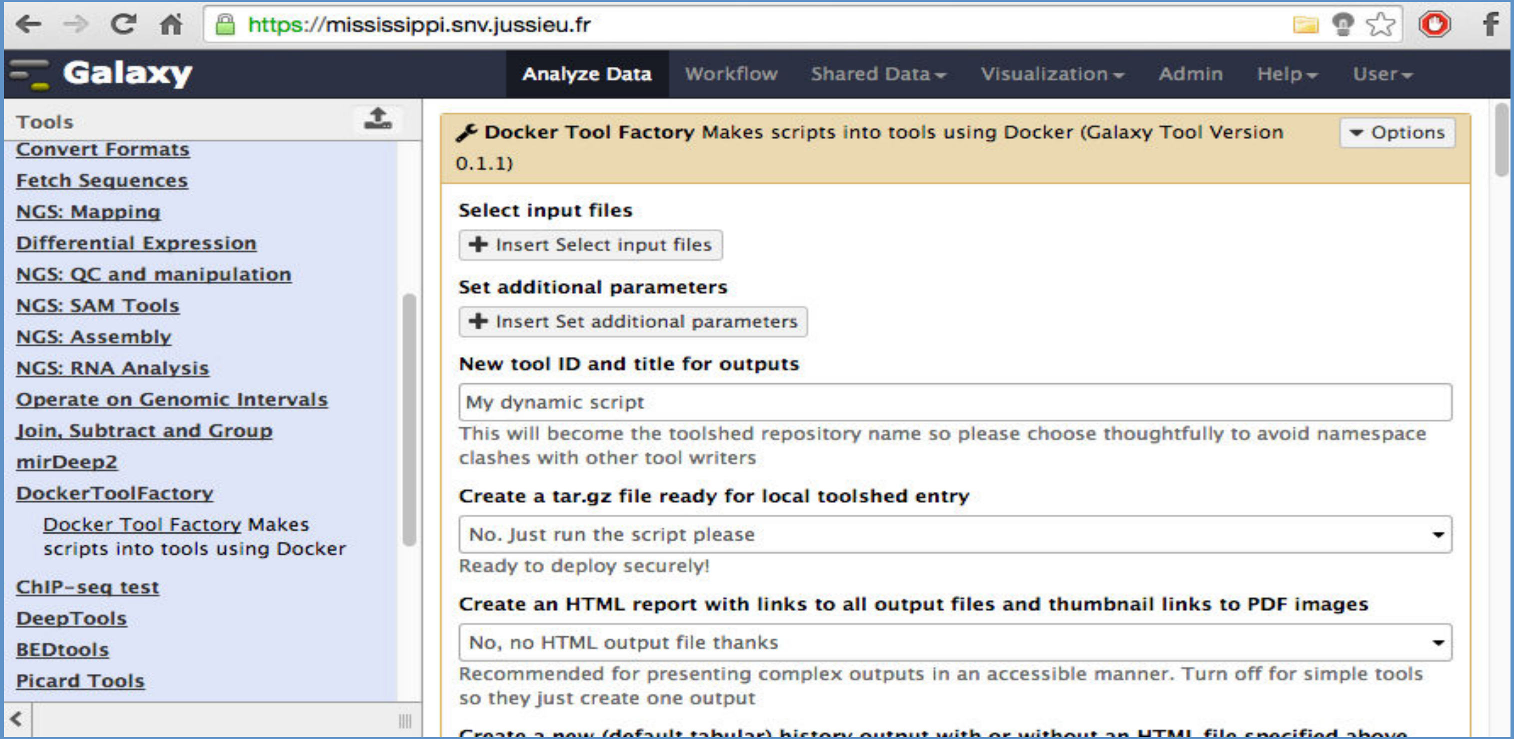
Opening up galaxy for script execution


https://mississippi.snv.jussieu.fr

Docker Tool Factory Makes scripts into tools using Docker (Galaxy Tool Version 0.1.1)

Select input files

+ Insert Select input files

Set additional parameters

+ Insert Set additional parameters

New tool ID and title for outputs

My dynamic script

This will become the toolshed repository name so please choose thoughtfully to avoid namespace clashes with other tool writers

Create a tar.gz file ready for local toolshed entry

No. Just run the script please

Ready to deploy securely!

Create an HTML report with links to all output files and thumbnail links to PDF images

No, no HTML output file thanks

Recommended for presenting complex outputs in an accessible manner. Turn off for simple tools so they just create one output

Create a new (default tabular) history output with or without an HTML file specified above

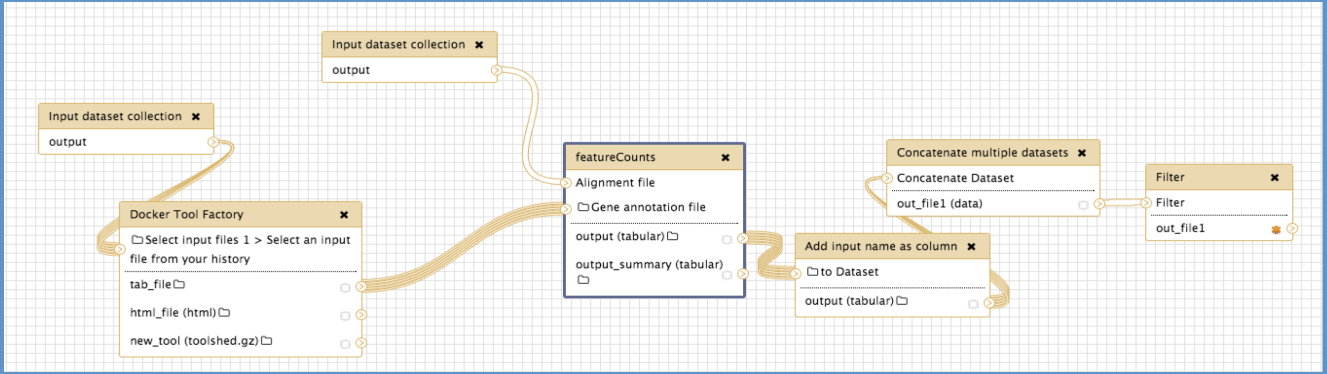
https://bitbucket.org/mvdbeek/dockertoolfactory
https://mississippi.snv.jussieu.fr/

Why do we use galaxy?

