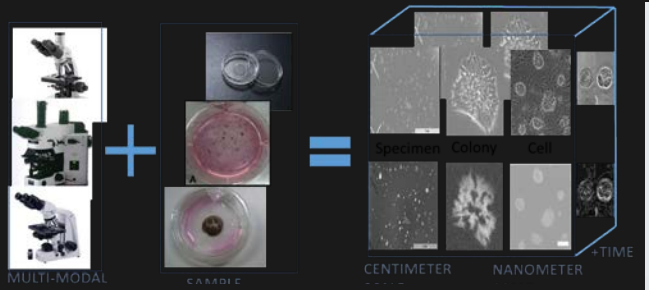
Web Image Processing Pipeline (WIPP) and Polus Services

[1]Nathan Hotaling & [2]Peter Bajcsy
[1]National Center for Advancing Translational Sciences (NCATS) NIH
[2]Information Technology Laboratory (ITL) NIST

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!



Multi-modal Instruments

Sample Variety

Centimeter-Nanometer Scale

Scale

Complexity

Speed

Number [unit]

Challenge: Incorporate all measurement attributes

Institutional Challenges

Software Features for

- Users
- Developers
- System Administrators

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 + Paraviewer	2	3			3	3		2	3	2	0	
Polus/WIPP 3.0	3	3	3	3	3	33	3	3	3	3	3	3

Fully Supported | Partially Supported | Not Supported

Link to Sheet

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		3		2
Ingestion + OMERO/OpenBIS + IMOD + MATLAB	0	2	2	3	2	0	0	3		3		2
Ingestion + Columbus + Palantir/Spotfire/etc	0		2	3		0	2		0	2	2	3
Ingestion + Midas + CMake + VTK + ITK + Paraviewer	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

Link to Sheet

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

Link to Sheet

Technical Approach

The computational and institutional challenges cannot be addressed without the involvement of communities of stakeholders

- Open Source Web Image Processing Pipeline (WIPP)
- Cloud service called Polus

High-Level Perspective on WIPP

Automated Processing:
- Flat field correction
- Noise filtering
- Stitching
- Image pyramid building

Cloud Computing

Servers

Interactive Analyses:
- Tabular and Image Annotations
- Visual Verification and Validation
- Sub-setting and Measurements

