

<http://aflak.jp>

aflak

Advanced Framework for Learning Astrophysical Knowledge

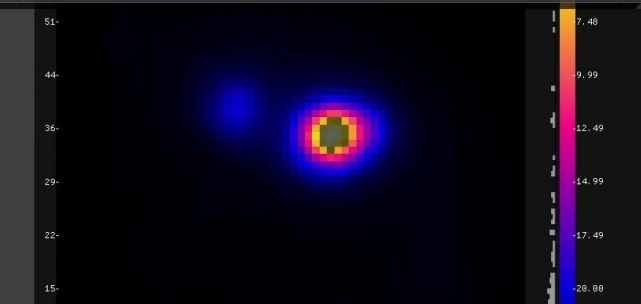
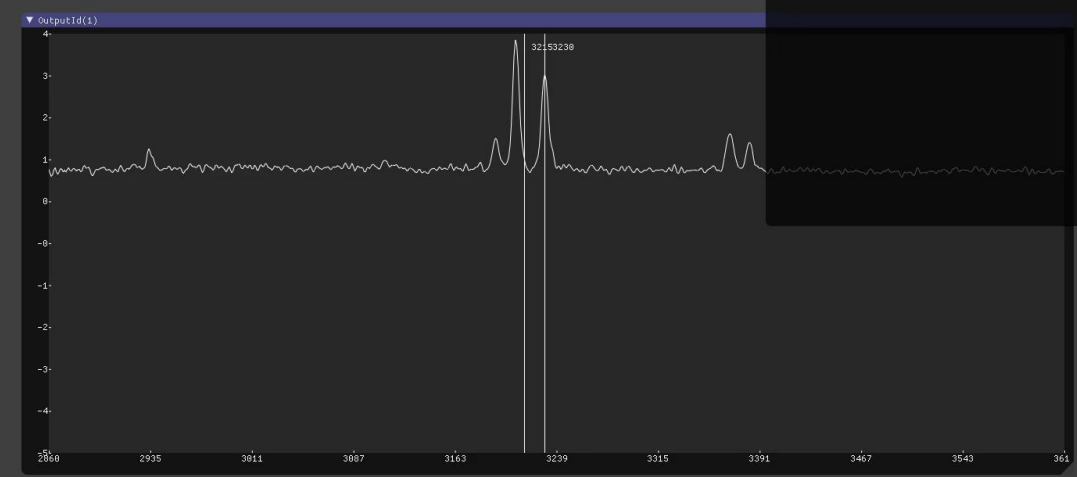
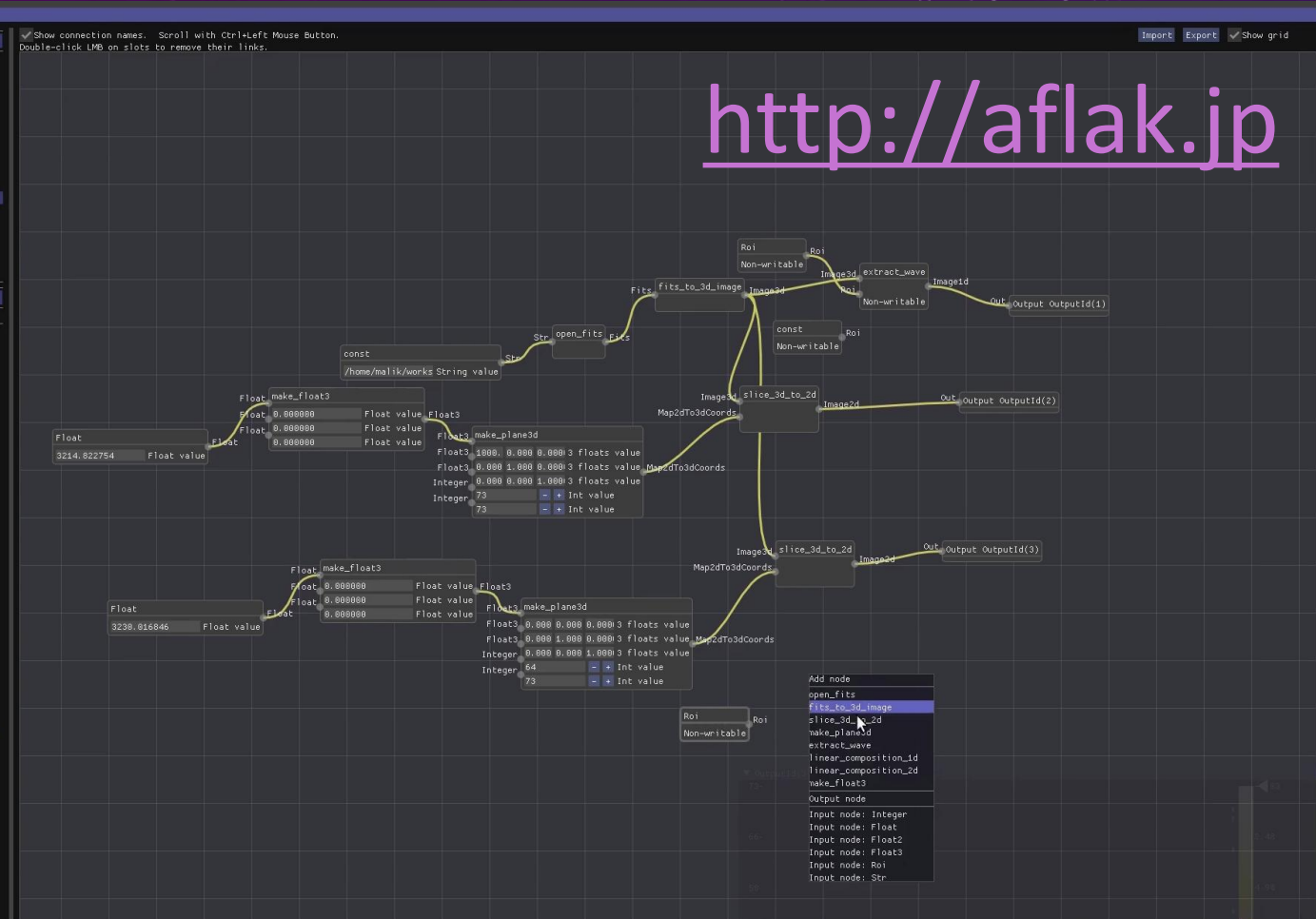
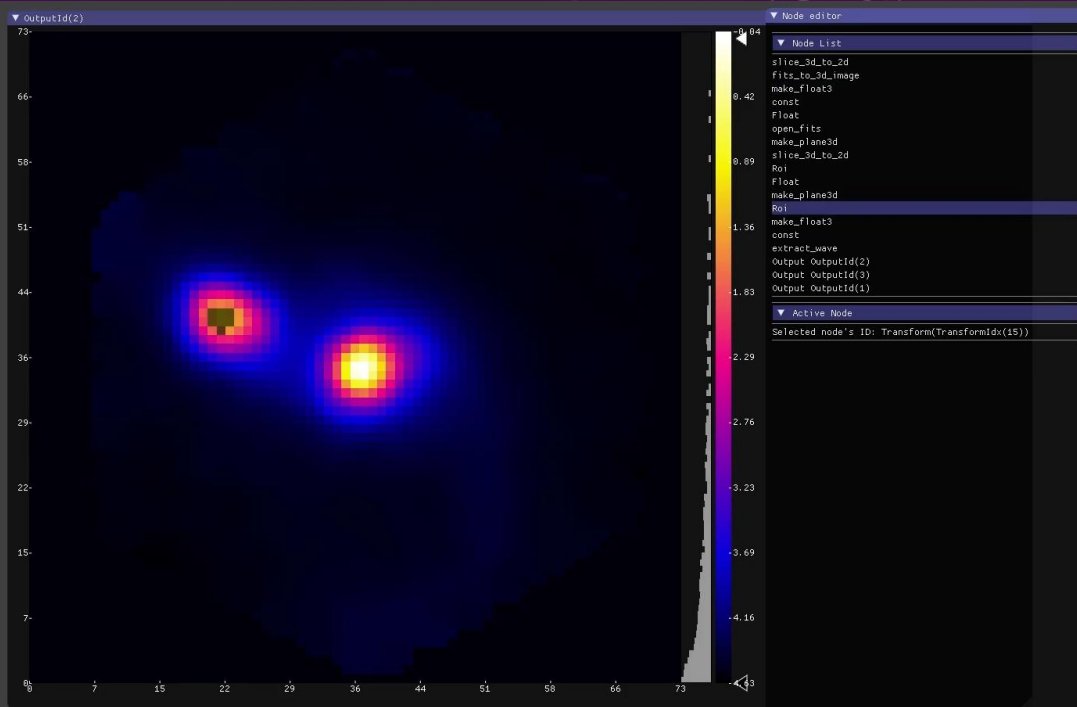
VISUAL PROGRAMMING ENVIRONMENT ESPECIALLY TUNED FOR
MULTI-SPECTRAL ASTROPHYSICAL OBSERVATIONS

