Documentation of the SOHO/EPHIN Level3 data product

Patrick Kuehl

February 1, 2022

Contacts

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

Contents

1 Introduction

"The *Electron Proton Helium Instrument* (EPHIN, Müller-Mellin et al. 1995) is part of the *Comprehensive Suprathermal and Energetic Particle Analyzer* (COSTEP) instrument suite onboard the *Solar and Heliospheric Observatory* (SOHO). SOHO was launched in December 1995 and has an orbit around the Lagrangian point L1. Figure 1 (left) shows a sketch of the instrument, which consists of six solid-state detectors (labeled A - F) enclosed in a scintilator that acts as anticoincidence (G). The measurements of EPHIN rely on the dE/dx-E method, which yields count rates for different ranges in the silicon detector stack. As described by Müller-Mellin et al. (1995), different ions and even isotopes can be identified based on the energy deposition in the first detector ΔE_A and the sum of the energy depositions E in all detectors. In addition to the total counts of these different coincidence conditions, energy losses in each detector are available for a statistical sample of individual particle tracks, allowing a detailed analysis of the measured particles including the calculation of energy spectra for electrons up to ≈ 10 MeV and ions up to ≈ 50 MeV/nucleon." (Adapted from Kühl et al., 2015)

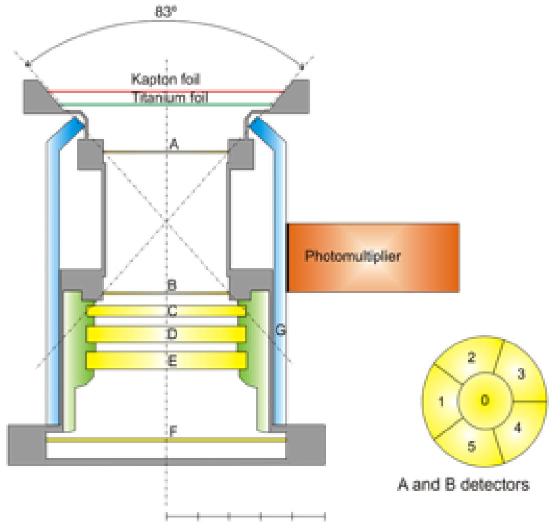

Figure 1: Sketch of the EPHIN instrument (Kühl et al., 2015).

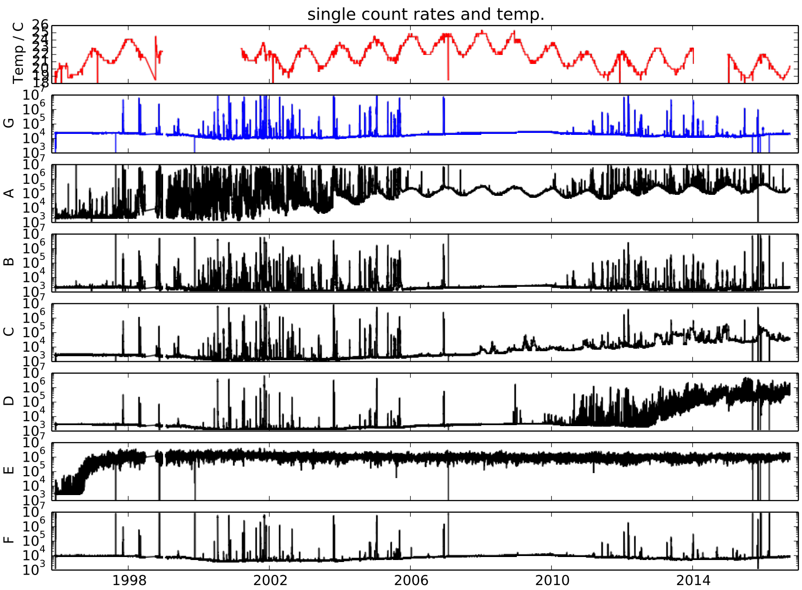

Figure 2: Temperature and count rates of the single detectors of SOHO/EPHIN.

In order to monitor the state of the different detectors of the instrument, the count rates of the different detectors without any coincidence condition are recorded. These count rates shown in figure 2 indicate a significant increase in the detector noise of detector E and D around 1997 and 2013, respectively. The energy loss histograms of these detectors shown in figure 3 further validate this statement. To account for the noise issue, the EPHIN instrument was switched to failure modes E and D October 31, 1996 and October 4, 2017, respectively. The failure modes prevent dead-time issues but are reducing the energy resolution, i.e. the nominal four proton channels are merged into three and two channels with failure modes E and D, respectively.

In order to restore all four energy channels and to ensure constant data quality over the last two decades and for as long as the mission will continue, a new data analysis solely based on detecotrs A, B and C has been performed.

The aims of this new Level3 data are

- to compensate for the loss of detector D and E
- to make the use of the *.kor files obsolete
- to produce a consistent data product over the entire mission

This will be achieved by

- the dE/dx-E method for coincidences AB
- the dE/dx-dE/dx method for ABC, ABCD, ABCDE
- both using only PHA (Level 2, *.PHA files) and coincidence counters (Level 1, *.SCI files)

Note that electron channels can not be recovered by the method presented here. Hence, for electron intensities, the user is refered to the nominal level2 data product.

In the following, the procedure to calculate Level3 proton and helium intensities is derived and explained for different coincidences in sections 2 and 3. Section 4 shows comparisons of the new data product to measurements from other missions/instruments. The new data product is explained in section 5, a short introduction to the Python code producing level3 data is given in section 6. The actual code can be found in sections 7 and 8. Section 9 contains geometry files required by the code.

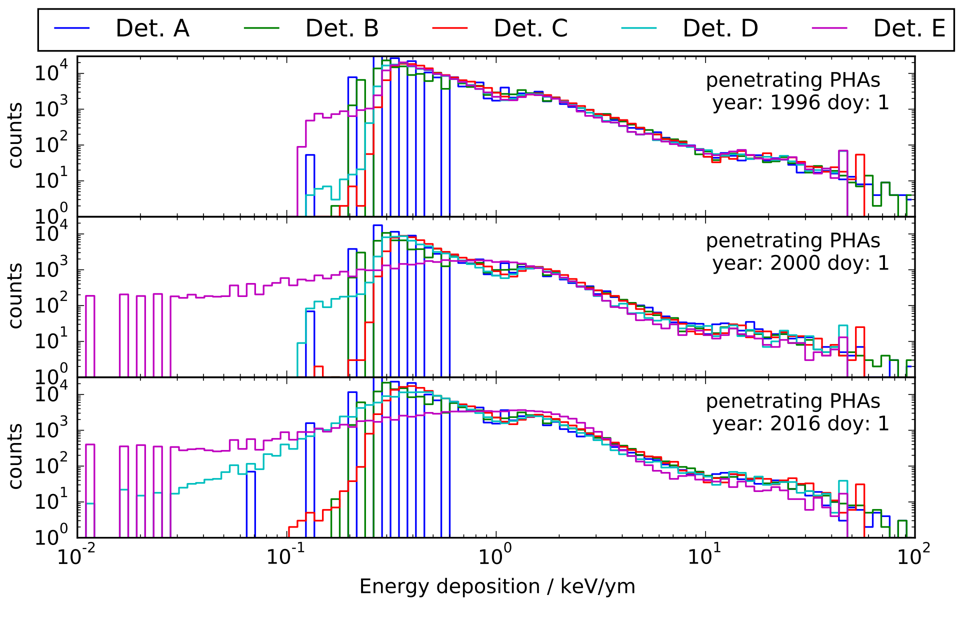

Figure 3: Histogram of the differential energy loss for penetrating particles. Shown are detectors A to E for the exemplary years 1996, 2000 and 2016.

2 A B ¬C ¬F coincidence

2.1 Particle Identification

In this chapter we restrict our analysis to the coincidence $A \wedge B \wedge \neg C \wedge \neg F$.
Figure 4 presents the energy deposition in det. B E_B as function of the energy deposition in det. A E_A for this coincidence (E150, P4 and H4 channel) for simulations of protons, helium and electrons, respectively. Simulations have been performed for isotropic fluxes with an energy independent intensity. The different particle populations can be clearly distinguished.
However, the figure shows an influence of δ-electrons for protons and helium, i.e. the populations with barely any energy loss in detector B but significant deposition in detector A. In order to remove this contribution, a lower threshold for the energy deposition in B was defined. Figure 5 shows the influence of the threshold value for protons in the coincidences E150, P4 and H4. Based on this plot, the threshold has been choosen to be set to 0.13 MeV (shown as magenta-black dashed line in figure 4) in order to supress the detection of low energy protons and helium.

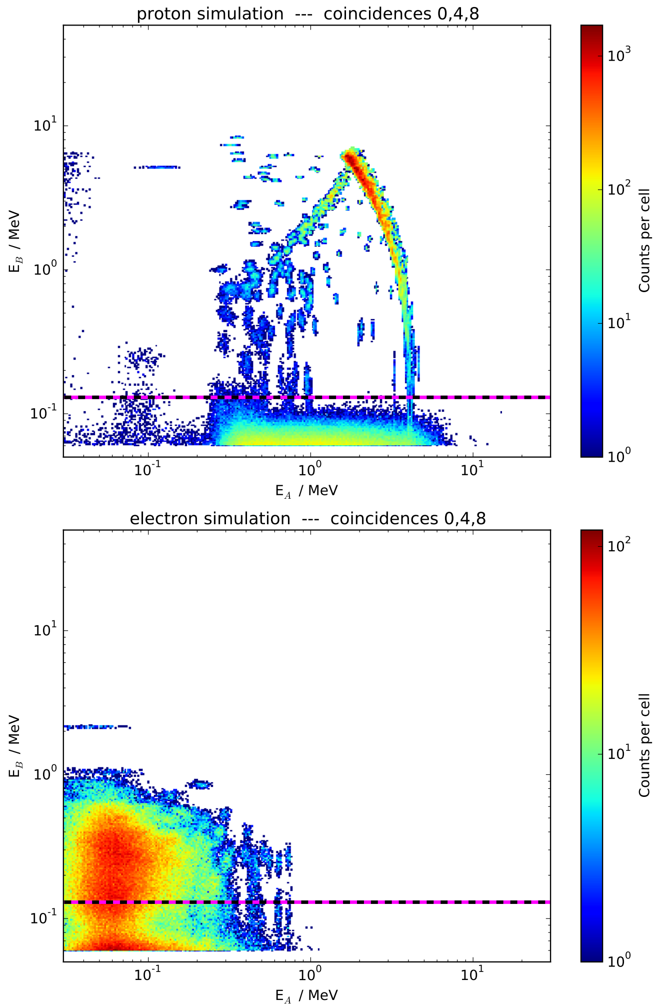

Figure 4: Energy deposition in det. B as a function of det. A for coincidences E150, P4 and H4. Shown are the results from proton (upper left), helium (upper right) and electron (lower left) simulations. Simulations have been performed for isotropic fluxes with an energy independent intensity.

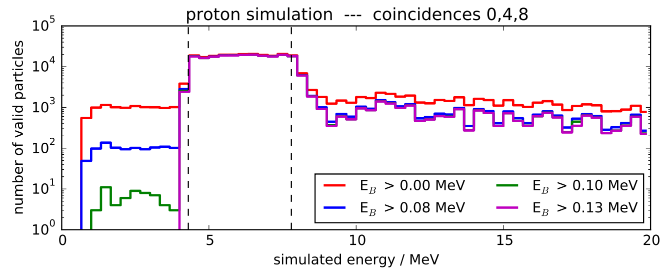

Figure 5: Analysis of thresholds in the energy loss in detector B in order to exclude δ electrons from the data set.

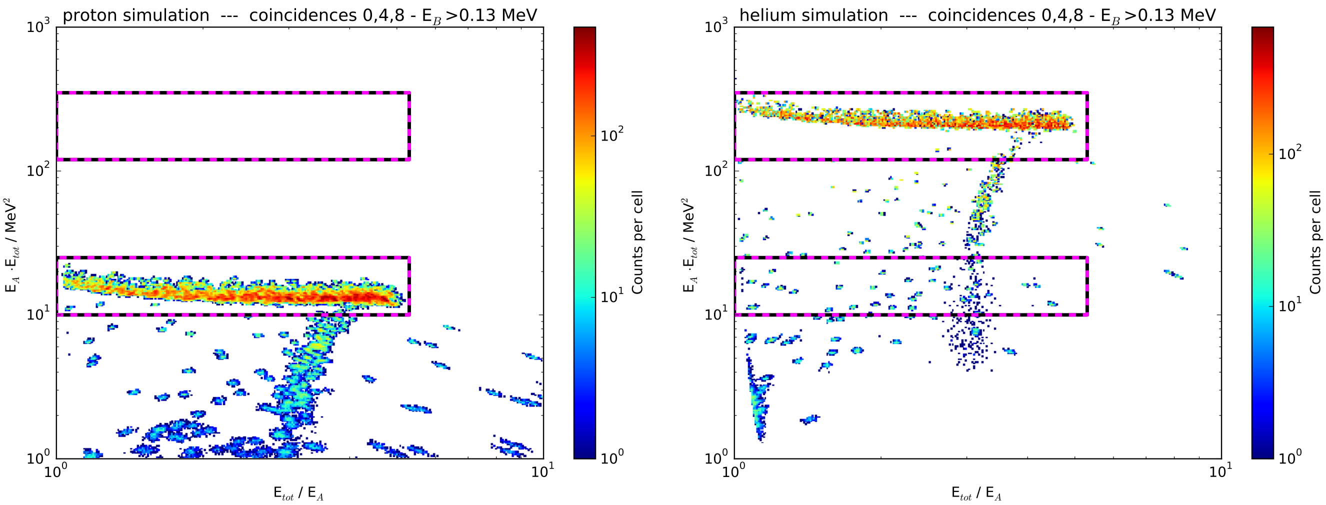

Figure 6: Product of total energy and energy deposition in detector A as function of their ratio. Shown are the results from proton (left) and helium (right) simulations.

Figure 6 shows the product of total energy and energy deposition in detector A as a function of their ratio for proton and helium simulations, respectively. Based on these histograms, we can define thresholds (boxes) for proton and helium identification:

$$\kappa := E_B \quad (1)$$

and

$$\lambda := (E_A + E_B) \cdot E_A \quad (2)$$

and

$$\mu := (E_A + E_B)/E_A \quad (3)$$

Proton selection criteria:

- $\kappa > 0.13$ MeV
- 10 MeV$^2 < \lambda < 25$ MeV^2
- $1.0 < \mu < 5.3$

Helium selection criteria:

- $\kappa > 0.13$ MeV
- 120 MeV$^2 < \lambda < 350$ MeV^2
- $1.0 < \mu < 5.3$

Note that the helium box has been chosen to be larger than the proton box in order to include both, ^{3}He and ^{4}He particles.

2.2 Energy Determination

The total kinetic energy of the measured particles is given by the sum of the energy depositions in detectors A and B ($E_{tot}=E_A+E_B$). It has to be noted though, that a small fraction of particles with higher energies can miss detector C depositing a significant amount of energy in the aluminium housing.

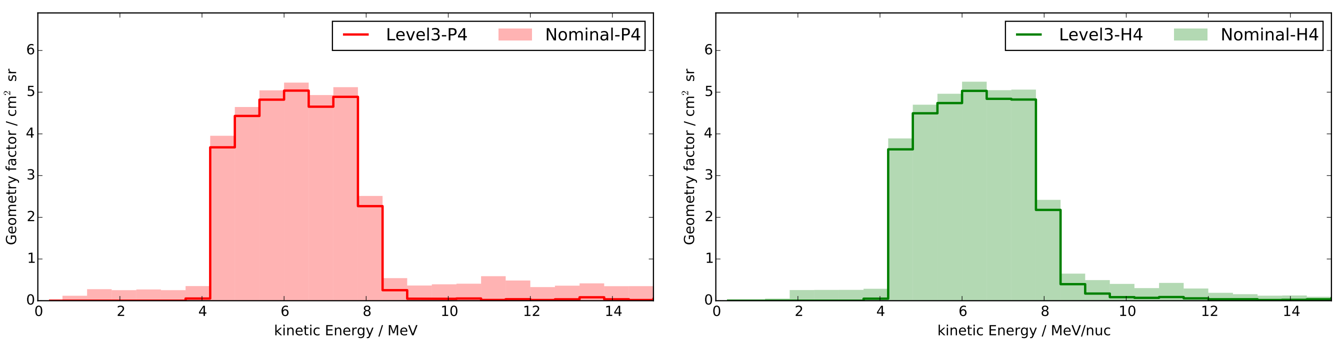

Figure 7: Nominal and Level3 response function for protons (left) and helium (right).

