Chapter 2

Analysis of Next-Generation Sequencing Data Using Galaxy

Daniel Blankenberg and Jennifer Hillman-Jackson

Abstract

The extraordinary throughput of next-generation sequencing (NGS) technology is outpacing our ability to analyze and interpret the data. This chapter will focus on practical informatics methods, strategies, and software tools for transforming NGS data into usable information through the use of a web-based platform, Galaxy. The Galaxy interface is explored through several different types of example analyses. Instructions for running one's own Galaxy server on local hardware or on cloud computing resources are provided. Installing new tools into a personal Galaxy instance is also demonstrated.

Key words NGS, Genomics, Informatics, RNA-seq, ChIP-seq, Workflows, Reproducibility, Open source, Web-based workbench, Big data analysis

1 Introduction

Recent advances in next-generation sequencing (NGS) technology have created a situation where raw data generation is no longer a rate-limiting factor in many genome-scale studies. However, the scale of the data presents not only difficulties for individual researchers attempting to analyze the data but also significant informatics issues for collaboration and reproducibility. Furthermore, simply generating data does not, in itself, lead to an increase in knowledge. Any technological advancement is limited in its ability to uncover new biological meaning if the ability of researchers to interact seamlessly with the data at any step is lacking. An important aspect of this interaction is providing researchers access to software tools.

Galaxy [1–3] is an open source, web-based platform for accessible, reproducible, and transparent computational biomedical research. Galaxy makes bioinformatics analyses accessible to users lacking programming knowledge by enabling them to interactively specify parameters for running tools and Workflows through a point-and-click interface. Every computational analysis is made reproducible by automatically capturing tool parameters and other

Benjamin L. Kidder (ed.), *Stem Cell Transcriptional Networks: Methods and Protocols*, Methods in Molecular Biology, vol. 1150, DOI 10.1007/978-1-4939-0512-6_2, © Springer Science+Business Media New York 2014

Table 1
Common galaxy terminology

Term	Description
Dataset	These are the inputs and outputs from each step in an analysis. Each time a tool is executed, a new Dataset is created that contains the results
Tool	An operation within Galaxy that acts upon Datasets as an analysis step. The underlying function may be developed by the Galaxy team or may be a third party program
History	A persistent container for an analysis. Each Dataset belongs to at least one History. As tools are run, Datasets appear chronologically within the active History. Each step of an analysis is recorded as the Datasets within a History
Workflow	A reusable analysis pipeline that allows any number of analysis steps to be performed automatically. Workflows that group tools into a single functional unit can be created de novo or by extracting directly from a History
Instance	A Galaxy instance is a single occurrence of a Galaxy server. There can be any number of Galaxy instances in existence at any particular time. Running a local Galaxy server would be an example of a Galaxy instance. Every Galaxy instance is independent and has its own set of Users, Datasets, and other objects
Main	The primary public Galaxy instance located at http://usegalaxy.org

information so that any user can repeat and understand the complete analysis. Transparency is maintained by allowing users simple access to share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis. A free public instance of Galaxy is available at http://usegalaxy.org. A local instance of Galaxy can be installed by following the directions at http://getgalaxy.org or run in the cloud by accessing http://usegalaxy.org/cloudlaunch. New tools can be easily installed into a Galaxy instance from Galaxy's application store, known as the ToolShed, available at http://usegalaxy.org/toolshed.

In addition to this chapter, users that are not familiar with Galaxy are directed to follow the tutorial available at http://usegalaxy.org/galaxy101 and to explore the resources at http://galaxyproject.org and https://vimeo.com/galaxyproject. *See* Table 1 for a list of common Galaxy terminology.

The live supplemental is located at https://usegalaxy.org/u/galaxyproject/p/ngs-analysis-2013.

2 Materials

2.1 Requirements for Using the Galaxy Interface

A computer with a modern web browser that supports JavaScript and HTML5 is required to use the interactive Galaxy interface. Most current Internet browsers are supported, such as Firefox,

Chrome, Safari, and Opera. JavaScript must be enabled and any plug-ins that the user has installed that block JavaScript should be disabled (e.g., NoScript).

2.2 Requirements for Analyzing Next-Generation Sequencing Data

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and a user account at the public Galaxy server (http://usegalaxy.org) or a local Galaxy installation or Galaxy cloud instance are needed.

