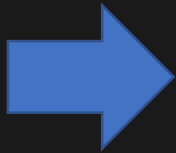


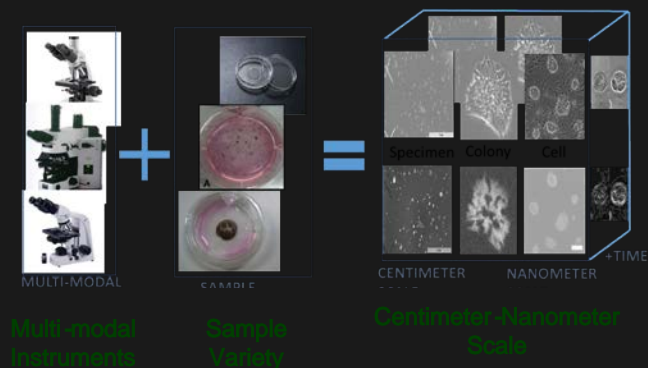


# Foundational Problem

How can a biologist harness petabytes of image data to derive quantitative measurements and gain meaningful insights?



# NIST Measurement: It is not just a number!



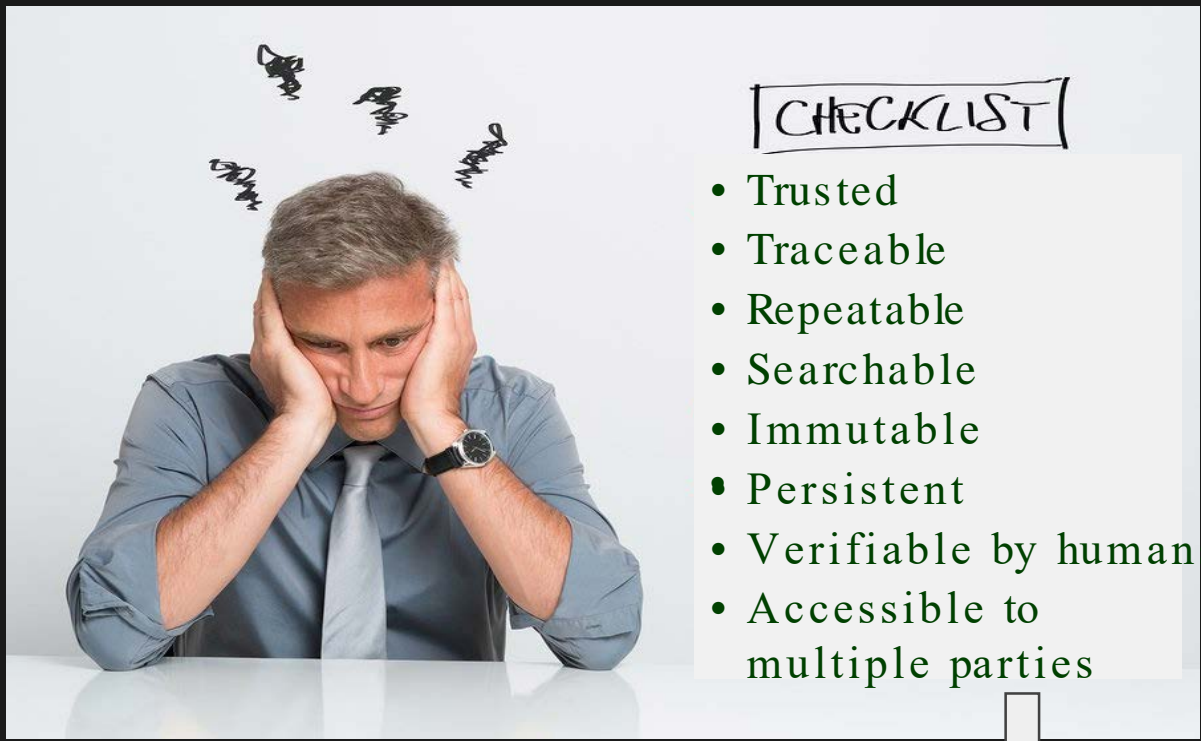
Scale

Complexity

Speed

Number [unit]

Challenge: Incorporate all measurement attributes



## CHECKLIST

- Trusted
- Traceable
- Repeatable
- Searchable
- Immutable
- Persistent
- Verifiable by humans
- Accessible to multiple parties



# Data Independent User Desired Features

	Automated Data Ingestion	Fast Data Previews	GUI-Workflows and Analysis	Hierarchical Catalog GUI	Collaborative Sharing & Coding	Advanced Automated Analysis	Simple Metadata Association	Easily Extensible	Native Broad Support	Scales Easily & Efficiently	Built-in Traceability and Reproducibility	Secure
Ingestion + OMERO/OpenBIS + CryoSpark + Python	2	2	3	3		3		3	3	2	2	
Ingestion + OMERO/OpenBIS + IMOD + MATLAB	2	3		3		3	2	3	3	2		
Ingestion + Columbus + Palantir/Spotfire/etc.	2	3	3	3	2	3	3		2	3	2	3
Ingestion + Midas + CMake + VTK + ITK + Paraview	2	3			3	3		2	3	2		0
Polus/WIPP 3.0	3	3	3	3	3	3	3	3	3	3	3	3

Fully Supported
Partially Supported
Not Supported

# Data Independent Developer Desired Features

	Open Source	Robust Code Documentation and Installation	Access to All Feature and Resource APIs	Well Documented APIs	Long Term Legacy Support APIs	Easy to Customize GUI	Simple Multi-Lingual Plugins Support	Open Plugin Repository	Easy & Automated Porting of Plugins	Native Broad Microscope Support	Collaborative Sharing/ Coding	Scales Easily & Efficiently
Ingestion + OMERO/OpenBIS + CryoSpark + Python	3	3	2	2	2	0	0	3	0	3	0	2
Ingestion + OMERO/OpenBIS + IMOD + MATLAB	0	2	2	3	2	0	0	3	0	3	0	2
Ingestion + Columbus + Palantir/Spotfire/etc	0	0	2	3	0	0	2	0	0	2	2	3
Ingestion + Midas + CMake + VTK + ITK + Paraview	3	3	3	3	3	0	0	3	0	3	3	2
Polus/WIPP 3.0	3	3	3	3	2	3	3	3	3	3	3	3

Fully Supported

Partially Supported

Not Supported

# Data Independent Admin. Desired Features

	Simple Automated Deployment	Generalizable & Flexible Deployment	Well Documented Deployment Instructions	Simple & Robust Reporting, Patching, & Updates	Application Performance Management	System Resource Management	User Utilization Management	Role Based Access Control (UI, plugins, storage, compute)	API Management	Vulnerability Management	Capacity Management
Ingestion + OMERO/OpenBIS + CryoSpark + Python	2	2	2								
Ingestion + OMERO/OpenBIS + IMOD + MATLAB	2	2	2					2			
Ingestion + Columbus + Palantir/Spotfire/etc.	2	2	2					2			
Ingestion + Midas + CMake + VTK + ITK + Paraviewer	2		2								
Polus/WIPP 3.0	3	3	3	2	3	3	3	2			2

Fully Supported
Partially Supported
Not Supported





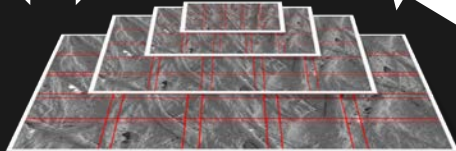
# High-Level Perspective on WIPP

## Automated Processing:

- Flat field correction
- Noise filtering
- Stitching
- Image pyramid building



Servers



Pyramid Representation of One Large Field of View

## Interactive Analyses:

- Tabular and Image Annotations
- Visual Verification and Validation
- Sub-setting and Measurements

