Galaxy for MS software dissemination: How to easily publish your tools

ASMS Workshop
Tuesday, June 5, 2018

Galaxy-P research team
Tim Griffin
Pratik Jagtap
Praveen Kumar
Caleb Easterly

Objectives for workshop

- Learn about the Galaxy platform: Why use it?
- Learn about process for tool development and implementation in Galaxy
- Learn about community resources for help and participation
- Ask questions and discuss

What is Galaxy?

Goecks, J, Nekrutenko, A, Taylor, J and The Galaxy Team.. ***Genome Biol.*** 2010, **11**: R86.

- A web-based, community developed bioinformatics framework/platform/workbench
- Originally designed to address issues in *genomic* informatics
- **In a nutshell:** Galaxy provides an open framework into which disparate software programs can be deployed, integrated into customized, automated workflows for typical to advanced applications, which can be shared in their entirety with other users

Integration

~~Reinvention~~

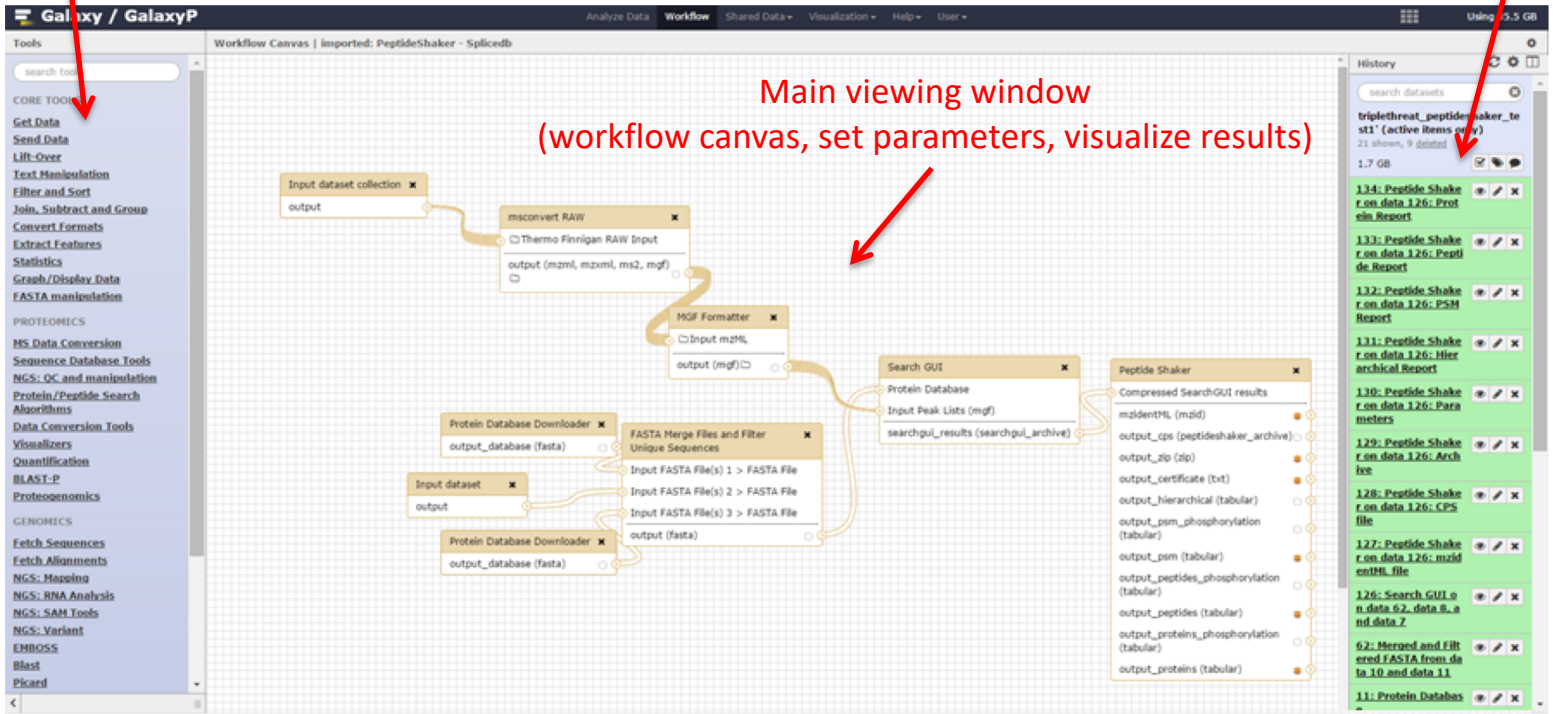
- *Flexible, accessible, scalable, community-minded (developers/users)*

Operation

Tool menu

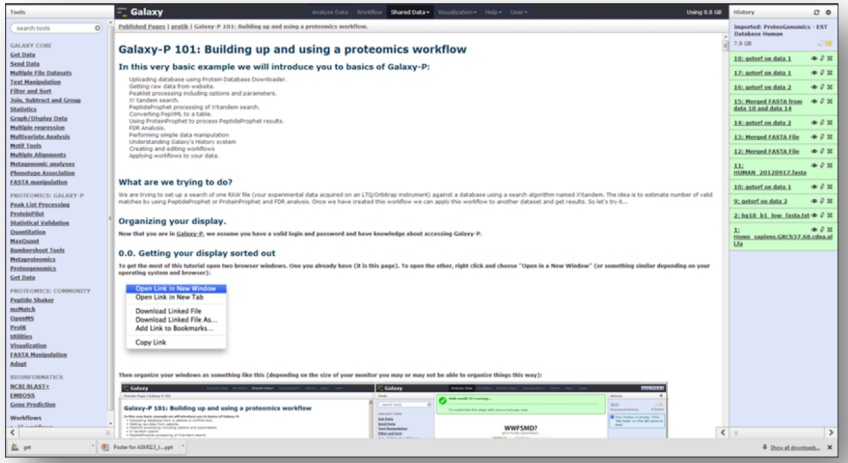
History

Main viewing window
(workflow canvas, set parameters, visualize results)



