**Chandra X-Ray Observatory Center** **MIT Center for Space Research**

# MEMORANDUM

Updated: August 1, 2000

**To:** CXC-SDS

**From:** John Houck

**Subject:** Removing Streaks from ACIS-S4

Images of the ACIS-S4 chip produced using Level 1 or Level 2 event files show a variable pattern of linear streaks which fall along rows of pixels with constant CHIPY (Fig. 1). These streaks appear to be caused by a flaw in the serial readout which randomly deposits significant amounts of charge along pixel rows as they are read out. This excess charge appears as a string of spurious X-ray events which are not removed by standard grade filtering and, because the streak locations vary from frame to frame, the streak events cannot be removed using standard methods appropriate for filtering bad columns and bad pixels.

Instead, one can identify and remove ACIS-S4 streak events using the fact that each streak occurs within a single frame and deposits events along a single CHIPY row of a single node (Fig. 2). For a low count-rate source it is relatively rare for two or more source events to fall in the same CHIPY row of the same CCD node within a single frame-time, therefore most such occurrences are likely to be instrumental artifacts. To identify the streak events in a given frame, one first generates a histogram of S4 events vs. CHIPY for each node; rows with two or more events are identified as streaks and all events within such rows are discarded or flagged with a particular STATUS value. **Important Usage Note:** To avoid confusion with cosmic-ray events and events generated in bad pixels and bad columns, this streak filter should be applied only *after* filtering on standard grades and removing bad pixels and bad columns. Discarding streak-rows in this fashion results in a negligible loss of effective exposure time compared to the length of a typical observation (Fig. 3); this is a reflection of the fact that individual streaks are quite “bright” but relatively rare. One could refine this algorithm to examine spatial coincidence within individual rows in an attempt to preserve more source counts but, given the success of the basic algorithm, it is not clear that the additional effort would be worthwhile.

The single-event cutoff criterion seems to work well for most observations, but it is not optimal for high count-rate sources. The higher the source count-rate, the more likely it becomes that two or more source-counts might fall in a single node-row within a single frame-time. In this regime, the single-event cutoff may be too strict and may filter out an unacceptable number of source counts. For high count-rate sources, an optimal cutoff can be selected using an approach suggested by Herman Marshall. He found that the number of streak events ($N$) in a given CHIPY row follows an exponential distribution so that the probability of $N$ events falling in a given row is $P(N) \propto e^{-aN}$. Examining the row-count distribution of S4 events from an actual data set (Fig. 4), one can see the exponential streak distribution which dominates at high $N$ and the large excess at low $N$ due to the source-counts. The optimal cutoff value of $N$ can be chosen as the value of $N$ where the source-count distribution and the streak-event distribution cross.

One can define a more intuitive filter criterion by expressing it in terms of the maximum allowable contamination due to streaks. By fitting an exponential to the observed row-count distribution, one can "predict" the number of streak-events which should remain for a given choice of the row-count cutoff $N$. The predicted fractional streak contamination for a given row-count cutoff $N$ is then

$$f(N) = \frac{\sum_{k=1}^{N} kP(k)}{\sum_{k=1}^{N} kO(k)}, \tag{1}$$

where $P(k)$ is the predicted number of occurrences of $k$ streak events in a single row and $O(k)$ is the observed number of occurrences of $k$ events in a single row. Examining $f(N)$ for an actual data set (Fig. 5), one can see that the streak contamination fraction is a sensitive function of the chosen row-count cutoff $N$. At this point, one can construct a two-pass algorithm which, on the first pass, computes $f(N)$ and determines the optimal cutoff $N$ consistent with the allowable streak contamination fraction; the second pass then identifies streak events using the determined cutoff.

After identifying streak events, it is straightforward to determine how well the filter criterion worked. For a given cutoff $N$, one can first examine an image of the filtered data set to see if any streaks remain, then examine an image of the streak-events to see if any traces of the source are visible. For the data set mentioned above, traces of the dispersed spectrum are visible in the $N = 1$ streak-event image but are not present in the $N = 2$ streak-event image.

