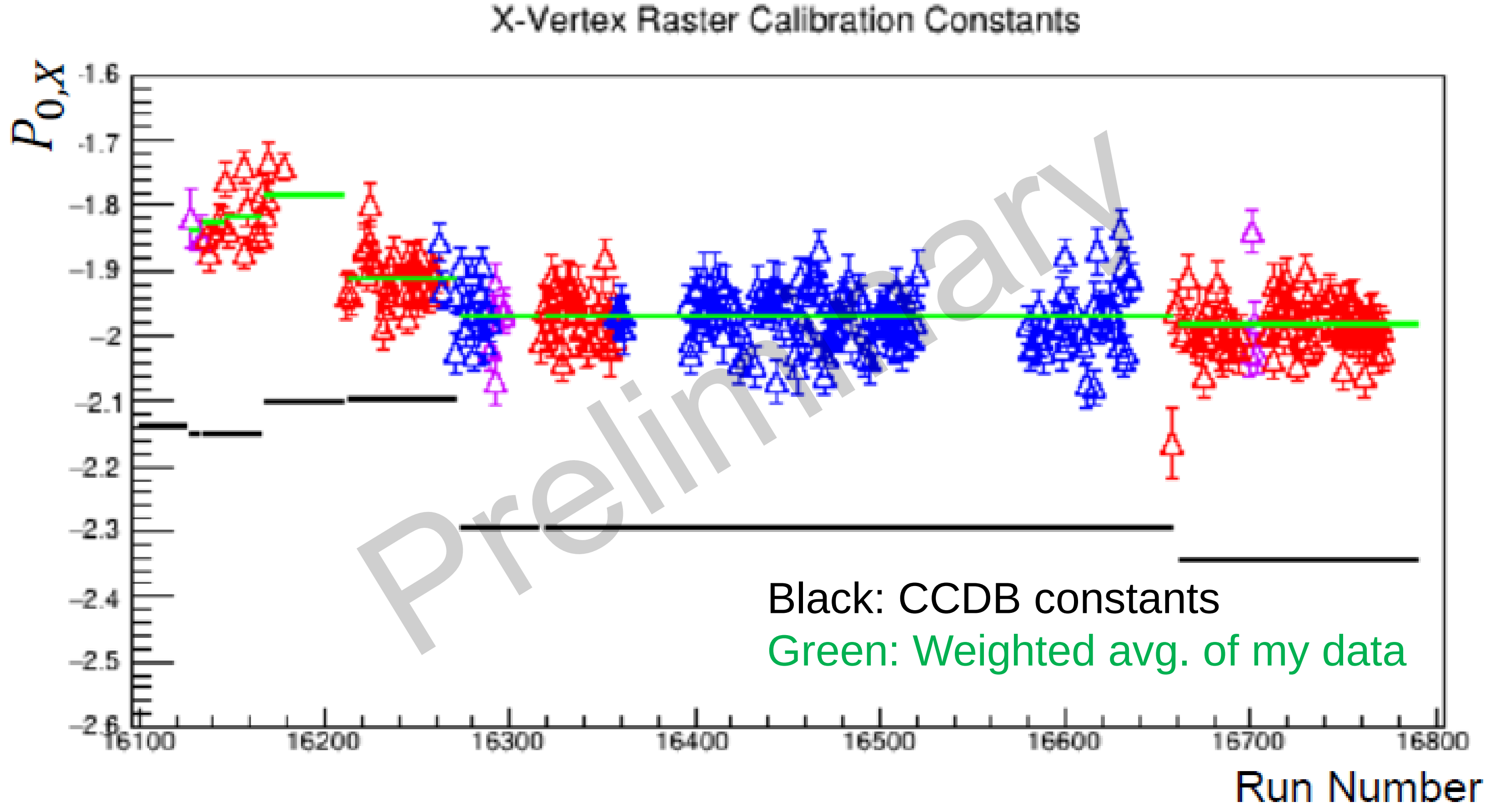
Running the Analysis

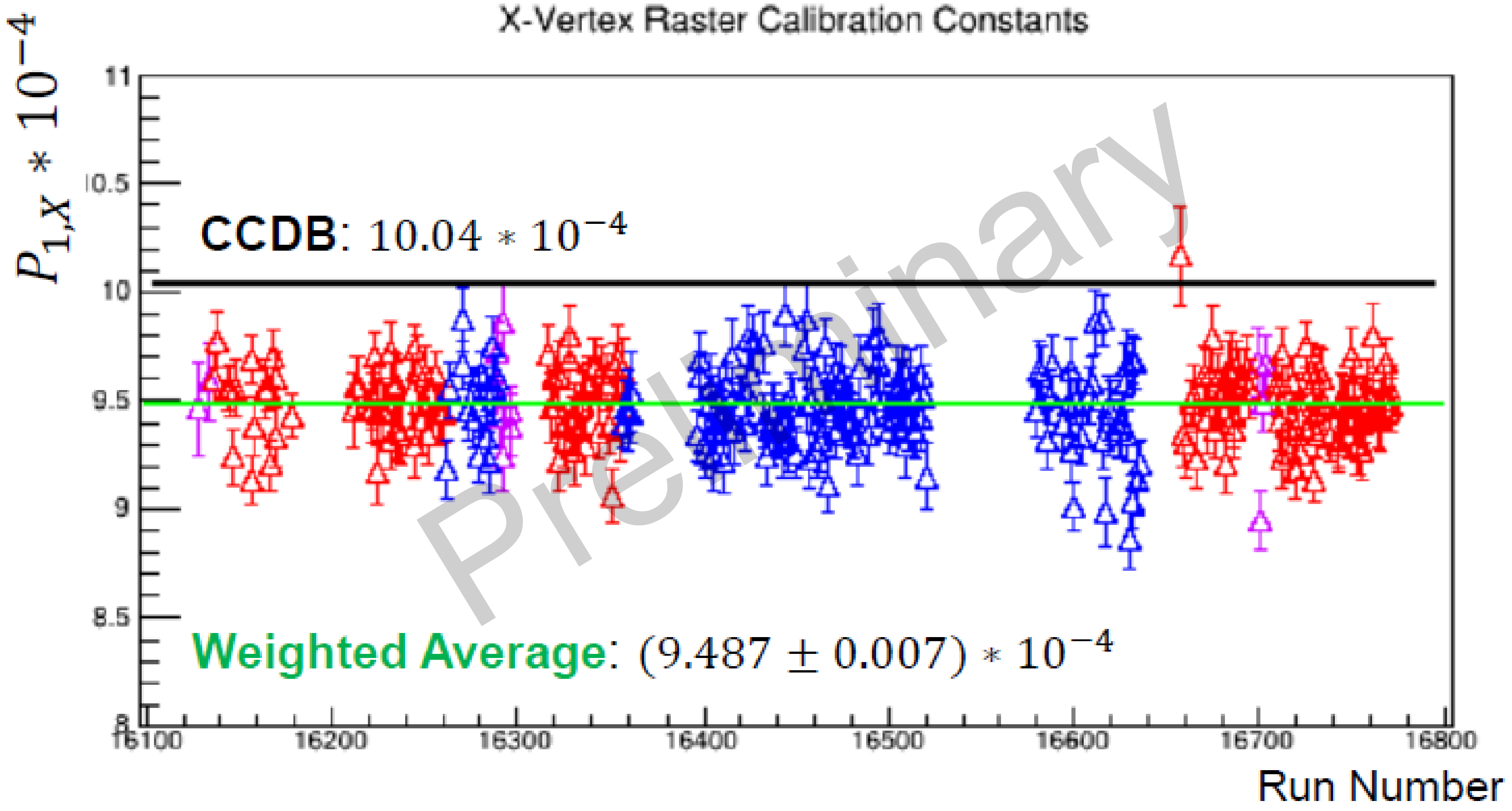
- Analyzed all runs over the entire Summer '22 run period (inbending electrons)
  - Using roughly 330 runs from across the run period
  - Calibration, empty target, luminosity scans skipped
  - /volatile/clas12/rg-c/production/pass0/mon/v0.17/mon/recon

- Runs ranging from 16128 to 16772
  - Runs 16292 – 16297 from /v0.15/ directory
  - Carbon, NH3, and ND3 runs were used

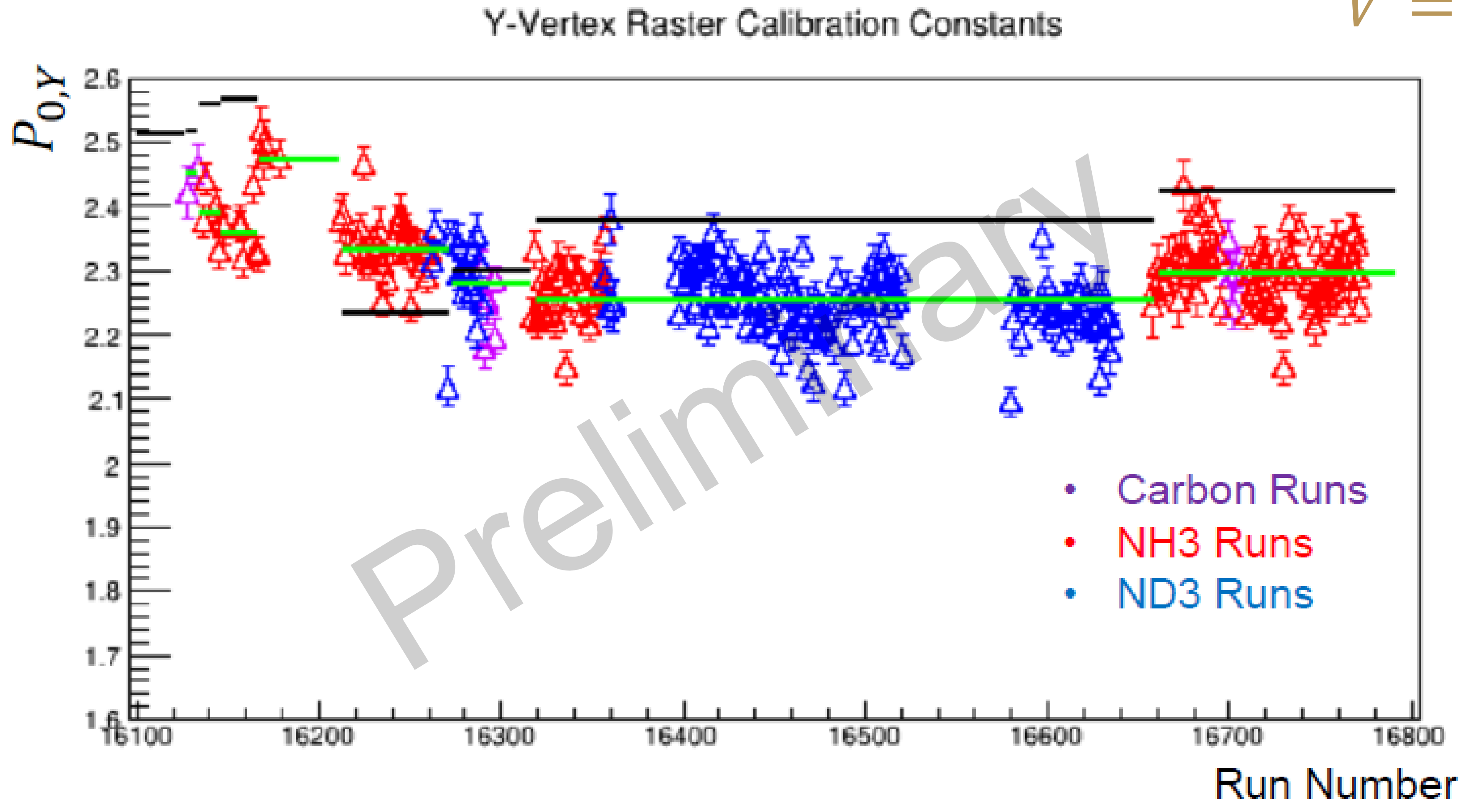
- Used events where e^- is in the forward detector, π^+ is in the central detector
  - $\pi^+\pi^-$ cases had too few events (<1000 per run) to have useful results

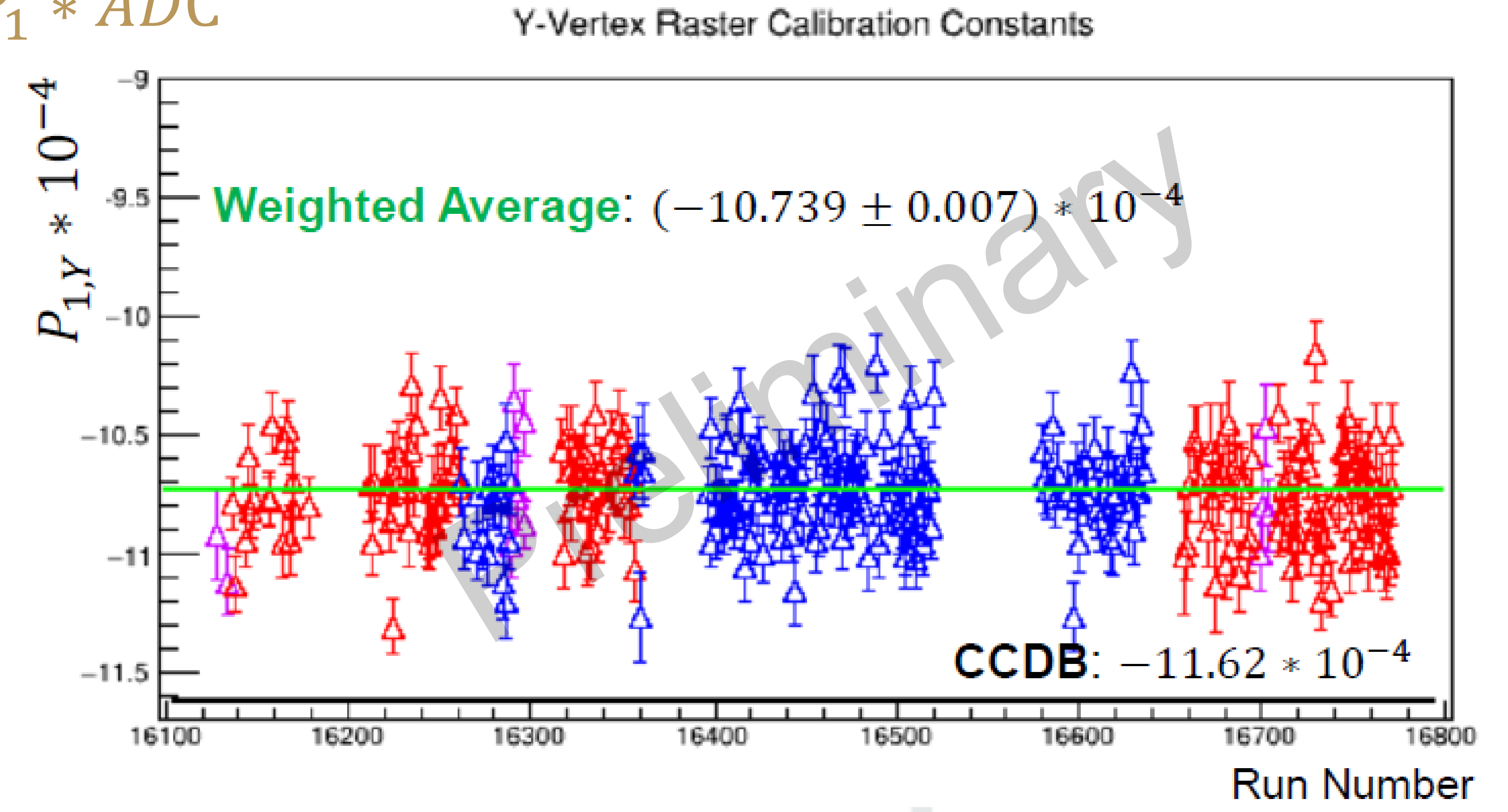
- Used analysis procedure as outlined, fitting to $V_{x,y} = P_0 + P_1 * ADC_{x,y}$


X-Vertex Raster Calibration Constants
$P_{0,X}$
Run Number
Black: CCDB constants
Green: Weighted avg. of my data
Preliminary


X-Vertex Raster Calibration Constants
$P_{1,X} * 10^{-4}$
Run Number
CCDB: $10.04 * 10^{-4}$
Weighted Average: $(9.487 \pm 0.007) * 10^{-4}$
Preliminary

$$V = P_0 + P_1 * ADC$$


Y-Vertex Raster Calibration Constants
$P_{0,Y}$
Run Number
Carbon Runs
NH3 Runs
ND3 Runs
Preliminary


Y-Vertex Raster Calibration Constants
$P_{1,Y} * 10^{-4}$
Run Number
Weighted Average: $(-10.739 \pm 0.007) * 10^{-4}$
CCDB: $-11.62 * 10^{-4}$
Preliminary

