

Planemo

A Galaxy Tool SDK

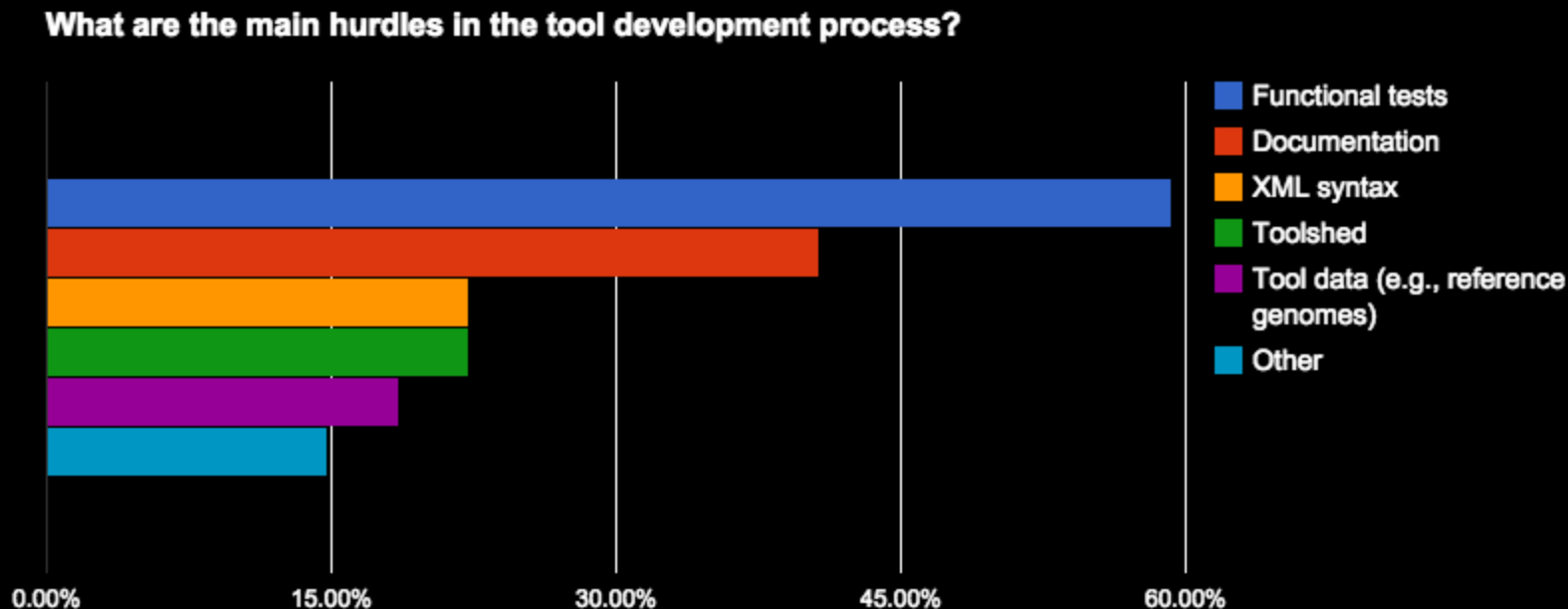
John Chilton, Björn Grüning,
Eric Rasche, Martin Čech, Kyle
Ellrott, and
the Galaxy Team

<http://bit.ly/planemo15>



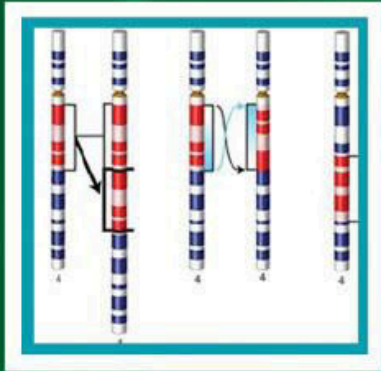
Why? - By Request

Why? - By Request



Why? - Reaching a new audience...

Why? - Reaching a new audience...



ICGC-TCGA DREAM Somatic Mutation Calling Challenge – Tumour Heterogeneity & Evolution



The Cancer Genome Atlas



Understanding genomics to improve cancer care

Galaxy

The DREAM competition will result in a lot of new tool development by developers with no knowledge of Galaxy.

