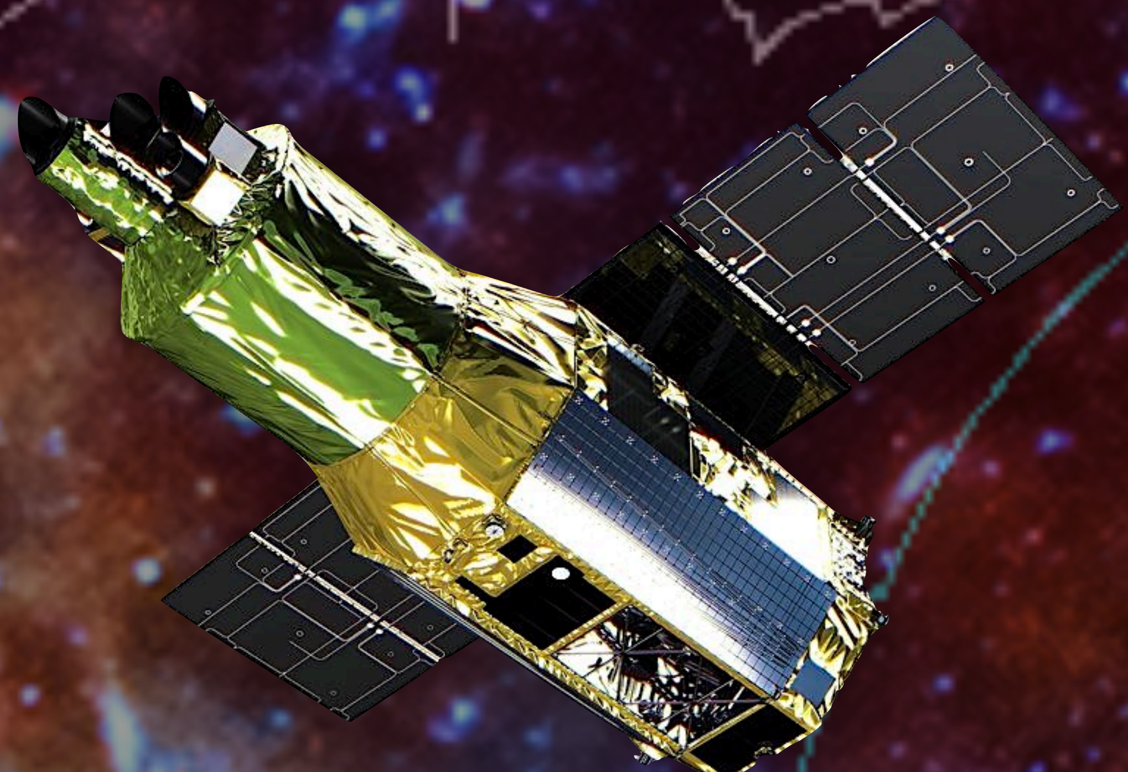
18 January 2024, *The 2nd XRISM Community Workshop, University of Maryland College Park (USA)*

Make your own XRISM responses

François Mernier

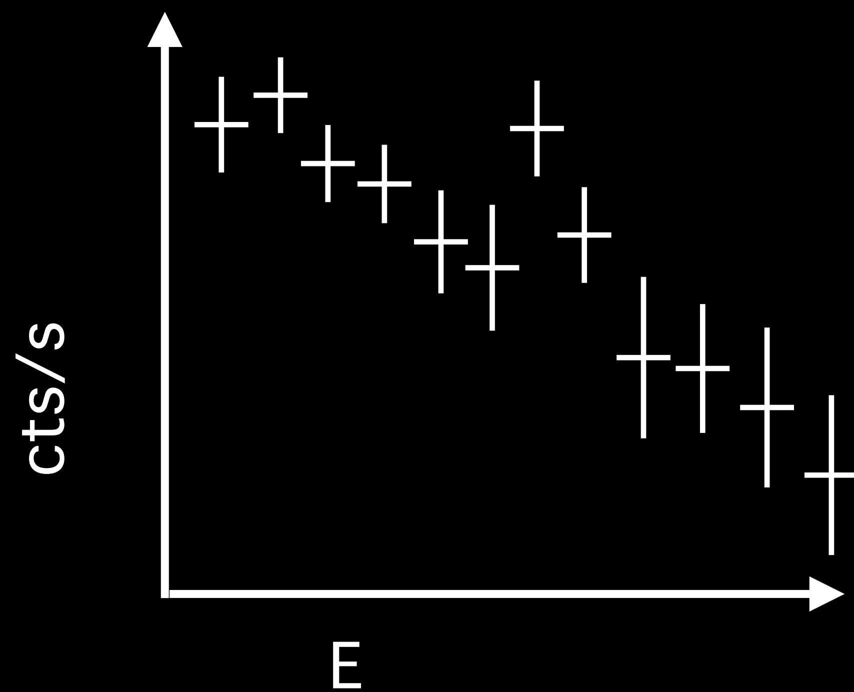
fmernier@umd.edu



Reminder

$$C(k) = \int_0^\infty \sum_i A_i(E)\, R_i(k,E)\, S_i(E)\, dE\, dT \;+\; B_i(k)$$

Spec =

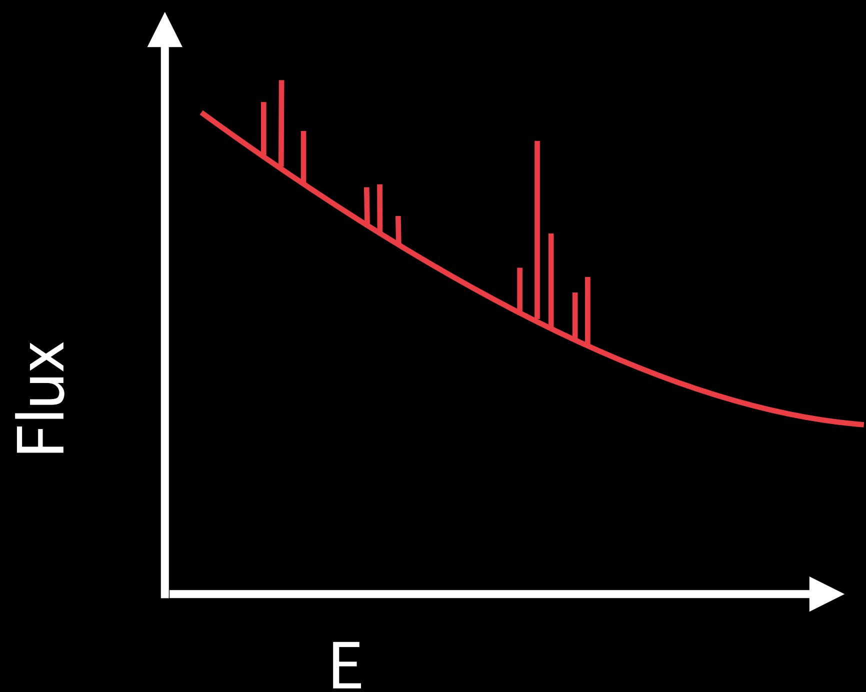


- ✓ **Spec** = observed ***spectrum***
- ✓ **Mod** = ***spectral model***
- ✓ **RMF** = ***response matrix***
- ✓ **ARF** = ***effective area***

Reminder

$$C(k) = \int_0^\infty \sum_i A_i(E)\, R_i(k,E)\, S_i(E)\, dE\, dT \; + \; B_i(k)$$

