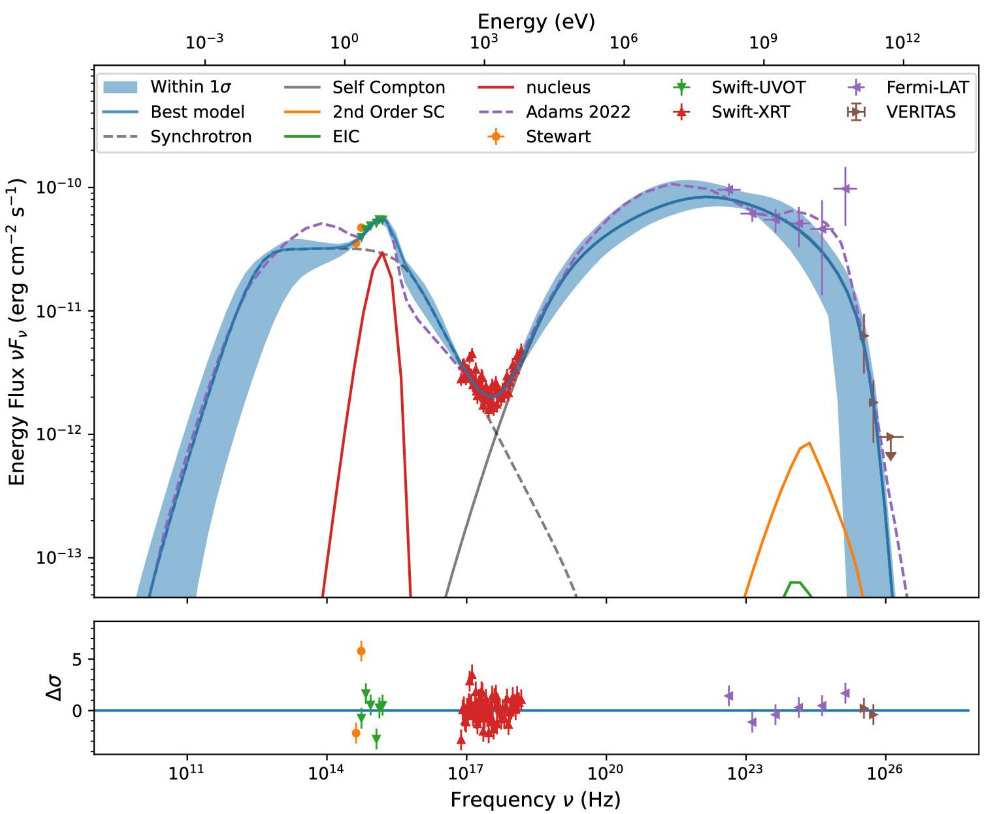
Using *gammapy* in your multi-wavelength workflow

Mireia Nievas Rosillo (IAC),
Fabio Acero, Jorge Otero, Daniel Morcuende, Régis Terrier,
Mónica Vázquez, Axel Arbet-Engels

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
AGENCIA ESTATAL DE INVESTIGACIÓN
Cofinanciado por la Unión Europea

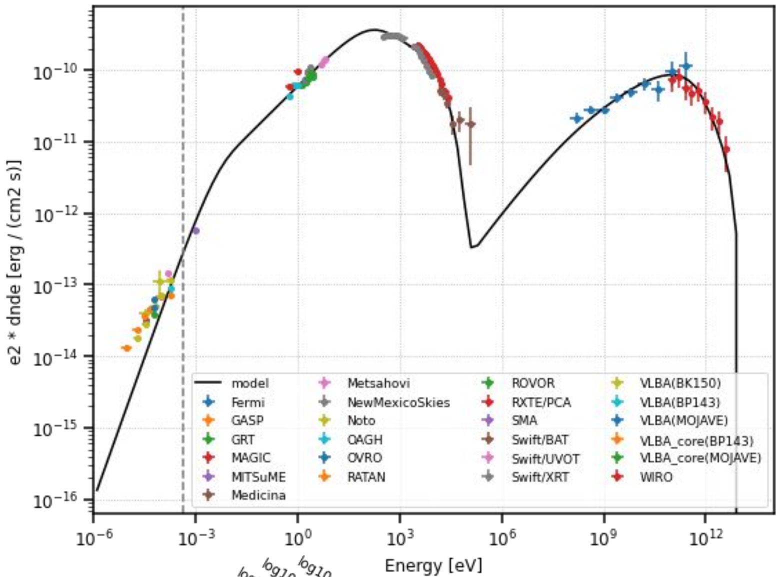
Motivation

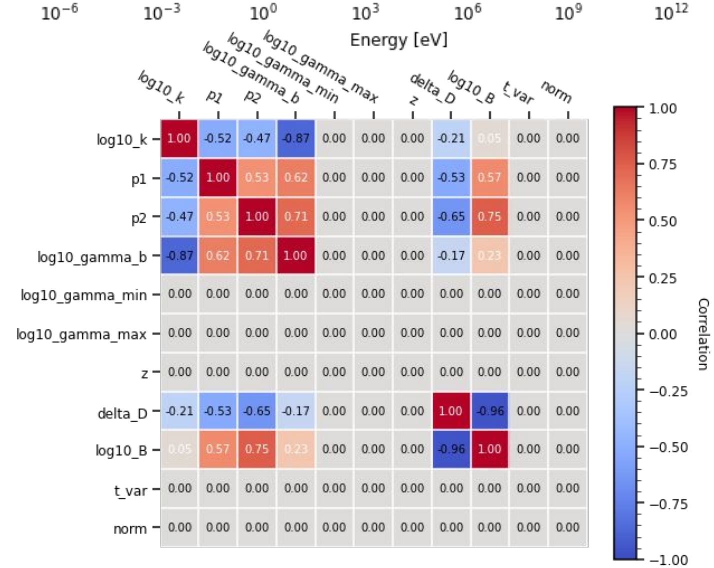
Most blazar MWL papers end up with
a collection of points + some modelling



BJet: Olivier Hervet *et al* 2024 *ApJ* 962 140

From the agnpy ***readthedocs*** page. Nigro et al. 2023





Not ideal

- Statistically not-correct. Poisson statistics vs Gaussian statistics
- No easy way to include upper-limits!.
- X-rays: `xspec` is not designed to produce flux points. Yet we ***misuse*** it for that.
- X-rays: ‘dirty tricks’ often used to correct for hydrogen absorption.
- Optical-UV: magnitudes are flux densities → conversion to our $E^2 dN/dE$ not trivial (requires assuming a spectral shape …).
- Ignores ‘flux point’ correlations (X-rays, LAT, UV-Optical).
- Ignores the interplay between source emission and the instrument model (i.e. redistribution matrices, PSF).
- Systematics usually added in quadrature to statistical errors → *too simplistic*.
- Backgrounds: they are ‘frozen’ when one reconstructs the flux points.

A better way

Forward folding + emission modeling. Currently two ways of doing it.

3ML
Multi-Mission Maximum Likelihood Framework

VS

$\gamma\pi$
A **Python** package for **gamma-ray** astronomy

Wrappers around each native tool to 'expose' the likelihood functionality.

- Pros: using 'native' tools
- Cons: i) all tools need to 'coexist' in the same env; ii) different data formats; iii) limits data reproducibility

Convert the data and instrument description to a common format.

- Pros: common likelihood code, "data portability".
- Cons: need to convert the data and the instrument responses.

A better way

GitHub https://github.com/mireianievas/gammapy_mwl_workflow

Forward folding + emission modeling. Currently two ways of doing it.

3ML
Multi-Mission Maximum Likelihood Framework

VS

A **Python** package for **gamma-ray** astronomy

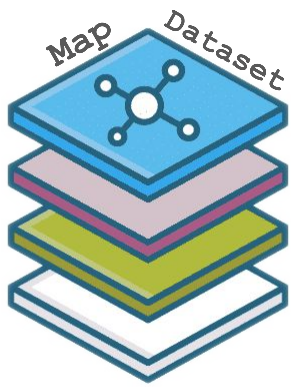
Wrappers around each native tool to 'expose' the likelihood functionality.

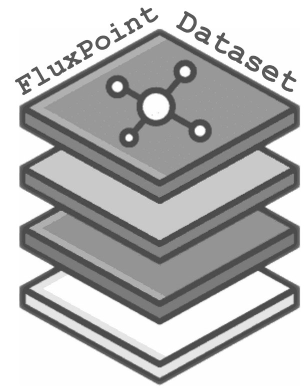
- Pros: using 'native' tools
- Cons: i) all tools need to 'coexist' in the same env; ii) different data formats; iii) limits data reproducibility

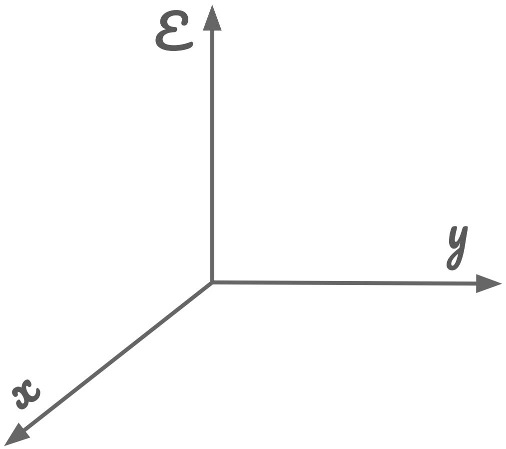
Convert the data and instrument description to a common format.

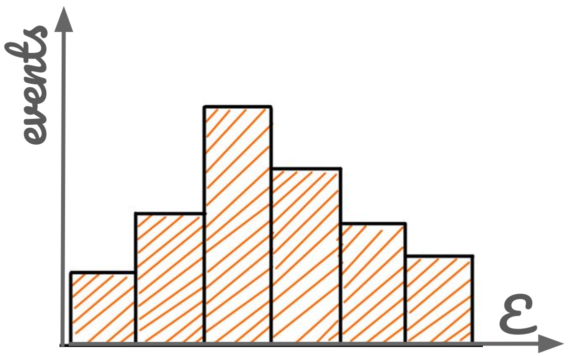
- Pros: common likelihood code, "data portability".
- Cons: need to convert the data and the instrument responses.

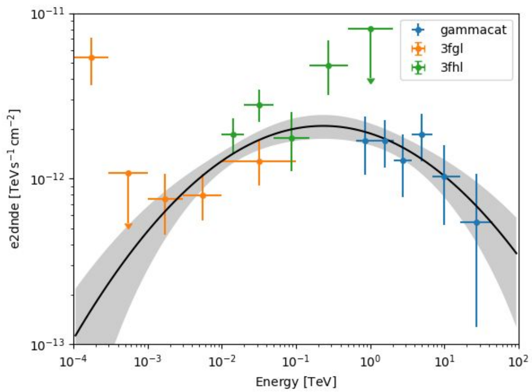
Datasets in `gammapy`.