- Integration of tools into sophisticated workflows
- Provenance tracking (all analyses are saved in completeness, including results and tool parameters)
- Visualizations via plug-ins, web-based queries through APIs

Promoting reproducibility, transparency and dissemination



URL export

File export (.ga)

HISTORY: https://galaxyp.msi.umn.edu/u/pjagtap/h/itraq-search-yang-2-xtandem-scaffold

WORKFLOW: https://galaxyp.msi.umn.edu/u/pjagtap/w/workflow-for-4-plex-itraq-xtandem-search-scaffold-processing

Galaxy-Workflow-Workflow_for_4-plex_iTRAQ_X_tandem_Search_Scaffold_Processing.**ga**

Labbies

Professor

Community-minded

- Many resources for users and developers:
  - Galaxy Training Network (GTN)
  - usegalaxy.org
  - galaxyp.org
  - Annual Galaxy community conference (June 25-28, Portland, OR)

- Tool development, publishing and sharing:
  - Tool Shed https://toolshed.g2.bx.psu.edu/
  - Github for proteomics: https://github.com/galaxyproteomics

Deposit tool in GalaxyP Github → automated testing → Tool Shed deposition

Galaxy Training Network (GTN)

https://galaxyproject.org/teach/gtn/

Galaxy COMMUNITY HUB | Use | Community | Education | Deploy & Develop | Support | Search Galaxy | Edit

Galaxy Training Network

- Training Resources Directory
- Trainer Directory
- Compute Platform Advice
- Best Practices
- Mailing List
- Events
- Interested?

The Galaxy Training Network (GTN) is a network of people and groups that present Galaxy and Galaxy-based training around the world. GTN helps researchers find online training material, and trainers in their geographic area, and helps advertise training events as well.

These pages describes the Galaxy Training Network, and everything else related to teaching Galaxy or teaching Bioinformatics with Galaxy:

Training Resources Directory

OPEN CHAT

- **Galaxy Interface**
- **Inputs / Datatypes**
- **Tools**
- **Workflow**
- **Jobrunner**
- **Summary**

Galaxy Interface

galaxyp-proteogenomics.duckdns.org

Galaxy / Omicron

Analyze Data | Workflow | Shared Data | Visualization | Help | User | Using 4.0 GB

Tools

search tools

Get Data
Send Data
Collection Operations
Text Manipulation
Filter and Sort
Join, Subtract and Group
Convert Formats
Extract Features
Fetch Sequences
Fetch Alignments
Statistics
Graph/Display Data
Text Manipulation
FASTX
Proteogenomics
NGS: QC
NGS: Mapping
NGS: Variant calling
NGS: SAM/BAM tools
NGS: VCF manipulation
Operate on Genomic Intervals
NGS: RNA Tools
GATK
NCBI BLAST+
BED Tools

Workflows

- All workflows

tools

An accessible proteogenomics informatics resource for cancer researchers

Instructions for accessing workflows and training material

This page provides links to instructional materials describing how to access and use the proteogenomic workflows described in the manuscript text.

Please access the links indicated below for the following materials:

Training on usegalaxyp.org

- How to access and use the usegalaxyp.org training site: http://z.umn.edu/accessusegalaxyporg

Interactive Galaxy Tours

- Accessing an interactive Galaxy Tour describing the RNA-seq-to-protein-database workflow: http://z.umn.edu/rnaseqtoproteindbtour
- Accessing an interactive Galaxy Tour describing Dataset Collections: http://z.umn.edu/datasetcollectiontour

History

search datasets

imported: ABRF_test_040918
38 shown, 46 hidden

2.2 GB

85: Final_Summary_Novel_Peptides_Output
84: PepPointer on data 83
83: Peptide Genomic Coordinate on data 37, data 72, and data 82
82: Novel_Peptides
81: PSM_Novel_Peptides
80: BlastP_output
78: Peptides_for_Blast-P_analysis
72: mz to sqlite on data 34, data 7, and others
67: PS_PSMreport
64: PS_mzidentml
61: query results on data 43 and data 56
60: query results on data 36 and data 56
59: query results on data 36 and data 56
57: PSM_Novel_Peptides
56: BlastP_output
54: Peptides_for_Blast-P_analysis
48: mz to sqlite on data 34, data 7, and others
43: PS_PSMreport

Inputs / Datatypes

- **Galaxy is designed to work with many different datatypes.**
- **Upon file upload, datatype can be detected and assigned (when possible) or user specified (before or after load).**
- ***It is important to note that many tools will only accept as input datasets with the appropriate datatype assigned.***

RAW **MGF** **mzidentML**

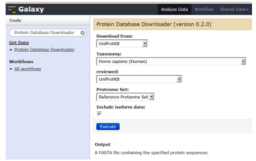
https://galaxyproject.org/learn/datatypes/

Tools

tools

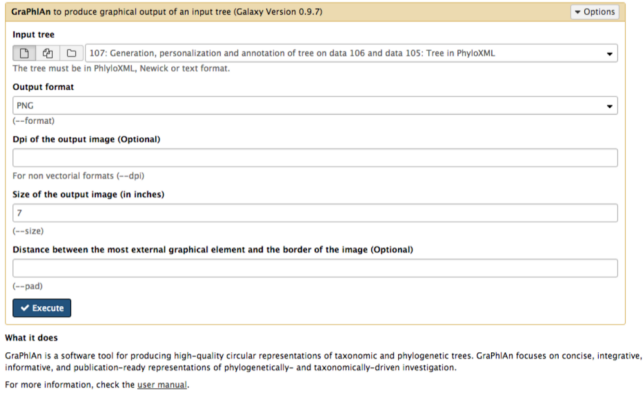
Galaxy-P has multiple application tools – some that are proteomics application specific and others from the genomics Galaxy framework.

