Increasing the Utility of Galaxy Workflows

John Chilton (@jmchilton) and the Galaxy Team

The *Illusion* of Galaxy Workflows
Galaxy Workflows - Kind of **Awesome**

- Designed for biologists, **accessibility** - easy to build and easy to run.
- Sharable, Publishable
  - e.g. NCBI BLAST+ integrated into Galaxy - Cock et al. dx.doi.org/10.1101/014043
- Data Flow

“Best Galaxy feature Galaxy users don’t know about.”
The **Illusion of Workflows**

- Galaxy didn’t “schedule” workflows - it would just queue up a bunch of jobs.
  - Therefore Galaxy had no way to conditionally evaluate branches or handle various dynamic functionality one would expect from a workflow.
Data Flow Limitations

"An Automated Pipeline for High-Throughput Label-Free Quantitative Proteomics (J. Proteome Res., 2013, PMID: 23391308)."

Applications run in parallel (once per input)

Merged into one output for subsequent steps.

http://www.slideshare.net/mygrid/2014-taverna-tutorial-advanced-taverna
Addressing These Problems

- Map/reduce style data flow using dataset collections.
- Implemented a workflow engine.
Collection Types

Currently two supported type pseudo-plugins - “list” and “paired”.

- Lists can contain arbitrary number of named elements
- Pairs contain a “forward” and “reverse” element.

Types can be combined to build nested types - for instance “list:paired” describes a list of paired datasets.
Upload Some Data...

Download data directly from web or upload files from your disk

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Type</th>
<th>Genome</th>
<th>Settings</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>M236C4-ch_1.fq</td>
<td>45.4 MB</td>
<td>fastq</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
<tr>
<td>M236C4-ch_2.fq</td>
<td>45.4 MB</td>
<td>fastq</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
<tr>
<td>M48C2-ch_1.fq</td>
<td>46.9 MB</td>
<td>fastq</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
<tr>
<td>M48C2-ch_2.fq</td>
<td>46.9 MB</td>
<td>fastq</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
<tr>
<td>SC14-ch_1.fq</td>
<td>74.4 MB</td>
<td>fastq</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
<tr>
<td>SC14-ch_2.fq</td>
<td>74.4 MB</td>
<td>fastq</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
<tr>
<td>sequence.fasta</td>
<td>16.9 KB</td>
<td>fasta</td>
<td>?</td>
<td>✔️</td>
<td>100%</td>
</tr>
</tbody>
</table>

You can Drag & Drop files into this box.
Select the Pairs
Create a Collection...

### Create a collection of paired datasets

3 pairs created: all datasets have been successfully paired

0 unpaired forward (0 filtered out)

<table>
<thead>
<tr>
<th>Forward Pair</th>
<th>Reverse Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>M236CA-ch_1.fq</td>
<td>M236CA-ch_3.fq</td>
</tr>
<tr>
<td>M486C2-ch_1.fq</td>
<td>M486C2-ch_2.fq</td>
</tr>
<tr>
<td>SOC14-ch_1.fq</td>
<td>SOC14-ch_2.fq</td>
</tr>
</tbody>
</table>

Remove file extensions from pair names? ☑️

Name: [Enter a name for your new list]

Create list

[1] [2] [3]
Collection Mapping (1 / 3)

Tool consumes a FASTQ file.

- List of Paired Datasets

- Individual FASTQ datasets.

What it does

This tool offers several conversion options relating to the FASTQ format.

When using Basic options, the output will be Sanger formatted or cssanger formatted (when the input is Color Space Sanger).

When converting, if a quality score falls outside of the target score range, it will be coerced to the closest available value (i.e. the minimum or maximum).
Collection Mapping (2 / 3)

Collection map icon replaces input options with valid collections.

Runs tool over every dataset in list of pairs and produces groomed list of pairs.
Like hiding workflow datasets, they are visible initially and hidden after completion (only collection remains visible).

Collection always green regardless of contents (**stateless**).

Need to do better on both points... not scalable enough.
Sample Tracking: Identifiers + Indices

Paired mt Datasets

- Element - 0:M236C4 (paired collection)
  - forward
    - hda - M236C4-ch_1.fq
  - reverse
    - hda - M236C4-ch_2.fq
- Element - 1:M486C2 (paired collection)
  - forward (hda)
    - hda - M486C2-ch_1.fq
  - reverse (hda)
    - hda - M486C2-ch_2.fq

FASTQ Groomer across collection 8

- Element - 0:M236C4 (paired collection)
  - forward
    - hda - FASTQ Groomer on data 1
  - reverse
    - hda - FASTQ Groomer on data 2
- Element - 1:M486C2 (paired collection)
  - forward (hda)
    - hda - FASTQ Groomer on data 3
  - reverse (hda)
    - hda - FASTQ Groomer on data 4

