Run-5 MUID Efficiency Results and Technical Details

David Silvermyr, Jamie Nagle, MinJung Kweon, DongJo Kim, Vince Cianciolo

May 23, 2006

We present a complete set of technical details on the method for determining the MUID efficiency from real data (DataMethod) method – previously referred to as the Colorado Method) and from the current draw measurements (HVMethod). We include the detailed study and final determination of MUID efficiencies for Run-5 Proton-Proton 200 GeV and Run-5 Copper-Copper 200 GeV and 62 GeV data sets. In this note, all figures labeled Copper-Copper are 200 GeV.

1. Introduction

- Macros, scripts and results in CVS
- Files on disk and in HPSS

2. Results and Method Comparisons

- Run-5 Proton-Proton
- Run-5 Copper-Copper

3. Final Results and Systematic Errors

Appendix A: MUID Geometry

Appendix B: Data (Colorado) Method

Appendix C: HVMethod

Appendix D: Overlap Correction

1. **Introduction**

We utilize two different methods to estimate the MUID efficiency.

The first one, which we will refer to as the DataMethod (previously called the Colorado method) uses offline reconstructed roads. We make the requirement that the roads should have passed all cuts with and without a hit in the MUID plane in question, and then check if we have a hit or not in that plane to calculate the efficiency.

The second method, which we will refer to as the HVMethod, examines the High Voltage (HV) log files from the experiment, which contain frequent snapshots (one approx. every 60 seconds) of the status for each MUID HV chain. Based on the determined no-beam currents, known broken wires, etc., we estimate what the beam-related current draw, and resulting efficiency loss due to Voltage sag. Note that this method will not catch other (non-HV) problems, such as Low Voltage (LV), Front End Electronics (FEE) or from Voltage sag present without beam.

We then compare the estimates from the two methods for the final result. Since the DataDriven method should in principle catch all problem HV Chains, the DataDriven method results are used except for the chains that have very low statistics (and therefore large uncertainties) in the DataDriven method, where we then replace the values with the HV method results. We have found that these low statistics HV groups correspond to regions on the edge of the acceptance.

This note focuses mainly on the details of the MUID efficiency determination method, and the final results from Run5 Proton-Proton and Run-5 Copper-Copper data sets. The Run-4 Gold-Gold analysis and results are described in a separate PHENIX Analysis Note.

Macros, Scripts and Results in CVS

The macros and scripts mentioned later reside in the CVS analysis areas:

offline/analysis/muideff/macros/HVMethod *HV Method macros*
offline/analysis/muideff/macros/DataMethod *DataMethod macros*

Details for running these macros and the overall procedure are outlined in README files in these directories.

There is also a results area where the final files (after merging HV and Data method results and after overlap corrections have been applied):

offline/analysis/muideff/results

The result files come in two flavors (hvgroup or twopack levels) and have a specific naming convention (see examples below). Note that the hvgroup files are used for visualization and plotting, while the twopack files are the direct input for the PHENIX Monte Carlo simulations.

muideff_run5_CuCu_200GeV_hvgroup_north.dat
muideff_run5_CuCu_200GeV_twopack_north.dat

Files on disk and in HPSS

The Run5 MuidEffic ntuple root files are on disk and in HPSS:

/phenix/data25/MUON_GROUP_PROJECTS/MuidEff/run5{pp,CuCu,CuCu624}

/home/silvermy/MuidEff/run5{pp,CuCu200,CuCu62}

The original MUID HV log files are in HPSS and the digested run-by-run ntuple info that we used are also in HPSS.

/home/claudius/

/home/silvermy/MuidHV/4.runbyrun_HVntuple/run{4AuAu,5CuCu,5pp}.tar

/phenix/data25/silvermy/work/042.run{5pp,5CuCu,4AuAu}_MUIDHV/4.runbyrun_HVntuple/

2. Results and Method Comparison

Checking Run Dependencies

Using the DataMethod, described in detail in Appendix B, we determined the MUID efficiencies panel by panel, and also HVgroup by HVgroup for each run number in the good run list for Run-5 Proton-Proton and Copper-Copper at 200 GeV. The good run lists are currently located at:

/phenix/WWW/p/draft/bickleya/pp200/DataQuality/PR05_pp_goodrun_South.list
/phenix/WWW/p/draft/bickleya/pp200/DataQuality/PR05_pp_goodrun_North.list

/phenix/WWW/p/draft/silvermy/run5/CuCu/runlists/final.txt

We then want to determine if there is any significant run to run variation that would require determining different MUID efficiency values for different subsets of the running period. In order to plot the run dependence and avoid large statistical errors, we have plotted the efficiency for each plane in the MUID (horizontal and vertical separately). If a change in efficiency is observed for a given running period, we may then need to investigate which panel or HVgroup contributes to this change. Note that these results are not applying the full set of tight cuts in order to preserve sufficient statistics. The results are shown in the following four figures.

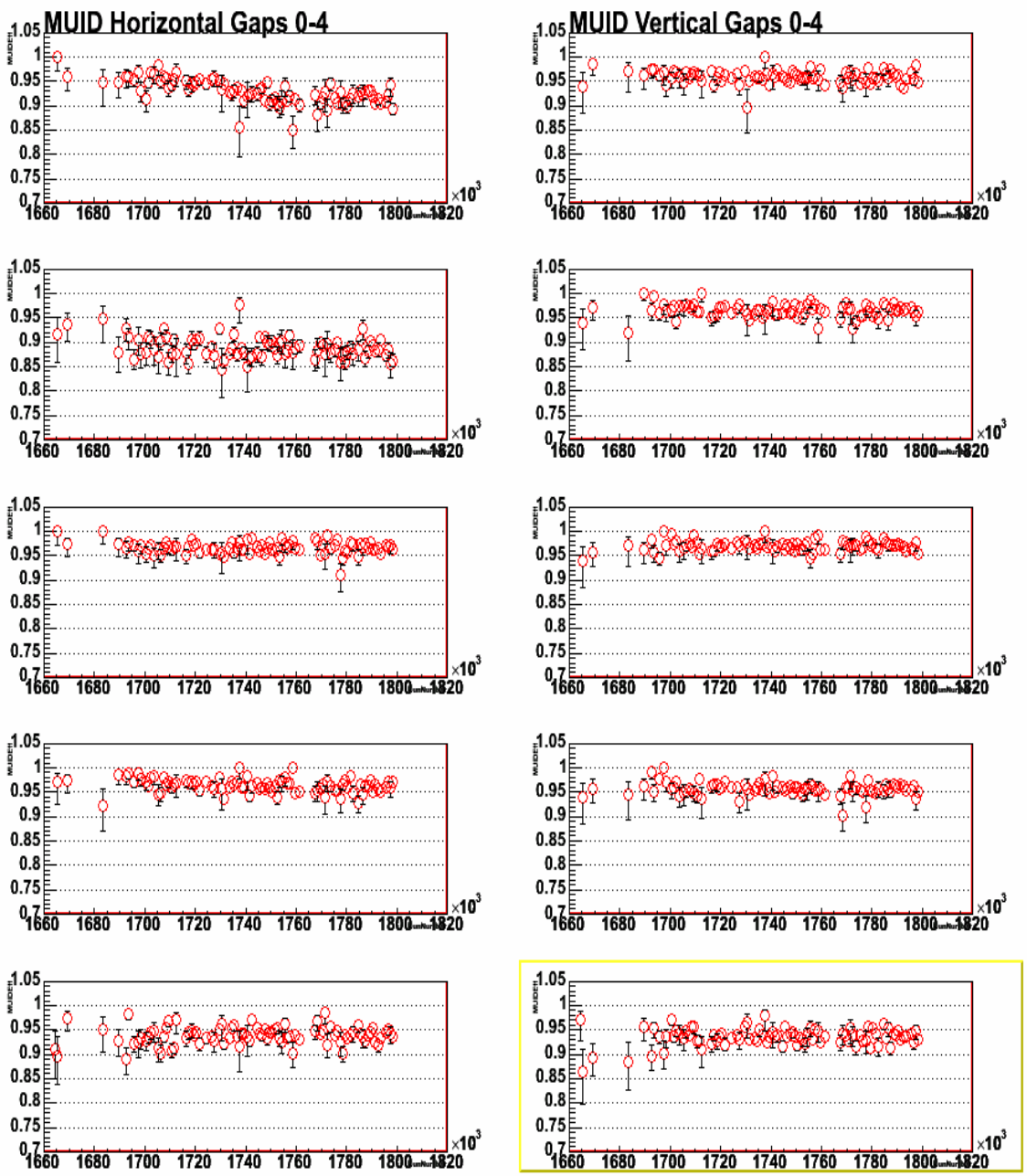


Figure 2.1: Run-5 Proton-Proton South Arm MUID efficiency as a function of run number. The ten plots are for gaps 0-4 (top to bottom) and horizontal and vertical orientations (left and right respectively).

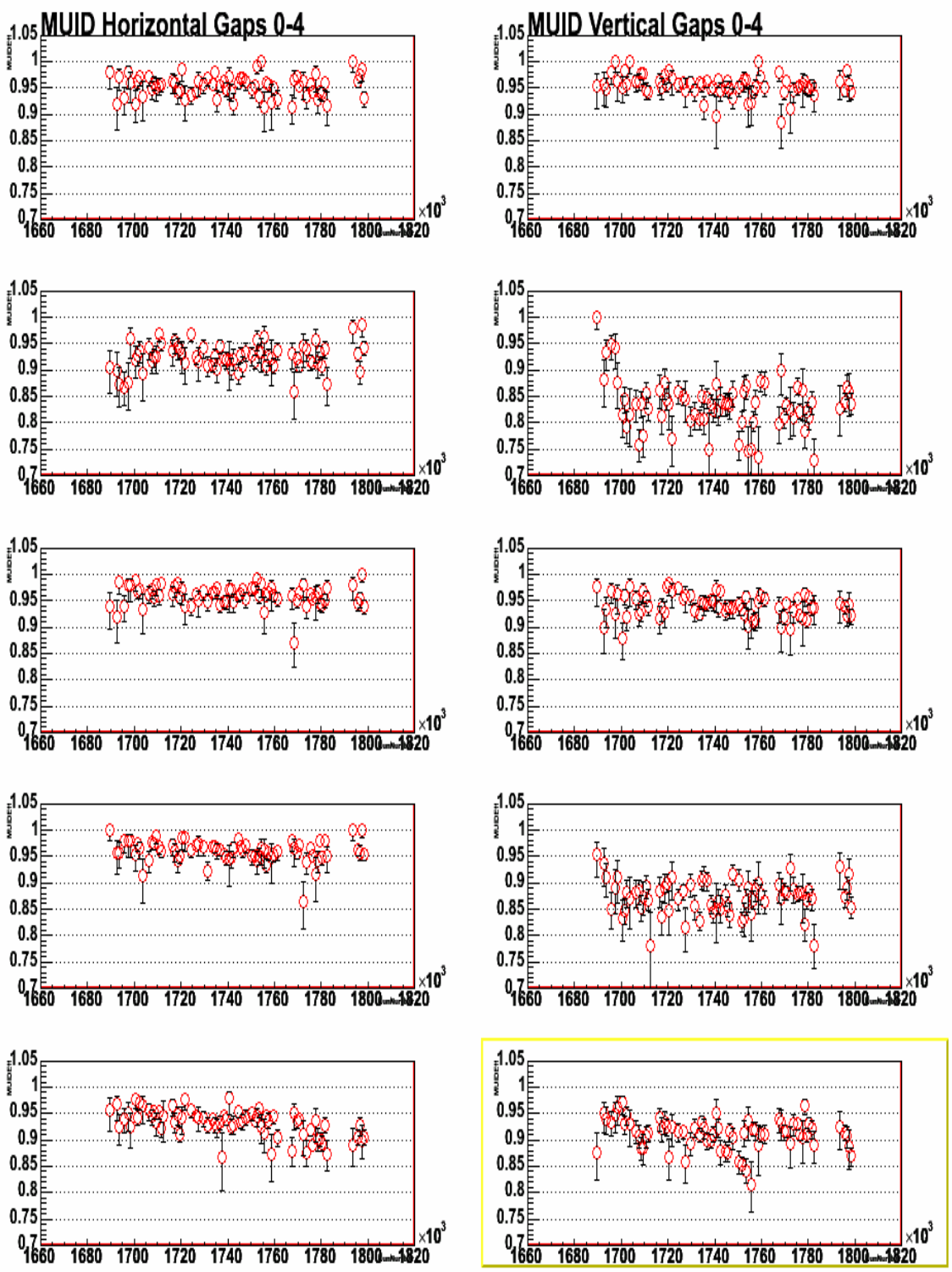


Figure 2.2: Run-5 Proton-Proton North Arm MUID efficiency as a function of run number. The ten plots are for gaps 0-4 (top to bottom) and horizontal and vertical orientations (left and right respectively).

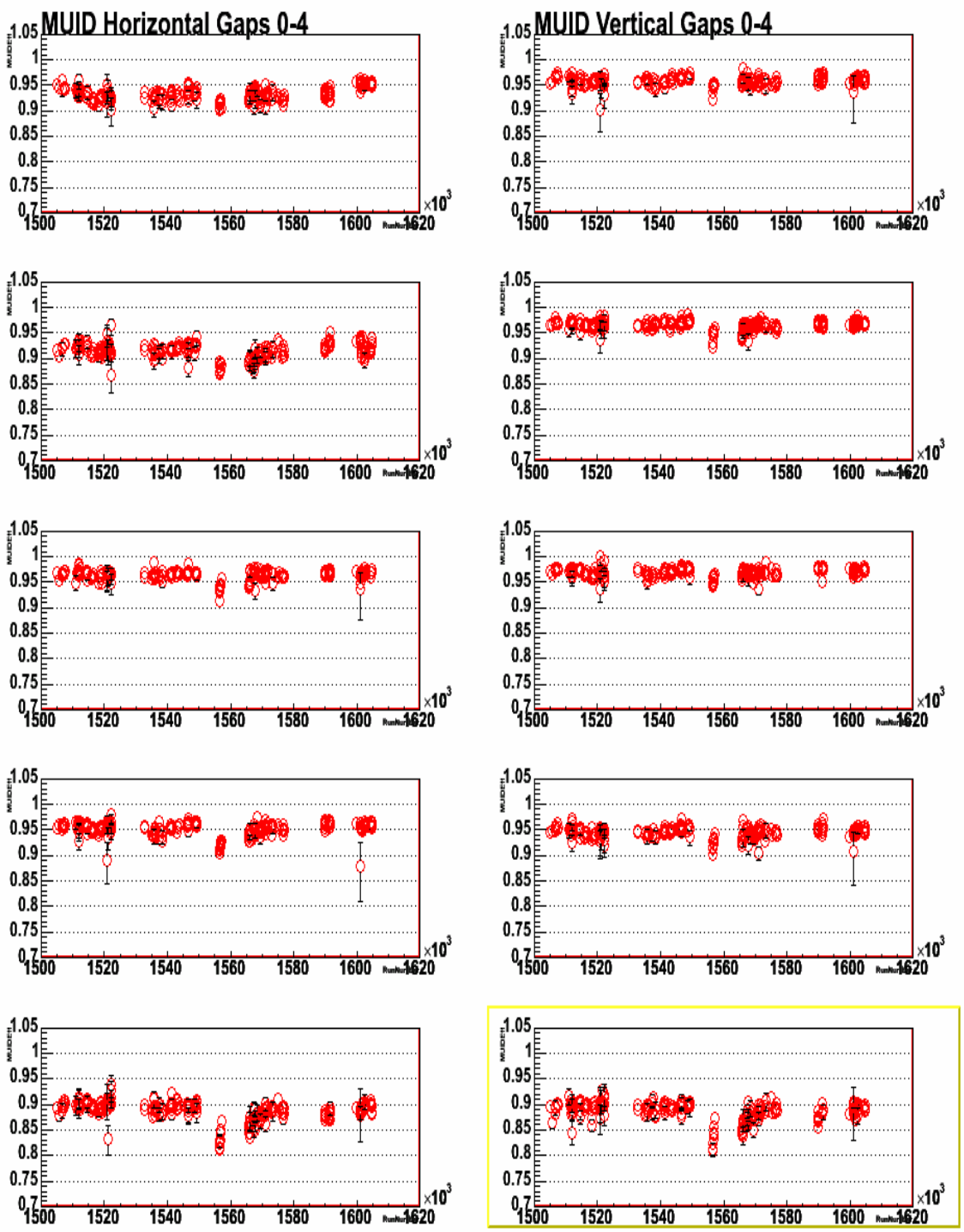


Figure 2.3: Run-5 Copper-Copper 200 GeV South Arm MUID efficiency as a function of run number. The ten plots are for gaps 0-4 (top to bottom) and horizontal and vertical orientations (left and right respectively).