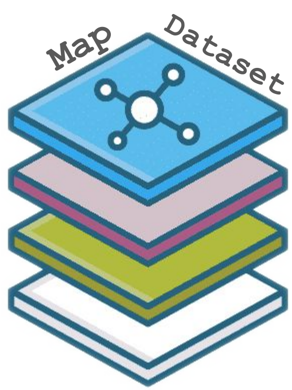


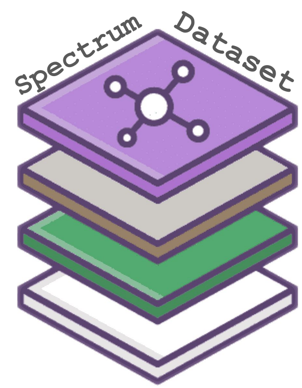


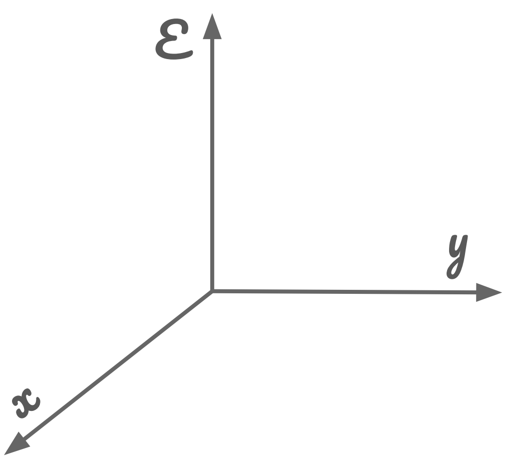


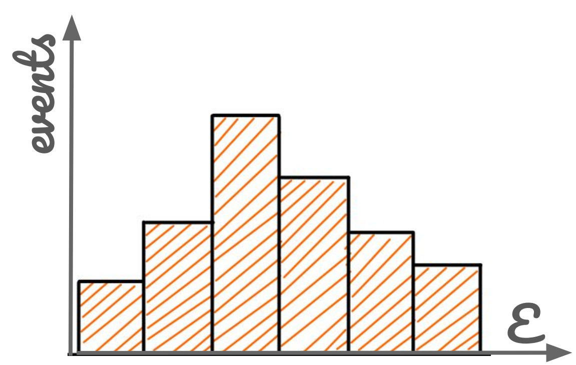


Datasets in `gammapy`.



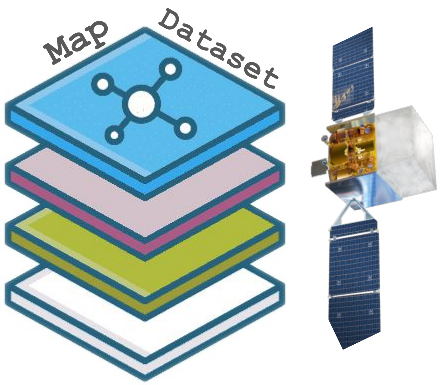






Fermi-LAT.

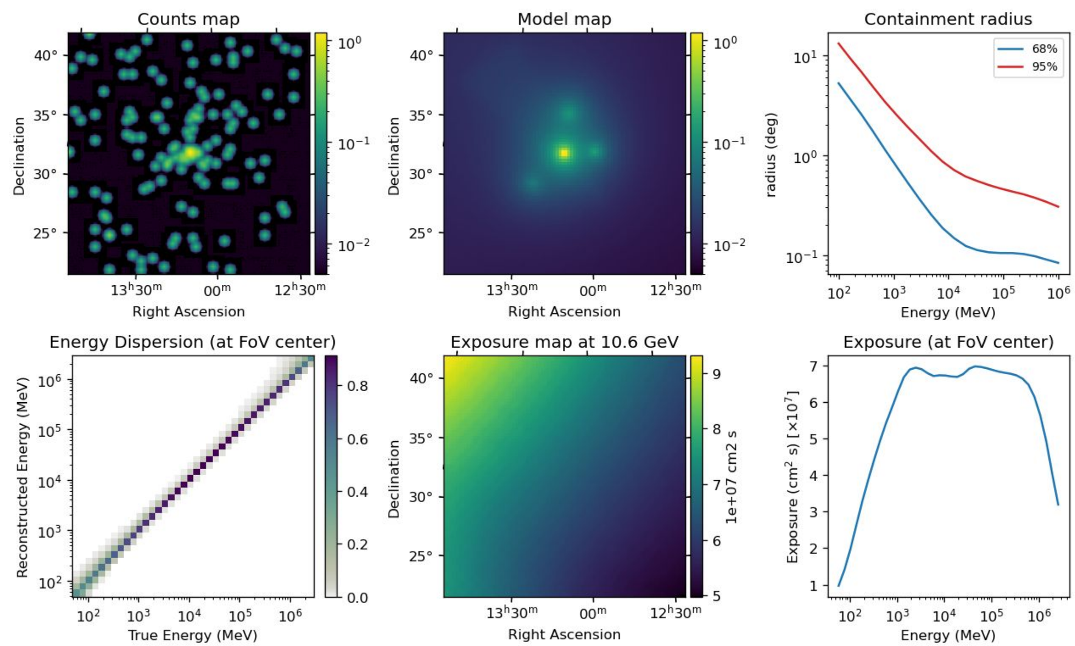
Quentin's **PR** to simplifies Fermi data handling
https://github.com/gammapy/gammapy/pull/5649



Tutorial:
https://docs.gammapy.org/dev/tutorials/data/fermi_lat.html

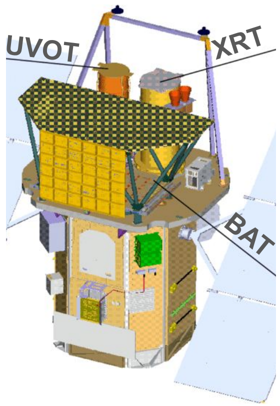
Computing your skymodel:
`SourceCatalog4FGL().to_models()`

```
Fermi = MapDataset(
    models=list_of_models,
    counts=counts_map,
    exposure=exposure_interp,
    psf=psfmap,
    edisp=edisp_kernel_map,
    mask_safe=None,
    name=dataset_name,
    gti=gtis,
)
```



Ingredients (from the `Fermitools`)
evtfile → output of gtmktime (photons +gtis)
bgd → None (*other sources inc. diffuse*)
psffile → output of gtpsf
expfile → output of gtexpcube2
drmfile → output of gtdrm
model → includes all sources and bkgs

Swift-XRT / NuSTAR.



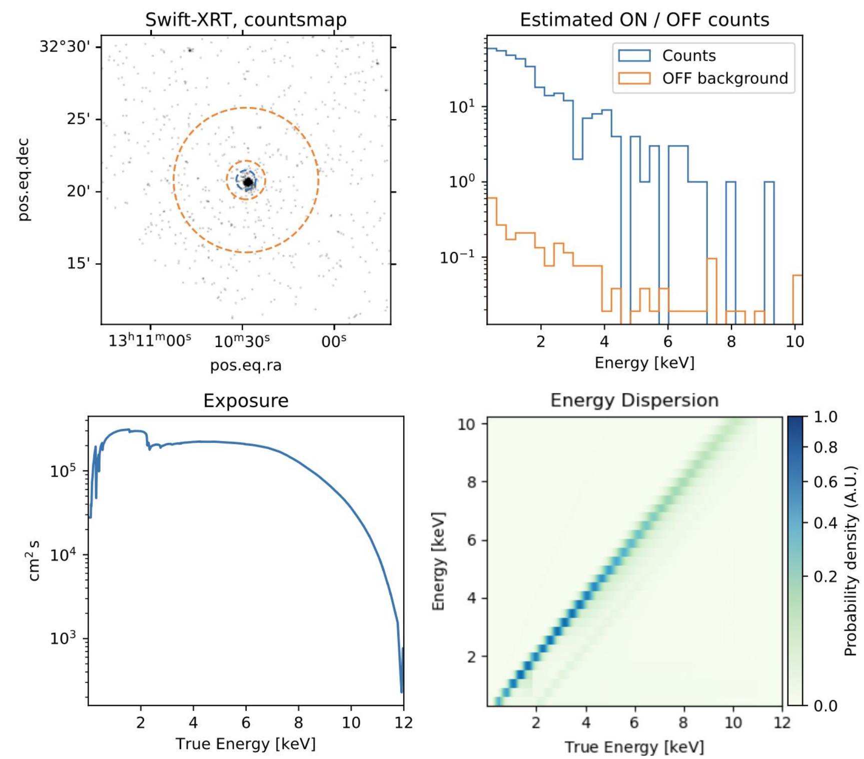

```
Swift = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```

Ingredients (from `xrtproducts/nuproducts`)
evtfile → source pha (Poisson stats)
bgd → background pha (Poisson stats)
psffile → None! (1D dataset)
expfile → arf file (xrtmkarf computes it)
drmfile → from CALDB
model → includes *only* source.

Converter (xspec OGIP → gammapy dataset)

https://github.com/luca-giunti/gammapyXray
http://github.com/mireianievas/gammapy_mwl_workflow

The data is already in OGIP format: ON / OFF histograms + *exposure* (ARF file) + *edisp* (RMF file).

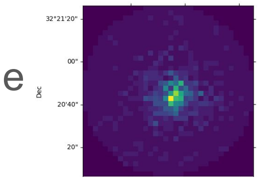


CTAO | LST COLLABORATION

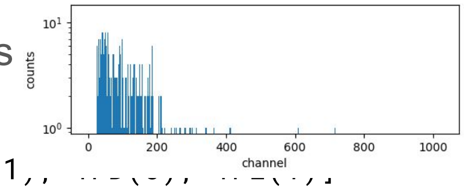
Swift-XRT / NuSTAR. *ogip format*

Output of xrtproducts: on / off spectra + arf (+rmf from *Heasarc*)

skymap for the on region



counts spec

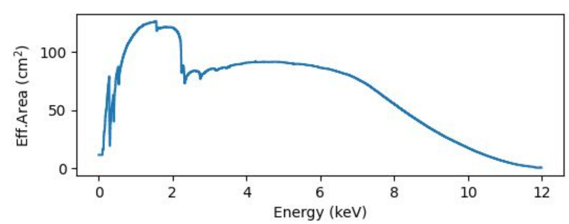



```
Filename: reproc/00036384112/XRT/source.pha
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1 PrimaryHDU      756   (34, 34)     float32
  1  SPECTRUM        1 BinTableHDU     720   1024R x 2C    [J, J]
  2  GTI             1 BinTableHDU     103   2R x 2C    [1D, 1D]
  3  REG00101        1 BinTableHDU     130   1R x 6C    [1PD(1), 1PD(1), 16A, 1PD(1

Filename: reproc/00036384112/XRT/background.pha
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1 PrimaryHDU      756   (254, 254)   float32
  1  SPECTRUM        1 BinTableHDU     720   1024R x 2C    [J, J]
  2  GTI             1 BinTableHDU     103   2R x 2C    [1D, 1D]
  3  REG00101        1 BinTableHDU     130   1R x 6C    [1PD(1), 1PD(1), 16A, 1PD(2), 1PD(0), 1PI(1)]

Filename: reproc/00036384112/XRT/exposure.arf
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1 PrimaryHDU       11   ()
  1  SPECRESP        1 BinTableHDU      69   2400R x 3C    [1E, 1E, 1E]

