SPACIOUS

Gaia visualisation services on the SPACIOUS platform

André Moitinho
University of Lisbon
andre@sim.ul.pt

Space astronomy science platforms focus week, 9–12 December
https://indico.ph.ed.ac.uk/event/374/

Funded by the European Union

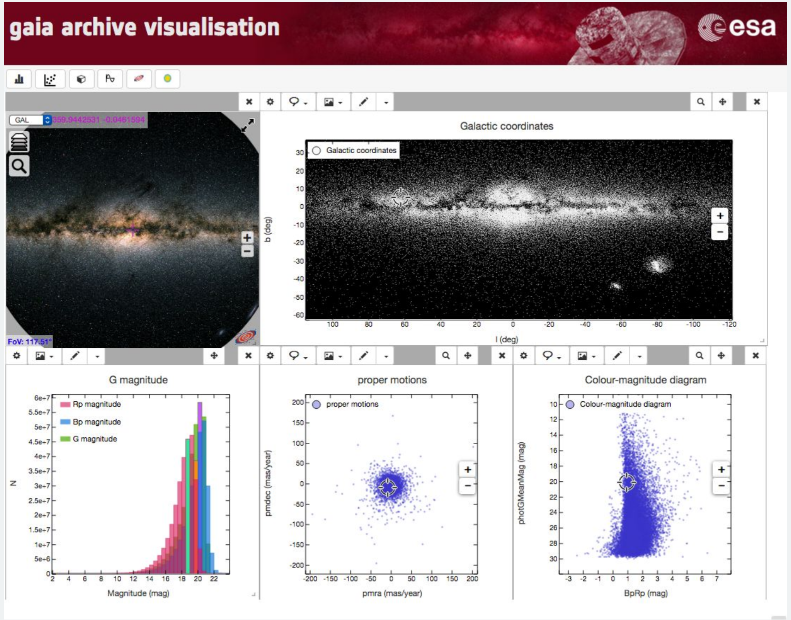
SPACIOUS is a Horizon Europe HORIZON-CL4-2023-SPACE-01-71 project funded under grant agreement no. 101135205

https://spacious.ub.edu/

GAVS

Gaia Archive Visualisation Service (GAVS)
http://gea.esac.esa.int/visualization

- Interactive visual exploration service
- Client-server architecture with REST API
- Server@ESA: Heavy lifting (and most of the code). **Code-to-data paradigm**
- Handles users (thousands? many simultaneously) on limited hardware resources.
- Scalable

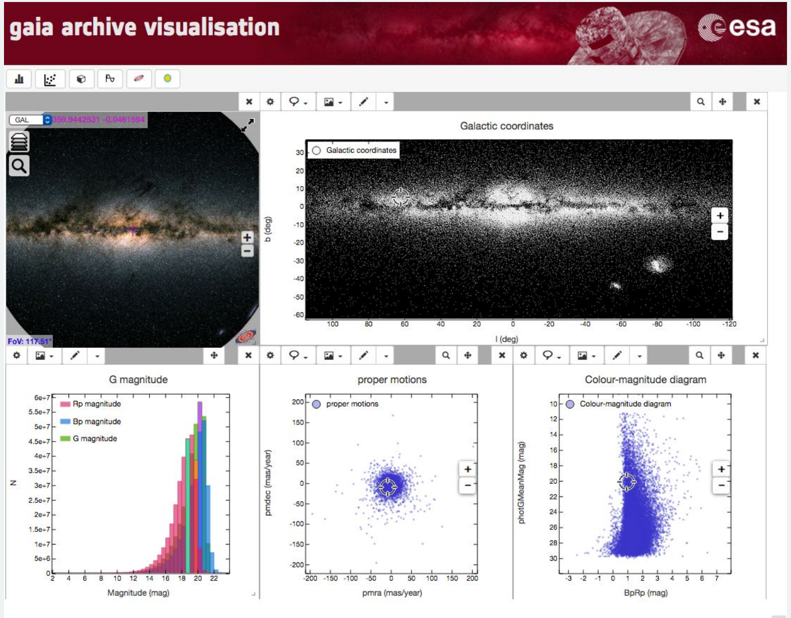


SPACIOUS

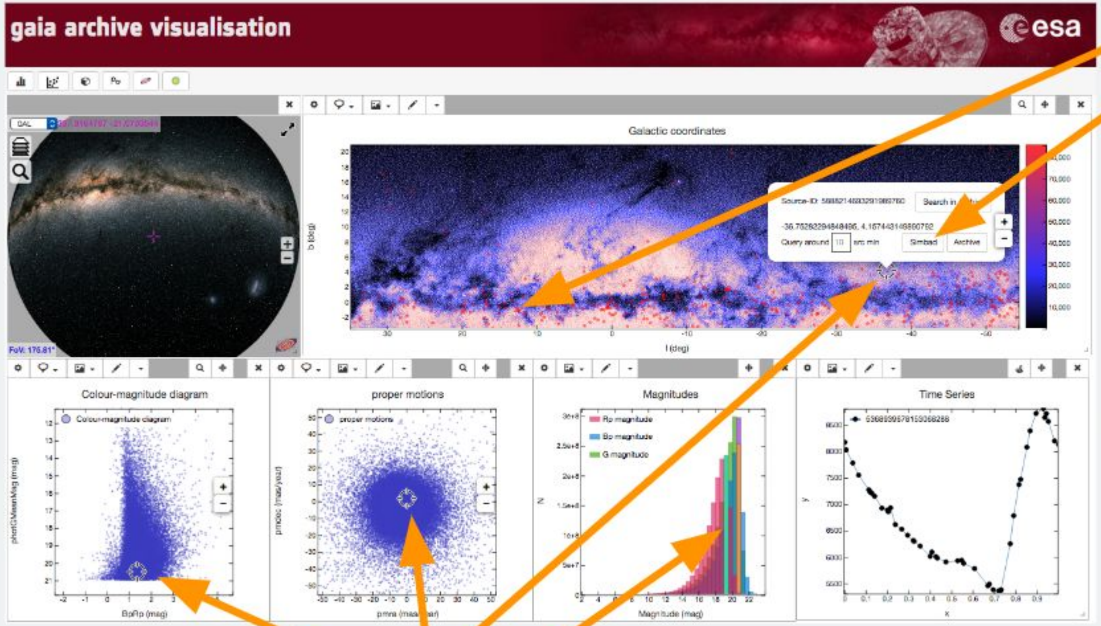
GAVS

Goals

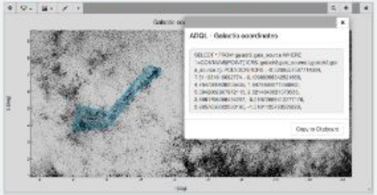
- Ability to display essential plots; scatter plots, density plots, histograms,...
- At any level of detail. From the overall catalog to individual stars.
- Interactive (zoom, pan, select, etc)
- Facilitate archive queries based on visual inspection at any level of detail



