

Introduction to using and developing IEs

Eric Rasche, Björn Grüning, John Chilton, Cameron Smith

What are IEs

- IE - interactive Environments
- Galaxy view into a Docker container
- Wrap your web services and embed it into Galaxy

In case you don't have the docker image pre-installed, please run this at the terminal:

```
$ docker pull bgruening/docker-ipython-notebook
```

What are IEs

Launch IPython
notebook



`ipython.mako`

Configure Docker Container
Run Docker Container

authenticate with Password
load default notebook

Docker Container

IPython + scipy stack installed

`ipython_galaxy_notebook.ipynb`

IPython Webservice



While connection active

Tools

search 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

Multiple Alignments

Metagenomic analyses

FASTA/FASTQ manipulation

NGS: QC and manipulation

NGS: Assembly

NGS: Mapping

NGS: RNA Analysis

NGS: Peak Calling

NGS: Simulation

DNA methylation

SNP/WGA: Data; Filters

SNP/WGA: Statistical Models

Phenotype Association

Annotation

IP[y]: Notebook

ipython_galaxy_notebook (unsaved changes)

File Edit View Insert Cell Kernel Help

        Code Cell Toolbar: None 

Welcome to the interactive Galaxy IPython Notebook.

You can access your data via the dataset number. For example, `handle = get(42)`. To save data, write your data to a file, and then call `put('filename.txt')`. The dataset will then be available in your galaxy history. Notebooks can be saved to Galaxy by clicking the large green button at the top right of the IPython interface.

More help and informations can be found on the project [website](#).

```
In [1]: import numpy
import scipy
import pandas
import matplotlib
import rpy2
import Bio
```

```
In [2]: !pip install pygal --user

Downloading/unpacking pygal
  Downloading pygal-1.7.0.tar.gz (737kB): 737kB downloaded
  Running setup.py (path:/tmp/pip_build_galaxy/pygal/setup.py) egg_info for package pygal

Installing collected packages: pygal
  Running setup.py install for pygal
    changing mode of build/scripts-2.7/pygal_gen.py from 664 to 775

    changing mode of /home/ipython/.local/bin/pygal_gen.py to 775
Successfully installed pygal
Cleaning up...
```

History

search datasets

CTB June 201540 shown, 1 [deleted](#), [hide hidden](#)

29.8 GB

41:
FBgn0039909.pdf

33.7 KB

format: **pdf**, database: ?

uploaded pdf file

Image in pdf format

40:
sashimi_plot.ipynb**39:**
dm3.ensGene.gff3

~350,000 lines

format: **gff3**, database: ?

uploaded gff3 file

   display with IGV [local](#)

1.Seqid	2.Source	3.Type	4.Start
##gff-version 3			
chr3Rhet	ensGene	gene	19352
chr3Rhet	ensGene	mRNA	19352
chr3Rhet	ensGene	five_prime_UTR	19352
chr3Rhet	ensGene	exon	19352
chr3Rhet	ensGene	exon	19367

38: 461178.bam

2.4 MB

format: **bam**, database: ?

Tools

search 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

Multiple Alignments

Metagenomic analyses

FASTA/FASTQ manipulation

NGS: QC and manipulation

NGS: Assembly

NGS: Mapping

NGS: RNA Analysis

NGS: Peak Calling

NGS: Simulation

DNA methylation

SNP/WGA: Data; Filters

SNP/WGA: Statistical Models

Phenotype Association

Annotation

OpenMS

ChemicalToolBox

OBO Ontology manipulation

Test Tools

IP[y]: Notebookipython_galaxy_notebookAutosave Failed

FileEditViewInsertCellKernelHelp

CodeCell Toolbar: None

Install Misopy

In [1]: !pip install --user --quiet misopy

Restart the kernel (Kernel -> Restart), so that IPython finds Misopy

Get Samtools

In [11]: !mkdir output
sash_plot.plot_event("FBgn0039909", "indexed", "sashimi_settings.txt", "output")

Skipping read with CIGAR 35M65N2M
Skipping read with CIGAR 35M65N2M
Skipping read with CIGAR 2M70N35M
Saving plot to: /import/output/FBgn0039909.pdf

FBgn0039909

In [1]: from urllib import urretrieve
urretrieve("http://depot.galaxyproject.org/package/linux/x86_64/samtools/samtools-0.1.19-Linux-x86_64.tgz",
!tar -xf samtools.tgz

