

INTEGRAL/SPI data segmentation to retrieve source intensity variations[★]

L. Bouchet^{1,2}, P. R. Amestoy³, A. Buttari⁴, F.-H. Rouet^{3,5}, and M. Chauvin^{1,2}

¹ Université de Toulouse, UPS – OMP, IRAP, 31028 Toulouse Cedex 04, France
e-mail: lbouchet@irap.omp.eu

² CNRS, IRAP, 9 Av. colonel Roche, BP 44346, 31028 Toulouse Cedex 4, France

³ Université de Toulouse, INPT – ENSEEIHT – IRIT, 31071 Toulouse, Cedex 7, France

⁴ CNRS-IRIT, 31071 Toulouse Cedex 7, France

⁵ Lawrence Berkeley National Laboratory, Berkeley CA 94720, USA

Received 15 May 2012 / Accepted 22 April 2013

ABSTRACT

Context. The INTEGRAL/SPI, X/γ-ray spectrometer (20 keV–8 MeV) is an instrument for which recovering source intensity variations is not straightforward and can constitute a difficulty for data analysis. In most cases, determining the source intensity changes between exposures is largely based on a priori information.

Aims. We propose techniques that help to overcome the difficulty related to source intensity variations, which make this step more rational. In addition, the constructed “synthetic” light curves should permit us to obtain a sky model that describes the data better and optimizes the source signal-to-noise ratios.

Methods. For this purpose, the time intensity variation of each source was modeled as a combination of piecewise segments of time during which a given source exhibits a constant intensity. To optimize the signal-to-noise ratios, the number of segments was minimized. We present a first method that takes advantage of previous time series that can be obtained from another instrument on-board the INTEGRAL observatory. A data segmentation algorithm was then used to synthesize the time series into segments. The second method no longer needs external light curves, but solely SPI raw data. For this, we developed a specific algorithm that involves the SPI transfer function.

Results. The time segmentation algorithms that were developed solve a difficulty inherent to the SPI instrument, which is the intensity variations of sources between exposures, and it allows us to obtain more information about the sources’ behavior.

Key words. methods: data analysis – methods: numerical – techniques: miscellaneous – techniques: imaging spectroscopy – methods: statistical – gamma rays: general

1. Introduction

The SPI X/γ-ray spectrometer, on-board the INTEGRAL observatory is dedicated to the analysis of both point-sources and diffuse emission (Vedrenne et al. 2003). The sky imaging is indirect and relies on a coded-mask aperture associated to a specific observation strategy that is based on a dithering procedure. Dithering is needed since a single exposure does not always provide enough information or data to reconstruct the sky region viewed through the ~30° field-of-view (FoV), which may contain hundreds of sources for the most crowded regions. It consists of small shifts in the pointing direction between exposures, the grouping of which allows us to increase the amount of available information on a given sky target through a growing set of “non-redundant” data. The standard data analysis consists of adjusting a model of the sky convolved with the instrument transfer function plus instrumental background to the data.

However, source intensities vary between exposures. Thus, a reliable modeling of source variability, at least of the most intense ones, is needed to accurately model the data and intensity measurement of the sources. In addition, ignoring these intensity variations can prevent the detection of the weakest sources.

Ideally, the system of equations that connects the data to the sky model (unknowns) should be solved for intensities as well as for the variability timescales of sources and background. We propose methods based on data segmentation to fulfill these requirements as best as possible and we report in detail on the improved description and treatment of source variability during the data reduction process.

2. Functioning of SPI: necessity of modeling the intensity variations of sources

2.1. Instrument

SPI is a spectrometer provided with an imaging system sensitive to both point-sources and extended source/diffuse emission. The instrumental characteristics and performance are described in Vedrenne et al. (2003) and Roques et al. (2003). Data are collected between 20 keV and 8 MeV using 19 high-purity Ge detectors illuminated by the sky through a coded mask. The

[★] Based on observations with INTEGRAL, an ESA project with instruments and science data centre funded by ESA member states (especially the PI countries: Denmark, France, Germany, Italy, Spain, and Switzerland), Czech Republic and Poland with participation of Russia and the USA.

resulting FoV is $\sim 30^\circ$. The instrument can locate intense sources with an accuracy of a few arc minutes (Dubath et al. 2005).

2.2. SPI imaging specifics and necessity of dithering

A single exposure provides only 19 data values (one per Ge detector), which is not enough to correctly sample the sky image. Indeed, dividing the sky into $\sim 2^\circ$ pixels (the instrumental angular resolution) yields about 225 unknowns for a 30° FoV (i.e., $(30^\circ/2^\circ)^2$). Hence, reconstructing the complete sky image enclosed in the FoV is not possible from a single exposure because the related system of equations is undetermined. From these 19 data values, and the parameters describing the instrumental background we can, at best, determine the intensity of only a few sources. This is sufficient for sparsely populated sky regions and for obtaining a related coarse image (the intensities of the few pixels corresponding to the source positions). However, some of the observed sky regions are crowded (the Galactic center region, for example, contains hundreds of sources) and obtaining more data is mandatory for determining all parameters of the sky model. The dithering technique (Sect. 2.3) allows to obtain this larger set of data through multiple exposures.

2.3. Dithering

By introducing multiple exposures of the same field of the sky that are shifted by an offset that is small compared to the FoV size, the number of data related to this sky region is increased. In the dithering strategy used for INTEGRAL, the spacecraft continuously follows a dithering pattern throughout an observation (Jensen et al. 2003). In general, the exposure direction varies around the pointed target by steps of 2° within a five-by-five square or a seven-point hexagonal pattern. An exposure lasts between 30 min and 1 h.

Through grouping of exposures related to a given sky region, the dithering technique allows one to increase the number of “non-redundant” data points¹ without significantly increasing the size of the FoV that is spanned by these exposures. This provides enough data to recover the source intensity by solving the related system of equations. The source intensity variations within an exposure do not constitute a problem with the coded mask imaging system. The difficulty comes from the intensity variations between exposures. Hence, in our case, sources are variable on various timescales, ranging from an hour (roughly the duration of an exposure) to years. This information should be included in the system of equations to be solved.

2.4. Handling source intensity variation

We chose to model the intensity variation of a source as a succession of piecewise constant segments of time. In each of the segments (also called “time bins”), the intensity of the source is supposed to be stable. The higher energy bands ($E \gtrsim 100$ keV) contain a few emitting/detectable sources. Their intensities are rather stable with time according to the statistics and only a few piecewise constant segments of time or “time bins” are needed to model source intensity variations.

¹ The measurements correspond to data points that are not completely independent, if one considers that the shadow (of the mask projected onto the camera by a source) just moves as a function of the pointing direction of the exposure. Dithering is also used to extend the FoV and to separate the background from the source emission. It does not always provide a better coding of the sky.

At lower energies ($E \lesssim 100$ keV), the source intensity varies more rapidly. When inappropriate timescales are used for the sources contained in the resulting large FoV ($>30^\circ$) and for the background, the model does not fit the data satisfactorily (the chi-squares of the corresponding least-squares problem (Appendix A) can be relatively high). Moreover, allowing all sources to vary on the timescale of an exposure is an inappropriate strategy because, for crowded regions of the sky the problems to be solved are again undetermined in most cases. Generally, to estimate the source variability timescale, a crude and straightforward technique consists of testing several mean timescale values until the reduced chi-square of the associated least-squares problem is about 1 or does not decrease anymore. Unfortunately, this method is rapidly limited. Including variability in the system of equations always increases the number of unknowns that need to be determined (Appendix A) since the intensity in each segment of time (“time bins”) is to be determined. Using too many “time bins” will increase the solution variance and does not necessarily produce the best solution. Similarly, it is better to limit the number of “time bins” to the strict necessary minimum to set up a well-conditioned system of equations.

When one manually defines the “time bins”, one might soon be overwhelmed by the many timescales to be tested and the number of sources. It is difficult to simultaneously search for the variability of all sources contained in the instrument FoV and to test the various combinations of “time bins” (not necessarily the same length or duration, location in time). As a consequence, constructing light curves turns out to be rather subjective and tedious, and relies most of the time on some a priori knowledge. To make this step more rational, we propose methods based on a partition of the data into segments, to model source intensity variations between exposures.

3. Material

The objective is to find the “time bins” or segment sizes and locations corresponding to some data. This is related to the partition of an interval of time into segments. We propose two methods. The first one, called “image space”, relies on some already available light curves (or equivalently on a time series)². In our application the time series come mainly from INTEGRAL/IBIS and from *Swift*/BAT (Barthelmy et al. 2005). The purpose is to simplify an original light curve to maximize the source signal-to-noise ratio (S/N), hence to reduce the number of “time bins” through minimizing of the number of time segments³. These “time bins” are used to set up the SPI system of equations. This partitioning is made for all sources in the FoV. This algorithm is not completely controlled by SPI data, but at least it allows one to obtain a reasonable agreement (quantified by the chi-square value) between the data and the sky model.

The second method, called “data space”, starts from the SPI data and uses the instrument transfer function. In contrast to the “image-space” method where the segmentation is made for a single source only, the “data-space” method performs the

² In some cases it is also possible to use SPI light curves directly, for example when the FoV contains only a few sources. A first analysis exposure-by-exposure provides a light curve.

³ The number of naturally defined “time bins” using for example the IBIS “imager”, is the number of exposures (hence, exposure timescales), which can exceed the number of SPI data if we have more than 19 sources (which corresponds to the number of detectors) in the FoV varying on the exposure timescale), leading to an undetermined system of equations. In short, these light curves on the exposure timescale cannot be used directly.

Table 1. Index of mathematical symbols.

n_p	number of exposures
n_d	number of detectors (data) per exposure
N	number of parameters (to determine)
L	number of data points in a time series
M	number of data points $M = \sum_{i=1}^{n_p} n_d(i)$
D_p	data recorded during the exposure (pointing) p D_p is a vector of length $n_d(p)$
y	data sorted in sequential order (vector of length M) $y \equiv (D_1, D_2, \dots, D_{n_p})$ where D_p are the data points accumulated during exposure p .
x	solution of the equation $Hx = y$ (vector of length N). Also time series (“image space”) $x \equiv x_{1:L} = (x_1, \dots, x_L)$ where x_i are supposed sequentially ordered in time
H	real rectangular matrix of size $M \times N$
$H^{(J)}$	the submatrix (columns of H) related to source named or numbered J , such that $H \equiv [H^{(1)}, \dots, H^{(N)}]$
A	square real symmetric matrix of order N ($A = H^T H$)
ϵ	measurement errors on x (time series vector of length L) assumed to have zero mean, to be independent and normally distributed with a known variance Σ ($\epsilon_i \sim N(0, [\sigma_i^2])$)
T	t_i is the time where the data point x_i is measured with measurement error ϵ_i . $T = (t_1, \dots, t_L)$.
$f(t_i)$	the model of a series, this is a piecewise constant function.
m	number of change points that defined the $m + 1$ segments such that $f(t)$ is constant between two successive change points
$n_{\text{seg}}(J)$	number of segments to describe source named or numbered J
τ	change points such that $\tau_0 = \min(T)$ and $\tau_{m+1} = \max(T)$ or in point number units, $\tau_0 = 1$ and $\tau_{m+1} = L + 1$ ($\tau_0 < \tau_1 < \dots < \tau_{m+1}$). τ^* denotes the last found change point
s	s_k is the value of the model $f(t)$ for the segment defined by $\tau_{k-1} \leq t < \tau_k$ (vector of length $m + 1$)
$F(\tau)$	recursive function to be minimized with respect to τ
β	penalty to prevent overfitting
$x_{(\tau_{i-1}+1):\tau_i}$	subset of the vector x , $x_{(\tau_{i-1}+1):\tau_i} = (x_{\tau_{i-1}+1}, \dots, x_{\tau_i})$
$C(x_{(\tau_{i-1}+1):\tau_i})$	Cost function or effective fitness for the vector x subset $x_{(\tau_{i-1}+1):\tau_i}$
n	Iteration number which corresponds to time t_n

Notes. Initial state is denoted with the upper subscript 0 : (H^0, N^0, \dots). The temporary state, with the upper subscript * : ($H^*, x^*, N^*, n_{\text{seg}}^*, \dots$). The time t is sometimes replaced by the iteration number n .

time interval partitioning simultaneously for all sources in the FoV and is used to separate their individual contributions to the data. While this is more complex, the great advantage is that it is based solely on SPI data (Sect. 5).

4. Theory and calculation

4.1. The “image-space” algorithm – partition of a time series

The partition of an interval into segments is closely related to the topic of change points, which is widely discussed in the literature. There is a variety of efficient ways to analyze a time series if the parameters associated with each segment are independent (Fearhead 2006; Hutter 2007, and references therein). Scargle (1998) proposed an algorithm for best data partitioning within an interval based on Bayesian statistics. Applications to astronomical time series (BATSE bursts characterization) can be found in Scargle (1998).

4.2. Problem formulation

A list of notations used throughout this paper is given in Table 1. We consider the time series $x \equiv x_{1:L} = (x_1, \dots, x_L)$, comprising L sequential elements, following the model

$$x_i \equiv f(t_i) + \epsilon_i \quad i = 1, 2, \dots, L, \quad (1)$$

where x_i are the measured data and ϵ_i their measurement errors. The data are assumed to be ordered in time, although they may be spaced irregularly, meaning that each x_i is associated with a time t_i and contained in a time interval $T = (t_1, \dots, t_L)$. $f(t_i)$ is the model to be determined. We chose to model the time series as a combination of piecewise constant segments of time or blocks. This set of non-overlapping blocks that add up to form the whole interval forms a partition of the interval T . Hence there are, m change points $\tau_{1:m} = (\tau_1, \dots, \tau_m)$, which define $m + 1$ segments, such that the function $f(t)$ is constant between two successive change points,

$$f = \sum_{k=1}^{m+1} s_k \mathcal{I}_k \quad \text{with} \quad \begin{cases} \mathcal{I}_k = 1 & \text{if } t \in [\tau_{k-1}, \tau_k[\\ \mathcal{I}_k = 0 & \text{otherwise.} \end{cases} \quad (2)$$

Here $\tau_0 = \min(T)$ and $\tau_{m+1} = \max(T)$ or, equivalently, in point number units, $\tau_0 = 1$ and $\tau_{m+1} = L + 1$ ($\tau_0 < \tau_1 < \dots < \tau_{m+1}$). Since these data are always corrupted by observational errors, the aim is to distinguish statistically significant variations through the unavoidable random noise fluctuations. Hence, the problem is fitting a function through a noisy one-dimensional series, where the function f to be recovered is assumed to be a piecewise constant function of unknown segment numbers and locations (see Appendix A).

4.3. Algorithm

Yao (1984) and Jackson et al. (2005) proposed a dynamic programming algorithm to explore these partitions. The algorithm described in Jackson et al. (2005) finds the exact global optimum for any block additive fitness function (additive independent segments) and determines the number and location of the necessary segments. For L data points, there are 2^{L-1} possible partitions. In short, these authors proposed a search method that aims at minimizing the following function (see also Killick et al. 2012):

$$\min_{\tau} \left\{ \sum_{i=1}^{m+1} [C(x_{(\tau_{i-1}+1):\tau_i}) + \beta] \right\}, \quad (3)$$

where C represents some cost function. The i th segment contains the data block $x_{(\tau_{i-1}+1):\tau_i} = (x_{\tau_{i-1}+1}, \dots, x_{\tau_i})$ (in point number units), and the cost function or effective fitness for this data block is $C(x_{(\tau_{i-1}+1):\tau_i})$. The negative log-likelihood, for example the chi-square, is a commonly used cost function in the change point literature, but other cost functions can be used instead (e.g. Killick et al. 2012, and references therein). β is a penalty to prevent overfitting. Jackson et al. (2005, but see also Killick et al. 2012) proposed a convenient recursive expression to build the partition in L passes,

$$F(n) = \min_{\tau} \{F(\tau^*) + C(x_{(\tau^*+1):n}) + \beta\} \quad n = 1, \dots, L. \quad (4)$$

This enables calculating the global best segmentation using best segmentations on subsets of the data. Once the best segmentation for the data subset $x_{1:\tau^*}$ has been identified, it is used to infer the best segmentation for data $x_{1:\tau^*+1}$.