2.3 Response Factor

In order to calculate the intensity of a given particle type from the measured count rate, the geometry factor has to be calculated. The energy dependent geometry factor is also called response function. In order to calculate differential intensities (i.e. in units of $(cm^2\ sr\ s\ MeV)^{-1}$), the energy range of a channel has to be taken into account as well. This factor can be interpreted as product of geometry factor and energy width of a channel or - for a non-ideal detector - as integration of the response function. In the following, the corresponding factor is called response factor.

Figure 7 show the response functions for P4 and H4 for both the nominal channel and the selection described above. To illustrate common issues with the response of channels with rather wide energy coverage, figure 8 shows several simulated power-laws (solid lines) as well as the resulting intensities in the nominal proton (left) and helium (right) channels. For an ideal detector, the intensity of a channel would agree with the intensity of the simulated spectra at a given energy, independent on the spectral shape. However, due to the broad energy range of the channel the exact energy at which the channel intensity equals the one of the input spectra changes as a function of the spectral shape (i.e. the power-law index γ). Due to the finite response of the channels to energies lower/higher than the ideal response (see tails in the nominal response function in figure 7) the channel intensity can be even lower/higher than the minimum/maximum intensities simulated in the nominal energy range. In order to account for this issue and to calculate valid response factors and reasonable systematic uncertainties, simulations with several power-laws have been evaluated with the Level3 selection criteria as defined above. Based on the counts in these artificial data sets, the response factors that would result in intensities in the level3 selection that would match the simulated intensities at the geometric mean of the energy in the channel have been calculated. Figure 9 shows these respons factors for proton (left) and helium (right) simulations as a function of the power-law index γ. Reasonable values for the response factors and their systematic uncertainties are than given by the mean as well as the standard deviation of these factors. The factors are also calculated for the ring-off mode (i.e. only the inner segments of A and B are active).

Response Factor in nominal observation mode:

- P4: (20.08 ± 2.32) cm^2 sr MeV
- H4: (23.26 ± 4.09) cm^2 sr MeV/nuc

Response Factor in ring-off mode (i.e. only the inner segments of A and B are active):

- P4: (0.69 ± 0.04) cm^2 sr MeV
- H4: (0.76 ± 0.08) cm^2 sr MeV/nuc

Thus, the intensity I_x (in units of $(\text{cm}^2\ \text{sr s MeV/nuc})^{-1}$) for a given channel x is given by

$$I_x = \frac{1}{R_x} \cdot \frac{1}{t_{acc}} \sum_{n_x} w_{fact}(n) \tag{4}$$

with

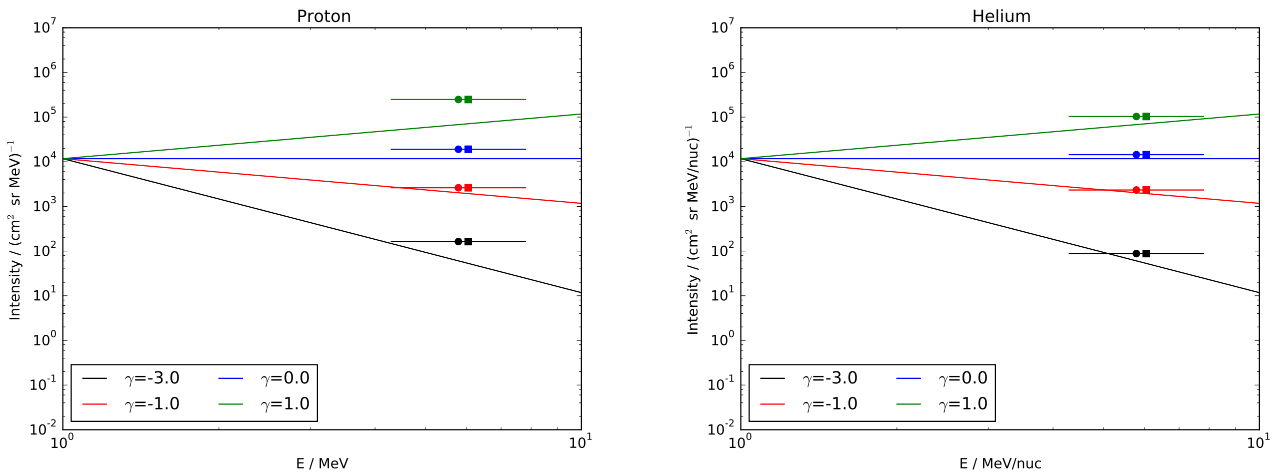

Figure 8: Simulated power-law spectra and the resulting intensities in the nominal channel (arithmetic and geometric means as squares and circles) for protons (left) and helium (right).

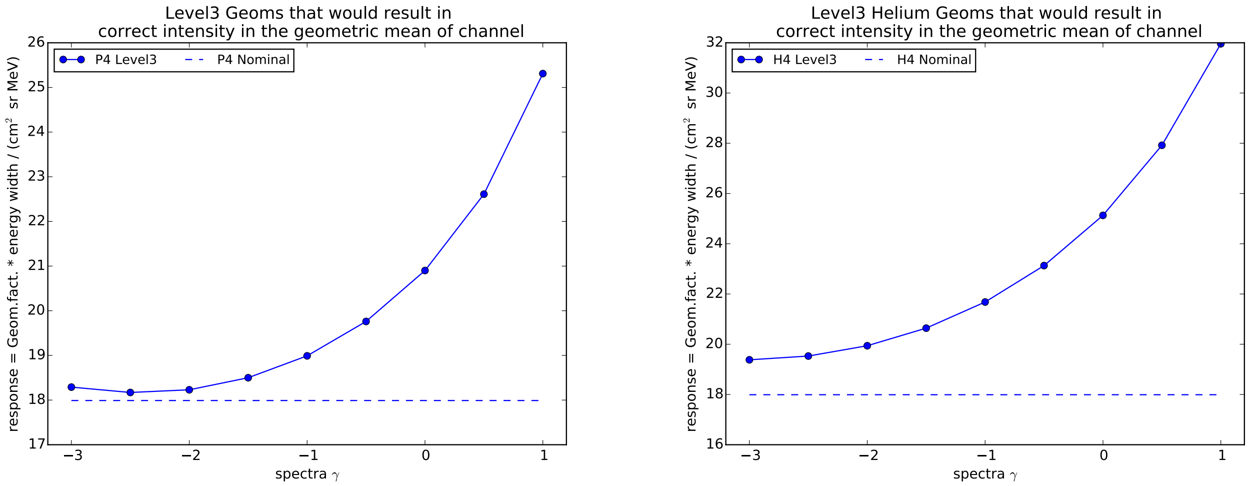

Figure 9: Proton (left) and helium (right) response factors that would result in the correct intensity in the geometric mean of channel.

- $\sum_{n_x}$: the sum over all PHA words in the related box (as defined above)
- $w_{fact}(n)$: ratio of total coincidence counts to number of PHA words for this minute and coincidence
- R_x: response as given above (in units of (cm^2 sr s MeV/nuc$)^{-1}$))
- t_{acc}: accumulation time (i.e. 59.953 seconds)

2.4 Comparison with Nominal Data Products

Figure 10 shows a comparison between Level3 and nominal intensities of the P4 and H4 channels, respectively. From the figures it is evident that the new level3 intensities are compareable to the nominal data product for both, protons and helium particles. The deviation between the H4 channels at higher intensities (i.e. nominal intensities are higher than the level3 intensities) can be explained by protons mistakenly being identified as helium particles in the nominal data product.

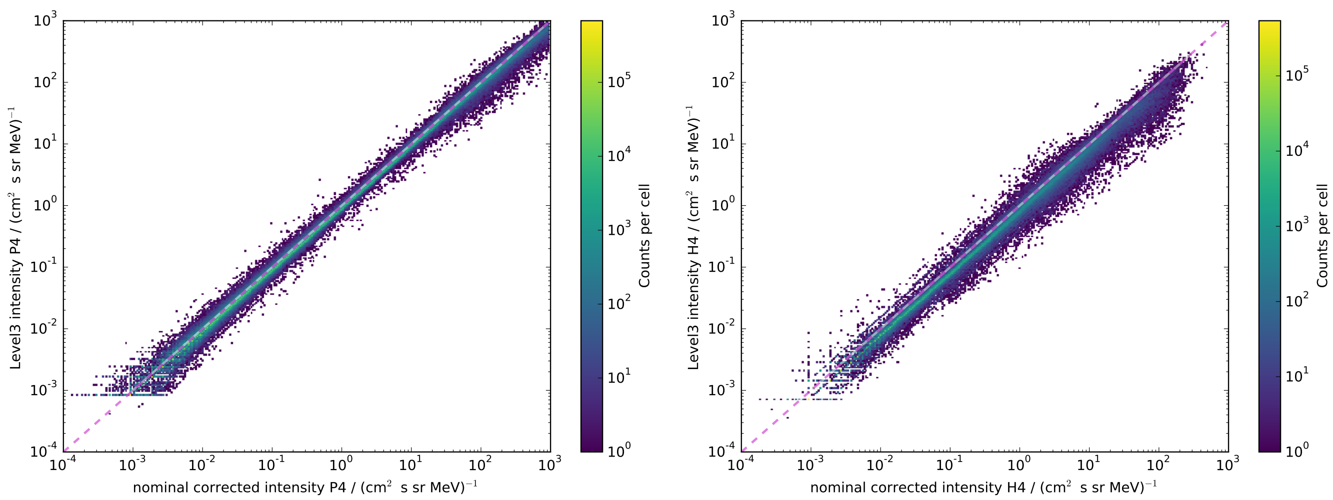

Figure 10: Comparison between nominal (Level2 data corrected using the *.kor files) and level3 intensities in the P4 (left) and H4 (right) channel for the entire mission.

3 A B C ¬F coincidence

3.1 Particle Identification

In this chapter we restrict our analysis to the coincidence $A \wedge B \wedge C \wedge \neg F$.
We define two quantities depending on E_A, E_B and E_C, the energy losses in detector A, B and C, respectively:

$$\kappa := (2 \cdot E_A - E_B)/(2 \cdot E_A + E_B) \tag{5}$$

and

$$\lambda := E_A + E_B + E_C \tag{6}$$

Figure 11 shows λ as a function of κ for more than 20 years of data. Clearly, the tracks of protons and helium can be seen as indicated by the text in the figure. Note that the positions of the particles is in agreement to estimations via the Bethe-Bloch equation, e.g. protons with energies from 8 to 25 MeV are depositing their entire energy in detectors A to C while the energy losses per pathlength in detector A and B are converging with higher energies (8 MeV protons are at the lower left (white square), protons with 25 MeV at the center on top of the proton population (white circle)). Protons with energies between 25 and 53 MeV are depositing an increasing amount of energy in detectors D and E and hence, according to the Bethe-Bloch equation, the energy loss in detectors A to C is decreasing (53 MeV protons deposit about 8.1 MeV in detectors A to C (white triangle)).
In order to define boxes, we can restrict our analysis to data around the proton and helium tracks as seen in figure 12. Based on these histograms, we can define thresholds (boxes) for proton and helium identification:

Proton selection criteria:

- -0.35 $< \kappa <$ 0.15
- 7.8 MeV $< \lambda <$ 27.5 MeV

Helium selection criteria:

- -0.35 $< \kappa <$ 0.15
- 29.5 MeV $< \lambda <$ 110 MeV

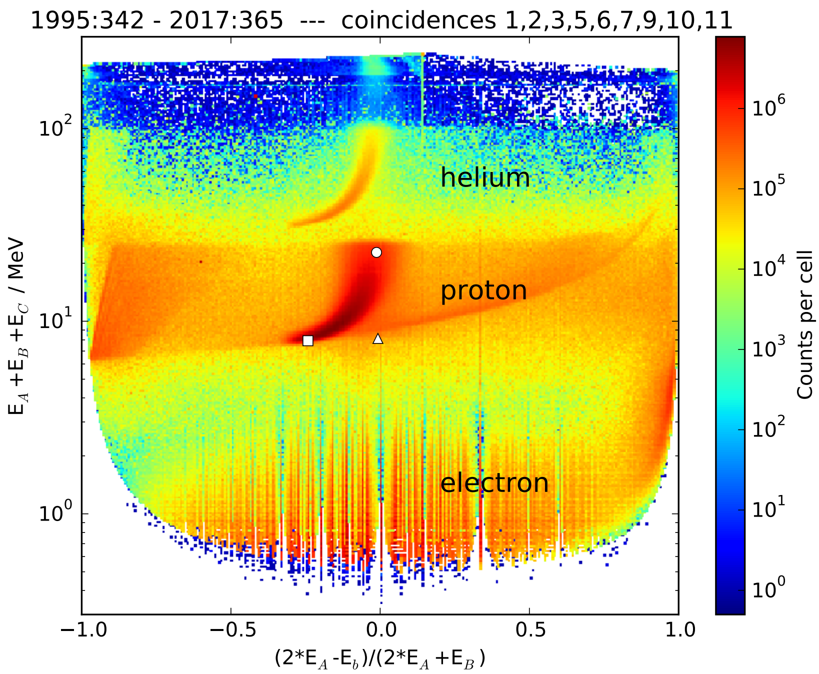

Figure 11: λ (equation 6) as function of κ (equation 5) for more than 20 years of data. The white square, circle and triangle represent calculations based on the Bethe-Bloch equation for protons with 8, 25 and 53 MeV, respectively.

