**cvt\_l1tol2.py Documentation**

**1. compute\_moments Function: Compute Corrected Data**

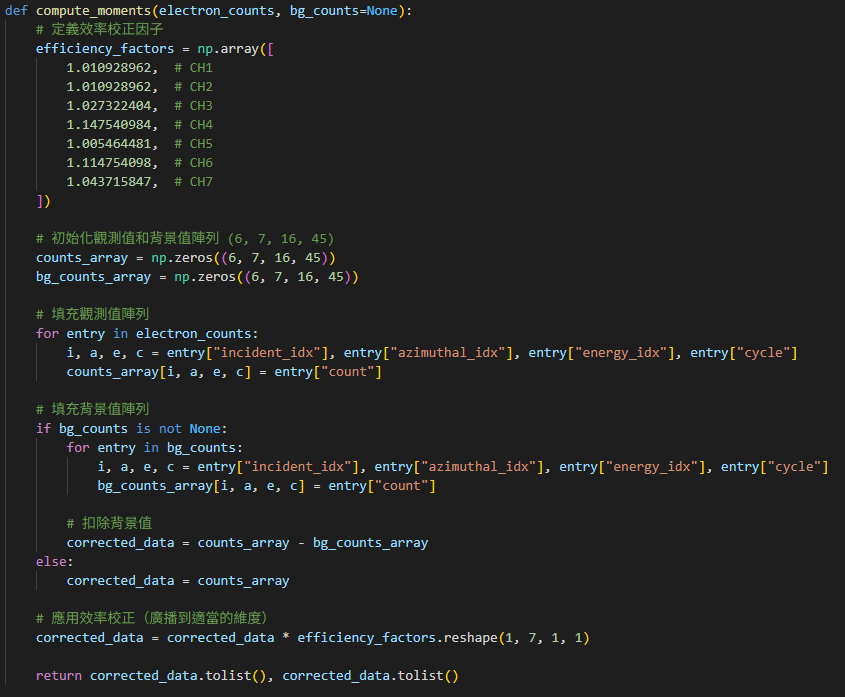
**Functionality:**  
Performs **efficiency correction** and **background subtraction** on electron count data.

**Inputs:**

* electron\_counts: A list of dictionaries, each containing raw observed electron count data.
* bg\_counts *(optional)*: Background noise count data.

**Processing Logic:**

1. Define **efficiency correction factors** (efficiency\_factors) corresponding to channels CH1~CH7.
2. Initialize a **counts data array** with shape (6, 7, 16, 45), representing different directions, azimuths, energies, and cycles.
3. Populate counts\_array and bg\_counts\_array with values from electron\_counts and bg\_counts, respectively.
4. If background values are provided, subtract them from the observed data.
5. Apply **efficiency correction** by multiplying with the correction factors.
6. Return both the **total** and the **mean** (currently returns two identical corrected\_data.tolist() arrays, designed for extensibility to include statistical metrics like average values).



**2. save\_as\_cdf Function: Save Results as a CDF File**

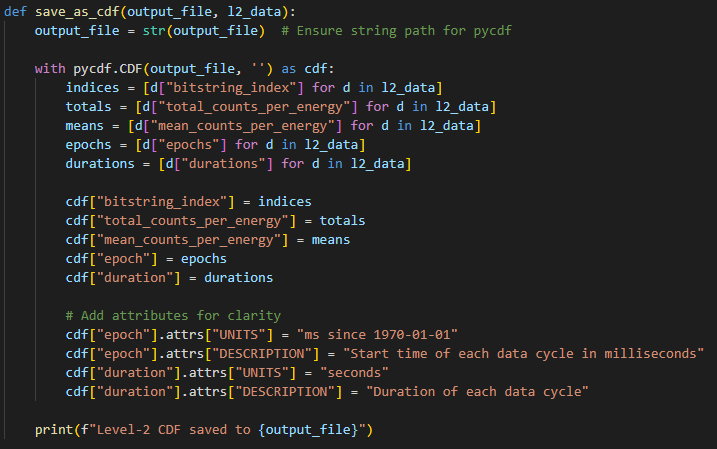
**Functionality:**  
Saves the processed data into a **CDF (Common Data Format)** file, a format commonly used in space physics research.  
Utilizes the spacepy.pycdf package.