Get Planemo.

With Homebrew/linuxbrew:

```
brew install galaxyproject/tap/planemo
```

With pip

```
pip install planemo
```

Also available as virtual appliances for Docker, Vagrant, Virtualbox, Google Cloud.

But what is it?

```
% planemo --help
```

A simple command-line executable driven by sub-commands (like git) - meant to help with tool development

.

Testing without Planemo

TODO: Spend 5 minutes describing how setup a Galaxy instance for testing tools.

TODO: Spend 5 more minutes summarizing how to use the install and test framework discussed at last year's GCC.

Testing with Planemo

```
% planemo test
```

Finds tools in the current directory and tests them.

Looks for Galaxy, test data, etc... in various reasonable locations relative to the tools and the current working directory.

Testing Better w/Planemo - TDD

- Tests are *much faster*.
- Allows re-running only **failed tests**.

Together these make it much easier to build tools iteratively and quickly via tests (**test-driven development**).

Testing Better w/Planemo - Reports

Tool Test Results (powered by Planemo)

Overview

tophat2 (Test #1)

tophat2 (Test #2)

tophat2 (Test #3)

tophat2 (Test #4)

Overview

All 4 test(s) successfully executed.

Tests

The remainder of this contains a description for each test executed to run these jobs.

tophat2 (Test #1)

status: success

command:

```
tophat2 --num-threads ${GALAXY_SLOTS:-4} /home/john/workspace/galaxy-st-data/bowtie2/tophat_test /tmp/tmp67W4CDfiles/000/dataset_1.dat
```

job standard output:

```
Log: tool progress
```

job standard error:

```
Log: tool progress
```

% planemo share_test



jmchilton commented on Dec 6, 2014

- There was an ambiguous parameter read_mismatches when full parameters were specified twice, so I broke all the test parameters out into conditional blocks to disambiguate.
- Added library test data so the two tests depending on indices work with stand-alone and/or pip testing.
- Last test seemed to cause tophat2 to fail with message "the read mismatches (5) and the read length (2) should be less than or equal to the read edit dist (2)" - so I set the edit distance parameter to 5 to match read mismatches.
- Updated test data to reflect these changes.

Update Tophat2 tests. ...



jmchilton commented on Dec 8, 2014

View test summary [here](#).

Linting with Planemo

```
% planemo lint
```

Finds tools and “lint”s them - checking for

- Problems (bad XML, misspelled fields, etc....)
- Best practices - citations, tests, other IUC best practices.

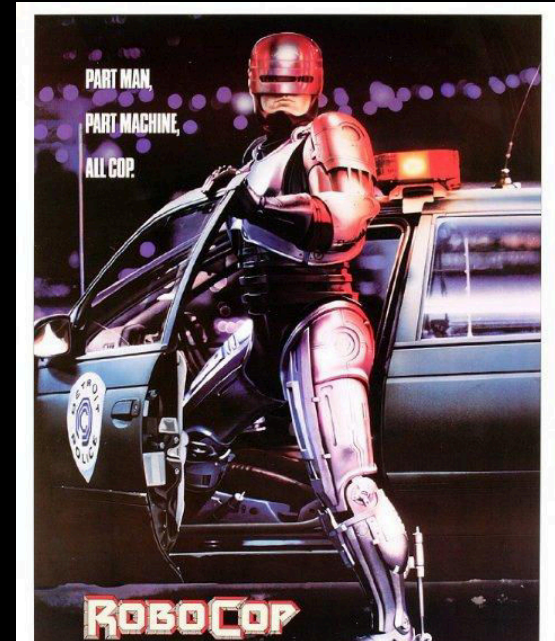
Again speeding tool development by catching problems before even running tests.

IUC Best Practices

Community-maintained set of best practice guidelines for writing consistent and high-quality tools.

<http://bit.ly/iucstandards>

Role models are important.

