Jenkins as a Scientific Data and Image Processing Platform

Ioannis K. Moutsatsos, Ph.D., M.SE.
Novartis Institutes for Biomedical Research
www.novartis.com

June 18, 2014
Life Sciences are Computational Sciences

- Modern life sciences (biomedical research, systems biology) are heavily dependent on
  - Data Management
  - Computational Analysis
  - Computational Modeling

- Modern laboratory technologies and instrumentation generate data that are
  - Big
  - Heterogeneous
  - Complex
Computational Challenges & Opportunities

Scientists

• Face daily challenges by continuing increases in computational complexity
• Focused on the biology and not the compute problem
• Have varying and rapidly changing requirements

Life Sciences Research

• Benefits from computational systems that are
  – Easy to use
  – Fast to implement
  – Flexible
  – Support
    • Collaboration
    • Transparency
    • Automation
    • Reproducible Research
    • Open standards
Talk Outline

• A life sciences computational challenge
  – High Content Image Analysis
    • What is it?

• Jenkins-CI as a scientific data/image processing platform
  – Functionality with standard plugins
  – How Jenkins-CI provided a HP image analysis platform for lab scientists

• Jenkins as a data analytics platform
  – Domain specific analysis and visualization plugins
  – The Jenkins pros and cons
    • What are we missing?

• Where do we want to take Jenkins?
High Content Screening: HCS
High throughput automated fluorescent microscopy for drug discovery

- **Wet Lab Workflow**
  - Cells grown on high density arrays
  - Cells treated with large number of chemical or biological factors
  - Cells stained with fluorescent antibodies

- **Data Acquisition**
  - Stained cells are imaged in high throughput mode using a computerized microscope

- **Computational Workflow**
  - Cell images processed to extract phenotypic measurements
  - Measurements analyzed to understand factor effects
High Content Screening

• Novartis
  – **High Throughput Biology** (my group)
    • Data from 2010-2013

• Captured
  – *83 Terabytes* of high content image data
    • 17.5 million wells
    • 27 million images
    • ~540 days of imaging time
    • ~1.5 years of computing time
HCS: Workflow and Data Stream

Raw Data
- Images
- Channels
- Fields
- Metadata
  - Acquisition
  - Experiment

Measurements
- Raw (>500 parameters)
  - Aggregated or cell by cell
    - Filtered
  - Metadata
  - Image Processing

Results
- Assay QC
- Hit Identification
- Multi-parametric Statistics
- Correlations
- Machine Learning etc.
HCS-High Performance Image Analysis

Initial Focus: Remove Image Processing Bottleneck

Focus

- Image Processing
  - high throughput
  - accessible to lab scientists
  - integrated in data workflow
  - Monitored
  - Recoverable
HCS: Image Measurements and Analytics

Easily Accessible, High Performance Image Analytics

- **Vision**
  - Image and data analysis using high performance (HP) image processing tools
    - Accessible, scalable, affordable, flexible and well-supported

- **Strategy**
  - Evaluate and adopt open-source, community supported tools
    - CellProfiler, ImageJ, Jenkins-CI
  - Utilize NIBR-IT systems and resources
    - Linux Compute Engine (cluster) / Network Attached Storage
    - Development expertise (UI, data management and web-services)
  - Increase usability of NIBR-IT systems and resources
  - Engage and provide timely and practical functionality to both expert and casual imaging platform users

- **Tactics**
  - Develop functional prototypes (Jenkins-CellProfiler, Test Mosaic, R-Analytics)
  - Collaborate to develop new image/data analysis systems
  - Explore imaging tools and data space. Define HP image processing requirements.
  - Provide training, support and engage in community building
CellProfiler

- Open Source Image Processing
- Platform independent
- Desktop client for defining an arbitrarily complex image processing pipeline
- Pipeline can be used by the command line CellProfiler executable
  - Suitable for high throughput analysis
  - Suitable for deployment on a Linux grid engine
  - Can process large image sets (300K + images)
- Developed and supported by the Broad Institute and a sizable scientific user community
- Supports additional imaging tools (ImageJ)
CellProfiler – general anatomy

Nuclear Translocation Assay

- Pipeline
- Modules
- Add/Subtract Modules
- Module details
HCS Image/Data Processing
Programming and Prototyping Functional Requirements

• Scripting
  – Pros
    • Quick prototyping
    • Flexibility
    • Platform independence
  – Cons
    • Unsuitable for end users
      – Requires installation of scripting tools
      – Command Line Interface

• Scripting for end users
  • Requires a user interface
    – Most UI prototypes are either
      » hard
      » Pretty but not functional
      » Or...