Pyramid Representation of One Large Field of View

Upload

Workflow Construction

Visualization

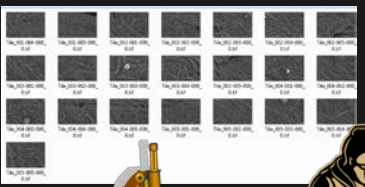
Computation

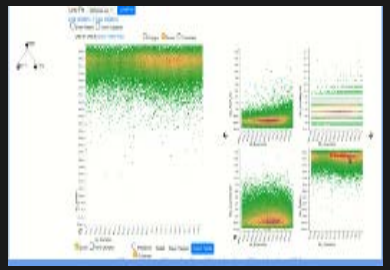
Many Small Image Fields of View

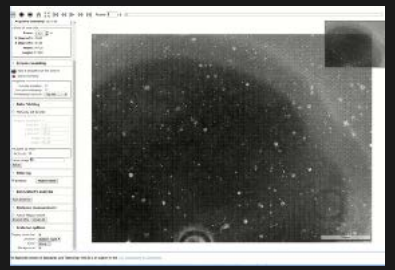
Algorithmic Plugins

{dev}

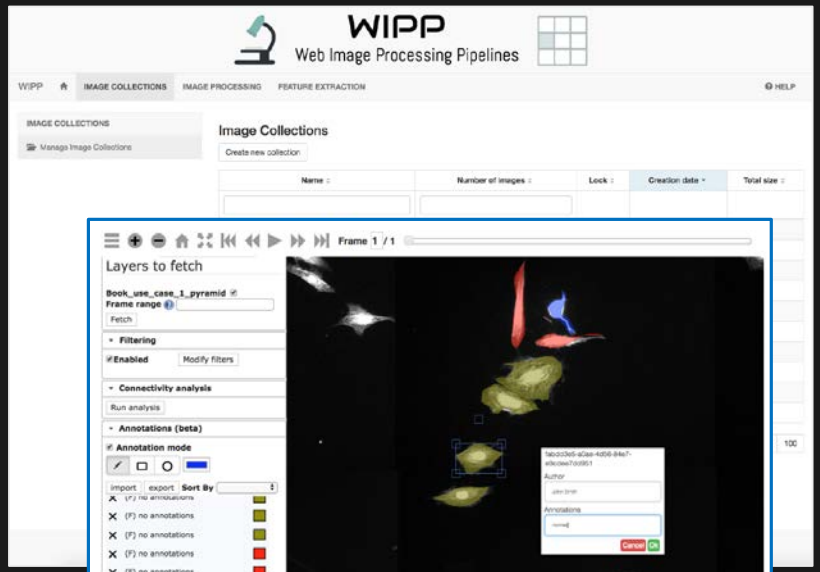
Interactive Interfaces

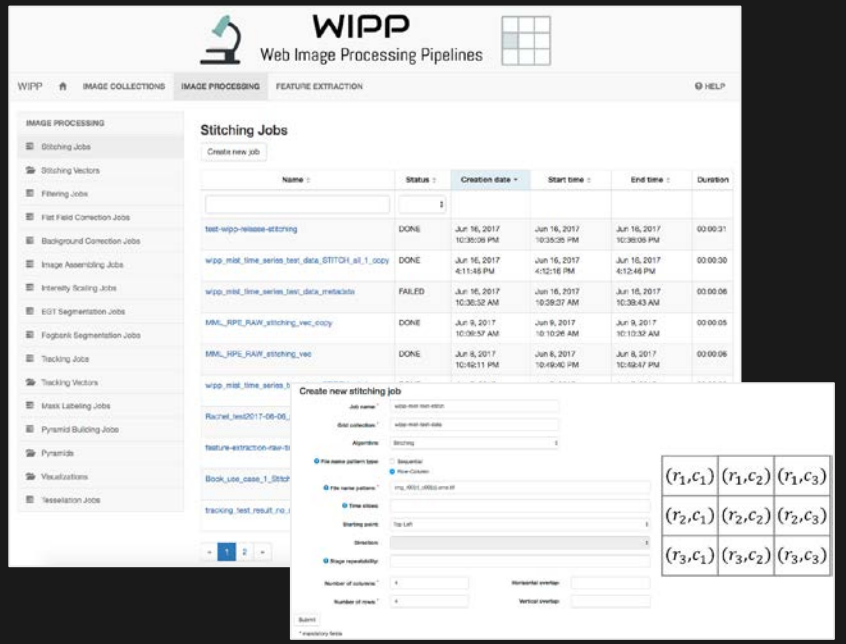




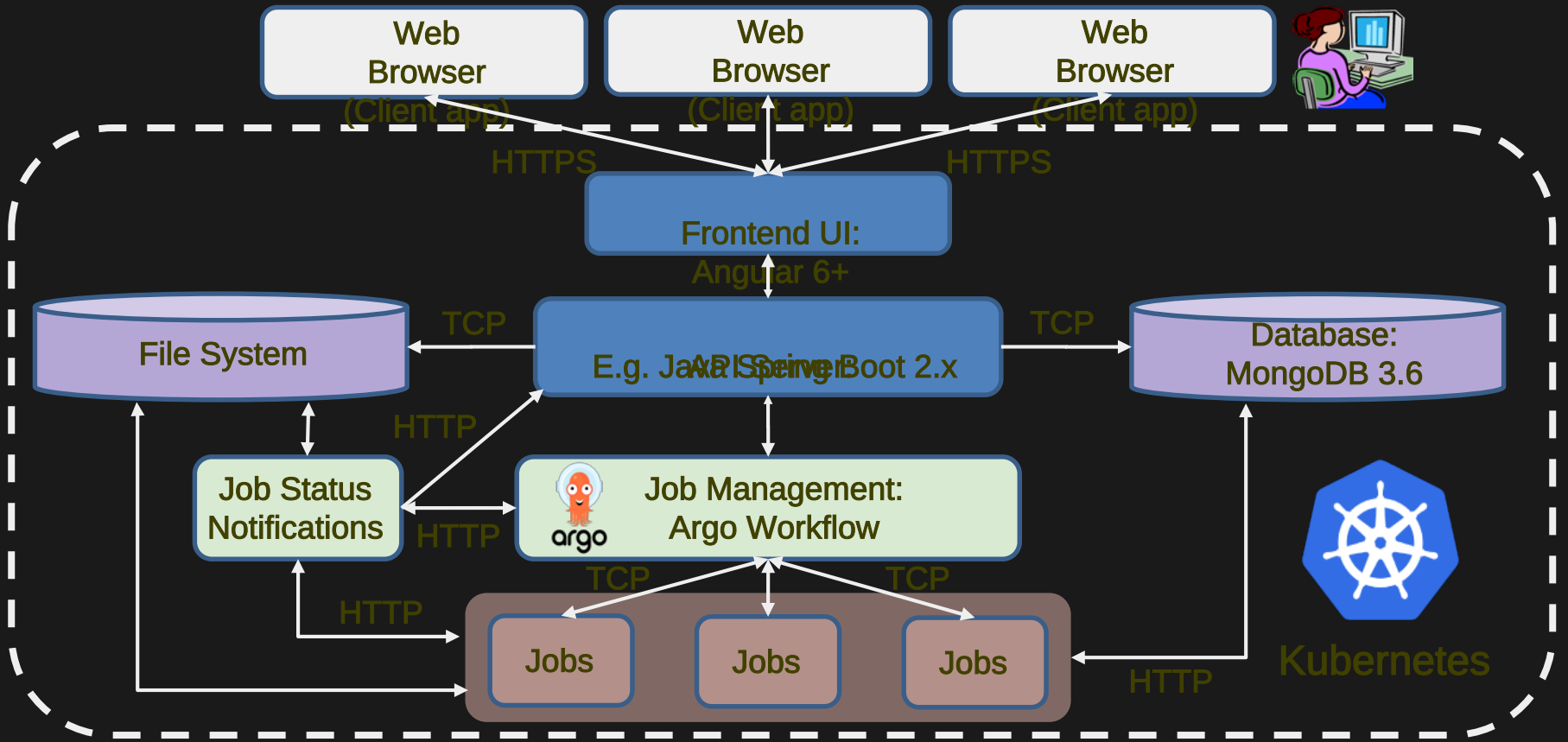


WIPP: Client-Server System - Image Processing + Analysis



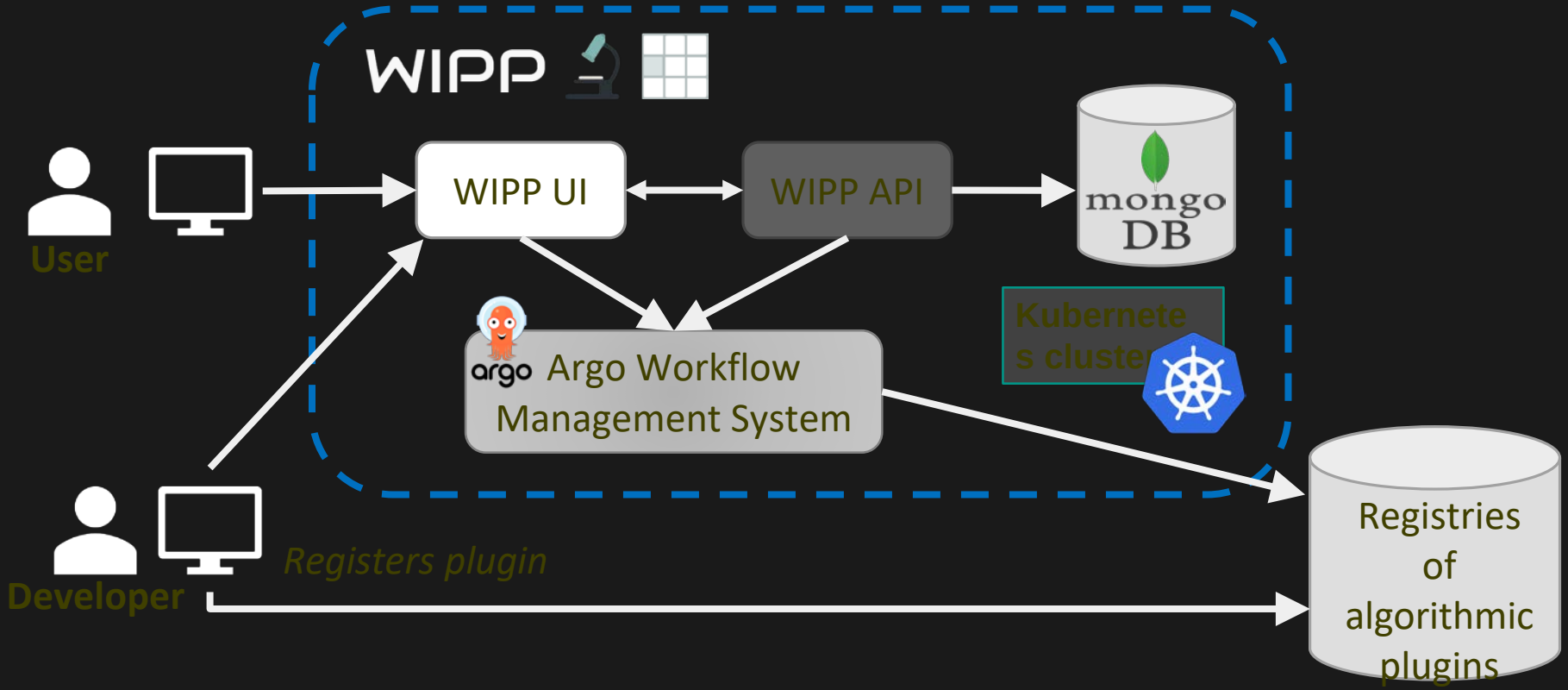