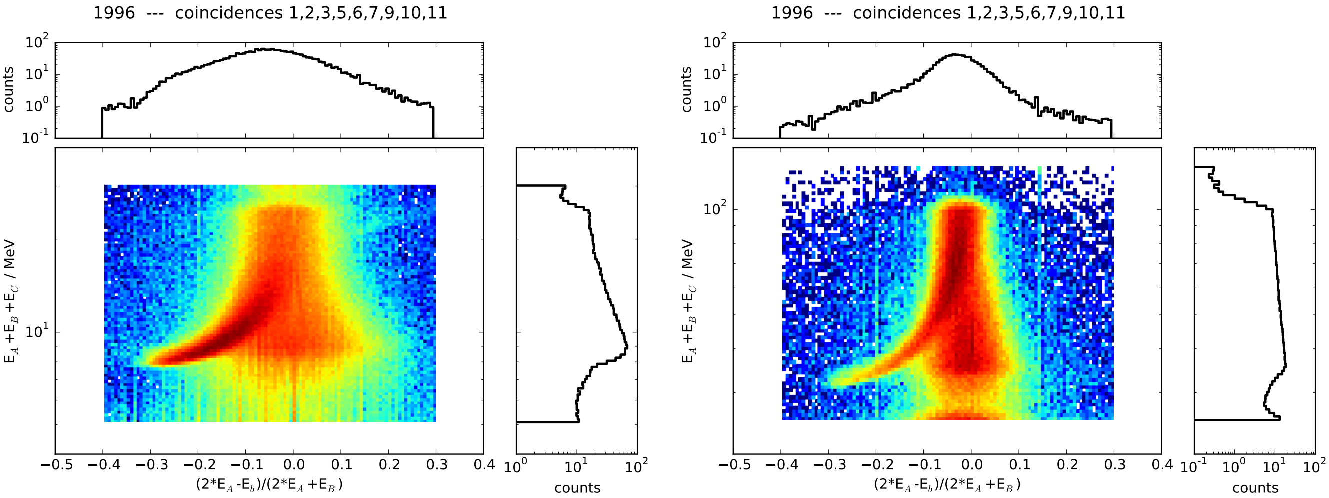

Figure 12: λ (equation 6) as a function of κ (equation 5) in 1996, restricted to data around the proton (left) and helium (right) track

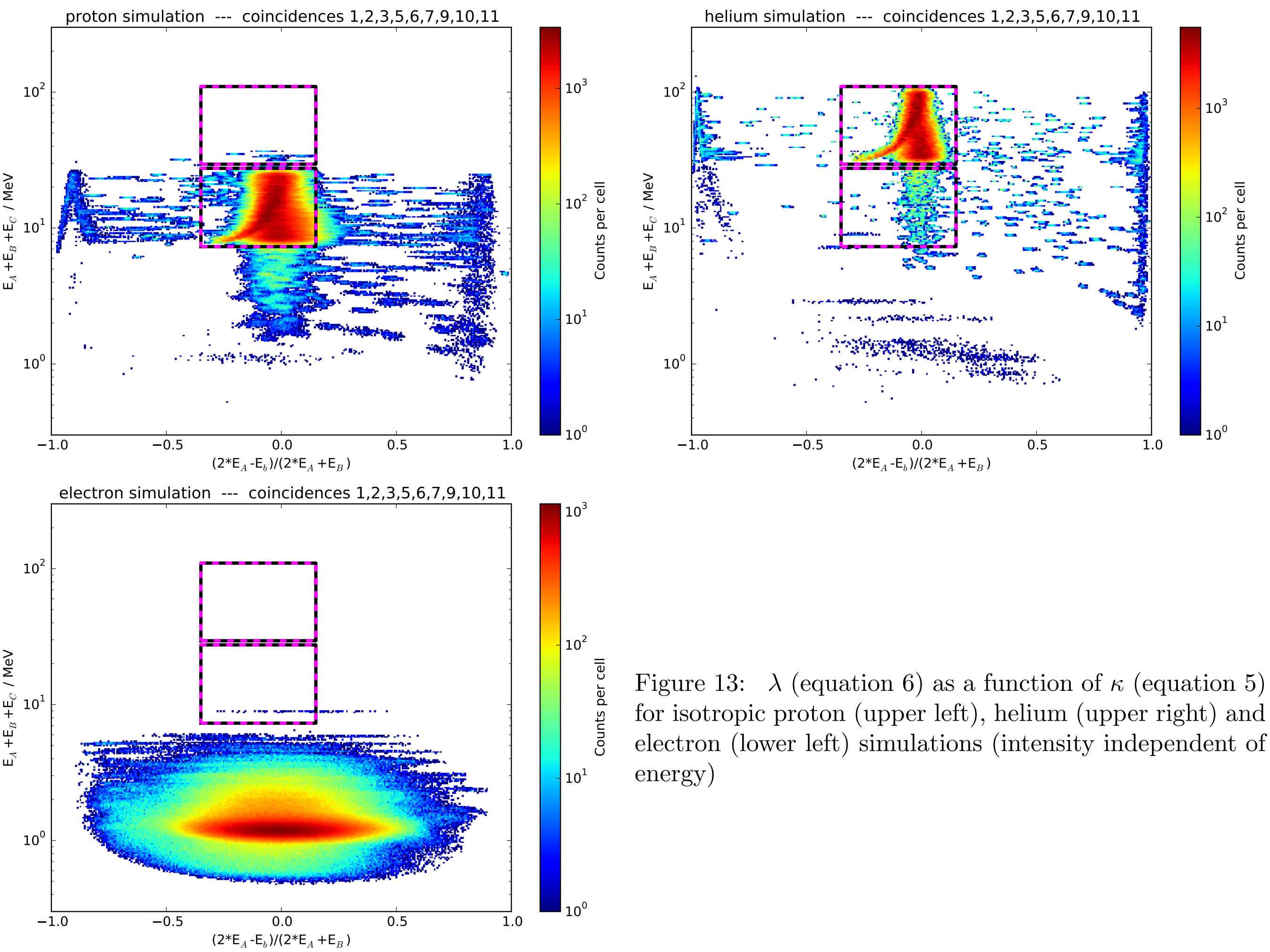

Figure 13: λ (equation 6) as a function of κ (equation 5) for isotropic proton (upper left), helium (upper right) and electron (lower left) simulations (intensity independent of energy)

Figure 13 shows λ as a function of κ for proton, helium and electron simulations, respectively. Simulations have been performed for isotropic fluxes with an energy independent intensity. The magenta-black boxes present the thresholds for proton and helium identification as defined above. From these results we can conclude that the majority of protons and helium particles are in their designated boxes while an electron contribution is surpressed.

3.2 Energy Determination

For the determination of the total kinetic energy of a measured particles, we define

$$\mu := E_A + E_B \tag{7}$$

Figure 14 shows the total energy (as simulated) as a function of the resulting energy deposit μ for protons (left) and helium (right). The black-magenta lines delimit the energy ranges of the channels P8, P25, P41 and H8, H25, H41, respectively.

Figure 15 shows the normalized histrograms of μ for protons and helium restricted to the nominal energy ranges of P8, P25, P41 and Int or H8, H25, H41 and Int., respectively (blue, green, red and teal lines). The intersection of the histrograms are used as thresholds between the different energy channels. In conclusion, boxes for κ and λ (equations 5 and 6) are used for the identification of proton and helium particles while the energy deposition μ (equation 7) is used in order to destinguish between different energy channels:

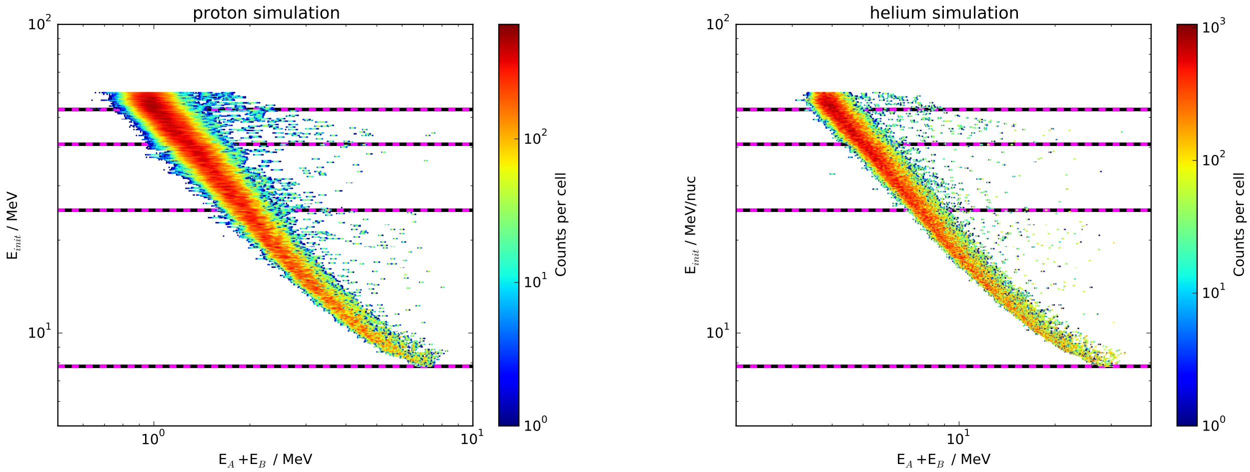

Figure 14: Total energy as function of the resulting energy deposit μ (equation 7) for an isotropic proton (left) and helium (right) simulation (intensity independent of energy)

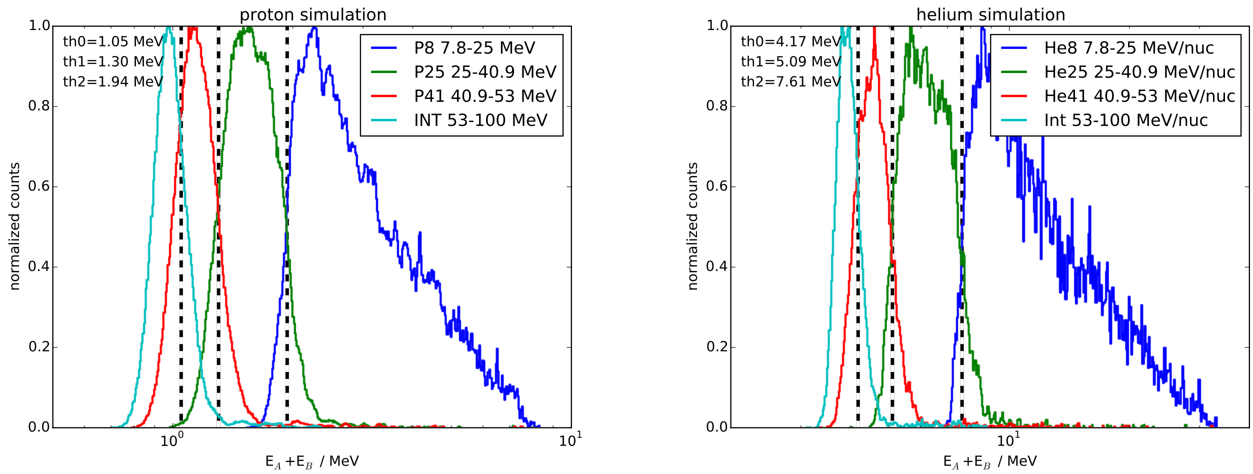

Figure 15: Normalized histograms of the resulting energy deposit μ for an isotropic proton (left) and helium (right) simulation (intensity independent of energy) for the nominal energy ranges of P8, P25, P41 and H8, H25, H41, respectively (blue, green, red and teal lines).

Proton selection criteria:

- -0.35 < κ < 0.15
- 7.8 MeV < λ < 27.5 MeV
- energy thresholds:
  - P8: μ > 1.94 MeV
  - P25: 1.94 MeV > μ > 1.30 MeV
  - P41: 1.30 MeV > μ > 1.05 MeV

Helium selection criteria:

- -0.35 < κ < 0.15
- 29.5 MeV < λ < 110 MeV
- energy thresholds:
  - H8: μ > 7.61 MeV
  - H25: 7.61 MeV > μ > 5.09 MeV
  - H41: 5.09 MeV > μ > 4.17 MeV

3.3 Response Factor

In order to calculate the intensity of a given particle type from the measured count rate, the geometry factor has to be calculated. The energy dependent geometry factor is also called response function. In order to calculate differential intensities (i.e. in units of $(cm^2\ sr\ s\ MeV)^{-1}$), the energy range of a channel has to be taken into account as well. This factor can be interpreted as product of geometry factor and energy width of a channel or - for a non-ideal detector - as integration of the response function. In the following, the corresponding factor is called response factor.
The response functions for protons and helium are shown in figure 16 for the new Level3 selection as well as the nominal responses. As expected and well known, the nominal responses are almost ideal box-shaped for both protons and helium. While similar in the total amplitude, the responses of the Level3 selection defined in the previous chapter show a significant higher overlap between the different channels due to the energy determination based on thresholds in μ (equation 7, see e.g. figure 15). In the following, the response functions for the level3 selection have to be quantified.

To illustrate common issues with the response of channels with rather wide energy coverage, figure 17 shows several simulated power-laws (solid lines) as well as the resulting intensities in the nominal proton (left) and helium (right) channels. For an ideal detector, the intensity of a channel would agree with the intensity of the simulated spectra at a given energy, independent on the spectral shape. However, due to the broad energy range of the channels the exact energy at which the channel intensities equals the one of the input spectra changes as a function of the spectral shape (i.e. the power-law index γ). Due to the finite responses of the channels to energies lower/higher than the ideal response (see tails in the nominal response function in figure 16) the channel intensities can be even lower/higher than the minimum/maximum intensities simulated in the nominal energy range.

In order to account for this issue and to calculate valid response factors and reasonable systematic uncertainties, simulations with several power-laws have been evaluated with the Level3 selection criteria as defined above. Based on the counts in these artificial data sets, the response factors that would result in intensities in the level3 selection that would match the simulated intensities at the geometric mean of the energy in the channel have been calculated. Figure 18 shows these respons factors for proton (left) and helium (right) simulations as a function of the power-law index γ. Reasonable values for the response factors and their systematic uncertainties are than given by the mean as well as the standard deviation of these factors. The factors are also calculated for the ring-off mode (i.e. only the inner segments of A and B are active).

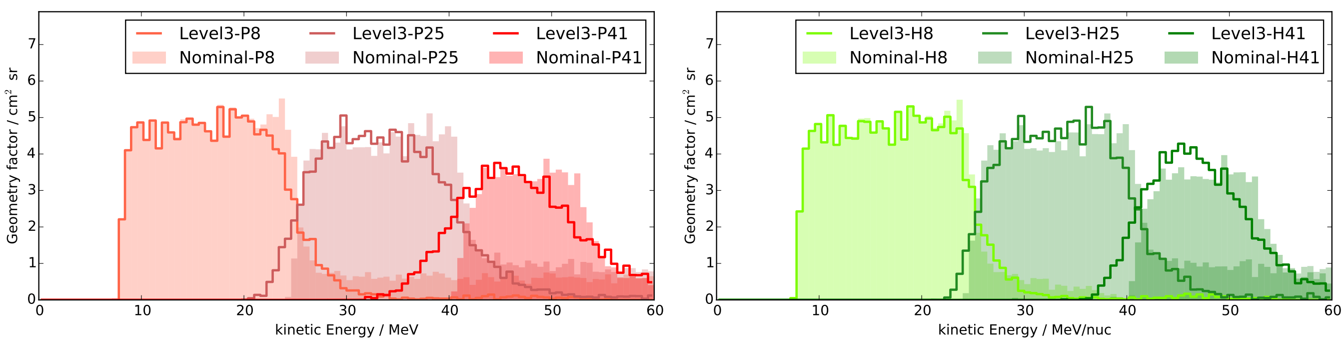

Figure 16: Nominal and Level3 response functions for protons (left) and helium (right).

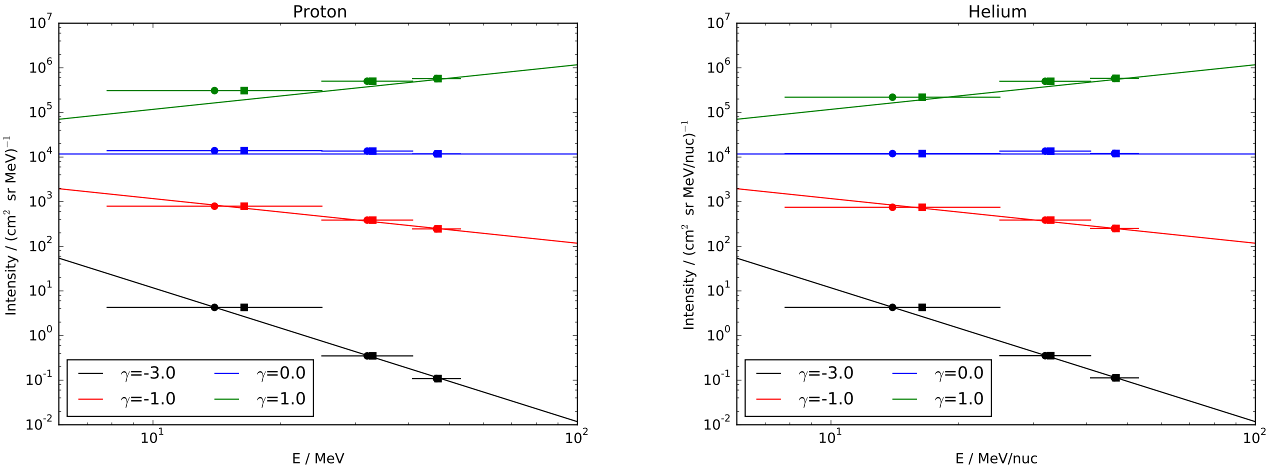

Figure 17: Simulated power-law spectra and the resulting intensities in the nominal channels (arithmetic and geometric means as squares and circles) for protons (left) and helium (right).

