Reviewer's report

Title:NCBI BLAST+ integrated into Galaxy

Version:1**Date:**19 January 2015

Reviewer: Stian Soiland-Reyes

Reviewer's report:

This review is also available at
https://gist.github.com/stain/a107379fc1adbb3e392b

The article describes a mechanism to
add the BLAST+ functionality to the Galaxy workflow system.
This is a very useful feature, and so in principle I would
want to see this article published. I do however have some
concerns with the aspects of reproducibility and
documentation, which are detailed below.

Major Compulsory Revisions

I am afraid I will have to ask for major compulsory revisions as I was unable to reproduce any the claims of the paper.

1: Docker image is not BLAST enabled

p5.
> the command docker ... start a BLAST enabled Galaxy instance

I tried the docker image. It starts up fine, and presents a
Galaxy that includes a list of BLAST tools - so the BLAST
tools have been installed.

The docker instance is however *not* BLAST enabled, as the
BLAST tools requires further configuration/download of the
external BLAST reference database to align against. This
procedure is loosely documented at
https://registry.hub.docker.com/u/bgruening/galaxy-blast/ -
but I was unable to follow through with this installation
as it was quite complicated and seems to require
manual downloading and configuration of many GB of

reference data spread over more than 300 files.

I was assuming that a docker image would be
'usable out of the box' - but this is far from the truth in this case.
Accessing "NCBI BLAST+ database info" gives an empty dropdown
list in

The article mentions that the public Galaxy instance
usegalaxy.com does not provide the BLAST tools by
default due to concerns over
computational load - but I am also worried if it could be
because configuring the BLAST+ tools is quite a complicated job.

The article does not mention at all the excessive amount of
system administraton that is required in order to finalize
the BLAST installation, and the docker image does not provide
any helper scripts to assist with this.

In fact, the example database configuration
files uses a totally different path, e.g.
/depot/data2/galaxy/blastdb/nt/nt.chunk - while
the docker image would require these under
/data/nt/nt.chunk.

The article or Docker README does not mention which subset of the
databases would commonly need to be downloaded - or even the fact
that all of the numbered fragments need to be downloaded.

The dataset referenced from the example configuration, e.g.
nt.chunk and wgs.chunk do not exist on
ftp://ftp.ncbi.nlm.nih.gov/blast/db/ - only non-chunk version
exist.

I tried to download a subset of the datasets from
ftp://ftp.ncbi.nlm.nih.gov/blast/db/

stain@biggie-utopic:/galaxy_store/data/blast_databases$ ls
human_genomic.00.nhd nt.00.nhd refseq_genomic.148.nhr refseq_protein.00.pin
refseq_protein.15.pnd wgs.00.nhi
human_genomic.00.nhi nt.00.nhi refseq_genomic.148.nin refseq_protein.00.pnd

refseq_protein.15.pni wgs.00.nhr

human_genomic.00.nhr nt.00.nhr refseq_genomic.148.nnd refseq_protein.00.pni refseq_protein.15.pog wgs.00.nin

human_genomic.00.nin nt.00.nin refseq_genomic.148.nni refseq_protein.00.pog refseq_protein.15.ppd wgs.00.nnd

human_genomic.00.nnd nt.00.nnd refseq_genomic.148.nog refseq_protein.00.ppd refseq_protein.15.ppi wgs.00.nni

human_genomic.00.nni nt.00.nni refseq_genomic.148.nsd refseq_protein.00.ppi refseq_protein.15.psd wgs.00.nog

human_genomic.00.nog nt.00.nog refseq_genomic.148.nsi refseq_protein.00.psd refseq_protein.15.psi wgs.00.nsd

human_genomic.00.nsd nt.00.nsd refseq_genomic.148.nsq refseq_protein.00.psi refseq_protein.15.psq wgs.00.nsi

human_genomic.00.nsi nt.00.nsi refseq_genomic.148.tar.gz refseq_protein.00.psq refseq_protein.15.tar.gz wgs.00.nsq

human_genomic.00.nsq nt.00.nsq refseq_genomic.nal refseq_protein.15.phr refseq_protein.pal wgs.nal

human_genomic.nal nt.nal refseq_protein.00.phr refseq_protein.15.pin wgs.00.nhd

and configured these in blastdb.loc according to the Docker readme. The readme says:

> you need to add the paths to your blast databases and they need to look like /export/swissprot/swissprot

but I have followed the instructions three lines above which mounted the datasets at /data - hence

I used /data/ instead of `/export`. Some consistency would help here.

stain@biggie-utopic:/galaxy_store/data/blast_databases$ grep -v ^# /tmp/galaxy/galaxy-central/tool-data/blastdb*loc

