Computation of Hardness Ratios using BEHR

Revised July 10, 2008

Note:

- This description of the computation of hardness ratios using BEHR is based on the IDL routines **`colbehr.pro`**, **`BEHR_prior_propagate.pro`**, and **`eqt_interval.pro`** written by Vinay Kashyap, and has been adapted for use in Level 3 pipeline processing by Ian Evans. ***For correct operation, the required version of BEHR is 09 July 2008.***

Inputs:

- Required inputs
  - C_s^n = Number of counts in source region of n^{th} observation in soft energy band
  - C_m^n = Number of counts in source region of n^{th} observation in medium energy band
  - C_h^n = Number of counts in source region of n^{th} observation in hard energy band
  - B_s^n = Number of counts in background region of n^{th} observation in soft energy band
  - B_m^n = Number of counts in background region of n^{th} observation in medium energy band
  - B_h^n = Number of counts in background region of n^{th} observation in hard energy band
  - S_s^n = Background scaling factor of n^{th} observation in soft energy band
    - S is defined as the fraction (area of background region / area of source region)
  - S_m^n = Background scaling factor of n^{th} observation in medium energy band
  - S_h^n = Background scaling factor of n^{th} observation in hard energy band
  - A_s^n = Effective area at location of source region of n^{th} observation in soft energy band (in units of cm^2 s)
  - A_m^n = Effective area at location of source region of n^{th} observation in medium energy band (in units of cm^2 s)
  - A_h^n = Effective area at location of source region of n^{th} observation in hard energy band (in units of cm^2 s)
  - L_{c} = Confidence level at which to report error — default to 0.683 (*i.e.*, 1σ)
- Optional inputs
  - N_{d} = Number of draws — default to 100000
  - N_{b} = Number of burn-in draws — default to 50000
  - N_{bin} = Maximum number of bins used to construct the histogram prior distributions — default to 100
  - P_{bs} = Minimum bin size used to construct the histogram prior distributions — default to 0.0
  - P_{box} = Boxcar filter size used to smooth the histogram prior distributions — default to 3

Outputs:

- Hardness Ratios
  - H_{ms} = Medium band to soft band hardness ratio
  - H_{ms-} = Medium band to soft band hardness ratio lower confidence limit
  - H_{ms+} = Medium band to soft band hardness ratio upper confidence limit
  - H_{hm} = Hard band to medium band hardness ratio
  - H_{hm-} = Hard band to medium band hardness ratio lower confidence limit
  - H_{hm+} = Hard band to medium band hardness ratio upper confidence limit
  - H_{hs} = Hard band to soft band hardness ratio
  - H_{hs-} = Hard band to soft band hardness ratio lower confidence limit
  - H_{hs+} = Hard band to soft band hardness ratio upper confidence limit
  - F_{vs} = Variable spectrum flag

Algorithm:

1. Assume there are *m* observations of a source. For each observation, compute the net total source counts, $C_t^n = C_s^n - B_s^n / S_s^n + C_m^n - B_m^n / S_m^n + C_h^n - B_h^n / S_h^n$. Reorder the observations according to increasing number of net total source counts, C_t^n, and designate the reordered observations 1, 2, ..., $m - 1$, m.
2. Set F_{vs} = **`FALSE`**.
3. For each observation $n = 1, ..., m$ perform steps 2–6 as follows.
4. Set $C_s = C_s^n$, $C_m = C_m^n$, $C_h = C_h^n$, $B_s = B_s^n$, $B_m = B_m^n$, $B_h = B_h^n$, $S_s = S_s^n$, $S_m = S_m^n$, $S_h = S_h^n$, $A_s = A_s^n$, $A_m = A_m^n$, and $A_h = A_h^n$.
5. Set default values:
   - If $N_b < \min(5000, N_d / 2)$ then set $N_b = \min(5000, N_d / 2)$, where min() is the minimum of the values included in the parentheses.
   - If $N_b \geqslant N_d$ then set $N_b = N_d - 1$
   - If $\{[(C_s - B_s / S_s) < 15.0] \,||\, [(C_m - B_m / S_m) < 15.0] \,||\, [(C_h - B_h / S_h) < 15.0]\}$ &&
     $\{[(C_s - B_s / S_s) > 100.0] \,||\, [(C_m - B_m / S_m) > 100.0] \,||\, [(C_h - B_h / S_h) > 100.0]\}$ &&
     $\{N_d < 50000\}$ then set $N_d = 50000$.
6. Run BEHR on the soft and medium bands:
   - If $n = 1$, then run the BEHR executable as follows, capturing all of the lines of output written to **`stdout`**:
     - **`BEHR softsrc=`**C_s **`hardsrc=`**C_m **`softbkg=`**B_s **`hardbkg=`**B_m **`softarea=`**S_s **`hardarea=`**S_m **`softeff=`**A_s **`hardeff=`**A_m **`softidx=1 hardidx=1 softscl=0 hardscl=0 post=true level=`**$L_c \times 100.0$ **`algo=gibbs details=true nsim=`**N_d **`nburnin=`**N_b **`HPD=true output=SM outputMC=true`**

- If $n > 1$, then run the BEHR executable as follows capturing all of the lines of output written to **`stdout`**:
  - **`BEHR softsrc=`**C_s **`hardsrc=`**C_m **`softbkg=`**B_s **`hardbkg=`**B_m **`softarea=`**S_s **`hardarea=`**S_m **`softeff=`**A_s **`hardeff=`**A_m **`softidx=1 hardidx=1 softscl=0 hardscl=0 post=true level=`**$L_c \times 100.0$ **`algo=gibbs details=true nsim=`**N_d **`nburnin=`**N_b **`HPD=true output=SM outputMC=true softtbl='tblprior_soft.txt' hardtbl='tblprior_med.txt'`**
- Read the simulated draws:
  - Open the file **`SM_draws.txt`** created by the BEHR executable for reading.
  - Read $N_d - N_b$ rows of data containing two columns per row; set D_s equal to the array of length ($N_d - N_b$) of values extracted from the 1st column of data read from the file and set D_m equal to the array of length ($N_d - N_b$) of values extracted from the 2nd column of data read from the file.
- Search through the lines of output written to **`stdout`** by the BEHR executable for a line that contains any of the following substrings:
  - "**`WARNING: THE TABULATED PRIOR CONTRADICTS THE LIKELIHOOD OF THE DATA`**" (10 words on a single line, separated by single spaces, not including the double quotes), or
  - "**`WARNING: THE TABULATED PRIOR FOR THE 'SOFT' BAND CONTRADICTS THE`**" (10 words on a single line, separated by single spaces, not including the double quotes), or
  - "**`WARNING: THE TABULATED PRIOR FOR THE 'HARD' BAND CONTRADICTS THE`**" (10 words on a single line, separated by single spaces, not including the double quotes).
- If any of the substrings are found, then set F_{vs} = **`TRUE`**.

7. Run BEHR on the soft and hard bands:
   - If $n = 1$, then run the BEHR executable as follows capturing all of the lines of output written to **`stdout`**:
     - **`BEHR softsrc=`**C_s **`hardsrc=`**C_h **`softbkg=`**B_s **`hardbkg=`**B_h **`softarea=`**S_s **`hardarea=`**S_h **`softeff=`**A_s **`hardeff=`**A_h **`softidx=1 hardidx=1 softscl=0 hardscl=0 post=true level=`**$L_c \times 100.0$ **`algo=gibbs details=true nsim=`**N_d **`nburnin=`**N_b **`HPD=true output=SH outputMC=true`**
   - If $n > 1$, then run the BEHR executable as follows capturing all of the lines of output written to **`stdout`**:
     - **`BEHR softsrc=`**C_s **`hardsrc=`**C_h **`softbkg=`**B_s **`hardbkg=`**B_h **`softarea=`**S_s **`hardarea=`**S_h **`softeff=`**A_s **`hardeff=`**A_h **`softidx=1 hardidx=1 softscl=0 hardscl=0 post=true level=`**$L_c \times 100.0$ **`algo=gibbs`**

```
details=true nsim=N_d nburnin=N_b HPD=true output=SH
outputMC=true softtbl='tblprior_soft.txt'
hardtbl='tblprior_hard.txt'
```