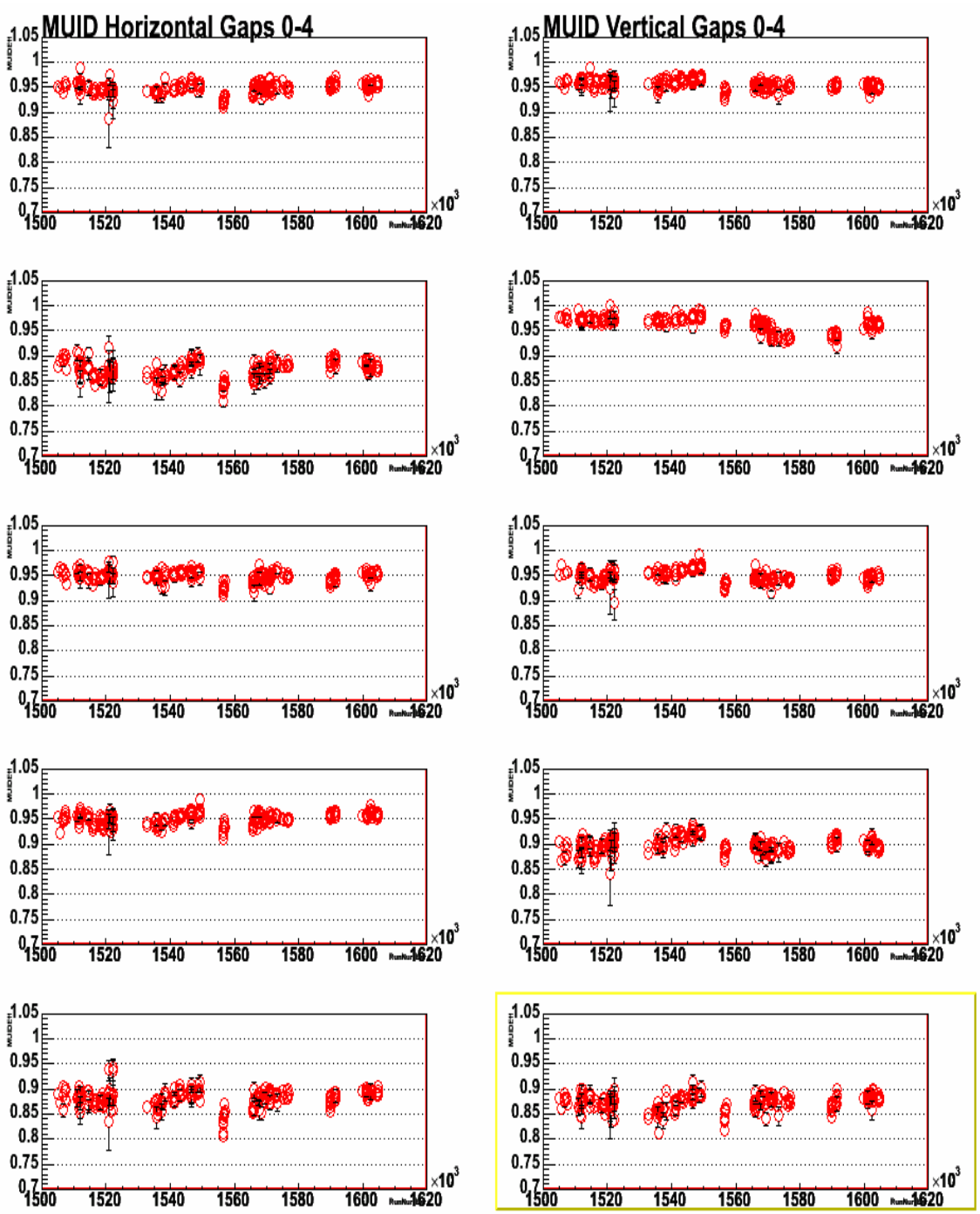


Figure 2.4: Run-5 Copper-Copper 200 GeV North Arm MUID efficiency as a function of run number. The ten plots are for gaps 0-4 (top to bottom) and horizontal and vertical orientations (left and right respectively).

We observe no significant run dependencies in either the Copper-Copper or Proton-Proton data samples. There is a slight change in efficiency in Run-5 Proton-Proton south arm, gap 0 horizontal in the middle of the running period of order < 2 %. There is also a slight change at the beginning of the run period in the Run-5 Proton-Proton north arm, gap 1, vertical. These changes are very minor and are thus not considered for separate MUID efficiency files.

Data Method and HV Method Results

We determine following the methods outlined in Appendix B (DataMethod) and Appendix C (HVMethod) the efficiency values averaged over the entire running period since we observe no significant run dependence. Note that for the DataMethod, we nominally should weight the efficiency statistics for each run by the integrated luminosity for that run. However, since there is negligible run dependence, we simply add the statistics from all runs together, thus weighting the runs by the number of reconstructed tracks and roads that enter the MUID efficiency ntuples.

In the following sequence of plots, we show the comparisons between the values from the Data method and HV method. These plots show the efficiency values (Y-axis, in %) for both methods and all HV groups, as outlined in the legend. The panels on the plots again correspond to the real MUID panels and the X-axis to the MUID gaps (0-4). We make separate plots for each arm (South, North) and orientation (Horizontal, Vertical), i.e. a total of four plots per dataset.

The plots are all also available on the web at:

https://www.phenix.bnl.gov/phenix/WWW/publish/silvermy/muon/muid_hv/run5{pp,CuCu}/all/ploteff-run_0-{0,1}-{0,1}.gif

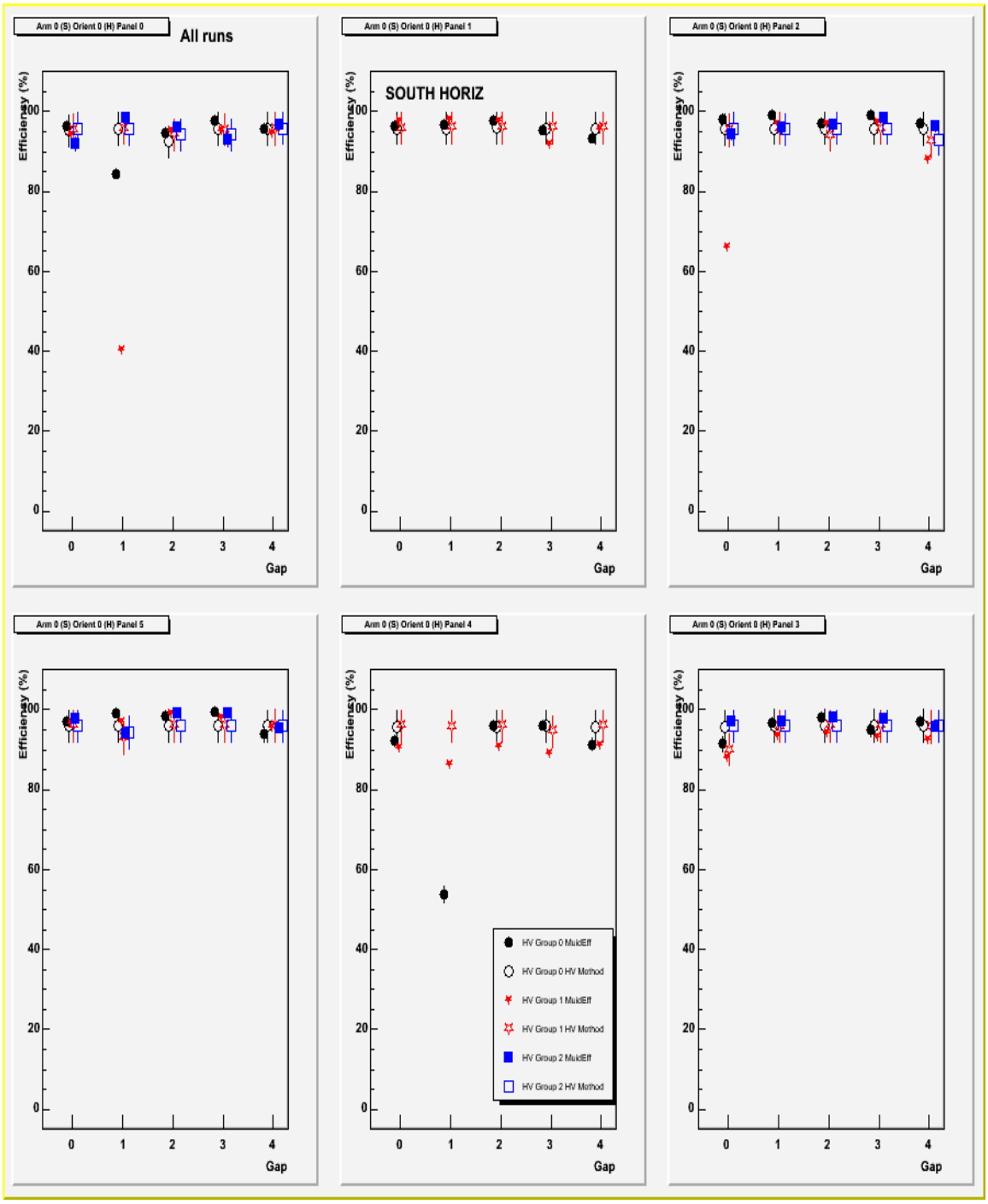


Figure 2.5: Run-5 Proton-Proton South Arm Horizontal MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

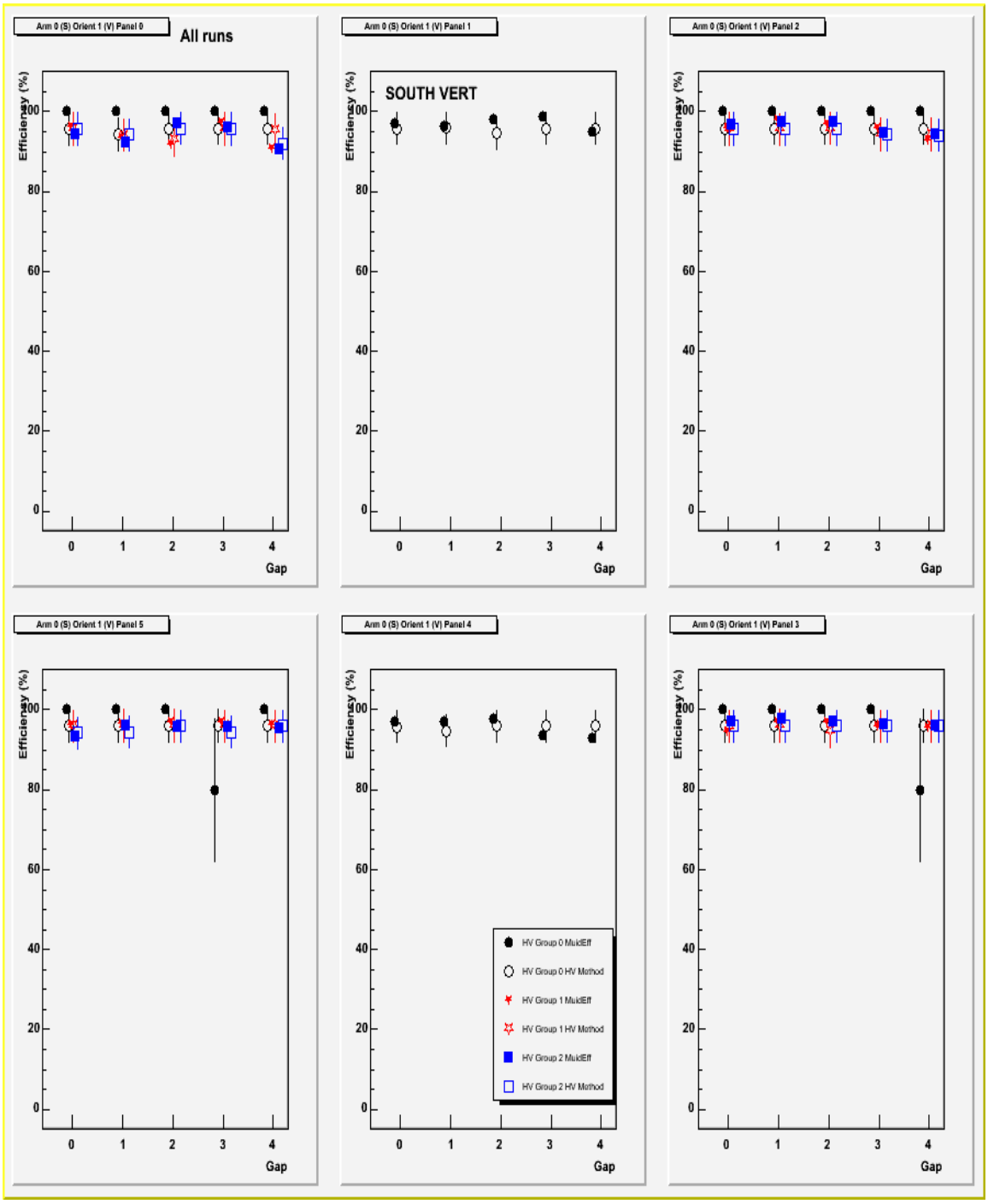


Figure 2.6: Run-5 Proton-Proton South Arm Vertical MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

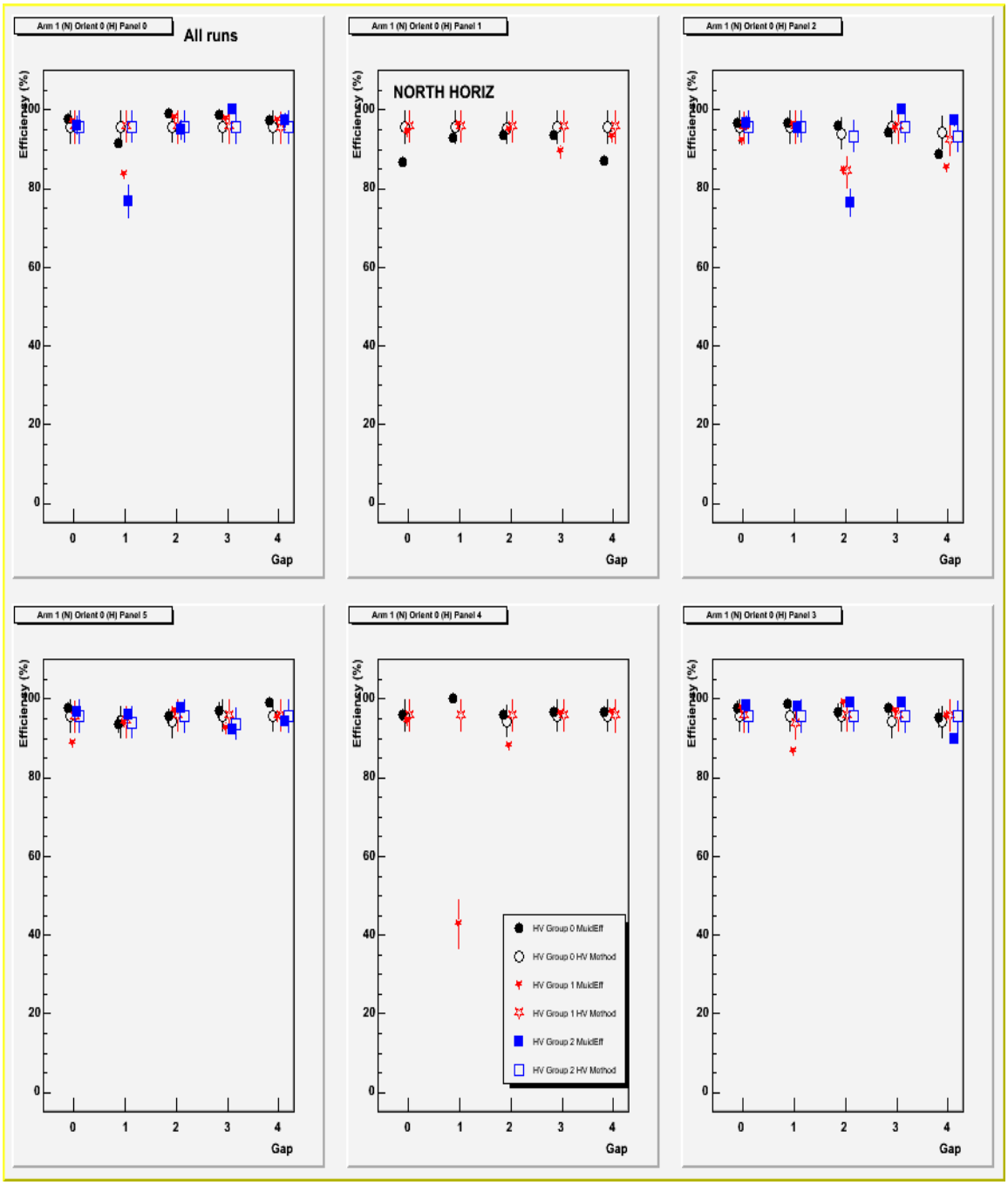


Figure 2.7: Run-5 Proton-Proton North Arm Horizontal MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