Upload



Workflow Construction

Visualization

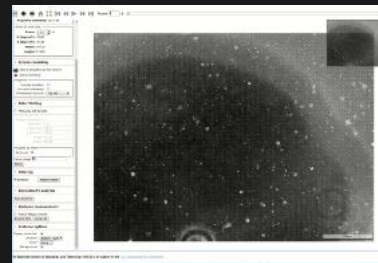
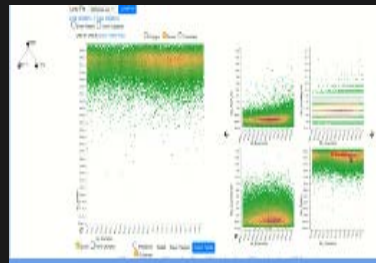
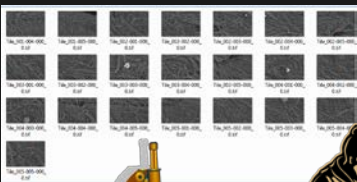


Computation

Many Small Image Fields of View

Algorithmic Plugins

Interactive Interfaces



# WIPP: Client - Server System - Image Processing + Analysis

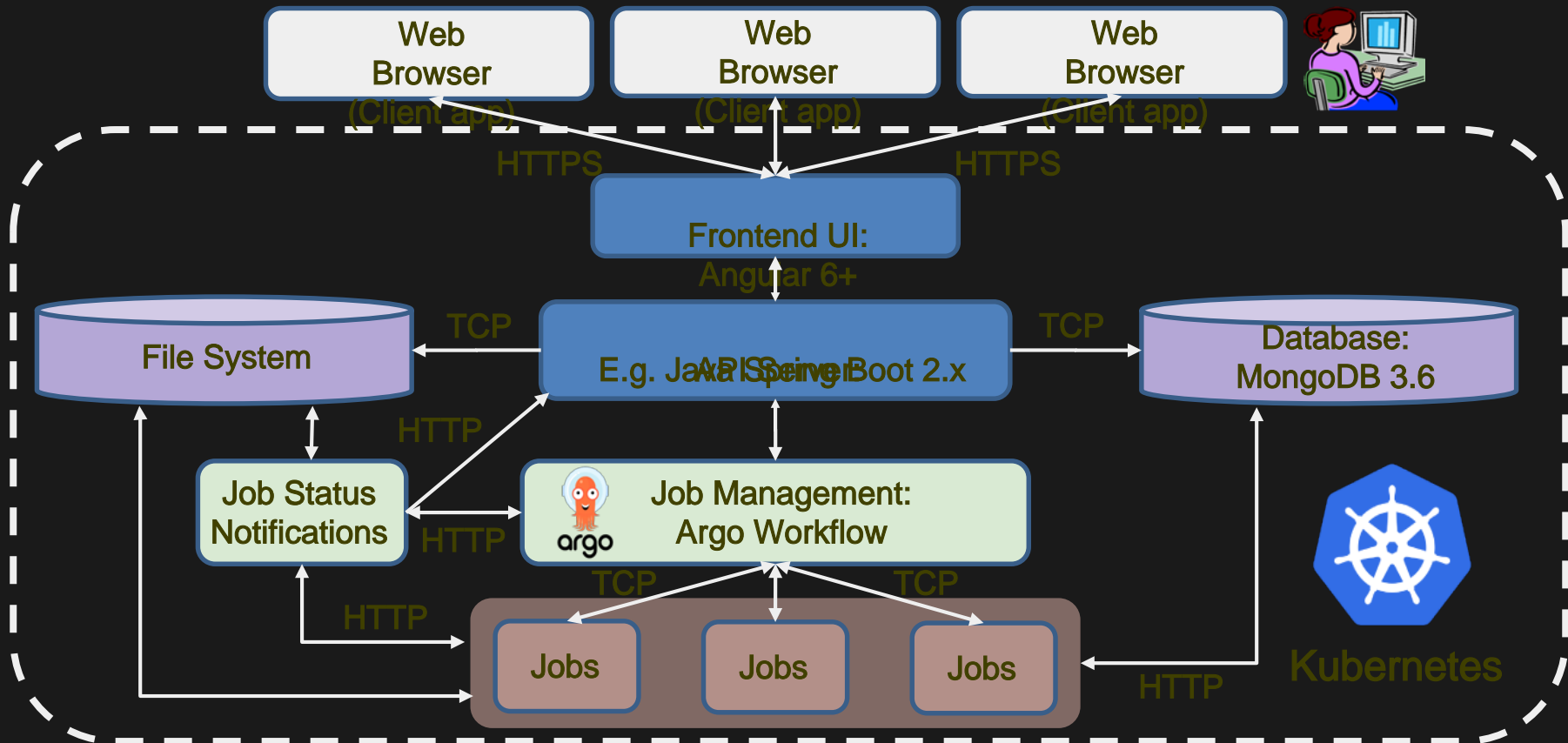
The screenshot shows the WIPP web interface. At the top, there's a navigation bar with 'WIPP Web Image Processing Pipelines' and a grid icon. Below it, there are tabs for 'IMAGE COLLECTIONS', 'IMAGE PROCESSING', and 'FEATURE EXTRACTION'. The main content area is titled 'Image Collections' and includes a 'Create new collection' button. A table below has columns for 'Name', 'Number of images', 'Lock', 'Creation date', and 'Total size'. A central window displays a multi-frame image with various layers and annotations. The interface includes navigation controls, a 'Layers to fetch' sidebar, and a 'Frame 1 / 1' indicator.

The screenshot shows the WIPP web interface. At the top, there's a navigation bar with 'WIPP Web Image Processing Pipelines' and a grid icon. Below it, there are tabs for 'IMAGE COLLECTIONS', 'IMAGE PROCESSING', and 'FEATURE EXTRACTION'. The main content area is titled 'Stitching Jobs' and includes a 'Create new job' button. A table below has columns for 'Name', 'Status', 'Creation date', 'Start time', 'End time', and 'Duration'. A detailed view of a 'Create new stitching job' form is shown, including fields for 'Job name', 'Job collection', 'Algorithm', 'File name pattern', 'Time slices', 'Stitching search', 'Direction', 'Number of columns', 'Number of rows', 'Horizontal overlap', and 'Vertical overlap'. A 3x3 grid of coordinates is also visible.

Name	Status	Creation date	Start time	End time	Duration
test-wipp-release-stitching	DONE	Jun 16, 2017 10:35:09 PM	Jun 16, 2017 10:35:09 PM	Jun 16, 2017 10:36:09 PM	00:00:31
wipp_mid_time_series_test_data_STITCH_all_1_copy	DONE	Jun 16, 2017 4:11:48 PM	Jun 16, 2017 4:12:16 PM	Jun 16, 2017 4:12:48 PM	00:00:30
wipp_mid_time_series_test_data_metadata	FAILED	Jun 16, 2017 10:30:32 AM	Jun 16, 2017 10:30:27 AM	Jun 16, 2017 10:30:43 AM	00:00:06
MMAL_RPL_RAN_stitching_test_copy	DONE	Jun 9, 2017 10:30:37 AM	Jun 9, 2017 10:10:28 AM	Jun 9, 2017 10:10:39 AM	00:00:05
MMAL_RPL_RAN_stitching_test	DONE	Jun 9, 2017 10:10:11 PM	Jun 8, 2017 10:49:46 PM	Jun 8, 2017 10:50:17 PM	00:00:06

