EXTENDING THE RUN04 BAD AREA CUT

Kr-83m data

Rachel Mannino
30 August 2017

RUN04 ^{83M}KR DATASET

- Using single-scatter ^{83m}Kr data from Run04
  - kr83mRun4_MATforRachel.mat = Matlab version of kr83mAllRun4.npz
    - Evan created this with his filter code
    - Residual ^{83m}Kr events from WS2014-16 dataset, rather than ^{83m}Kr injections.
- Plan: Use ^{83m}Kr data to extend the bad area cut as this provides high statistics dataset of 32.1 keV + 9.4 keV IC electrons.
  - May merge within an event to look like a 41.5 keV signal.

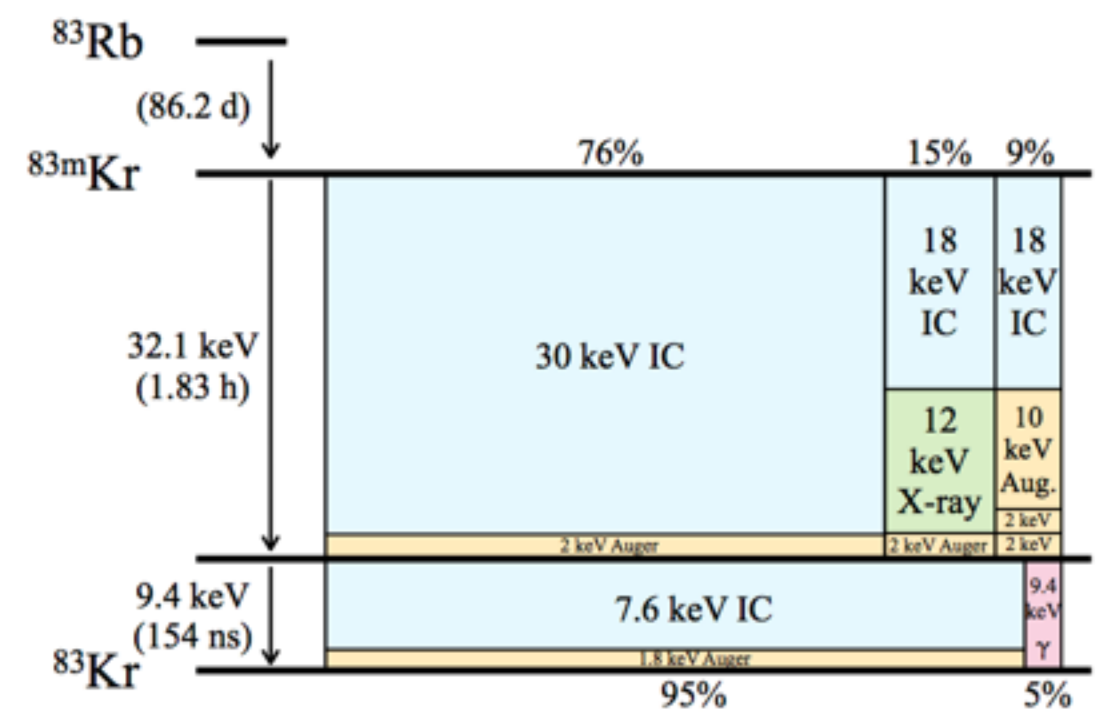
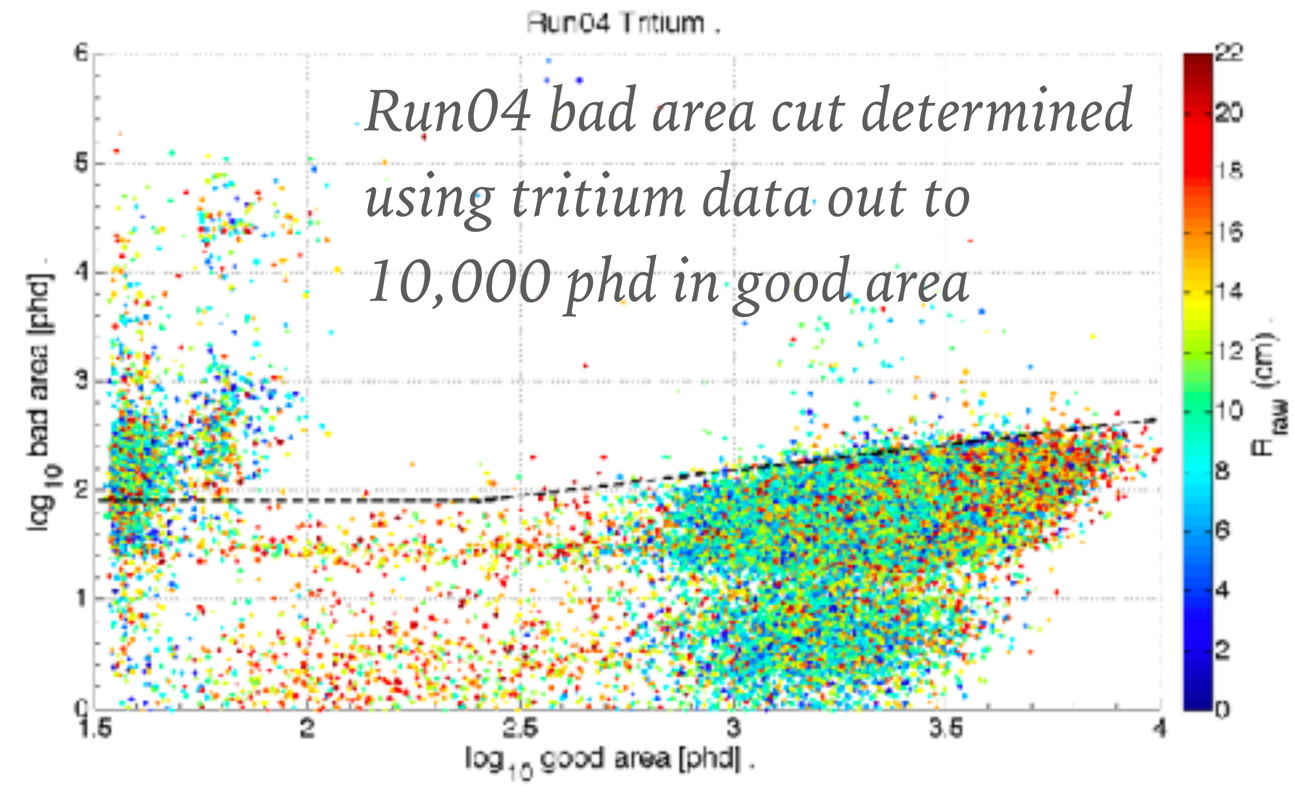