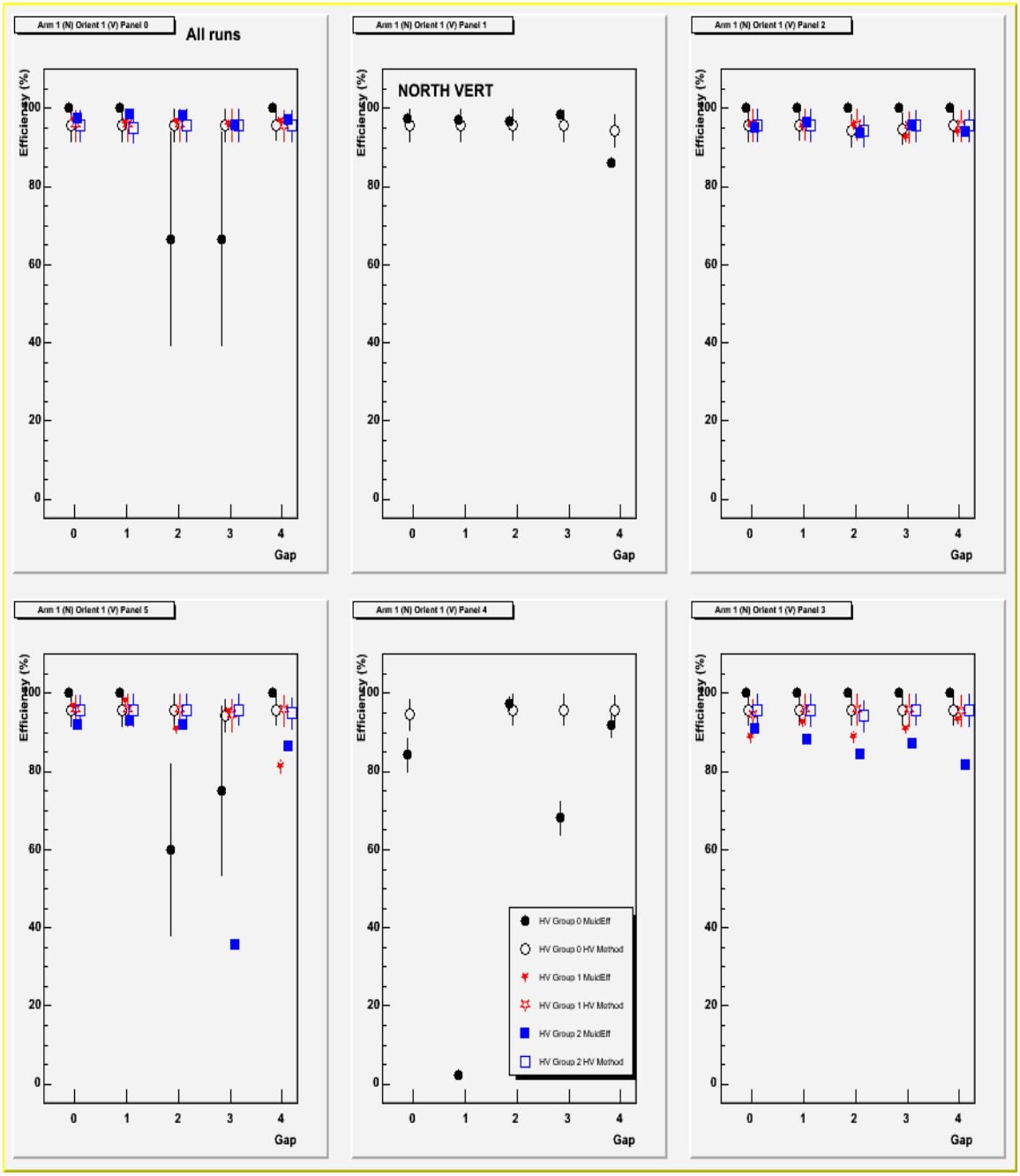


Figure 2.8: Run-5 Proton-Proton North Arm Vertical MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

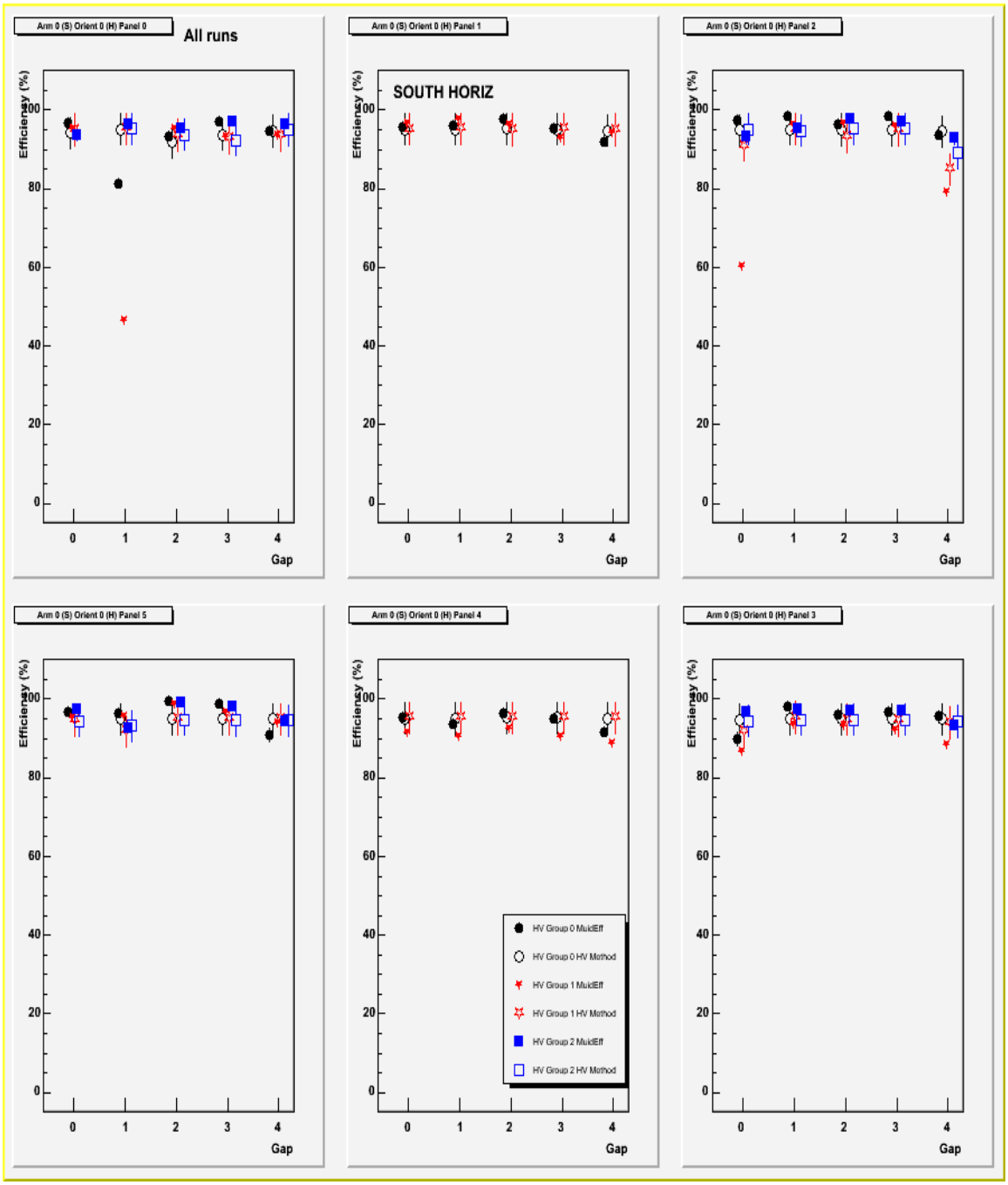


Figure 2.9: Run-5 Copper-Copper South Arm Horizontal MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

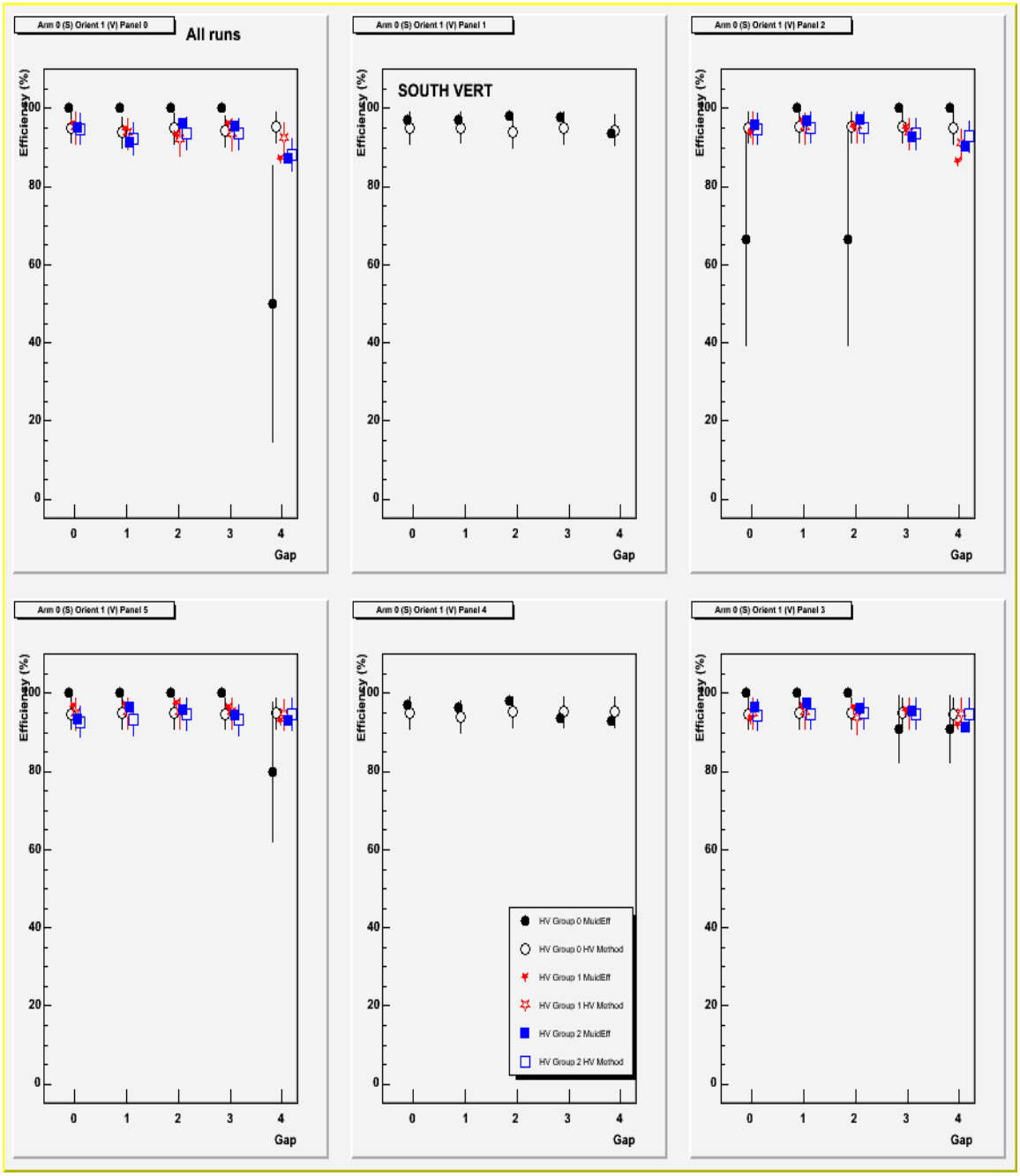


Figure 2.10: Run-5 Copper-Copper South Arm Vertical MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

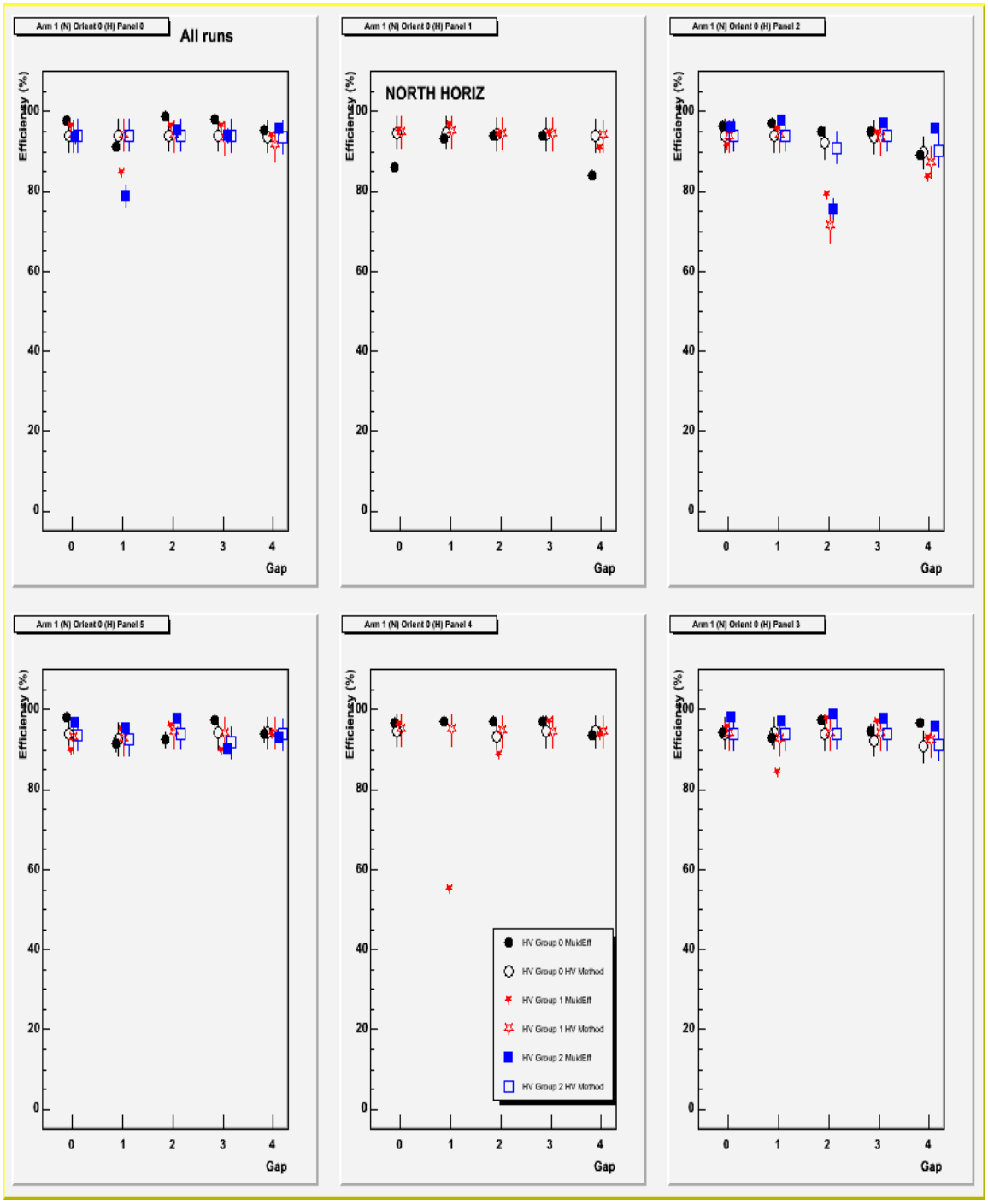


Figure 2.11: Run-5 Copper-Copper North Arm Horizontal MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

