Documentation of the SOHO/EPHIN histflux data product

Patrick Kuehl

April 29, 2021

Contacts

- Patrick Kühl, kuehl@physik.uni-kiel.de
- Bernd Heber, heber@physik.uni-kiel.de

Contents

1 TODO

TODO:

- check issue with jumping fluxes (i.e. finer energy range or sth like that?)

2 Introduction

The SOHO/EPHIN histogram spectra are a data product based on the EPHIN histogram data. Some previous, older documentation is listed in section A.1. For general informations regarding EPHIN the reader is refered to Mueller-Mellin, 1995 or any publication from Kuehl, P. and Heber, B. between 2010 and 2020. Information to the EPHIN level3 data product to which this document refers to various times can be found here: `http://ulysses.physik.uni-kiel.de/costep/level3/l3i/DOCUMENTATION-COSTEP-EPHIN-L3-20181002.pdf`. A sketch of the EPHIN instrument is given in figure 1.
The motivation for the histogram data is as follows: Main data products of EPHIN are either based on coincidence counters or PHA. While the former do include all measured particles (great statistics), the partice seperation and energy determination is limited. The PHA data sets on the other hand offer the best paticle seperation and energy determination. Due to data limitation, however, the PHA data set is only a statistical sample of the particles measured and hence often show statistical limitations. The histogram data set sits somewhere in-between: The particle identification and energy determination is superb to coincidence counters (as will be shown in sections 4 and 5), while having a counting statistics up to 30 times as high as the PHA one. This is shown in figure 2 which presents the ratio of PHA and histogram counts for the AB penetration depth (i.e. detector A and B are triggered, while C is not) as function of the total coincidence counts at the same time. During quite times conditions (where the PHA buffer is large enough to include all measured particles), the ratio is unity. However, during periods of higher flux (higher total count rate), this ratio goes down to as low as 0.03.

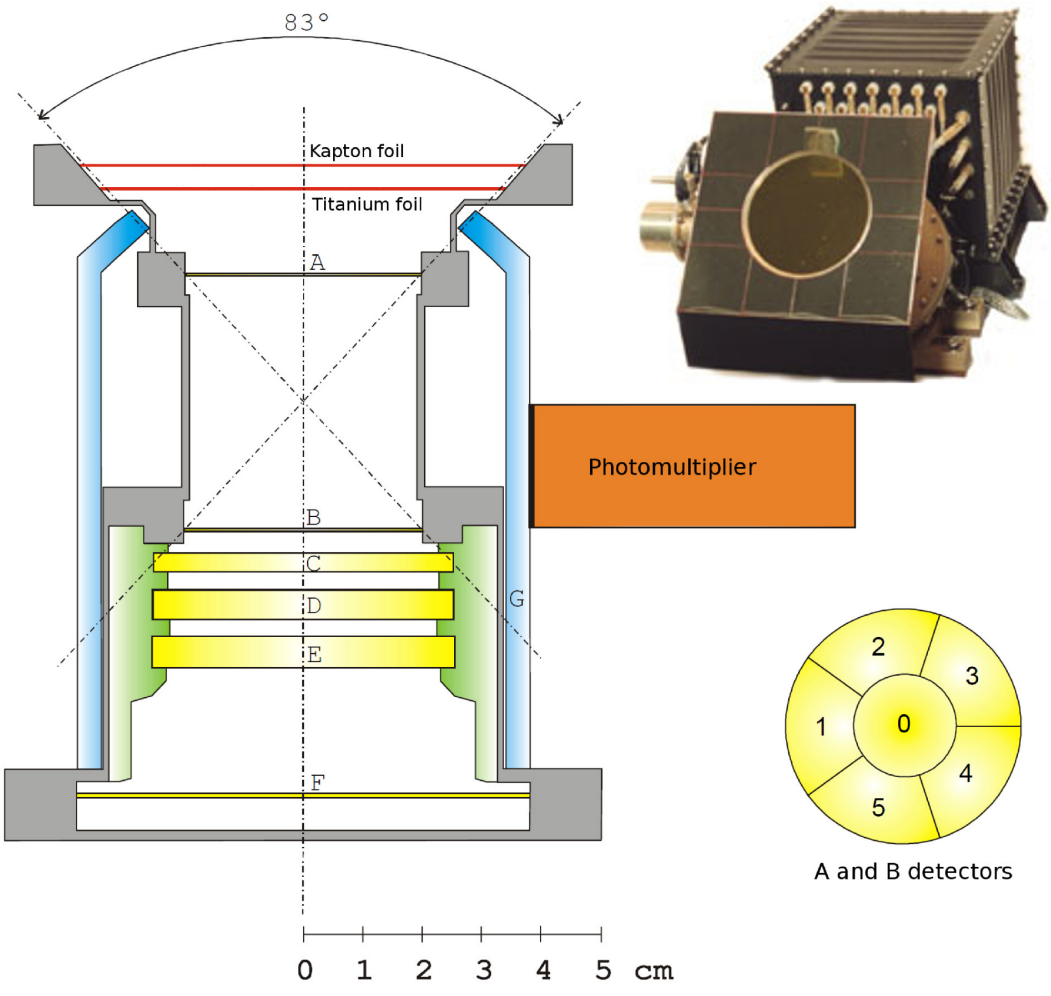

Figure 1: EPHIN

The document is structured as follows: section 3 presents and explains the histogram data structure as received from the instrument. Section 4 and 5 present the particle identification scheme as well as the calculated and validated response factors. Section 6 shows comparisons of the newly obtained data to EPHIN level3 data as well as various other datasets. Sections 7 and 8 present example plots of the new data product as well as detailed information regarding the new data product. The code for this production is explained in section 9.
The appendix include some previous documentations (sec. A.1) as well as deeper analysis on the influence of Helium3 (sec. A.2) and the response to higher energetic particles (sec A.3).
JSON files including all necessary information for the production of the data set are given in section B.1, the required python functions are defined in section B.2 while section B.3 presents a minimal example of how to use that code.

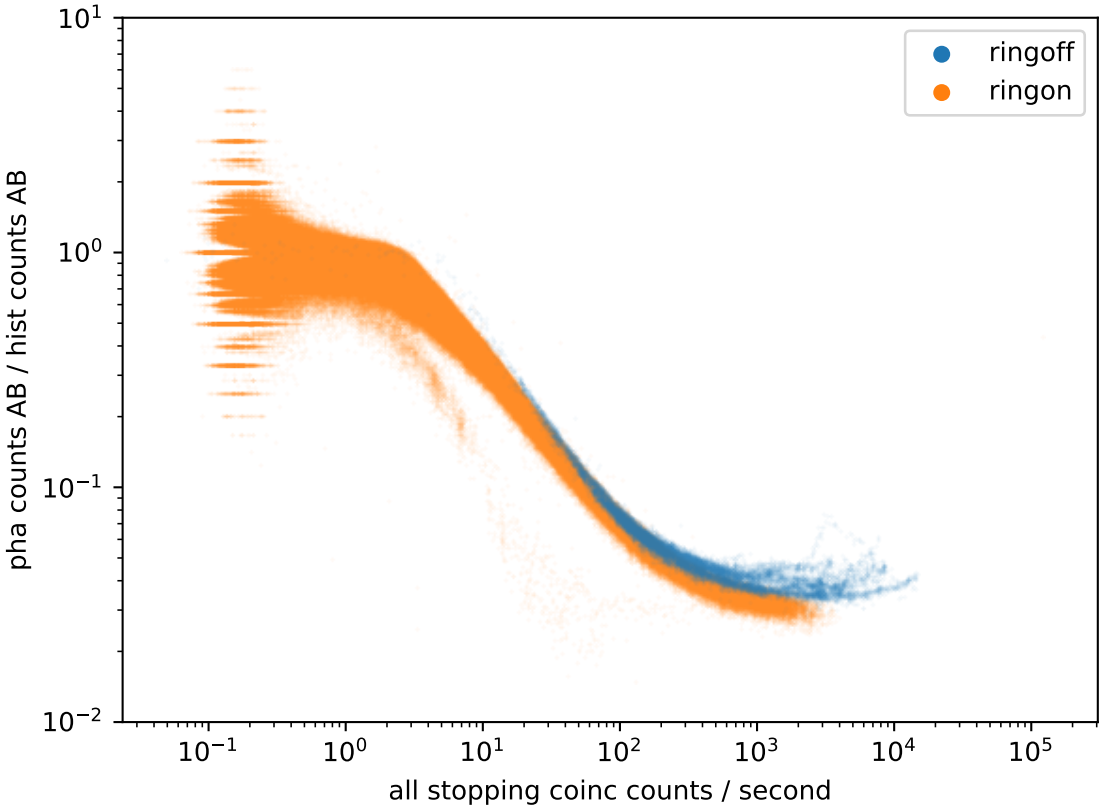

Figure 2: Ratio of PHA to hist counts in the AB coincidence depending on the total stopping coincidence rate.

3 Lvl1 Histogram data

3.1 Overview of the histogram data

The histogram data of EPHIN works as follows: During an 8 minute interval, histogram data is accumulated. For that purpose, particles are 1) analysed based on their penetration depth (AB, ABC, ABCD or ABCDE), 2) their total energy deposition is calculated from look-up tables (i.e. if penetration depth is ABC the energy deposition is E=E_A+E_B+E_C) and 3) the energy loss histogram for that penetration depth is incremented at the caluclated energy. (The energy ranges for the different penetration depths are listed in figure 36 in the appendix A.1.) Hence, after this 8 minute interval, four histograms have been filled, each presenting the total deposited energy for a certain penetration depth.
These histograms are than each split into two halfs for data transmission reasons. These eigth half-parts of a histogram are than stored in the SCI data that is transmitted. Hence over a 8 minute period after the histogram has been derived, the 8 parts of the 4 histograms are transmitted. During this time the next histogram is filled and the cycle repeats. The level1 SCI file is explained in 35 in appendix A.1, the histograms are stored in columns 62-93 . Column 94 includes the histbit (i.e. the three LSB itarate from 0 to 7 indicating which part of the histogram is currently transmited).

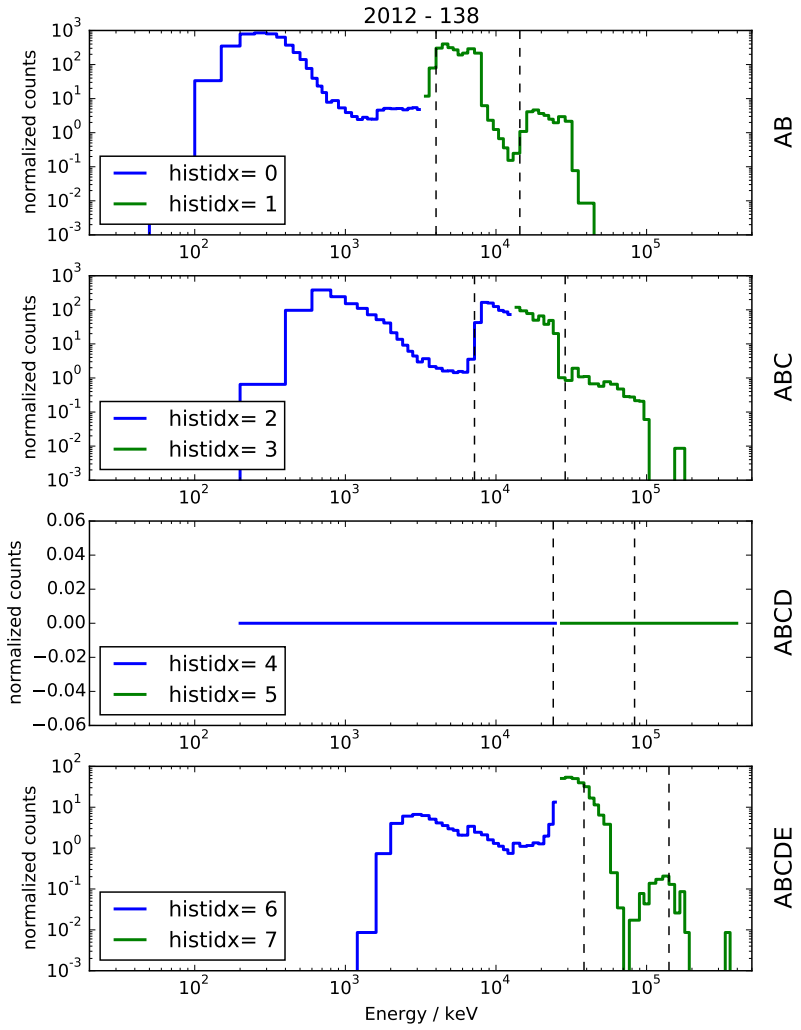

Figure 3: Histogram data for the May, 2012 event.

Figure 3 shows the measured histograms for May 17th, 2012 (i.e. GLE 71 SEP event) by presenting counts as function of deposited energy. The rows present the different penetration depths (AB, ABC, ABCD and ABCDE), each histogram half is shown seperately. Note that the ABCD histogram is empty due to failure mode E (FME). Dashed vertical lines presents the particle seperation as suggested in the 90ies that are supposed to seperate electrons, protons and helium particles. And in fact that seperation seems to be okayish as one can easily identify the different particle populations which are more or less seperated by these thresholds.
While the overall histograms are rather smooth, some unexpected jumps can be seen (cf. ABCDE histogram at 6.5 MeV). These are caused by the different bin widths (cf. figure 36 in appendix A.1) and can thus be easily corrected. Figure 4 shows the same daa but counts being normalized by their binwidths. As expected the resulting spectra does not show the artificial jumps seen in figure 3.

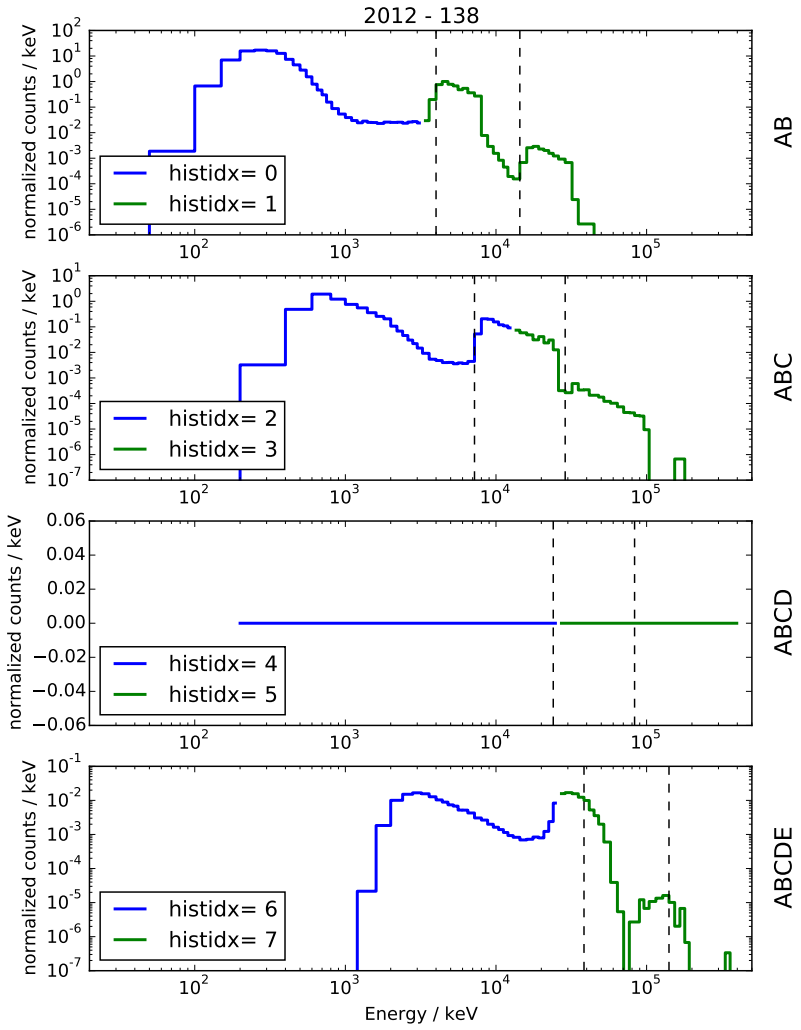

Figure 4: Histogram data for the May, 2012 event normalized by energy channel widths.

3.2 Parallel patch

The EPHIN patch table shown in figure 5 lists an "upload histogram patch (parallel incidence)". While we were not oble to find any information on this patch, the most obvious idea is that this patch limits the histogram data to parallel incidence during ringon mode (i.e. during ringon the same segment has to be triggered in B as has been triggered in A, i.e. SegA=3, SegB=3,see figure 1). This has been verified by the analysis presented in figure 6 which shows the total counts of histograms after the ring was switched on again in declining phases of SEP events as function of the total counts that has been observed while the ring was still turned off. Based on the differences in responses for arbitrary, parallel and central incidence, on can conclude that the histogram data is indeed only observing parallel incidence since all meaningful data points (orange points, the blue ones include multiple switch-on and -offs during several minutes) are on the dashed line which is the expected increase between parallel and central incidence. At very high count rates (upper part of the curve), other effects such as upper limit of histogram counts discussed in section 3.3 come into play.

Year	Month	Day	Hour	Minute	Status Flag	Change
1995	12	07	17	28	16	EPHIN ON (standby or maintenance)
1995	12	07	17	35	0	Nominal observation
1995	12	09	17	03	8	Commissioning
1995	12	10	05	15	0	Nominal observation
1996	10	31	14	47	1	Failure mode E
1996	12	12	16	42	3	Ring A/B OFF (for check-out only)
1996	12	18	14	45	1	Ring A/B ON (back to normal)
1997	02	15	10	25	-1	EPHIN OFF (no data)
1997	02	16	16	37	16	EPHIN ON (standby or maintenance)
1997	02	16	16	45	0	Nominal observation
1997	02	19	14	57	1	Failure mode E
1997	03	13	14	41	5	Upload E patch (salvage PHA_E)
1997	04	23	16	19	37	Calibration
1997	04	23	16	28	5	End calibration
1997	08	22	13	10	5	Upload histogram patch (parallel incidence)
1997	10	10	14	39	69	Enable ring segment switching

Figure 5: Table showing uploaded patches to EPHIN till 1997 (extracted from ephispec.doc).

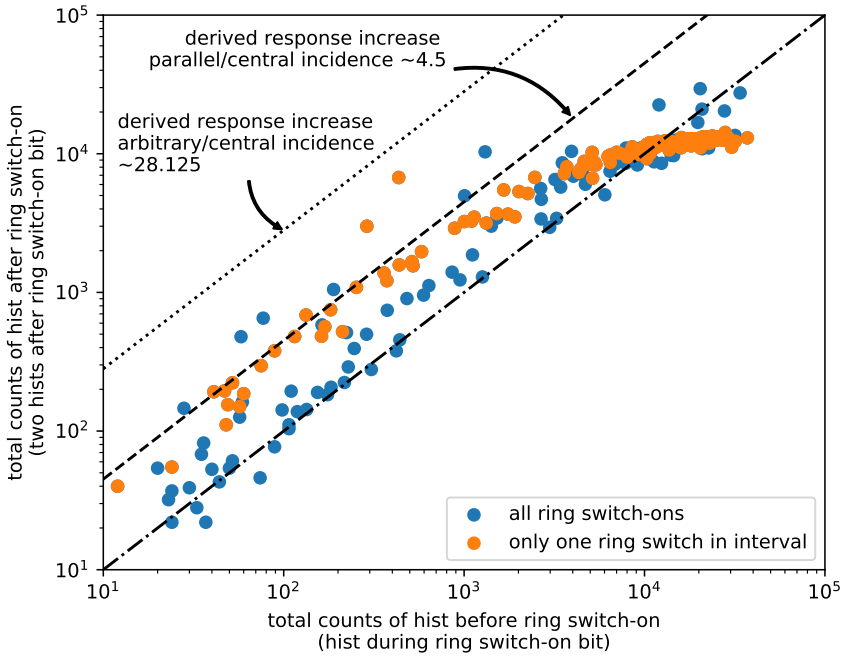

Figure 6: Hist counts before and after ring switching.

3.3 Scaling with coincidence counts and time constant

Due to the parallel patch, the usual scaling based on the ratio of total counts over counts in a particular dataproduct (as used in PHA analysis) can only be performed during ringoff conditions for the histogram data. This is due to the fact that parallel counters are only available for proton and helium coincidences but not for electron coincidences. Since particle in the electron coincidences are counted in the histogram as well, another method has to be defined to scale the histogram counts in order to account for the fact that not every particle detected is analysed in the histograms.