By default, my C implementation of this streak-filtering algorithm (`destreak.c`) supports a two-pass algorithm by examining the streak event distribution function and automatically selecting a streak threshold $N = j$ where $j$ is the smallest number of events in a single row for which

$$O(j) - P(j) < 3\sqrt{P(j)}. \tag{2}$$

To override this automatic threshold selection, it includes a command-line option which allows the user to specify the row-count cutoff for each node. Streak events may be either removed or labeled with a specific STATUS value. For each S4 node, the program can optionally output the net exposure time loss per row, the observed row-count distribution, the predicted streak row-count distribution, and the corresponding cumulative distribution function. Despite the need for frame-by-frame processing, the algorithm can be implemented efficiently and does not require extensive computing resources. For example, on a Sun Ultra-1, my implementation requires <3 Mb of run-time memory and processes a standard event file at a rate of about $\sim 3$ Mb/sec so that a "typical" $\sim$30 Mb grade-filtered Level 1 event file containing $\sim 10^4$ frames may be processed through a single pass in $\sim$10 cpu-seconds; the two-pass algorithm necessarily takes somewhat longer.

## Figures:

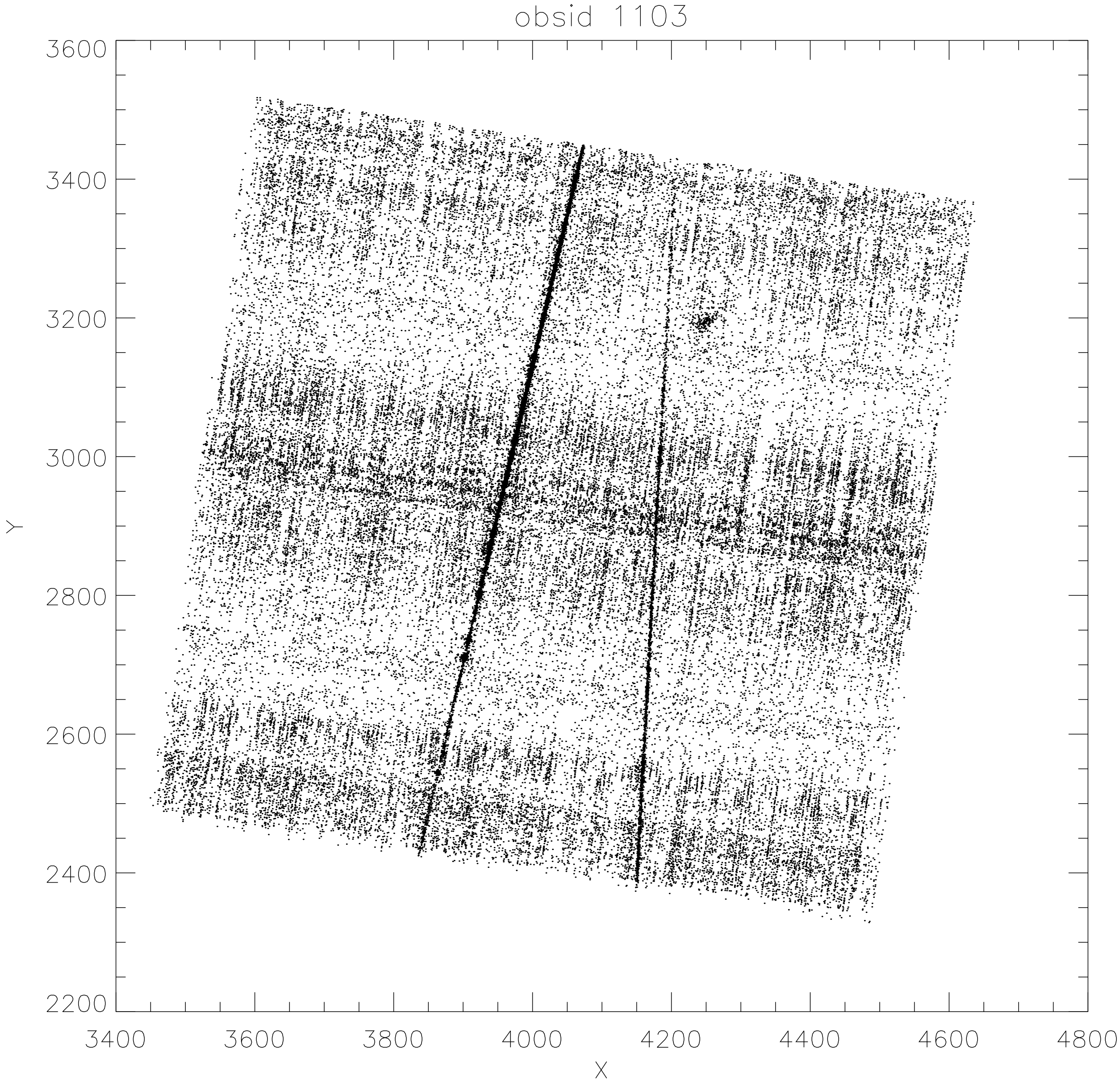
Fig. 1 – A spot-diagram of ACIS-S4 (obsid 1103) showing the streak distribution in sky X,Y coordinates along with HEG and MEG dispersed spectra. Several dithered bad columns are also visible as broad straight features crossing the dispersed spectra perpendicular to the streak direction.

