MIT Kavli Institute **Chandra X-Ray Center**

MEMORANDUM
April 5, 2021

To: Jonathan McDowell, SDS Group Leader (______________)
Dale Graessle, DS Calibration Data Coordinator (______________)
Brad Wargelin, Calibration Scientist (______________)
Ian Evans, DS End-to-End Scientist (______________)
From: David P. Huenemoerder, SDS (initial / date)
Subject: Interface Control Document: HRC-S PI Reference Data & Algorithm
Revision: 2.0 (February 2018)
URL: `http://space.mit.edu/cxc/docs/docs.html#hrcstgain`
File: `hrcs_spi_icd-2.0.tex`

1 Overview of HRC-S PI Computation

An algorithm for computing the HRC-S pulse height (PI, for Pulse Invariant Amplitude) was originally documented in the memo *"A New Gain Map and Pulse-Height Filter for the Chandra LETG/HRC-S Spectrometer"* by B. J. Wargelin, P. W. Ratzlaff, and M. Juda,[1] henceforth referred to as the *Memo*. This memo is now found under "Previous Work", the link labled "2008 gain calibration" at `http://cxc.harvard.edu/cal/Letg/Hrc_bg/` ("Development Documentation" link). Work in-progress can be found at `http://hea-www.cfa.harvard.edu/~bradw/Gain2017/`. Changes described there in the SPI-correction formula are described here.

The PI-correction algorithm can result in a reduction of HRC-S background by about a factor of two. There are two parts to the processing:

1. Computation of the PI for every event, which requires a Calibration Database (`CALDB`) coefficients file, and algorithmic support in the tool, `hrc_process_events`.

2. Application of a position-dependent filter before binning an LETGS spectrum, which requires a new `CALDB` `PI`-region file.

---

[1] `http://cxc.harvard.edu/cal/Letg/Hrc_bg/gain.pdf`, November 25, 2008

This memo will define the `CALDB` files, the processing algorithm for `hrc_process_events`, and the filter application method.

In Revision 2.0, we introduce a second algorithm for application of the time-dependent gain correction. For back-compatibility, we introduce a new keyword, `GMAPVER` into the `CALDB` file to indicate whether we require Legacy mode (value of 0; no time-dependent correction), 1 (first time-dependent correction algorithm), or 2 (new algorithm described here).

2 PI Computation

The essence of the the PI computation is [2]:

$$P_n^{(1)} = \frac{a_1}{m_{ij}} \left(\frac{D_n}{C_i(t) S_{ij}^{norm}} - b_{ij} \right) + a_0, \tag{1a}$$

$$P_n^{(2)} = \frac{1}{C_i(t)} \left(\frac{a_1}{m_{ij}} \left(\frac{D_n}{S_{ij}^{norm}} - b_{ij} \right) + a_0 \right). \tag{1b}$$

We used notation to make the two-dimensional nature of the calibration data explicit in detector raw coordinates by subscripts ij, and we have removed explicit numerical values which are subject to calibration updates. We use a "row, column" notation, in which row i is a `RAWY` index, and column j indexes `RAWX`. For `HRC-S`, the LETG dispersion is along `RAWY`. In this equation the terms are

$P_n^{(ver)}$ is the desired value for PI for the $n^{\rm th}$ event, at detector raw pixel coordinate $(\mathtt{RAWX}, \mathtt{RAWY})$, which will be mapped to calibration bin, (i, j).

The superscript refers the value of `GMAPVER` and the evaluation of either Equation 1a or 1b contingent upon `GMAPVER`.

D_n represents the uncorrected event pulse-height data value ("`SAMP`" in *Memo* terms), for the $n^{\rm th}$ event.

Other terms are calibration data, two of which are scalars (a_n), three are two-dimensional maps (m, S, and b), and another is a time-dependent term with one spatial dimension (C).

For purposes of efficiency in `CALDB` representation and event processing, we can collect terms into a more compact forms:

$$P_n^{(1)} = G_{ij0} + \frac{G_{ij1}}{C_i(t)} D_n, \tag{2a}$$

$$P_n^{(2)} = \frac{G_{ij0}}{C_i(t)} + \frac{G_{ij1}}{C_i(t)} D_n \tag{2b}$$

---

[2] Originally given by Equation 5 in the *Memo*, but here modified slightly

G_{ijk} is the "Gain Map", with two spatial dimension, i and j, and one polynomial order dimension, k.

$C_i(t)$ is the time-dependent gain correction, with one spatial dimension, i.

The `CALDB` file will hold one 3D image, G_{ijk}, one 2D image, C_{in}, and coordinate grids for each axis. Since the grids may be non-uniform, axes will be enumerated rather than specified by FITS image coordinate terms.

2.1 File Structure: HRC-S Gain Coefficients

2.1.1 File Names

The `CALDB` type for time dependent gain is `t_gmap`. File names will be of the form

`hrcsD1999-07-22t_gmapN`$nnnn$`.fits`

in which $nnnn$ is a 4-digit version counter (which starts at `0001`).