Spec = **Mod**



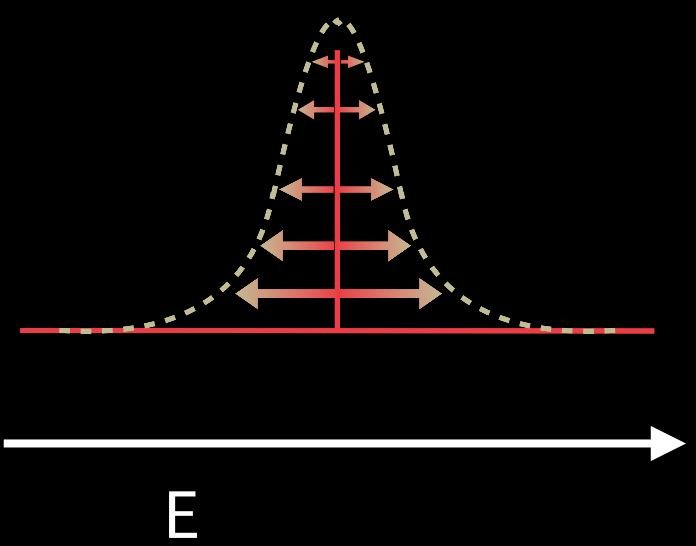
- ✓ **Spec** = observed ***spectrum***
- ✓ **Mod** = ***spectral model***
- ✓ **RMF** = ***response matrix***
- ✓ **ARF** = ***effective area***

Reminder

$$C(k) = \int_0^\infty \sum_i A_i(E)\, R_i(k, E)\, S_i(E)\, dE\, dT \; + \; B_i(k)$$

Spec = { **RMF** * **Mod** }

- ✓ **Spec** = observed ***spectrum***
- ✓ **Mod** = ***spectral model***
- ✓ **RMF** = ***response matrix***
- ✓ **ARF** = ***effective area***

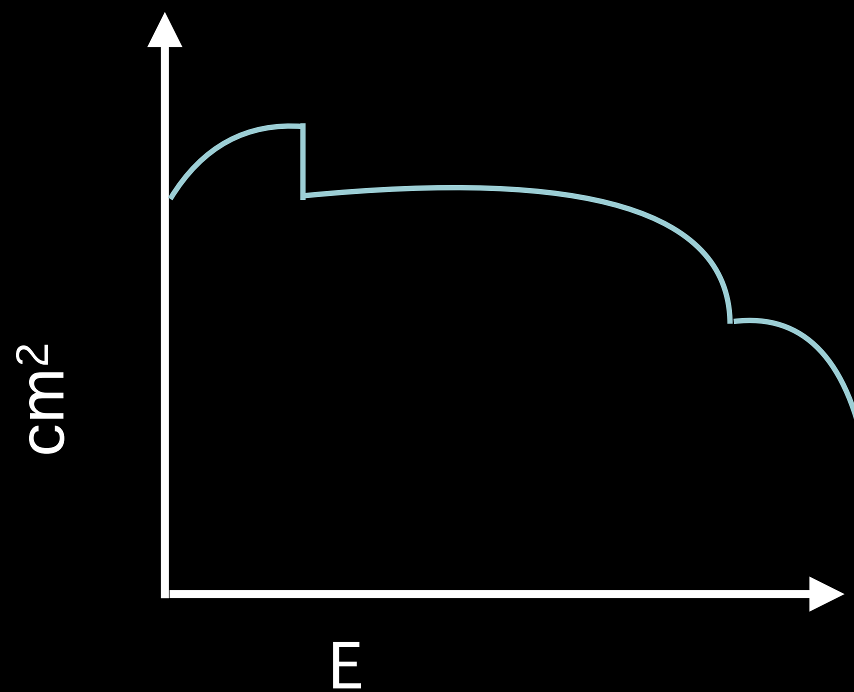


Reminder

$$C(k) = \int_0^\infty \sum_i A_i(E)\, R_i(k,E)\, S_i(E)\, dE\, dT \;+\; B_i(k)$$

Spec = ARF * { RMF * Mod }

- ✓ **Spec** = observed ***spectrum***
- ✓ **Mod** = ***spectral model***
- ✓ **RMF** = ***response matrix***
- ✓ **ARF** = ***effective area***



Reminder

Resolve

RMFs

- ✓ rsl_Hp_5eV.rmf
- ✓ rsl_Mp_6eV.rmf
- ✓ rsl_Lp_18eV.rmf

ARFs (GV closed)

- ✓ rsl_standard_GVclosed.arf
- ✓ rsl_pointsource_fwBe_GVclosed.arf
- ✓ rsl_pointsource_fwND_GVclosed.arf
- ✓ rsl_pointsource_off_GVclosed.arf
- ✓ rsl_extflat_GVclosed.arf
- ✓ rsl_extbeta_GVclosed.arf

Xtend

RMFs

- ✓ xtd_standard.rmf

ARFs

- ✓ xtd_standard.arf
- ✓ xtd_extflat.arf

Reminder

Resolve

"I want an RMF for Hp events at pixel 27 exactly."

"I want an RMF for Hp events at pixel 27 with electron loss continuum."

"I want an ARF for a point source located exactly on the detector's upper right corner."

"I want an ARF for a beta extended source with other parameters than provided."

Xtend

"I want an ARF for a circular source with 1 arcmin radius and 2 arcmin off-axis."

The solution…

Make RMFs (and ARFs) from real data! 😉

…But we are only at Cycle 1. No data is available yet!

Fair point… (Then let's dig into the XRISM software and CALDB.)

Warning: we encourage you to make your own responses ONLY if you have a good reason to do so!

(Remember, canned responses are provided too…)

Generating Resolve responses

What you need:

✓ **rmfparam file**: File containing basic RMF parameters. (Available in CALDB)

✓ **rsl_1att_b7optaxis.expo:** Dummy exposure map file for making Resolve non-observation ARFs. OPEN filter, gate valve OPEN. (Provided separately)

