The HDF5/HTM large catalog format, data/catsHTM directory and access functions in the MATLAB Astronomy & Astrophysics Toolbox

Description

The HDF5/HTM format designed to store and provide fast access for large astronomical catalogs with >10^6 rows. The format is based on the HDF5 file format and HDF5 file access utilities which are available on many platforms. The catalog format is designed to allow fast access for cone searches in the range of 1 arcsec to about 1 deg. For fast access, the sources are sorted into hierarchical triangular mesh (HTM).

Several tools for creating and accessing the HDF5/HTM files are provided. Using the matlab tools, the typical search and retrival time for a 10 arcsec cone search is about 2-4 milliseconds (almost independent of the number of sources in the catalog).

Python version

Python version is available from github.

The Python version has only a cone search capabilities.

Available catalogs

Currently, the following catalogs are available in this format (alphabetical order):

1. 2MASS -
2. 2MASSxsc -
3. AKARI
4. APASS - (~5.5x10^7 sources)
5. Cosmos
6. DECaLS - DECaLS/DR5
7. FIRST - (~9.5x10^5 sources)
8. GAIADR1 - GAIA/DR1 (~1.1x10^9 sources).
9. GALEX - GALAEX/GR6Plus7 (~1.7x10^8 sources).
10. HSC (not yet available)
11. NVSS - (~1.8x10^6 sources)
12. PS1 - Pan-STARRS (~2.6x10^9 sources; A cleaned version of the PS1 stack catalog)
13. ROSATfsc -
14. SDSSDR10 - SDSS/DR10 -
15. SpecSDSS/DR14
16. UCAC4 (~1.1x10^8 sources)
17. UKIDSS/DR10
18. USNOB1 (not yet available)
19. VISTA/Viking/DR3
20. VST/ATLAS/DR3
21. VST/KiDS/DR3
22. WISE (~5.6x10^8 sources)
23. XMM - 7.3x10^5 sources) 3XMM-DR7 (Rosen et al. 2016; A&A 26, 590)

Credit

See catsHTM credit page.

License

Unless specified otherwise this code and products are released under the GNU general public license version 3.

Installation

See http://weizmann.ac.il/home/eofek/matlab/doc/install.html for installation instruction and additional documentation.

The catsHTM directory is very large and therefore available on request. After copying the data you need to add the directories path to your startup.m file.

The HDF5/HTM catalogs requires about 1.5TB of disk space.

Structure and methodlogy

In order to allow fast access, we first divide the celestial sphere into a grid of equal area hierarchical triangular mesh (HTM). The level of the HTM (i.e., the smallest HTM size) is typically between 6 and 9, depanding on the catalog size.

Next, groups of typically 100 HTMs are stored in a single HDF5 file, where each HTM is stored in its own dataset. The dataset names are "htm_%06d", where %06d is the format of the HTM index we use here. These indices are generated by the `celestial.htm.htm_build` function. In each HTM the sources are sorted by declination.

For each HTM dataset, the HDF file contain additional dataset named "htm_%06d_Ind". This dataset is an index file of [LineIndex Declination] in steps of about 30 to 100 lines. This maybe use in order to upload smaller subsets of the HTM sources.

The full HTM hirarchy from level 0 to the final level (corresponds to the HTM datasets) are stored in an HDF file name "<CatName>_htm.hdf5", in dataset named "<CatName>_HTM". The dataset contains a table of 13 columns (stored in single precision format). The columns are:

The line number in the table corresponds to the HTM index.

1. Level - The HTM level (first level is 0).
2. Father - The HTM index (line number) of the HTM father.
3. Son1 - The HTM index of the 1st son of the HTM. The sons of the final HTM (in which data is actually stored) are NaNs.
4. Son2 - The HTM index of the 2nd son of the HTM.
5. Son3 - The HTM index of the 3rd son of the HTM.
6. Son4 - The HTM index of the 4th son of the HTM.
7. Pole1 Long - Longitude (J2000 R.A.) of the pole of the first arc of the HTM.
8. Pole1 Lat - Latitude (J2000 Dec.) of the pole of the first arc of the HTM.
9. Pole2 Long - Longitude (J2000 R.A.) of the pole of the 2nd arc of the HTM.
10. Pole2 Lat - Latitude (J2000 Dec.) of the pole of the 2nd arc of the HTM.
11. Pole3 Long - Longitude (J2000 R.A.) of the pole of the 3rd arc of the HTM.
12. Pole3 Lat - Latitude (J2000 Dec.) of the pole of the 3rd arc of the HTM.
13. Nsrc - Number of sources in HTM dataset. NaN or 0, if HTM dataset doesn't exist