For example, Protein Database Downloader downloads UniProt protein FASTA databases of various organisms.



- ***msconvert***
- ***SearchGUI***
- ***PeptideShaker***

Galaxy tool / wrapper




```
graphlan.py --format "png" --size 7 input_tree.txt png_image.png
```

Workflow

Software tools can be used in a sequential manner to generate analytical workflows that can be reused, shared and creatively modified.

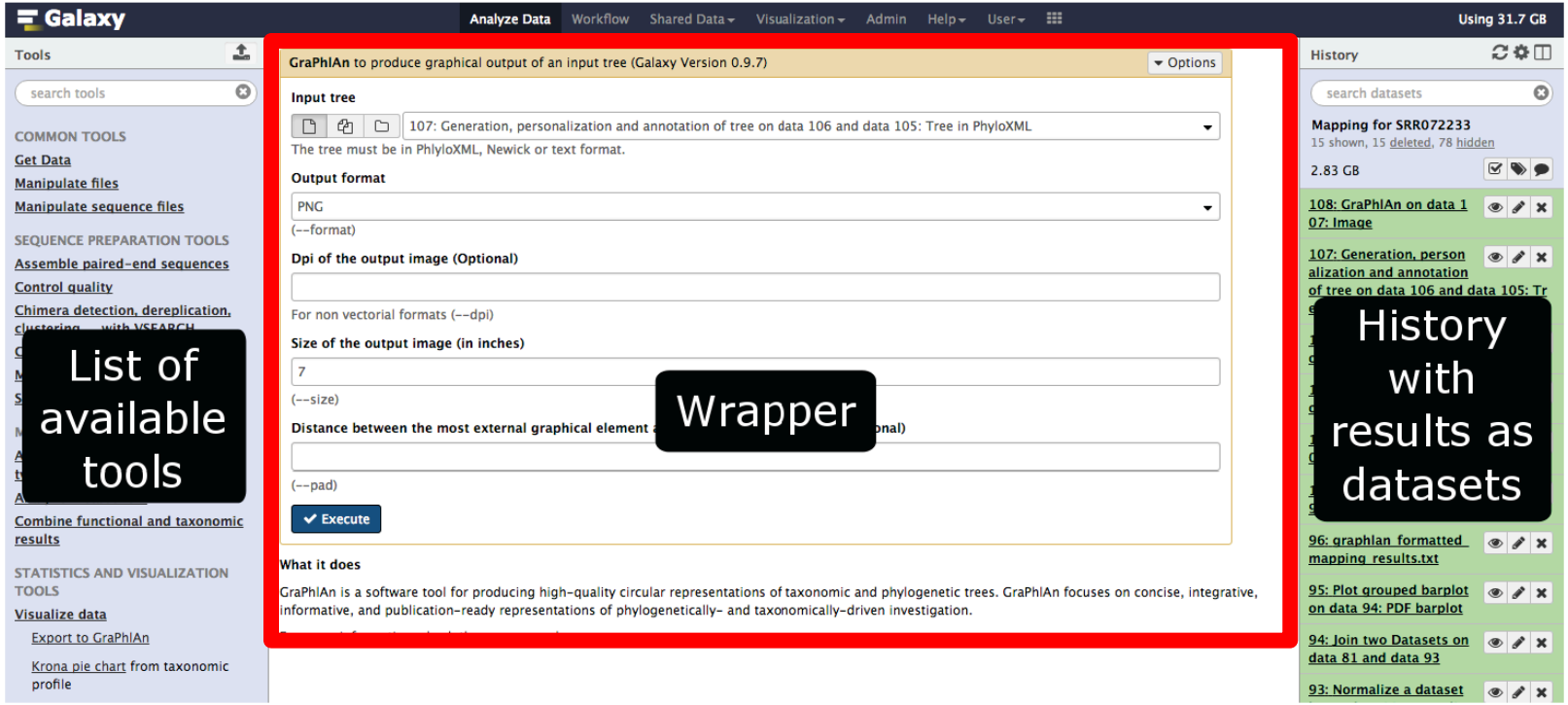
RAW —*msconvert*→ **MGF** —*SearchGUI PeptideShaker*→ **mzidentML**

Jobrunner

Job runners can target appropriate compute environments (Large Memory, multiple nodes, windows software, etc.)

RAW —*msconvert*, run job on windows server→ **MGF** —*SearchGUI* *PeptideShaker*, run job on linux cluster→ **mzidentML**

A tool in Galaxy



Command Line Interface (CLI)

- `blastp -query input_file -db database_name -out output_file`
- `graphlan.py --format "png" --size 7 input_tree.txt png_image.png`
- `head -100 input_file > output_file`
- `echo “Hello World! Some more string” > output_file`

Executable code/program

Input file(s)

Output file(s)

Option(s)/Parameter(s)

CLI to GUI

```
graphlan.py --format "png" --size 7 input_tree.txt png_image.png
```

GraPhlAn to produce graphical output of an input tree (Galaxy Version 0.9.7)

Options

Input tree

107: Generation, personalization and annotation of tree on data 106 and data 105: Tree in PhyloXML

The tree must be in PhlyloXML, Newick or text format.

