

Integrating the NCBI BLAST+ suite into Galaxy

https://github.com/peterjc/galaxy_blast

<http://slideshare.net/pjacock/>

1 July 2014, GCC2014, Baltimore



The James
Hutton
Institute

Peter Cock¹, John Chilton², Björn Grüning³, Jim Johnson⁴, Nicola Soranzo⁵

¹ James Hutton Institute, Scotland, UK; ² Penn State University, USA; ³ Albert-Ludwigs-University, Germany; ⁴ University of Minnesota, USA; ⁵ CRS4, Italy



Everyone loves BLAST!

- Altschul *et al.* (1990)
Basic local alignment search tool.
[http://doi.org/10.1016/S0022-2836\(05\)80360-2](http://doi.org/10.1016/S0022-2836(05)80360-2)
- ...
- Camacho *et al.* (2009)
BLAST+: architecture and applications.
<http://doi.org/10.1186/1471-2105-10-421>



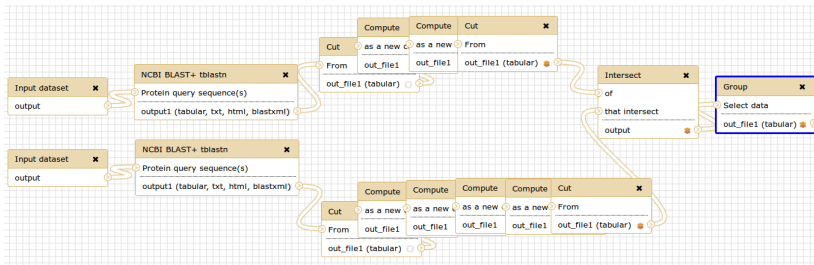
Example use cases and workflows

- Homology searches / functional annotation
- Contamination detection
- Identification of unmappable reads
- Reciprocal best BLAST hits
- BLAST2GO
- Identification of nearby located genes (Gene cluster searches)



Example use cases and workflows

- Identification of nearby located genes (Gene cluster searches)





Open Source Licensing → Ecosystem

- NCBI BLAST and BLAST+
 - Public Domain (United States Government Work)
- Galaxy
 - Academic Free License version 3.0
- BLAST+ wrappers for Galaxy
 - MIT Licence

Like *most* Bioinformatics tools, these are all open source :)

- This is important for redistribution via the Galaxy Tool Shed



Galaxy BLAST wrapper early history

- Jan 2008 – Galaxy team wrapped NCBI ‘legacy’ MEGABLAST
- Sep 2010 – I contributed wrappers for NCBI BLAST+ core
 - Development within main [Galaxy bitbucket.org repository](https://bitbucket.org/galaxy)
- August 2012 – BLAST+ wrapper moved to ToolShed
 - Development on [my Galaxy bitbucket.org branch](https://bitbucket.org/galaxy)
- Summer 2013 – Published Galaxy tools paper
 - Cock, Grüning, Paszkiewicz, and Pritchard (2013)
Galaxy tools and workflows for sequence analysis with
applications in molecular plant pathology
<http://doi.org/10.7717/peerj.167>
- Summer 2013 – Reorganised to facilitate contributions



Galaxy BLAST wrapper recent history

- July 2013
 - Held [Birds-of-a-Feather meeting at GCC2013](#)
 - [BLAST+ wrapper development moved to GitHub](#)
 - Adopted MIT Licence
 - Merged first pull request on GitHub
- Sept 2013 – updated to BLAST+ 2.2.26
 - Added RPSBLAST, RPSTBLASTN, and protein domain databases
- Dec 2013 – updated to BLAST+ 2.2.28
 - Added description in tabular output, \$GALAXY_SLOTS, macros, masking, etc
- March 2014 – updated to BLAST+ 2.2.29
 - Added pick-your-own columns, more masking, etc
- We're also working on a paper...