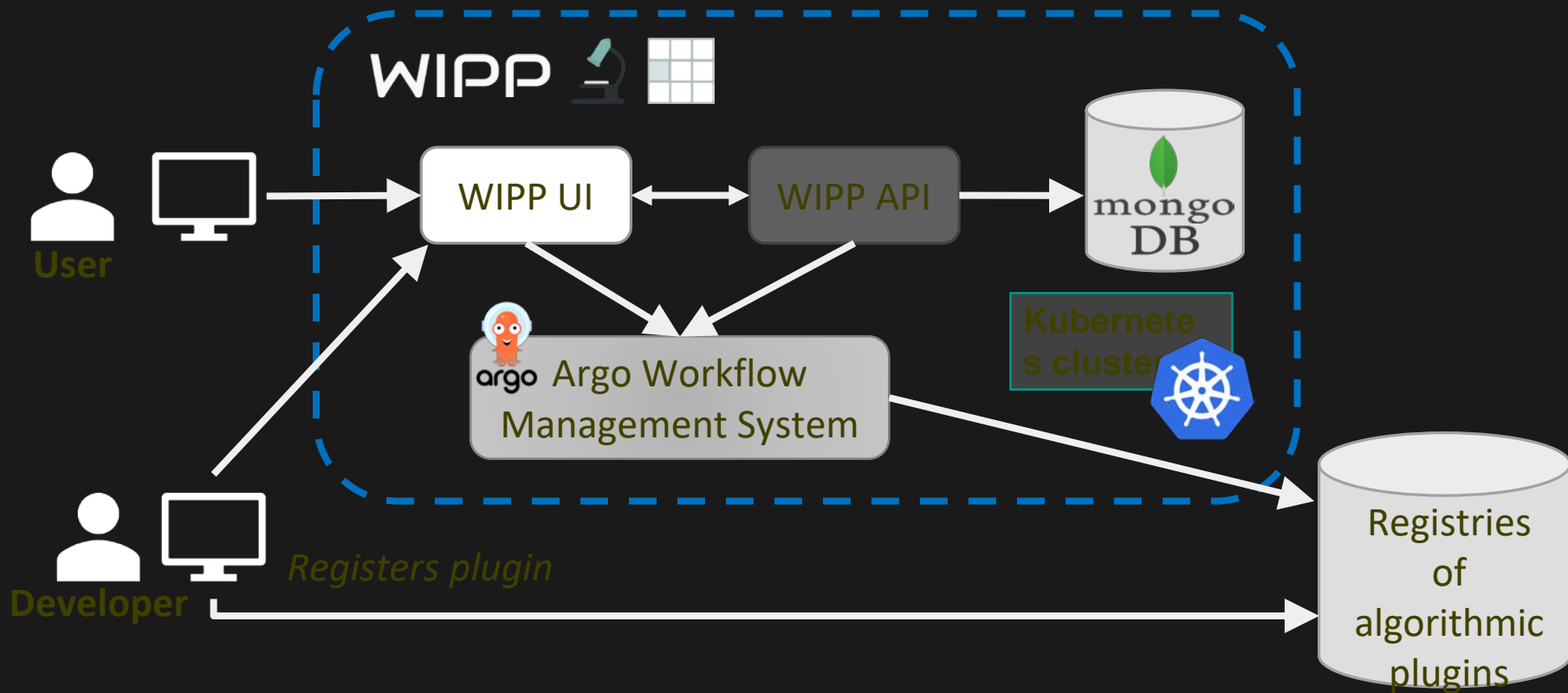
$(r_1, c_1)$	$(r_1, c_2)$	$(r_1, c_3)$
$(r_2, c_1)$	$(r_2, c_2)$	$(r_2, c_3)$
$(r_3, c_1)$	$(r_3, c_2)$	$(r_3, c_3)$

# WIPP 3.0 ClientServer Architecture



Focused on Functionalities for Users, Developers, and System Administrators

# WIPP 3.xx Architecture



# Containerization and Orchestration Technologies

- Docker



*"Docker containers wrap up software and its dependencies into a standardized unit for software development that includes everything it needs to run: code, runtime, system tools and libraries."*

- Kubernetes

8



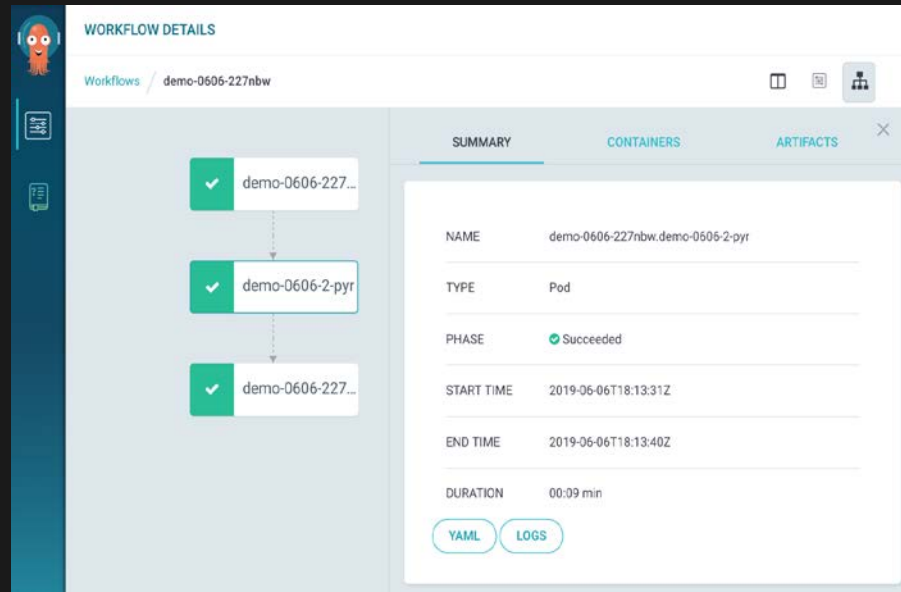
*"Kubernetes is an opensource system for automating deployment, scaling, and management of containerized applications. Production-Grade Container Orchestration."*

# Container Workflow Technologies

- Argo Workflows



*“Argo Workflows is an open source container-native workflow engine for orchestrating parallel jobs on Kubernetes.”*



The screenshot displays the 'WORKFLOW DETAILS' page for a workflow named 'demo-0606-227nbw'. The workflow is shown as a vertical sequence of three steps, each with a green checkmark indicating success. The steps are labeled 'demo-0606-227...', 'demo-0606-2-pyr', and 'demo-0606-227...'. The right-hand panel shows the 'SUMMARY' tab with the following details:

NAME	demo-0606-227nbw.demo-0606-2-pyr
TYPE	Pod
PHASE	✔ Succeeded
START TIME	2019-06-06T18:13:31Z
END TIME	2019-06-06T18:13:40Z
DURATION	00:09 min

At the bottom of the summary panel, there are buttons for 'YAML' and 'LOGS'.

# Algorithmic Plugins

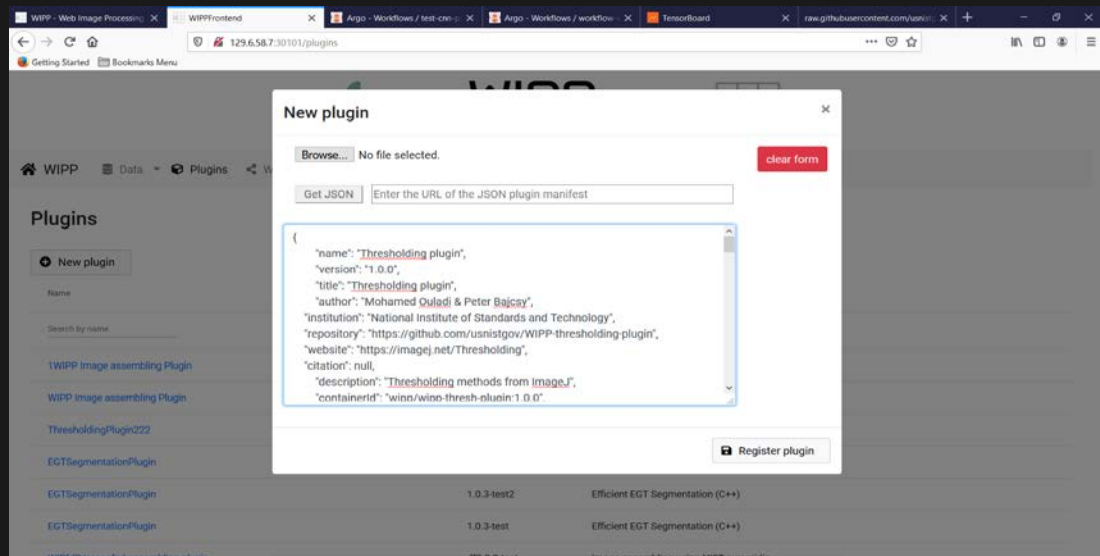
## Plugin Components

- JSON Manifest

- Location of docker file
- Inputs/Outputs
- User interface definition

- Docker Container

- Algorithms
- Libraries
- Abstraction  
Middleware



wipp/wipp-thresh-plugin

By wipp • Updated 3 months ago

WIPP thresholding plugin based on ImageJ/Fiji thresholding methods.