WIPP 3.0 Client-Server Architecture



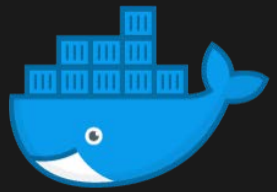
Focused on Functionalities for Users, Developers, and System Administrators

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.”

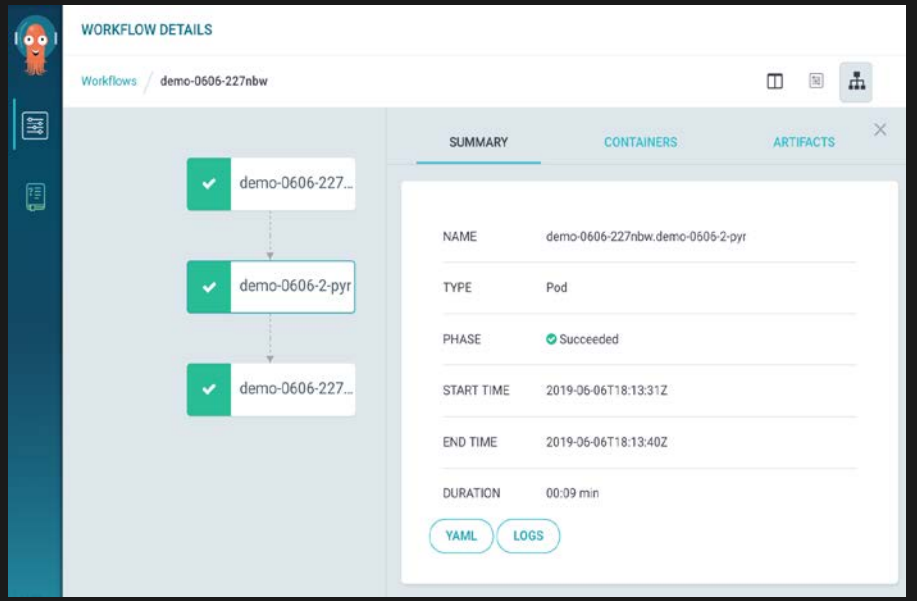
- Kubernetes

“Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. Production-Grade Container Orchestration.”

Container Workflow Technologies

- Argo Workflows

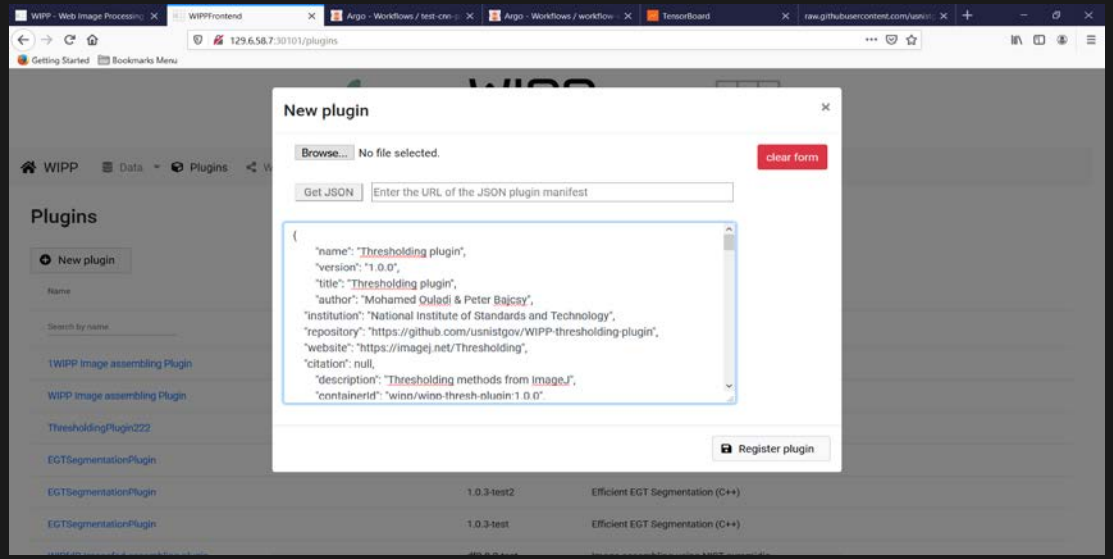
"Argo Workflows is an open source container-native workflow engine for orchestrating parallel jobs on Kubernetes."