• ...not very pretty
  – But quite functional
Why choose Jenkins-CI?

Why Jenkins-CI?

- Jenkins allows us to rapidly wrap any command line script or program in a web interface
  
  • Excellent support for [Groovy](https://groovy-lang.org) a Java based, dynamic, modern scripting language
  
  • Straight forward integration with other languages, tools, OS, frameworks

- Jenkins has [broad community support](https://plugins.jenkins.io) that provides access to over 800 plugins
  
  • Plugins allow easy customization of Jenkins for a variety of tasks

- Jenkins provides basic workflow and web server functionality
  
  • Which works well in combination with CellProfiler

- Jenkins is used extensively by the NIBR-IT group to build all kinds of internal software
  
  • Many software developers know a lot about Jenkins

- Jenkins is now emerging as a useful Bioinformatics tool
  
  • The [BioUno](https://biouno.org) project
Jenkins-CellProfiler
HP Image Processing Workflow: Outline

Contribute Image Processing Pipeline → Assemble Images, Metadata → CellProfiler HP Image Processing → Retrieve & Use Data

Upload your CellProfiler pipeline
Upload a CellProfiler image processing pipeline from your repository. The pipeline will be available for use on the Jenkins-HK server.

Generate a CellProfiler image list
The source folder must contain one or more subfolders. Use the `Generate Linux Image List Now` button to create the list.

Process images using CellProfiler
The CellProfiler pipeline and the image list are selected. Processing can be restricted to a subset of the plates. Use the `Execute CellProfiler` button to run the pipeline.
Jenkins-HCS Workflow Engine
High Level UI Components
**Typical Workflow**

*Step 1: Contribute a pipeline*

- **Project:** Contribute_Pipeline
  - Upload and annotate a standard CellProfiler image analysis pipeline. Uploaded pipelines are usable in other projects

- **Assumptions**
  - The pipeline has been designed and successfully tested on the CellProfiler desktop client

- **Outcome**
  - The CellProfiler pipeline file will be uploaded and stored on Jenkins
  - Additional annotation will be extracted and attached to the pipeline
Build report from a contributed pipeline

Uses: Summary Display Plugin

- Contributed pipelines are annotated by a combination of user-provided and automatically extracted metadata
  - Presented as a tab panel
  - Pipeline can be downloaded and further modified

Use PIPELINE FILE tab to download or quickly browse the pipeline
A CellProfiler Pipeline from Jenkins Server

Additional Usage

- CellProfiler pipelines on the Jenkins server can be used as follows:
  - For inspection
  - For re-use:
    - On CellProfiler desktop client
    - On Jenkins-CellProfiler
  - For further experimentation:
    - Load in desktop client and further customize
Typical Workflow

Step 3: Execute CellProfiler on the Linux Cluster

- **Project:** CellProfiler_JClustSelect
  - Executes a series of image processing steps using the Jenkins-CI CellProfiler
  - Uses the SSH Plugin

- **Typical Assumptions**
  - CellProfiler pipeline and a CP formatted image list are stored on the Jenkins server
    - Jenkins build artifacts

- **Outcome**
  - Summary report
  - A file containing combined measurements from all the images processed.
    - Results file is in CSV format
Monitoring CellProfiler runs on the cluster

Uses: Build Pipeline Plugin

Users switch to the graphical review of the workflow!
Monitoring CellProfiler runs on the cluster

Uses: Build Pipeline Plugin and the Console
Run Report & Measurement Retrieval

Uses: Associated Files and HTML Publisher plugins

If all goes well, final results are found in the merged measurements folder.
Jenkins –CI: CellProfiler Image Processing