Container

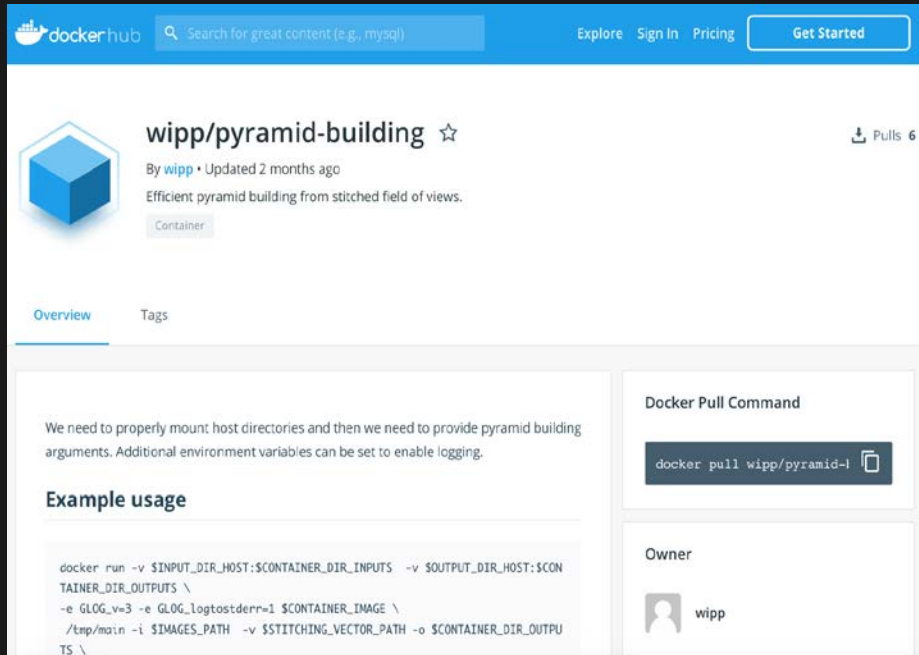
Linux

x86-64

[https://  
hub.docker.com/](https://hub.docker.com/)

# Plugin Component: Docker Image

WIPP plugin as a Docker image available on DockerHub



The screenshot shows the Docker Hub page for the `wipp/pyramid-building` image. The page includes the Docker Hub logo, a search bar, and navigation links for Explore, Sign In, Pricing, and Get Started. The image details include the name `wipp/pyramid-building`, the maintainer `wipp`, and the description "Efficient pyramid building from stitched field of views." There are 6 pulls and a star icon. The page is categorized as a "Container" and has tabs for Overview and Tags. A section titled "Example usage" provides a `docker run` command. A "Docker Pull Command" box shows `docker pull wipp/pyramid-building`. The "Owner" section identifies the maintainer as `wipp`.

docker run -v \$INPUT\_DIR\_HOST:\$CONTAINER\_DIR\_INPUTS -v \$OUTPUT\_DIR\_HOST:\$CONTAINER\_DIR\_OUTPUTS \

-e GLOG\_v=3 -e GLOG\_logtostderr=1 \$CONTAINER\_IMAGE \

/tmp/main -i \$IMAGES\_PATH -v \$STITCHING\_VECTOR\_PATH -o \$CONTAINER\_DIR\_OUTPUTS \

Docker image definition (Dockerfile)

```
24 lines (18 sloc) | 734 Bytes
1 FROM wipp/pyramid-building:1.0.0
2 LABEL maintainer="National Institute of Standards and Technology"
3
4 ARG EXEC_DIR="/opt/executables"
5 ARG DATA_DIR="/data"
6
7 # Create folders
8 RUN mkdir -p ${EXEC_DIR} \
9     && mkdir -p ${DATA_DIR}/inputs \
10    && mkdir ${DATA_DIR}/outputs
11
12 # Install java 8 jdk
13 RUN apt-get update \
14     && apt-get install -y openjdk-8-jdk \
15     && update-alternatives --set java /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
16
17 # Copy wipp-pyramid-plugin JAR
18 COPY target/wipp-pyramid-plugin*.jar ${EXEC_DIR}/wipp-pyramid-plugin.jar
19
20 # Set working directory
21 WORKDIR ${EXEC_DIR}
22
23 # Default command. Additional arguments are provided through the command line
24 ENTRYPOINT ["usr/bin/java", "--jar", "wipp-pyramid-plugin.jar"]
```



# Dynamic Plugin UI: JSON Manifest

```
{
  "name": "WIPP Pyramid plugin",
  "version": "0.0.1",
  "title": "WIPP Pyramid building",
  "description": "Pyramid building using NIST accelerated C++ algorithm",
  "creator": "National Institute of Standards and Technology",
  "containerId": "wipp/wipp-pyramid-plugin:0.0.1",
  "inputs": [
    {
      "name": "inputImages",
      "type": "collection",
      "options": {
        "format": "tiledtiff"
      },
      "description": "Input Images",
      "required": true
    },
    {
      "name": "inputStitchingVector",
      "type": "stitchingVector",
      "description": "Input Stitching Vector",
      "required": true
    },
    {
      "name": "blending",
      "type": "enum",
      "options": {
        "values": [
          "overlay",
          "max"
        ]
      },
      "description": "Blending method when assembling tiles",
      "required": false
    }
  ],
  "outputs": [
    {
      "name": "output",
      "type": "pyramid",
      "description": "Output pyramid"
    }
  ]
}
```

## Information about the plugin:

- name, version, description
- Docker image to use

## Inputs description

- name, type, description

## Outputs description

- name, type, description

WIPP will use this manifest to generate the job configuration form

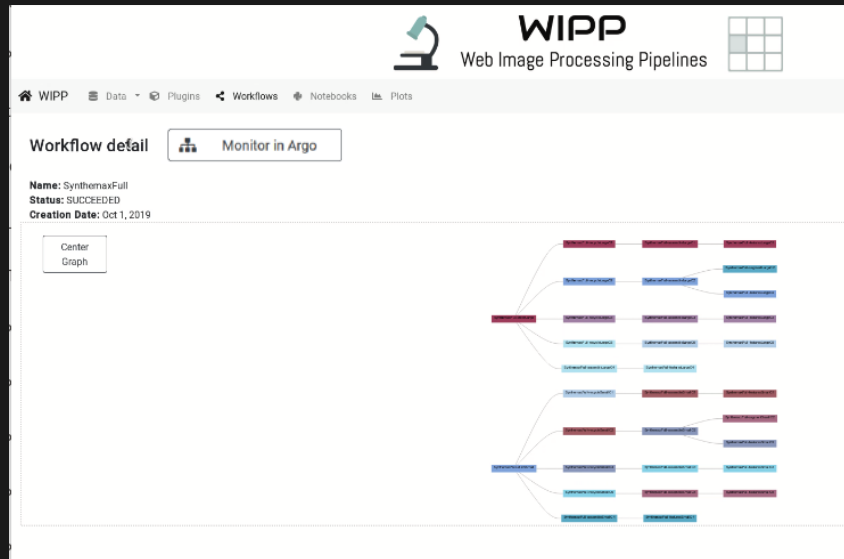
# Workflow Creation from Algorithmic Plugins