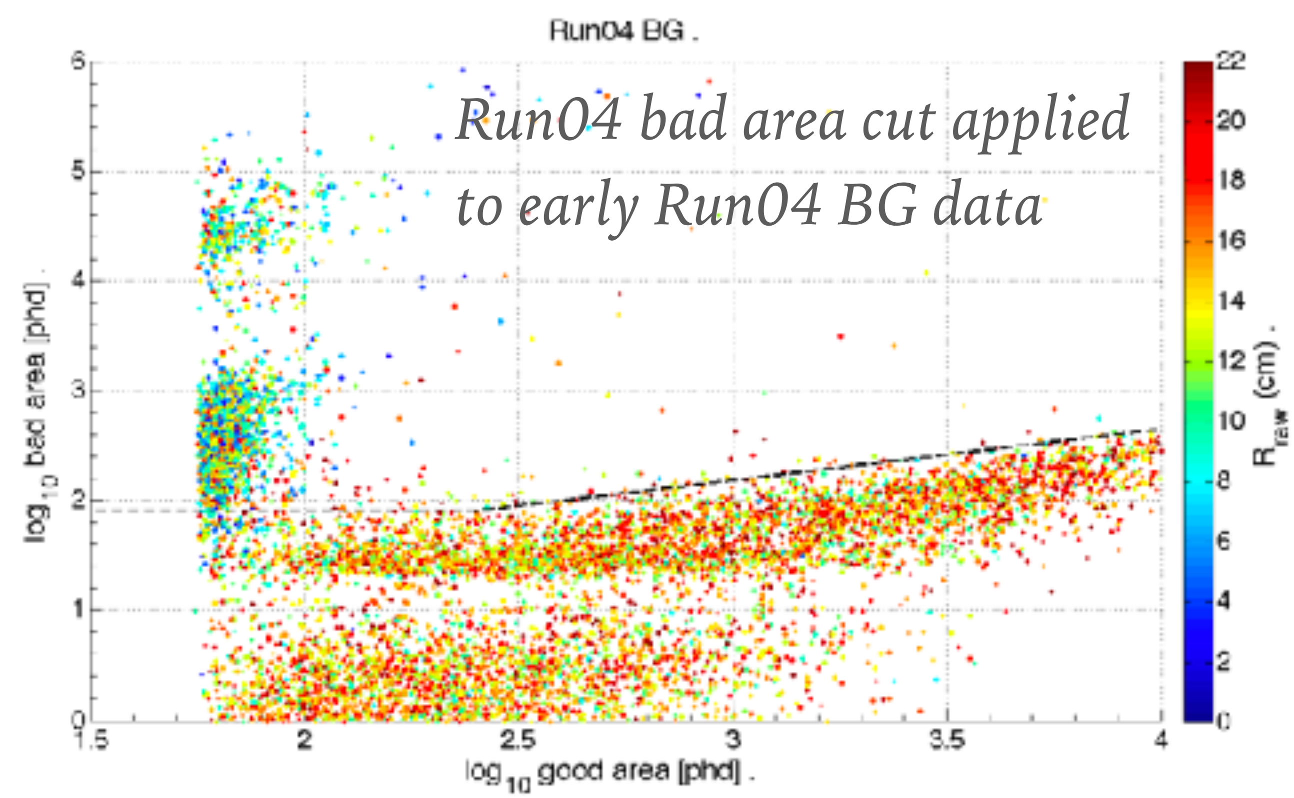
FIG. 1. Decay schematic of ^{83m}Kr . The width of each column is proportional to the branching fraction of that decay mode, the vertical divisions are proportional to energy partitioning among internal conversion electrons, Auger electrons, x-rays, and gamma-rays. Numerical values from Reference [2].

