



#usegalaxy

# Increasing the Utility of Galaxy Workflows

John Chilton (@jmchilton) and the Galaxy Team

<http://bit.ly/bosc2015workflows>

# 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](https://doi.org/10.1101/014043)
- Data Flow
  - Blog by Samuel Lampa <http://bionics.it/posts/workflows-dataflow-not-task-deps>

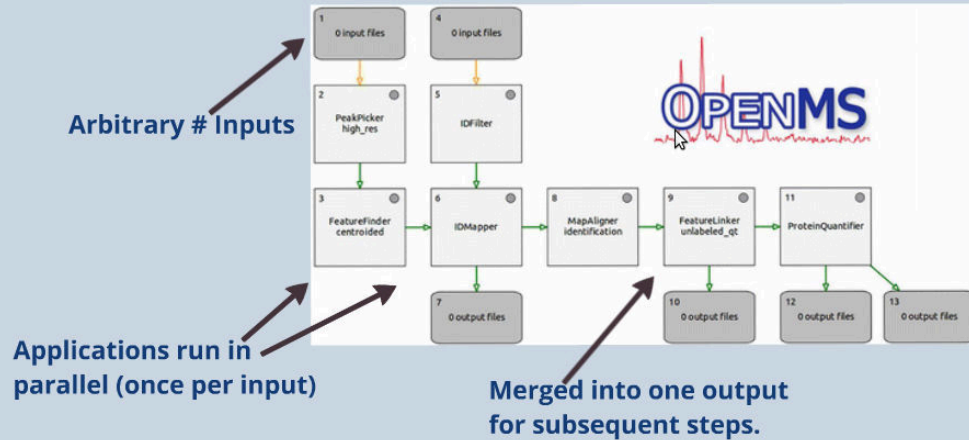
“Best Galaxy feature Galaxy users don’t know about.”

# 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.

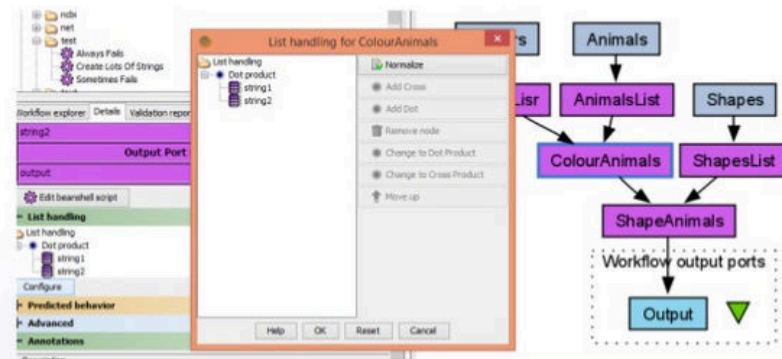
# Data Flow Limitations

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



MANCHESTER  
1824  
The University of Manchester

List handling— configuring -



<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

Name	Size	Type	Genome	Settings	Status
 M236C4-ch_1.fq	45.4 MB	fastqsanger  	unspecified (?) 		100% 
 M236C4-ch_2.fq	45.4 MB	fastqsanger  	unspecified (?) 		100% 
 M486C2-ch_1.fq	46.9 MB	fastqsanger  	unspecified (?) 		100% 
 M486C2-ch_2.fq	46.9 MB	fastqsanger  	unspecified (?) 		100% 
 SC14-ch_1.fq	74.4 MB	fastqsanger  	unspecified (?) 		100% 
 SC14-ch_2.fq	74.4 MB	fastqsanger  	unspecified (?) 		100% 
 sequence.fasta	16.9 KB	fasta  	unspecified (?) 		100% 

You can Drag & Drop files into this box.

Choose local file

Paste/Fetch data

Start

Pause

Reset

Close



# Select the Pairs

History

Map/Reduce Test

318.1 MB

All None

Operations on multiple datasets

- 7: sequence.fasta
- 6: SC14-ch 2.fq
- 5: SC14-ch 1.fq
- 4: M486C2-ch 2.fq
- 3: M486C2-ch 1.fq
- 2: M236C4-ch 2.fq
- 1: M236C4-ch 1.fq

All None For all selected...

- 9: SC...
- 8: SC...
- 7: se...
- 6: M...
- 5: M...
- 4: M236C4-ch 2.fq
- 3: M486C2-ch 2.fq

- Hide datasets
- Unhide datasets
- Delete datasets
- Undelete datasets
- Build Dataset List
- Build Dataset Pair
- Build List of Dataset Pairs

# Create a Collection..

Galaxy Analyze Data Workflow Shared Data Visualization Admin Help User

### Create a collection of paired datasets

3 pairs created: all datasets have been successfully paired

0 unpaired forward - (0 filtered out) [Choose filters](#) [Clear filters](#) 0 unpaired reverse - (0 filtered out)

3 paired [Unpair all](#)

M236C4-ch_1.fq →	M236C4-ch	← M236C4-ch_2.fq	🗑
M486C2-ch_1.fq →	M486C2-ch	← M486C2-ch_2.fq	🗑
SC14-ch_1.fq →	SC14-ch	← SC14-ch_2.fq	🗑

Remove file extensions from pair names?

Name:

# Collection Mapping (1 / 3)

The image shows a screenshot of the FASTQ Groomer application interface. The main window is titled "FASTQ Groomer (version 1.0.4)". It features a "File to groom:" section with a dropdown menu showing "6: SC14-ch\_2.fq". Below this is the "Input FASTQ quality scores type:" section with a dropdown menu showing "Sanger & Illumina 1.8+". There is also an "Advanced Options:" section with a "Hide Advanced Options" dropdown. An "Execute" button is located at the bottom left of the main window. To the right of the main window is a "History" panel with a refresh and settings icon. The history panel lists several datasets, including "Map/Reduce Test" (318.1 MB), "8: Paired mt Datasets", "7: sequence.fasta", "6: SC14-ch\_2.fq", "5: SC14-ch\_1.fq", "4: M486C2-ch\_2.fq", "3: M486C2-ch\_1.fq", "2: M236C4-ch\_2.fq", and "1: M236C4-ch\_1.fq". Each item in the history panel has an eye icon, a pencil icon, and an 'x' icon. Annotations in orange text with blue arrows point to various parts of the interface: "Tool consumes a FASTQ file." points to the file dropdown; "-List of Paired Datasets" points to the "Execute" button; "-Individual FASTQ datasets." points to the "What it does" section; and another arrow points from the "What it does" section to the "History" panel.

FASTQ Groomer (version 1.0.4)

File to groom:

Input FASTQ quality scores type:

Advanced Options:

*Tool consumes a FASTQ file.*

*-List of Paired Datasets*

*-Individual FASTQ datasets.*

History

Map/Reduce Test  
318.1 MB

8: Paired mt Datasets

7: sequence.fasta

6: SC14-ch\_2.fq

5: SC14-ch\_1.fq

4: M486C2-ch\_2.fq

3: M486C2-ch\_1.fq

2: M236C4-ch\_2.fq

1: M236C4-ch\_1.fq

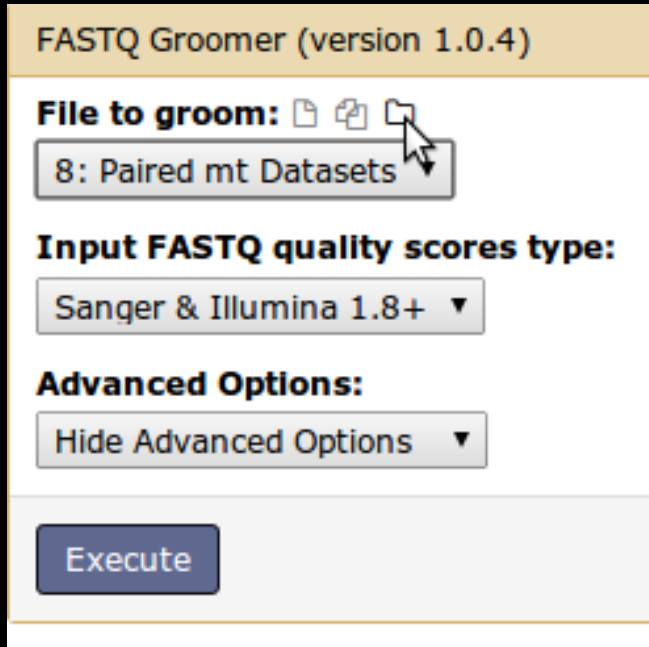
What it does

This tool offers several conversions 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)



FASTQ Groomer (version 1.0.4)

**File to groom:**

**Input FASTQ quality scores type:**

**Advanced Options:**

Collection map icon replaces input options with valid collections.