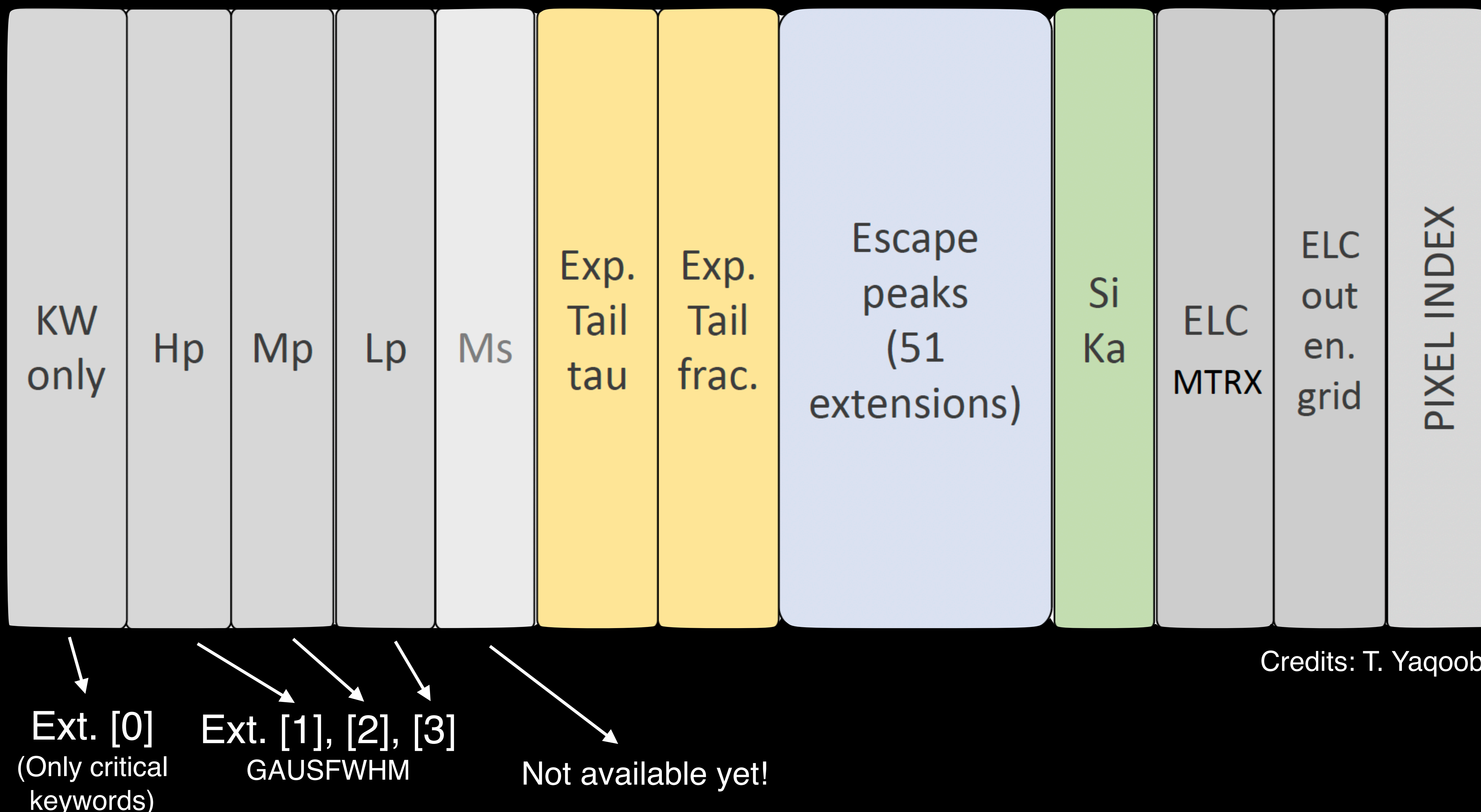
✓ **rsl_35pix_det.reg:** Resolve region file for all pixels, in DET coordinates, used to make non-observation ARFs. (Provided separately)

✓ **rsl_onaxiscfile_0p3to18kev.fits:** Energy grid file needed for raytracing to make canned ARFs. (Provided separately)

Generating Resolve RMFs

Anatomy of the rmfparam file

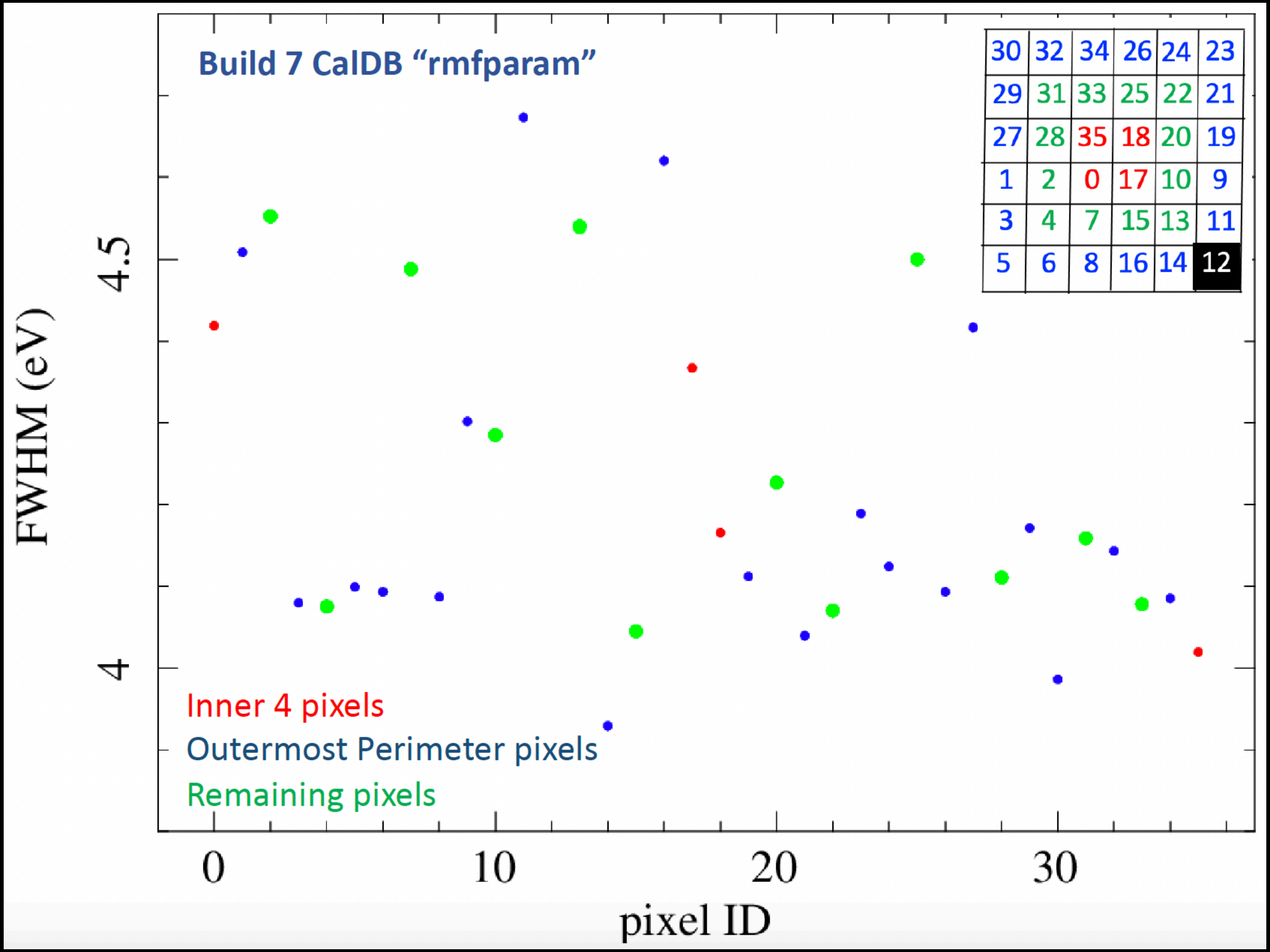
(e.g. /path/to/CALDB/data/xrism/resolve/bcf/response/**xa_rsl_rmfparam_20190101v005.fits.gz**)



Credits: T. Yaqoob

Generating Resolve RMFs

Anatomy of the rmfparam file
(e.g. `/path/to/CALDB/data/xrism/resolve/bcf/response/`**`xa_rsl_rmfparam_20190101v005.fits.gz`**)