Uses: HTML Publisher plugin

20130430  2 IC50


<table>
<thead>
<tr>
<th>Build Parameters</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>CellProfiler Pipeline</td>
<td>Review</td>
</tr>
<tr>
<td>Total Source Images</td>
<td>69120</td>
</tr>
<tr>
<td>Mapped Image List Parameters</td>
<td>Review</td>
</tr>
<tr>
<td>Measurements Folder</td>
<td>\incell\HTBJENKINS\JCP_2013-04-30_11-58-03</td>
</tr>
<tr>
<td>Merged Data Folder</td>
<td>\OUTPUT\CELLPROFILER\2013-04-30_11-58-03\ALL</td>
</tr>
<tr>
<td>Progress Monitor</td>
<td>Progress Monitor</td>
</tr>
</tbody>
</table>

Analysis Provenance

Links to Data

Visual QC Feedback (green all is good!)
Build report of a visual QC Jenkins pipeline
Advanced/Experimental Functionality

Exploring the parameter space (a.k.a. Test Mosaic)

- Optimization of imaging module parameters
  - A typical pipeline development requirement
- Test Mosaic
  - Allows systematic and documented exploration of the parameter space
  - Evaluation is based on visual and quantitative interpretation of the results
HCS-Multi-Parametric Data Analysis

Current Focus: Prototype powerful and easy to use analytics

- Jenkins
- Perl
- R

Raw readout files

- QC for missing values, columns
- Generate data file for R
  - Plate level QC
    - DMSO boxplot
      - Plate heatmap plot
    - Distribution plot of CellCount of DMSO wells
    - ZeroVariance
      - Ftr correlation
        - linearCombination
    - Feature selection
    - Plate correction
      - LOESS correction plot
  - DMSO CellCount QC
    - Which CellCount threshold?
      - Features excluded will not be deleted, just masked
    - Which features to include in analysis?
    - Plate normalization
      - Normalization plot
    - Feature correlation
      - Correlation plot

One file per plate, rows are well/cols, columns are "Barcode", "Row", "Column", "WellType", features, with exact name and order (also which column is "CellCount")

Plates, Features, Annotations

Boxplot for DMSO wells only for all features. Plate heatmap for randomly selected N features + CellCount.

Doesn't apply to positive controls; doesn't apply to plates expected to have pattern (e.g. DRC)

Contributed by Stanley Lazic
Statistics, Visualization, Reporting
My current Jenkins toolkit

- **Jenkins R-Plugin**
  - Supplies build step for executing R scripts
    - This plug-in was created by the BioUno project (sponsored by TupiLabs), and released to Jenkins as well.

- **Image Gallery Plugin**
  - This plug-in reads a job workspace and collects images to produce an image gallery
  - Useful for visualizing various statistical plots and graphs
    - This plug-in was created by the BioUno project (sponsored by TupiLabs), and released to Jenkins as well.

- **Reporting Plugins**
  - HTML Publisher, Summary Display
Jenkins for Interactive Analytics
Using R in a Jenkins pipeline interactively

• Opportunities
  – Quickly prototype functional analysis for multi-parametric data
    • Improve analysis requirements
    • Experiment with required data management and analysis workflows
  – Provide lab scientists with an easy to use, yet sophisticated, standardized and validated platform for MP data analysis tools
Jenkins for Interactive Analytics

*Using R in a Jenkins pipeline interactively*

**Challenges**

- Limitations of the Jenkins user interface
  - Limited interaction between UI controls
- Large and varied HC measurement metadata
  - A challenge for creating HC data schemata as well

**Strategies**

- Open source collaboration with BioUno project
  - Uno-choice UI control greatly facilitates dynamic updating of the UI
- Initial design supports flexible (but still controlled) data schema
  - Low tech, cumulative, shared key-value Java properties
Analytical Builds

A build may create a new transform of the data or simply add metadata.

Workflow Requirement

Ability to select ad hoc Artifacts (data, metadata, results) from previous project builds.
The Uno-Choice plugin

- Provides a list of dynamically generated options
  - Driven by a Groovy script
  - Single/Multi-select (Check Boxes, Radio Buttons)
  - References one or more other UI parameters
  - Dynamically refreshes when referenced UI parameters change
The Uno-Choice plugin

- Provides **reference parameters**
  - Dynamically rendered in the UI but not used in the build
  - Rendered as lists, ‘free-form’ HTML, or an image gallery

![Uno-Choice plugin interface](image)
Soon to be released Jenkins-HCS Analytics
Introducing Jenkins to Life Sciences!