NGS data in the FASTQ format is needed (*see* **Note 1**). You may supply your own Datasets or use the example data listed at the start of the methods.

2.3 RNA-Seq Analysis with Galaxy

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and a user account at the public Galaxy server (http://usegalaxy.org) or a local Galaxy installation or Galaxy cloud instance are needed.

RNA sequencing data in the FASTQ format is needed. You may supply your own Datasets or use the example data listed at the start of the methods.

2.4 ChIP-Seq Analysis with Galaxy

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and a user account at the public Galaxy server (http://usegalaxy.org) or a local Galaxy installation or Galaxy cloud instance are needed.

Chromatin immunoprecipitation (ChIP) sequencing data in the FASTQ format is needed. You may supply your own Datasets or use the example data listed at the start of the methods.

2.5 Creating and Running Galaxy Workflows

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and a user account at the public Galaxy server (http://usegalaxy.org) or a local Galaxy installation or Galaxy cloud instance are needed.

A Galaxy History used to create Workflows is also needed; these can be created in the previous subsections or can be other Histories created independently of this chapter.

2.6 Sharing and Publishing with Galaxy

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and a user account at the public Galaxy server (http://usegalaxy.org) or a local Galaxy installation or Galaxy cloud instance are needed.

Galaxy objects to share are also needed; these can be created in the previous subsections or can be other items created independently of this chapter.

2.7 Installing a Local Galaxy Instance

Installing Galaxy requires command-line access to a computer running a POSIX compliant operating system (OS) such as a Linux distribution or Mac OS X. Python2.6 or Python2.7 (http://www.python.org/getit) and Mercurial (http://mercurial.selenic.com) are

required to run and download Galaxy, respectively. Internet access is required during installation, but the computer does not need to have a fully public IP.

When a POSIX noncompliant OS, e.g., Windows, is on the computer, it is possible to use virtualization software, such as VirtualBox (https://www.virtualbox.org), recommended, or VMware Player (http://www.vmware.com), to install a compatible guest OS, such as Ubuntu (see, e.g., https://help.ubuntu.com/community/VirtualBox), in order to run Galaxy inside of a virtual machine. A virtual machine can also be used even if the host OS is POSIX compliant.

2.8 Running Galaxy in the Cloud

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and an Amazon Web Services (AWS) account with a valid payment method are required to use Galaxy CloudLaunch. Access to a Secure Shell (SSH) client is needed for advanced configuration that is not covered in this chapter. Currently, Amazon EC2 is supported. OpenStack-based cloud services have also been used, such as within the Australian NeCTAR cloud, but will not be covered here.

2.9 Installing New Tools via the Galaxy ToolShed

A computer with a modern web browser that supports JavaScript and HTML5 is required. Access to the Internet and Admin access to a Galaxy instance are needed to install tools (*see* Subheadings 2.7 and 2.8).

3 Methods

3.1 Using the Galaxy Interface

Galaxy is divided into several different operational interfaces. The most commonly used interface is the Analysis interface and is the first one encountered upon loading a Galaxy instance. The Analysis interface is accessed by clicking on "Analysis" within the Galaxy masthead or by clicking on "Galaxy" in the top left. Additional interfaces include the Workflow interface, the Data Library and Shared Data interfaces, the Visualization interface, and the Admin interface. Access to Galaxy can also occur via a RESTful application programming interface (API; *see* **Note 2**) but is beyond the scope of this chapter.

1. Start the web browser and load http://usegalaxy.org. This is the Main public Galaxy instance run by the Galaxy Project team. The Galaxy interface is divided into four main parts: the masthead, the tools menu, the tool interface, and the user History (*see* Fig. 1):
   (a) At the top of the page is the masthead. This allows the user to access help, user account settings, and shared data and to change the interface view. Also visible are the user's current data usage and quota status.