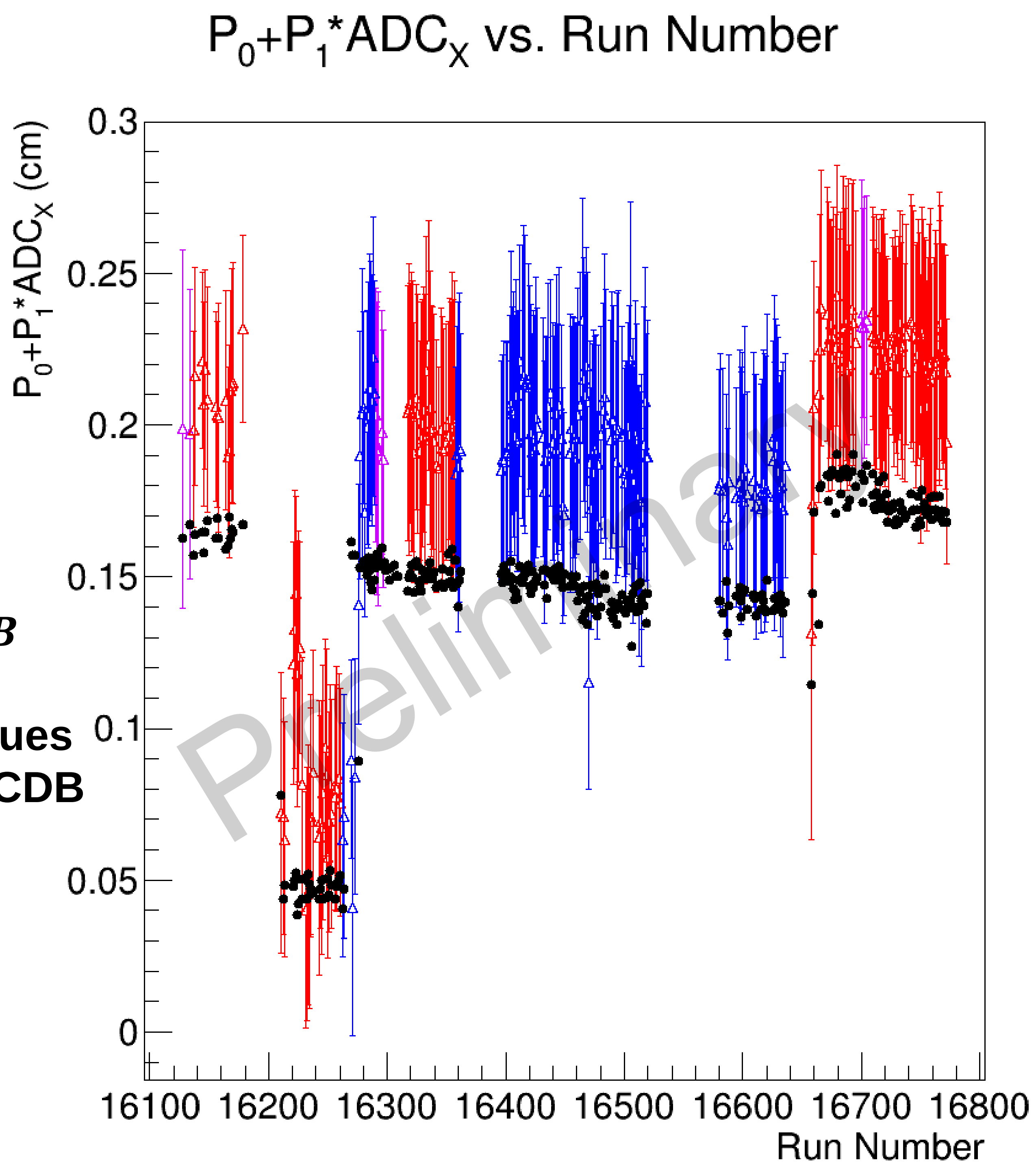
Calculating Average Beam Position

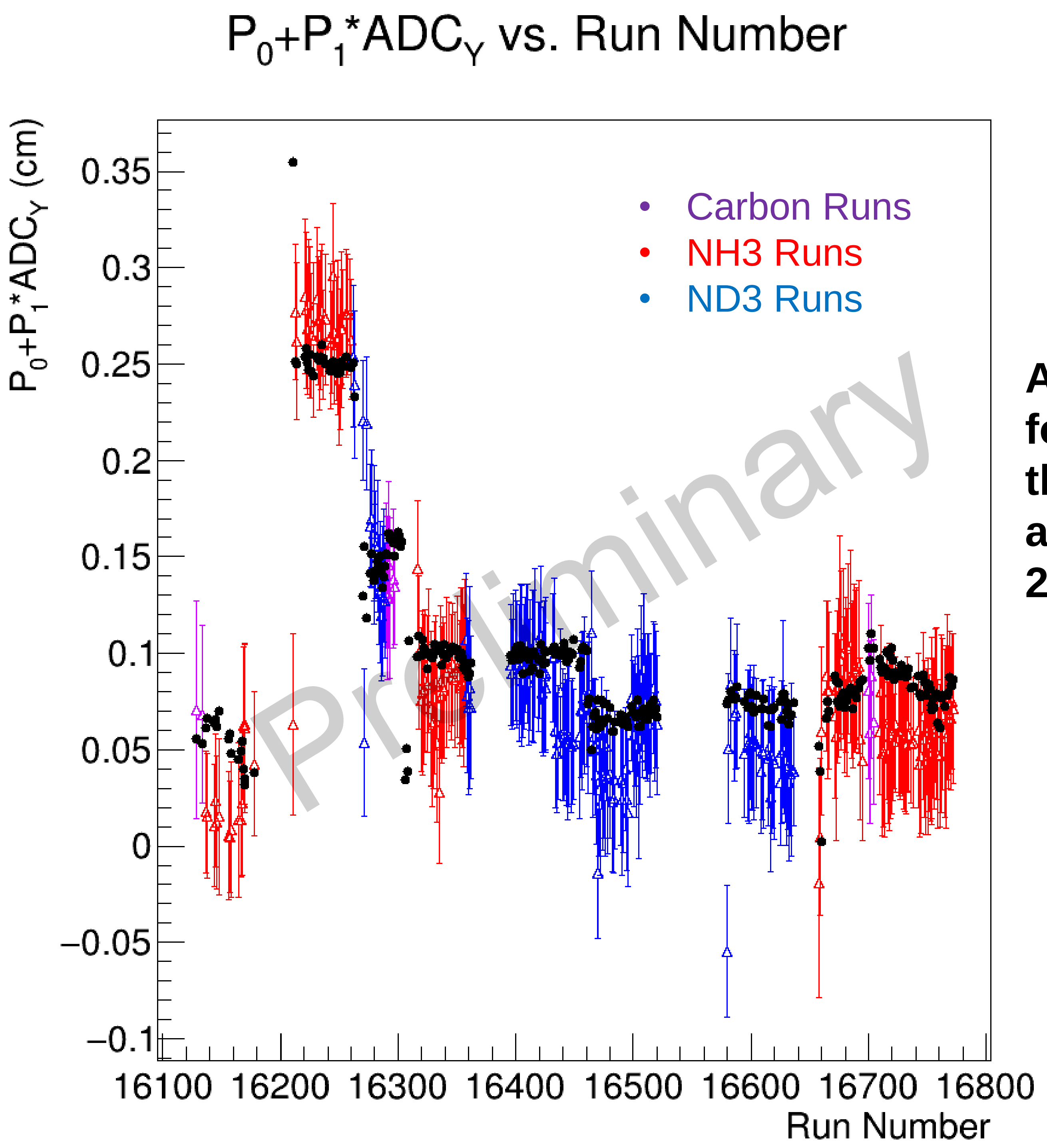
- Average beam position is the average position of the beam over the entire run
- For each of the runs, I calculated the average value of all ADC signals measured
  $ADC_{run} = \sum_i ADC_i / N, \;\; N =$ total counts over run
- Plotted $P_0 + P_1 * ADC_{run}$ for X,Y using my raster calibration constants
- Also plotted against $P_0 + P_1 * ADC_{run} + B$ using the values currently in CCDB, where B is the beam offset

$P_0 + P_1 * ADC$ (Average Beam Position)

$P_0 + P_1 * ADC + B$

(Using CCDB values where B is the CCDB beam offset)





ADC values used for the plots are the average value across the first 20 HIPO files

Work Still To Be Done

- Recooked multiple runs using my raster calibration constants; carbon foil runs 16296, 16297 and empty target run 16194
- Plot the reconstructed vertex distributions to check for any systematic effects of the calibration

- My Raster Calibration Values:
  - $P_{0,X} = -1.984, \;\; P_{1,X} = 0.0009487; \;\; P_{0,Y} = 2.299, \;\; P_{1,Y} = -0.0010739$
  - Beam Offset: All Zero
- Current Raster CCDB Values:
  - $P_{0,X} = -2.29552 \;\; P_{1,X} = 0.001004; \;\; P_{0,Y} = 2.29812, \;\; P_{1,Y} = -0.001162$
  - Beam Offset: $X_{offset} = 0.1544833; \;\; Y_{offset} = 0.150292$

- Still a work in progress; checking slight discrepancies in vertex distributions

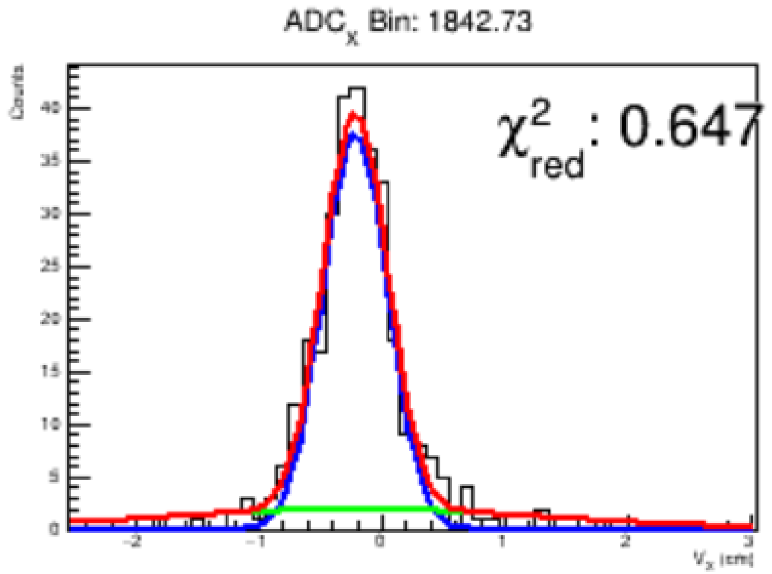
Any Questions?

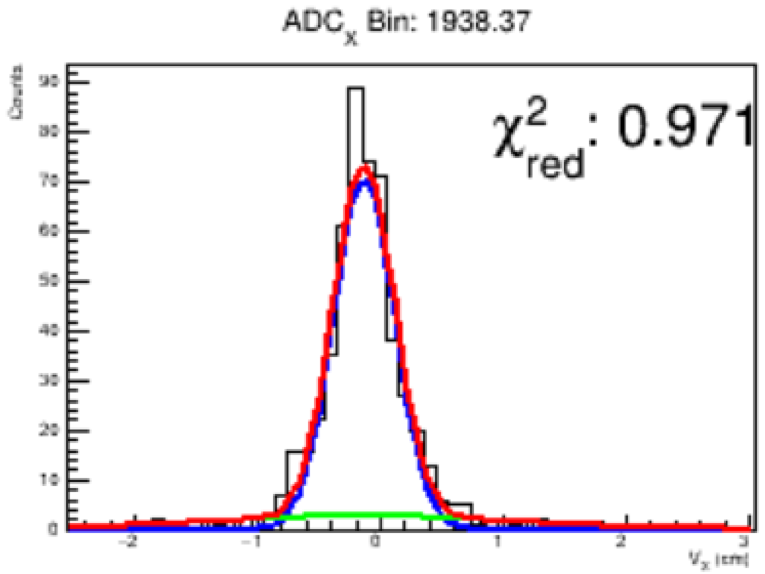
Backup Slides

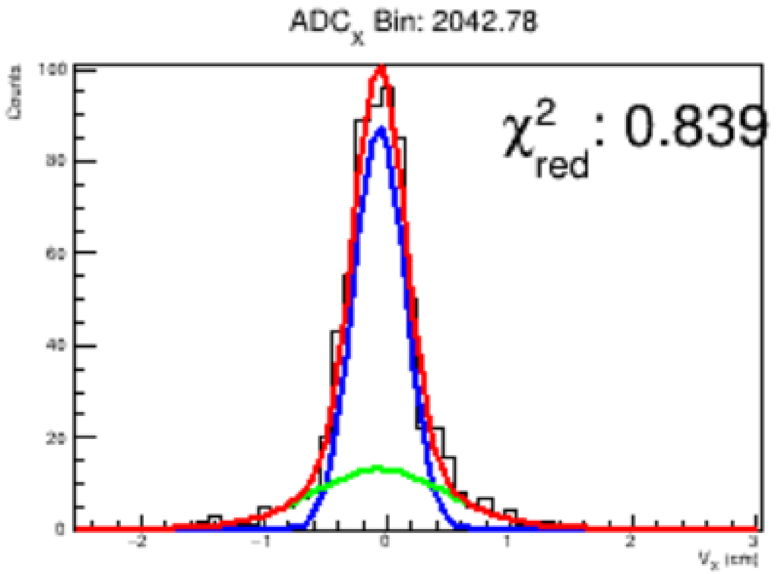
- What do the histograms look like?
- Effects of binning on raster calibration constants
- Changes in reconstructed vertices using my constants versus what's in CCDB

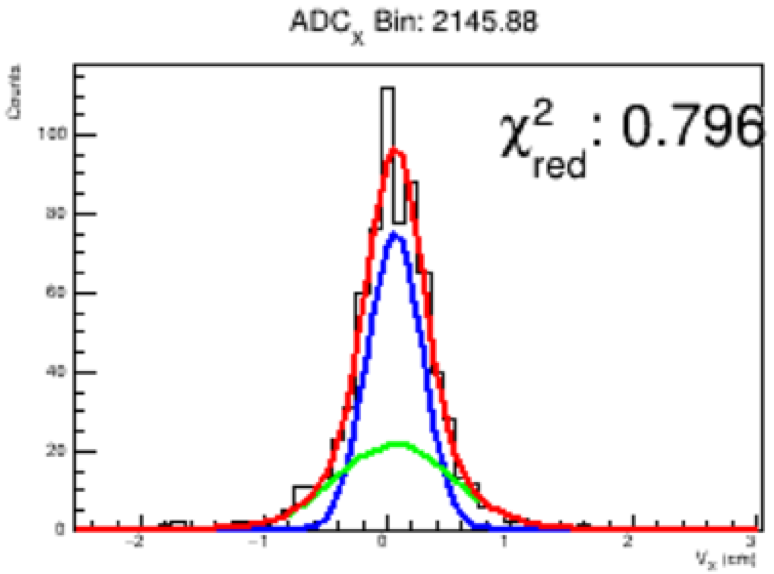
Backup Slides

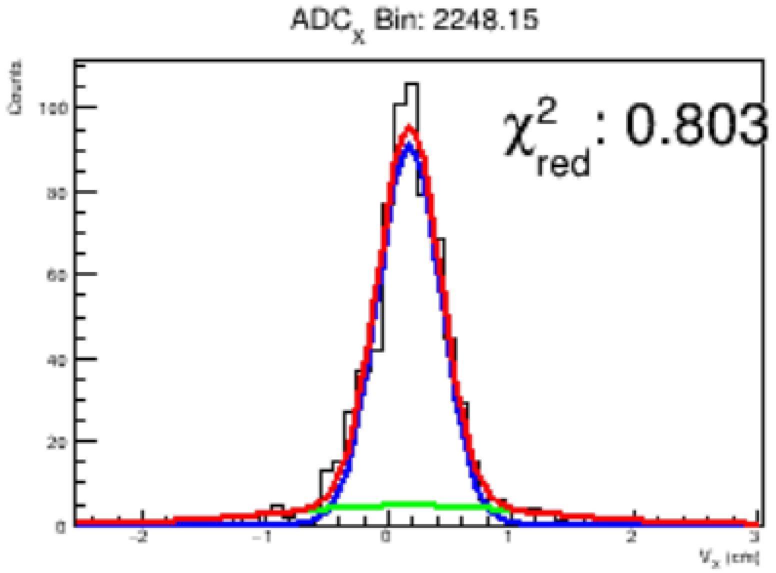
- What do the histograms look like?
- Effects of binning on raster calibration constants
- Changes in reconstructed vertices using my constants versus what's in CCDB