At each step of the recursion, the best segmentation up to τ^* is stored; for that, the position of the last change point τ^* is recorded in a vector (Jackson et al. 2005). By backtracking this vector, the positions of all change points that constitute the best segmentation at a given time or iteration number can be recovered. Figure 1 shows a flowchart of the algorithm. Our implementation is essentially the same as that of Jackson et al. (2005), but see also Killick et al. (2012). The best partition of the data is found in $O(L^2)$ evaluations of the cost function.

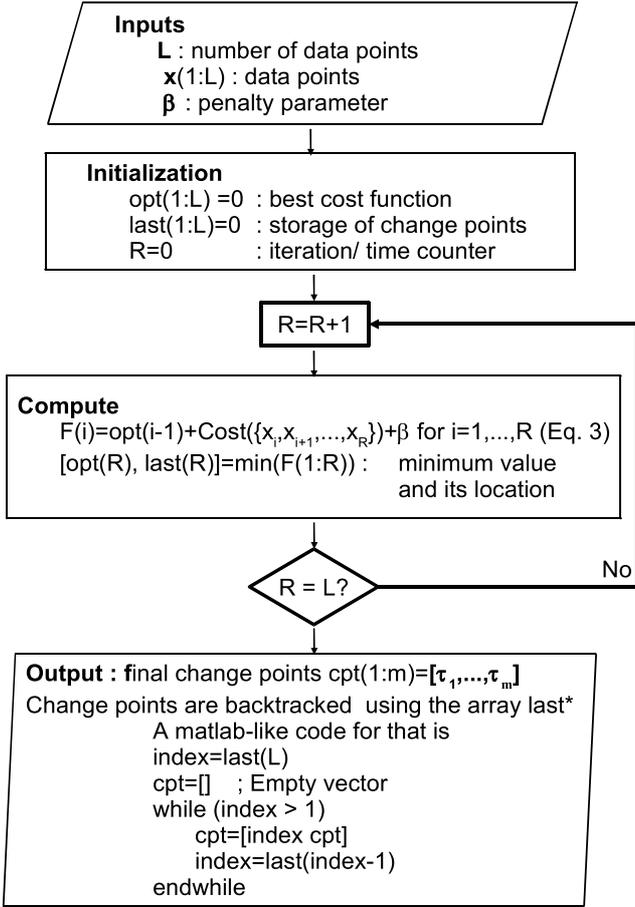


Fig. 1. Flowchart of the “image-space” algorithm (pseudo-code in Appendix F.1). The vector *last* gives the starting point of individual segments of the best partition in the following way; let $\tau_m = \text{last}(L)$, $\tau_{m-1} = \text{last}(\tau_m - 1)$ and so on. The last segment contains cells (data points) $\tau_m, \tau_m + 1, \dots, n$, the next-to-last contains cells $\tau_{m-1}, \tau_{m-1} + 1, \dots, \tau_m - 1$, and so on.

5. “Data-space” algorithm

The SPI data contain the variable signals of several sources. The contribution of each of the sources to the data through a transfer function is to be retrieved. As in Sect. 4.1 and for each of the sources, the number and positions of segments are the parameters to be estimated, but this time, the estimates are interdependent because of the nature of the instrument coding. While many algorithms exist to analyze models where the parameters associated with each segment are independent of those linked to the other segments, there are only a few approaches, not of general applicability, where the parameters are dependent from one segment to another (Punskaya et al. 2002; Fearnhead & Liu 2011, and references therein). However, our problem involves a transfer function, which requires a specific treatment.

5.1. Contribution of the sources to the data

The elements related to source J are denoted with the upper subscript (J) . The variable signal of the source numbered J is represented by $n_{\text{seg}}(J)$ ($= m^{(J)} + 1$) piecewise constant segments (“time bins”) delimited by $0 \leq m^{(J)} \leq n_p - 1$ change points (Sect. 4.2). The response $H^0(1 : M, J)$, which corresponds to a constant source intensity, is a vector of length M ($= \sum_{i=1}^{n_p} n_d(i)$).

This vector is expanded into a submatrix with $n_{\text{seg}}(J)$ orthogonal columns. The signal can be written (see also Eq. (2))

$$x^0(J) \rightarrow f^{(J)}(t) = \sum_{k=1}^{m^{(J)}+1} s_k^{(J)} \mathcal{I}_k^{(J)}(t) \text{ with}$$

$$\begin{cases} \mathcal{I}_k^{(J)}(t) = 1 & \text{if } t \in [\tau_{k-1}^{(J)}, \tau_k^{(J)}[\\ \mathcal{I}_k^{(J)}(t) = 0 & \text{otherwise.} \end{cases}$$

The corresponding predicted contribution to the data, at time t (in units of exposure number), is

$$\sum_{k=1}^{n_{\text{seg}}(J)} H^{(J)}(t, k) s_k^{(J)} t = 1, \dots, n_p.$$

The submatrices $H^{(J)}(t, k)$ of dimension $n_d(t) \times n_{\text{seg}}(J)$ are derived from the response vector $H^0(1 : M, J)$ as follows:

$$H^{(J)}(t, k) = \begin{cases} H^0(i, J) & \text{if } \sum_{q=1}^{\tau_{k-1}^{(J)}} n_d(q) + 1 \leq i \leq \sum_{q=1}^{\tau_k^{(J)}} n_d(q) \\ 0 & \text{otherwise.} \end{cases}$$

The response associated with the N^0 sources is obtained by grouping subarrays $H^{(J)}$ of all sources,

$$H \equiv [H^{(1)}, \dots, H^{(N^0)}]. \quad (5)$$

Matrix H has size $M \times N$ with $N = \sum_{J=1}^{N^0} n_{\text{seg}}(J)$. Therefore, the predicted contribution of the N^0 variable sources to the data is

$$\sum_{j=1}^N H(i, j) x(j) \quad i = 1, \dots, M,$$

where $x = (s_{1:n_{\text{seg}}(1)}^{(1)}, \dots, s_{1:n_{\text{seg}}(N^0)}^{(N^0)})$ is the flux in the N “time bins” (see also Appendix A).

5.2. Algorithm

The search path or the number of possible partitions for N^0 sources and the n_p exposures is $(2^{n_p-1})^{N^0}$. An exhaustive search is not feasible for our application. In addition, computing the cost function for a particular partition implicitly involves solving a system of equations (computing the cost function), which renders the search even more inefficient. We essentially make a simplification or approximation that reduces the computation time, allows significant optimizations, and hence makes the problem tractable.

Instead of searching for the exact change points, we can settle for reliable segments and hence derive an algorithm that aims at significantly reducing the search path. Rather than exploring the space for all sources simultaneously, we explore the reduced space associated to a single source at a time. The algorithm closely follows the pattern of Eq. (4). It is an iterative construction of time segments that processes the sources sequentially and independently. In practice, the best set of change points using the data recorded up to time $t - 1$ is stored. The $n_d(t)$ new data points obtained at time t are added to the dataset. The procedure starts for the best change points recorded so far. For a given source, an additional change point is added at the various possible positions. Its best position is recorded along with the best new cost function. The operation is repeated for all sources. The best change point (and cost function value) of all sources

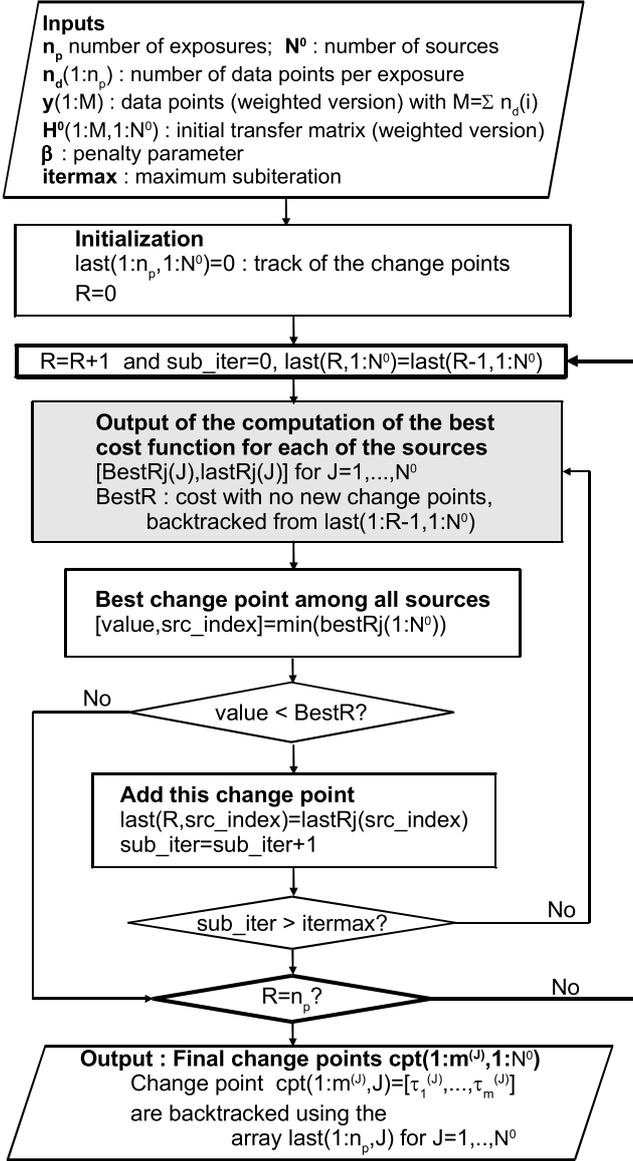


Fig. 2. Flowchart of the “data-space” algorithm (pseudo-code in Appendix F.2). The cost function calculation, here the shaded box, is detailed in Fig. 3.

is kept (location and source number) and added to the growing set of change points. This procedure is repeated several times, for instance $N_{\text{iter}} (\leq N^0)$ subiteration) times, until no more change points are detected or until the cost function cannot be improved any further. Then, another iteration at $t + 1$ can start. Figure 2 schematically shows the resulting procedure.

This experimental approach is exact when the columns of the transfer matrix are orthogonal. Although there is no mathematical proof that this holds for a matrix that does not have this property, it is efficient on SPI transfer function and data (Sect. 6.3).

5.3. Cost function calculation

The computation of the cost function, used to adjust the data to the model, is detailed in Appendix B. The derived expression involves the minimization of the chi-square function. Indeed, the cost function must be computed many times, each time that a new partition is tested. This is by far the most time-consuming

part of the algorithm. Fortunately, these calculations can be optimized.

At iteration n , for a subiteration, we have to solve about $n \times N^0$ systems of equations to compute the cost functions corresponding to the different change-point positions. Instead of solving all these systems, it is possible to solve only a few of them and/or reduce the number of arithmetic operations by relying on the structure of the matrices⁴.

The matrix \mathbf{H} of size $M \times N$ using the change points obtained at iteration $n - 1$ is the starting point. To search for the best partition for source J , we have to replace only the submatrix $\mathbf{H}^{(J)}$ by its updated version $\mathbf{H}^{*(J)}$. The corresponding transfer matrix \mathbf{H}^* (of dimension $M \times N^*$) can be written (Eq. (5)) as

$$\mathbf{H}^*(1 : M, 1 : N^*) \equiv [\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(J-1)}, \mathbf{H}^{*(J)}, \mathbf{H}^{(J+1)}, \dots, \mathbf{H}^{(N^0)}].$$

The dimension of the submatrix $\mathbf{H}^{*(J)}$ corresponding to the new data partition is $M \times n_{\text{seg}}^*(J)$. Here $n_{\text{seg}}^*(J)$ is the resulting number of segments for source J . It varies from 1, which corresponds to a single segment for the source J , to the $n_{\text{seg}}^{\text{max}}(J) + 1$, where $n_{\text{seg}}^{\text{max}}(J)$ is the highest number of segments obtained so far for source J for iteration up to $n - 1$. In most cases, only the few last columns of the matrix $\mathbf{H}^{*(J)}$ are not identical to those of $\mathbf{H}^{(J)}$, since the very first columns tend to be frozen (Sect. 5.4). If $n_{\text{seg}}^{\text{keep}}$ is the number of consecutive identical columns, then

$$\mathbf{H}^{*(J)} \equiv [\mathbf{H}^{(J)}(1 : M, 1 : n_{\text{seg}}^{\text{keep}}), \mathbf{H}_{\text{add}}(1 : M, 1 : N_{\text{add}})].$$

Here \mathbf{H}_{add} is the modified part of $\mathbf{H}^{(J)}$. Let $k_{\text{min}}(J) = \sum_{k=1}^{J-1} n_{\text{seg}}(k)$ and $k_{\text{max}}(J) = \sum_{k=1}^J n_{\text{seg}}(k)$. Then, the columns, from $k_{\text{min}}(J) + n_{\text{seg}}^{\text{keep}} + 1$ to $k_{\text{max}}(J)$ are deleted in the matrix \mathbf{H} and N_{add} columns are added at position $k_{\text{min}}(J) + n_{\text{seg}}^{\text{keep}} + 1$ to $k_{\text{min}}(J) + n_{\text{seg}}^{\text{keep}} + N_{\text{add}}$. Therefore, it is not necessary to compute the full matrix \mathbf{H}^* from scratch each time a new change point is tested (position and source number), but simply to update some columns of the matrix \mathbf{H} . This construction of the transfer function significantly saves on run time and provides clues to further optimization.

However, in our application, we solve a least-squares system, and hence the matrix $\mathbf{A} = \mathbf{H}^T \mathbf{H}$ ($\mathbf{A}^* = \mathbf{H}^{*T} \mathbf{H}^*$) is to be implicitly inverted. We use an expression based on the Sherman-Morrison-Woodbury formula and the matrix partition lemma (E) to obtain \mathbf{A}^* from \mathbf{A} after subtracting N_{sub} and added N_{add} columns to \mathbf{H} (Appendix E). With this optimization, we need only one matrix decomposition ($\mathbf{A} = \mathbf{L} \mathbf{L}^T$ where \mathbf{L} is a lower triangular matrix) of a symmetric sparse matrix⁵ \mathbf{A} of order N per iteration and solve the system ($\mathbf{A} \mathbf{x} = \mathbf{H}^T \mathbf{y}$) (store factors and solution). Then, for each tested change-point position tested, only one system of equations is to be solved (using the matrix decomposition stored previously), other operations are relatively inexpensive. To compute the cost function, it can be necessary for our application, depending on the expression of the cost function, to compute a determinant. However, in our application, only the ratio of the determinant of the created matrix (\mathbf{A}^*) of rank $N^* = N + N_{\text{add}} - N_{\text{sub}}$ to matrix \mathbf{A} of rank N matters. This ratio is very stable because it involves only matrices of small rank. In addition, as a cross-product of the update of \mathbf{A} , it is inexpensive to compute.

⁴ In our implementation, the matrices are sparse and the estimated search path number and cost in arithmetic operations are lower than those given in this section, which correspond to full matrices.

⁵ We used the MUMPS (MUltifrontal Massively Parallel Solver) package (<http://mumps.enseeiht.fr/> or <http://graal.ens-lyon.fr/MUMPS>). The software provides stable and reliable factors and can process indefinite symmetric matrices. We used a sequential implementation of MUMPS.