Runs tool over every dataset in list of pairs and produces groomed list of pairs.

# Collection Mapping (3 / 3)

15: FASTQ Groomer across collection 8 [x]

14: FASTQ Groomer on data 6 [eye] [pencil] [x]

13: FASTQ Groomer on data 5 [eye] [pencil] [x]

12: FASTQ Groomer on data 4 [eye] [pencil] [x]

11: FASTQ Groomer on data 3 [eye] [pencil] [x]

10: FASTQ Groomer on data 2 [eye] [pencil] [x]

9: FASTQ Groomer on data 1 [eye] [pencil] [x]

8: Paired mt Datasets [x]

15: FASTQ Groomer across collection 8 [x]

14: FASTQ Groomer on data 6 [eye] [pencil] [x]

13: FASTQ Groomer on data 5 [eye] [pencil] [x]

8: Paired mt Datasets [x]

15: FASTQ Groomer across collection 8 [x]

8: Paired mt Datasets [x]

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

list:paired collection

Element - 0:M236C4 (paired collection)

Element - 0:forward

hda - M236C4-ch\_1.fq

Element - 1:reverse

hda - M236C4-ch\_2.fq

Element - 1:M486C2 (paired collection)

Element - 0:forward (hda)

hda - M486C2-ch\_1.fq

Element - 1:reverse (hda)

hda - M486C2-ch\_2.fq

...

## FASTQ Groomer across collection 8

list:paired collection

Element - 0:M236C4 (paired collection)

Element - 0:forward

hda - *FASTQ Groomer on data 1*

Element - 1:reverse

hda - *FASTQ Groomer on data 2*

Element - 1:M486C2 (paired collection)

Element - 0:forward (hda)

hda - *FASTQ Groomer on data 3*

Element - 1: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

Bowtie2 (version 0.2)

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

**FASTQ Paired Dataset:**   
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):**

**Will you select a reference genome from your history or use a built-in index?:**  
Use one from the history

Built-ins were indexed using default options

**Select the reference genome:**  
7: sequence.fasta

History

Map/Reduce Test  
636.1 MB



- 15: FASTQ Groomer across collect ion 8
- 8: Paired mt Datasets
- 7: sequence.fasta
- 6: SC14-ch 2.fq
- 5: SC14-ch 1.fq
- 4: M486C2-ch 2.fq
- 3: M486C2-ch 1.fq
- 2: M236C4-ch 2.fq
- 1: M236C4-ch 1.fq

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:

Maximum insert size for valid paired-end alignments:

Write unaligned reads to separate file(s):

History

Map/Reduce Test  
636.1 MB

- 15: FASTQ Groomer across collection 8
- 8: Paired mt Datasets
- 7: sequence.fasta
- 6: SC14-ch 2.fq
- 5: SC14-ch 1.fq
- 4: M486C2-ch 2.fq

Map/Reduce Test  
636.1 MB

- 19: Bowtie2 across collection 1 5
- 18: Bowtie2 on data 7, data 9, and others: ali gned reads
- 17: Bowtie2 on data 7, data 9, and others: ali gned reads
- 16: Bowtie2 on data 7, data 9, and others: ali gned reads
- 15: FASTQ Groomer across collection 8



# Subcollection Mapping (Identifiers)

## Paired mt Datasets

list:paired collection

Element - 0:M236C4 (paired collection)

Element - 0:forward

hda - M236C4-ch\_1.fq

Element - 1:reverse

hda - M236C4-ch\_2.fq

Element - 1:M486C2 (paired collection)

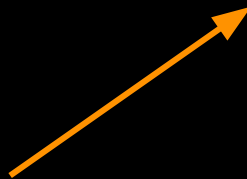
Element - 0:forward (hda)

hda - M486C2-ch\_1.fq

Element - 1:reverse (hda)

hda - M486C2-ch\_2.fq

...



## Bowtie 2 across collection 13

list collection

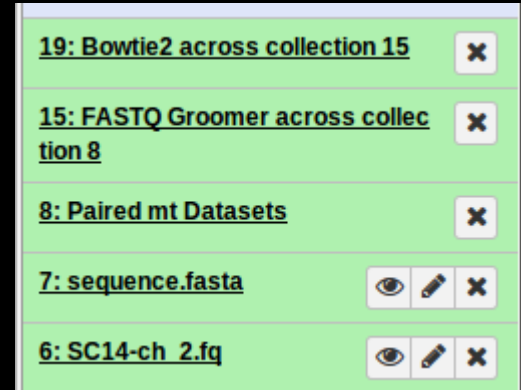
Element - 0:M236C4

hda - *Bowtie 2 on data 9 and data 10*

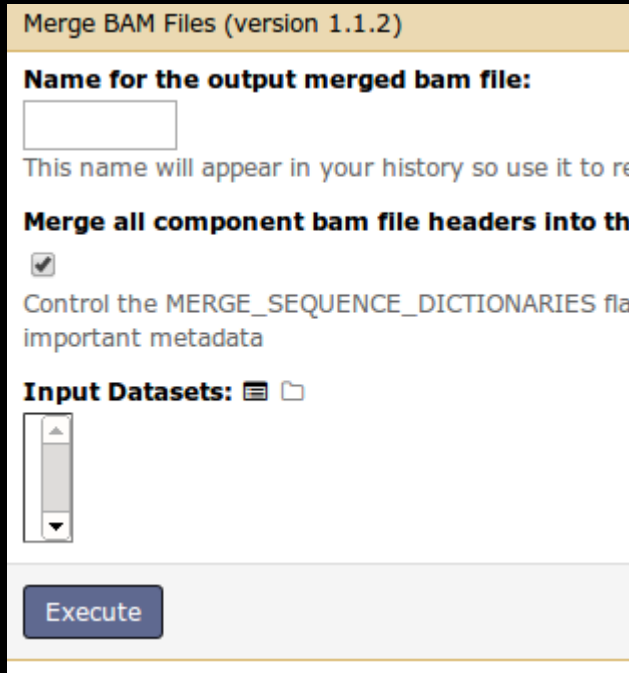
Element - 1:M486C2

hda - *Bowtie 2 on data 11 and data 12*

...





# Reducing Collections



Merge BAM Files (version 1.1.2)

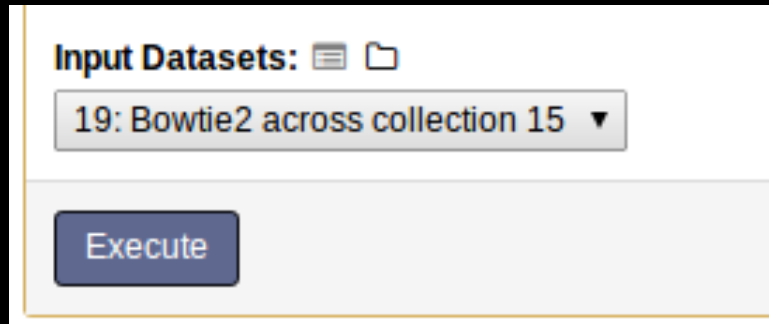
**Name for the output merged bam file:**  
  
This name will appear in your history so use it to re

**Merge all component bam file headers into th**  
  
Control the MERGE\_SEQUENCE\_DICTIONARIES fla  
important metadata

**Input Datasets:**  

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

The screenshot displays a workflow extraction tool interface. On the left, a list of workflow steps is shown, each with a title and a checkbox to include it in the workflow. The steps are:

- Dataset Collection Creation (3 instances)
- FASTQ Groomer (checked)
- Bowtie2 (checked)
- Merge BAM Files (checked)
- flagstat (checked)