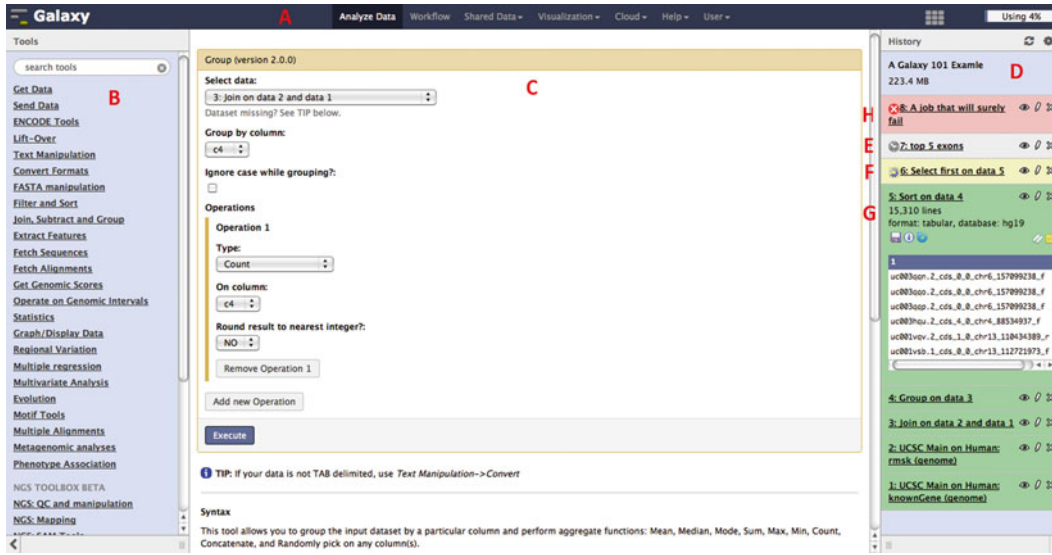

Fig. 1 The Galaxy Analysis interface. The Galaxy Analysis interface is constructed of four main parts: (*A*) the masthead at the *top*, (*B*) the tools menu on the *left*, (*C*) the tool interface in the *center*, and (*D*) the analysis History located on the *right*. The current status of a Dataset is indicated by its color and associated icon. Datasets that are queued (*E*) for execution, either due to limited compute resources or due to waiting for an input Dataset to become available, are *gray*. Datasets that are currently running (*F*) are *yellow*. A *green* Dataset indicates that the Dataset is ready to be used (*G*) and that the creating tool has finished executing successfully. A Dataset will be *red* in color when there has been an error with the analysis step (*H*)

(b) On the left-hand side is the Tools menu. Here, tools are organized into sections based upon function; clicking on a section will expand the available tools. Clicking on the underlined name of a tool will cause that particular tool interface to appear in the middle pane.

(c) The tool interface in the middle pane allows users to select input Datasets and to configure tool parameter settings. As a result of Galaxy's datatype system, only input Datasets that are valid for a particular tool input will be selectable. Clicking the "Execute" button here will cause an analysis job to run in the background and create one or more output Datasets in the user History.

(d) The user History is located on the right-hand side and contains the output Datasets from every step performed during a particular analysis. Every time a Dataset is uploaded or an analysis step is performed, one or more output Datasets are created within the History, with the newest steps appearing at the top. Clicking on the name of a Dataset within the History will expand to show a peek of the Dataset content and allow the user to access additional information and actions that can be performed on the Datasets, such as downloading, viewing the tools and

parameters that generated the Dataset, and allowing the job to be rerun.

2. In the masthead, click "User" and then click "Register." Enter the information requested and click "Submit" to register an account. A confirmation email will be sent for verification to activate the request. While user registration is optional, accessing advanced Galaxy features, such as multiple analysis Histories, Sharing, and Workflows, requires the user to be logged in. Additionally, at the Main public Galaxy server, registered users have a larger Dataset storage quota, allowing more data to be uploaded and more analyses to be performed.
3. In the History pane, click the gear icon to show History options, click "Create New," to create a new empty analysis History.
4. In the History pane, click "Unnamed history" and enter a new name for the History, such as "Learning Galaxy" (*see* **Note 3**).
5. In the Tools menu, click "Get Data" to expand the tool section.
6. Under the "Get Data" tool section, click "Upload File" to display the tool form interface. The Upload tool allows users to get external data into Galaxy by four methods: uploading from the user's computer, entering free text, copying and pasting one or more URLs for Datasets, or through FTP. Video examples can be found at http://vimeo.com/galaxyproject/upload.
7. In the URL/text box, enter "http://goo.gl/8Y3K8r" (without the quotes) and click "Execute." A new Dataset containing sequencing reads in the FASTQ format will appear in the History. In this case it is not necessary to change the file format parameter from "Auto-detect," as Galaxy will determine that the file is in the FASTQ format (*see* **Note 4**).
8. Datasets can also be loaded from a Data Library. Data Libraries are available by clicking on "Shared Data" in the masthead and selecting the "Data Library" option. This will present you with a list of the available Data Libraries on your current Galaxy instance.
9. Open a Data Library by clicking on the name of the Library. The Library will load into the center pane.
10. Datasets within Data Libraries are contained within folders. To expand folders, click on the folder icon next to the name of the folder.
11. Datasets can be loaded into a History by selecting the checkbox next to the name of the desired Dataset and then clicking the "Go" next to the "Import into current History" action at the bottom of the Library.
12. Multiple Datasets can be imported at one time and all of the folder's Datasets can be imported by checking the box next to the desired folder.