Accessibility

Reproducibility

Transparency



Can we benefit from galaxy during data exploration projects?

Re-usable components?

Quick changing of parameters?

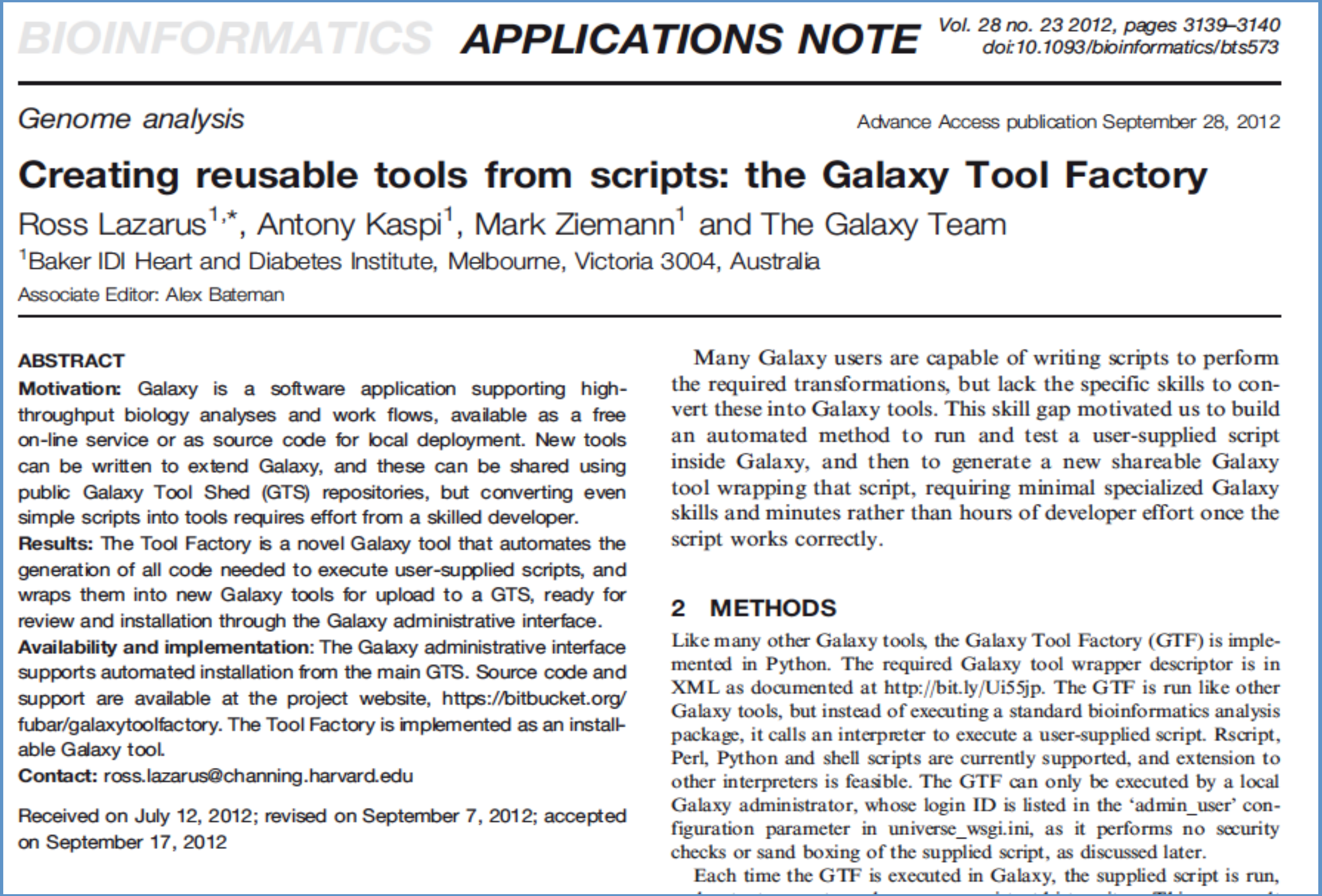
Testing out a new tool/procedure/script?

Steep learning curve – from available tools to new tools

Setting up your own instance

Installing existing tools

Writing and maintaining new tools



BIOINFORMATICS APPLICATIONS NOTE Vol. 28 no. 23 2012, pages 3139–3140
doi:10.1093/bioinformatics/bts573

Genome analysis

Advance Access publication September 28, 2012

Creating reusable tools from scripts: the Galaxy Tool Factory

Ross Lazarus[1,*], Antony Kaspi[1], Mark Ziemann[1] and The Galaxy Team
[1]Baker IDI Heart and Diabetes Institute, Melbourne, Victoria 3004, Australia

Associate Editor: Alex Bateman

ABSTRACT

Motivation: Galaxy is a software application supporting high-throughput biology analyses and work flows, available as a free on-line service or as source code for local deployment. New tools can be written to extend Galaxy, and these can be shared using public Galaxy Tool Shed (GTS) repositories, but converting even simple scripts into tools requires effort from a skilled developer.

Results: The Tool Factory is a novel Galaxy tool that automates the generation of all code needed to execute user-supplied scripts, and wraps them into new Galaxy tools for upload to a GTS, ready for review and installation through the Galaxy administrative interface.

Availability and implementation: The Galaxy administrative interface supports automated installation from the main GTS. Source code and support are available at the project website, https://bitbucket.org/fubar/galaxytoolfactory. The Tool Factory is implemented as an installable Galaxy tool.

Contact: ross.lazarus@channing.harvard.edu

Received on July 12, 2012; revised on September 7, 2012; accepted on September 17, 2012

Many Galaxy users are capable of writing scripts to perform the required transformations, but lack the specific skills to convert these into Galaxy tools. This skill gap motivated us to build an automated method to run and test a user-supplied script inside Galaxy, and then to generate a new shareable Galaxy tool wrapping that script, requiring minimal specialized Galaxy skills and minutes rather than hours of developer effort once the script works correctly.

