Event Lists Manipulation and Screening

13th ESAC SAS Workshop

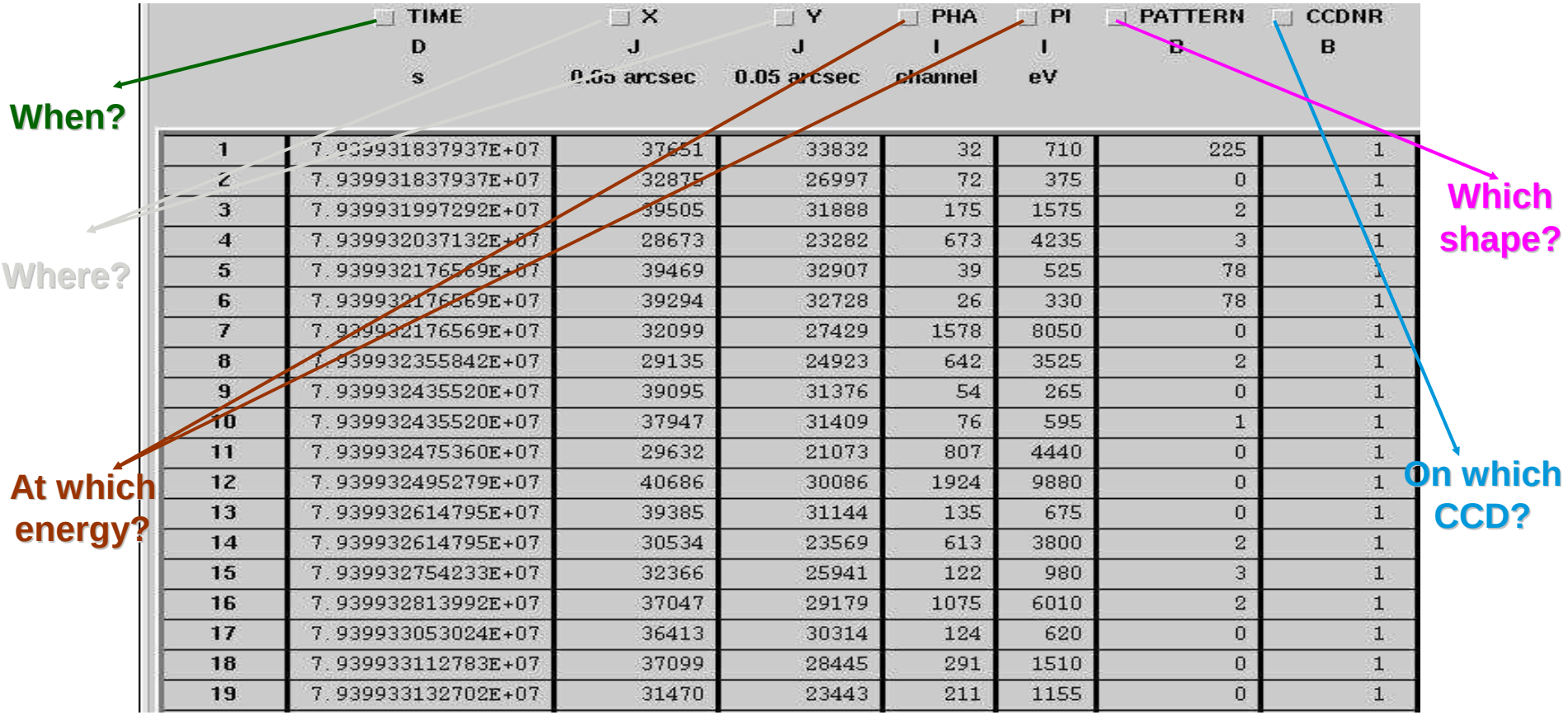
June 10 – June 14, 2013

Ignacio de la Calle

XMM-Newton Science Operations Centre

EPIC event lists

`e[m/p]proc` and `e[m/p]chain` produce calibrated & concatenated *event lists*.
Each event is individually time-tagged, and its spatial, energy … properties are registered.