13. Datasets can also be loaded through Shared or Published Histories.
14. Shared History links may be entered as URLs into the browser and submitted. At the top right corner, the smaller green plus icon will read "Import" when moused over. Click on this icon to import the History and contained Datasets into the current History. Creating a new History as in **step 3** is advised.
15. Shared Histories can also be found by selecting "Histories Shared with Me" from the History menu. This menu is represented by a small gear icon at the top of the History pane, far right of the History name. From here, Shared Histories can be copied and worked with or unshared if no longer needed.
16. Published Histories are located under the masthead menu "Shared Data" and then "Published Histories." They are also often included in "Published Pages" as embedded data. From either location, the same icon will be present and the same process can be used as in **step 14** to import the History and/or Datasets.
17. Video walkthrough of Dataset attributes is at http://vimeo.com/galaxyproject/datasets1.

3.2 Analyzing Next-Generation Sequencing Data

Typically, NGS analysis in Galaxy begins with raw sequencing data in the FASTQ format. To facilitate the use of supplemental data required for certain tools as well as to support alternate analysis paths, Galaxy permits users to upload or import SAM/BAM, BED, GTF/GFF3, VCF, and other common bioinformatics file formats from local file systems or from integrated external sources including BioMart [4], UCSC Table Browser [5], GenomeSpace (http://genomespace.org), and others:

1. We will start with the History created in the previous section (Subheading 3.1), which now contains a set of sequencing reads as Datasets (*see* **Note 3**).
2. In the Tools menu, click "NGS: QC and manipulation" to expand the tool section.
3. Access the FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc) tool by clicking the name "FastQC: Read QC" (*see* **Note 5**).
4. Make sure that the newly uploaded FASTQ Dataset is selected under "Short read data from your current History."
5. Click "Execute" to run the FastQC tool and to create a new Dataset containing an HTML report containing various metrics about the FASTQ Dataset.
6. Once the tool has finished (Dataset has turned green; *see* Fig. 1), click on the Eye icon to display the FastQC report in the middle pane. Pay particular attention to the reported FASTQ variant type.

7. In the Tools menu, click "FASTQ Groomer" to access the FASTQ conversion and validation tool (*see* **Notes 6** and **7**).
8. Make sure that the uploaded FASTQ Dataset is selected under "File to groom" and click "Execute." The default settings are acceptable in this case and a fastqsanger Dataset will be created (*see* **Note 8**). If the Dataset input is different than the Sanger variant, select the variant type reported by the FastQC tool under "Input FASTQ quality scores type." If an output variant other than fastqsanger is desired, change "Advanced Options" to "Show Advanced Options" and select the appropriate settings.
9. The quality scores for sequenced bases generally decrease along the length of the read. Low-quality bases can have a significant impact upon alignment and other downstream steps. Based upon the FastQC report, it may be desirable to trim the ends of reads or to remove low-quality reads altogether by filtering.
10. This video on FASTQ data demonstrates how to groom or assign the proper Galaxy datatype for Illumina data (http://vimeo.com/galaxyproject/fastqprep).

3.3 RNA-Seq Analysis with Galaxy

RNA-seq is the practice of sequencing RNA. Generally, RNA is harvested from an organism and converted to DNA using reverse transcriptase. The resultant cDNA is then sequenced. While methods for sequencing each type of RNA have been developed, we will discuss the most commonly studied, mRNA. mRNA is usually purified by taking advantage of the poly-A tail. Two different types of information are usually investigated during an RNA-seq experiment: (1) alternative splicing and (2) differential expression of genes.