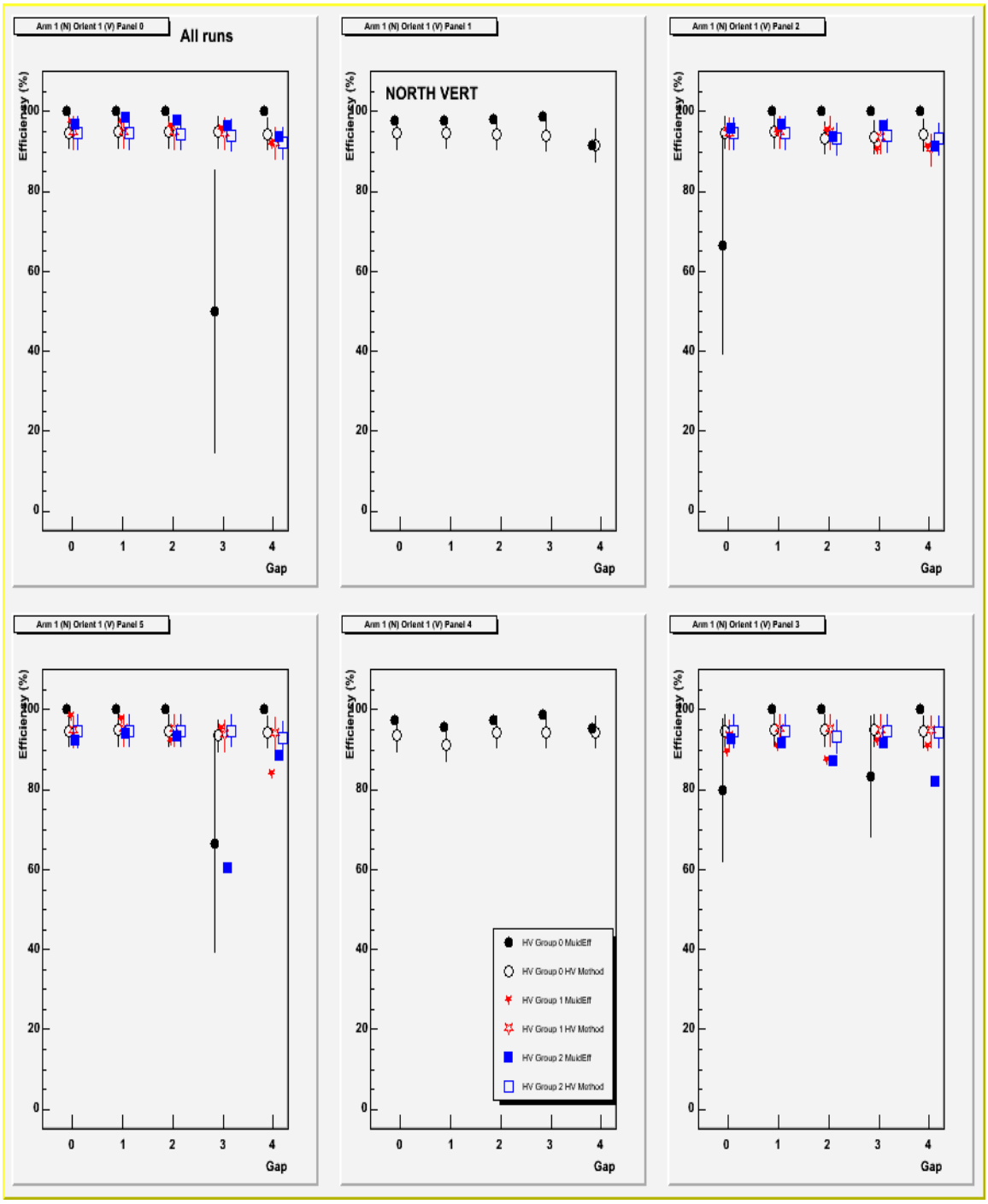


Figure 2.12: Run-5 Copper-Copper South Arm Vertical MUID Efficiencies from DataMethod and HVMethod for the six panels, different HV groups and as a function (x-axis) of gap number.

Overall the efficiencies are quite good (typical values in the range of 96%). Qualitatively there appears to be good agreement between the two methods. There are a handful of HV chains where the Data method gives a low efficiency, but the HV method does not. This may be due to a low Voltage problem (for example) which is not caught by the HV method. Additionally, there are a set of low occupancy HV chains for which the Data method has large errors. The errors shown on the plots are binomial and give an indication of these chains. We have also compared the exact list of low occupancy chains as labeled in the MUID diagram in Appendix A.

For the vast majority of chains where the agreement is quite good, we can quantify the exact agreement between the methods. We have compared the difference between the values from the Data method and HV method. These plots show the efficiency values difference (X-axis) distribution. A value of 1 here means 100% efficiency difference between the methods. The panels on the plots now correspond to the MUID gaps (0-4). We make separate plots for each arm (South, North), i.e. a total of 2 plots per dataset.

The plots are all available under:
https://www.phenix.bnl.gov/phenix/WWW/publish/silvermy/muon/muid_hv/run5{pp,CuCu}/all/compare_{south, north}.gif

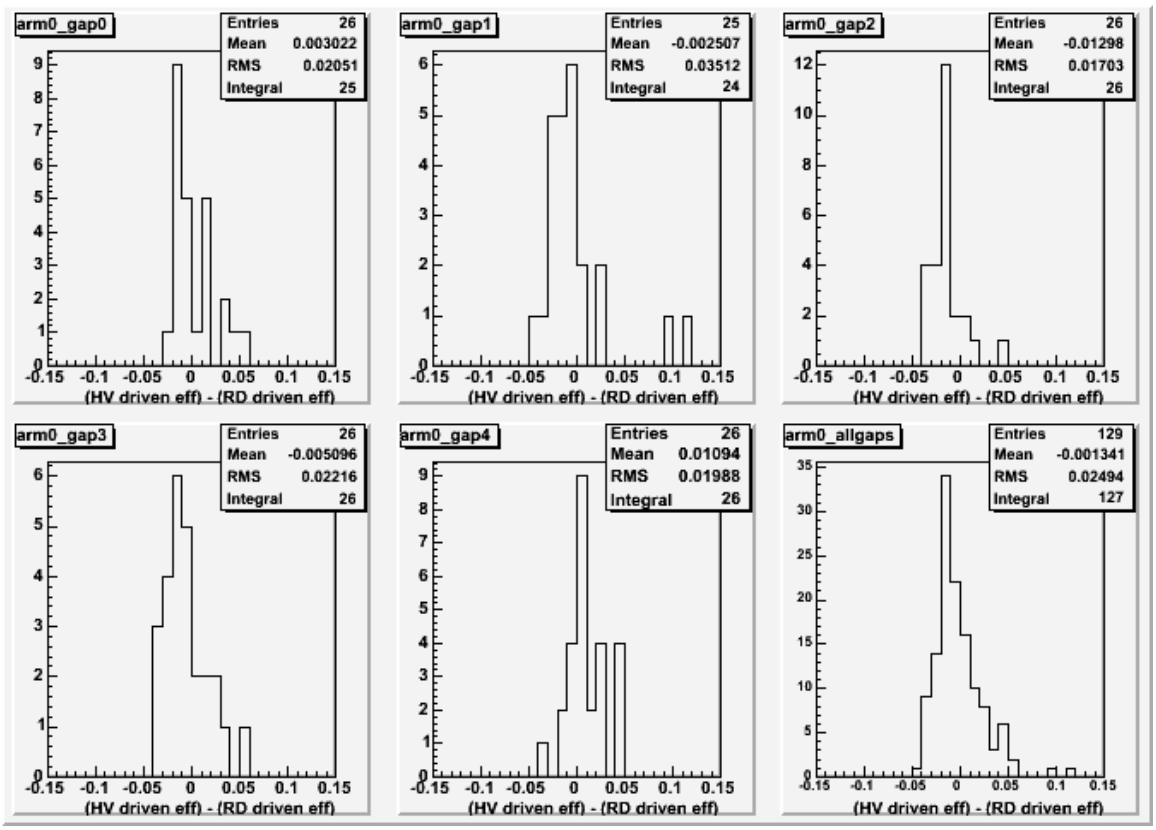


Figure 2.13: Run-5 Proton-Proton South Arm HV Method efficiency (as a fraction 0-1) minus the Data Method efficiency (as a fraction 0-1). The panels are gaps 0, 1, 2 from left to right on the top row, and gaps 3, 4 and all gaps from left to right on the bottom row. Each entry is a different HV group corresponding to that gap.

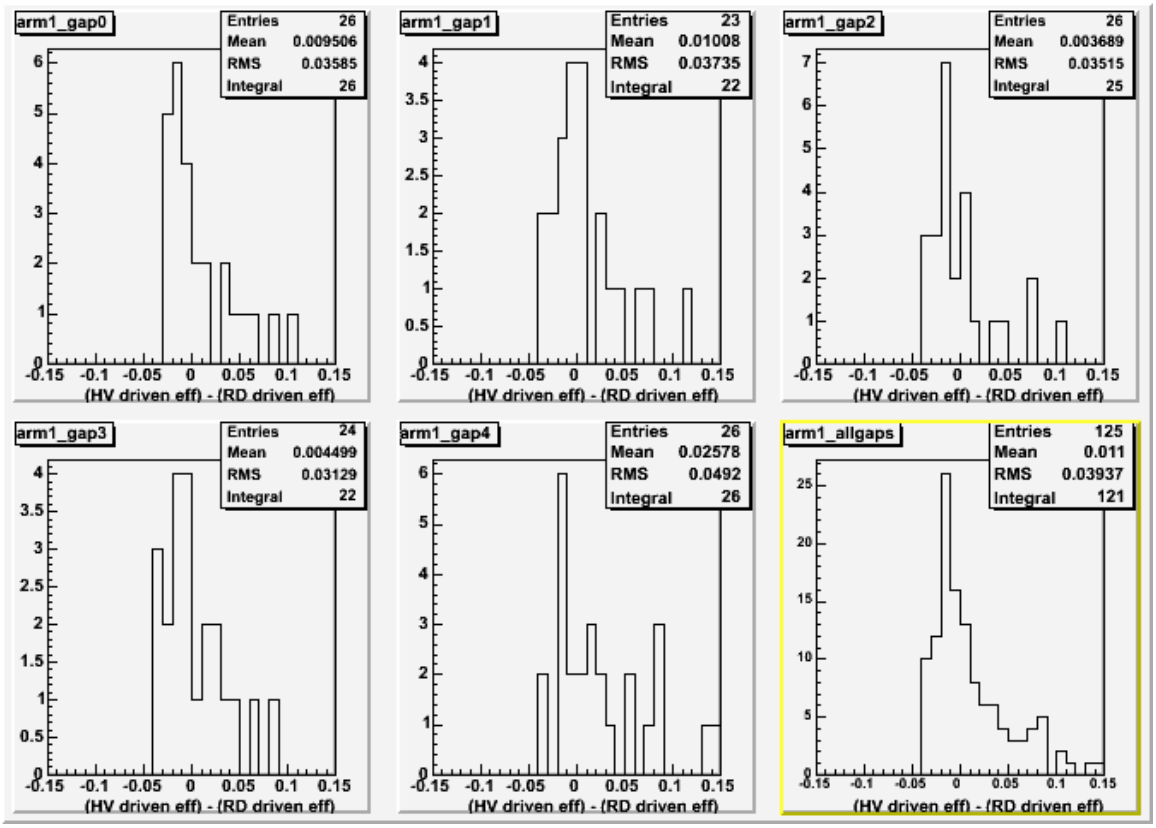


Figure 2.14: Run-5 Proton-Proton North Arm HV Method efficiency (as a fraction 0-1) minus the Data Method efficiency (as a fraction 0-1). The panels are gaps 0, 1, 2 from left to right on the top row, and gaps 3, 4 and all gaps from left to right on the bottom row. Each entry is a different HV group corresponding to that gap.

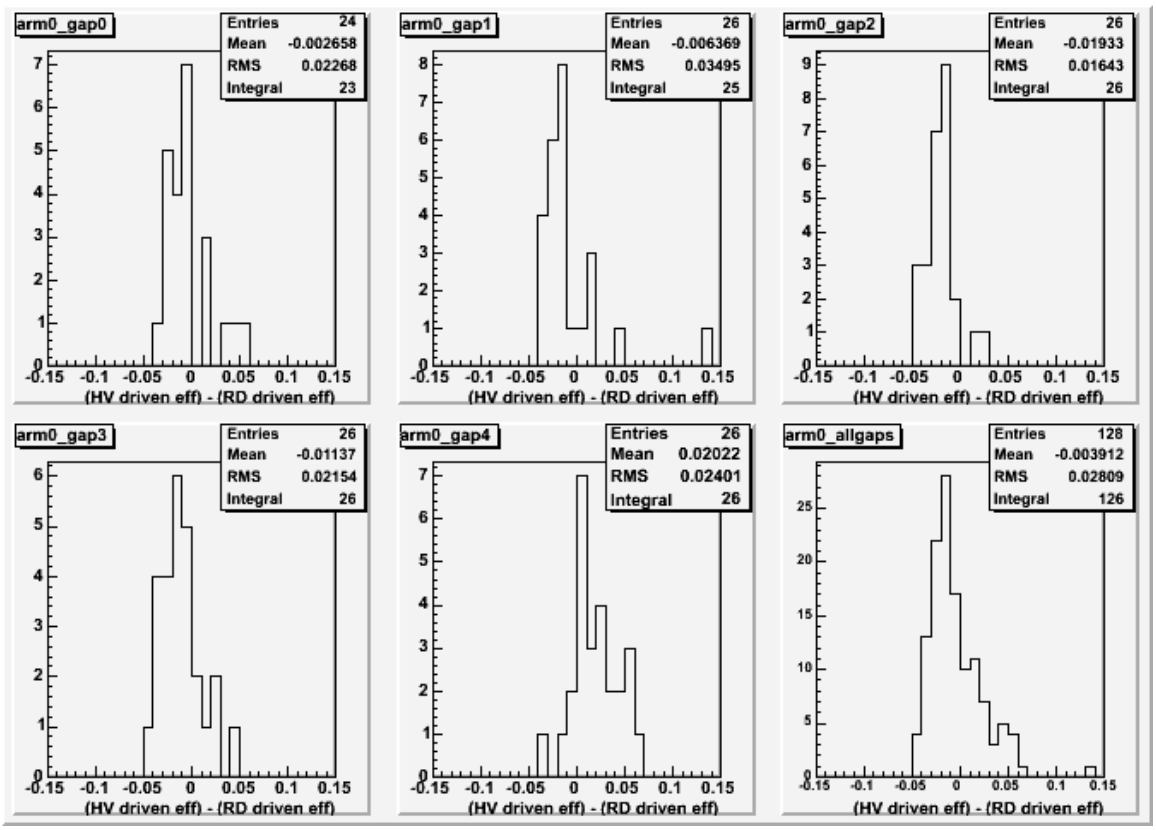


Figure 2.15: Run-5 Copper-Copper South Arm HV Method efficiency (as a fraction 0-1) minus the Data Method efficiency (as a fraction 0-1). The panels are gaps 0, 1, 2 from left to right on the top row, and gaps 3, 4 and all gaps from left to right on the bottom row. Each entry is a different HV group corresponding to that gap.