For this purpose, figure 7 presents the ratio of histogram counts (all penetratio depths, i.e. AB+ABC+...) over coincidence counts (i.e. total counts) as function of the total amount of stopping (coincidence) counts for ringoff (blue) and rinon (orange) conditions. The ringon ratio is expected to be ≈ 0.2 due to the differences in responses of parallel incidence (histogram) and arbitrary incidence (coincidence counts). Apart from that, both curves show a decrease at higher total count numbers. Thus, the curves have been fitted with a function describing non-paralyseable deadtime effects

$$r = 1/(1 + \tau * c) \tag{1}$$

where τ is a free parameter describing the time constant of the electronics and r, c describing the ratio of counts (y-axis) and total counts per seconds (x-axis), respectively. Based on the fit shown in fig. 7, the parameter were determined to be

$$\tau_{ring-off} = 0.0102s \tag{2}$$

and

$$\tau_{ring-on} = 0.00747s \tag{3}$$

While no detailed information on the electronic time constants could be found in the aging documentations of EPHIN, the PhD thesis from Sierks 1997 gives some indication that the order of magnitude is correct (citing from page 40):

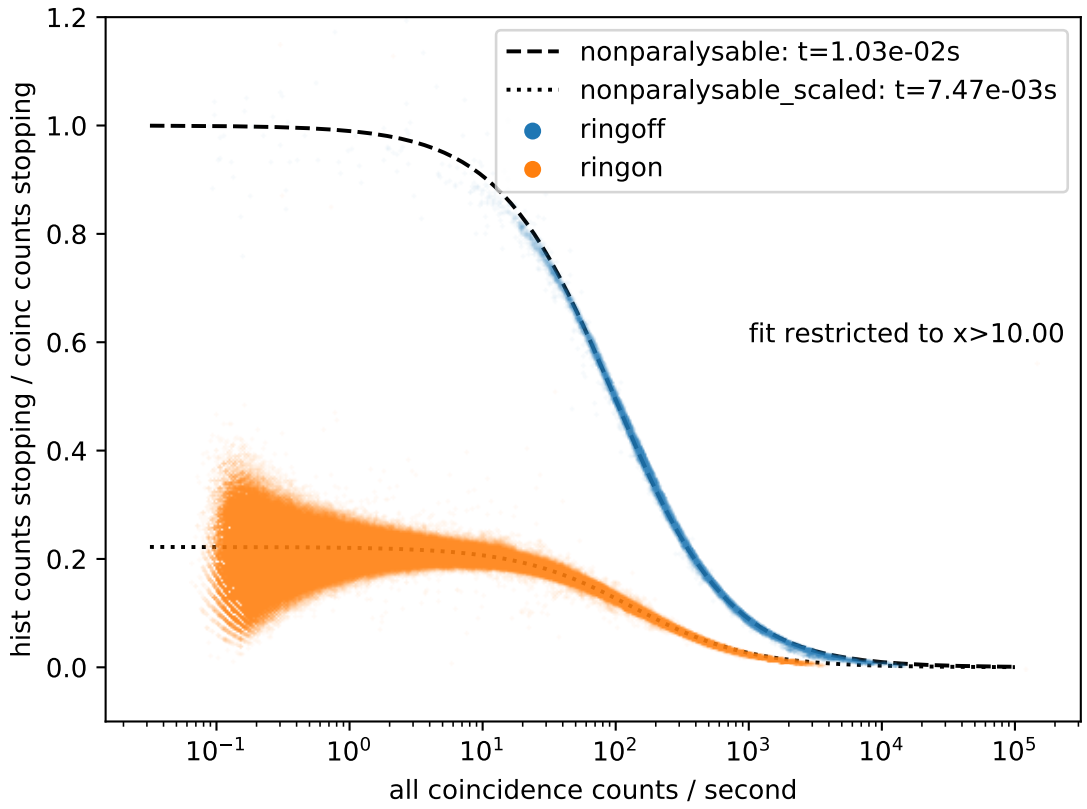

Figure 7: Ratio of histogram counts to coincidence counts for coincidence AB depending on total coincidence count rate (all channels)

> Mit einer zeitlichen Auflösung von 8 Minuten werden die auf (8+4) bit komprimierten Zählerstände der Histogramm-bins übertragen. In den Daten dieser 8 Minuten können bis zu 50 000 analysierte Teilcheneinfälle enthalten sein, so dass mit guter Statistik der Spektralverlauf der verschiedenen Spezies rekonstruiert werden kann.

The mentioned 50000 counts per 8 minute interval translates to a time constant of

$$8min/50000 = 0.0096s \tag{4}$$

which is in the same order as the ones given in equations 2 and 3.
Utilizing the dervied τ and a given total count rate (per second) of the coincidence counters, the scaling factor S for the histogram data is given by the reciprocal of equation 1:

$$S(\tau, c) = 1 + \tau * c \tag{5}$$

For the ringoff condition, however, this is not necessary since during that mode the coincidence counter have the same conditions as the histograms and hence, the scaler can be derived directly from the ratio of counts

$$S_{ratio}(c, h) = c/h \tag{6}$$

with h and c being the number of counts in histogram and coincidence, respectively. This can be also used to validate the time constant approach explained above. Figure 8 shows the relative deviation of the ratio scaler (eq. 6) and the time constant scaler (5) as function of the time constant scaler based on data with ring off between 1997 and 2017. From the figures it can be concluded that the method of deriving the scaling parameter based on timeconstants and the total count rate is valid (at least for ringoff conditions) with systematic uncertainties in the regime of single digit percentages.
For the derived histogram flux data set, the time constant method is used for ringon conditions (eq. 5 with $\tau = 0.00747s$) while the ratio of counts (eq. 6) is used for the ringoff conditions. Note that this scaling is not taking care of the electronic deadtime effects due to the analog chain (i.e. the one based on the single detector count rates and the coincidence logic).

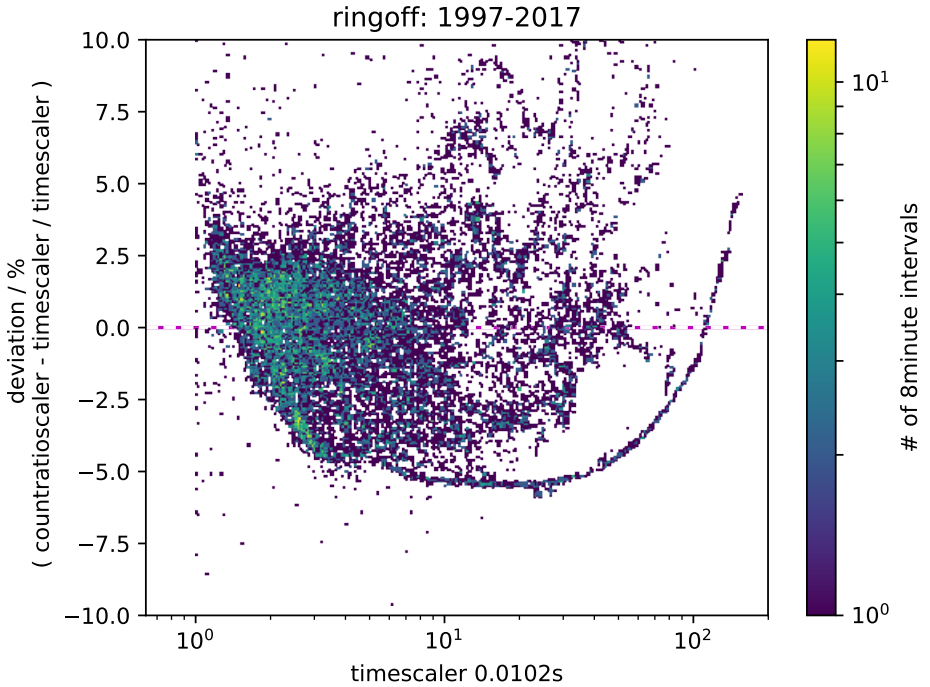

Figure 8: Relative difference of scaler for ringoff condition (1997-2017) based on ratio of coincidence counts over hist counts and the derived scaler based on time constant of 0.0102s.

4 Simulation: Particle Identification

Geant4 simulations for electrons up to 20 MeV, proton and helium particles up to 100 MeV/nuc have been performed for isotropic fluxes. The resulted energy depositions in the detector have been analysed in the same scheme as the histograms onboard, e.g. 1) the range (AB, ABC, ABCD or ABCDE) was determined and 2) the total energy deposition in that range is histogramed in the same energy bins as onboard. The simulation is designed such that in postanalysis, different spectral shapes and particle flux ratios can be rescaled.
Figure 9 shows the expected histogram for power-laws with $\gamma = -3$ and a helium to proton ratio of 0.01 for ringon and ringoff conditions. The comparison with figure 4 shows already quite an agreement despite the fact that not detailed fitting to spectral gammas was performed. Hence, the simulation is performing as expected.

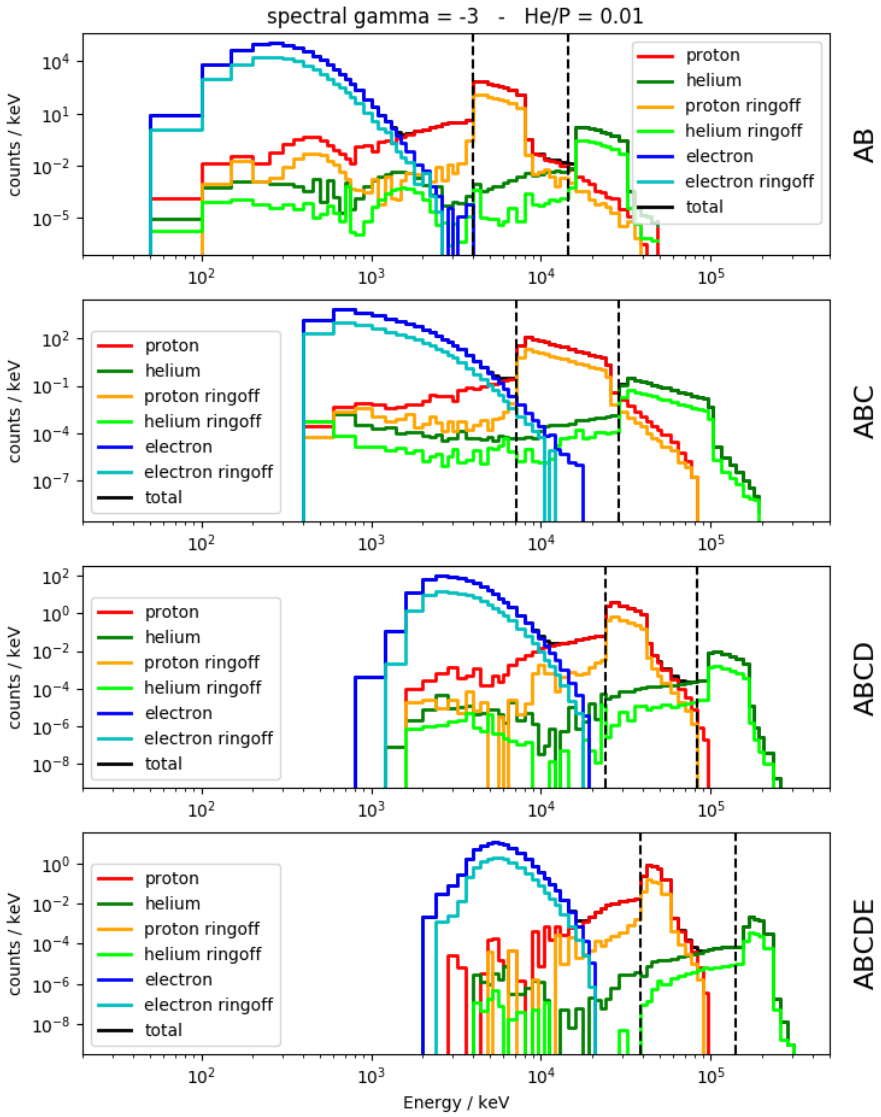

Figure 9: Histogram of simulation results.

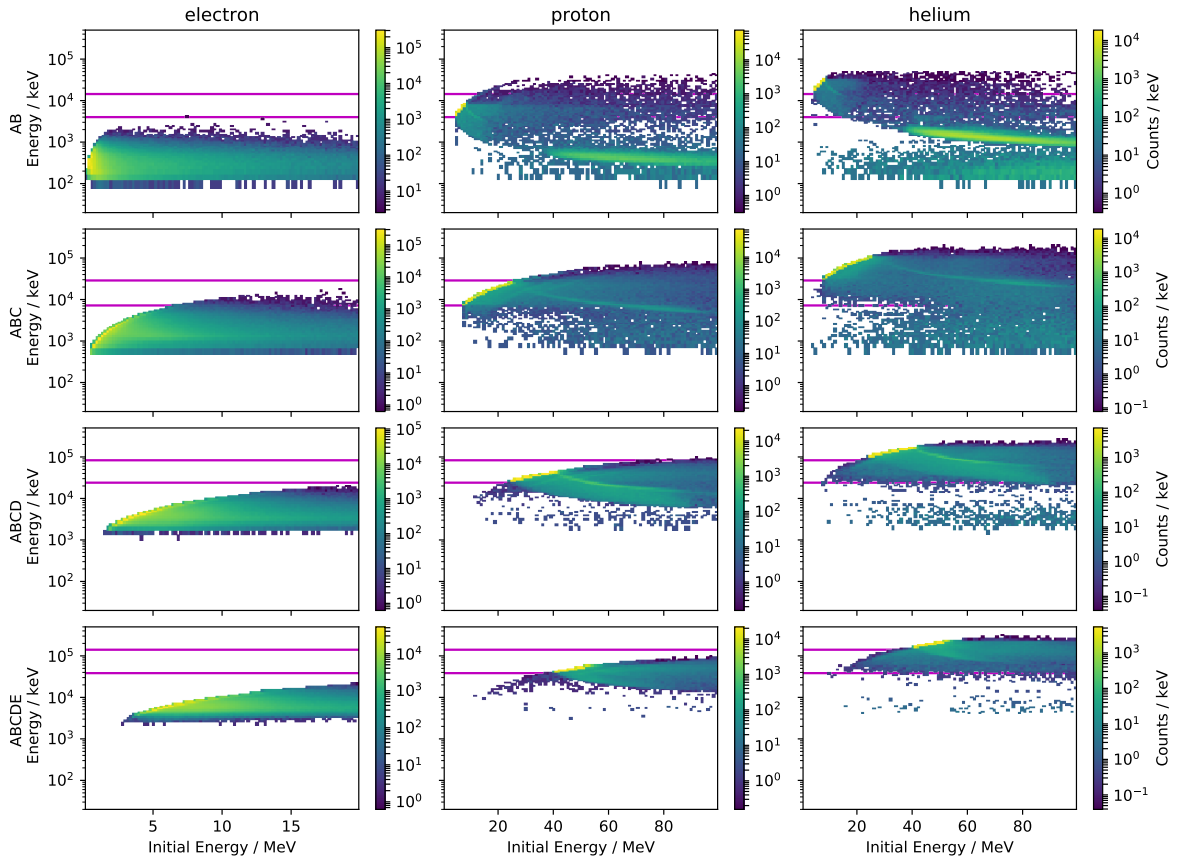

Figure 10: energy vs edep

Figures 10 and 11 show a more detailed analysis of the simulation results for ringon and ringoff conditions, respectively. Each row presents another penetration depth while the columns represent the electron, proton and helium simulations. Shown in each figure is the histogramed energy as function of the initial (simulated) energy. The magenta lines show the particle seperation thresholds defined by Sierks and others in the 90ies. The figures show 1) large contributions of electrons to all channels below the simulated energy (non-identiy contributions), 2) a clear, distinct response to proton and helium channels in certain energy ranges where the entire energy is deposited in the instrument and 3) individual tracks of higher energy protons and helium that deposit only a fraction of their entire energies (i.e. particles missing the deeper detectors, depositing energy in the housing, cf. section A.3). Since we dont investigate electrons here, the following sections focus on a) defining channels and their responses for protons and helium that deposit their entire energies (2) and how higher energetic particles (3) falsly contribute to these channels.
Based on results as shown in figure 9 thresholds for particle seperation have been defined and are given in table 1. Channels for protons and helium particles (which are used for the data product described in this document) as well as electron, background and overflow channels (all not used in here) have been defined.

Particle	AB	ABC	ABCD	ABCDE
electron	[0.0, 1.2]	[0.0, 6.4]	[0.0, 16.0]	[0.0, 16.0]
background	[1.2, 4.0]	[6.4, 7.2]	[16.0, 24.0]	[16.0, 24.0]
proton	[4.4, 8.0]	[7.2, 25.6]	[24.0, 51.2]	[24.0, 51.2]
helium	[17.6, 32.0]	[28.8, 102.4]	[96.0, 204.8]	[96.0, 204.8]
overflow	[32.0, 51.2]	[102.4, 204.8]	[230.4, 409.6]	[230.4, 409.6]

Table 1: Thresholds for the histogram channels in units of MeV.

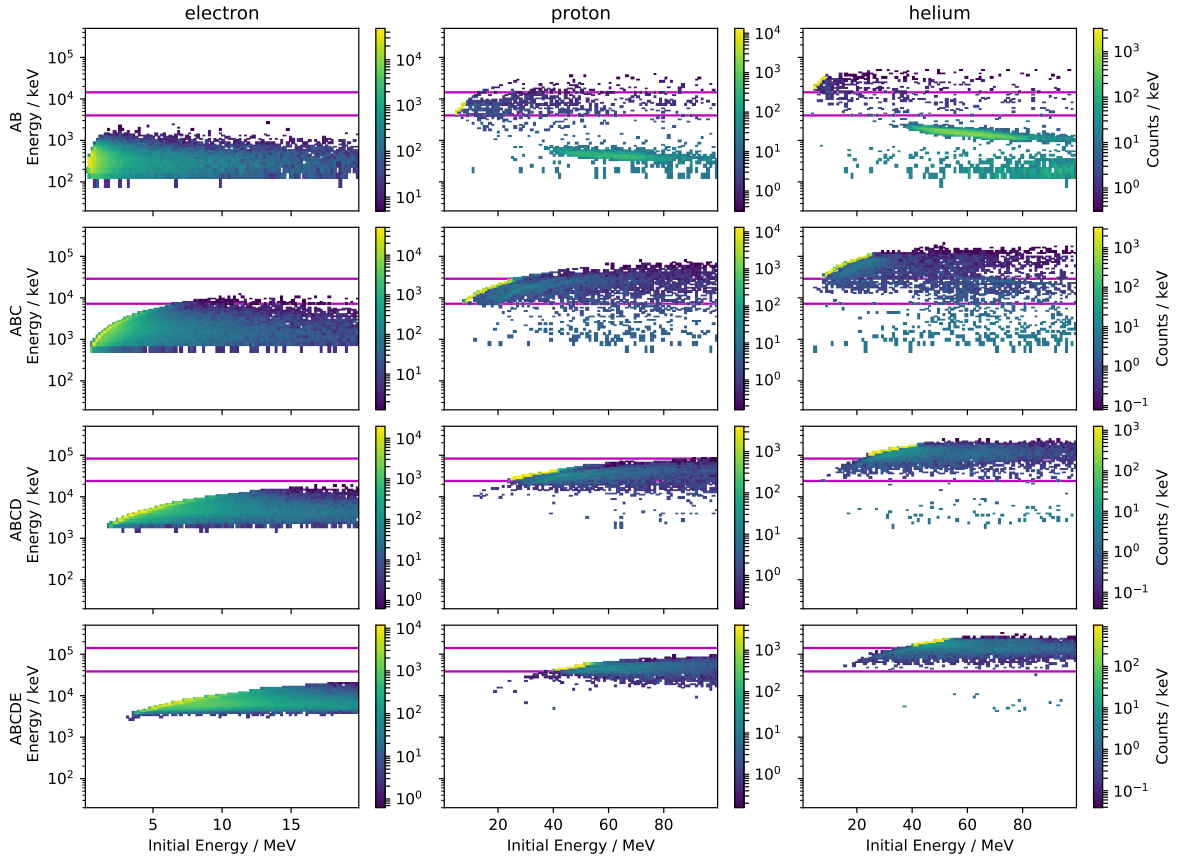

Figure 11: ring off: energy vs edep

5 Simulation: Response function

5.1 Channel response

Responses to electrons, protons and helium paticles have been derived for all 64 channels in each range histogram. The results are shown in figures 12 till 15 for ranges AB till ABCDE for ringon conditions. The letter in the top left of each panel referes to the channel type as defined in table 1. From the figures it can be concluded that the channel definition is working fine since the major contributions to a channel is caused by the correct particle type.