2.1.2 HDU Components

The following table describes the file structure by Header-Data Unit number, type, extension name, content, and HDU classes. An asterisk (*) denotes the principal HDU.

HDU	HDU	EXTNAME	EXTVER	CONTENT	HDUCLASS	Description
0	PRIMARY	N/A	N/A	N/A	N/A	NULL
1 (*)	BINTABLE	AXAF_TGAIN	1	CDB_HRCS_TGAIN	ASC DETCHAR TGAIN	Binary table extension listing coefficients for PI gain correction as a function of time & position.

2.1.3 Columns and Coordinate Systems

Columns for HRC-S gain coefficients are given in the following table. Column ordering in the FITS table is arbitrary.

TTYPE	TUNIT	TFORM	TLMIN	TLMAX	Description
`RAWXGRID`	pixel	n_xE	0	4095	`RAWX` grid, low bin edges. n_x is the number of elements in the `RAWX` grid.
`RAWYGRID`	pixel	n_yE	0	49151	`RAWY` grid, low bin edges. n_y is the number of elements in the `RAWY` grid.
`TIMEGRID`	MJD	n_tD	N/A	N/A	Time axis for data stored in column `TGAIN`. Units are MJD. `MJDREF` is 50814, which corresponds to `1998-01-01T00:00:00 (TT)`, and $MJD = JD - 2400000.5$. n_t is the number of elements in the `TIME` grid.
`GAINMAP`	1	$2n_xn_yE$	N/A	N/A	Coefficients map vs raw coordinate. $2 \times n_x \times n_y$ is the total length fo the array.
`TGAIN`	1	n_tn_yE	N/A	N/A	Normalization coefficient vs time and `RAWY`. $n_t \times n_y$ is the total length of the array.

Grids are required to be monotonic and in ascending order. They are, however, not necessarily uniform. Hence, the grids are tabulated rather than specified by standard FITS World Coordinate System (WCS) image keywords, `CRPX`, `CDLT`, and `CRVL`.

2.1.4 Special Coordinate Keywords

The image column require several keywords in order to define its coordinate system and relation to other columns. These follow the conventions given in "Representation of celestial coordinates in FITS", by Greisen and Calabretta (Sept. 9, 1996)[3].

i`CTYP`n: These string-valued keywords define the axes of image columns.

For `GAINMAP` we require:

1CTYPn = 'ORDER'
2CTYPn = 'RAWX',
3CTYPn = 'RAWY'.

Here, n, is the column index for `GAINMAP`. `ORDER` refers to the order of the polynomial coefficient (not spectral diffraction order).

For `TGAIN`, we require:

1CTYPn = 'RAWY',
2CTYPn = 'TIME'.

with n being replaced by the column number of `TGAIN`.

`TDIM`n: This string-valued keyword specifies the dimensionality of each axis of image columns. It has the form, `'(`N_z, N_x, N_y`)'`

For `GAINMAP` we require:

---

[3] `http://www.cv.nrao.edu/fits/documents/wcs/wcs.all.ps`

TDIMn = '$(2, n_y, n_x)$'.

For TGAIN we require:

TDIMn = '(n_y, n_t)'.

CREFn: This keyword has a string value which maps the image axes to TTYPEn keywords which specify grids for those axes.

For GAINMAP we require:

CREFn = '(ORDER, RAWX, RAWY)'.

For TGAIN we require:

CREFn = '(RAWY, TIME)'.

For non-enumerated axes (uniform grids), the coordinate strings would be what the WCS mapping requires (i.e., 2CRPX, 2CDLT, and 2CRVL).

2.1.5 Relevant Header Keywords

Relevant keywords are:

GMAPVER: The algorithm version. A value of 1 indicates the gain correction defined by Equations 1a or 2a, value of 2 by 1b or 2b. A value of 0 indicates the original, static gain correction ($\mathtt{HDUNAME} = \mathtt{AXAF\_GAIN}$, $\mathtt{HDUCLAS} = \mathtt{GAINMAP}$, $\mathtt{CCNM} = \mathtt{GMAP}$).

TELESCOP: The only allowed value is CHANDRA.

INSTRUME: The only allowed value is HRC.

DETNAM: The only allowed values are HRC-S.

CIPn: For configuration control, this keyword (or list if n is present to enumerate several keywords) names the input CIP, or Calibration Interface Product files delivered by calibration and formatted into this ARD. For example, the original calibration prototype file was called `spimean.fits`.

CIPREF: For configuration control, this keyword cites a relevant calibration memo (or preferably, URL). The origninal calibration prototype was provided at

$$\mathtt{http}: //\mathtt{cxc.harvard.edu}/\mathtt{contrib}/\mathtt{letg}/\mathtt{GainFilter}/.$$

2.1.6 `CALDB` Keywords

The required `CALDB` keywords are as follows:

```
CCLS0001  'BCF'
CDTP0001  'DATA'
CCNM0001  'T_GMAP'
CVSD0001  1999-07-22T00:00:00
CVST0001  00:00:00
CDES0001  'Detector gain map coefficients'
```

2.2 Size and File Number Estimates

There will be one HRC-S coefficients file which will cover the time of the mission. The minimal spatial gain map is 576×48 real-valued elements, and the minimal temporal map is 576×18 real-valued elements. Including the 2-plane gain map and grids for each axis, the size is about 500 kB. With finer grids, this could be 4 to 8 times the size, or up to 4 MB.