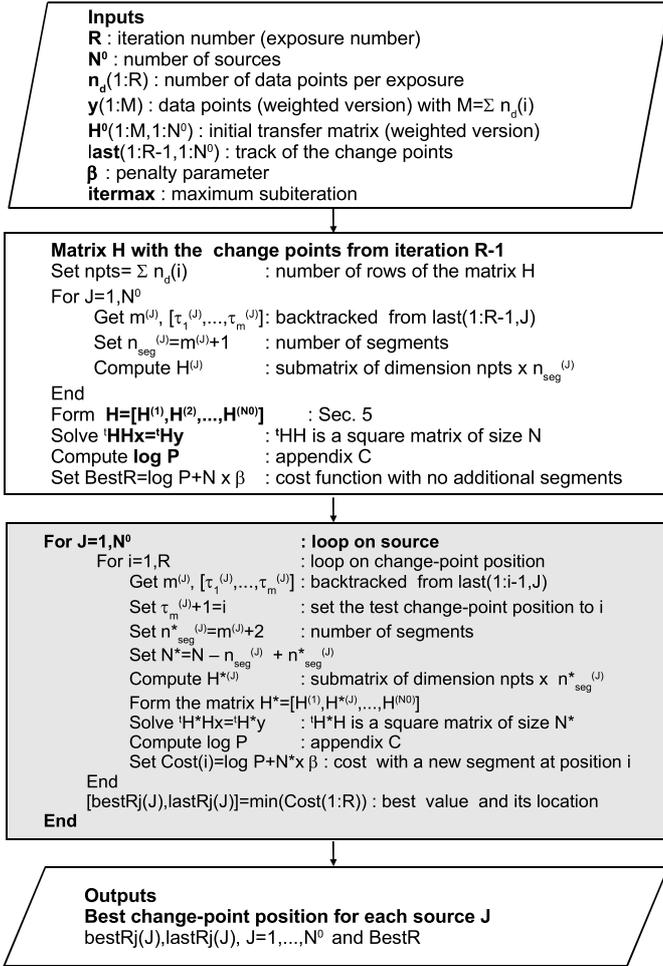


Fig. 3. Flowchart of the calculation of the cost function for the “data-space” algorithm.

5.4. “Limited memory” version

The number of possible partitions, and hence possible change-point positions, increases with the number of exposures (Sect. 5.2). This makes the problem difficult to solve in terms of computing time for data sets containing a large number of exposures. The positions of the change points are not necessarily fixed until the last iteration. In the worst case, there can be no common change points (hence segments) between two successive iterations. In practice, it turns out that they become more or less frozen once a few successive change points have been located. Then, for each source, instead of exploring the n possible positions of the change point, we consider only the data accumulated between iteration $n_0 > 1$ and n (The “full” algorithm starts at $n_0 = 1$). This defines a smaller exploration window size, which further reduces the number of cost function evaluations (in the shaded box of Fig. 3, the loop on change-point positions starts at $i = n_0 > 1$). The number of change-point positions to test is nearly stable as a function of the number of accumulated exposures. Section 6.3.4 compares the “limited memory” and “full” version.

5.5. Parallelization

The algorithm sequentially divides each source interval into segments. Hence, it is straightforward to parallelize the code and process several sources simultaneously. To do so, the “source

loop” (in the shaded box of Fig. 3, the loop on J) is parallelized using Open-MP. Matrix operations are performed in sequential mode, although a parallel implementation is also possible. Experiments on parallelization are presented in Sect. 6.3.3.

6. Results

The sky model for each dataset consists of sources plus the instrumental background, whose intensity variations are to be determined. The background is also the dominant contributor to the observed detector counts or data. Its spatial structure on the detector plane (relative count rates of the detectors) is assumed to be known thanks to “empty-field” observations, but its intensity is variable (Appendix A). By default, its intensity variability timescale is fixed to ~ 6 h, which was shown to be relevant for SPI (Bouchet et al. 2010). Its intensity variation can also be computed, as for a source, with the “data-space” algorithm.

Owing to observational constraints (solar panel orientation toward the Sun, etc.), a given source is “visible” only every ~ 6 months and is observed only during specific dedicated exposures (Fig. 4). For these reasons, it is more convenient to present the “light curves” as a function of the exposure number, so the abscissa does not actually reflect the time directly.

The fluxes obtained with SPI and IBIS in each defined “time bin”, are also compared whenever possible. However, IBIS light curves are provided by a “quick-look analysis” and are intended to give an approximate measurement of the source intensity. These time series, used as external data input to the “image-space” method, are not fully converted into units that are independent of the instrument (such as photons $\text{cm}^2 \text{s}^{-1}$) and can even be affected by the specific characteristics of the instrument.

6.1. Datasets/training data

The various datasets used in this paper are shorter subsets related to a given sky region of a larger database that consists of $\sim 39\,000$ exposures covering six years of observations of the entire sky (Bouchet et al. 2011). An SPI dataset, related to a particular source, consists of all exposures whose angular distance, between the telescope pointing axis and the source of interest direction (central source), is less than 15° . This procedure gathers the maximum number of exposures containing the signal from the source of interest, but at the same time the datasets span a $\sim 30^\circ$ radius FoV sky region containing numerous sources.

Concerning the energy bandwidth of the datasets, the transfer function is computed for the same mean energy. However, because of the spectral evolution of the source with time, this approximation introduces some inaccuracies that increase with the energy bandwidth. At first glance, it is preferable to use a narrow energy band, but at the same time, the S/N of the sources are lower (since the error bars on the fluxes are large enough, we detected virtually no variation in intensity). In other words, for a narrower energy band, the reduced chi-square between the data and the model of the sky is lower, but at the same time, the source intensity variations are less marked. Finally, as a compromise for our demonstration purpose, we used the 25–50 keV and 27–36 keV energy bands.

6.2. “Image-space” method

6.2.1. Application framework

We rely on auxiliary information, such as available light curves, to define the “time bins” for SPI. These light curves come from

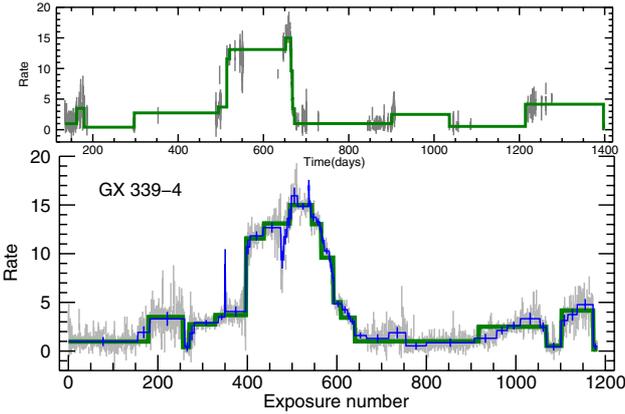


Fig. 4. GX 339-4 intensity variations. The 26–40 keV IBIS time series (gray), which contains 1183 data points (one measurement per exposure), is segmented into 17 constant segments or “time bins” (green). The $\chi_r^{2(i)}$ between the time series and its segmented version is 1.0006 for 1166 d.o.f.. These curves are plotted as a function of time (*top*) and the exposure number (*bottom*). The raw time series (without S/N scaling) is directly segmented into 46 segments (bottom blue curve).

the IBIS “quick-look analysis”⁶, more precisely, its first detector layer ISGRI (Ubertini et al. 2003). The light curves of the known sources are not all available, but usually the strongest ones are, if not always exactly in the requested energy bands, but this is sufficient to define some reliable “time bins”.

The “image-space” method consists of two separate steps. The first does not necessarily involve the SPI instrument, while the second one involves the SPI only.

6.2.2. “Time bins”: segmentation of an existing time series

The basic process to set up the “time bin” characteristics (position and length) is the time series segmentation. The raw time series is the light curve linked to a given source, which comes from the “quick-look analysis” of IBIS data. Below 100 keV, IBIS is more sensitive than SPI. However, it depends on the source position relative to the instrument pointing axis, the number of sources in the FoV, but also on the source’s spectral shape. The measured S/N of the two instruments for strong sources is around ~ 2 and increased to ~ 3 for crowded sky regions. To have roughly similar S/N per sources between IBIS and SPI, especially in crowded sky regions, random Gaussian statistical fluctuations are added to raw time series to obtain statistical error bars on IBIS light curves increased roughly by a factor 3 below 100 keV. This forms the time series used. This ad hoc procedure is also retrospectively coherent with the number of segments obtained with the “data-space” method (Sect. 6.5). Above 150 keV SPI is more sensitive, but at these energies many sources can be considered to be stable in intensity because their detection significance is low (fluxes have large error bars), hence we do not expect as many “time bins”.

In principle, the model of the time series, hence the number of segments, can be adjusted to the data with the desired chi-square value by varying the penalty parameter β of Eq. (3). Intuitively, one readily sees that a too high value will underfit, while a too low value will overfit the data. To be conservative, we adjusted this parameter to obtain a reduced chi-square ($\chi_r^{2(i)}$) of 1; the segmentation algorithm is fast enough to test several values of β to achieve this objective.

⁶ Provided by the INTEGRAL Science Data Centre (ISDC).

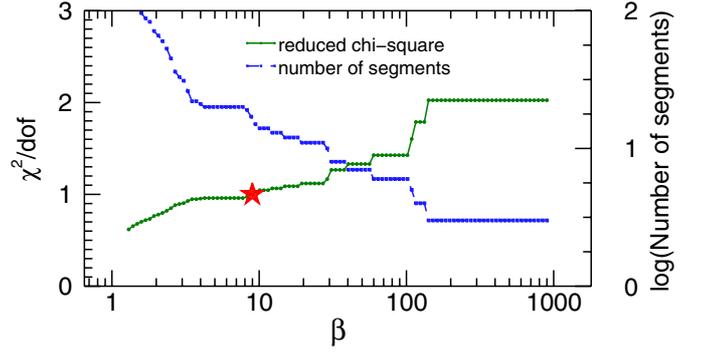


Fig. 5. GX 339-4 time series segmentation. Variation of the $\chi_r^{2(i)}$ (green circles) and number of segments (blue squares) as a function of the penalty parameter value for the time series of Fig. 4. The red star indicates the value of the penalty parameter ($\beta = 8.99$) where $\chi_r^{2(i)} = 1.0006$ and which corresponds to 17 segments.

We proceeded as follows: if a single segment (constant flux) gives $\chi_r^{2(i)}$ above 1, the time series was segmented. Many expressions of the cost function can be used; expressions C.3 and C.5 described in Appendix C give similar results. Finally, we chose the most simple one (C.3), which is the chi-square function.

Figures 4 and 5 show the application to the GX 339-4 source, whose intensity variation is characterized by a short rise time followed by an exponential decay on a timescale of about one month. The IBIS time series for the 26–40 keV band contains 1183 exposures. The $\chi_r^{2(i)}$ is 1.0006 for 1166 degrees of freedom (d.o.f.), the number of “time bins” is 17 for a penalty parameter value of 8.99 (if the IBIS raw time series is directly segmented⁷, the number of segments is 46 instead of 17).

6.2.3. Application to SPI

We applied the “image-space” algorithm to the common IBIS/SPI 27–36 keV datasets related to the GX 339-4, 4U 1700-377, and GX 1+4 sources, which contain 1183, 4112, and 4246 exposures. The persistent source 4U 1700-377 is variable on hour timescales, while GX 1+4 is variable on a timescale from days to about 1–2 months. In the sky model are 120, 142, and 140 sources included for the GX 339-4, 4U 1700-377, and GX 1+4 datasets.

For 4U 1700-377, the reduced chi-square (χ_r^2) between the predicted and measured data is 6.15; assuming that all sources have constant intensity, this value is clearly not acceptable. Each available IBIS individual light curve was segmented as indicated in Sect. 6.2.2 to define the “time bins”. A total number of 4005 “time bins” was obtained to describe the dataset, and the resulting 4U 1700-377 was partitioned into 2245 segments. The SPI-related system of equations (Appendix A) was then solved using these segments. The χ_r^2 is 1.28 for 68 455 d.o.f., which is a clear improvement compared with the value of 6.15 obtained when all sources are assumed to remain constant over time. It is also possible to directly use the “time bins” defined with the IBIS light curves without scaling the sensitivities of the two instruments (Sect. 6.2.2), in which case the χ_r^2 reaches 1.13. For strong and highly variable sources such as 4U 1700-377, one can fix the

⁷ The direct segmentation of IBIS time series will generally produce more “time bins” than necessary. This will overfit the SPI model of the source intensity variations. Obviously, this reduces the chi-square, but does not necessarily produce the best source S/N after SPI data processing.

Table 2. “Image-space” method final chi-square.

Dataset	4U 1700-377	GX 339-4*	GX 1+4*	4U 1700-377*
exposures	4112	1183	4246	4112
sources	142	120	140	142
All sources are assumed to have constant intensity				
χ_r^2 (d.o.f.)	6.15(71 594)	2.46(19 308)	1.81(69 361)	2.01(67 483)
n_{seg}	1	1	1	4112
“Time bins” from IBIS light curves after S/N scaling (Sect. 6.2.2)				
χ_r^2 (d.o.f.)	1.28(68 455)	1.186(18 880)	1.193(68 557)	1.186(66 588)
n_{seg}	2245	17	122	4112
“Time bins” from IBIS light curves directly				
χ_r^2 (d.o.f.)	1.132(65 768)	1.106(18 317)	1.143(66 939)	1.124(64 841)
n_{seg}	3185	46	675	4112

Notes. (*) The source 4U 1700-377 is also contained in the FoV and is set to be variable on the exposure duration timescale (~ 1 h).

Table 3. Characteristics of the datasets.

Dataset	Number of exposures	Energy range (keV)	Number of sources
V0332+53	391	25–50	17
GX 339-4 ^a	1183	27–36	120
GRS 1915+105	2980	27–36	61
Cyg X-1	2351	27–36	32
IGR J17464-3213	7147	27–36	132

Notes. (a) The dataset is restricted to common IBIS and SPI exposures.

variation timescale to the exposure duration (~ 1 h). All these results are summarized in Table 2.

For all three datasets, the source 4U 1700-377 was assumed to vary on the exposure timescale. Assuming that all other sources are constant in intensity yields a final χ_r^2 of 2.46, 1.81, and 2.01 for the GX 339-4, GX 1+4, and 4U 1700-377 datasets.

The segmentation of the available IBIS time series permits one to obtain better “time bins” and the χ_r^2 improves to values below 1.2 (Table 2). The resulting GX 339-4 and GX 1+4 light curves were partitioned into 17 and 122 segments (Table 2). Figure 6 shows the intensity variations of these two sources, obtained from SPI data.

The correlation of the fluxes in the “time bins” of the two instruments is purely indicative since IBIS time series are obtained from a “quick-look analysis”. However, the fluxes are well correlated (Fig. 8). The linear correlation coefficients are 0.995, 0.941, and 0.816, for GX 339-4, 4U 1700-377, and GX 1+4.

To illustrate the decreasing number of “time bins” with increasing energies, we used the GRS 1915+105 dataset. There are 1185 common IBIS and SPI exposures for this source in our database. For each energy band, the procedure is the same as before, but with 66 sources contained in the FoV. The number of “time bins” after segmentation of the IBIS time series in the 27–36, 49–66, 90–121 and 163–220 keV bands, is 174, 50, 8, and 1 and the χ_r^2 (d.o.f.), after solving the SPI related system of equations, is 1.20 (20 257), 1.05(20 545), 1.02(20 562), and 1.01(20 616). Figure 7 shows the resulting intensity variations.