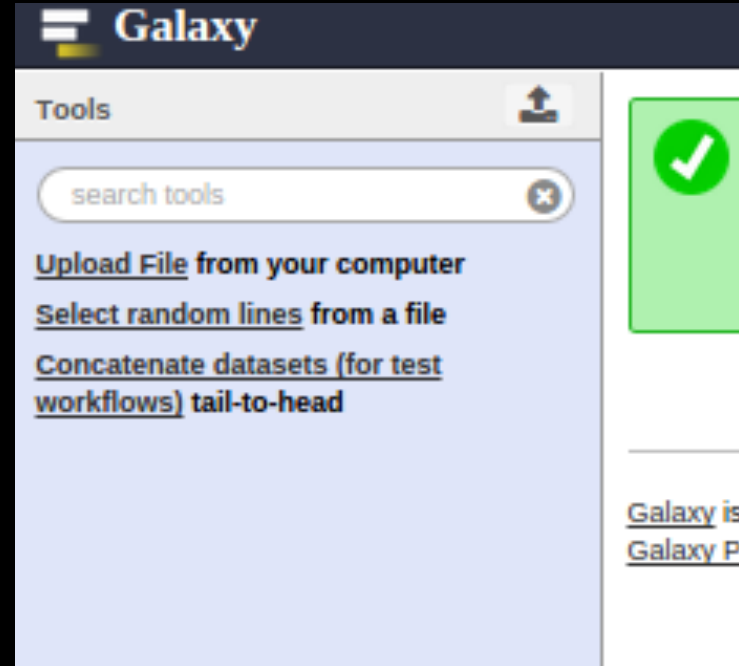


Planemo Serve

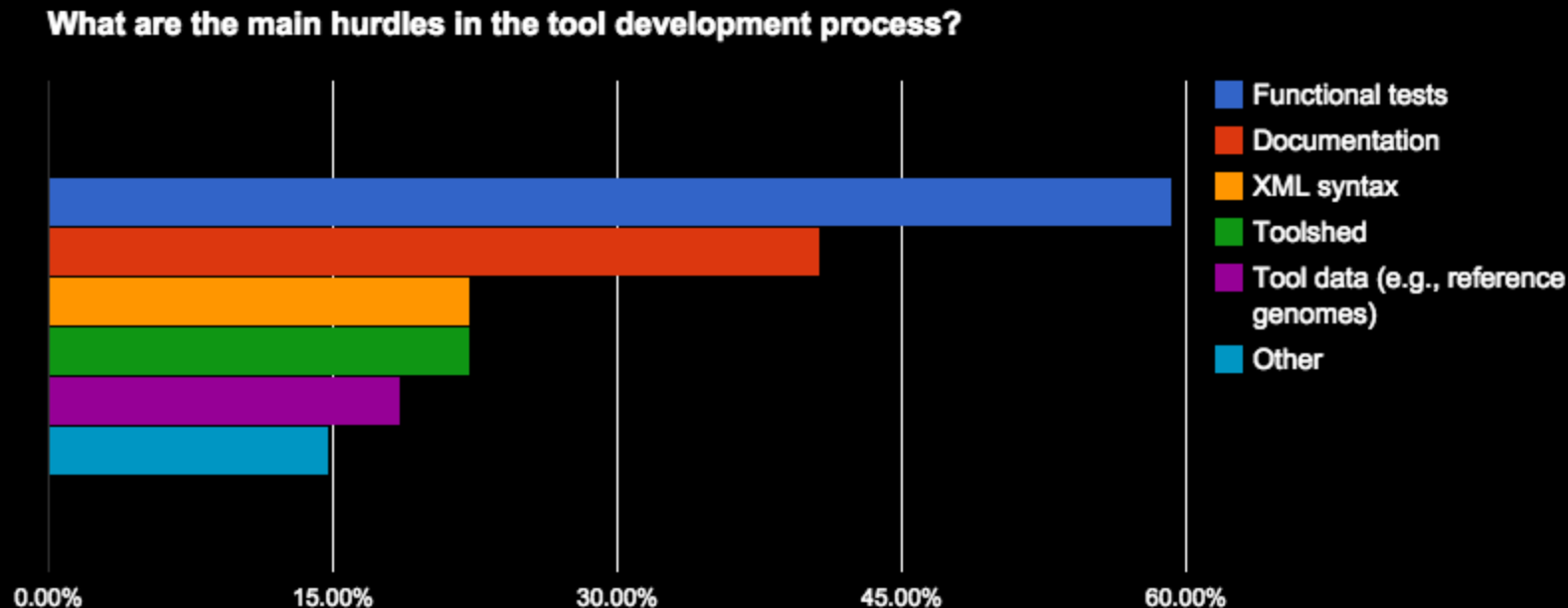
`% planemo serve`

Once done linting and testing - then and hopefully only then should you actually need to see the tool in Galaxy.