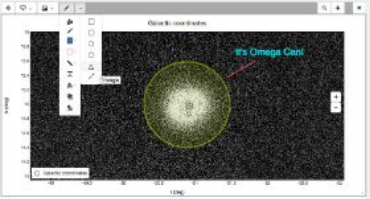
GAVS



- overlay catalogues
- identify sources in Simbad
- visual ADQL queries



- annotate and save hard copies



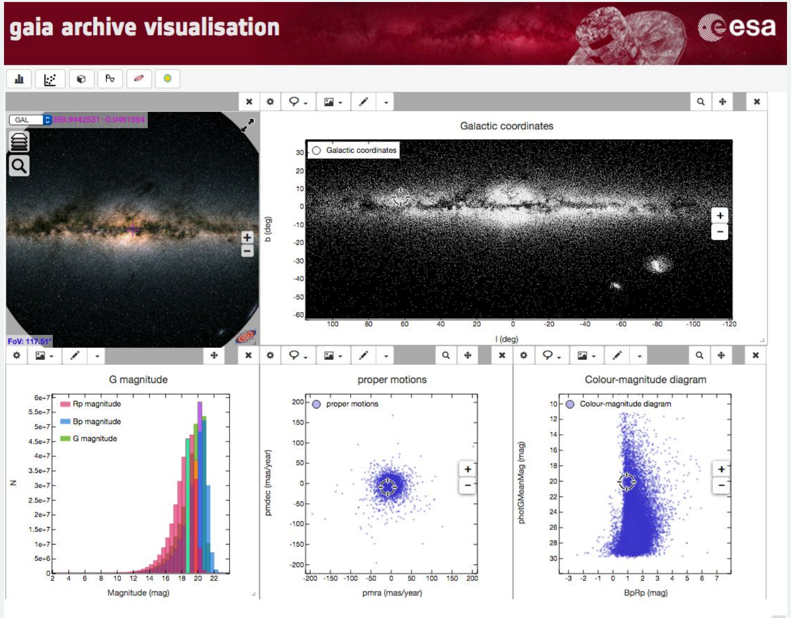
- linked views
- time series analysis

http://gea.esac.esa.int/visualization

SPACIOUS

GAVS - Limitations

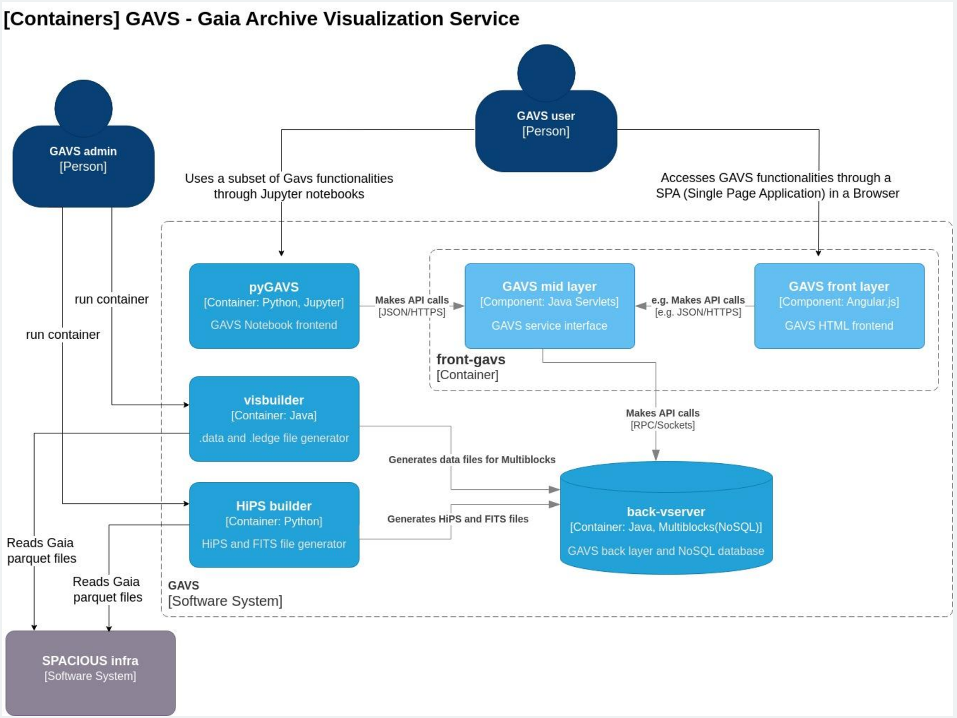
- Fixed set of plots: Arbitrary, user-defined, not currently supported
- (no) interoperability with archive
- Functionalities offered via GUI
- Pre-computed indices (plots) done by an operator
- Limited computational resources: HD, also CPU+RAM for computing indices.



