juliacon

My JuliaCon proceeding

1st author[1], 2nd author[1, 2], and 3rd author[2]

[1]University
[2]National Lab

To study the cosmos, astronomers use data cubes with many dimensions representing images with axes for sky position, time, wavelength, polarization, and more. Since these large datasets often span many orders of magnitude in intensity and typically include colours invisible to humans, astronomers like to visualize their images using a variety of non-linear stretching and contrast adjustments. Additionally, images may contain metadata specifying arbitrary mappings of pixel positions to multiple celestial coordinate systems. Julia[1] is a powerful language for processing astronomical data, but these visualization tasks are a challenge for any tool.

Background. One of the most ubiquitous data formats used in astronomy is FITS, or the Flexible Image Transport System. Compared to a traditional raster image formats, FITS files are more like containers or small filesystems. Each file contains one or more header data units (HDUs) that pair a dataset with a header. The FITS format and header conventions have developed organically over several decades, first by convention, and then in a series of papers proposing new extensions and standardizing existing behaviour now summarized in [2].
In each HDU, headers are stored in a plain text ASCII format and are followed by data. The data may be in the format of an N-dimensional binary array (an "image" HDU), a binary table, or an ASCII table ("table HDUs").
The metadata described by FITS headers are quite rich. A header consists of `KEY = VALUE / COMMENT` entries, long form `COMMENT` sections, and long form `HISTORY` sections used to desrcibe the sequence of transformations used to generate the data. FITS headers often contain several hundred header entries most of which are specific to the instrument and software packages used to record and process the data. A subset of these are standardized and used to describe the physical units and coordinates of pixels in the image, called world coordinate system (WCS) header entries.
These WCS entries record pixel locations in one or more celestial coordinate systems including, right ascension and declination, galactic coordinates, velocity, frequency, wavelength, polarization, and more. For most coordinate schemes, pixel coordinates are encoded using an affine transformation matrix combined with a `CTYPE` specifier. Images are typically sampled regularly in the plane of a detector; however, coordinate projections are in general non-linear, and the physical coordinates of each pixel, spacings between pixels, etc. may vary across the image. In many instances, image HDUs contain 3, 4, or even 5 dimensional data cubes, and thanks to the affine transformation matrix, moving along any one of these dimensions can shift the coordinates along the other dimesions. This point is important since it means that the full coordinates of a given pixel are needed to calculate it's physical position along any one axis.

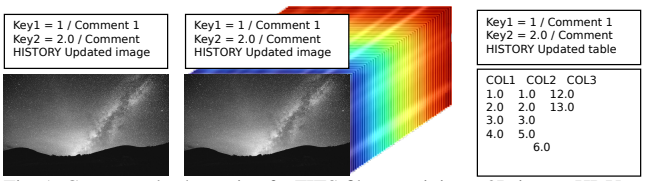

Fig. 1: Conceptual schematic of a FITS file containing a 2D image HDU, a 3D hyperspectral image HDU, and a table HDU.

For image data itself, FITS files do not in general store colour information that can be displayed directly. In contrast to digital cameras, astronomical data is almost invariably captured outside the human visible range or at least with filters that are not well matched to human colour perception. Instead, astronomers use visualizations that maps raw numerical data to a false colour image. These steps include a linear transformation from arbitrary numerical ranges to a standard 0 to 255 display range that produces a desired contrast and brightness at an intensity level of interest. The comparatively low dynamic range of electronic displays is often compensated for by applying non-linear mapping, or "stretching" such as log scales or arcsinh scales. Once the data is normalized to a reasonable intensity level, one typically presents a grayscale image or applies a false colour map. Finally, one may also create a colour composite image combining multiple information layers, especially when preparing data for public consumption.

Existing Tools. Julia has a robust package ecosystem for loading FITS files (`FITSIO.jl`), manupulating images (`Images.jl` ecosystem), calculating world coordinates (`WCS.jl`), and plotting (e.g. `Plots.jl`). For instance, the `FITSIO` package is capable of reading and writing FITS files, headers, image HDUs, and table HDUs; the `Images` packages support generating and displaying RGB images; and the the `WCS` package supports converting between pixel and world coordinates given a FITS header. The Julia package ecosystem lacks, however, a package that combines these tools into a high-level interface for loading and visualizing astronomical image data.

The AstroImages.jl package. The aim of `AstroImages.jl` is to tie together relevant utility packages from the Julia ecosystem to facilitate the loading, manipulation, analysis, and visualization of astronomical data.
`AstroImages.jl` is registered with `FileIO.jl` so that images and tables can be loaded from FITS files with the `load` and `save` functions.

```
using AstroImages
img = load("jwst-carina.fits")
```

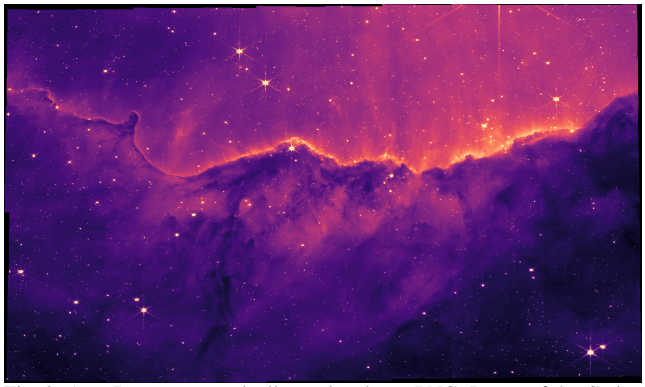

Fig. 2: AstroImage automatically rendered as a PNG. Image of the Carina nebula from JWST commisioning data (NASA, ESA, and CSA).