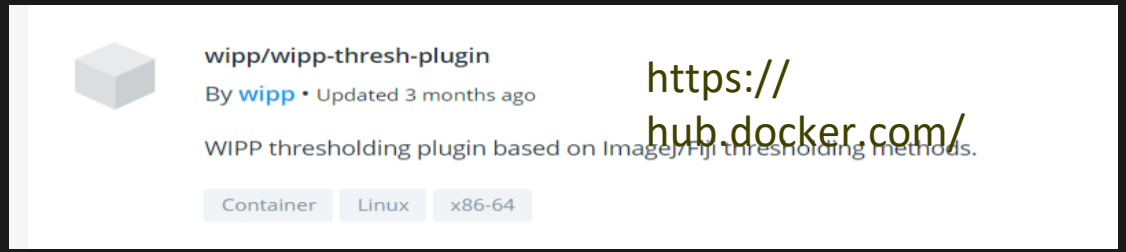


Algorithmic Plugins

Plugin Components

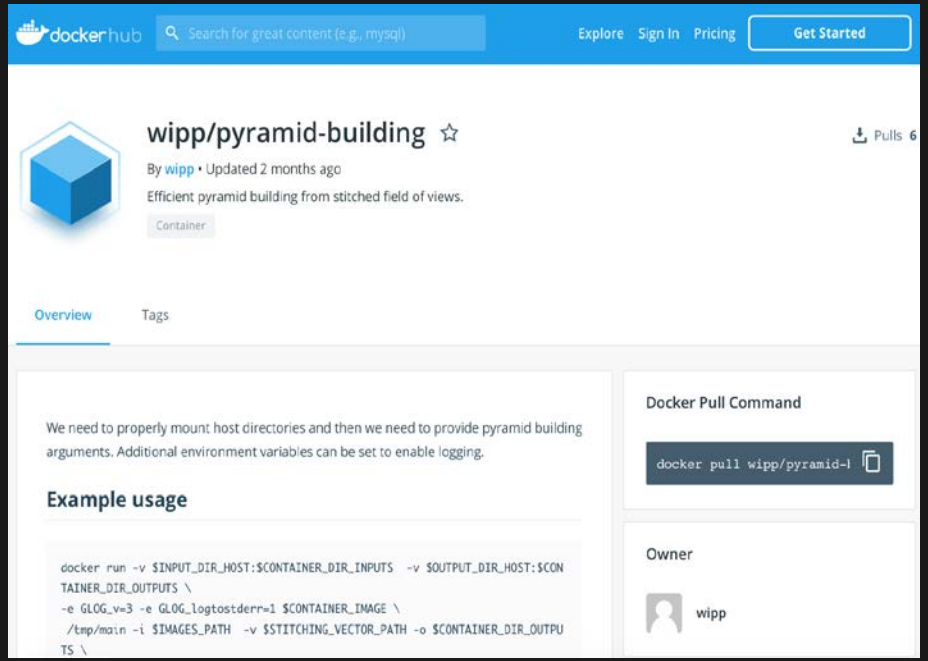
- JSON Manifest
  - Location of docker file
  - Inputs/Outputs
  - User interface definition
- Docker Container
  - Algorithms
  - Libraries
  - Abstraction Middleware





Plugin Component: Docker Image

WIPP plugin as a Docker image available on DockerHub



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

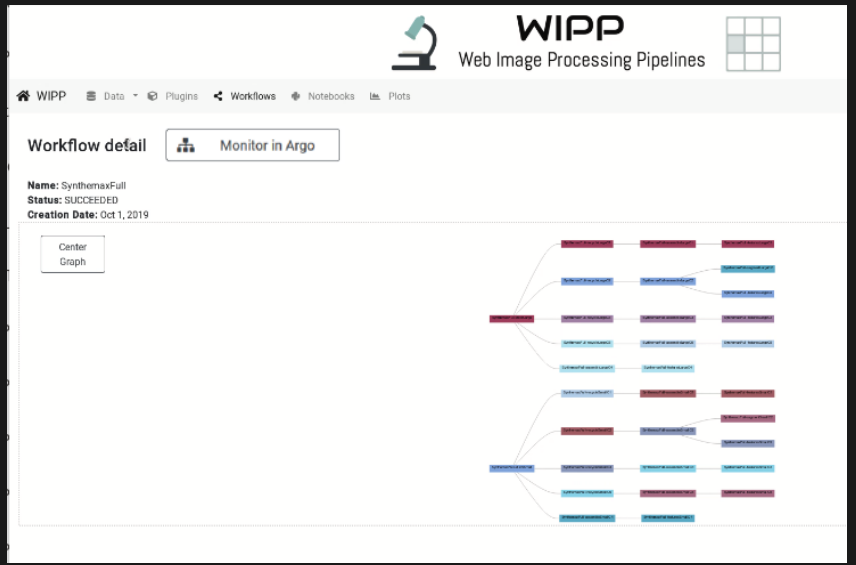
Inputs description
- name, type, description

Outputs description
- name, type, description

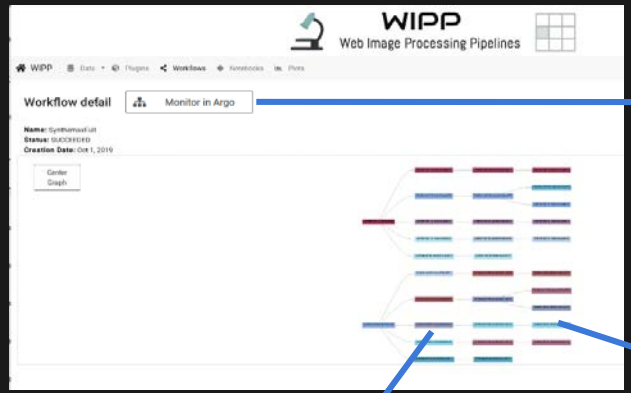
WIPP will use this manifest to generate the job configuration form

Workflow Creation from Algorithmic Plugins

- Interactive workflow creation and visualization
- Configure workflow tasks
- Chain job outputs