SPACIOUS addresses these limitations

GAVS @ SPACIOUS

- **Containerisation and Deployment**
  - Refactored and packaged the GAVS components for cloud-native environments (and python interfacing).
- Solves
  - Extended resources
  - User precomputation of indices
  - Interoperability with data archive



[Containers] GAVS - Gaia Archive Visualization Service

GAVS admin
[Person]

GAVS user
[Person]

Uses a subset of Gavs functionalities through Jupyter notebooks

Accesses GAVS functionalities through a SPA (Single Page Application) in a Browser

run container

run container

pyGAVS
[Container: Python, Jupyter]
GAVS Notebook frontend

Makes API calls
[JSON/HTTPS]

GAVS mid layer
[Component: Java Servlets]
GAVS service interface

e.g. Makes API calls
[e.g. JSON/HTTPS]

GAVS front layer
[Component: Angular.js]
GAVS HTML frontend

front-gavs
[Container]

visbuilder
[Container: Java]
.data and .ledge file generator

Makes API calls
[RPC/Sockets]

Generates data files for Multiblocks

back-vserver
[Container: Java, Multiblocks(NoSQL)]
GAVS back layer and NoSQL database

HiPS builder
[Container: Python]
HiPS and FITS file generator

Generates HiPS and FITS files

Reads Gaia parquet files

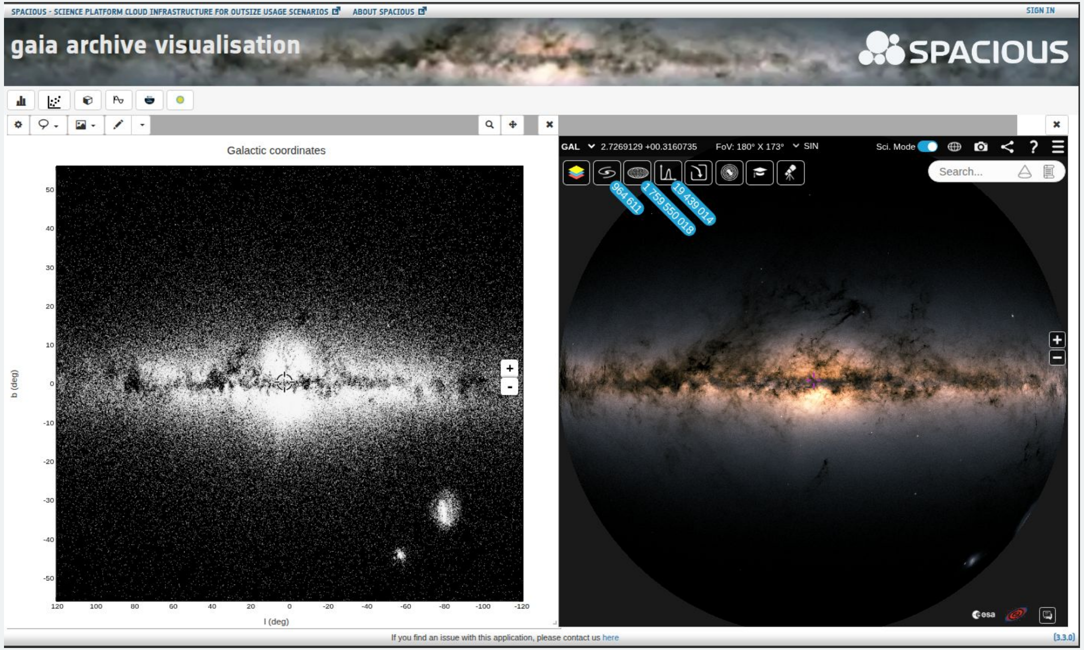
Reads Gaia parquet files

GAVS
[Software System]

SPACIOUS infra
[Software System]

GAVS @ SPACIOUS

- Frontend skin support. GUI customisation for branding consistency in different projects.
- ESASky support:
  - Archive interoperability a la EUCLID archive
  - With linked views!