By default, this will load the first image HDU from the file. If a different HDU or a table HDU is desired, the HDU number can be passed explicitly as a second argument. Multiple HDUs can be loaded at once by passing a tuple of HDU numbers or `:`.
The values returned from `load` are `AstroImage` objects which combine a parent array, a FITS header, and implements the `DimensionalData` `AbstractDimArray` interface. Any array type can be wrapped in an `AstroImage` by passing it to the contructor.
By implementing the `AbstractDimArray` interface, the dimensions of each axis of the array can be named and paired with coordinates to enable concise indexing of multi-dimensional cubes.
Once loaded, the user should be able to pass the AstroImage through any analysis code that accepts an AbstractArray. Indexing and slicing are tracked using `DimensionalData.jl` so that pixel locations in the originating parent data cube are recoverable.
FITS header keys can be accessed by indexing the image with a string key, the `Comment` type, and or the `History` type to read and write to the header. `WCS.jl` `Transformation` objects can be generated using the `wcs(img, wcsnum)` function and are cached unless/until the user modifies a relevant header field.
Finally, when the user wishes to display an `AstroImage`, they may use either `imview` to create a lazy mapping to RGB data or the `implot` Plots.jl recipe. `imview` accepts keyword arguments for controlling the image stretching, colour limits, colour scheme, bias, and contrast:

```
imview(img; clims=Percent(95), cmap=:magma,
    stretch=asinhstretch, contrast=1.2, bias=0.4)
```

`imview` is also called automatically with configurable default values whenever an `AstroImage` is displayed. In this case, the output is automatically downscaled to a reasonable size using `ImageTransformations.restrict`. For example, all that's needed to display a 2D image HDU from a FITS file is to type `load("filename.fits")` at an interactive prompt provided that prompt supports rich outputs.
The `implot` Plots.jl recipe supports additional functionality. It uses `imview` to first render the image and then uses Plots.jl to present an image series.
The plot recipe automatically applies WCS grid lines (which may be arbitrarily tilted and warped) and labels the axes with their coordinate types. It also creates a custom colour bar labelled with ticks at the correct, possibly non-linear spacing caused by chosen stretch. When applied to slices of multi-dimensional cubes, `implot` further titles the plot with the slice's location along the other axes.

```
using AstroImages, Plots
HIcube = load("HIdat.fits")
slice = HIcube[Z=228]
implot(slice; cmap=:turbo, stretch=asinh)
```

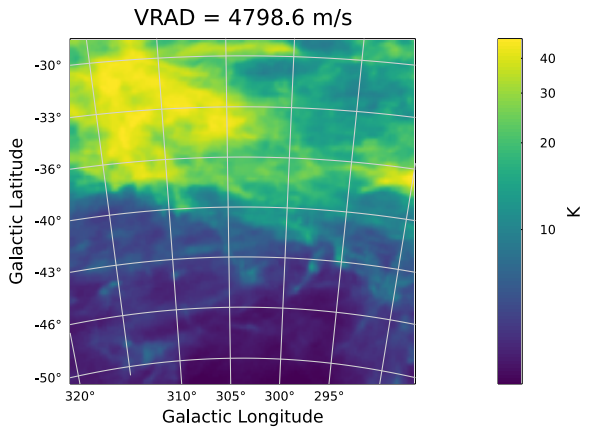

Fig. 3: implot augments an image with world coordinates and automatically labelled axes, titles, and colourbar. Image is a slice through galactic latitude, longitude, and velocity space of neutral Hydrogen density (HI4PI survey [3]).

`AstroImages.jl` is a complete, high-level package for working with astronomical data in Julia.

1. References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. doi:10.1137/141000671.

[2] FITS Working Group. Definition of the flexible image transport system (FITS). Technical report, International Astronomical Union, 2016.

[3] HI4PI Collaboration, N. Ben Bekhti, L. Flöer, R. Keller, J. Kerp, D. Lenz, B. Winkel, J. Bailin, M. R. Calabretta, L. Dedes, H. A. Ford, B. K. Gibson, U. Haud, S. Janowiecki, P. M. W. Kalberla, F. J. Lockman, N. M. McClure-Griffiths, T. Murphy, H. Nakanishi, D. J. Pisano, and L. Staveley-Smith. HI4PI: A full-sky H I survey based on EBHIS and GASS. , 594:A116, October 2016. doi:10.1051/0004-6361/201629178. 1610.06175.