3 HRC-S Time-dependent Gain Correction Algorithm Specification

The time-dependent gain correction and PI computation is to be a part of the `hrc_process_events` program. The algorithm's PI values will become the PI column of the output event file.

1. Read the `DATE-OBS` from the event or reference file.

2. Read the gain file as specified by the `hrc_process_events` parameter, "`gainfile`". If the "`gainfile`" parameter's value is `CALDB`, then the implied default is to read the most recent time-dependent file from the Calibration Database.

   Determine the file type from the header (`EXTNAME` or `HDUNAME` keywords).

   Read the `GMAPVER` keyword from the header. The value should be 0, 1, 2, or null (missing).

   If the type is time-dependent (`TGAIN`), then proceed as enumerated below according to the value of `GMAPVER`.

   Otherwise (type `GAINMAP`) and `GMAPVER` is 0 or Null, proceed in "legacy" mode (no time-dependent correction).

   If the type is `GAINMAP` and `GMAPVER` is ≥ 1, this is an error. *(TBD: Either issue a warning and proceed with* `GMAPVER` $= 0$ *or exit with an error.)*

   The time-dependent gain file elements are the `GAINMAP` 3D image, the `TGAIN` 2D image, and the corresponding enumerated coordinate grids, `RAWXGRID`, `RAWYGRID`, `TIMEGRID`.

3. Interpolate `TGAIN` to the observed date. `TGAIN` is comprised of one array in `RAWY` for each MJD of the `TIMEGRID`. Each element along the `RAWY` axis should be linearly interpolated in time to produce a new array in `RAWY`.

   Find the times in the `TIMEGRID` grid which bracket the observed date, t_{obs}.

   If the observed date is after the greatest `TIMEGRID`, then use the greatest `TIMEGRID`.

   Otherwise, linearly interpolate. If j is the index for the greatest `TIMEGRID` less than the observed date, and i is the `RAWY` grid index, then for each i,

$$\begin{aligned}\mathtt{TGAIN}_i(t_{\text{obs}}) = {} & \mathtt{TGAIN}_{i,j} \\ & + \left(\frac{t_{\text{obs}} - \mathtt{TIMEGRID}_j}{\mathtt{TIMEGRID}_{j+1} - \mathtt{TIMEGRID}_j}\right)(\mathtt{TGAIN}_{i,j+1} - \mathtt{TGAIN}_{i,j})\end{aligned} \tag{3}$$

   The result, $\mathtt{TGAIN}_i(t_{\text{obs}})$, is a 1D array of the same length as `RAWY`.

4. Divide each column (index j) of the `GAINMAP`'s coefficients by the interpolated `TGAIN` (contingent upon `GMAPVER`, as given by the parenthetic superscripts):

$$\mathtt{GAINMAP}'^{(1)}_{ij0} = \mathtt{GAINMAP}_{ij0} \tag{4a}$$

$$\mathtt{GAINMAP}'^{(2)}_{ij0} = \mathtt{GAINMAP}_{ij0}/\mathtt{TGAIN}_i(t_{\text{obs}}) \tag{4b}$$

$$\mathtt{GAINMAP}'_{ij1} = \mathtt{GAINMAP}_{ij1}/\mathtt{TGAIN}_i(t_{\text{obs}}) \tag{5}$$

   The results are 2D maps of the same size as $\mathtt{GAINMAP}_{ijk}$.

5. For each event, n, read the $\mathtt{RAWX}_n$, $\mathtt{RAWY}_n$, $\mathtt{AMP\_SF}_n$, and $\mathtt{SUMAMPS}_n$ from the event file.

6. Compute the data value,

$$D_n = \mathtt{SUMAMPS}_n \times 2^{(\mathtt{AMP\_SF}_n - 8)} \tag{6}$$

7. From the event's raw coordinates, use the gain file's enumerated grids to look up the indices of the gain-correction coefficients. Let $\mathcal{I}$ be the row index (`RAWY` coordinate), and $\mathcal{J}$ be the column index (`RAWX` coordinate). Then

$$\mathcal{I} = \text{search}(\mathtt{RAWYGRID} \leq \mathtt{RAWY}_n) \tag{7}$$

$$\mathcal{J} = \text{search}(\mathtt{RAWXGRID} \leq \mathtt{RAWX}_n) \tag{8}$$

   In words, this means *"Find the greatest index in the* `RAW` *grid such that the grid's value there is less than or equal to the event's* `RAW` *coordinate."*

8. Compute the new PI value and write to the output event file:

$$PI_n = \texttt{GAINMAP}'_{\mathcal{IJ}0} + \texttt{GAINMAP}'_{\mathcal{IJ}1} \times D_n \tag{9}$$

   The result may contain non-physical values $PI < 0$ and $PI = NaN$. These represent less than 0.5% of the events and should be set to 0.