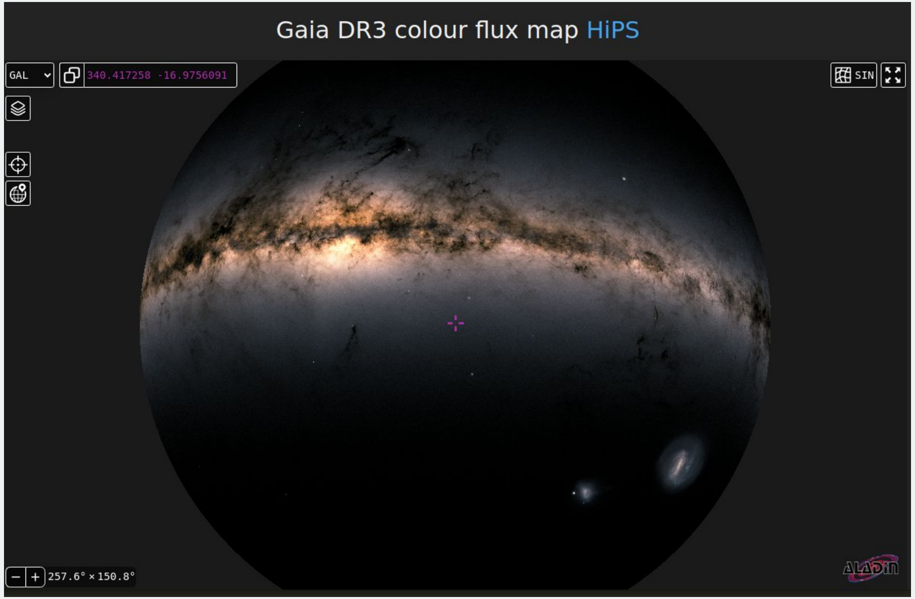


If you find an issue with this application, please contact us here

SPACIOUS

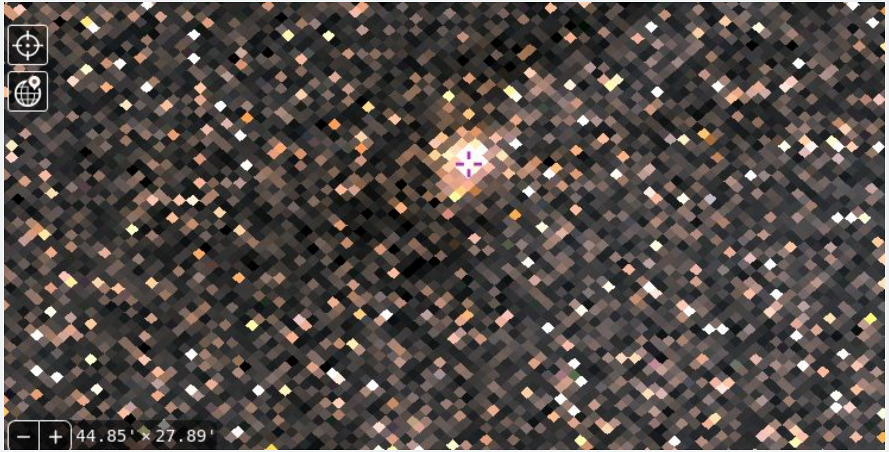
GAVS @ SPACIOUS

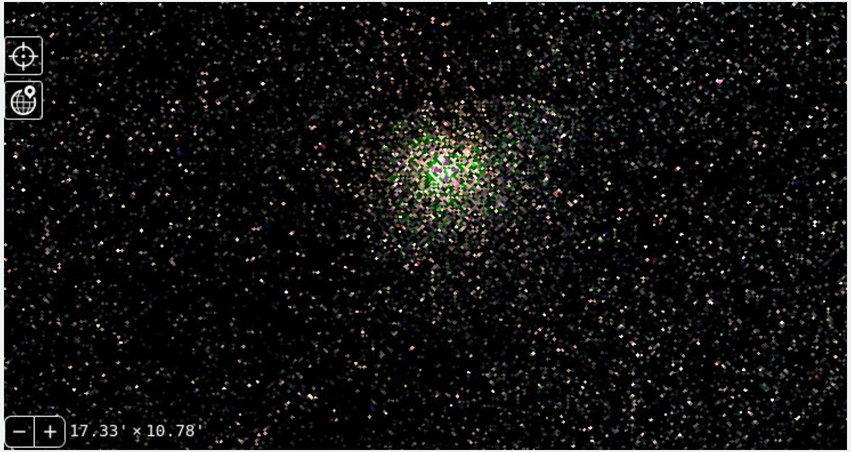
- **Creation of Visualisations (preprocessing)**
  - Support for Parquet input files
  - Containerised visbuilder to integrate with the BDAF infrastructure
  - Optimised memory handling (previously requiring operator oversight)
  - Complete rewrite of HiPS creator Memory limits removed. High resolution reachable.



Gaia DR3 colour flux map HiPS

Higher res HiPS with SPACIOUS





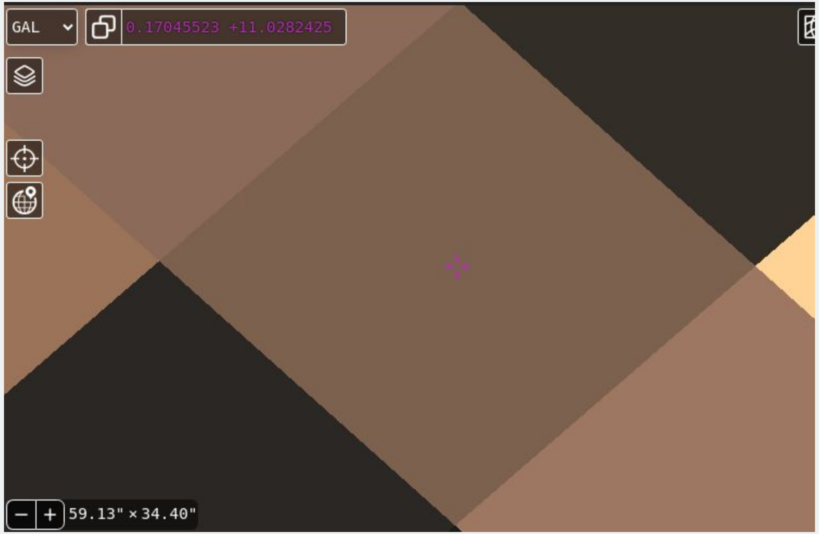
NGC 6287

Left: HiPS order 4 (~34.5”) - current resolution at GAVS.

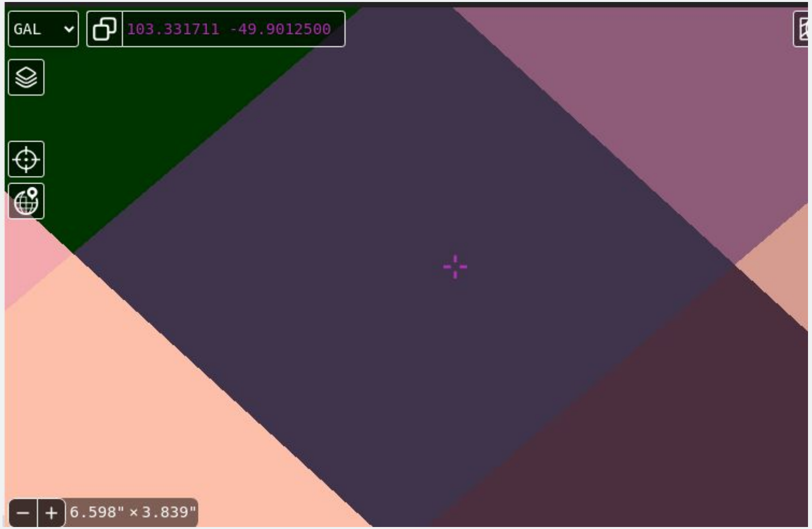
Right: HiPS level 7 (~4”) - resolution with SPACIOUS

SPACIOUS

Higher res HiPS with SPACIOUS



Left: HiPS order 4 (~34.5”) - current resolution at GAVS.