Output format

PNG

(--format)

Dpi of the output image (Optional)

For non vectorial formats (--dpi)

Size of the output image (in inches)

7

(--size)

Distance between the most external graphical element and the border of the image (Optional)

(--pad)

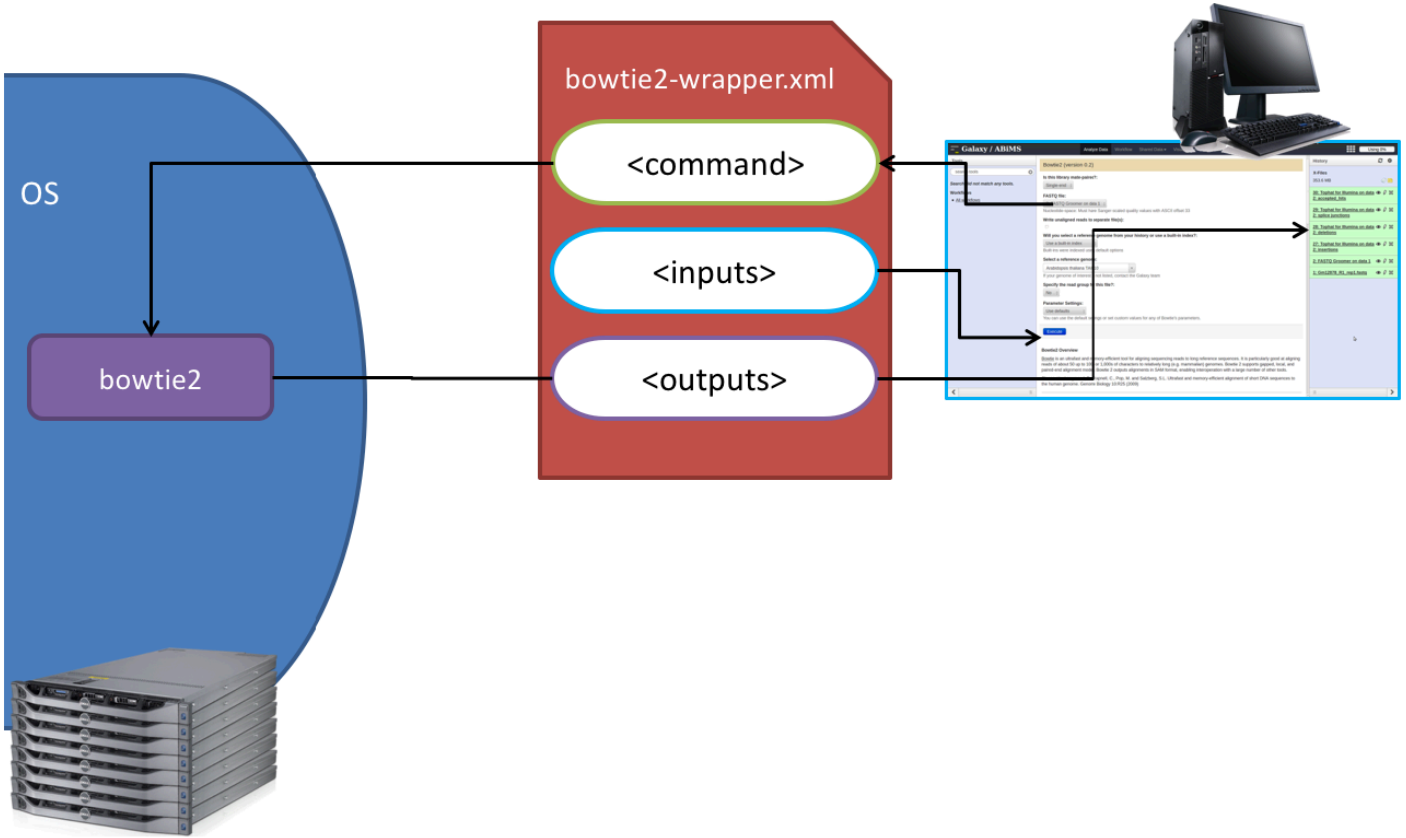
Execute

What it does

GraPhlAn is a software tool for producing high-quality circular representations of taxonomic and phylogenetic trees. GraPhlAn focuses on concise, integrative, informative, and publication-ready representations of phylogenetically- and taxonomically-driven investigation.

For more information, check the user manual.

Wrapper Flow



Wrapper: Key Ingredients

- Description of the user interface
- Link between the underlying tool and the Galaxy instance
- How to invoke the tool
- Which files and options to pass
- Which files the tool will produce as output

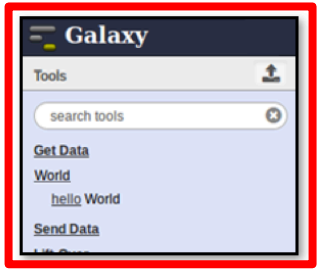
Wrapper: What's in here

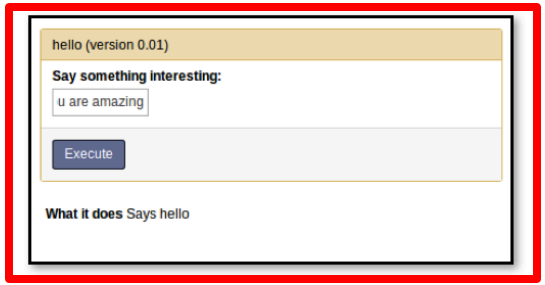
```
<tool id="hello" name="hello" version="0.01">
    <description>World</description>
    <command><![CDATA[
        /bin/echo 'Hello World! $mystring' > '$output1'
    ]]></command>
    <inputs>
```

```
echo "Hello World! Your String" > output_file
```

Wrapper

```
<tool id="hello" name="hello" version="0.01">
    <description>World</description>
    <command><![CDATA[
        /bin/echo 'Hello World! $mystring' > '$output1'
    ]]></command>
    <inputs>
        <param name="mystring" type="text" label="Say something interesting"/>
    </inputs>
    <outputs>
        <data format="tabular" name="output1" label="hello_world"/>
    </outputs>
    <help><![CDATA[
        **What it does**
        Says hello
    ]]></help>
</tool>
```





Wrapper

```
<tool id="hello" name="hello" version="0.01">
    <description>World</description>
    <command><![CDATA[
        /bin/echo 'Hello World! $mystring' > '$output1'
    ]]></command>
    <inputs>
        <param name="mystring" type="text" label="Say something interesting"/>
    </inputs>
    <outputs>
        <data format="tabular" name="output1" label="hello_world"/>
    </outputs>
    <help><![CDATA[
        **What it does**
```

Job Command-Line:

```
/bin/echo 'Hello World You are amazing' > '/usr/local/galaxy/database/files/000/dataset_9.dat'
```

```
</tool>
```

Hello World! You are amazing

Cheetah code in <command>

```
<command><![CDATA[
graphlan.py
--format $format
#if $dpi
    --dpi $dpi
#end if
--size $size
'$input_tree'
]]></command>
```

GraPhlAn to produce graphical output of an input tree (Galaxy Version 0.9.7) Options

Input tree

107: Generation, personalization and annotation of tree on data 106 and data 105: Tree in PhyloXML

The tree must be in PhlyloXML, Newick or text format.