Starts a throw away Galaxy server with just your tools.

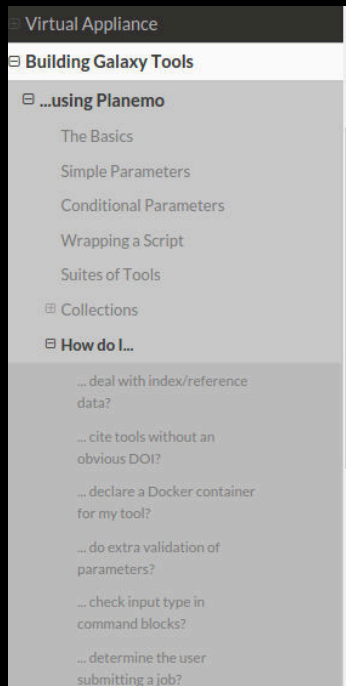


Revisiting Requests



Tool Documentation

<http://planemo.readthedocs.org>



How do I...

This section contains a number of smaller topics with links and examples meant to provide relatively concrete answers for specific tool development scenarios.

... deal with index/reference data?

Galaxy's concept of [data tables](#) are meant to provide tools with access reference datasets or index data not tied to particular histories or users. A common example would be FASTA files for various genomes or mapper-specific indices of those files (e.g. a BWA index for the hg19 genome).

Galaxy [data managers](#) are specialized tools designed to populate tool data tables.

... cite tools without an obvious DOI?

In the absence of an obvious [DOI](#), tools may contain embedded [BibTeX](#) directly.

Further reading:

- [bibtex.xml](#) (test tool with a bunch of random examples)
- [bwa-mem.xml](#) (BWA-MEM tool by Anton Nekrutenko demonstrating citation of an arXiv article)
- [macros.xml](#) (Macros for vcflib tool demonstrating citing a github repository)

declare a Docker container for my tool?

Tool XML Syntax

```
planemo tool_init --id 'seqtk_seq' \  
                  --name 'Convert to FASTA (seqtk)' \  
                  --requirement seqtk@1.0-r68 \  
                  --example_command 'seqtk seq -a 2.fastq > 2.fasta' \  
                  --example_input 2.fastq --example_output 2.fasta \  
                  --test_case --help_from_command 'seqtk seq'
```

tool_init can bootstrap a working tool from an example command - complete with configured test case, test data, and help text.

Tool Shed Publishing Shortcomings

Two prior methods for publishing to the tool shed.

- hg upload
 - Shed wouldn't auto-populate fields
 - No very compatible with storing tools on Github
- Upload form
 - Complicated, difficult to build the correct archive
 - Gave rise to many custom scripts

Both methods still required **lots of GUI interaction** for maintenance (creating repositories, updating metadata).
Very **onerous to manage many repositories** at once.

.shed.yml

A very simple YAML file that describes how a directory (presumably in a Github repository) maps into a ToolShed repository.

Create a .shed.yml file.

```
planemo shed_init --name 'seqtk' --owner 'devteam' \  
    --description 'Tools wrapping seqtk application'\  
    --category 'Sequence Analysis'
```

Very easy to create a .shed.yml file from scratch - but shed_init has "--help" and does some simple validation.

shed_create

```
% planemo shed_create --shed_target testtoolshed
```

Use newly created `.shed.yml` to create a tool shed repository. Here we will publish to the test Tool Shed first.

shed_lint

```
% planemo shed_lint
```

Shed lint checks tool shed related artifacts (`.shed.yml`, `tool_dependencies.xml`, `repository_dependencies.xml`) for common problems.

Run with `--tools` argument to also lint tools.

shed_update

```
% planemo shed_update --shed_target testtoolshed
```

Update metadata and content differences with target tool shed.