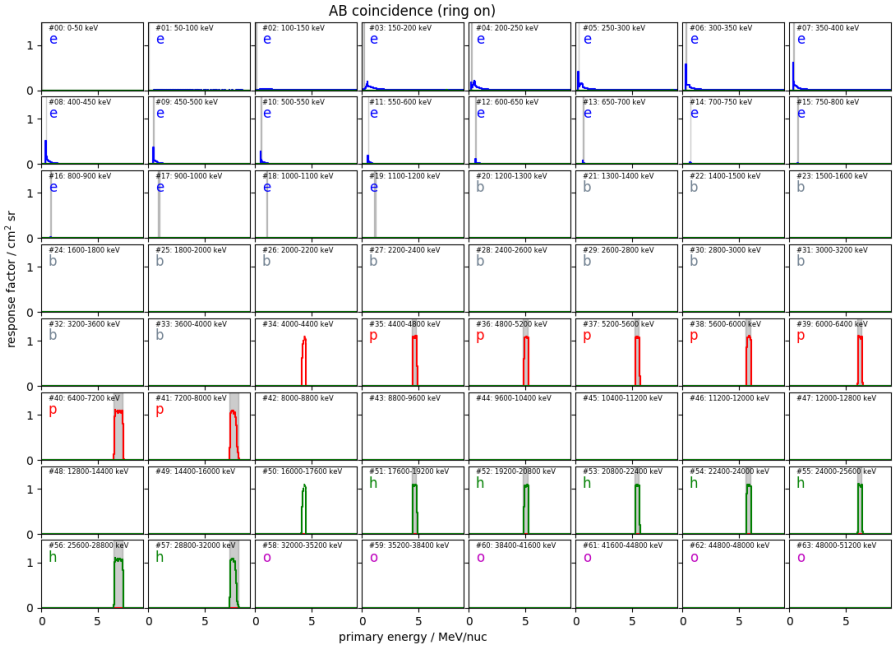

Figure 12: type response ab

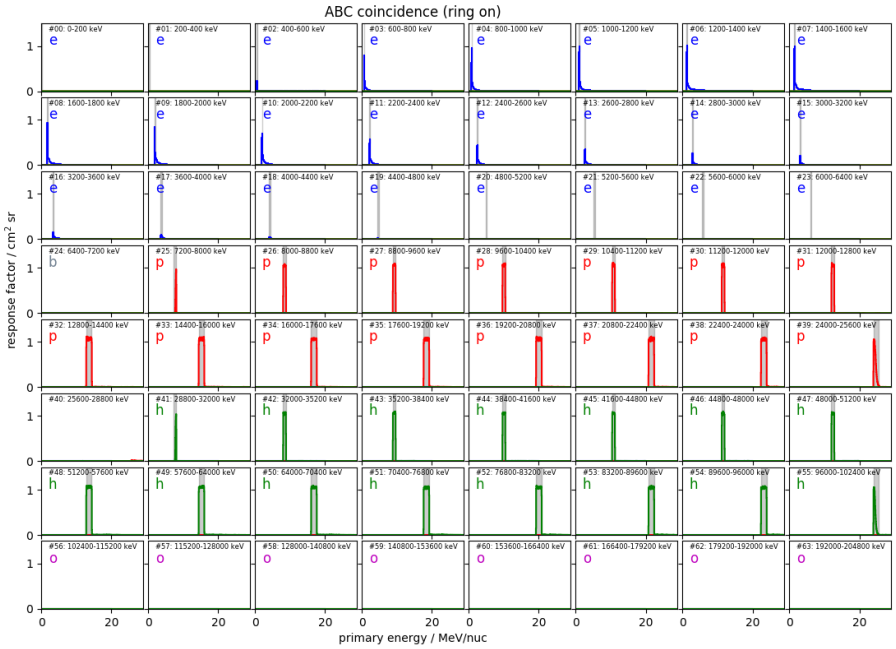

Figure 13: type response abc

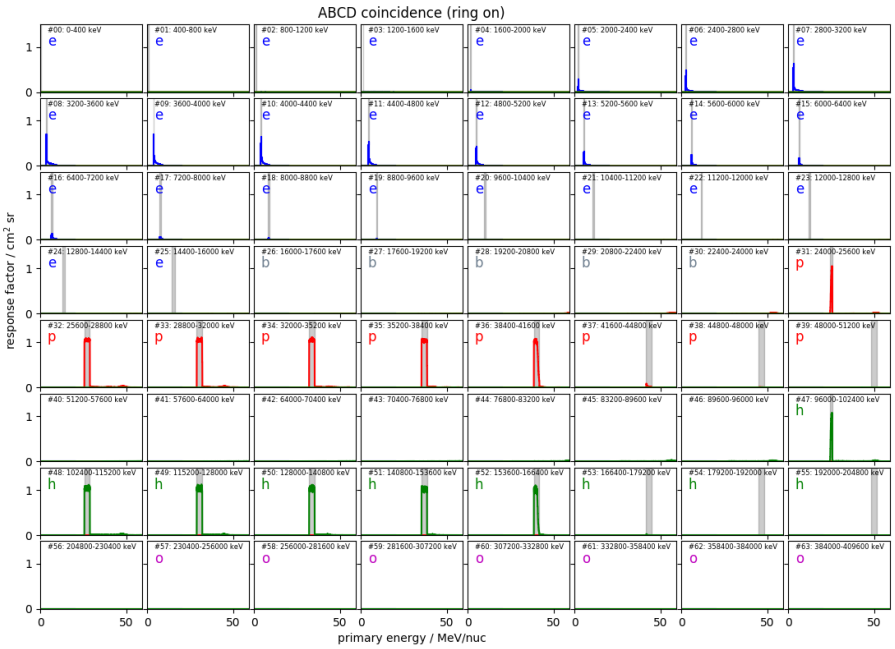

Figure 14: type response abcd

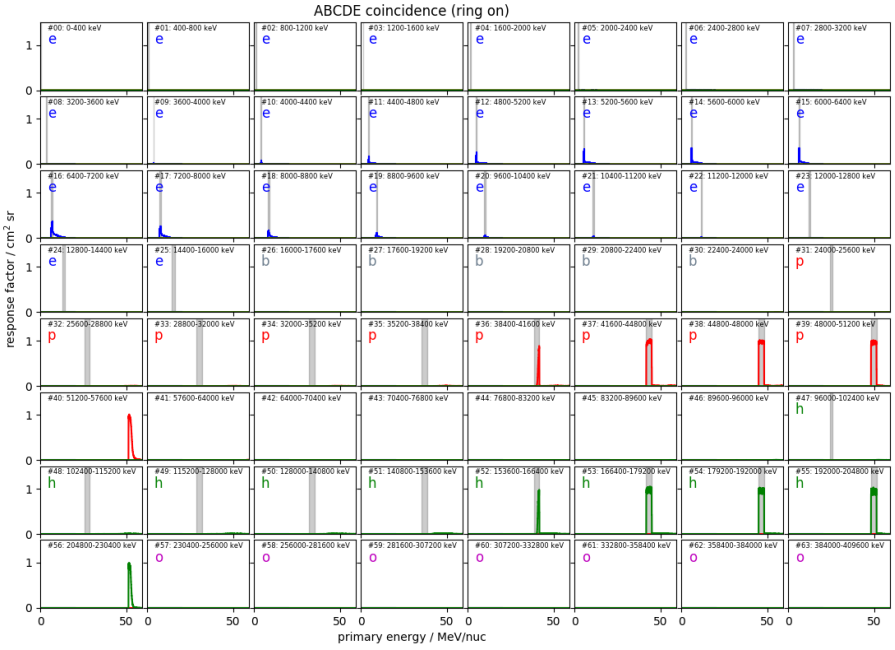

Figure 15: type response abcde

This can be further seen in figure 16 which shows the response for the channels electron, background, proton and helium (columns) each summed up for different penetration depths (rows). I.e. all responses for channels noted as "p" in figure 12 are summed up and shown in the first row, third column. The figure shows that the overall response to ions is constant over the nominal energy range (as expected). However, also so finite response (note the logarithmic scale) to particles of higher energies is noteable.

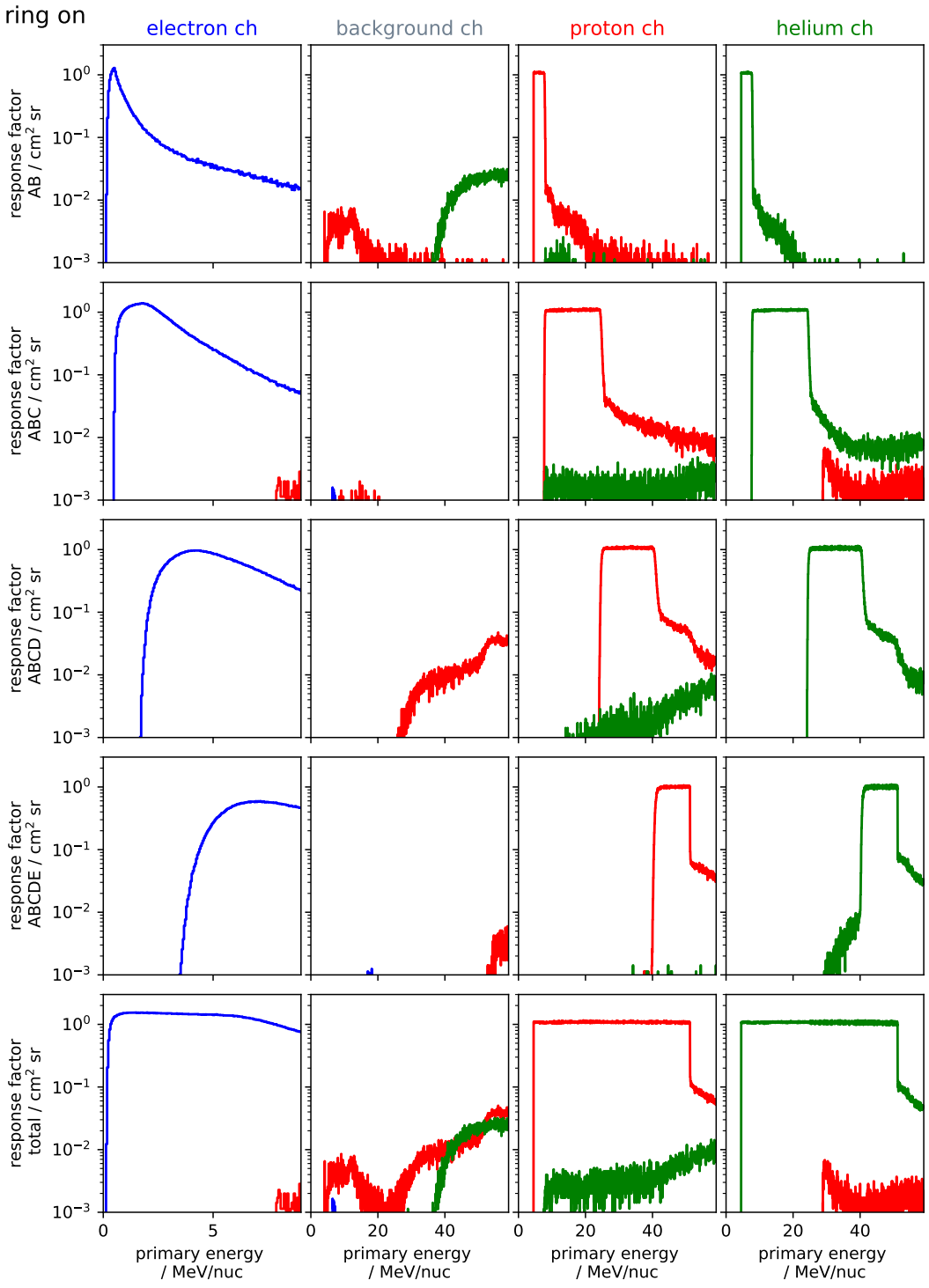

Figure 16: type response

5.2 Masterchannels

Since channels of the same energy are present in multiple penetration depths (i.e. a channel with 7.2-8 MeV mostly sensitive for protons exists in AB and ABC), these channels and their corresponding responses have to be merged in order to provide channels with constant responses. This can be seen in figure 17 where these combined channels (from now on called masterchannels) are presented for protons. Each panel shows the response as function of the simulated energy for individual histogram energy ranges (grey hatching as well as vertical text in each panel). The response to protons for all penetration depths for which a particular channel is noted as proton (cf. figure 12 to 15) are shown (see legend in each panel). The black lines correspond to the sum of the responses (in most channels its only one, its only necessary in overlap channels and in the identical ABCD and ABCDE channels. All masterchannels show a constant response. The channels at lower energies are sensitive to slightly higher energies than the nominal energy range of the histogram. This is due to the foils on top of the instrument in which particles (especially lower energy particles) deposit a fraction of their total energy. To account for that, the masterchannels will be labeled based on the actual response thresholds which are shown as magenta lines in the plots. They have been calculated as the energy at which the previous/following channel has a higher response. For the lowest and highest energy channel, the threshold has been defined as the energy at which the response is above 0.4 and 0.1 $cm^2\ sr$ for ringon and ringoff conditions, respectively. The response of a channel in $cm^2\ sr$ is than derived as mean value in this energy range. For the differential response $cm^2\ sr\ MeV$ this value is multiplied by the energy width of that channel (range between magenta lines in the plot). Both the response and differential response are shown in figure 19 for protons (row 1) and helium (row 2) individually for ringon (left) and ringoff conditions (right).
For the further processing and data production, the channel definitions and response factors are stored in a JSON files (Python dictionary) for further applications. The proton and helium JSONs are both given in section B.1.

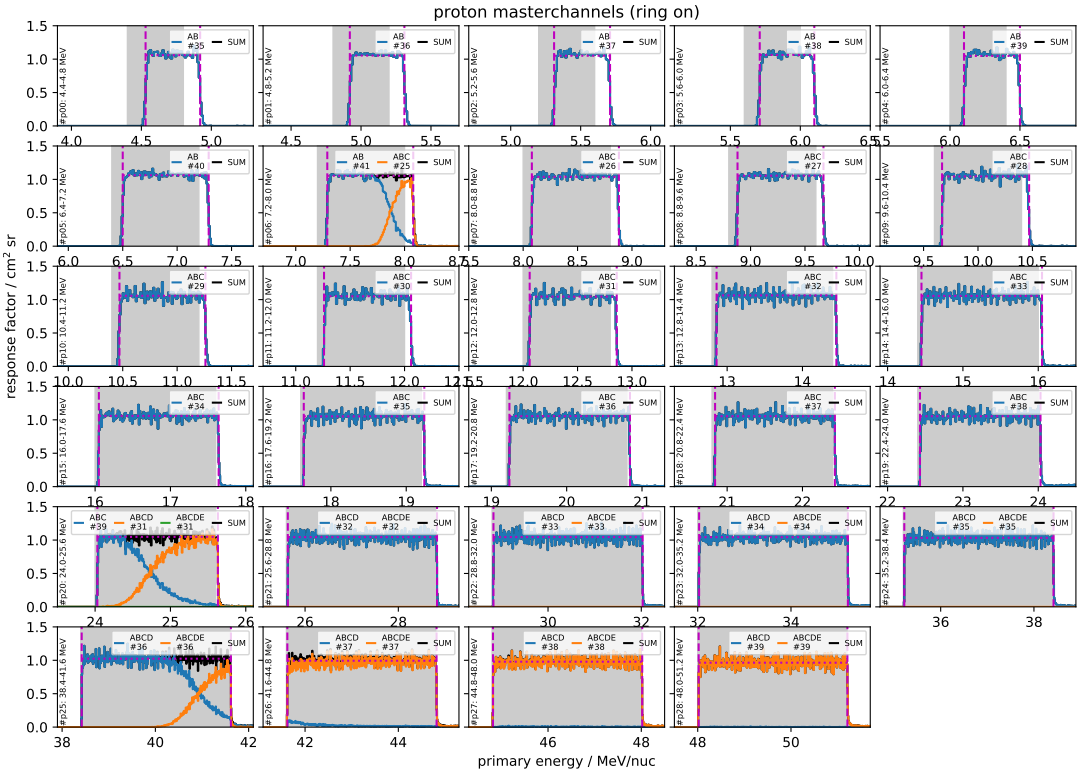

Figure 17: master proton response

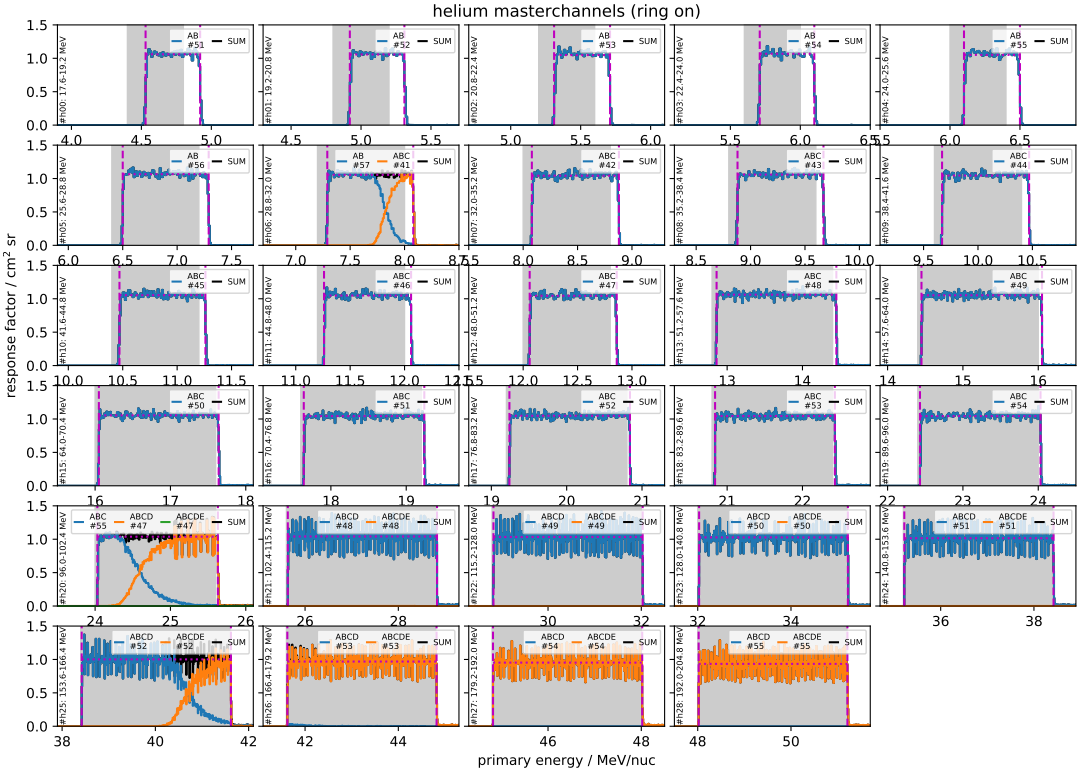

Figure 18: master helium response

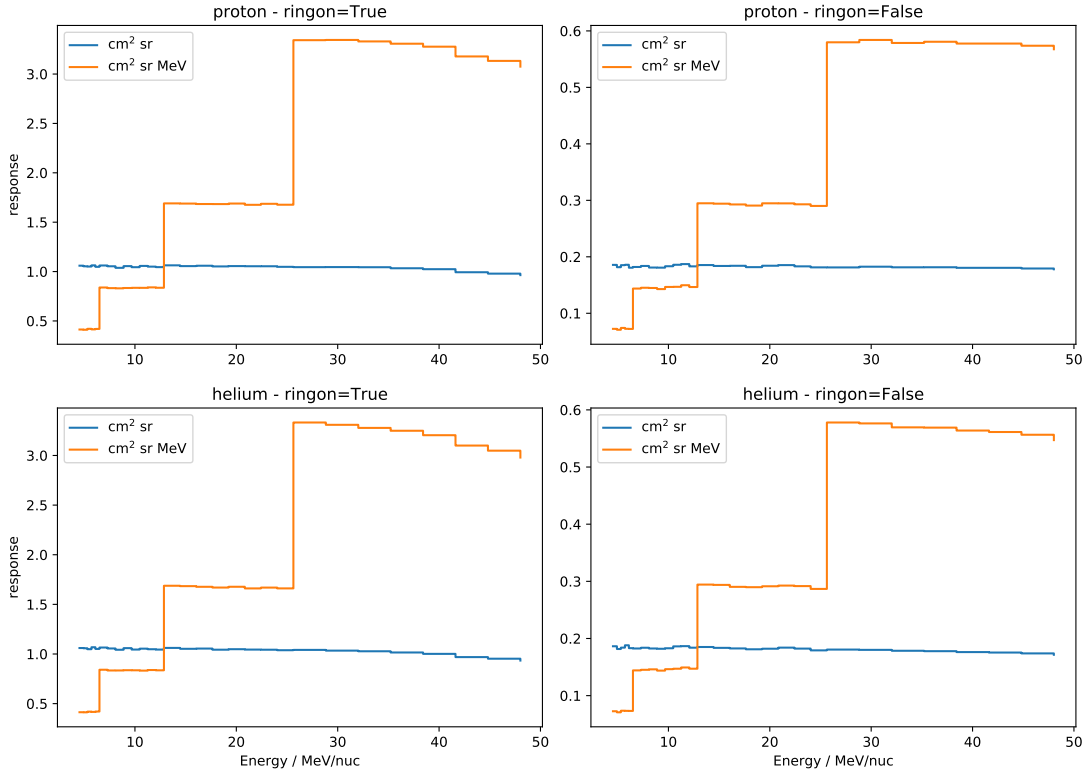

Figure 19: master response function

5.3 Validation based on simulation input spectra

Based on the simulation results, the overall systematic uncertainties and the contribution of higher enery particles (as discussed in sections 4 and 5 as well as A.3) can be analysed. For that purpose, the input spectra of protons and helium particles (figure 16 shows that electrons can be neglected) has been rescaled to power-law spectra with γ values from -5 to 1 and Helium to proton ratios of 0.01, 0.1 and 1. Figure 20 shows these rescaled input spectra as dashed grey lines (assuming a Helium to proton ratio of 1) in comparison to the derived proton and helium fluxes using the thresholds and responses given in sections 4 and 5 (also listed in the JSONS in B.1). While the colors represent the simulation scaling (power-law index), the circle and squares represent the calculated fluxes based on this artificial data for ringon condition for protons and helium particles, respectively. The check and asterix symbols reprenset the same for ringoff conditions. The figure as well as the relative deviations shown in the bottom panel show that the method is presenting reasonable results for all spectra and particle types with only the hardest spectra causing significant overestimates of the fluxes (which are also only seen at those channels with energies corresponding to a change of penetration depth, cf. section A.3).

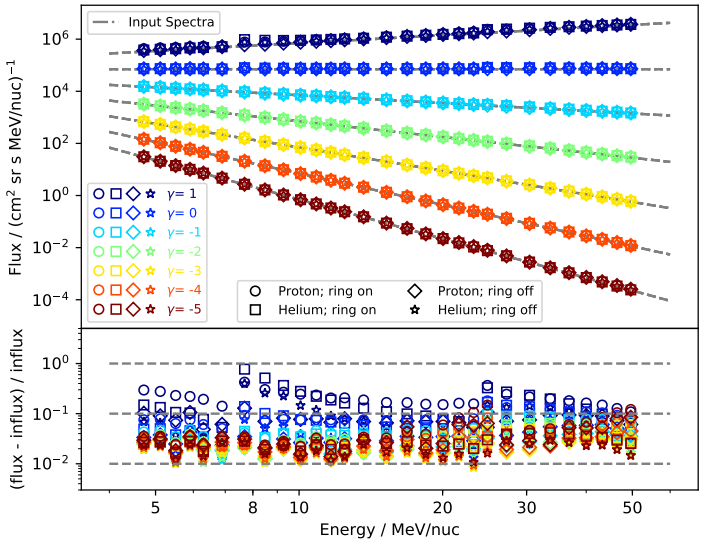

Figure 20: recalculated proton and helium spectra in comparison to the input spectra.