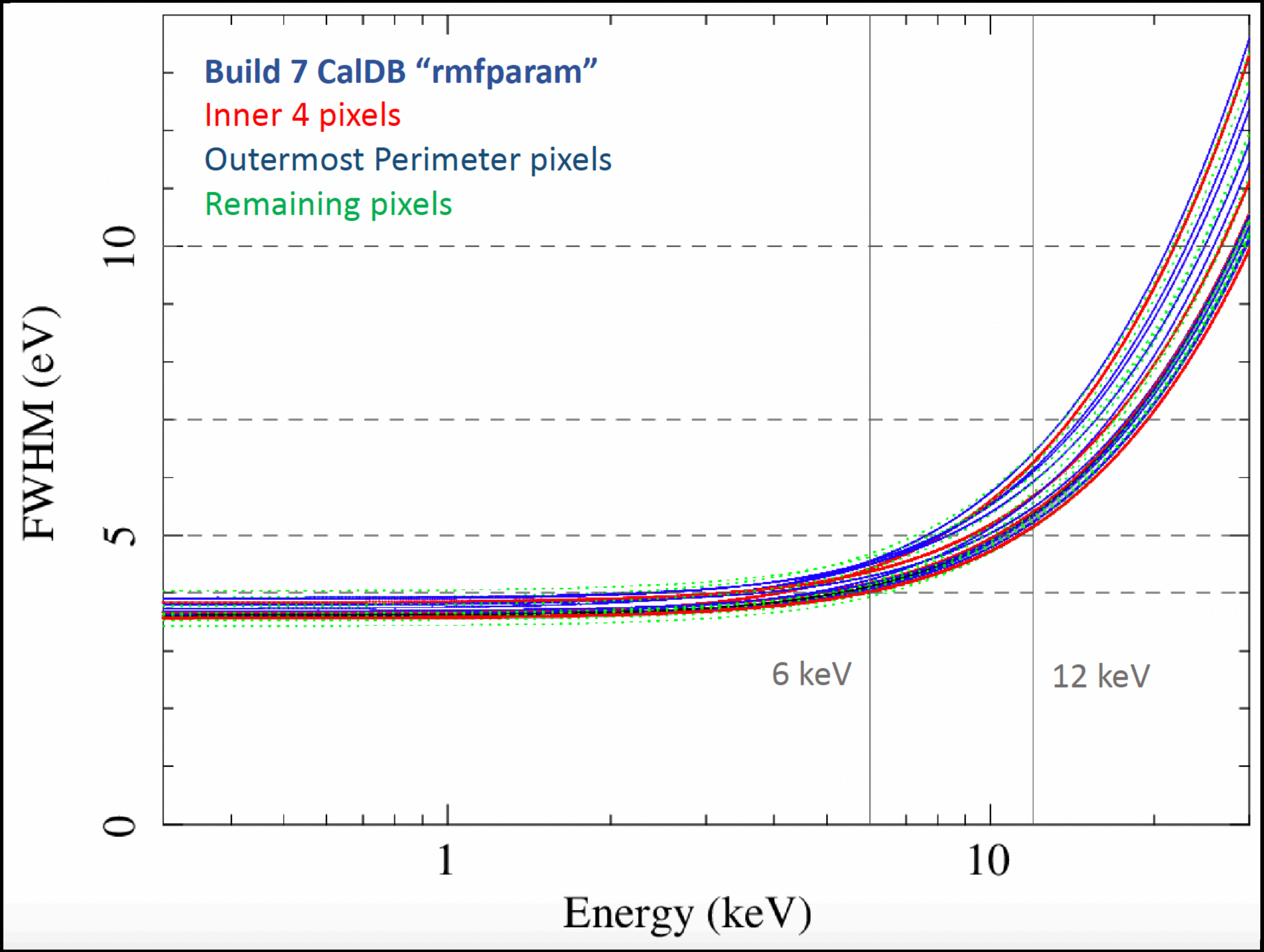


Credits: T. Yaqoob

Generating Resolve RMFs

Anatomy of the rmfparam file

(e.g. /path/to/CALDB/data/xrism/resolve/bcf/response/**xa_rsl_rmfparam_20190101v005.fits.gz**)



Credits: T. Yaqoob

Generating Resolve RMFs

I want an RMF for Hp events at pixel 27 exactly.

```
$ fhelp rslrmf
```

Set to NONE to select a pixel value directly

```
$ rslrmf infile=NONE outfile="rsl_Hp_pix27" pixel=27
  resol=Hp rmfparamfile="CALDB"
```

Name of my output file (.rmf extension added automatically)

Either Hp (= GAUSFWHM1), Mp (= GAUSFWHM2), or Lp (= GAUSFWHM3)

If set to CALDB, the RMF will take info from the latest rmfparamfile in the CALDB directory

I want an RMF with a resolution of 10 eV exactly.

```
$ rslrmf infile=NONE outfile="rsl_10eV" pixel=27
  resol=Hp rmfparamfile="my_edited_rmfparam_file.fits"
```

Modified rmfparam file with all values of all pixel columns (or, at the very least, pixel 27) in GAUSFWHM1 are set to 10 (using ftcalc, python,...)

Generating Resolve RMFs

I want an RMF for Hp events at pixel 27 with electron loss continuum.

```
$ rslrmf infile=NONE outfile="rsl_Hp_pix27_XL" pixel=27
  resol=Hp rmfparamfile="CALDB" whichrmf=X splitrmf=yes
  elcbinfac=32
```

Coarser bin on the ELC part of the matrix

Creates an Xtra Large RMF that includes ELC

Splits the RMF into two files (necessary to avoid a >2 GB size)

whichrmf=S / M / L / X

- S: Gaussian core …
- M: … + exponential tail …
- L: … + escape peaks + Si Kα …
- X: … + electron loss continuum

Caveats (RMFs)

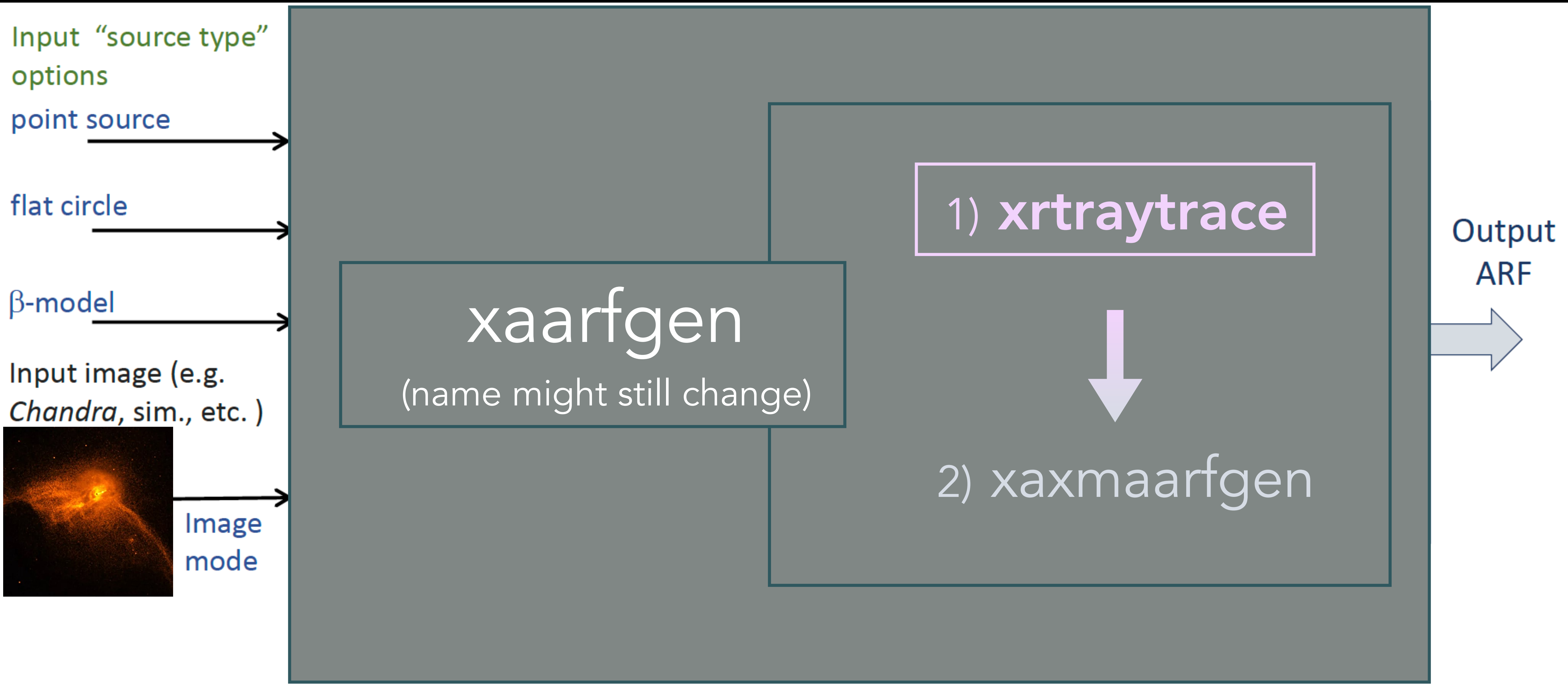
✓ These RMFs do **NOT** account for **branching ratios**! (i.e. they assume only the values given at energies and pixels from the rmfparam file)

