Hypercat User Manual

The HC team
Contact: robert.nikutta@gmail.com

June 27, 2021

Contents

1 Hypercat short introduction

1.1 Purpose

Hypercat is a Python software package developed to ease operating on N-dimensional hypercubes of model images, produced with Clumpy or other codes, while hiding the complexity of dealing with hundreds of gigabytes of data.

1.2 Attribution and software license

Hypercat is open-source software and freely available on GitHub and PyPI under a permissive BSD 3-clause license – in short, if you are using in your research any of the Hypercat software or its components, and/or the Hypercat data hypercubes, and/or telescope pupil images, please cite these two papers:

Nikutta, Lopez-Rodriguez, Ichikawa, Levenson, Packham, Hönig, Alonso-Herrero, "Hypercubes of AGN Tori (Hypercat) – I. Models and Image Morphology", ApJ (2021, accepted)

Nikutta, Lopez-Rodriguez, Ichikawa, Levenson, Packham, Hönig, Alonso-Herrero, "Hypercubes of AGN Tori (Hypercat) – II. Resolving the torus with Extremely Large Telescopes", ApJ (2021, under review)
[TODO: Add final URLs]

1.3 Code contributions are welcome

We welcome contributions to the Hypercat GitHub repository, be it bug fixes, documentation / examples / notebooks, or enhanced functionality.

Just fork the repository on GitHub, make edits on your fork, and when ready, make a pull request against the Hypercat master branch. Thank you!

2 Download, verification, and content of hypercube model files

2.1 Download

Hypercat needs to access the hypercubes of Clumpy images and dust maps. They can be downloaded as `hdf5` files (Hierarchical Data Format, version 5) from the link given at https://www.clumpy.org/images/ (which currently is ftp://ftp.noao.edu/pub/nikutta/hypercat/). Several `hdf5` files are available, each covering a different range of wavelengths:

Filename	Size (GB) compr. / raw	Wavelengths contained in the image hypercube (micron)
hypercat_20200830_all.hdf5.gz	271 / 913	All of the below
hypercat_20200830_nir.hdf5.gz	44 / 146	1.2, 2.2, 3.5, 4.8
hypercat_20200830_mir.hdf5.gz	120 / 402	8.7, 9.3, 9.8, 10, 10.3, 10.6, 11.3, 11.6, 12, 12.5, 18.5
hypercat_20200830_fir.hdf5.gz	65 / 219	31.5, 37.1, 53, 89, 154, 214
hypercat_20200830_submm.hdf5.gz	42 / 146	350, 460, 690, 945

Note that Hypercat interpolates smoothly between the sampled wavelengths in the hypercube. The `hypercat_20200830_all.hdf5.gz` file contains the image hypercube at all sampled wavelengths. This is the maximally compressed version of the hdf5 file, which must be uncompressed on the user's computer system.
To reduce the peak storage required on the target computer, both steps can be executed in one go (all commands in a single line):

```
lftp -e 'set net:timeout 10; cat /pub/nikutta/hypercat/hypercat_20200830_all.hdf5.gz; bye' ftp.noao.edu | gunzip > hypercat_20200830_all.hdf5
```

The program `lftp` must be installed on the target system, and 914 GB of space must be available on it (but only 271 GB of compressed data will be downloaded).

2.2 Verification

The reader should also download the checksums file ftp://ftp.noao.edu/pub/nikutta/hypercat/hypercat_20200830.md5 file and verify the hypercube file:

```
# this can take 30 minutes even on a modern computer
md5sum --ignore-missing -c hypercat_20200830.md5
hypercat_20200830_all.hdf5: OK

# or on MacOS and BSD variants
md5 hypercat_20200830_all.hdf5
#... and compare the printed hash with the one in the .md5 file
```

The download was successful if the computed checksum is identical to the one in the MD5 sums file. The same procedure applies to the other `*hdf5.gz` files analogously (which hold subsets of the wavelength sampling).

2.3 Content of the `hdf5` model files

The Hierarchical Data Format is a well-established format for packaging structured and heterogeneous datasets and meta-data, and the format is agnostic about the operating system.
We use `h5py`, a Python implementation, to pack our hypercubes, and the Hypercat software uses it to access said data. HDF implementations in other languages are readily available (e.g. the HDF5 module in IDL).