- Interactive workflow creation and visualization



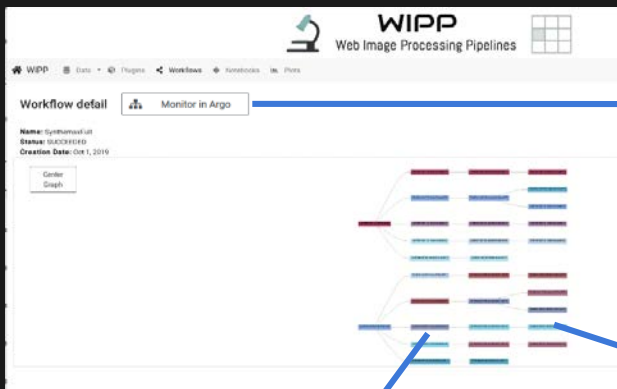
The screenshot shows the WIPP interface with a navigation bar at the top containing 'WIPP', 'Data', 'Plugins', 'Workflows', 'Notebooks', and 'Plots'. The main content area is titled 'Workflow detail' and includes a 'Submit workflow' button. Below this, the workflow information is displayed: 'Name: s12-demo', 'Status: CREATED', and 'Creation Date: Oct 2, 2019'. A 'Workflow tasks' section is visible with an 'Add task' button.

- Configure workflow tasks
- Chain job outputs



The screenshot shows the WIPP interface with a navigation bar at the top containing 'WIPP', 'Data', 'Plugins', 'Workflows', 'Notebooks', and 'Plots'. The main content area is titled 'Workflow detail' and includes a 'Monitor in Argo' button. Below this, the workflow information is displayed: 'Name: SynthmaxFull', 'Status: SUCCEEDED', and 'Creation Date: Oct 1, 2019'. The main area shows a complex workflow graph with multiple tasks connected by arrows, illustrating the chaining of job outputs.

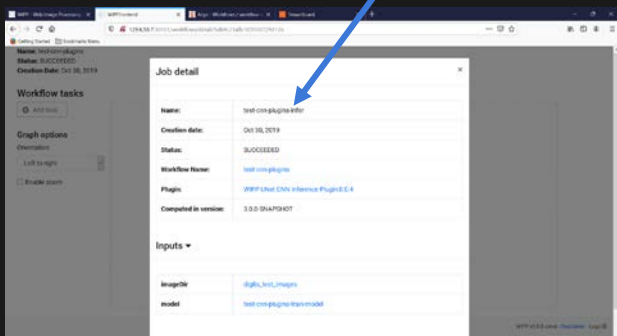
# Workflow Monitoring and Traceability



## Monitor Execution in Argo

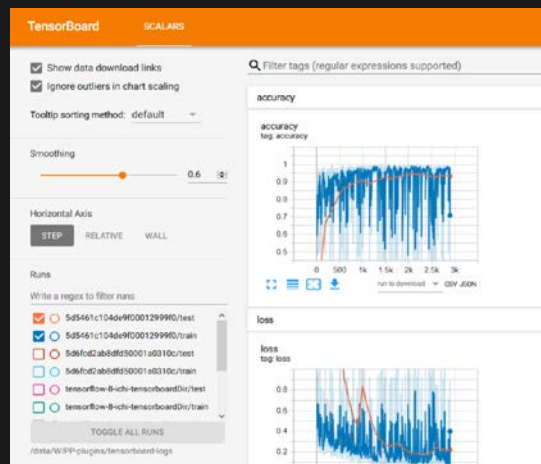


U-Net training  
U-Net inference



Traceable Parameters

Monitor Convergence in TensorBoard



# Interoperable Algorithmic Plugins

## Pros of Container Plugins

- Multi-lingual
  - C++, Python, Java, TensorFlow, etc.
- Containerized
  - Reproducibility of results
  - Compatibility of library dependencies
  - Simplicity of deployment
  - Modularity of analyses
- OSS or Paid
- Interoperable between NIH & NIST

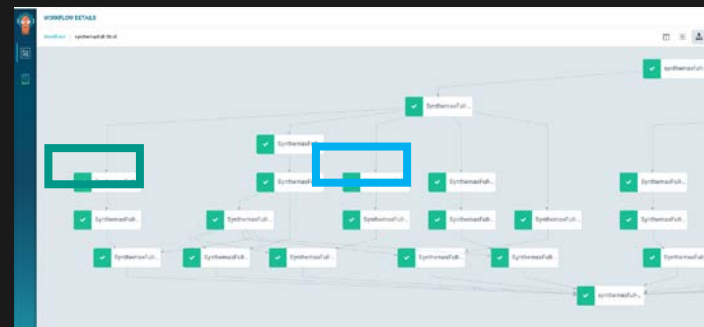
```
Declare variable in one language

[1]: %use Python3
      %put x
      x = [1,2,3]
      x

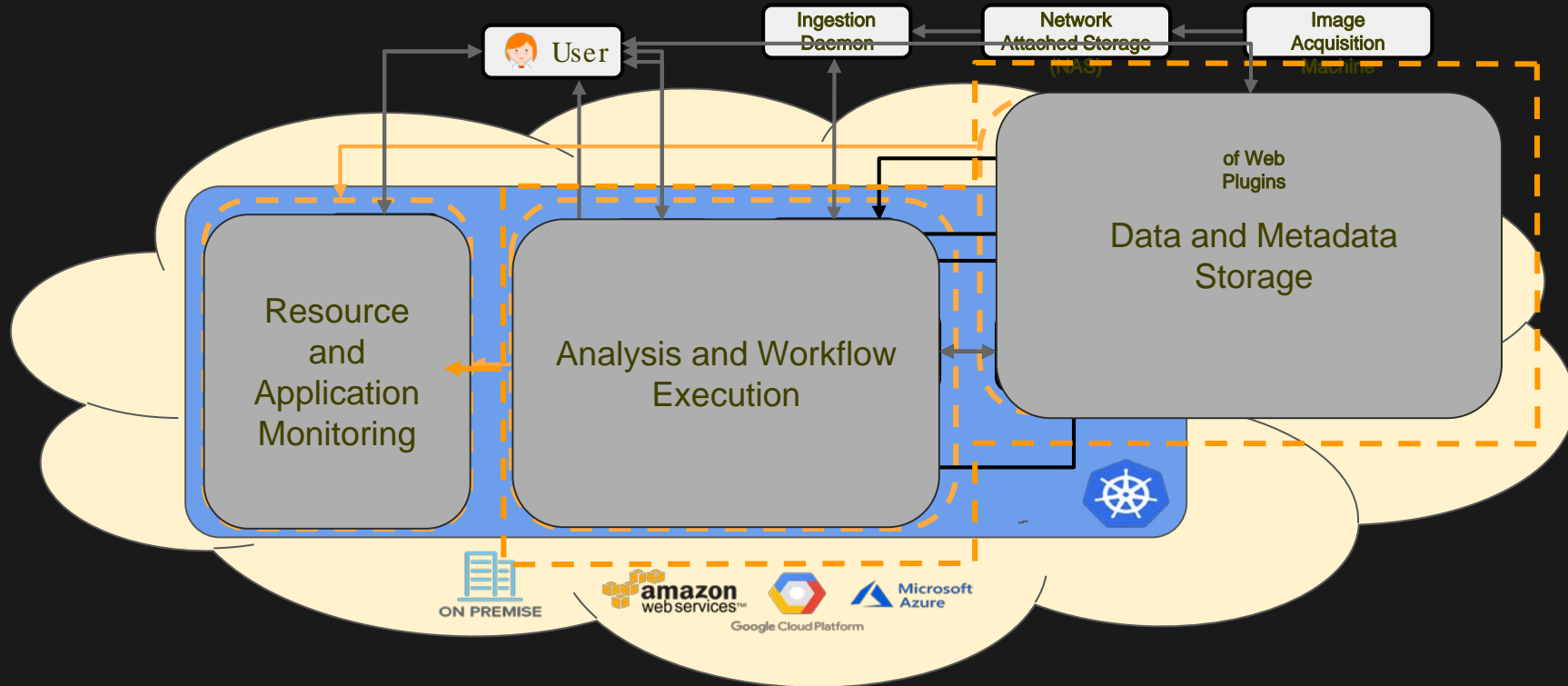
[1]: [1, 2, 3]

Access in other languages

[2]: %use R
      %get x
      x
```