✓ Can**NOT** be representative of the **entire detector** (including all pixels) because the sum of Gaussians is **NOT** a Gaussian!

Then how do I do to get a super realistic RMF?

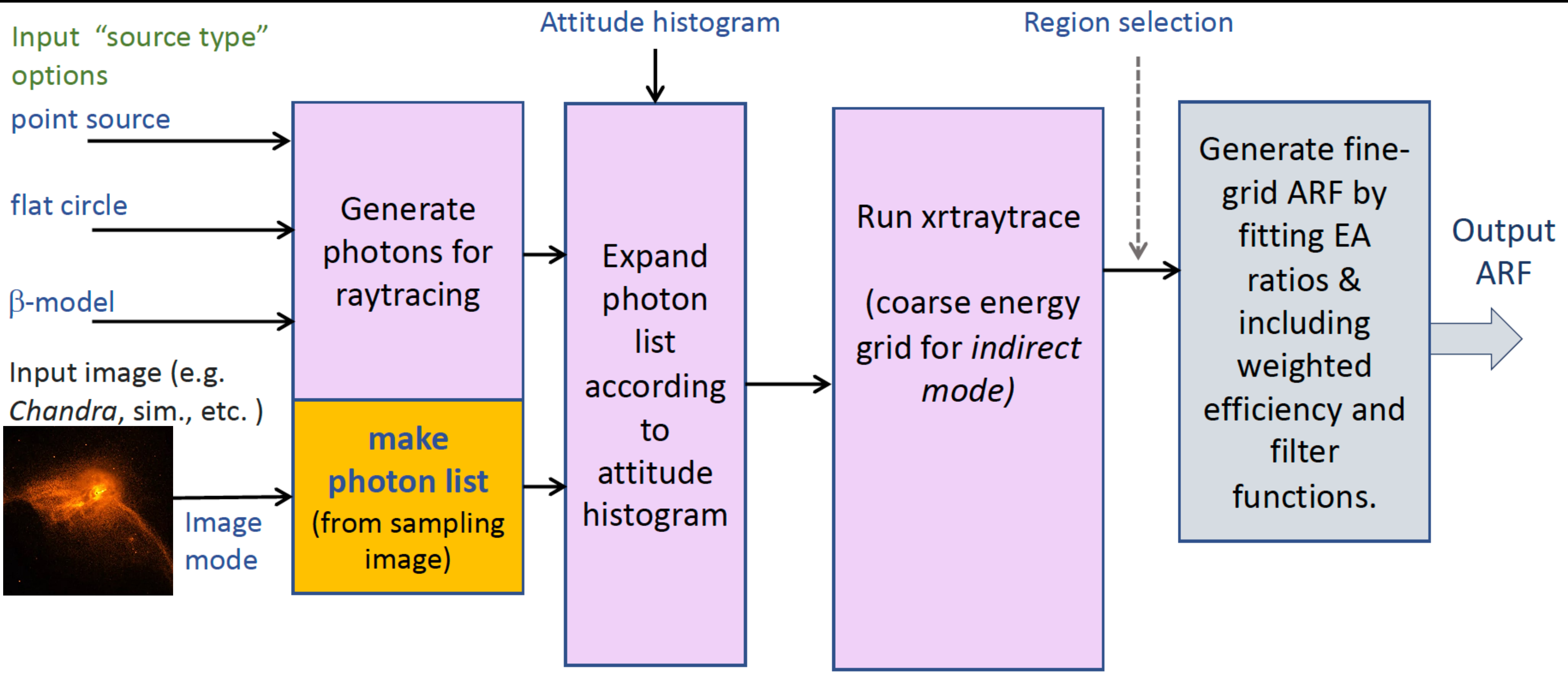
Make RMFs (and ARFs) from real data! 😉

Generating Resolve ARFs



Credits: T. Yaqoob

Generating Resolve ARFs



Credits: T. Yaqoob

Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector's upper right corner.

Step 1: Run **xrtraytrace**

```
$ fhelp xrtraytrace
```

Number of photons injected in the ray-tracing simulation

```
$ xrtraytrace telescop="XRISM" instrume="RESOLVE"
  energy="rsl_onaxiscfile_0p3to18kev.fits[1]" numphoton=600000
  fastmode=yes offaxis=2.317 roll=49.744 source="POINT"
  outphistfile="rsl_pointsource_uppercorner_phist.fits"
  outeafile="rsl_pointsource_uppercorner_ea.fits"
  outpsffile="rsl_pointsource_uppercorner_psf.fits"
  logfile="rsl_pointsource_uppercorner_log.log"
  mirrorfile="CALDB" obstructfile="CALDB" frontreffile="CALDB"
  backreffile="CALDB" pcolreffile="CALDB" scatterfile="CALDB"
  transmode="ALL" scattermode="ALL" psfpars="1 100 0.25"
  resplaneonly=yes phisttype=BRIEF
```

The most important! (See next slide)

Output files (photon events list, effective area, PSF, log)

Energy grid (can also be entered manually)