Filename: reproc/00036384112/XRT/swxpc0to12s6_20210101v016.rmf
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1 PrimaryHDU       10   ()
  1  MATRIX          1 BinTableHDU      78   2400R x 6C    [D, D, I, PJ(189), PJ(189), PE(982)]
  2  EBOUNDS         1 BinTableHDU      46   1024R x 3C    [I, D, D]
```

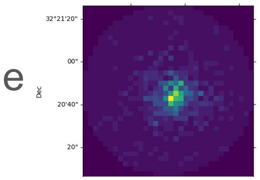


[('ENERG_LO', '>f8'), ('ENERG_HI', '>f8'), ('N_GRP', '>i2'), ('F_CHAN', '>i4', (2,)), ('N_CHAN', '>i4', (2,)), ('MATRIX', '>i4', (2,))])

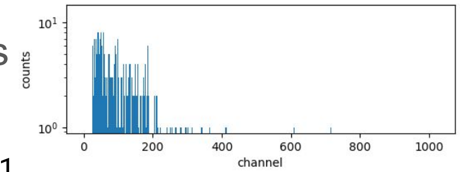
Swift-XRT / NuSTAR. *ogip format*

Output of xrtproducts: on / off spectra + arf (+rmf from *Heasarc*)

skymap for the on region



counts spec

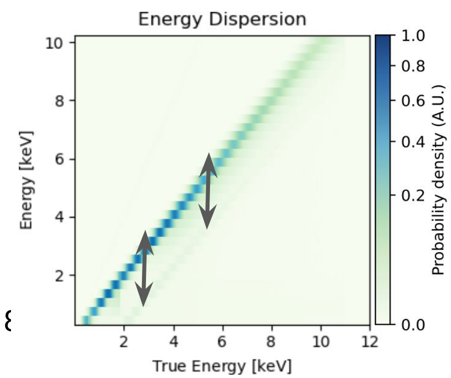



```
Filename: reproc/00036384112/XRT/source.pha
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1  PrimaryHDU     756   (34, 34)     float32
  1  SPECTRUM        1  BinTableHDU    720   1024R x 2C    [J, J]
  2  GTI             1  BinTableHDU    103   2R x 2C    [1D, 1D]
  3  REG00101        1  BinTableHDU    130   1R x 6C    [1PD(1), 1PD(1), 16A, 1PD(1
```

```
Filename: reproc/00036384112/XRT/background.pha
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1  PrimaryHDU     756   (254, 254)   float32
  1  SPECTRUM        1  BinTableHDU    720   1024R x 2C    [J, J]
  2  GTI             1  BinTableHDU    103   2R x 2C    [1D, 1D]
  3  REG00101        1  BinTableHDU    130   1R x 6C    [1PD(1), 1PD(1), 16A, 1PD(2), 1PD(0), 1PI(1)]
```

```
Filename: reproc/00036384112/XRT/exposure.arf
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1  PrimaryHDU      11   ()
  1  SPECRESP        1  BinTableHDU     69   2400R x 3C    [1E, 1E, 1E]
```

```
Filename: reproc/00036384112/XRT/swxpc0to12s6_20210101v016.rmf
No.    Name        Ver    Type        Cards   Dimensions    Format
  0  PRIMARY         1  PrimaryHDU      10   ()
  1  MATRIX          1  BinTableHDU     78   2400R x 6C    [D, D, I, PJ(189), PJ(18
  2  EBOUNDS         1  BinTableHDU     46   1024R x 3C    [I, D, D]
```



[('ENERG_LO', '>f8'), ('ENERG_HI', '>f8'), ('N_GRP', '>i2'), ('F_CHAN', '>i4', (2,)), ('N_CHAN', '>i4', (2,)), ('MATRIX', '>i4', (2,))]])

Swift-XRT / NuSTAR. *ogip format*

Output of xrtproducts: on / off spectra + arf (+rmf from *Heasarc*)

```
Filename: reproc/00036384112/XRT/swxpc0to12s6_20210101v016.rmf
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      10   ()
  1  MATRIX        1 BinTableHDU     78   2400R x 6C   [D, D, I, PJ(189), PJ(189), PE(982)]
  2  EBOUNDS       1 BinTableHDU     46   1024R x 3C   [I, D, D]
```

[('ENERG_LO', '>f8'), ('ENERG_HI', '>f8'), ('N_GRP', '>i2'), ('F_CHAN', '>i4', (2,)), ('N_CHAN', '>i4', (2,)), ('MATRIX', '>i4', (2,))])

- *N_GRP: number of channel groups for each energy bin*
- *F_CHAN: first channel number in the energy range that the corresponding row in the response matrix*
- *N_CHAN: It indicates the number of channels over which the given row of the response matrix applies*

e.g. element number 100

```
(0.5, 0.5049999952316284, 5, array([ 24, 109, 114, 121, 135], dtype=int32),
array([84,  4,  1,  1,  1], dtype=int32),
array([1.0009960e-06, 5.6055775e-05, 3.6436255e-04, 7.2372012e-04,
       9.1591134e-04, 1.1551494e-03, 1.3733665e-03, 1.9349252e-03,
       2.3683566e-03, 2.9699551e-03, 3.8478286e-03, 4.7867629e-03,
       5.9098802e-03, 7.1741384e-03, 8.8468026e-03, 1.0652599e-02,
       1.2379318e-02, 1.4683610e-02, 1.6885802e-02, 1.8872779e-02,
       2.1070966e-02, 2.2509396e-02, 2.3899781e-02, 2.4859736e-02,
       2.5050925e-02, 2.4776652e-02, 2.4029911e-02, 2.2407295e-02,
       2.0509407e-02, 1.8115025e-02, 1.5207131e-02, 1.2858794e-02,
       1.0418367e-02, 8.3302883e-03, 6.3282968e-03, 4.7987746e-03,
       3.4744572e-03, 2.5595468e-03, 1.8618526e-03, 1.4084014e-03,
       1.0450399e-03, 7.7577191e-04, 6.1160856e-04, 4.9048802e-04,
       3.6936751e-04, 2.9629481e-04, 2.3923804e-04, 2.0220119e-04,
       1.4714641e-04, 1.1311255e-04, 8.7086650e-05, 6.0059760e-05,
       5.4053784e-05, 3.7036851e-05, 2.3022907e-05, 1.8017929e-05,
     …], dtype=float32))
```

Swift-XRT / NuSTAR. *ogip format*

converted into **gammapy-ready** ogip.

rebinning / grouping
(wstat limitations)

```
Filename: Datasets/Swift-XRT/DL4/pha_obssw00030976001_xrt.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4   ()
  1  SPECTRUM       1 BinTableHDU      38   33R x 4C    [I, J, L, D]
  2  EBOUNDS        1 BinTableHDU      26   33R x 3C    [I, D, D]
  3  GTI            1 BinTableHDU      18   3R x 2C    [D, D]
```

reg description goes away. Added ebounds.

```
Filename: Datasets/Swift-XRT/DL4/pha_obssw00030976001_xrt_bkg.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4   ()
  1  SPECTRUM       1 BinTableHDU      38   33R x 4C    [I, J, L, D]
  2  EBOUNDS        1 BinTableHDU      26   33R x 3C    [I, D, D]
```

```
Filename: Datasets/Swift-XRT/DL4/pha_obssw00030976001_xrt_arf.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4   ()
  1  SPECRESP       1 BinTableHDU      25   2400R x 3C    [D, D, E]
```

basically the same

```
Filename: Datasets/Swift-XRT/DL4/pha_obssw00030976001_xrt_rmf.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4   ()
  1  MATRIX         1 BinTableHDU      31   2400R x 6C    [E, E, I, PI(5), PI(5), PE(33)]
  2  EBOUNDS        1 BinTableHDU      26   33R x 3C    [I, D, D]
```

same, but rebinned

Swift-XRT / NuSTAR. *ogip format*

converted into **gammapy-ready** ogip.

```
Filename: Datasets/Swift-XRT/DL4/pha_obssw00030976001_xrt_rmf.fits.gz
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU       4   ()
  1  MATRIX        1 BinTableHDU     31   2400R x 6C   [E, E, I, PI(5), PI(5), PE(33)]
  2  EBOUNDS       1 BinTableHDU     26   33R x 3C   [I, D, D]
```

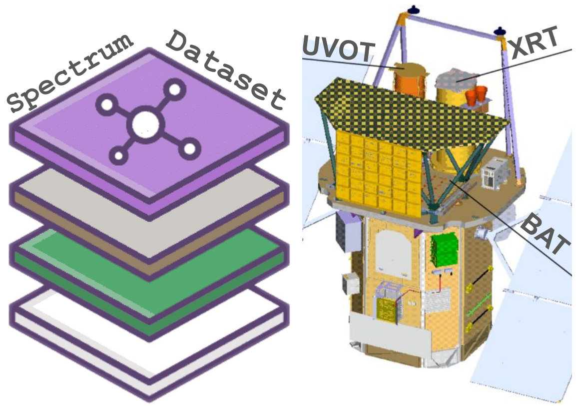
[('ENERG_LO', '>f4'), ('ENERG_HI', '>f4'), ('N_GRP', '>i2'), ('F_CHAN', '>i4', (2,)), ('N_CHAN', '>i4', (2,)), ('MATRIX', '>i4', (2,))]

- *N_GRP: number of channel groups for each energy bin*
- *F_CHAN: first channel that each channel for each channel set*
- *N_CHAN: number of channels in each channel set*

e.g. element number 100

```
(0.5, 0.505, 1, array([0], dtype=int16), array([3], dtype=int16),
array([4.1869560e-01, 1.7593596e-03, 1.1195924e-04], dtype=float32))
```

Swift-UVOT.




```
Swift = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```

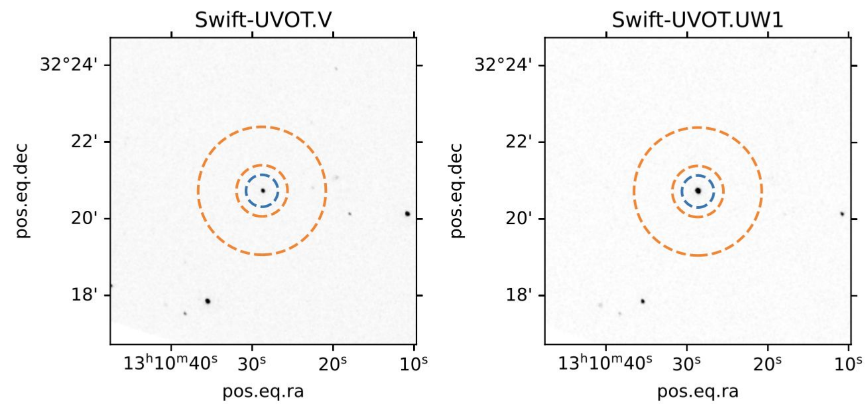
Ingredients (from `uvotproducts/uvot2pha`)

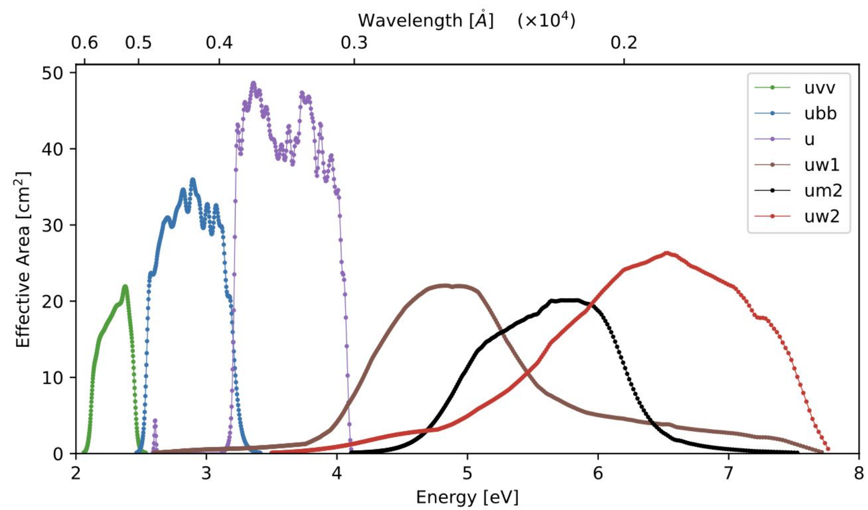
evtfile → source pha (Poisson stats)
bgd → background pha (Poisson stats)
psffile → None! (1D dataset)
expfile → from CALDB
drmfile → None!
model → includes *only* source.