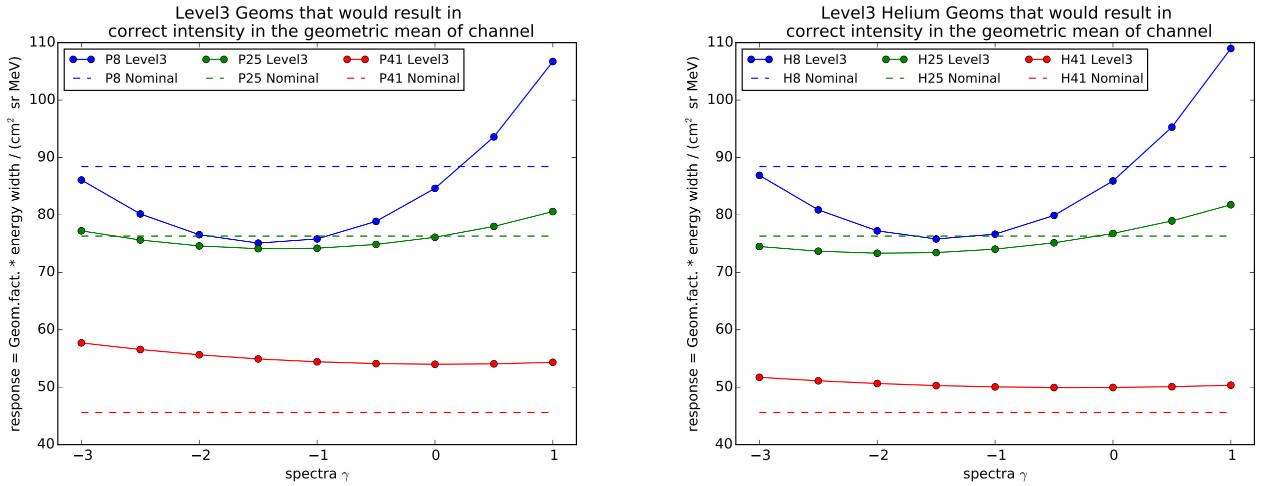

Figure 18: Proton (left) and helium (right) response factors that would result in the correct intensity in the geometric mean of channel.

Response Factor in nominal observation mode:

- P8: (84.17 ± 9.75) cm^2 sr MeV
- P25: (76.15 ± 2.01) cm^2 sr MeV
- P41: (55.08 ± 1.23) cm^2 sr MeV
- H8: (85.28 ± 10.23) cm^2 sr MeV
- H25: (75.74 ± 2.75) cm^2 sr MeV
- H41: (50.46 ± 0.57) cm^2 sr MeV

Response Factor in ring-off mode (i.e. only the inner segments of A and B are active):

- P8: (3.03 ± 0.31) cm^2 sr MeV/nuc
- P25: (2.75 ± 0.22) cm^2 sr MeV/nuc
- P41: (2.07 ± 0.19) cm^2 sr MeV/nuc
- H8: (2.94 ± 0.26) cm^2 sr MeV/nuc
- H24: (2.72 ± 0.17) cm^2 sr MeV/nuc
- H41: (1.86 ± 0.19) cm^2 sr MeV/nuc

Thus, the intensity I_x (in units of (cm^2 sr s MeV/nuc)$^{-1}$) for a given channel x is given by

$$I_x = \frac{1}{R_x} \cdot \frac{1}{t_{acc}} \sum_{n_x} w_{fact}(n) \tag{8}$$

with

- $\sum_{n_x}$: the sum over all PHA words in the related box (as defined above)
- $w_{fact}(n)$: ratio of total coincidence counts to number of PHA words for this minute and coincidence
- R_x: response as given above (in units of $(cm^2 \text{ sr s MeV/nuc})^{-1}$))
- t_{acc}: accumulation time (i.e. 59.953 seconds)

3.4 Comparison with Nominal Data Products

Figure 19 shows comparisons between Level3 and nominal intensities of the P8 and H8 channels, respectively. From the figures it is evident that the new level3 intensities are compareable to the nominal data product for the protons. The deviation between the H8 channels (i.e. nominal intensities are higher than the level3 intensities) can be explained by protons mistakenly being identified as helium particles in the nominal data product. The level3 intensities of higher energy channels (P25, P41, H25 and H41) are not compared to the nominal data products due to limited amount of data available for the nominal data products (the instrument was switched into failure mode E as early as October 31, 1996).

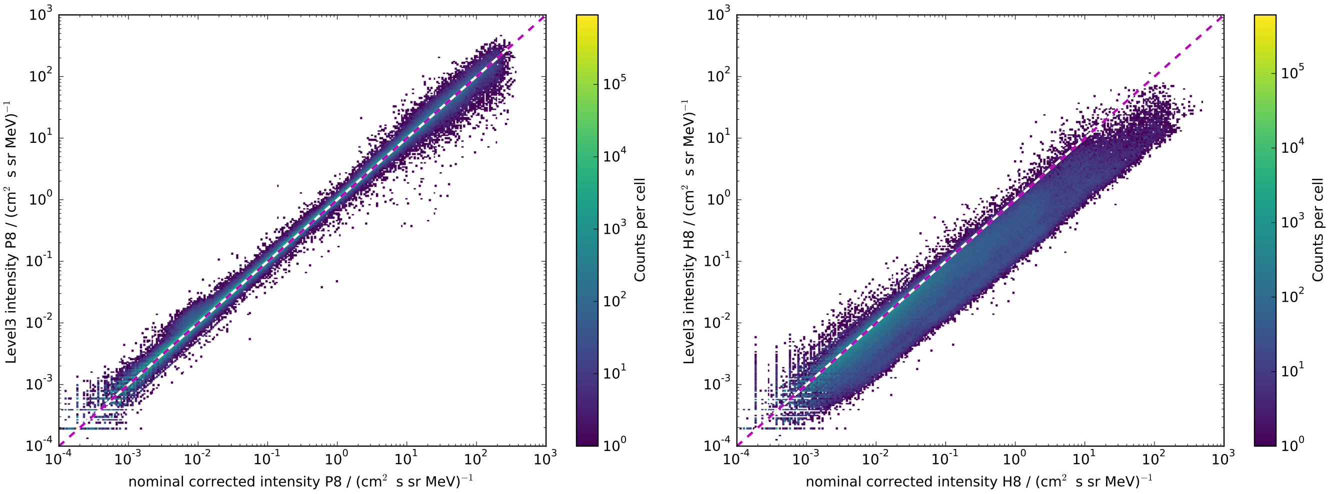

Figure 19: Comparison between nominal (Level2 data corrected using the *.kor files) and level3 intensities in the P8 (left) and H8 (right) channel for the entire mission.

4 Comparison with Data from other Missions

In order to validate the level3 intensities, comparisons to several instrument with similar energy ranges have been performed. These comparisons include all proton and helium channels for several different time resolutions (from 1 minute up to 1 hour). The reference measurements are from GOES/EPS (channels P4, P8), ACE/SIS (H4, H8, H25) and SOHO/ERNE (P25, P41, H25, H41).

4.1 GOES/EPS (P4, P8)

Figures 20 and 21 show comparisons of the Level3 P4 intensity with the GOES11/EPS intensities in a similar energy window for 1 and 5 minute resolution, respectively. The figures on the left hand side show a direct comparison between the intensities, those on the right hand side shows the differences of both intensities divided by the uncertainties of the EPHIN data. Figures 22 and 23 show the same analysis for the P8 channel in comparison the GOES11/EPS intensities in a slightly lower energy window. Data has been taken from the entire GOES11 mission starting in 2000 till 2011 with several data gaps. In addition to an overall agreement, the significant better sensitivity (lower background) for EPHIN is clearly pronounced.

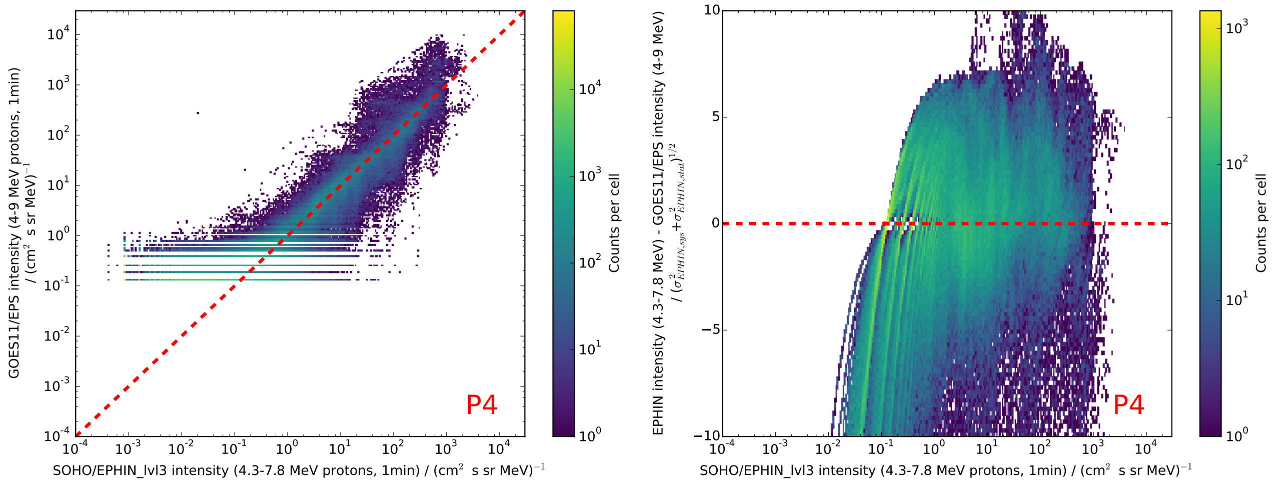

Figure 20: Comparison between Level3 P4 intensity and GOES11/EPS proton intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

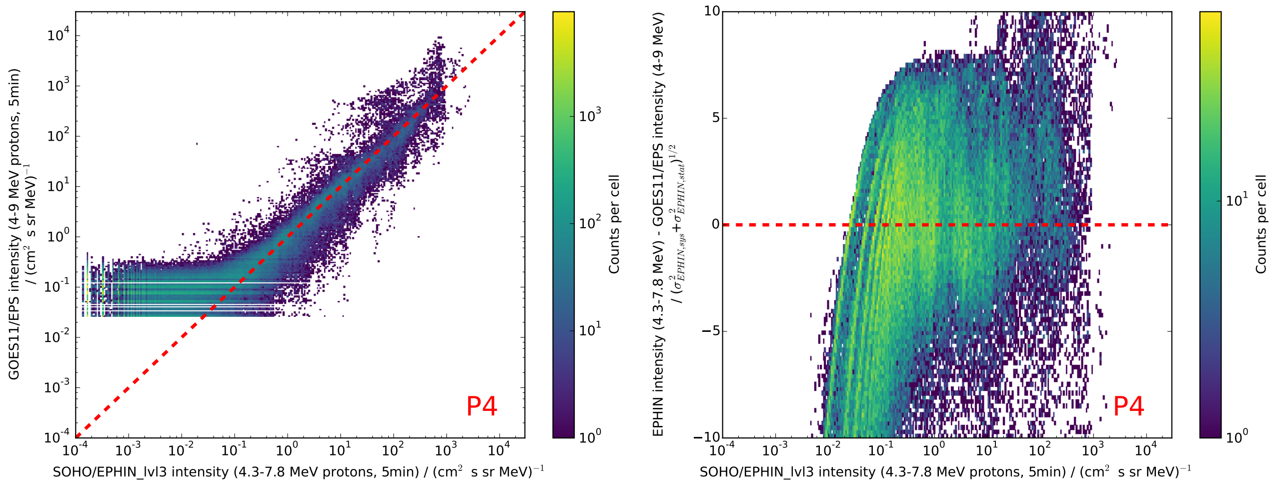

Figure 21: Comparison between Level3 P4 intensity and GOES11/EPS proton intensities in a similar energy window (5 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

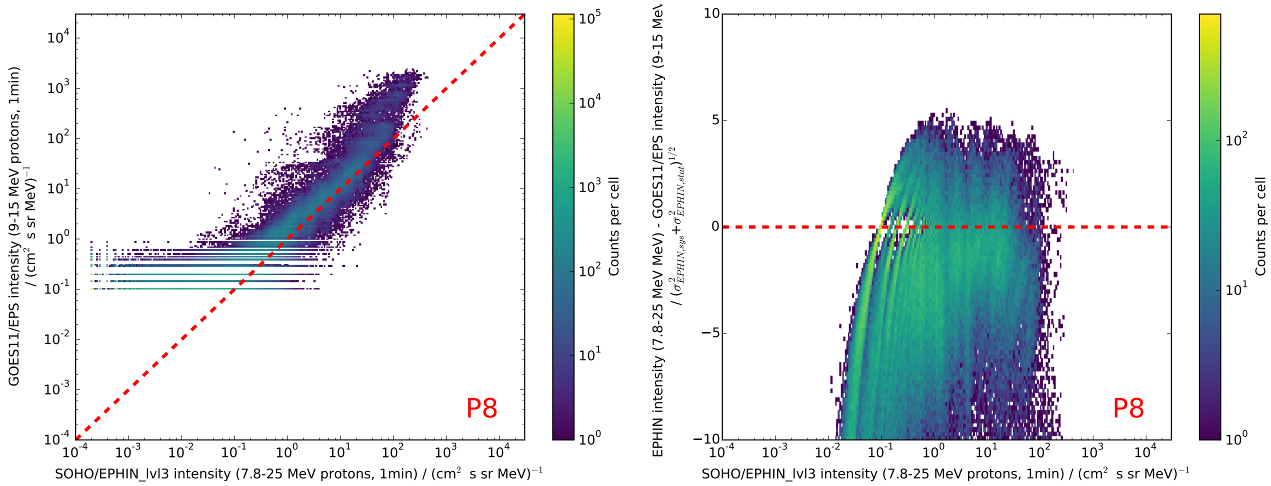

Figure 22: Comparison between Level3 P8 intensity and GOES11/EPS proton intensities in a slightly lower energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

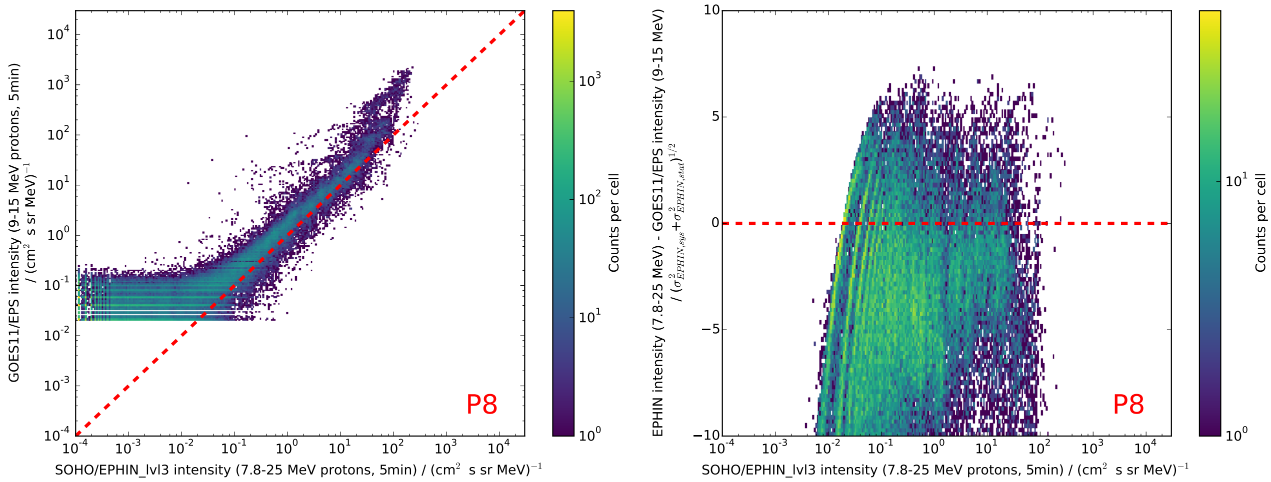

Figure 23: Comparison between Level3 P8 intensity and GOES11/EPS proton intensities in a slightly lower energy window (5 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

4.2 ACE/SIS (H4, H8, H25)

Figures 24 till 27 show comparisons of the Level3 H4, H8 and H25 hourly averaged intensities with ACE/SIS Helium intensities in similar energy windows, respectively. The figures on the left hand side show a direct comparison between the intensities, those on the right hand side shows the differences of both intensities divided by the uncertainties of the EPHIN data. Data has been taken from 1997 till 2018 for figures 24 till 26 while figure 27 shows data untill end of 2015. Both instruments show remarkable similar intensities although it has to be noted that the comparison below 10^{-4} (cm^2 s sr MeV/nuc)$^{-1}$ is limited by statistical uncertainties.