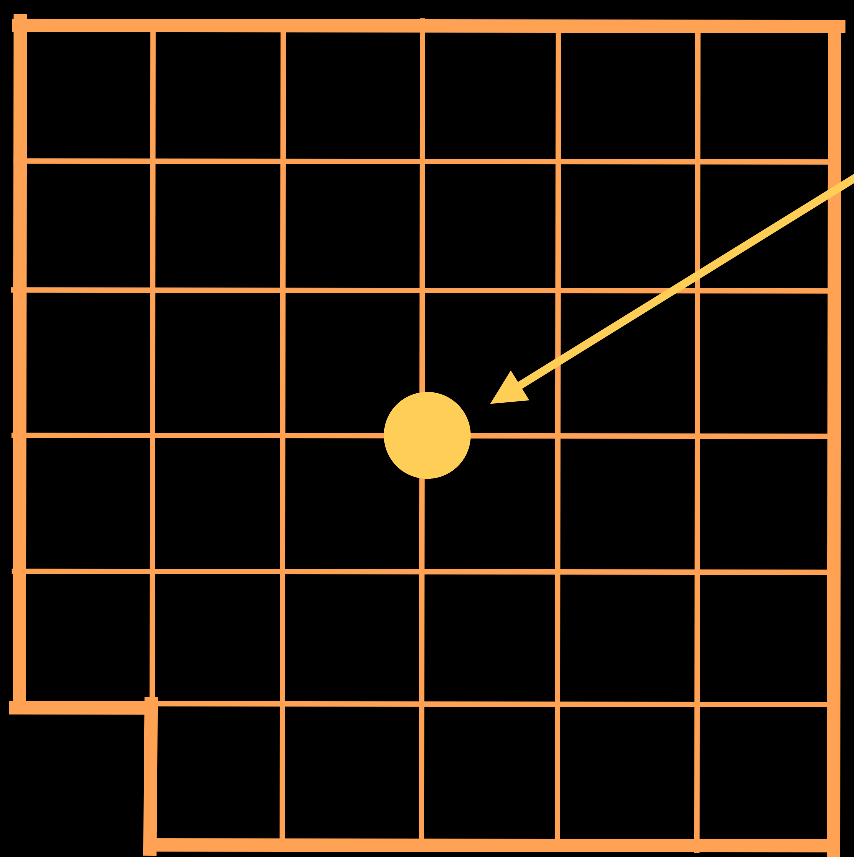
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector's upper right corner.

Step 1: Run **xrtraytrace**

- ✓ `offaxis`: offset from the **<u>TELESCOPE optical axis</u>** (**<u>NOT</u>** the detector!)

- ✓ `roll`: rotation angle from DETX (**<u>NOT</u>** the roll angle!)



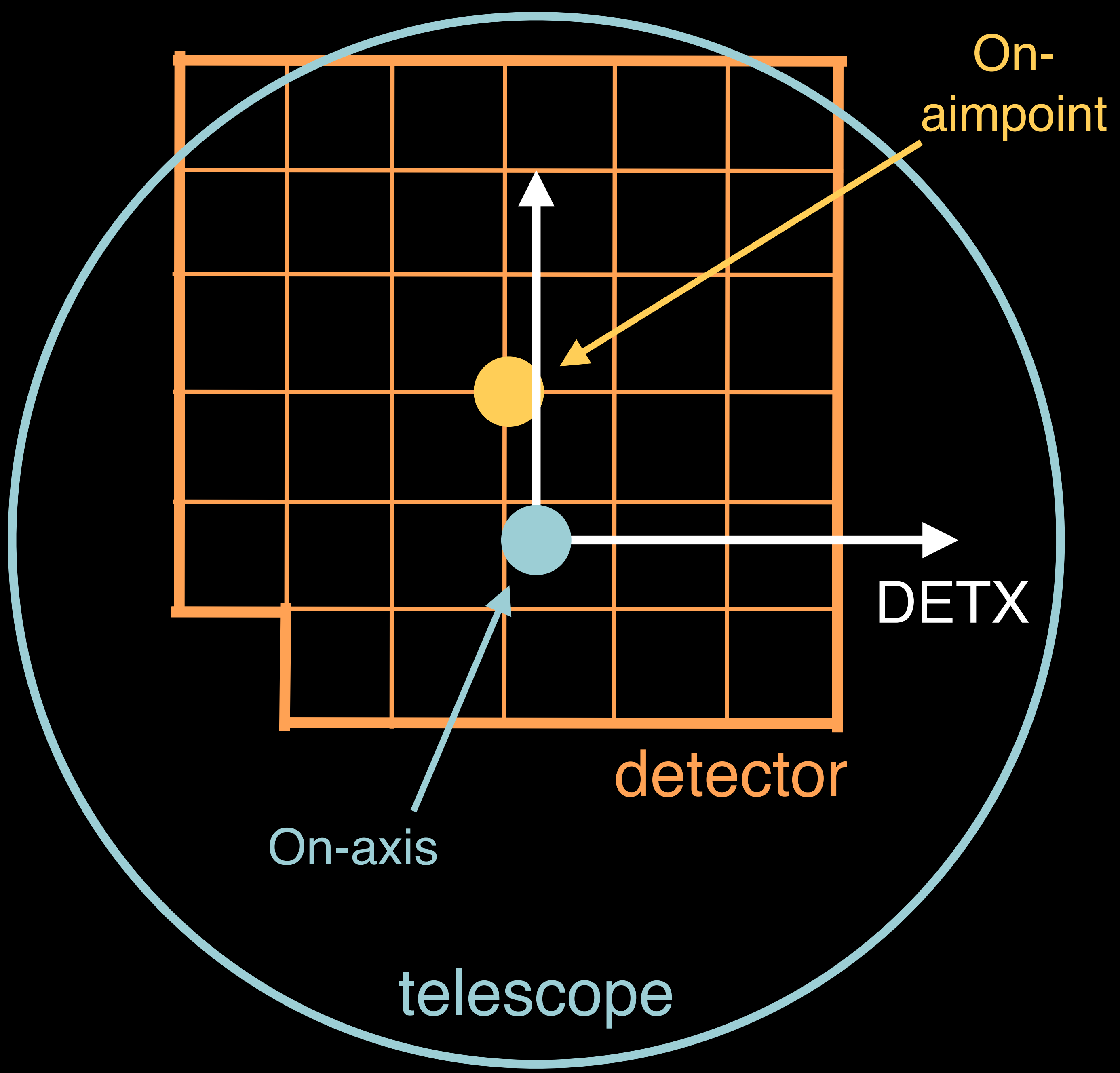
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector's upper right corner.

Step 1: Run **xrtraytrace**

✓ `offaxis`: offset from the **<u>TELESCOPE optical axis</u>** (**<u>NOT</u>** the detector!)

✓ `roll`: rotation angle from DETX (**<u>NOT</u>** the roll angle!)