Converter (xspec OGIP → gammapy dataset)

https://github.com/luca-giunti/gammapyXray

http://github.com/mireianievas/gammapy_mwl_workflow





Swift-UVOT. *ogip format*

Output of uvot2pha: on / off ‘spectra’ (+rsp from *Heasarc*)

```
Filename: reproc/00036384001/uvotproduct/sw00036384001ubb_sk.src.pha
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU       7   ()
  1  SPECTRUM      1 BinTableHDU     91   1R x 3C    [I, J, E]
```

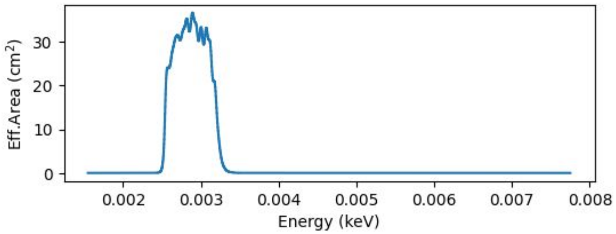
basically a number !

('CHANNEL', '>i2'),
('COUNTS', '>i4'),
('STAT_ERR', '>f4')])

```
Filename: 00036384001/uvotproduct/sw00036384001ubb_sk.bkg.pha
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU       7   ()
  1  SPECTRUM      1 BinTableHDU     91   1R x 3C    [I, J, E]
```

```
Filename: 00036384001/uvotproduct/swubb_20041120v105.rsp
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      11   ()
  1  MATRIX        1 BinTableHDU     53   1282R x 6C    [E, E, I, 1I, 1I, 1E]
  2  EBOUNDS       1 BinTableHDU     46   1R x 3C    [I, E, E]
```

('ENERG_LO', '>f4'),
('ENERG_HI', '>f4'),
('N_GRP', '>i2'),
('F_CHAN', '>i2'),
('N_CHAN', '>i2'),
('MATRIX', '>f4')



Swift-UVOT. *ogip format*

converted into **gammapy-ready** ogip.

```
Filename: Datasets/Swift-UVOT/DL4/pha_obssw00030976001ubb.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4    ()
  1  SPECTRUM       1 BinTableHDU      38    1R x 4C     [I, J, L, D]
  2  EBOUNDS        1 BinTableHDU      26    1R x 3C     [I, D, D]
  3  GTI            1 BinTableHDU      18    1R x 2C     [D, D]
```

need to add backscal (4th value).
this was missing. Fill it from the eff.area dist.
this was missing. Get it from the headers.

```
Filename: Datasets/Swift-UVOT/DL4/pha_obssw00030976001ubb_bkg.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4    ()
  1  SPECTRUM       1 BinTableHDU      38    1R x 4C     [I, J, L, D]
  2  EBOUNDS        1 BinTableHDU      26    1R x 3C     [I, D, D]
```

```
Filename: Datasets/Swift-UVOT/DL4/pha_obssw00030976001ubb_arf.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4    ()
  1  SPECRESP       1 BinTableHDU      25    1282R x 3C     [D, D, D]
```

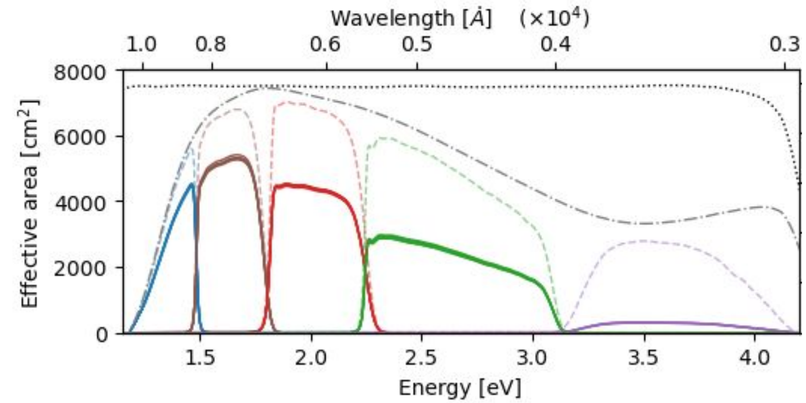
actual effective area.
extracted from the rsp file
('ENERG_LO', '>f8'),
('ENERG_HI', '>f8'),
('SPECRESP', '>f8')])

```
Filename: Datasets/Swift-UVOT/DL4/pha_obssw00030976001ubb_rmf.fits.gz
No.    Name        Ver    Type        Cards   Dimensions   Format
  0  PRIMARY        1 PrimaryHDU        4    ()
  1  MATRIX         1 BinTableHDU      31    1282R x 6C     [E, E, I, PI(1), PI(1), PE(1)]
  2  EBOUNDS        1 BinTableHDU      26    1R x 3C     [I, D, D]
```

there is no migration matrix when you just have a channel,
copy from the rsp and set the MATRIX elements to 1

Optical photometry.

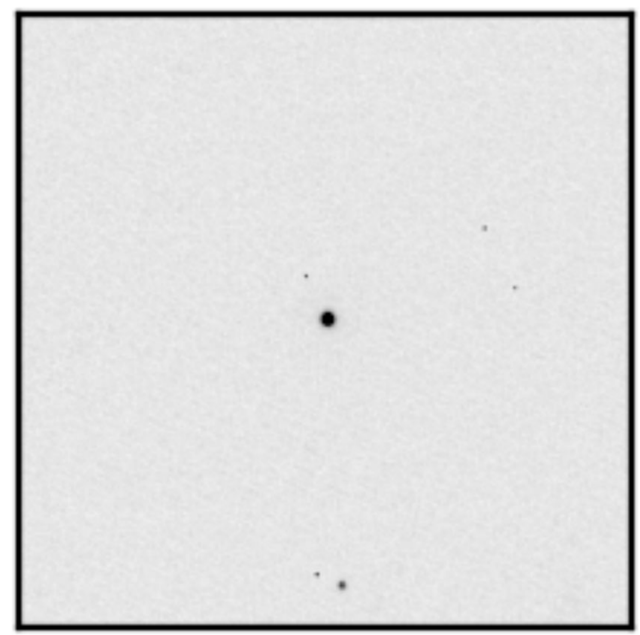

```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



Steps

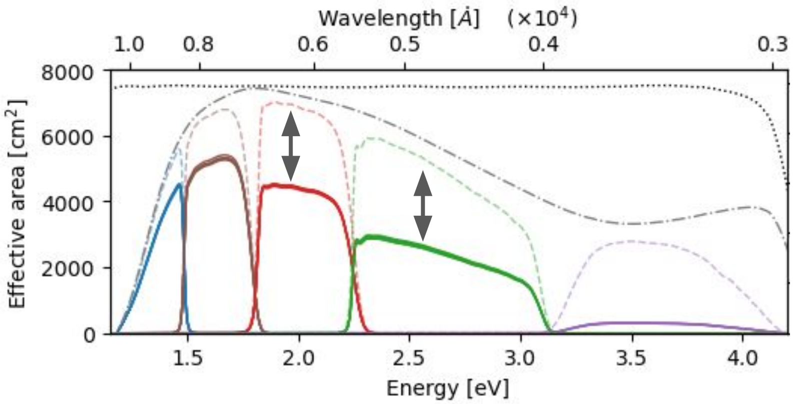
1. **Fetch your images & filter profiles**

h_e_20240314_44_1_1_1.fits

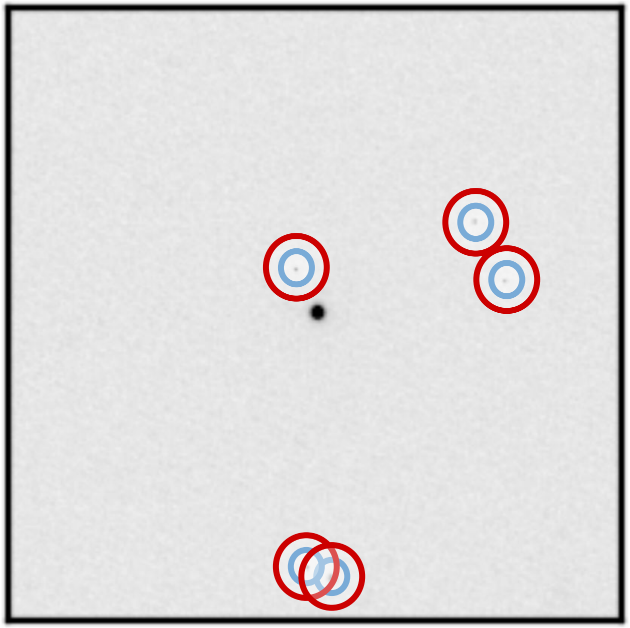


Optical photometry.


```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



h_e_20240314_44_1_1_1.fits

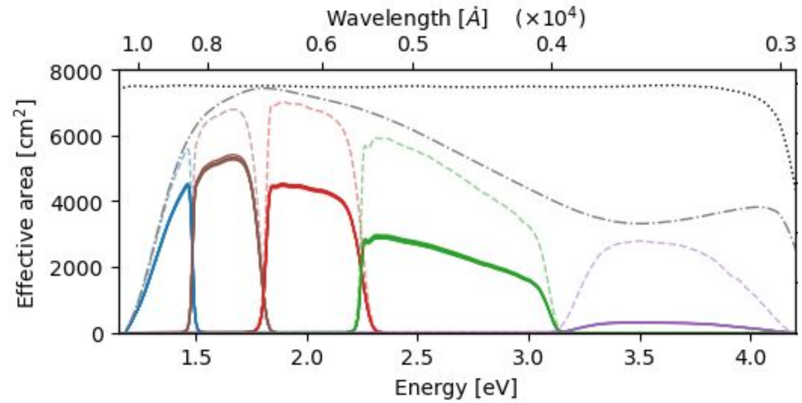


Steps

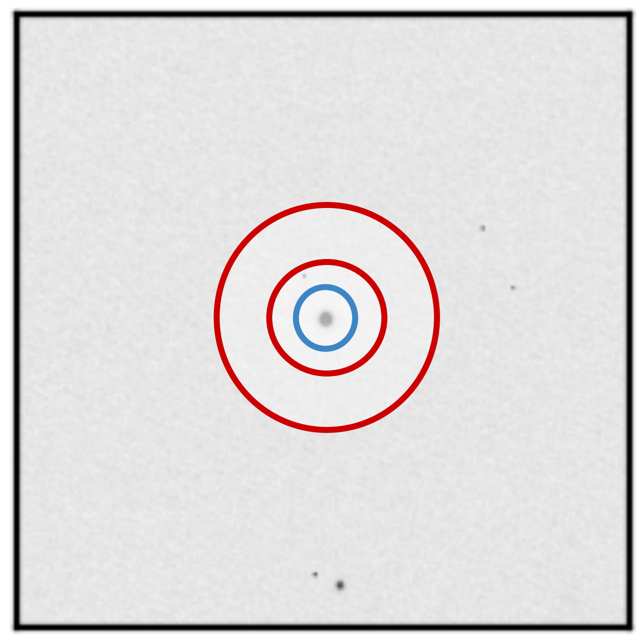
1. Fetch your images & filter profiles
2. **Measure excess counts for calibration stars with known magnitude.**
   gives you the zero point, i.e. eff.area norm.