1. Load RNA-seq Datasets into a new History. Some sample FASTQ sequencing reads are available within a Data Library called "2013—MiMB—Stem Cell Transcriptional Networks" under the folder "RNA-seq."
2. Prepare the paired-end FASTQ sequencing reads using the techniques utilized in Subheading 3.2 on each Dataset.
3. Map the sequencing reads to a reference genome (hg19) using TopHat. Open the "Tophat2 Gapped-read mapper for RNA-seq data" tool [6], found under the "NGS: RNA-seq" section.
4. Set "Is this Library mate-paired?:" to "Paired-end."
5. Set "RNA-Seq FASTQ file, forward reads:" to the FASTQ Dataset that has been imported and prepared in **step 2** that corresponds to the forward reads (names end in "/1").
6. Set "RNA-Seq FASTQ file, reverse reads:" to the FASTQ Dataset that has been imported and prepared in **step 2** that corresponds to the reverse reads (names end in "/2").
7. Select the proper reference genome ("hg19").
8. Leave "TopHat settings to use" set to "Commonly Used."

9. Click "Execute" to start the job.
10. Repeat **steps 1–9** for each set of paired-end data.
11. Access the tool "Cufflinks transcript assembly and FPKM (RPKM) estimates for RNA-Seq data" [7].
12. For each mapped set of data, individually, set "SAM or BAM file of aligned RNA-Seq reads:" to the output of TopHat.
13. Click "Execute."
14. The output of Cufflinks provides a set of assembled transcripts along with estimates of isoform-level relative abundance known as FPKM.
15. Access the "Cuffcompare compare assembled transcripts to a reference annotation and track Cufflinks transcripts across multiple experiments" tool.
16. Set "GTF file produced by Cufflinks:" to the GTF output from one of the Cufflinks results.
17. Click "Add Additional GTF Input Files" for each additional Cufflinks output that was created previously.
18. Set the additional "GTF file produced by Cufflinks:" inputs to the remaining Cufflinks outputs.
19. Set "Use Reference Annotation:" to "Yes" and select the GTF reference annotation file that was imported from the Library.
20. Click "Execute."
21. Examine the output Dataset to locate transcripts that were assembled in each of the input Datasets.
22. Access the "Cuffdiff find significant changes in transcript expression, splicing, and promoter use" tool to assess differential expression.
23. Under "Transcripts:" select the combined transcripts Dataset created by Cuffcompare.
24. Set the "Replicates" input to the TopHat accepted hits output for each of the RNA-seq conditions. To include additional conditions, click "Add new Conditions."
25. Click "Execute." Several outputs will be created that assess any significant changes in transcript expression, splicing, and promoter use between the RNA samples.

3.4 ChIP-Seq Analysis with Galaxy

Chromatin immunoprecipitation followed by sequencing (ChIP-seq) is an essential technique for genome-wide profiling of protein binding, histone modification, and nucleosome positioning. Generally, within a ChIP-seq experiment, to determine protein-binding locations, bound proteins are cross-linked to their bound DNA locations and then sheered. The cross-linked protein–DNA molecules are passed over a column where the molecules bind to a treated substrate and the non-bound DNA is washed away.

The protein-bound DNA is then eluted from the column and the cross-linking is reversed. The precipitated DNA is purified and then sequenced to determine the DNA sequence around the previously bound loci. The sequencing reads are then mapped to a reference genome and enriched regions are determined (a process commonly called peak calling):

