The Swiss Army Knife of Astronomical Data

Francesco D’Eugenio
francesco.deugenio@ugent.be
The Swiss Army Knife of Astronomical Data

By three methods we may learn wisdom: **First**, by reflection, which is noblest; **second**, by imitation, which is easiest; and **third** by experience, which is the bitterest.
Overview

I. Storage (how to store the data)
II. Procurement (how to get the data)
III. Analysis (how to interrogate the data)
I. Data storage

- State of the art: HDF5, VOTable, ASDF
- Outdated: FITS
- Really, really outdated: plain text (csv, etc.)
I. Data storage

- State of the art: HDF5, VOTable, ASDF
- Outdated: FITS
- Really, really outdated: plain text (csv, etc.)

*the best is also the least used!*

*the worst are amongst the most common.*
I. Data storage

- State of the art: HDF5, VOTable, ASDF
- Outdated: FITS
- Really, really outdated: plain text (csv, etc.)

*the best is also the least used!*

*the worst are amongst the most common.*
I. Data storage: FITS files

- interface (quick): topcat, ds9, QFitsView
- interface (analysis): cfitsio, astlib, astropy
- tables & images
- multiple extensions
- headers → WCS
I. Data storage: FITS files

A FITS file (single extension) contains either an “image” or a table (also called binary table).
FITS Images
FITS “Images” (Cubes)
### FITS Tables

<table>
<thead>
<tr>
<th>CATID</th>
<th>phot</th>
<th>mag</th>
<th>n_comp_r</th>
</tr>
</thead>
<tbody>
<tr>
<td>int64</td>
<td>bytes4</td>
<td>float64</td>
<td>int64</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>7839</td>
<td>GAMA</td>
<td>15.003</td>
<td>4</td>
</tr>
<tr>
<td>8487</td>
<td>GAMA</td>
<td>14.068</td>
<td>7</td>
</tr>
<tr>
<td>15481</td>
<td>GAMA</td>
<td>14.888</td>
<td>5</td>
</tr>
<tr>
<td>16525</td>
<td>GAMA</td>
<td>16.892</td>
<td>3</td>
</tr>
<tr>
<td>16526</td>
<td>GAMA</td>
<td>16.816</td>
<td>4</td>
</tr>
<tr>
<td>16926</td>
<td>GAMA</td>
<td>16.395</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>9403800763</td>
<td>VST</td>
<td>15.0276728</td>
<td>8</td>
</tr>
<tr>
<td>9403800813</td>
<td>VST</td>
<td>13.8577576</td>
<td>8</td>
</tr>
<tr>
<td>9403800833</td>
<td>VST</td>
<td>15.4592133</td>
<td>5</td>
</tr>
<tr>
<td>9403800911</td>
<td>VST</td>
<td>15.5672979</td>
<td>5</td>
</tr>
<tr>
<td>9403801002</td>
<td>VST</td>
<td>15.1227217</td>
<td>7</td>
</tr>
<tr>
<td>9403801010</td>
<td>VST</td>
<td>14.4940319</td>
<td>9</td>
</tr>
</tbody>
</table>
### FITS Tables

<table>
<thead>
<tr>
<th>CATID</th>
<th>phot</th>
<th>mag</th>
<th>n_comp_r</th>
</tr>
</thead>
<tbody>
<tr>
<td>int64</td>
<td>bytes4</td>
<td>float64</td>
<td>int64</td>
</tr>
<tr>
<td>7839</td>
<td>GAMA</td>
<td>15.003</td>
<td>4</td>
</tr>
<tr>
<td>8487</td>
<td>GAMA</td>
<td>14.068</td>
<td>7</td>
</tr>
<tr>
<td>15481</td>
<td>GAMA</td>
<td>14.888</td>
<td>5</td>
</tr>
<tr>
<td>16525</td>
<td>GAMA</td>
<td>16.892</td>
<td>3</td>
</tr>
<tr>
<td>16526</td>
<td>GAMA</td>
<td>16.816</td>
<td>4</td>
</tr>
<tr>
<td>16926</td>
<td>GAMA</td>
<td>16.395</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>9403800763</td>
<td>VST</td>
<td>15.0276728</td>
<td>8</td>
</tr>
<tr>
<td>9403800813</td>
<td>VST</td>
<td>13.8577576</td>
<td>8</td>
</tr>
<tr>
<td>9403800833</td>
<td>VST</td>
<td>15.4592133</td>
<td>5</td>
</tr>
<tr>
<td>9403800911</td>
<td>VST</td>
<td>15.5672979</td>
<td>5</td>
</tr>
<tr>
<td>9403801002</td>
<td>VST</td>
<td>15.1227217</td>
<td>7</td>
</tr>
<tr>
<td>9403801010</td>
<td>VST</td>
<td>14.4940319</td>
<td>9</td>
</tr>
</tbody>
</table>
FITS structure: multiple extensions

- Data | Header
- Data 1 | Header 1
- Data 2 | Header 2
- Data N | Header N

a FITS file (single extension)
a FITS file (multiple extensions)
FITS structure: multiple extensions

a FITS file (multiple extensions)

...done right.
Why use FITS extensions?