Output format

PNG

(--format)

Dpi of the output image (Optional)

For non vectorial formats (--dpi)

Size of the output image (in inches)

7

(--size)

Distance between the most external graphical element and the border of the image (Optional)

(--pad)

Execute

What it does

GraPhlAn is a software tool for producing high-quality circular representations of taxonomic and phylogenetic trees. GraPhlAn focuses on concise, integrative, informative, and publication-ready representations of phylogenetically- and taxonomically-driven investigation.

For more information, check the user manual.

Find more about Cheetah at: **https://pythonhosted.org/Cheetah/**

inputs > param to command

How to pass the parameters?

```
<inputs>
    <param name="input_tree" type="data" label="..."/>

    <param argument="--dpi" type="integer" optional="true" label="..."
        help="For non vectorial formats" />
</inputs>
```

Directly linked to `<command>`

```
<command><![CDATA[
graphlan.py
...
#if $dpi
    --dpi $dpi
#end if
'$input_tree'
...
]]></command>
```

inputs > param > data

Input tree

107: Generation, personalization and annotation of tree on data 106 and data 105: Tree in PhyloXML

The tree must be in PhlyloXML, Newick or text format.

```
<param name="..." type="data" format="txt" label="..." help="..." />
```

inputs > param > text

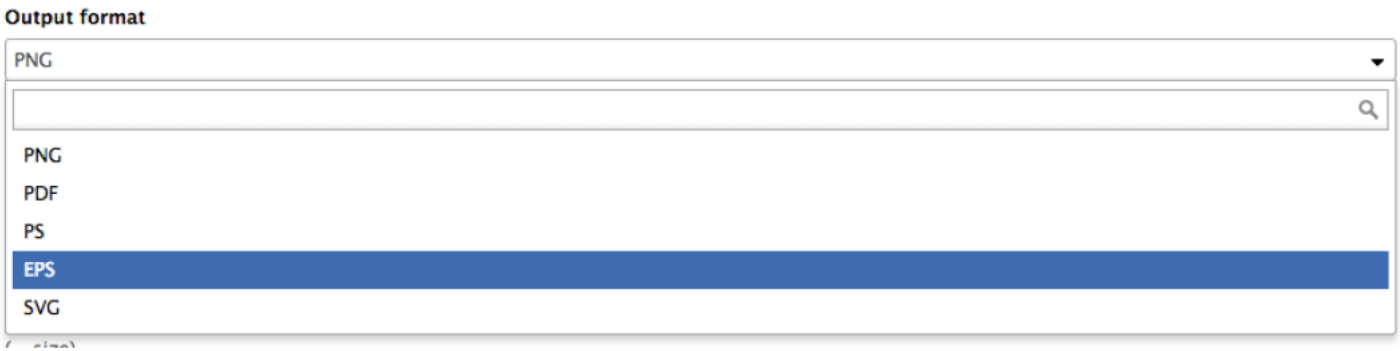
Label for x axis

text

(--xlab)

```
<param name="..." type="text" value="" label="..." help="..."/>
```

inputs > param > select




```
<param name="..." type="select" label="..." help="...">
    <option value="png" selected="true">PNG</option>
    <option value="pdf">PDF</option>
    <option value="ps">PS</option>
    <option value="eps">EPS</option>
    <option value="svg">SVG</option>
</param>
```

If no `option` has `selected="true"`, the first one is selected by default.

inputs > param > select

Type of quality score calculation to use

- (●) Mininum
- ( ) Mean
- ( ) Max
- ( ) Sum

By default, min is used. (-trim_qual_type)

```
<param name="..." type="select" display="radio" label="..." help="...">
    <option value="min" selected="true">Minimum</option>
    <option value="mean">Mean</option>
    <option value="max">Max</option>
    <option value="sum">Sum</option>
</param>
```

inputs > param > select

Which statistics should be calculated included in the graph_data file

☑ Select/Unselect all

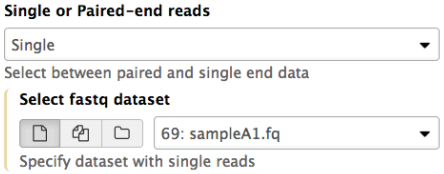
× Length distribution × GC content distribution × Base quality distribution × Occurence of N × Poly-A/T tails × Tag sequence check
× Assembly quality measure × Sequence duplication - exact only × Sequence duplication - exact + 5'/3' × Sequence complexity
× Dinucleotide odds ratios, includes the PCA plots

(-graph_stats)