1. Load ChIP-seq Datasets into a new History. Some sample FASTQ sequencing reads are available within a Data Library called "2013—MiMB—Stem Cell Transcriptional Networks" under the folder "ChIP-seq."
2. Prepare the FASTQ sequencing reads labeled as "Enriched" using the techniques utilized in Subheading 3.2.
3. Map the sequencing reads to a reference genome (mm9) using BWA [8] or Bowtie [9] (not shown). Open the "Map with BWA for Illumina" tool, found under the "NGS: Mapping" section.
4. Select the proper reference genome ("mm9").
5. Set "FASTQ file:" to the FASTQ Dataset that has been imported and prepared in **step 2**.
6. Leave "BWA settings to use" set to "Commonly Used."
7. Click "Execute" to start the job.
8. Repeat **steps 2–6** on the Dataset labeled "Control."
9. Within the "NGS: Peak Calling" section, access the tool named "MACS Model-based Analysis of ChIP-Seq" [10].
10. Set "ChIP-Seq Tag File:" to the prepared and mapped Enriched Dataset.
11. Set "ChIP-Seq Control File:" to the prepared and mapped Control Dataset.
12. Set the reference genome to match the previously used reference genome (mm9).
13. Optionally check "Parse xls files into distinct interval files:," set "Save shifted raw tag count at every bp into a wiggle file:" to be "Save," and set "Resolution for saving wiggle files:" to be "1."
14. Click "Execute" to start the peak calling job.
15. The output Datasets consist of one or more result files (a–e) and an HTML summary report (f). These outputs take the form of the following:
    (a) Standard output—peaks: bed.
    (b) Optional output—peaks: interval.
    (c) Optional output—negative peaks: interval.
    (d) Optional output—treatment: wig.
    (e) Optional output—control: wig.
    (f) Standard output—html report.

16. Click the name of the Dataset labeled as "peaks: bed" to expand its contents.
17. Click on the link labeled "display at UCSC main" to view called peaks in a genomic context at the UCSC Genome Browser [11]. Are there any visual correlations between the called peaks and other annotation tracks, such as genes or conservation?

3.5 Creating and Running Galaxy Workflows

Workflows are groups of tools linked together to form an analysis pipeline that can be launched in batch, reused, edited, and started through the user interface or API and are an invaluable aid in ensuring exact replication of experimental conditions when used on a series of input data. Galaxy Workflows can be extracted from existing Histories or created de novo using the Workflow Editor. Once created, any Workflow can be viewed, copied, downloaded, edited again, shared, published, renamed, or deleted at any time.

Galaxy Workflows can be exported and imported across Galaxy instances and many from the Galaxy team and community are available both on the Main public Galaxy (http://usegalaxy.org) at "Shared Data" under "Published Workflows" and in the Main ToolShed (http://usegalaxy.org/toolshed) under "Search for Workflows" (leave search box empty and click on button "search repositories" to view all).

If it has been some time since you have run the original Workflow and the "Extract Workflow" function is invoked or you are uploading or importing a Workflow from another Galaxy server (click on masthead's "Workflows" menu to bring up the "Your Workflows" home page, and find the button in the upper right corner named "Upload or import workflow"), updates or notices about missing tools may be presented. Accept these changes when saving the Workflow on the current server, or, if using a local or cloud instance, adjust tool content on the server as needed (*see* Subheadings 3.7, 3.8, and 3.9). More Workflow help is available at http://wiki.galaxyproject.org/Learn/AdvancedWorkflow.

The first method describes how to extract a Workflow from an existing History, edit it, and then run it:

1. We will start with the History created by Subheading 3.3, which now contains a populated analysis History (*see* **Note 3**).
2. In the History pane, click on the top right gear icon to expand the "Histories List" menu.
3. Select the option "Extract Workflow" to bring up the associated form in the center pane of the Galaxy interface.
4. The form contains the following informative text:

   "The following list contains each tool that was run to create the Datasets in your current History. Please select those that you wish to include in the Workflow.

Tools which cannot be run interactively and thus cannot be incorporated into a Workflow will be shown in gray."

5. In the box under the text "Workflow name," modify the name to be something more meaningful than the default, if desired.
6. Review the tools on the left side of the form and the Datasets on the right side of the form for accuracy. Uncheck any that are not needed in the final analysis path.
7. Input Datasets do not have automated tools to retrieve the data and must be loaded, imported, or created by the user.
8. It is important to make a careful note of the input Dataset's expected content and format, which tools they are used in first, and even the ordering in the History (while scientifically arbitrary, the Workflow Editor will by default organize inputs during runtime in the same order as for display). This will help with labeling the Workflow's data inputs during editing.
9. Alternatively, a second browser window opened to this same page/view can be opened and retained for reference, while the first window moves forward with the next steps.
10. Click on the box "Create Workflow" when finished.
11. Click on "Workflows" in the top masthead menu choices.
12. On the Workflow home page, under "My Workflows," the Workflow just created will be listed first by default if sort by creation date is preserved.
13. Click on the end of the button named for your Workflow, near the down arrow on the right side. A menu will open.
14. Click on "Edit" from the "Workflow List" to open the Workflow Editor.
15. Orientation:
    (a) *Tool icons* are on the left pane.
    (b) *Canvas model* is in the middle pane.
    (c) *Tool options* are on the right pane.
    (d) *Navigation* is in the bottom right corner of the middle pane.
    (e) *Editor List* is in the small gear icon found right above the middle pane.
16. Best practices:
    (a) Make a second copy (backup) of anything important before you begin to edit it. Use the "Copy" function from the "Workflow List."
    (b) Save your Workflow periodically while editing, especially before or after significant edits. Using "Editor List" choice "Save."