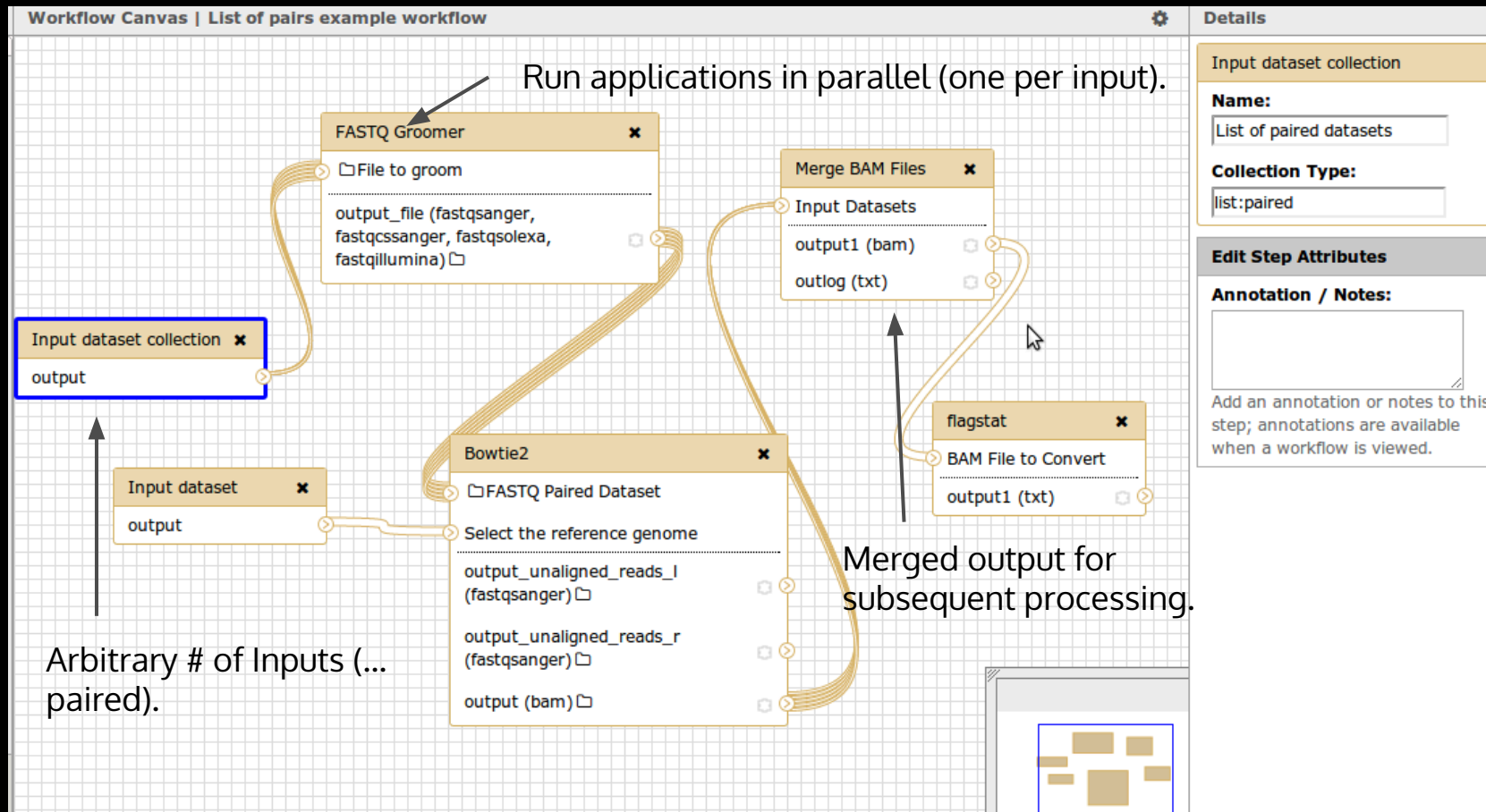
In the center, a detailed view of the selected steps is shown, including:

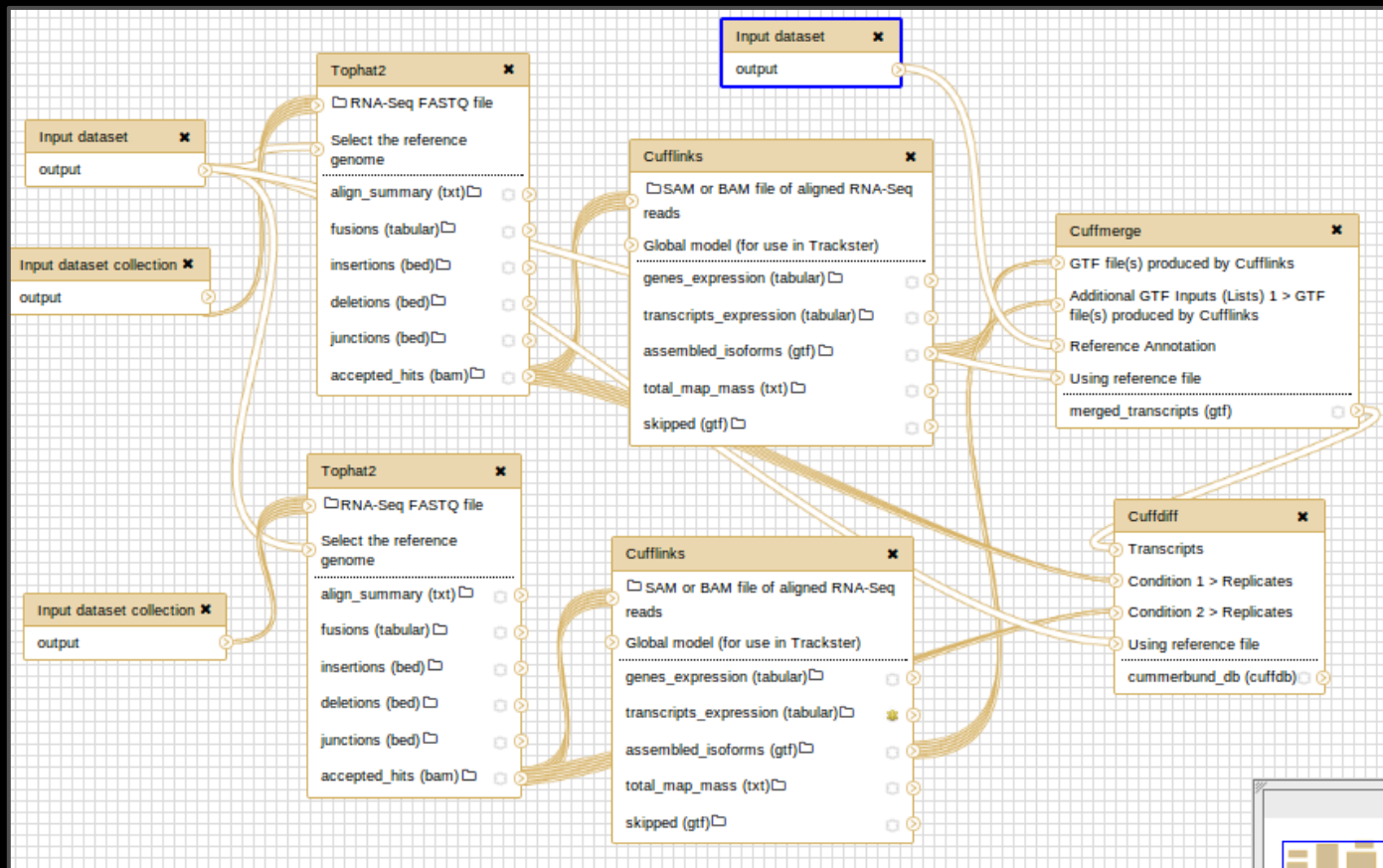
- 7: M486C2 (checkbox unchecked)
- 8: Paired mt Datasets (checkbox checked)
- 13: FASTQ Groomer across collection 8
- 16: Bowtie2 across collection 13
- 17: NewBam.bam
- 18: NewBam\_Merge BAM Files.log
- 19: flagstat on data 17

On the right, a list of workflow steps is shown, each with a title and a set of icons (eye, pencil, x) to manage the step. The steps are:

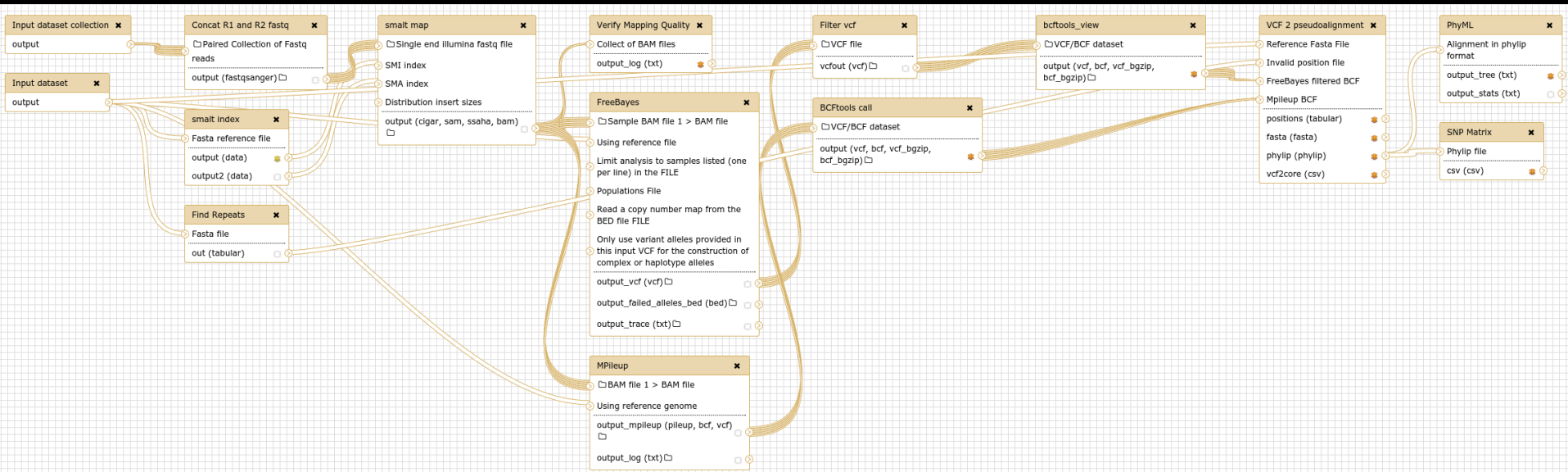
- 19: flagstat on data 17
- 18: NewBam\_Merge BAM Files.log
- 17: NewBam.bam
- 16: Bowtie2 across collection 13
- 13: FASTQ Groomer across collection 8
- 8: Paired mt Datasets
- 7: M486C2
- 6: M236C4
- 5: sequence.fasta
- 4: M486C2-ch 2.fg
- 3: M486C2-ch 1.fg
- 2: M236C4-ch 2.fg
- 1: M236C4-ch 1.fg

# More Powerful Workflows



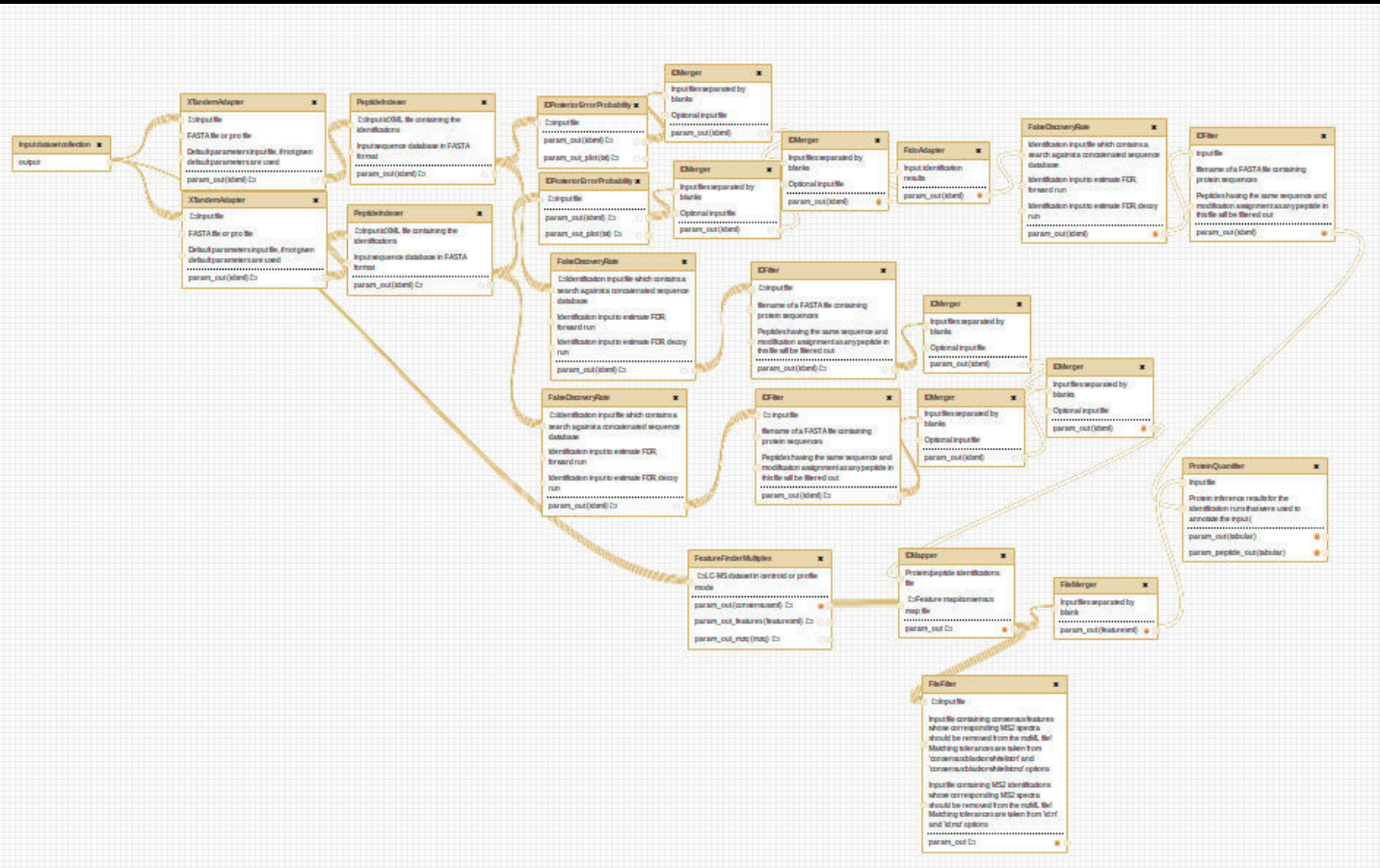


RNA-Seq workflow based using the **Tuxedo suite**.



[Core phylogenomics SNP pipeline](#) by Aaron Petkau, Gary Van Domselaar, Philip Mabon, and Lee Katz. Used to assist in outbreak response for food-borne illness by the public health agency of Canada. Process hundreds of paired strains at a time.