Let’s start by explaining away ‘artifacts’!

1. any object made by human beings, especially with a view to subsequent use
2. a handmade object, as a tool, or the remains of one, as a shard or pottery, characteristic of an earlier time or cultural stage, especially such an object found at an archaeological excavation.
3. any mass-produced, usually inexpensive object reflecting contemporary society or popular culture: artifacts of the pop rock generation.
4. a substance or structure not naturally present in the matter being observed but formed by artificial means, as during preparation of a microscope slide.
5. a spurious observation or result arising from preparatory or investigative procedures.

http://dictionary.reference.com/browse/artifact

Impedance Mismatch!

Developer

Scientist
Introducing Jenkins to Life Sciences
Let’s improve the User Interface/Experience

- Let’s start by improving the default Jenkins UI
  - Layout
  - Navigation
  - Refreshing
  - Interactivity
- This is an active Jenkins community discussion
What We are Missing
Configuration Explorer

• Structured
• Graphical
• Dynamic
What We are Missing
Bi-Directional Build Interaction

- Build A uses artifacts of Build B
  - Limited support by Run Type parameter
    - Missing flexible and dynamic filtering

- Build D modifies build/publisher artifacts of Build C
  - Sometimes not do-able
  - Sometimes requires a reload

Build C produces an intermediate report that will get updated once Build D is finished successfully.

Build D monitors output of long running job and updates report of Build C.

‘Progress Monitor’ Link and cell color are updated
What We are Missing
A good, deep search and metadata framework

<table>
<thead>
<tr>
<th>Supported</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Searches</td>
<td>Build Searching</td>
</tr>
<tr>
<td>Build Browsing</td>
<td>Parameter Search</td>
</tr>
<tr>
<td></td>
<td>Metadata Search</td>
</tr>
<tr>
<td></td>
<td>– Metadata plugin</td>
</tr>
<tr>
<td></td>
<td>(currently limited to</td>
</tr>
<tr>
<td></td>
<td>adding metadata at</td>
</tr>
<tr>
<td></td>
<td>project level)</td>
</tr>
<tr>
<td></td>
<td>Artifact Search</td>
</tr>
<tr>
<td></td>
<td>Tagging</td>
</tr>
<tr>
<td></td>
<td>Dynamic Metadata</td>
</tr>
</tbody>
</table>

- By timeline
- By view
- By user
The BioUno project is filling the gap

Interested in plugins that

- Integrate bio-informatic, statistical and visualization tools
- Connect to life-science data repositories
- Generate artifacts and reports in LifeSci formats
In Summary

- We have demonstrated that Jenkins-CI can be used for life-science applications
  - Using standard functionality
  - Using domain specific plugins
  - In demanding environments of big data and high performance
- We have observed that scientist are able and willing to use the platform despite it’s ‘domain impedance mismatch’
- There is some fundamental interest in the larger Jenkins-CI community to expand the boundaries of the framework beyond continuous integration
Where do we want to take Jenkins-CI?

- Discussion
  - No changes?
  - Gradual improvements?
    - User interface
    - API
    - New life-science plugins
  - Fundamental changes?
  - Integration framework for orchestrating more granular pipelines?
    - CellProfiler
    - Galaxy
    - Knime
    - Others?
Acknowledgments

• Novartis
  – Fred Harbinski
  – Christian Parker
  – Stanley Lazic
  – Imtiaz Hossain
  – Josh Snyder
  – Erik Sassaman

• BioUno
  – Bruno Kinoshita

• The Jenkins Community
Thank You To Our Sponsors

Platinum

CloudBees

Gold

black diamond software
MidVision™
Release the innovation
choose digital
JFrog
XebiaLabs
the deployment automation company

Silver

SOASTA
Test Faster. Release Sooner
Kensho
SERENA
assembla
ELIASSEN GROUP
Agile Consulting Services