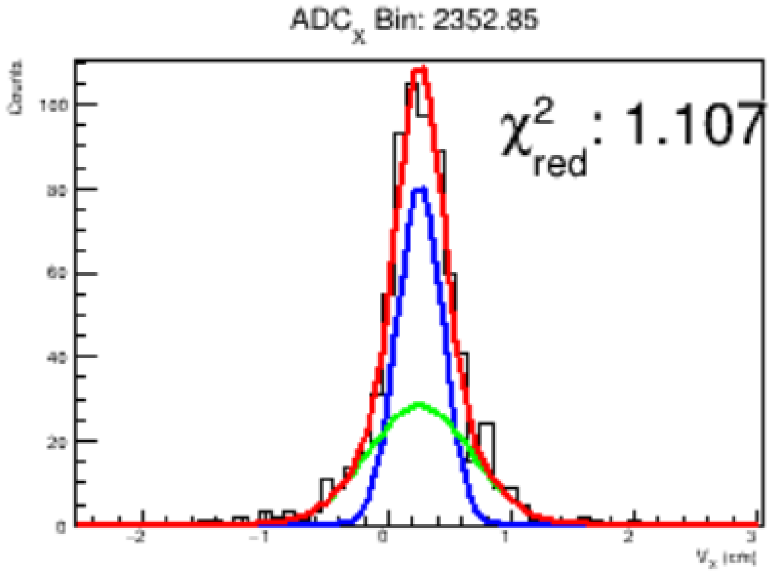


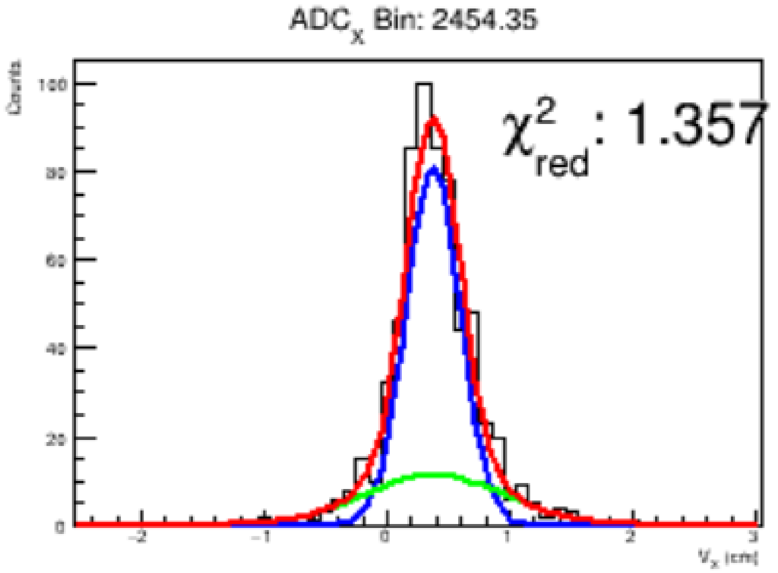


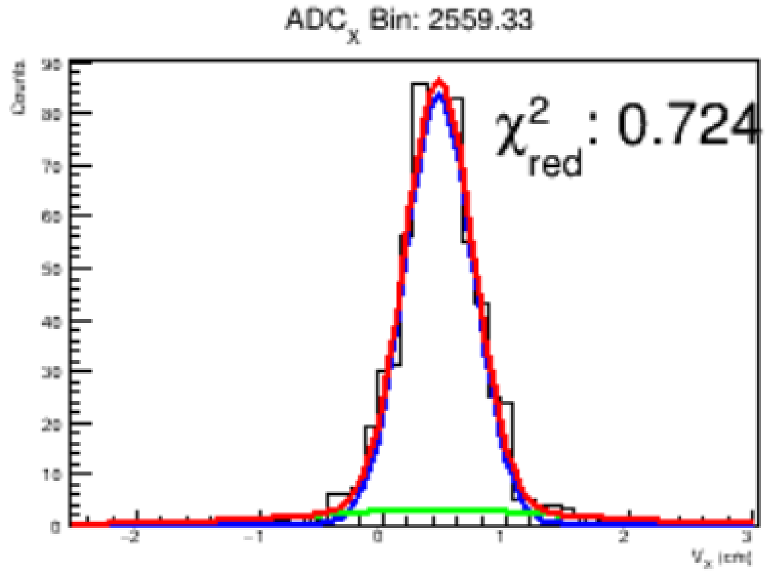


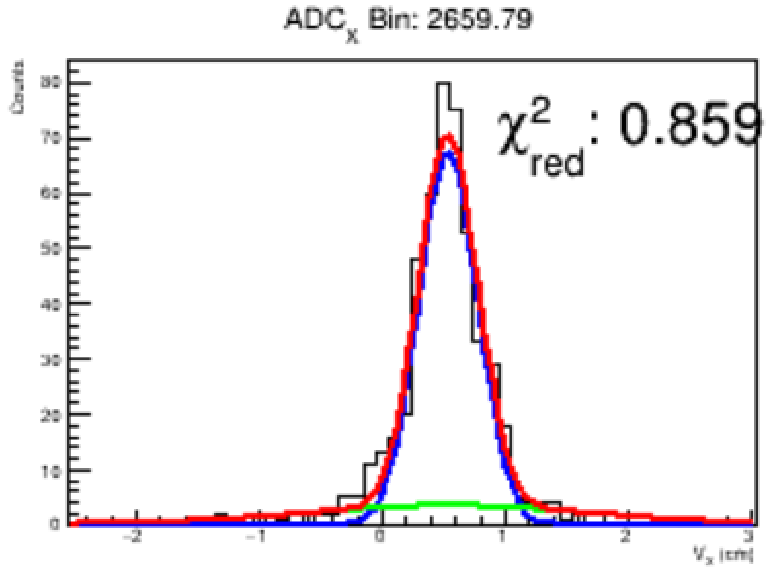


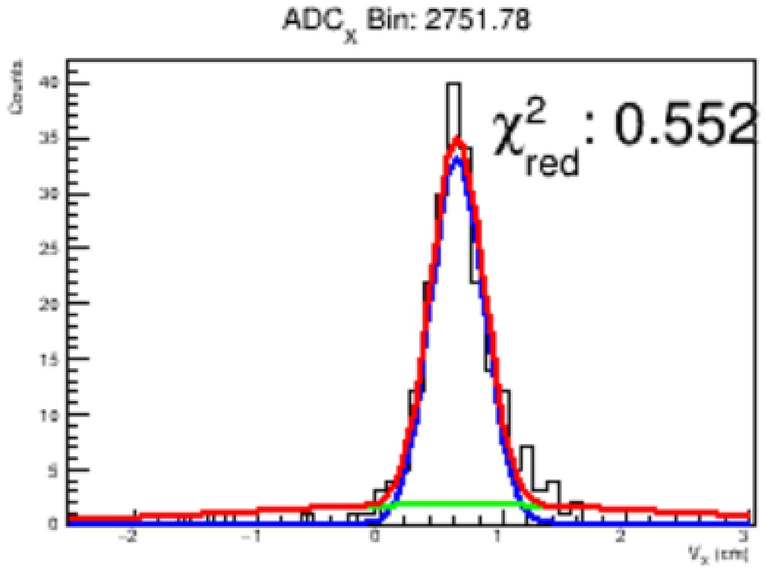




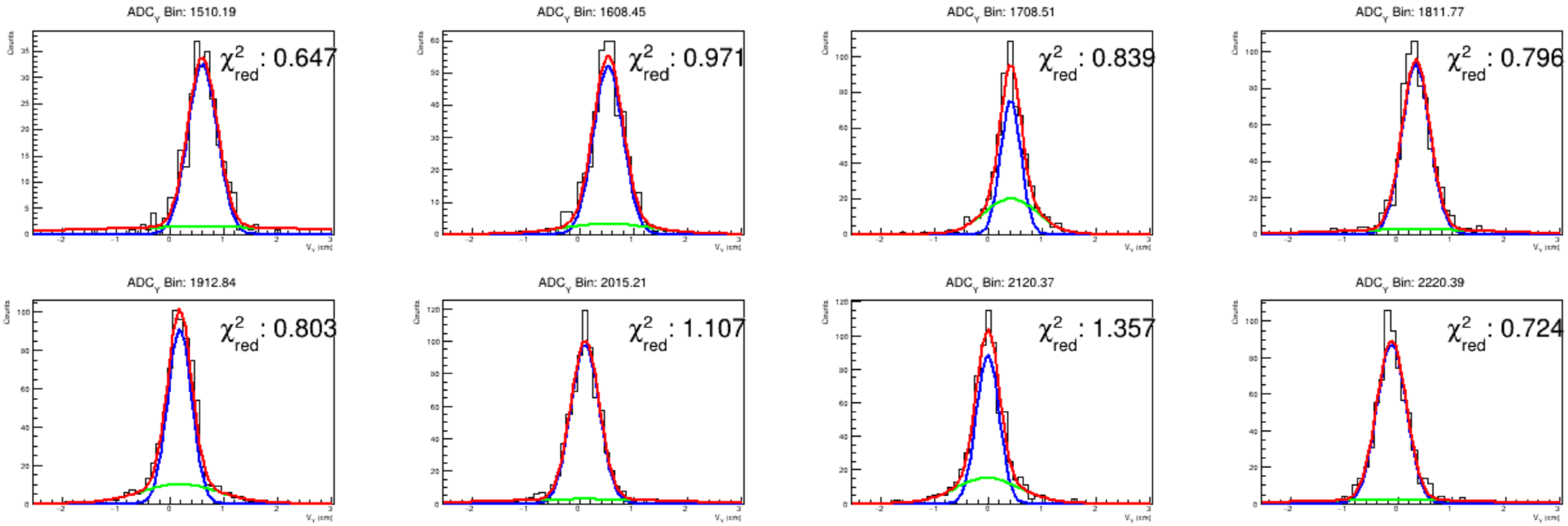


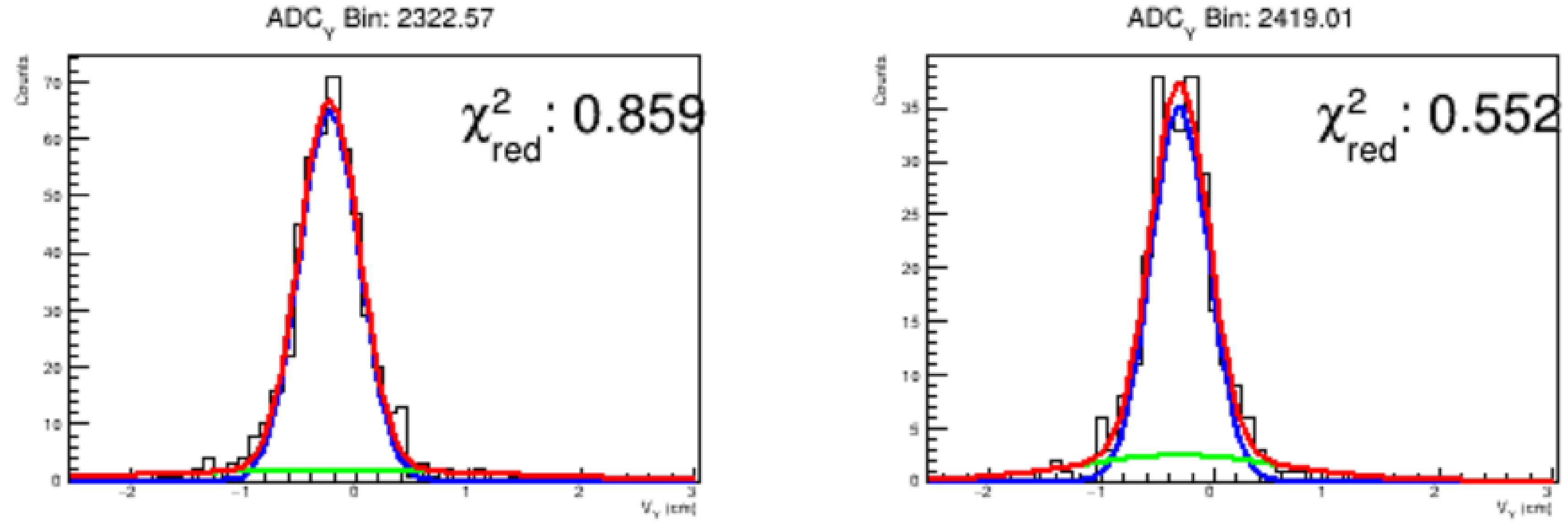






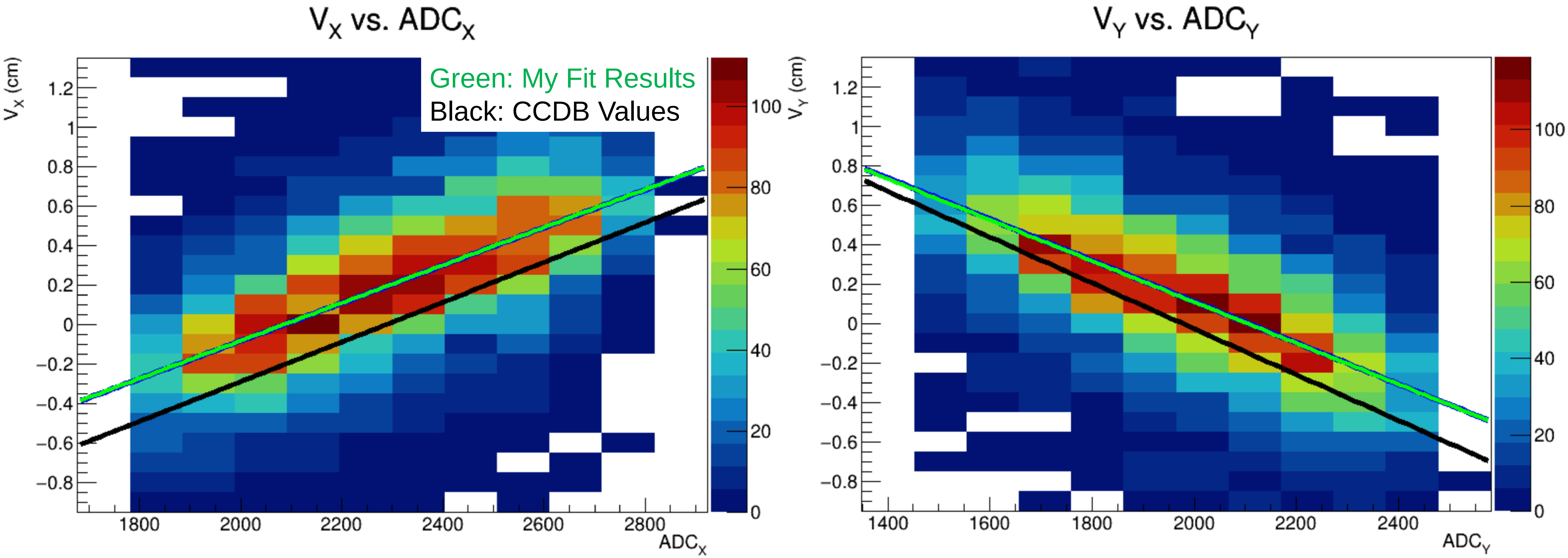
Run 16296: X-Vertex Histograms





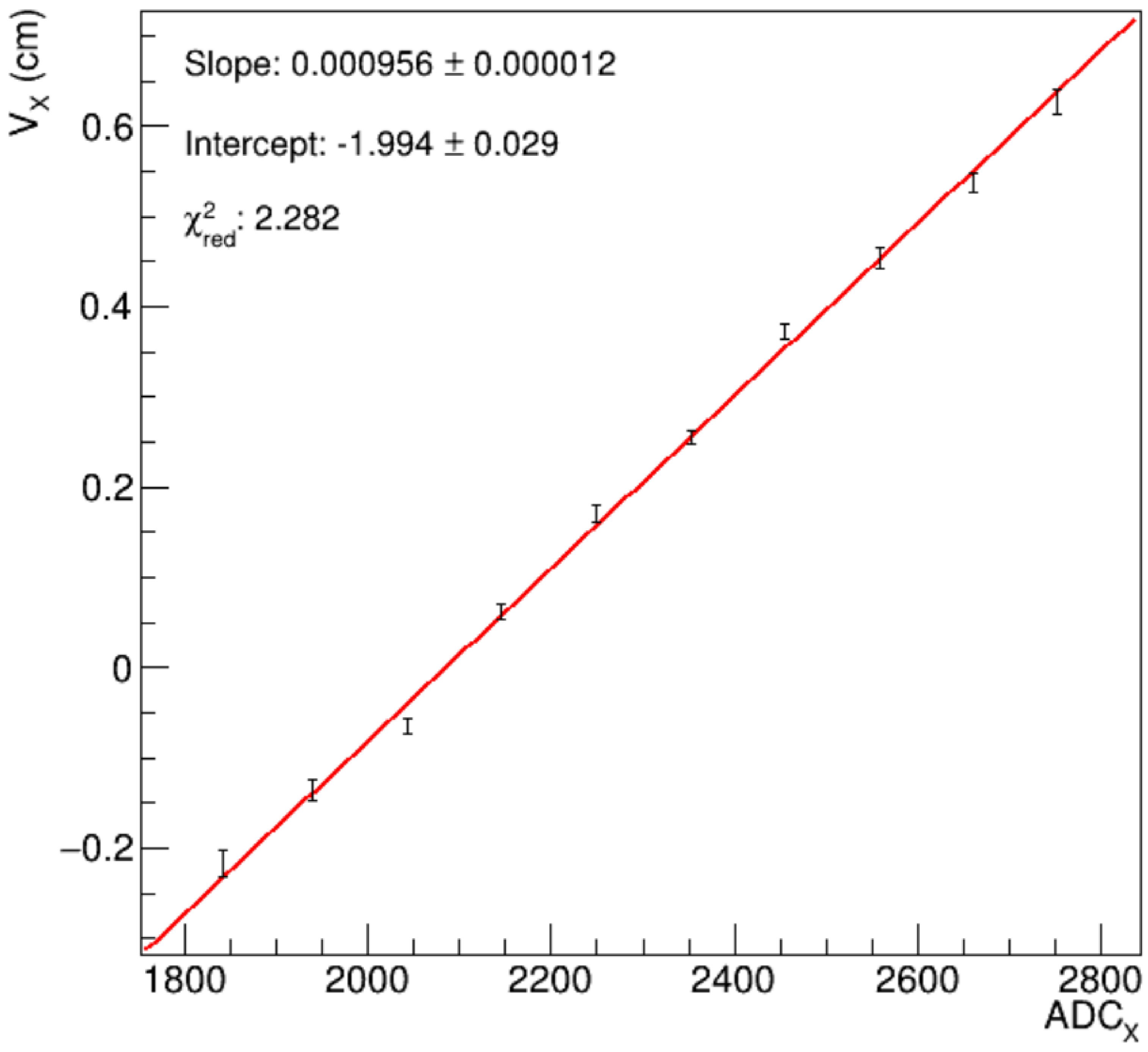
Run 16296: Y-Vertex Histograms

Run 16296: 2D Histograms

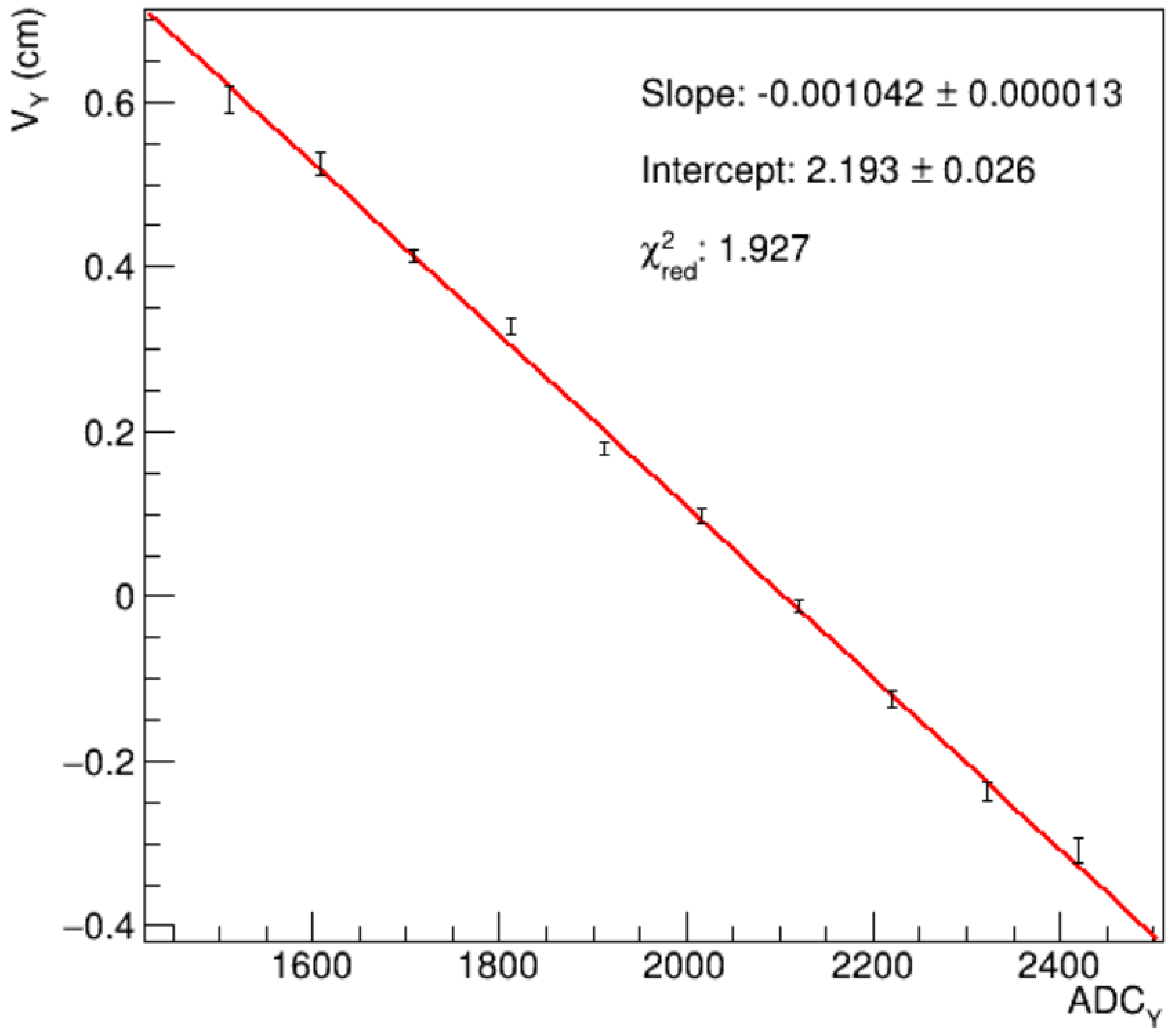


(Same as Before)

Run 16296: Plot of V_X vs. ADC_X



Run 16296: Plot of V_Y vs. ADC_Y



Backup Slides

- What do the histograms look like?
- Effects of binning on raster calibration constants
- Changes in reconstructed vertices using my constants versus what's in CCDB

Effects of Binning on Data

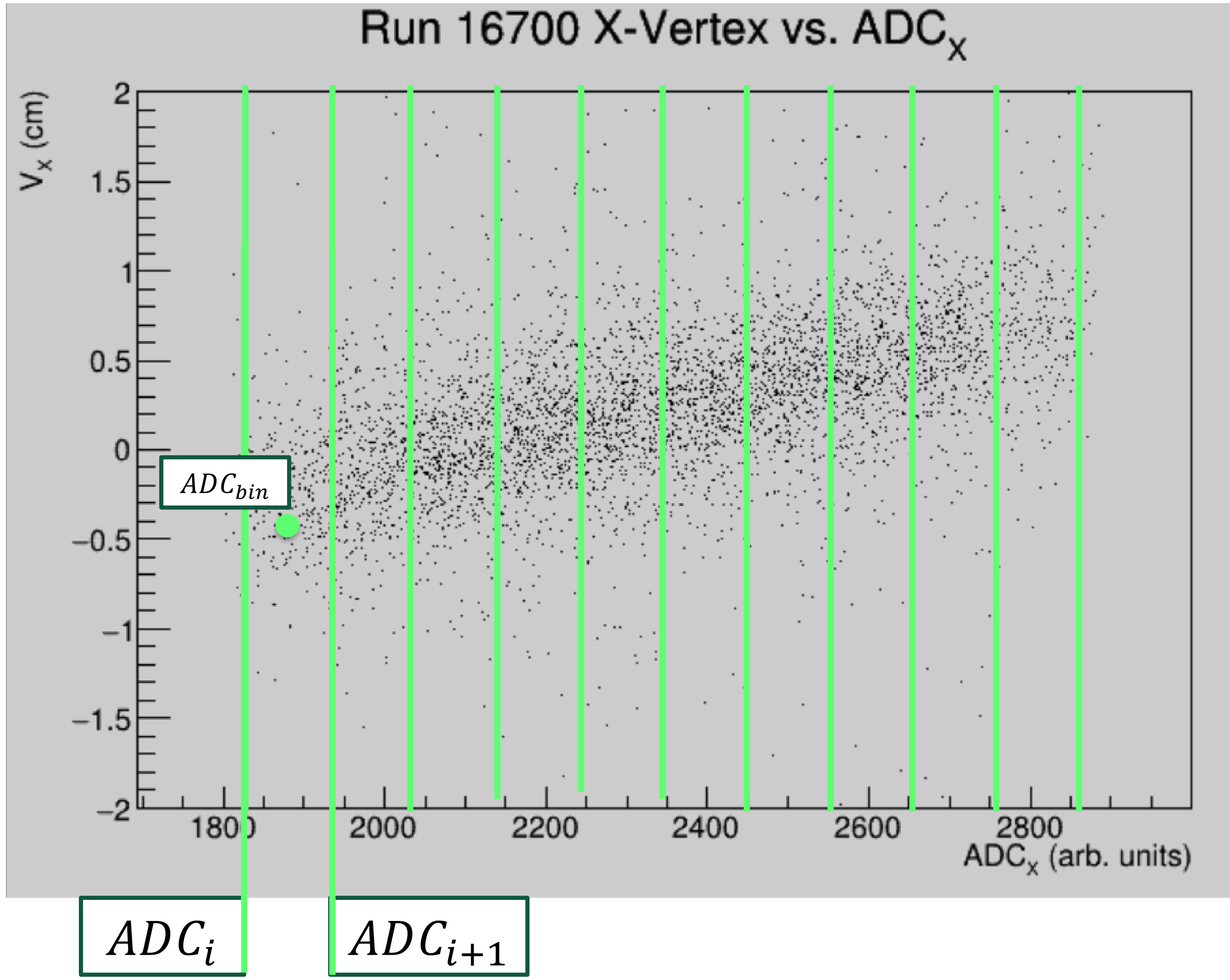
- Current goal is to recook at least one run using the raster calibration constants calculated using my method

- However, I investigated how binning the ADC data would affect the extracted constants
  - Does the ADC value used for the bin affect the constants?