Source code & history on GitHub

 This repository ▾ ⓘ [Explore](#) [Gist](#) [Blog](#) [Help](#)  [peterjc](#) [+](#) [✂](#) [📄](#)

 [peterjc](#) / [galaxy_blast](#) 👁 Unwatch ▾ 5 ★ Star 5 🍴 Fork 10

📁 branch: master ▾ [galaxy_blast](#) / [tools](#) / [ncbi_blast_plus](#) / [+](#) ⌵ ⓘ

Correct help text for TBLASTX

 **peterjc** authored on 19 May latest commit a314af2506 📄

..

README.rst	Update README & markup	2 months ago
blastxml_to_tabular.py	More detailed help if try to use the script's old API	2 months ago
blastxml_to_tabular.xml	Remove 1 to 25 column numbers from pick-your-own interface	4 months ago
check_no_duplicates.py	Call new script check_no_duplicates.py before makeblastdb to avoid bug	7 months ago
make_ncbi_blast_plus.sh	Fix minor typo	2 months ago
ncbi_blastdbcmd_info.xml	Switch dependency to BLAST+ 2.2.29; Bump wrapper version to v0.1.00	5 months ago
ncbi_blastdbcmd_wrapper.xml	Switch dependency to BLAST+ 2.2.29; Bump wrapper version to v0.1.00	5 months ago
ncbi_blastn_wrapper.xml	Just want 'megablast' or 'blastp' etc in description, not the long de...	4 months ago
ncbi_blastp_wrapper.xml	Just want 'megablast' or 'blastp' etc in description, not the long de...	4 months ago
ncbi_blastx_wrapper.xml	Test picking all the tabular columns as output.	4 months ago

https://github.com/peterjc/galaxy_blast/tree/master/tools/ncbi_blast_plus



Issue Tracker on GitHub (For Bug Reports etc)

11 Open 31 Closed Sort: Recently updated ▾

☐

Close

Label ▾

Assignee ▾

Milestone ▾

☐		Add option to run blastdbcmd tools on blast db in history	#41
Opened by ericenns ⌚ updated on 18 Apr 💬 7 comments			
☐		Data manager for the BLAST database *.loc files? enhancement	#22
Opened by peterjc ⌚ updated on 17 Apr 💬 36 comments			
☐		Sometimes BLAST+ warning messages treated as errors	#40
Opened by peterjc ⌚ updated on 9 Apr			
☐		Support for -remote enhancement	#39
Opened by peterjc ⌚ updated on 19 Mar 💬 7 comments			
☐		Filter BLAST by taxid enhancement	#36
Opened by peterjc ⌚ updated on 19 Mar 💬 1 comment			
☐		Write tests with (history) databases as input	#3
Opened by peterjc ⌚ updated on 19 Mar 💬 4 comments			

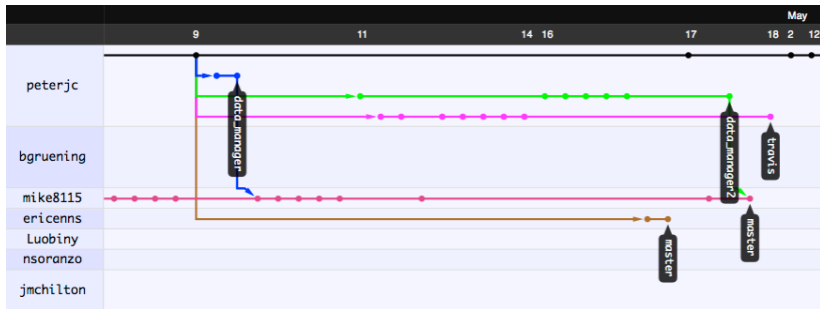
https://github.com/peterjc/galaxy_blast/issues



GitHub contributors

Git & GitHub facilitate an open development model:

- Anyone can “fork” the code to try out modifications
- Can then offer their improvements to be merged back



https://github.com/peterjc/galaxy_blast/network



Changes on GitHub tested via TravisCI

 ravis

HomeBlogStatusHelp

Travis CI for Private Repositories

 Peter Cock

peterjc/galaxy_blast

build passing

Galaxy wrappers for NCBI BLAST+ and related BLAST tools.