   Values of $PI > 1023$ should be set to 1023.

   The data written to the FITS file should be of integer type.

9. Record the gain map file used in the header keyword, `GAINCORF`.

10. Record the gain map version used in the header keyword, `GMAPVER`.

3.1 Interface Changes to `hrc_process_events`

The event processing program, `hrc_process_events`, requires no user-interface changes to support the time-dependent gain correction. Instead, the code will determine the appropriate algorithm based on the input gain file (from its type and the `GMAPVER` keyword).

Downstream Contingencies: The use of `tgain` in `hrc_process_events` is important for proper use of event filtering in `tgextract`. If the gain file name is of type, "`t_gmap`", then an appropriate filter should be used, which also matches in the value of `GMAPVER`. These files are of content type, `TGPIMASK2`, with the `CALDB` filename identifier `*pireg_tgmap*`.

The standard recommendation for filtering HRC events is to exclude $PHA = 255$. This will no longer be necessary.

4 Spectral Filtering

The computation of PI is the first step in LETGS background reduction. A filter on coordinates $(\texttt{tg\_mlam}, PI)$ must be applied before binning into a spectrum. We define a `CALDB` file which defines a polygonal filter region appropriate for filtering HRC-S `tgain`-corrected events. [4]

4.1 File Structure: LETGS tgain Region Filter

4.1.1 File Names

The `CALDB` type for time dependent gain corrected events filter is `pireg_tgmap`. File names will be of the form

---

[4] This is similar to the current user-option to apply a $(\texttt{tg\_lam}, PI)$ filter which offers mild background reduction. The `tgmask2` files are in the `CALDB`, but are not applied during pipeline processing. There are made available for application at the user's discretion.

`letgD1999-07-22pireg_tgmap_N`*nnnn*`.fits`

in which $nnnn$ is a 4-digit version counter (starting at `0001`).

4.1.2 HDU Components

The following table describes the file structure by Header-Data Unit number, type, extension name, content, and HDU classes. An asterisk (*) denotes the principal HDU.

HDU	HDU	EXTNAME	EXTVER	CONTENT	HDUCLASS	Description
0	PRIMARY	N/A	N/A	N/A	N/A	NULL
1 (*)	BINTABLE	REGION	1	TGPIMASK2	ASC TG	Binary table extension holding a region for filtering HRC-S tgain-corrected events before binning LETGS spectra.

4.1.3 Columns and Coordinate Systems

Columns for LETGS tgain-corrected events filter are given in the following table. Column ordering in the FITS table is arbitrary.

TTYPE	TUNIT	TFORM	TLMIN	TLMAX	Description
SHAPE	1	nA	N/A	N/A	Region shape designation, of value “POLYGON”.
TG_MLAM	angstrom	nD	N/A	N/A	Array of grating event file `tg_mlam` coordinates, of length n.
PI	channel	nD	0	N/A	Array of HRC-S event file tgain-corrected coordinates, of length n, which must correspond to the `tg_mlam` length.

Other region-specific columns are not required for the `POLYGON` shape.

4.1.4 Relevant Header Keywords

Relevant keywords are:

`GMAPVER:` The algorithm version. A value of 1 indicates the gain correction defined by Equations 1a or 2a, value of 2 by 1b or 2b. A value of 0 indicates the original, static gain correction (`HDUNAME` = `AXAF_GAIN`, `HDUCLAS` = `GAINMAP`, `CCNM` = `GMAP`).

`TELESCOP:` The only allowed value is `CHANDRA`.

`INSTRUME:` The only allowed value is HRC.

`DETNAM:` The only allowed value is HRC-S.

`GRATING`: The only allowed value is LETG.

`GRATTYPE`: The only allowed value is LEG.

`MFORM1`: is a CXC datamodel column association; the only allowed value is `'TG_MLAM,PI'`.

`MTYPE1`: is a CXC datamodel column association name; the only allowed value is `'mlam_pi'`.

`CIP`n: For configuration control, this keyword (or list if n is present to enumerate several keywords) names the input CIP, or Calibration Interface Product files delivered by calibration and formatted into this ARD. For example, the original calibration prototype file was called `spimfilter.spec`.

`CIPREF`: For configuration control, this keyword cites a relevant calibration memo (or preferably, URL). The origninal calibration prototype was provided at

> http://cxc.harvard.edu/contrib/letg/GainFilter/.

4.1.5 `CALDB` Keywords

The required `CALDB` keywords are as follows:

```
CCLS0001  'BCF'
CDTP0001  'DATA'
CCNM0001  'TGPIMASK2'
CBD10001  'TGMAPCOR(T,TRUE)'
CVSD0001  1999-07-22T00:00:00
CVST0001  00:00:00
CDES0001  'LETGS m*lambda - PI region filter'
```

4.2 Size and File Number Estimates

There will be one to a few LETGS region filter files which will cover the time of the mission. Each has a size of about 10 kB.

