Basic properties of the INTEGRAL data analysis system OSA 5.1

Igor Telezhinsky,

Astronomical Observatory of Kiev University

What can you use OSA for?

- Corrections, if needed
- Dead-time calculation, good-time interval selections
- Image reconstruction (including mosaics)
- Source identification and extraction
- Count extraction (spectra, light curves)
- Handling of Instrument Characteristics (IC)

What can't you use OSA for?

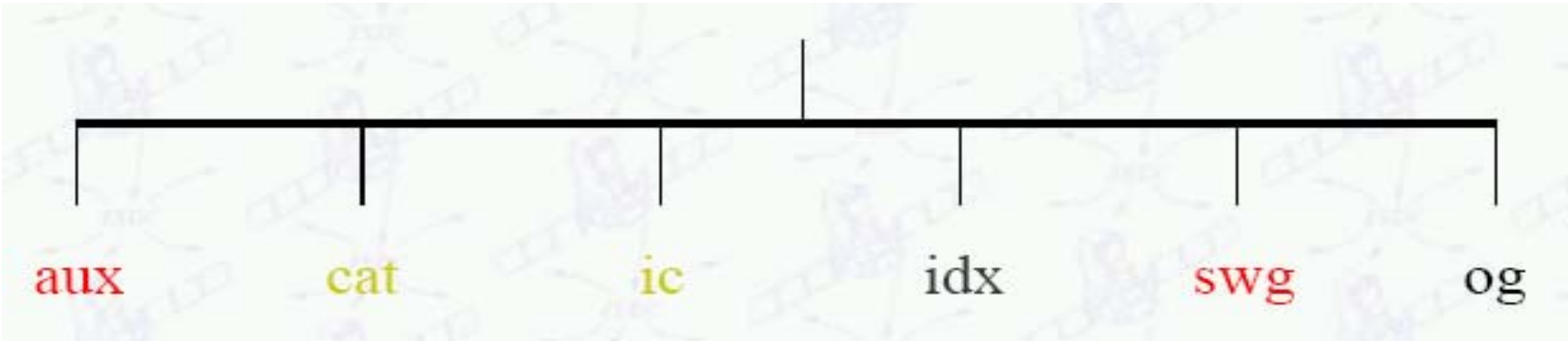
- Image analysis
- Spectral fitting
- Timing analysis (period search,...)
- etc.

OSA Components

- ***Generic tools***
  - Developed and maintained by ISDC
  - FTOOL-like

    ```
    dal_list dol=my_file.fits[1]
    ```
  - Written in C
- ***Instrument-specific executables***
  - Developed and maintained by Instrument Teams
  - FTOOL-like
  - Some components written in F90
- ***Analysis scripts***
  - Developed and maintained by ISDC/Instrument Teams
  - C++ scripts based on ROOT and isdcroot
  - Work in script, command-line or GUI modes

Data repository

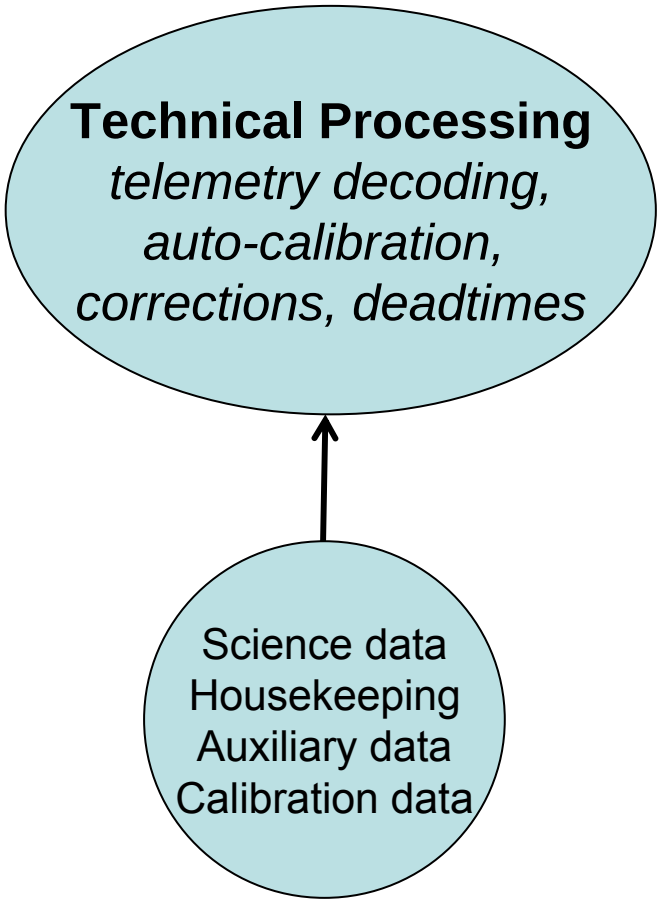


- **aux**: Auxiliary data; attitude, orbit, program, ...
- (downloaded from the Archive)
- **cat**: Source catalogue (downloaded with OSA)
- **ic**: Instrument characteristics; calibration, instrument models, ... (downloaded with OSA)
- **idx**: Index tables; sort of databases...
- **swg**: Science Window data; instrument data
- (downloaded from the Archive)
- **og**: Observation data; products generated by OSA