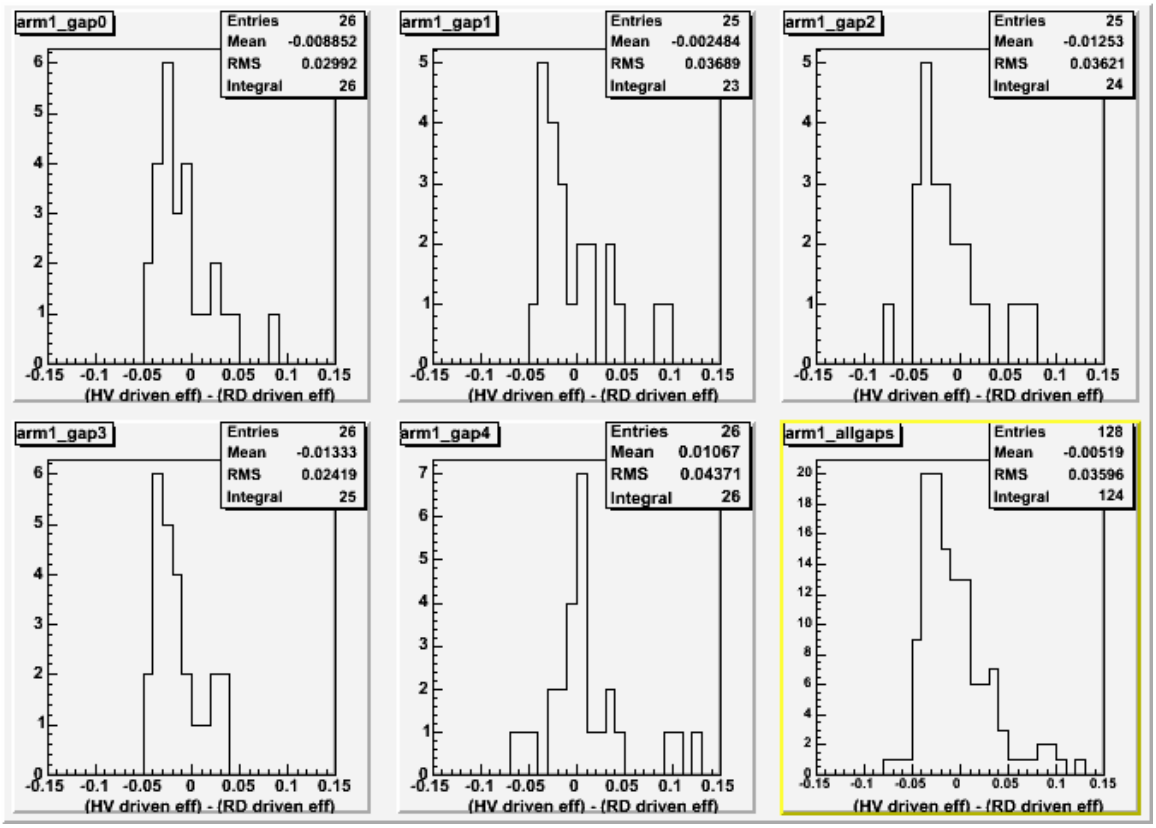


Figure 2.16: Run-5 Copper-Copper North Arm HV Method efficiency (as a fraction 0-1) minus the Data Method efficiency (as a fraction 0-1). The panels are gaps 0, 1, 2 from left to right on the top row, and gaps 3, 4 and all gaps from left to right on the bottom row. Each entry is a different HV group corresponding to that gap.

Note that the mean value differences in the plots above are typically less than 1%. A few gaps show a difference of order 2%, but this is often due to one or two HV groups where the HV method gives a higher result, which may be due to periodic low voltage problems or other issues. Overall the agreement is quite good.

Random Benefit Check

In this section we compare the values from Run-5 Copper-Copper to Run-5 Proton-Proton for both methods as a means to gauge the possible influence of random benefit. Similarly to the previous section, we require that the HV groups included in the comparison plots should have absolute efficiency errors larger than 0% and less than 100%, to exclude low statistics and poorly determined efficiencies. The errors displayed in the earlier plots are all binomial, i.e. if we label the efficiency p, and the number of trials N, the error is given by Error = sqrt(p*(1-p) / N). So, if we only had 2 trials and both were successful we'd currently assign an efficiency value of 1 (100%) and an error of 0. So, this cut is just meant to remove such groups.

Note that for the Data method, as described in Appendix B, we have applied a peripheral event selection for Copper-Copper to reduce the influence of random hits that get associated with the road. All the results shown previously and in this comparison are for this specific selection.

The plots are all available under:
https://www.phenix.bnl.gov/phenix/WWW/publish/silvermy/muon/muid_hv/comparison/CuCu-pp//compare_{data,hv}_{south, north}.gif

Similar plots for the difference between Run-4 Gold-Gold and Run-5 Proton-Proton are available in a parallel directory (AuAu-pp) but not included in this note. We include the full +/- 1 scale in efficiency difference to show (and have the y-axis in log-scale) that there is not much grass around the central part of ~0% difference. One can see that there is not that much of difference between the Copper-Copper and Proton-Proton results.

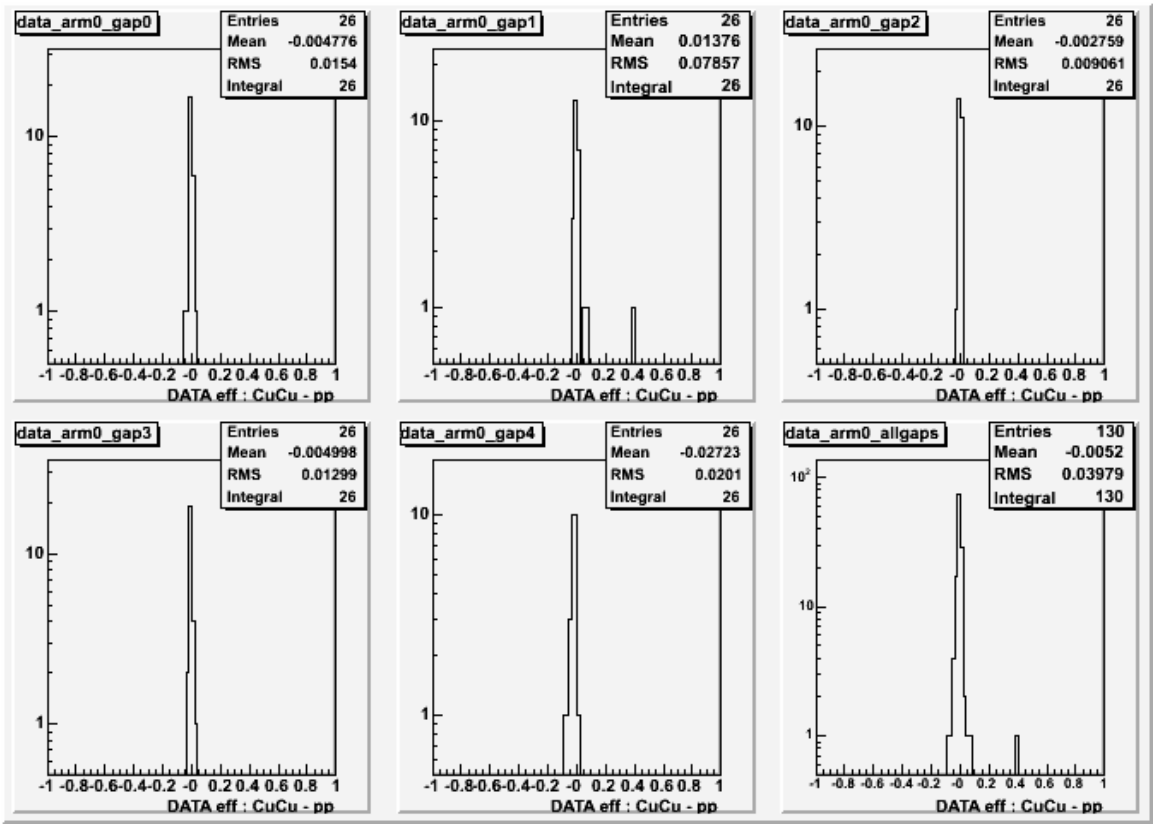


Figure 2.17: Run-5 South Arm Copper-Copper MUID efficiency from the Data method minus the equivalent for Proton-Proton.

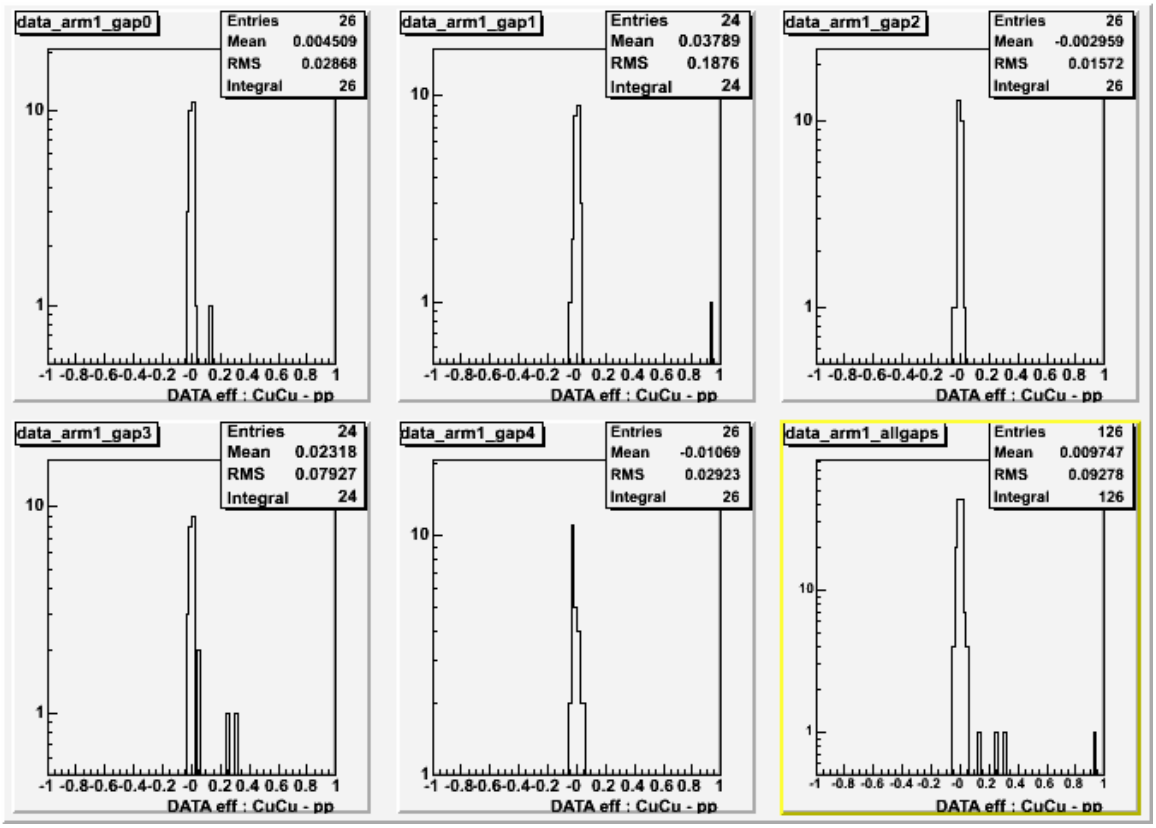


Figure 2.18: Run-5 North Arm Copper-Copper MUID efficiency from the Data method minus the equivalent for Proton-Proton.

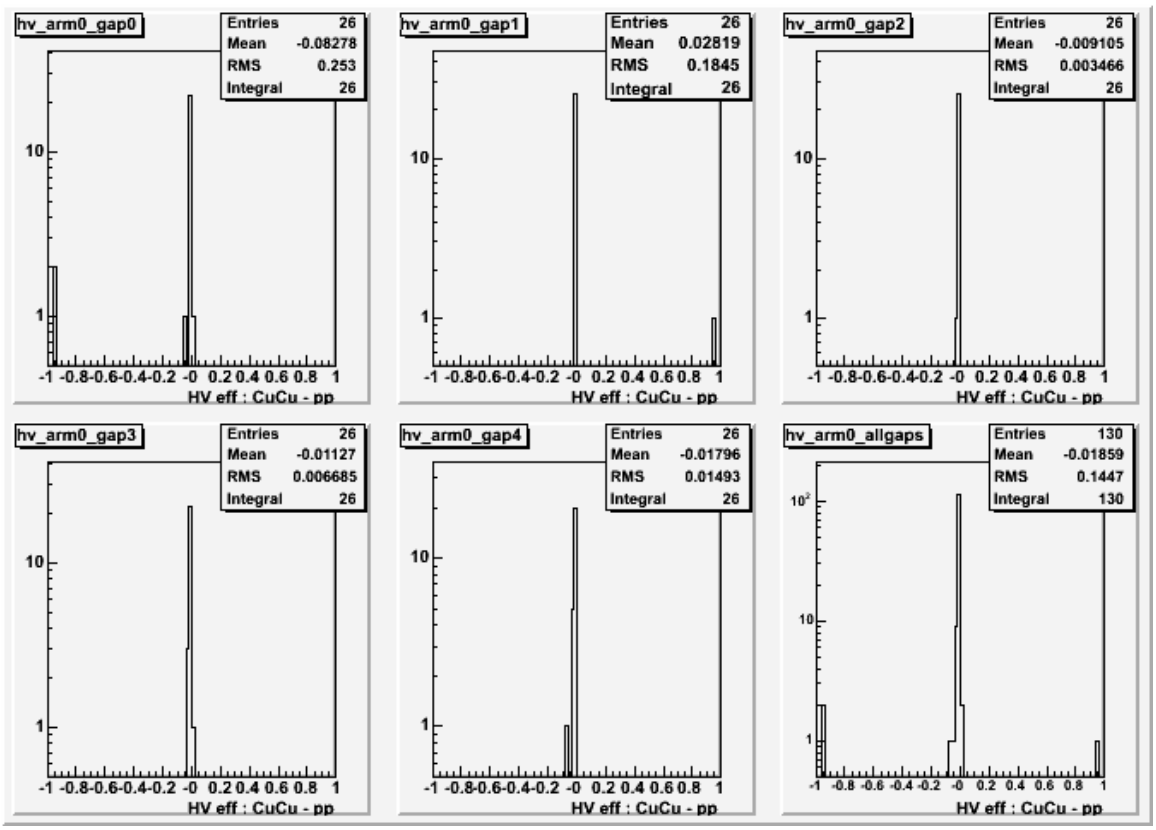


Figure 2.19: Run-5 South Arm Copper-Copper MUID efficiency from the HV method minus the equivalent for Proton-Proton.