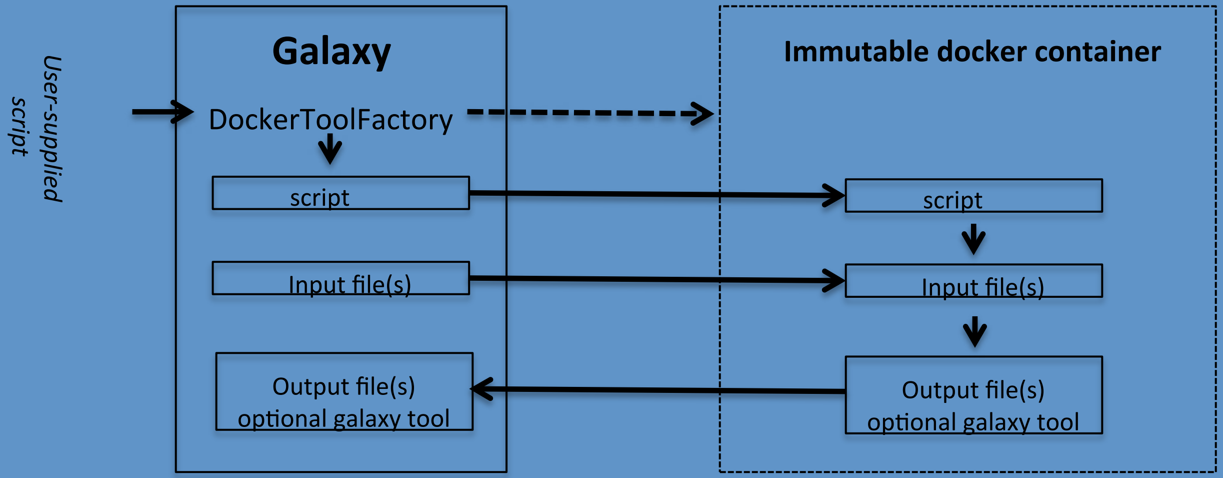
2 METHODS

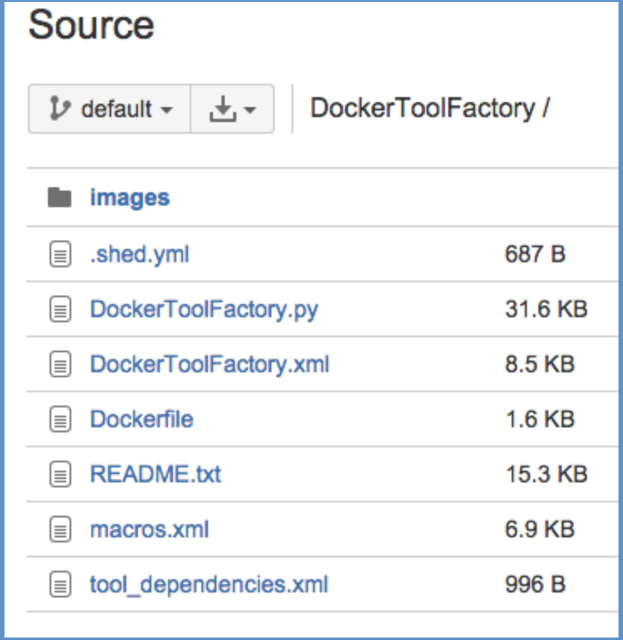
Like many other Galaxy tools, the Galaxy Tool Factory (GTF) is implemented in Python. The required Galaxy tool wrapper descriptor is in XML as documented at http://bit.ly/Ui55jp. The GTF is run like other Galaxy tools, but instead of executing a standard bioinformatics analysis package, it calls an interpreter to execute a user-supplied script. Rscript, Perl, Python and shell scripts are currently supported, and extension to other interpreters is feasible. The GTF can only be executed by a local Galaxy administrator, whose login ID is listed in the 'admin_user' configuration parameter in universe_wsgi.ini, as it performs no security checks or sand boxing of the supplied script, as discussed later.

Each time the GTF is executed in Galaxy, the supplied script is run,

https://github.com/galaxyproject/tools-iuc/tree/master/tools/tool_factory_2

Isolating script execution in a docker layer


User-supplied script
Galaxy
DockerToolFactory
script
Input file(s)
Output file(s) optional galaxy tool
Immutable docker container
script
Input file(s)
Output file(s) optional galaxy tool



images	
.shed.yml	687 B
DockerToolFactory.py	31.6 KB
DockerToolFactory.xml	8.5 KB
Dockerfile	1.6 KB
README.txt	15.3 KB
macros.xml	6.9 KB
tool_dependencies.xml	996 B

https://bitbucket.org/mvdbeek/dockertoolfactory/

What can we do with the Docker toolfactory?