Malik Olivier Boussejra,¹ Shunya Takekawa,² Rikuo Uchiki,¹ Kazuya Matsubayashi,³ Yuriko Takeshima,⁴
Makoto Uemura,⁵ and Issei Fujishiro¹

¹Keio University

malik@boussejra.com

²Nobeyama Radio Observatory ³Kyoto University ⁴Tokyo University of Technology ⁵Hiroshima University



QUICK DEMO: <https://vimeo.com/290328343>

RUST



[Documentation](#)

[Install](#)

[Community](#)

[Contribute](#)

Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

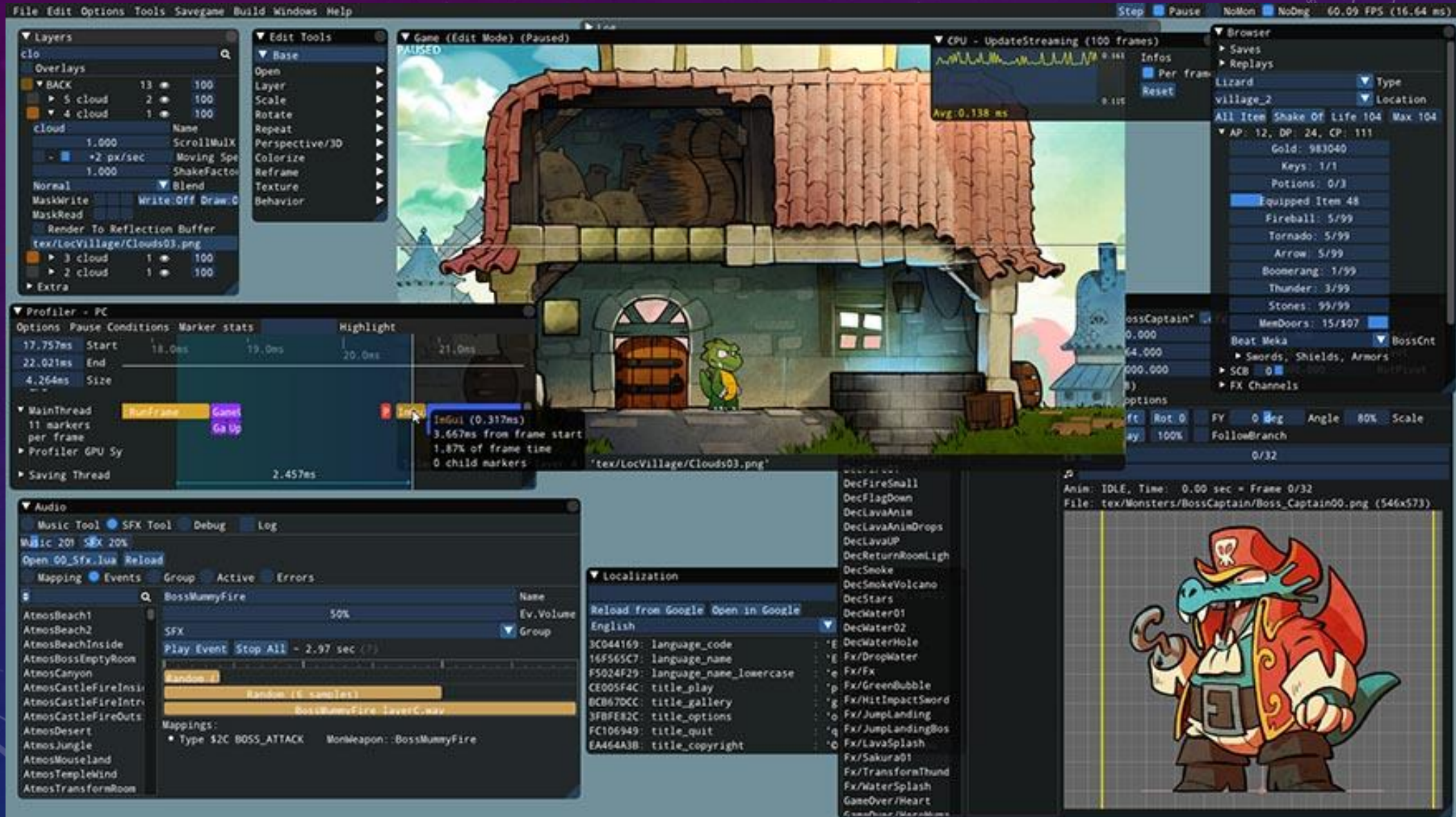
[Install Rust 1.30.1](#)