	TIME D s	X J 0.05 arcsec	Y J 0.05 arcsec	PHA I channel	PI I eV	PATTERN B	CCDNR B
1	7.939931837937E+07	37651	33832	32	710	225	1
2	7.939931837937E+07	32875	26997	72	375	0	1
3	7.939931997292E+07	39505	31888	175	1575	2	1
4	7.939932037132E+07	28673	23282	673	4235	3	1
5	7.939932176569E+07	39469	32907	39	525	78	1
6	7.939932176569E+07	39294	32728	26	330	78	1
7	7.939932176569E+07	32099	27429	1578	8050	0	1
8	7.939932355842E+07	29135	24923	642	3525	2	1
9	7.939932435520E+07	39095	31376	54	265	0	1
10	7.939932435520E+07	37947	31409	76	595	1	1
11	7.939932475360E+07	29632	21073	807	4440	0	1
12	7.939932495279E+07	40686	30086	1924	9880	0	1
13	7.939932614795E+07	39385	31144	135	675	0	1
14	7.939932614795E+07	30534	23569	613	3800	2	1
15	7.939932754233E+07	32366	25941	122	980	3	1
16	7.939932813992E+07	37047	29179	1075	6010	2	1
17	7.939933053024E+07	36413	30314	124	620	0	1
18	7.939933112783E+07	37099	28445	291	1510	0	1
19	7.939933132702E+07	31470	23443	211	1155	0	1

Browsing an event list: SAS and FTOOLS

Event lists (as most of the XMM-Newton data) are FITS files, which can be manipulated with FTOOLS (HEASOFT), alongside with specific SAS tasks:

FTOOLS

- **dump FITS files to ASCII:**

```
fdump infile=file.fits outfile=file.asc columns=- rows=-
```

- **visualise header keywords *(attributes):***

```
fkeyprint infile=file.fits keynam=KEYWORD outfile=STDOUT
```

- **show the structure of a FITS file:**

```
fstruct infile=file.fits
```

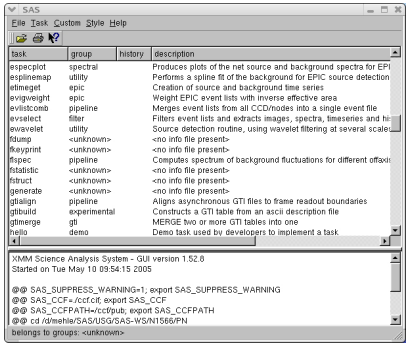
- **calculate statistics on a column of a FITS file:**

```
fstatistic infile=file.fits colname=COLUMN rows=-
```

SAS

- **SAS provides a GUI interface to run these and other LHEASOFT tasks. From the command line run:**