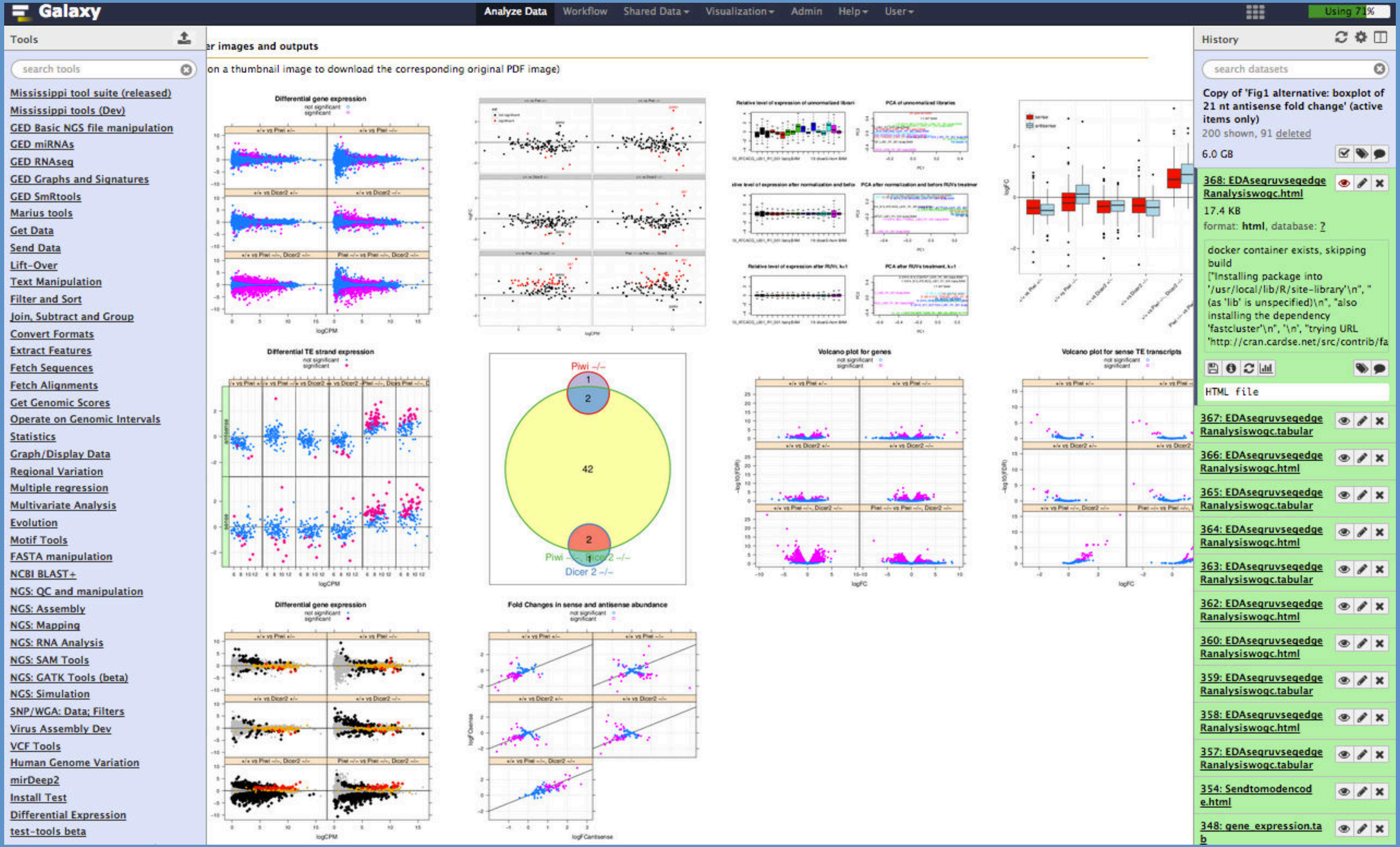
- Stay in galaxy while using scripts

  → data and scripts side-by-side
  (less “dark script matter”, R. Lazarus)
  → from input to figure

- Learn scripting

- run API scripts directly from within galaxy

What can we do with the Docker toolfactory?



r images and outputs

on a thumbnail image to download the corresponding original PDF image)

Mississippi tool suite (released)
Mississippi tools (Dev)
GED Basic NGS file manipulation
GED miRNAs
GED RNAseq
GED Graphs and Signatures
GED SmRtools
Marius tools
Get Data
Send Data
Lift-Over
Text Manipulation
Filter and Sort
Join, Subtract and Group
Convert Formats
Extract Features
Fetch Sequences
Fetch Alignments
Get Genomic Scores
Operate on Genomic Intervals
Statistics
Graph/Display Data
Regional Variation
Multiple regression
Multivariate Analysis
Evolution
Motif Tools
FASTA manipulation
NCBI BLAST+
NGS: QC and manipulation
NGS: Assembly
NGS: Mapping
NGS: RNA Analysis
NGS: SAM Tools
NGS: GATK Tools (beta)
NGS: Simulation
SNP/WGA: Data; Filters
Virus Assembly Dev
VCF Tools
Human Genome Variation
mirDeep2
Install Test
Differential Expression
test-tools beta