Optical photometry.


```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



h_e_20240314_44_1_1_1.fits

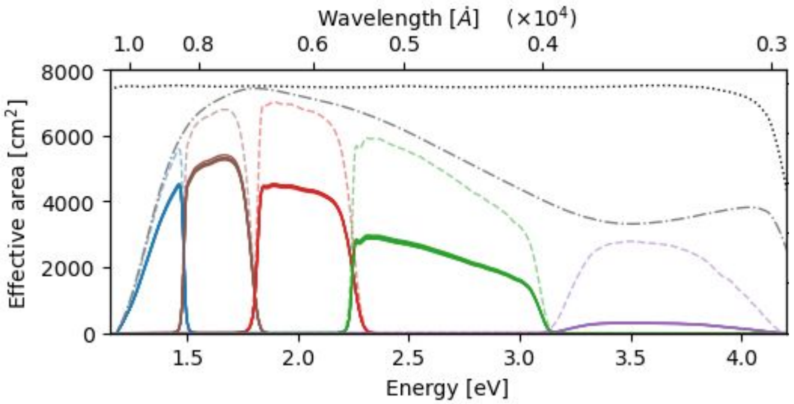


Steps

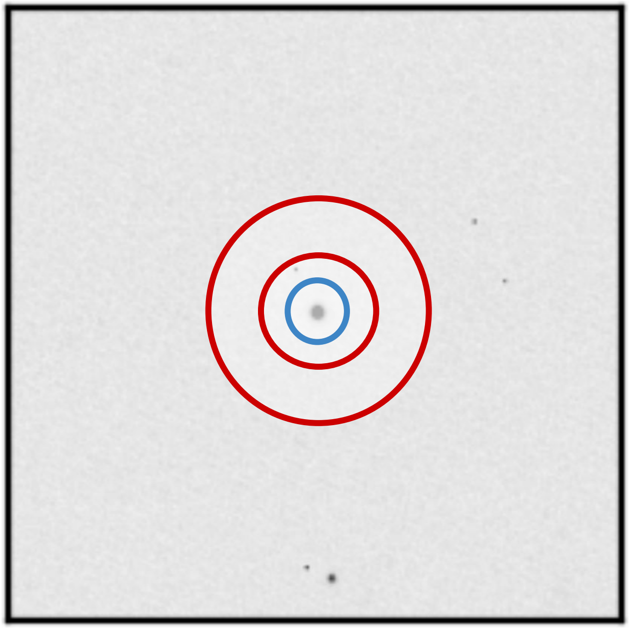
1. Fetch your images & filter profiles
2. Measure excess counts for calibration stars with known magnitude.
   gives you the zero point, i.e. eff.area norm.
3. **Measure ON/OFF counts & α (region size ratio, i.e. acceptances) for your source.**

Optical photometry.


```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



h_e_20240314_44_1_1_1.fits

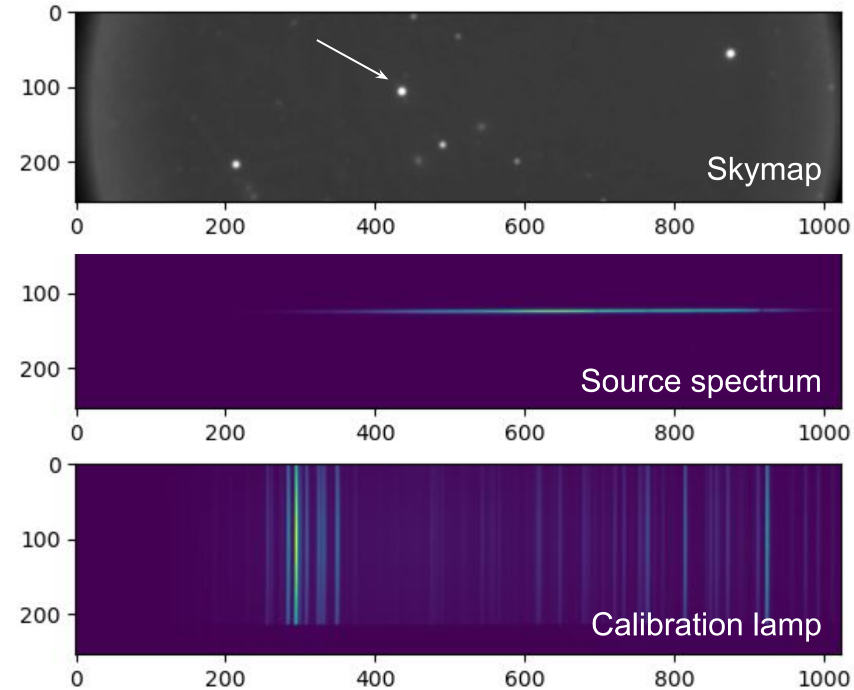


Steps

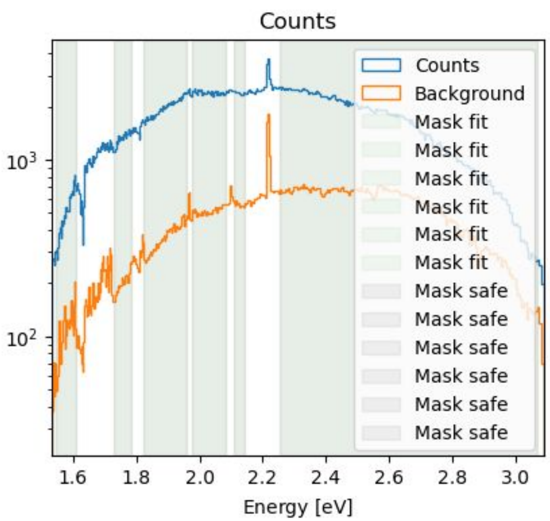
1. Fetch your images & filter profiles
2. Measure excess counts for calibration stars with known magnitude.
   gives you the zero point, i.e. eff.area norm.
3. Measure ON/OFF counts & α (region size ratio, i.e. acceptances) for your source.
4. **Build the dataset**

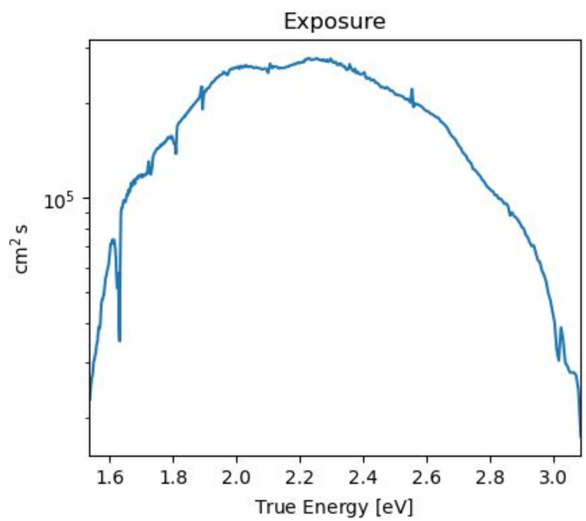
Optical spectroscopy.

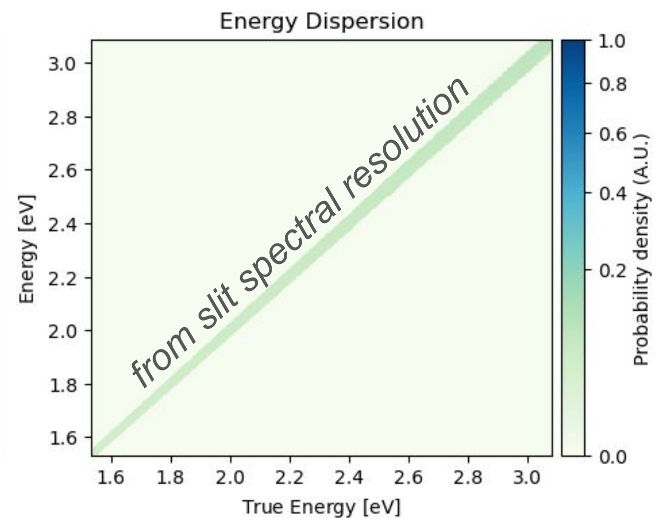

```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



Example: 4C+27.50 (Liverpool *SPRAT*)

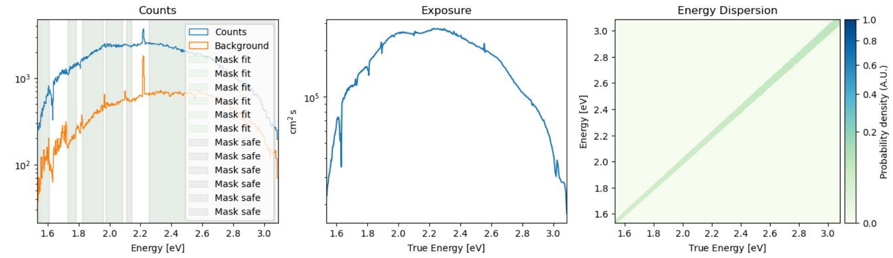






Optical spectroscopy.