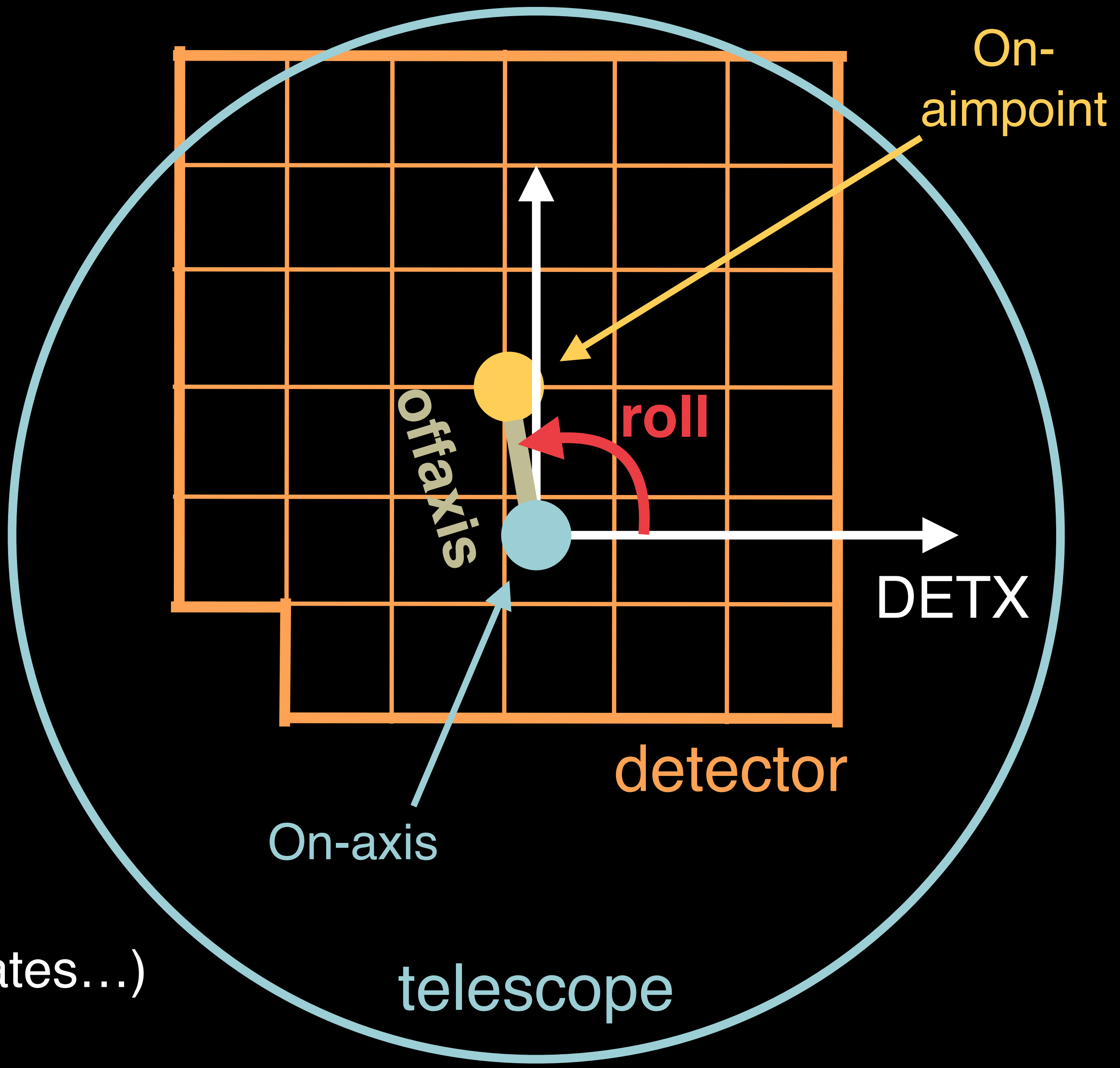
Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector's upper right corner.

Step 1: Run **xrtraytrace**

✓ `offaxis`: offset from the **TELESCOPE optical axis** (**NOT** the detector!)

✓ `roll`: rotation angle from DETX (**NOT** the roll angle!)



At detector aimpoint:
`offaxis=0.245 roll=96.577`
(could change with in-flight calibration updates…)

Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector's upper right corner.

Step 1: Run **xrtraytrace**

```
$ fhelp xrtraytrace
```

Warning: can take a long time! (Depending on the energy grid…)

Step 2: If necessary, change the FILTER and GATEVALV keywords in the (dummy) **exposure map** file

```
$ fthedit "rsl_1att_b7optaxis.expo[1]" FILTER add OPEN
  comment="Filter state”

$ fthedit "rsl_1att_b7optaxis.expo[1]" GATEVALV add CLOSED
  comment=“Gatevalve state”
```

Generating Resolve ARFs

I want an ARF for a point source located exactly on the detector's upper right corner.

Step 3: Run **xaxmaarfgen** (with the simulated event list as input file)

Exposure map used only for partial pixel exp. fractions (and FILTER and GATEVALVE keywords)

```
$ xaxmaarfgen telescop="XRISM" instrume="RESOLVE"
 emapfile="rsl_1att_b7optaxis.expo"
 rmffile="rsl_Hp_5eV.rmf"
 onaxiscfile="rsl_onaxiscfile_0p3to18kev.fits[1]"
 outfile="rsl_pointsource_uppercorner.arf"
 regionfile="rsl_35pix_det.reg"
 xrtevtfile="rsl_pointsource_uppercorner_phist.fits"
 qefile="CALDB" contamifile="CALDB"
 gatevalvefile="CALDB" onaxisffile="CALDB"
```

Detector region (on which the ARF is extracted). In DET coordinates!

Input event list (simulated from xrtraytrace)

Generating Resolve ARFs

I want an ARF for a beta extended source with other parameters than provided.

$ fhelp **xrtraytrace**

Step 1: Run **xrtraytrace**

Center of the source is on-aimpoint

The source is now a beta model!

```
$ xrtraytrace telescop="XRISM" instrume="RESOLVE"
  energy="rsl_onaxiscfile_0p3to18kev.fits[1]" numphoton=600000
  fastmode=yes offaxis=0.245 roll=96.577
  source="BETAMODEL" betapars="1.26 0.53 5.7"
  outphistfile="rsl_extbeta_phist.fits"
  outeafile="rsl_extbeta_ea.fits"
  outpsffile="rsl_extbeta_psf.fits"
  logfile="rsl_extbeta_log.log" mirrorfile="CALDB"
  obstructfile="CALDB" frontreffile="CALDB" backreffile="CALDB"
  pcolreffile="CALDB" scatterfile="CALDB" transmode="ALL"
  scattermode="ALL" psfpars="1 100 0.25" resplaneonly=yes
  phisttype=BRIEF
```

Parameters to change (see next slide)

Generating Resolve ARFs

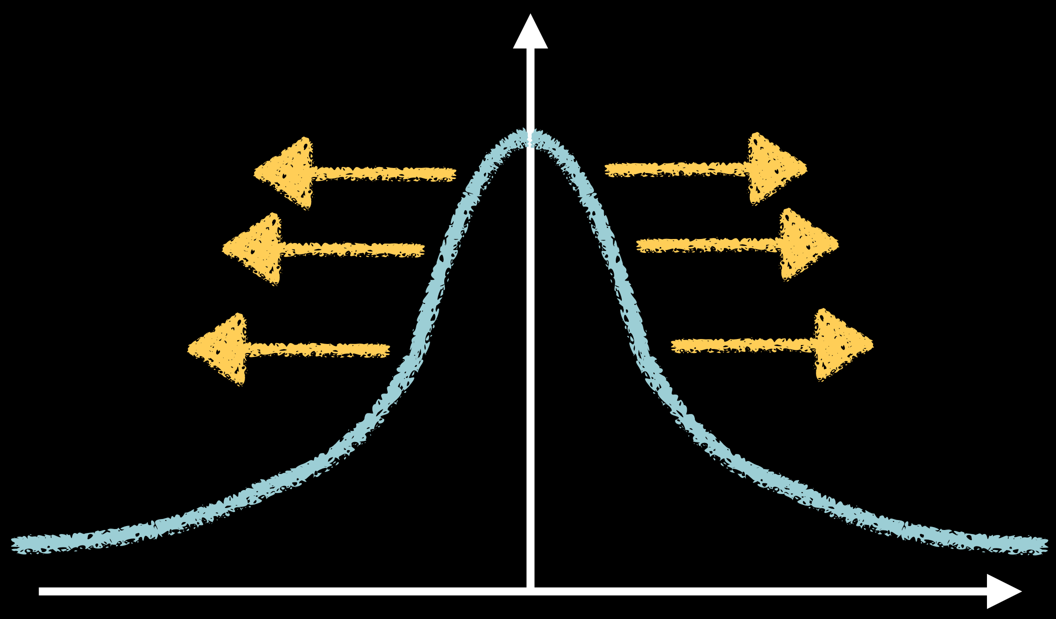
I want an ARF for a beta extended source with other parameters than provided.