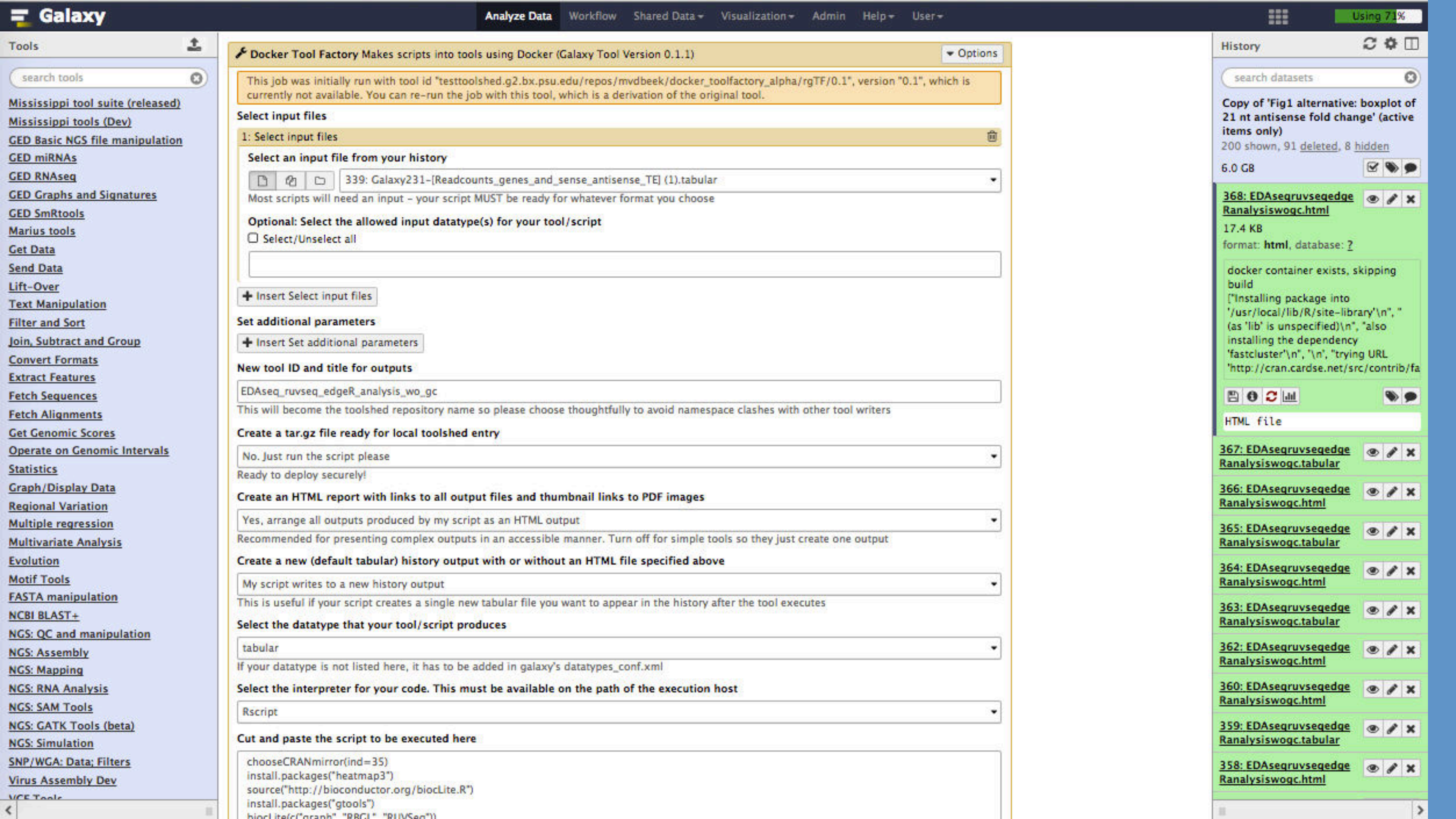
Copy of 'Fig1 alternative: boxplot of 21 nt antisense fold change' (active items only)
200 shown, 91 deleted
6.0 GB

368: EDAseqruvseqedgeRanalysiswogc.html
17.4 KB
format: html, database: ?

```
docker container exists, skipping
build
["Installing package into
'/usr/local/lib/R/site-library'\n", "
(as 'lib' is unspecified)\n", "also
installing the dependency
'fastcluster'\n", '\n', "trying URL
'http://cran.cardse.net/src/contrib/fa
```

HTML file

367: EDAseqruvseqedgeRanalysiswogc.tabular
366: EDAseqruvseqedgeRanalysiswogc.html
365: EDAseqruvseqedgeRanalysiswogc.tabular
364: EDAseqruvseqedgeRanalysiswogc.html
363: EDAseqruvseqedgeRanalysiswogc.tabular
362: EDAseqruvseqedgeRanalysiswogc.html
360: EDAseqruvseqedgeRanalysiswogc.html
359: EDAseqruvseqedgeRanalysiswogc.tabular
358: EDAseqruvseqedgeRanalysiswogc.html
357: EDAseqruvseqedgeRanalysiswogc.tabular
354: Sendtomodencode.html
348: gene_expression.tab



Docker Tool Factory Makes scripts into tools using Docker (Galaxy Tool Version 0.1.1)

This job was initially run with tool id "testtoolshed.g2.bx.psu.edu/repos/mvdbeek/docker_toolfactory_alpha/rgTF/0.1", version "0.1", which is currently not available. You can re-run the job with this tool, which is a derivation of the original tool.

Select input files

1: Select input files

Select an input file from your history

339: Galaxy231-[Readcounts_genes_and_sense_antisense_TE] (1).tabular

Most scripts will need an input - your script MUST be ready for whatever format you choose

Optional: Select the allowed input datatype(s) for your tool/script

☐ Select/Unselect all

Set additional parameters

New tool ID and title for outputs

EDAseq_ruvseq_edgeR_analysis_wo_gc

This will become the toolshed repository name so please choose thoughtfully to avoid namespace clashes with other tool writers

Create a tar.gz file ready for local toolshed entry

No. Just run the script please

Ready to deploy securely!

Create an HTML report with links to all output files and thumbnail links to PDF images

Yes, arrange all outputs produced by my script as an HTML output

Recommended for presenting complex outputs in an accessible manner. Turn off for simple tools so they just create one output

Create a new (default tabular) history output with or without an HTML file specified above

My script writes to a new history output

This is useful if your script creates a single new tabular file you want to appear in the history after the tool executes

Select the datatype that your tool/script produces

tabular

If your datatype is not listed here, it has to be added in galaxy's datatypes_conf.xml

Select the interpreter for your code. This must be available on the path of the execution host

Rscript

Cut and paste the script to be executed here

```
chooseCRANmirror(ind=35)
install.packages("heatmap3")
source("http://bioconductor.org/biocLite.R")
install.packages("gtools")
biocLite(c("graph", "RBGL", "RUVSeq"))
```

Copy of 'Fig1 alternative: boxplot of 21 nt antisense fold change' (active items only)

200 shown, 91 deleted, 8 hidden

6.0 GB

368: EDAseqruvseqedge Ranalysiswogc.html

17.4 KB

format: **html**, database: ?

```
docker container exists, skipping
build
["Installing package into
'/usr/local/lib/R/site-library'\n", "
(as 'lib' is unspecified)\n", "also
installing the dependency
'fastcluster'\n", '\n', "trying URL
'http://cran.cardse.net/src/contrib/fa
```