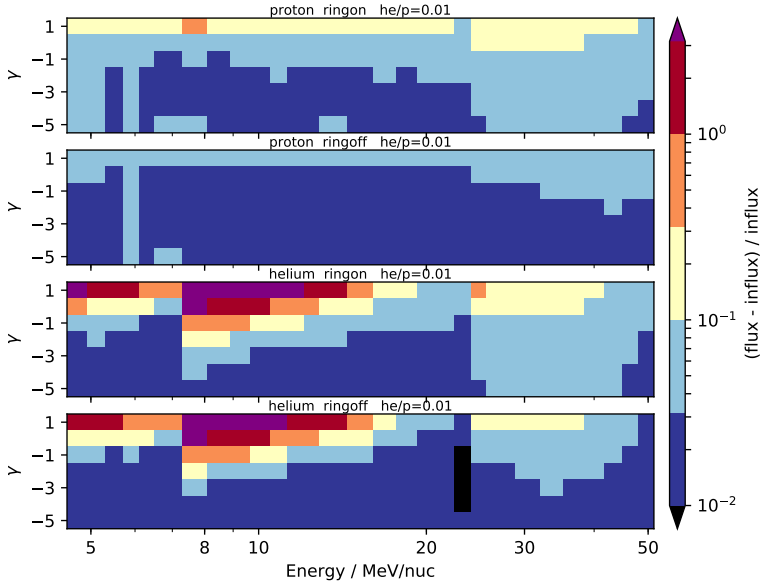

Figure 21: recalc spectra deviation: p/he=0.01

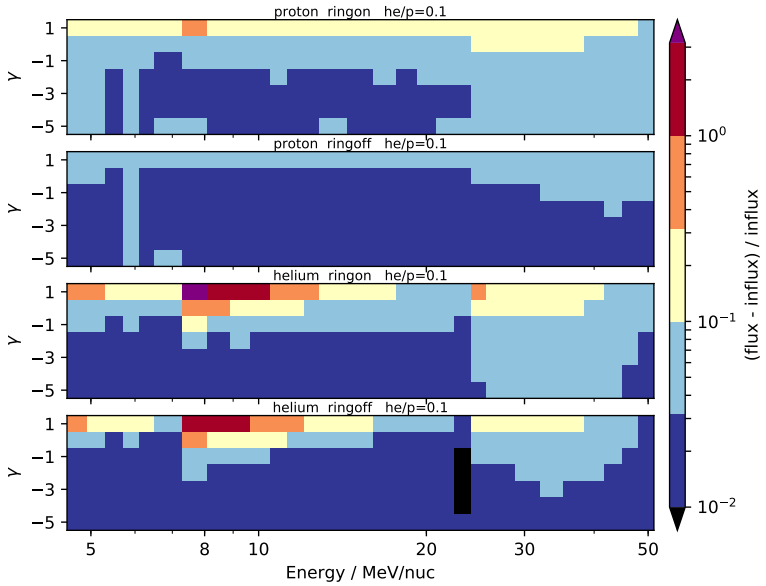

Figure 22: recalc spectra deviation: p/he=0.1

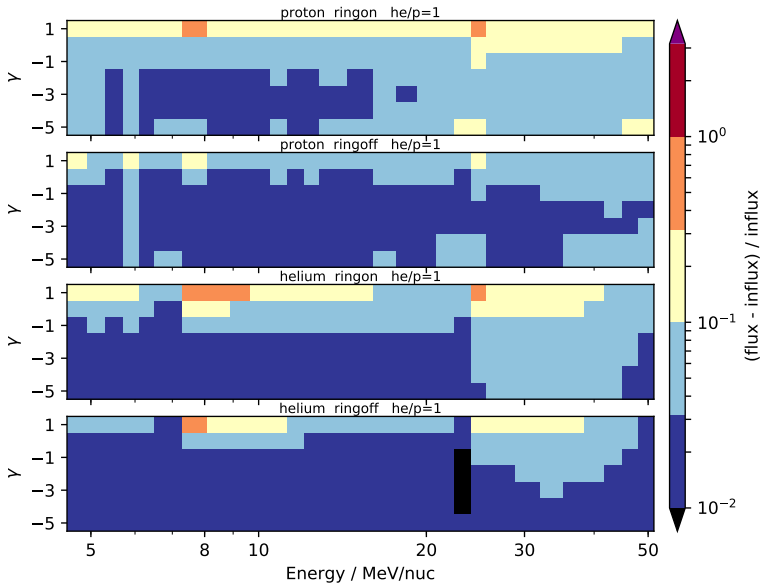

Figure 23: recalc spectra deviation: p/he=1

The scenario with a helium to proton ratio of 1, however, is far from being the norm. Hence, this analysis has been also performed for different ratios as presented in figures 21 to 23 for ratios of 0.01, 0.1 and 1. Here, the panels correspond to 1) proton ringon, 2) proton ringoff, 3) helium ringon, 4) helium ringoff. The colorbar presents the relative deviation (as shown in bottom panel of 20), while the y and x-axis correspond to the analysed spectral power-law index γ and the energychannel, respectively. From the figure it can be concluded:

- ringon proton:
  - contamination if γ >-1 due to particles missing deeper detectors
  - minor contamination if soft spectrum (γ <=-4) and he/p large (≈1) due to nominal helium barely reaching ABC with E<28.8 MeV being count as P barely stopping in ABC - or due to helium missing deeper detector
    - However: soft spectrum (γ <=-4) and he/p large (≈1) does not occour, see fig. 24

- ringon helium:
  - large contamination if γ >-1 due to particles missing deeper detectors
  - more corrupted if background (γ=+1) and he/p small due to nominal protons barely stopping in ABC with E>28.8 MeV being count as He (28.8 MeV/4. = 7.2 MeV)
    - However: background (γ=+1) and he/p small does not occour, see fig. 24
- ringoff proton:
  - always fine
- ringoff helium:
  - only corrupted if background (γ=+1) and he/p small due to nominal protons barely stopping in ABC with E>28.8 MeV being count as He (28.8 MeV/4. = 7.2 MeV)
    - However: background (γ=+1) and he/p small does not occour, see fig. 24

Based on that we can give the following verdict:

- ringoff proton channels always fine, ringon proton channels only valid if $\gamma < 0$!
- helium channels (especially ringoff) shows significant contribution of high energy protons, especially visible in hard spectra
- Word of caution I: Note that we used $\gamma_p = \gamma_{he}$ - an assumption that is not always valid in reality!
- Word of caution II: Note that the contribution of He3 was neglected (see section A.2)
- Word of caution III: Note that the failure mode E (FME) was neglected in simulation analysis, i.e. it is assumed that the energy loss in E is correctly accounted for. Hence, without any further analysis, data above 25 MeV/nuc should be taken with uttermost caution!

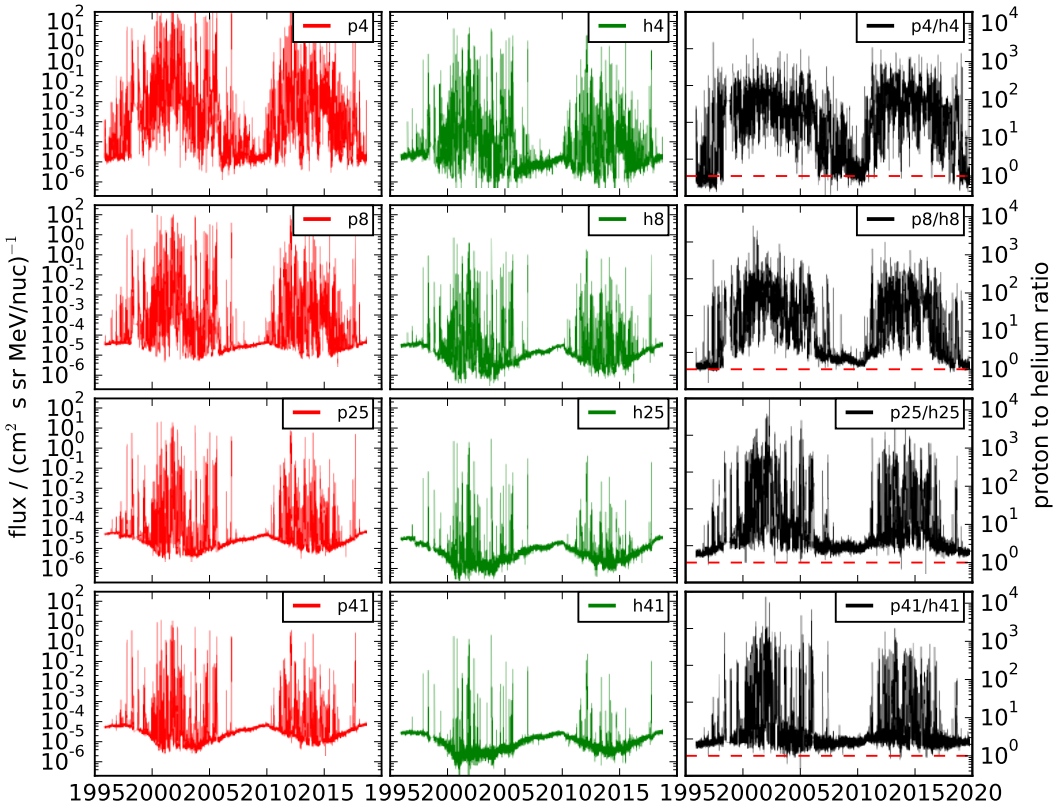

Figure 24: from Kuehl et al. 2019

6 Validation

6.1 Flux comparison with EPHIN lvl3

Figure 25 shows a direct comparison between the derived histogram fluxes and level 3 fluxes (cf. `http://ulysses.physik.uni-kiel.de/costep/level3/l3i/DOCUMENTATION-COSTEP-EPHIN-L3-20181002.pdf`) derived for identical energy channels and the same 8 minute intervals. The different rows correspond to several energy channels (4.72, 6.89, 8.48 and 20.04 MeV/nuc). The left column presents protons, the right one helium particles. While the data sets are in general in very good agreement, certain aspects discussed in section 5.3, such as the overestimation of helium fluxes during hard spectra (GCR background, low fluxes), can be already observed with this simple comparison.
For a more detailed comparison, figure 26 presents the relative deviation of the same data. Note that the level 3 fluxes are based on PHA data and hence have a significant higher statistical uncertainty than the histogram data explaining part of the broad distribution in the relative deviation.
The distributions of relative deviations are also shown in figure 27 as 1D-histograms. While symmetric distributions can be explained with statistics, the asymetric part in the higher energies can be explained by the FME, causing particles between 25 and 50 MeV/nuc to be registered at lower energies due to the fact that the energy deposition in E is not considered in the histogram. The fact that the center of the distribution is not at relative deviations of zero for certain energies has yet to be explained. This effect is also seen during ringoff conditions and the magnitude (shifts of up to 20%) can not be explained by the uncertainties of the response factors (i.e. fig. 23). Either differences between simulation and reality (e.g. uncorect foil thickness) or calibration uncertainties (e.g. the energy binning of the histogram differs from table 36) could explain such results. Some of these effects, however, would also be true for the level3 data. Furthermore, note that the level3 data was not designed/validated to feature such small energy channels either.
Figure 28 presents the deviations for all energy channels. The red hatched area indicates the crossing from AB to ABC channels, for which level 3 data could not be provided in identical channels (cf. level3 documentation). From the figure the overall agreement is once again confirmed while the fact that certain distributions are not centered at zero relative deviation is also clearly visible. Furthermore, this offset is changing from one channel to another. Besides the argument of systematic uncertainties in the response given above, it also has to be noted that the level 3 fluxes are based on a dE/dx method without direct measurement of the total energy and hence a broad non-rectangular response. In combination with changing bin widths (cf. figure 36 and grey dotted lines in the figure), i.e. at 6 MeV, this could cause uncertainties in the level3 flux calculation.

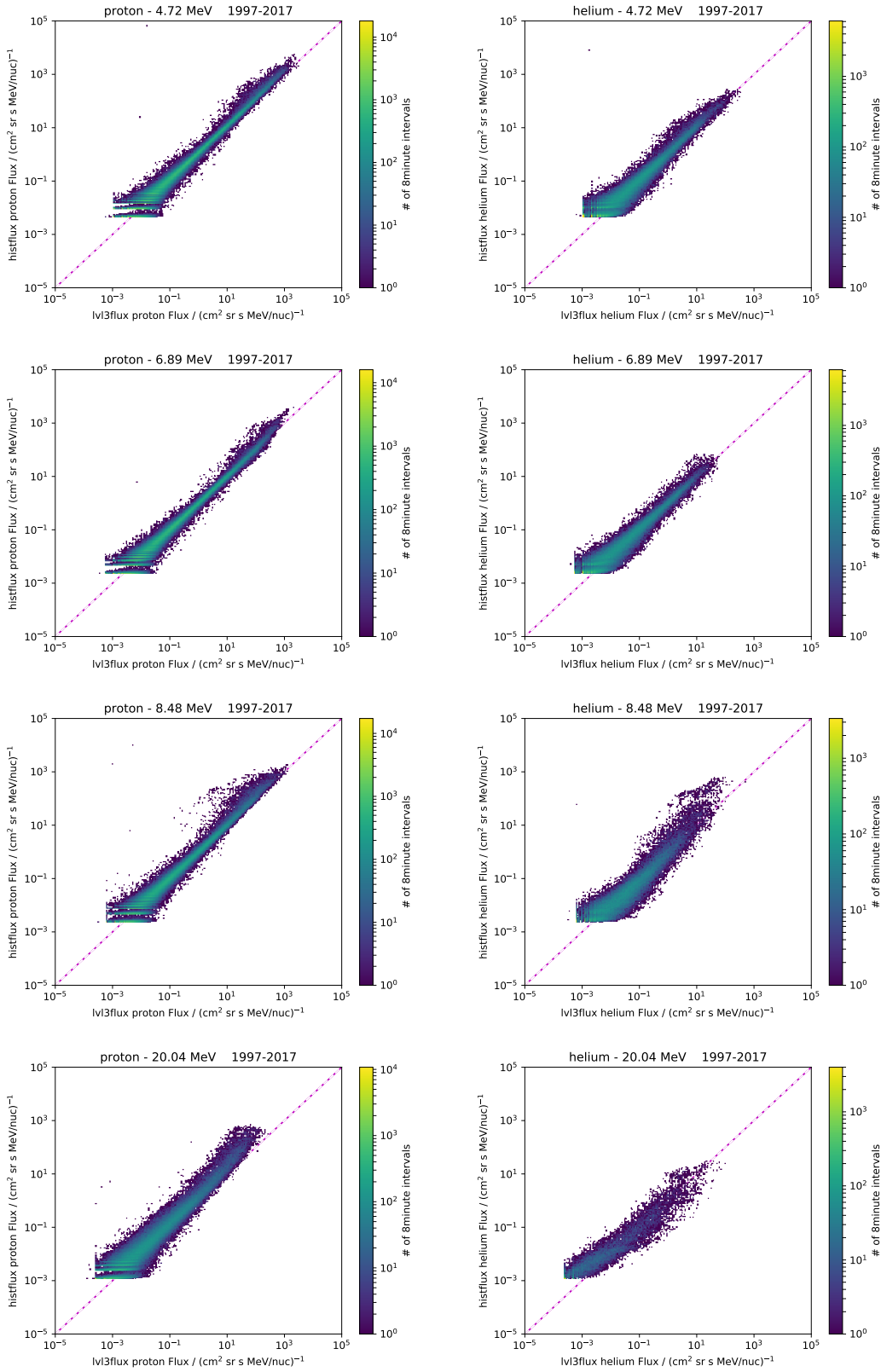

Figure 25: Comparison between EPHIN histogram fluxes and EPHIN lvl3 fluxes for some channels of protons (left) and helium particles (right)

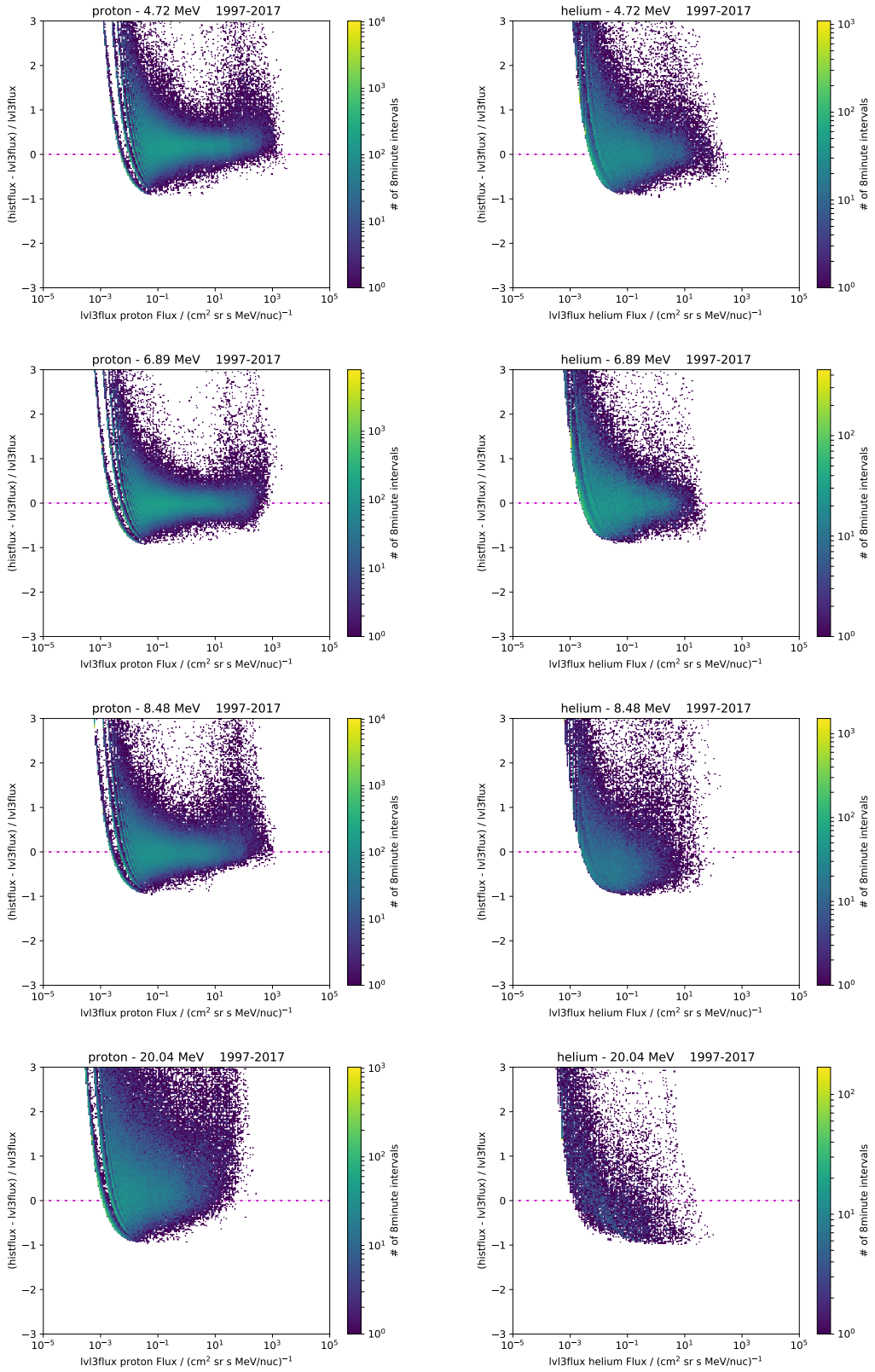

Figure 26: Relative comparison between EPHIN histogram fluxes and EPHIN lvl3 fluxes for some channels of protons (left) and helium particles (right)

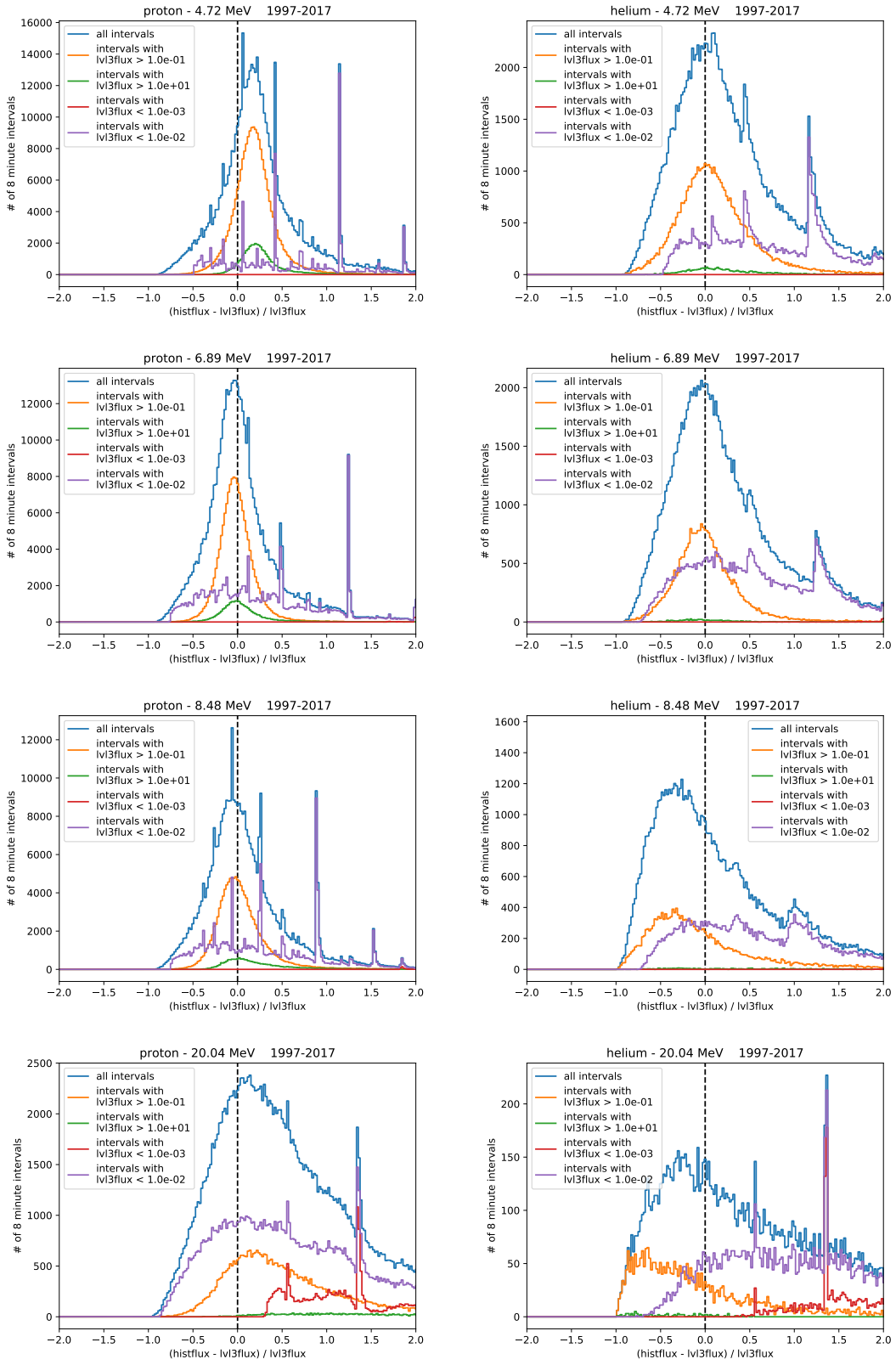

Figure 27: Relative comparison between EPHIN histogram fluxes and EPHIN lvl3 fluxes for some channels of protons (left) and helium particles (right)

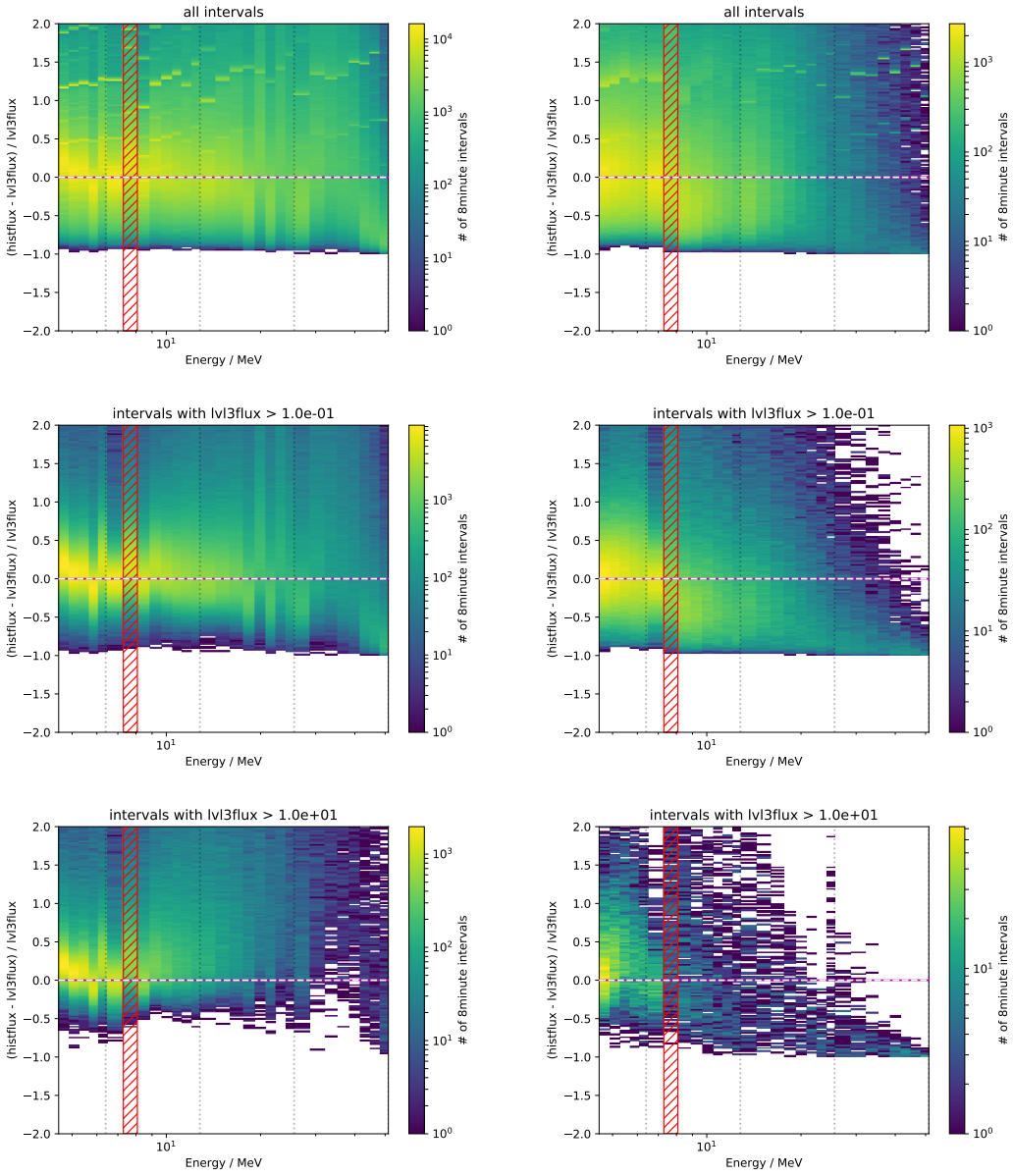

Figure 28: Deviations between EPHIN histogram fluxes and EPHIN lvl3 fluxes for all channels of protons (left) and helium particles (right) for all intervals (row 1) as well as intervals with lvl3flux above a threshold of 0.1 (row 2) and 10 cm^2 sr s MeV/nuc9 (row 3).

