Advanced Framework for Learning Astrophysical Knowledge

木本真理究（Malik Olivier Boussejra）1・竹川俊也2・内木陆雄1・松林和也3・竹岛由里子4・植村诚5・藤代一成1

1慶應義塾大学 2国立天文台 3京都大学 4東京工科大学 5広島大学

http://aflak.jp
malik@boussejra.com
QUICK DEMO: https://vimeo.com/290328343
Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

https://www.rust-lang.org/
BLUEPRINTS – UNREAL ENGINE

経緯

- February 2018 – START

- VO standards are important
- aflak: モジュール可視化環境による等価幅マップの生成 日本天文学会 2019 年春季年会 [TBP]
WORKFLOW

Issue request

Query

File Path

Open file

Astronomical data repositories

Local filesystem

Load FITS dataset

Edit query and change loaded data

Execute

Visual program

Output(s)

Edit visual program
OVERVIEW

Node editor

Output window #1

Output node #1

Output node #2

Output window #2

http://aflak.jp
- New nodes can be added from a list of nodes
- Nodes can be wired together
- When an output node is created, a corresponding output window opens
- Export and import with *ad hoc* format

**Question**
- Any kind of VO standard for this kind of information?

[http://aflak.jp](http://aflak.jp)

This document is provided by JAXA.
SAVE OUTPUT AS FITS FILE CONTAINING END-TO-END LINEAGE OF HOW THE DATA WAS CREATED

Question

Any kind of “standard” procedure for recording data provenance?

Embed node graph into FITS header
CASE STUDY: EXTRACTION OF EQUIVALENT WIDTH

This document is provided by JAXA.
DOUBLE-FEEDBACK BETWEEN NODE EDITOR’S VARIABLES AND VISUALIZATION OUTPUT

Node A’s value is equal to the position of this vertical line.
SELECTING REGION OF INTEREST

↑ select ROI from pixel by pixel
QUERY DATA REPOSITORIES USING VO STANDARDS
USE CASES

• Analysis that requires gradual and manual fiddling with many parameters
  aflak aims at putting the astronomer in the loop
• Denoising, preprocessing?
WORKFLOW: ASTRONOMER IN THE LOOP
PRIMITIVES

● A set of algorithms applying transformations on datasets

● New nodes can be created
  - by combination of existing nodes (macro)
  - by direct implementation (currently Rust only, but C or Python solutions are explored)
FUTURE WORK

- https://github.com/aflak-vis/aflak/issues
- Batch processing
- Full macro support (Sit back! Writing the code as we speak, will be released very soon!)
- Full WCS support. Currently only partial and most probably buggy support is implemented.
- VO standards for communication with Aladin / Topcat / etc.
- Node primitives implementable in languages other than Rust (e.g. Python, C)
- Have more primitives included by default
ONE (TWO) – COMMAND INSTALL!

- Currently supported OSes:
  - Debian 9.X
  - Ubuntu 18.04
  - macOS
  - Windows

- Run on normal laptop. RAM requirements depend on the open datasets. 4GB or more is advised. A wide screen (2K or more) is better.

- Bug report / Feature requests / Comment / Anything

https://github.com/aflak-vis/aflak/issues/

$ curl https://sh.rustup.rs -sSf | sh
$ cargo install --git https://github.com/aflak-vis/aflak aflak

http://aflak.jp