**Inputs:**

* output\_file: The output file name (with .cdf extension).
* l2\_data: A list of processed data entries, each being a dictionary containing:
  + bitstring\_index
  + total\_counts\_per\_energy
  + mean\_counts\_per\_energy
  + epochs
  + durations

**Additional Notes:**

* Descriptive **metadata** is added to the epoch and duration fields.

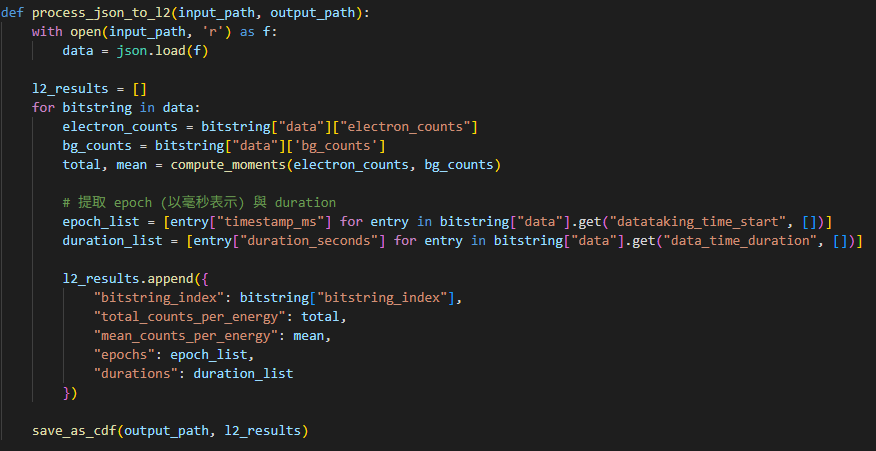


**3. process\_json\_to\_l2 Function: Process JSON Input Data**

**Functionality:**  
Reads L1 data in .json format, computes corrected results, and packages them into L2 format.

**Workflow:**

1. Use json.load() to read the input file.
2. Iterate over each bitstring:
   * Extract electron count and background data.
   * Call compute\_moments() to apply corrections.
   * Extract timestamps (epoch) and duration.
   * Assemble a dictionary and append it to the result list.
3. Call save\_as\_cdf() to save the output.

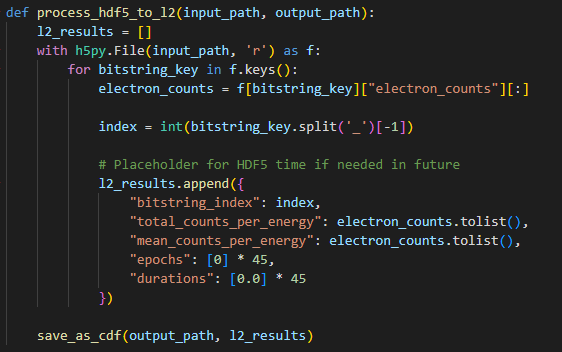


**4. process\_hdf5\_to\_l2 Function: Process HDF5 Input Data**

**Functionality:**  
Reads L1 data in .h5 format and packages it into a basic L2 structure.

**Notes:**

* Does **not** call compute\_moments() — assumes the data has already been corrected.
* epoch and duration fields are set to 0 as **placeholders**, and can be refined in future updates.

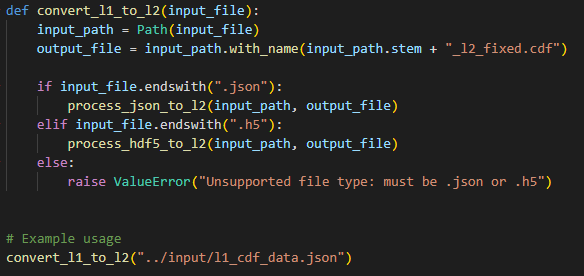


**5. convert\_l1\_to\_l2 Function: Main Controller Function**

**Functionality:**  
Automatically determines the input file format and selects the appropriate processing function.

**Supported Formats:**

* .json → Calls process\_json\_to\_l2()
* .h5 → Calls process\_hdf5\_to\_l2()



**Workflow:**

.json / .h5 → convert\_l1\_to\_l2()

↓

Select corresponding processing function

↓ ↓

process\_json\_to\_l2() process\_hdf5\_to\_l2()

↓ ↓

compute\_moments() [Use values directly]

↓ ↓

Package into dict and append to L2 result list

↓

save\_as\_cdf()