November 8, 2018

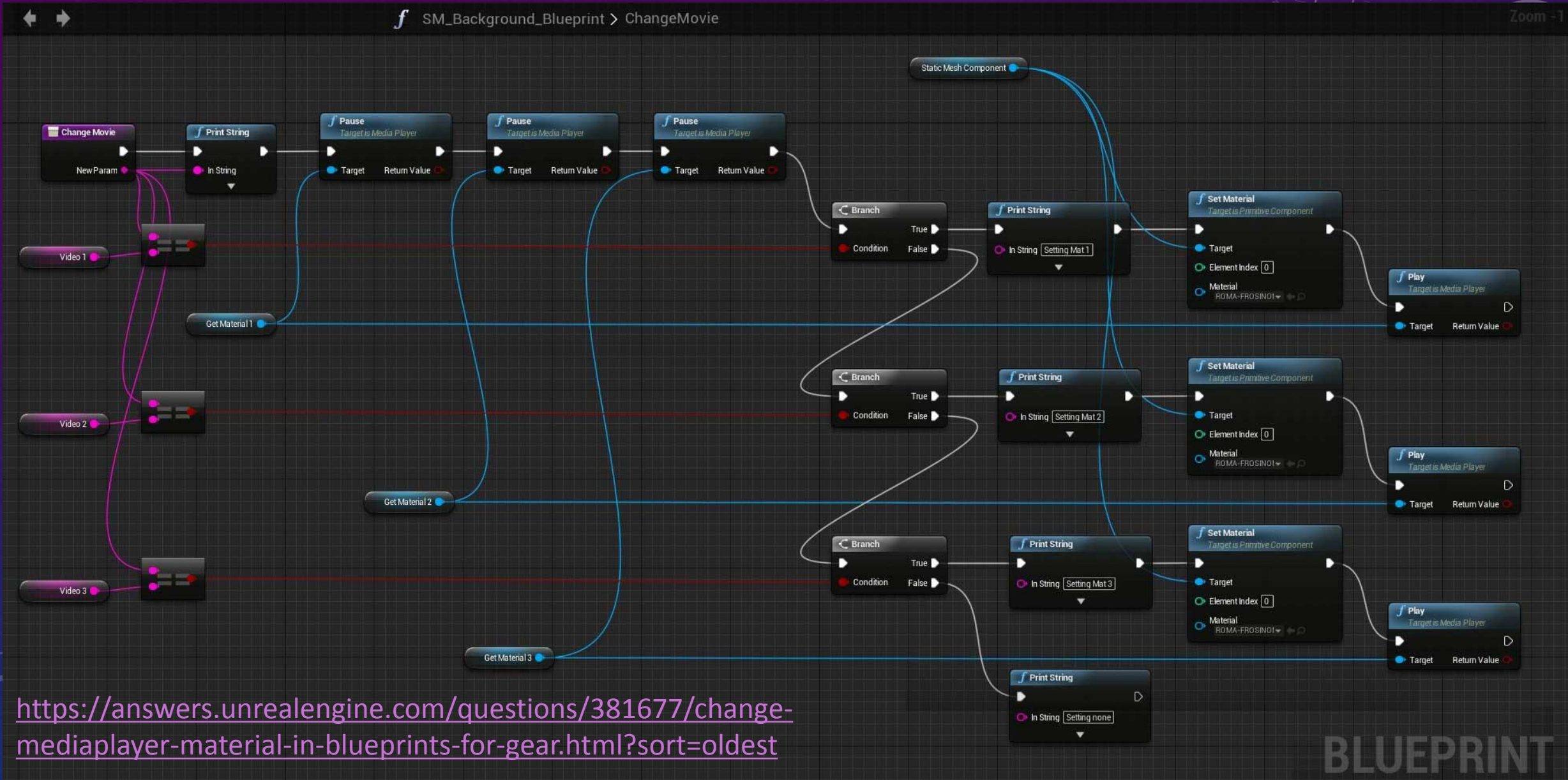
<https://www.rust-lang.org/>

DEAR IMGUI (C++)

<https://github.com/ocornut/imgui>



BLUEPRINTS – UNREAL ENGINE



<https://answers.unrealengine.com/questions/381677/change-mediaplayer-material-in-blueprints-for-gear.html?sort=oldest>