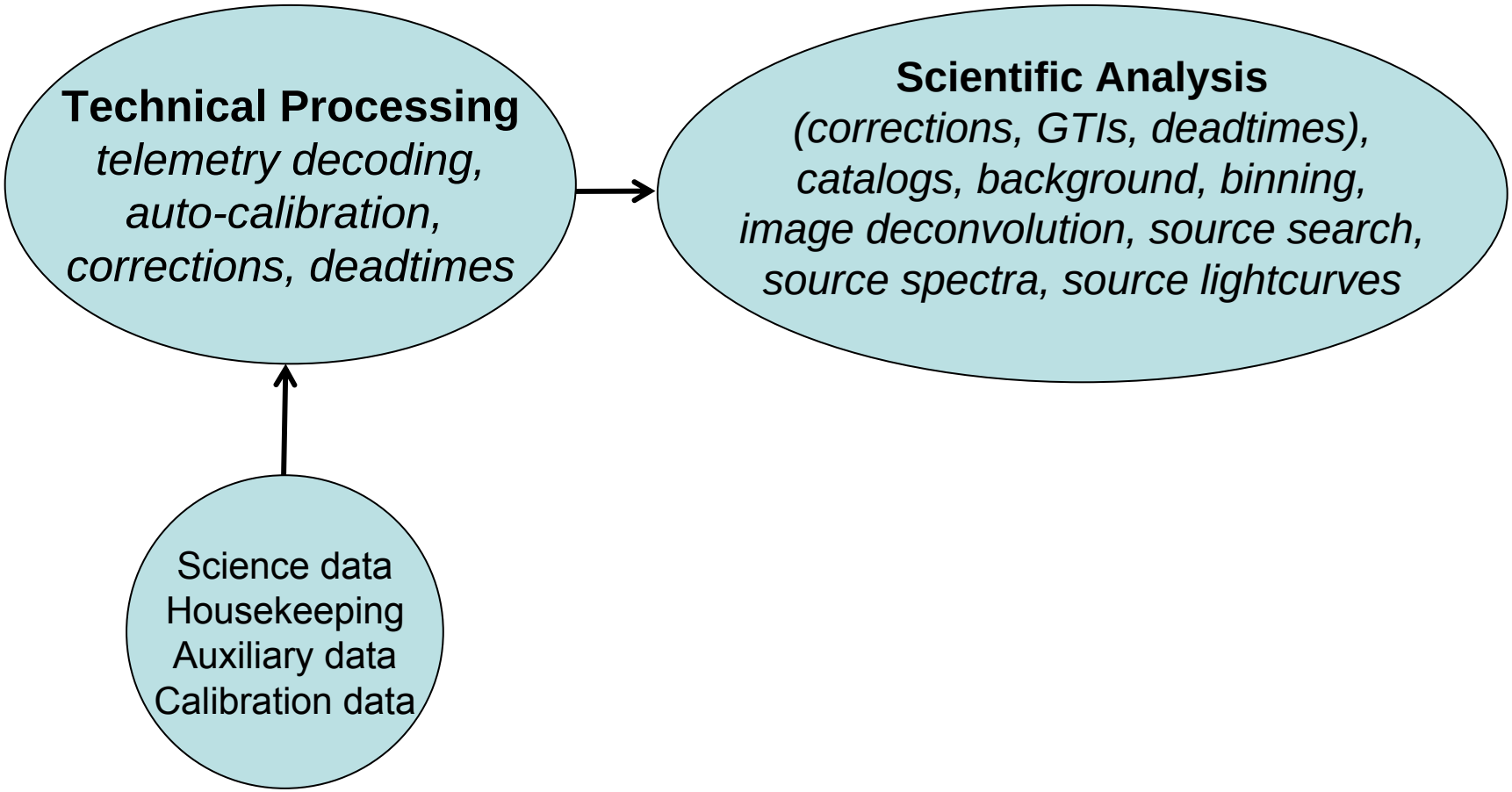
Data way of life till final results

Technical Processing
telemetry decoding, auto-calibration, corrections, deadtimes