  - Does the number of data points in each bin affect the constants?
  - Does the number of bins used affect the constants?

- My old method had some issues, but I believe they may be fixed now!

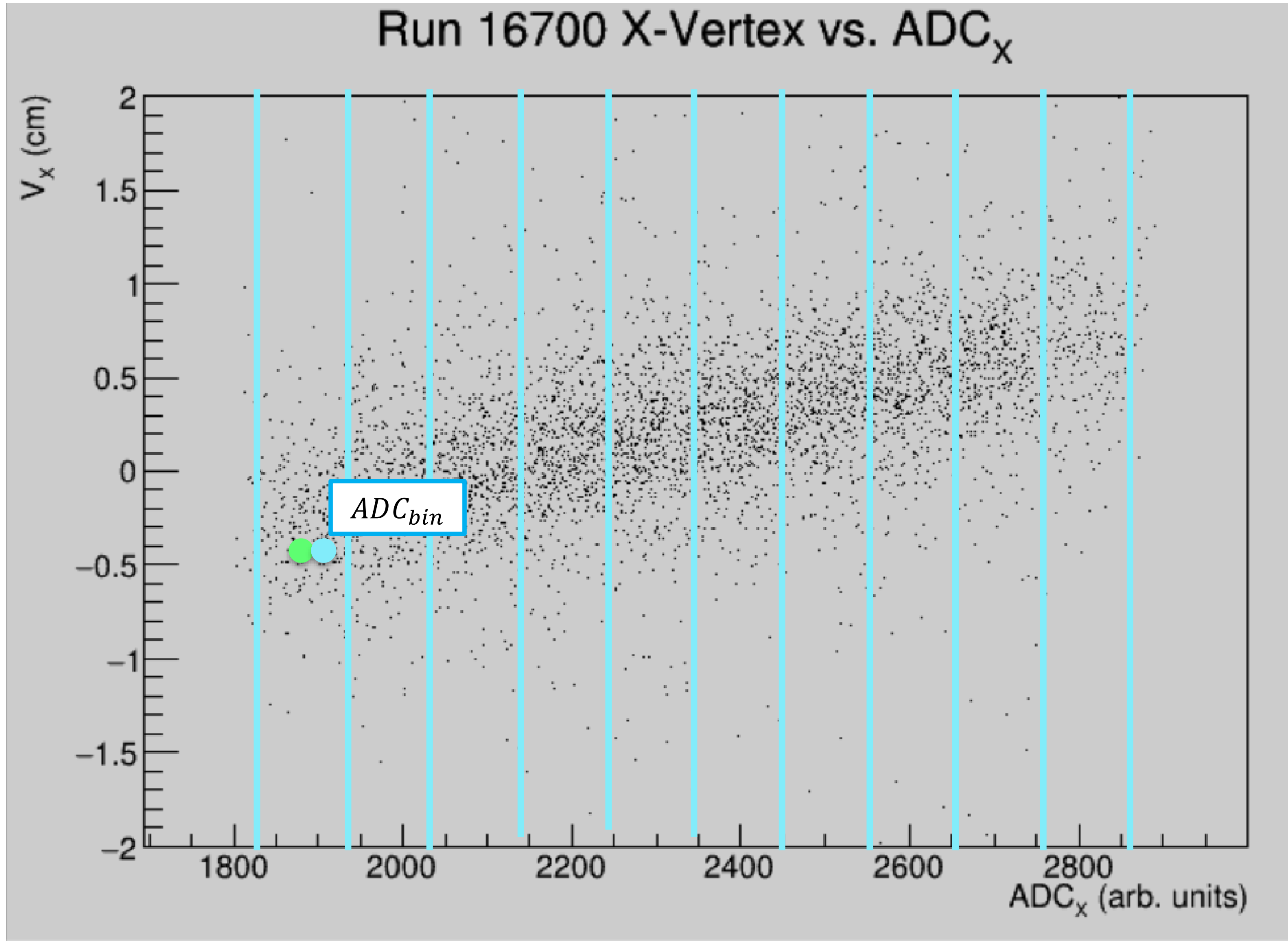
Effects of Binning on Data

- Plot extracted x, y vertices vs. $ADC_{x,y}$ signals and perform profile fits to find $V_{x,y} = P_0 + P_1 * ADC_{x,y}$
- **Old method**: partition data into 10 bins, with the ADC value used to represent that bin taken as the average of the bin edges: $ADC_{bin} = (ADC_i + ADC_{i+1})/2$



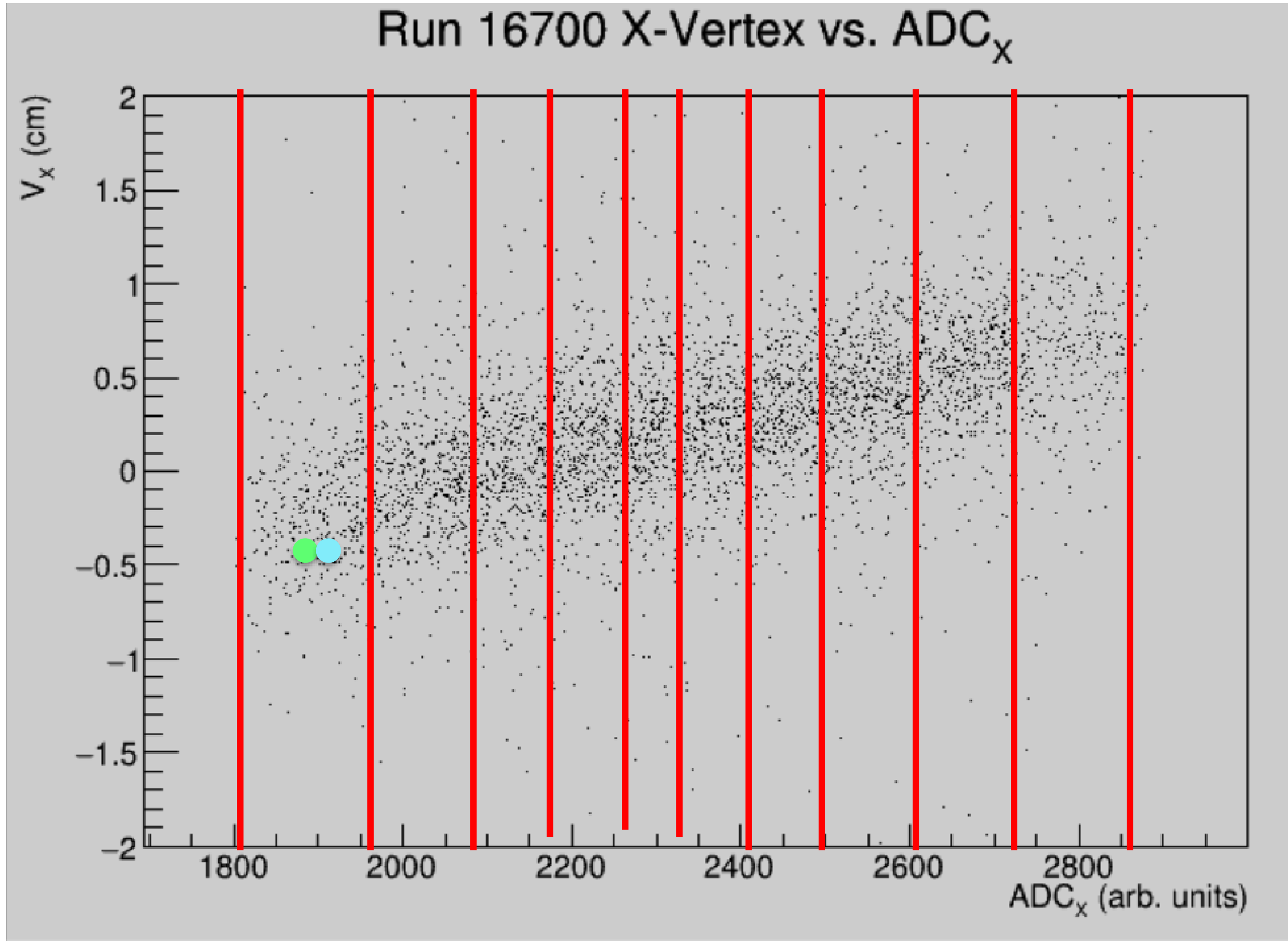
Effects of Binning on Data

- Plot extracted x, y vertices vs. $ADC_{x,y}$ signals and perform profile fits to find $V_{x,y} = P_0 + P_1 * ADC_{x,y}$
- **Old method**: partition data into 10 bins, with the ADC value used to represent that bin taken as the average of the bin edges: $ADC_{bin} = (ADC_i + ADC_{i+1})/2$
- **New method**: use the average value of ADC signal in each bin instead of central value: $ADC_{bin} = \sum_i^N ADC_i \,/N$
    - N = number of data points in bin



