

Exploring Gaia data with TOPCAT and STILTS

Mark Taylor,
University of Bristol,
m.b.taylor@bristol.ac.uk

Adapted by Enrique Solano & Francisco Jimenez-Esteban

February 25, 2020

TOPCAT: <http://www.starlink.ac.uk/topcat/>

STILTS: <http://www.starlink.ac.uk/stilts/>

Mailing list: topcat-user@jiscmail.ac.uk

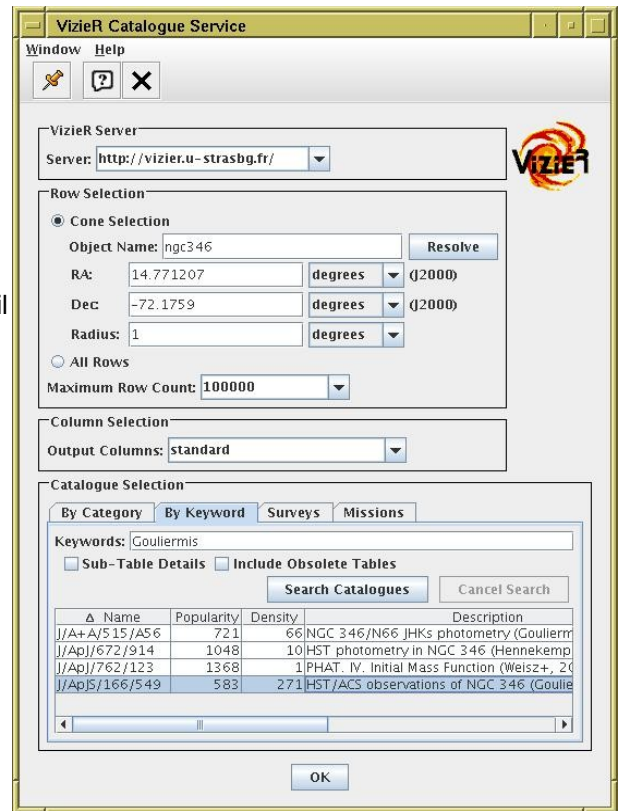
1 Match Gaia and HST observations

In this example we have a local catalogue from a publication by Gouliermis et al. 2006 [3], available in VizieR as J/ApJS/166/549. This contains about 100,000 sources observed by the ACS instrument on the Hubble Space Telescope at epoch J2004.6 of stars in NGC346, a cluster in the Small Magellanic Cloud. We match these positions with positions in the main Gaia catalogue at J2015.5 and see what we can find out about the relative astrometry of the two catalogues.

1.1 Acquire HST observations

There are various ways to do this, but here we will use TOPCAT's VizieR dialogue window, which talks directly to the VizieR catalogue service.

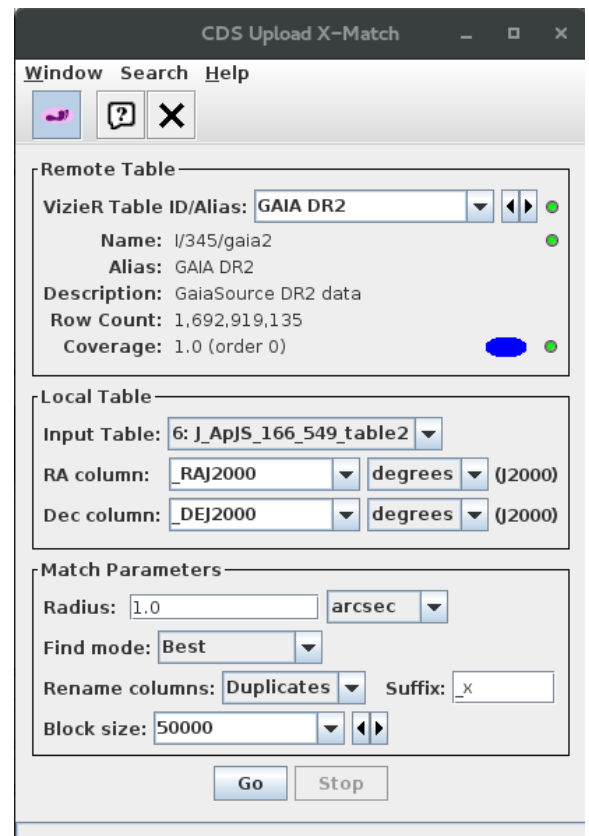
1. In the TOPCAT main window: [VO --> VizieR Catalogue Service](#)
2. **Object Name:** ngc346 --> Click **Resolve**. The RA and Dec boxes will be filled in.
3. **Radius:** 1 (degrees)
4. **Maximum Row Count:** unlimited
5. **Output Columns:** standard
6. Catalogue selection panel: **By Keyword** tab
7. **Keywords:** Gouliermis. Hit **Search catalogues** button
8. Select J/ApJS/166/549
9. Hit the **OK** button at the bottom. A new table with 99,079 rows should be loaded



