**cvt\_l1tol2.py 說明文件**

**1. compute\_moments 函數：計算校正後的資料**

**功能：**

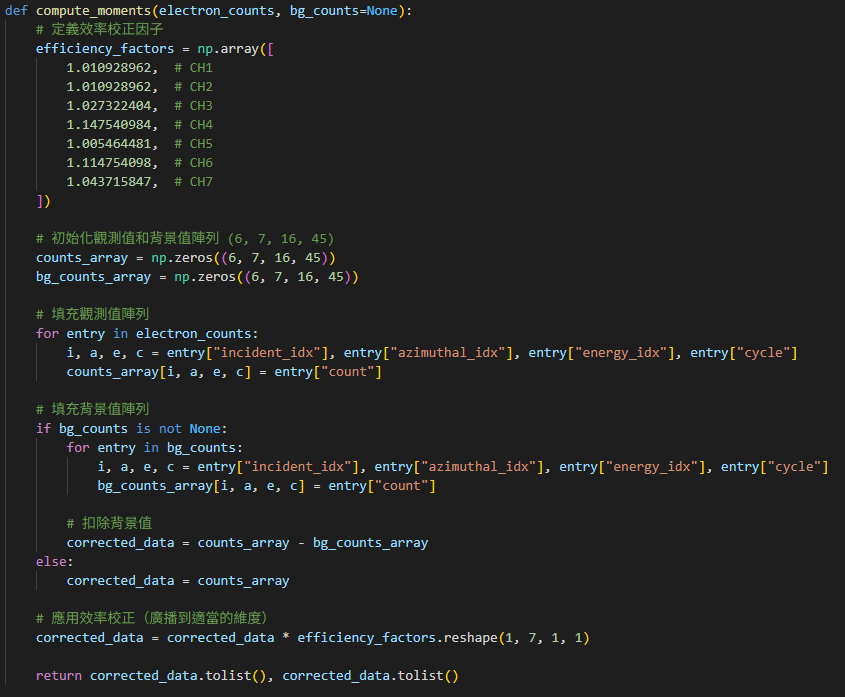
對電子計數資料進行 **效率校正** 並 **扣除背景值**。

**輸入：**

* electron\_counts：包含原始觀測電子計數的清單，每一筆是 dict。
* bg\_counts（可選）：背景雜訊計數資料。

**運作邏輯：**

1. 定義效率校正因子 efficiency\_factors（對應通道 CH1~CH7）。
2. 初始化形狀為 (6, 7, 16, 45) 的計數資料陣列（分別對應不同的方向、方位角、能量、循環）。
3. 將 electron\_counts 和 bg\_counts 分別填入對應的 counts\_array 和 bg\_counts\_array。
4. 如果有背景值，則從觀測值中扣除。
5. 對資料套用效率校正（乘上因子）。
6. 回傳總計與平均值（實際是兩份相同的 corrected\_data.tolist()，可擴充成計算平均值等統計資料的架構）。



**2. save\_as\_cdf 函數：儲存結果為 CDF 檔案**

**功能：**

將處理後的資料儲存成 CDF 格式，這是太空物理資料常用的一種格式。

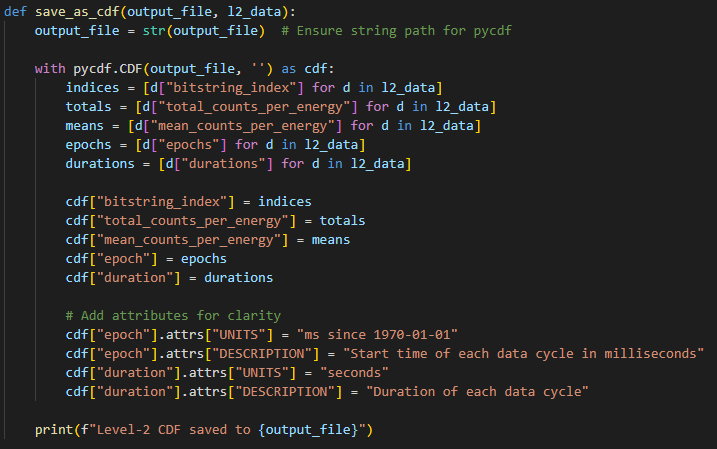
**使用了 spacepy.pycdf 套件。**

**輸入：**

* output\_file：輸出檔名（.cdf）
* l2\_data：處理後的資料列表，每筆是 dict，包含：
  + bitstring\_index
  + total\_counts\_per\_energy
  + mean\_counts\_per\_energy
  + epochs
  + durations