6.3. “Data-space” method

6.3.1. Application framework

As examples, we built datasets related to the sources V0332+53, Cyg X-1, GRS 1915+105, GX 339-4, and IGR J17464-3213 as described in Sect. 6.2. The characteristics of these datasets are indicated in Table 3.

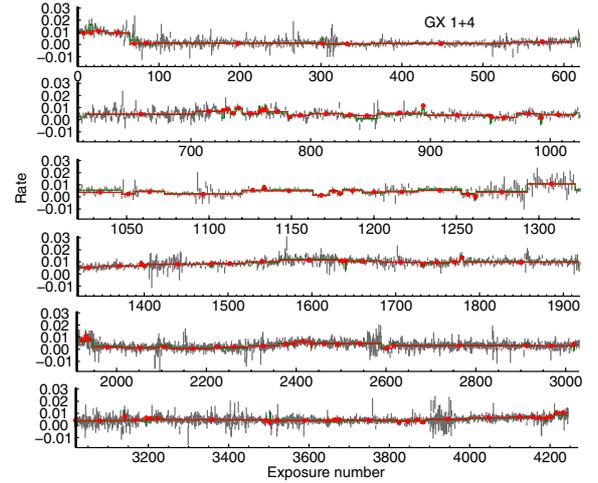
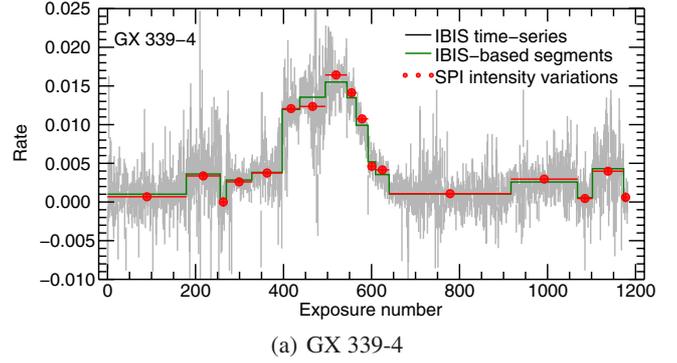


Fig. 6. Intensity variations of GX 339-4 and GX 1+4 in the 27–36 keV band as a function of the exposure number. The SPI segments are plotted in red and the IBIS raw light curves (26–40 keV) are drawn in gray. The segmented IBIS time series (scaled S/N) is shown in green. The count rate normalization between IBIS and SPI is arbitrary.

6.3.2. Cost function, penalty parameter, and chi-square

The penalty parameter value determines the final chi-square value between the data and the sky model, but its choice is more problematic than with the “image-space” algorithm. Figure 9 illustrates the influence of parameter β on the final number of segments and the χ_r^2 value for several datasets of Table 3. Because of the very schematic representation of the data, it is not always possible to reach a χ_r^2 value of 1 for all these datasets, but this value can be approached; if necessary, the width of the energy bands can be reduced (Sect. 6.1) to obtain a lower χ_r^2 value. Two expressions of the cost function are used using different assumptions (Appendix C), but there is no strong justification for preferring one expression over the other.

The simplest cost function expression C.8 is essentially the chi-square. The value of the penalty parameter β prescribed by the Bayesian information criterion is $\beta_0 \sim \log(M)$, M being the number of data points (Appendix D). A penalty parameter value lower than this prescription is required to reach a χ_r^2 value closer to 1. These lower values of β produce a greater number of “time bins”, which has the effect of reducing the chi-square at the expense of the reliability of the solution. However, the value β_0 remains a reliable guess value of the penalty parameter and gives a number of “time bins” similar to what is expected from the direct segmentation of the light curves (Sect. 6.5).

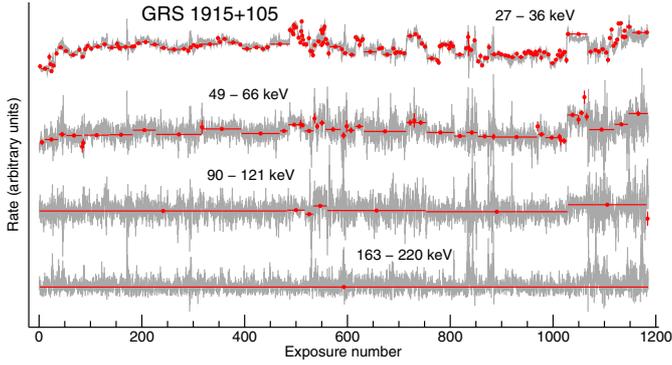


Fig. 7. “Image-space” method applied to GRS 1915+105. The intensity variations as a function of the exposure number are shown in the 27–36, 49–66, 90–121 and 163–220 keV band as indicated in the figure. The different bands are translated on-to the y -axis (the intensities are in arbitrary units). The SPI time series (scaled S/N) are drawn in gray (26–40, 51–63, 105–150, and 150–250 keV). The number of “time bins” deduced from the time series segmentation are 174, 50, 8, and 1. The global normalization factor in each energy band between the fluxes of the two instruments is arbitrary.

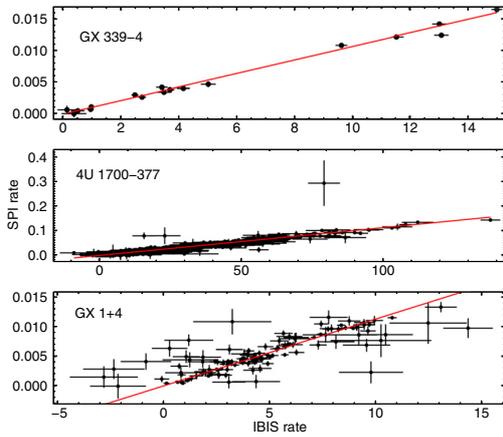


Fig. 8. IBIS/SPI flux correlation in 17, 2245, and 122 “time bins” for GX 339-4, 4U 1700-377, and GX 1+4. The red line is a straight-line fitting (with errors in both directions) with a chi-square of 25 (15 d.o.f.), 3961 (2243 d.o.f.), and 321 (120 d.o.f.).

Expression C.9 of the cost function has an additional term with respect to the expression C.8, which acts as a penalty parameter and further limits the increase of the number of “time bins”. The minimum cost function value is reached for penalty parameters β below one tenth of β_0 . A value $\beta = 0$ can be used since it gives the minimum chi-square or at least a value close to the lowest one. The χ_r^2 value is then comparable to the value obtained with expression C.8 and the Bayesian information criterion prescription for the value of β .

Figure 9 shows the evolution of the χ_r^2 and total number of segments to describe the intensity variations of the central source as a function of the penalty parameter β for several datasets and configurations of the “data-space” algorithm. The C.8 expression of the cost function permits one to reach a lower value of the chi-square compared to C.9. Tables 6 and 7 summarize the results obtained with the “data-space” method. We quite arbitrarily chose to use the expression C.9 as the baseline cost function with penalty parameter $\beta = 0$.

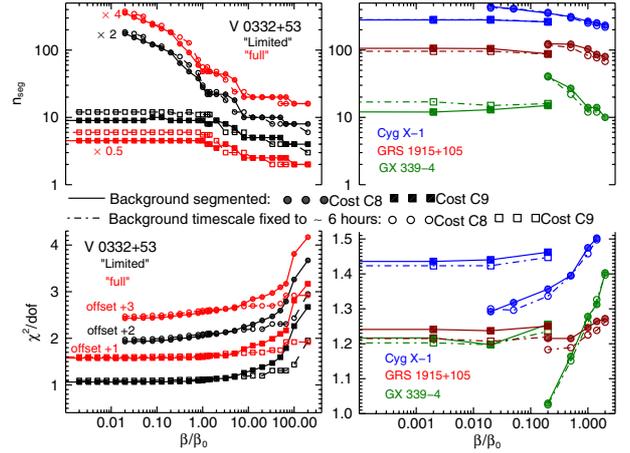


Fig. 9. (Bottom) Reduced chi-square (χ_r^2) as a function of the penalty parameter in units of $\beta_0 = \log(M)$, where M is the number of data points of the dataset. (Top) Number of segments to describe the intensity variation of the central source. Left panels are related to V0332+53 and right panels to GX 339-4, GRS 1915+105, and Cyg X-1. The results obtained with different configurations of the “limited version” of the “data-space” method (background timescale fixed to ~ 6 h, cost function expression C.8 or C.9) are compared with the default configuration (cost function C.9 and background timescale to be computed). The “full” version is compared with the ‘limited’ version of the “data-space” algorithm for the V0332+53 dataset.

Table 4. Gain in time achieved by “source loop” parallelization of the “data-space” algorithm.

Dataset	Algorithm version	Average time (s) per “source loop”					
		Number of threads					
		1	2	4	8	16	32
V0332+53 (391 exposures)	Full	22.8	15.7	10.1	8.1	7.7	5.2
	limited	3.1	2.0	1.8	1.1	1.0	0.7
GX 339-4 (1183 exposures)	Full				607		
	limited	119			43	30	23

Notes. The “limited” version is parameterized as indicated in Sect. 6.3.4. The dataset characteristics are detailed in Table 3. A computer with 64 (2.62 GHz) processors was used. The total number of “source loops” performed for the “full” version was 519 and 2359 for V0332+53 and GX 339-4. Similarly, 508 and 2296 loops were performed with the “limited memory” version. The total time spent in the “source loop” for the “full” (“limited memory”) “data-space” algorithm was obtained by multiplying the average single-loop time by the number of performed loops.

6.3.3. Parallelization

Table 4 shows experimental results for two datasets (V0332+53 and GX 339-4 (Table 3)). A system with 64 (2.62 GHz) cores was used. The run time was measured in the “source loop” (Sect. 5.5). Using 16 threads instead of 1 speeds up the computation by a factor ~ 3 for the V0332+53 dataset and by a factor ~ 4 for the GX 339-4 with the “limited memory” code. The gain is the same for the V0332+53 dataset with the “full” code, but the average time to perform a loop is about seven times higher than the “limited memory” code.

6.3.4. “Limited memory” and “full” version

The “limited memory” version (Sect. 5.4) is parameterized such that at iteration n , it starts the change-points position exploration

Table 5. Mean time spent in the “source loop” by the “full” and “limited memory” versions of the ‘data-space’ algorithm.

Iteration	“Source loop” duration (s)							Total time(s)
	20	50	125	250	500	1000	1183	
Full	6	18	50	107	288	1701	2748	1 432 864
“Limited”	8	18	20	20	30	107	100	98 746

Notes. The characteristics of the dataset related to GX 339-4 are detailed in Table 3. The “source loop” was parallelized on 8 threads of a 64 processors (2.62 GHz) machine. The “limited” version was parameterized as follows: $n_0 = \min(n - 30 \text{ exposures}, n - 2 \text{ change points backward})$, where n is the current exposure, individually obtained for each source as explained in Sect. 5.4.

at exposure $n_0 = \min(n - 30 \text{ exposures}, n - 2 \text{ change points backward})$, individually for each source.

For the GX 339-4 dataset, the average time spent in a single “source loop” by the “full” version increases with the iteration number (number of exposures processed) while it is comparable per iteration number with the “limited memory” version. Table 5 reports the gain in time achieved with the “limited memory” version compared with the full “data-space” algorithm. Finally, the “limited memory” version is almost ~ 15 times faster than the “full” version on the GX 339-4 dataset. Table 7 compares the results of the “limited memory” and “full” version for the V0332+53 and GX 339-4 and the GRS 1915+105 and Cyg X-1 datasets. The number of segments found by the “limited” version is in general slightly higher than the number found by the “full” version of the code; however, this effect is also due to the particular parametrization used for the “limited” version. The results remain quite similar in terms of number of segments and reached χ_r^2 , showing that the “limited memory” version is a good alternative to the full “data-space” algorithm, especially for the gain in computation time. The gain in time of the “limited” (with the above parametrization) over the “full” version reaches a factor ~ 30 when the background timescale is fixed (not to be segmented).

6.3.5. Background segmentation

It is no longer necessary to fix the variation timescale of instrumental background to about six hours. The “data-space” algorithm models the change in intensity of the background similarly to the other sources of the sky model (Table 7). If we let the algorithm determine the background intensity variation, it is modeled with a fewer number of segments than when the background variation is fixed for a quantitatively comparable chi-square. This is because of a better localization of the change points. The other parameters, such as the number and locations of the source “time bins”, stay essentially the same.

6.3.6. Application

The “limited memory” version, where the intensity variations of both sources and background are computed, was used as the default algorithm. For the dataset relative to V0332+53, this consisted of a χ_r^2 of 1.06 for 6595 d.o.f. and a total of 97 “time bins”. The resulting V0332+53 intensity evolution is displayed in Fig. 10 as a function of the exposure number (nine segments).

Next, we processed the highly variable source Cyg X-1. The dataset χ_r^2 is 1.44 for 40 068 d.o.f. and a total of 676 “time bins”. The number of segments needed to describe Cyg X-1 is 280. The relatively high value of the chi-square may be due to the strong

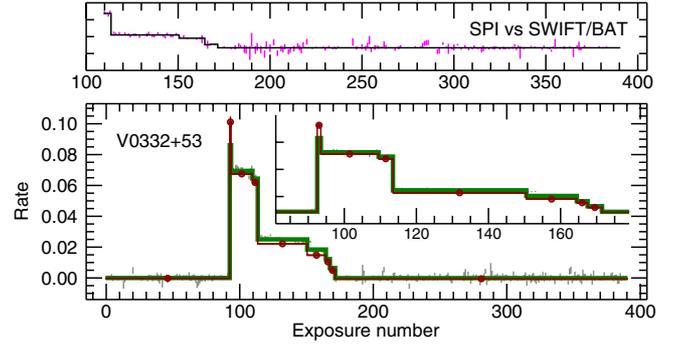


Fig. 10. “Data-space” method applied to V0332+53. The source intensity variations (25–50 keV) were modeled by nine segments (red) and were compared with the IBIS time series (26–51 keV, gray). The green curve corresponds to the IBIS flux averaged on SPI-obtained segments. The insert is a zoom between exposure number 81 and 179. (Top) SPI intensity variations model (black) compared with *Swift*/BAT time series (24–50 keV, purple line). The scale between the different instruments is arbitrary and was chosen such that their measured total fluxes are equal. It should be noted that the *Swift*/BAT and IBIS data are not necessarily recorded at the same time as the SPI data, nor exactly in the same energy band.

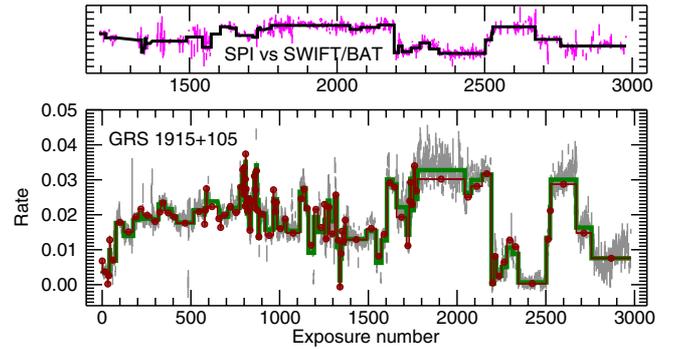


Fig. 11. Same caption as Fig. 10, but for GRS 1915+105 in the 27–36 keV band. The intensity linear correlation coefficients are 0.98 between IBIS (26–40 keV) and SPI, and 0.97 between *Swift*/BAT (24–35 keV) and SPI.

intensity and hence to the high S/N of the source. However, the systematic errors due to the finite precision of the transfer function start to be important for this strong source, which may be in part responsible for the high chi-square value.

For the dataset related to GRS 1915+105, a moderately strong and variable source, the χ_r^2 is 1.24 for 51 573 d.o.f. and a total of 440 “time bins”. GRS 1915+105 intensity variation is displayed in Fig. 11 and consists of 106 segments.