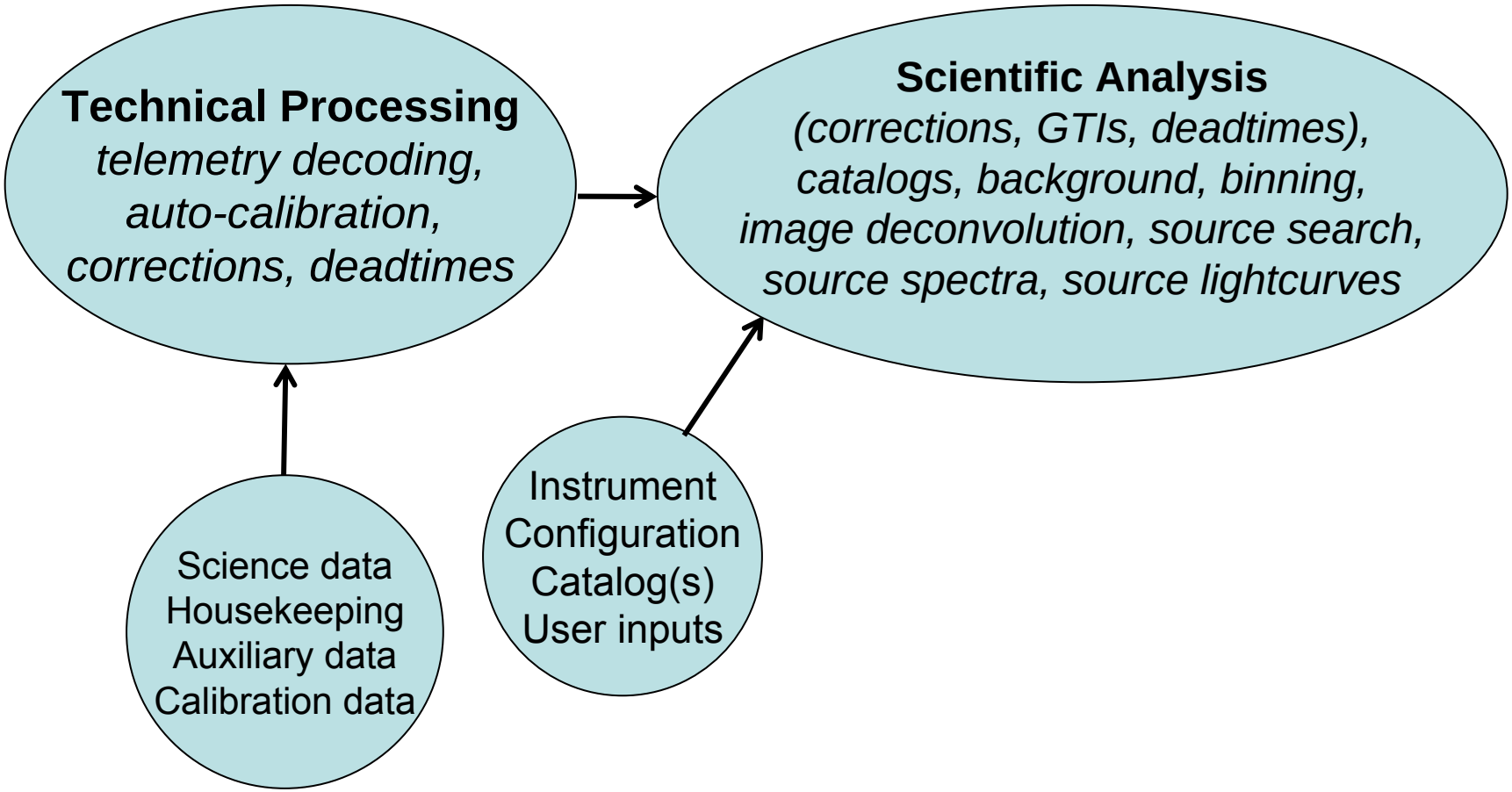
Data way of life till final results



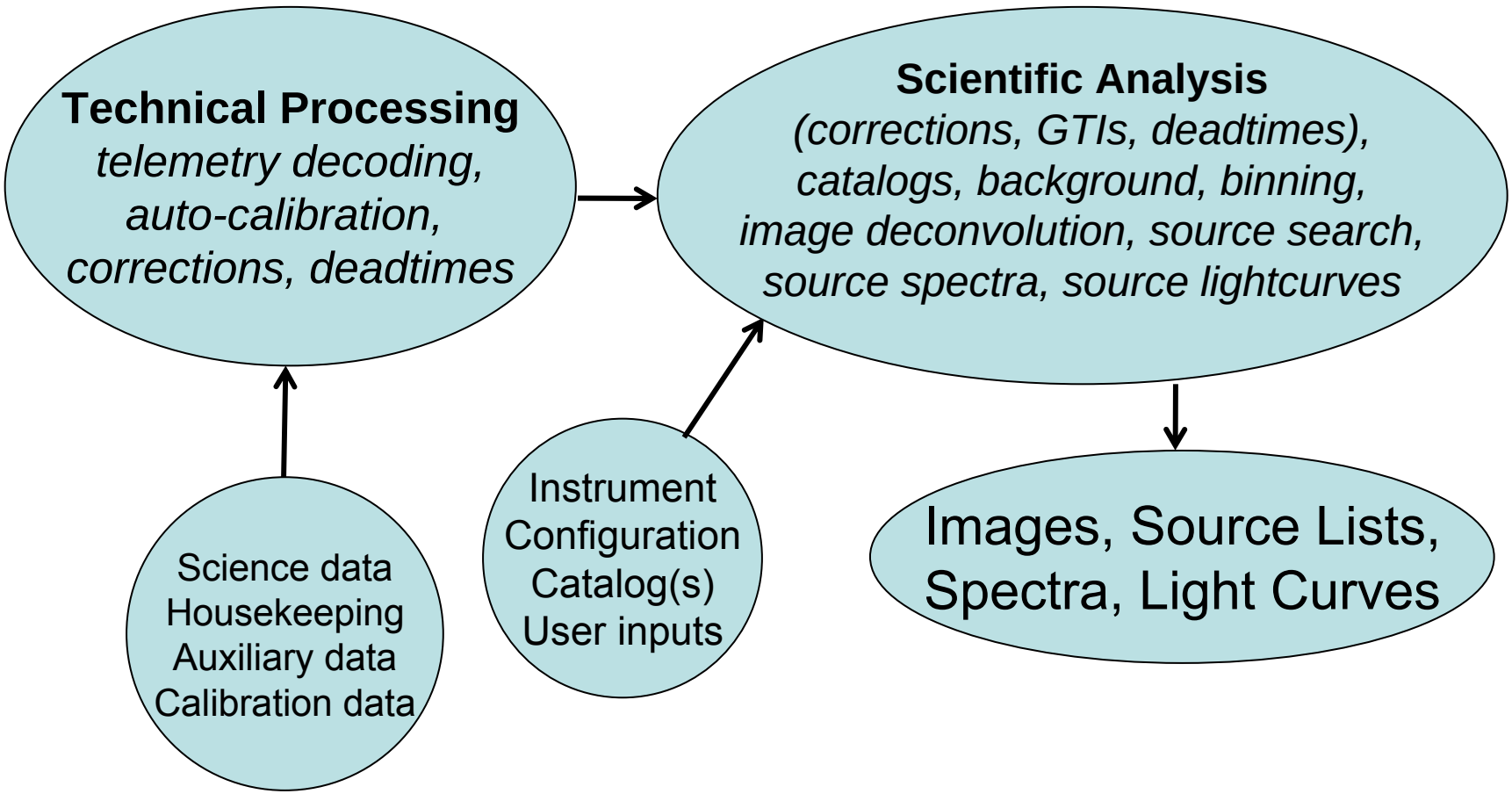
Data way of life till final results



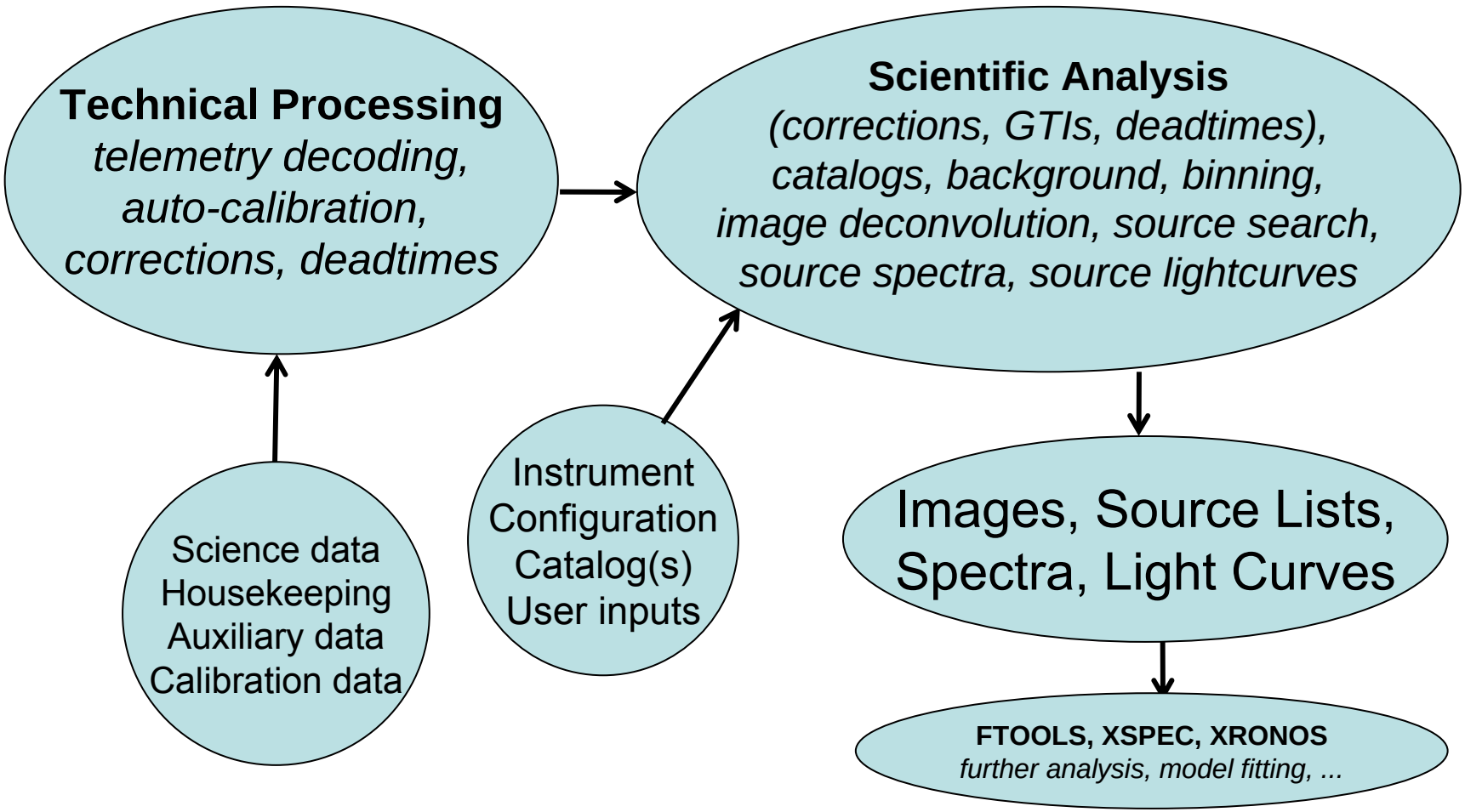
Data way of life till final results



Data way of life till final results



Data way of life till final results



The way OSA works

- One concept for all instruments.
- Data analysis pipeline is a chain of executables each having a parameter file (so called Parameter Interface Layer implementation).
- Parameters can be queried or hidden
- Parameters can be set on the command line

```
dal_list dol=my_file.fits[1]
```

- Or through the scripts and GUIs
- For the users, OSA allows easy control of the output (screen vs log file), whether in interactive or in batch mode (so called Reporting Interface Layer implementation).

The way OSA works

- OSA uses FITS files for any kind of data (images, tables, spectra etc.)
- FITS allows special data structures that do not contain any data, but only points to other data structure.
- The basic building block of data for the analysis system is a GROUP.
- ***Science Window groups*** contain all the data generated by INTEGRAL during a Science Window (pointing, slew,...).