1.2 Crossmatch with Gaia

Now we want to find associations of the HST objects with sources from Gaia DR2. Use the CDS X-Match service from TOPCAT. This uploads a local table to the CDS X-Match service, where the match is made against the Gaia DR2 catalogue (or any other catalogue in VizieR). The resulting matched catalogue is then received as a new table in TOPCAT.

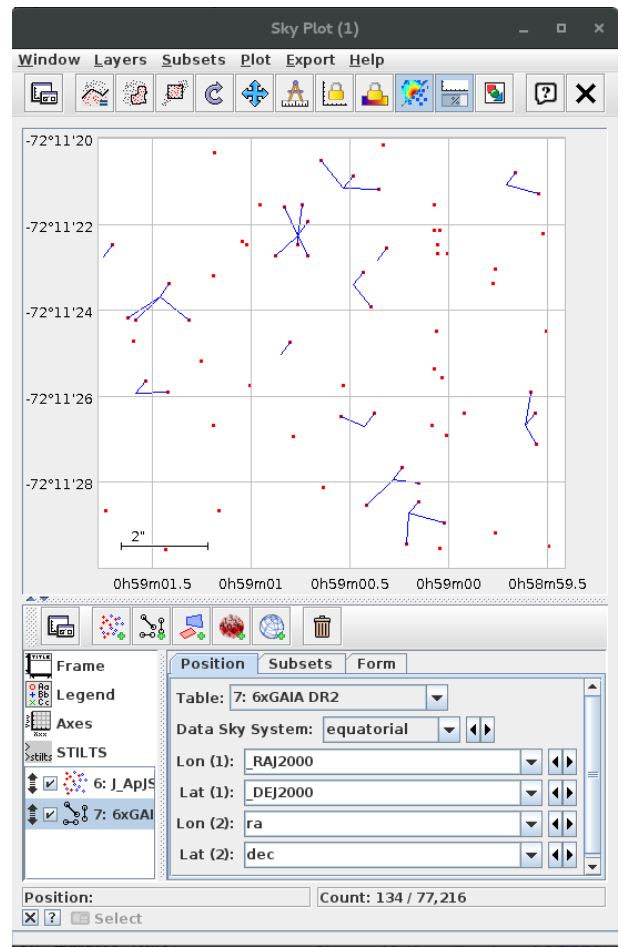
1. In the TOPCAT main window: [VO --> CDS Upload X-match](#)
2. **VizieR Table ID/Alias:** GAIA DR2
3. **Input Table:** J_ApJS_166_549_table2 (or whatever the HST table is called in your TOPCAT main window)
4. **RA column:** _RAJ2000", **Dec column:** _DEJ2000" (should be filled in automatically)
5. **Radius:** 1 (arcsec)
6. **Find Mode:** Best
7. Hit **Go**; within a few seconds, it should inform you that a new table has been loaded, with 8721 rows.
8. Look at the columns of the new table (all HST followed by all Gaia): [Views --> Column info](#)