(c) Save your Workflow before navigating away from the Workflow Editor or changes will be lost, same method as b. above.

17. Assign a name to each input of the extracted Workflow loaded into the editor:
    (a) Click on each box named "Input Dataset" in the canvas, one at a time.
    (b) Name is entered in the "Tool Options," top of the right pane, in the text box labeled "Name:."
    (c) Name each so that proper inputs from the History can be matched up with Workflow inputs during run setup and execution.

        For example, if a particular input is a reference annotation Dataset in the GTF format, label the input "GTF reference annotation."
18. Hide intermediate analysis steps:
    (a) Click on the "Flag output" icon (asterisk, sometimes referred to as a snowflake) within the final output step(s).
    (b) This small icon is located in the bottom right corner of tool icons in the Workflow canvas. Hovering over the icon will pop-up display the text:

        "Flag this as Workflow output. All non-flagged outputs will be hidden."
19. Save Workflow using "Editor List" choice "Save."
20. Run Workflow using "Editor List" choice "Run."
21. Set "Input Datasets" in the Workflow displayed in the center panel to be the proper Datasets from the right History panel (the original History). This Workflow can be run on any similar data ongoing after a test run.
22. Click on the box next to "Send the results to new History," and when the option expands, type in a meaningful History name into the new text box.
23. Click on "Run Workflow."
24. Once launched, a message will appear in the middle pane indicating the location of the new History along with the tools used, analysis steps performed, and resulting Datasets that will be produced.
25. Follow the link to view the new results and compare to the original results, first setting hidden Datasets to viewable using the "Histories List" menu option "Include Hidden Datasets" (*see* **Note 9**).
26. Results will be complete, identical, and reproducible unless one or more of the inputs changed (tool version, data content) or a nondeterministic tool is used.

The second method describes how to create a Workflow de novo using the Workflow Editor.

27. In this example, a simple Workflow duplicating the analysis steps performed in Subheading 3.2 will be created. The tools used in this example, in actual use, or by following the above instructions will guide the contents and construction of the Workflow.
28. We will start with a copy of the History created in the previous section (Subheading 3.1), which contains a set of sequencing reads as Datasets (create a copy if needed; *see* **Note 3**). These Datasets will be the "Input Datasets" of our Workflow when executed.
29. Click on the masthead menu "Workflow" to reach the Workflow homepage.
30. Click on the button "Create new workflow" located near the upper right corner.
31. Within the Workflow Editor, start by selecting the "Input Dataset" icons from the left tool icon pane. These are located at the bottom. For each input Dataset, click to to create the input module and drag into the desired position. Name/label as in the prior Workflow method as desired.
32. Next, add in the other tools to represent the analysis steps for the Subheading 3.2 methods. Drag each tool over and arrange.
33. When a tool is clicked on, the right pane will display the tool options. These are exactly the same options as presented when using tools directly. Modify parameters and add names or annotation to suit requirements.
34. Connect tools together by using the flexible "noodles" that extend out from the right side of tool icons (representing outputs) and insert into the left side of tool icons (representing inputs).
35. Choose which output Datasets will be hidden or not using the same method explained in **step 18** above (or a variation).
36. Duplicate **steps 19–26** above substituting the input Datasets and History for the one prepared in **step 28** above.
37. Reuse, expand, and explore more ways to use Workflows!

Bonus method: How to make your Workflows appear as tools in your tool menu on any server (including the Main public Galaxy instance).