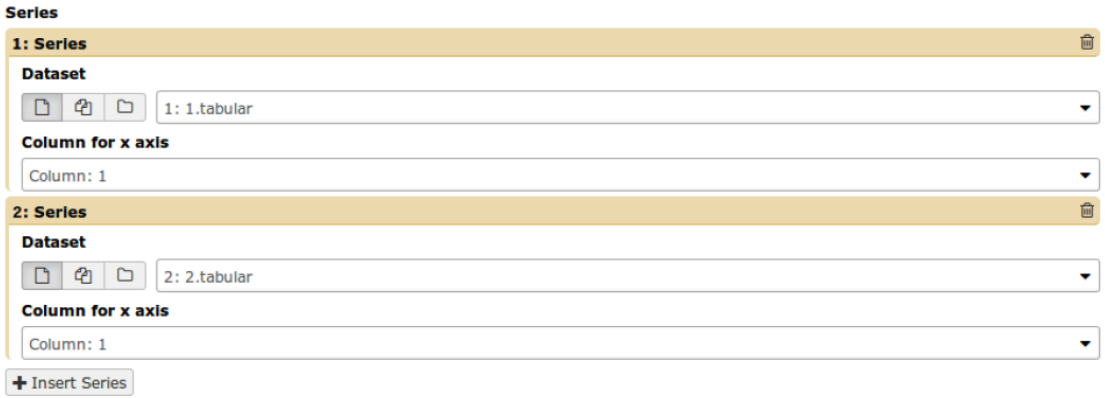
```
<param name="..." type="select" multiple="true" label="..." help="...">
    <option value="ld" selected="true">Length distribution</option>
    <option value="gc" selected="true">GC content distribution</option>
</param>
```

inputs > params > conditional

```
<command><![CDATA[
#if $fastq_input.selector == 'paired':
    '$fastq_input.input1' '$fastq_input.input2'
#else:
    '$fastq_input.input'
#end if
]]></command>
<inputs>
    <conditional name="fastq_input">
        <param name="selector" type="select" label="Single or paired-end reads?">
            <option value="paired">Paired-end</option>
            <option value="single">Single-end</option>
        </param>
        <when value="paired">
            <param name="input1" type="data" format="fastq" label="Forward reads"
            <param name="input2" type="data" format="fastq" label="Reverse reads"
        </when>
        <when value="single">
            <param name="input" type="data" format="fastq" label="Single reads" />
        </when>
    </conditional>
</inputs>
```



inputs > param > repeat



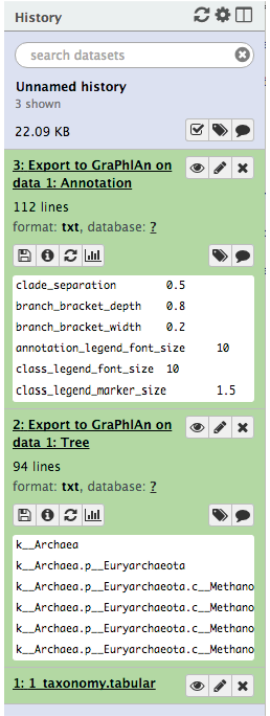

```
<command><![CDATA[
#for $i, $s in enumerate( $series )
    rank_of_series=$i
    input_path=${s.input}
    x_column=${s.xcol}
#end for
]]></command>

<inputs>
    <repeat name="series" title="Series">
        <param name="input" type="data" format="tabular" label="Dataset"/>
        <param name="xcol" type="data_column" data_ref="input" label="Column for x
    </repeat>
</inputs>
```

outputs

Which files the tool will produce as output?

```
<outputs>
    <data name="tree" format="txt" label="${tool.name} on ${on_string}: Tree" />
    <data name="annotation" format="txt"
        label="${tool.name} on ${on_string}: Annotation" />
</outputs>
```



outputs > filter

Conditional

```
<inputs>
    <param type="select" name="format" label="Output format">
        <option value="png">PNG</option>
        <option value="pdf">PDF</option>
    </param>
</inputs>
<outputs>
    <data name="png_output" format="png" label="${tool.name} on ${on_string}: PNG"
        <filter>format == "png"</filter>
    </data>
    <data name="pdf_output" format="pdf" label="${tool.name} on ${on_string}: PDF"
        <filter>format == "pdf"</filter>
    </data>
</outputs>
```

Dependencies (<requirements>)

```
<tool id="hello" name="hello" version="0.01">
    <description>World</description>
    <command><![CDATA[
        /bin/echo 'Hello World! $mystring' > '$output1'
```

Definition of requirements in `.xml` file

```
<requirements>
    <requirement type="package" version="1.66">biopython</requirement>
    <requirement type="package" version="1.0.0">graphlan</requirement>
</requirements>
```

Local installation using Conda packages

```
    <help><![CDATA[
        **What it does**
        Says hello
    ]]></help>
</tool>
```

<help>

GraPhlAn to produce graphical output of an input tree (Galaxy Version 0.9.7)

Options

Input tree

107: Generation, personalization and annotation of tree on data 106 and data 105: Tree in PhyloXML

The tree must be in PhlyloXML, Newick or text format.

Output format

PNG

(--format)

Dpi of the output image (Optional)

For non vectorial formats (--dpi)

Size of the output image (in inches)

7

(--size)

Distance between the most external graphical element and the border of the image (Optional)

(--pad)

Execute

What it does

GraPhlAn is a software tool for producing high-quality circular representations of taxonomic and phylogenetic trees. GraPhlAn focuses on concise, integrative, informative, and publication-ready representations of phylogenetically- and taxonomically-driven investigation.

For more information, check the user manual.

<citations>

Citations Show BibTeX

Kopylova, E. and Noe, L. and Touzet, H. (2012). SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. In *Bioinformatics, 28 (24), pp. 3211-3217.* [doi:10.1093/bioinformatics/bts611][Link]

Quast, C. and Pruesse, E. and Yilmaz, P. and Gerken, J. and Schweer, T. and Yarza, P. and Peplies, J. and Glockner, F. O. (2012). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. In *Nucleic Acids Research, 41 (D1), pp. D590-D596.* [doi:10.1093/nar/gks1219][Link]

Burge, S. W. and Daub, J. and Eberhardt, R. and Tate, J. and Barquist, L. and Nawrocki, E. P. and Eddy, S. R. and Gardner, P. P. and Bateman, A. (2012). Rfam 11.0: 10 years of RNA families. In *Nucleic Acids Research, 41 (D1), pp. D226-D232.* [doi:10.1093/nar/gks1005][Link]