```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



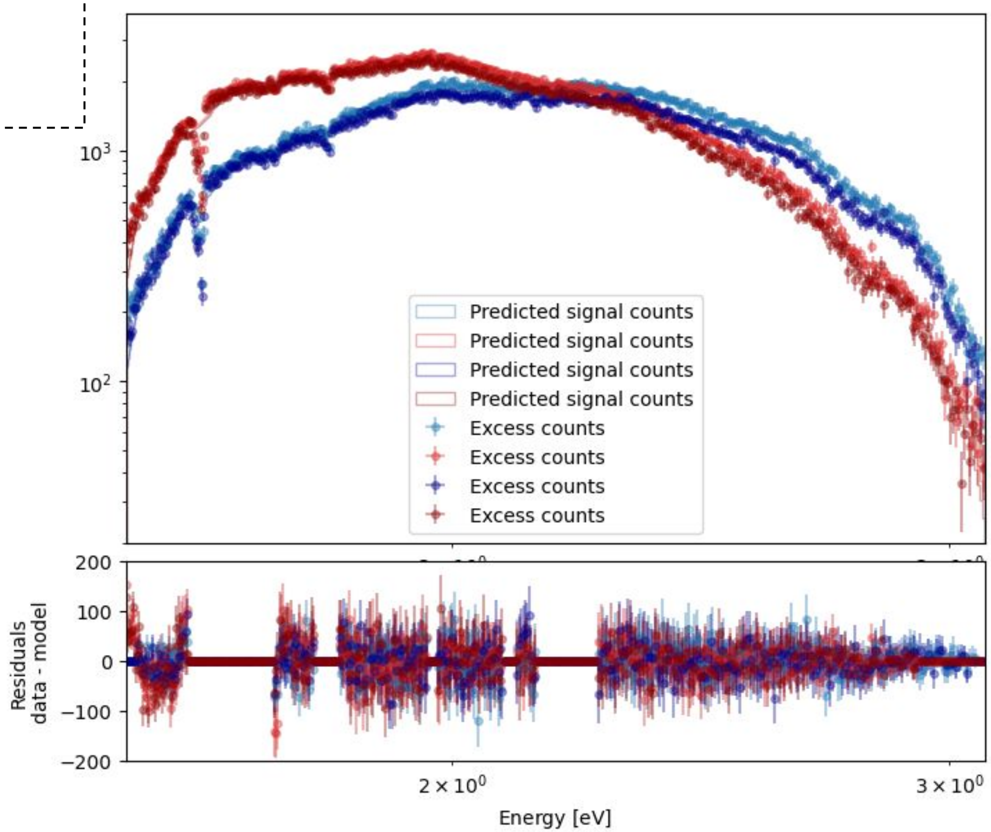
Datasets:

- Two grisms (blue & red) for two nights. Joint fit of the four.

Model:

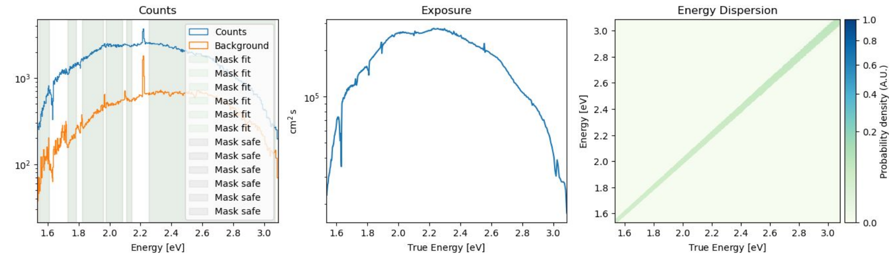
- Source (PowerLaw). Changes from night to night.

- Emission lines (Mg II): constant



Optical spectroscopy.

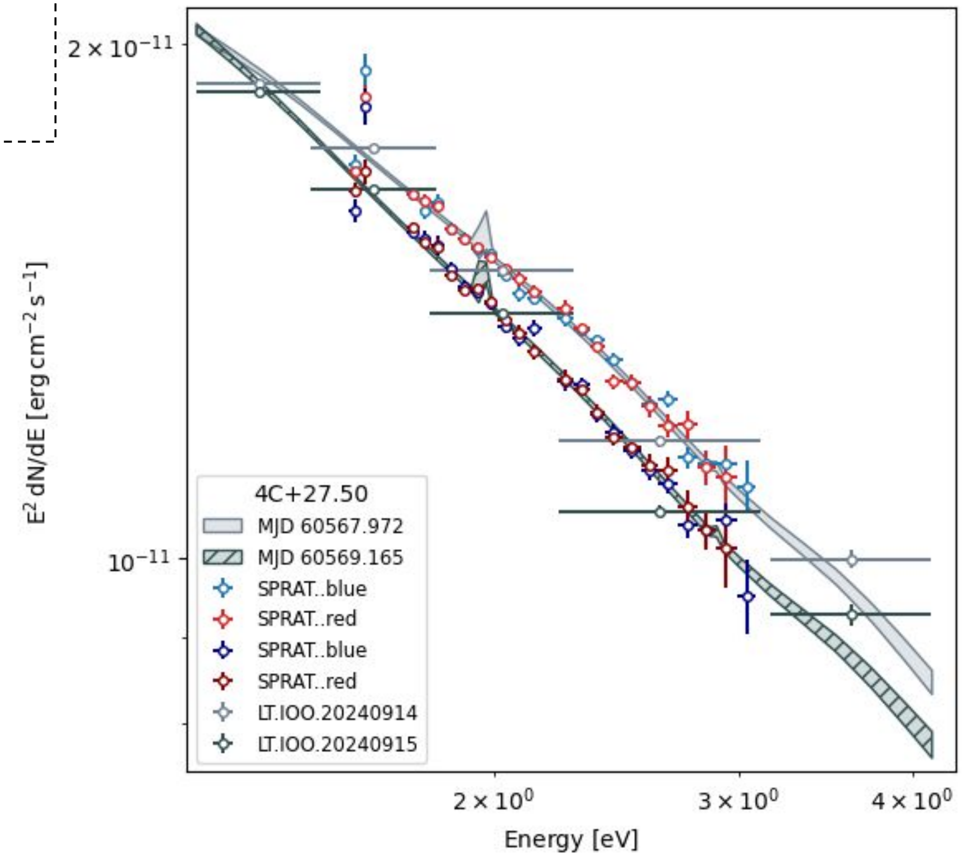

```
Optical = SpectrumDatasetOnOff(
    counts=myreg_on,
    counts_off=myreg_off,
    acceptance=1,
    acceptance_off=1./alpha,
    exposure=myreg_exposure,
    mask_fit=None,
    psf=None,
    edisp=myedisp,
    name=name,
    gti=gti)
```



You can even combine photometry (e.g. Liverpool IO:O) and spectroscopy (Liverpool SPRAT) !

Two nights:

- Night 1: 2 grisms + 5 filters
- Night 2: 2 grisms + 5 filters



Use cases: *MWL modeling*

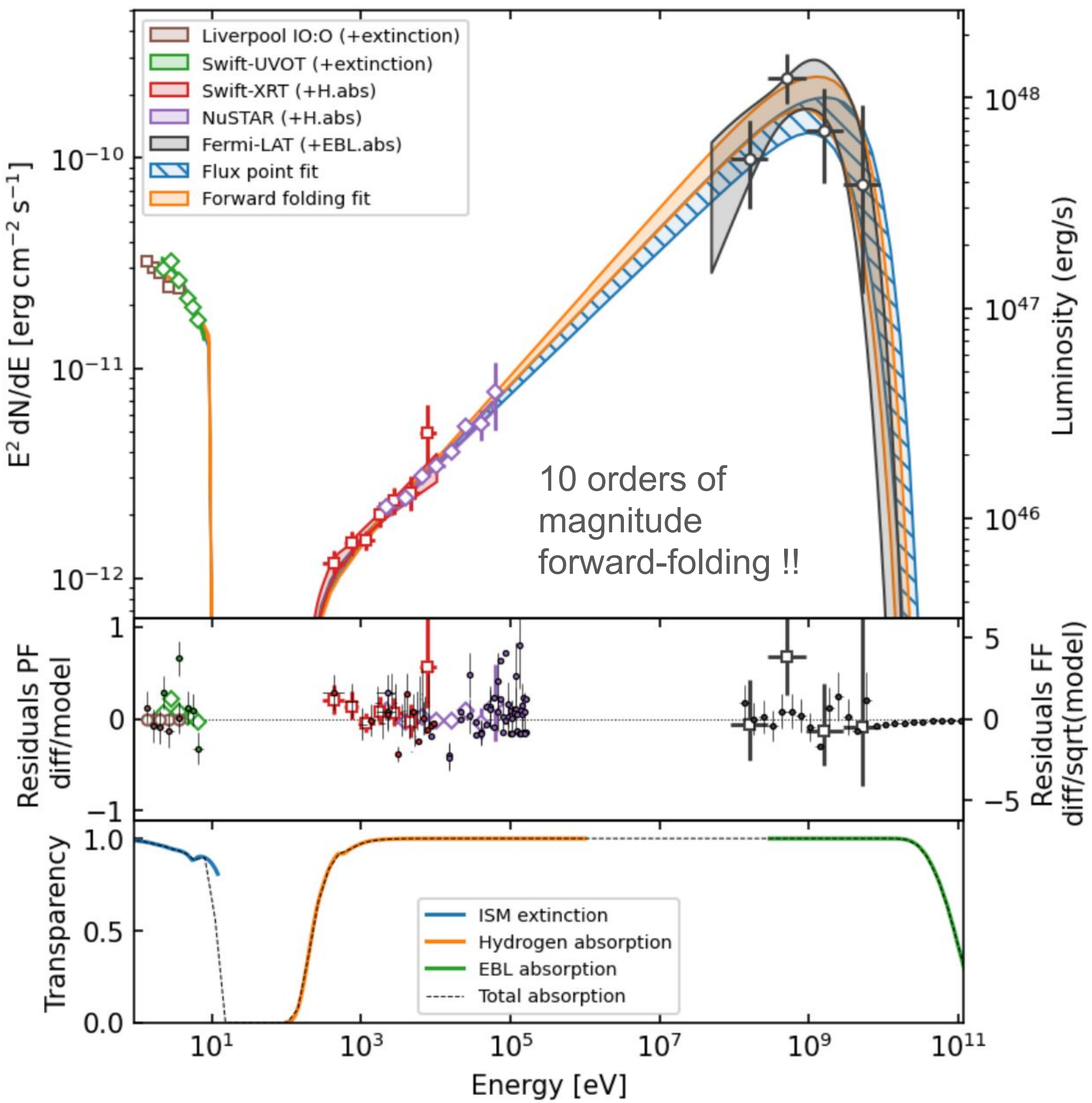
Multi-instrument multi-wavelength forward folding analysis, including:

- multiple sources in LAT + diffuse components
- instrumental background in NuSTAR
- Absorption components: EBL, hydrogen, reddening.