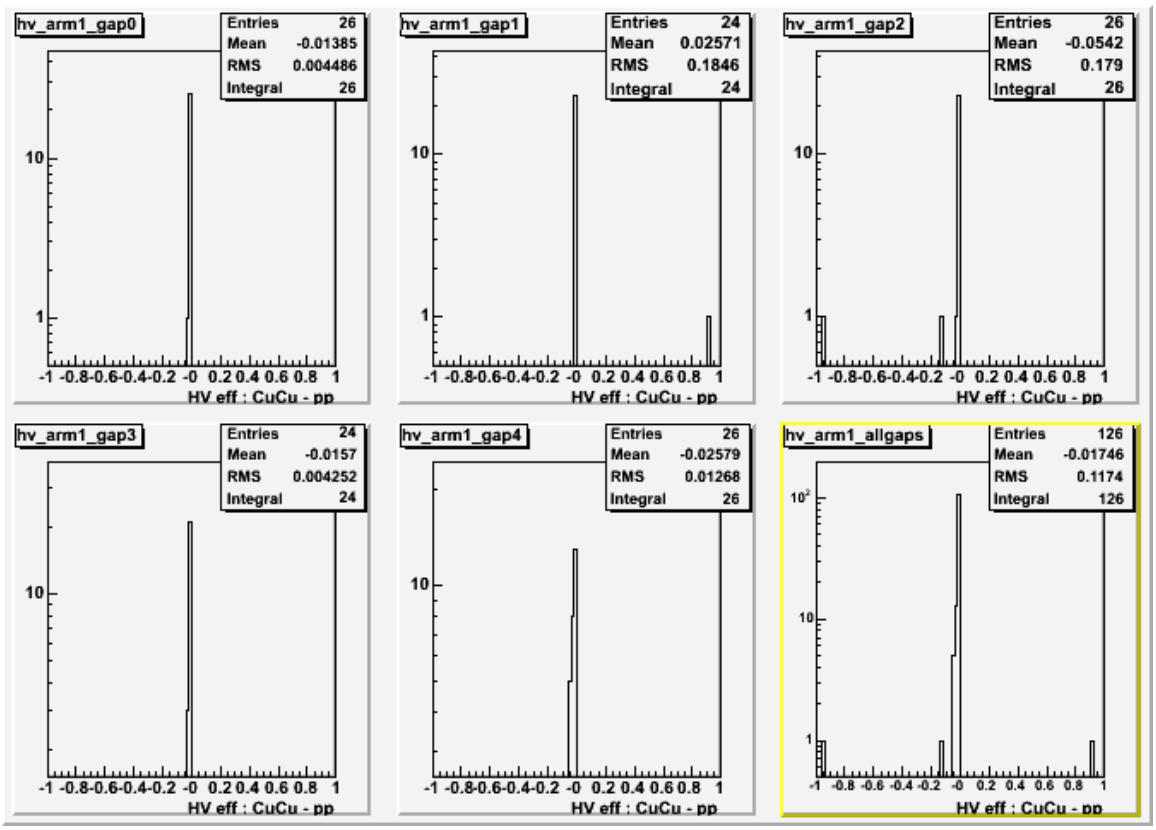


Figure 2.20: Run-5 North Arm Copper-Copper MUID efficiency from the HV method minus the equivalent for Proton-Proton.

3. Final Results and Systematic Errors

We have chosen to use the Data method for determining the final MUID efficiencies since overall the two methods are in good agreement except for a handful of HV groups which may have issues such as low Voltage problems that the HV method does not catch.

As detailed in Appendix C, we determine a list of low occupancy HV chains – which is of order 40 chains corresponding to the expected low occupancy regions of the detector shown in Appendix A. We then use the HV method for these chains only. Details of this step are provided in the HVMethod/README file.

After merging the files as above, we then apply the overlap correction described in Appendix E. Then we convert the efficiency file as a function of HVgroup to the proper MC input format as a function of two-pack. This step is done using
offline/analysis/muideff/macros/DataMethod/GenMuidEfficTwoPack.C

The final MUID efficiency results have been committed to CVS on a HVgroup basis for ease of plotting and analyzing, and also on a two-pack basis as direct input for the PHENIX Monte Carlo simulations. They are located at:

offline/analysis/muideff/results:

Note that since the Run-5 Copper-Copper 200 GeV and Run-5 Proton-Proton results agree very well, for the Run-5 Copper-Copper 62 GeV one can simply use the 200 GeV MUID efficiency files.

We also need to quote a systematic error on the efficiency results in order to quantify a final systematic error on the J/Psi or other muon physics topics invariant yield.

We consider four separate sources of systematic error that may contribute to our final result.

1. MC not doing the overlap correction properly.

This systematic error is quote as an RMS error +/- 0.7%. Details of this systematic error estimate are given in Appendix E.

2. Run Dependencies

Since we observe no significant run dependencies as detailed in this note, we consider this systematic to be << 1 % and is negligible compared with other systematic error sources.

3. Random benefit

We have compared the results of the HV method (which does not suffer from random benefit hits) and the Data method. We have also compared the Proton-Proton and Copper-Copper (utilizing peripheral events). In both cases the agreement is < 1%. Random benefit may cause extra hits to be found when the real detector did not fire on the particle of interest. This can lead to a slightly higher efficiency from the Data method. Also, one can have falsely extended roads where it then appears as an artificially low efficiency since you get a random gap 4 hit in vertical and thus often are missing the gap 4 horizontal. We have made additional cuts as detailed in Appendix B and find there is a negligible effect in Copper-Copper after the cuts are applied. We estimate this systematic as an RMS error +/- 0.5%.

4. HV comparison with Data Driven method

We find the methods agree at the 1% level overall. Being somewhat conservative, we apply an RMS error +/- 1% for both the Proton-Proton and Copper-Copper results. Note that this systematic error is somewhat redundant with the error found in item 3. Thus, we only include this contribution.

The final RMS systematic error for Run-5 Proton-Proton and Run-5 Copper-Copper is thus determined by the quadrature sum of 0.7% (item 1) and 1.0% (item 4). The final RMS systematic error is:

RMS systematic error = +/- 1.2%

We note two final items. For the J/Psi analysis, if we require 8 of 10 hits for both (two) muons and the average MUID efficiency is 96%, the J/Psi systematic error from the above quoted MUID efficiency RMS systematic error will only be of order RMS +/- 1%. Thus, this should be a very minor contribution to the final J/Psi systematic error.

Second, there has been some discussion of other methods to check the efficiency results. Some have looked at the hit distribution for roads in the north and south arms. However, one needs to have a clean sample of “good particles” to do this study in a controlled fashion. One such study is to use the filtered data sample and require 8 of 10 hits in both real data and MC. One can then also require 9 of 10 hits in both real data and MC. In the second case, the real data J/Psi counts is reduced, the efficiency should be reduced the same amount, and thus the yield should be the same within errors. This check was made for Run-4 Gold-Gold and appears consistent.

Appendix A: MUID HV Chain and Panel Geometry

Each gap of the MUID is composed of two layers of two-packs with different orientation (horizontal and vertical). These are grouped into six panels. For the purposes of this document, the convention for numbering the MUID panels as used in our display of DataMethod and HVMethod efficiencies is shown below (as viewed from the South). Note that this is not in every place the PHENIX standard convention.

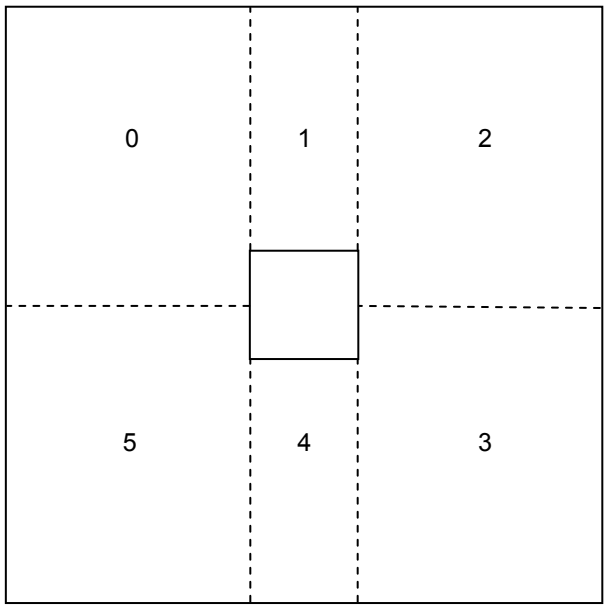


Figure A1: MUID Panel Numbering Scheme from collision vertex view.

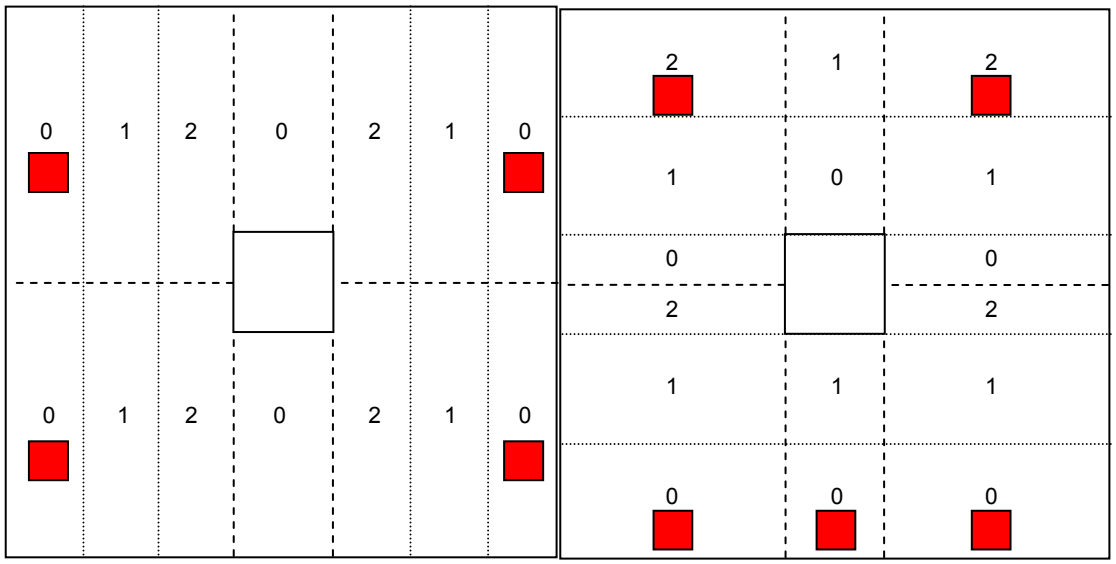


Figure A2: The left panel shows the numbering scheme for the vertically oriented tubes for the HV Chains (numbers 0-2) for each panel. The right panel shows the numbering scheme for the horizontally oriented tubes for the HV Chains (numbers 0-2) for each panel. The red squares indicate which HV Chains are almost completely out of the acceptance for good vertex originating tracks in the MUTR and MUID combined. Thus, these chains are expected to have very low track statistics for determining the tube efficiencies.

Appendix B: DataMethod (Colorado) Details

When running over real data PRDF or DST we use the code in offline/packages/mutoo_subsysreco/MuidEffic.cxx and thus when registered properly run MuidEffic::process_event which then outputs a MUID Efficiency ntuple file. All the process_event method does is loop over reconstructed MuTR tracks and find the closest associated MUID road. Then it checks at each plane whether there is a MUID hit associated with the road and its position relative to the road fit and records this and some additional event information to the ntuples.

This means that only hits associated with the MUID road count in the efficiency. Note that the roadfinder has a 30 cm window from the projected road, but this is a bit complex since the fit is redone as each new hit is added.

Analysis Note 255 (associated with Run-3 preliminary results) has the previous documentation on the Colorado MUID efficiency method, though that note is quite incomplete on the efficiency calculation.

One runs on the ntuples with **MuidEffic::do_analysis** and for each road it checks that the road passes all offline road finding and all trigger requirements assuming the plane of interest is hit and then assuming the plane of interest is not yet. Note that sometimes an extra hit can cause a road to be lost.

All the macros for running this analysis method are in CVS in:

Offline/analysis/muideff/macros/DataMethod

Also, included in this directory is a README file which is included in this note for completeness. We then also include the piece of code from MuidEffic that outlines the exact cut selections for Run-5 Proton-Proton and Run-5 Copper-Copper data sets.

README

J.Nagle - MuidEffic Data Driven (Colorado) Method: How to Instructions

Last Updated 05-11-2006

---

1. When running a production job on PRDFF files, one needs to register the mutoo_subsysreco/MuidEffic class module to run MuidEffic::process_event. This will output the muid efficiency ntuple.

One can make a separate ntuple for each run segment, for each run, or for all runs combined. In the Run-4 AuAu case, from stripe-1, the ntuples were made for each run segment. MinJung then combined these segment files into a single file for each run in the good run list. In the Run-5 CuCu and PP case, a single ntuple was made from the level-2 filtered files for all runs.

Below we outline the procedure for running on a single file over all runs, but then detail changes for the other case.

2. When analyzing the ntuples, one runs root -b -q offline/analysis/muideff/macros/DataMethod/GenMuidEffic.C. Below is an example csh script for running.

```
rm generic_goodrunlist_south.txt
rm generic_goodrunlist_north.txt
rm generic_muid_effic_ntuple.root
ln -sf runfiles_south/goodrun_$1.txt generic_goodrunlist_south.txt
ln -sf runfiles_south/goodrun_$1.txt generic_goodrunlist_north.txt
ln -sf run5_pp_muid_eff.root generic_muid_effic_ntuple.root
root -b -q 'GenMuidEffic.C(0,1)'
mv muid_south_hvgroup_eff.dat efffiles/muid_south_hvgroup_eff_$1.dat
mv muid_south_panel_eff.dat efffiles/muid_south_panel_eff_$1.dat
```

One needs to make soft links to the good run list files, and to the muid efficiency ntuple file. The two arguments of the macro are the arm (0=south, 1=north, 2=both) and the cut selection (1 = Run5pp, 2 = Run5CuCu, 3 = Run4AuAu). You can find details on the cut selections in MuidEffic::do_analysis().

If you want to use the one ntuple and output muid efficiency files for each run, you need to make a set of goodrun_$runnumber.txt files. Create a directory called runfiles_south and runfiles_north and the below commands generate the goodrun files.

```
awk '{print "echo "$1" > runfiles_south/goodrun_"$1".txt"}' run5_pp_goodrun_south.txt > create_singlerun_south_parsed.csh

awk '{print "echo "$1" > runfiles_north/goodrun_"$1".txt"}' run5_pp_goodrun_north.txt > create_singlerun_north_parsed.csh

source create_singlerun_south_parsed.csh
source create_singlerun_north_parsed.csh
```

Then you want to create a "super" csh script that calls for every run number the csh script above that calls GenMuidEffic.C. How to create the above run0_super_south.csh?

```
awk '{print "source run0_analyze_run5pp_south.csh " $1}' run5_pp_goodrun_south.txt > run0_super_south.csh

awk '{print "source run0_analyze_run5pp_north.csh " $1}' run5_pp_goodrun_north.txt > run0_super_north.csh
```

Sourcing these csh files will then create muid efficiency files for every run in the directory efffiles. There will be one by panel and one by hvgroup.

3. If you want to create one file with the results over all runs, just create a goodrun file with all run numbers listed and go from there.

4. If you want to then make plots of the run dependence, follow the below instructions. First, we want to modify all of the

efficiency files produced above by adding the run number as the first field in each line. Do this by following the procedure below.

a. Create a list of all the relevant files

```
ls efffiles/*.dat > tempfilelist
```

b. I then in emacs removed the ".dat" from the end of each file (there should be a way around this step?)

c. Then I parse with awk the tempfilelist and make a csh script of awk commands that puts the run number at the first entry of each file
and puts the output with an extension back in the same area.

```
awk -F_ '{ print "awk {print " $5 " $0 } " $0 ".dat > " $0 "_withrunnumber.dat" }' tempfilelist > create_output_withrunnumber.csh
```

emacs to add ' around {print} - must be a work around later

d. Then I just run the csh script

```
source create_output_withrunnumber.csh
```

e. Concatonate the relevant output files

```
ls efffiles/*north*panel*withrunnumber.dat > tempfilelist2
awk '{print "cat " $0 " >> fullfileout_muid_south_panel_eff_withrunnumber.dat"}' tempfilelist2 > create_concat.csh
source create_concat.csh
```

Now you have one giant ascii file with the efficiency values from each individual run.

You can plot the results using offline/analysis/muideff/macros/DataMethod/GenMuidEffic_PlotRunDep.C

The macro takes arguements of the input filename, minimum run number, maximum run number, runs per bin, and the low scale on the efficiency plots.

5. To create one efficiency file averaged over all runs there are two options.

If you have one ntuple file for all runs, you can just create a goodrun list including all runs and run the first part of the step 2 procedure. Easy as pie.

If you have many ntuples, one for each run, then run the entire above procedure first. Then you can run the following steps to add them all together and make a run averaged file. Just run the macro offline/analysis/muideff/macros/DataMethod/GenMuidEffic_RunAveraging.C and give the arguments of the arm and the input file name and pipe > the output to the desired filename. Note that you need to run this on the concatonated file (described above) with the runnumber inserted at the start and by hvgroup!