Finally, the catalog column names and units are stored in a mat file named: "<CatName>_htmColCell.mat". The column names and units are stored in cell array named ColCell and ColUnits, respectively.

Example

The data/catsHTM/FIRST/ directory contains 153 files. The names of 151 of the files are <CatName>_htm_%06d.hdf5, where CatName is the catalog name (e.g., 'FIRST'). Each file contains up to 100 datasets, named "htm_%06d", that contain all the sources in a specific HTM of level 7. For each dataset, there is an index dataset named "htm_%06d_Ind". Example:

```
Info=h5info('GAIADR1_htm_021500.hdf5');
% There are 200 datasets (100 pairs of data + ind)
Info.Datasets
```

```
ans = 
  200×1 struct array with fields:

    Name
    Datatype
    Dataspace
    ChunkSize
    FillValue
    Filters
    Attributes
```

```
Info.Datasets(1)
```

```
ans = 
          Name: 'htm_021500'
      Datatype: [1×1 struct]
     Dataspace: [1×1 struct]
     ChunkSize: []
     FillValue: 0
       Filters: []
    Attributes: [2×1 struct]
```

```
Info.Datasets(2)
```

```
ans = 
          Name: 'htm_021500_Ind'
      Datatype: [1×1 struct]
     Dataspace: [1×1 struct]
     ChunkSize: []
     FillValue: 0
       Filters: []
    Attributes: []
```

```
ans = 
        2722           8

ans = 
        2722           8

ans = 
        2722           8

ans = 
        2722           8
```

Search the catalogs

Cone searches

The catalogs are optimized for cone searches. The basic low-level function for searching the catalogs is `catsHTM.cone_search`. The arguments of this function are: catalog name, J2000 R.A. in radians, J2000 Dec. in radians, search radius in arcsec, and optional pairs of keyword values.

```
[Cat,ColCell]=catsHTM.cone_search('APASS',1,1,100)
```

```
Cat =
      0.99972            1         0.711         0.518    2.0122e+07            2           10        15.358

ColCell =
    'RA'     'Dec'     'RAerr'     'Decerr'     'Name'     'Nobs'     'Mobs'     'V'     'BV'     'B'     'g'     '
```

```
[Cat,ColCell]=catsHTM.cone_search('GAIADR1',1,1,10,'OutType','astcat')
```

```
Cat =
  AstCat with properties:

            Cat: [0×8 double]
            Col: [1×1 struct]
        ColCell: {'RA'  'Dec'  'ErrRA'  'ErrDec'  'MagG'  'ErrMagG'  'ExcessNoise'  'ExcessNoiseSig'}
       ColUnits: []
       SortedBy: []
    SortedByCol: []
           Name: []
         Source: []
      Reference: []
        Version: []
         Header: {0×3 cell}
            WCS: []
       UserData: []

ColCell =
    'RA'     'Dec'     'ErrRA'     'ErrDec'     'MagG'     'ErrMagG'     'ExcessNoise'     'ExcessNoiseSig'
```

Additional high and low level functions to access these catalogs as well as other online and offline catalog is available in the VO package.

Specifically, the `VO.search.cat_cone` function can search for catalogs in the `catsHTM` format as well as all the catalogs in the `+cats` package.