4.3 Filter Application

The filter can be applied as a data-model virtual file specification on the input to `tgextract` or by applying the filter with `dmcopy` to create an actual filtered event file, as in these examples:

```
tgextract
   infile=evt2"[(tg_mlam,pi)=region(letgD1999-07-22pireg_tgmap_N0001.fits)]"
   outfile=pha2

dmcopy
   infile=evt2"[(tg_mlam,pi)=region(letgD1999-07-22pireg_tgmap_N0001.fits)]"
   outfile=evt2_filtered
```

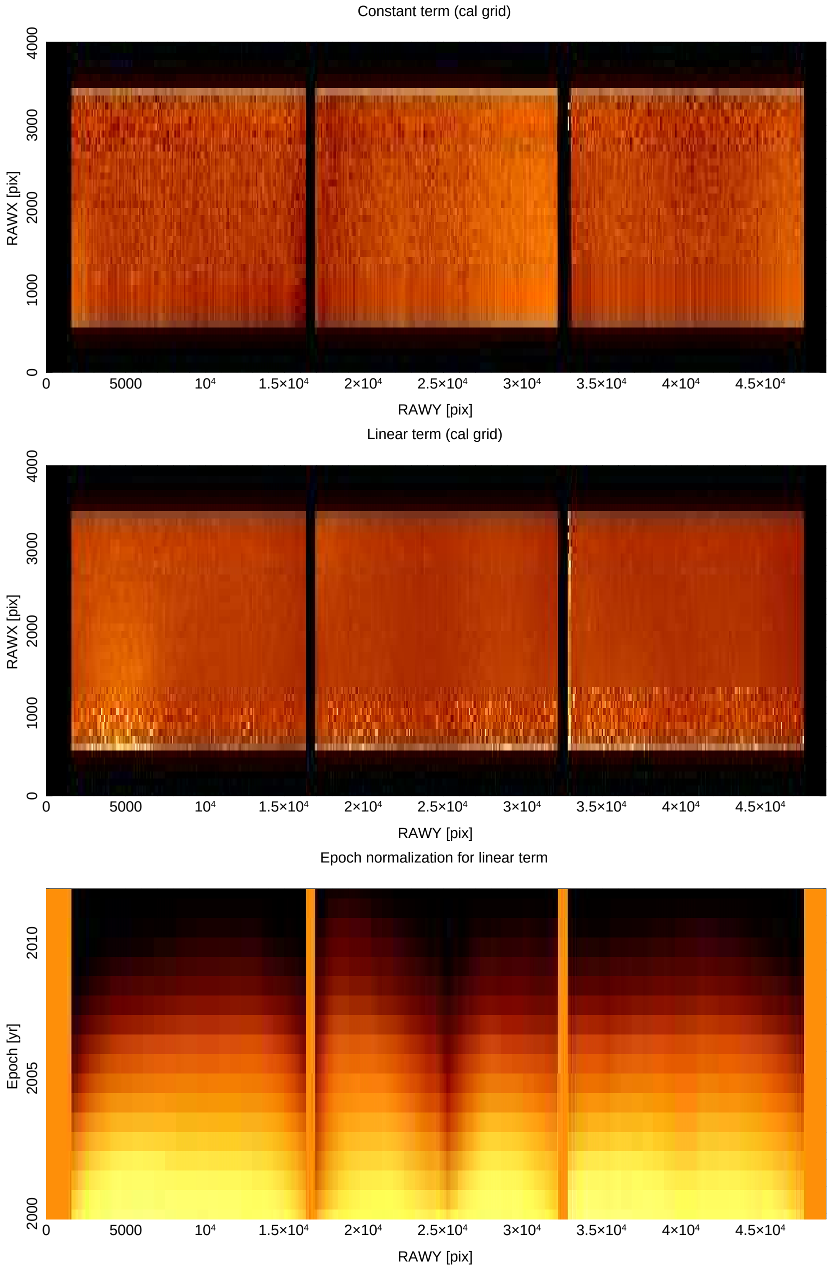
4.4 Caveats

Effective area vs. order has not been re-calibrated to include the effects of the PI filter for high orders. The filter has been optimized for first orders; it will clip the PI distributions for different the different energies at a given location by different amounts. The magnitude of the effect is, however, presumed to be very small compared with other calibration uncertainties (TBR).

A Prototype & Tests

A.1 v1.4 tests

A prototype was written in ISIS to read the calibration file, an events file, and compute the TGAIN-corrected PI. This was compared to the output of the Calibration Group scripts. The latter scripts' output was also run through spectral extraction with application of the filter file. The following plots show some of the characteristics of the calibration data and of resulting events and spectra for standard processing and for TGAIN-corrected data. **NOTE: These tests are for `GMAPVER` $= 1$, but are representative the the changes.**



Constant term (cal grid)

Linear term (cal grid)

Epoch normalization for linear term

Figure 1: Tgain-correction coefficient maps. Top: constant term; Middle: linear term; Bottom: time-dependent factor.