CurrentBuild HistoryPull RequestsBranch Summary

Build	Message	Commit	Duration	Finished
 209	Forgot to bundle new test file	6afb2b7 (master)	57 min 51 sec	7 days ago
 208	Fix typo in blast_rbh tests	61d58cb (master)	1 hr 10 min 33 sec	8 days ago
 207	Use optparse in blast_rbh, easier to use standalone now	c2f624c (master)	1 hr 8 min 49 sec	8 days ago
 206	blast_rbh v0.1.1, support self-comparison	b16bf63 (master)	1 hr 6 min 41 sec	8 days ago
 205	Forgot to checkin k12_ten_proteins.fasta	6dd9c39 (master)	1 hr 5 min 33 sec	28 days ago
 204	Check return code	5579169 (master)	56 min 40 sec	28 days ago

<http://blastedbio.blogspot.co.uk/2013/09/using-travis-ci-for-testing-galaxy-tools.html>

https://travis-ci.org/peterjc/galaxy_blast/builds



Development & Testing Process

General process,

- Development & local testing on galaxy-central (on internal Galaxy server for development, using our cluster)
- Push to GitHub, runs TravisCI tests (also galaxy-central)

Then, for a release-candidate,

- Update *Test* Tool Shed (also on galaxy-central)
- Local testing on our live server (using galaxy-dist)
- Encourage others to test on their systems

Finally,

- Update *main* ToolShed (tested on galaxy-dist)



All dependencies via Tool Shed

- Main BLAST+ tool wrappers
 - http://toolshed.g2.bx.psu.edu/view/devteam/ncbi_blast_plus/
- Packages for NCBI BLAST+ binaries
 - http://toolshed.g2.bx.psu.edu/view/iuc/package_blast_plus_2_2_26
 - http://toolshed.g2.bx.psu.edu/view/iuc/package_blast_plus_2_2_27
 - http://toolshed.g2.bx.psu.edu/view/iuc/package_blast_plus_2_2_28
 - http://toolshed.g2.bx.psu.edu/view/iuc/package_blast_plus_2_2_29
- BLAST datatypes
 - http://toolshed.g2.bx.psu.edu/view/devteam/blast_datatypes/

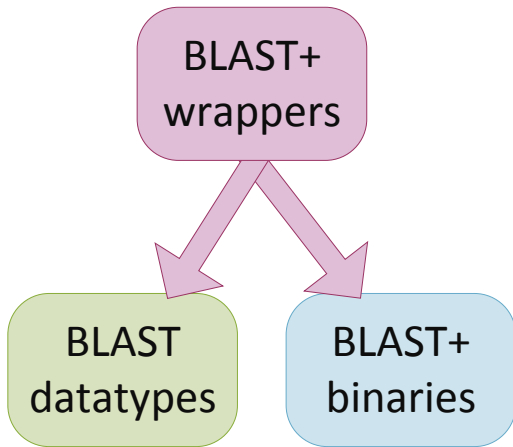
(and same set again for the Test Tool Shed)

See: Blankenberg *et al.* (2014)
<http://doi.org/10.1186/gb4161>



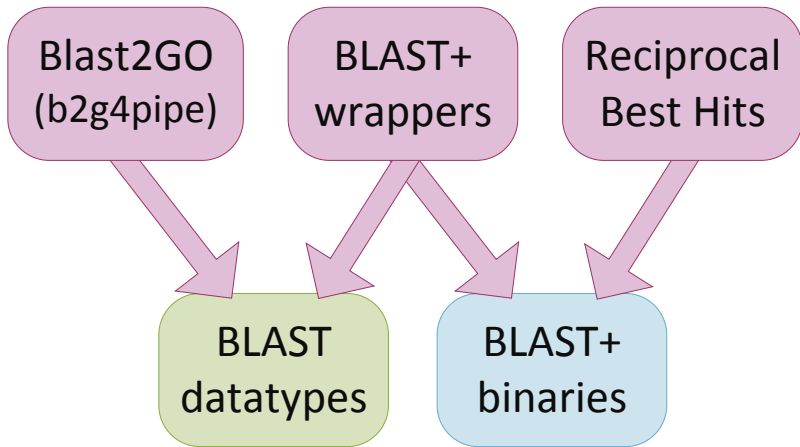


All dependencies via Tool Shed





All dependencies via Tool Shed





Galaxy datatypes used in BLAST

- `tabular` – BLAST tabular output
- `txt` – BLAST plain text output
- `html` – BLAST webpage output
- `fasta` – FASTA sequence files