<http://bit.ly/gcc2015irida>



# Protein identification of mass spectrometry data using Open MS.

Tools and workflow by Torsten Houwaart

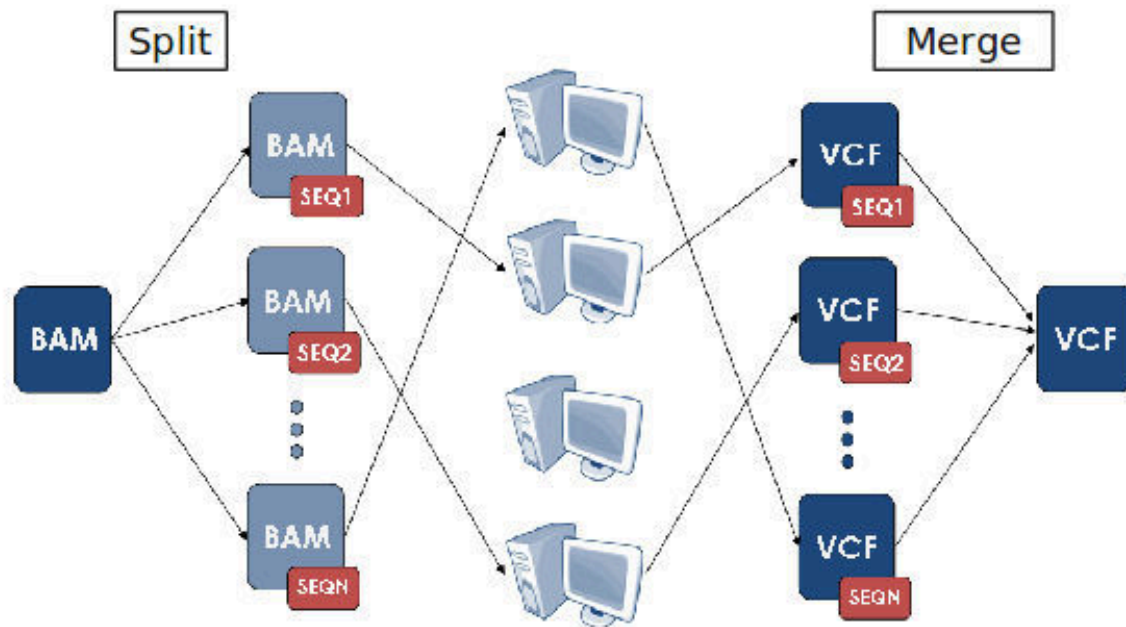
<http://bit.ly/gcc2015rna>



# Banner Year for Tool Development

- In 2015, we have 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.

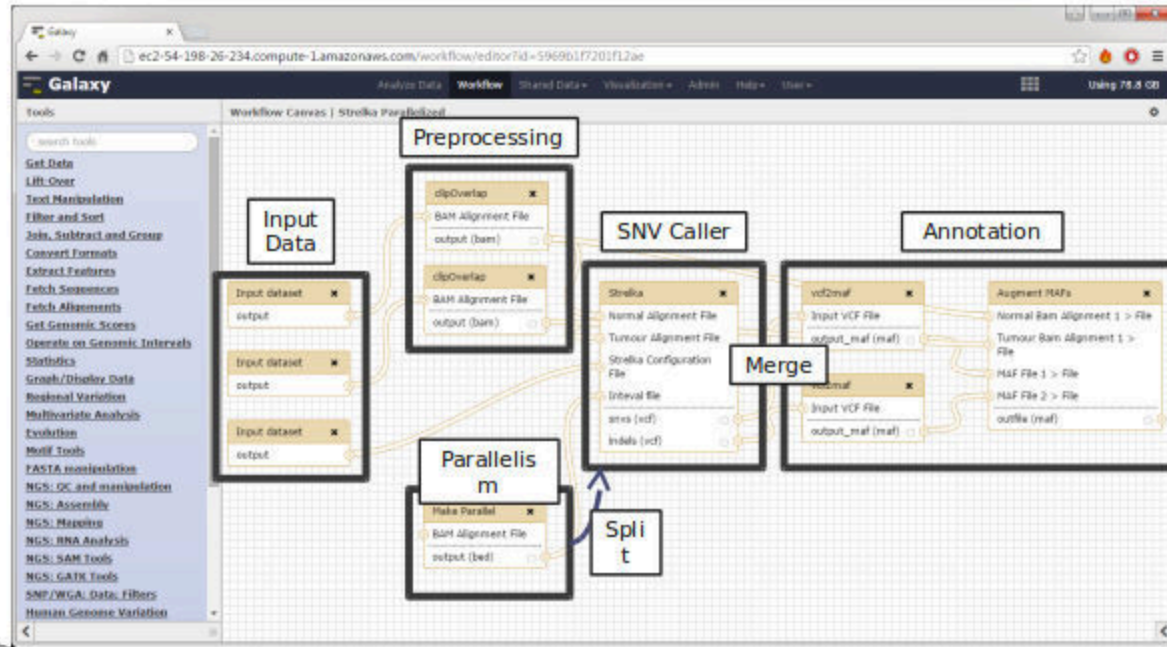
# Parallelization in Galaxy



<http://bit.ly/gcc2015cancerogenomics>

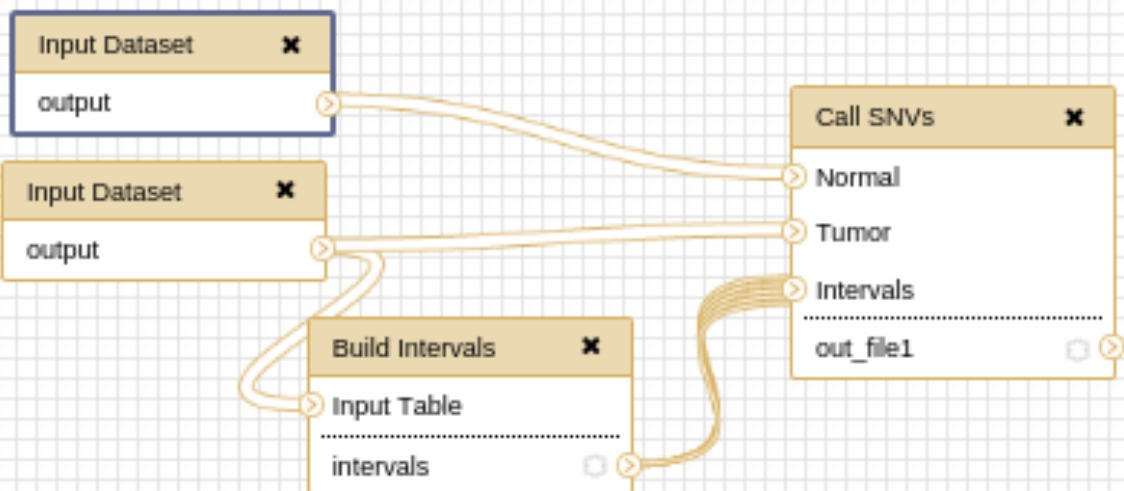
Marco Albuquerque, et. al.

# Somatic SNV Workflow



<http://bit.ly/gcc2015cancerogenomics>

Marco Albuquerque, et. al.

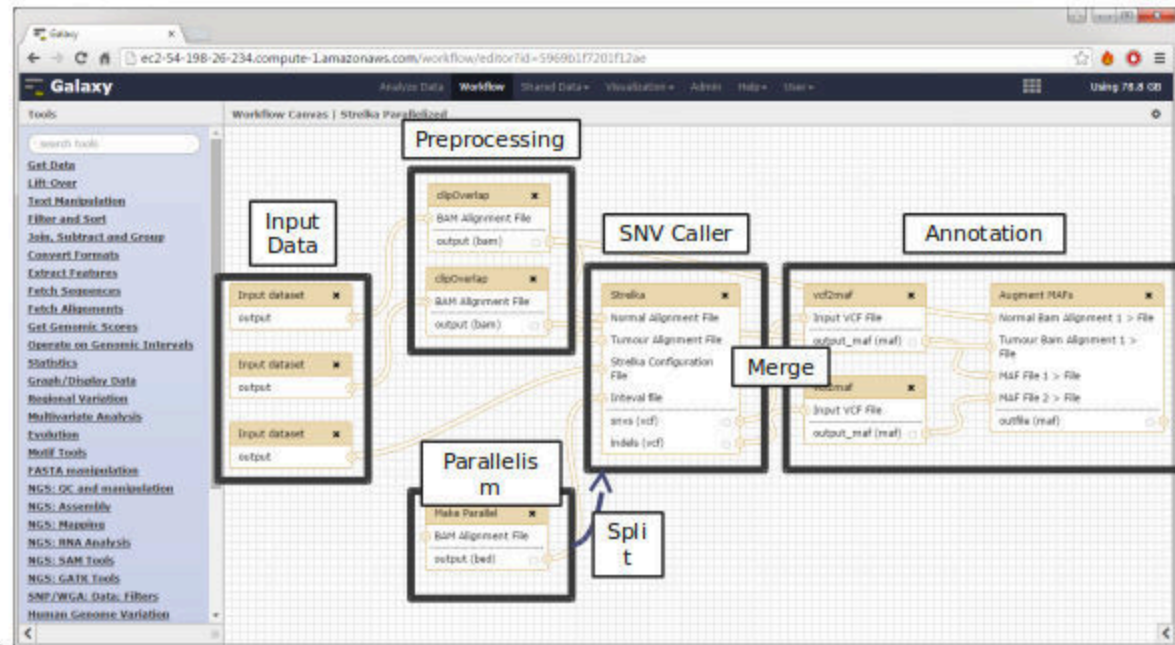


# Workflow - Rewrite

- **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/pcaWG\\_tools](https://github.com/ucscCancer/pcaWG_tools/tree/master/tools/pcaWG_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.