- ***Observation groups*** group Science Windows on the basis of user defined criteria and all the results from OSA run.

Scripts

- The analysis of the data from a single INTEGRAL instrument requires a few tens of executables
- Scripts are necessary to help users to run an entire analysis in the correct order
  - Steps are called “Level”: COR, DEAD, GTI, IMA, SPE, LCR, ...
  - There is complete flexibility in the Level sequence, except order
- ISDC scripts are actually C++ code with isdcroot/ROOT classes
- C++ is not mandatory... Advanced users can develop their own scripts in shell, Perl, Python, tcl, ..., but, for the moment, the need to provide scripts in more user-friendly languages that users can reuse, modify is not that obvious...

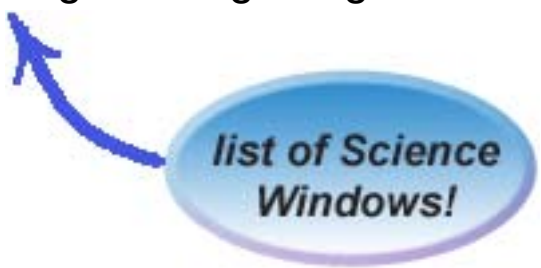
GUIs

- GUIs exist for all instruments.
- Simplify parameter filling for interactive sessions, with little loss of capabilities.
- Hides “hidden” parameters not to scare anyone ☺
- With OSA 5.1, GUIs really become parameter editors.
  - Save command lines.
  - Load parameters from scripts.
  - Can also be used as regular GUI's!