The samtools binary should now be in bin/samtools

Generate Indices of the bam Files

Now the bam files from the Galaxy history can be imported with get(history_id)
Then index files of the bam files can be generated via samtools

History

search datasets

CTB June 2015

40 shown, 1 deleted, hide hidden

29.8 GB

41: FBgn0039909.pdf33.7 KBformat: pdf, database: ?uploaded pdf file

40: sashimi_plot.ipynbImage in pdf format

39: dm3.ensGene.gff3~350,000 linesformat: gff3, database: ?uploaded gff3 file

display with IGV

1.Seqid2.Source3.Type4

##gff-version 3

chr3Rhet ensGene gene1

chr3Rhet ensGene mRNA1

chr3Rhet ensGene five_prime_UTR1

chr3Rhet ensGene exon1

chr3Rhet ensGene exon1

38: 461178.bam2.4 MBformat: bam, database: ?uploaded bam file

display with IGV

display in IGB

Install Misopy

```
In [1]: !pip install --user --quiet misopy
```

Restart the kernel (Kernel -> Restart), so that IPython finds Misopy

Get Samtools

```
In [1]: from urllib import urlretrieve
urlretrieve("http://depot.galaxyproject.org/package/linux/x86_64/samtools/samtools-0.1.19-Linux-x86_64.tgz", "samtools.tgz")
!tar -xf samtools.tgz
```

The samtools binary should now be in bin/samtools

Generate Indices of the bam Files

- Now the bam files from the Galaxy history can be imported with `get(history_id)`
- Then index files of the bam files can be generated via samtools

```
In [2]: get(5)
get(6)
!mv 5 461177.bam
!mv 6 461178.bam

!bin/samtools index 461177.bam
!bin/samtools index 461178.bam
```

If everything went well there should now be index files (*.bai) next to the bam files.

History

Unnamed history

6 shown, 6 [deleted](#)

93.6 MB

12:

[ipython_galaxy_notebook.ipynb](#)

11: [FBgn0039909.pdf](#)

36.0 KB

format: pdf, database: 2

uploaded pdf file

Image in pdf format

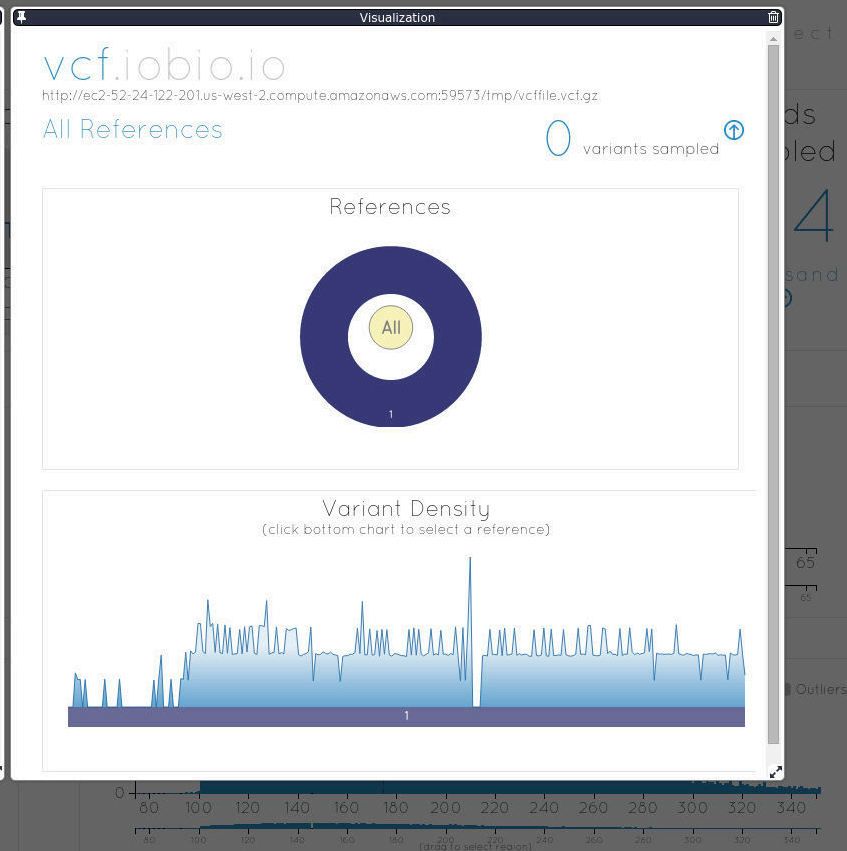
10: [out.gff3](#)