Mapping over collections - dataset naming is normal, but new collection created with identical tree structure and element identifiers preserved.
Subcollection Mapping

Bowtie 2, Tophat, BWA-mem, Picard, Hitsat, etc... have all been updated to consume paired datasets.
Subcollection Mapping

Bowtie2 (version 0.2)

Is this library mate-paired?:
Paired-end Dataset

FASTQ Paired Dataset:

15: FASTQ Groomer across collection 8
Nucleotide-space: Must have Sanger-scaled quality values with ASCII offset 33

Minimum insert size for valid paired-end alignments:
0

Maximum insert size for valid paired-end alignments:
250

Write unaligned reads to separate file(s):

History

Map/Reduce Test
636.1 MB

19: Bowtie2 across collection 1
5

18: Bowtie2 on data 7, data 9, and others: aligned reads

17: Bowtie2 on data 7, data 9, and others: aligned reads

16: Bowtie2 on data 7, data 9, and others: aligned reads

15: FASTQ Groomer across collection 8
Subcollection Mapping (Identifiers)

**Paired mt Datasets**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>File Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:M236C4</td>
<td>paired collection</td>
<td>M236C4-ch_1.fq</td>
</tr>
<tr>
<td>0:forward</td>
<td>(hda)</td>
<td>M236C4-ch_1.fq</td>
</tr>
<tr>
<td>1:reverse</td>
<td>(hda)</td>
<td>M236C4-ch_2.fq</td>
</tr>
<tr>
<td>1:M486C2</td>
<td>paired collection</td>
<td>M486C2-ch_1.fq</td>
</tr>
<tr>
<td>0:forward (hda)</td>
<td></td>
<td>M486C2-ch_1.fq</td>
</tr>
<tr>
<td>1:reverse (hda)</td>
<td></td>
<td>M486C2-ch_2.fq</td>
</tr>
</tbody>
</table>

**Bowtie 2 across collection 13**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>File Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:M236C4</td>
<td></td>
<td>Bowtie 2 on data 9 and data 10</td>
</tr>
<tr>
<td>1:M486C2</td>
<td></td>
<td>Bowtie 2 on data 11 and data 12</td>
</tr>
</tbody>
</table>

...
Reducing Collections

Modified “Merge BAM Files” tool to use multiple input data parameter instead of two input parameters and a repeat block.
Reducing Collections

Can dynamically substitute collection for the multiple selection of datasets.
# Extract a Workflow

## Dataset Collection Creation
- Dataset collection created in a way not compatible with workflows

## FASTQ Groomer
- Include "FASTQ Groomer" in workflow

## Bowtie2
- Include "Bowtie2" in workflow

## Merge BAM Files
- Include "Merge BAM Files" in workflow

## flagstat
- Include "flagstat" in workflow

## Steps:

1. **Treat as input dataset**
   - **7: M486C2**
   - **Treat as input dataset**
   - **8: Paired mt Datasets**
     - **Treat as input dataset**
     - **13: FASTQ Groomer across collection 8**
   - **16: Bowtie2 across collection 13**
   - **17: NewBam.bam**
   - **19: NewBam_Merge BAM Files.log**
   - **19: flagstat on data 17**

2. **NewBam.bam**
3. **NewBam_Merge BAM Files.log**
4. **sequence.fasta**
5. **M486C2-ch_2.fq**
6. **M236C4**
7. **M236C4-ch_1.fq**
8. **M486C2-ch_1.fq**
9. **flagstat**
10. **Bowtie2 across collection 13**
11. **FASTQ Groomer across collection 8**
12. **Paired mt Datasets**
13. **M486C2**
14. **Treat as input dataset**
15. **Dataset collection created in a way not compatible with workflows**
16. **Dataset collection created in a way not compatible with workflows**
17. **Dataset collection created in a way not compatible with workflows**
18. **Dataset collection created in a way not compatible with workflows**
More Powerful Workflows

Run applications in parallel (one per input).

Arbitrary # of Inputs (...paired).

Merged output for subsequent processing.
RNA-Seq workflow based using the Tuxedo suite.
Core phylogenomics SNP pipeline by Aaron Petkau, Gary Van Domselaar, Philip Mabon, and Lee Katz. Used to assist in outbreak response for food-born illness by the public health agency of Canada. Process hundreds of paired strains at a time.

Protein identification of mass spectrometry data using Open MS. Tools and workflow by Torsten Houwaart

Banner Year for Tool Development

- In 2015, we have a had a real focus on tool development - new & updated tools for many areas including RNA-seq and metagenomics - with collection compatibility being a large focus.