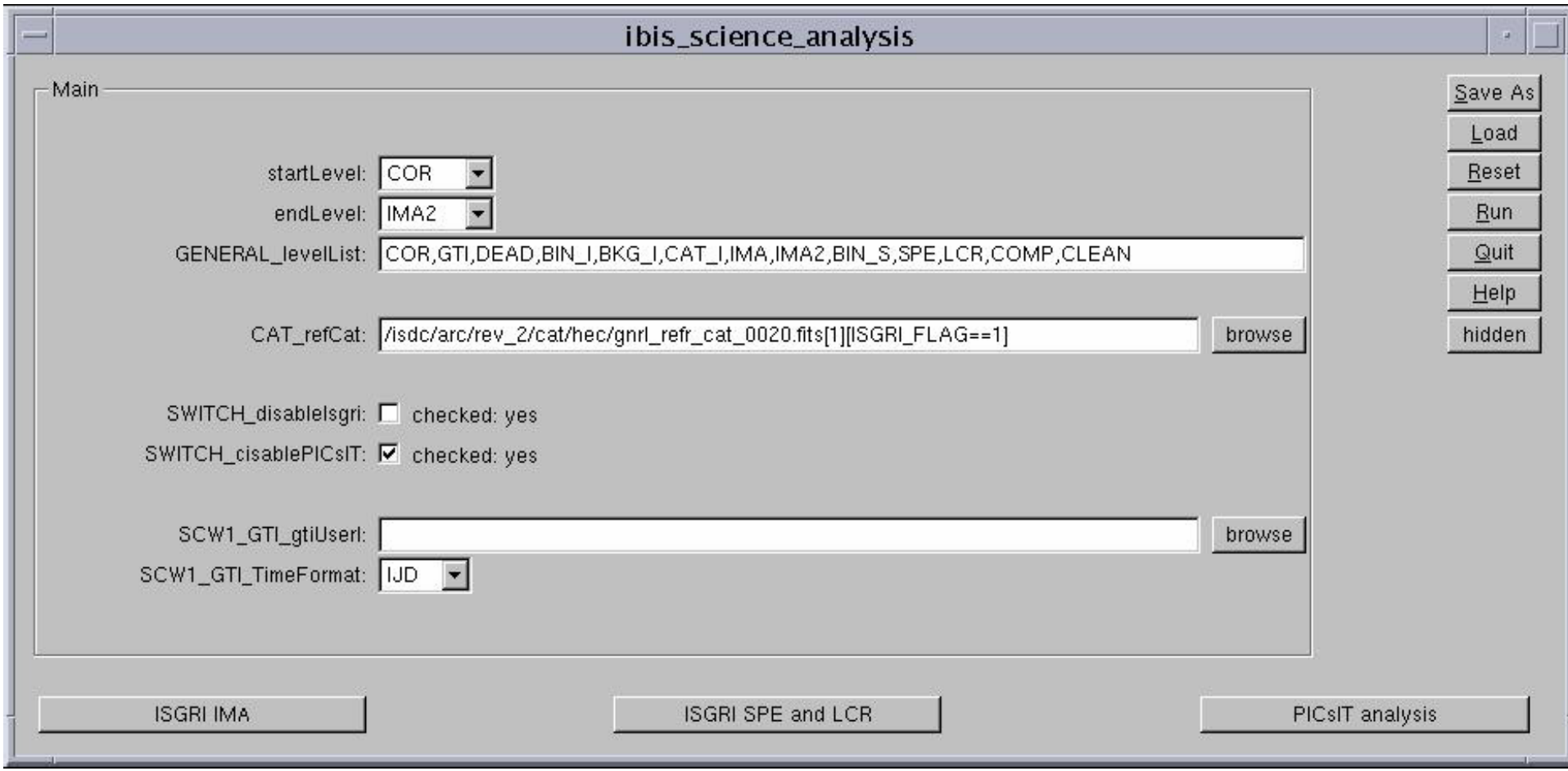
GUIs standard analysis usage

Just before the analysis…

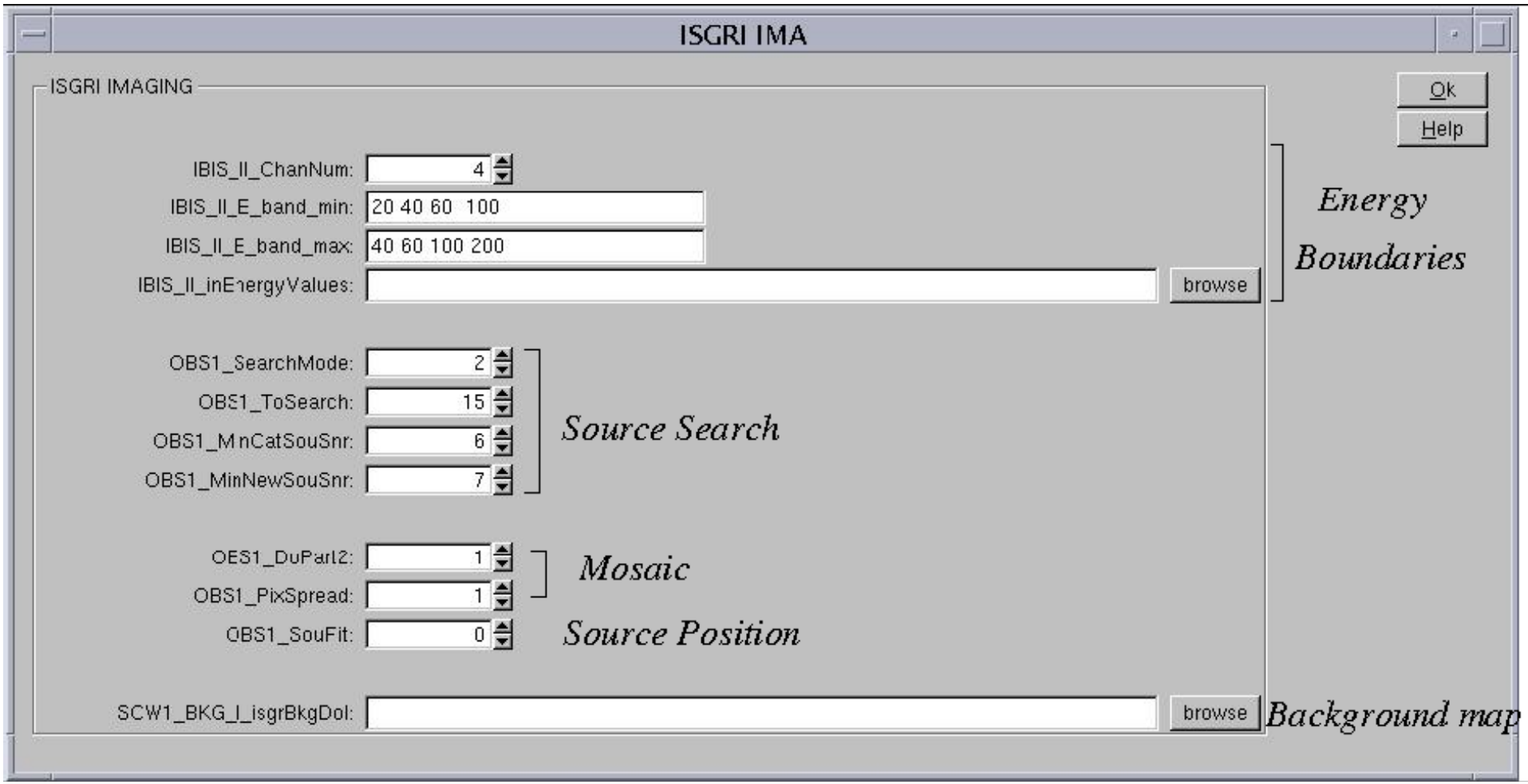
- Set the Environment…
  - `setenv ISDC_ENV directory_of_OSA_sw_installation`
  - `setenv ISDC_REF_CAT "$REP_BASE_PROD/cat/hec/gnrl_refr_cat_0023.fits[1]"`
  - `source $ISDC_ENV/bin/isdc_init_env.csh`
- … and create observation group
  - `cd $REP BASE PROD`
  - `og_create idxSwg=isgri_gc.lst ogid=isgri_gc baseDir="./" instrument=IBIS`