7:

https://raw.githubusercontent.com/bgruening/notebooks/master/Python/sashimi_plot.ipynb

6: [461178.bam](#)

5: [461177.bam](#)



History

search datasets

Interactive Environments

5 shown, 4 deleted

17.8 GB

9: **Select first on data**

2

~220,000 lines

format: **vcf**, database: ?

display with IGV [local](#)

```
1.Chrom 2.Pos 3.ID 4.Ref 5.Alt 6.qual
##fileformat=VCFv4.0
##fileDate=20150416
##source=dbSNP
##dbSNP_BUILD_ID=142
##reference=GRCh38
##phasing=partial
```

8: **Small Test VCF**

7: **vcf/lib-phix.vcf**

6: **HG04141.mapped.I**
LLUMINA.bwa.BEB.lo
w_coverage.20130415.bam

17.1 GB

format: **bam**, database: ?

uploaded bam file

display with IGV [local](#)

display in IGV [View](#)

Binary bam alignments file

2: **ftp://ftp.ncbi.nlm.nih.gov/snp/organism/s/human_9606/VCF/00-All.vcf**

The essentials

- All the magic is in: `./config/plugins/interactive_environments/ipython`
- IE file (`.../config/ipython.xml`)
- IE config file (`.../config/ipython.ini`)
- IE mako file (`.../templates/ipython.mako`)
- Example notebook (`.../templates/notebook.ipynb`)

The essentials :: ipython.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE interactive_environment SYSTEM "../..//interactive_environments.dtd">
<interactive_environment name="IPython">
... <data_sources>
...   <data_source>
...     <model_class>HistoryDatasetAssociation</model_class>
...     <test type="isinstance" test_attr="datatype" result_type="datatype">tabular.Tabular</test>
...     <test type="isinstance" test_attr="datatype" result_type="datatype">data.Text</test>
...     <to_param param_attr="id">dataset_id</to_param>
...   </data_source>
... </data_sources>
... <params>
...   <param type="dataset" var_name_in_template="hda" required="true">dataset_id</param>
... </params>
... <entry_point entry_point_type="mako">ipython.mako</entry_point>
</interactive_environment>
```

The essentials :: ipython.ini

```
[main]
# Following options are ignored if using the Galaxy dynamic proxy but
# are useful if mapping a range of ports for environment consumption.
#apache_urls = False
#password_auth = False
#ssl = False

[docker]
# Command to execute docker. For example `sudo docker` or `docker-lxc`.
command = docker

# The docker image name that should be started.
image = bgruening/docker-ipython-notebook

# Additional arguments that are passed to the `docker run` command.
command_inject = --sig-proxy=true -e DEBUG=false

# URL to access the Galaxy API with from the spawn Docker container, if empty
# this falls back to galaxy.ini's galaxy_infrastructure_url and finally to the
# Docker host of the spawned container if that is also not set.
#galaxy_url =
```

```
## IPython Specific
# Prepare an empty notebook
notebook_id = ie_request.generate_hex(64)
with open(os.path.join(ie_request.attr.our_template_dir,
..... 'notebook.ipynb'), 'r') as nb_handle:
    ... empty_nb = nb_handle.read()
empty_nb = empty_nb % notebook_id
# Copy over default notebook, unless the dataset
# this viz is running on is a notebook
empty_nb_path = os.path.join(temp_dir, 'ipython_galaxy_notebook.ipynb')
if hda.datatype.__class__.__name__ != "Ipynb":
    ... with open(empty_nb_path, 'w+') as handle:
        .... handle.write(empty_nb)
else:
    .... shutil.copy(hda.file_name, empty_nb_path)
```

```
## General IE specific
# Access URLs for the notebook from within galaxy.
notebook_access_url = ie_request.url_template
.... ('${PROXY_URL}/ipython/${PORT}/notebooks/ipython_galaxy_notebook.ipynb')
notebook_login_url = ie_request.url_template
.... ('${PROXY_URL}/ipython/${PORT}/login?next=%2Fipython%2F${PORT}%2Ftree')

# Add all environment variables collected from Galaxy's IE infrastructure
ie_request.launch(env_override={
.... 'notebook_password': PASSWORD,
})
```