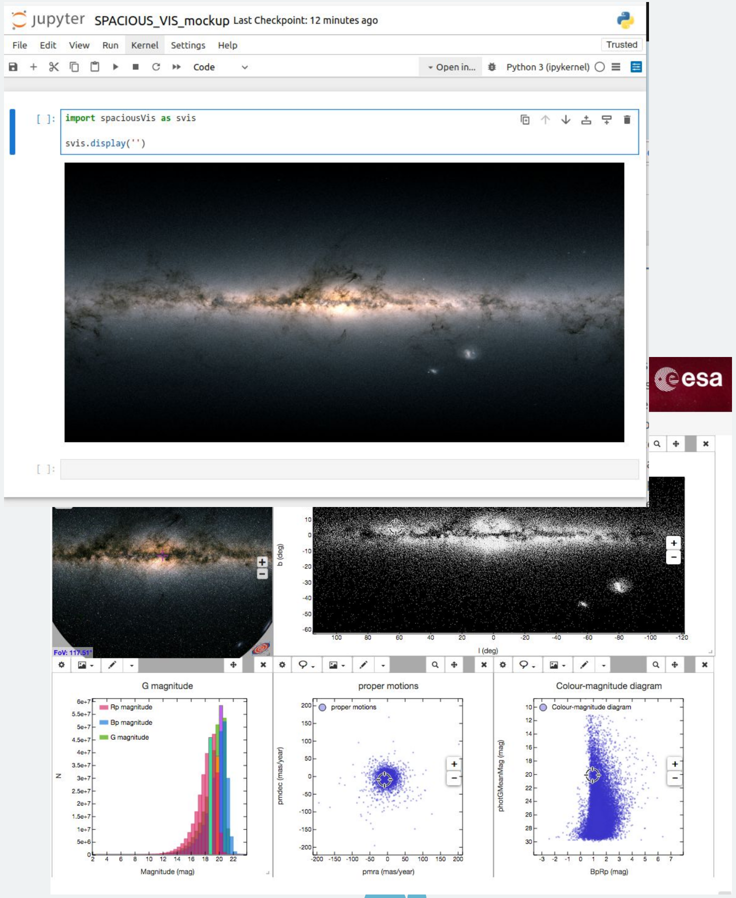


Right: HiPS level 7 (~4”) - resolution with SPACIOUS

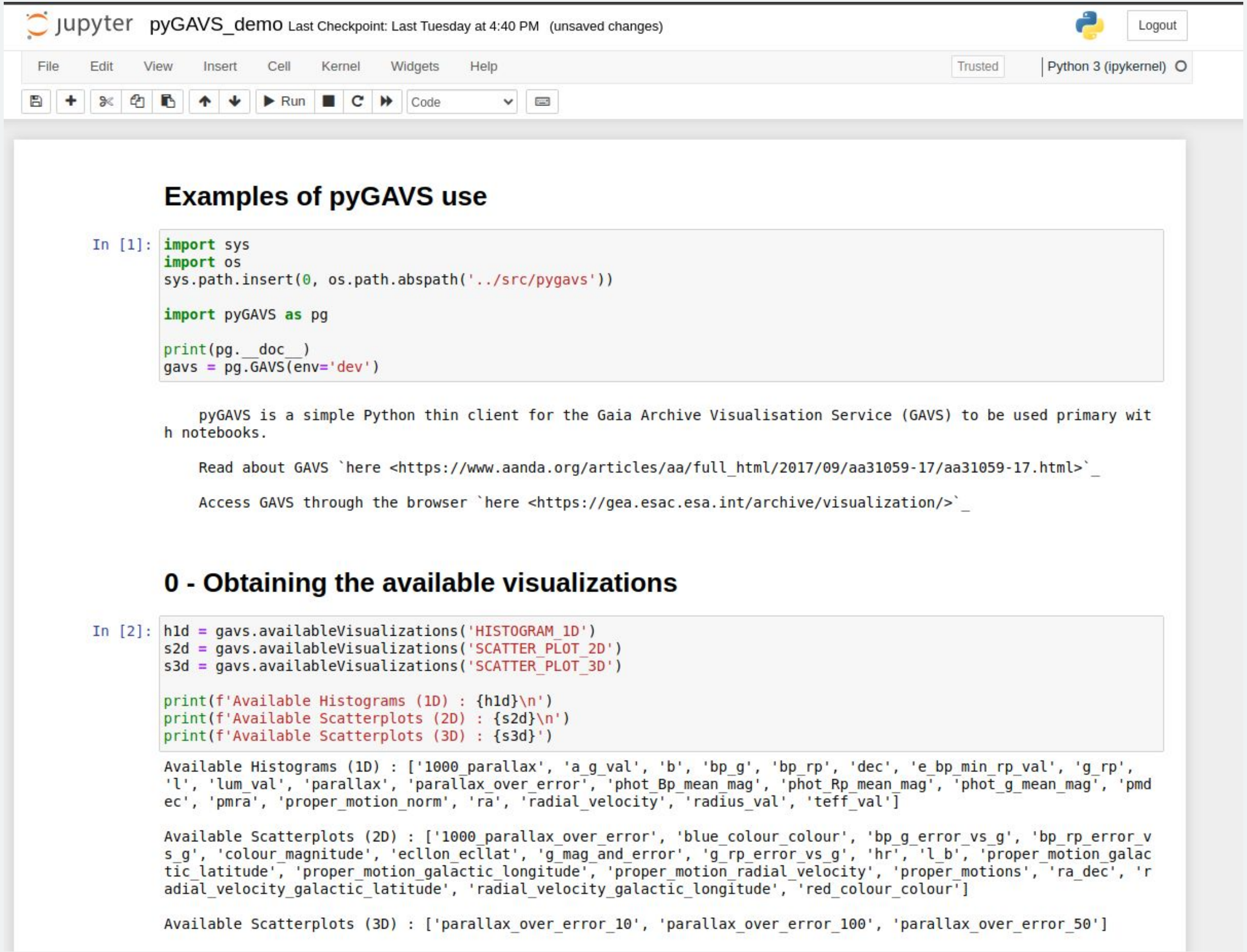
SPACIOUS

GAVS @ SPACIOUS

- The web page (webapp) is “just” an example of a visualisation client
- **SPACIOUS provides**
  - Python client wrapping the REST API
  - Total visualisation and analysis flexibility
  - Tighter integration with archives (ESA, SPACIOUS cloud, etc): visualise queries, visually generated queries
  - Reproducibility, collaborative - sharing workspaces



```
[ ]: import spaciousVis as svis

     svis.display('')
```



Examples of pyGAVS use

In [1]:

```
import sys
import os
sys.path.insert(0, os.path.abspath('../src/pygavs'))

import pyGAVS as pg

print(pg.__doc__)
gavs = pg.GAVS(env='dev')
```

```
    pyGAVS is a simple Python thin client for the Gaia Archive Visualisation Service (GAVS) to be used primary wit
h notebooks.

    Read about GAVS `here <https://www.aanda.org/articles/aa/full_html/2017/09/aa31059-17/aa31059-17.html>`_

    Access GAVS through the browser `here <https://gea.esac.esa.int/archive/visualization/>`_
```