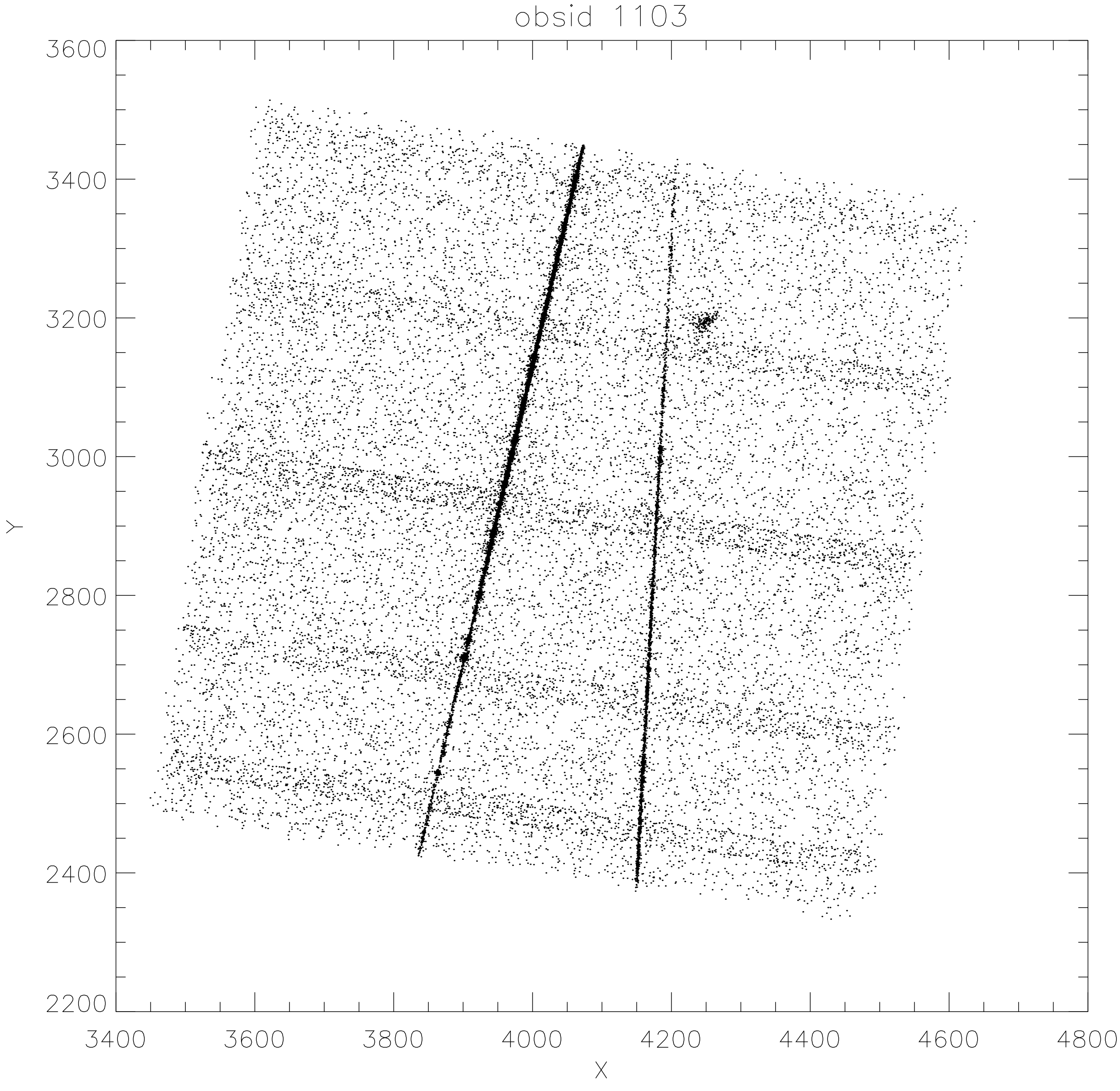
Fig. 2 – A spot-diagram of ACIS-S4 (obsid 1103) after removing streak events using the algorithm described with a default cutoff of $N$=1 (e.g. two or more events in a single CHIPY row/node/frame were identified as streak events and discarded). In this case, out of 85002 input S4 events, 36923 (43.4%) were identified as streak events. Events associated with bad columns have not been removed.