M. Nievas Rosillo et al 2025

https://www.aanda.org/articles/aa/full_html/2025/01/aa52349-24/aa52349-24.html

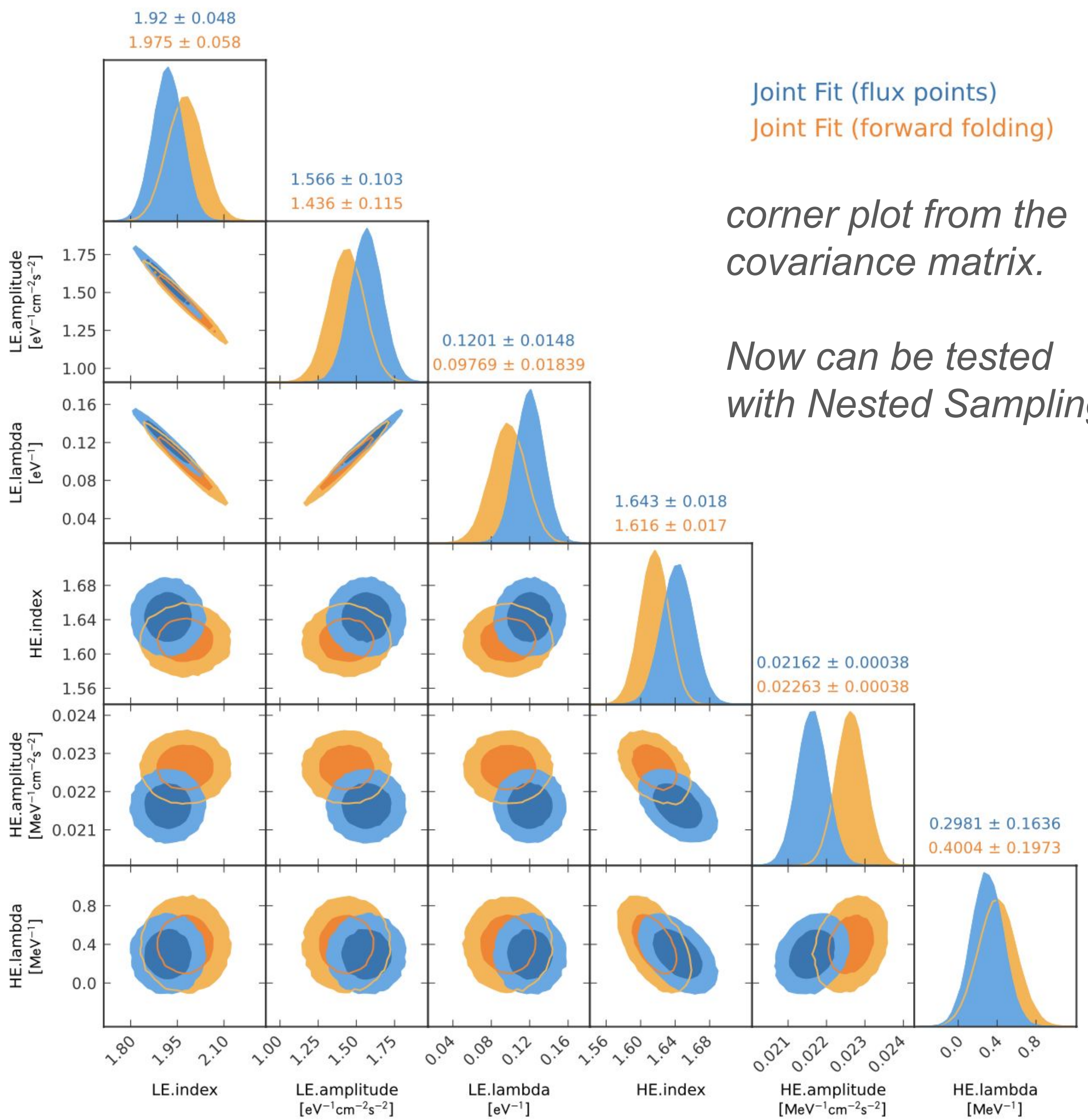
https://zenodo.org/records/13837637



Use cases:
MWL modeling

Multi-instrument multi-wavelength forward folding analysis, including:

- multiple sources in LAT + diffuse components
- instrumental background in NuSTAR
- Absorption components: EBL, hydrogen, reddening.



corner plot from the covariance matrix.

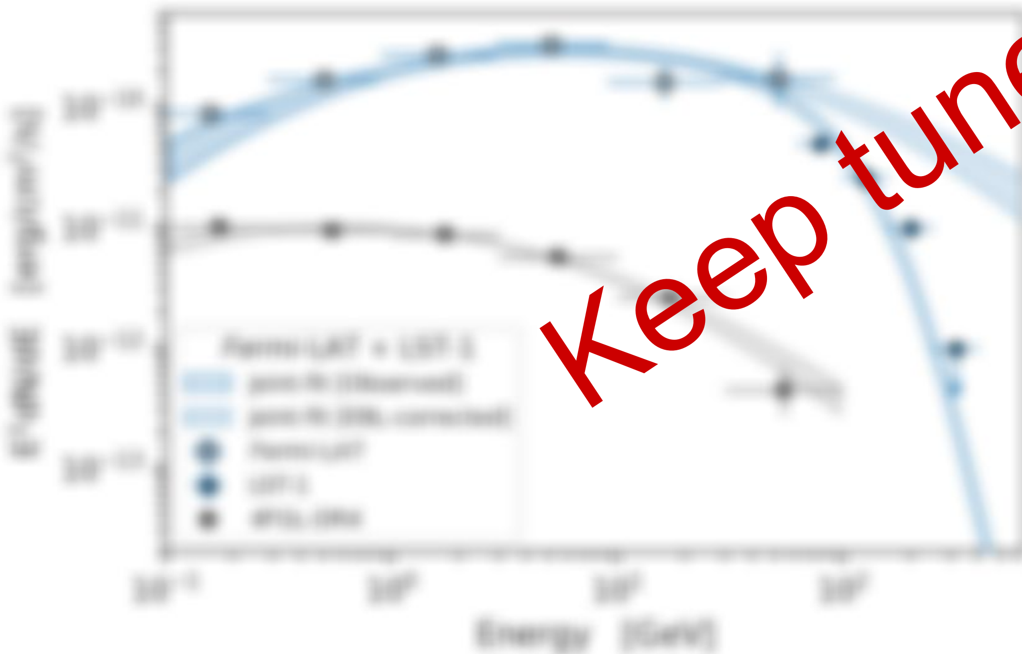
Now can be tested with Nested Sampling.

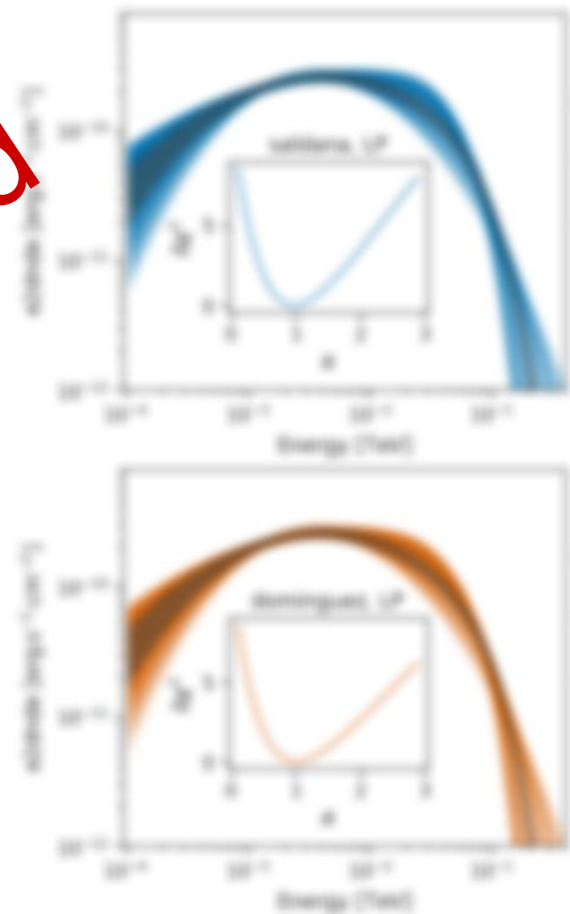
M. Nievas Rosillo et al 2025

https://www.aanda.org/articles/aa/full_html/2025/01/aa52349-24/aa52349-24.html

https://zenodo.org/records/13837637

Use cases:
EBL constraints (LAT + IACTS)





Keep tuned

Conclusions

Gammapy makes it possible to analyse *simultaneously* datasets:

- from multiple bands (optical to γ-rays, ~ 10 orders of magnitude in energy)
- different nature (3D cubes like in LAT, 1D spectra like XRT, even photometry and spectroscopy from optical telescopes).
- with correct statistics (*Poisson* instead of *Gaussian*).
- with better handling of backgrounds, contaminating sources.
- avoiding issues: e.g. flux point correlation.
- allowing to use non-detections.

As a by-product, it offers a flexible data format structure (~ DL4), instrument description and emission model description that can be archived for future usage.

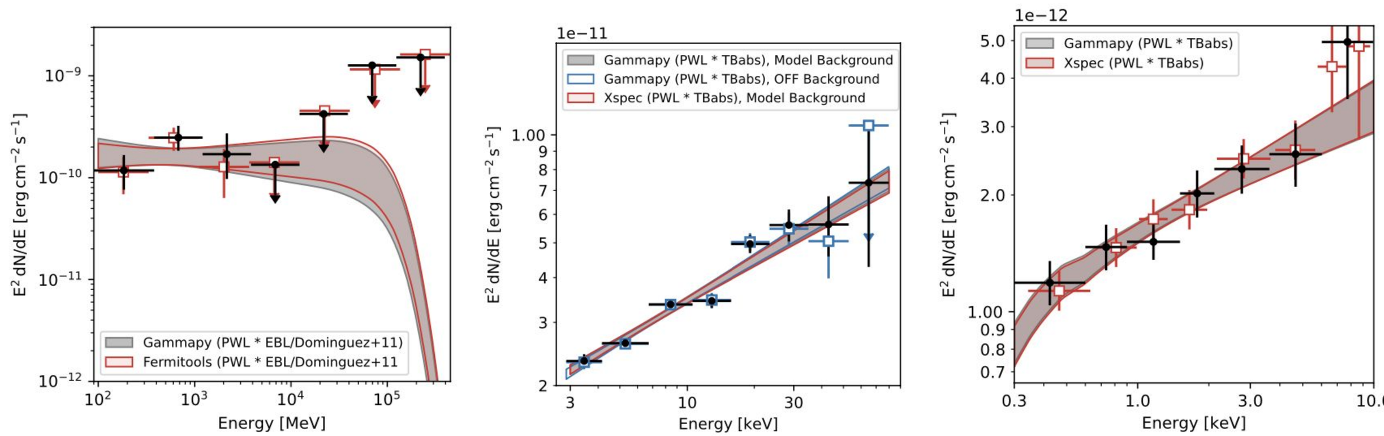
What would be nice

- Systematics for IACTs: *energy-scale, background & acceptance*
- Astrophysical + Instrumental backgrounds. *Looks like now you have to choose?*
- Easier way to digest Fermi-LAT products (and to stack_reduce datasets)
- position-dependent IRFs for Fermi-LAT (DRM, PSF).
- Simpler way to read directly standard ogip files from UVOT/XRT/NuSTAR/XMM.
- We plan to explore X-ray polarimetry (IXPE). May need to work on the data format.