0 - Obtaining the available visualizations

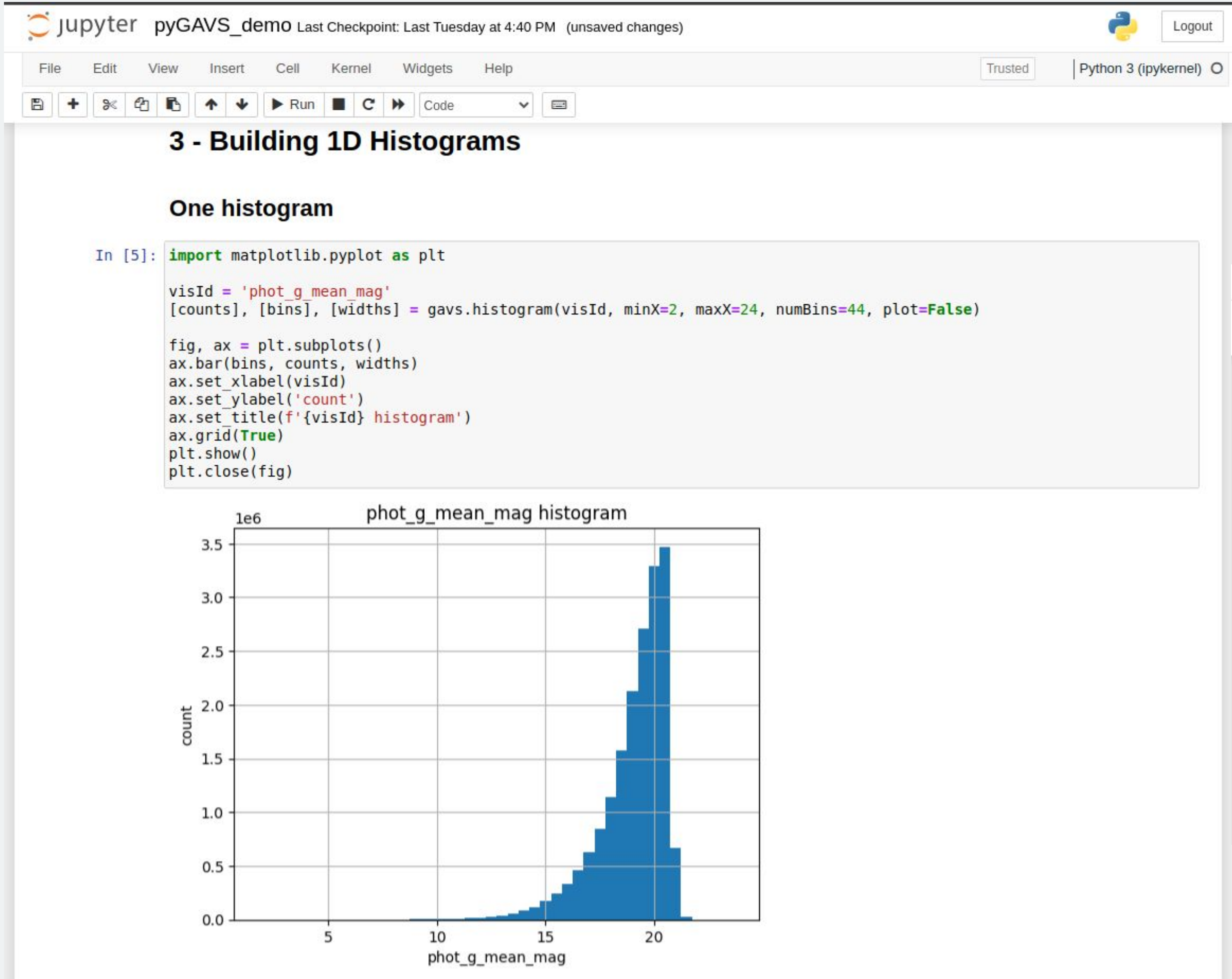
In [2]:

```
h1d = gavs.availableVisualizations('HISTOGRAM_1D')
s2d = gavs.availableVisualizations('SCATTER_PLOT_2D')
s3d = gavs.availableVisualizations('SCATTER_PLOT_3D')

print(f'Available Histograms (1D) : {h1d}\n')
print(f'Available Scatterplots (2D) : {s2d}\n')
print(f'Available Scatterplots (3D) : {s3d}')
```

```
Available Histograms (1D) : ['1000_parallax', 'a_g_val', 'b', 'bp_g', 'bp_rp', 'dec', 'e_bp_min_rp_val', 'g_rp',
'l', 'lum_val', 'parallax', 'parallax_over_error', 'phot_Bp_mean_mag', 'phot_Rp_mean_mag', 'phot_g_mean_mag', 'pmd
ec', 'pmra', 'proper_motion_norm', 'ra', 'radial_velocity', 'radius_val', 'teff_val']

Available Scatterplots (2D) : ['1000_parallax_over_error', 'blue_colour_colour', 'bp_g_error_vs_g', 'bp_rp_error_v
s_g', 'colour_magnitude', 'ecllon_ecllat', 'g_mag_and_error', 'g_rp_error_vs_g', 'hr', 'l_b', 'proper_motion_galac
tic_latitude', 'proper_motion_galactic_longitude', 'proper_motion_radial_velocity', 'proper_motions', 'ra_dec', 'r
adial_velocity_galactic_latitude', 'radial_velocity_galactic_longitude', 'red_colour_colour']

Available Scatterplots (3D) : ['parallax_over_error_10', 'parallax_over_error_100', 'parallax_over_error_50']
```

- Connects to any service running a vserver (GAVS, SPACIOUS BDAF, ...)
- Tutorials provided