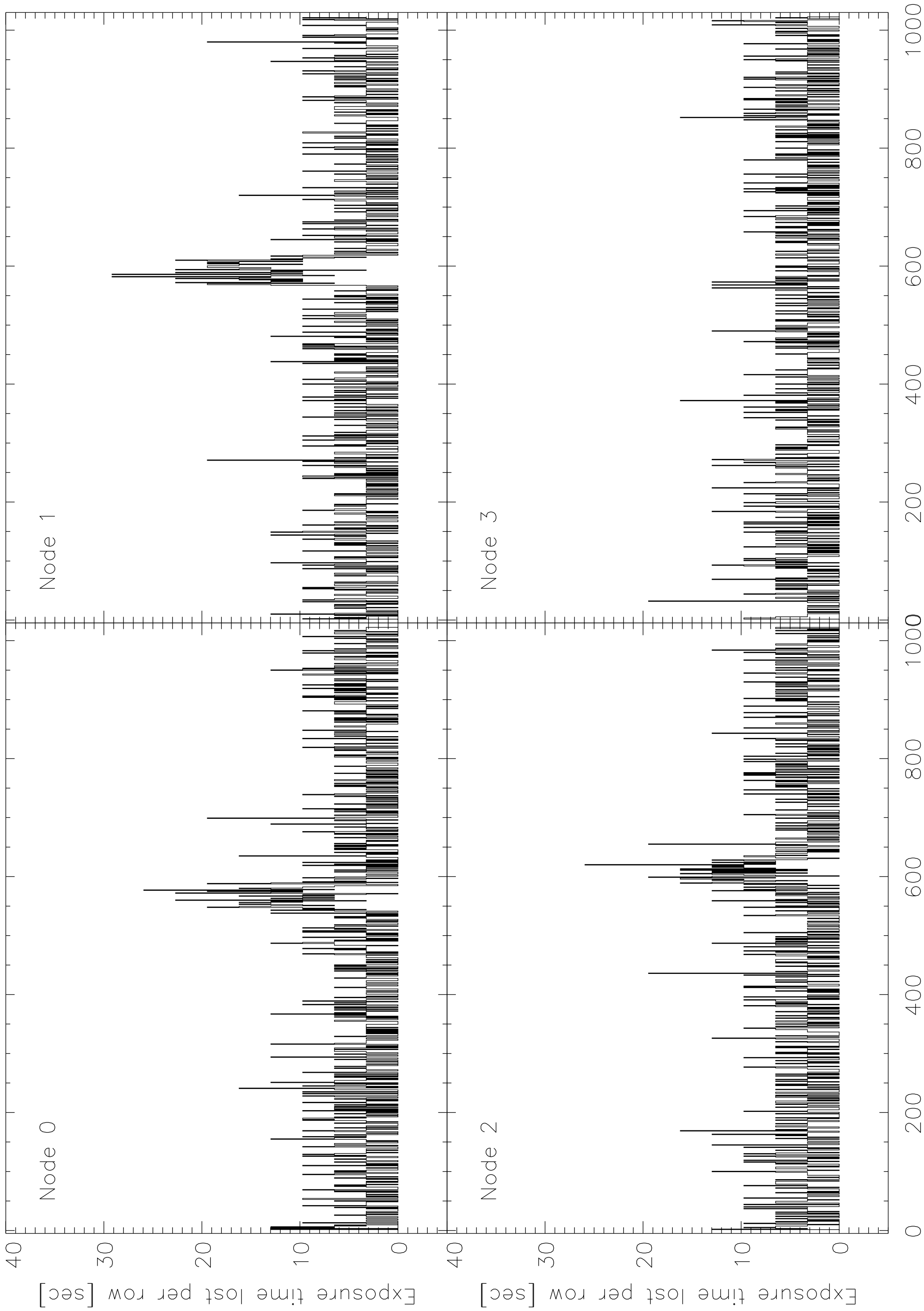
Fig. 3 – Effective loss of exposure time vs. ACIS-S4 CHIPY (obsid 1103) as a result of applying the default streak filter with $N$=1. Given that the total exposure time was 44 ksec, a loss of $<$ 30 sec exposure per row in each node is probably negligible. The peaks visible in the plots for nodes 0, 1 and 2 correspond to the location of the dispersed spectrum, indicating that the value $N$=1 is too strict, allowing a few source counts from this high count-rate source to be incorrectly identified as streak events; applying a cutoff of $N$=2 solves this problem.