6.2 Comparison with other missions

Figure 29 shows monthly spectra for various different missions in comparison to the obtained histogram flux. While this can not give a complete picture, the months were selected to represent all different cases:

- most monthly spectra are in agreement between all data sets (examples: 06.1999, 04.2011, 01.2002)
- harder spectra cause overestimations of fluxes (especially for helium) as expected from section 5.3 (examples: 07.2000, 01.2005, 05.2007)
- two monthly spectra (from 1997 to 2017) obviously failed totally (03.2007, 11.2015 - this has to be investigated)
- in general: histogram fluxes above 25 MeV/nuc are doubtful (FME caused)

Note that 1) erne is often underestimating the flux due to deadtime issues and 2) goes is always a difficult instrument to compare to due to various instrumental issues.

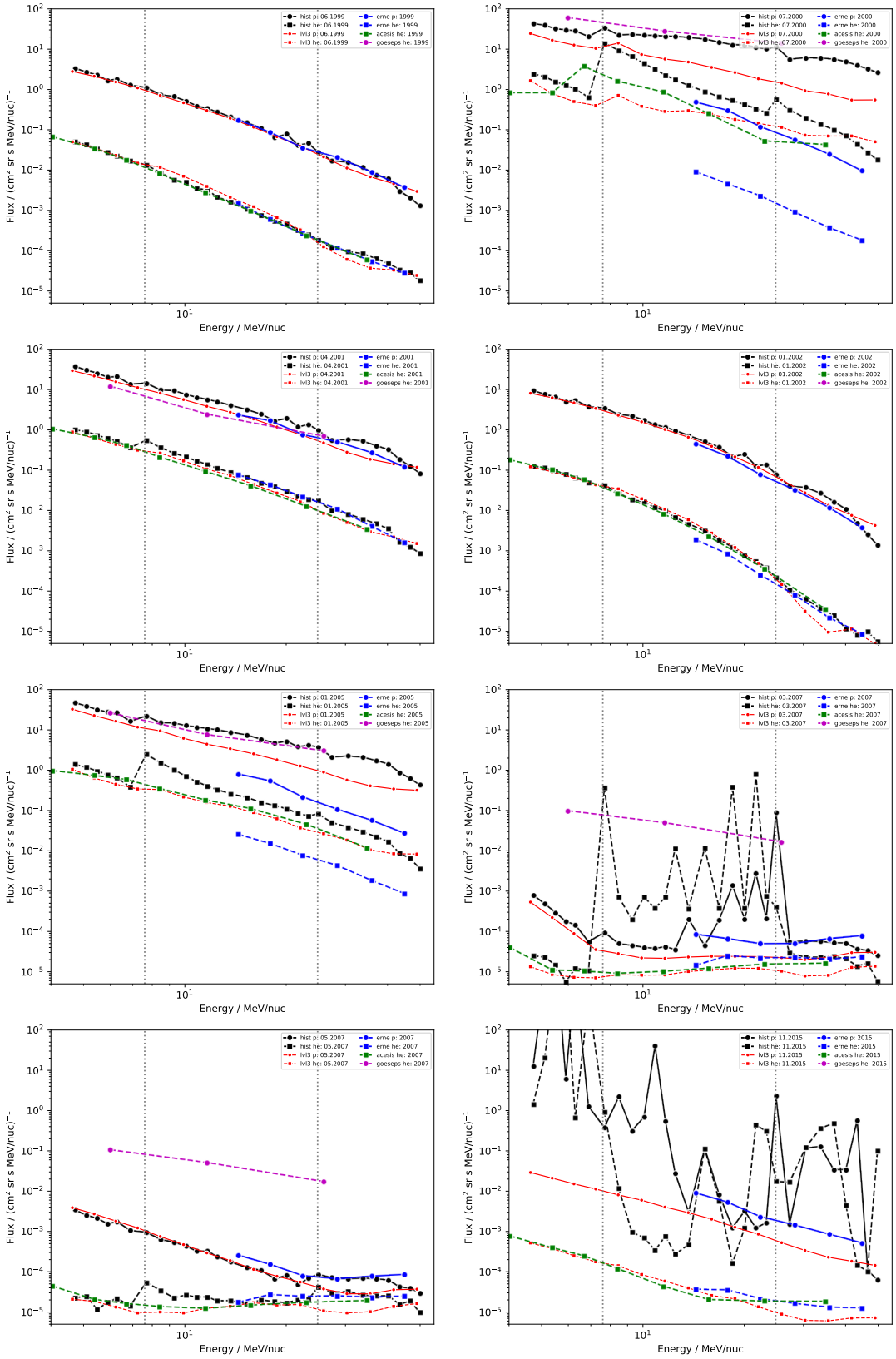

Figure 29: Monthly spectra for various missions in comparison to the histogram fluxes

7 Example plots

Figures 30 till 34 present different example plots for the application of the new data product. Note 1) the issues in figures 30, 33 and 34 due to hard spectra (cf. 5.3) but also 2) the nice velocity dispersion seen in figures 31 and 32.