- Read the simulated draws:
  - Open the file **`SH_draws.txt`** created by the BEHR executable for reading.
  - Read $N_d - N_b$ rows of data containing two columns per row; set D_h equal to the array of length $(N_d - N_b)$ of values extracted from the 2nd column of data read from the file.
  - Search through the lines of output written to **`stdout`** by the BEHR executable for a line that contains any of the following substrings:
    - "**`WARNING: THE TABULATED PRIOR CONTRADICTS THE LIKELIHOOD OF THE DATA`**" (10 words on a single line, separated by single spaces, not including the double quotes), or
    - "**`WARNING: THE TABULATED PRIOR FOR THE 'SOFT' BAND CONTRADICTS THE`**" (10 words on a single line, separated by single spaces, not including the double quotes), or
    - "**`WARNING: THE TABULATED PRIOR FOR THE 'HARD' BAND CONTRADICTS THE`**" (10 words on a single line, separated by single spaces, not including the double quotes).
  - If any of the substrings are found, then set F_{vs} = **`TRUE`**.

8. If $n \neq m$, then create the prior distribution histograms to be applied to the next observation:
   - For each band k = s, m, h separately, compute the histograms $\mathcal{H}_k$ of the elements of the arrays D_k as follows:
     - Set $P_{\max,k} = \max(D_k)$, where max() is the maximum value of the elements of all of the arrays included in the parentheses. If the resulting $P_{\max,k} < 0$, then set $P_{\max,k} = 0$.
     - Set $P_{\min,k} = \min(D_k)$, where min() is the minimum value of the elements of all of the arrays included in the parentheses. If the resulting $P_{\min,k} < 0$, then set $P_{\min,k} = 0$.
     - Set $P_{\text{bin},k} = \max(P_{\text{bs}}, (P_{\max,k} - P_{\min,k}) / N_{\text{bin}})$, where max() is the maximum of the values included in the parentheses.
     - If $P_{\text{bin},k} \neq 0$, then reset $N_{\text{bin},k} = \text{ceil}[(P_{\max,k} - P_{\min,k}) / P_{\text{bin},k}]$, where ceil[] is the smallest integer not less than the value included in the brackets. Otherwise ($P_{\text{bin},k} = 0$) reset $N_{\text{bin},k} = 1$.
     - Construct the histogram $\mathcal{H}_s$ of the elements of the array D_s. Set the left edge of the first bin to $P_{\min,s}$, the bin size to $P_{\text{bin,s}}$, and the number of bins to $N_{\text{bin,s}}$. The right edge of the last bin should be set to $P_{\min,s} + N_{\text{bin,s}} \times P_{\text{bin,s}}$.
     - Construct the histograms $\mathcal{H}_m$ and $\mathcal{H}_h$ of the elements of the arrays D_m and D_h, respectively, using bin parameters $P_{\min,m}$, $P_{\text{bin,m}}$, $N_{\text{bin,m}}$ when computing $\mathcal{H}_m$, and bin parameters $P_{\min,h}$, $P_{\text{bin,h}}$, $N_{\text{bin,h}}$ when computing $\mathcal{H}_h$.
   - Compute the normalized, smoothed histogram $\mathcal{H}_{s,n}$ by smoothing $\mathcal{H}_s$ with a boxcar filter of length P_{box}, and then dividing the result by the product of the sum of the smoothed histogram

values and the bin width, $P_{bin,s}$.