A MODULAR STRUCTURE

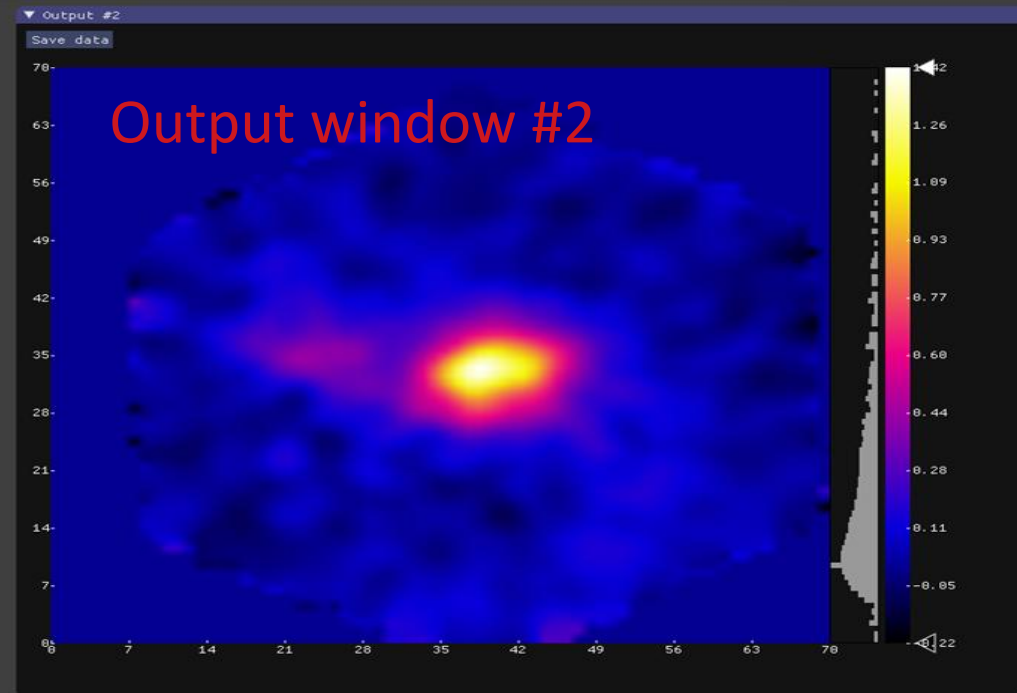
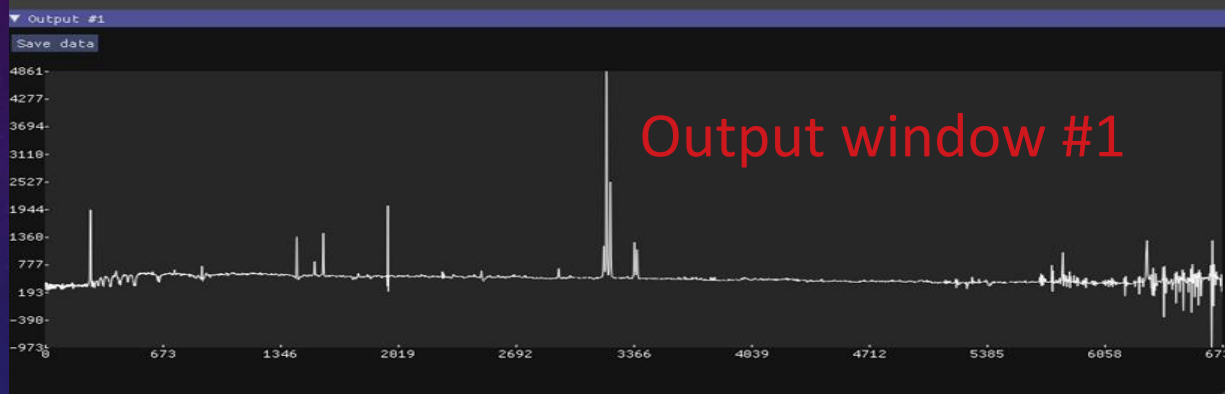
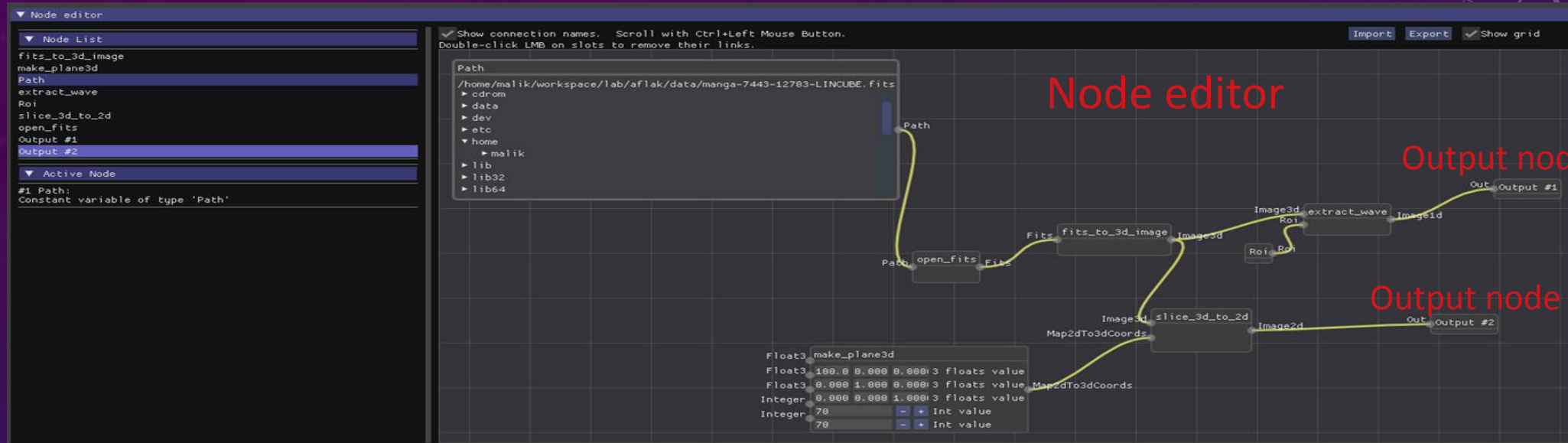
<http://aflak.jp>



Integrating together game technology for astronomy!

OVERVIEW

<http://aflak.jp>



NODE EDITOR

<http://aflak.jp>

The screenshot displays the Node Editor interface. On the left, there is a 'Node List' panel containing a list of nodes: fits_to_3d_image, make_plane3d, Path, extract_wave, Roi, slice_3d_to_2d, open_fits, Output #1, and Output #2. Below this is the 'Active Node' panel, which shows '#1 Path: Constant variable of type 'Path''. The main workspace contains a workflow graph with nodes connected by yellow lines. The nodes include 'Path', 'open_fits', 'fits_to_3d_image', 'Image3d', 'extract_wave', 'Image1d', 'Output #1', 'Image3d', 'Roi', 'slice_3d_to_2d', 'Image2d', 'Output #2', and 'Map2dTo3dCoords'. A 'make_plane3d' node is also visible with its parameters: Float3 (100.0, 0.000, 0.000), Float3 (0.000, 1.000, 0.000), Integer (0.000, 0.000, 1.000), and Integer (70).