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. In *Bioinformatics, 26 (19), pp. 2460-2461.* [doi:10.1093/bioinformatics/btq461][Link]

Loman, Nicholas J and Misra, Raju V and Dallman, Timothy J and Constantinidou, Chrystala and Gharbia, Saheer E and Wain, John and Pallen, Mark J (2012). Performance comparison of benchtop high-throughput sequencing platforms. In *Nature Biotechnology, 30 (5), pp. 434-439.* [doi:10.1038/nbt.2198][Link]


```
<citations>
    <citation type="doi">10.1093/bioinformatics/bts611</citation>
    <citation type="doi">10.1093/nar/gks1219</citation>
    <citation type="doi">10.1093/nar/gks1005</citation>
    <citation type="doi">10.1093/bioinformatics/btq461</citation>
    <citation type="doi">10.1038/nbt.2198</citation>
</citations>
```

If no DOI is available, a BibTeX citation can be specified with `type="bibtex"`

Tips and Best Practices

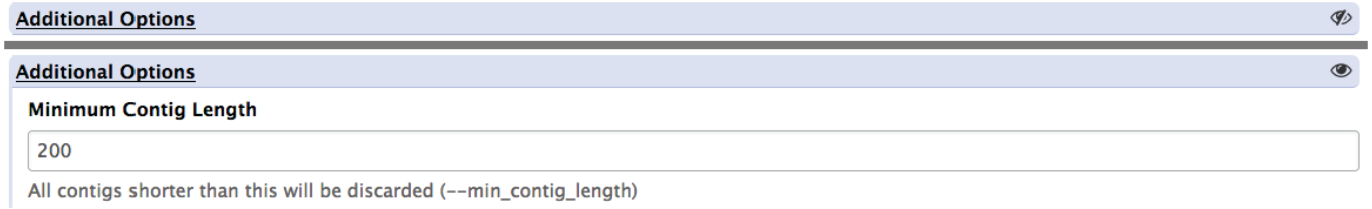
Always quote `text` and `data` parameters and output `data`

```
<command><![CDATA[
graphlan.py
...
'$input_tree'
'$png_output_image'
]]></command>
```

- For security reasons
- Paths may contain spaces

Tips and Best Practices

Use sections to group related parameters



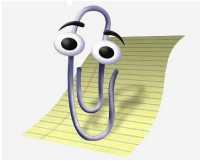

```
<section name="advanced" title="Advanced options" expanded="False">
    <param argument="--size" type="integer" value="7" label="..." help="..."/>
    ...
</section>
```

Find more at:

https://docs.galaxyproject.org/en/latest/dev/schema.html

Life Cycle of a Galaxy Tool

- Dependency management and distribution via `conda`
- Planemo
  - Galaxy tool development helper



- Community

Dependency Resolution

- Dependencies: impede reproducibility, a common source of frustration
  - “Application *X* needs version 1.5 of software *Y* and version 2.3.1b of software *Z* and …”
- Galaxy performs dependency resolution with `conda`, a system- and language-agnostic package manager
  - each tool gets its own `conda` environment
  - Bioconda: special `conda` channel for bioinformatics

Dependency Resolution

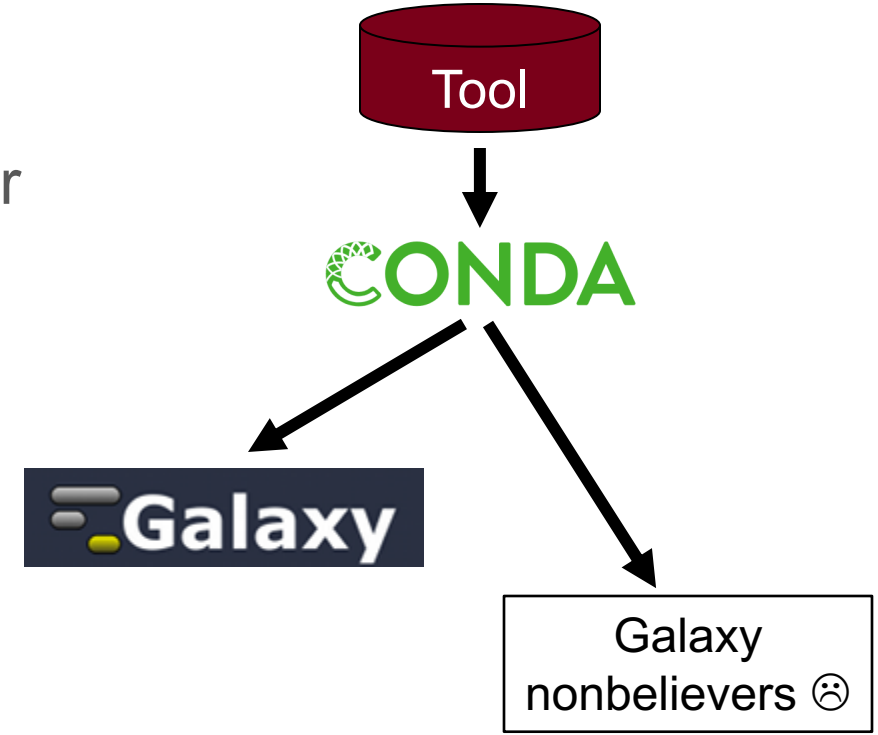
- Galaxy's solution: pull in dependencies in wrapper.

```
<requirements>
    <requirement type="package" version="3.3.1">searchgui</requirement>
</requirements>
```

- Each dependency needs a `conda` "recipe" – a set of instructions for building the application