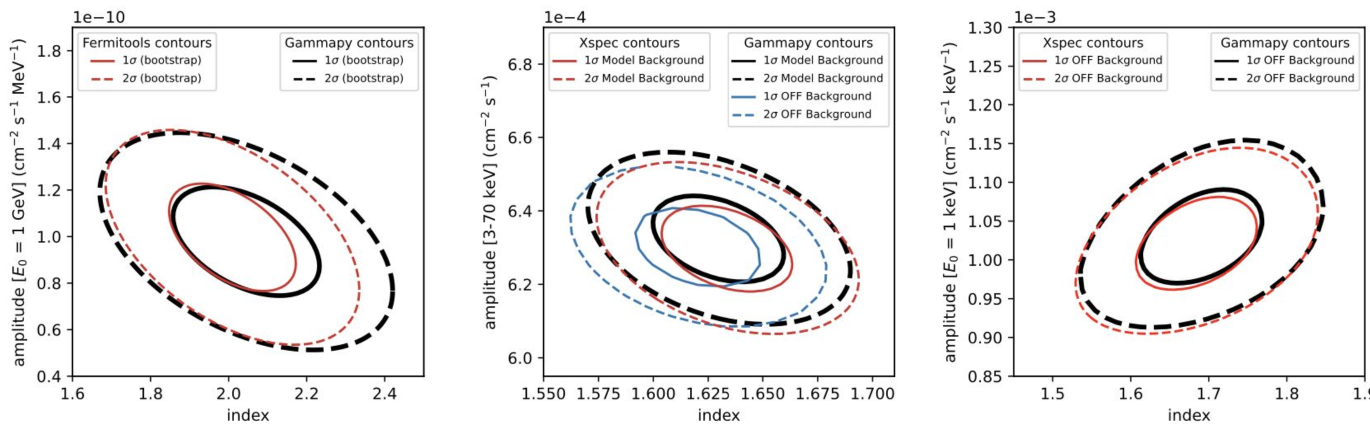
Backup slides …

Validation

	Fermi-LAT		NuSTAR A+B		*Swift*-XRT	
	@1 GeV		3 – 70 keV		@1 keV	
	amplitude	index	integral flux	index	amplitude	index
native	1.00 ± 0.23	2.01 ± 0.16	6.30 ± 0.12	1.633 ± 0.030	10.20 ± 0.60	1.683 ± 0.077
gammapy	0.98 ± 0.23	2.05 ± 0.19	6.32 ± 0.12	1.630 ± 0.030	10.29 ± 0.60	1.690 ± 0.078
scale	$\times 10^{-11}$		$\times 10^{-4}$		$\times 10^{-4}$	
units	$\mathrm{MeV^{-1}s^{-1}cm^{-2}}$		$\mathrm{s^{-1}cm^{-2}}$		$\mathrm{keV^{-1}s^{-1}cm^{-2}}$	

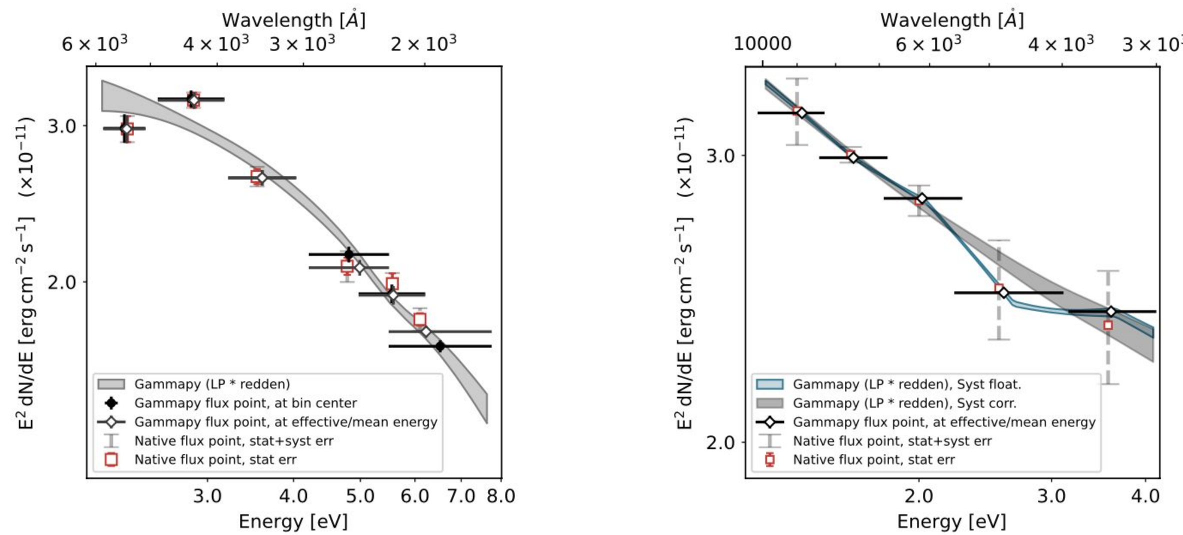


(a) *Fermi*-LAT. (b) NuSTAR (c) *Swift*-XRT

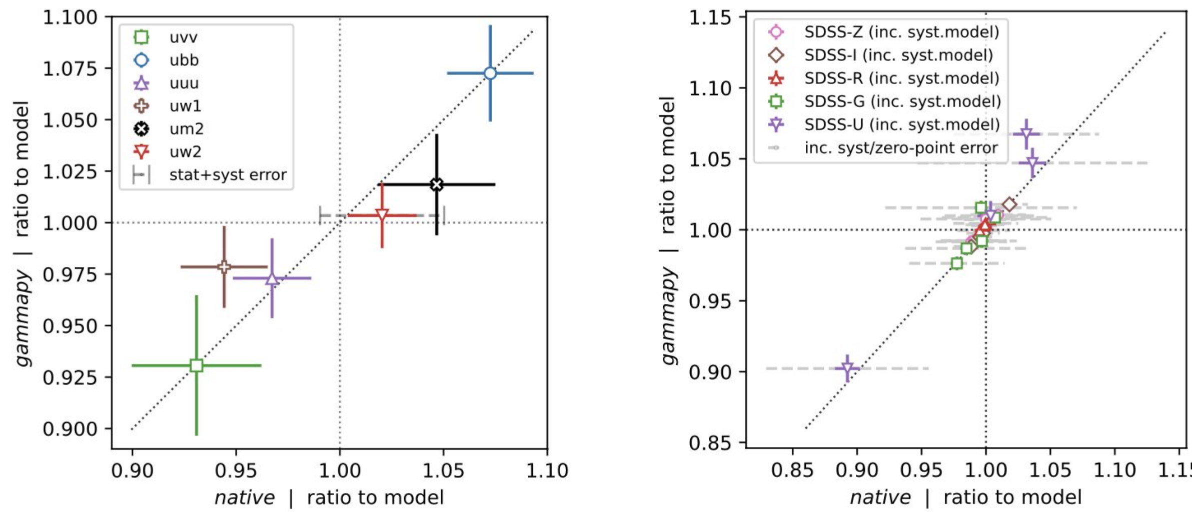


(a) *Fermi*-LAT. (b) NuSTAR (c) *Swift*-XRT

Validation



(a) *Swift*-UVOT (b) Liverpool Telescope IO:O



(a) *Swift*-UVOT (b) Liverpool Telescope IO:O

Optical photometry.

What if … you already have the photometry

1. from the definition of excess (assuming a PWL spectrum)

$$\text{excess} \equiv \int_{\text{Emin}}^{\text{Emax}} \Phi(E)\Psi(E)dE \approx \Phi_0\Psi_0 \int_{\text{Emin}}^{\text{Emax}} E^{-\alpha}T(E)dE$$

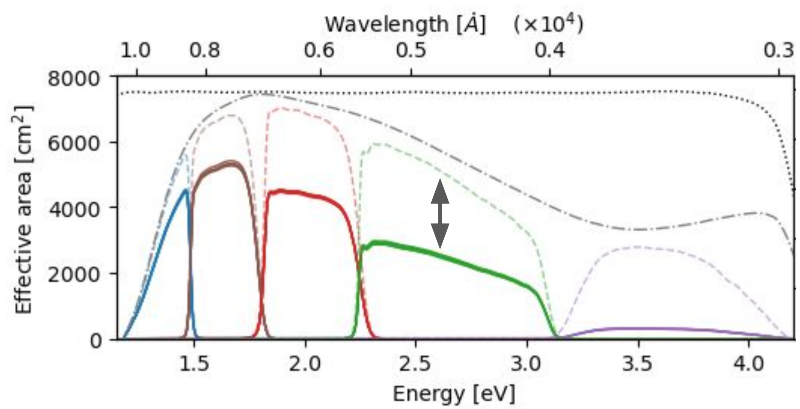
2. from the definition of “energy flux”:

$$\text{eflux} \equiv \int_{\text{Emin}}^{\text{Emax}} E\Phi(E)dE \approx \Phi_0 \int_{\text{Emin}}^{\text{Emax}} E^{1-\alpha}dE \approx \text{W}_{\text{eff}} \times \text{fluxdensity}$$

3. The ‘normalization’ of the effective area vs the transmission becomes:

$$\Psi_0 = \frac{\text{excess}}{\text{W}_{\text{eff},\lambda} \times \text{fluxdensity}} \frac{\int_{\text{Emin}}^{\text{Emax}} E^{1-\alpha}dE}{\int_{\text{Emin}}^{\text{Emax}} E^{-\alpha}T(E)dE}$$

$$\text{W}_{\text{eff},\lambda} = \frac{\int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} T(\lambda)\times\lambda^{\alpha}d\lambda}{\int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} \lambda^{\alpha}d\lambda} \times [\lambda_{\text{max}} - \lambda_{\text{min}}]$$



Cofinanciado por la Unión Europea

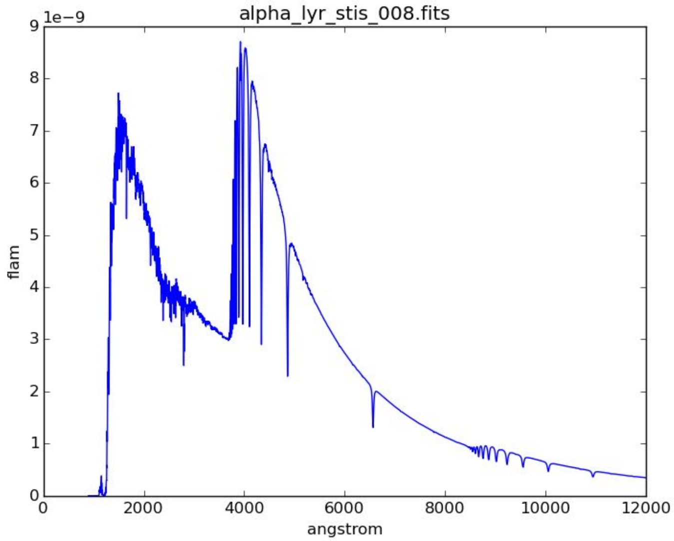
Optical photometry.

What if … you already have the photometry

How to convert

1. excess to Vega magnitudes:

$$\mathrm{mag} = -2.5\log_{10}(\mathrm{excess}) + \mathrm{ZP}_{\mathrm{mag}}$$



2. Vega magnitudes to flux densities

$$\mathrm{fluxdensities} = \mathrm{VegaZP} \times 10^{-0.4\times\mathrm{mag}}$$

$$\mathrm{VegaZP} = \frac{\int_{\lambda_{\mathrm{min}}}^{\lambda_{\mathrm{max}}} \Phi_{\mathrm{Vega},\lambda}(\lambda)T(\lambda)d\lambda}{\int_{\lambda_{\mathrm{min}}}^{\lambda_{\mathrm{max}}} T(\lambda)d\lambda}$$

Optical photometry.

What if … you already have the photometry

In summary, what you (minimally) need:

- ON, OFF, ratio of integration region sizes
- Zero Point (counts to mag)
- Filter/band properties (e.g. from SVO)
- Vega spectrum (e.g. STIS)

Big caveat … we are ignoring here a few important details:

1. Gain! CCDs give us Poisson statistics, in p.e., not in counts. One would need to know the gain to do it properly.
2. Atmosphere … only considered as part of the differential photometry. Changes within the filter are ignored.
3. Exposure does not follow the filter transmission, but a combination of filter, CCD (efficiency), cryostat window, atmosphere!.
4. Colors of calibration star and our source (color terms).