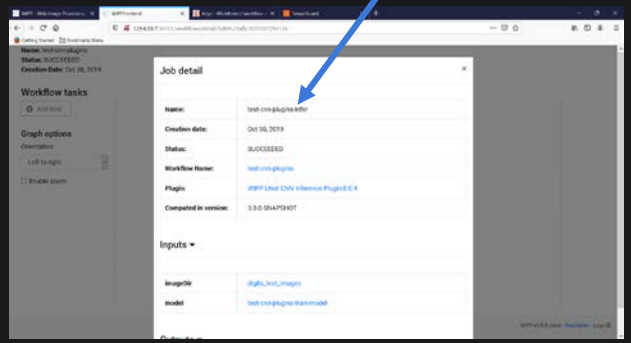


Workflow Monitoring and Traceability



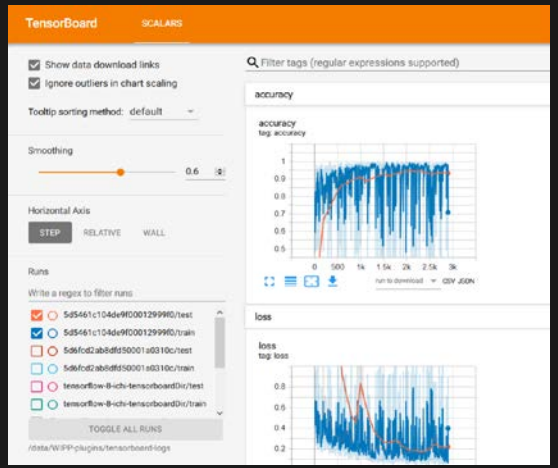
Monitor Execution in Argo

U-Net training
U-Net inference



Traceable Parameters

Monitor Convergence in TensorBoard

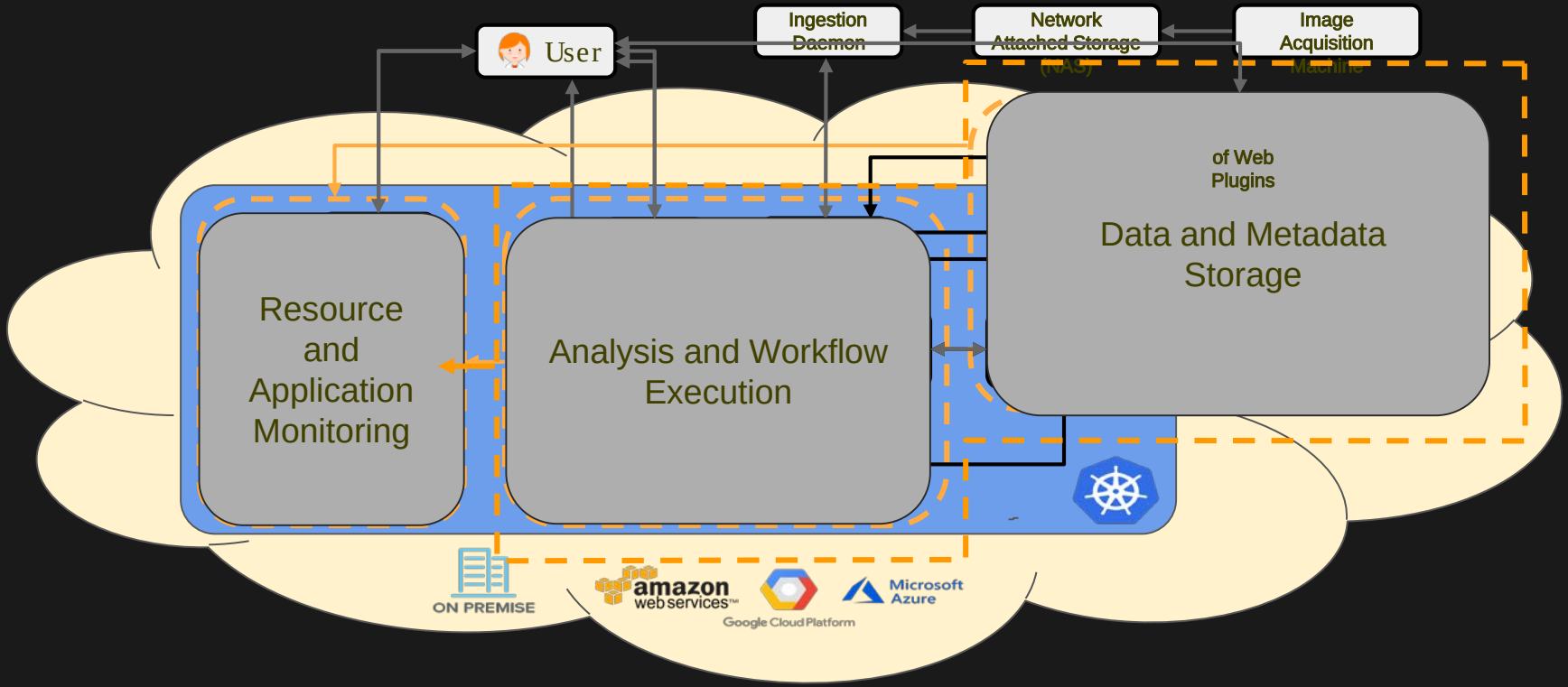


Interoperable Algorithmic Plugins

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

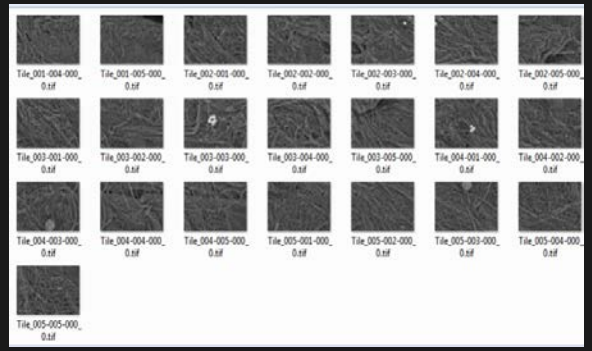
Polus Services: Monitoring, Execution, and Data Storage - Managed Infrastructure!



Plugins and Interaction

Algorithmic Plugins

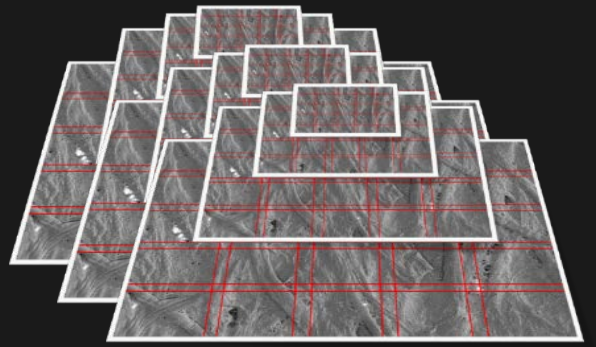
- Image stitching (MIST algorithm)
- Image thresholding (ImageJ/Fiji multiple algorithms)
- Pyramid building (accelerated algorithm with low RAM requirements)
- Universal Notebook Execution (Scalably Execute Jupyter Notebooks as plugins)