Serial searches

You can perform serial search with that execute a function on the entire catalog and optionaly save the results. The following example, execute the command `sin` on the entire APASS catalog. Note that by default this will run on multiple (24) processors, but you can specify the number of parallel processors using the 'NparPool' keyword.

```
[ColCell]=catsHTM.serial_search('APASS',@sin)
```

Cross match the catalogs

You can cross match two catalogs and execute a function on the output. For example, the following command will cross match the entire APASS catalog with the 2MASS catalog:

```
catsHTM.xmatch_2cats('APASS','TMASS')
```

Reading specific HTM tiles

We can read one of these datasets - these can be done using several functions. First, using the built in MATLAB function:

```
Data=h5read('GAIADR1_htm_021500.hdf5','/htm_021502');
size(Data)
```

or using functions in te HDF5 static class:

```
Data=HDF5.load('GAIADR1_htm_021500.hdf5','htm_021502');
size(Data)
```

or using the functions in the catsHTM static class:

```
Data = catsHTM.load_cat('GAIADR1_htm_021500.hdf5','htm_021502');
size(Data)
```

However, you can also use the more convinient style:

```
Data = catsHTM.load_cat('GAIADR1',21502);
size(Data)
```

Additional functions

There are several low level functions for accessing and generating the catalogs, constructing the file and dataset names, and catalog statistics and display.

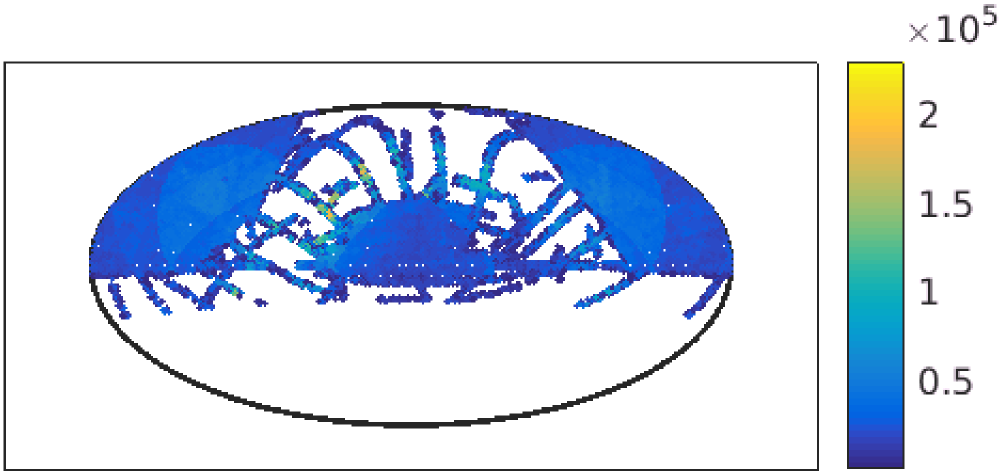
- `catsHTM.filename2base` - Convert HDF5/HTM file name to catalog name (file base name).
- `catsHTM.get_file_var_from_htmid` - Construct file and var name for HTM file stored in HDF5.
- `catsHTM.get_index_filename` - Get HDF5/HTM index file name and variable name from CatName.
- `catsHTM.save_cat` - save catalog data in HDF5 file.
- `catsHTM.save_htm_ind` - Save HTM indinces of the celestial sphere in an HDF5 file.
- `catsHTM.save_cat_colcell` - Save ColCell cell array of an HTM catalog
- `catsHTM.load_cat` - Load catalog stored in an HDF5 file.
- `catsHTM.load_multiple_cats` - Load HDF5/HTM catalog from multiple files/datasets
- `catsHTM.load_colcell` - Load ColCell and ColUnits for an HDF5/HTM catalog
- `catsHTM.load_htm_ind` - load HTM data into structure from an HDF5 file
- `catsHTM.search_htm_ind` - A coordinate cone search in an HTM stored in HDF5 file.

- `catsHTM.htm_search_cone` - Search for all HTM leafs interscting a small circle (cone search)
- `catsHTM.get_nsrc` - Count number of sources over all HTM in HDF5 files
- `catsHTM.nsrc` - Count sources in the HDF5/HTM index file

Plot functions

You can plot the sky surface density in a catalog using the `catsHTM.plot_density` command.

```
% plot the SDSS/DR10 source density per deg^2
% see help catsHTM.plot_density for additional options
catsHTM.plot_density('SDSSDR10');
```


$\times 10^5$
2
1.5
1
0.5