Add node

open_fits
fits_to_3d_image
slice_3d_to_2d
make_plane3d
extract_wave
linear_composition_1d
linear_composition_2d
make_float3

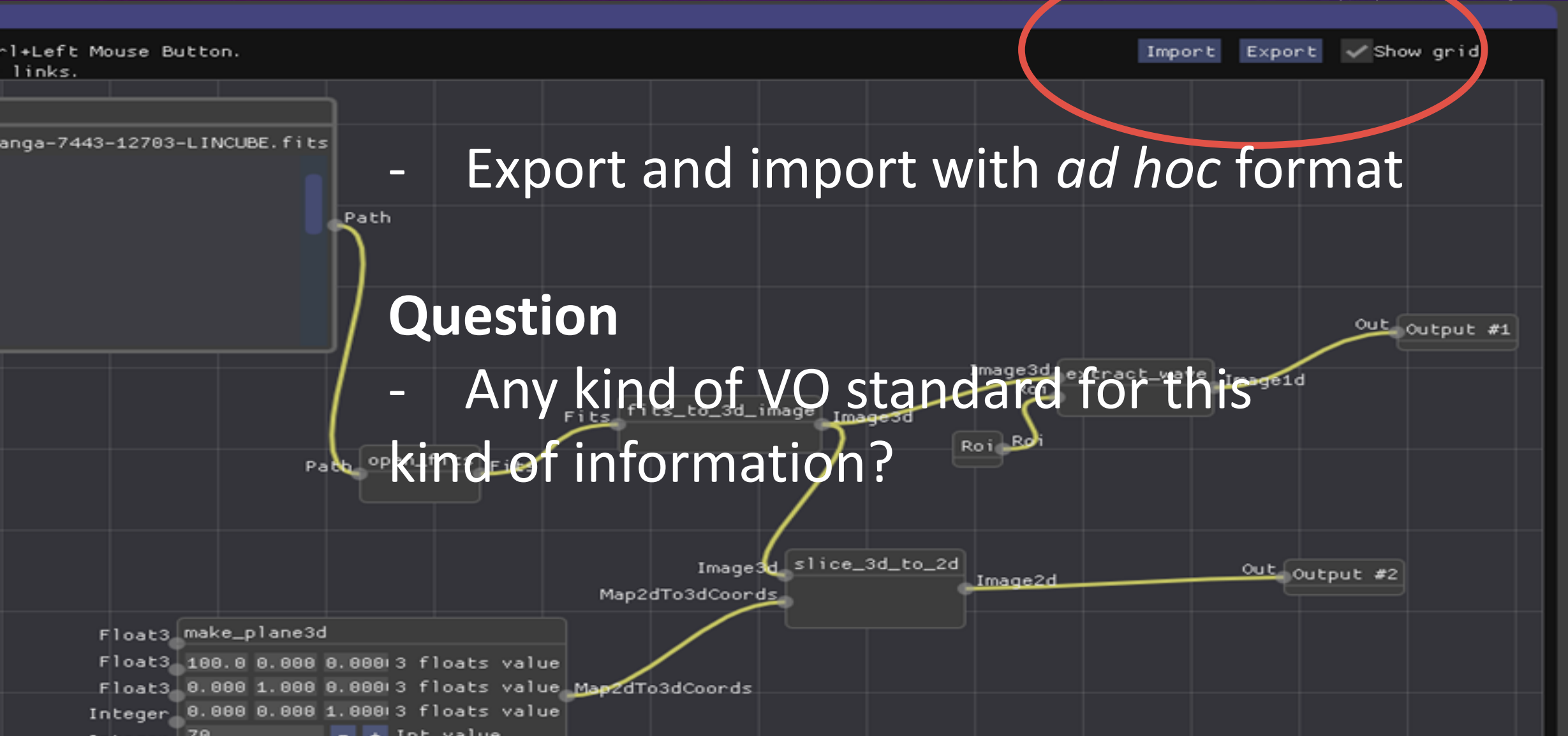
Output node

Input node: Integer
Input node: Float
Input node: Float2
Input node: Float3
Input node: Roi
Input node: Str
Input node: Path