Many Overlapping Microscope Fields of View

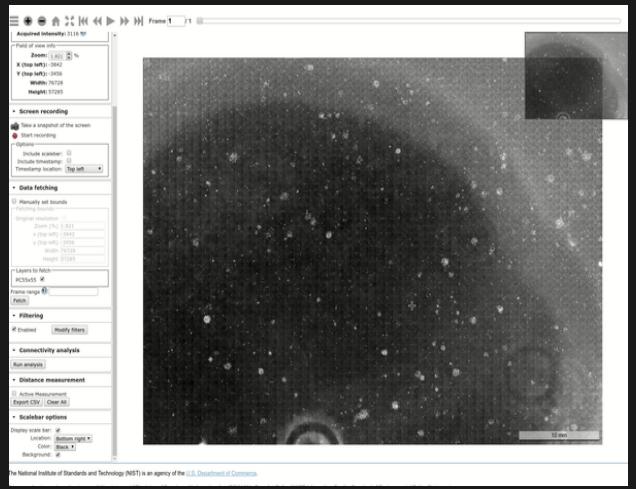
Dynamic Workflow Builder

Image Stitching and Pyramid Building



Stitched Interactive Image Pyramids

Interactive Data-Driven Visual Discovery



Interactive whole sample visualization/manipulation

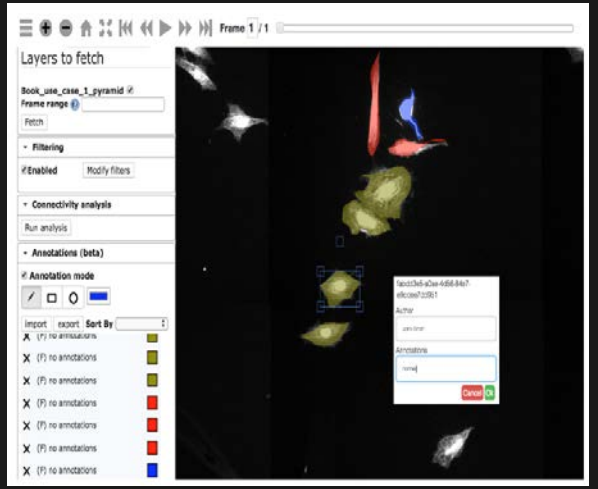


Illustration of WDZT annotation tool

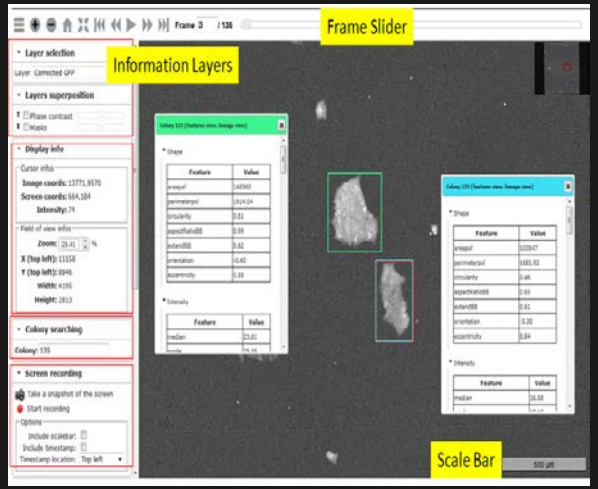
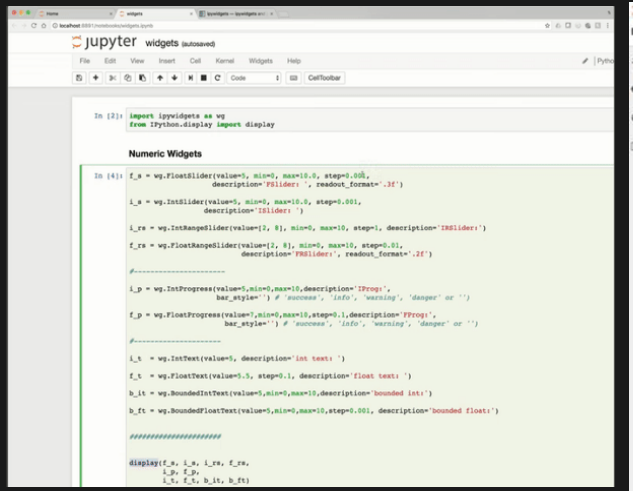
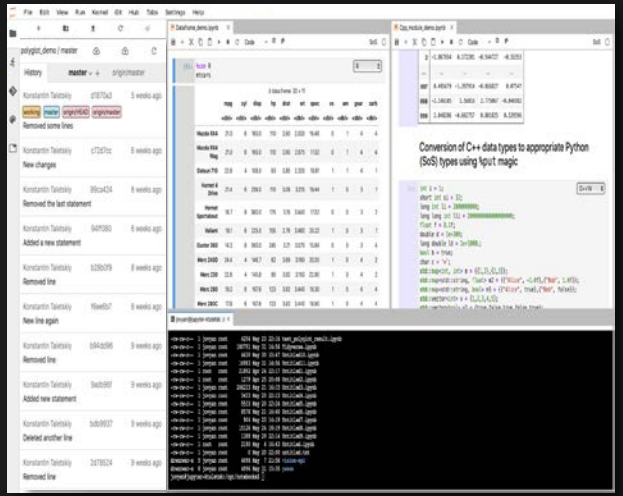


Illustration of linking features and ROIs

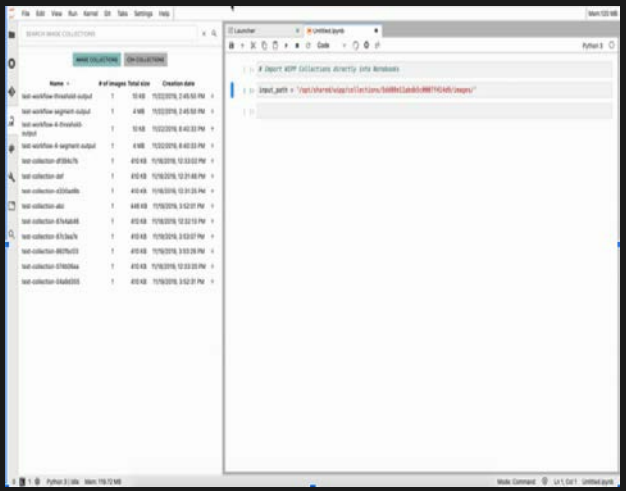
Prototyping Using Jupyter Notebook Interactive Programming