```
%>
<html>
<head>
${.ie.load_default_js().}
</head>
<body>

<script.type="text/javascript">
....${.ie.default_javascript_variables().}
....var.notebook_login_url.='${.notebook_login_url.}';
....var.notebook_access_url.='${.notebook_access_url.}';
....${.ie.plugin_require_config().}
|
....//.Load.notebook

....requirejs(['interactive_environments', 'plugin/ipython'], function(){
.....load_notebook(ie_password, notebook_login_url, notebook_access_url);
....});
</script>
<div.id="main".width="100%".height="100%">
</div>
</body>
</html>
```

Running the Docker Image, Manually

```
$ docker --help
```

```
$ docker run -d -p 9000:6789 bgruening/docker-  
ipython-notebook
```

```
$ docker ps
```

Browser (FF, Chrome) on localhost:9000/ipython

```
$ docker stop <funny name>
```

Preparing Galaxy for IEs

In config/galaxy.ini, set the following variables:

```
interactive_environment_plugins_directory =  
config/plugins/interactive_environments
```

```
dynamic_proxy_manage=True
```

Preparing Galaxy for IEs

Change directory into: lib/galaxy/web/proxy/js/

```
$ cd lib/galaxy/web/proxy/js/
```

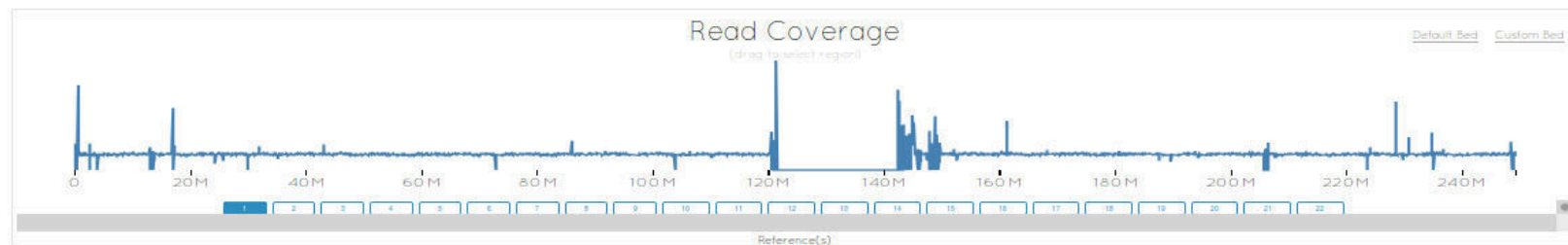
Then install the node proxy with “npm”:

```
$ npm install .
```


Preparing Galaxy for IEs

We need to start the Galaxy proxy

```
> node lib/galaxy/web/proxy/js/lib/main.js  
  --verbose --port 8800 --ip 0.0.0.0  
  --sessions database/session_map.sqlite
```

Reads
Sampled207
thousand

Mapped Reads



Forward Strand



Proper Pairs



Singletons



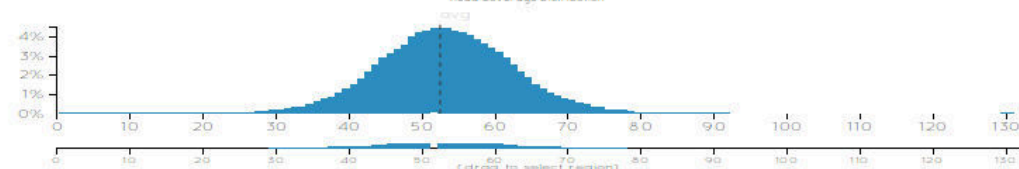
Both Pairs Mapped



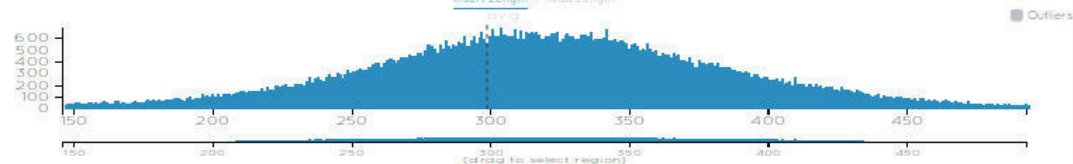
Duplicates



Read Coverage Distribution



Insert Length



Mapping Quality



Getting BAM io.bio running

```
> docker pull qiaoy/iobio-galaxy.bam_iobio_io  
(should be already pulled)
```