1.3 Visualise the crossmatch

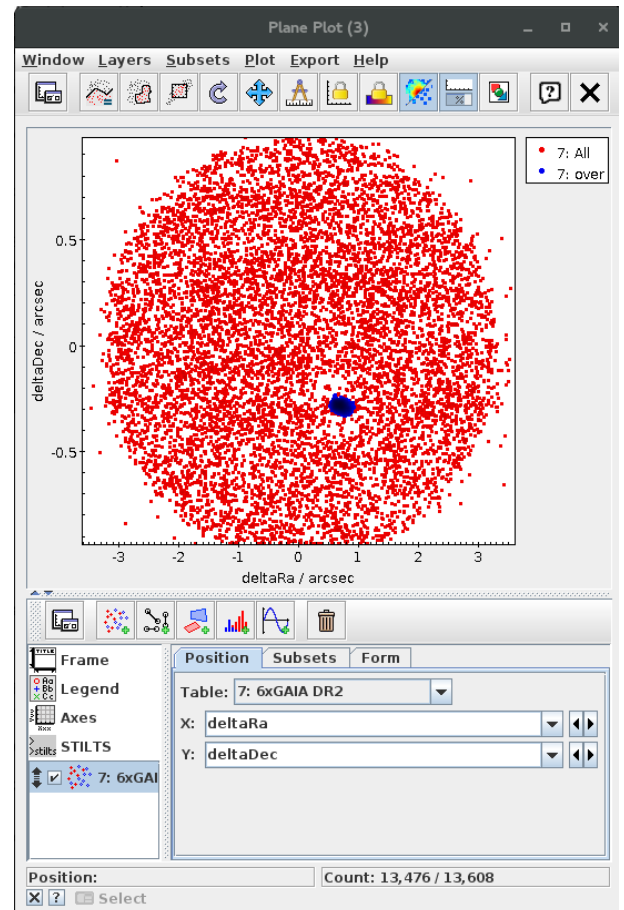
1. in the TOPCAT main window: [Graphics --> Sky Plot](#)
2. Plot the HST observations: **Table:** J_ApJS_166_549_table2, **Lon:** _RAJ2000, **Lat:** _DEJ2000
3. Overplot the actual matches. In the Sky Plot window -->
 - o [Layers --> Add pair control](#)
 - o **Table:** 2xGAIA DR2 (or whatever the xmatch result table is called)
 - o **Lon(1):** _RAJ2000, **Lat(1):** _DEJ2000 (HST)
 - o **Lon(2):** ra, Lat(2): dec (Gaia)
4. Zoom in to look at the associations. There are too many!

Visualising the results of a crossmatch is very often a good idea, unless you're pretty sure what you're going to get. Here, you can see it was crucial to understand the results: each Gaia source has been matched to several HST sources.



1.4 Investigate and identify matches


1. Add new columns giving RA/Dec discrepancies between HST and Gaia positions:
 - o In the TOPCAT main window: [Views --> Column Info.](#)
then hit
 - Name: deltaRa, Expression: $3600*(ra - _RAJ2000)$,
 - Units: arcsec
 - Hit **OK**
 - **Repeat the same process**
 - Name: deltaDec, Expression: $3600*(dec - _DEJ2000)$,
 - Units: \arcsec"
 - Hit **OK**
2. In the TOPCAT main window:
 - o [Grapics > Plane Plot](#)
 - o **X:** deltaRa
 - o **Y:** deltaDec
3. Select the overdense region (zoom in for a better selection)
 - o In the Plane Plot window
 - [Subset --> Draw algebraic subset: Polygon](#)
 - [Subset -> Finish drawing subset to end](#)
 - A new window ("Add Figure Subset(s)") pops up. Include a name in the **Subset Name** box.
4. In the TOPCAT main window, hit to determine mean and standard deviation for deltaRa and deltaDec, giving apparent movement of NGC346 between epoch of Gouliermis observations (J2004.6) and Gaia (J2015.5).



2. Local Hertzsprung-Russell Diagram

In this example we will use a TAP query to download all the nearby Gaia sources with good astrometry and photometry, and calculate their absolute magnitudes to construct an HR diagram, performing a couple of cleaning operations to improve the data.