4.3 SOHO/ERNE (P25, P41, H25, H41)

Figures 28 till 33 show comparisons of the Level3 P25, P41, H25 and H41 intensities with the SOHO/ERNE intensities in a similar energy window for 1 minute resolution, respectively. Data has been taken from 1996 (starting at day of year 152 due to ERNE caveats) till the end of 2000. In addition to an overall agreement, a dead-time issue for ERNE is clearly pronounced.

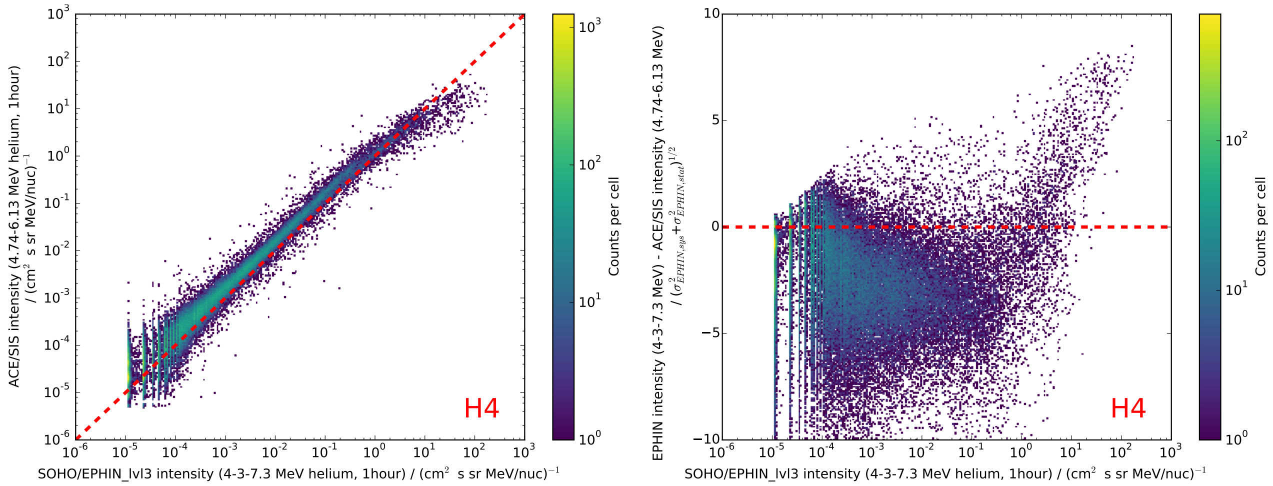

Figure 24: Comparison between Level3 H4 intensity and ACE/SIS Helium intensities in a similar energy window (1 hour time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

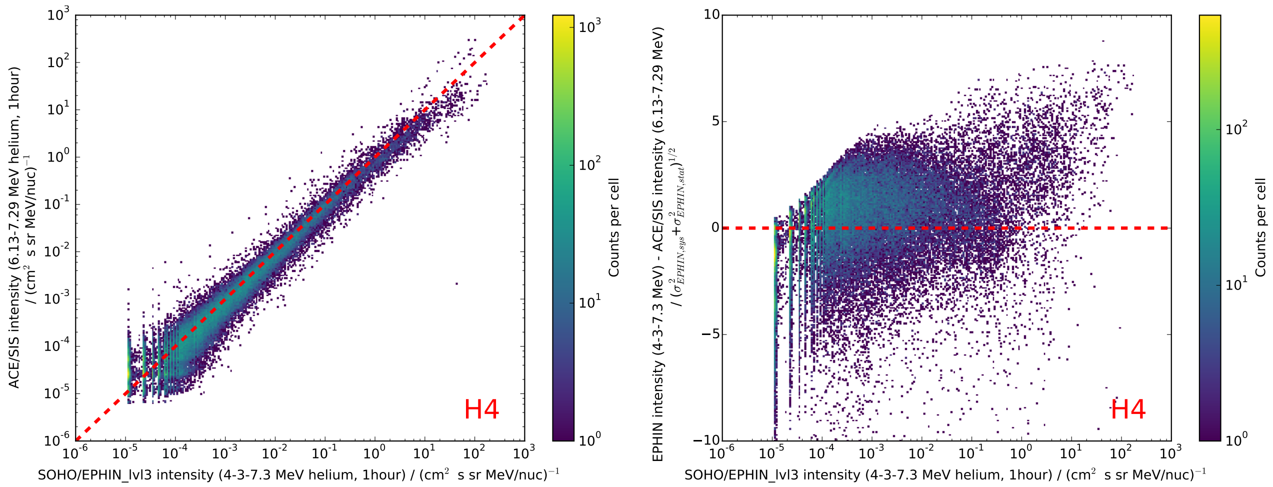

Figure 25: Comparison between Level3 H4 intensity and ACE/SIS Helium intensities in a similar energy window (1 hour time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

4.4 Comparison results

Figure 34 shows the differences of intensities between EPHIN and other missions in units of EPHIN uncertainties (red). The black curves indicate normal distributions with 1, 2 and 3 σ respectively. Since all measurements are comparable to these distributions, the uncertainties as derived in sections 2 and 3 are in the correct order of magnitude.

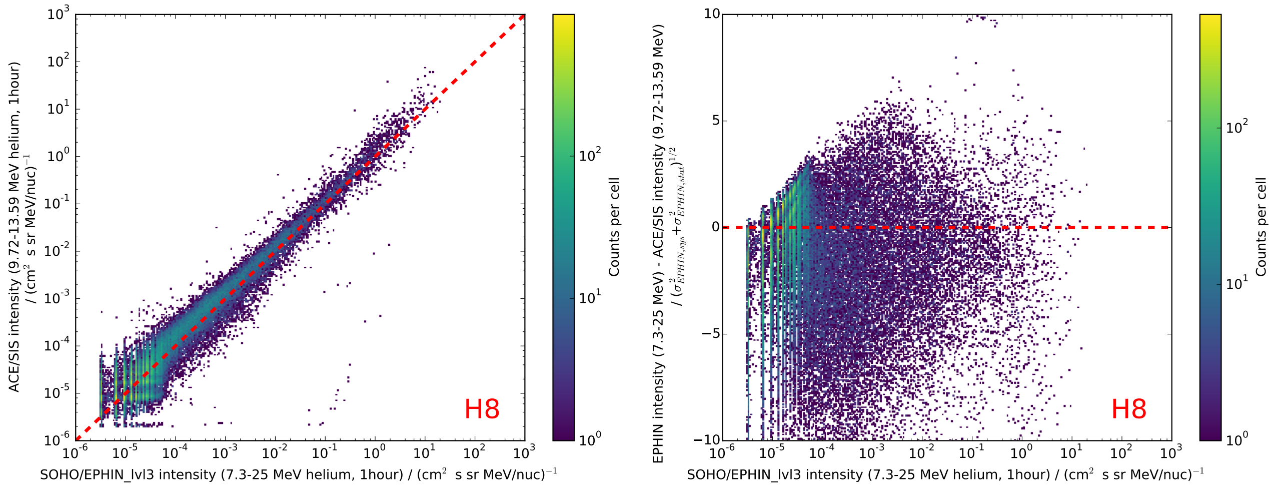

Figure 26: Comparison between Level3 H8 intensity and ACE/SIS Helium intensities in a similar energy window (1 hour time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

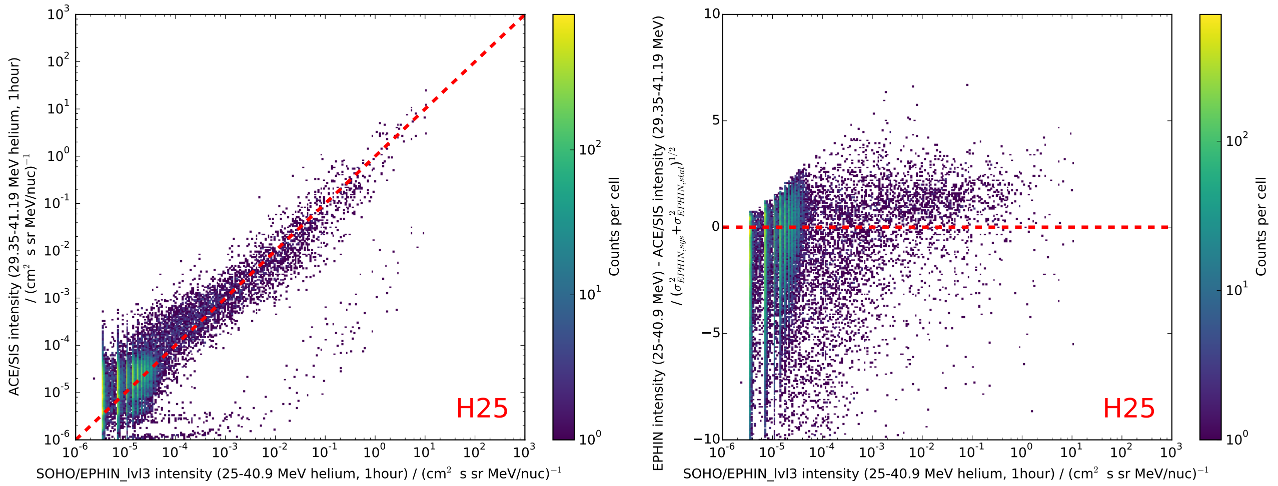

Figure 27: Comparison between Level3 H25 intensity and ACE/SIS Helium intensities in a similar energy window (1 hour time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

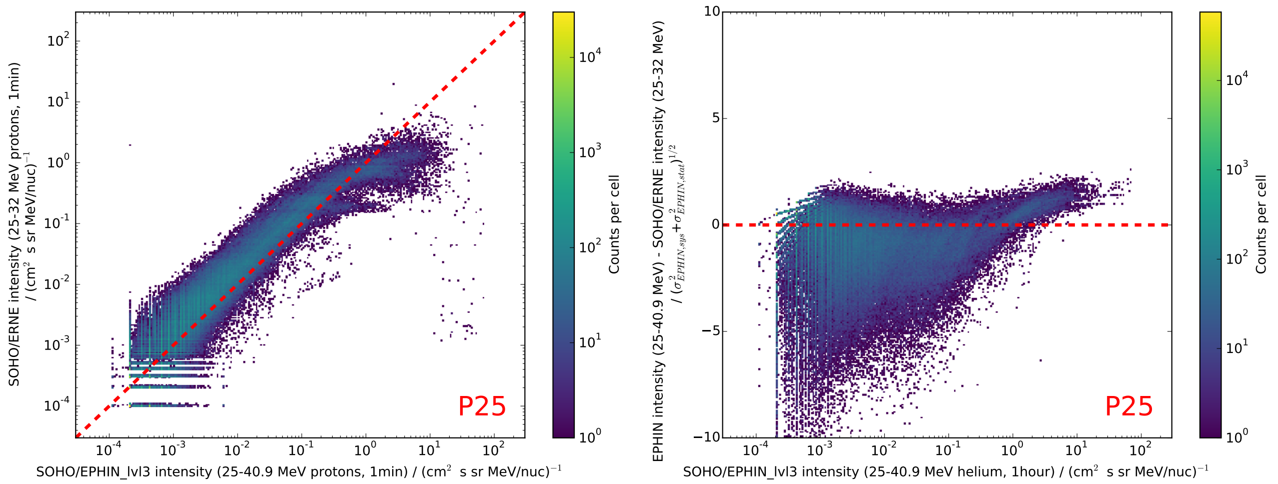

Figure 28: Comparison between Level3 P25 intensity and SOHO/ERNE proton intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

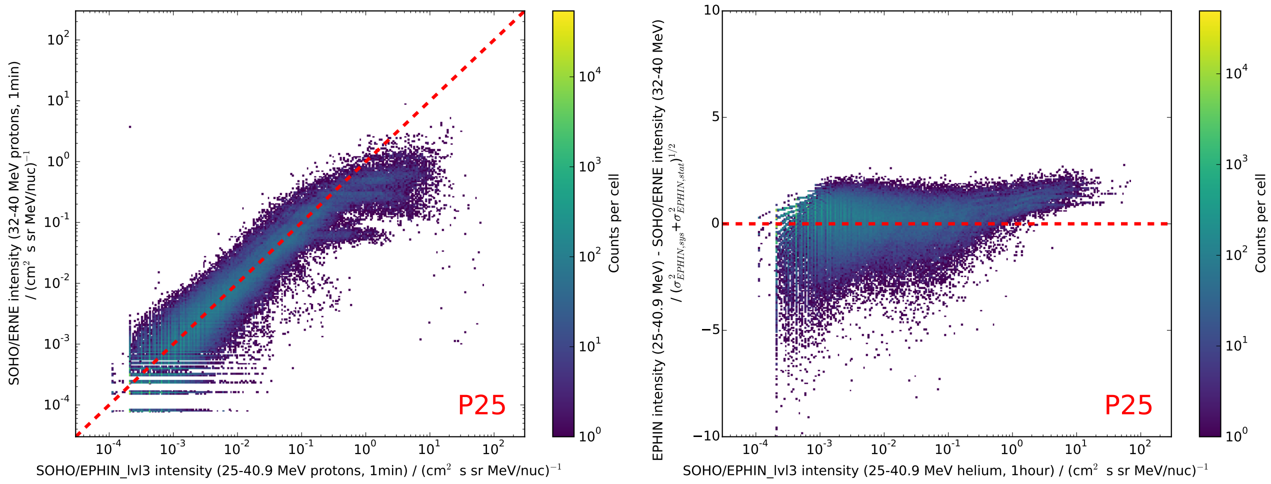

Figure 29: Comparison between Level3 P25 intensity and SOHO/ERNE proton intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

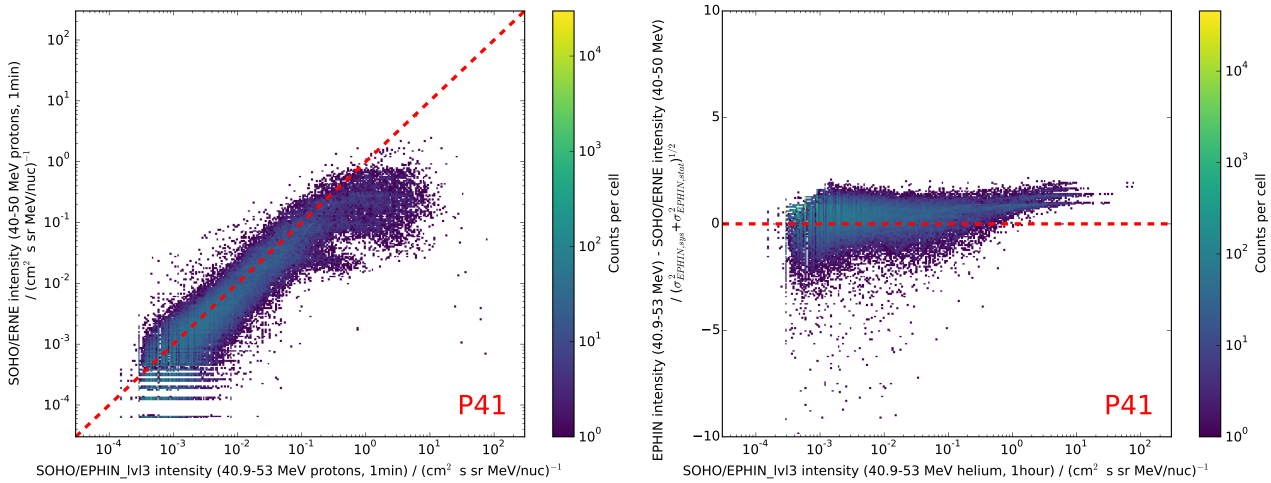

Figure 30: Comparison between Level3 P41 intensity and SOHO/ERNE proton intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