HTML file

367: EDAseqruvseqedge Ranalysiswogc.tabular

366: EDAseqruvseqedge Ranalysiswogc.html

365: EDAseqruvseqedge Ranalysiswogc.tabular

364: EDAseqruvseqedge Ranalysiswogc.html

363: EDAseqruvseqedge Ranalysiswogc.tabular

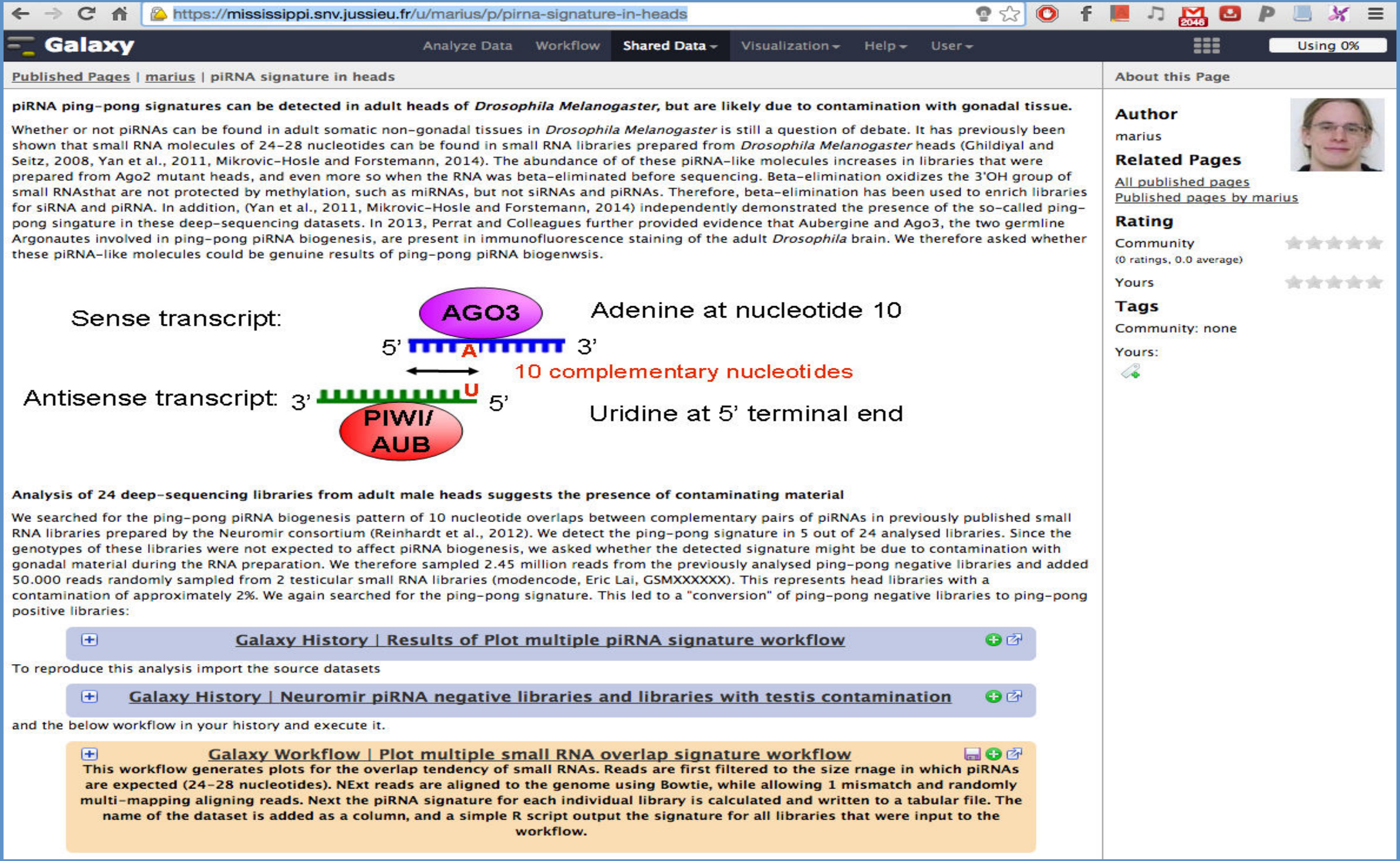
362: EDAseqruvseqedge Ranalysiswogc.html

360: EDAseqruvseqedge Ranalysiswogc.html

359: EDAseqruvseqedge Ranalysiswogc.tabular

358: EDAseqruvseqedge Ranalysiswogc.html

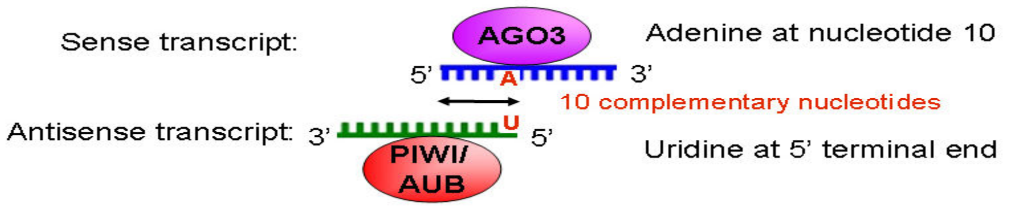
Example 1: Plotting


https://mississippi.snv.jussieu.fr/u/marius/p/pirna-signature-in-heads
Galaxy
Analyze Data
Workflow
Shared Data
Visualization
Help
User
Using 0%

Published Pages | marius | piRNA signature in heads

piRNA ping-pong signatures can be detected in adult heads of *Drosophila Melanogaster*, but are likely due to contamination with gonadal tissue.

Whether or not piRNAs can be found in adult somatic non-gonadal tissues in *Drosophila Melanogaster* is still a question of debate. It has previously been shown that small RNA molecules of 24-28 nucleotides can be found in small RNA libraries prepared from *Drosophila Melanogaster* heads (Ghildiyal and Seitz, 2008, Yan et al., 2011, Mikrovic-Hosle and Forstemann, 2014). The abundance of of these piRNA-like molecules increases in libraries that were prepared from Ago2 mutant heads, and even more so when the RNA was beta-eliminated before sequencing. Beta-elimination oxidizes the 3'OH group of small RNAsthat are not protected by methylation, such as miRNAs, but not siRNAs and piRNAs. Therefore, beta-elimination has been used to enrich libraries for siRNA and piRNA. In addition, (Yan et al., 2011, Mikrovic-Hosle and Forstemann, 2014) independently demonstrated the presence of the so-called ping-pong singature in these deep-sequencing datasets. In 2013, Perrat and Colleagues further provided evidence that Aubergine and Ago3, the two germline Argonautes involved in ping-pong piRNA biogenesis, are present in immunofluorescence staining of the adult *Drosophila* brain. We therefore asked whether these piRNA-like molecules could be genuine results of ping-pong piRNA biogenwsis.