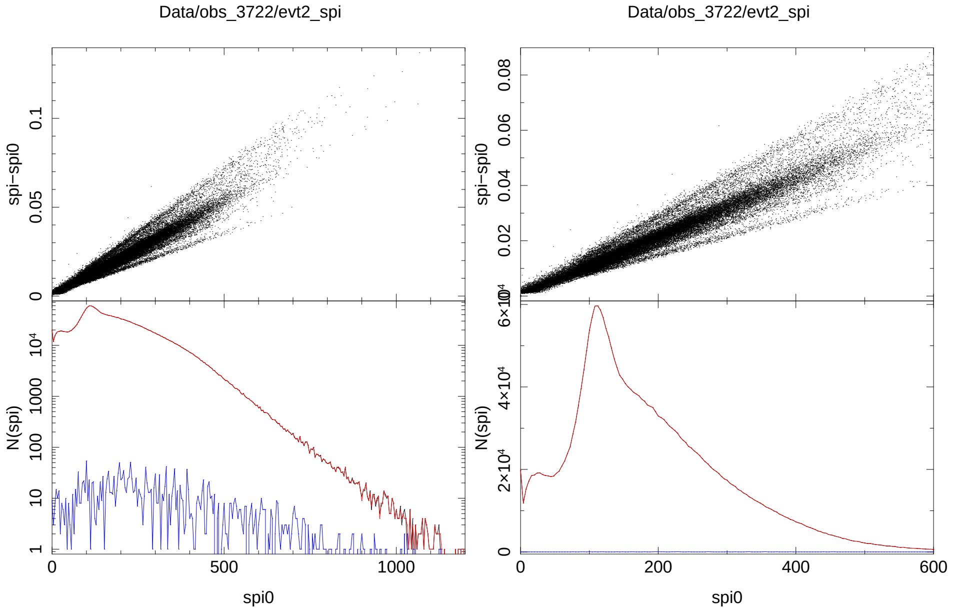


Figure 2: PI residuals, compared to Cal script output, $spi0$. Top panels: PI residuals by event. Bottom panels: PI histograms and difference. The right-hand plot has a linearly scaled histogram (lower panels) and a smaller $spi0$ range. Filter cutoffs range from $PI \sim 120 - 300$ (see Figure 3).

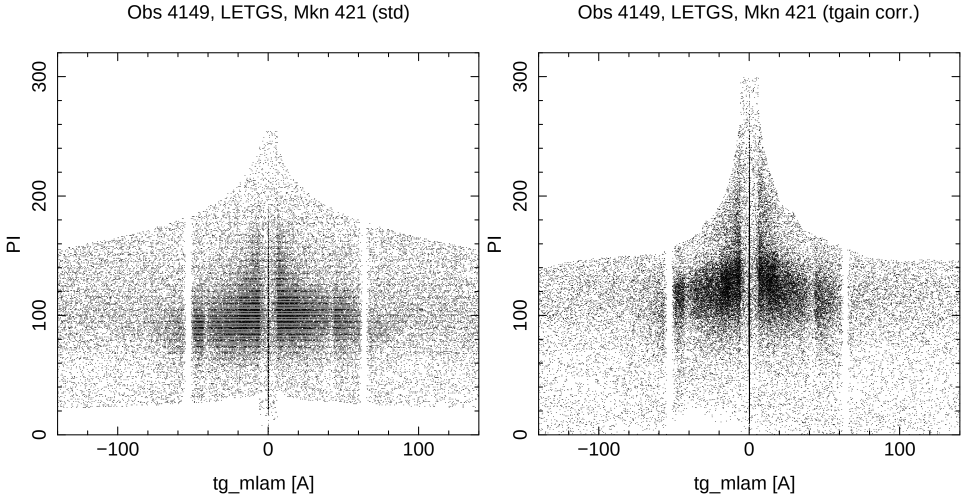


Figure 3: The PI-region filters as a function of `tg_mlam`. Left: current filter. Right: TGAIN-PI filter. The events are shown for ObsID 4149, Mkn 421.