Dependency Resolution

- Optional, but helpful, to write `conda` recipe for your tool



You're Not Alone, Pt. 1: Planemo

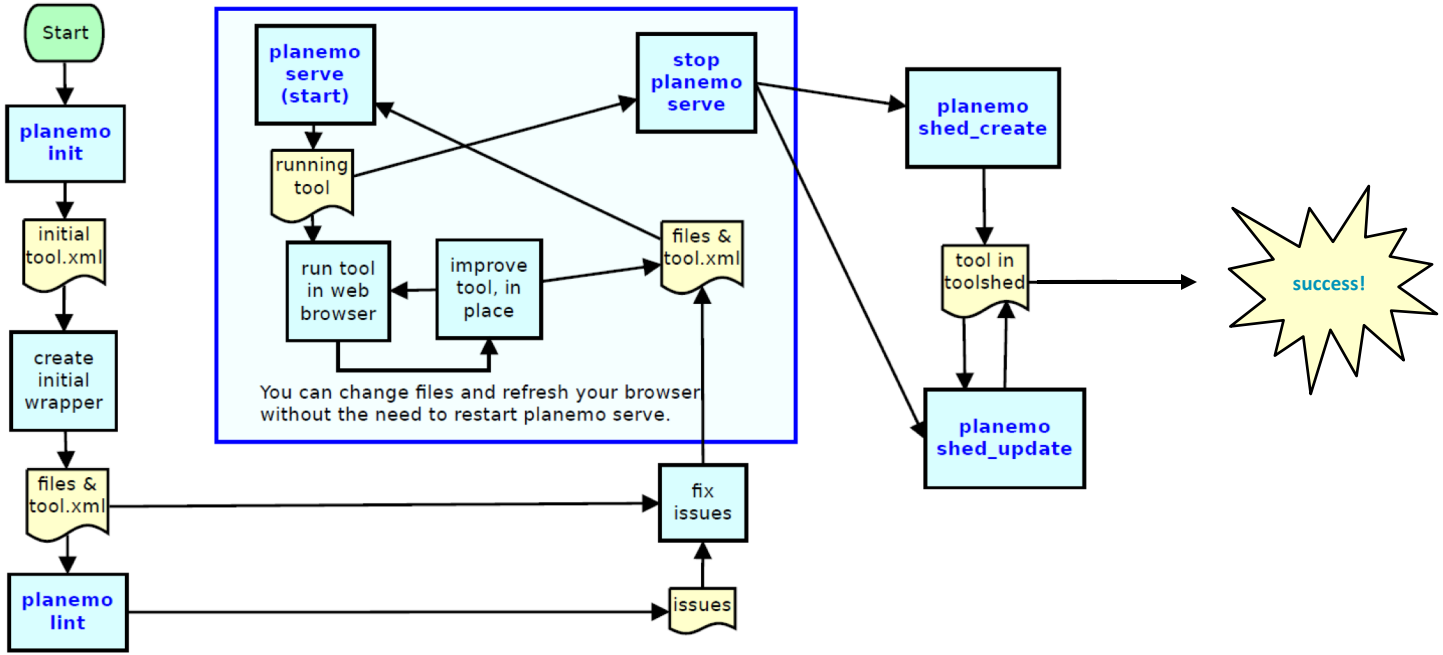
PLANEMO

- The Galaxy team developed Planemo: a set of command-line tools that helps write Galaxy tool wrappers
- Contains many modules, some shown to the right

Module	What it does
`init`	Create an 'outline' for the tool wrapper
`lint`	Check tool XML for mistakes
`test`	Run tool tests
`serve`	Run a clean local Galaxy server equipped with tool
`shed_create`	Create Toolshed repo
`shed_update`	Update tool in Toolshed

You're Not Alone, Pt. 1: Planemo

Example of a tool development workflow:



You're Not Alone, Pt. 2: Community

- Extensive documentation (https://docs.galaxyproject.org/)
  - Including extensive XML
- Github community (https://github.com/galaxyproject)
  - Issues, questions
  - Many example tools
- Us! http://galaxyp.org/

COLLABORATIONS

GalaxyP
galaxyp.org

University of Minnesota
Timothy Griffin
Pratik Jagtap
Praveen Kumar
Candace Guerrero
Subina Mehta
Adrian Hegeman (Co-I)
Art Eschenlauer
Shane Hubler
Ray Sajulga
Caleb Easterly
Andrew Rajczewski

Minnesota Supercomputing Institute
James Johnson
Thomas McGowan
Lee Parsons
Michael Milligan

Harald Barsnes
Marc Vaudel
University of Bergen, Norway

Carolin Kolmeder
University of Helsinki, Finland

Lennart Martens (Co-I)
Bart Mesuere
Robbert G Singh
VIB, UGhent, Belgium

Brook Nunn
U of Washington

Thilo Muth
Bernhard Renard
Robert Koch Institut

Biologists / collaborators
Laurie Parker
Joel Rudney
Maneesh Bhargava
Amy Skubitz
Chris Wendt
Brian Crooker
Steven Friedenberg
Kevin Viken
Kristin Boylan
Marnie Peterson
Somiah Afiuni
Brian Sandri
Alexa Pragman
Wanda Weber
Amy Treeful

Josh Elias
Stanford University

Judson Hervey
Naval Research Institute Washington, D.C.

Lloyd Smith (Co-I)
Michael Shortreed
UW-Madison

Alessandro Tanca
Porto Conte Ricerche, Italy

Anamika Krishanpal
Priyabrata Panigrahi
Persistent Systems Limited

Stephan Kang
Intero Life Sciences

Bjoern Gruening
Bérénice Batut
University of Freiburg, Freiburg, Germany

Thomas Doak
Jeremy Fisher
Indiana University

Matt Chambers
Nashville, TN

Karen Reddy
Mo Heydarian
Johns Hopkins University

Ira Cooke
Melbourne , Australia

INTERESTED IN JOINING THE TEAM?

Workshop Material:
z.umn.edu/asms2018

Website: galaxyp.org
Contact: http://galaxyp.org/contact/
Github: https://github.com/galaxyproteomics
Twitter: twitter.com/usegalaxyp
Funding: NIH NATIONAL CANCER INSTITUTE Informatics Technology for Cancer Research
NSF