New Galaxy datatypes for BLAST

- maskinfo-asn1 and maskinfo-asn1-binary – Masking Info
- pssm-asn1 – Position Specific Scoring Matrices (PSSMs)
- Simple subclasses of Galaxy's ASN.1 datatypes
- Each defined by one line of XML in `datatypes_conf.xml`:

```
<datatype extension="pssm-asn1"  
  type="galaxy.datatypes.data:GenericAsn1"  
  mimetype="text/plain" subclass="True"  
  display_in_upload="true" />
```

https://github.com/peterjc/galaxy_blast/blob/master/datatypes/blast_datatypes/datatypes_conf.xml



New Galaxy datatypes for BLAST

- blastxml – BLAST XML output
 - Python subclass of Galaxy's XML datatype
 - Defines `.sniff(...)` method for detecting on upload
 - Defines `.merge(...)` method for job splitting

```
class BlastXml(GenericXml):  
    """NCBI Blast XML Output data."""  
    file_ext = "blastxml"  
  
    def sniff(self, filename):  
        ....  
  
    def merge(split_files, output_file):  
        ...
```



New Galaxy datatypes for BLAST

- blastdbn, blastdbp, blastdbd – BLAST databases
 - Composite datatypes made up of multiple files
 - Python subclasses of Galaxy's base datatype

```
class BlastNucDb(...):  
    """Nucleotide BLAST database files."""  
    file_ext = "blastdbn"  
    allow_datatype_change = False  
    composite_type = "basic"  
  
    ...
```

https://github.com/peterjc/galaxy_blast/blob/master/datatypes/blast_datatypes/blast.py



Galaxy Macros

- BLAST+ tool suite had a lot of repeated XML
 - Consistency/maintenance was hard
- Galaxy does not support:
 - XML `<!ENTITY ...>` tags
 - XInclude
- Galaxy has its own macros and token system
 - Downside is more complexity
 - Ideal for tool suites like BLAST+



Galaxy Macros - XML macros

Turn re-used chunks of XML into macros:

```
<xml name="input_conditional_nucleotide_db">
  <conditional name="db_opts">
    <param name="db_opts_selector" type="select" label="Database/sequences">
      <option value="db" selected="True">Locally installed BLAST DB</option>
      <option value="histdb">BLAST database from your history</option>
      <option value="file">FASTA file from history (see warning)</option>
    </param>
    <when value="db">
      <param name="database" type="select" label="Nucleotide BLAST database">
        <options from_file="blastdb.loc">
          ...
        </options>
      </param>
      <param name="histdb" type="hidden" value="" />
      <param name="subject" type="hidden" value="" />
    </when>
    <when value="histdb">
      ...
    </when>
    <when value="file">
      ...
    </when>
  </conditional>
</xml>
```

https://github.com/peterjc/galaxy_blast/blob/master/tools/ncbi_blast_plus/ncbi_macros.xml



Galaxy Macros - Cheetah tokens

Turn re-used chunks of Cheetah markup (or help text) into tokens:

```
<token name="@BLAST_DB_SUBJECT@">
#if $db_opts.db_opts_selector == "db":
    -db "${db_opts.database.fields.path}"
#elif $db_opts.db_opts_selector == "histdb":
    -db "${os.path.join($db_opts.histdb.
        extra_files_path, 'blastdb')}"
#else:
    -subject "$db_opts.subject"
#end if
</token>
```

https://github.com/peterjc/galaxy_blast/blob/master/tools/ncbi_blast_plus/ncbi_macros.xml