Step 1: Run **xrtraytrace**

```
$ fhelp xrtraytrace
```

```
betapars="1.26 0.53 5.7"
```

$$N(r) = C\,[1 + (r/r_c)^2]^{(1.5-3\beta)}$$



Generating Resolve ARFs

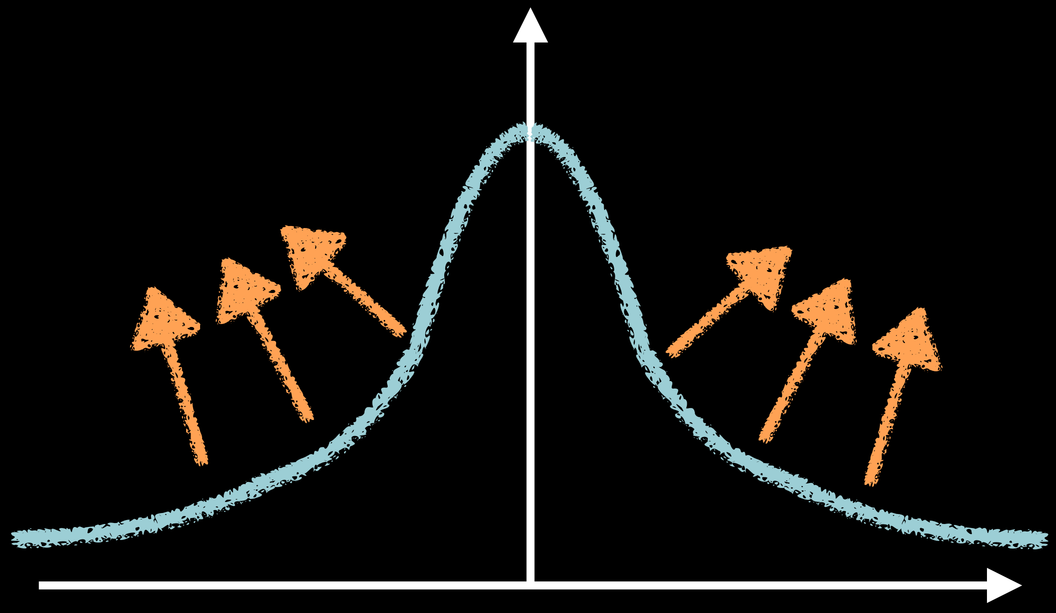
I want an ARF for a beta extended source with other parameters than provided.

```
$ fhelp xrtraytrace
```

Step 1: Run **xrtraytrace**

```
betapars="1.26 0.53 5.7"
```

$$N(r) = C\,[1 + (r/r_c)^2]^{(1.5-3\beta)}$$



Generating Resolve ARFs

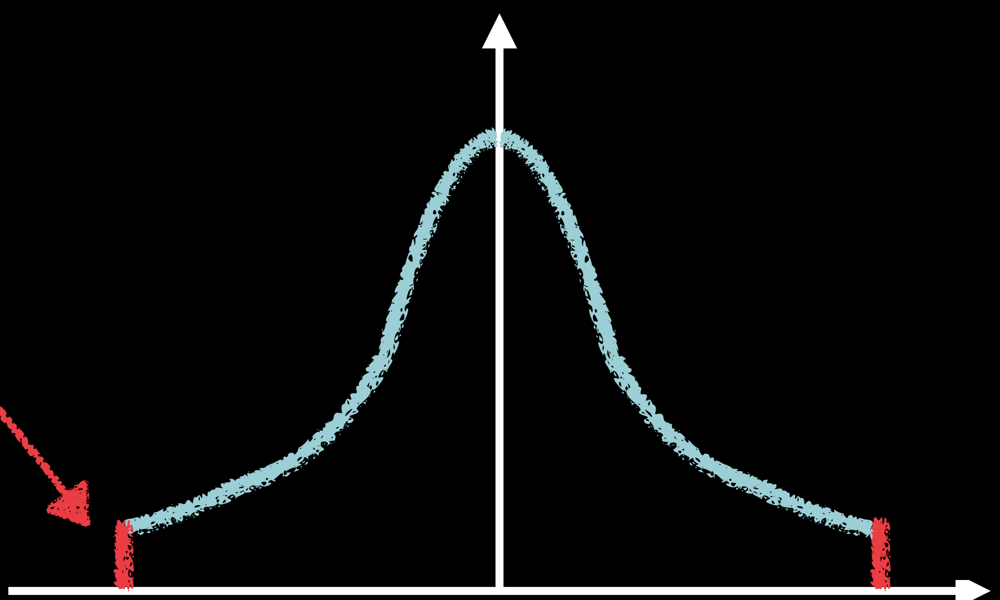
I want an ARF for a beta extended source with other parameters than provided.

Step 1: Run **xrtraytrace**

```
$ fhelp xrtraytrace
```

```
betapars="1.26 0.53 5.7"
```

$$N(r) = C\left[1 + (r/r_c)^2\right]^{(1.5-3\beta)}$$



Step 2 & 3: As before

Generating Xtend responses

What you need:

✓**xtd_1att_nobadpix_b7optaxis.expo:** Dummy exposure map file for making Xtend non-observation ARFs. OPEN filter, gate valve OPEN. (Provided separately)

✓ **xtd_det_r2p50_b7optaxis.reg:** Xtend 2.5' radius circle region file in DET coordinates, centered on the optical axis position, used to make non-observation ARFs. (Provided separately)

✓ **xtd_onaxiscfile_0p3to18kev.fits:** Energy grid file needed for raytracing to make canned ARFs. (Provided separately)

Generating Xtend ARFs

I want an ARF for a circular source with 1 arcmin radius and 2 arcmin off-axis.

```
$ fhelp xrtraytrace
```

Step 1: Run **xrtraytrace**

The source is now a flat circle

Now using Xtend (also in input and output files)

```
$ xrtraytrace telescop="XRISM" instrume="XTEND"
energy="xtd_onaxiscfile_0p3to18kev.fits[1]" numphoton=600000
fastmode=yes offaxis=2.0 roll=0.0 source="FLATCIRCLE"
flatradius=1.0 outphistfile="xtd_extflatoff_phist.fits"
outeafile="xtd_extflatoff_ea.fits"
outpsffile="xtd_extflatoff_psf.fits"
logfile="xtd_extflatoff_log.log" mirrorfile="CALDB"
obstructfile="CALDB" frontreffile="CALDB"
backreffile="CALDB" pcolreffile="CALDB" scatterfile="CALDB"
transmode="ALL" scattermode="ALL" psfpars="1 100 0.25"
resplaneonly=yes phisttype=BRIEF
```

Radius of the flat circular source is set to 1 arcmin

Generating Xtend ARFs

I want an ARF for a circular source with 1 arcmin radius and 2 arcmin off-axis.

Step 1: Run **xrtraytrace**

```
$ fhelp xrtraytrace
```

✓ In the case of Xtend, the on-aimpoint (i.e. center of the detector) almost coincides with the on-axis (i.e. center of the telescope)!

At detector aimpoint:

`offaxis=0.0` `roll=0` is a good approximation

Step 2 & 3: As before

Caveats (ARFs) and Takeaways

- ✓ Generating accurate ray-tracing events at many energies can take a long time!
- ✓ Try to find the best compromise between science case accuracy vs. computing cost
- ✓ **Remember**: this is an advanced tutorial! It is **VERY** likely that your science justification can reasonably be done with the responses already available online (provided by the GOF)