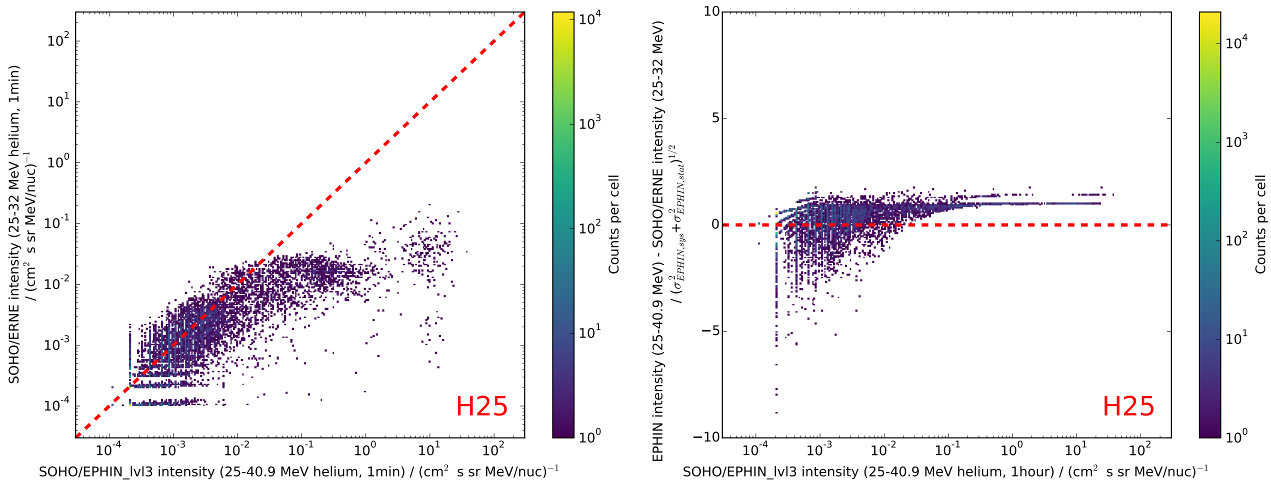

Figure 31: Comparison between Level3 H25 intensity and SOHO/ERNE helium intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

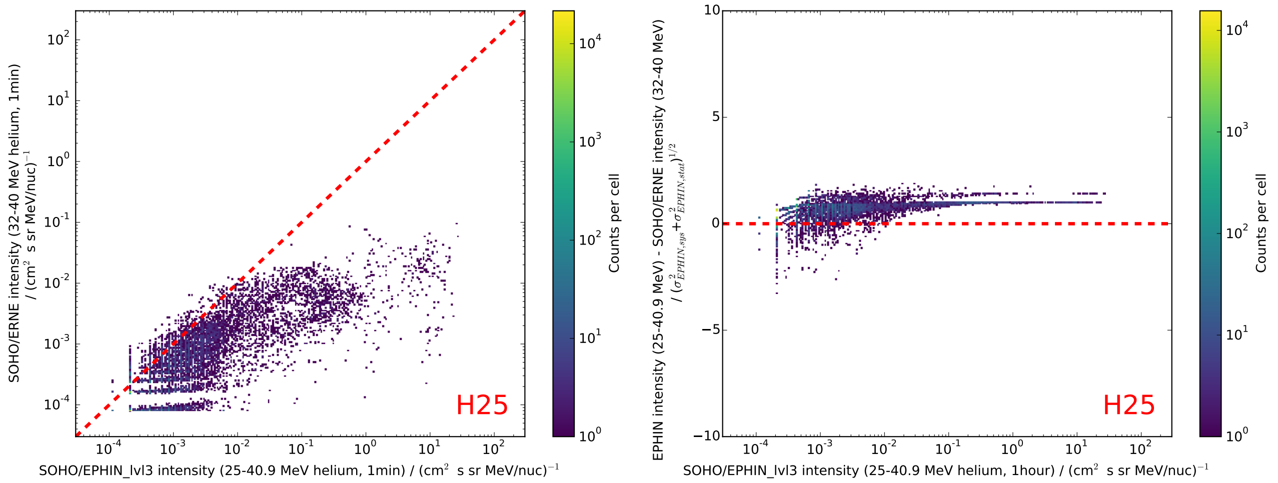

Figure 32: Comparison between Level3 H25 intensity and SOHO/ERNE helium intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

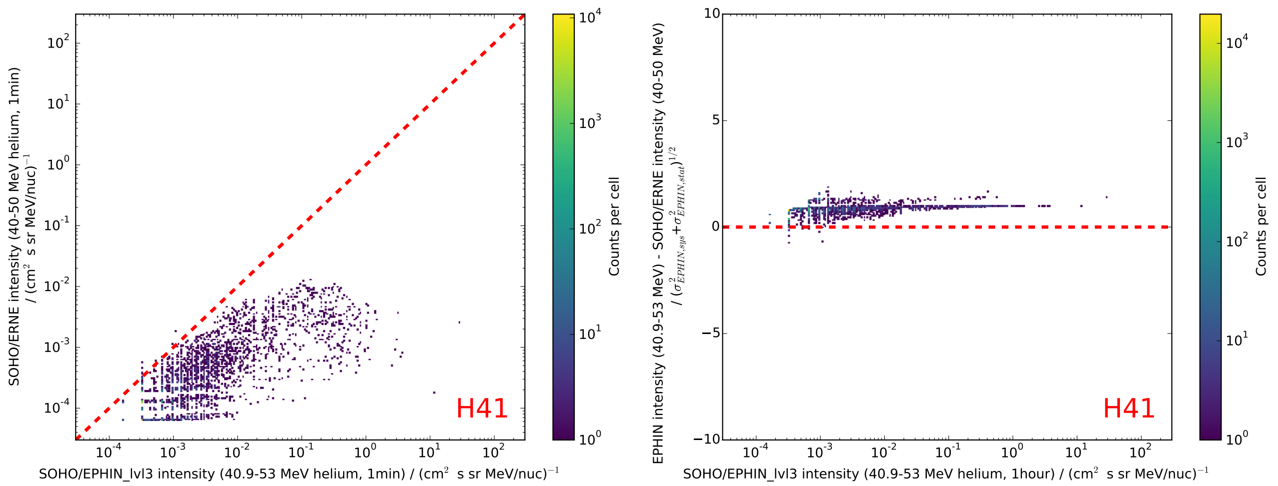

Figure 33: Comparison between Level3 H41 intensity and SOHO/ERNE helium intensities in a similar energy window (1 minute time resolution). On the right hand side, the deviation between both data sets are shown in units of the Level3 uncertainties.

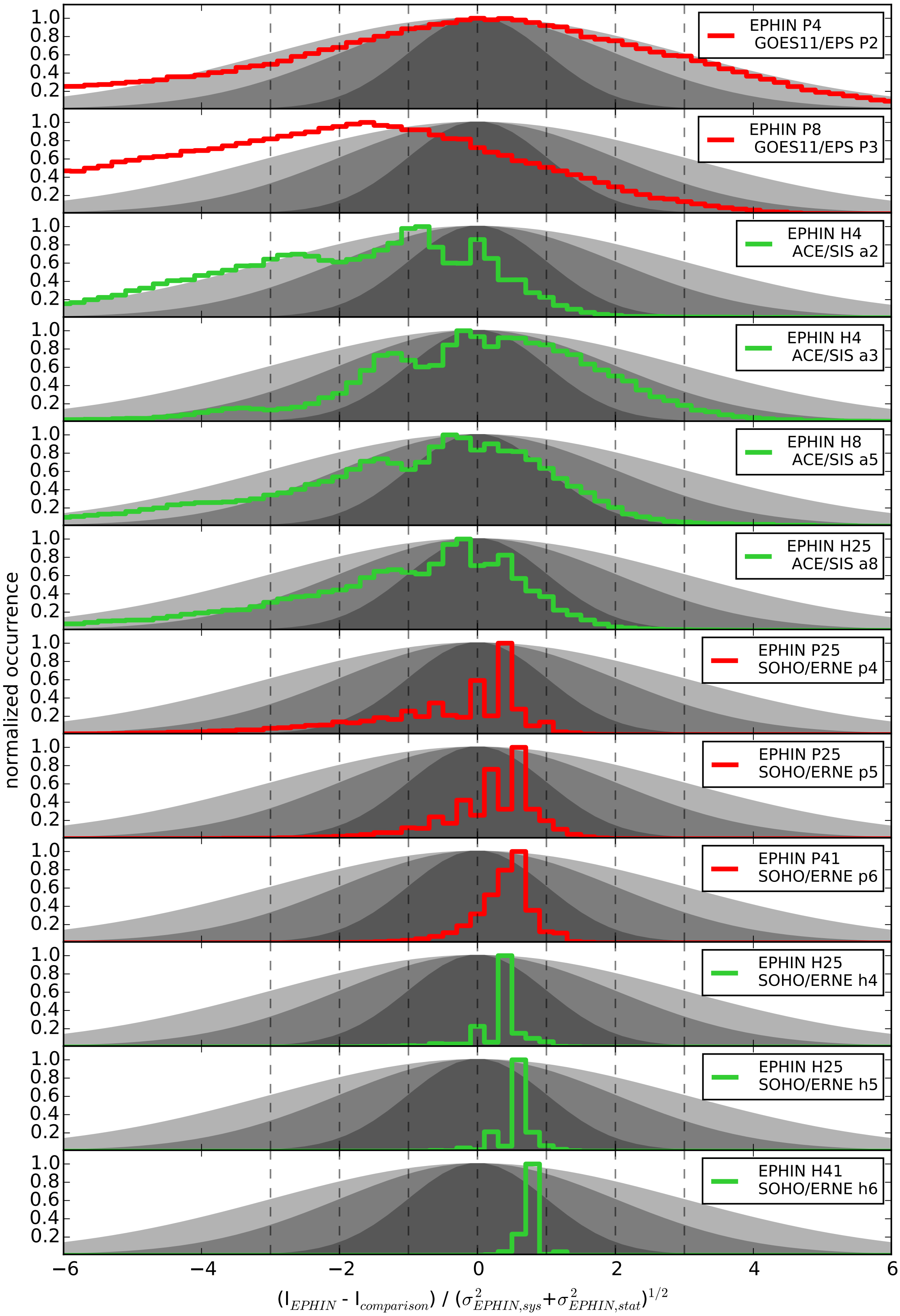

Figure 34: Differences of intensities between EPHIN and other missions in units of EPHIN uncertainties (protons in red, helium in green). The black curves indicate normal distributions with 1, 2 and 3 σ respectively.

5 Data Product

The created Level3 intensity files will be provided in different time resolutions as ASCII text files: 1, 5, 10, 30, 60 and 1440 minutes. The format of the data product is given in table 1. Note that

- the time given in the data set marks the beginning of the time interval
- the statistical and systematic uncertainties of a given channel are set to '-999' if the channel has zero counts in a time interval (the intensity will be '0' though)
- 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 status flag is a decimal code which results from the summation of the flag bit values given in table 2
- the energy ranges of the different channels are given in table 3

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	status	status flag	binary status word	int
8	accum.time	accumulation time	seconds	4.4f
9	int_p4	proton intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
10	int_p8	proton intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
11	int_p25	proton intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
12	int_p41	proton intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
13	sys_p4	proton systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
14	sys_p8	proton systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
15	sys_p25	proton systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
16	sys_p41	proton systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
17	stat_p4	proton statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
18	stat_p8	proton statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
19	stat_p25	proton statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
20	stat_p41	proton statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
21	int_h4	helium intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
22	int_h8	helium intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
23	int_h25	helium intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
24	int_h41	helium intensity	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
25	sys_h4	helium systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
26	sys_h8	helium systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
27	sys_h25	helium systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
28	sys_h41	helium systematic uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
29	stat_h4	helium statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
30	stat_h8	helium statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
31	stat_h25	helium statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e
32	stat_h41	helium statistical uncertainty	$(cm^2$ s sr Mev/nuc$)^{-1}$	4.4e

Table 1: Explaination of the data product of Level3 intensities.

Flag Bit Value	Remarks
0	Nominal observation, i.e. High Voltage ON, no failure mode, ring segment switching disabled
1	Failure Mode E
2	Ring A/B OFF
4	E patch uploaded
8	Commissioning
16	Standby or maintenance, i.e. High Voltage OFF
32	Calibration, i.e. test mode
64	Automatic ring switch enable
128	Failure Mode D

Table 2: EPHIN status flag description (source: 'ephispec.doc')

channel	min energy (MeV/nuc)	max energy (MeV/nuc)	arithmetic mean energy (MeV/nuc)	geometric mean energy (MeV/nuc)
P4	4.3	7.8	6.05	5.79
P8	7.8	25	16.15	13.51
P25	25	40.9	32.95	31.98
P41	40.9	53	46.95	46.56
H4	4.3	7.8	6.05	5.79
H8	7.8	25	16.15	13.51
H25	25	40.9	32.95	31.98
H41	40.9	53	46.95	46.56

Table 3: Energy ranges of the Level3 channels.

The data structure of the Level3 files for the SOHO archive at GSFC and ESAC is as follows:

- one mission-long file per resolution (1, 5, 10, 30, 60 and 1440 minutes).