The structure of every Hypercat hdf5 file is simple. The hierarchy tree of the main Hypercat hdf5 file hypercat_20200830_all.hdf5 is shown below:

```
/Nhypercubes : 2
/hypercubenames : ['imgdata', 'clddata']
/imgdata/ : (group)
  /Nparam : 9
  /paramnames : ['sig','i','Y','N','q','tv','wave','x','y']
  /theta : [[15,30,45,60,75],[0,10,20,30,40,50,60,70,80,90],...]
  /hypercubeshape : [5,10,16,12,5,7,25,121,241]
  /hypercube : (9-d array)
/clddata/ : (group)
  /Nparam : 6
  /paramnames : ['sig','i','Y','q','x','y']
  /theta : [[15,30,45,60,75],[0,10,20,30,40,50,60,70,80,90],...]
  /hypercubeshape : [5,10,16,5,121,241]
  /hypercube : (6-d array)
```

There are Nhypercube = 2 hypercubes in the file. Their names are imgdata and clddata, each stored in a separate HDF group.

Group imgdata contains the thermal emission maps, stored in a 9-dimensional (σ, i, Y, $\mathcal{N}_0$, q, τ_V, λ, x, y) array /imgdata/hypercube. Several other small datasets of auxiliary information are also stored inside the group, e.g. the mapping of parameter values to the hypercube grid vertices (theta). The hypercube shape is given in hypercubeshape, and every array axis corresponds to one list in theta, and to one parameter name in paramnames. theta in the listing above shows the sampling of the parameters sig and i, which is the torus angular thickness σ, and of the viewing angle i. Other parameter samplings are truncated here for brevity.

The clddata group does the same for the 2D maps of cloud number along the LOS. These maps do not depend on wavelength and cloud optical depth τ_V, and $\mathcal{N}_0$ is just a multiplicative factor (i.e. each dust cloud map obtained must be muliplied by $\mathcal{N}_0$), thus this hypercube is 6-dimensional (σ, i, Y, q, x, y).

3 Installing Hypercat

Hypercat can be installed in several ways. Please select the route you may be most familar with.

3.1 Installing via pip

In most cases running `pip install` in your Python environment will be easiest:

```
pip install hypercat
```

Currently, HYPERCAT requires a specific version of the `h5py` package, so it might be a ood idea to install hypercat in a new virtual Python enviroment.
If you are familiar with using Anaconda to manage Python environments, the steps are:

```
# make a new environment called 'hypercat' with Python 3.7.2 installed (higher version are ok too)
conda create -n hypercat-env python=3.7.2

# activate this environment
conda activate hypercat-env

# install hypercat from pypi
pip install hypercat
```

Or maybe you are coordinating Python environments with pyenv. Then do:

```
# make a new environment called 'hypercat' with Python 3.7.2 installed (higher version are ok too)
pyenv install 3.7.2

# activate this environment
. .venv/bin/activate

# install hypercat from pypi
pip install hypercat
```

3.2 Clone directly from GitHub

You can also clone the Hypercat repository directly from Github, but you will have to make sure that all external Python modules are already installed on your computer:

```
cd foo/ # your target directory
git clone https://github.com/rnikutta/hypercat.git
cd hypercat/
```

4 Running Hypercat

The full functionality of HYPERCAT lies in its several Python modules. To use all of its capabilities, you can import `hypercat` into your Python workflow, including in Jupyter notebooks, and use the various functions. The examples below show how to import HYPERCAT into a Python workflow after the initial installation.

For the most common task – obtaining an image at given model parameters and wavelength – the user may opt to launch the [[HYPERCAT GUI Program][Hypercat GUI; it is very easy to use.

4.1 If you installed Hypercat via pip

Just start e.g. Ipython, and import Hypercat:

```
ipython --pylab
Python 3.8.10 (default, May 19 2021, 18:05:58)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.24.0 -- An enhanced Interactive Python. Type '?' for help.
Using matplotlib backend: TkAgg

In [1]: import hypercat as hc
In [2]: hc.__version__
Out[2]: '0.1.5'
```

4.2 If you cloned Hypercat from Github

Change into the `hypercat/` directory, launch e.g. Ipython, and import Hypercat:

```
cd hypercat/hypercat
ipython --pylab

In [1]: import hypercat as hc
In [2]: hc.__version__
Out[2]: '0.1.5'
```

5 Hypercat GUI Program

The Hypercat graphical user interface (GUI) application provides a simple way to generate, view, and export model images. It may be most convenient for exploring the effect of model parameters on the resulting image. With the initial release of Hypercat the Clumpy model cube(s) are available, and the section demonstrates operation using these Clumpy models.