The flux linear correlation factors between SPI and IBIS (“quick-look analysis”) fluxes measured in the same “time bins” are 0.991, 0.948, and 0.983 for V0332+53, Cyg X-1, and GRS 1915+105. The linear correlations with *Swift*/BAT are 0.993, 0.934, and 0.973. The flux correlation for Cyg X-1 and GRS 1915+105 are shown in Fig. 12. In addition, the average flux in a given segment is not necessarily based on the same duration of the observation. The number of usable SPI and IBIS exposures contained in the same segment is not always the same and there are not always simultaneous observations available of the same sky region by *Swift*/BAT and INTEGRAL. Despite these limitations, the fluxes measured by these instruments are quantitatively well correlated.

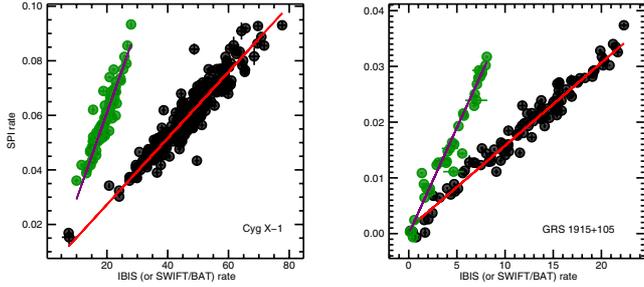


Fig. 12. Black denotes the SPI-IBIS correlation and green the SPI-*Swift*/BAT flux correlation in the “time bins” defined using the “data-space” method. The best SPI-IBIS and SPI-*Swift*/BAT linear regression are shown in red and purple. *Left* shows Cyg X-1 and *right* GRS 1915+105.

Table 6. “Limited-memory” “data-space” algorithm results.

Dataset	Cost function	Penalty (β/β_0)	χ_r^2 (d.o.f.)	Total number of “time bins”
V0332+53	C.9	0	1.062(6595)	97
	C.8	1	1.059(6592)	100
	C.8	0.32	0.995(6492)	200
GX 339-4	C.8	0.05	0.997(16 533)	2147
	C.9	0	1.202(19 930)	940
	C.9	0.02	1.197(19 906)	964
	C.8	1	1.278(20 108)	762
GRS 1915+105	C.8	0.2	1.030(18 951)	1919
	C.9	0	1.241(51 553)	424
	C.9	0.02	1.237(51 565)	412
	C.8	1	1.246(51 585)	392
Cyg X-1	C.8	0.2	1.215(51 023)	954
	C.9	0	1.436(40 068)	676
	C.8	1	1.475(40 120)	624
	C.8	0.02	1.292(38 948)	1796

Notes. The “limited” version is parameterized as indicated in Sect. 6.3.4. The penalty parameter is measured in units of $\beta_0 = \log(M)$, where M is the number of data points of the dataset. The bold penalty parameters used throughout the analysis are indicated in bold. **C.8** indicates that the cost function is the chi-square and **C.9** the cost function obtained with a “flat-prior” hypothesis.

Lower values of the chi-square than those indicated in Table 7 can be obtained by reducing the width of the energy band (Sect. 6.1) or by using a lower penalty parameter (Sect. 6.3.2) if the cost function is given by expression C.8.

The IGR J17464-3213 dataset corresponds to the central, crowded, region of the Galaxy. The sky model consists of 132 sources and the background timescale was fixed to about six hours. The dataset is relatively large (7147 exposures) and was artificially split into three subsets to reduce the computation time. The characteristics of these three subsets are displayed in Table 6. The intensity variations of the central source, IGR 17464-3213, was modeled with 29 segments. We also extracted the intensity evolutions of GRS 1758-258, GX 1+4, GS 1826-24, and GX 354-0 which are derived simultaneously (Fig. 13). There are a few spikes in the GX 1+4 intensity variations, which are not investigated here in more detail. The comparison of the source intensity variations obtained with the IBIS instrument (the light curves are assembled from a “quick-look analysis”, the intensities are computed/averaged in the temporal segments found by the “data-space” method) gives linear correlation coefficients of 0.995, 0.800, 0.786, 0.845, and 0.901 for IGR J17464-3213, GRS 1758-258, GX 1+4, GS 1826-24, and

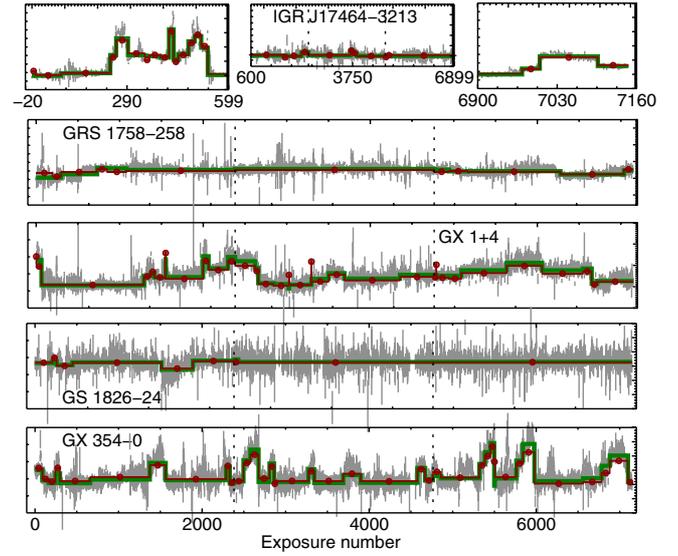


Fig. 13. Intensity evolutions (red) of IGR J17464-3213, GRS 1758-258, GX 1+4, GS 1826-24, and GX 354-0 in the 27–36 keV band. The IGR J17464-3213 dataset is divided in-to three subsets (dashed vertical lines show the limits) during the computation. The IBIS time series (30–40 keV) is shown in gray and the averaged fluxes in SPI “time bins” are plotted in green, as in Fig. 10. The distance of GRS 1758-258, GX 1+4, GS 1826-24, and GX 354-0 to IGR J17464-3213 are 7.3, 8.1, 12.7, and 3.4°.

GX 354-0. The comparison with *Swift*/BAT gives 0.996, 0.912, 0.909, 1.00, and 0.968 for the same sources.

6.4. Comparison of the “image-space” and “data-space” methods

The comparison was made on the SPI and IBIS common 1183 exposures related to the GX 339-4 dataset. The “image-space” algorithm used the IBIS (26–40 keV) light curve as external input to define SPI “time bins”. It achieved a χ_r^2 of 1.19 (18 880 d.o.f.) and a total of 1990 “time bins” and used 17 segments to describe the intensity evolution of GX 339-4 (Sect. 6.2). The “data-space” algorithm displays a similar χ_r^2 of 1.20 and used 15 segments. Both light curves are compared in Fig. 14. If the background intensity is forced to vary on a timescale of about six hours, the number of segments is 13, if the source 4U 1700-377 is forced to vary exposure-by-exposure, the number of segments is 13 as well. The SPI curves are more difficult to compare quantitatively because the total number of segments and the χ_r^2 are not the same with the two methods (Sect. 6.5). Nevertheless, we compared them indirectly with the IBIS binned ones. This demonstrates the effectiveness of the “data-space” algorithm.

6.5. Studying a particular source: Obtaining its light curve on the exposure timescale

The data related to a given source also contain the contribution of the many other sources in the FoV. To study this particular source, one must take into account the influence of the other sources and distinguish the different contributions to the data. This means knowing the model of the intensity variations of all other sources. At the same time, the number of unknowns to be determined in the related system of equations must be minimum to achieve a high S/N for the source of interest. The intensity variation of all sources of the FoV is not generally known a priori

Table 7. “Data-space” algorithm comparison.

Dataset	Version	χ_r^2 (d.o.f.)	C.8 (Chi-square)			χ_r^2 (d.o.f.)	C.9 (“Flat” prior)		
			Number of “time bins”				Number of “time bins”		
			Central source ^a	Background ^b	Total ^c		Central source ^a	Background ^b	Total ^c
V0332+53	Full	1.062(6603)	12	53	89	1.075(6612)	9	51	80
	Limited	1.059(6592)	14	50	100	1.062(6595)	9	50	97
	Full*	1.094(6545)	14	96	147	1.100(6550)	12	96	142
	Limited*	1.089(6537)	15	96	155	1.094(6539)	12	96	153
GX 339-4						1.218(20 097)	13	71	773
	Limited	1.278(20 108)	14	57	762	1.202(19 930)	15	87	940
GRS 1915+105	Limited*	1.259(19 932)	12	267	938	1.189(19 730)	13	267	1140
	Limited	1.246(51 585)	100	65	392	1.241(51 553)	102	94	424
Cyg X-1	Limited*	1.225(51 162)	79	515	791	1.215(51 133)	92	515	840
	Limited	1.475(40 120)	267	82	624	1.436(40 068)	280	121	676
IGR J17464-3213	Limited*	1.458(39 798)	252	448	946	1.423(39 729)	290	448	1015
	Limited*					1.263(132 486)	29	1132	3307

Notes. The algorithm uses the penalty parameter $\beta = 0$ for cost function C.9 and $\beta = \beta_0$ for cost function C.8 as indicated in bold in Table 6. ^(*) The background variability timescale is fixed to about six hours. ^(a) The central source from which the processed dataset is named. ^(b) Number of segments to describe the background intensity variations and ^(c) the total number of segments of the dataset.

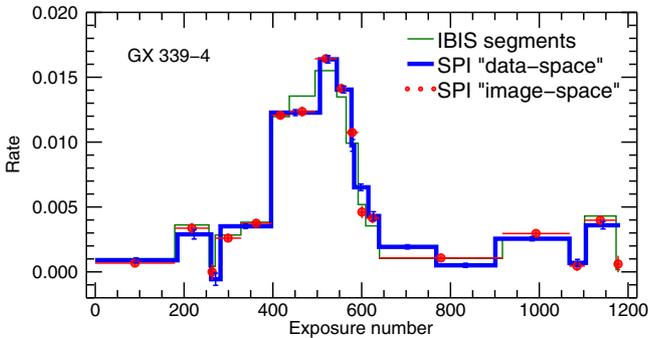


Fig. 14. Comparison of GX 339-4 (27–36 keV) intensity variations obtained with the “image-space” and the “data-space” algorithms. The common SPI/IBIS database contains 1183 exposures. The “image space” method describes GX 339-4 intensity variations with 17 segments (red) for a χ_r^2 of 1.19. The “data-space” method uses 15 segments (blue) and achieves χ_r^2 of 1.20. The GX 339-4 segmented version of the IBIS (26–40 keV) time series is shown in green. The absolute rate normalization between SPI and IBIS is arbitrary.

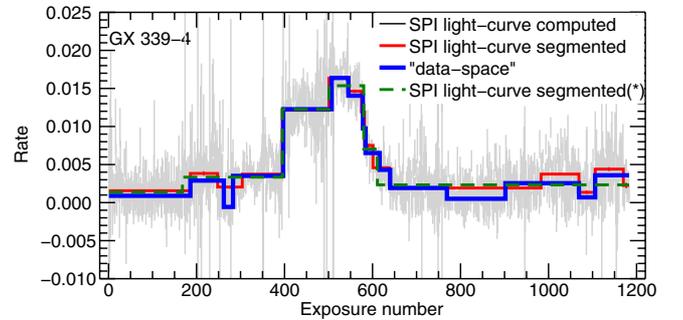


Fig. 15. “Data-space” algorithm has modeled the temporal evolution of GX 339-4 in the 27–36 keV band with 15 constant segments (blue). The light curve (in gray) of GX 339-4 was obtained by directly processing SPI data (after fixing the temporal evolution of the other sources). It contains 1183 data points (one measurement per exposure). The red curve is the segmentation of the gray curve into 15 segments with the “image space” algorithm. (*) If the background is forced to vary exposure-by-exposure, the GX 339-4 light curve is segmented only into six segments (dashed-green).

and constitutes a difficulty. The “image-space” or “data-space” methods are designed to solve this difficulty.

The first step consists of processing data using methods that allow one to construct synthetic intensity variations of the sources; then one determines these variations by constructing “time bins” while minimizing their number. The intensity variations of all sources are fixed, excepted those of the source of interest, which can be therefore studied in more detail.

As an example, the procedure was applied to the GX 339-4 light curve study exposure-by-exposure. Figure 15 shows the light curve obtained with SPI, compared with its segmented version. The light curve is segmented into 15 segments, this number is comparable to the number found using IBIS time series of 17 segments, which also confirms the idea that it is useful to adjust the SPI and IBIS sensitivity.

Next, both GX 339-4 source and background were forced to vary exposure-by-exposure. The S/N of this GX 339-4 light curve is degraded compared with the case where the background varies on a timescale of about six hours or is segmented. The resulting light curve is segmented in-to only six segments. This illustrates the importance of having a minimum number of parameters to model the sky and the background.

7. Discussion and summary

7.1. Discussion

We proposed two methods for modeling the intensity variations of sky sources and instrumental background. The first one, called “image space”, is relatively easy to implement and benefits from the IBIS instrument on-board INTEGRAL, which simultaneously observes the same region of the sky as SPI. Hence in this case, the main weakness of this method is that it requires information from other instruments and then depends both on the characteristics of these instruments (FoV, sensitivity, etc.) and on the level of processing performed on these available external data.

The second method, called “data space” (Sect. 5), is based solely on SPI data and does not depend on external data. With this algorithm, the background intensity variations can be computed (6.3.5). The dependence across segments through the transfer function greatly increases the complexity of the algorithm, which is very computer-intensive and hence time-consuming. We have made some simplifications to be able to handle the problem and optimizations that in most cases accelerate the computations, in particular the way the matrices

are updated during the change-point position search. More or alternative optimizations are possible, although we did not explore them in this paper; for example a (sparse) Cholesky update/down-date may be used at each iteration to avoid the decomposition of matrix \mathbf{H} at each iteration. Similarly, an alternative to the procedure used here could be solving the orthogonal-least-squares problem through QR rank update. Because of the approximations we made, the best segmentation (in the mathematical sense) is probably not achieved, but at least we obtained a reliable one, which is sufficient to improve the sky model.

The final value of the chi-square is not really directly controlled for the “image-space” method. For the “data space”, the value achieved depends mainly on the value of the penalty parameter. We empirically derived some initial guess values. The determination of the penalty parameter, even if a good guess of its initial value is possible, and the cost function formulation are difficulties of the “data-space” method, which need more investigation.

In addition, the “data-space” method is more suitable for exploring the interdependence of the source contributions to the data. It takes into account the co-variance of the parameters during the data reduction process. In contrast, the “image-space” method uses only the variance since the light curves are already the product of another analysis and the covariance information related to the sources in the FoV is lost.

We chose to use a very simple representation of the source intensity variations by using piecewise constant functions. With this modeling of the time series, the individual segments are disjoint. For the “data-space” method and for a given (single) source, the columns of the corresponding submatrix are orthogonal. With a more complex light curve model, this property no longer holds. However, it is possible to derive a formulation where the intensity variations are continuous, e.g., using piecewise polynomials (the quadratic functions have been studied by [Fearnhead & Liu \(2011\)](#)).

7.2. Summary

With only 19 pixels, the SPI detector does not provide enough data to correctly construct and sample the sky image viewed through the aperture. The dithering technique solves this critical imaging problem by accumulating “non-redundant” data on a given sky region, but at the same time it raises important questions of data reduction and image/data combination because of the variability of the sources.