- 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

IMPORT / EXPORT OF NODE PROGRAM

<http://aflak.jp>

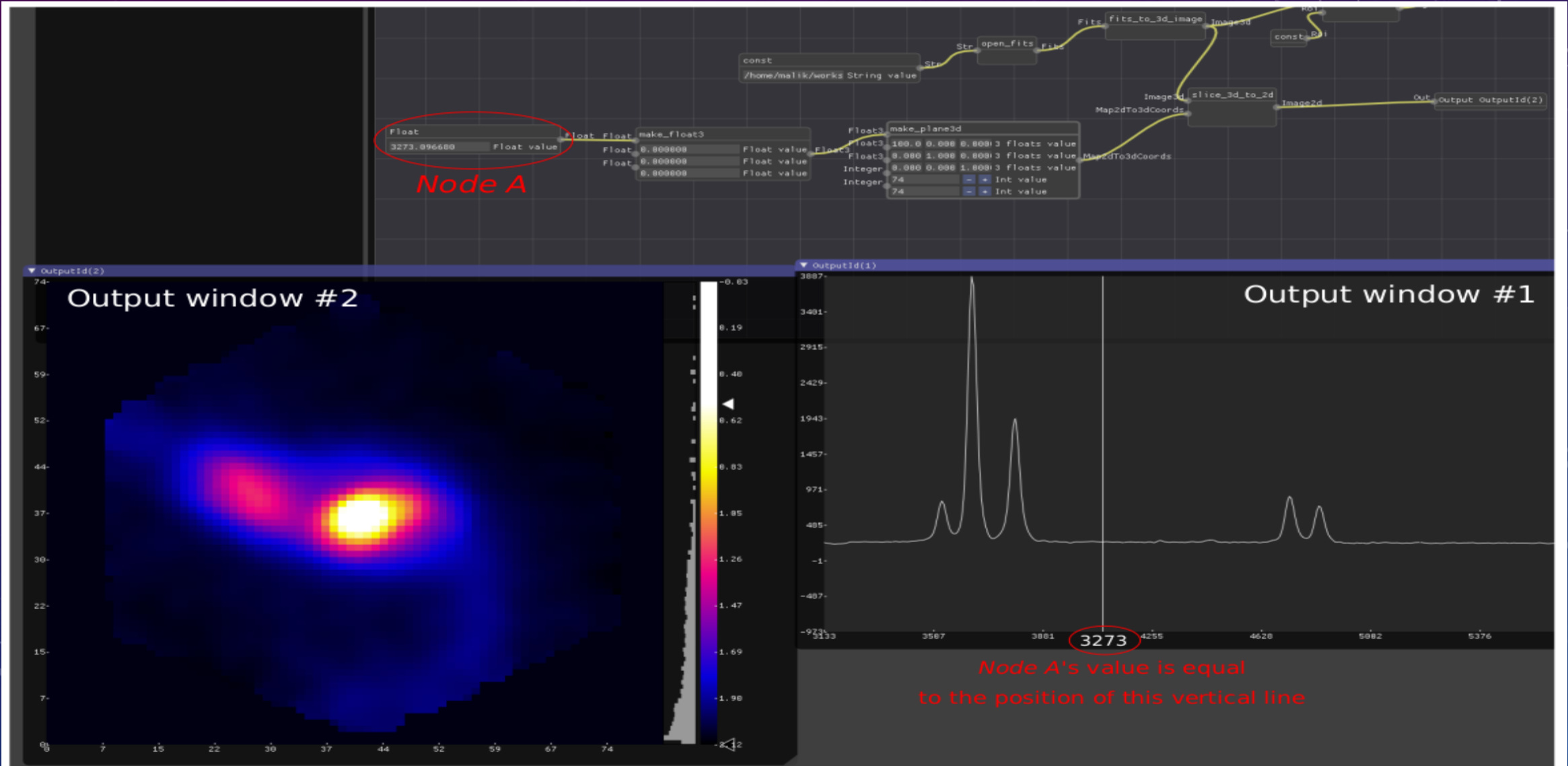


- Export and import with *ad hoc* format

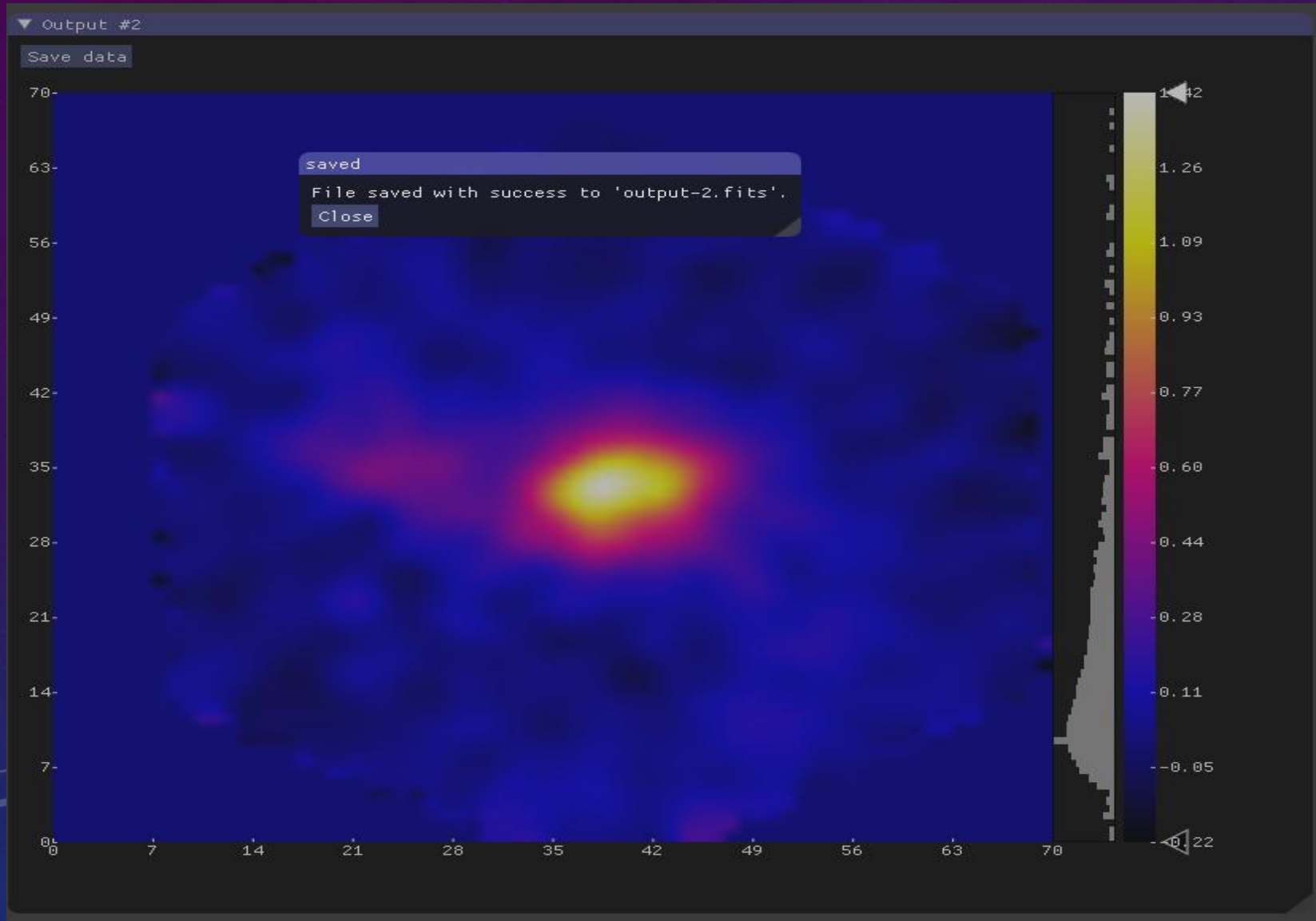
Question

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