The internal directory structure of the Level3 files is as follows:

- the main folder contains sub-directories for all time resolutions (e.g. 1, 5, 10, 30, 60 and 1440 minutes)
- all time resolution sub-directories have further sub-directories for each year which contain daily files
  - e.g. for 1 minute time resolution, year 2017 and day of year 1: main_directory/1min/2017/2017_001.l3i
- all time resolution sub-directories contain also annual files
  - e.g. for 10 minute time resolution, year 2001: main_directory/10min/2001.l3i
- the time resolution sub-directories for 60 and 1440 minutes also contain files for the entire mission
  - e.g. for 60 minute time resolution: main_directory/60min/entire_mission_60min.l3i

6 Code

All function required to create the Level3 intensity files (as defined in section 5) are defined in the file

level3_funcs.py

The entire code can be found in section 7, an example of how to apply the code in order to set-up automatic data production can be found in section 8. In the following, an explanation of all defined functions is given:

load_level2_pha Loads Level2 PHA data

load_level1_sci Loads Level1 SCI data

check_coinc Checks for and deletes wrong coincidences in the PHA

add_wfact_to_pha Synchronizes Level1 SCI and Level2 PHA data. Adds ratio of total counts and number of PHA words as well as the status word to PHA files (so-called PHAWS files)

phaws_from_year_doy Creates the PHAWS file for a given year and doy

extract_lvl3_geoms_ab Extracts the Level3 Geometry factors for AB coincidences from the Geometry file

counts_in_lvl3_ab_ch_from_ea_eb Calculates valid counts in AB coincidences

int_in_lvl3_ch_from_ea_eb Calculates counts/(cm^2 sr MeV) for AB

extract_lvl3_geoms_abc Extracts the Level3 Geometry factors for ABC coincidences from the Geometry file

counts_in_lvl3_ch_from_ea_eb_ec Calculates valid counts in ABC coincidences

int_in_lvl3_ch_from_ea_eb_ec Calculates counts/(cm^2 sr MeV) for ABC

calc_lvl3_intensities_timeresolution Calculates complete Level3 intensity files for a given timeresolution (in minutes)

merge_level3_daily_to_annual Merges daily files of a given time resolution to annual files

Furthermore, the code requires a set of paths (input and output) that have to be defined either in 'level3_funcs.py' or in a given executable script:

lvl2_pha_path Location of the Level2 PHA data set (e.g. '/data/missions/soho/costep/level2/pha/')

lvl1_sci_path Location of the Level1 SCI data set (e.g. '/data/missions/soho/costep/level1/sci/')

geompath Location of the Geometry factor files (cf. section 9)

phaws_path Storage location for PHAWS files (can be temporarily). Note: PHAWS is a combined dataset of PHA and SCI information that is created during the calculation of the Level3 intensities.

lvl3_out_path Output location for the Level3 intensity files

7 Appendix I: level3_funcs.py

```python
# This scripts includes all function necessary in order to derive EPHIN lvl3 ion intensities
# Patrick Kuehl, June 7 2018 kuehl@physik.uni-kiel.de
#from pylab import *
import numpy as np
import numpy.ma as ma
import time as time
import datetime as dt
import os
np.seterr(divide='ignore', invalid='ignore')
import subprocess
import warnings
warnings.filterwarnings("ignore")

# sections and defined functions
"""
functions for the PHAWS data processing
    load_level2_pha(year,doy,unpack=False)
    load_level1_sci(year,doy)
    check_coinc(co,a,b,c,d,e)
    add_wfact_to_pha ( year , doy )
    phaws_from_year_doy ( year , doy , save = True )

level3 AB-coincidence functions
    extract_lvl3_geoms_ab()
    counts_in_lvl3_ab_ch_from_ea_eb(ea,eb)
    int_in_lvl3_ch_from_ea_eb(ea,eb, mywfact,myringoff,myaseg,mybseg, i_p_ron,s_p_ron,i_h_ron,s_h_ron, i_p_roff,s_p_roff,i_h_roff,s_h_roff)

level3 ABC-coincidence functions
    extract_lvl3_geoms_abc()
    counts_in_lvl3_ch_from_ea_eb_ec(ea,eb,ec)
    int_in_lvl3_ch_from_ea_eb_ec(ea,eb,ec, mywfact,myringoff,myaseg,mybseg, i_p_ron,s_p_ron,i_h_ron,s_h_ron, i_p_roff,s_p_roff,i_h_roff,s_h_roff)

functions for the actual level3 data processing
    calc_lvl3_intensities_timeresolution(year,doy,tres,create_phaws=True, delete_phaws=True)
    merge_level3_daily_to_annual(year,timeres,header_lines=3)
"""

# paths (shall be defined in actual processing code)
"""
lvl2_pha_path="/data/missions/soho/costep/level2/pha/"
lvl1_sci_path="/data/missions/soho/costep/level1/sci/"
geompath="/data/missions/soho/python/l3i/GEOM_FACTORS/"
phaws_path="/data/missions/soho/python/l3i/tmp/"
lvl3_out_path="/data/missions/soho/costep/level3/l3i/"
"""

""" functions for the PHAWS data processing """
# load level2 pha file for given year and doy
def load_level2_pha(year,doy,unpack=False):
  pha_path=lvl2_pha_path  # /data/missions/soho/costep/level2/pha/
  thisyear=year
  thisdoy=doy
  if thisyear <2000:
    thisyear2d=thisyear-1900
    prefix='eph'
  else:
    thisyear2d=thisyear-2000
```

```
    prefix='epi'
  data=np.loadtxt("%s%s/%s%02d%03d.pl2" %(pha_path,thisyear,prefix, thisyear2d,thisdoy))
  if (year>=2017 and doy>276) or year>2017: fmd=True
  else: fmd=False
  if True:  # remove wrong coincidences
    cc=[]
    for q in range(len(data[:,1])):
      if check_coinc(data[q,1],data[q,5],data[q,6],data[q,7],data[q,8],data[q,9],fmd=fmd):
        cc.append(q)
    data=data[cc]
  if unpack==False:
    return data
  else:
    time=data[:,0] #  ms since year 0
    coinc=data[:,1]
    aseg=data[:,2]
    bseg=data[:,3]
    ea=data[:,5]
    eb=data[:,6]
    ec=data[:,7]
    ed=data[:,8]
    ee=data[:,9]
    etot=data[:,10]
    return time,coinc,aseg,bseg,ea,eb,ec,ed,ee,etot

# load level1 sci file for given year and doy
def load_level1_sci(year,doy):
  if year <2000:
    thisyear2d=year-1900
    prefix='eph'
  else:
    thisyear2d=year-2000
    prefix='epi'
  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,status=np.loadtxt("%s%s/%s%02d%03i.sci"%(lvl1_sci_path,year,prefix,thisyear2d,doy),usecols=(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,-1),unpack=True)
  p1=p1_1+p1_2+p1_3
  p2=p2_1+p2_2+p2_3
  p3=p3_1+p3_2+p3_3
  p4=p4_1+p4_2+p4_3
  h1=h1_1+h1_2+h1_3+h1_4
  h2=h2_1+h2_2+h2_3+h2_4
  h3=h3_1+h3_2+h3_3+h3_4
  h4=h4_1+h4_2+h4_3+h4_4
  lvl1_counts=[year,doy,msdoy, e1,e2,e3,e4, p1,p2,p3,p4, h1,h2,h3,h4, total_int_counts,status]
  return lvl1_counts

# checks for wrong coincidences
def check_coinc(co,a,b,c,d,e, fmd=False):
  t=0
  # def ths:
  a0,a1,a2,a3,a4=0.03,0.27,0.97,2.1,5.3
  b0,c0,d0,e0=0.06,0.37,0.58,0.58
  # electrons
  if co<4 and a>a0 and a<a1 and b>b0:
    if co==0 and c<c0 and d<d0 and e<e0: t=1
    if co==1 and c>c0 and d<d0 and e<e0: t=1
```

```
    if co==2 and c>c0 and d>d0 and e<e0: t=1
    if co==3 and c>c0 and d>d0 and e>e0: t=1
  # protons
  if 3<co<8 and a>a1 and b>b0:
    if fmd==False:
      if co==4 and a<a4 and c<c0 and d<d0 and e<e0: t=1
      if co==5 and a<a3 and c>c0 and d<d0 and e<e0: t=1
      if co==6 and a<a2 and c>c0 and d>d0 and e<e0: t=1
      if co==7 and a<a2 and c>c0 and d>d0 and e>e0: t=1
    else:   # if failure mode d: threshold in a changes
      if co==4 and a<a4 and c<c0 and d<d0 and e<e0: t=1
      elif a<a3: t=1

  # helium
  if 7<co and b>b0:
    if fmd==False:
      if co==8 and a>a4 and c<c0 and d<d0 and e<e0: t=1
      if co==9 and a>a3 and c>c0 and d<d0 and e<e0: t=1
      if co==10 and a>a2 and c>c0 and d>d0 and e<e0: t=1
      if co==11 and a>a2 and c>c0 and d>d0 and e>e0: t=1
    else:   # if failure mode d: threshold in a changes
      if co==8 and a>a4 and c<c0 and d<d0 and e<e0: t=1
      elif a>a2: t=1

  # returner
  if t==0: return False
  if t==1:  return True

# returns a pha like data product that includes wfacts (ratio counts/num_of_pha) and status bit
def add_wfact_to_pha(year,doy):
  scidata= load_level1_sci(year,doy)
  sci_msdoy=scidata[2]
  sci_status=scidata[-1]
  phadata= load_level2_pha(year,doy,unpack=False)
  phadata=phadata[phadata[:,1]!=12] # remove penetrating
  pha_time=phadata[:,0] #  ms since year 0
  coinc=phadata[:,1]
  msoffset=(dt.datetime(year,1,1)+dt.timedelta(doy-1))-dt.datetime(1,1,1)+dt.timedelta(366)
  pha_msdoy= pha_time-msoffset.total_seconds()*1e3
  wfacts=np.zeros(len(pha_msdoy))
  pha_status=np.ones(len(pha_msdoy))*-1
  for thismsec in sci_msdoy:
    # add status to pha
    pha_status[(pha_msdoy==thismsec)]=sci_status[sci_msdoy==thismsec][0]
    # get coinc counts in this minute
    coinccounters=[]
    for q in range(13): coinccounters.append(scidata[3+q][sci_msdoy==thismsec][0])
    # calc wfact for each coinc in this minute
    thiswfacts=[]
    for thiscoinc in range(13):
      numphas= len(pha_msdoy[(pha_msdoy==thismsec)&(coinc==thiscoinc)])
      ### care for failure modes!
      if (year>=1997 and doy>50) or year>1997:  #failure mode e as well as failure mode d (fmE: pha: 0,1,3  ,rl2: 0,1,2   fmDE: pha 0,3, rl2: 0,2)
            if thiscoinc in [3,7,11]:
              thiswfacts.append(coinccounters[thiscoinc-1]/numphas)
            else:
              thiswfacts.append(coinccounters[thiscoinc]/numphas)
      else:
```

```python
        thiswfacts.append(coinccounters[thiscoinc]/numphas)
    # dump wfacts in wfacts-array
    for thiscoinc in range(13):
      wfacts[(pha_msdoy==thismsec)&(coinc==thiscoinc)]=thiswfacts[thiscoinc]
  aseg=phadata[:,2]
  bseg=phadata[:,3]
  ea=phadata[:,5]
  eb=phadata[:,6]
  ec=phadata[:,7]
  ed=phadata[:,8]
  ee=phadata[:,9]
  etot=phadata[:,10]
  return pha_msdoy,coinc,aseg,bseg,ea,eb,ec,ed,ee,etot,wfacts,pha_status

#  makes a phaws from year and doy
def phaws_from_year_doy(year,doy,save=True):
  os.system("mkdir %s%i -p" %(phaws_path,year))
  msdoy,coinc,aseg,bseg,ea,eb,ec,ed,ee,etot,wfacts,pha_status=add_wfact_to_pha(year,doy)
  list_of_arrays=[msdoy.astype(int),coinc.astype(int),aseg.astype(int),bseg.astype(int),ea,eb,ec,ed,ee,etot,wfacts,pha_status.astype(int)]
  shape = list(list_of_arrays[0].shape)
  shape[:0] = [len(list_of_arrays)]
  arr = np.concatenate(list_of_arrays).reshape(shape).T
  if save==True:
    np.savetxt("%s%i/%i-%03d.phaws"%(phaws_path,year,year,doy),arr,fmt="%i %i %i %i %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f %4.4f %i")
  else:
    return arr

""" level3 AB-coincidence functions """
# returns lvl3 geom factors for ab coinc
def extract_lvl3_geoms_ab():
  #geompath="/home/pacifix/kuehl/work/simulations/G4ET_2015/build_level3_stopping/data/AB_COINC/"
  geomfile="LEVEL3_GEOMS_AB.DAT"
  geoms=np.loadtxt(geompath+geomfile)
  i_p_ron=geoms[0,0]
  s_p_ron=geoms[0,1]
  i_h_ron=geoms[1,0]
  s_h_ron=geoms[1,1]
  i_p_roff=geoms[2,0]
  s_p_roff=geoms[2,1]
  i_h_roff=geoms[3,0]
  s_h_roff=geoms[3,1]
  return i_p_ron,s_p_ron,i_h_ron,s_h_ron, i_p_roff,s_p_roff,i_h_roff,s_h_roff

# calc number of counts in ab coinc masks
def counts_in_lvl3_ab_ch_from_ea_eb(ea,eb):
  kappa=eb
  lam=(ea+eb)*ea
  mu=(ea+eb)/ea
  mask_kappa=(kappa>0.13)
  mask_lam_proton=(lam>10)&(lam<25)
  mask_lam_helium=(lam>120)&(lam<350)
  mask_mu=(mu>1.0)&(mu<5.3)
  p4=len( ea[ (mask_kappa)&(mask_lam_proton)&(mask_mu) ] )
  h4=len( ea[ (mask_kappa)&(mask_lam_helium)&(mask_mu) ] )
  return p4,h4

# calc intensity in ab coinc masks
def int_in_lvl3_ch_from_ea_eb(ea,eb, mywfact,myringoff,myaseg,mybseg, i_p_ron, s_p_ron,i_h_ron,s_h_ron, i_p_roff,s_p_roff,i_h_roff,s_h_roff):
```

```
  if 1 in myringoff:
    mask_center=(myaseg==0)&(mybseg==0)
    ea=ea[mask_center]
    eb=eb[mask_center]
    mywfact=mywfact[mask_center]
  kappa=eb
  lam=(ea+eb)*ea
  mu=(ea+eb)/ea
  mask_kappa=(kappa>0.13)
  mask_lam_proton=(lam>10)&(lam<25)
  mask_lam_helium=(lam>120)&(lam<350)
  mask_mu=(mu>1.0)&(mu<5.3)
  acctime=59.953
  p4=sum( mywfact[ (mask_kappa)&(mask_lam_proton)&(mask_mu) ] )
  c_p4=len( mywfact[ (mask_kappa)&(mask_lam_proton)&(mask_mu) ] )
  h4=sum( mywfact[ (mask_kappa)&(mask_lam_helium)&(mask_mu) ] )
  c_h4=len( mywfact[ (mask_kappa)&(mask_lam_helium)&(mask_mu) ] )
  if 1 in myringoff:
    i_p=i_p_roff
    i_h=i_h_roff
    s_p=s_p_roff
    s_h=s_h_roff
  else:
    i_p=i_p_ron
    i_h=i_h_ron
    s_p=s_p_ron
    s_h=s_h_ron
  i_p4=p4/i_p#/acctime
  sys_p4=i_p4 * s_p/i_p
  stat_p4=i_p4*1/np.sqrt(c_p4)
  i_h4=h4/i_h#/acctime
  sys_h4=i_h4 * s_h/i_h
  stat_h4=i_h4*1/np.sqrt(c_h4)
  return i_p4,sys_p4,stat_p4, i_h4,sys_h4,stat_h4


""" level3 ABC-coincidence functions """
# returns lvl3 geom factors for abc coinces
def extract_lvl3_geoms_abc():
  #geompath="/home/pacifix/kuehl/work/simulations/G4ET_2015/
      build_level3_stopping/data/"
  geomfile="LEVEL3_GEOMS_ABC.DAT"
  geoms=np.loadtxt(geompath+geomfile)
  i_p_ron=geoms[0,0:3]
  s_p_ron=geoms[0,3:]
  i_h_ron=geoms[1,0:3]
  s_h_ron=geoms[1,3:]
  i_p_roff=geoms[2,0:3]
  s_p_roff=geoms[2,3:]
  i_h_roff=geoms[3,0:3]
  s_h_roff=geoms[3,3:]
  return i_p_ron,s_p_ron,i_h_ron,s_h_ron, i_p_roff,s_p_roff,i_h_roff,s_h_roff

# calc number of counts in abc coinc masks
def counts_in_lvl3_ch_from_ea_eb_ec(ea,eb,ec):
  kappa=(2*ea-eb)/(2*ea+eb)
  lam=ea+eb+ec
  mu=ea+eb
  mask_kappa=(kappa>-0.35)&(kappa<0.15)
  mask_lam_proton=(lam>7.8)&(lam<27.5)
  mask_lam_helium=(lam>29.5)&(lam<110)
  p8=len( ea[ (mask_kappa)&(mask_lam_proton)&(mu>1.94) ] )
  p25=len( ea[ (mask_kappa)&(mask_lam_proton)&(mu<1.94)&(mu>1.3) ] )
```

```
  p41=len( ea[ (mask_kappa)&(mask_lam_proton)&(mu<1.3)&(mu>1.05) ] )
  h8=len( ea[ (mask_kappa)&(mask_lam_helium)&(mu>7.61) ] )
  h25=len( ea[ (mask_kappa)&(mask_lam_helium)&(mu<7.61)&(mu>5.09) ] )
  h41=len( ea[ (mask_kappa)&(mask_lam_helium)&(mu<5.09)&(mu>4.17) ] )
  return p8,p25,p41,h8,h25,h41

# calc intensity in abc coinc masks
def int_in_lvl3_ch_from_ea_eb_ec(ea,eb,ec, mywfact,myringoff,myaseg,mybseg,
    i_p_ron,s_p_ron,i_h_ron,s_h_ron, i_p_roff,s_p_roff,i_h_roff,s_h_roff):
  if 1 in myringoff:
    mask_center=(myaseg==0)&(mybseg==0)
    ea=ea[mask_center]
    eb=eb[mask_center]
    ec=ec[mask_center]
    mywfact=mywfact[mask_center]
  kappa=(2*ea-eb)/(2*ea+eb)
  lam=ea+eb+ec
  mu=ea+eb
  mask_kappa=(kappa>-0.35)&(kappa<0.15)
  mask_lam_proton=(lam>7.8)&(lam<27.5)
  mask_lam_helium=(lam>29.5)&(lam<110)
  acctime=59.953
  p8=sum( mywfact[ (mask_kappa)&(mask_lam_proton)&(mu>1.94) ] )
  c_p8=len( mywfact[ (mask_kappa)&(mask_lam_proton)&(mu>1.94) ] )
  p25=sum( mywfact[ (mask_kappa)&(mask_lam_proton)&(mu<1.94)&(mu>1.3) ] )
  c_p25=len( mywfact[ (mask_kappa)&(mask_lam_proton)&(mu<1.94)&(mu>1.3) ] )
  p41=sum( mywfact[ (mask_kappa)&(mask_lam_proton)&(mu<1.3)&(mu>1.05) ] )
  c_p41=len( mywfact[ (mask_kappa)&(mask_lam_proton)&(mu<1.3)&(mu>1.05) ] )
  h8=sum( mywfact[ (mask_kappa)&(mask_lam_helium)&(mu>7.61) ] )
  c_h8=len( mywfact[ (mask_kappa)&(mask_lam_helium)&(mu>7.61) ] )
  h25=sum( mywfact[ (mask_kappa)&(mask_lam_helium)&(mu<7.61)&(mu>5.09) ] )
  c_h25=len( mywfact[ (mask_kappa)&(mask_lam_helium)&(mu<7.61)&(mu>5.09) ] )
  h41=sum( mywfact[ (mask_kappa)&(mask_lam_helium)&(mu<5.09)&(mu>4.17) ] )
  c_h41=len( mywfact[ (mask_kappa)&(mask_lam_helium)&(mu<5.09)&(mu>4.17) ] )
  if 1 in myringoff:
    i_p=i_p_roff
    i_h=i_h_roff
    s_p=s_p_roff
    s_h=s_h_roff
  else:
    i_p=i_p_ron
    i_h=i_h_ron
    s_p=s_p_ron
    s_h=s_h_ron
  i_p8=p8/i_p[0]#/acctime
  sys_p8=i_p8 * s_p[0]/i_p[0]
  stat_p8=i_p8*1/np.sqrt(c_p8)
  i_p25=p25/i_p[1]#/acctime
  sys_p25=i_p25 * s_p[1]/i_p[1]
  stat_p25=i_p25*1/np.sqrt(c_p25)
  i_p41=p41/i_p[2]#/acctime
  sys_p41=i_p41 * s_p[2]/i_p[2]
  stat_p41=i_p41*1/np.sqrt(c_p41)
  i_h8=h8/i_h[0]#/acctime
  sys_h8=i_h8 * s_h[0]/i_h[0]
  stat_h8=i_h8*1/np.sqrt(c_h8)
  i_h25=h25/i_h[1]#/acctime
  sys_h25=i_h25 * s_h[1]/i_h[1]
  stat_h25=i_h25*1/np.sqrt(c_h25)
  i_h41=h41/i_h[2]#/acctime
  sys_h41=i_h41 * s_h[2]/i_h[2]
  stat_h41=i_h41*1/np.sqrt(c_h41)
  return i_p8,i_p25,i_p41, sys_p8,sys_p25,sys_p41, stat_p8,stat_p25,stat_p41,
```

```
    i_h8,i_h25,i_h41, sys_h8,sys_h25,sys_h41, stat_h8,stat_h25,stat_h41

""" functions for the actual level3 data processing """
# calcs intensity all coincs for given time resolution
def calc_lvl3_intensities_timeresolution(year,doy,tres,create_phaws=True, delete_phaws=True):
  i_p_ron_ab,s_p_ron_ab,i_h_ron_ab,s_h_ron_ab, i_p_roff_ab,s_p_roff_ab, i_h_roff_ab,s_h_roff_ab=extract_lvl3_geoms_ab()
  i_p_ron_abc,s_p_ron_abc,i_h_ron_abc,s_h_ron_abc, i_p_roff_abc,s_p_roff_abc, i_h_roff_abc,s_h_roff_abc=extract_lvl3_geoms_abc()
  os.system("mkdir %s%imin/ -p"%(lvl3_out_path,tres))
  os.system("mkdir %s%imin/%i -p"%(lvl3_out_path,tres,year))
  if True:
    try:
      if create_phaws==True:
        phaws_from_year_doy(year,doy,save=True)
      msdoy,coinc,aseg,bseg,ea,eb,ec,ed,ee,etot,wfacts,pha_status=np.loadtxt("%s%i/%i-%03d.phaws"%(phaws_path,year,year,doy),unpack=True)
      mask=[(coinc==1)+(coinc==2)+(coinc==3)+(coinc==5)+(coinc==6)+(coinc==7)+(coinc==9)+(coinc==10)+(coinc==11)]
      ringoff=np.zeros(len(pha_status))
      for q in range(len(pha_status)):
        binaries='{0:08b}'.format(int(pha_status[q]))
        if int(binaries[-2]): ringoff[q]=1
      f=open("%s%imin/%i/%i_%03d.l3i"%(lvl3_out_path,tres,year,year,doy) ,"w")
      f.write("# year month day doy hour minute status accum.time int_p4 int_p8 int_p25 int_p41  sys_p4 sys_p8 sys_p25 sys_p41  stat_p4 stat_p8 stat_p25 stat_p41   int_h4 int_h8 int_h25 int_h41  sys_h4 sys_h8 sys_h25 sys_h41  stat_h4 stat_h8 stat_h25 stat_h41 \n# all values except for time and status are intensities in units of (cm^2 s sr mev/nuc)^-1\n# zero counts in given channel are indicated by a '-999' in the intensity, stat and sys uncertainty\n")
      tinter=[0]
      while tinter[-1]<1440:
        tinter.append(tinter[-1]+tres)
      #for mytime in np.unique(msdoy):
      for tidx in range(len(tinter)-1):
        smin,emin=tinter[tidx],tinter[tidx+1]
        tmask=(msdoy>=smin*60000)&(msdoy<emin*60000)
        # write time and status
        if not any(tmask):
          continue
        hour,minutes=divmod(smin,60)
        mydate=dt.datetime(int(year),1,1)+dt.timedelta(int(doy)-1)
        month,day=mydate.month,mydate.day
        mystatus=np.max(pha_status[tmask])  #pha_status[tmask][0]
        f.write("%i %i %i %i %i %i %i " %(year,month,day,doy,hour,minutes, mystatus ) )
        f.write("  ")
        tnorm=len(np.unique(msdoy[tmask]))*59.953  # timeinterval in seconds
        f.write("%4.4f   "%tnorm)
        # calc lvl3 intensities AB
        lvl3_coinc_mask=((coinc==0)+(coinc==4)+(coinc==8))
        myea=ea[(tmask)&(lvl3_coinc_mask)]
        myeb=eb[(tmask)&(lvl3_coinc_mask)]
        myec=ec[(tmask)&(lvl3_coinc_mask)]
        myaseg=aseg[(tmask)&(lvl3_coinc_mask)]
        mybseg=bseg[(tmask)&(lvl3_coinc_mask)]
        mywfact=wfacts[(tmask)&(lvl3_coinc_mask)]
        myringoff=ringoff[(tmask)&(lvl3_coinc_mask)]
        i_p4,sys_p4,stat_p4, i_h4,sys_h4,stat_h4 = int_in_lvl3_ch_from_ea_eb(myea,myeb,mywfact,myringoff,myaseg,mybseg, i_p_ron_ab,s_p_ron_ab, i_h_ron_ab,s_h_ron_ab, i_p_roff_ab,s_p_roff_ab,i_h_roff_ab,
```

```
            s_h_roff_ab)
        i_p4,sys_p4,stat_p4, i_h4,sys_h4,stat_h4=i_p4/tnorm,sys_p4/tnorm,
            stat_p4/tnorm, i_h4/tnorm,sys_h4/tnorm,stat_h4/tnorm
        # calc lvl3 intensities ABC
        lvl3_coinc_mask=((coinc!=0)&(coinc!=4)&(coinc!=8)&(coinc!=12))
        myea=ea[(tmask)&(lvl3_coinc_mask)]
        myeb=eb[(tmask)&(lvl3_coinc_mask)]
        myec=ec[(tmask)&(lvl3_coinc_mask)]
        myaseg=aseg[(tmask)&(lvl3_coinc_mask)]
        mybseg=bseg[(tmask)&(lvl3_coinc_mask)]
        mywfact=wfacts[(tmask)&(lvl3_coinc_mask)]
        myringoff=ringoff[(tmask)&(lvl3_coinc_mask)]
        i_p8,i_p25,i_p41, sys_p8,sys_p25,sys_p41, stat_p8,stat_p25,stat_p41,
            i_h8,i_h25,i_h41, sys_h8,sys_h25,sys_h41, stat_h8,stat_h25,
            stat_h41=int_in_lvl3_ch_from_ea_eb_ec(myea,myeb,myec,mywfact,
            myringoff,myaseg,mybseg, i_p_ron_abc,s_p_ron_abc,i_h_ron_abc,
            s_h_ron_abc, i_p_roff_abc,s_p_roff_abc,i_h_roff_abc,s_h_roff_abc)
        i_p8,i_p25,i_p41, sys_p8,sys_p25,sys_p41, stat_p8,stat_p25,stat_p41,
            i_h8,i_h25,i_h41, sys_h8,sys_h25,sys_h41, stat_h8,stat_h25,
            stat_h41=i_p8/tnorm,i_p25/tnorm,i_p41/tnorm, sys_p8/tnorm,sys_p25/
            tnorm,sys_p41/tnorm, stat_p8/tnorm,stat_p25/tnorm,stat_p41/tnorm,
             i_h8/tnorm,i_h25/tnorm,i_h41/tnorm, sys_h8/tnorm,sys_h25/tnorm,
            sys_h41/tnorm, stat_h8/tnorm,stat_h25/tnorm,stat_h41/tnorm
        # if int=0 => set sys,stat uncertainties = -999
        set_zeros_invalid=True
        if set_zeros_invalid:
          keyword=-999
          if i_p4==0: sys_p4,stat_p4=keyword,keyword
          if i_p8==0: sys_p8,stat_p8=keyword,keyword
          if i_p25==0: sys_p25,stat_p25=keyword,keyword
          if i_p41==0: sys_p41,stat_p41=keyword,keyword
          if i_h4==0: sys_h4,stat_h4=keyword,keyword
          if i_h8==0: sys_h8,stat_h8=keyword,keyword
          if i_h25==0: sys_h25,stat_h25=keyword,keyword
          if i_h41==0: sys_h41,stat_h41=keyword,keyword
        # write lvl3 intensities
        f.write("%2.4e %2.4e %2.4e %2.4e  %2.4e %2.4e %2.4e %2.4e  %2.4e %2.4e
             %2.4e %2.4e   %2.4e %2.4e %2.4e %2.4e  %2.4e %2.4e %2.4e %2.4e
            %2.4e %2.4e %2.4e %2.4e"%(i_p4,i_p8,i_p25,i_p41, sys_p4,sys_p8,
            sys_p25,sys_p41, stat_p4,stat_p8,stat_p25,stat_p41,  i_h4,i_h8,
            i_h25,i_h41, sys_h4,sys_h8,sys_h25,sys_h41, stat_h4,stat_h8,
            stat_h25,stat_h41))
        f.write("\n")
      f.close()
    except:
      d=1
    if delete_phaws:  os.system("rm -f %s%i/%i-%03d.phaws"%(phaws_path,year,
        year,doy))

# merge level3 daily files to annual
def merge_level3_daily_to_annual(year,timeres,header_lines=3):
  init=1
  g=open("%s%imin/%i.l3i"%(lvl3_out_path,timeres,year),"w")
  for doy in range(1,370):
    try:
      f=open("%s%imin/%i/%i_%03d.l3i"%(lvl3_out_path,timeres,year,year,doy),"r
          ")
      if init==0: #skip header
        for i in range(header_lines): f.readline()
      for line in f:
        g.write(line)
      init=0
      f.close()
```

```
    except:
      continue  #print "no file", year, doy
  g.close()
```