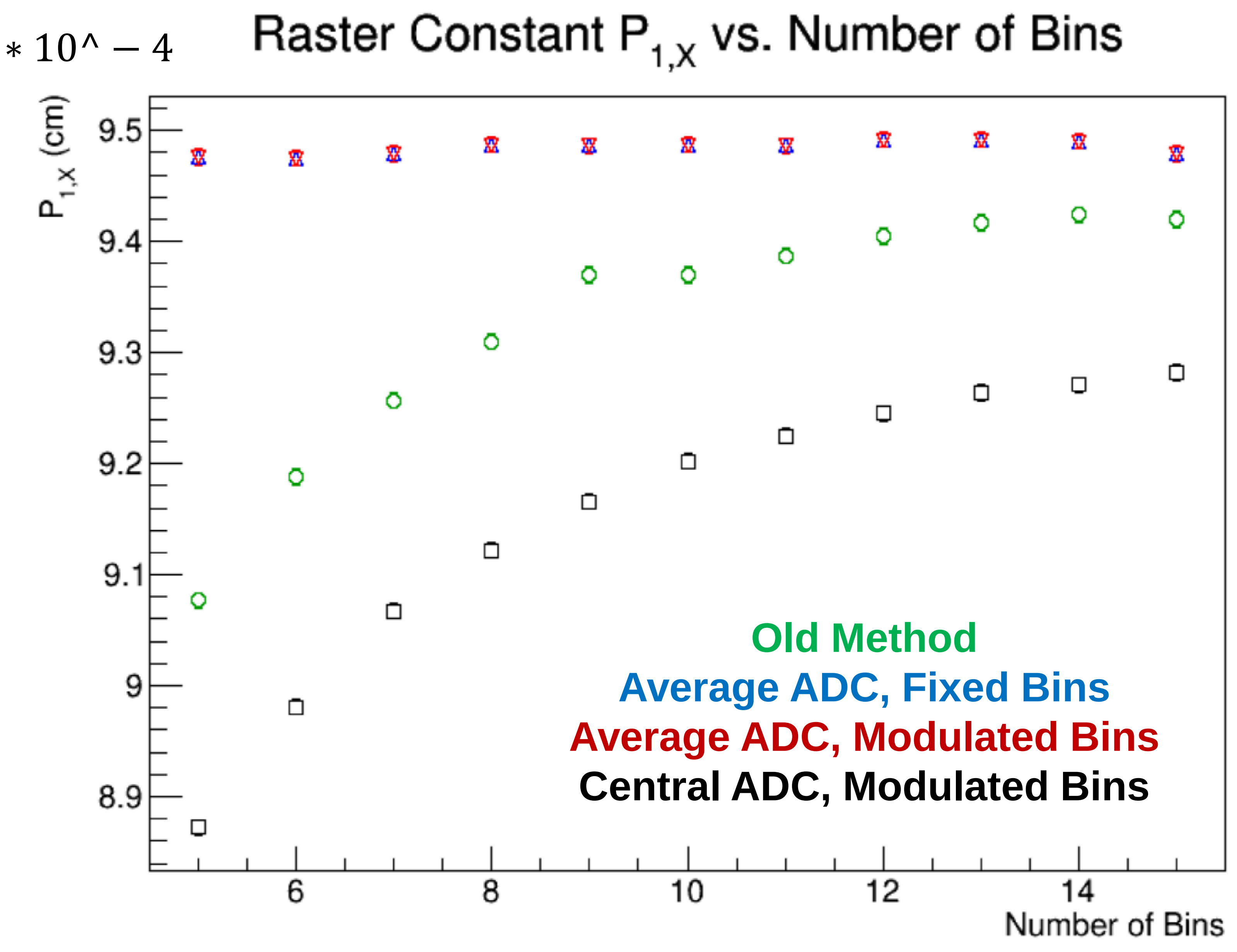
Effects of Binning on Data

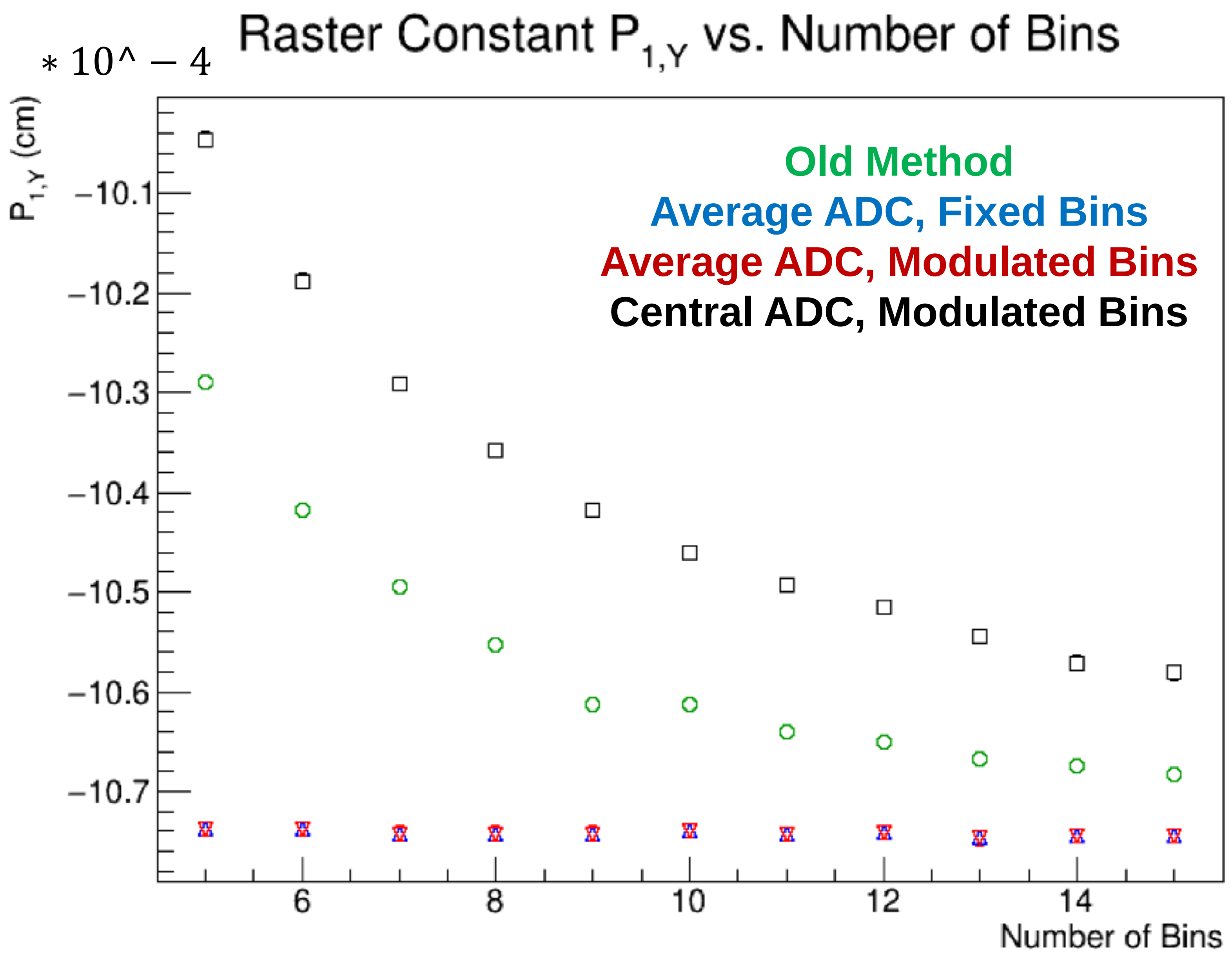
- Plot extracted x, y vertices vs. $ADC_{x,y}$ signals and perform profile fits to find $V_{x,y} = P_0 + P_1 * ADC_{x,y}$
- **Old method**: partition data into 10 bins, with the ADC value used to represent that bin taken as the average of the bin edges: $ADC_{bin} = (ADC_i + ADC_{i+1})/2$
- **New method**: use the average value of ADC signal in each bin instead of central value: $ADC_{bin} = \sum_i^N ADC_i \,/N$
  - N = number of data points in bin
- **Another Method**: use modulated bin widths so that each bin contains roughly the same number of data points
- Tested all of these using fixed and modulated bins
- Also used tighter vertex cut: $-8\ cm < V_z < 2\ cm$, $|V_{x,y}| < 2.05\ cm$



Updates From Last Time

- Using the average ADC signal for each bin gave consistent results, no matter the binning
- Should we use these new values for any future cooking?





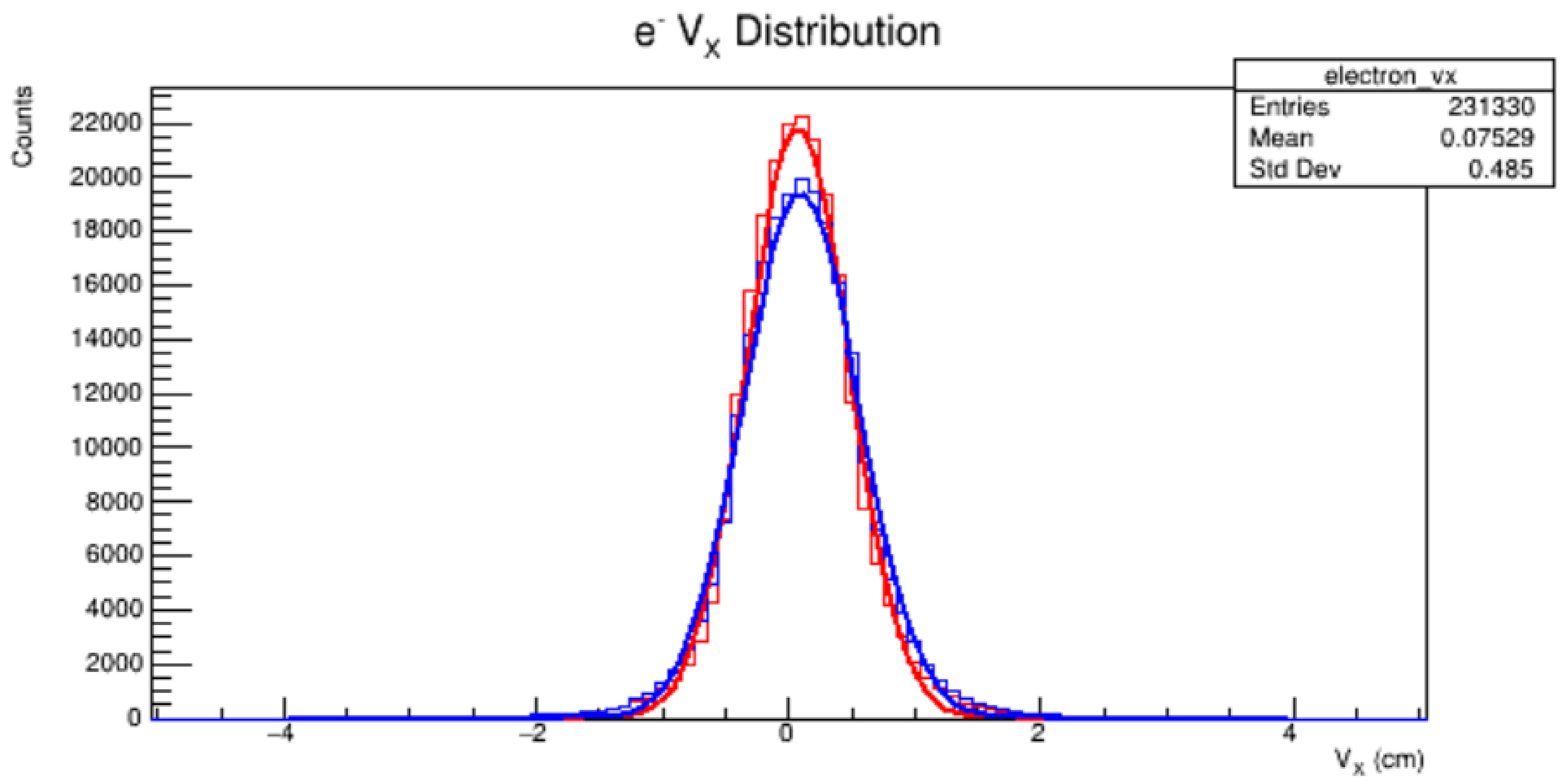
Backup Slides

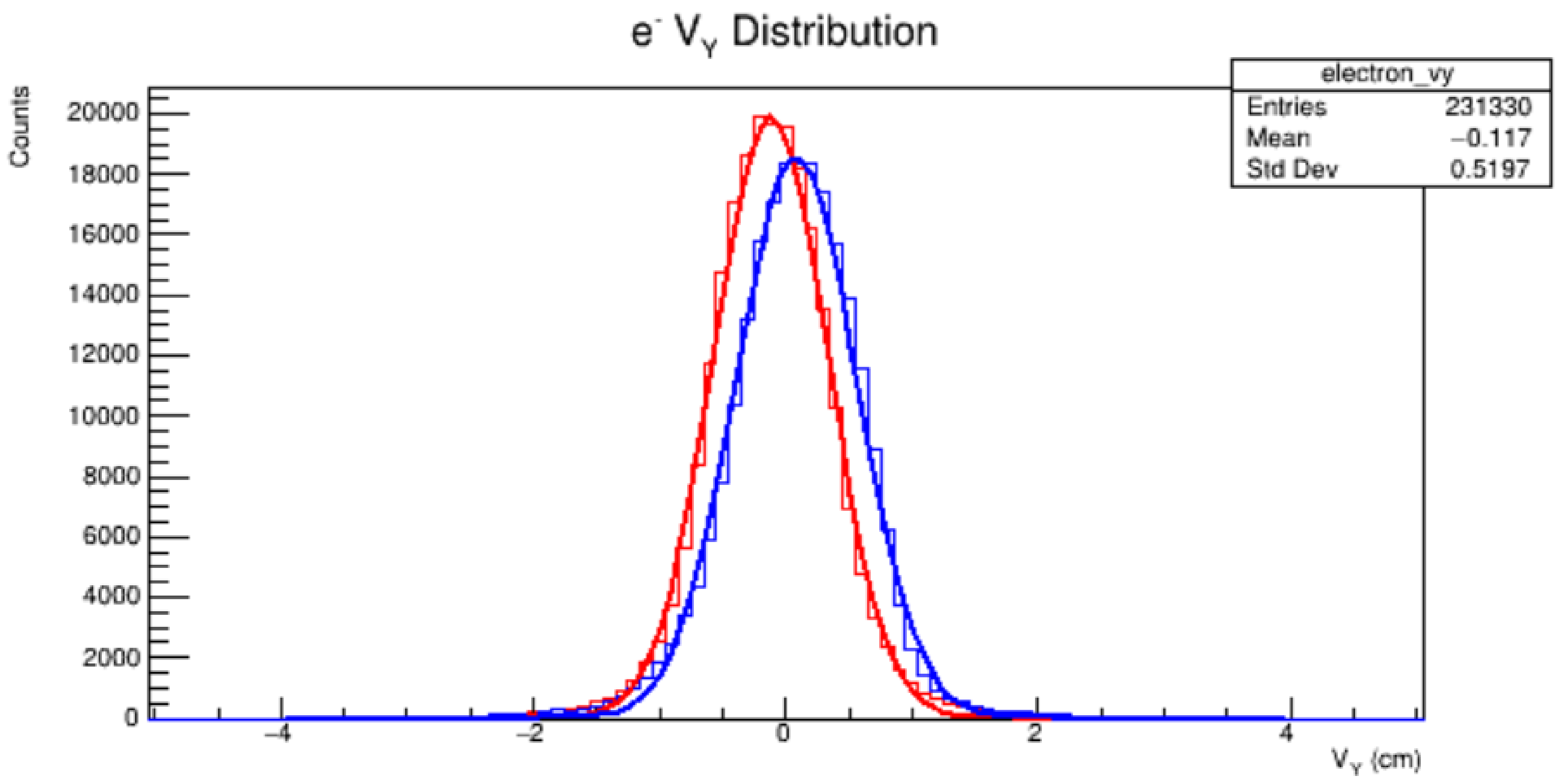
- What do the histograms look like?
- Effects of binning on raster calibration constants
- Changes in reconstructed vertices using my constants versus what’s in CCDB