See talk by Dan Blankenberg at the 2015 GCC - Less Click, More Quick. <http://bit.ly/gcc2015lessclick>

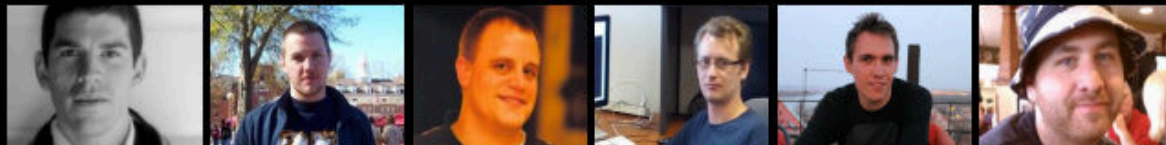


## Toward 10,000 samples (beyond collections)

- 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)

The **Galaxy Team**

# Thanks!



Enis Afghan    Dannon 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.

- biobambam 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**

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 CIGARs are not otherwise needed to be decoded.

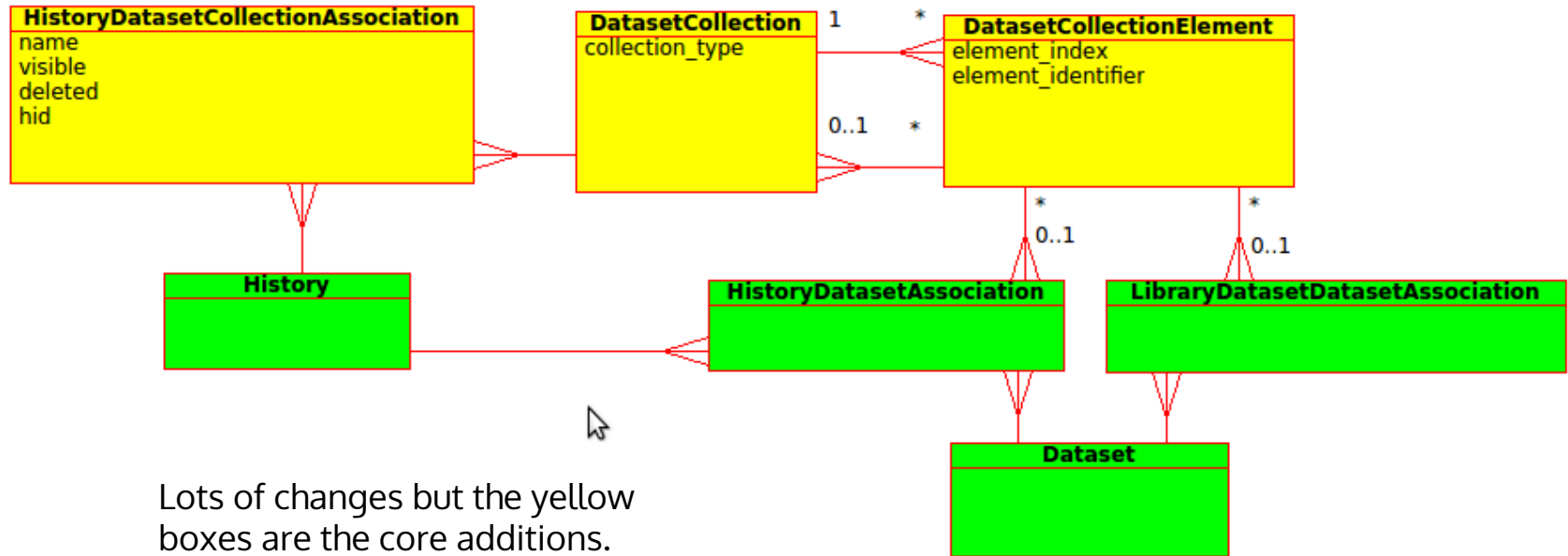
# Handful of Reduction Tools...

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.