Configurable in-browser Polyglot IDE located where images are stored

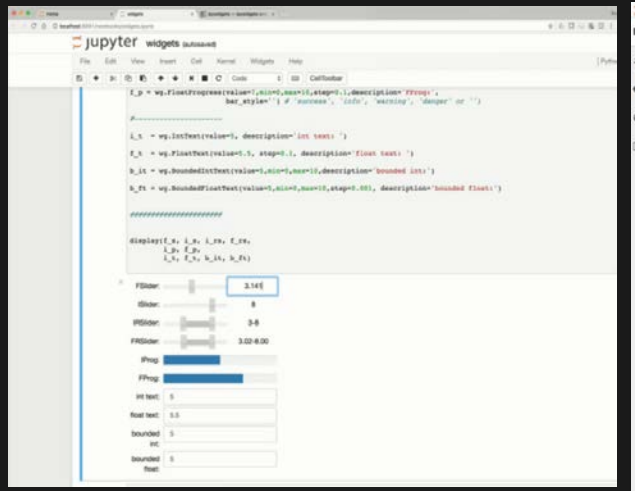


Collaborative Notebooks using GitHub GUI or terminal

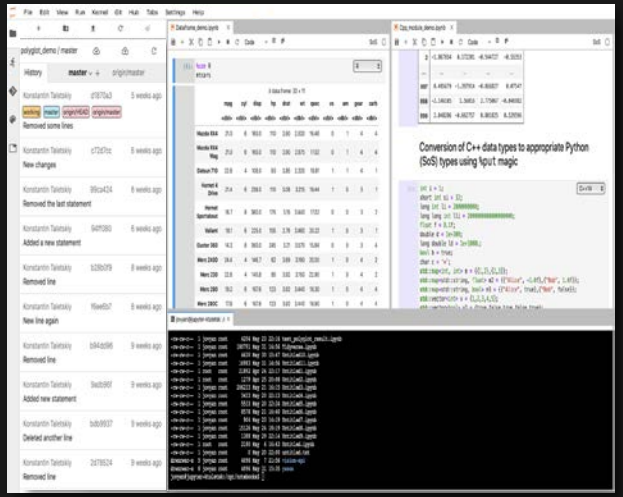


Tight Integration with WIPP Jupyterlab Extension

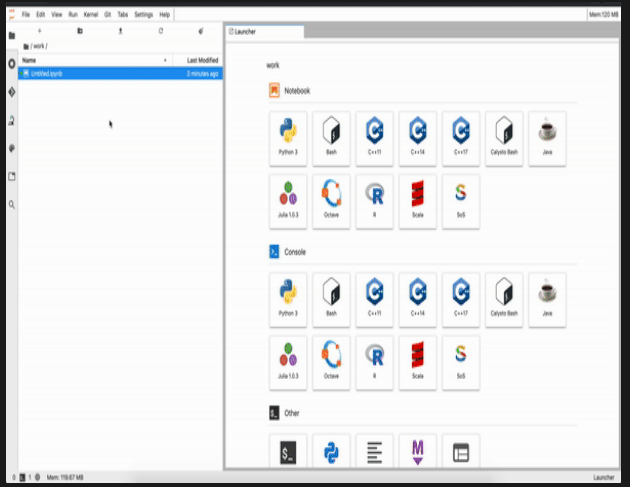
Prototyping Using Jupyter Notebook Interactive Programming