- Support for collection aware read-group handling for BWA, Bowtie 2, Picard.
Parallelization in Galaxy

Marco Albuquerque, et. al.
Workflow - Rewrite

- **Stateful** models allowing re-evaluating workflows over time. Large or complex workflows will now be evaluated in the background.
- Plugin framework for describing how scheduling occurs.

... groundwork for future enhancements - still must build new UI elements and modules (loops, conditionals) to maximize the utility of this...
Somatic SNV Workflow
More output collections.

Similar approach by Kyle Ellrott @ UCSC.

Using biobambam to split a bam file, mapping with BWA, and then merging the results.

https://github.com/ucscCancer/pcawg_tools/tree/master/tools/pcap_tools
“Implicit” Connections between Steps

Steps can wait arbitrarily on other steps without needing to specify an explicit input-to-output “data flow”.

Use for admin workflows to populate reference data.

Toward 10,000 samples (beyond collections)

- **Optimize database** interactions, tool execution.
- Move **workflow scheduling** into own process, optimize.
- Differentiate between cluster failures and tool failures.
  - Retry later on cluster failures.
  - Retry on different cluster or with different resource params on failures.
- Optimize disk usage - **streaming**
- More diverse and bigger compute and storage
  - Separate metadata calculation out into its own “job”
  - XSEDE
  - More portable dependency management (docker, Homebrew, tool shed installs without galaxy)
Thanks!

The Galaxy Team

Enis Afghan  Daniel 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

The Galaxy Community for building awesome stuff with Galaxy and pushing the platform forward - especially Philip Mabon.

With special thanks to Carl Eberhard - for building UI powering this work.
Should I CWL?

- Definitely - but it is not the best way to reach the large Galaxy community today.
- CWL is not in Galaxy today and may never be.
- CWL tools and workflows might never provide user experience of Galaxy native.
- Tool authors should usegalaxy.
Extra Content
Bam Splitting Workflow

Add a BAM splitting workflow example.
- biobam split
- bwa-mem
- reheader merge
Splitting a BAM File

Something as simple as splitting a BAM file though is a real problem. The number and nature of one job cannot be determined until the previous one completed.
**MergeSamFiles** merges multiple SAM/BAM datasets into one (Galaxy Tool Version 1.126.0)

**Select SAM/BAM dataset or dataset collection**

- [ ] No sam or bam dataset available.
- Dataset collection

If empty, upload or import a SAM/BAM dataset.

**Merge the sequence dictionaries of the datasets being merged**

- [ ] Yes
- [ ] No

MERGE_SEQUENCE_DICTIONARIES; default=False

**Assume the input file is already sorted**

- [ ] Yes
- [ ] No

ASSUME_SORTED; default=False

**Comment**

[Insert Comment]
You can provide multiple comments.

**Select validation stringency**

- [ ] Lenient

Setting stringency to SILENT can improve performance when processing a BAM file in which variable-length not otherwise need to be decoded.

[Execute]
A handful of reduction tools need to be updated (so will tools consuming pairs). Using multiple input data parameters instead of repeat parameters will still allow these tools to work with uncollected dataset.

repeat blocks - while cumbersome - allow duplicated entries & control of order. Multiple input data parameters should be enhanced to have same control.
Models

Lots of changes but the yellow boxes are the core additions.
"An Automated Pipeline for High-Throughput Label-Free Quantitative Proteomics (J. Proteome Res., 2013, PMID: 23391308)."

Arbitrary # Inputs

Applications run in parallel (once per input)

Merged into one output for subsequent steps.
More Powerful Workflows

- Run applications in parallel (one per input).
- Merged output for subsequent processing.

Arbitrary # of Inputs (... paired).
API First Development

Initial work focused on building an API for creating and using dataset collections.

Upshot - API is richer than UI currently (especially in stable).

bioblend contains high-level functionality for creating and “viewing” collections in different ways.
Tool Parameters - Cheetah-isms

Common paired data idiom:

```python
bowtie $collect_param.forward $collect_param.reverse
```

Common list data idiom:

```python
#for $f in $collect_param# $f #end for#
```

-or-

```python
#for $name in $collect_param.keys()# $f[$name] #end for#
```

Nested data:

```python
#for $f in $collect_param# $f.is_collection ...
```
Tool Parameters - Testing

<test>
  <param name="collect_param">
    <collection type="paired">
      <element name="forward" value="simple_line.txt" />
      <element name="reverse" value="simple_line_alternative.txt" />
    </collection>
  </param>
</test>

...
Plan: Multiple-Data Improvements

Enhance multiple input data parameters to allow control of order and repeated entries.