2.1 Acquire data from TAP service

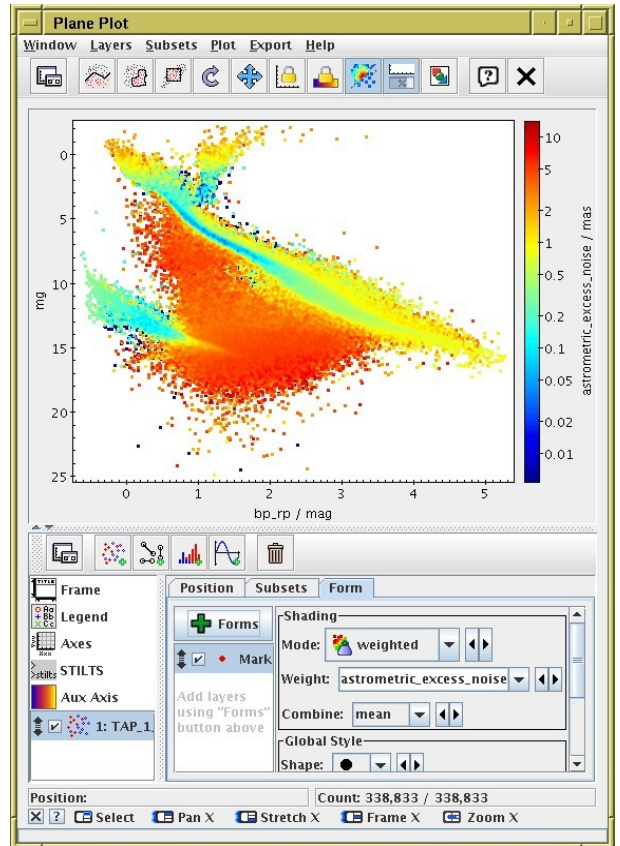
1. Open the TAP window  **VO | Table Access Protocol (TAP) Query**
2. Select one of the Gaia services (probably the ESA one) and **Use Service**
3. Choose **Mode**: "Asynchronous" (just above the ADQL text entry panel). This query may take a minute or two, so a synchronous query might time out (with the unhelpful result, probably, "TAP response is not a VOTable").
4. Execute the following query:

```
SELECT ra, dec, parallax, phot_g_mean_mag, bp_rp,
       astrometric_excess_noise, phot_bp_rp_excess_factor
FROM gaiadr2.gaia_source
WHERE parallax > 10
      AND parallax_over_error > 10
      AND phot_bp_mean_flux_over_error > 10
      AND phot_rp_mean_flux_over_error > 10
```



You should get a table with 338833 sources; they are nominally within 100pc, and have parallax, BP and RP with small errors. In particular, the parallax error is small enough that its inverse is a reasonable estimate of distance.

2.2 Plot HRD


1. Add a new column calculating absolute G magnitude, using parallax:
Columns | New Synthetic Column in Column Info window:
Name: "g abs",
Expression: "phot_g_mean_mag + 5*log10(parallax/100)", **Units:** "mag"
2. Make a  **Graphics | Plane Plot**, with X: "bp rp" (BP - RP colour), Y: "g abs" (absolute G magnitude). Use the  **Axes** control, **Coords** tab, **Y Flip** checkbox to flip it the right way round. Structure visible, but lots of interlopers.
3. Play around with different **Shading Modes** in **Form** tab.
4. In **Form** tab, use **Mode:weighted**, **Weight:astrometric_excess_noise**. Most of the interlopers have a high value of this parameter, which characterises the goodness of fit of the astrometric solution to the observations.