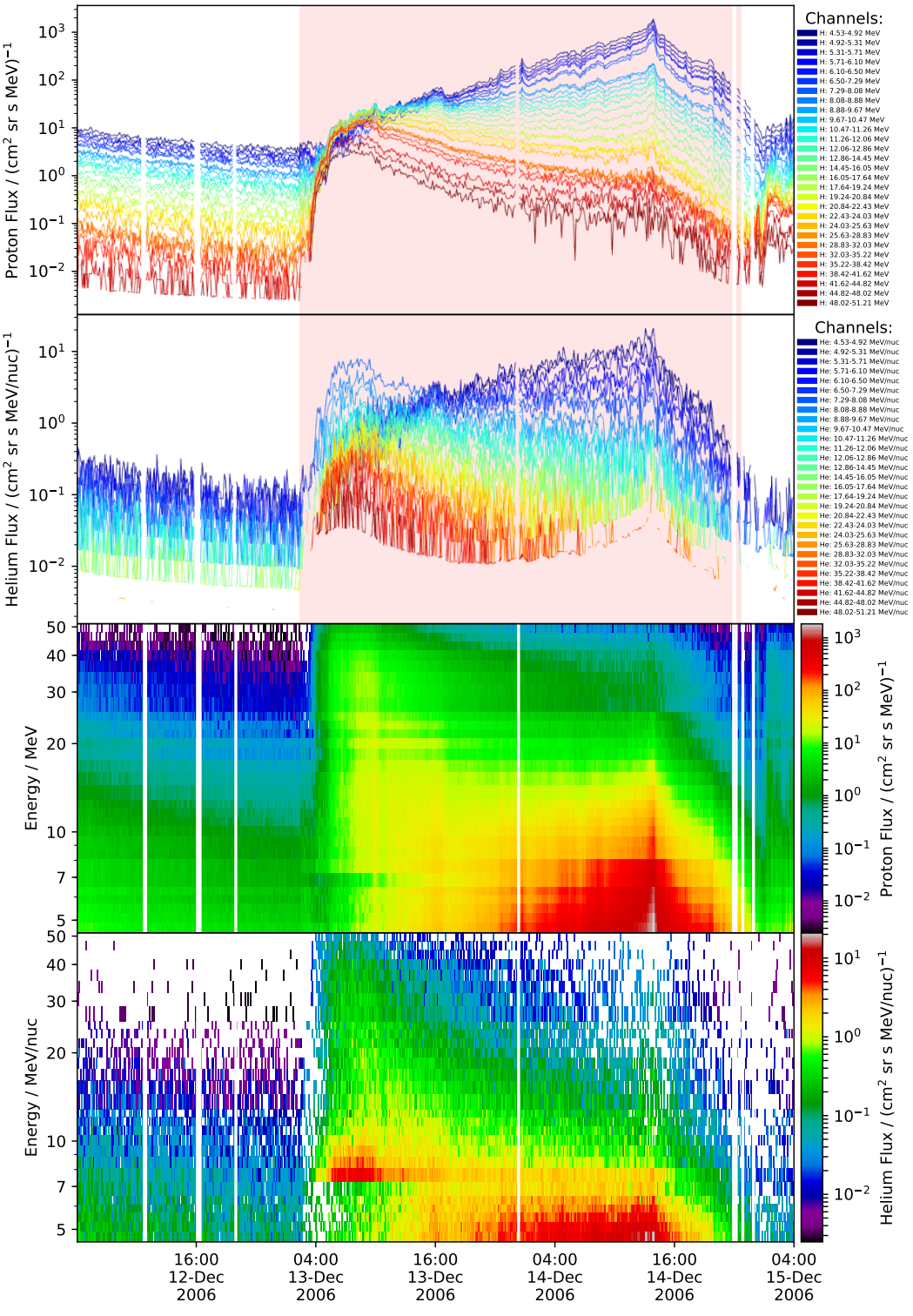

Figure 30: example plot 1

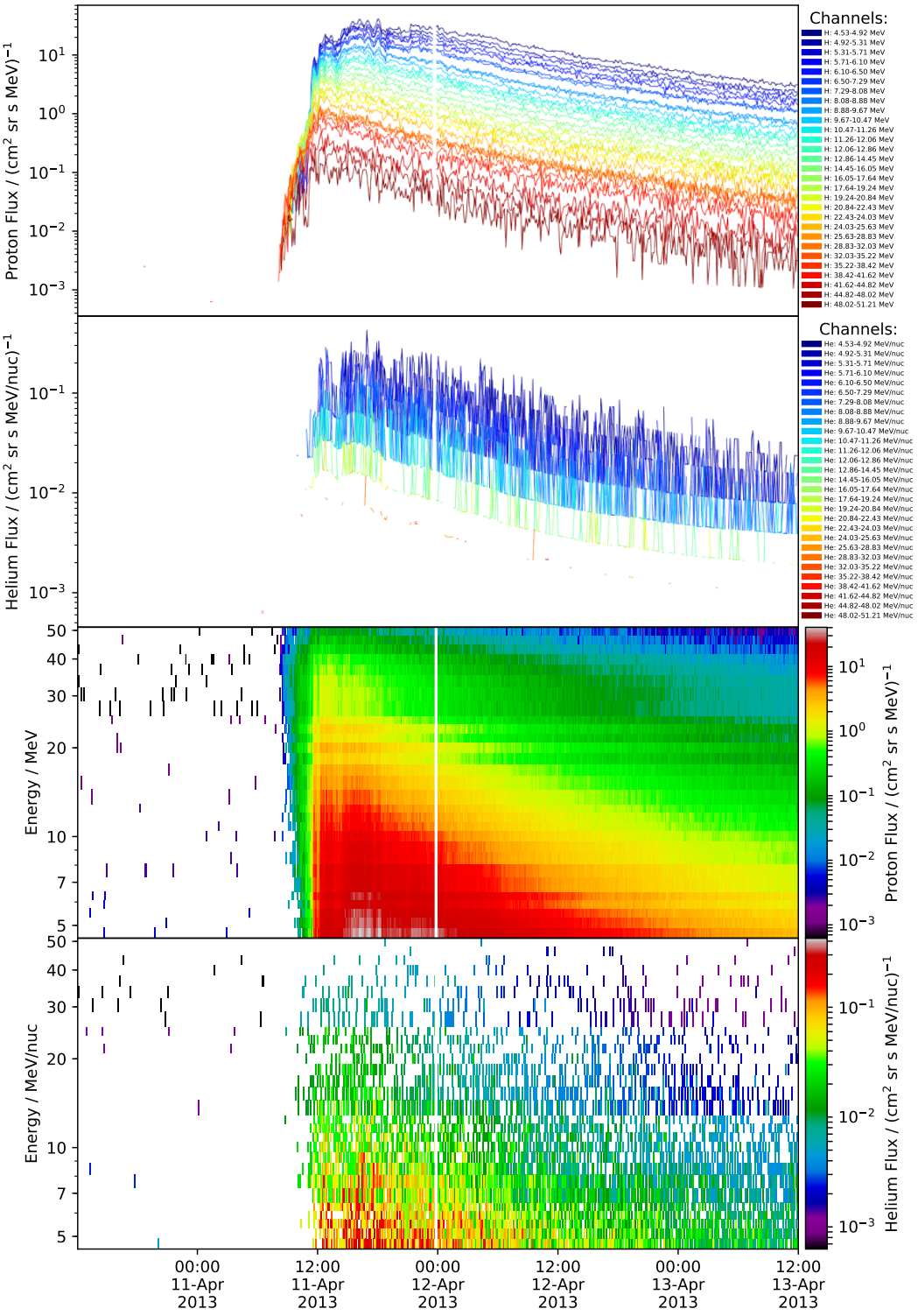

Figure 31: example plot 2

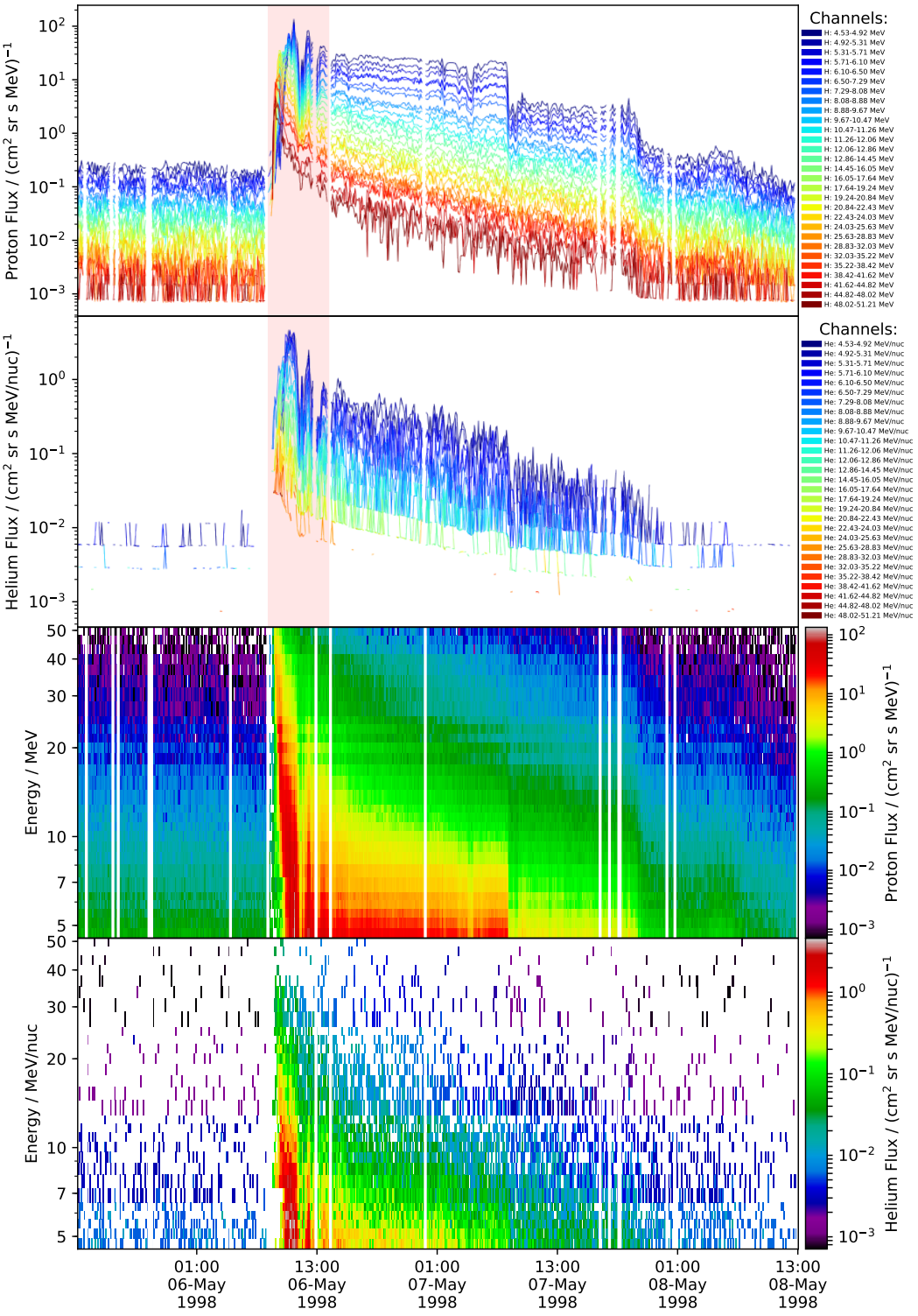

Figure 32: example plot 3

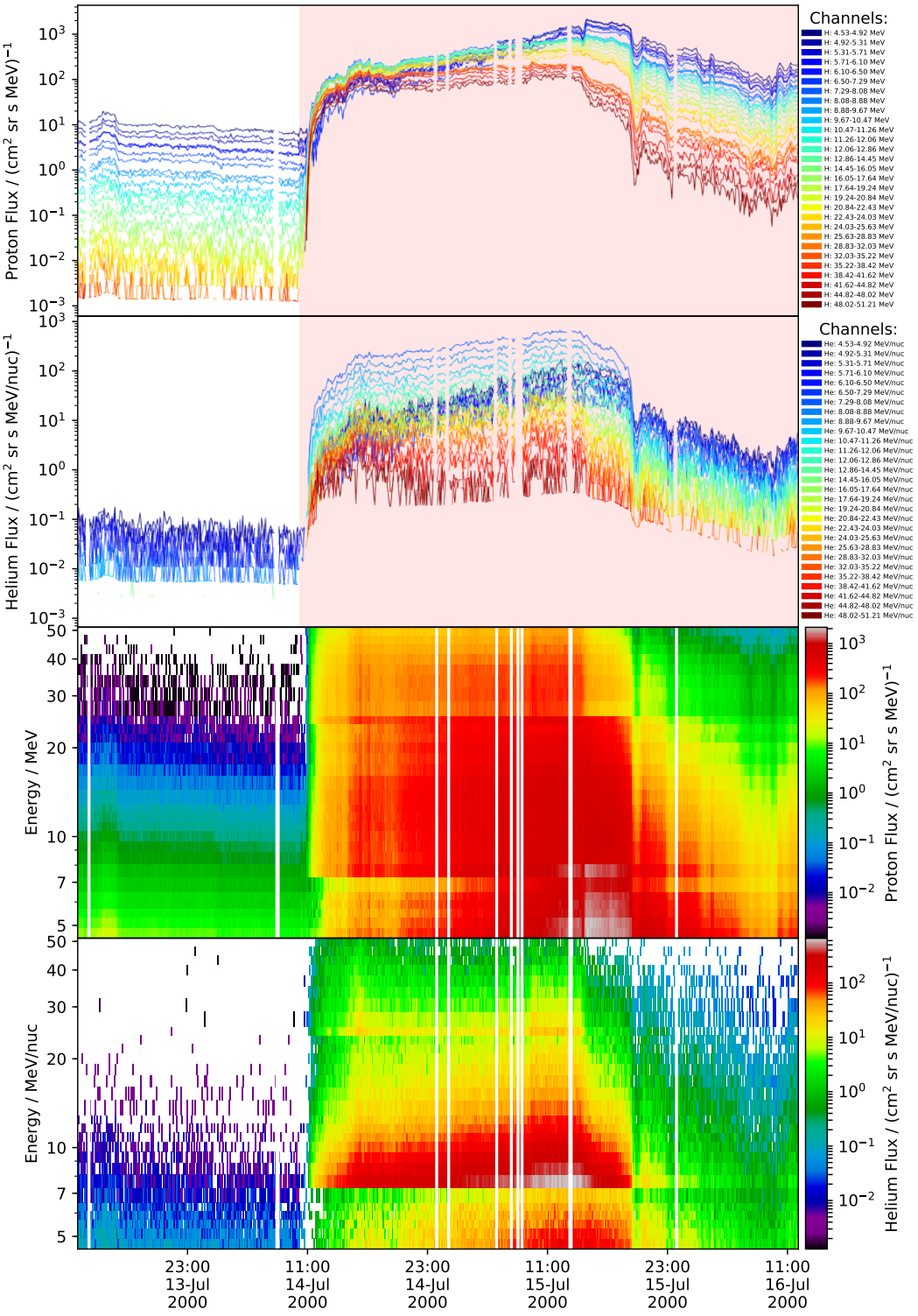

Figure 33: example plot 4

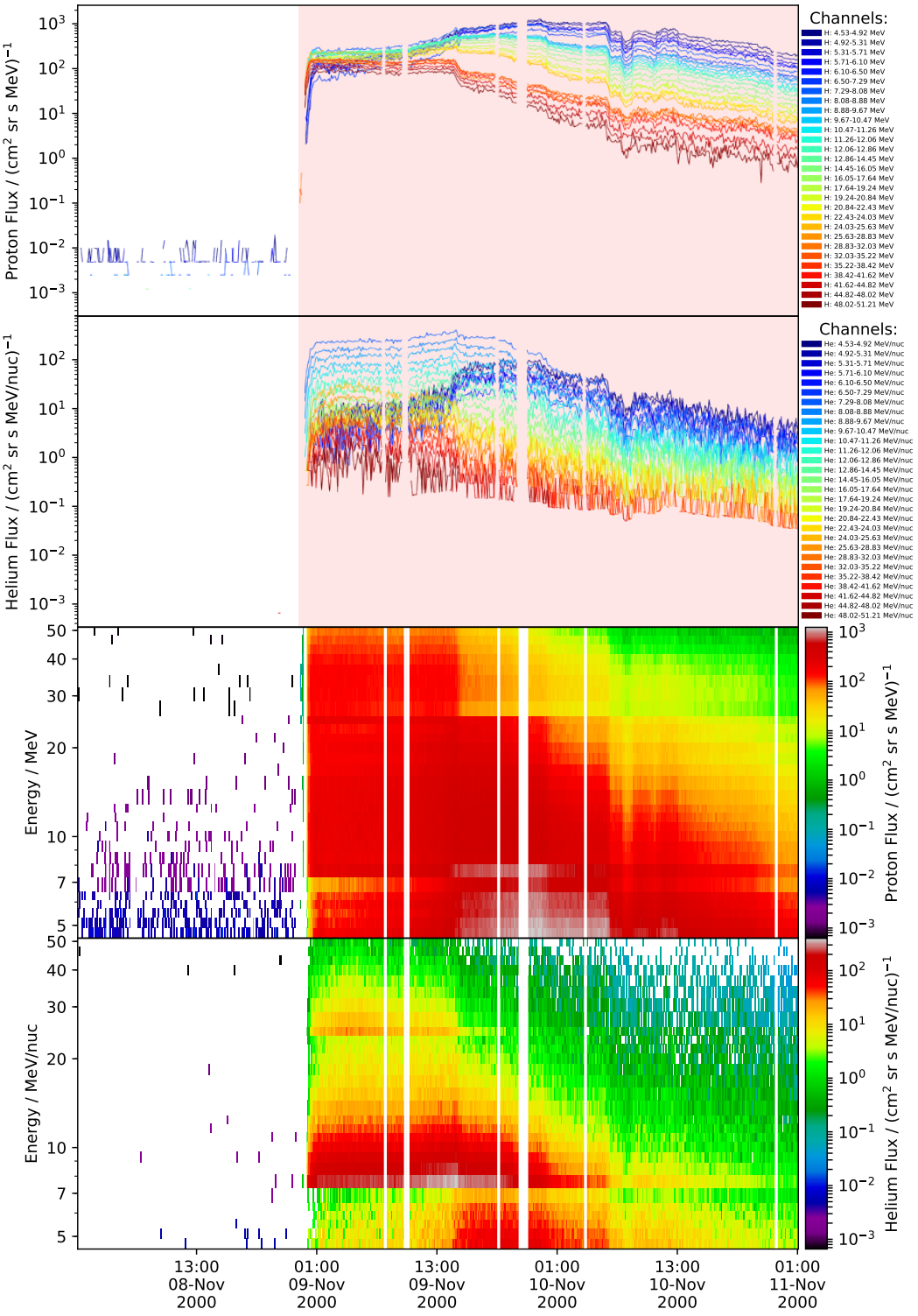

Figure 34: example plot 5

8 Data Product

The created histogram flux files will be provided as ASCII text files, separate files for protons and helium and for each individual year.
The format of the data product is given in table 2. The number of columns in the data product depends on the number of masterchannels (cp. section 5.2). For the nominal data product as defined in this channel the number of channels for both the proton and helium files is n=29.
Note that

- the particle type ('proton' or 'helium') is given in both, the filename as well as the header
- the time given in the data set marks the beginning of a 8 minute time interval
- the number of counts columns can be used to determine statistical uncertainties (sigma=flux/sqrt(N))
- the ringbit column will be "1" if the outer segments of detectors A and B are turned off (high flux mode) or "0" for during normal observations
- the coinccounter scaler presents the applied scaling of the histogram counts based on the total coincidence count rates (compare section 3.3). This is included for additional information only (i.e. the user should not (!) use is for additional scaling)
- the 'type' column in the table describes the format of the data product with 'int', '4.4f' and '4.4e' refering to integer, float and scientific (float and exponent), respectively

The total size of the entire data set from 1997-08-22 (instrument patch sets the histogram to parallel only) till 2017-10-05 (instrument set to failure mode D) amounts to roughly 1 GByte with the calculation and production of these data sets taking about 30 minutes.

item	label	data content	units	type
1	year	year	years	int
2	month	month	months	int
3	day	day	days	int
4	doy	day of year	days of year	int
5	hour	hour	hours	int
6	minute	minute	minutes	int
7	ringbit	ringbit	binary status word	int
8	coincidence counter scaler	applied scaling factor	scaling factor	4.4f
9	flux_ch1	channel 1 flux	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
:	:	:	:	:
8+n	flux_chn	channel n flux	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
9+n	counts_ch1	channel 1 number of counts	number of counts	int
:	:	:	:	:
8+2*n.	counts_chn	channel n number of counts	number of counts	int

Table 2: Explanation of the data product of histogram fluxes.

9 Code

All functions to create histogram flux files are defined in the file 'ephin_histograms_funcs.py'. The entire code can be found in section B.2. In the following, an explanation of all defined functions is given:

year_doy_iterator returns the year and day one day afer input date

time_from_msec returns hours and minutes from input millisec

timediff returns timedelta between two lines

write_header writes header to output file

load_level1_sci load level1 sci files

load_and_merge_consecutive_scis merges two consecutive sci files

create_ringbit_array extracts ringbit information from data array

create_hist_array extracts histograms from data array

create_histbit_array extracts histbit information from data array

create_total_coinc_counter_array extracts total sum of coincidence counters from data array

create_all_the_arrays executes all data extractions listed above

load_masterchannel_response_dict loads the histogram response JSON

calc_coinc_counter_per_min_during_hist calculates coincidence counters during hist integration interval

calc_coinc_deadtime_correction calculates counter scaling during ringon (compare section 3.3)

calc_coinc_counter_ratio calculates counter scaling during ringoff (compare section 3.3)

extract_and_unfold_hists unfold histograms from hist array

calc_and_save_hist calculates and saves fluxes for a given histogram

create_histogram_flux_files main function incorporating the above ones in order to create histogram flux files

Furthermore, the code requires a set of paths (input and output) and parameter that have to be defined in a given executable script. A working example that includes these is given in B.3. The paths and parameter are as follows:

output_path Output path (Note: this folder will be purged first when executed!)

sci_path Path to the EPHIN Level1 .sci files (e.g. 'data/missions/soho/costep/level1/sci/')

response_dict_path Path to the histogram response JSONs (i.e. those given in sec. B.1)

timeconstant_coinc_counter Time constant for counter scaling (compare section 3.3). Should be set to 0.00747.

time_accumulation Instrument accumulation time. Should be set to 59.953.

A Complementary documentation

A.1 Previous documentations

Some limited documentation of the histogram data was previously created.
The level1 spec shows the column definition of the lvl1 *.sci files that include the histogram data as shown in figure 35. The entire level1 documentation is available here: `ulysses.physik.uni-kiel.de/costep/doc/L1-EPHINspec.doc`.
The energy ranges of the different histogram channels were given in a table in the appendix of the PhD thesis of Holger Sierks 1997 which is shown in figure 36.
Some technical information was summed up in a document by Reinhold Müller-Mellin in May, 2019, given in figure 37.

```
File formats
------------
1. Counter & Histogram File: COSyydoy.SCI


Time resolution 1 min, one L-1 data set consists of 101 elements:

Time tags
-------------------------------------------------------------------------------
                PB5-Time:       integer :       3 elements
                                year
                                day of year
                                ms of day

          min95:                float     :     minutes since launch 2 dec'95 +
                                                2 digits decim. part of min

                SC_EPOCH:  float     :       ms since year 1

Science DATA          integer  :        88 elements
------------

--SINGLE DETECTOR COUNTER----------------------------------------
         1        '_G0:'
         2 : 7            'A00:','A01:','A02:','A03:','A04:','A05:'
         8 :13   'B00:','B01:','B02:','B03:','B04:','B05:'
        14 :17            '_C0:','_D0:','_E0:','_F0:'
--COINCIDENCE COUNTER--------------------------------------------
        18 :20            'P4GM:','P4GR:','P4S:'
        21 :23            'P8GM:','P8GR:','P8S:'
        24 :27            'H4GM:','H4GR:','H4S1:','H4S23:'
        28 :31            'H8GM:','H8GR:','H8S1:','H8S23:'
        32 :35            'E150:','E300:','E1300:','E3000:'
        36       'INT:'
        37 :39            'P25GM:','P25GR:','P25S:'
        40 :42            'P41GM:','P41GR:','P41S:'
        43 :46            'H25GM:','H25GR:','H25S1:','H25S23:'
        47 :50            'H41GM:, H41GR:','H41S1:','H41S23:'
        51 :56            ,'CT0:','CT1:','CT2:','CT3:','CT4:','CT5:'
--HISTOGRAM Channel----------------------------------------------
        57 :88            ' Bin1'...' Bin32'


--EPHIN Instrument Status info-----------------------------------
        integer : 8 elements
            1:5 first 5 EPHIN status bytes
              6  LION data set lenght
```

Figure 35: Level 1 documentation.

			HIST1		HIST2		HIST3 & HIST4		
Chn	Exp	Mant	Σ dE/dx [keV]		Σ dE/dx [keV]		Σ dE/dx [keV]		
1	000	000	– 50	e	– 200	e	– 400	e	e
2	000	001	– 100	e	– 400	e	– 800	e	e
3	000	010	– 150	e	– 600	e	– 1200	e	e
4	000	011	– 200	e	– 800	e	– 1600	e	e
5	000	100	– 250	e	– 1000	e	– 2000	e	e
6	000	101	– 300	e	– 1200	e	– 2400	e	e
7	000	110	– 350	e	– 1400	e	– 2800	e	e
8	000	111	– 400	e	– 1600	e	– 3200	e	e
9	001	000	– 450	e	– 1800	e	– 3600	e	e
10	001	001	– 500	e	– 2000	e	– 4000	e	e
11	001	010	– 550	e	– 2200	e	– 4400	e	e
12	001	011	– 600	e	– 2400	e	– 4800	e	e
13	001	100	– 650	e	– 2600	e	– 5200	e	e
14	001	101	– 700	e	– 2800	e	– 5600	e	e
15	001	110	– 750	e	– 3000	e	– 6000	e	e
16	001	111	– 800	e	– 3200	e	– 6400	e	e
17	010	000	– 900	e	– 3600	e	– 7200	e	e
18	010	001	– 1000	e	– 4000	e	– 8000	e	e
19	010	010	– 1100	e	– 4400	e	– 8800	e	e
20	010	011	– 1200	e	– 4800	e	– 9600	e	e
21	010	100	– 1300	e	– 5200	e	– 10400	e	e
22	010	101	– 1400	e	– 5600	e	– 11200	e	e
23	010	110	– 1500	e	– 6000	e	– 12000	e	e
24	010	111	– 1600	e	– 6400	e	– 12800	e	e
25	011	000	– 1800	e	– 7200	e	– 14400	e	e
26	011	001	– 2000	e	– 8000	p	– 16000	e	e
27	011	010	– 2200	e	– 8800	p	– 17600	e	e
28	011	011	– 2400	e	– 9600	p	– 19200	e	e
29	011	100	– 2600	e	– 10400	p	– 20800	e	e
30	011	101	– 2800	e	– 11200	p	– 22400	e	e
31	011	110	– 3000	e	– 12000	p	– 24000	e	e
32	011	111	– 3200	e	– 12800	p	– 25600	p	e
33	100	000	– 3600	e	– 14400	p	– 28800	p	e
34	100	001	– 4000		– 16000	p	– 32000	p	e
35	100	010	– 4400	p	– 17600	p	– 35200	p	e
36	100	011	– 4800	p	– 19200	p	– 38400	p	e
37	100	100	– 5200	p	– 20800	p	– 41600	p	p
38	100	101	– 5600	p	– 22400	p	– 44800	p	p
39	100	110	– 6000	p	– 24000	p	– 48000	p	p
40	100	111	– 6400	p	– 25600	p	– 51200	p	p
41	101	000	– 7200	p	– 28800	p	– 57600	p	p
42	101	001	– 8000	p	– 32000	he	– 64000	p	p
43	101	010	– 8800	p	– 35200	he	– 70400	p	p
44	101	011	– 9600	p	– 38400	he	– 76800		p
45	101	100	– 10400	p	– 41600	he	– 83200		p
46	101	101	– 11200	p	– 44800	he	– 89600	he	p
47	101	110	– 12000	p	– 48000	he	– 96000	he	
48	101	111	– 12800	p	– 51200	he	– 102400	he	
49	110	000	– 14400		– 57600	he	– 115200	he	
50	110	001	– 16000	he	– 64000	he	– 128000	he	
51	110	010	– 17600	he	– 70400	he	– 140800	he	
52	110	011	– 19200	he	– 76800	he	– 153600	he	he
53	110	100	– 20800	he	– 83200	he	– 166400	he	he
54	110	101	– 22400	he	– 89600	he	– 179200	he	he
55	110	110	– 24000	he	– 96000	he	– 192000	he	he
56	110	111	– 25600	he	– 102400	he	– 204800	he	he
57	111	000	– 28800	he	– 115200	he	– 230400	he	he
58	111	001	– 32000	he	– 128000	he	– 256000	he	he
59	111	010	– 35200	he	– 140800	he	– 281600		he
60	111	011	– 38400	he	– 153600	he	– 307200		he
61	111	100	– 41600	he	– 166400	he	– 332800		he
62	111	101	– 44800	he	– 179200		– 358400		
63	111	110	– 48000	he	– 192000		– 384000		
64	111	111	– 51200	he	– 204800		– 409600		

e = electron, p = proton, he = helium

Tabelle E.1: Die Zuordnung der Teilchenpopulationen und Energien zur Kanalnummer (*Chn*) der On-Board Histogramme. Mit einer Akkumulationsperiode von 8 Minuten werden in Echtzeit aus den Pulshöhendaten vier Histogramme in Abhängigkeit von der Eindringtiefe der Teilchen erstellt (HIST1: A B, HIST2: A B C, HIST3: A B C D, HIST4: A B C D E).

Figure 36: Calibration table from PhD thesis from Holger Sierks, 1997.

RM² 13-MAY-2019

EPHIN/SOHO Histogram Specification

EPHIN onboard electronics can process pulse height information from incident particle at a higher rate than can be transmitted to ground due to limited telemetry capacity. Therefore, EPHIN generates onboard histograms which provide more detailed spectral information than the counting rates though less detailed than the full PHA information:

- The statistics is greatly increased, however, the energy resolution is reduced such that no isotope information can be obtained. This results from the method of energy determination and histogram generation:
  - conversion of the 10-bit ADC result of each detector into keV using look-up tables
  - summation of 2, 3, 4 or 5 results to a 14-bit total energy value
  - reduction of the total energy to 10 significant bits
  - compression of the 10-bit total energy to 3-bit exponent + (hidden leading 1) + 3-bit mantissa
  - incrementing a 64-channel histogram
- The ability to discriminate particles is reduced, as particle signatures from electrons, protons, and helium nuclei may overlap in certain energy ranges. This results from the histogram generation process, which is solely controlled by the range of the particle in the detector stack:
  - histogram 0: detectors AB
  - histogram 1: detectors ABC
  - histogram 2: detectors ABCD
  - histogram 3: detectors ABCDE
- The time resolution is 8 minutes. In the EPHIN counting rate files, there are 32 channels per minute, i.e. half a histogram. See Specification for Level-1 *.SCI files:
  - Bin1 … Bin 32 words 57 : 88 (or words 61 : 92 when including header words)
  - Use the 3-bit minute counter (= frame counter in EPHIN status word) to address the 4 histograms

Figure 37: Histogram specifications given by Reinhold Müller-Mellin in May, 2019.

A.2 Helium3

The histogram data does not allow for isoptopic identification, e.g. the separation between ^3He and ^4He. However, since certain solar energetic particle events can feature high $^3He/^4He$ ratios, here we show the expected energy deposition of ^3He and hence contributions to the histograms. Figure 38 shows in addition to the electron, proton and ^4He contributions (cp. section 4) the histogram contributions of ^3He for the different coincidences/penetration depths. From the figure, it can be concluded that ^3He populates the ^4He channels and thus, either the ^4He should be considered as helium channels (including both isotopes) or otherwise during ^3He events the ^4He flux will be overestimated.

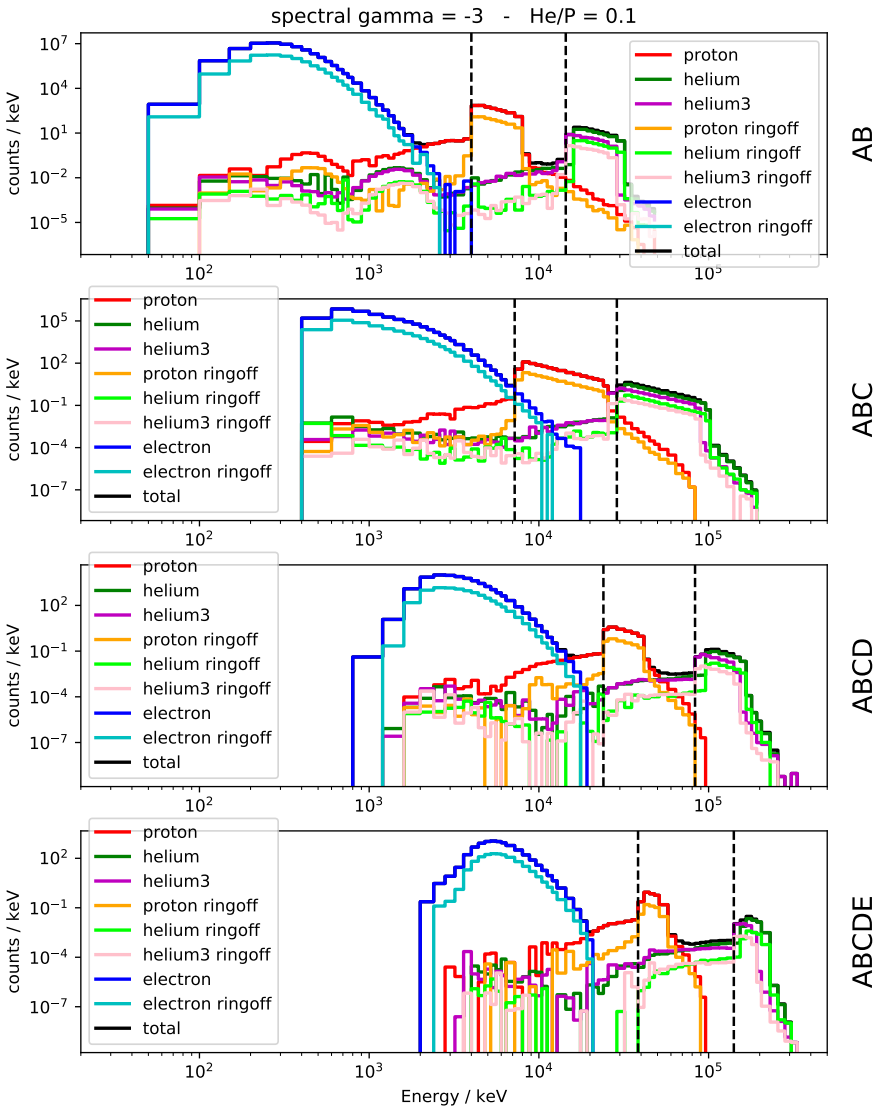

Figure 38: Expected counts including ^3He contributions.

A.3 Response to higher energies

In order to shed some ligth into the question how particles with sufficient energy do reach deeper detectors and hence coincidence conditions can still contribute to the lower channels, the start position and angle of a subsample of the simulated particles was tracked. Note that a more detailed tracking was not performed due to the relative small propability of this effect and the high computational demands of such a tracking. Furthermore, these tracking plots are for arbitrary incidence rather than the restricted parallel incidence to further enhance statistics. Figure 39 shows the response of the different histogram channels for the AB coincidence. A small but finite (note the logarithmic scale) contribution of protons and helium partiles with higher energies can be observed for the electron and proton as well as the background channel. Note that ions above 8 MeV/nuc are supposed to penetrate the B detector and hence causing an ABC coincidence.
Figure 40 shows trajectories of such particles being counted in the AB coincidence. The start position of the particles are marked in green, the trajectory was than (under the assumption of negectable scattering) extrapolated from the starting angle. The trajectories were colored in blue (before reaching detector A) and red (between detector A and B) for visualisation purposes. Figure 41 shows the derived position in the plane of the detectors at which the particles entered the detectors. The black circle indicates the extend of the detector. Figrue 42 shows the same results, however, as 1-D histogram over the radial distance of the particles wrt to the center of the detector. From the figure it can be seen that the particles were supposed to miss detector B. Since they are not hitting the C detector, a possibility is the creation of secondary electrons in the alumnium housing around A and B which are than measured in detector B while the primary ion is stopped in the housing.
Figure 43 shows the responses for the ABC coincidence in the same presentation as figure 39 presented the results for the AB coincidence. Particles with higher energies than the expected energy of 25 MeV necessary to penetrate C and causing an ABCD coincidence show again a small, finite response. Figures 44, 45 and 46 show the tracking of these particles in the same way as presented above for the AB coincidence. Note that larger radius of detector C compared to A and B in the figures. From the figure it is clear that the contribution of the higher energetic particles is due to particles penetrating A, B and C while slightly missing the D detector due to the geometry and design of the detector stack.