arXiv:0905.1766

GOOD AREA AND BAD AREA

- Good area = S1 + S2;
- Bad area = full event area - good area;
- Bad area cut removes events where the event window has anomalies such as electron trains, glow, etc.
  - LUX only keeps 10 pulses/event, so using the full_event_area_phe RQ captures the area of all signal area above baseline, even if the PulseFinder did not classify it as a pulse.
  - Designed for single-scatter events.
  - Calibrate bad area cut using high statistics datasets such as tritium (*earlier incarnation of Run04 bad area cut*) or ^{83m}Kr (*now*).
- Filter code creates "goodarea" and "badarea" RQs using uncorrected, raw S1 and S2 areas.



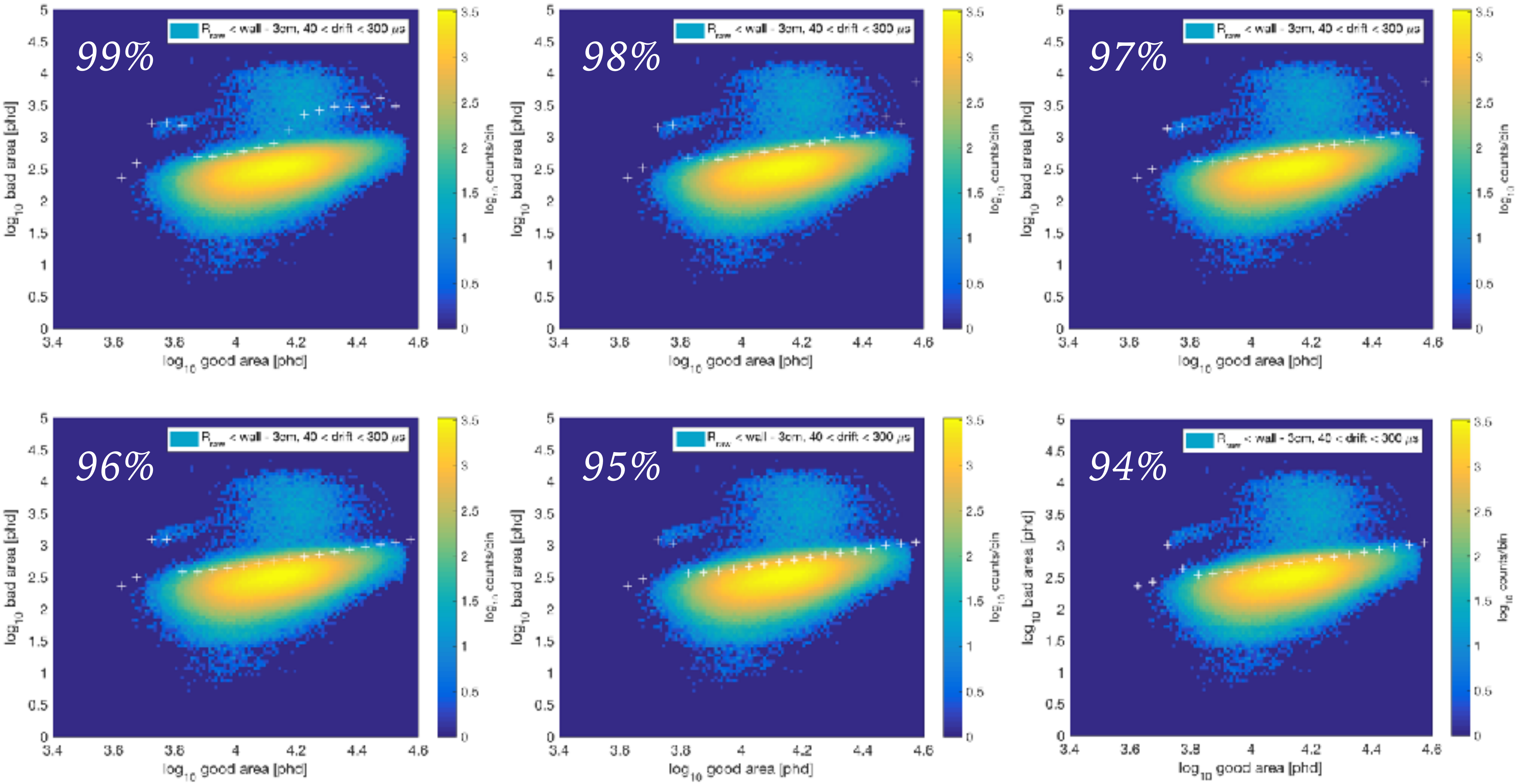


CONSTRUCTING A NEW BAD AREA CUT WITH ^{83m}KR, FIDUCIAL CUT

1. Bin log10(good area) in the vicinity of ^{83m}Kr data distribution *(see table)*.
   - ***Bins 1, 2, and 20 have poor statistics once a fiducial cut is applied.*** These bins should be neglected for the fit.
2. Calculate the log10(bad area) value at which X% of the data in the bin of log10(good area) is below.
   - Initial Run04 bad area cut determined from tritium data kept 99% (X=99) of the data within the log10(good area) bin. This only cut 1% of the events as having too much bad area.
3. Determine the best value of X (ie., what percentile to keep).
4. Fit the log10(bad area) values at X% to calculate a cut line as a function of log10(good area) and log10(bad area).

Bin	Min log10(good area)	Max log10(good area)	Counts/bin
1	***3.6***	***3.65***	***4***
2	***3.65***	***3.7***	***70***
3	3.7	3.75	471
4	3.75	3.8	2180
5	3.8	3.85	7829
6	3.85	3.9	21655
7	3.9	3.95	48342
8	3.95	4	82281
9	4	4.05	112664
10	4.05	4.1	133439
11	4.1	4.15	143502
12	4.15	4.2	137757
13	4.2	4.25	112339
14	4.25	4.3	72783
15	4.3	4.35	36850
16	4.35	4.4	16615
17	4.4	4.45	7057
18	4.45	4.5	2644
19	4.5	4.55	828
20	***4.55***	***4.6***	***61***