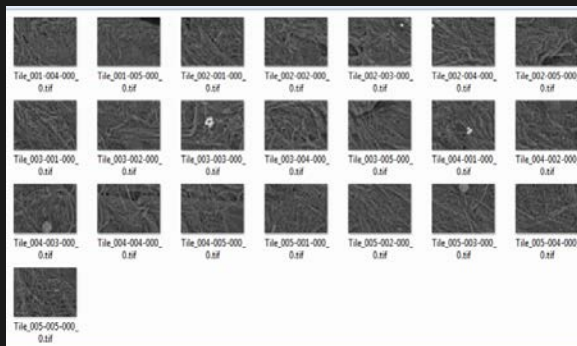
# Polus Services: Monitoring, Execution, and Data Storage - Managed Infrastructure!





# Algorithmic Plugins

- Image stitching (MIST algorithm)
- Image thresholding (ImageJ/Fiji multiple algorithms)
- Pyramid building (accelerated algorithm with low RAM requirements)
- Universal Notebook Execution (Scalably Execute Jupyter Notebooks as plugins)

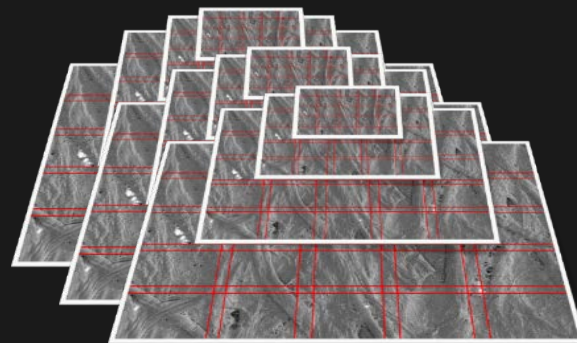


Many Overlapping Microscope Fields of View



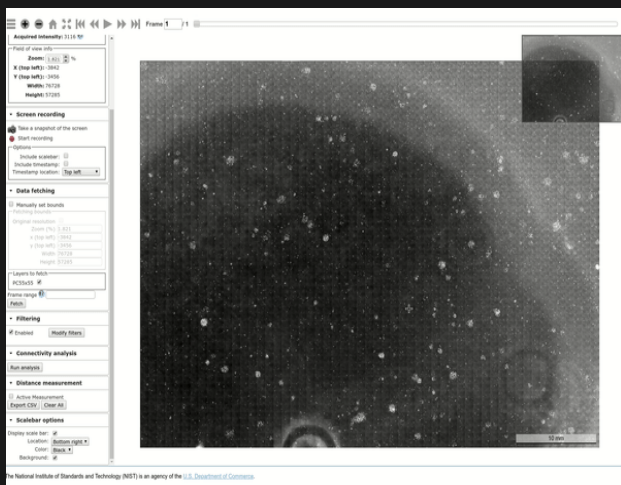
Dynamic  
Workflow Builder

Image Stitching and  
Pyramid Building



Stitched Interactive Image Pyramids

# Interactive Data-Driven Visual Discovery



Interactive whole sample visualization/manipulation

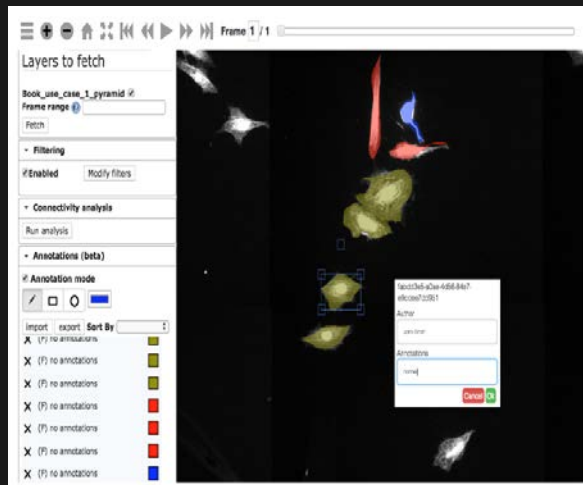


Illustration of WZDT annotation tool

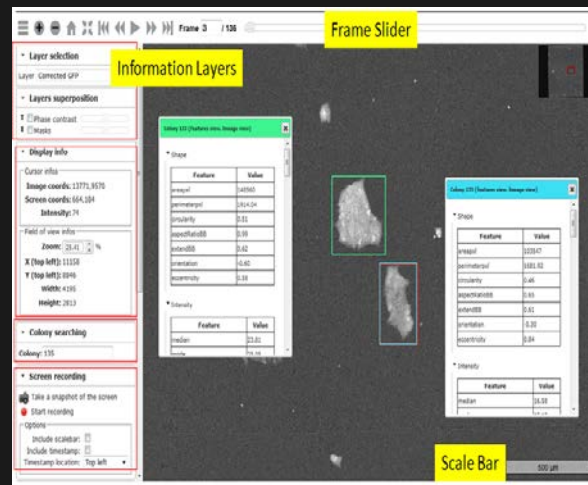
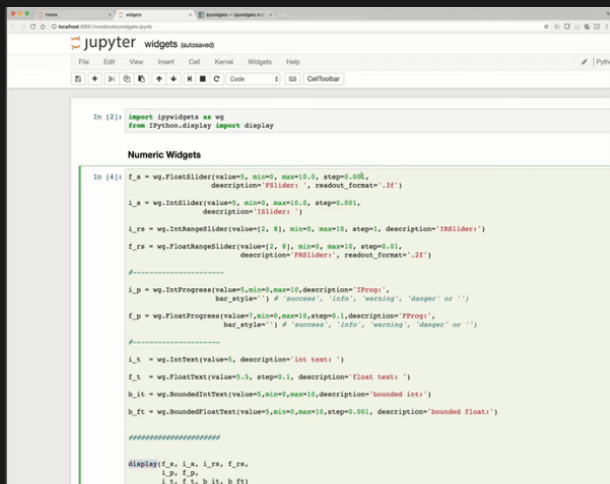


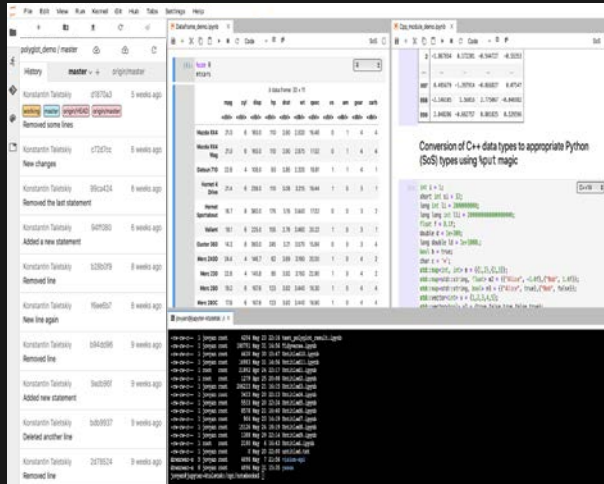
Illustration of linking features and ROIs