Analysis of 24 deep-sequencing libraries from adult male heads suggests the presence of contaminating material

We searched for the ping-pong piRNA biogenesis pattern of 10 nucleotide overlaps between complementary pairs of piRNAs in previously published small RNA libraries prepared by the Neuromir consortium (Reinhardt et al., 2012). We detect the ping-pong signature in 5 out of 24 analysed libraries. Since the genotypes of these libraries were not expected to affect piRNA biogenesis, we asked whether the detected signature might be due to contamination with gonadal material during the RNA preparation. We therefore sampled 2.45 million reads from the previously analysed ping-pong negative libraries and added 50.000 reads randomly sampled from 2 testicular small RNA libraries (modencode, Eric Lai, GSMXXXXXX). This represents head libraries with a contamination of approximately 2%. We again searched for the ping-pong signature. This led to a "conversion" of ping-pong negative libraries to ping-pong positive libraries:

Galaxy History | Results of Plot multiple piRNA signature workflow

To reproduce this analysis import the source datasets

Galaxy History | Neuromir piRNA negative libraries and libraries with testis contamination

and the below workflow in your history and execute it.

Galaxy Workflow | Plot multiple small RNA overlap signature workflow

This workflow generates plots for the overlap tendency of small RNAs. Reads are first filtered to the size rnage in which piRNAs are expected (24-28 nucleotides). NExt reads are aligned to the genome using Bowtie, while allowing 1 mismatch and randomly multi-mapping aligning reads. Next the piRNA signature for each individual library is calculated and written to a tabular file. The name of the dataset is added as a column, and a simple R script output the signature for all libraries that were input to the workflow.

About this Page

Author

marius

Related Pages

All published pages
Published pages by marius

Rating

Community
(0 ratings, 0.0 average)

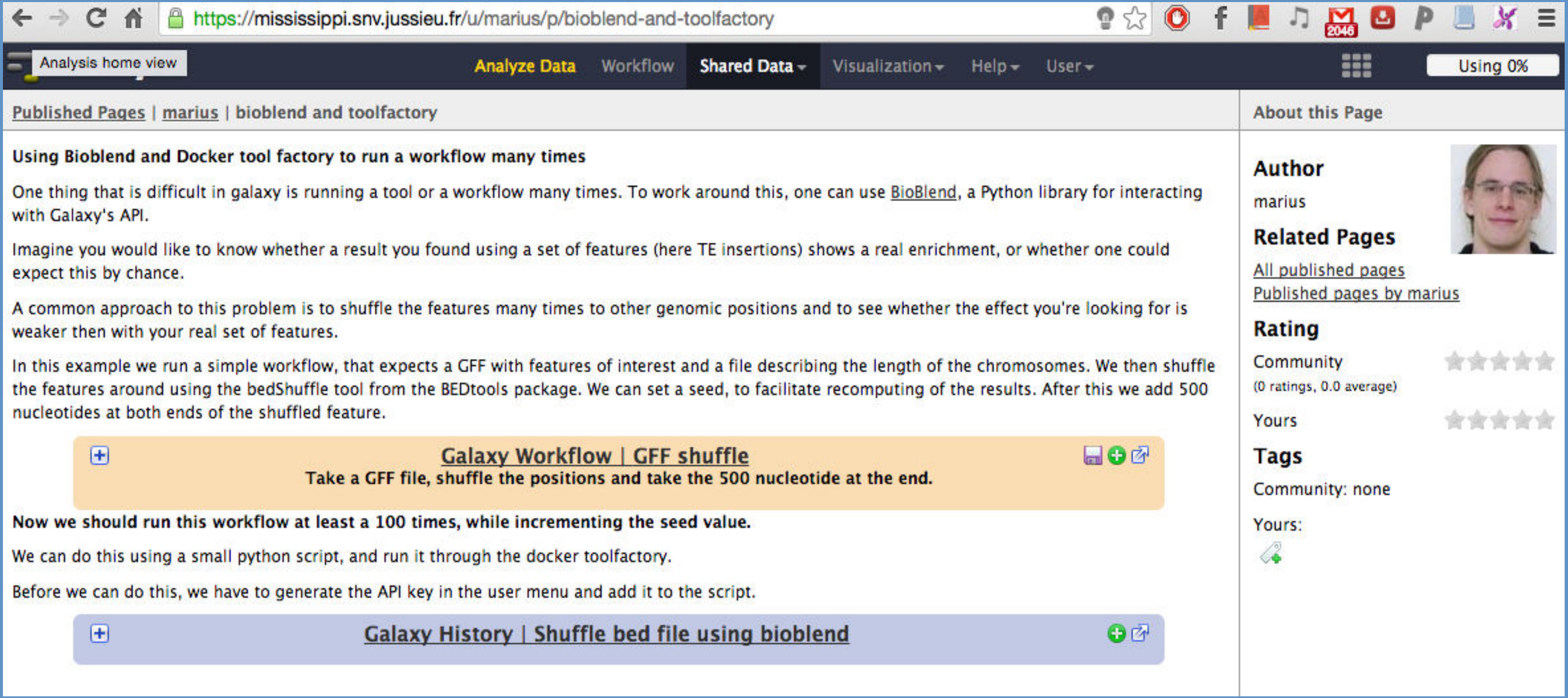
Yours

Tags

Community: none

Yours:

Example 2: Complex workflows



Published Pages | marius | bioblend and toolfactory

Using Bioblend and Docker tool factory to run a workflow many times

One thing that is difficult in galaxy is running a tool or a workflow many times. To work around this, one can use BioBlend, a Python library for interacting with Galaxy's API.