**附加說明：**

* 給 epoch 和 duration 欄位加上描述性的 metadata。



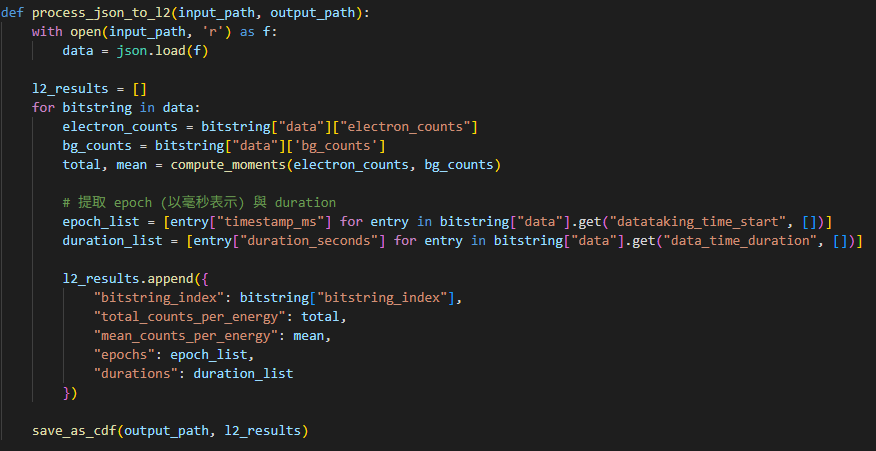
**3. process\_json\_to\_l2 函數：處理 JSON 輸入資料**

**功能：**

讀取 .json 格式的 L1 資料，計算校正後的結果並封裝成 L2 格式。

**流程：**

1. 使用 json.load() 讀取檔案。
2. 遍歷每筆 bitstring：
   * 取得電子計數與背景資料。
   * 呼叫 compute\_moments() 做校正。
   * 擷取時間戳（epoch）與 duration。
   * 組成 dict 加入結果清單。
3. 呼叫 save\_as\_cdf() 儲存。



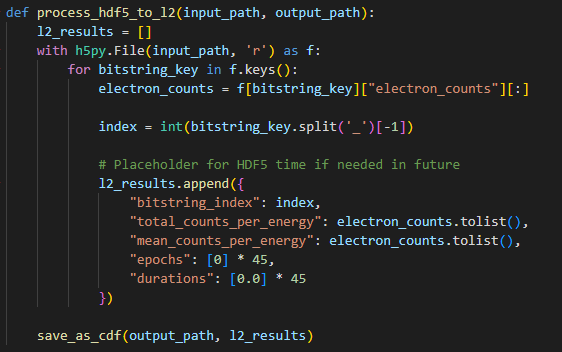
**4. process\_hdf5\_to\_l2 函數：處理 HDF5 輸入資料**

**功能：**

讀取 .h5 格式的 L1 資料，封裝成基本的 L2 結構。

**注意：**

* 沒有做 compute\_moments()，假設數據已校正。
* 時間戳和 duration 被設定為 0，是 placeholder，可日後加強。



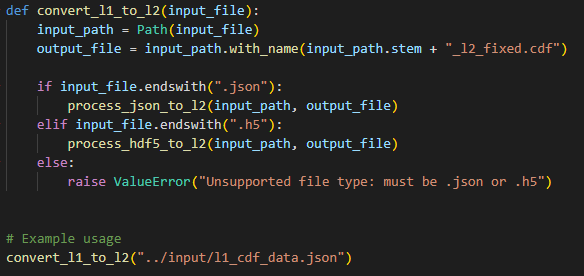
**5. convert\_l1\_to\_l2 函數：主控制函數**

**功能：**

自動判斷輸入檔案格式並選擇正確的處理函數。

**支援：**

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



流程：

.json / .h5 → convert\_l1\_to\_l2()

↓

選擇對應處理函數

↓ ↓

process\_json process\_hdf5

↓ ↓

compute\_moments() [直接使用值]

↓ ↓

封裝成 dict，加入 L2 結果列表

↓

save\_as\_cdf()