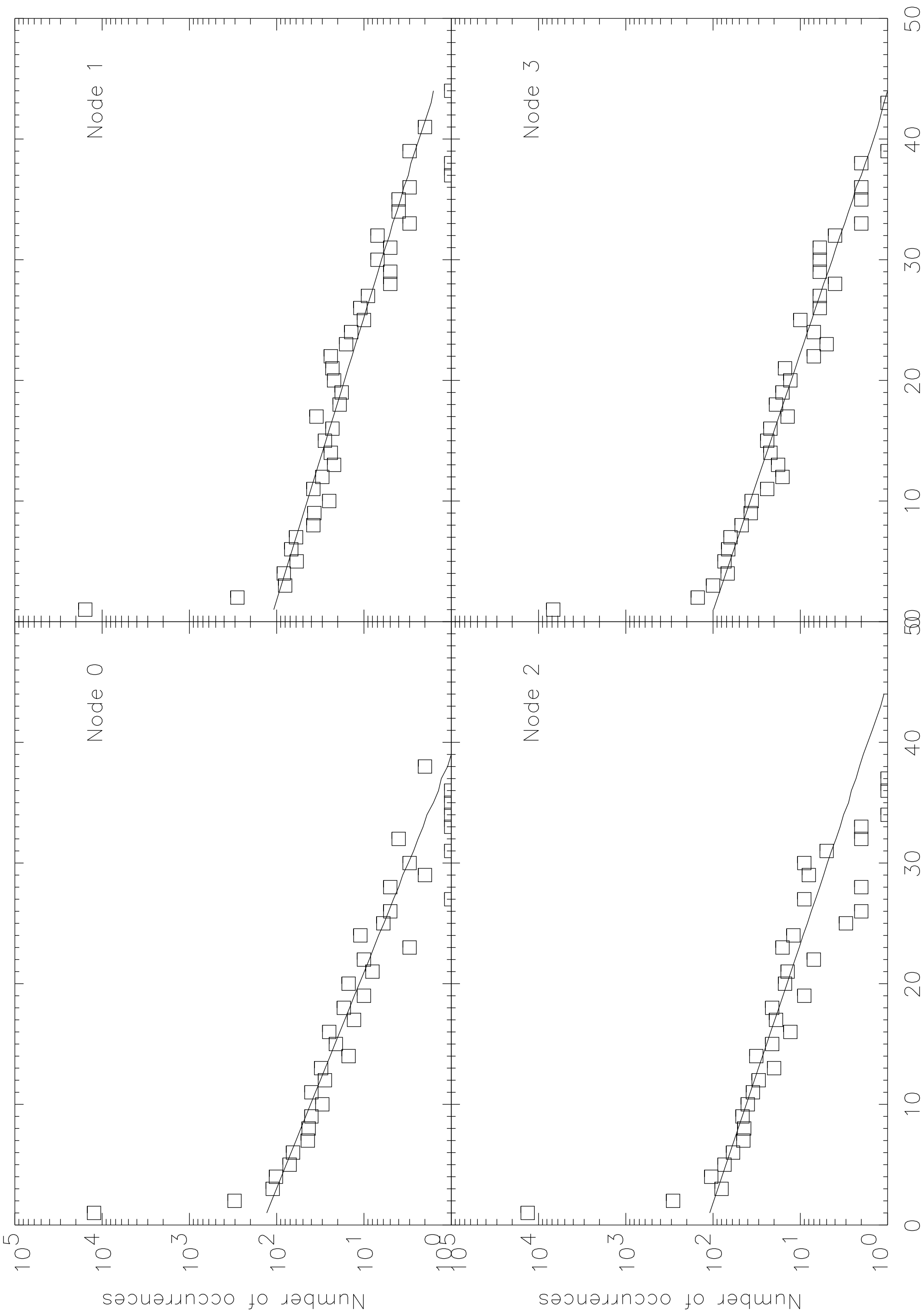
Fig. 4 – Row-count distribution for each node of ACIS-S4 (obsid 1103). The plotted points show the observed distribution of events. The straight lines are least-squares fits to the exponential distribution of streak events.