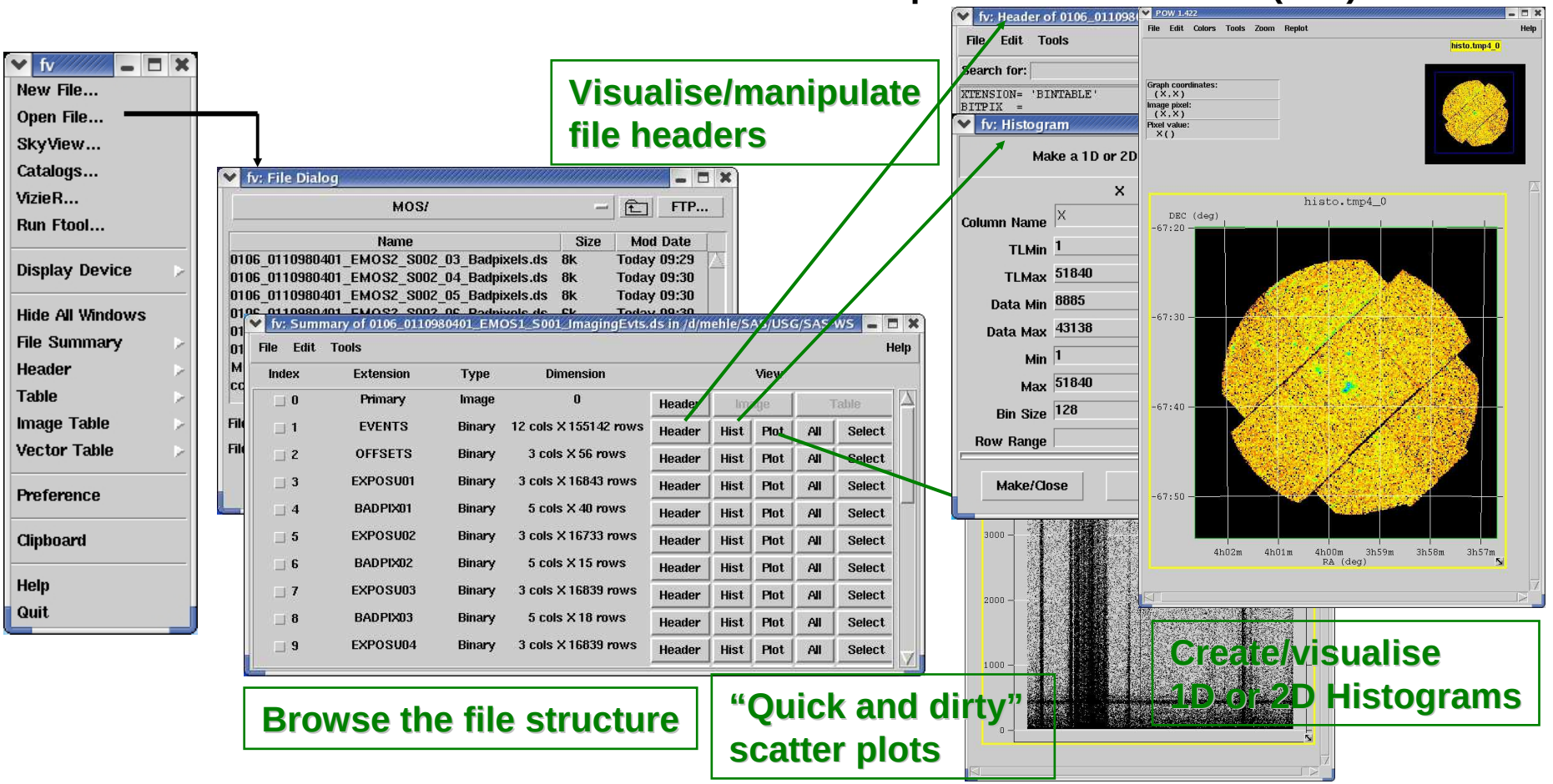
> sas



xmm-newton
European Space Agency

Browsing an event list: "fv"

Event files can also be browsed with an ftools Graphical User Interface (GUI): fv



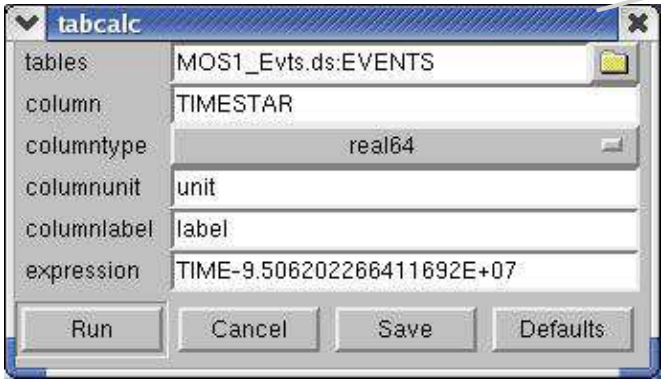
Manipulating event list columns

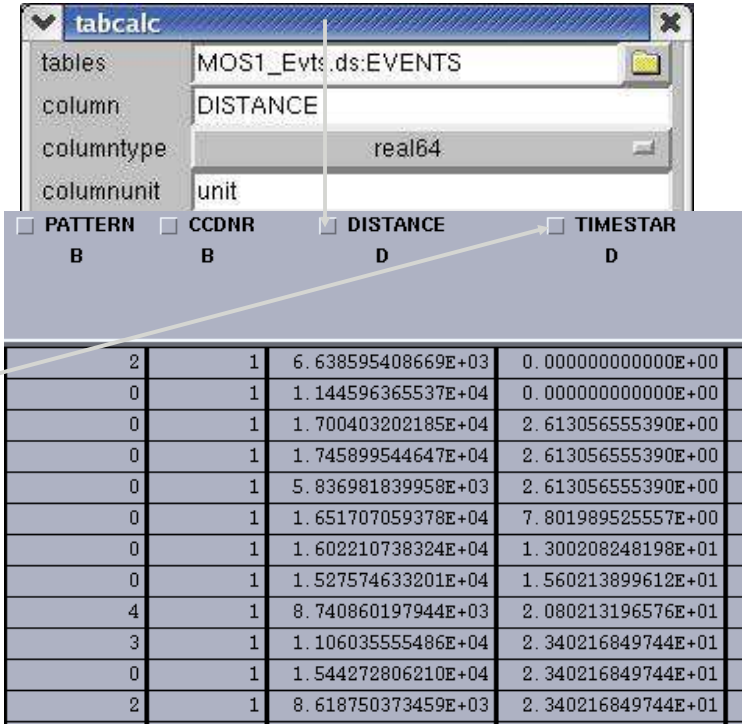
Event list columns can be algebraically manipulated to produce new, or to modify existing columns with the SAS task `tabcalc`. Examples:

1. Run from the command line: tabcalc -d

2. Generation of a column containing the DISTANCE from a given pixel [in the example: (18000, 18000) in sky coordinates]

3. Generation of a new TIME column, where times are expressed as seconds from the observation start:





PATTERN	CCDNR	DISTANCE	TIMESTAR
B	B	D	D
2	1	6.638595408669E+03	0.000000000000E+00
0	1	1.144596365537E+04	0.000000000000E+00
0	1	1.700403202185E+04	2.613056555390E+00
0	1	1.745899544647E+04	2.613056555390E+00
0	1	5.836981839958E+03	2.613056555390E+00
0	1	1.651707059378E+04	7.801989525557E+00
0	1	1.602210738324E+04	1.300208248198E+01
0	1	1.527574633201E+04	1.560213899612E+01
4	1	8.740860197944E+03	2.080213196576E+01
3	1	1.106035555486E+04	2.340216849744E+01
0	1	1.544272806210E+04	2.340216849744E+01
2	1	8.618750373459E+03	2.340216849744E+01

Filtering Event Files

Event list can be filtered to contain only a given subsample of the total events.

They can be filtered according to:

- **Time**
- **Event Characteristics**
- **Spatial region**
- **Any parameter in the Event File**

	TIME D s	X J 0.05 arcsec	Y J 0.05 arcsec	PHA I channel	PI I eV	PATTERN B	CCDNR B
1	7.939931837937E+07	37651	33832	32	710	225	1
2	7.939931837937E+07	32875	26997	72	375	0	1
3	7.939931997292E+07	39505	31888	175	1575	2	1
4	7.939932037132E+07	28673	23282	673	4235	3	1
5	7.939932176569E+07	39469	32907	39	525	78	1
6	7.939932176569E+07	39294	32728	26	330	78	1
7	7.939932176569E+07	32099	27429	1578	8050	0	1
8	7.939932355842E+07	29135	24923	642	3525	2	1
9	7.939932435520E+07	39095	31376	54	265	0	1
10	7.939932435520E+07	37947	31409	76	595	1	1
11	7.939932475360E+07	29632	21073	807	4440	0	1
12	7.939932495279E+07	40686	30086	1924	9880	0	1
13	7.939932614795E+07	39385	31144	135	675	0	1
14	7.939932614795E+07	30534	23569	613	3800	2	1
15	7.939932754233E+07	32366	25941	122	980	3	1
16	7.939932813992E+07	37047	29179	1075	6010	2	1
17	7.939933053024E+07	36413	30314	124	620	0	1
18	7.939933112783E+07	37099	28445	291	1510	0	1
19	7.939933132702E+07	31470	23443	211	1155	0	1

Filtering is done through *selection expressions*, using the SAS task:

`evselect`

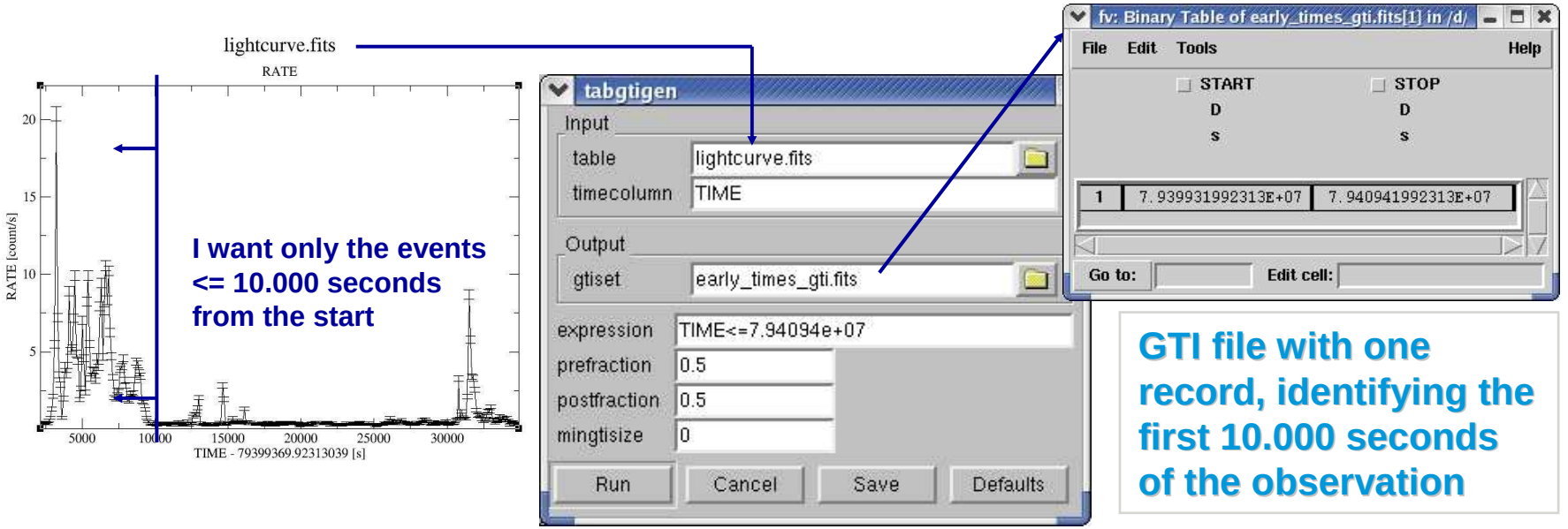
Example: select all events within energies greater than 2 keV:

```
evselect table=MyFITS_file.fits expression=PI>2000
```

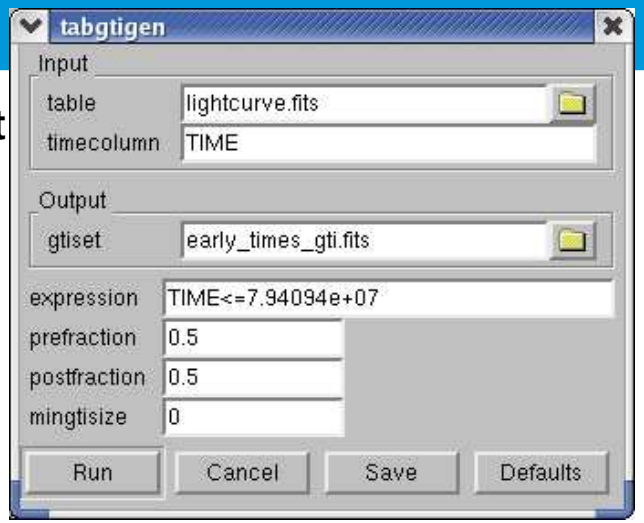
The concept of Good Time Intervals (GTI)

Good Time Intervals: a set of time intervals where a given scientific product (*e.g.:* an event list) is accumulated.
EPIC event lists: one GTI extension for each chip. Important in calculation of exposure times, or to remove high particle background phases.

GTIs can be generated with SAS task `tabgtigen`.
GTI files can be subsequently applied to generate customised scientific products:



selectlib: a selection/manipulation library



All operations to manipulate tables in an XMM-Newton event list are driven by the `selectlib` library.

http://xmm.esac.esa.int/sas/current/selectlib/

Examples of allowed operations:

- **boolean: “==“, “>”, “<=“, “||”, “&&”, “!” … E.g.: `(CCDNR==1)&&(PHA>=300)`**
- **arithmetic/trigonometric: “+”, “abs(x)”, “sin(x)”, “log(x)” … E.g.: `(log(PI)>0)`**
- **string manipulation: “upper/lower”, “=“, “>”, “+”, “ascii” … E.g.: `’W’ + ‘ XMM’ ⇒ ‘W XMM’`**
- **definition of a selection expression as a keyword. E.g.: `#DISTANCE < 128` if a keyword `DISTANCE == SQRT((X-18000)**2+(Y-18000)**2)` exists in a to-be-screened file**
- **bitwise (BW) operators: “BW AND/OR”, “left/right shift”**
- **built-in constants: “#PI”, “#RAD”, “#E”, “TRUE/FALSE” … E.g.: `PATTERN>#PI`**