We proposed two algorithms that model the intensity variation of sources in the form of combinations of piecewise segments of time during which a given source exhibits a constant intensity. For our purposes, these algorithms can help to solve a specific difficulty of the SPI data processing, which is to take into account the variability of sources during observations and consequently optimize the signal-to-noise ratio of the sources. A first algorithm uses existing time series to build segments

of time during which a given source exhibits a constant intensity. This auxiliary information is incorporated into the SPI system of equations to be solved. A second algorithm determines these segments directly using SPI data and constructs some “synthetic” light curves of the sources contained in the FoV. It separates the contribution to the data of each of the sources through the transfer function. Both algorithms allow one to introduce more objective parameters, here the “time bins”, in the problem to be solved. They deliver an improved sky model that fits the data better.

The algorithms also have various applications, for example for studying a particular source in a very crowded region. One typical example is the extraction of the diffuse emission components in addition to all source components. Indeed, the contribution of variable sources should be “eliminated” across the whole sky, so that the diffuse low surface brightness emission is as free as possible from residual emission coming from the point sources.

In addition, the “image-space” method allows one to merge databases of the various instruments on-board the INTEGRAL observatory and uses the complementary information of different telescopes.

Acknowledgements. The INTEGRAL/SPI project has been completed under the responsibility and leadership of CNES. We are thankful to ASI, CEA, CNES, DLR, ESA, INTA, NASA, and OSTC for their support. We thank the referees for their very useful comments, which helped us to strengthen and improve this paper.

References

- Akaike, H. 1974, *IEEE Trans. Autom. Contr.*, 19, 716
 Barthelmy, S. D., Barbier, L. M., Cummings, J. R., et al. 2005, *Space Sci. Rev.*, 120, 143
 Bouchet, L., Roques, J. P., & Jourdain, E. 2010, *ApJ*, 720, 177
 Bouchet, L., Strong, A., Porter, T. A., et al. 2011, *ApJ*, 739, 29
 Dubath, P., Knödseder, J., Skinner, G. K., et al. 2005, *MNRAS*, 357, 420
 Fearnhead, P. 2006, *Stat. Comp.*, 16, 203
 Fearnhead, P., & Liu, Z. 2011, *Stat. Comp.*, 21, 217
 Hannan, E. J., & B. G. Quinn 1979, *J. Roy. Stat. Soc.*, B, 41, 190
 Hutter, M. 2007, *Bayesian Anal.*, 4, 635
 Jackson, B., Scargle, J. D., Barnes, et al. 2005, *IEEE, Sig. Proc. Lett.*, 12, 105
 Jensen, P. L., Clausen, K., Cassi, C., et al. 2003, *A&A*, 411, L7
 Killick, R., Fearnhead, P., & Eckley, I. A. 2012, *J. Am. Stat. Assoc.*, 107, 1590
 Koop, G. 2003, *Bayesian Econometrics (The Atrium, Southern Gate, Chichester, UK: John Wiley & Sons Ltd)*
 Poirier, D. 1995, *Intermediate Statistics and Econometrics: A comparative approach (Cambridge: The MIT Press)*
 Punsakaya, E., Andrieu, C., Doucet, A., & Fitzgerald, W. J. 2002, *IEEE Trans. Sig. Proc.*, 50, 3
 Roques, J. P., Schanne, S., Von Kienlin, A., et al. 2003, *A&A*, 411, L91
 Scargle, J. D. 1998, *ApJ*, 504, 405
 Schwarz, G. E. 1978, *Ann. Stat.*, 6, 461
 Ubertini, P., Lebrun, F., Di Cocco, G., et al. 2003, *A&A*, 411, L131
 Vedrenne, G., Roques, J. P., Schonfelder, V., et al. 2003, *A&A*, 411, L63
 Yao, Y. 1984, *Ann. Stat.*, 12, 1434
 Zellner, A. 1971, *An Introduction to Bayesian Inference in Econometrics (New York: John Wiley & Sons)*

Appendix A: Stating the problem: from the physical phenomenon to the mathematical equation

The signal (counts and energies) recorded by the SPI camera on the 19 Ge detectors is composed of contributions from each source (point-like or extended) in the FoV convolved with the instrument response plus the background. For n_s sources located in the FoV, the data D_{dp}^{raw} obtained during an exposure p in detector d for a given energy band can be expressed by the relation

$$D_{dp}^{\text{raw}} = \sum_{j=1}^{n_s} R_{dp,j} I_{p,j}^s + B_{dp}^{\text{bg}} + \epsilon_{dp}, \quad (\text{A.1})$$

where $R_{dp,j}$ is the response of the instrument for source j (function of the incident angle of the source relative to pointing axis), $I_{p,j}^s$ is the flux of source j during exposure p and B_{dp}^{bg} the background both recorded during the exposure p for detector d . ϵ_{dp} are the measurement errors on the data D_{dp}^{raw} , they are assumed to have zero mean, to be independent, and to be normally distributed with a known variance σ_{dp} ($\epsilon_{dp} \sim N(0, [\sigma_{dp}^2])$) and $\epsilon_{dp} = \sqrt{D_{dp}^{\text{raw}}}$.

For a given exposure p , D_{dp}^{raw} , ϵ_{dp} , $R_{dp,j}$, and B_{dp}^{bg} are vectors of $n_d(p)$ elements (say, $n_d = 19$) elements). However, $n_d(p)$ corresponds to the number of functioning detectors, the possible values are 16, 17, 18 or 19 for single events and up to 141 when all the multiple events are used in addition to the single events (Roques et al. 2003). For a given set of n_p exposures, we have to solve a system of M ($= \sum_{p=1}^{n_p}$) equations (Eq. (A.1)).

To reduce the number of free parameters related to the background, we take advantage of the stability of relative count rates between detectors and rewrite the background term as

$$B_{dp}^{\text{bg}} = I_p^{\text{bg}} \times U^d \times t_{dp}. \quad (\text{A.2})$$

In this expression, either U^d or I_p^{bg} is supposed to be known, depending on the hypotheses made on the background structure. To illustrate the idea, suppose that U^d is known. U is an “empty field” (Bouchet et al. 2010), a vector of n_d elements. Then, the number of parameters necessary to model the background reduces to n_p . The number of unknowns (free parameters) of the set of M equations is then $(n_s + 1) \times n_p$ (for the n_s sources and the background intensities, namely I^s and I^{bg}). Fortunately, a further reduction of the number of parameters can be obtained since many sources vary on timescales longer than the exposure timescale. In addition, many point sources are weak enough to be considered as having constant flux within the statistical errors, especially for the higher energies. Then the $n_p \times n_s$ parameters related to sources will reduce to N_s^{eff} parameters and, similarly, N_b for the background. As these parameters also have a temporal connotation, they are also termed, hereafter, “time bins”. Finally, we have $N = N_s^{\text{eff}} + N_b$ free parameters to be determined. If the source and background relative intensities are supposed to be constant throughout all observations (time spanned by the exposures), omitting the detector index, the equation can be simplified as

$$D_p^{\text{raw}} = \sum_{j=1}^{n_s} R_{p,j} I_j^s + P_p I^{\text{bg}} + \epsilon_p \quad (\text{A.3})$$

with $P_p = t_{d,p} \times U^d$.

Here P_p is a vector of $n_d(p)$ elements. The aim is to compute the intensities $I^s(j)$ of the n_s sources and the background relative intensity I^{bg} . Therefore, the problem can be written in matrix form as

$$\begin{pmatrix} P(1) & R(1,1) & \dots & R(1,n_s) \\ P(2) & \ddots & \ddots & R(2,n_s) \\ \vdots & \ddots & \ddots & \vdots \\ P(n_p) & \dots & \dots & R(n_p,n_s) \end{pmatrix} \begin{pmatrix} I^{\text{bg}} \\ I^s(1) \\ I^s(2) \\ \vdots \\ I^s(n_s) \end{pmatrix} \epsilon = \begin{pmatrix} D(1) \\ D(2) \\ \vdots \\ D(n_p-1) \\ D(n_p) \end{pmatrix}.$$

At last, Eq. (A.1) reduces to the general linear system $\mathbf{H}^0 \mathbf{x}^0 + \epsilon = \mathbf{y}$, where \mathbf{H}^0 is the transfer matrix, \mathbf{y} the data, ϵ the measurement errors and \mathbf{x}^0 the unknowns \mathbf{H}^0 (elements \mathbf{H}_{ij}) is an $M \times N_0$ matrix and $N_0 = n_s + 1$. $\mathbf{x}^0 = (I^{\text{bg}}, I^s(1), \dots, I^s(n_s))^T$ is a vector of length N_0 . $\mathbf{y} = (D_1, D_2, \dots, D_{n_p})^T$ and $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_{n_p})^T$ are vectors of length M . Now, if all N_0 sources (here sources and background) are assumed to vary and the signal of source J is represented by $n_{\text{seg}}(J)$ piecewise constant segments of time or “time bins”, then each column of \mathbf{H}^0 will be expanded in $n_{\text{seg}}(J)$ columns. Equation (A.1) reduces to $\mathbf{H} \mathbf{x} + \epsilon = \mathbf{y}$, where the matrix \mathbf{H} is of dimension $M \times N$, with $N = \sum_{j=1}^{N_0} n_{\text{seg}}(j)$ and \mathbf{x} a vector of length N representing the intensity in the “time bins”. Schematically and to illustrate the purpose of Sect. 5.1, \mathbf{H} can be derived from \mathbf{H}^0 :

$$\mathbf{H}^0 = \begin{pmatrix} H_{11} & H_{12} & H_{13} & \dots & H_{1N} \\ H_{21} & H_{22} & H_{23} & \ddots & H_{2N} \\ H_{31} & H_{32} & H_{33} & \ddots & H_{3N} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ H_{M1} & H_{M1} & H_{M3} & \dots & H_{MN} \end{pmatrix}$$

$$\mapsto \mathbf{H} = \begin{pmatrix} H_{11} & 0 & 0 & 0 & H_{12} & 0 & 0 & H_{13} & 0 & H_{1N} & \dots & 0 \\ 0 & H_{21} & 0 & 0 & H_{22} & 0 & 0 & 0 & H_{23} & H_{2N} & \dots & 0 \\ 0 & 0 & H_{31} & 0 & 0 & H_{32} & 0 & 0 & H_{33} & 0 & \dots & H_{3N} \\ \vdots & \ddots & \vdots \\ 0 & 0 & 0 & H_{M1} & 0 & 0 & H_{M2} & 0 & H_{M3} & 0 & \dots & H_{MN} \end{pmatrix}.$$

For the training data given in Sect. 2, which contain $\approx 39\,000$ exposures and for the 25–50 keV energy band, we used $N = N_s^{\text{eff}} + N_b \approx 22\,500$ “time bins” (Bouchet et al. 2011). To fit the sky model consisting of sources plus instrumental background to the data, we minimized the corresponding chi-square value:

$$\chi^2 = \sum_{i=1}^M \left[\frac{y_i - \sum_{j=1}^N H_{ij} x_j}{\sigma_i} \right]^2, \quad (\text{A.4})$$

where σ_i is the measurement error (standard deviation) corresponding to the data point y_i , formally $\sigma_i = \sqrt{y_i}$. By replacing \mathbf{H} and \mathbf{y} by their weighted version,

$$y_i \equiv \frac{y_i}{\sigma_i} \text{ and } H_{ij} \equiv \frac{H_{ij}}{\sigma_i}. \quad (\text{A.5})$$

The least-squares solution \mathbf{x} is obtained by solving the following normal equation:

$$(\mathbf{H}^T \mathbf{H}) \mathbf{x} = \mathbf{H}^T \mathbf{y} \text{ or } \mathbf{A} \cdot \mathbf{x} = \mathbf{b}. \quad (\text{A.6})$$

Here \mathbf{A} is a symmetric matrix of order N , \mathbf{x} and \mathbf{b} are vectors of length N .

Appendix B: Segmentation/partition of a time series

B.1. Bayesian model comparison

The literature about model comparisons is vast, especially in the field of econometric (e.g. Zellner 1971; Poirier 1995; Koop 2003). We have some data y , a model $Model_i$ containing a set of parameters θ_i . In our application, we have to choose between models of light curves ($Model_1$ or $Model_2$), based on observations over a time interval T :

$$\begin{cases} Model_1 : & \text{Constant intensity over } T \\ Model_2 : & \text{Possibly different constant intensities} \\ & \text{in two subintervals, } T_1 + T_2 = T. \end{cases}$$

Using Bayesian inference (see for example Scargle 1998), the posterior odds ratio makes it possible to compare models i and j : it is simply the ratio of their posterior model probabilities:

$$PO_{ij} = \frac{p(Model_i|y)}{p(Model_j|y)} = \frac{p(y|Model_i)p(Model_i)}{p(y|Model_j)p(Model_j)} = BF_{ij} \frac{p(Model_i)}{p(Model_j)}, \quad (B.1)$$

where BF_{ij} is the Bayes factor comparing $Model_i$ to $Model_j$. The quantity $p(y|Model_i)$, the marginal likelihood

$$p(y|Model_i) \equiv L(Model_i, y) = \int P(y|\theta_i, Model_i)P(\theta_i|Model_i)d\theta_i. \quad (B.2)$$

To compare $Model_2$ with $Model_1$, non-informative choices are commonly made for the prior model probabilities (i.e. $P(Model_1) = P(Model_2) = \frac{1}{2}$). Then, to prevent overfitting, we introduce a penalty parameter γ that chooses model $Model_2$ instead of $Model_1$ if

$$BF_{21} = \frac{p(y|Model_2)}{p(y|Model_1)} > \gamma. \quad (B.3)$$

To fit the framework of expression Eq. (3), this equation can be equivalently rewritten as

$$-\log[p(y|Model_2)] + \beta < -\log[p(y|Model_1)] \quad \text{with } \beta = \log(\gamma). \quad (B.4)$$

Appendix C: Cost or likelihood function computation

C.1. Normal distributions

The model for the time series observations (Eq. (1)) is

$$x_i \equiv f(t_i) + \epsilon_i \quad i = 1, 2, \dots, L,$$

where x_i is the value measured at time t_i , f is the unknown signal and ϵ_i the measurement errors. These errors are assumed to have zero mean, to be independent, and to be normally distributed with a known variance σ_i^2 , thus the probability for bin i is

$$P(\epsilon_i|\sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\epsilon_i}{\sigma_i}\right)^2}.$$

The parameters associated with each segment are independent of each other, the likelihood can be calculated for the data within

each segment or block. For the block K where the true signal is λ , the likelihood is,

$$\mathcal{L}^{(K)} = \prod_{i \in K} \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - \lambda}{\sigma_i}\right)^2}. \quad (C.1)$$

The product is over all i such that t_i falls within block K , for instance L_K points fall into block K . We can simply maximize the likelihood for instance, $P_K = \max(\mathcal{L}^{(K)})$; its maximum is reached for $\lambda = \lambda_{\max}$. With the following notations

$$a_K = \frac{1}{2} \sum_{i \in K} \frac{1}{\sigma_i^2} \quad b_K = - \sum_{i \in K} \frac{x_i}{\sigma_i^2} \quad c_K = \frac{1}{2} \sum_{i \in K} \frac{x_i^2}{\sigma_i^2}$$

and $C_K = \frac{(2\pi)^{-L_K/2}}{\prod_{i \in K} \sigma_i}$.