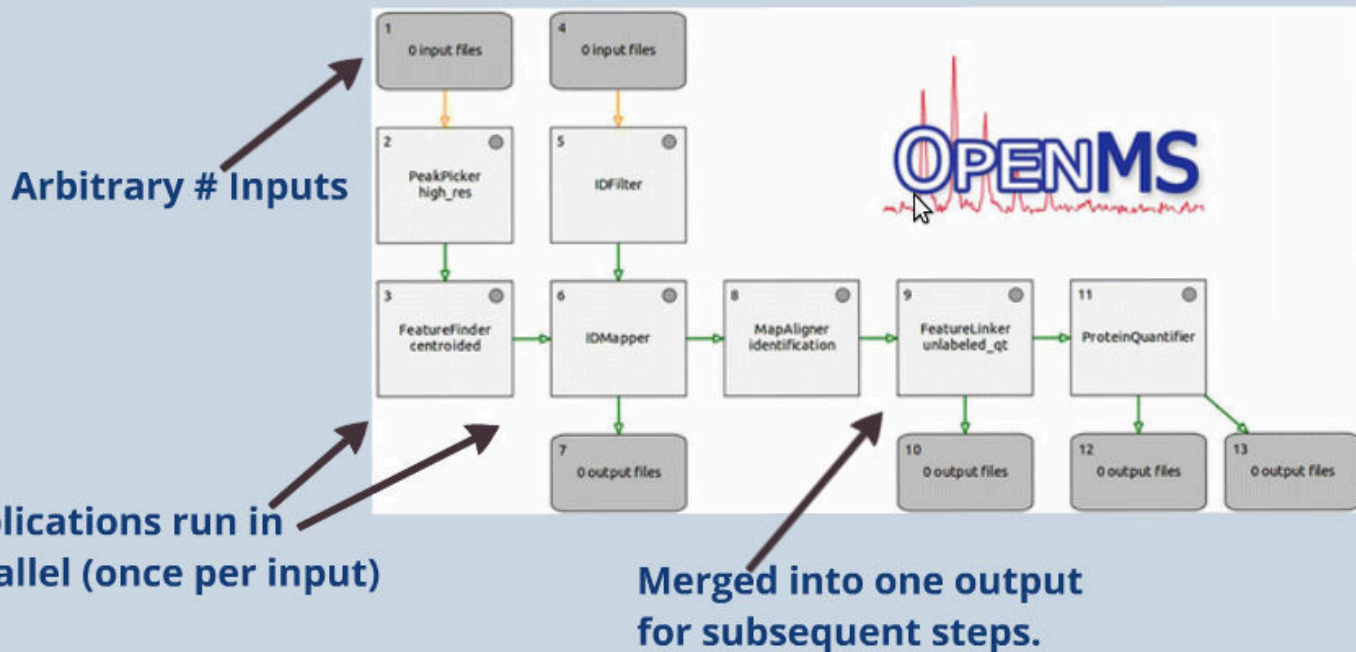
# Models



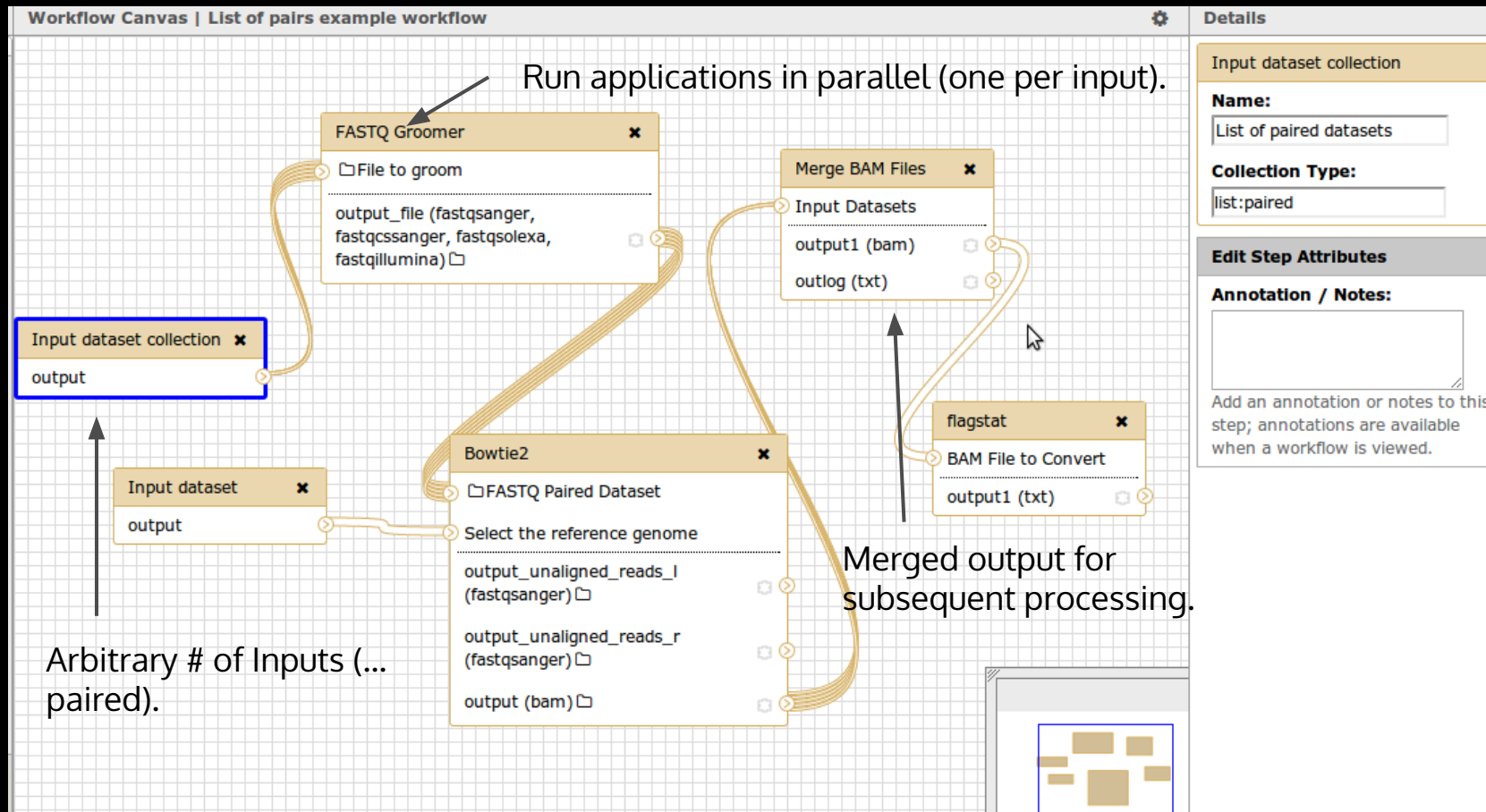
Lots of changes but the yellow boxes are the core additions.

# John @ GCC 2012, 2013 - Workflows... not good enough!

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



# More Powerful Workflows



# 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:

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

Common list data idiom:

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

-or-

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

Nested data:

```
#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>
```

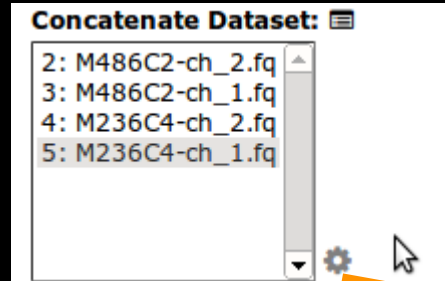
```
  </param>
```

```
...
```

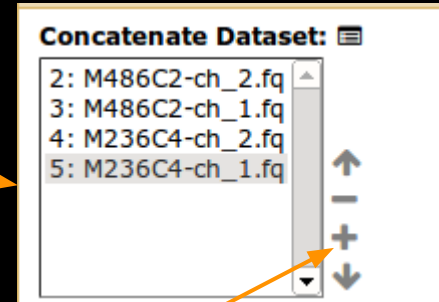
# Plan: Multiple-Data Improvements

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.