Functions for spatial filters

In order to facilitate extraction of scientific products in spatial regions, a number of pre-defined selection regions are available in `selectlib`:

http://xmm.esac.esa.int/sas/current/doc/selectlib/index.html

- `point(x0,y0,Xcolumn,Ycolumn)`
- `line(x0,y0,x1,y1,Xcolumn,Ycolumn)`
- `circle(xCenter,yCenter,radius,Xcolumn,Ycolumn)`
- `sector(xCenter,yCenter,fromAngle,toAngle,Xcolumn,Ycolumn)` or `pie(xCenter,yCenter,fromAngle,toAngle,Xcolumn,Ycolumn)`
- `ring(xCenter,yCenter,radius1,radius2,Xcolumn,Ycolumn)` or `annulus(xCenter,yCenter,radius1,radius2,Xcolumn,Ycolumn)`
- `ellipse(xCenter,yCenter,xHalfWidth,yHalfWidth,rotation,Xcolumn,Ycolumn)`
- `elliptannulus(xCenter,yCenter,xHalfWidthInner,yHalfWidthInner xHalfWidthOuter,yHalfWidthOuter,rotationInner,rotationOuter,Xcolumn,Ycolumn)` or `elliptring(xCenter,yCenter,xHalfWidthInner,yHalfWidthInner xHalfWidthOuter,yHalfWidthOuter,rotationInner,rotationOuter,Xcolumn,Ycolumn)`
- `box(xCenter,yCenter,xHalfWidth,yHalfWidth,rotation,Xcolumn,Ycolumn)`
- `rectangle(xLoLeft,yLoLeft,xUpRight,yUpRight,rotation,Xcolumn,Ycolumn)`
- `rhombus(xCenter,yCenter,xHalfWidth,yHalfWidth,rotation,Xcolumn,Ycolumn)` or `diamond(xCenter,yCenter,xHalfWidth,yHalfWidth,rotation,Xcolumn,Ycolumn)`
- `polygon(x1,y1,x2,y2,x3,y3,x4,y4,....,Xcolumn,Ycolumn)`

Example: to select all events within 128 pixels from the sky pixel (18000, 18000):

```
circle (18000, 18000, 128, X, Y)
```

File-based filters

Three file-based filters exist within `selectlib`:

- **GTI-filter: `gti(gti.fits, TIME):`** selects all events, whose TIME belongs to at least on of the GTIs defined in `gti.fits` (assuming that `TIME` is the event list time column)

- **Mask filter: `mask(mask.fits, X0, Y0, X, Y):`** selects all events which fall on a position `[(X0-X),(Y0-Y)]`, whose corresponding mask value is non-zero. It can be applied to sky coordinates positions, if the mask contains WCS information

- **Region filter: `region(region.fits, X, Y):`** selects all events whose position (in sky pixels in this case) belongs to `region.fits`

IN-operator

A generic operator family exists, allowing expressions of form `arith in (...)`

- **IN-intervals:**

interval specification	alternative expression	meaning
: or (:] or [:) or (:)	true	$-\infty < x < +\infty$
val or [*val]*	*val* == *x*	*x=val*
val: or [*val:]* or [*val:)*	*val* <= *x*	$val <= x < +\infty$
(*val:]* or (*val:)*	*val* < *x*	$val < x < +\infty$
: *val* or [: *val]* or (: *val]*	*val* >= *x*	$-\infty < x <= val$
[: *val)* or (: *val)*	*val* > *x*	$-\infty < x < val$
lo: hi or [*lo: hi]*	*lo* <= *x* && *hi* >= *x*	*lo<=x<=hi*
(*lo: hi]*	*lo* < *x* && *hi* >= *x*	*lo<x<=hi*
[*lo: hi)*	*lo* <= *x* && *hi* > *x*	*lo<=x<hi*
(*lo: hi)*	*lo* < *x* && *hi* > *x*	*lo<x<hi*

Example:

```
PI in [100, 300)
```

is the same as:

```
(PI=>100)&&(PI<300)
```

- **IN-GTI:** `TIME IN gti(gti.fits)` **is the same as** `gti(gti.fits, TIME)`

- **IN-filter:** `(X, Y) in circle(18000, 18000, 128)` **is the same as**

```
circle (18000, 18000, 128, X, Y)
```

If you are scared enough, you may ask: do I really need to learn all this stuff to extract my customised scientific products?
The answer is no … as it will be shown in the next presentation.