Another command `shed_diff` can be used to check these differences.

shed_test

```
% planemo shed_test --shed_target testtoolshed
```

A variant of testing that tests the artifacts published in the Tool Shed (the tools as well as their dependencies specified via `tool_dependencies.xml` files) instead of the local files.

This is a much simpler way to check both tool shed installs and tools together - than the install and test framework discussed last year at the GCC.

Future Plans (Shed, Planemo, and the Web)

Simple **web interface** for managing repositories.

- Maintain links to .shed.yml files in Git(hub) repos.
- Allow **one click testing and publishing**.
- Configurable **auto-synchronization**.
- See Marius van den Beek


Other Tool Development... *Developments*

Citations

Embed DOIs into tools,
Galaxy resolves these
per tools or for a full
analysis histories into
exportable citation list.

SIFTWeb (version 1.0.0)

Input Variants:


Citations 

Kumar, Prateek and Henikoff, Steven and Ng, Pauline C (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. In *Nat Protoc*, 4 (8), pp. 1073–1081. [[doi:10.1038/nprot.2009.86](https://doi.org/10.1038/nprot.2009.86)][[Link](#)]

Ng, P. C. and Henikoff, S. (2001). Predicting Deleterious Amino Acid Substitutions. In *Genome Research*, 11 (5), pp. 863–874. [[doi:10.1101/gr.176601](https://doi.org/10.1101/gr.176601)][[Link](#)]

Ng, P. C. (2002). Accounting for Human Polymorphisms Predicted to Affect Protein Function. In *Genome Research*, 12 (3), pp. 436–446. [[doi:10.1101/gr.212802](https://doi.org/10.1101/gr.212802)][[Link](#)]

Choudhary, S. K. and Noronha, S. B. (2014). GalDrive: Pipeline for comparative identification of driver mutations using the Galaxy framework. [[doi:10.1101/010538](https://doi.org/10.1101/010538)][[Link](#)]

 **bioRxiv**
beta
THE PREPRINT SERVER FOR BIOLOGY

HOME | ABOUT | SEARCH

New Results


GalDrive: Pipeline for comparative identification of driver mutations using the Galaxy framework




Saket Kumar Choudhary, Santosh B Noronha
doi: <http://dx.doi.org/10.1101/010538>


[Abstract](#) [Info/History](#) [Metrics](#) [Preview PDF](#)

[Saket Kumar Choudhary / galaxy_tools](#)

Add tool citations for tools in GalDrive paper and citations for paper itself. #1

 [jinchilton wants to merge 1 commit into saketic:master from jinchilton:citation_support](#)

 Conversation  Commits 1  Files changed 5

 jinchilton commented 2 hours ago

Tool Test Improvements

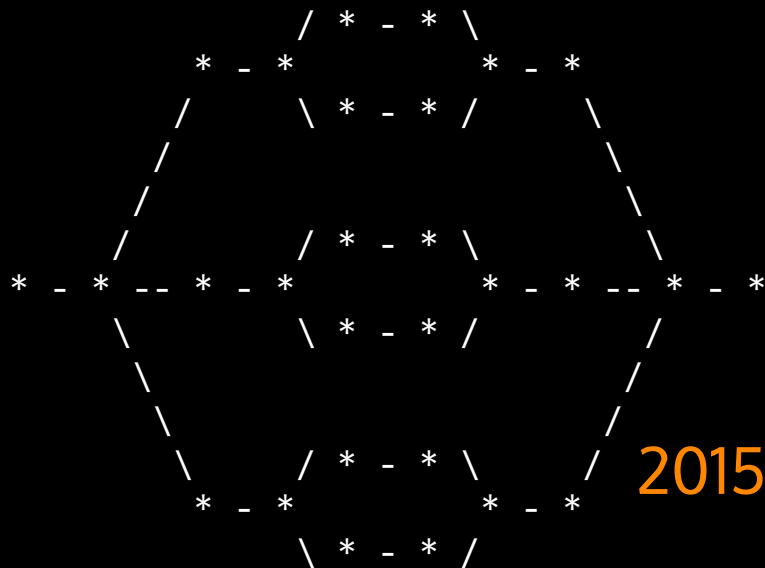
- Allow testing command-line generation directly
- Allow checking binary data via MD5 sums
- Allow testing failure states
- XUnit support
- Allow testing data tables

Data Flow in Workflows

* _ * _ * _ * _ * _ *



2014




2015

<https://bitbucket.org/galaxy/galaxy-central/pull-request/634/>

Convenience Methods

- Easier to force error checking by **exit code**.
- Reference files relative to the tool with **`$__tool_directory__`**.
- Label command-line **arguments** (highlighted in UI to help sophisticated users).
- Color picker widget.