A way to group together different FITS files
What are FITS extensions?
What are FITS headers?

The place for “everything else”

CRPIX1 = 25.5   Pixel coordinate
CRPIX2 = 25.5   Pixel coordinate
CDELT1 = -0.00013888888888888889 / [deg] Coordinate increment at reference point
CDELT2 = 0.00013888888888888889 / [deg] Coordinate increment at reference point
CTYPE1 = 'RA---TAN' / Right ascension, gnomonic projection
CTYPE2 = 'DEC--TAN' / Declination, gnomonic projection
CRVAL1 = 181.112792 / [deg] Coordinate value at reference point
CRVAL2 = 1.895972 / [deg] Coordinate value at reference point
EQUINOX = 2000.0 / [yr] Equinox of equatorial coordinates
WCS_SRC = 'Nominal' / WCS Source

The place for “everything else”
Want to know more?

https://fits.gsfc.nasa.gov/fits_standard.html

II. Data procurement

- Observations
- Archives
- Various
II. Data procurement: observations

- Guaranteed time or competitive proposals
- Proposals have generally biannual cadence
Proposal tools

- Exposure time calculator (telescope website)
- Observability charts (iobserve, staralt)
- Proposal template (Latex)
## Proposal success criteria

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Success Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compelling science case</td>
<td>Somewhat arbitrary</td>
</tr>
<tr>
<td>Quantitative time justification</td>
<td>Objective</td>
</tr>
<tr>
<td>Quantitative sample size</td>
<td>Objective</td>
</tr>
<tr>
<td>Quantitative expected results</td>
<td>Objective</td>
</tr>
<tr>
<td>Convincing telescope justification</td>
<td>Somewhat arbitrary</td>
</tr>
<tr>
<td>Informative, pretty figures</td>
<td>Absolutely objective</td>
</tr>
<tr>
<td>Robustness</td>
<td>Objective</td>
</tr>
</tbody>
</table>
Moon: on or off?

- Off for blue wavelengths and/or faint targets
- Moon has the same spectrum as the Sun → bluer than the sky background
- At red enough wavelength (>800 nm), sky emission dominated by other processes
II. Data procurement: various

- dedicated website
- email / usb stick / dropbox

→ difficult to obtain, unless searching for something specific.
II. Data procurement: archives

- CDS – Strasbourg astronomical data center (e.g. VizieR)
- SciServer (e.g. casjobs)
- Telescope archives (e.g. archive.eso.org, HST legacy)
Telescope archives

- ESO archive
- Keck archive
- HST Legacy archive
- Gemini archive
- Subaru …and many more
CDS: Aladin Sky Atlas

Useful to view the data, and to search data around a particular sky location.

There is a “lite” version that runs in the browser, and a desktop version, which allows to download catalogues, over plot.

See here, looking for all the sources around M87 (these are mostly globular clusters)
CDS: VizieR “interactive” query

Useful to look what’s around.
CDS: VizieR SQL-like query

Why?
- easy to reproduce
- easy to pass around
- pre-select the data

How?
- know table/column names (schema browser)
- learn the SQL language (the basics)
CDS: VizieR SQL-like query

Structured Query Language – the most common query language.

Astronomers also made their own version, called Astronomy Data Query Language ADQL
A working example with SDSS

```sql
SELECT TOP 5000
  s.ra, s.dec,
  s.z, s.zErr,
  s.velDisp, s.velDispErr,
  s.snMedian,
  l Sigma_Ha_6562 as sigHa, l Sigma_Ha_6562 Err as sigHaErr,
  l Sigma_Hb_4861 as sigHb, l Sigma_Hb_4861 Err as sigHbErr,
  mpa.lgm tot p50 as lgm, (mpa.lgm tot p84 - mpa.lgm tot p16) / 2. as lgm Err,
  mpa.sfr_tot_p50 as lsfr, (mpa.sfr_tot_p84 - mpa.sfr_tot_p16) / 2. as lsfr_Err
INTO mydb.meshclass
FROM SpecObj AS s
JOIN emissionLinesPort as l ON s.specobjid=l.specobjid
JOIN galSpecExtra as mpa on s.specobjid=mpa.specobjid
WHERE mpa.lgm_tot_p50 > 9 AND s.velDispErr > 0 AND l Sigma_Ha_6562 err > 0
```
A working example with SDSS

```
SELECT TOP 5000
  s.a, s.dec,
  s.z, s.zErr,
  s.velDisp, s.velDispErr,
  s.snMedian,
  l.Sigma_Ha_6562 as sigHa, l.Sigma_Ha_6562_Err as sigHaErr,
  l.Sigma_Hb_4861 as sigHb, l.Sigma_Hb_4861_Err as sigHbErr,
  mpa.lgm_tot_p50 as lgm, (mpa.lgm_tot_p84-mpa.lgm_tot_p16)/2. as lgm_Err,
  mpa.sfr_tot_p50 as lsfr, (mpa.sfr_tot_p84-mpa.sfr_tot_p16)/2. as lsfr_Err
FROM SpecObj AS s
JOIN emissionLinesPort AS l ON s.specobjid=l.specobjid
JOIN galSpecExtra AS mpa on s.specobjid-mpa.specobjid
WHERE mpa.lgm_tot_p50 > 9 AND s.velDispErr > 0 AND l.Sigma_Ha_6562_Err > 0
```
How do I know table/column names?
Example from SDSS Schema:

Search for keywords
Search by table/view
Our query:

```
SELECT TOP 5000
 s.ra, s.dec,
 s.z, s.zErr,
 s.velDisp, s.velDispErr,
 s.snMedian,
 l.Sigma_Ha_6562 as sigHa, l.Sigma_Ha_6562_Err as sigHaErr,
 l.Sigma_Hb_4861 as sigHb, l.Sigma_Hb_4861_Err as sigHbErr,
 mpa.lgm_tot_p50 as lgm, (mpa.lgm_tot_p84-mpa.lgm_tot_p16)/2. as lgm_Err,
 mpa.sfr_tot_p50 as lsfr, (mpa.sfr_tot_p84-mpa.sfr_tot_p16)/2. as lsfr_Err
INTO mydb.methclass
FROM SpecObj AS s
JOIN emissionLinesPort as l ON s.specobjid=l.specobjid
JOIN galSpecExtra as mpa on s.specobjid=mpa.specobjid
WHERE mpa.lgm_tot_p50 > 9 AND s.velDispErr > 0 AND l.Sigma_Ha_6562_err > 0
```
II. Data procurement: Summary

- Archives (catalogues) (SQL, Schema)
- Archives (images/spectra)
- Observations
III. Quick analysis tools

- What is it? A test that takes < 30’
- Why? Evaluate data quality, feasibility
- What tools? Plotting & robust fitting algorithms
Topcat: see the data
Topcat: see the data
Visually inspecting the data

identify outliers
identify problems with the data
Visually inspecting the data

- Identify outliers
- Identify problems with the data
Visually inspecting the data can help identify outliers and problems with the data.
Visual inspection in SDSS

Total time: 1 minute (top)
A running median can help...

In Python:

```python
>>> binned_statistic(sigstar, sigHa, bins=..., statistic=np.median)
```

Python3 `methclass_running_median.py`
A running median can help...

...but be sensible!
Least Squares

Something’s wrong…
Need to reject outliers!

Difficult and often lengthy process
Labour intensive

$No\ outlier\ rejection$

\[a = 0.976 \pm 0.017\]
\[b = 1.945 \pm 0.003\]
Least Trimmed Squares (LTS)

Least squares minimization + sigma clipping but:

➢ proceeds inside-out
➢ guaranteed to converge

For the record:
slope = b = 1.08 ± 0.02

python3 methclass_lts_linefit.py
Contour-based rejection

Quick to implement
Can be recycled for Bayesian approach
Applying contour rejection.

Least squares minimization + sigma clipping but:

➢ proceeds inside-out
➢ guaranteed to converge

For the record:
slope = b = 1.08 +/- 0.02

Least squares + contour rej.

\[ a = 1.019 \pm 0.016 \]
\[ b = 1.942 \pm 0.003 \]
Towards a probabilistic model

Least squares minimization + sigma clipping but:

- proceeds inside-out
- guaranteed to converge

For the record:
slope = b = 1.08 ± 0.02
Least Trimmed Squares (LTS)

Least squares minimization + sigma clipping but:

- proceeds inside-out
- guaranteed to converge

For the record:
slope = b = 1.08 \pm 0.02

\begin{align*}
a &= 1.052 \pm 0.014 \\
b &= 1.945 \pm 0.003
\end{align*}
How to construct an adequate model?

create a probabilistic model

\[
p(model|data) \propto p(model) \times p(data|model)
\]

ALL algorithms we have seen so far assume a model

...know thy model
Towards a probabilistic model

Least squares minimization + sigma clipping but:

➢ proceeds inside-out
➢ guaranteed to converge

For the record:
slope = b = 1.08 ± 0.02
Towards a probabilistic model

Least squares minimization + sigma clipping but:

- proceeds inside-out
- guaranteed to converge

For the record:
slope = \( b = 1.08 \pm 0.02 \)
Towards a probabilistic model

Least squares minimization + sigma clipping but:

➢ proceeds inside-out
➢ guaranteed to converge

For the record:

slope = $b = 1.08 \pm 0.02$

Bayesian Contour Rejection

$a = 1.268^{+0.021}_{-0.022}$

$b = -0.622^{+0.045}_{-0.045}$
Towards a probabilistic model

Least squares minimization + sigma clipping but:

- proceeds inside-out
- guaranteed to converge

For the record:
slope = b = 1.08 ± 0.02
Locally wEighted Scatter-plot Smoother = LOESS