DOUBLE-FEEDBACK BETWEEN NODE EDITOR'S VARIABLES AND VISUALIZATION OUTPUT



SAVE OUTPUT AS FITS FILE CONTAINING END-TO-END LINEAGE OF HOW THE DATA WAS CREATED

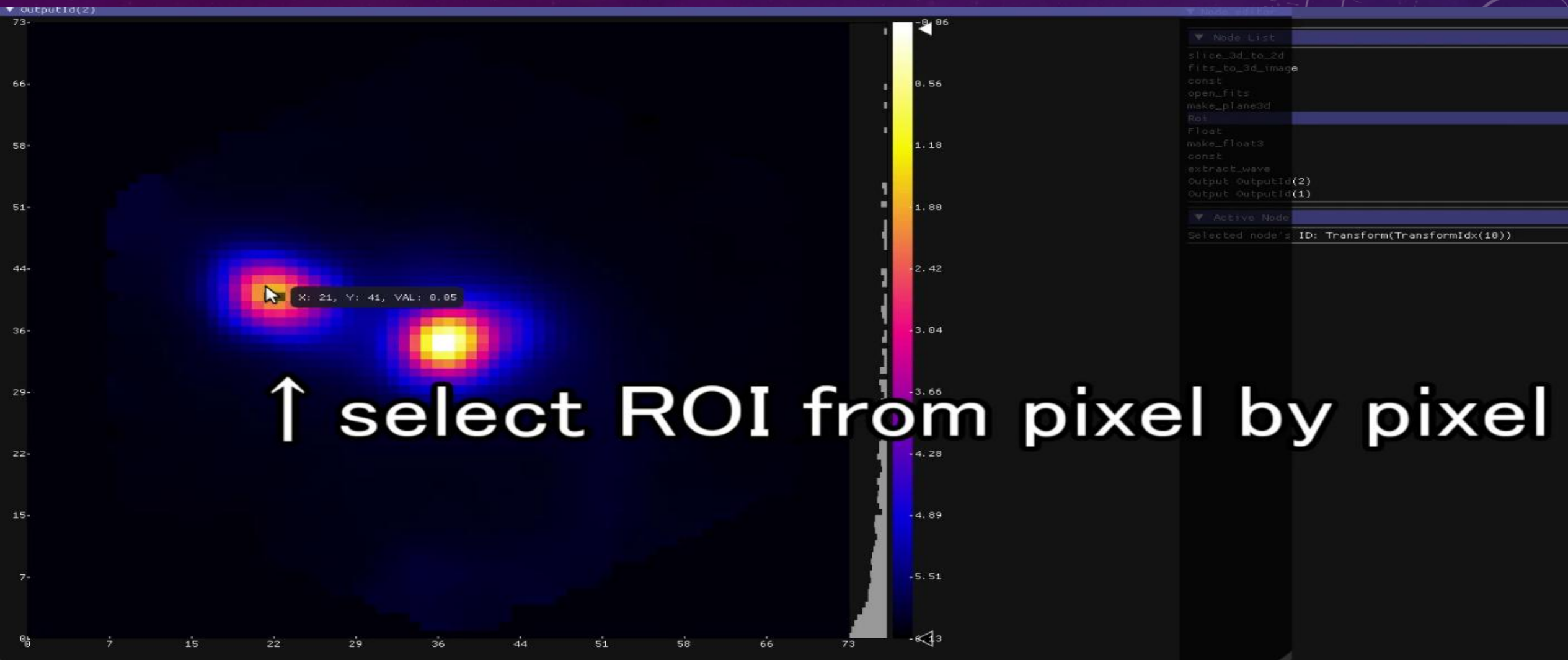


Question

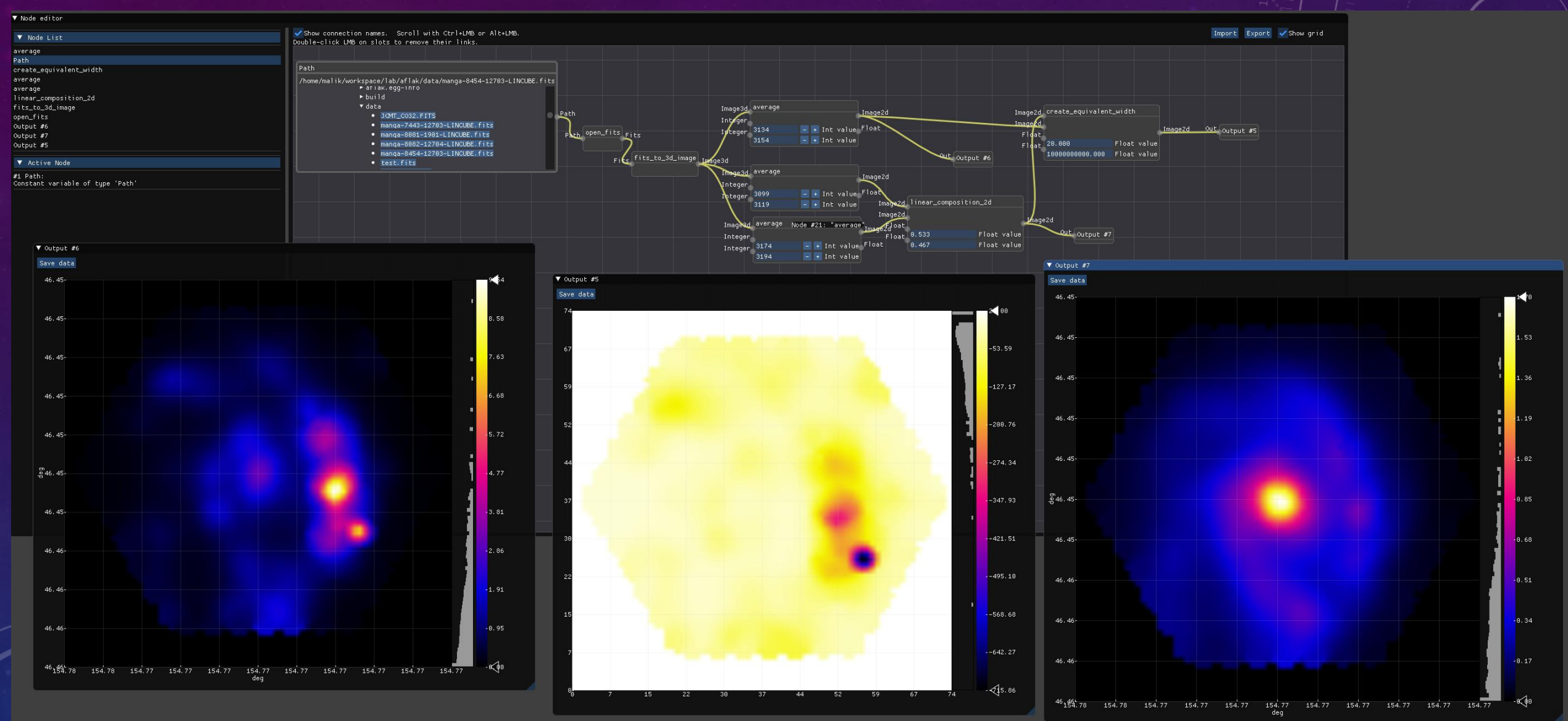
Any kind of
“standard”
procedure
for recording
data provenance?