/tmp/galaxy/galaxy-central/tool-data/blastdb.loc:nt_02_Dec_2009 nt 02 Dec 2009 /data/nt

/tmp/galaxy/galaxy-central/tool-data/blastdb.loc:wgs_30_Nov_2009 wgs 30 Nov 2009 /data/wgs/wgs

/tmp/galaxy/galaxy-central/tool-data/blastdb.loc:refseq_genomic_148 refseq 148 /data/refseq_genomic

/tmp/galaxy/galaxy-central/tool-data/blastdb.loc:

/tmp/galaxy/galaxy-central/tool-data/blastdb.loc:

/tmp/galaxy/galaxy-central/tool-data/blastdb_p.loc:nt_02_Dec_2009 nt 02 Dec 2009 /data/nt

/tmp/galaxy/galaxy-central/tool-data/blastdb_p.loc:wgs_30_Nov_2009 wgs 30 Nov 2009 /data/wgs/wgs

/tmp/galaxy/galaxy-central/tool-data/blastdb_p.loc:refseq_protein refseq protein /data/refseq_genomic

/tmp/galaxy/galaxy-central/tool-data/blastdb_p.loc:

/tmp/galaxy/galaxy-central/tool-data/blastdb_p.loc:

A BLAST Data Manager is available at
at https://github.com/peterjc/galaxy_blast/ - in theory
this can download and populate the blastdb data table. This is
mentioned as "Future work" in the article, so presumably it
is not yet production ready.

This data manager does not appear under Data Libraries in the docker
image, and it is not included in the installation at
https://registry.hub.docker.com/u/bgruening/galaxy-blast/dockerfile/

2: Galaxy Tool Shed not working with the docker image

The article says:

> The recently published Galaxy Tool Shed [9] allows anyone hosting a Galaxy
> instance to install tools and defined dependencies with a few clicks right
> from the Galaxy web application itself.

I am unable to verify this claim using the provided Docker image.

I am unable to install any tools from the Galaxy Tool Shed from
the web interface of the Docker image.

I am logged in as
admin@galaxy.org according to the instructions, but if I
go to Admin -> Search and browse tool sheds
http://localhost:8080/admin_toolshed/browse_tool_sheds
and click the dropdown list for "Browse valid sheds", this hangs for a
while before failing with "Can't find the server".

On the console I get many error messages like:

URLError: <urlopen error [Errno -2] Name or service not known>

tool_shed.util.shed_util_common ERROR 2015-01-19 10:22:08,698 Error attempting to get tool shed status for installed repository ncbi_blast_plus: <urlopen error [Errno -2] Name or service not known>

Traceback (most recent call last):

File "lib/tool_shed/util/shed_util_common.py", line 772, in get_tool_shed_status_for_installed_repository

encoded_tool_shed_status_dict = common_util.tool_shed_get(app, tool_shed_url, url)

File "lib/tool_shed/util/common_util.py", line 345, in tool_shed_get

response = urlopener.open(uri)

File "/usr/lib/python2.7/urllib2.py", line 404, in open

response = self._open(req, data)

File "/usr/lib/python2.7/urllib2.py", line 422, in _open

'_open', req)

File "/usr/lib/python2.7/urllib2.py", line 382, in _call_chain

result = func(*args)

File "/usr/lib/python2.7/urllib2.py", line 1214, in http_open

return self.do_open(httplib.HTTPConnection, req)

File "/usr/lib/python2.7/urllib2.py", line 1184, in do_open

raise URLError(err)

Inspecting the internal frame I see the link is

http://toolshed.g2.bx.psu.edu/repository/browse_valid_categories?galaxy_url=http://localhost:80

I somehow feel that toolshed.g2.bx.psu.edu will try to connect to my galaxy instance

at http://localhost:8080 - which is not going to work.

The actual toolshed site is unavailable

http://www.downforeveryoneorjustme.com/toolshed.g2.bx.psu.edu

> It's not just you! http://toolshed.g2.bx.psu.edu looks down from here.

..so this might be an temporary network problem that is not related to the "localhost" bit.

I am nevertheless unable to verify the claim of the ease of using the Tool Shed to install

the BLAST+ tool because of this.

If it is true that the Docker image does not work with the Galaxy Tool Shed - which now

host most of the tools required in a Galaxy installation, then this should be duely noted in the

article and the README of the Docker image.

3: Supporting data has no usage instructions

The article links to https://github.com/peterjc/galaxy_blast
as the supporting data - but this website has no instructions on
how to install/use with Galaxy or the Galaxy docker image.
I could execute .travis.yml "by hand" - but I do not feel this
is sufficient documentaton for a supporting data set.