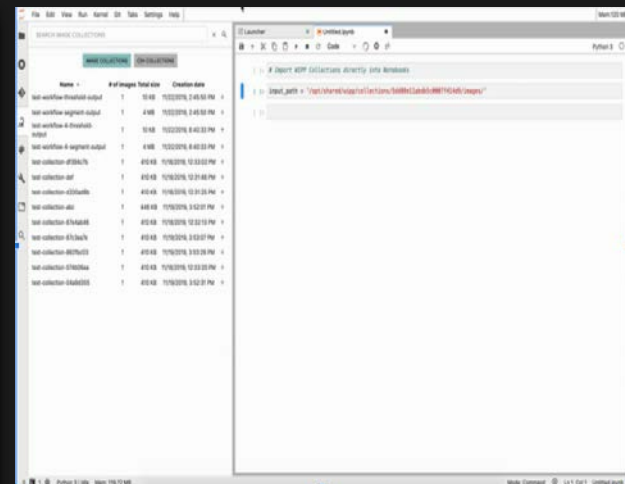
# Prototyping Using Jupyter Notebook Interactive Programming



Configurable in-browser Polyglot IDE located where images are stored

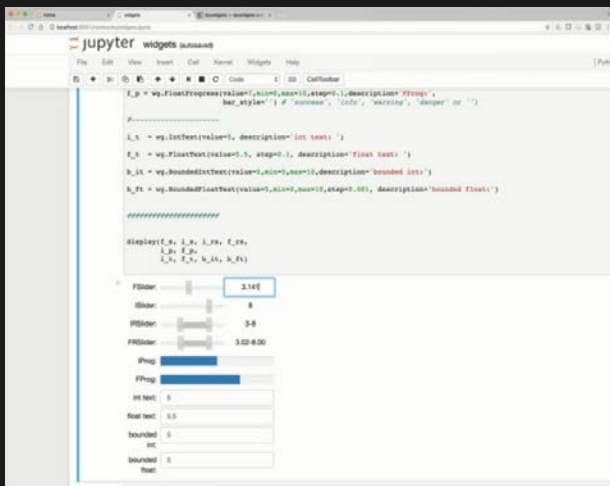


Collaborative Notebooks using GitHub GUI or terminal

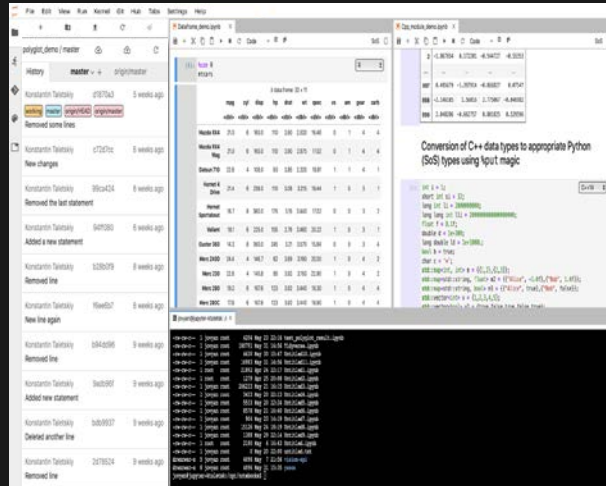


Tight Integration with WIPP Jupyterlab Extension

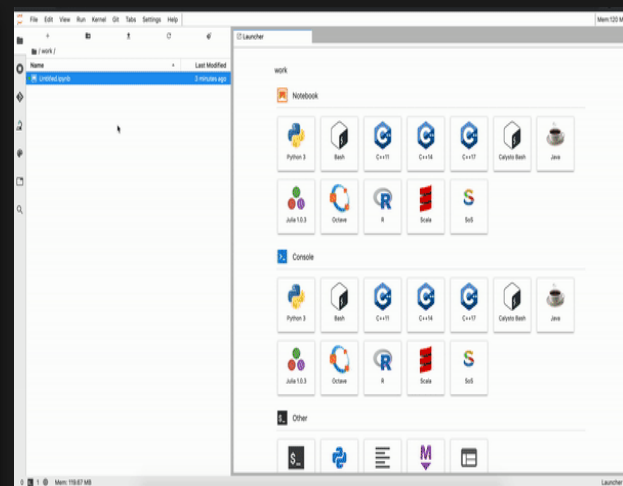
# Prototyping Using Jupyter Notebook Interactive Programming



Configurable in-browser Polyglot IDE located where images are stored

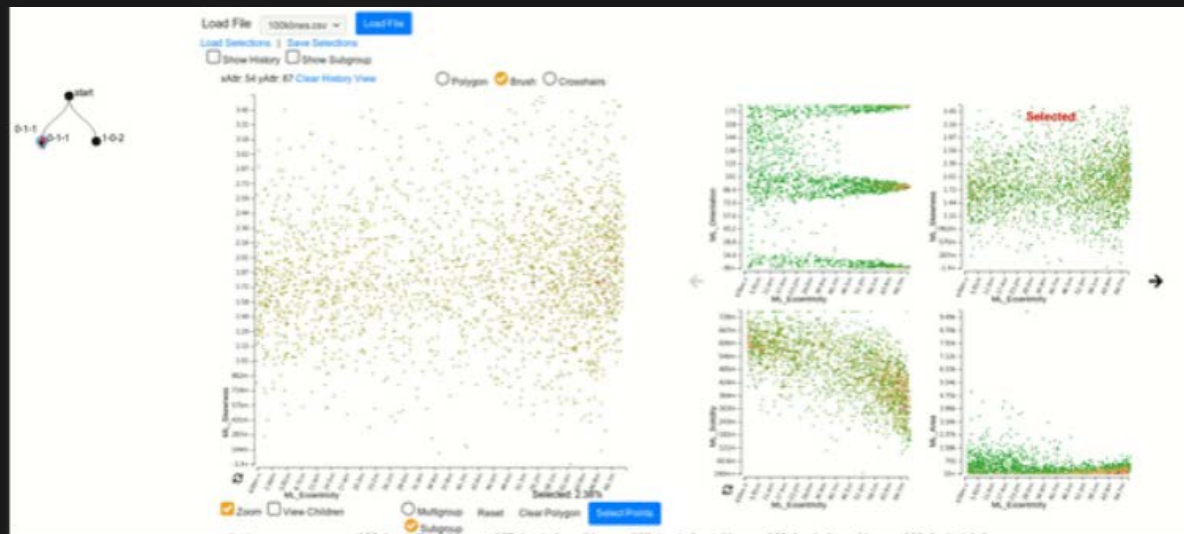
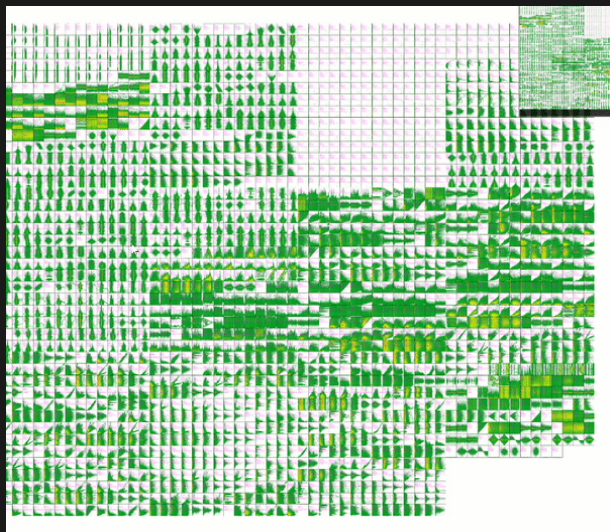