Configurable in-browser Polyglot IDE located where images are stored

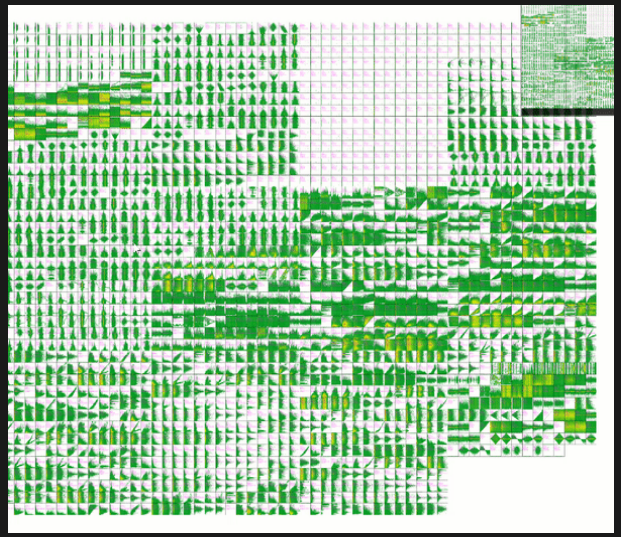


Collaborative Notebooks using GitHub GUI or terminal

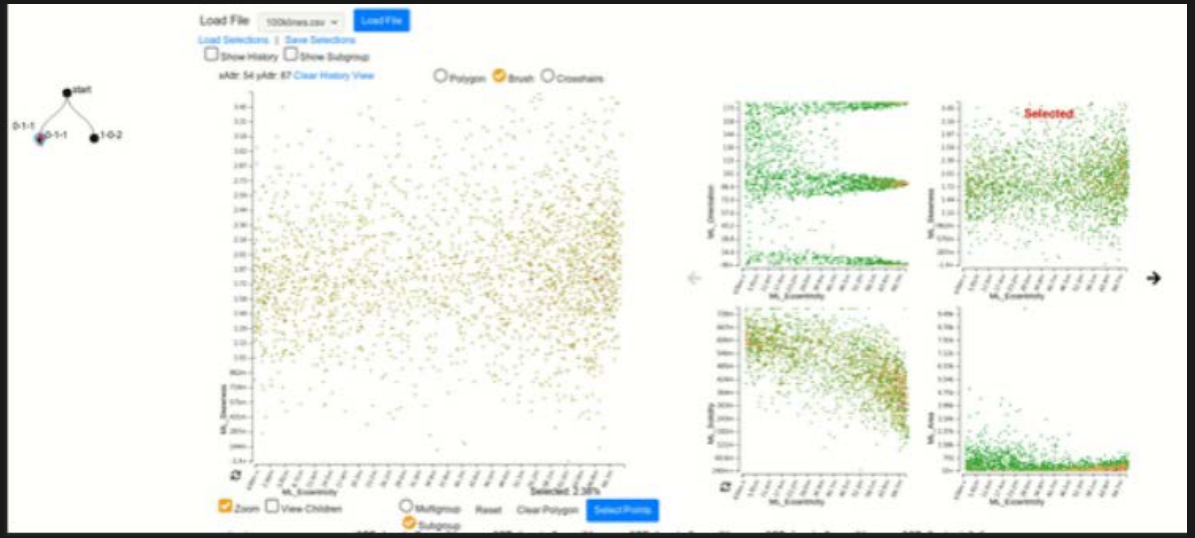


Tight Integration with WIPP Jupyterlab Extension

Scalable Interactive Visualization of Tabular Data

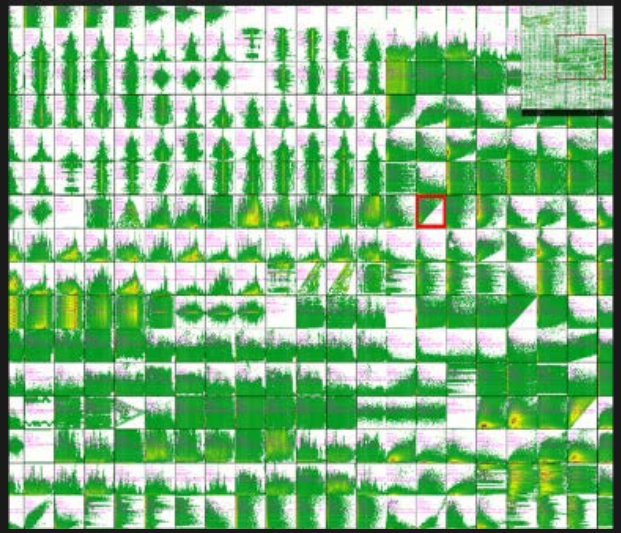


- DeepZoom of all pair-wise feature scatter plots

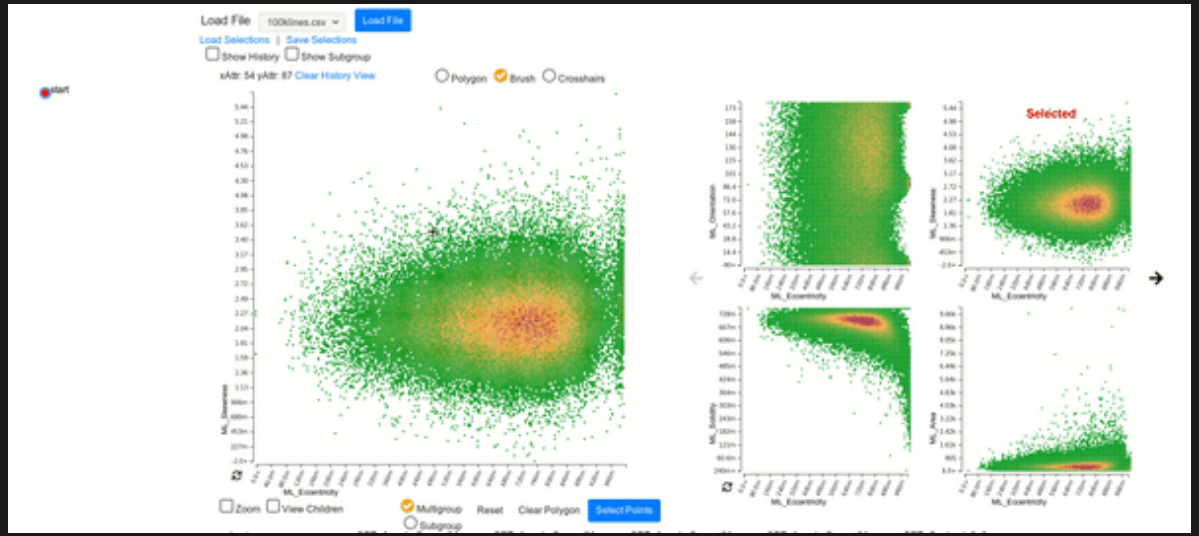


- Interactive data sub-setting across dimensions, zoomed and global views
- Familiar dynamic group selection tools with a history dendrogram and plot view

Scalable Interactive Visualization of Tabular Data



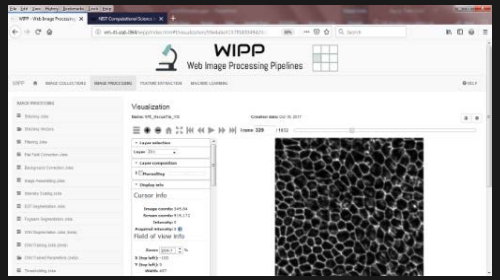
- DeepZoom of all pair-wise feature scatter plots



- Interactive data sub-setting across dimensions, zoomed and global views
- Familiar dynamic group selection tools with a history dendrogram and plot view

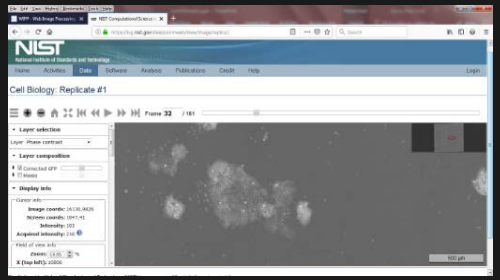
Examples of Data Sharing and Dissemination

- **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

Export

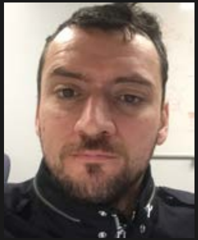


Persistent large size image data

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/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 OS10
- 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.

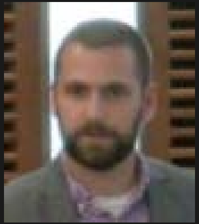
NIH & NIST Development Team





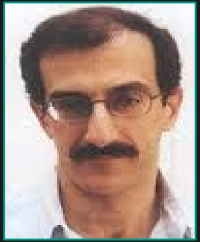


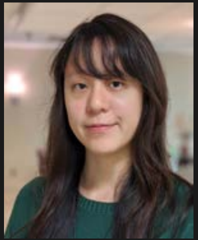










Credit

NIST

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

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

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 Florczyk
- Nick Schaub
- Sumona Sarkar
- John Henry Scott
- Nicholas Ritchie
- Keana Scott
- Charlie Camp
- Young Lee
- Marcus Cicerone

External Collaborators

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

(1) Technical University of Lodz, Poland
(2) NIH National Cancer Institute
(3) University of Maryland at College Park
(4) The Lieber Institute for Brain Development
(5) George Mason University
(6) Axle Informatics
(7) Rockefeller University

Disclaimer

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the products identified are necessarily the best available for the purpose.