All the ease of multiple data inputs with actually greater versatility than placing simple data inputs into repeat blocks.

An advanced “add to selection” modal would provide interesting room to grow - options for importing library datasets, digging into collections, etc....
Plans - Other

- [https://trello.com/c/WodW2sLb](https://trello.com/c/WodW2sLb)
- Subcollection mapping over multiple data parameters.
- Fix history import/export for data collections.
- Implicit conversion
- Allow batch input of collections to workflows
Tool Parameters - Tool XML

```
<param name="collect_param1" type="data_collection" format="bam" collection_type="paired" />
```

Optional - filter collections by contained formats.

Optional - filter collections by collection_type.
Todo:

- Screenshots of building up workflow from scratch?
- Extra Slides (post presentation)...
- Comparison with multiple file datasets.

Redo initial screenshots with correct history name on bigger monitor.
Building Collections...

```python
>>> from bioblend import galaxy
>>> gi = galaxy.GalaxyInstance(url="localhost:8080", key="db53bb4500dfaeda25ceb378069b722b")
>>> hist = gi.histories.get_histories(name="Map/Reduce Test")[0]
>>> gi.histories.show_history(hist["id"], contents=True, deleted=False)
>>> pair1_id = [d for d in gi.histories.show_history(hist["id"], contents=True) if d["hid"] == 5][0]["id"]
>>> pair2_id = [d for d in gi.histories.show_history(hist["id"], contents=True) if d["hid"] == 6][0]["id"]
>>> gi.histories.update_dataset_collection(hist["id"], pair1_id, name="M236C4")
>>> gi.histories.update_dataset_collection(hist["id"], pair2_id, name="M486C2")
```

bioblend contains support for creating, reading, updating (name, annotations, etc…), and deleting history dataset collections.

https://github.com/afgane/bioblend/commit/f8d40b687be4c699d608e930c5972673922fa0a
Tool consumes a FASTQ file.

- List of Paired Datasets
- Paired Datasets
- Individual FASTQ datasets.
Like hiding datasets in workflow execution, datasets are visible running or queued and they are hidden after (and only collection is visible).

Collection is always green regardless of contents - is currently stateless.

Need to do a better job on both points - this is not too scalable - but it was an easy quick win.
Plans - UI for Creating Collections

https://trello.com/c/CIIdaxl2  Mockup @ mybalsamiq

The middle section is a scrollable table divided into two parts: the upper paired section and the lower unpaired section. Filtering only affects the unpaired section.

A: Color background color, font, and justification can all be used to differentiate paired/unpaired.

When the user clicks on an unpaired forward then an unpaired reverse (or vice versa) a pair is created. That pair is moved to the bottom of the paired section of the table.

Each row in the 'Pairs' section of the list will have some control to unpair that pair: When clicked, the row disappears and the two files go back to the unpaired/lower section of the table in the appropriate, sorted order.

Alternately, we can send the user to a second pane (2nd 'Wizard' step) to review and reorder the final list. (An option to move back to this step should also be there.)
Why not repeat replacements?

In its most simple form - allowing replacement of one repeat block with a collection - this feature would be gross to implement - it would add a lot of complexity to already complex parts of Galaxy.

... and it would not work with any tools.
Concatenate (Easiest Reduction)

Not just a repeat, would need to be able to dynamically replace input + repeat to work with this. That will be ugly and will have implications all over.
Merging Bams

Second most common reduction - has two inputs and a repeat. So we need to be able to dynamically replace any number inputs and a repeat. Hmm....
Merging BedGraph

Found another reduction tool on main. Multiple inputs, multiple extra options. How could this reasonably allow collection replacement at the infrastructure level.
Plans - More Options in History Panel

https://trello.com/c/hnmWWKlB

Currently can hide, delete, and see name.

Cannot rename, rerun, see type, see contents, see/add annotations, see/add tags, download, etc...
Incorporate collection builder when uploading files (or vise versa).
Plans - UI for Viewing Collections

https://trello.com/c/PVdbbpQS
Plans - Store Collections in Data Libraries

https://trello.com/c/3axmjaxE
Plans - Improved Reductions

https://trello.com/c/lp5YmA1O

Improvements to multiple data parameters described earlier and/or ability to reduce across repeat statements.
Main Goal: Filter out the failed datasets and keep going.

Would like more general filters - filter on metadata (file size, number of sequences, etc...)

Needs to be trackable so can extract and execute in workflows. May require delayed workflow evaluation.
Docker... Docker... Docker...

https://github.com/jmchilton/gcc2014_demo