Fig. 5 – Estimated fractional streak-event contamination on ACIS-S4 vs. row-count cutoff for obsid 1103.

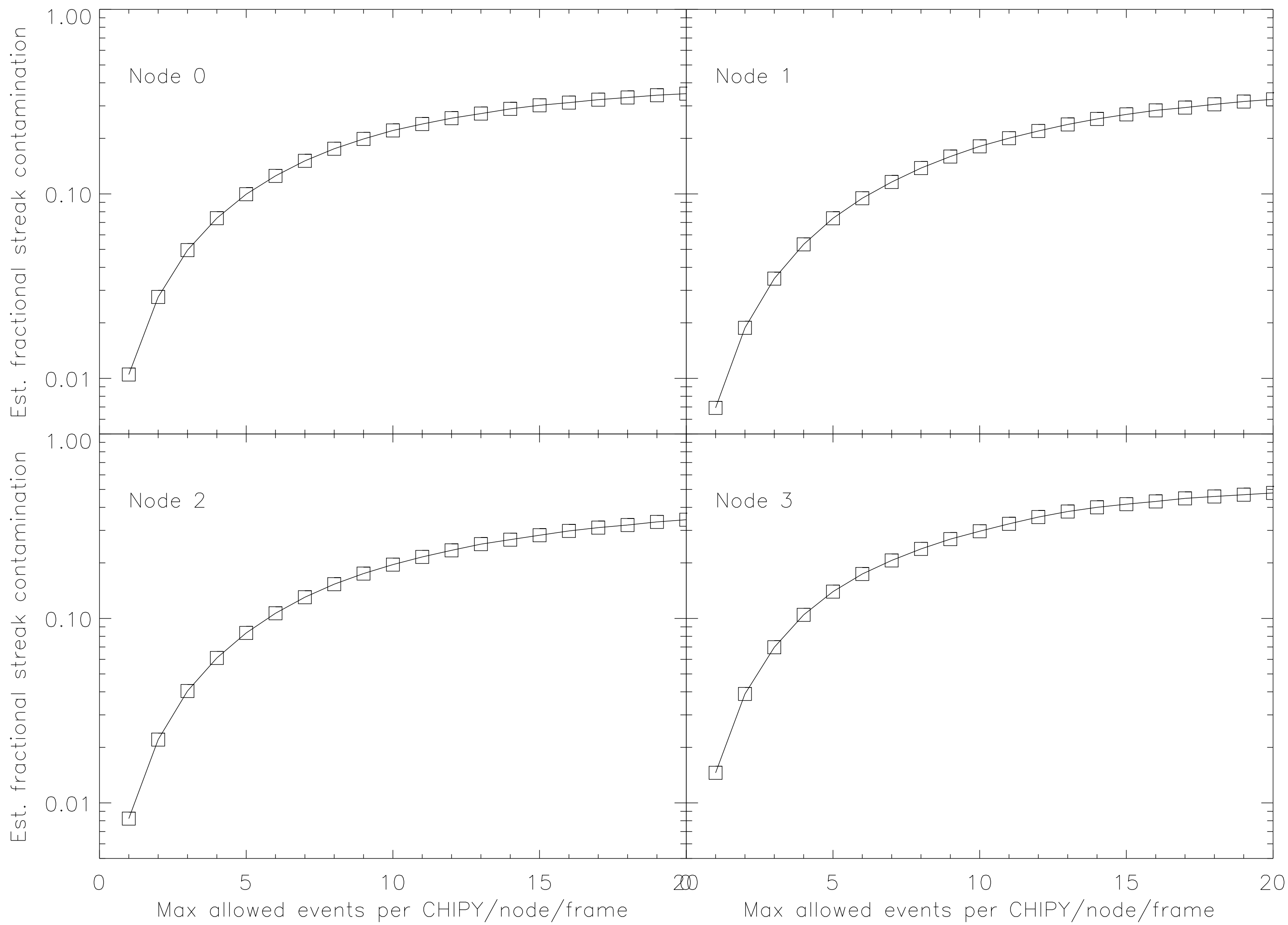

obsid 1103
3600
3400
3200
3000
2800
2600
2400
2200
Y
3400
3600
3800
4000
4200
4400
4600
4800
X


obsid 1103
3600
3400
3200
3000
2800
2600
2400
2200
Y
3400
3600
3800
4000
4200
4400
4600
4800
X


Node 0
Node 1
Node 2
Node 3
CHIPY
CHIPY
Exposure time lost per row [sec]
Exposure time lost per row [sec]



Node 0
Node 1
Node 2
Node 3
Number of occurrences
Number of occurrences
Counts in a single CHIPY
Counts in a single CHIPY


Node 0
Node 1
Node 2
Node 3
Est. fractional streak contamination
Est. fractional streak contamination
1.00
0.10
0.01
0
5
10
15
20
Max allowed events per CHIPY/node/frame
Max allowed events per CHIPY/node/frame