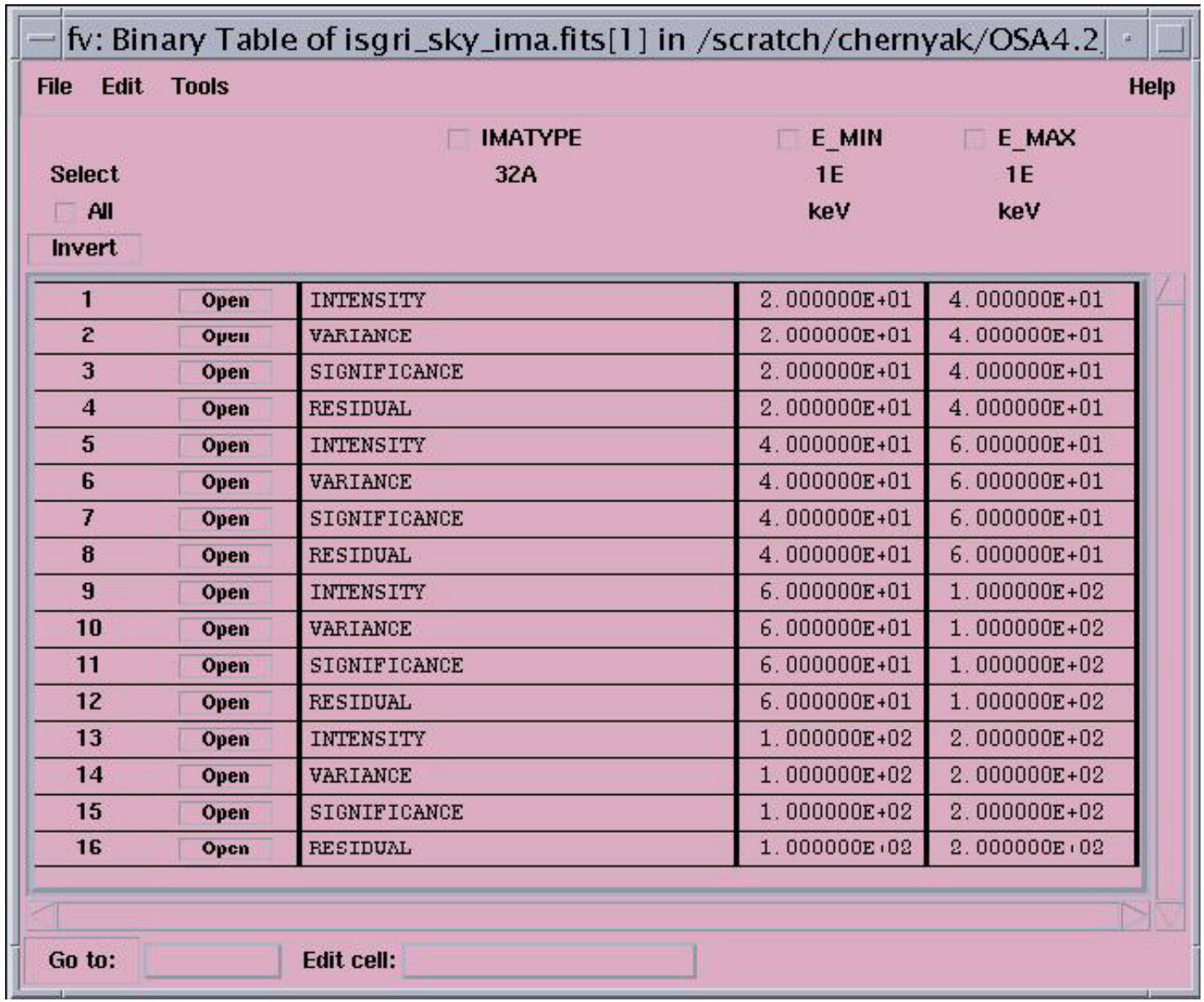
ibis_science_analysis



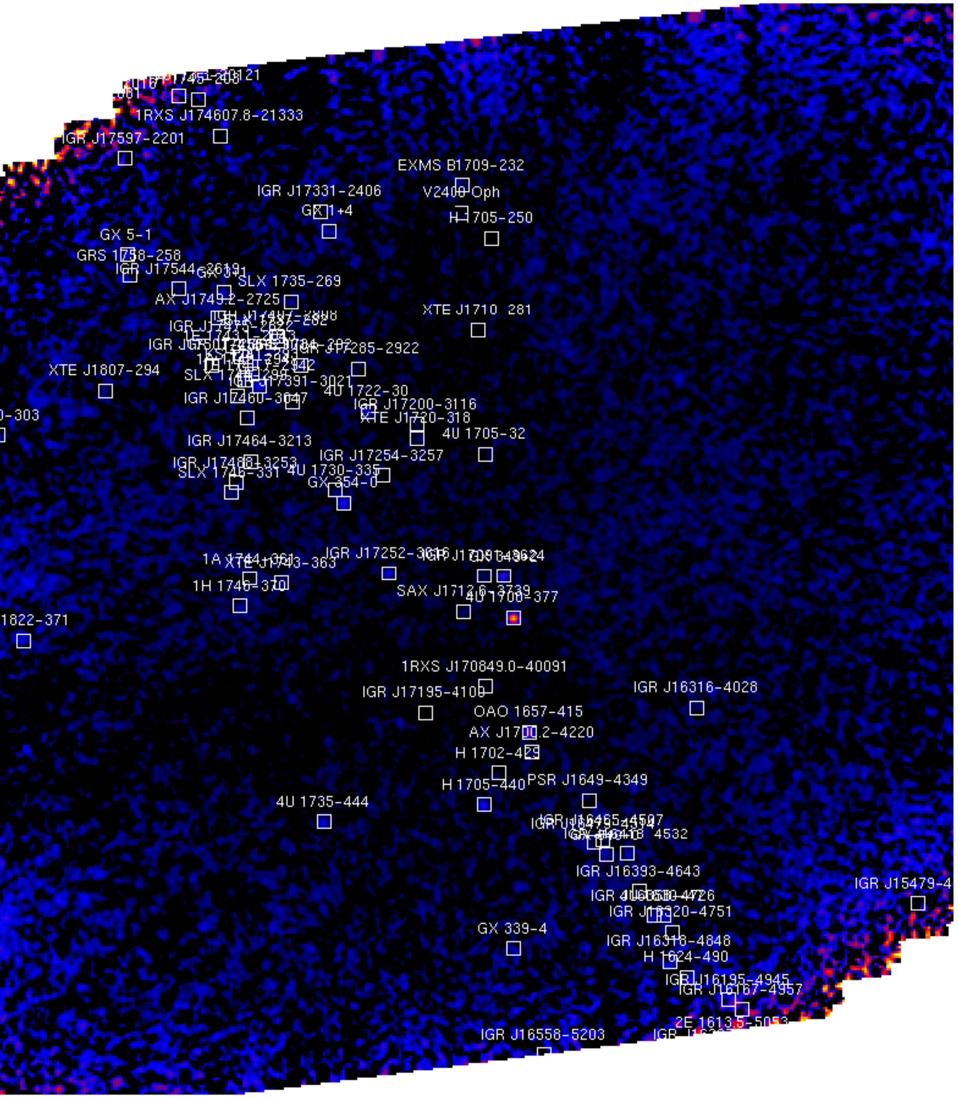
isgri_ima parameters

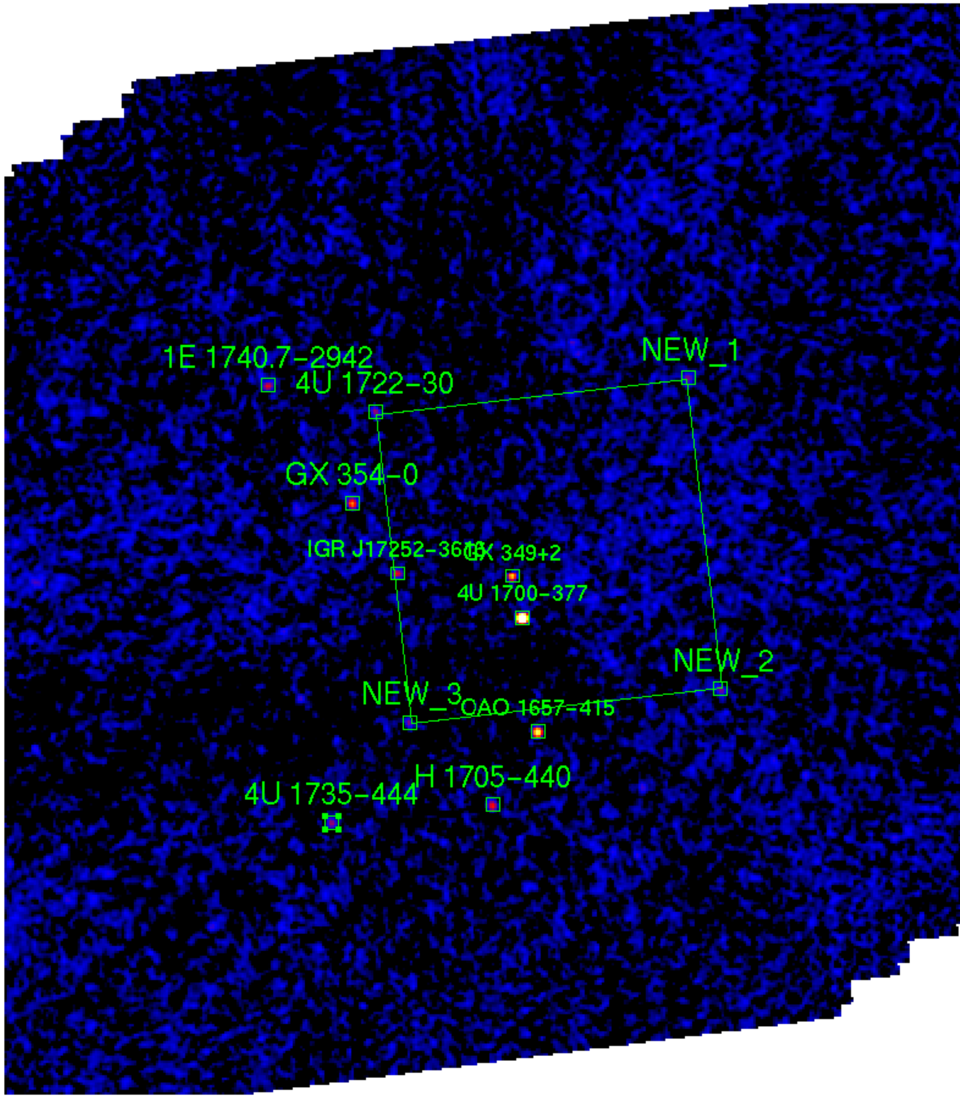


Inside isgri_sky_ima.fits file

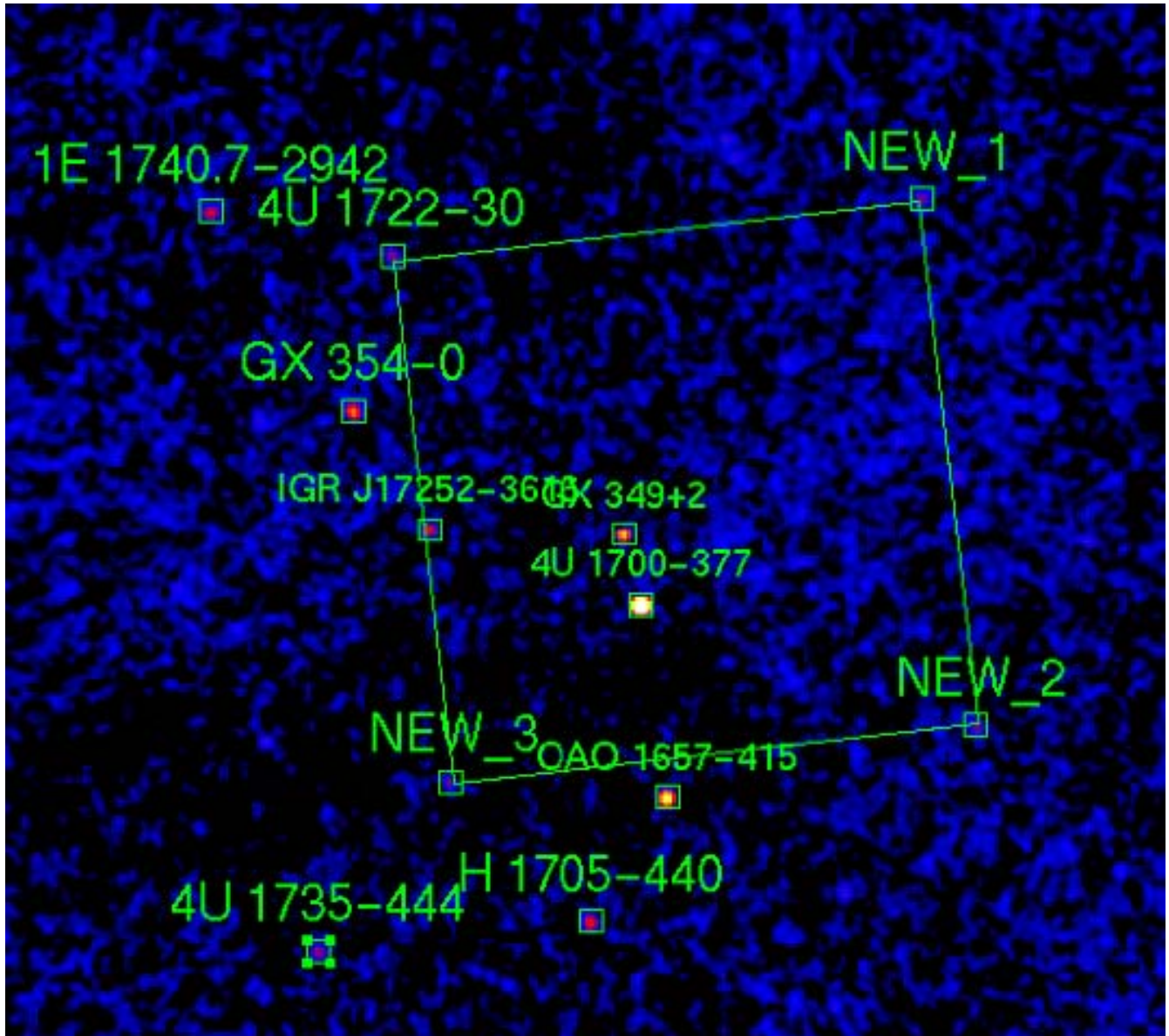


Results…


1RXS J174607.8-21333
IGR J17597-2201
EXMS B1709-232
IGR J17331-2406
V2400 Oph
GX 1+4
H 1705-250
GX 5-1
GRS 1758-258
SLX 1735-269
AX J1749.2-2725
XTE J1710 281
IGR J17285-2922
XTE J1807-294
4U 1722-30
IGR J17200-3116
XTE J1720-318
4U 1705-32
IGR J17464-3213
IGR J17254-3257
4U 1730-335
GX 354-0
IGR J17252-3616
1H 1746-370
SAX J1712.6-3739
4U 1700-377
1822-371
1RXS J170849.0-40091
IGR J17195-4100
OAO 1657-415
IGR J16316-4028
AX J1700.2-4220
H 1702-429
H 1705-440
PSR J1649-4349
4U 1735-444
IGR J16393-4643
IGR J16320-4751
GX 339-4
IGR J16318-4848
H 1624-490
IGR J16195-4945
IGR J16167-4957
IGR J16558-5203


1E 1740.7-2942
4U 1722-30
NEW_1
GX 354-0
IGR J17252-3616
GX 349+2
4U 1700-377
NEW_2
NEW_3
OAO 1657-415
H 1705-440
4U 1735-444

Let's have a closer look…



THANKS FOR ATTETION!