I have therefore not been able to verify that the supporting
data actually supports the article, beyond inspecting the
Travis-CI build logs at
https://travis-ci.org/peterjc/galaxy_blast/builds

.. which except for a single error seem to be verifying
the tools. https://travis-ci.org/peterjc/galaxy_blast/builds/45137901

> OperationalError: (OperationalError) unable to open database file None None

Minor Essential Revisions

4: Provenance and update issue not addressed

Workflow systems are commonly praised in bioinformatics because they enable reproducibility and sharing of analytical pipelines.

One challenge in this aspect is that the domain of bioinformatics commonly update software tools and reference datasets.

In fact, BLAST+ 2.2.30 was
released just 6 weeks ago [ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/]
and the latest BLAST reference dataset taxdb.tar.gz was updated
today [ftp://ftp.ncbi.nlm.nih.gov/blast/db/].

The blast FTP site does not seem to contain any version number of the dataset, and as datasets are split over multiple files, it would be difficult to know if you have downloaded half an old dataset and half a new dataset. (A new version of the dataset could be released in the middle of your lengthy download). The sanity of the dataset could potentially be verified by downloading the *.md5 files both before and after the large download -- but this should be automated by a script to be done correctly.

I would say the main challenges are in this respect, assuming a successful workflow
run using the described Galaxy BLAST tools:

a) Which version of the BLAST tool was used?

b) Which reference data set was used?

c) Which version?

d) Was the install complete/sane? (ref. MD5 files and updates)

e) Are there any later versions of tool or reference data set? How do I keep my Galaxy instance up to date? (going through the lengthy database download+config again?)

f) How can a Galaxy workflow using BLAST+ be shared with another Galaxy instance (supposedly easily started with Docker), when manual download and configuration of databases are required?

Your article does not mention how the BLAST+ tool for Galaxy addresses any of these
concerns. The use of the Galaxy Tool Shed should in theory allow for automatic updating of the tool - and I believe the BLAST tools would output log information that includes at least version number.

I am worried that the dataset description that is entered manually by the system administrator into /galaxy-central/tool-data/blastdb.loc and friends contain an element of "manual versioning", as the example contains

#nt_02_Dec_2009 nt 02 Dec 2009 /depot/data2/galaxy/blastdb/nt/nt.chunk
#wgs_30_Nov_2009 wgs 30 Nov 2009 /depot/data2/galaxy/blastdb/wgs/wgs.chunk

This sounds very errorprone, and as older datasets not available from NCBI, definitiely not reproducible.

I would expect the article to at least acknowledge these concerns - and ideally for the tooling to support this (e.g. through the BLAST Data Manager and additional provenance output from the BLAST+ tools, e.g. in W3C PROV format).

5: Results workflows unavavailable

> We now describe some use-cases and workflows combining
> these tools within Galaxy.

The first two examples:
- Assessing a de novo assembly
- Finding genes of interest in a de novo assembly

do not link to any actual Galaxy workflow descriptions, but
are only described as bullet point lists.

The descriptions do not link to any examples for "Upload ** sequence" or of the expected outputs.

"Identifying candidate genes clusters" is described in more detail,
but the workflow is only included as a visual Figure, and not
in the supporting data or uploaded/linked to an external repository like the mentioned myExperiment.

The citation for this workflow, [22] http://dx.doi.org/10.1021/ja501630w is not Open Access, and I was required to use the University of Manchester.
The article does not mention the word "workflow" once, and do not seem to contain any data citations for the workflow, only for the sequence.

The only supporting information provided at http://pubs.acs.org/doi/suppl/10.1021/ja501630w is a PDF with tables, graphs and sequence views. Again the word "workflow" is not mentioned.

A direct link to the workflow definition should be included for all three examples.

Discretionary Revisions

Spelling corrections for product/company names

p2:
- MyExperiment -> myExperiment
- Amazon Inc. -> Amazon AWS
- "Cloud Computing" -> Cloud Computing
- Galaxy "CloudMan" -> Galaxy CloudMan
- "Galaxy Tool Shed" -> Galaxy Tool Shed

p5:
- "such FASTA format" -> "such as FASTA format"
- Docker Inc. -> Docker Inc. (https://www.docker.com/)

- Galaxy "CloudMan" -> Galaxy CloudMano

Useful hyperlinks:

- whose functional tests are then run.
-> whose ..then run (https://travis-ci.org/peterjc/galaxy_blast).
- The Galaxy-P project -> The Galaxy-P project https://usegalaxyp.org/

Level of interest: An article of importance in its field

Quality of written English: Acceptable

Statistical review: No, the manuscript does not need to be seen by a statistician.

Declaration of competing interests:

I declare that I have no competing interests.

I follow the main author's Twitter account @pjacock.