5.1 Launching Hypercat GUI

If you installed Hypercat using ‘`pip`‘ or similar, you should be able to simply start Hypercat GUI in the terminal:

```
hypercatgui
```

If you instead just cloned the Hypercat repository from GitHub, you can change to the directory subdirectory ‘`hypercat/hypercatgui/` inside where you cloned the code into, and run

```
python ./hypercatgui.py
```

The application should launch in either case, and if it is the first launch, it will ask you which Hypercat model cub to load. Navigate to the `.hdf5` file (e.g. `hypercat_20181031_all.hdf5`), and an image with default settings will appear.

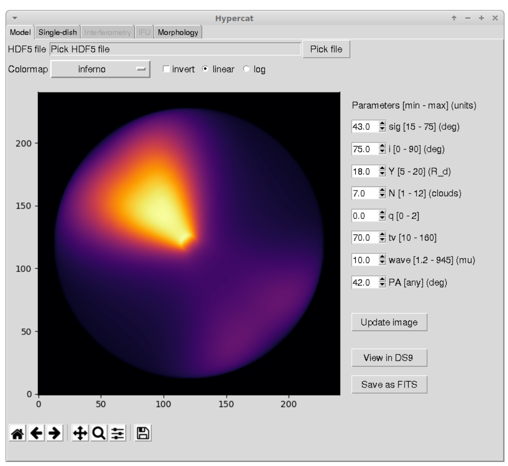

Figure 1: The Hypercat GUI displaying the emission map for a set of parameter values.

Currently the GUI application is quite simple. For a loaded model, a number of color maps and their scaling are selectable. The displayed image can be saved to a `.fits` file, and if SAO DS9 is installed, the model image can be sent to DS9. Figure 1 shows the GUI displaying one Clumpy emission map.

5.2 Changing model parameters

You can change all model parameter in the fields to the right of the displayed image, either by typing the new value (it overrides the currently displayed one), or by using the small up/down arrows next to each parameter (the finite stepping is some useful values for the given parameter).

Please note that for the `Y` parameter (the torus radial extent) we recommend that only integer values be used, as the image interpolation for fractional numbers will lead to have artifacts in the last `Y` through `Y-1` outer annulus.

When you have changed one parameter you can either hit `ENTER` or the `Update image` button to compute the new model, or change another parameters before updating. Note that the `TAB` keystroke brings you to the next parameter field. Upon updating the model a new image is computed from the hypercube through N-dimensional linear interpolation and displayed quickly.

Note that your most recent parameter selection is saved in a local configuration file. When you next time launch Hypercat GUI, the same model will be loaded again, using the values stored in the configuration file.

5.3 Save as FITS

If the currently displayed model is to your liking and you would like to export it to a FITS file, click on the `Save as FITS` button and decide where on your local disk to save the FITS file. The save file will have a header with the necessary information about the stored model, including Clumpy parameters, etc.

5.4 View in DS9

If the SAO DS9 software is installed on your system (found out my typing `which ds9` in the terminal), you can send the currently displayed model to DS9 directly from Hypercat GUI. Simply click on the `View in DS9` button.

You can open several models in the same DS9 (as new frames), but changing the displayed model, and then clicking on the `View in DS9` button again.

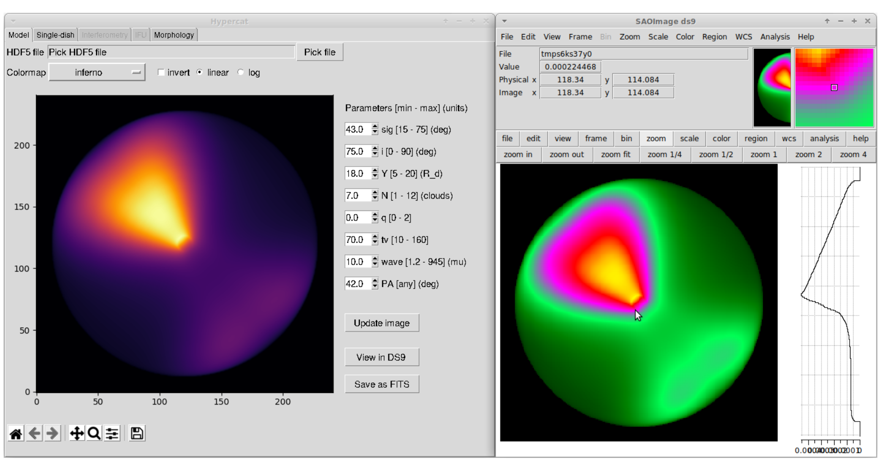

Figure 2: The Hypercat GUI and SAO DS9 displaying the same Clumpy model.

6 Using Hypercat

This section will be expanded in the near future. Until then, we refer the reader to the Jupyter notebook examples in the Hypercat GitHub repository: https://github.com/rnikutta/hypercat/examples/