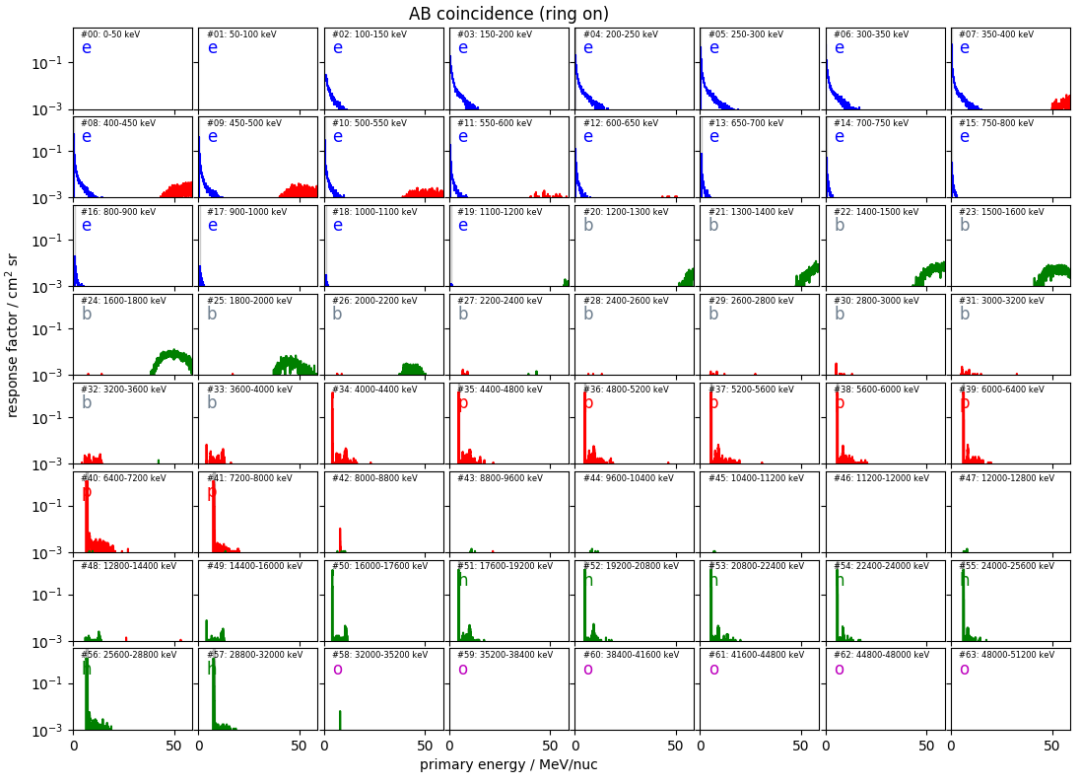

Figure 39: response high energy ab

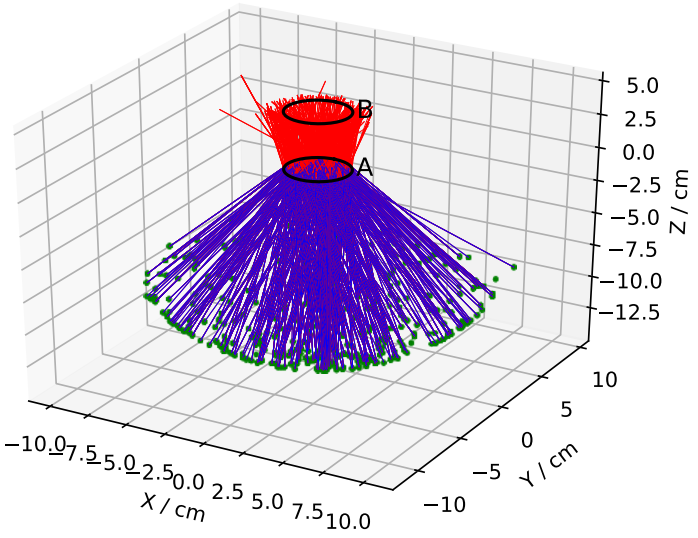

Figure 40: tracking ab

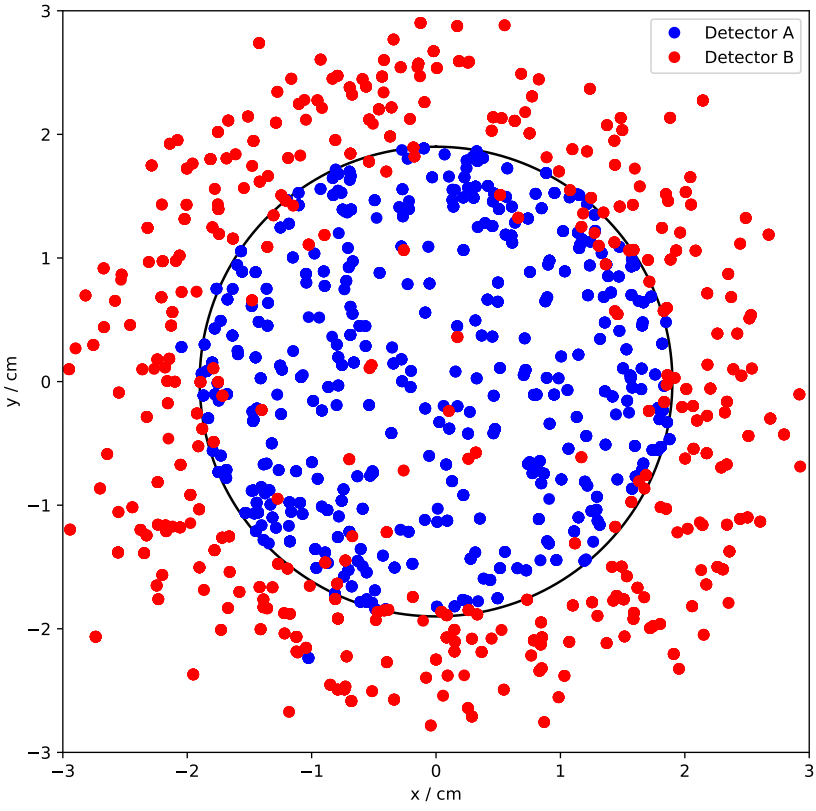

Figure 41: tracking pos ab

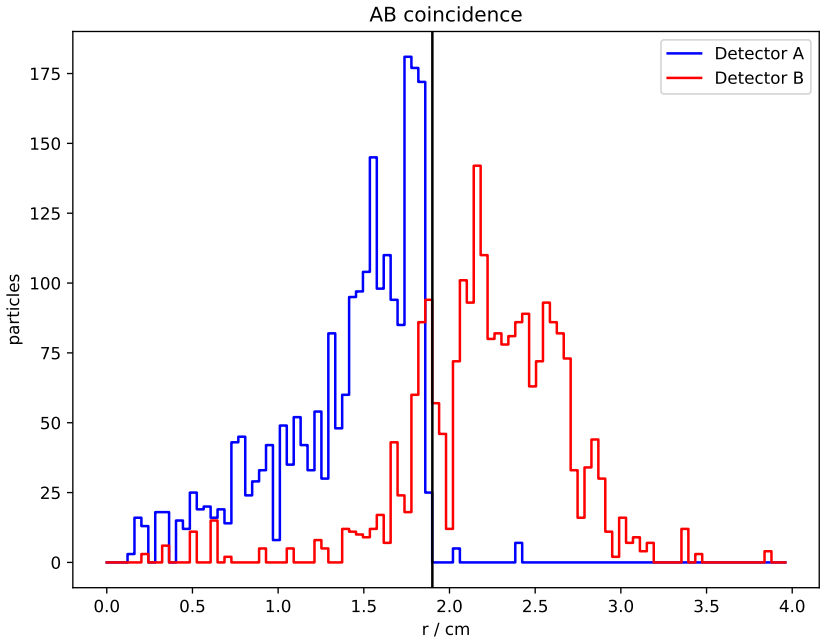

Figure 42: tracking pos r ab

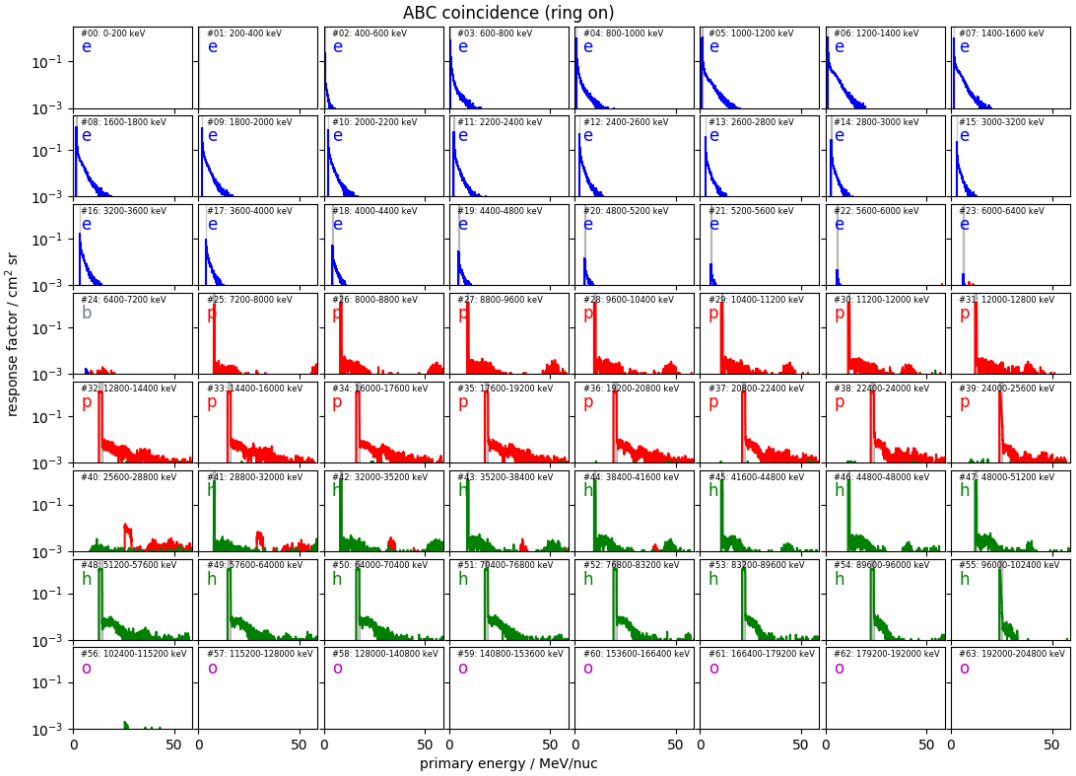

Figure 43: response high energy abc

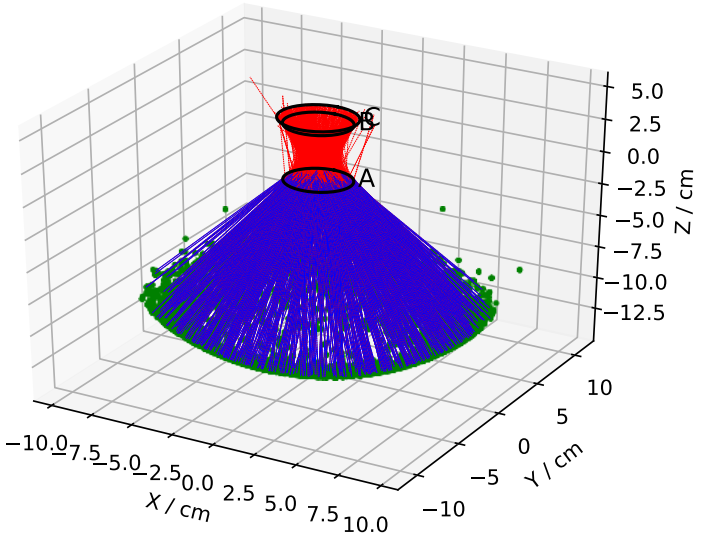

Figure 44: tracking abc

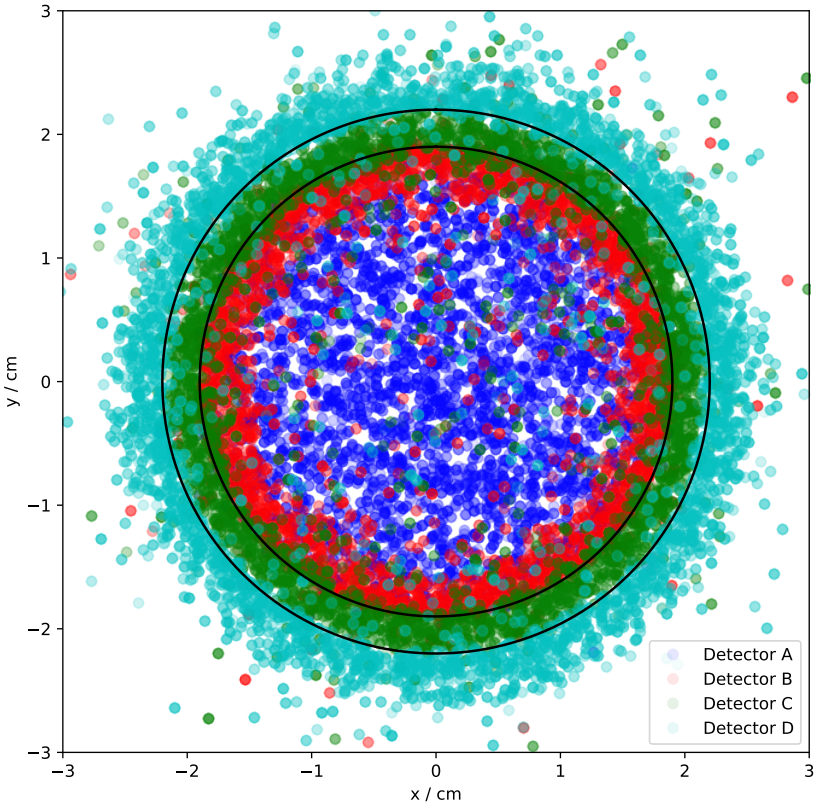

Figure 45: tracking pos abc

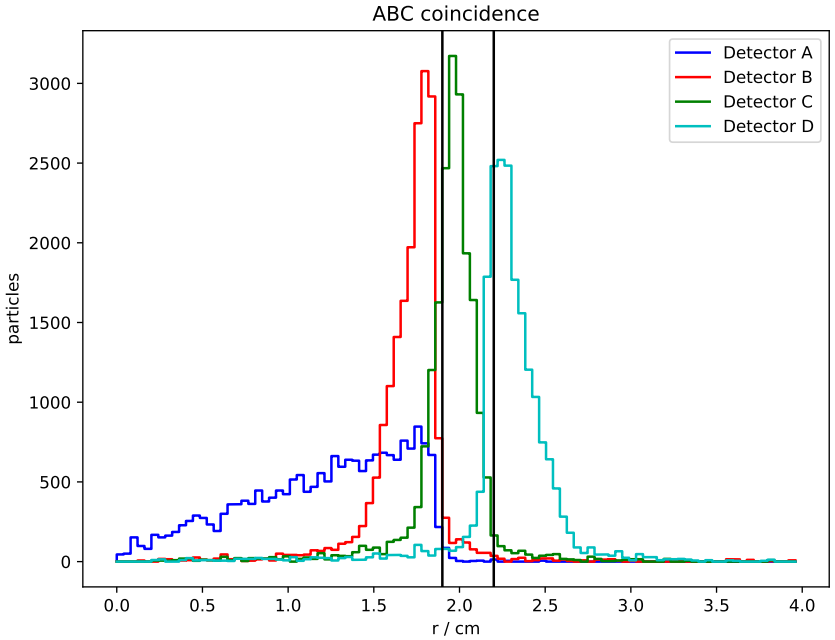

Figure 46: tracking pos r abc

B CODE and necessary files to produce files

B.1 JSONs including channel definitions and responses

The responses as well as channel definitions are stored as JSONs for further processing. Listing 1 gives a brief example of how to load these JSONs as a dictionary into python. Listings 2 and 3 provide the JSONs necessary for the calculation of proton and helium histogram spectra, respectively.

Listing 1: Exemplaric code for loading the response json as dictionary into python.

```
import json
type = "proton"                 # or "helium"
dict_name="histomasterch%sparalleldict.json"%type
histdict=json.load(open(dict_name))
print histdict.keys()           # print all available keys
print histdict["ecenter"]       # print energy center of bins
```

Listing 2: Proton JSON including channel definitions and responses


```
{"ch_in_coinces": [[35], [36], [37], [38], [39], [40], [41, 25], [26], [27],
    [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39, 31,
     31], [32, 32], [33, 33], [34, 34], [35, 35], [36, 36], [37, 37], [38,
    38], [39, 39]], "coinces": [["AB"], ["AB"], ["AB"], ["AB"], ["AB"], ["AB"
    ], ["AB", "ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["
    ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC", "ABCD
    ", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["AB
    CD", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["
    ABCD", "ABCDE"]], "ecenter": [4.725, 5.115, 5.51, 5.905, 6.3, 6.895,
    7.685, 8.48, 9.275, 10.07, 10.865, 11.66, 12.46, 13.655, 15.25, 16.845,
    18.44, 20.04, 21.635, 23.23, 24.83, 27.23, 30.43, 33.625, 36.82, 40.02,
    43.22, 46.42, 49.615], "emax": [4.92, 5.31, 5.71, 6.1, 6.5, 7.29, 8.08,
    8.88, 9.67, 10.47, 11.26, 12.06, 12.86, 14.45, 16.05, 17.64, 19.24, 20.84,
     22.43, 24.03, 25.63, 28.83, 32.03, 35.22, 38.42, 41.62, 44.82, 48.02,
    51.21], "emin": [4.53, 4.92, 5.31, 5.71, 6.1, 6.5, 7.29, 8.08, 8.88, 9.67,
     10.47, 11.26, 12.06, 12.86, 14.45, 16.05, 17.64, 19.24, 20.84, 22.43,
    24.03, 25.63, 28.83, 32.03, 35.22, 38.42, 41.62, 44.82, 48.02], "ewidth":
    [0.39, 0.39, 0.4, 0.39, 0.4, 0.79, 0.79, 0.8, 0.79, 0.8, 0.79, 0.8, 0.8,
    1.59, 1.6, 1.59, 1.6, 1.6, 1.59, 1.6, 1.6, 3.2, 3.2, 3.19, 3.2, 3.2, 3.2,
    3.2, 3.19], "masterch_id": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
    14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28], "
    masterch_name": ["H: 4.53-4.92 MeV", "H: 4.92-5.31 MeV", "H: 5.31-5.71 MeV
    ", "H: 5.71-6.10 MeV", "H: 6.10-6.50 MeV", "H: 6.50-7.29 MeV", "H:
    7.29-8.08 MeV", "H: 8.08-8.88 MeV", "H: 8.88-9.67 MeV", "H: 9.67-10.47 MeV
    ", "H: 10.47-11.26 MeV", "H: 11.26-12.06 MeV", "H: 12.06-12.86 MeV", "H:
    12.86-14.45 MeV", "H: 14.45-16.05 MeV", "H: 16.05-17.64 MeV", "H:
    17.64-19.24 MeV", "H: 19.24-20.84 MeV", "H: 20.84-22.43 MeV", "H:
    22.43-24.03 MeV", "H: 24.03-25.63 MeV", "H: 25.63-28.83 MeV", "H:
    28.83-32.03 MeV", "H: 32.03-35.22 MeV", "H: 35.22-38.42 MeV", "H:
    38.42-41.62 MeV", "H: 41.62-44.82 MeV", "H: 44.82-48.02 MeV", "H:
    48.02-51.21 MeV"], "masterch_type": "proton", "response_ringoff": [0.1857,
     0.1816, 0.1852, 0.1859, 0.1807, 0.182, 0.1838, 0.181, 0.1808, 0.1833,
    0.1859, 0.1871, 0.1831, 0.1854, 0.1838, 0.1841, 0.1817, 0.1842, 0.1853,
    0.1831, 0.1813, 0.1812, 0.1825, 0.1814, 0.1815, 0.1805, 0.1805, 0.1793,
    0.1779], "response_ringon": [1.0596, 1.0533, 1.0509, 1.0629, 1.0479,
    1.0616, 1.0536, 1.0378, 1.0552, 1.0443, 1.0572, 1.05, 1.045, 1.0631,
    1.0553, 1.0591, 1.0521, 1.0555, 1.054, 1.0538, 1.048, 1.0447, 1.0454,
    1.044, 1.0335, 1.0241, 0.9932, 0.9791, 0.964]}
```

The dictonary keys are as follows:

- ch_in_coinces: used histogram channel(s) for this masterchannel
- coinces: used histogram(s) for this masterchannel
- ecenter: mean energy for this masterchannel / MeV
- emax: max. energy for this masterchannel / MeV
- emin: min. energy for this masterchannel / MeV
- ewidth: energy width for this masterchannel / MeV
- masterch_id: ID of this masterchannel
- masterch_name: name of this masterchannel
- masterch_type: particle type for this histogram (proton or helium)
- response_ringoff: response for this masterchannel (ring off, /cm^{-2} sr^{-1})
- response_ringon: response for this masterchannel (ring on, /cm^{-2} sr^{-1})