Work Still To Be Done

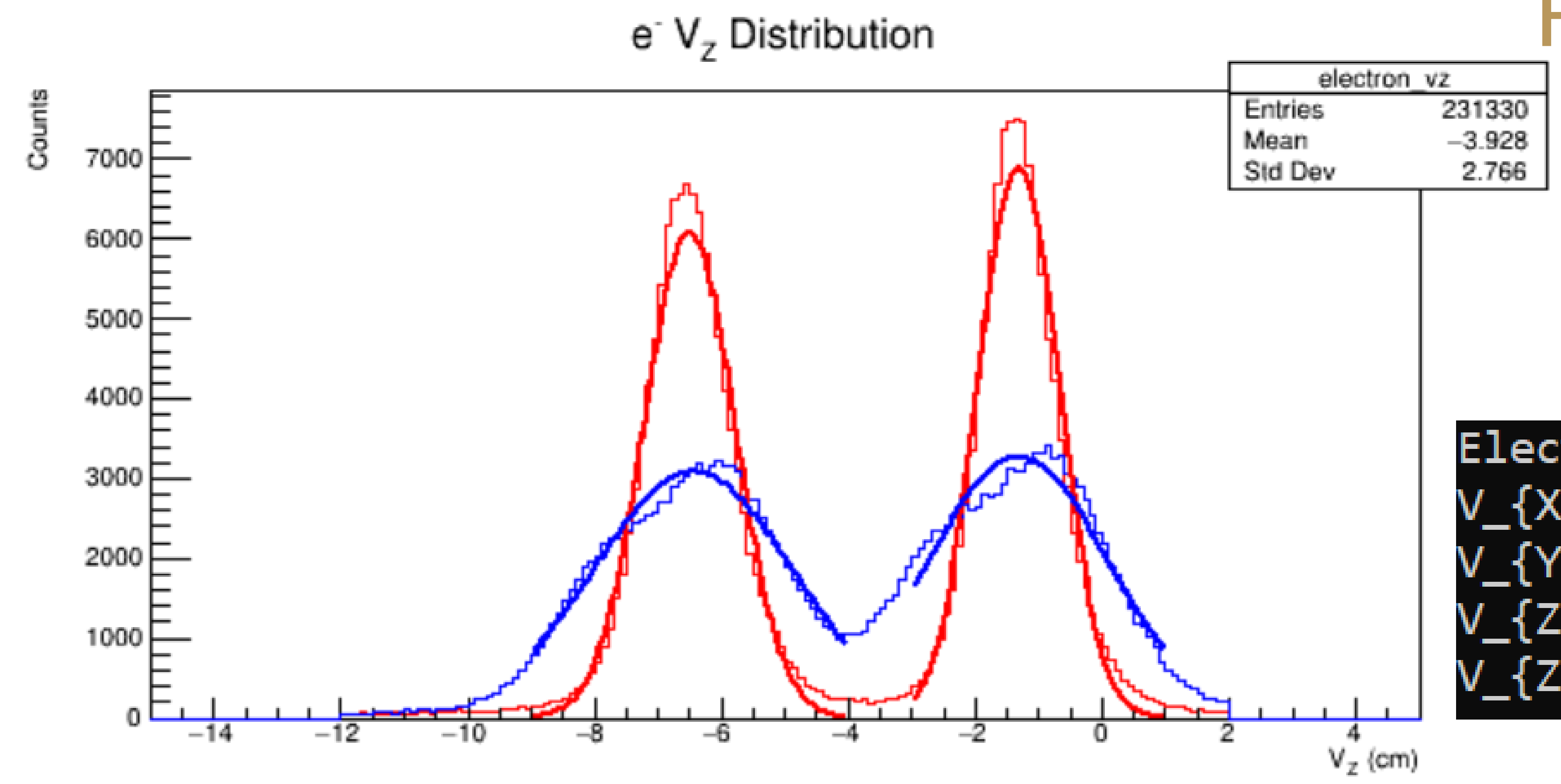
- Made new cuts to the data:
  - $V_X^2 + V_Y^2 < 10\ cm^2, \quad -8\ cm < V_Z < 2\ cm$
  - All particles have $\chi^2_{pid} < 3$, e^- seen in the FD, π^+ seen in the CD
  - Made cuts to the track reconstruction $\chi^2_{red} \equiv \chi^2_{track}/NDF < 10.0$
- New method: for π^+, use vertices from REC::UTrack instead of REC::Track
- Made new plots for runs 16296, 16297 using REC::UTrack and comparing it to REC::Track
- Also made plots for for run 16194 (using REC::UTrack for π^+)

- Using REC::UTrack didn't shift the z-vertices of the π^+, but it was harder to tell if the x,y vertices shifted
  - Poor results for the gaussian fits of x,y vertices
  - There does seem to be some sort of small shift, but it's hard to explicitly quantify





Run 16194 Electron REC::Track Data



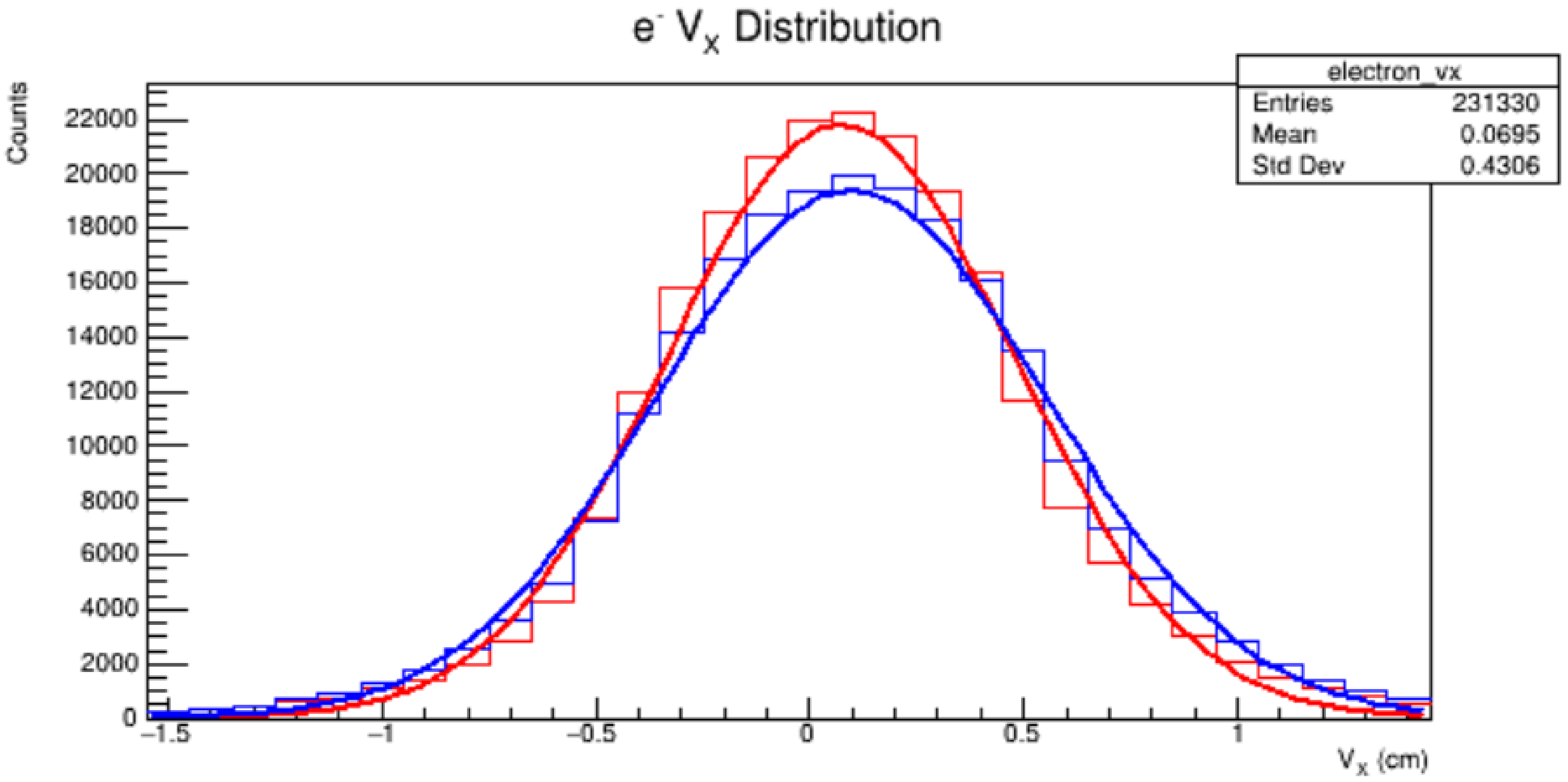
Derek's Raster Calibration Values

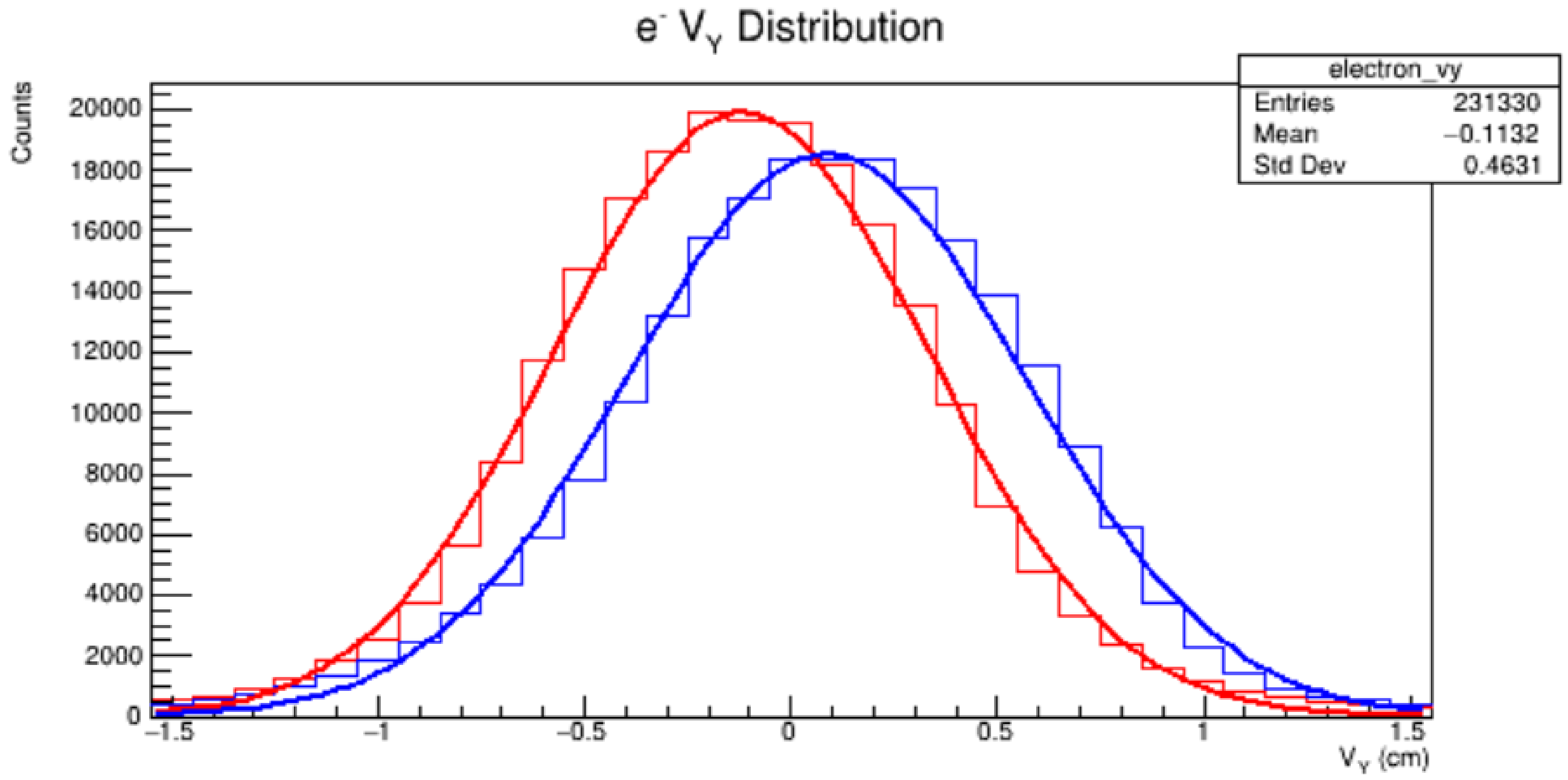
CCDB Values (Beam Offset and Raster)

$V_i = \mu \pm \sigma$ of gaussian fit

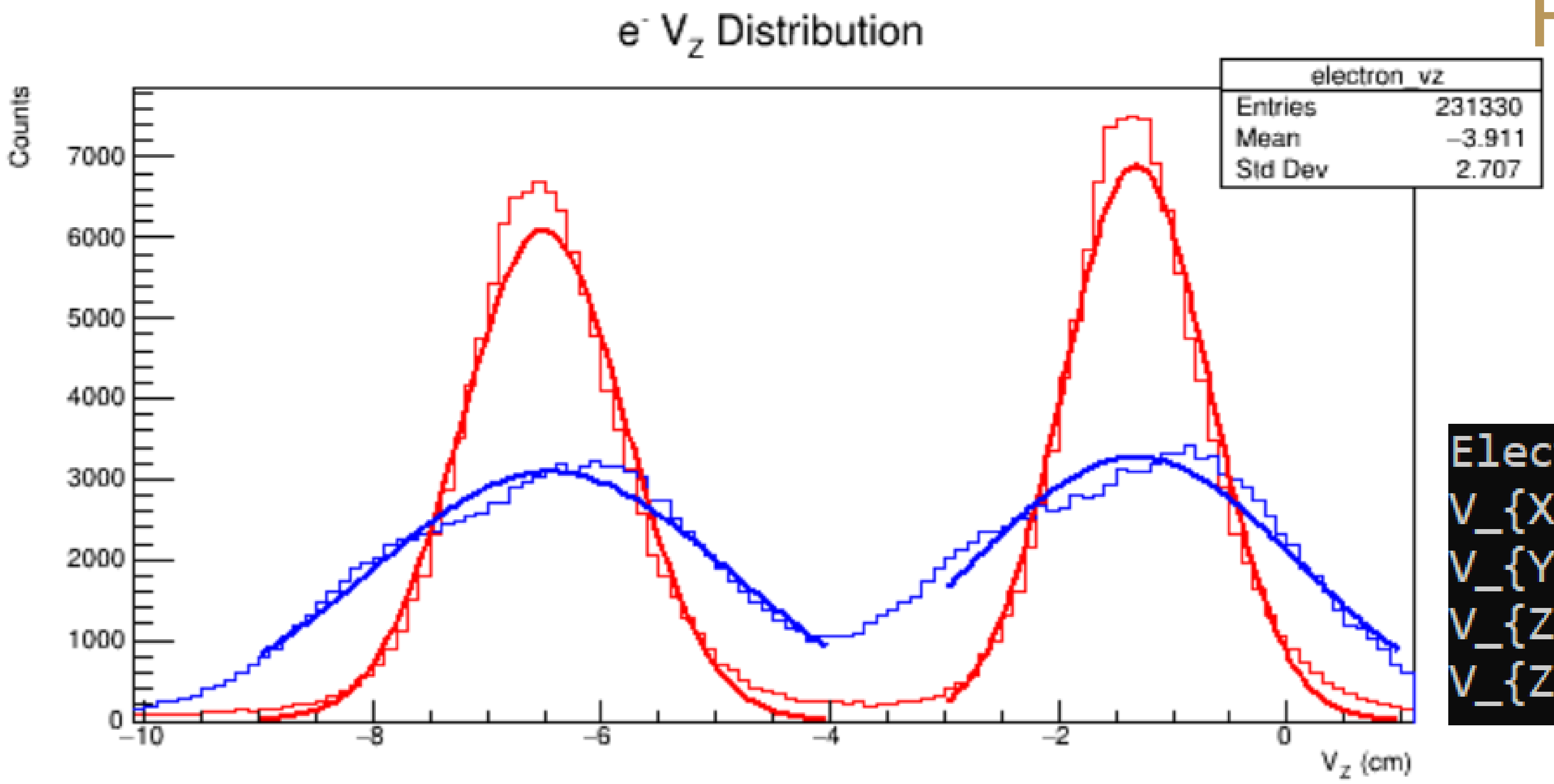
```
Electron Vertices (My Constants)
V_{X} = 0.0730246 +- 0.408821
V_{Y} = -0.119356 +- 0.452616
V_{Z1} = -6.50846 +- 0.720149
V_{Z2} = -1.32131 +- 0.641867
```

```
Electron Vertices (CCDB Constants)
V_{X} = 0.0973734 +- 0.458523
V_{Y} = 0.0848283 +- 0.479743
V_{Z1} = -6.45113 +- 1.56239
V_{Z2} = -1.32759 +- 1.41781
```