SPACIOUS

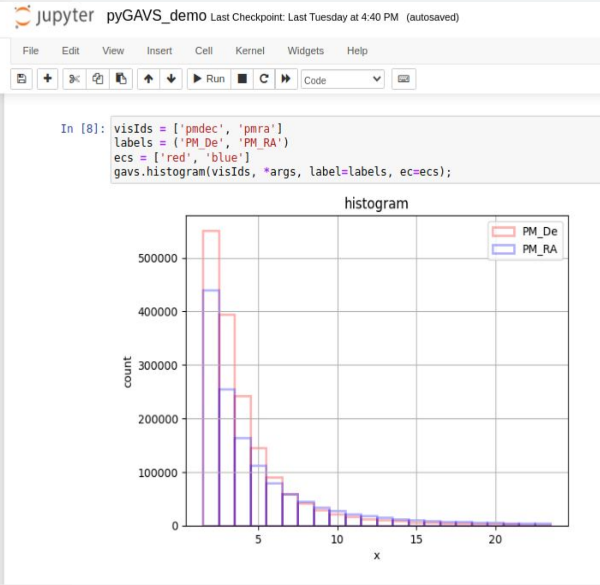

jupyter pyGAVS_demo Last Checkpoint: Last Tuesday at 4:40 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

3 - Building 1D Histograms

One histogram

In [5]:
```
import matplotlib.pyplot as plt

visId = 'phot_g_mean_mag'
[counts], [bins], [widths] = gavs.histogram(visId, minX=2, maxX=24, numBins=44, plot=False)

fig, ax = plt.subplots()
ax.bar(bins, counts, widths)
ax.set_xlabel(visId)
ax.set_ylabel('count')
ax.set_title(f'{visId} histogram')
ax.grid(True)
plt.show()
plt.close(fig)
```

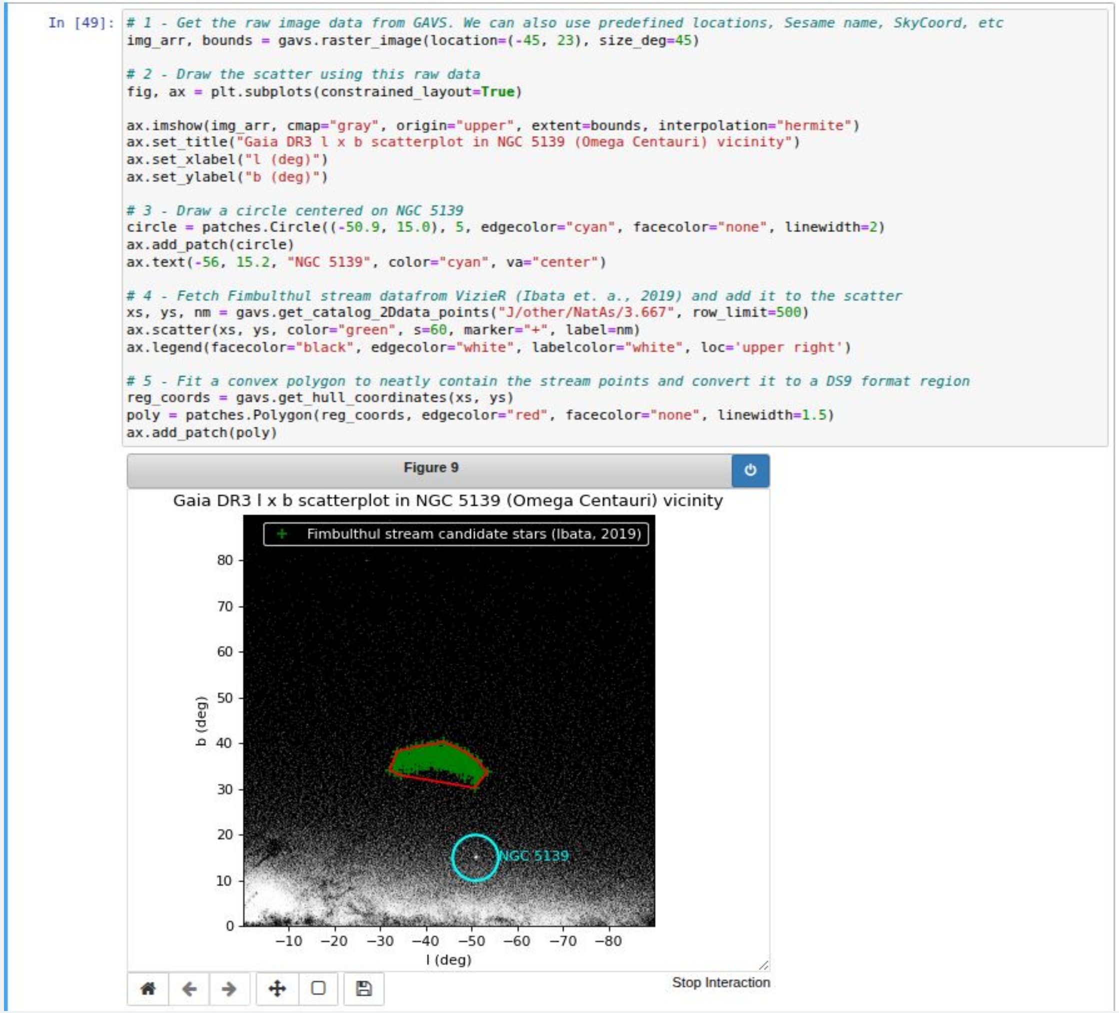
phot_g_mean_mag histogram
1e6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
count
5
10
15
20
phot_g_mean_mag


jupyter pyGAVS_demo Last Checkpoint: Last Tuesday at 4:40 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run
Code

In [8]:
```
visIds = ['pmdec', 'pmra']
labels = ('PM_De', 'PM_RA')
ecs = ['red', 'blue']
gavs.histogram(visIds, *args, label=labels, ec=ecs);
```