SELECTING REGION OF INTEREST

<http://aflak.jp>



CASE STUDY: EXTRACTION OF EQUIVALENT WIDTH



USE CASES

- Analysis that requires gradual and manual fiddling with many parameters
aflak aims at putting the astronomer in the loop
- Denoising, preprocessing?

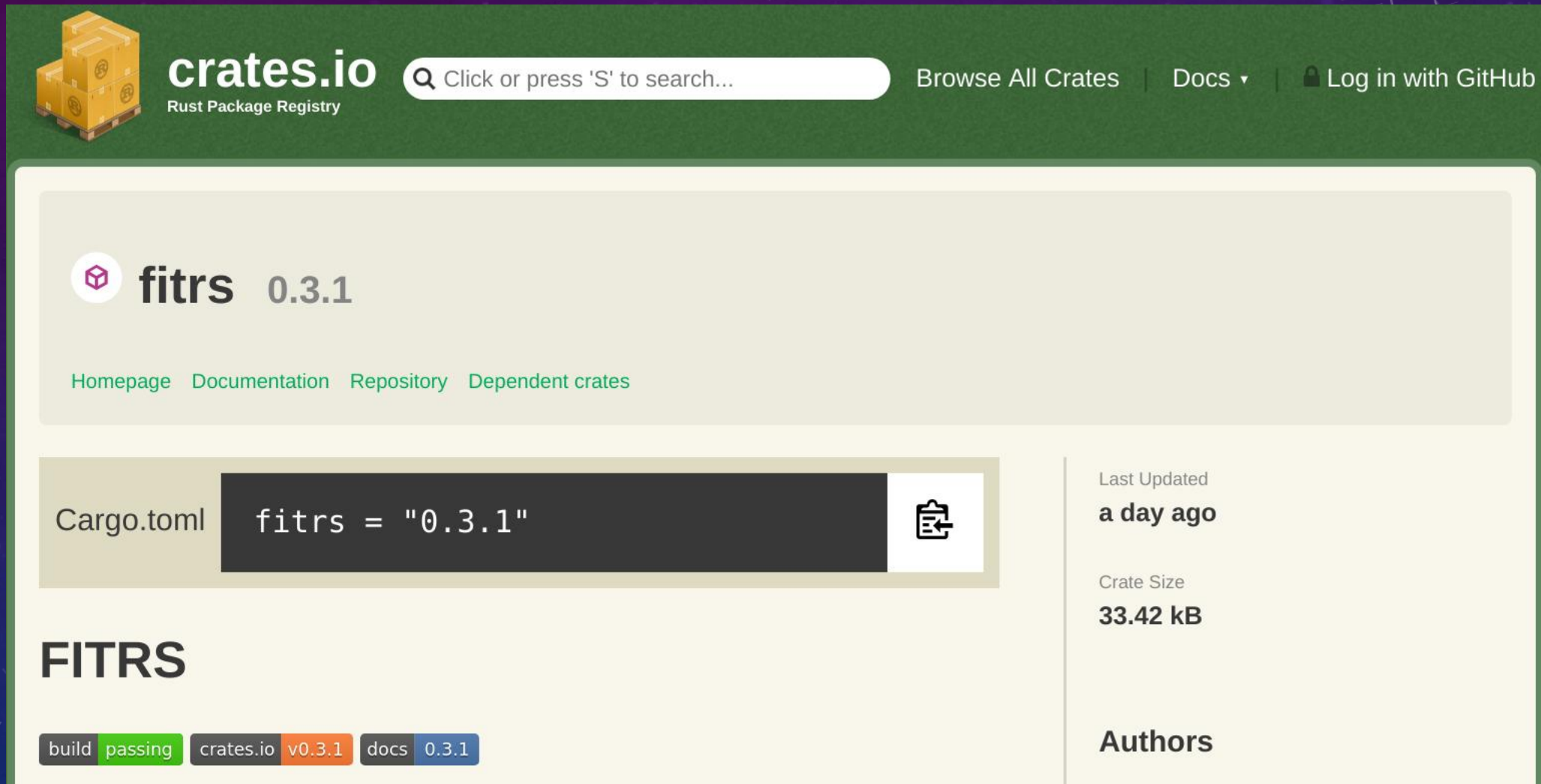
PRIMITIVES

<http://aflak.jp>


- 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)

FITS LIBRARY IN RUST: FITRS


<https://github.com/malikolivier/fitrs>




The screenshot shows the crates.io interface for the 'fitrs' crate. At the top, there's a search bar and navigation links. The main content area displays the crate name 'fitrs' and version '0.3.1'. Below this, there are links for 'Homepage', 'Documentation', 'Repository', and 'Dependent crates'. A 'Cargo.toml' snippet is shown with the line 'fitrs = "0.3.1"'. On the right, there are statistics: 'Last Updated a day ago' and 'Crate Size 33.42 kB'. At the bottom, there are status indicators for 'build passing', 'crates.io v0.3.1', and 'docs 0.3.1'. The 'Authors' section is partially visible at the bottom right.

 **crates.io**
Rust Package Registry

Click or press 'S' to search... | Browse All Crates | Docs | Log in with GitHub

 **fitrs** 0.3.1

[Homepage](#) [Documentation](#) [Repository](#) [Dependent crates](#)

Cargo.toml `fitrs = "0.3.1"` 

FITRS

build passing crates.io v0.3.1 docs 0.3.1

Last Updated
a day ago

Crate Size
33.42 kB

Authors

FUTURE WORK

- Batch processing
- Full macro support (Sit back! Writing the code as we speak, will be release very soon!)
- Full WCS support . Currently only partial and most probably buggy support is implemented.
- VO standards for communication with Aladin / Topcat
- Node primitives implementable in languages other than Rust (e.g. Python, C)
- Have more primitives included by default

ONE (TWO) –COMMAND INSTALL!

<http://aflak.jp>

```
$ curl https://sh.rustup.rs -sSf | sh
```

```
$ cargo install --git https://github.com/aflak-vis/aflak aflak
```

- Currently supported OSes:
 - Debian 9.X
 - Ubuntu 18.04
 - macOS (partial)
 - Crashes on Windows
- Run on normal laptop. RAM requirements depend on the open datasets. 4GB or more is advised.
- Bug report / Feature requests / Comment / Anything

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