6. You can make various plots of the efficiencies by HVgroup using macros in the directory... We use the same file formats, so the plotting macros are the same.

offline/analysis/muideff/macros/HVMethod/

7. Our current procedure is to merge the hvgroup efficiency file determined from the DataMethod and then replace the efficiency values for very low occupancy hvchains. Check the macros/HVMethod for this description.

Then we apply an overlap tube correction using the script GenMuidEfficOverlapCorrect.prl on the HVGroup files. Then we generate the two pack efficiency files.

8. The MC takes a input an efficiency file by two-pack. To generate this file from the efficiency file by HVgroup, follow these instructions.

```
void GenMuidEfficTwoPack(const char *input_file="HVchainbase_eff.txt", char *output_file="twopackbase_eff.txt")
```

Run this macro in root. The only issue is that the macro needs to instantiate the muid geometry object. It does this by reading one event of a PRDFF file. Thus, you need to create a soft link ln -sf "----" generic.prdff Thus, you need to find a real PRDFF file to run this step. We should find a workaround for this.

CUT SELECTION CODE FROM MUIDEFF.cxx

```
    if (_cutselect == 1) { // FINAL CUT SELECTION FOR RUN5 PP ANALYSIS

        // dg0 < 20 cm (just for general matching)
        if (! (road_track_distance < 20.0)) continue;
        // |pz(at station1)| > 1.3 GeV (to ensure large enough to penetrate MUID)
        if (! (fabs(pz) > 1.3)) continue;
        // require agreement of hit in other view in same plane
        if (planeindex==0) if (! (v0 > 0.01 && v0 < 5.0)) continue;  // hori gap 0
        if (planeindex==1) if (! (h0 > 0.01 && h0 < 5.0)) continue;  // vert gap 0
        if (planeindex==2) if (! (v1 > 0.01 && v1 < 5.0)) continue;  // hori gap 1
        if (planeindex==3) if (! (h1 > 0.01 && h1 < 5.0)) continue;  // vert gap 1
        if (planeindex==4) if (! (v2 > 0.01 && v2 < 5.0)) continue;  // hori gap 2
        if (planeindex==5) if (! (h2 > 0.01 && h2 < 5.0)) continue;  // vert gap 2
        if (planeindex==6) if (! (v3 > 0.01 && v3 < 5.0)) continue;  // hori gap 3
        if (planeindex==7) if (! (h3 > 0.01 && h3 < 5.0)) continue;  // vert gap 3
        if (planeindex==8) if (! (v4 > 0.01 && v4 < 5.0)) continue;  // hori gap 4
```

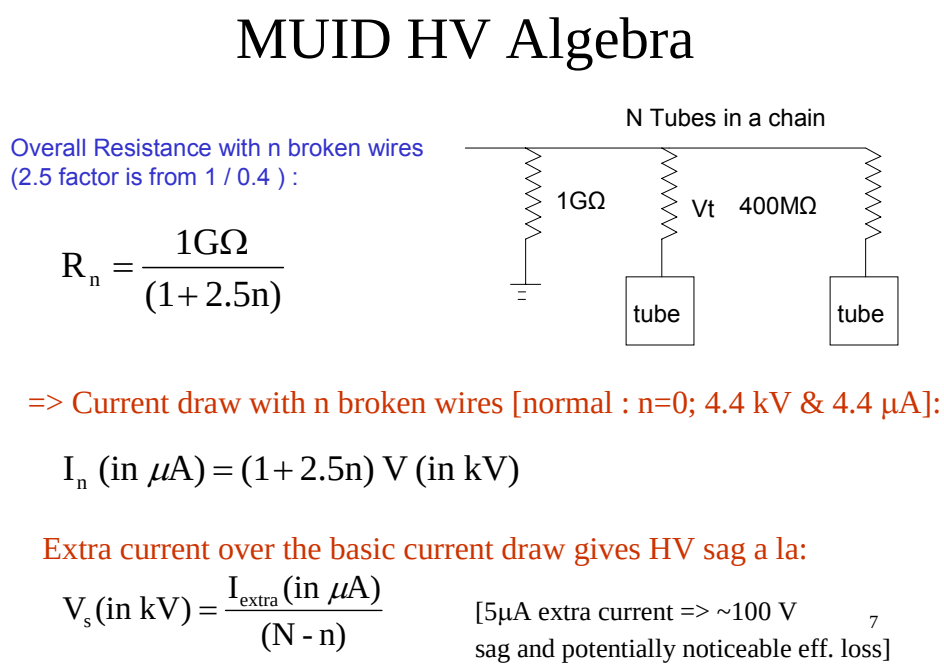
```
        if (planeindex==9) if (! (h4 > 0.01 && h4 < 5.0)) continue;  // vert
gap 4

    } else if (_cutselect == 2) { // FINAL CUT SELECTION FOR RUN5
CUCU ANALYSIS

        // select peripheral events based on BBC charge sum
        if     (!     (((bbc_ch_sum_0+bbc_ch_sum_1)     <     200)     &&
((bbc_ch_sum_0+bbc_ch_sum_1) > 20))) continue;
        // remove events with too many roads (nroads < 25)
        if (! (nroads < 25)) continue;
        // dg0 < 20 cm (just for general matching)
        if (! (road_track_distance < 20.0)) continue;
        // |pz(at station1)| > 1.3 GeV (to ensure large enough to
penetrate MUID)
        if (! (fabs(pz) > 1.3)) continue;
        // require agreement of hit in other view in same plane
        if (planeindex==0) if (! (v0 > 0.01 && v0 < 5.0)) continue;  // hori
gap 0
        if (planeindex==1) if (! (h0 > 0.01 && h0 < 5.0)) continue;  // vert
gap 0
        if (planeindex==2) if (! (v1 > 0.01 && v1 < 5.0)) continue;  // hori
gap 1
        if (planeindex==3) if (! (h1 > 0.01 && h1 < 5.0)) continue;  // vert
gap 1
        if (planeindex==4) if (! (v2 > 0.01 && v2 < 5.0)) continue;  // hori
gap 2
        if (planeindex==5) if (! (h2 > 0.01 && h2 < 5.0)) continue;  // vert
gap 2
        if (planeindex==6) if (! (v3 > 0.01 && v3 < 5.0)) continue;  // hori
gap 3
        if (planeindex==7) if (! (h3 > 0.01 && h3 < 5.0)) continue;  // vert
gap 3
        if (planeindex==8) if (! (v4 > 0.01 && v4 < 5.0)) continue;  // hori
gap 4
        if (planeindex==9) if (! (h4 > 0.01 && h4 < 5.0)) continue;  // vert
gap 4
```

Appendix C: HVMethod Details

Overview / Algebra

Each MUID channel or two-pack, consists (as the name suggests) of two individual Iarocci tubes. The two tubes are part of two different HV chains. Each HV chain typically serves 20 or so tubes. The two HV chains that serve the same two-packs make up what we will refer to as a HV 'group'. The large panels have 3 HV groups in either orientation while the smaller panels have 2 groups in the horizontal orientation and 1 in the vertical orientation.

The HV setup, and expected resistance and current algebra is shown in the following image, for a chain with N tubes, and n broken wires:



Efficiency scans with cosmics

We have estimated the efficiency for the chains using our cosmics trigger as a function of applied HV. An example (for the South Horizontal view) is shown in the following image.

The plot has 6 panels, following the MUID layout with the large panels in the corners, and up to 3 HV groups per panel. The X-axis is the applied HV in V and the Y-axis is the estimated efficiency. The points are fitted with a function (to asymptotically reach full eff as HV approaches infinity): par[0] * (TMath::TanH(par[1]*(x[0] - par[2])));

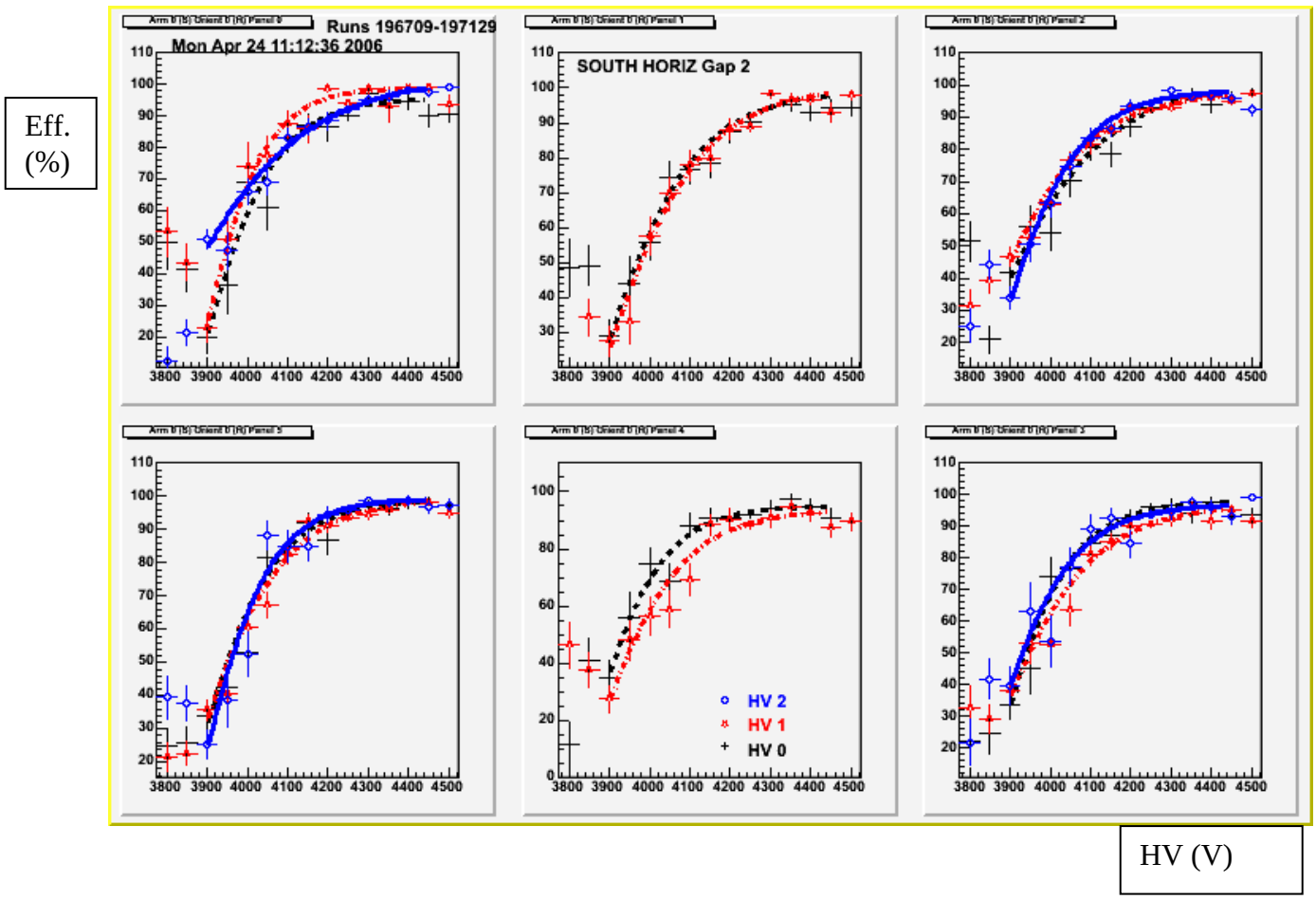


Figure C.1: MUID efficiency as a function of HV setting from Cosmic Ray test running.

The default MUID HV operating value is at 4400 V, i.e. safely on the plateau, but one can see that if one experiences a voltage sag of about 100 V (to 4300 V) , which corresponds to an extra current of about 5 □A, the loss in efficiency can be significant. Averaging the results from the fits to the response curves gives us a formula for how the efficiency E (in %) can be described as a function of voltage sag V_S (down from 4400 V):

$$E = 0.96\ (1 - 2.4e\text{-}6\ V_S^2)$$

Note that this equation is for efficiency as a fraction, but Figure C.1 is multiplied by 100 to be a percentage (0-100).

The first step is then to estimate the voltage sag from the current draw.

Current draw

The beam conditions and backgrounds can vary from run-to-run and store-to-store. The current draw can also deviate from the normal 4.4 A due to to broken wires and/or other problems. To see what the beam-related extra current is, we use a snapshot of the current values at the operating voltage but without any beam in the machine. We can then compare the calculated average current run-by-run with the subtracted current. The difference between the average (with beam) current, and the subtracted (no beam) current for the Run5 pp & CuCu and Run4 AuAu for all chain is shown in the images below.

Run5pp:

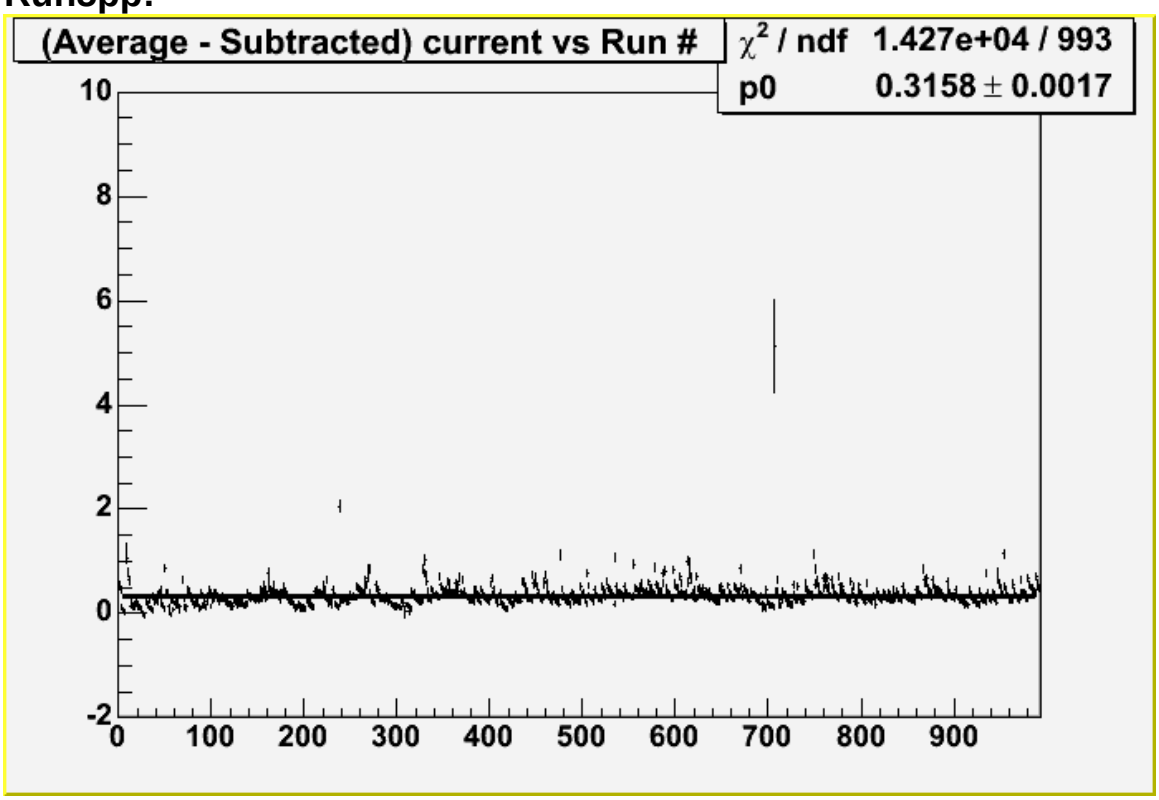


Figure C.2: Average Current Draw minus Non-Beam Current Draw as a function of run number for the Run-5 Proton-Proton data set.

Each bin corresponds to a particular run. Y-axis is the difference between the average and subtracted currents, in □A.
As one can see, the current draw is quite low and stable throughout Run5.

Run5CuCu:

For CuCu, the current draw is marginally higher, but not too much worse. The store-to-store fluctuations are more apparent, but not too large.