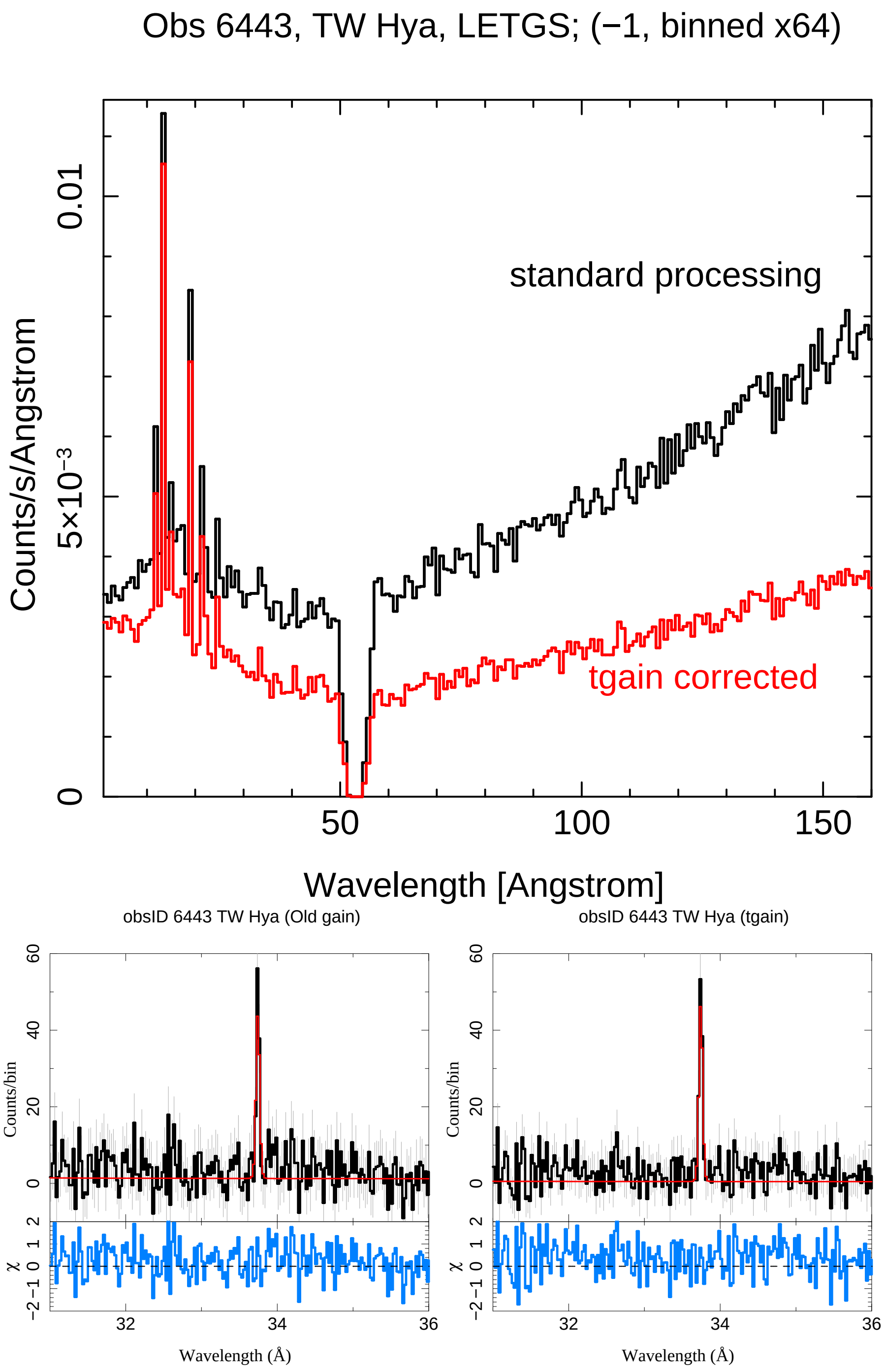


Figure 4: TW Hya, ObsID 6443. Top: comparison of the count rate density for -1^{st} order for standard processing (black) and for tgain-corrected data (red), binned to 0.8 A/bin. Bottom: detail of the C VI 33.7 Å line net counts for standard processing (left) and TGAIN-corrected (right).