2.3 Exclude astrometrically suspect sources

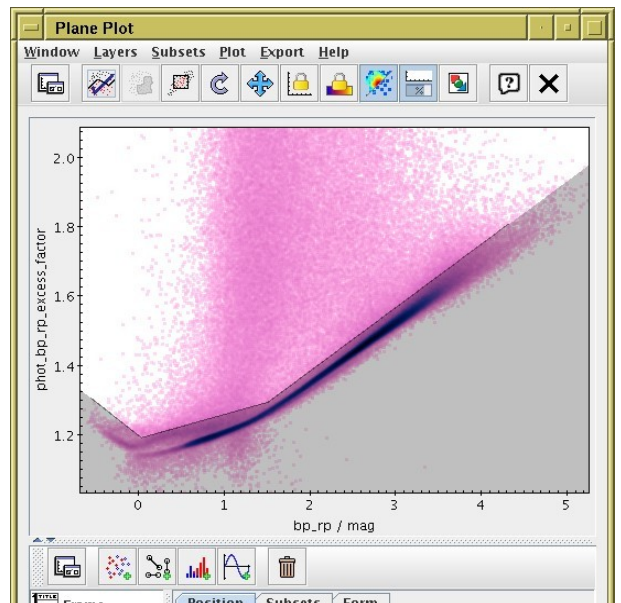
1. Create a subset using an algebraic expression that excludes the spurious points: go to the  **Views | Row Subsets** window and use the  **Subsets | New Subset** action: **Subset Name:** "astro_ok", **Expression:** "astrometric excess noise < 1"
2. Go back to the plot, and make sure only the astro ok subset is plotted (**Subsets** tab). Now there are fewer spurious sources.

2.4 Exclude photometrically suspect sources

1. Open a new  **Graphics | Plane Plot**, with X: "bp rp" (BP - RP colour), Y: "phot bp rp excess factor".

The quantity on the Y axis is some measure of photometric reliability. High values are bad, but how high is colour-dependent. We will define a region in this space to exclude the unusually high values.

2. Click the  **Subsets | Draw Subset Polygon** action
3. Select **Point inclusion mode**: “BELOW” in the popup
4. Click on a few points above the overdense region until the shaded area roughly covers it.
5. When you’re done, click on the  button again.
6. Fill in the **Subset Name** field in the popup (e.g. phot_ok) and hit **OK**
7. Go to the  **Views | Row Subsets** window, where you can see the new subset alongside astrom ok.
8. Create a new subset combining the two:
 -  **Subsets | New Subset** action: **Subset Name**: “ok” **Expression**: “astrom ok && photom ok”
9. Go back to the plot, and make sure only the ok subset is plotted (**Subsets** tab). Now there are fewer spurious sources.



3. Local Hertzsprung-Russell Diagram using STILTS

In this section we will reproduce the cleaned HRD from the previous section, but this time from the command line. Using TOPCAT interactively to work out how to write STILTS scripts for later batch use like this is a common pattern of work.

3.1 Acquire data from TAP service

1. Run the query and output the result to a local FITS file hrd.fits:

```
java -jar stilts.jar tapquery sync=false maxrec=1000000 out=hrd=fits
```

tapurl - Base URL of TAP service: <http://gaia.ari.uni-heidelberg.de/tap>

adql - ADQL query text:

```
select ra,dec,parallax,phot_g_mean_mag,bp_rp,astrometric_excess_noise,phot_bp_rp_excess_factor from
gaiadr2.gaia_source where parallax>10 and parallax_over_error>10 and phot_bp_mean_flux_over_error>10 and
phot_rp_mean_flux_over_error>10
```

3.2 Manipulate the downloaded table using tpipes

We will perform some processing on the downloaded table using the **tpipes** command. This reads an input table, optionally performs operations on it, and writes it to output, either to a file (in the same or different format) or some other destination. It works like a Unix pipeline.

The operations are defined by adding *filters* such as “select” (retain only some rows) or “addcol” (add a new column). Filters are specified by adding using zero or more “cmd=” parameters on the command line, and documented in Section 6.1: “Processing Filters” of the STILTS manual.

The default output mode is write to a file, but other options such as “meta” (display column metadata), “count” (count rows) and “stats” (calculate mean, st.dev etc) are also available. Select non-default output modes using the “omode=” parameter on the command line, documented in Section 6.4: “Output Modes” of the manual.