Run 16194 Electron REC::Track Data



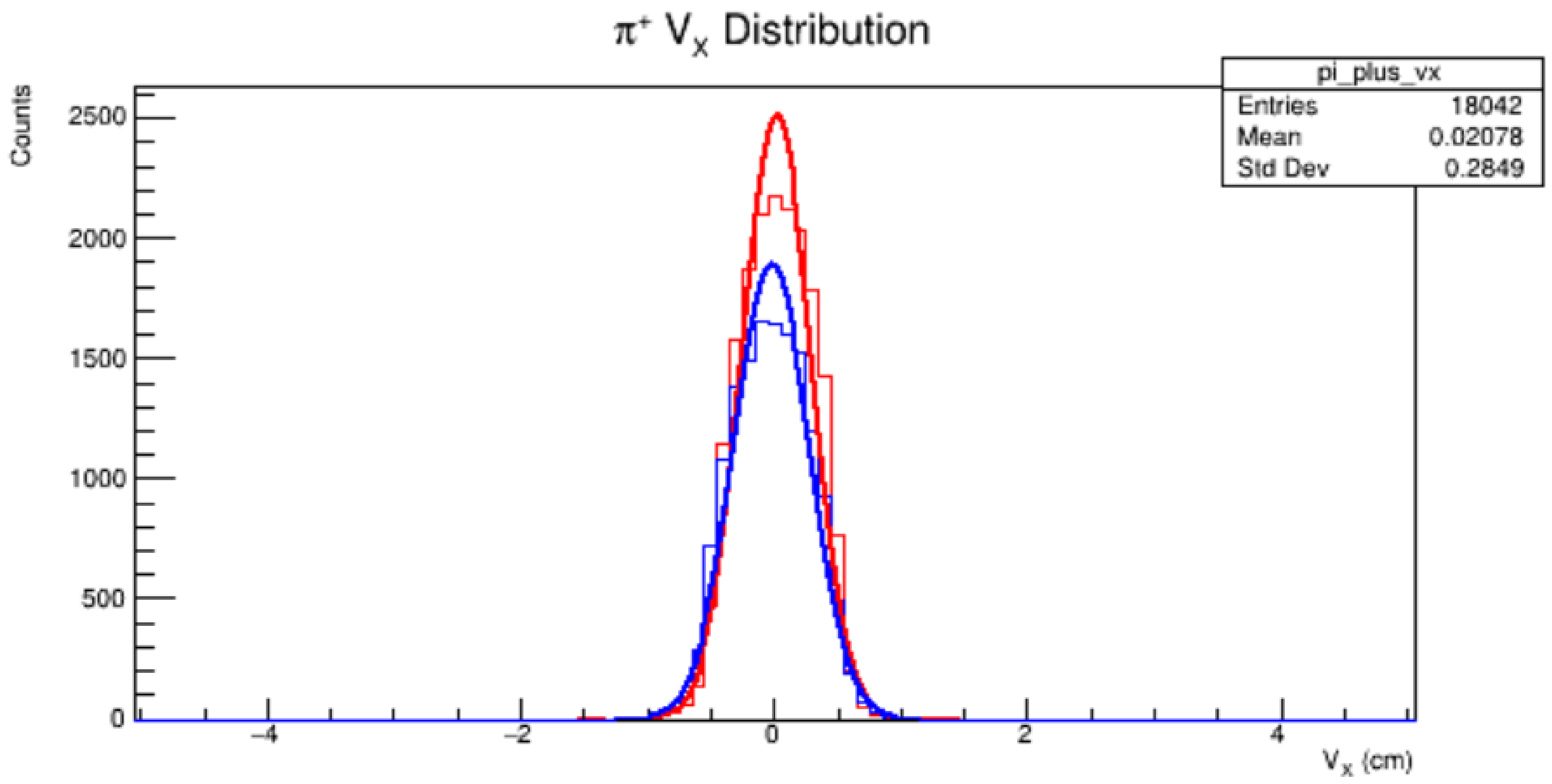
Derek's Raster Calibration Values

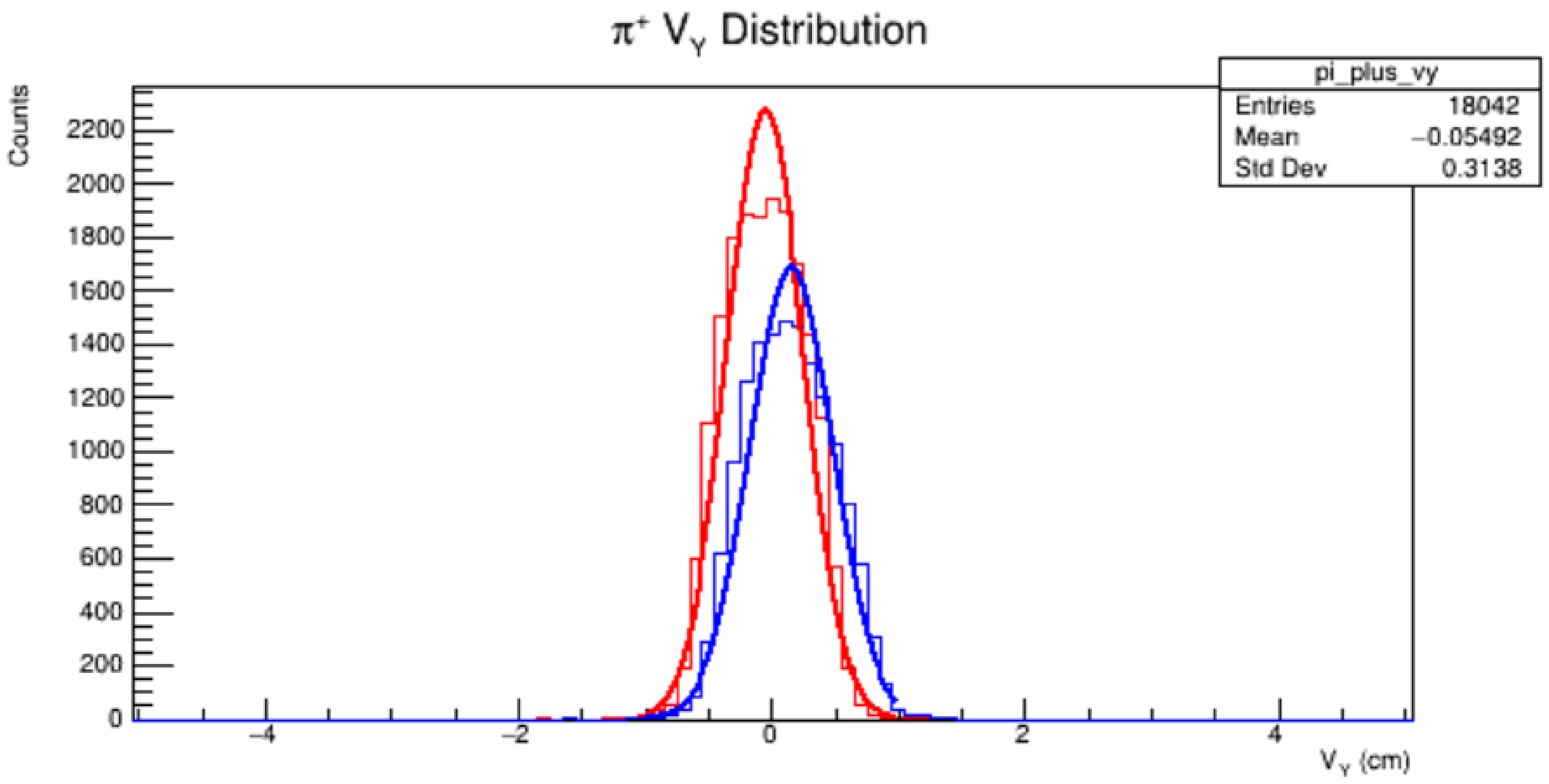
CCDB Values (Beam Offset and Raster)

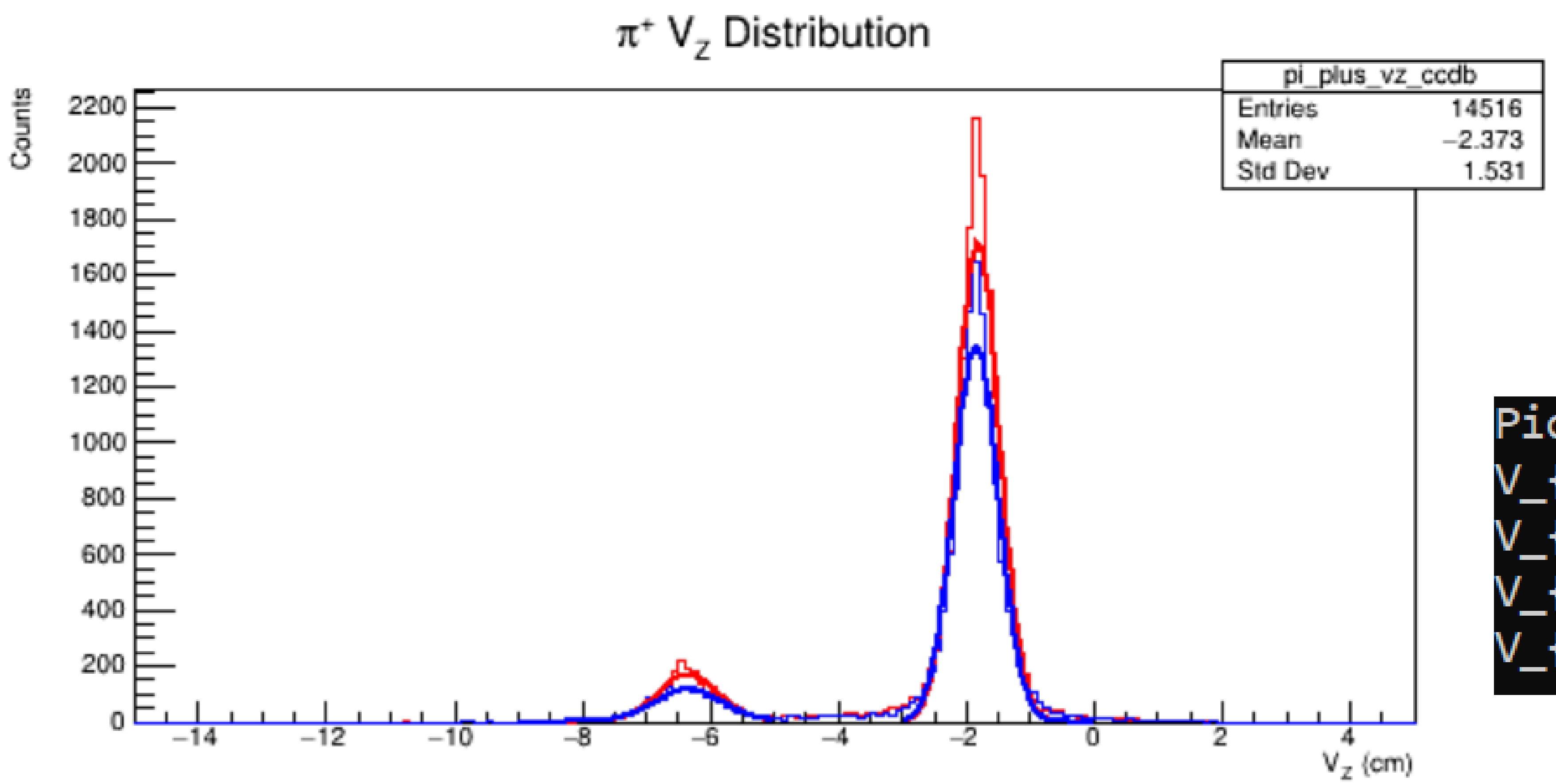
$V_i = \mu \pm \sigma$ of gaussian fit

```
Electron Vertices (My Constants)
V_{X} = 0.0730246 +- 0.408821
V_{Y} = -0.119356 +- 0.452616
V_{Z1} = -6.50846 +- 0.720149
V_{Z2} = -1.32131 +- 0.641867
```

```
Electron Vertices (CCDB Constants)
V_{X} = 0.0973734 +- 0.458523
V_{Y} = 0.0848283 +- 0.479743
V_{Z1} = -6.45113 +- 1.56239
V_{Z2} = -1.32759 +- 1.41781
```







Run 16194 Pion REC::UTrack Data

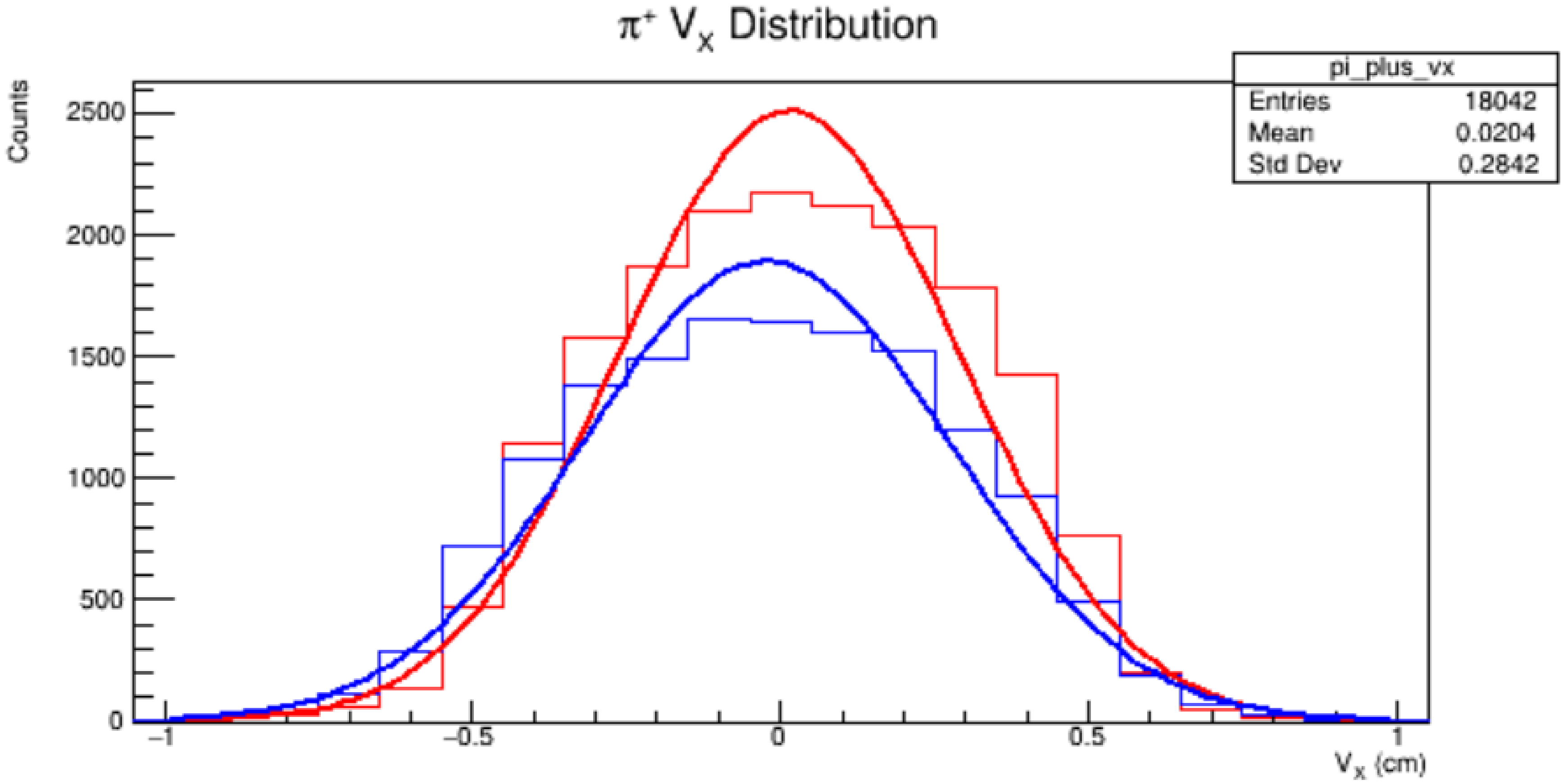
Derek's Raster Calibration Values

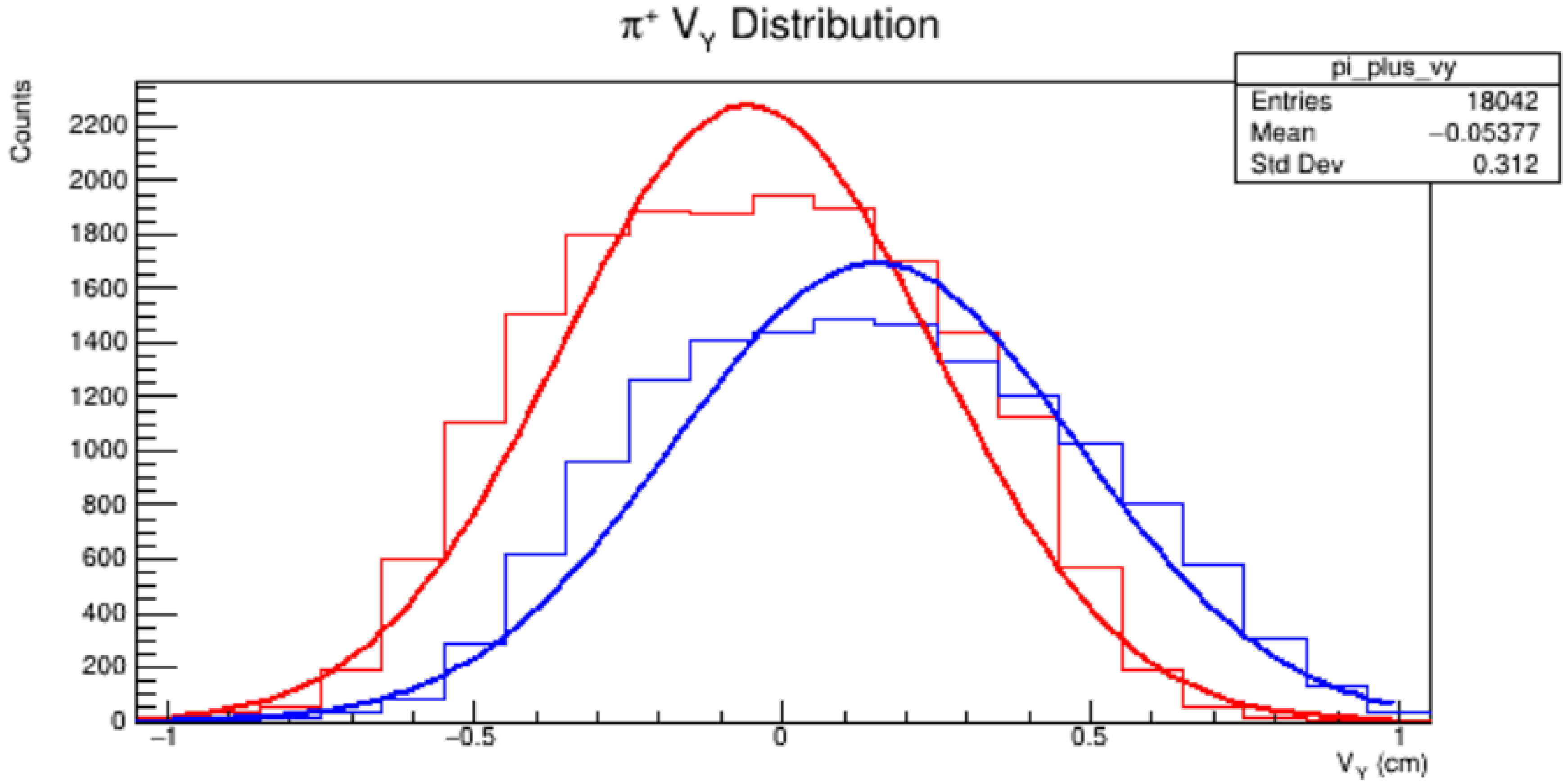
CCDB Values (Beam Offset and Raster)

