Remote Sensing in Julia

an absolute beginners perspective

António Almeida

July 2022

preamble

YMMV

I've nothing much to offer
There's nothing much to take
I'm an absolute beginner
When I'm absolutely sane

David Bowie in *Absolute Beginners*, Virgin Records (1986)

why I am interested in Julia

- Lisp was my first love in programming language terms
- Julia takes a lot of cues from Lisp
- I also like numerical analysis
- Julia is a modern language
- Promising to bridge the scientific and non-scientific worlds

my day job

- I work for a company that is a marketplace for remote sensing data
- mostly Earth Observation data
- mostly optical, some radar
- very large images with many bands
- Python is my day job language

Julia envy

- while working with Python I have been following Julia
- what is promised and delivered makes it ideal
- e.g., native implementation for efficient handling of large arrays
- true to its scientific computing roots, focus on HPC
- HPC comes naturally for handling remote sensing images

passing judgment after a parachute drop

- hoping it will be useful for others in a similar quest

limited to optical satellite images

- GDAL looms large
- GDAL.jl
- ArchGDAL (more Julian)
- Rasters.jl: abstracting further than ArchGDAL
- GMT.jl: completely different approach

GDAL and its discontents

- GDAL supports an impressive list of formats and processing algorithms
- is mostly mantained by one person
- championed a lot of ideas (VRTs, GeoTIFF, COGs, etc.)
- but is a behemoth with quirks
- e.g., makes it hard to port a lot of existing Python to Web Assembly

uninformed opinions I

- Julia seems poised to ditch GDAL and do things differently
- performance wise C is not required
- build a more consistent interface for users
- leverage the JuliaImages ecosystem

uninformed opinions II

- performance from the start without the 2 languages problem
- modern package management system
- leverage the SciML ecosystem
- not dependent on the humours of big (kind?) corporations
- energetic & welcoming community

modest proposals

- expand RasterDataSources.jl to include other data sets (e.g. PlanetaryComputer, USGS, etc.)
- embrace the SpatioTemporal Asset Catalogs (STAC) specification
- provide easy solutions for high-performance image processing while escaping the spell of C/C++
- contribute decisively to bring geospatial processing to the web by routing around GDAL and the concomitant Emscripten issues

an happy-ending of sorts

- after properly disposing of my parachute I am eager to contribute as much as I can

starting the journey

- there is a long road ahead

fin

- slides and more https://github.com/perusio/juliacon-2022
- thank you