1. See what columns the query result table has:

```
java -jar stilts.jar tpipes in=hrd.fits omode=meta
```

2. Prepare an expression for a new column giving absolute magnitude (see item 1 from section 4.2)
3. Prepare expressions for rows corresponding to the astrometry and photometry selections from sections 4.3 and 4.4; go back to TOPCAT's **Subsets** window, and see the expressions used to restrict the sources in the final HRD.
4. Put items 3 and 4 together to produce a plot-ready table (note use of quotes):

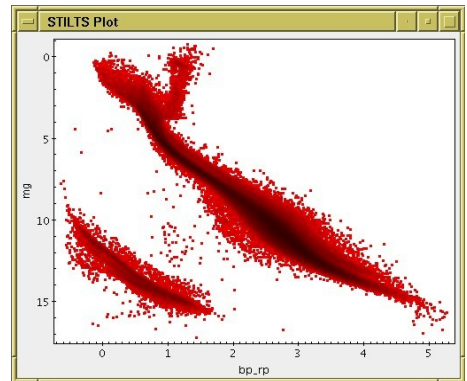
```
java -jar stilts.jar tpipe in=hrd.fits cmd='addcol g_abs "phot_g_mean_mag+5*log10(parallax/100)" ' cmd='select "astrometric_excess_noise<1" ' cmd='select "phot_bp_rp_excess_factor < polyLine(bp_rp,-0.56,1.307,0.03,1.192,1.51,1.295,4.31,1.808)" ' out=hrd_clean.fits
```

3.3 Plot the diagram

1. Plot the HRD on the screen using the 2d plotting command plot2plane:

```
java -jar stilts.jar plot2plane layer=mark in=hrd_clean.fits x=bp_rp  
y=g_abs yflip=true
```

2. By default, a “live” plot shows up on the screen: use the mouse to zoom and drag it around like in TOPCAT.
3. Output it to a graphics file instead, by adding the “out” parameter, e.g. “out=hrd.png” or “out=hrd.pdf”.



4 Playing with STILTS

A few more examples of operations you can do with STILTS

Download the file **sample.txt** from the school web page.

See how big it is:

```
java -jar stilts.jar tpipe in=sample.txt ifmt=ascii omode=count
```

Calculate statistics for some columns:

```
java -jar stilts.jar tpipe in=sample.txt ifmt=ascii cmd='keepcols "PLL Gmag"' omode=stats
```

Find the 5 sources with smallest / largest parallax:

```
java -jar stilts.jar tpipe in=sample.txt ifmt=ascii cmd='keepcols "PLL"' cmd='sorthead 5 PLL'
```

```
java -jar stilts.jar tpipe in=sample.txt ifmt=ascii cmd='keepcols "PLL"' cmd='sorthead -tail 5 PLL'
```

You can do plots as well. Note that these plots are interactive: you can resize the windows and navigate with the mouse just like in topcat. If instead you add the parameter `out=foo.png` it will write the image directly to a PNG. Similarly for `pdf`, `eps`, `jpeg`, `gif`.

Plot all the points on the sky:

```
java -jar stilts.jar plot2sky in=sample.txt ifmt=ascii layer1=mark lon1=ra lat1=dec
```

Plot histogram of parallaxes:

```
java -jar stilts.jar plot2plane layer1=histogram in=sample.txt ifmt=ascii x1=PLL ylog=true
```

STILTS does a ton of other stuff that there's not space to explain here. Here is an example of downloading the Gliese catalogue from the VizieR TAP service and crossmatching it remotely against TGAS, to identify Gliese sources with TGAS counterparts in 3arcsec. The result is automatically uploaded in TOPCAT.

```
java -jar stilts.jar tapquery sync=true out=gliese.fits
```

tapurl - Base URL of TAP service: <http://tapvizier.u-strasbg.fr/beta/TAPVizieR/tap>

adql - ADQL query text: `select * from "V/70A/catalog"`

```
java -jar stilts.jar cdsskymatch cdstable=I/337/TGAS in=gliese.fits ra=_ra_icrs  
dec=_de_icrs radius=3 find=best omode=topcat
```