The AuAu case shows more scatter in the average currents, particularly towards the later parts of the Run. But the average current is still not too much higher than the no-beam current.

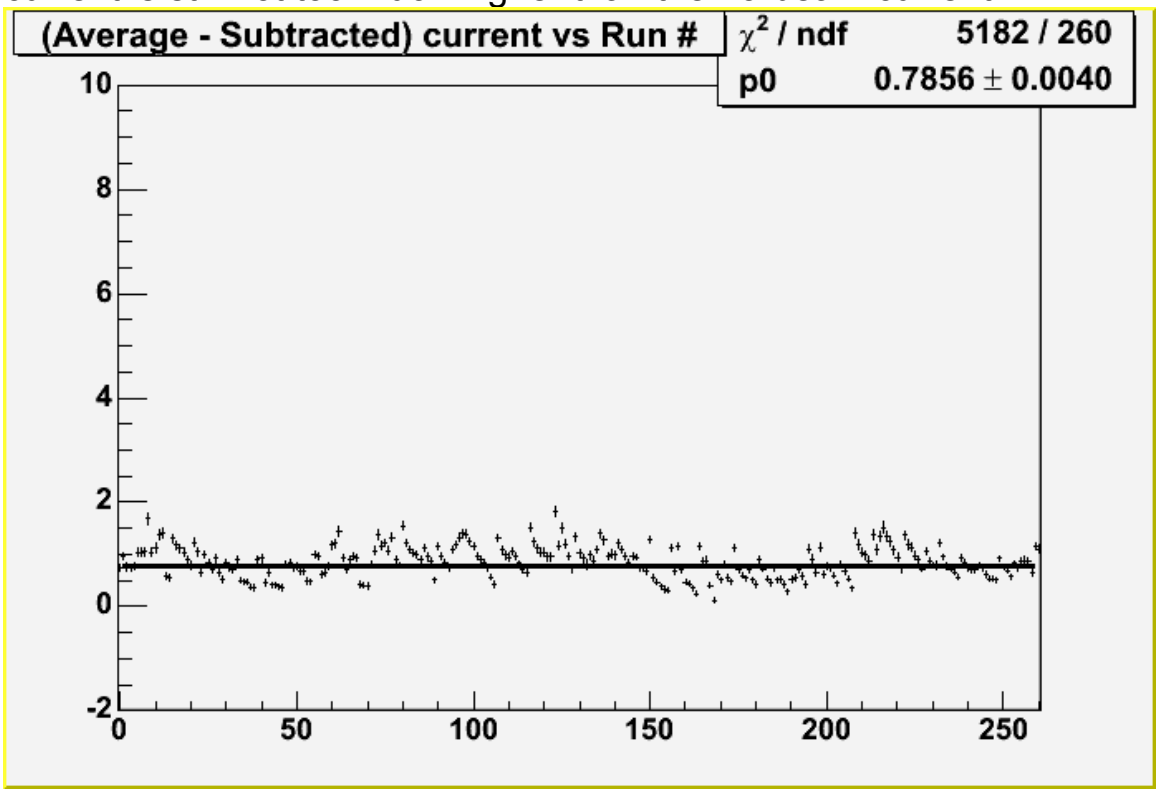


Figure C.3: Average Current Draw minus Non-Beam Current Draw as a function of run number for the Run-5 Copper-Copper data set.

Run4AuAu:

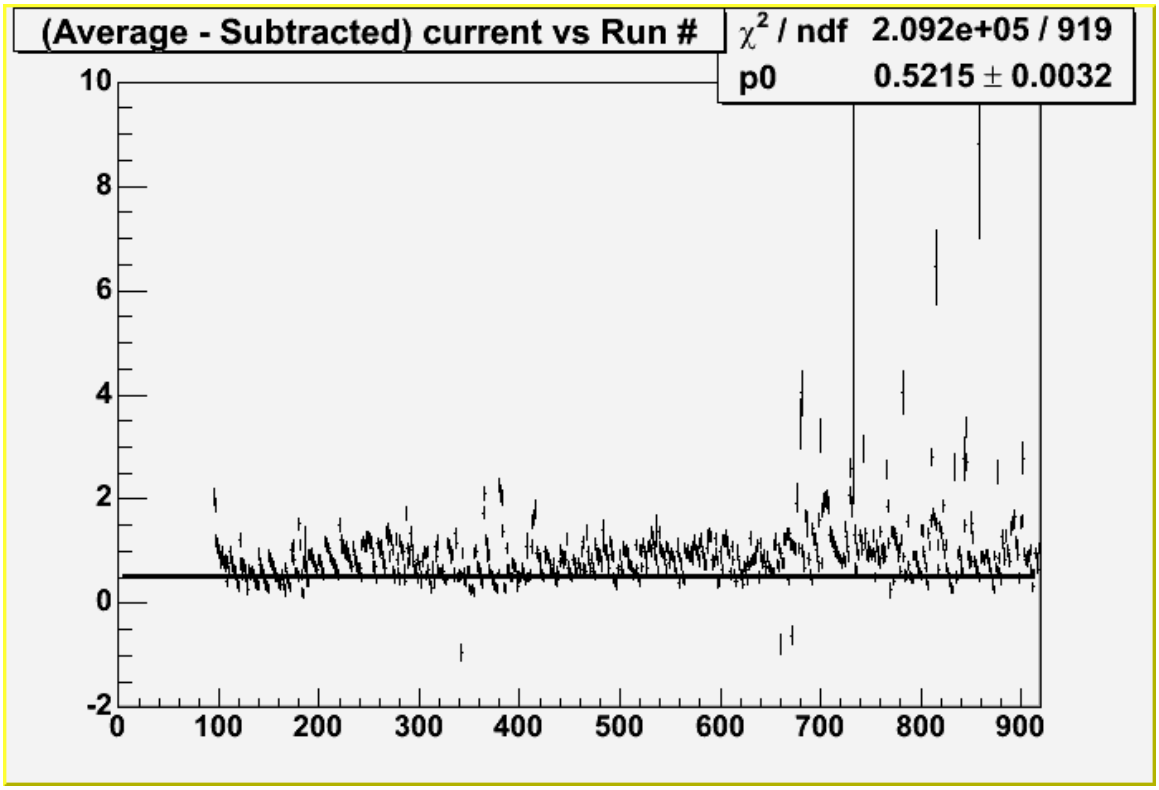


Figure C.4: Average Current Draw minus Non-Beam Current Draw as a function of run number for the Run-4 Gold-Gold data set.

HV Method

Since this method has not previously been documented (unlike the DataDriven) method, and has in fact evolved somewhat recently, from just using a broken wire parameterization, we describe the involved steps in some detail.
The README (and DoAnalysis) files in the CVS area, which also holds all macros contain the full/exact description.

Step 1: SETUP / From HV logs to run-by-run ntuples

Description: After getting the raw HV log/ascii files from HPSS, and providing a list of runs that should be studied, we run two perl scripts to first extract begin and end times for each run in the list, and extract

all HV readings that fall within the run's time window into separate files. Then we convert the ascii files into ntuples, with the info still separated run-by-run. I.e. this step only involves translation of data from one form the other.

Step 2: Determining the HV status for each chain (and each run)

Description**:** Here we use the macro **average.C** for each run, and as the name suggests, take an average of the current and voltage values from the usually tens of snapshots readings for each chain. Checks are done that the voltages are up, and that no trips or ramps appear to have taken place.
After this step we can then produce (with **scan_average.C, plot_average.C**) the current draw plots that were shown above.

Step 3: Estimating the efficiency for each HV group (and each run)

Description**:** We run the macro **muidHV_2pack_root.C** for each run, which checks the average current draw, compares it with the normal current without beam, and consider the approximate acceptance loss from broken wires. Based on these numbers and the efficiency loss formula as a function of voltage sag, an efficiency value is estimated for each HV group.
We subsequently produce a combined ROOT file with the info for all runs that have passed our good run selection using the macro **combine.C**.

Step 4: Comparing results with the DataDriven Method.

Description**:** We compare the results from the HV method and the DataDriven method with the macro **muidploteff.C**, which makes the panel-by-panel plots that are shown first in the Results section. We also do a more detailed comparison of the difference between the two methods with the the **compare.C** macro. Finally, we merge the results from the methods (**compare.C**) whereby low statistics values (approx. 40 out of 600 chains; 41 for pp) from the DataDriven method are replaced by values from the HV method. These 40 groups all are from 'group 0' in the vertical orientation for the large panels (0,2,3,5)

in all 5 gaps. There was one extra group in run5pp, namely North, gap 1, horizontal, panel 4, group 0.

Appendix D: Overlap Correction

In the DataMethod (Colorado method) we determine the plane efficiency (probability to have at least one hit in a plane). This is not the same as the tube efficiency due to overlapping tubes. Note that the same issue is relevant for the HV method since the overall efficiencies are determined using Cosmic Ray data analyzed with the Colorado method (i.e. plane efficiencies). There are two relevant overlap issues we discuss below.

1) Some panels overlap and thus a single particle may pass completely through two different panels in a given gap and orientation.

2) Tube two-packs also overlap. In the current MC, if there is a hit overlapping two two packs it gives an equal efficiency for each to get a hit even though the particle can only have passed through one tube of each two pack.

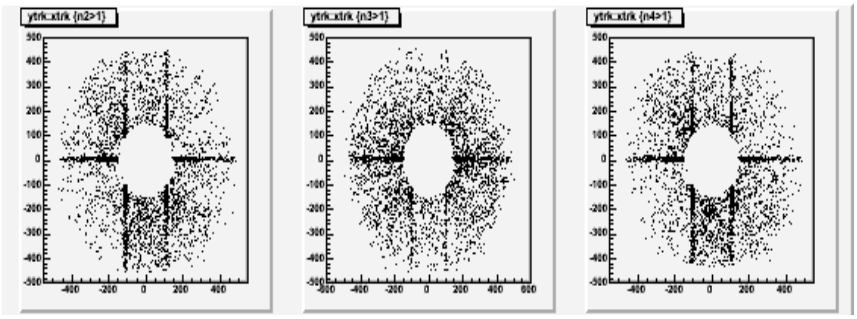


Figure D.1: One can see from the hit distribution in MC, that the panel overlap effect is not uniform, but only focused around the regions where the panels overlap. The second effect of overlapping two packs is uniform across the detector.

If we are willing to assume there is no difference if the effect is uniform across the detector, then we can calculate a correction factor to convert from the plane efficiencies (probability to have at least one hit in a plane) as determined from the Colorado method, to the two-

pack efficiency which is the proper input to the Monte Carlo. Sean Kelly determined exactly this correction, which was then re-verified by a reflexive test.

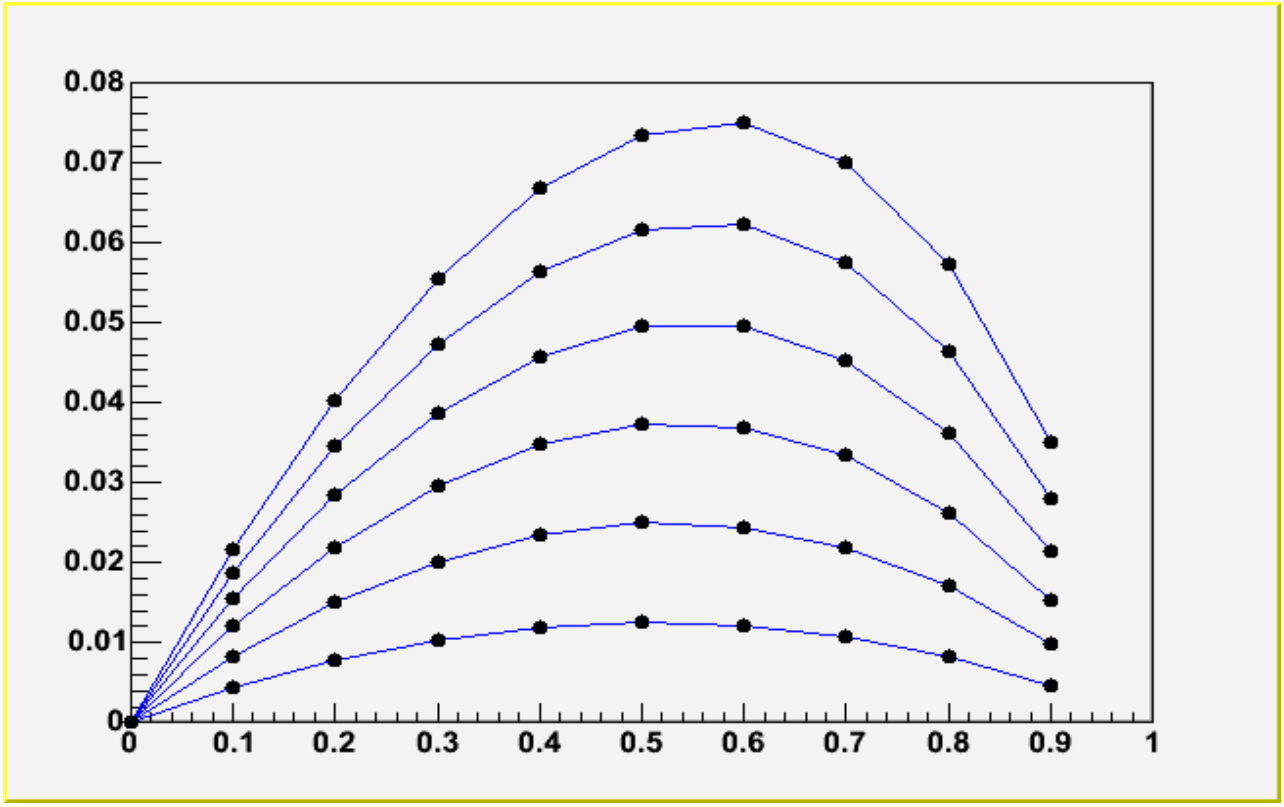


Figure D.2: The magnitude of the overlap correction factor from plane to two-pack efficiency as a function of the overall efficiency and for different overlap fractions (f=0.05, f=0.10, f=0.15, … , f=0.30 from bottom curve to top). The x-axis is the nominal plane efficiency; the y-axis is the level of the correction from plane efficiency to two-pack efficiency.

If E is the plane efficiency (for getting one or more hits), e is the two-pack efficiency, and f is the fraction of the time when the track goes through an overlap region with more than one chance to get a hit.

$$E = (1-f)e + f(1-(1-e)^2)$$

$$e = \frac{(1+f) - \sqrt{(1+f)^2 - 4fE}}{2f}$$

We have determined that there is a 14% overlap correction (f=0.14) needed to have a perfect reflexive test. This correction is applied prior to committing the final MUID efficiency files to CVS and is accomplished by using the perl script on the HVgroup efficiency file.

offline/analysis/muideff/macros/DataMethod/GenMuidEffOverlapCorrect.prl

However, there is a systematic error in this method currently. The overlap correction is 14%, with 8% from overlapping panels and 6% from overlapping two-packs. If a particle passes through two neighboring two-packs, the particle only traverses one tube of each two-pack. The current Monte Carlo gives each two packs the equivalent efficiency as if the particle passed through both tubes of both two packs. It is estimated that the true efficiency for a properly operating detector when passing through only one tube is of order 65%. In principle this can be done properly by fixing the Monte Carlo, but this error is small and this Monte Carlo modification was deferred.

If the nominal two-pack efficiency is 93%, then in the MC when a particle passes through a two-pack overlap region, one actually has two chances for a hit at 93% each. Thus, the probability for at least one hit is (1.0 – 0.07*0.07) = 0.995.

If the Monte Carlo were modified to handle this overlap correctly, and one had only a 65% two pack efficiency, the probability for at least one hit is (1.0 – 0.35*0.35) = 0.877. Since this applies to 6% of all particles that strike the two pack overlap region, the difference in the overall tube efficiency = 0.06 * (0.995 – 0.877) = 0.007. Thus, as a conservative estimate, the RMS systematic error is +/- 0.7%. In principle it is not a symmetry error, but the contribution is small and so is treated as such for simplicity.