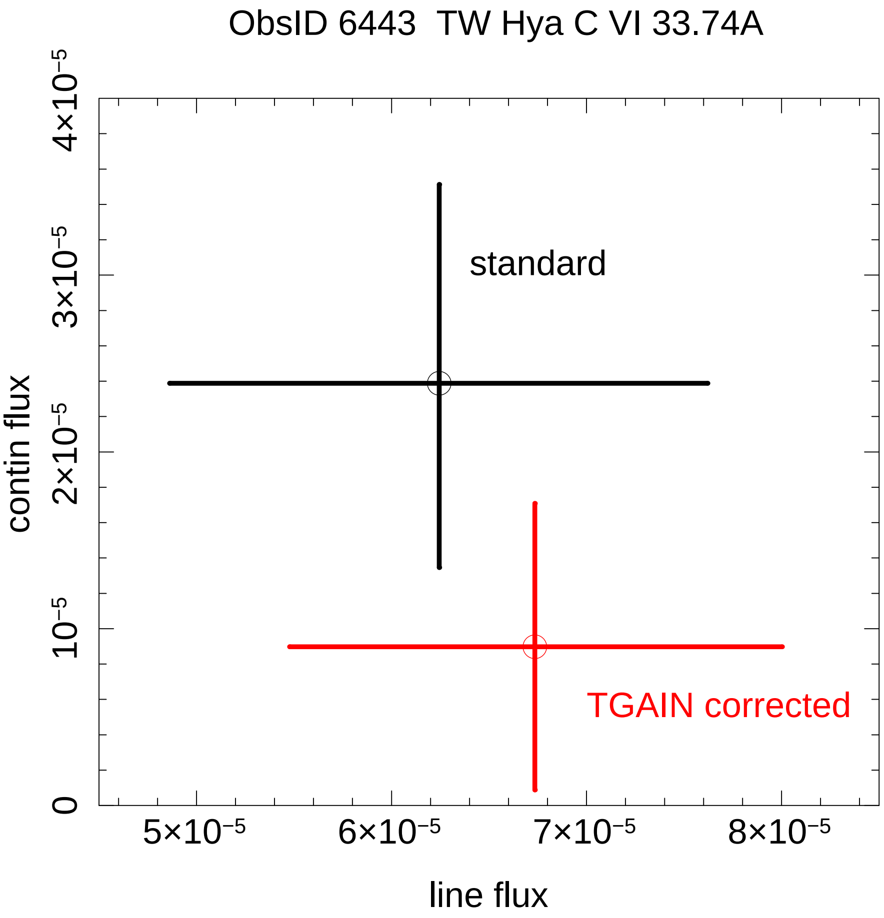


Figure 5: TW Hya, ObsID 6443; Comparison of fits of a constant plus Gaussian function to the C VI 33.7 Å line region shown in Figure 4. The x-axis is the line flux, the y-axis is the continuum flux density. The best fit and 90% confidence intervals are shown for standard processing (black) and for TGAIN-corrected data (red). Error-bars are slightly smaller for the TGAIN corrected data.

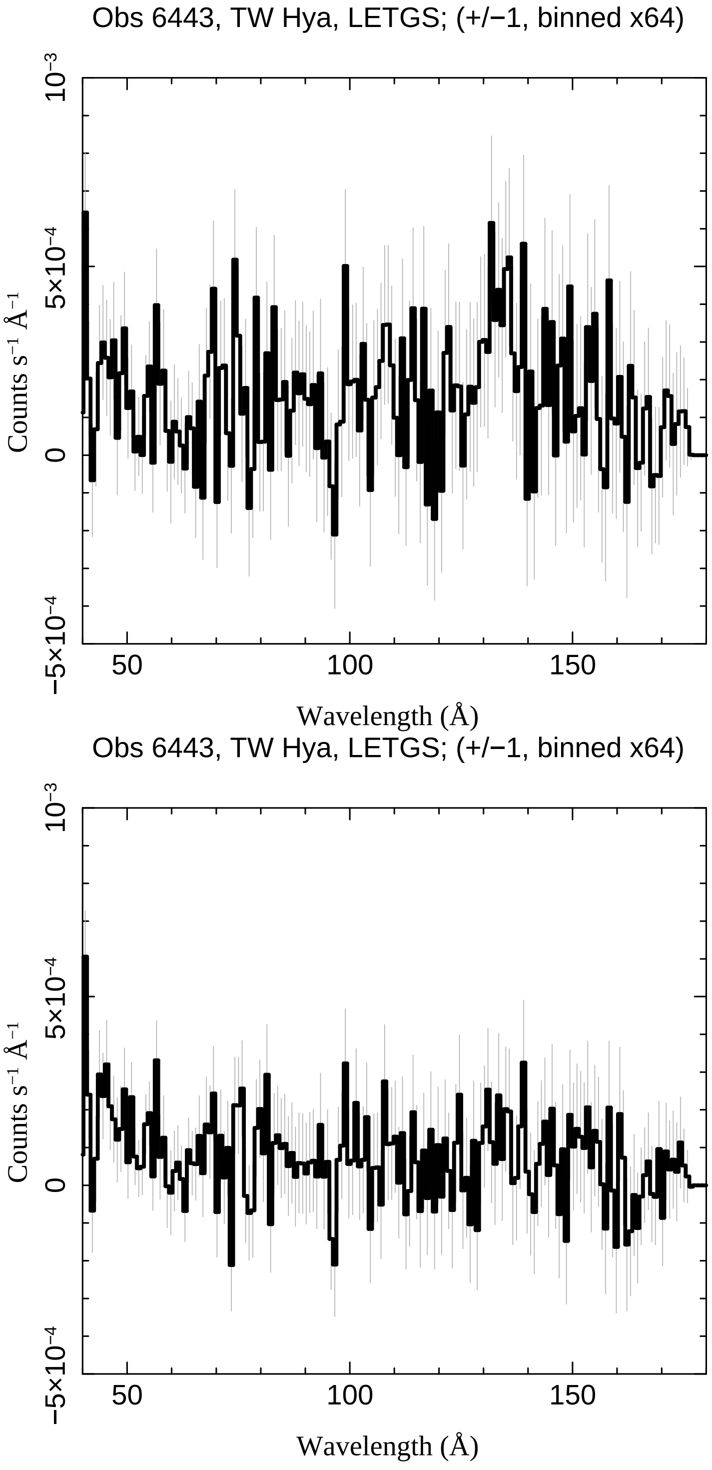


Figure 6: TW Hya, ObsID 6443; Comparison of the continuum net count rate density binned to 0.08 Å/bin for standard processing (top), and TGAIN-corrected data (bottom) in a spectral range where background dominates. Reduction of the noise is apparent in the TGAIN-corrected extraction.