histogram
PM_De
PM_RA
500000
400000
300000
200000
100000
0
count
5
10
15
20
x

Displaying and annotating a visualisation with GAVS with matplotlib



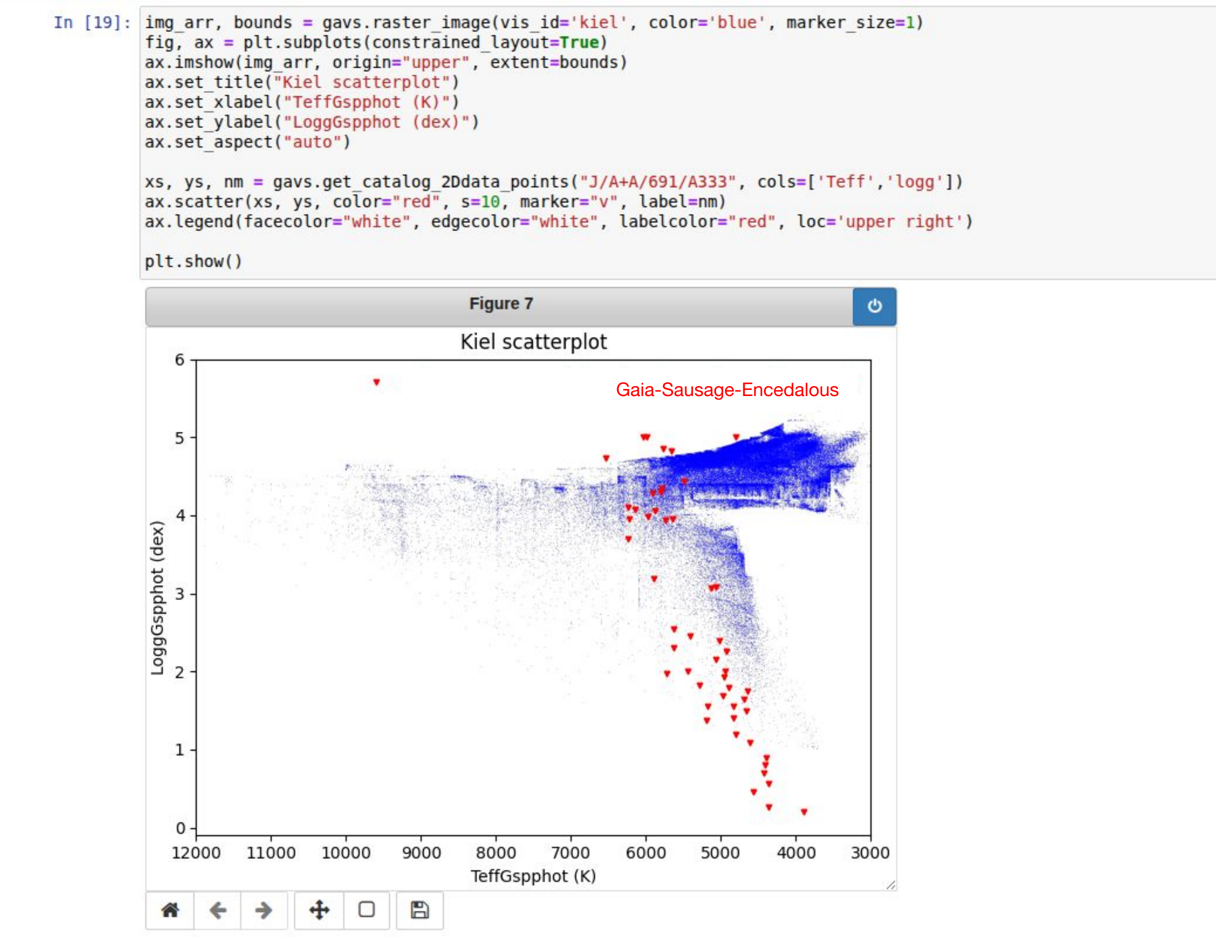
In [49]:

```
# 1 - Get the raw image data from GAVS. We can also use predefined locations, Sesame name, SkyCoord, etc
img_arr, bounds = gavs.raster_image(location=(-45, 23), size_deg=45)

# 2 - Draw the scatter using this raw data
fig, ax = plt.subplots(constrained_layout=True)

ax.imshow(img_arr, cmap="gray", origin="upper", extent=bounds, interpolation="hermite")
ax.set_title("Gaia DR3 l x b scatterplot in NGC 5139 (Omega Centauri) vicinity")
ax.set_xlabel("l (deg)")
ax.set_ylabel("b (deg)")

# 3 - Draw a circle centered on NGC 5139
circle = patches.Circle((-50.9, 15.0), 5, edgecolor="cyan", facecolor="none", linewidth=2)
ax.add_patch(circle)
ax.text(-56, 15.2, "NGC 5139", color="cyan", va="center")

# 4 - Fetch Fimbulthul stream datafrom VizieR (Ibata et. a., 2019) and add it to the scatter
xs, ys, nm = gavs.get_catalog_2Ddata_points("J/other/NatAs/3.667", row_limit=500)
ax.scatter(xs, ys, color="green", s=60, marker="+", label=nm)
ax.legend(facecolor="black", edgecolor="white", labelcolor="white", loc='upper right')

# 5 - Fit a convex polygon to neatly contain the stream points and convert it to a DS9 format region
reg_coords = gavs.get_hull_coordinates(xs, ys)
poly = patches.Polygon(reg_coords, edgecolor="red", facecolor="none", linewidth=1.5)
ax.add_patch(poly)
```

Figure 9
Gaia DR3 l x b scatterplot in NGC 5139 (Omega Centauri) vicinity
Fimbulthul stream candidate stars (Ibata, 2019)
80
70
60
50
40
30
20
10
0
b (deg)
NGC 5139
−10
−20
−30
−40
−50
−60
−70
−80
l (deg)
Stop Interaction

Works with

- Matplotlib
- Numpy and pandas
- Astroquery
- Astropy SkyCoords, units, etc
- Sesame name resolver
- ESASky
- ...



In [19]:
```python
img_arr, bounds = gavs.raster_image(vis_id='kiel', color='blue', marker_size=1)
fig, ax = plt.subplots(constrained_layout=True)
ax.imshow(img_arr, origin="upper", extent=bounds)
ax.set_title("Kiel scatterplot")
ax.set_xlabel("TeffGspphot (K)")
ax.set_ylabel("LoggGspphot (dex)")
ax.set_aspect("auto")

xs, ys, nm = gavs.get_catalog_2Ddata_points("J/A+A/691/A333", cols=['Teff','logg'])
ax.scatter(xs, ys, color="red", s=10, marker="v", label=nm)
ax.legend(facecolor="white", edgecolor="white", labelcolor="red", loc='upper right')

plt.show()
```

Works with

- Matplotlib
- Numpy and pandas
- Astroquery
- Astropy SkyCoords, units, etc
- Sesame name resolver
- ESASky
- …

SPACIOUS