The expression can be written in terms of ordinary least-squares quantities:

$$\lambda_{\max} = \frac{-b_K}{2a_K} \quad \text{least-squares solution}$$

$$SSE_K = \sum_{i \in K} \left(\frac{x_i - \lambda_{\max}}{\sigma_i} \right)^2 = -2c_K + \frac{b_K^2}{2a_K} \quad \text{sum of square errors.}$$

The block likelihood can be rewritten as

$$\log P_K = -\frac{SSE_K}{2} + \log C_K. \quad (C.2)$$

Finally, the probability for the entire dataset is

$$\log P = \log \left[\prod_K P_K \right] = \sum_K \log P_K. \quad (C.3)$$

The choice of the prior of λ is another question, although many prior probabilities may be possible, we marginalize it by choosing the flat unnormalized prior ($p(x|Model) = \text{constant}$). If the variable λ is marginalized,

$$P_K \equiv p(y|Model_K) = C_K \int_{-\infty}^{+\infty} e^{-\frac{1}{2} \sum_{i \in K} \left(\frac{x_i - \lambda}{\sigma_i}\right)^2} d\lambda. \quad (C.4)$$

The expression can be integrated to give

$$\log P = \sum_K \left[-\frac{1}{2} \log a_K - \frac{SSE_K}{2} + \frac{1}{2} \log \pi + \log C_K \right]. \quad (C.5)$$

C.2. Generalization to multivariate normal distributions

We assume to have a linear regression model that assesses the relation between the dependent variables y_i (the data) and the N -vector of regressors x_i (unknowns). This relation is written in vector form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon}. \quad (C.6)$$

Thus, \mathbf{y} is an M -vector, \mathbf{H} an $M \times N$ matrix, $\boldsymbol{\varepsilon}$ an M -vector of errors, and \mathbf{x} an N -vector. The errors are assumed to be independently normally distributed with mean 0 and a variance σ , that is

$$\boldsymbol{\varepsilon} \sim N(\mathbf{0}_M, [\sigma^2]),$$

where $\mathbf{0}_M$ is an M -vector of zeroes and $[\sigma^2]$ is an $M \times M$ diagonal matrix whose diagonal is $\{\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2\}$. Using the properties

of the multivariate normal distribution, it follows that the likelihood function is given by

$$p(\mathbf{y}|\mathbf{x}, \sigma) = \frac{1}{2\pi^{M/2} \prod_{i=1}^M \sigma_i} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})'[\sigma^{-2}](\mathbf{y}-\mathbf{H}\mathbf{x})}. \quad (\text{C.7})$$

The likelihood function can be rewritten in terms of ordinary least-squares quantities:

$$\begin{aligned} \hat{\mathbf{x}} &= (\mathbf{H}'[\sigma^{-2}]\mathbf{H})^{-1}\mathbf{H}'[\sigma^{-2}]\mathbf{y} && \text{least-squares solution} \\ SSE &= (\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})'[\sigma^{-2}](\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) && \text{sum of squared errors.} \end{aligned}$$

If we take the highest likelihood value,

$$\log P = \log C - \frac{SSE}{2} \quad \text{with } C = \frac{1}{2\pi^{M/2} \prod_{i=1}^M \sigma_i}. \quad (\text{C.8})$$

If the variable x is marginalized by choosing the flat unnormalized prior, it yields for the marginal posterior after integration

$$\log P = \log C + \frac{1}{2} \log \frac{(2\pi)^p}{\det(\mathbf{H}'[\sigma^{-2}]\mathbf{H})} - \frac{SSE}{2}. \quad (\text{C.9})$$

For $M = L$, $N = 1$ and \mathbf{H} a matrix with elements $H_{ij} = 1$, we recover Eqs. (C.2) and (C.4), with

$$a = \det(\mathbf{H}'[\sigma^{-2}]\mathbf{H}) = \sum_{i=1}^N \frac{1}{\sigma_i^2}.$$

Appendix D: Common model selection criteria

Other penalty cost functions are possible, the most widely known criteria for model selection among a class of parametric models with different numbers of parameters are *AIC* (Akaike 1974) and *BIC* (Schwartz 1978). Given \mathcal{L} , the likelihood of the data, their expressions are $AIC = 2k - 2 \log(\mathcal{L})$, $BIC = -2 \log \mathcal{L} + k \log(M)$. Here M is the number of data and k the number of parameters used in the model. Both *AIC* and *BIC* fit the general form given in Eq. (3) with ($\beta = 2$) and ($\beta = \log M$). The penalty term of *BIC* is more stringent than the penalty term of *AIC* (for $M > 8$, $k \times \log M$ exceeds $2k$). The *AIC* formula is based on an asymptotic behavior with M large and a small number of parameters k and *BIC* on an asymptotic approximation to a transformation of the Bayesian posterior probability of a candidate model. Thus, both are issued from asymptotic approximation and may have some drawbacks in some range of parameters. Modified versions of these criteria have also been developed to correct their main weakness. The Hannan-Quinn information criterion (HQC; Hannan & Quinn 1979) proposes something intermediate with ($HQC = -2 \log \mathcal{L} + 2k \log(\log M)$) and ($\beta = 2 \log(\log M)$).

Appendix E: Mathematical utilities

The formulas below avoid recomputing the inverse of matrix from scratch when a small rank adjustment is performed on the matrix.

E.1. Sherman-Morrison-Woodbury formula – inversion lemma

Suppose that one aims to invert the matrix $\mathbf{A} + \mathbf{UCV}$. Here \mathbf{A} , \mathbf{U} , \mathbf{C} , and \mathbf{V} are $n \times n$, $n \times k$, $k \times k$ and $k \times n$ matrices, respectively. \mathbf{UCV} is called rank- k update (or correction) of \mathbf{A} . If \mathbf{C} has a much smaller dimension than \mathbf{A} , it is more efficient to use the Sherman-Morrison-Woodbury formula than to invert $\mathbf{A} + \mathbf{UCV}$ directly. Now suppose that \mathbf{A}^{-1} has already been computed and that \mathbf{C} and $\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U}$ are invertible. The Sherman-Morrison-Woodbury formula says that the inverse of a rank- k correction of some matrix can be computed by applying a rank- k correction to the inverse of the original matrix. Alternative names for this formula are the matrix inversion lemma or simply the Woodbury formula. The Woodbury matrix identity is

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}.$$

The determinant of $\mathbf{A} + \mathbf{UCV}$ is obtained through the relation:

$$\det(\mathbf{A} + \mathbf{UCV}) = \det(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U}) \det(\mathbf{C}) \det(\mathbf{A}).$$

E.2. Inverse of a partitioned matrix

Let the $N \times N$ matrix \mathbf{A} be partitioned into a block form as below, where \mathbf{T} and \mathbf{W} are square matrices of size $t \times t$ and $w \times w$, respectively ($t + w = N$). Matrices \mathbf{U} and \mathbf{V} are not necessarily square, and have size $t \times w$ and $w \times t$, respectively. Let matrices \mathbf{T} , \mathbf{W} and $\mathbf{Q} = \mathbf{W} - \mathbf{VT}^{-1}\mathbf{U}$ being invertible. The inverse of \mathbf{A} can be written as

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{T} & \mathbf{U} \\ \mathbf{V} & \mathbf{W} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{T}^{-1} + \mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1} & -\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1} & \mathbf{Q}^{-1} \end{bmatrix}.$$

The determinant of \mathbf{A} is obtained through the relation:

$$\det(\mathbf{A}) = \det(\mathbf{T}) \det(\mathbf{W} - \mathbf{VT}^{-1}\mathbf{U}) = \det(\mathbf{W}) \det(\mathbf{T} - \mathbf{U}\mathbf{W}^{-1}\mathbf{V}).$$

E.3. Application

We have to solve system $\mathbf{H}\mathbf{x} = \mathbf{y}$ in the least-squares sense; \mathbf{y} is the right-hand side, \mathbf{H} a matrix and \mathbf{x} the solution, see also Appendix A. Now, we need to update the solution \mathbf{x} of the system of equations $\mathbf{A}\mathbf{x} = (\mathbf{H}^T\mathbf{H})\mathbf{x} = \mathbf{H}^T\mathbf{y}$ after adding or deleting some columns to \mathbf{H} . Here \mathbf{H} is a matrix of size $M \times N$, \mathbf{x} and \mathbf{y} are vectors of length N and M , respectively. We first derive the expression of the updated inverse of \mathbf{A} . We assume that the inverse of \mathbf{A} is already computed and denote by \mathbf{A}_* the updated corresponding matrix. To simplify, we just plug the formula for 1 column added or suppressed. First we add N_{add} columns to \mathbf{H} such that $\mathbf{H}_* = [\mathbf{H}\mathbf{v}]$, \mathbf{v} is a matrix of size $M \times 1$ representing the newly added column, then

$$\mathbf{A}_* = \begin{bmatrix} \mathbf{H}^T \\ \mathbf{v}^T \end{bmatrix} \begin{bmatrix} \mathbf{H} & \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{H}^T\mathbf{H} & \mathbf{H}^T\mathbf{v} \\ \mathbf{v}^T\mathbf{H} & \mathbf{v}^T\mathbf{v} \end{bmatrix}.$$

The inverse of this partitioned matrix is (Appendix E.2)

$$\begin{aligned} \mathbf{A}_*^{-1} &= \begin{bmatrix} \mathbf{H}^T\mathbf{H} & \mathbf{H}^T\mathbf{v} \\ \mathbf{v}^T\mathbf{H} & \mathbf{v}^T\mathbf{v} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{H}^T\mathbf{v}\Delta^{-1}\mathbf{v}^T\mathbf{H}\mathbf{A}^{-1} & \mathbf{A}^{-1}\mathbf{H}^T\mathbf{v}\Delta^{-1} \\ -\Delta^{-1}\mathbf{v}^T\mathbf{H}\mathbf{A}^{-1} & \Delta^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F} & -\mathbf{u}_3 \\ -\mathbf{u}_3^t & \Delta \end{bmatrix}, \end{aligned}$$

Algorithm 1 Implicit rank-1 update of $\mathbf{A} = (\mathbf{H}^T\mathbf{H})^{-1}$ after adding a column \mathbf{v} to \mathbf{H} at position j .

- 1: $\mathbf{u}_1 = \mathbf{H}^T\mathbf{v}$
- 2: $\mathbf{u}_2 = \mathbf{A}\backslash\mathbf{u}_1$
- 3: $\Delta = [\mathbf{v}^T\mathbf{v} - \mathbf{u}_1^T\mathbf{u}_2]^{-1}$
- 4: $\mathbf{u}_3 = \mathbf{u}_2\Delta$
- 5: $\mathbf{A}_*^{-1} \leftarrow \begin{bmatrix} \mathbf{A}^{-1} + \Delta\mathbf{u}_2^T\mathbf{u}_2 & -\mathbf{u}_3 \\ -\mathbf{u}_3^T & \Delta \end{bmatrix}$ {Not explicitly formed}
- 6: $\mathbf{z}(1 : N + 1) = [\mathbf{H}\mathbf{v}]^T\mathbf{y}$ {A vector of length $n + 1$ }
- 7: $\mathbf{x}_* \leftarrow \begin{cases} \mathbf{x}(1 : N) = \mathbf{x} + \mathbf{u}_3(\mathbf{u}_2^T\mathbf{z}(1 : N) - \mathbf{z}(N + 1)) \\ \mathbf{x}(N + 1) = -\mathbf{u}_3^T\mathbf{z}(1 : N) + \Delta\mathbf{y}(N + 1) \end{cases}$ {Update \mathbf{x}
without using \mathbf{A}_*^{-1} explicitly}
- 8: $\mathbf{y}_*^{\text{predicted}} \leftarrow [\mathbf{H}\mathbf{v}]\mathbf{x}_*$ {Predicted data}
- 9: Permute column J and row J of \mathbf{A}_*^{-1} to last column and last row
- 10: Permute element J of \mathbf{x}_* to last element

Output: Updated least-squares solution and predicted data $\mathbf{x}_*, \mathbf{y}_*^{\text{predicted}}$

where $\Delta = \mathbf{v}^T\mathbf{v} - \mathbf{v}^T\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T\mathbf{v}$. Algorithm 1 performs the operation without accessing \mathbf{A}^{-1} explicitly. We use the MATLAB notation “ \backslash ” to denote the solution of a linear system (“ $\mathbf{x} = \mathbf{A}\backslash\mathbf{b}$ ” \equiv “ $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ ”); it emphasizes the fact that the system is solved without forming the inverse of the matrix of the system. If we need to remove a column from \mathbf{H} , the algorithm is described in algorithm 2.

Algorithm 2 Implicit rank-1 update of $\mathbf{A} = (\mathbf{H}^T\mathbf{H})^{-1}$ after suppressing column j of \mathbf{H} .

- 1: $\mathbf{e}_j = 0$ and $\mathbf{e}_j(j) = -1$ { \mathbf{e}_j is a unit vector}
- 2: $\mathbf{A}\mathbf{u}_3 = \mathbf{e}_j$ {One system to solve to have $\mathbf{u}_3 = \mathbf{A}^{-1}\mathbf{e}_j$ }
- 3: $\Delta = (\mathbf{e}_j^T\mathbf{u}_3)^{-1}$ {Obtain element \mathbf{A}_{jj}^{-1} }
- 4: $\mathbf{A}_*^{-1} \leftarrow \mathbf{A}^{-1}(1 : j - 1, j + 1 : N, 1 : j - 1, j + 1 : N) - \mathbf{u}_3\Delta\mathbf{u}_3^T$ {Not explicitly formed}
- 5: $\mathbf{z} = \mathbf{H}^T\mathbf{y}$
- 6: $\mathbf{w} = \mathbf{x} - \mathbf{u}_3\mathbf{e}_j^T\mathbf{z}$
- 7: $\mathbf{x}_* \leftarrow \mathbf{w} - \mathbf{u}_3\Delta\mathbf{z}$
- 8: $\mathbf{y}_*^{\text{predicted}} \leftarrow \mathbf{H}\mathbf{x}_*$

Output: Updated least-squares solution and predicted data $\mathbf{x}_*, \mathbf{y}_*^{\text{predicted}}$

E.4. Adding N_{add} columns and suppressing N_{sub} columns

The two algorithms described above are also applicable without loss of generality when several columns are added or removed. Algorithm 3 first adds N_{add} columns and then suppresses N_{sub} columns from the transfer matrix \mathbf{H} and updates the solution \mathbf{x}_* and predicted data $\mathbf{y}_*^{\text{predicted}}$.

Algorithm 3 $\mathbf{A} = (\mathbf{H}^T \mathbf{H})^{-1}$ update after adding N_{add} and suppressing N_{sub} columns from \mathbf{H} .