- Compute the normalized, smoothed histograms $\mathcal{H}_{m,n}$ and $\mathcal{H}_{h,n}$ from $\mathcal{H}_m$ and $\mathcal{H}_h$, respectively, using the same method as the previous step with the bin widths $P_{bin,m}$, and $P_{bin,h}$, respectively.
- Create the text file **`tblprior_soft.txt`** to be used when processing the next observation:
  - Output the (integer) value N_{bin} to the first line of the file
  - Output the following header (excluding the quotation marks) to the next line of the file: "**`lamS	Pr_lamS`**". The two text elements are separated by a tab character.
  - Output $N_{bin,s}$ tab-delimited tuples containing (bin center value, bin value) for each bin of the smoothed histogram $\mathcal{H}_{s,n}$ to subsequent lines of the file, one tuple per line.

  Example:

  ```
  200
  lamS	Pr_lamS
  11.000000	0.0042700937
  11.250000	0.0074726640
  11.500000	0.0074726640
  11.750000	0.0021350469
  12.000000	0.0042700937
  ...
  ```

- Create the text files **`tblprior_med.txt`** and **`tblprior_hard.txt`** to be used when processing the next observation from the smoothed histograms $\mathcal{H}_{m,n}$ and $\mathcal{H}_{h,n}$, respectively, using the same method as the previous step. The headers for the text files should be (in order) the integer value $N_{bin,m}$ and "**`lamM	Pr_lamM`**" for the text file **`tblprior_med.txt`**, and the integer value $N_{bin,h}$ and "**`lamH	Pr_lamH`**" for the text file **`tblprior_hard.txt`**.

9. After all of the observations have been processed through steps 1–6, compute the hardness ratios and confidence intervals:
   - Set $D_t = (D_s + D_m + D_h)$
   - Create the arrays $D_{ms} = (D_m - D_s) / D_t$,
     $D_{hm} = (D_h - D_m) / D_t$, and
     $D_{hs} = (D_h - D_s) / D_t$, for all array elements i for which $D_t(i) \neq 0$.
   - Compute the hardness ratios H_{ms} = mean value of D_{ms} ,
     H_{hm} = mean value of D_{hm} , and
     H_{hs} = mean value of D_{hs} .
   - Determine the confidence intervals on the hardness ratios by computing the lower and upper bound equal-tail confidence intervals (H_{ms-} , H_{ms+}), (H_{hm-} , H_{hm+}), and (H_{hs-} , H_{hs+}), on the arrays D_{ms} , D_{hm} , and D_{hs} around the mean values H_{ms} , H_{hm} , and H_{hs} , respectively, with confidence level L_c (see below).

Computing equal-tail confidence intervals:

- Given an array of values D_{xy} with mean value $\bar{D}_{xy}$, and a confidence level L_c, proceed as follows:
  - Sort the array D_{xy} in ascending order. All subsequent references to D_{xy} refer to the sorted array.
  - Create an array C_{xy} with the same number of elements as D_{xy}, populated with values 0, $1/(N-1)$, $2/(N-1)$, ... , $(N-2)/(N-1)$, 1, where N is the number of elements of C_{xy}.
  - Identify the two consecutive elements $[D_{xy}(i), D_{xy}(i+1)]$ of D_{xy} for which $D_{xy}(i) \leq \bar{D}_{xy} < D_{xy}(i+1)$.
  - Compute the value $\bar{C}_{xy} = (C_{xy}(i+1) \times [\bar{D}_{xy} - D_{xy}(i)] + C_{xy}(i) \times [D_{xy}(i+1) - \bar{D}_{xy}]) / [D_{xy}(i+1) - D_{xy}(i)]$.
    - If no element $D_{xy}(i)$ satisfies the relation $D_{xy}(i) \leq \bar{D}_{xy}$, then set $\bar{C}_{xy} = 0$.
    - If no element $D_{xy}(i+1)$ satisfies the relation $\bar{D}_{xy} < D_{xy}(i+1)$, then set $\bar{C}_{xy} = 1$.
  - Compute the lower and upper confidence levels $L_{c-} = \bar{C}_{xy} - L_c \bar{C}_{xy}$ and $L_{c+} = \bar{C}_{xy} + L_c (1 - \bar{C}_{xy})$.
  - Identify the two consecutive elements $[C_{xy-}(i), C_{xy-}(i+1)]$ of C_{xy} for which $C_{xy-}(i) \leq L_{c-} < C_{xy-}(i+1)$.
  - Compute the lower confidence value $D_{xy-} = (D_{xy}(i+1) \times [L_{c-} - C_{xy-}(i)] + D_{xy}(i) \times [C_{xy-}(i+1) - L_{c-}]) / [C_{xy-}(i+1) - C_{xy-}(i)]$.
    - If no elements of C_{xy} satisfy the above relation, then set the lower confidence value $D_{xy-} = min(D_{xy})$.
  - Identify the two consecutive elements $[C_{xy+}(i), C_{xy+}(i+1)]$ of C_{xy} for which $C_{xy+}(i) < L_{c+} \leq C_{xy+}(i+1)$.
  - Compute the upper confidence value $D_{xy+} = (D_{xy}(i+1) \times [L_{c+} - C_{xy+}(i)] + D_{xy}(i) \times [C_{xy+}(i+1) - L_{c+}]) / [C_{xy+}(i+1) - C_{xy+}(i)]$.
    - If no elements of C_{xy} satisfy the above relation, then set the upper confidence value $D_{xy+} = max(D_{xy})$.
  - Return the lower and upper confidence values D_{xy-} and D_{xy+}.

Sample test data and expected results:

- **Input Data 1**
  - In the following, any values not specified are set to their default values

Observation #1:

	Soft Band	Medium Band	Hard Band
Source Counts	25	33	28
Background Counts	4	7	10

Observation #2:

	Soft Band	Medium Band	Hard Band
Source Counts	22	28	29
Background Counts	3	5	11

Observation #3:

	Soft Band	Medium Band	Hard Band
Source Counts	34	40	27
Background Counts	9	14	16

- **Expected Results 1**

  H_{ms} = **0.059349000**

  (H_{ms-}, H_{ms+}) = (**–0.020777198**, **0.13805908**)

  H_{hm} = **–0.14119626**

  (H_{hm-}, H_{hm+}) = (**–0.22746541**, **–0.056571940**)

  H_{hs} = **–0.081847257**

  (H_{hs-}, H_{hs+}) = (**–0.16322040**, **–0.0025638488**)

  F_{vs} = **FALSE**

- **Input Data 2**
  - In the following, any values not specified are set to their default values

Observation #1:

	Soft Band	Medium Band	Hard Band
Source Counts	234	123	45
Background Counts	3674	2150	786
Background Scale Factor	22.661348	22.661411	22.661809
Effective Area	1807566.205719	2992805.257055	1983521.552635

Observation #2:

	Soft Band	Medium Band	Hard Band
Source Counts	339	221	110
Background Counts	4477	3113	1046
Background Scale Factor	14.482264	14.482376	14.482441
Effective Area	34547345.885866	64336614.013749	45080717.873112

- **Expected Results 2**

  H_{ms} = **-0.54242507**

  (H_{ms-} , H_{ms+}) = (**-0.67159630**, **-0.40929599**)

  H_{hm} = **-0.068660230**

  (H_{hm-} , H_{hm+}) = (**-0.15790591**, **0.020057840**)

  H_{hs} = **-0.61108530**

  (H_{hs-} , H_{hs+}) = (**-0.72982033**, **-0.48616630**)

  F_{vs} = **TRUE**

Note:

Because the statistical distributions are computed from pseudo-random samples, the values are not expected to match exactly from run to run or across platforms.