Collaborative Notebooks using GitHub GUI or terminal



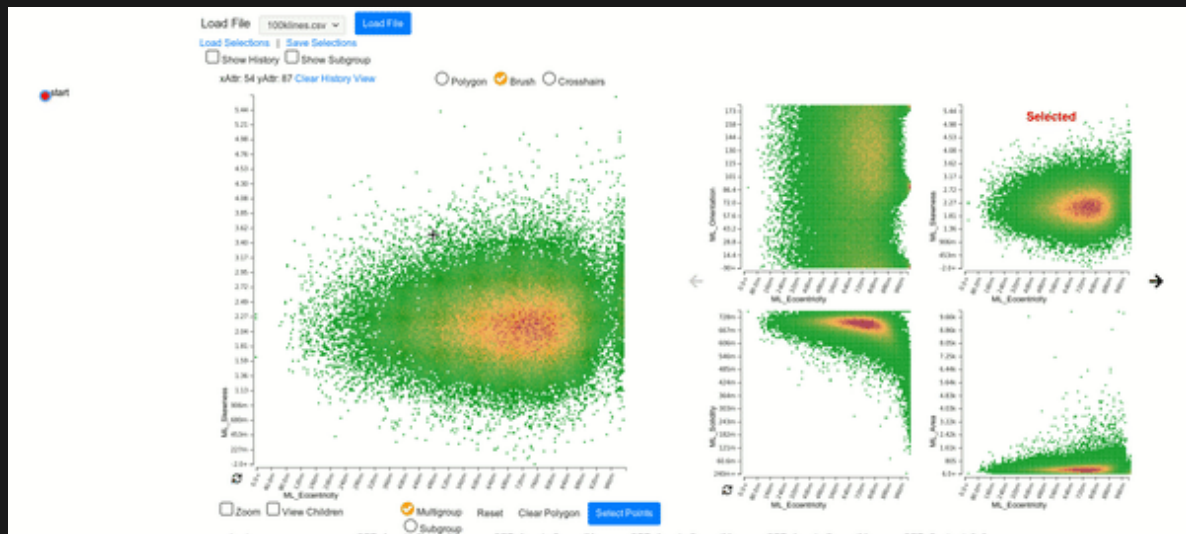
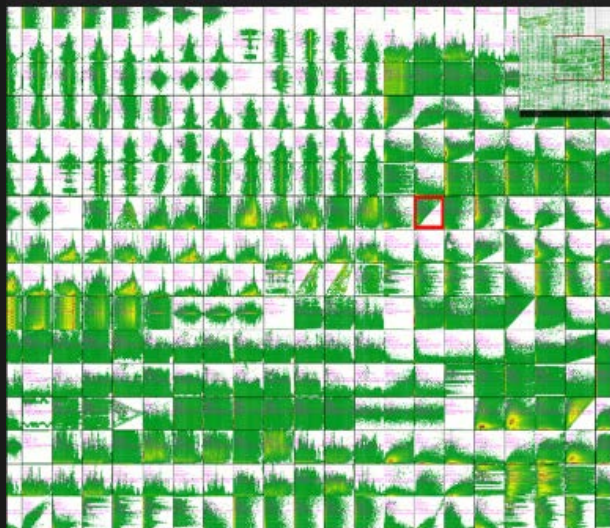
Tight Integration with WIPP Jupyterlab Extension

# Scalable Interactive Visualization of Tabular Data



- DeepZoom of all pair-wise feature scatter plots
- Interactive data sub-setting across dimensions, zoomed and global views
- Familiar dynamic group selection tools with a history dendrogram and plot view

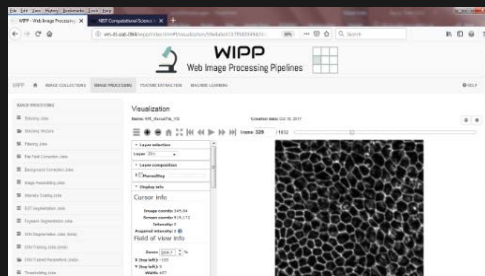
# Scalable Interactive Visualization of Tabular Data



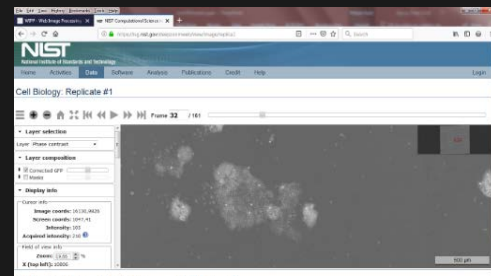
- DeepZoom of all pair-wise feature scatter plots
- Interactive data sub-setting across dimensions, zoomed and global views
- Familiar dynamic group selection tools with a history dendrogram and plot view

# Examples of Data Sharing and Dissemination

- **Provenance data:**hyperlinked computations with all metadata about data and execution configurations
- **Large size image data:**pyramid representation with Deep Zoom enabled browsing of images
- **Image-based measurements:**hyperlinked thumbnails and numerical values to persistent image ROIs



Provenance data associated with each computation



Persistent large size image data

# Dissemination (WIPP 3.0 Beta version)

- Source code is in GitHub repositories
  - WIPP (main repo with documentation and deployment instructions): <https://github.com/usnistgov/WIPP>
  - WIPP-backend (Java Spring REST API): <https://github.com/usnistgov/WIPP-backend>
  - WIPP-frontend (Angular 7 app): <https://github.com/usnistgov/WIPP-frontend>
- Docker containers are in Docker Hub
- We provide single node installation instructions for all platforms
  - Tested on Linux 18.04, Windows 10 Enterprise, and Mac OS 10
- WIPP 3.0 installation includes
  - Image processing and AI semantic segmentation plugins
  - Integrated Jupyter notebook
    - Prototyping in 13 programming languages
  - Interactive scatter plots
    - Visualization and sub-setting of millions of image-based measurements.



# NIH & NIST Development Team



# Credit

## NIST

- Mary Brady
- Peter Bajesy
- Joe Chalfoun
- Mike Majurski
- Mylene Simon
- Adele Peskin
- Petre Manescu
- Soweon Yoon
- Mohamed Ouladi
- Sarala Padi
- Antoine Gerardin
- Philippe Dessauw
- Walid Keyrouz
- Derek J uba
- Antonio Cardone
- Tim Blattner
- James Filliben
- Steve Lund
- Hari Iyer

- (1) Technical University of Lodz, Poland
- (2) NIH National Cancer Institute
- (3) University of Maryland at College Park

## NIH

- Sam Michael
- Anton Simeonov
- Robert Hohman
- Beth Fischer
- Marc Ferrer
- Sheldon Miller
- Brian Brooks
- Kapil Bharti
- Tiziana Cogliati
- Arvydas Maminishkis
- Bryan Hansen
- Yuchi Chen
- Kelli Wilson
- Nina Mezu
- Reid Simon
- Busola Grillo

- (4) The Lieber Institute for Brain Development
- (5) George Mason University
- (6) Axle Informatics

**Thank you to all  
who aren't  
specifically called  
out here!**

**The goal is huge  
and so is the  
team that got us  
here.**

## NIST BIO/MAT

- Anne Plant
- Kiran Bhadriraju
- Michael Halter
- John Elliott
- Jeff Stinson
- Ed Kwee
- Carl Simon
- Stephen Floreczyk
- Nick Schaub
- Sumona Sarkar
- John Henry Scott
- Nicholas Ritchie
- Keana Scott
- Charlie Camp
- Young Lee
- Marcus Cicerone

- (7) Rockefeller University

## External Collaborator

- Marcin Kociolek
- Piotr Szczypiński
- Carol Parent
- Christina Stuelten
- Michael Weiger
- Wolfgang Loser
- Amitabh Varshney
- Daniel Hoepfner
- Ron McKay
- Jana Kosecka
- Suhas Sharma
- Rafat Sarosh
- Gary Mays
- Dan Gareau