TESTING DIFFERENT X% VALUES, FIDUCIAL CUT APPLIED



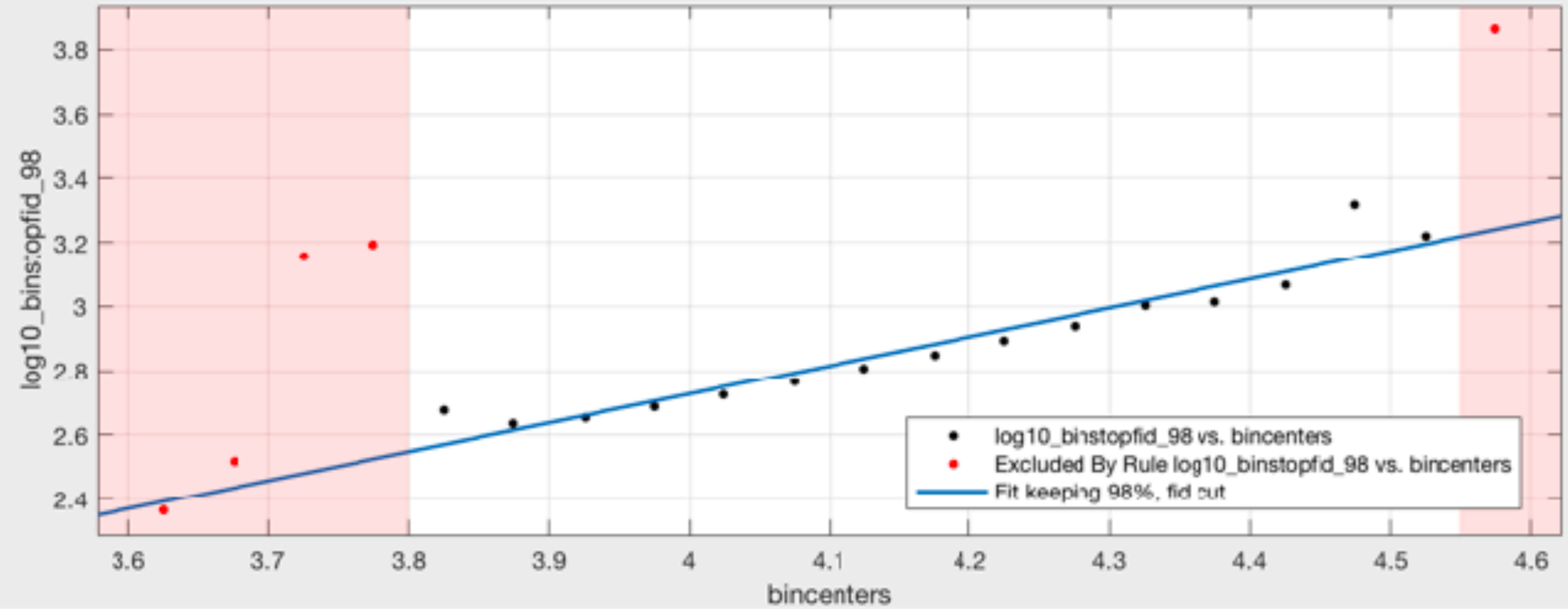
➤ Keeping large X% pushes the bad area cut into population 2 at large good areas and population 3 at low good areas, but decreasing the bad area cut risks removing population 1 events.

APPLY FIT TO 98% VALUES

Results

Linear model Poly1:
  f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
  p1 = 0.8953 (0.7348, 1.056)
  p2 = -0.855 (-1.526, -0.1839)

Goodness of fit:
  SSE: 0.05024
  R-square: 0.9178
  Adjusted R-square: 0.9115
  RMSE: 0.06217



- Exclude points with log10(good area) < 3.8 & log10(good area) > 4.55 to remove bins affected by the number of events in populations 3 or 2, respectively.
- Linear fit to the other bins yields a bad area cut:
  - log10(bad area) = 0.8953*log10(good area) - 0.855
  - Keep events with log10(bad area) below this line.