Mock Up



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>
- 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...

```
>>> 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/f8d40b687be4c699d608e930c59726793922fa0a>

Hide datasets  
Unhide datasets  
Delete datasets  
Undelete datasets  
Permanently delete datasets  
Build Dataset List (Experimental)  
**Build Dataset Pair (Experimental)**

GATCACAGGTCTATCAOCTATTAAOCCACTCAOGGGAGCTC  
GTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGC  
CTGCOCTCATOCTATTATTTATOGCAOCTAOGTTCAATATTA  
ATTAATTAATGCTGTGAGGACATAAATAAACAATTGAATC  
ATAACAATAAATTTCCACCAAAACCCCCOCTCCOCCOCTCTC

4: M486C2-ch 2.fq  
 3: M486C2-ch 1.fq  
 2: M236C4-ch 2.fq  
 1: M236C4-ch 1.fq

# Collection Mapping (1 / 3)

The screenshot shows the FASTQ Groomer (version 1.0.4) interface. On the left, the 'File to groom' field contains '4: M486C2-ch\_2.fq'. Below it, the 'Input FASTQ quality scores type' is set to 'Sanger & Illumina 1.8+'. Under 'Advanced Options', 'Hide Advanced Options' is selected. An 'Execute' button is visible. On the right, a 'History' panel lists several datasets, including '8: Paired mt Datasets', '7: M486C2', '6: M236C4', '5: sequence.fasta', '4: M486C2-ch\_2.fq', '3: M486C2-ch\_1.fq', '2: M236C4-ch\_2.fq', and '1: M236C4-ch\_1.fq'. Annotations with blue arrows point from the file path in the 'File to groom' field to the '8: Paired mt Datasets' entry in the history, and from the 'Execute' button to the '3: M486C2-ch\_1.fq' entry.

**FASTQ Groomer (version 1.0.4)**

**File to groom:** 4: M486C2-ch\_2.fq

**Input FASTQ quality scores type:** Sanger & Illumina 1.8+

**Advanced Options:** Hide Advanced Options

**Execute**

*-List of Paired Datasets  
-Paired Datasets  
-Individual FASTQ datasets.*

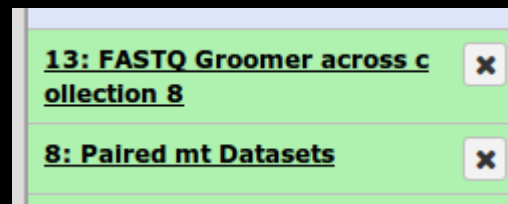
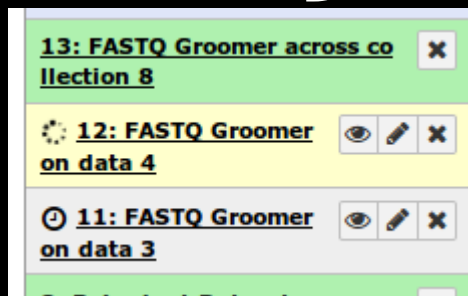
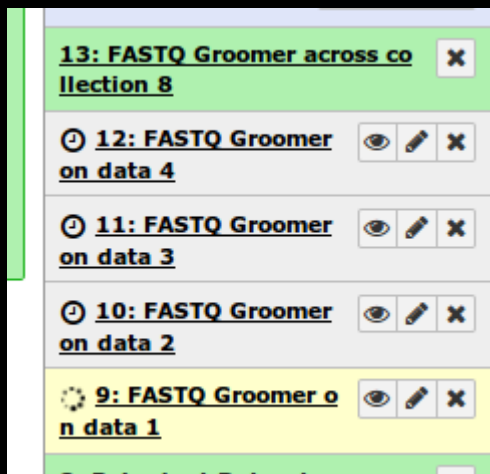
**History**

**Map/Reduce Test**  
519.8 MB

- 8: Paired mt Datasets
- 7: M486C2
- 6: M236C4
- 5: sequence.fasta
- 4: M486C2-ch\_2.fq
- 3: M486C2-ch\_1.fq
- 2: M236C4-ch\_2.fq
- 1: M236C4-ch\_1.fq

*Tool consumes a FASTQ file.*

# Collection Mapping (3 / 3)



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/CIIIdaxl2>

[Mockup @ mybalsamiq](#)

Create a list of paired datasets

(help text) Create a list of paired datasets by..

Forward	2 unpaired	Reverse	1 unpaired
<input type="text" value="Q_1"/>	9 Pairs	<input type="text" value="Q_2"/>	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
MRX3348_1.fastq	MRX3348	MRX3348_2.fastq	
exp_1000.bed		data_2.fasta	
yerinia_214_1.fastq			

Name of new list:

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 re-order the final list. (An option to move back to this step should also be there)

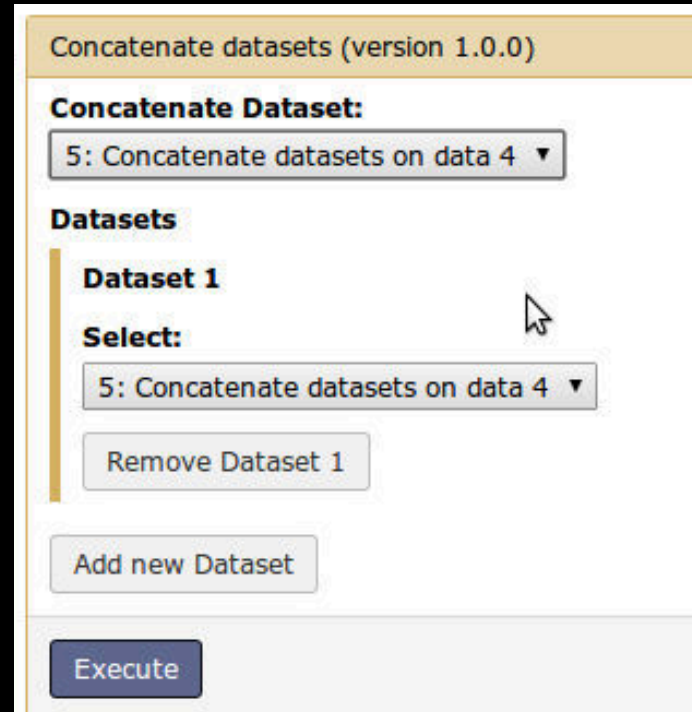
# Why not repeat replacements?

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.

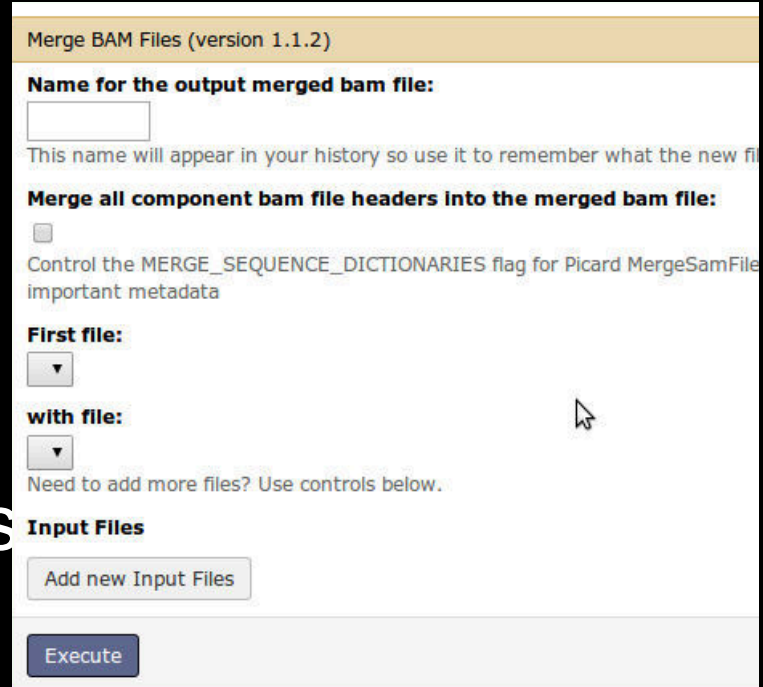


The screenshot shows a web-based interface for concatenating datasets. At the top, it says "Concatenate datasets (version 1.0.0)". Below that, there is a section titled "Concatenate Dataset:" with a dropdown menu showing "5: Concatenate datasets on data 4". Underneath, there is a "Datasets" section. The first dataset, "Dataset 1", has a "Select:" dropdown menu also showing "5: Concatenate datasets on data 4". Below the dropdown for Dataset 1 is a "Remove Dataset 1" button. At the bottom of the dataset list is an "Add new Dataset" button. Finally, at the very bottom of the interface is a blue "Execute" button.



# 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....



Merge BAM Files (version 1.1.2)

**Name for the output merged bam file:**

This name will appear in your history so use it to remember what the new file is

**Merge all component bam file headers into the merged bam file:**

Control the MERGE\_SEQUENCE\_DICTIONARIES flag for Picard MergeSamFile to preserve important metadata

**First file:**

**with file:**

Need to add more files? Use controls below.

**Input Files**

# Merging BedGraph

Found another reduction tool on main. Multiple inputs, multiple extra options. How could this reasonably allow collection replacement at the infrastructure level.

Merge BedGraph files (version 0.1.0)

**First BedGraph file:**

❌ History does not include a dataset of the required format / build

**Sample name:**

**Second BedGraph file:**

❌ History does not include a dataset of the required format / build

**Sample name:**

**Add'l BedGraph files**

**Add'l BedGraph files 1**

**BedGraph file:**

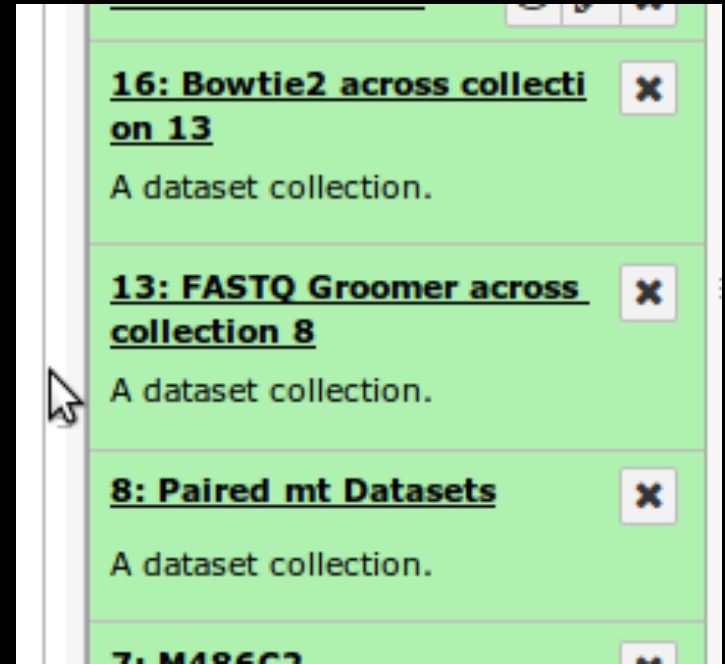
**Sample name:**

# Plans - More Options in History Panel

<https://trello.com/c/hnmVWKlB>

Currently can hide, delete, and see name.

Cannot rename, rerun, see type, see contents, see/add annotations, see/add tags, download, etc...



# Plans - UI for Uploading Collections

<https://trello.com/c/ZAXwWOZ2>

Incorporate collection builder when uploading files (or vice 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/lp5YmA10>

Improvements to multiple data parameters described earlier and/or ability to reduce across repeat statements.

# Plans - Filtering Collections

<https://trello.com/c/ryKJrsYc>

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.



# Plans - Rerun Tools / Resuming Workflows

<https://trello.com/c/lxVJy7fs>

**Docker... Docker... Docker...**



docker

[https://github.com/jmchilton/gcc2014\\_demo](https://github.com/jmchilton/gcc2014_demo)