Input: m, n size of the system
 \mathbf{H} : Transfer matrix ($m \times n$) to be updated
 $\mathbf{y}(m)$: data
 $j_{\text{submin}}, j_{\text{submax}}$: contiguous column of \mathbf{H} to be suppressed ($N_{\text{sub}} = j_{\text{submax}} - j_{\text{submin}} + 1$)
 $j_{\text{addmin}}, j_{\text{addmax}}$: contiguous column of \mathbf{H} to be added ($N_{\text{add}} = j_{\text{addmax}} - j_{\text{addmin}} + 1$)
 $\mathbf{v}(m, N_{\text{add}})$: columns to be added to \mathbf{H}
 \mathbf{x} : solution to $\mathbf{A}\mathbf{x} = (\mathbf{H}^T \mathbf{H})\mathbf{x} = \mathbf{H}^T \mathbf{y}$
 $\mathbf{z}(1:n) = \mathbf{H}^T \mathbf{y}$ & $\mathbf{z}(n+1:n+N_{\text{add}}) = \mathbf{v}^T \mathbf{y}$

- 1: {Add N_{add} columns from j_{addmin} to j_{addmax} (append them first to position $n+1:n+N_{\text{add}}$)}
- 2: $\mathbf{u}_1 = \mathbf{x}^T \mathbf{v}$
- 3: $\mathbf{A}\mathbf{u}_2 = \mathbf{u}_1$ {Solve $\mathbf{u}_2 = \mathbf{A}^{-1} \mathbf{u}_1$ }
- 4: $\mathbf{D}_i = \mathbf{v}^T \mathbf{v} - \mathbf{u}_1^T \mathbf{u}_2$ {Symmetric matrix of order N_{add} }
- 5: $\mathbf{D}_a = \mathbf{D}_i^{-1}$
- 6: $\mathbf{u}_3 = \mathbf{u}_2 \mathbf{D}_a$
- 7: $x_a(1:n) = x + \mathbf{u}_3 (\mathbf{u}_2^T \mathbf{z}(1:n) - \mathbf{z}(n+1:n+N_{\text{add}}))$ {Non-permuted solution after addition of N_{add} columns to \mathbf{H} }
- 8: $x_a(n+1:n+N_{\text{add}}) = -\mathbf{u}_3^T \mathbf{z}(1:n, 1) + \mathbf{D}_a \mathbf{z}(n+1:n+N_{\text{add}}, 1)$ {(same)}
- 9: {Suppress N_{sub} columns from j_{submin} to j_{submax} }
- 10: $e_j(n+N_{\text{add}}, N_{\text{sub}}) = 0$
- 11: **for** $i = 1$ **to** N_{sub} **do**
- 12: $e_j(j_{\text{submin}} + i - 1, i) = 1$
- 13: **end for**
- 14: $\mathbf{A}\mathbf{w} = \mathbf{e}_j$ {Solve $\mathbf{w} = \mathbf{A}^{-1} \mathbf{e}_j$ }
- 15: $\mathbf{u}_{3s}(1:n, 1:N_{\text{sub}}) = \mathbf{w} + \mathbf{u}_3 (\mathbf{u}_2^T \mathbf{e}_j(1:n, 1:N_{\text{sub}}) - \mathbf{e}_j(n+1:n+N_{\text{add}}, 1:N_{\text{sub}}))$
- 16: $\mathbf{u}_{3s}(n+1:n+N_{\text{add}}, 1:N_{\text{sub}}) = -\mathbf{u}_3^T \mathbf{e}_j(1:n, 1:N_{\text{sub}}) + \mathbf{D}_a \mathbf{e}_j(n+1:n+N_{\text{add}}, 1:N_{\text{sub}})$
- 17: $\mathbf{D}_i = \mathbf{e}_j^T \mathbf{u}_{3s}$ { \mathbf{D}_{sub} is a symmetric matrix of order N_{sub} }
- 18: $\mathbf{D}_{\text{sub}} = \mathbf{D}_i^{-1}$
- 19: $\mathbf{z} = x_a - \mathbf{u}_{3s} \mathbf{e}_j^T \mathbf{z}$
- 20: $\mathbf{x}_* = \mathbf{z} - \mathbf{u}_{3s} \mathbf{D}_{\text{sub}} \mathbf{e}_j^T \mathbf{z}$
- 21: $\mathbf{y}_* = \mathbf{H}\mathbf{x}_*(1:n) + \mathbf{v}\mathbf{x}_*(n+1:n+N_{\text{add}})$
- 22: $\mathbf{x}_* = \mathbf{x}_*([1:j_{\text{submin}}-1:j_{\text{submax}}+1:n+N_{\text{add}}])$ {After suppressing elements $j_{\text{submin}}:j_{\text{submax}}$ }
- 23: $\mathbf{x}_* = \mathbf{x}_*([1:j_{\text{addmin}}-1:n-N_{\text{sub}}+1:n+N_{\text{add}}-N_{\text{sub}}:j_{\text{addmin}}:n-N_{\text{sub}}])$ {Permute the output solution \mathbf{x}_* }

Output: Updated least-squares solution and predicted data: $\mathbf{x}_*(n+N_{\text{add}}-N_{\text{sub}}), \mathbf{y}_*(m)$

Appendix F: “Pseudo-codes”

The codes (simplified versions) for the ‘image-space’ and ‘data-space’ algorithms are written in IDL and Fortran, the time series of GX 339-4 (6.2.2) and the data for V0332+53 (6.3.1) that are to be used as input data to these codes can be downloaded at <http://sigma-2/integral/algorithms>. These codes and input data allow one to reproduce the main features of Figs. 4 and 10.

F.1. “Image-space” algorithm

Algorithm 4 Data best partitioning

Input:

L: data length
 x(1:L): dataset of the form (x_1, x_2, \dots, x_L)
 β : penalty parameter
 1: last(1:L)=0 ; best(1:L)=0.0 {Initialization}
 2: **for** R=1 **to** L **do** { Loop on the data subset 1 to R}
 3: cost(1:n)=C(x,R), see App. C {Compute $\text{cost}(i)=C(x_{i:R})$ }
 4: F(1)=cost(1)+ β {Segment starting at $i = 1$ and ending at $R : F(0) + C(x_{1:R})$ }
 5: **for** i=2 **to** R **do**
 6: F(i)=best(i-1)+cost(i)+ β {i.e. $F(t) + C(x_{t+1:R})$ with $t = i - 1$ }
 7: **end for**
 8: [best(R),last(R)]=minloc(F(1:R)){minimum value and its location}
 9: **end for**
 10: {The vector *last* gives the start point of individual segment of the optimal partition in the following way. Let $\tau_m = \text{last}(L)$, $\tau_{m-1} = \text{last}(\tau_m - 1)$, etc. The last segments contains cells (data points) $\tau_m, \tau_m + 1, \dots, L$, the next to last contains cells $\tau_{m-1}, \tau_{m-1} + 1, \dots, \tau_m - 1$ and so on.}
 11: index=last(n) and cpt=[] {Change points storage}
 12: **while** index > 1 **do**
 13: cpt=[index cpt]
 14: index=last(index-1)
 15: **end while**

Output:

cpt = [1 = $\tau_0, \tau_1, \dots, \tau_m, \tau_{m+1} = L + 1$]: change points

F.2. “Data-space” algorithm

Algorithm 5 “Data-space” algorithm**Input:**

nofp : number of exposures
nofs : number of sources
ndata : number of data points
Hzero(*ndata* : *nofs*) : weighted initial transfer matrix ($H^0 \equiv \frac{H_{ij}^0}{\sigma_i}$)
y(*ndata*) : weighted data ($y_i \equiv \frac{y_i}{\sigma_i}$)
index_data(*nofp* + 1) : vector of indexes such that *y*(*index_data*(*p*) : *index_data*(*p* + 1) – 1)) are data points of exposure *p*
beta : penalty parameter β
maxiter : number of maximum subiterations

```

1: lastarr(1, 1 : nofs) = 1 {Array of change points}
2: best(1 : nofp) = 1e77 {Initialize the cost function for each iteration}
3: for R=2 to nofp do {Add a new set of data at each iteration R}
4:   converge = true {The cost function at iteration is improved}
5:   iter = 0 {Subiteration counter}
6:   m = index_data(R + 1) – 1 {Total number of data points used at iteration R}
7:   lastarr(R, 1 : nofs) = lastarr(R – 1, 1 : nofs) {Extend the previous segments}
8:   while converge do
9:     {Compute the cost at iteration R}
10:    {next_bestJ(1 : nofs): best cost function for each source}
11:    {next_lastrJ(1 : nofs): and the corresponding change-point locations}
12:    {cost_without_new_chgpt: cost function with no new change points added}
13:    {nseg_array(1 : nofs): the number of segments per source with no change points added}
14:    N = sum(nseg_array(1 : nofs)) {Total number of segments/column of matrix H}
15:    if N <= m then {If the system of equations is not underdetermined}
16:      iter = iter + 1
17:      [bestRj, isonum] = minloc(next_bestJ) {Minimum cost value, its location in terms of source number}
18:      if bestRj < cost_without_new_chgpt then {Addition of a new change point improves the cost function}
19:        best(R) = bestRj
20:        lastarr(R, isonum) = lastRj(isonum) {Update the array of change points}
21:      else
22:        converge = false {No improvement of the cost function}
23:        best(R) = cost_without_new_chgpt
24:      end if
25:      if iter > maxiter then {Maximum subiterations reached}
26:        converge = false {Next iteration}
27:      end if
28:    else
29:      converge = false {Next iteration}
30:    end if
31:  end while
32: end for
  
```

Output:

lastarr(1 : *nofp*, 1 : *nofs*) : vector *lastarr*(*nofp*, *Jnum*) is backtracked/peeled off to obtain the location of the change points of source *Jnum*

Algorithm 6 “Data space” cost function computation**Input:**

R :	iteration number/number of used exposures
$nofp$:	number of exposures
$nofs$:	number of sources
$ndata$:	number of data points
$Hzero(ndata : nofs)$:	weighted initial transfer matrix $\left(H^0 \equiv \frac{H_{ij}^0}{\sigma_i}\right)$
$y(ndata)$:	weighted data ($y_i \equiv \frac{y_i}{\sigma_i}$)
$index_data(nofp + 1)$:	vector of indexes such that $y(index_data(p) : index_data(p + 1) - 1)$ are data points of exposure p
$beta$:	penalty parameter β
$maxiter$:	number of maximum subiterations
$lastarr(1 : nofp, 1 : nofs)$:	vector $lastarr(nofp, Jnum)$ is backtracked/peel-off to obtain the location of the change points of source $Jnum$

```

1: for  $iso = 1$  to  $nofs$  do {Find the  $nseg$  segments in the array  $lastarr(1 : R, iso)$ }
2:    $nseg\_array(iso) = nseg$ 
3: end for
4:  $m = index\_data(R + 1) - 1$  {Total number of data points used at iteration  $R$ }
5:  $valid\_source(1 : nofs) = 1$  {All sources are to be “partitioned”}
6: {Optionally, by setting  $valid\_source(J) = 0$  and  $nseg\_array(J) = 0$ , source  $J$  is not used to build the matrix  $H$ ; the source makes the system of equations not well-conditioned or the source is detected with a too low efficiency (low surface area ( $cm^2$ ) projected on the camera).}
7:  $N = sum(nseg\_array)$  {Number of columns in matrix  $H$ }
8: if  $N > m$  then {Return if this system is underdetermined}
9:    $cost\_without\_new\_chgpt = 0$ .
10:   $next\_bestJ = 1.0d + 77$ 
11:   $next\_lastrJ = -1$ 
12:  return
13: end if
14:  $H(1 : m, 1 : N) = 0$ . {Initialization of the matrix  $H$ }
15:  $kw = 0$ 
16: for  $iso = 1$  to  $nofs$  do {Build the transfer function}
17:   set  $ka(1 : nseg + 1)$  {Peel-off the vector  $last(1 : R, iso)$ , the segment  $j$  starts at  $ka(j)$  and ends at  $ka(j+1)-1$ }
18:   if  $valid\_source(iso) == 1$  then {Consider only the list of valid sources}
19:     for  $j = 1$  to  $nseg$  do {Expand column  $H(1 : m, iso)$ }
20:        $kw = kw + 1$ 
21:       for  $i = ka(j)$  to  $ka(j + 1) - 1$  do
22:          $ilow = index\_data(i); iupp = index\_data(i + 1) - 1$ 
23:          $H(ilow : iupp, kw) = Hzero(ilow : iupp, iso)$ 
24:       end for
25:     end for
26:   end if
27: end for
28: {Solve the system of equations and return the auxiliary quantities needed to compute the cost function}
29:  $cost\_without\_new\_chgpt = -log(P)$  {Cost function with  $N$  segments}
30:  $irowmin(1 : nofs) = 1$  {Contains the starting of the “loop i” for each sources}
31: {Optionally modify the starting values  $irowmin(1 : nofs)$  of the “loop i”}
32: for  $Jnum = 1$  to  $nofs$  do {Identify the best new change point for each source}
33:   {Loop on source (Algorithm 7) returns  $next\_bestJ(1 : nofs)$  and  $next\_lastrJ(1 : nofs)$ }
34: end for [Loop on source  $Jnum$ ]

```

Output:

$next_bestJ(1 : nofs)$:	best cost function for each sources
$next_lastrJ(1 : nofs)$:	and the corresponding vector of change points
$cost_without_new_chgpt$:	cost function with no new change points added
$nseg_array(1 : nofs)$:	the number of segments per source with no new change points added

Algorithm 7 “Loop on sources”**Output:**

```

next_bestJ(1 : nofs):    best cost function for each source
next_lastrJ(1 : nofs):  and the corresponding change-point locations
1: next_lastrJ(Jnum) = lastarr(R, Jnum) {Best change-point location}
2: next_bestJ(Jnum)1.0e77 {and the default cost function}
3: if valid_source(Jnum) = 1 then {If the source is to be “partitioned”}
4:   cost(1 : R) = 1.0e77 {Initialize to high value}
5:   for irow = irowmin(Jnum) to R do {Test the change point located at irow}
6:     if irow > 1 then { }
7:       {Identify the segments delimiter ka(1 : nseg + 1) and the number of segments nseg in array last(1 : irow - 1, Jnum)}
8:       ka(nseg + 2) = R + 1 ; nseg = nseg + 1
9:       ka(1) = 1, ka(2) = R + 1 ; nseg = 1
10:    else
11:      ka(1) = 1, ka(2) = R + 1 ; nseg = 1 {Special case of a single segment irow = 1}
12:    end if
13:    N_star = N - nseg_array(Jnum) + nseg {Number of segments of the new matrix H_star}
14:    if N_star ≤ m then {If the resulting system is not underdetermined}
15:      Hj_star(1 : m, 1 : nseg) = 0.0 {Initialize the matrix Hj_star}
16:      kw = 0
17:      for j = 1 to nseg do {Construct the new submatrix for source Jnum}
18:        kw = kw + 1
19:        for i = ka(j) to ka(j + 1) - 1 do
20:          ilow = index_data(i)(i) and iupp = index_data(i + 1) - 1
21:          Hj_star(ilow : iupp, kw) = Hzero(ilow : iupp, Jnum)
22:        end for
23:      end for
24:      set correct_conf = 1 {This submatrix has some desired mathematical properties}
25:      {If this submatrix does not have some desired mathematical properties, set correct_conf = 0, for example for a null column}
26:      if correct_conf = 1 then {Insert this submatrix in the place of the old submatrix to form the new matrix of the partition}
27:        kmin = sum(nseg_array(1 : Jnum - 1)) ; kmax = kmin + nseg_array(Jnum)
28:        H_star(1 : m, 1 : N_star) = 0 {Initialize the matrix H_star}
29:        H_star(1 : m, 1 : kmin) = H(1 : m, 1 : kmin) {Original matrix H}
30:        H_star(1 : m, kmin + 1 : kmin + nseg) = Hj_star(1 : m, 1 : nseg) {Submatrix corresponding to source Jnum}
31:        H_star(1 : m, kmin + nseg + 1 : N_star) = H(1 : m, kmax + 1 : N) {Original matrix H}
32:        { Solve the system equation and compute the auxiliary quantities to compute the cost function}
33:        if well_conditioned_system = 1 then {If the system of equation is well conditioned or fulfills some constraints}
34:          cost(irow) = -log(P) {Cost function with the N_star segments}
35:        end if{End of well_conditioned_system = 1 }
36:      end if{End Correct_conf = 1}
37:    else
38:      {Display optionally a WARNING because of a subdetermined system of equations: N_star > m}
39:    end if{Loop N_star < m}
40:  end for
41:  {Locate the most probable change point in the array cost(irowmin(Jnum) : R)}
42:  [next_bestJ(Jnum), next_lastrJ(Jnum)] = minloc(cost(irowmin(Jnum) : R)) {Best cost value and corresponding location}
43: end if{If the source is to be ‘partitioned’ valid_source(Jnum) = 1}

```

Output:

```

next_bestJ(1 : nofs) :    best cost function for each sources
next_lastrJ(1 : nofs) :  and the change-points position

```