Galaxy Macros – Example Tool

Now can embed the @TOKENS@ and XML fragments:

```
<tool id="ncbi_blastn_wrapper" ...>
  ...
  <macros>
    <import>ncbi_macros.xml</import>
  </macros>
  ...
  <command>
    blastn ... @BLAST_DB_SUBJECT@ ...
  </command>
  ...
  <inputs>
    ...
    <expand macro="input_conditional_nucleotide_db" />
    ...
  </inputs>
  ...
</tool>
```



Galaxy BLAST+ setup at James Hutton Institute

- Task splitting/job parallelization enabled
 - Batches of 1000 query sequences
- Use `$GALAXY_SLOTS=4` to exploit our entire cluster
 - Ensures even our older 4 core nodes are used
- Update NCBI BLAST databases using cron job
 - We don't keep date-stamped versions
- Wrapper script caches BLAST databases on cluster nodes,
<https://github.com/peterjc/picobio/tree/master/blast>
 - Databases on network storage were too slow
 - (BeeGFS currently under evaluation ... talk to Björn)



Work in progress

- Data manager(s) for BLAST databases
- PSI-BLAST
- DELTA-BLAST
- Tests for every tool
- ...
- See https://github.com/peterjc/galaxy_blast/issues



DB configuration via *.loc files

```
$ cat /mnt/galaxy/galaxy-central/tool-data/blastdb.loc
nt NCBI nt nucleotide sequence database /mnt/scratch/local/blast/ncbi/nt
est_others NCBI EST others (non-human, non-mouse) nucleotide sequence database /
  mnt/scratch/local/blast/ncbi/est_others
est_mouse NCBI EST mouse nucleotide sequence database /mnt/scratch/local/blast/
  ncbi/est_mouse
est_human NCBI EST human nucleotide sequence database /mnt/scratch/local/blast/
  ncbi/est_human
est NCBI EST nucleotide sequence database /mnt/scratch/local/blast/ncbi/est
oomycete_genes Oomycete predicted genes /mnt/shared/cluster/blast/galaxy/
  oomycete_genes
oomycete_ests Oomycete ESTs /mnt/shared/cluster/blast/galaxy/oomycete_ests
oomycete_scaffolds Oomycete genome scaffolds /mnt/shared/cluster/blast/galaxy/
  oomycete_scaffolds
...
```

- Data Managers could make this easier...
- Blankenberg *et al.* (2014) Wrangling Galaxy's reference data.
<http://doi.org/10.1093/bioinformatics/btu119>



Acknowledgements – Groups

- NCBI BLAST developers (and their help desk team)
- [Galaxy Community](#)
- [Galaxy Team](#), especially Core Developers & Tool Shed team
- [Intergalactic Utilities Commission](#) (IUC, *“Tool Shed Police”*)
- [GitHub](#) and [Bitbucket](#) for repository hosting
- [TravisCI](#) for testing open source projects free of charge
- Our various testers @ JHI and ALU-Freiburg
- Our funders



Acknowledgements – Individuals

- Björn Grüning (packaging, macros, datatypes, masking, etc)
- Dan Blankenberg (Data Managers)
- Dannon Baker (Tool Shed migration)
- Edward Kirton (datatype for BLAST databases)
- Jim Johnson (tabular output enhancements)
- John Chilton (macros, \$GALAXY_SLOTS, test framework, etc)
- Kanwei Li (merges while using main Galaxy repository)
- Luobin Yang (initial PSI-BLAST work)
- Nicola Soranzo (masking support, datatype work, etc)



Key References

- Camacho *et al.* (2009)
BLAST+: architecture and applications.
<http://doi.org/10.1186/1471-2105-10-421>
- Cock *et al.* (2013)
Galaxy tools and workflows for sequence analysis with
applications in molecular plant pathology.
<http://doi.org/10.7717/peerj.167>
- Galaxy BLAST+ wrappers
https://github.com/peterjc/galaxy_blast

Galaxy BLAST

https://github.com/peterjc/galaxy_blast

<http://slideshare.net/pjacock/>

@pjacock #usegalaxy