Listing 3: Helium JSON including channel definitions and responses


```
{"ch_in_coinces": [[51], [52], [53], [54], [55], [56], [57, 41], [42], [43],
    [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55, 47,
     47], [48, 48], [49, 49], [50, 50], [51, 51], [52, 52], [53, 53], [54,
    54], [55, 55]], "coinces": [["AB"], ["AB"], ["AB"], ["AB"], ["AB"], ["AB"
    ], ["AB", "ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["
    ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC"], ["ABC", "ABCD
    ", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["AB
    CD", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["ABCD", "ABCDE"], ["
    ABCD", "ABCDE"]], "ecenter": [4.725, 5.115, 5.51, 5.905, 6.3, 6.895,
    7.685, 8.48, 9.275, 10.07, 10.865, 11.66, 12.46, 13.655, 15.25, 16.845,
    18.44, 20.04, 21.635, 23.23, 24.83, 27.23, 30.43, 33.625, 36.82, 40.02,
    43.22, 46.42, 49.615], "emax": [4.92, 5.31, 5.71, 6.1, 6.5, 7.29, 8.08,
    8.88, 9.67, 10.47, 11.26, 12.06, 12.86, 14.45, 16.05, 17.64, 19.24, 20.84,
     22.43, 24.03, 25.63, 28.83, 32.03, 35.22, 38.42, 41.62, 44.82, 48.02,
    51.21], "emin": [4.53, 4.92, 5.31, 5.71, 6.1, 6.5, 7.29, 8.08, 8.88, 9.67,
     10.47, 11.26, 12.06, 12.86, 14.45, 16.05, 17.64, 19.24, 20.84, 22.43,
    24.03, 25.63, 28.83, 32.03, 35.22, 38.42, 41.62, 44.82, 48.02], "ewidth":
    [0.39, 0.39, 0.4, 0.39, 0.4, 0.79, 0.79, 0.8, 0.79, 0.8, 0.79, 0.8, 0.8,
    1.59, 1.6, 1.59, 1.6, 1.6, 1.59, 1.6, 1.6, 3.2, 3.2, 3.19, 3.2, 3.2, 3.2,
    3.2, 3.19], "masterch_id": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
    14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28], "
    masterch_name": ["He: 4.53-4.92 MeV/nuc", "He: 4.92-5.31 MeV/nuc", "He:
    5.31-5.71 MeV/nuc", "He: 5.71-6.10 MeV/nuc", "He: 6.10-6.50 MeV/nuc", "He:
     6.50-7.29 MeV/nuc", "He: 7.29-8.08 MeV/nuc", "He: 8.08-8.88 MeV/nuc", "He
    : 8.88-9.67 MeV/nuc", "He: 9.67-10.47 MeV/nuc", "He: 10.47-11.26 MeV/nuc",
     "He: 11.26-12.06 MeV/nuc", "He: 12.06-12.86 MeV/nuc", "He: 12.86-14.45
    MeV/nuc", "He: 14.45-16.05 MeV/nuc", "He: 16.05-17.64 MeV/nuc", "He:
    17.64-19.24 MeV/nuc", "He: 19.24-20.84 MeV/nuc", "He: 20.84-22.43 MeV/nuc"
    , "He: 22.43-24.03 MeV/nuc", "He: 24.03-25.63 MeV/nuc", "He: 25.63-28.83
    MeV/nuc", "He: 28.83-32.03 MeV/nuc", "He: 32.03-35.22 MeV/nuc", "He:
    35.22-38.42 MeV/nuc", "He: 38.42-41.62 MeV/nuc", "He: 41.62-44.82 MeV/nuc"
    , "He: 44.82-48.02 MeV/nuc", "He: 48.02-51.21 MeV/nuc"], "masterch_type":
    "helium", "response_ringoff": [0.1864, 0.1816, 0.184, 0.1881, 0.1831,
    0.1825, 0.1838, 0.1825, 0.182, 0.1829, 0.1862, 0.1866, 0.1839, 0.1851,
    0.1836, 0.1826, 0.1811, 0.1821, 0.184, 0.1823, 0.1792, 0.1806, 0.1801,
    0.1785, 0.1778, 0.1762, 0.1754, 0.1739, 0.1715], "response_ringon":
    [1.0601, 1.0578, 1.0484, 1.0681, 1.0511, 1.0651, 1.0557, 1.0427, 1.0591,
    1.0448, 1.0538, 1.0481, 1.0445, 1.0613, 1.0523, 1.0547, 1.0434, 1.0483,
    1.0445, 1.0431, 1.0381, 1.0413, 1.034, 1.0275, 1.0153, 1.0013, 0.9687,
    0.9526, 0.9339]}
```

B.2 CODE I: ephin_histograms_funcs.py

Listing 4: Collection of required functions for histflux calculation.

```
import numpy as np
import os
import json
import datetime as dt
import warnings

### required code for calculating EPHIN histfluxes from lvl1-sci files


### working example:
"""
execfile("ephin_histograms_funcs.py")
output_path="histfluxes/"
sci_path="/data/missions/soho/costep/level1/sci/"
response_dict_path="../SCRIPTS/"
timeconstant_coinc_counter=0.0102
time_accumulation=59.953
os.system("mkdir %s -p"%output_path)
os.system("rm -f %s*"%output_path)
create_histogram_flux_files(1997,234,2017,277,output_path,sci_path,
    response_dict_path,timeconstant_coinc_counter,time_accumulation)
"""


### included functions:
"""
# time helper functions
year_doy_iterator(year,doy) - returns the year and day one day afer input date
time_from_msec(millisec) - returns hours and minutes from input millisec
timediff(array_year,array_doy,array_msec,line1,line2) - returns timedelta
    between two lines

# write header
write_header(toutfile,type,tdict,timeconstant_coinc_counter,time_accumulation)
     - writes header to output file

# data loading
load_level1_sci(year,doy,sci_path) - load level1 sci files
load_and_merge_consecutive_scis(year,doy,sci_path) - merges two consecutive
    sci files
create_ringbit_array(data) - extracts ringbit information from data array
create_hist_array(data) - extracts histograms from data array
create_histbit_array(data) - extracts histbit information from data array
create_total_coinc_counter_array(data) - extracts total sum of coincidence
    counters from data array
create_all_the_arrays(data) - executes all data extractions listed above

# load masterchannel response json
load_masterchannel_response_dict(type) - loads the histogram response JSON

# flux calculation
calc_coinc_counter_per_min_during_hist(array_total_coinc_counter,line,
    thistdate,array_year,array_doy,array_msec) - calculates coincidence
    counters during hist integration interval
calc_coinc_deadtime_correction(coinc_counter_sum_per_min,
    timeconstant_coinc_counter,time_accumulation) - calculates counter scaling
     used during ringon
def calc_coinc_counter_ratio(coinc_counter_sum_per_min, array_hist,line) -
    calculated count ratio coinc/histo used for scaling during ringoff
extract_and_unfold_hists(array_hist,line) - unfold histograms from hist array
```

```
51 calc_and_save_hist(array_ringbit,array_hist,array_histbit,
       array_total_coinc_counter,array_year,array_doy,array_msec,line,
       masterch_dict_list,types,timeconstant_coinc_counter,time_accumulation,
       output_path) - calculates and saves fluxes for a given histogram
52
53 # actual data processing
54 create_histogram_flux_files(start_year,start_doy,end_year,end_doy,output_path,
       sci_path,response_dict_path,timeconstant_coinc_counter,time_accumulation)
       - main function incorporating the above ones in order to create histogram
       flux files
55 """
56
57 # time helper functions
58 def year_doy_iterator(year,doy):
59     # returns the year doy tuple of next day
60     if doy>370: year,doy=year+1,1
61     else: year,doy=year,doy+1
62     return year, doy
63
64 def time_from_msec(millisec):
65   minutes=millisec/60000.
66   hour=0
67   while minutes>=60:
68     hour+=1
69     minutes-=60
70   return hour,minutes
71
72 def timediff(array_year,array_doy,array_msec,line1,line2):
73     # calculates the time diffents of two lines
74     year1,doy1,msec1=array_year[line1],array_doy[line1],array_msec[line1]
75     year2,doy2,msec2=array_year[line2],array_doy[line2],array_msec[line2]
76     time1=dt.datetime(int(year1),1,1)+dt.timedelta(int(doy1)-1,milliseconds=
           msec1)
77     time2=dt.datetime(int(year2),1,1)+dt.timedelta(int(doy2)-1,milliseconds=
           msec2)
78     return time2-time1
79
80 # write header
81 def write_header(toutfile,type,tdict,timeconstant_coinc_counter,
       time_accumulation):
82     f=open(toutfile,"w")
83     f.write("# SOHO/EPHIN histogram fluxes - %s\n"%type)
84     f.write("# !!! Note that the histogram fluxes feature several caveats (
           especially during hard energy spectra) - Please read the Manual/
           Documentation !!!\n")
85     f.write("# %s fluxes and their statistical uncertainties are given in %i
           channels\n"%(type,len(tdict["masterch_id"])))
86     f.write("# The time given indicates the beginning of a 8 minute
           integration period\n")
87     f.write("# Councidence counter time constant (used during ringon) was set
           to %4.8ss (see Documentation)\n"%(timeconstant_coinc_counter))
88     f.write("# The columns represent:\n")
89     f.write("# Year, Month, Day, DayOfYear, Hour, Minute, Ringbit,
           Coinccounterscaler, %i flux columns, %i number of counts columns\n"%(
           len(tdict["masterch_id"]),len(tdict["masterch_id"])))
90     f.write("# If ringbit ==1: outer segments are turned off (high flux mode)\
           n")
91     f.write("# Fluxes are given in units of (cm^2 sr s MeV/nuc)^-1\n")
92     f.write("# The number of counts columns can be used to determine
           statistical uncertainties (sigma=flux/sqrt(N)) \n")
93     f.write("# The energy channels are as follows (given in units of MeV/nuc)
           :\n")
94     emeans="# E_center: "
```

```
95      for q in tdict["ecenter"]: emeans+="%4.2f "%q
96      f.write(emeans+"\n")
97      emins="# E_min: "
98      for q in tdict["emin"]: emins+="%4.2f "%q
99      f.write(emins+"\n")
100     emaxs="# E_max: "
101     for q in tdict["emax"]: emaxs+="%4.2f "%q
102     f.write(emaxs+"\n")
103     f.write("# !!! Note that the histogram fluxes feature several caveats (
            especially during hard energy spectra) - Please read the Manual/
            Documentation !!!\n")
104     f.close()
105
106 # data loading
107
108 def load_level1_sci(year,doy,sci_path):
109     if year>=2000:  fname="epi%02d%03d.sci"%(year-2000,doy)
110     else:  fname="eph%02d%03d.sci"%(year-1900,doy)
111     filename=sci_path+"%04d/"%year+fname
112     if os.path.isfile(filename):
113         with warnings.catch_warnings():
114             warnings.simplefilter("ignore")
115             data=np.loadtxt(filename)
116         if data != []:
117             return data
118         else:
119             return []
120     else:
121         return []
122
123
124 def load_and_merge_consecutive_scis(year,doy,sci_path):
125     data1,data2=[],[]
126     while data1==[]:
127         data1=load_level1_sci(year,doy,sci_path)
128         year,doy=year_doy_iterator(year,doy)
129     while data2==[]:
130         data2=load_level1_sci(year,doy,sci_path)
131         year,doy=year_doy_iterator(year,doy)
132         if data2!=[]:
133             if len(data2[:,0]) <=8:
134                 data2=[]
135     merged_data=np.vstack((data1,data2))
136     return merged_data
137
138 def create_ringbit_array(data):
139     # ring off == 1
140     # ring on == 0
141     status=data[:,-1]
142     ringoff=np.zeros(len(status))
143     for q in range(len(status)):
144         binaries='{0:08b}'.format(int(status[q]))
145         if int(binaries[-2]): ringoff[q]=1
146     return ringoff
147
148 def create_hist_array(data):
149     hist=data[:,-40:-8]
150     return hist
151
152 def create_histbit_array(data):
153     histbit=data[:,-8]-64
154     return histbit
155
```

```
156 def create_total_coinc_counter_array(data):
157     #sdata=np.split(data,len(data[0,:]),1)
158     #print sdata
159     columns= [0,1,2,36,37,38,39, 22,23,24, 25,26,27, 41,42,43, 44,45,46,
            28,29,30,31, 32,33,34,35, 47,48,49,50, 51,52,53,54, 40]
160     variables=[]
161     for q in columns: variables.append(data[:,q])
162     year,doy,msdoy,e1,e2,e3,e4,p1_1,p1_2,p1_3,p2_1,p2_2,p2_3,p3_1,p3_2,p3_3,
            p4_1,p4_2,p4_3, h1_1,h1_2,h1_3,h1_4,h2_1,h2_2,h2_3,h2_4,h3_1,h3_2,h3_3
            ,h3_4,h4_1,h4_2,h4_3,h4_4, total_int_counts= variables
163     p1=p1_1+p1_2+p1_3
164     p2=p2_1+p2_2+p2_3
165     p3=p3_1+p3_2+p3_3
166     p4=p4_1+p4_2+p4_3
167     h1=h1_1+h1_2+h1_3+h1_4
168     h2=h2_1+h2_2+h2_3+h2_4
169     h3=h3_1+h3_2+h3_3+h3_4
170     h4=h4_1+h4_2+h4_3+h4_4
171     ab=e1+p1+h1
172     abc=e2+p2+h2
173     abcd=e3+p3+h3
174     abcde=e4+p4+h4
175     totat_coinc_counter_array= ab+abc+abcd+abcde  #+total_int_counts
176     return totat_coinc_counter_array
177
178 def create_all_the_arrays(data):
179     array_ringbit=create_ringbit_array(data)
180     array_hist=create_hist_array(data)
181     array_histbit=create_histbit_array(data)
182     array_total_coinc_counter=create_total_coinc_counter_array(data)
183     array_year=data[:,0]
184     array_doy=data[:,1]
185     array_msec=data[:,2]
186     return array_ringbit,array_hist,array_histbit,array_total_coinc_counter,
            array_year,array_doy,array_msec
187
188
189 # load masterchannel response json
190 def load_masterchannel_response_dict(response_dict_path,type):
191     this_dict=json.load(open(response_dict_path+"histo_masterch_%
            s_parallel_dict.json"%type))
192     return this_dict
193
194
195
196 # flux calculation
197 def calc_coinc_counter_per_min_during_hist(array_total_coinc_counter,line,
        thistdate,array_year,array_doy,array_msec):
198     lineoffset=0
199     tcounter_year,tcounter_doy,tcounter_msec=array_year[line+lineoffset],
            array_doy[line+lineoffset],array_msec[line+lineoffset]
200     tcounterdate=dt.datetime(int(tcounter_year),1,1)+dt.timedelta(int(
            tcounter_doy)-1,milliseconds=int(tcounter_msec))
201     while tcounterdate-thistdate>=dt.timedelta(0):
202         lineoffset-=1
203         tcounter_year,tcounter_doy,tcounter_msec=array_year[line+lineoffset],
                array_doy[line+lineoffset],array_msec[line+lineoffset]
204         tcounterdate=dt.datetime(int(tcounter_year),1,1)+dt.timedelta(int(
                tcounter_doy)-1,milliseconds=int(tcounter_msec))
205     lineoffset+=1 # we went one too far ;)
206     coinc_counter_sum_per_min=np.mean(array_total_coinc_counter[line+
            lineoffset:line])
207     return coinc_counter_sum_per_min
```

```
208
209 def calc_coinc_deadtime_correction(coinc_counter_sum_per_min,
        timeconstant_coinc_counter,time_accumulation):
210     # returns the deadtime correction for the coinc counters. The returned
            scaler needs to be multiplied with the histogram counts
211     def func(x,p):
212         y = 1./(1.+p*x)
213         return y
214     coinc_counter_sum_per_second=coinc_counter_sum_per_min/time_accumulation
215     scaler=1./func(coinc_counter_sum_per_second,timeconstant_coinc_counter)
216     return scaler
217
218 def calc_coinc_counter_ratio(coinc_counter_sum_per_min, array_hist,line):
219     scaler=coinc_counter_sum_per_min*8./np.sum(array_hist[line:line+8,:])
220     return scaler
221
222 def extract_and_unfold_hists(array_hist,line):
223     unscaled_hist=array_hist[line:line+8,:]
224     unscaled_hist_ab=unscaled_hist[0:2,:].flatten()
225     unscaled_hist_abc=unscaled_hist[2:4,:].flatten()
226     unscaled_hist_abcd=unscaled_hist[4:6,:].flatten()
227     unscaled_hist_abcde=unscaled_hist[6:8,:].flatten()
228     unscaled_hist_dict={"AB": unscaled_hist_ab,"ABC":unscaled_hist_abc,"ABCD":
            unscaled_hist_abcd,"ABCDE":unscaled_hist_abcde}
229     return unscaled_hist_dict
230
231
232 def calc_and_save_hist(array_ringbit,array_hist,array_histbit,
        array_total_coinc_counter,array_year,array_doy,array_msec,line,
        masterch_dict_list,types,timeconstant_coinc_counter,time_accumulation,
        output_path):
233     # derive deadtime scaler
234     thist_year,thist_doy,thist_msec=array_year[line],array_doy[line],
            array_msec[line]
235     thistdate=dt.datetime(int(thist_year),1,1)+dt.timedelta(int(thist_doy)-1,
            milliseconds=int(thist_msec),minutes=-8)
236     coinc_counter_sum_per_min=calc_coinc_counter_per_min_during_hist(
            array_total_coinc_counter,line,thistdate,array_year,array_doy,
            array_msec)
237     coinc_counter_deadtime_scaler=calc_coinc_deadtime_correction(
            coinc_counter_sum_per_min, timeconstant_coinc_counter,
            time_accumulation)
238     coinc_counter_ringoff_scaler=calc_coinc_counter_ratio(
            coinc_counter_sum_per_min, array_hist,line)
239     unscaled_hist_dict=extract_and_unfold_hists(array_hist,line)
240     # loop over types
241     for typerunner, type in enumerate(types):
242         tdict=masterch_dict_list[typerunner]
243         # extract counts in master chs
244         masterch_counts=[]
245         for chrunner, masterchnummer in enumerate(tdict["masterch_id"]):
246             masterch_counts.append(0)
247             for coin_in_masterch_runner in range(len(tdict["coinces"][chrunner
                    ])):
248                 tcoinc= tdict["coinces"][chrunner][coin_in_masterch_runner]
249                 tchincoinc= tdict["ch_in_coinces"][chrunner][
                        coin_in_masterch_runner]
250                 masterch_counts[-1]+= unscaled_hist_dict[tcoinc][tchincoinc]
251         # derive flux/staterr in master chs
252         masterch_fluxes=[]
253         for chrunner, masterchnummer in enumerate(tdict["masterch_id"]):
254             if array_ringbit[line-1]==1.:
255                 tresponse=tdict["response_ringoff"][chrunner]
```

```
256                 countscaler=coinc_counter_ringoff_scaler
257             else:
258                 tresponse=tdict["response_ringon"][chrunner]
259                 countscaler=coinc_counter_deadtime_scaler
260             tflux=masterch_counts[chrunner] / (8.*time_accumulation) / tdict["
                    ewidth"][chrunner] / tresponse *countscaler
261             masterch_fluxes.append(tflux)
262         # write output
263         toutfile=output_path+"histfluxes_%s_%i.dat"%(type,thist_year)
264         if not os.path.isfile(toutfile): write_header(toutfile,type,tdict,
                timeconstant_coinc_counter,time_accumulation)
265         f=open(toutfile,"a")
266         f.write("%i %i %i %i  %i %i  %i %4.4e  "%(thistdate.year,thistdate.
                month,thistdate.day,int(thistdate.strftime("%j")), thistdate.hour,
                thistdate.minute,  array_ringbit[line-1], countscaler))
267         for tflux in masterch_fluxes:
268             f.write(" %4.4e"%tflux)
269         f.write("    ")
270         for counts in masterch_counts:
271             f.write(" %i"%counts)
272         f.write("\n")
273         f.close()
274
275
276
277 # actual data processing
278 def create_histogram_flux_files(start_year,start_doy,end_year,end_doy,
        output_path,sci_path,response_dict_path,timeconstant_coinc_counter,
        time_accumulation):
279     types=["proton","helium"]
280     masterch_dict_list=[]
281     for type in types: masterch_dict_list.append(
            load_masterchannel_response_dict(response_dict_path,type))
282     stat_good,stat_bad=0.,0.
283     tyear,tdoy=start_year,start_doy
284     printyear=start_year-1
285     data=load_and_merge_consecutive_scis(tyear,tdoy,sci_path)
286     array_ringbit,array_hist,array_histbit,array_total_coinc_counter,
            array_year,array_doy,array_msec=create_all_the_arrays(data)
287     line=9
288     while data[line,0] < end_year or (data[line,0] == end_year and data[line
            ,1]<=end_doy+1):
289         if data[line,0]!= printyear:
290             printyear+=1
291             print("Processing %i..."%printyear)
292         if array_histbit[line]==0:
293             no_hist_issue=True
294
295             # check if hist is complete and correctly ordered
296             comparison=array_histbit[line:line+8]==np.array([0,1,2,3,4,5,6,7])
297             if not comparison.all(): no_hist_issue=False
298
299             # check if no ringswitch occours during previous 8 minutes:
300             tringbit=array_ringbit[line-8:line]
301             if len(np.unique(tringbit))!=1:
302                 no_hist_issue=False
303
304             # check if the time stamp from line-8 till line+7 is not too long
                    or too short
305             # (should be 16, used threshold is 18 in order to allow two
                    scaling counters being missing - this is taken care of in the
                    scaling stuff)
306             thistimediff=timediff(array_year,array_doy,array_msec,line-8,line
```

```
                +7).seconds
307             if thistimediff < 14.5*60 or thistimediff > 18.5*60:
308                 no_hist_issue=False
309
310             if no_hist_issue==True:
311                 calc_and_save_hist(array_ringbit,array_hist,array_histbit,
                        array_total_coinc_counter,array_year,array_doy,array_msec,
                        line,masterch_dict_list,types,timeconstant_coinc_counter,
                        time_accumulation,output_path)
312                 stat_good+=1
313             else:
314                 stat_bad+=1
315
316             line+=1
317         else:
318             line+=1
319         if line>len(array_ringbit)-10:
320             tyear,tdoy,tmsec=array_year[line],array_doy[line],array_msec[line]
321             #print tyear,tdoy, line
322             data=load_and_merge_consecutive_scis(tyear,tdoy,sci_path)
323             array_ringbit,array_hist,array_histbit,array_total_coinc_counter,
                    array_year,array_doy,array_msec=create_all_the_arrays(data)
324             line=np.where((array_msec==tmsec)&(array_doy==tdoy))[0][0]
325
326     print("Hist issues were found in %2.2f %% of all cases"%(stat_bad/
            stat_good*100))
```

B.3 CODE II: 10_calc_annual_hist_files.py

Listing 5: Working example for histflux calculation

```
1  from ephin_histograms_funcs import *
2  output_path="histfluxes/"
3  sci_path="/data/missions/soho/costep/level1/sci/"
4  response_dict_path="../SCRIPTS/"
5  timeconstant_coinc_counter=0.00747      ###0.0102
6  time_accumulation=59.953
7
8  os.system("mkdir %s -p"%output_path)
9  os.system("rm -f %s*"%output_path)
10
11 create_histogram_flux_files(1997,234,2017,277,output_path,sci_path,
       response_dict_path,timeconstant_coinc_counter,time_accumulation)
```