New Tool Form

 **BWA-MEM** - map medium and long reads (> 100 bp) against reference genome (Galaxy Tool Version 0.1)

Options

Load reference genome from

Local cache

Using reference genome

C. elegans (WS220): ce10




Select genome from the list

Single or Paired-end reads

Paired

Select between paired and single end data




Select first set of reads



No fastqsanger dataset available.

Specify dataset with forward reads

Select second set of reads



No fastqsanger dataset available.

Specify dataset with reverse reads

Enter mean, standard deviation, max, and min for insert lengths.

-l; This parameter is only used for paired reads. Only mean is required while sd, max, and min will be inferred. Examples: both "250" and "250.25" will work while "250..10" will not. See below for details.

Thanks!

The **Galaxy Team**

and **planemo** contributors.



Enis
Afghan

Dannon
Baker

Dan
Blankenberg

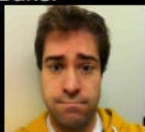
Dave Bouvier

Martin Čech

John Chilton



Dave
Clements



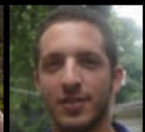
Nate
Coraor



Carl
Eberhard



Jeremy
Goecks



Aysam
Guerler



Jen
Jackson



Ross
Lazarus



Anton
Nekrutenko



Nick Stoler



James Taylor



Nitesh
Turaga



Björn Grüning



Peter Cock



Eric Rasche



Nicola Soranzo



Kyle Ellrott

Future Plans

BEFORE

Create Repository

Name:

Repository type:

Select the repository type based on the following criteria.
Unrestricted - contents can be any set of valid Galaxy utilities or files
Repository suite definition - contents will always be restricted to one file named repository_dependencies.xml
Tool dependency definition - contents will always be restricted to one file named tool_dependencies.xml

Synopsis:

Detailed description:

Categories

☐ Assembly
☐ ChIP-seq
☐ Combinatorial Selections
☐ Computational Chemistry

Multi-select list - hold the appropriate key while clicking to select multiple categories.

Contents:

☐ suite_linkyx_bundle_0.1
☐ repository_dependencies.xml

Click on a file to display its contents below. You may delete files from the repository by clicking the check box next to each file and clicking the Delete selected files button.

Message:

Deleted selected files

This is the commit message for the mercurial change set that will be created if you delete selected files.

```
<?xml version="1.0"?>
<repositories description="Metapackage for the installation of linkyx suite of tools.">
  <repository changeset_revision="09c1e388c20c" name="trinitymaseq" owner="jjohnson" toolshed="https://testtoolshed.g2.bx.psu.edu" />
  <repository changeset_revision="06c42572d7c0" name="fastq_groomer" owner="devteam" toolshed="https://testtoolshed.g2.bx.psu.edu" />
  <repository changeset_revision="3365c42b076d" name="bwa_mappers" owner="devteam" toolshed="https://testtoolshed.g2.bx.psu.edu" />
  <repository changeset_revision="7095d651c95f" name="sam_to_bam" owner="devteam" toolshed="https://testtoolshed.g2.bx.psu.edu" />
  <repository changeset_revision="93ace7e49295" name="picard" owner="devteam" toolshed="https://testtoolshed.g2.bx.psu.edu" />
  <repository changeset_revision="5e12c7427fa3" name="samtools_sort" owner="devteam" toolshed="https://testtoolshed.g2.bx.psu.edu" />
</repositories>
```

AFTER

Galaxy Tool Shed

<https://toolshed.galaxyproject.org/>

Khmer

This Tool Shed Tool is connected to the development repository at <https://github.com/ged-lab/khmer>.

We checked 18 minutes ago and there are updates **AVAILABLE**.

We tested 13 hours ago and the tool **PASSED**.