Imagine you would like to know whether a result you found using a set of features (here TE insertions) shows a real enrichment, or whether one could expect this by chance.

A common approach to this problem is to shuffle the features many times to other genomic positions and to see whether the effect you're looking for is weaker then with your real set of features.

In this example we run a simple workflow, that expects a GFF with features of interest and a file describing the length of the chromosomes. We then shuffle the features around using the bedShuffle tool from the BEDtools package. We can set a seed, to facilitate recomputing of the results. After this we add 500 nucleotides at both ends of the shuffled feature.

Galaxy Workflow | GFF shuffle

Take a GFF file, shuffle the positions and take the 500 nucleotide at the end.

Now we should run this workflow at least a 100 times, while incrementing the seed value.

We can do this using a small python script, and run it through the docker toolfactory.

Before we can do this, we have to generate the API key in the user menu and add it to the script.

Galaxy History | Shuffle bed file using bioblend

About this Page

Author

marius

Related Pages

All published pages
Published pages by marius

Rating

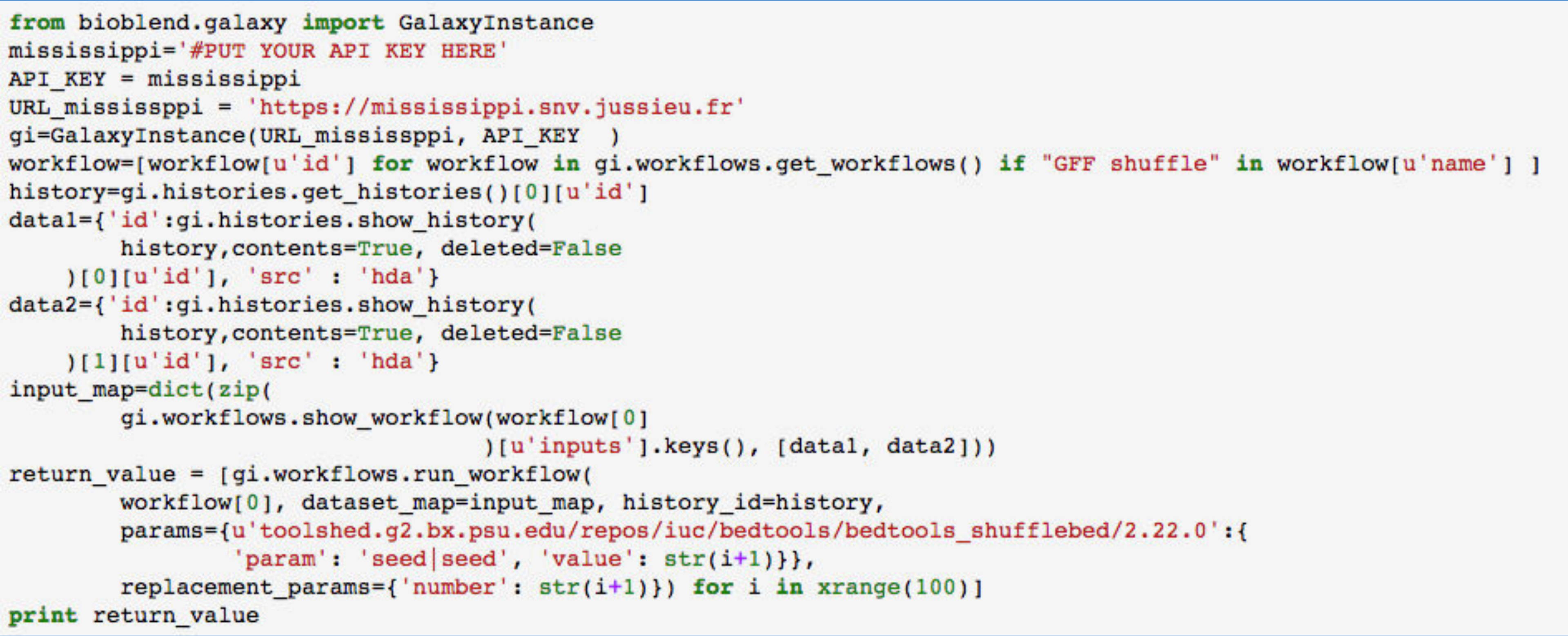
Community
(0 ratings, 0.0 average)

Yours

Tags

Community: none

Yours:



```python
from bioblend.galaxy import GalaxyInstance
mississippi='#PUT YOUR API KEY HERE'
API_KEY = mississippi
URL_mississppi = 'https://mississippi.snv.jussieu.fr'
gi=GalaxyInstance(URL_mississppi, API_KEY  )
workflow=[workflow[u'id'] for workflow in gi.workflows.get_workflows() if "GFF shuffle" in workflow[u'name'] ]
history=gi.histories.get_histories()[0][u'id']
data1={'id':gi.histories.show_history(
        history,contents=True, deleted=False
    )[0][u'id'], 'src' : 'hda'}
data2={'id':gi.histories.show_history(
        history,contents=True, deleted=False
    )[1][u'id'], 'src' : 'hda'}
input_map=dict(zip(
        gi.workflows.show_workflow(workflow[0]
                                   )[u'inputs'].keys(), [data1, data2]))
return_value = [gi.workflows.run_workflow(
        workflow[0], dataset_map=input_map, history_id=history,
        params={u'toolshed.g2.bx.psu.edu/repos/iuc/bedtools/bedtools_shufflebed/2.22.0':{
                'param': 'seed|seed', 'value': str(i+1)}},
        replacement_params={'number': str(i+1)}) for i in xrange(100)]
print return_value
```

Roadmap/plans

- Investigate if users could provide their own images/ or commit current images for custom dependencies
- Use javascript to aid in parameter selection
- Better multi-output support
- Simple API binding … `galaxy_push myscript.sh`

Acknowledgements

- Christophe Antoniewski
- Galaxy community