8 Appendix II: create_lvl3_files_etmd.py

```
#! /usr/bin/python
lvl2_pha_path="/data/missions/soho/costep/level2/pha/"
lvl1_sci_path="/data/missions/soho/costep/level1/sci/"
geompath="/data/missions/soho/python/l3i/GEOM_FACTORS/"
phaws_path="/data/missions/soho/python/l3i/tmp/"
lvl3_out_path="/data/missions/soho/costep/level3/l3i/"

timeresolutions=[1,5,10,30,60,1440] # in minutes
verbosity=0
execfile("/data/missions/soho/python/l3i/level3_funcs.py")
maxyear=dt.date.today().year+1

for year in range(1995,maxyear):
  for doy in range(1,370):
    if verbosity==1:  print year,doy

    # continue loop if level3 intensities exists in lvl3 output for all
       timeresolutions
    existences=[]
    for timeres in timeresolutions:
      existences.append( os.path.isfile("%s%imin/%s/%i_%03d.l3i" %(
         lvl3_out_path,timeres,year,year,doy)) )

    if all(existences):
      if verbosity==1: print "intensity files already exist for each
         timeresolution, skipping..."
      continue

    # create PHAWS
    try:
      phaws_from_year_doy(year,doy,save=True)
    except:
      if verbosity==1: print "ERROR: could not create PHAWS"
      continue

    # create lvl3 intensity files
    for timeres in timeresolutions:
      calc_lvl3_intensities_timeresolution(year,doy,timeres,create_phaws=False
         ,delete_phaws=False)

    # remove temp files (PHAWS)
    os.system("rm -rf %s*"%phaws_path)

  for timeres in timeresolutions:
    # check if annual file is already complete, otherwise merge daily files
    annual_file="%s%imin/%i.l3i"%(lvl3_out_path,timeres,year)
    if os.path.isfile(annual_file):
      last_annual_line=subprocess.check_output(['tail', '-1', annual_file])
      last_doy_in_annual=int(last_annual_line.split(" ")[3])

      daily_files=os.listdir("%s%imin/%i/"%(lvl3_out_path,timeres,year))
      daily_files.sort()
      last_doy=int(daily_files[-1].split("_")[1].split(".")[0])
      if last_doy == last_doy_in_annual: continue
    merge_level3_daily_to_annual(year,timeres)


execfile("/data/missions/soho/python/l3i/merge_entire_mission.py")
```

9 Appendix III: Geometry files

- LEVEL3_GEOMS_AB.DAT:

```
### level3 geoms. All values in (cm^2 sr MeV). G: geometry, S: systematic
    uncertainties

## ring on

# proton: G(p4) S(p4)
20.08 2.32

# helium: G(h4) S(h4)
23.26 4.09

## ring off

# proton: G(p4) S(p4)
0.69 0.04

# helium: G(h4) S(h4)
0.76 0.08
```

- LEVEL3_GEOMS_ABC.DAT:

```
### level3 geoms. All values in (cm^2 sr MeV). G: geometry, S: systematic
    uncertainties

## ring on

# proton: G(p8) G(p25) G(p41) S(p8) S(p25) S(p41)
84.17 76.15 55.08 9.75 2.01 1.23

# helium: G(h8) G(h25) G(h41) S(h8) S(h25) S(h41)
85.28 75.74 50.46 10.23 2.75 0.57

## ring off

# proton: G(p8) G(p25) G(p41) S(p8) S(p25) S(p41)
3.03 2.75 2.07 0.31 0.22 0.19

# helium: G(h8) G(h25) G(h41) S(h8) S(h25) S(h41)
2.94 2.72 1.86 0.26 0.17 0.19
```