$V_i = \mu \pm \sigma$ of gaussian fit

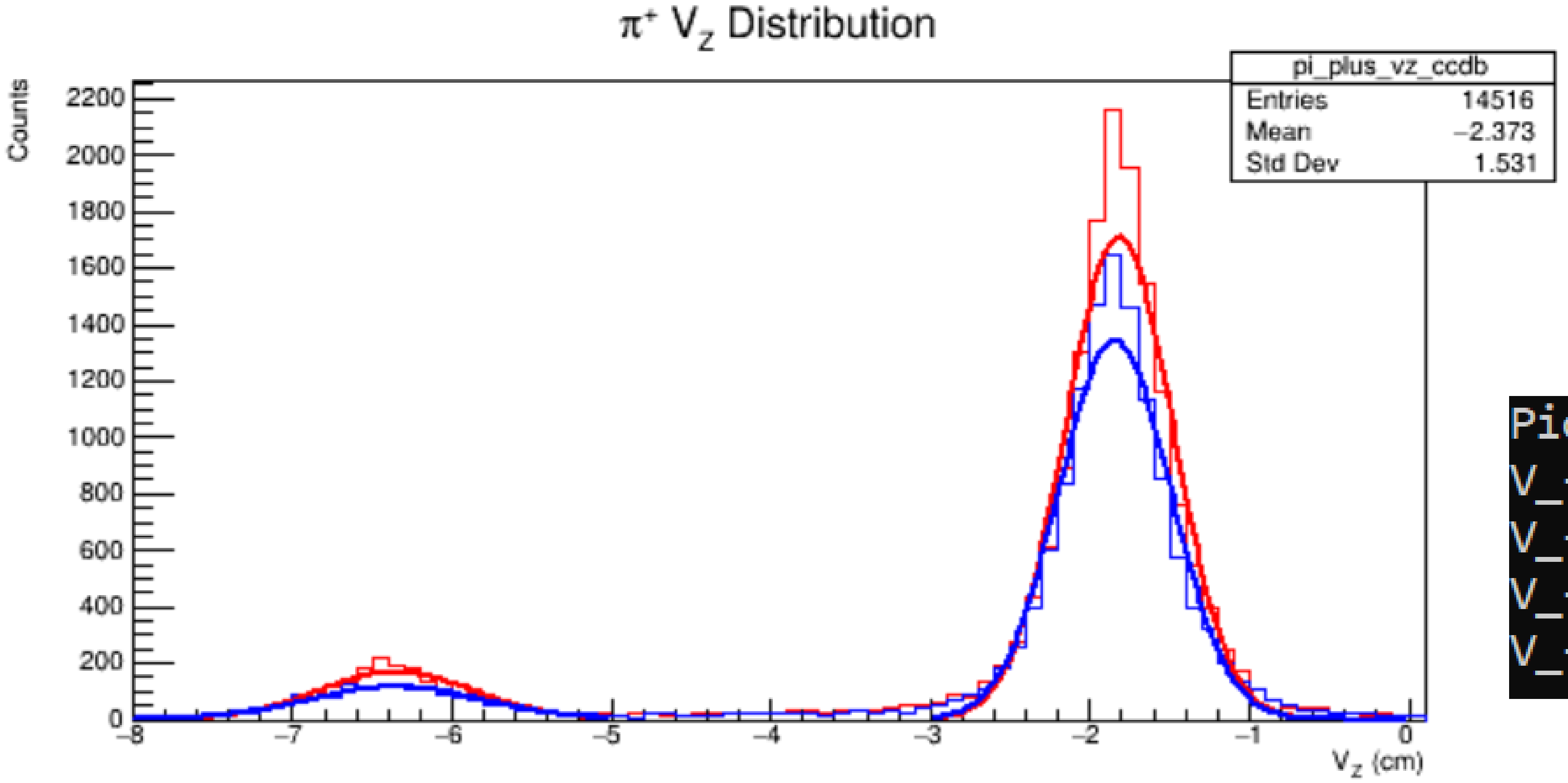
```
Pion Vertices (My Constants)
V_{X} = 0.0143408 +- 0.274806
V_{Y} = -0.0564504 +- 0.302426
V_{Z1} = -6.35221 +- 0.48992
V_{Z2} = -1.81707 +- 0.337596
```

```
Pion Vertices (CCDB Constants)
V_{X} = -0.0236244 +- 0.298057
V_{Y} = 0.150404 +- 0.327656
V_{Z1} = -6.35166 +- 0.543916
V_{Z2} = -1.845 +- 0.351192
```





Run 16194 Pion REC::UTrack Data



Derek’s Raster Calibration Values

CCDB Values (Beam Offset and Raster)

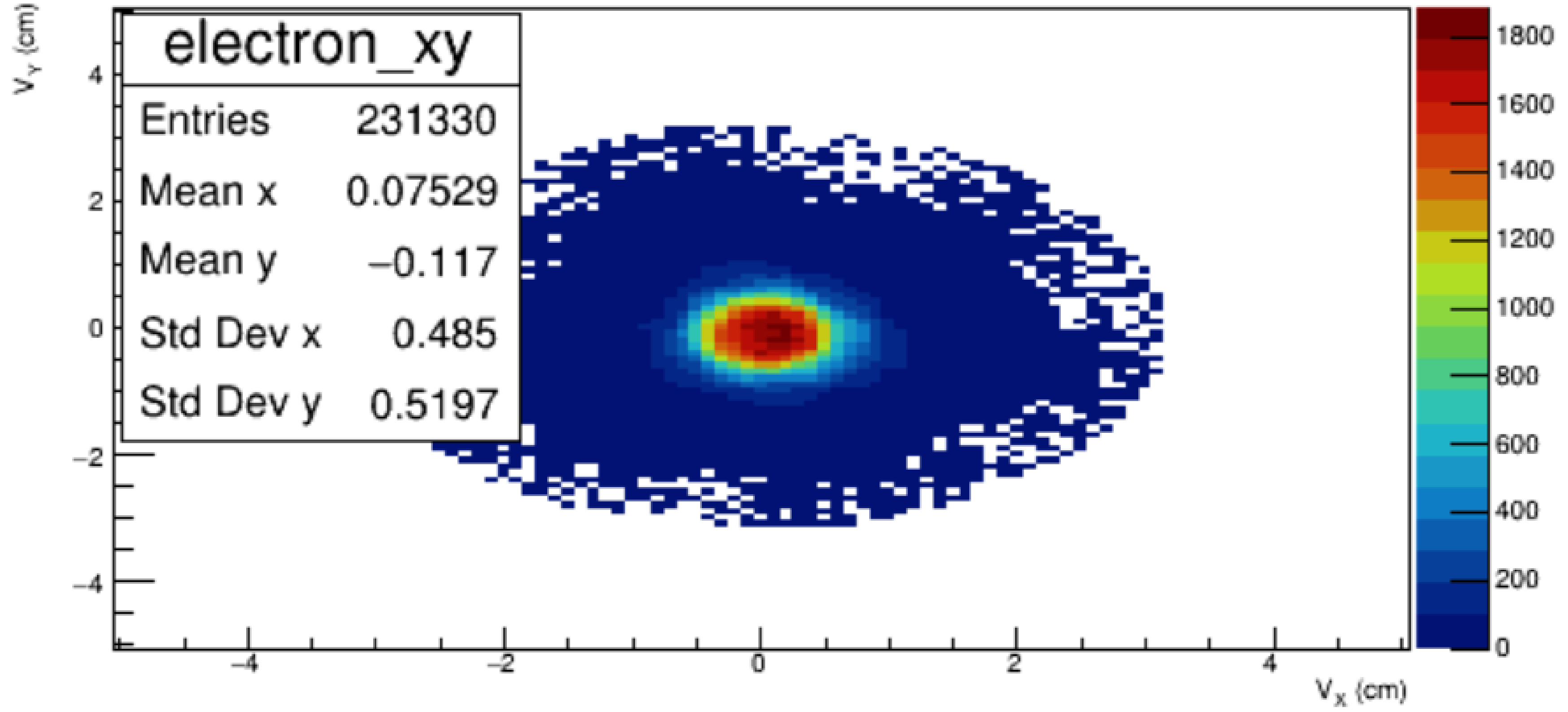
$V_i = \mu \pm \sigma$ of gaussian fit

```
Pion Vertices (My Constants)
V_{X} = 0.0143408 +- 0.274806
V_{Y} = -0.0564504 +- 0.302426
V_{Z1} = -6.35221 +- 0.48992
V_{Z2} = -1.81707 +- 0.337596
```

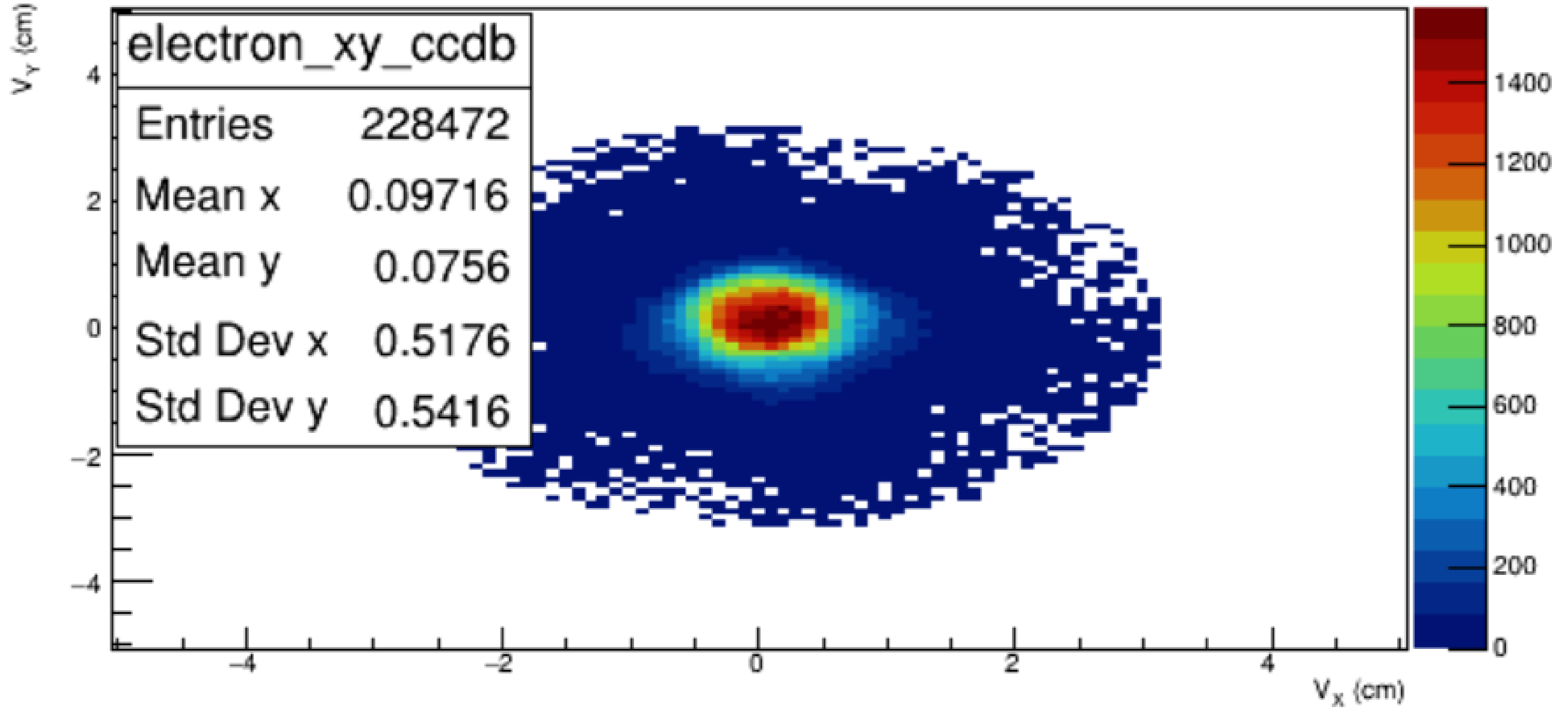
```
Pion Vertices (CCDB Constants)
V_{X} = -0.0236244 +- 0.298057
V_{Y} = 0.150404 +- 0.327656
V_{Z1} = -6.35166 +- 0.543916
V_{Z2} = -1.845 +- 0.351192
```

Run 16194 REC::UTrack Data

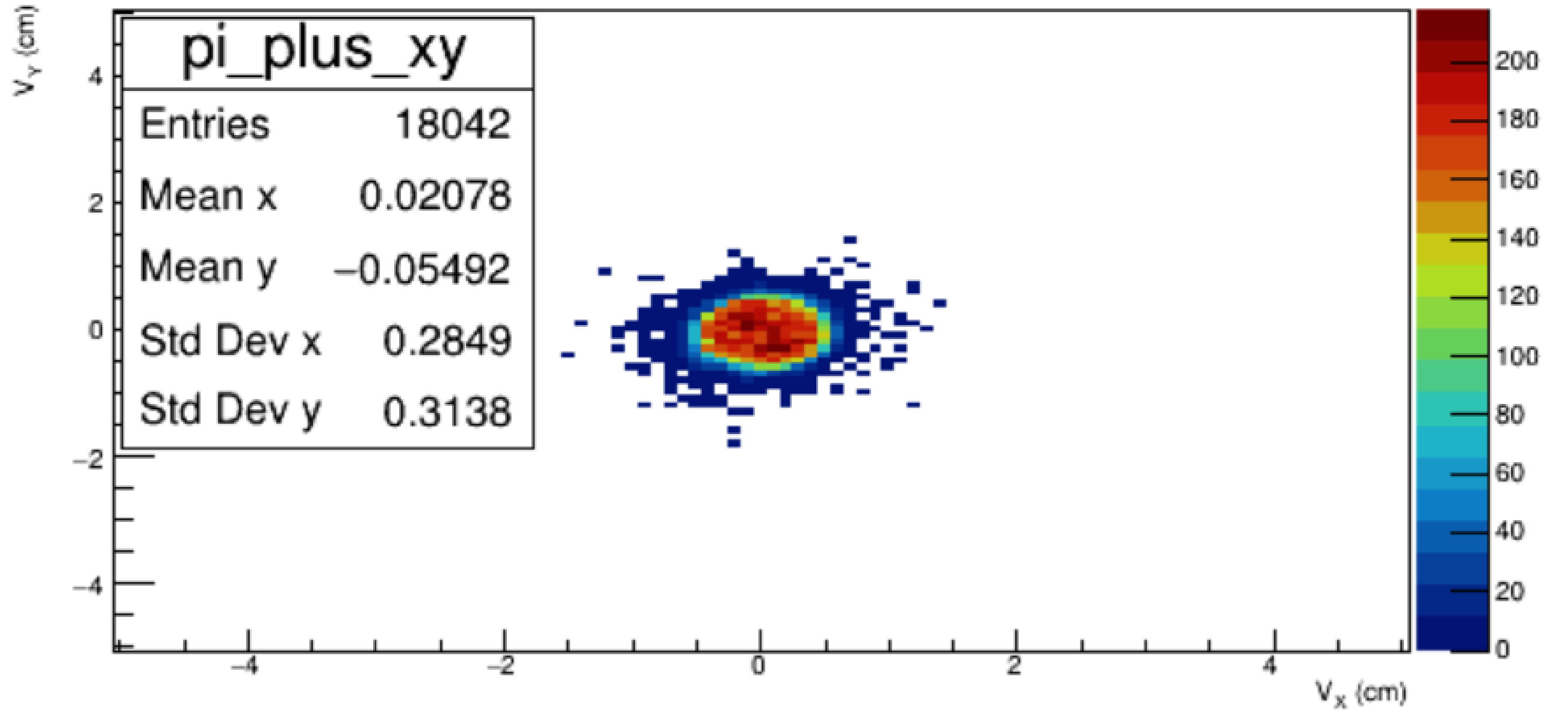
e^- V_X vs. V_Y (My Values)



e^- V_X vs. V_Y (CCDB)



π^+ V_X vs. V_Y (My Values)



π^+ V_X vs. V_Y (CCDB)