38. Click on the masthead menu "Workflow" to reach the Workflow homepage.
39. At the bottom of the page under "Other options," click on the button "Configure your Workflow menu."
40. On the next form, in the farthest right column labeled "Show in menu," check the box for the Workflows that you want to appear in your tool menu and click "Save."

41. To return to the analysis interface, click on the masthead menu "Analyze Data."
42. Scroll down to the bottom of the left Tool Pane, under the section "Workflows," the Workflows that you checked will be listed individually.
43. "All Workflows" is present by default and is a quick way to bring up Workflows in the center pane, without leaving the active History.

3.6 Sharing and Publishing with Galaxy

Central to Galaxy's mission of enabling accessible, reproducible, and transparent research are the Sharing and Publishing functions common to all core objects, namely, Workflows, Visualizations, Pages, and Histories (including the Datasets contained within). Galaxy objects can be Shared or Published using a single click within the Galaxy interface. Sharing a Galaxy object can be done directly with another Galaxy user by utilizing their account's e-mail address on the same Galaxy server or by generating and communicating a Share link (email, publication, or similar) that will allow anyone that knows the link to access the shared item. When an item is Published within Galaxy, it is made publicly available under a central "Shared Data" hub for the object type where all users can view, search, tag, vote, import, and use. A Galaxy account is not required to view the content included in a Share link. Shared and Published objects may also be embedded into Galaxy Pages, which in turn can be Shared or Published. Galaxy Pages are particularly significant in that they provide a customized and organized means to communicate exact data sources, methods, results, and discussion related to analysis. The creation of a Galaxy Page utilizes an interactive word-processing style editor directly within the web browser. These are commonly used as a platform for publication supplemental materials and tutorials. Workflows, Datasets, and Histories can be directly imported from a Page and into the user's own workspace to be modified or reused. To learn more about the Share or Publish features in Galaxy, please see http://wiki.galaxyproject.org/Learn/Share and http://vimeo.com/galaxyproject/sharepublish.

1. We will start with the History created in the prior section (Subheading 3.4), which now contains a populated analysis History (*see* **Note 3**).
2. In the History pane, click on the top right gear icon to expand the "Histories List" menu.
3. Select the option "Share or Publish" to bring up the associated form in the center pane of the Galaxy interface.
4. There are three Share or Publish options grouped into two sections. Option A or B, plus C may be made active for any single History at a time. Selections can be modified at any time while on the form or by returning to the form.

5. Group 1: "Make History Accessible via Link and Publish It." A single option may be chosen:
   (a) Option A, button: "Make History Accessible via Link." This generates a web link that you can share with other people so that they can view and import the History.
   (b) Option B, button: "Make History Accessible and Publish." This makes the History accessible via link (see above) and publishes the History to Galaxy's *Published Histories* section, where it is publicly listed and searchable.
6. Group 2: "Share History with Individual Users":
   (a) Option C, button: "Share with a user." This directly shares a History with another user having an account on the same Galaxy server.
7. Sharing and Unsharing the History with a single user directly:
   (a) Click on the button from Option C from **step 6a** above, to initiate sharing the History with a single user directly.
   (b) A new form will open with an empty text box labeled "Galaxy user emails with which to share Histories."
   (c) Type into the box either a known user's account email address or optionally enter the email address "outreach@galaxyproject.org." Click "Submit."
   (d) The user's email will now appear as a button under the section "Share History with Individual Users." Click on the button to "Unshare" when no longer needed.
8. Sharing and Unsharing the History with one or more users via a link:
   (a) Click on the button from Option A from **step 5a** above, to generate the Share link.
   (b) A new button "Disable Access to History Link" will also appear. This offers control to disable History's link so that it is not accessible when finished sharing.
   (c) The share link may be customized by clicking on the pencil icon at the far right end of the link and modifying the displayed text, if desired.
   (d) Copy the link and paste it into an email, text message, document, or any other means desired to communicate the location of the shared History.
   (e) Any recipient of the Share link may enter it into a web browser (as described in Subheadings 2 and 2.1) in order to access the History. An account is not required.
   (f) Unshare the History by clicking on the button "Disable Access to History Link."
9. Sharing and Publishing the History with all users, publicly into "Shared Data, Published Histories." This action also generates

the Share link from **step 8**. How to unshare and unpublish are included.

(a) Click on the button from Option B from **step 5b** above, to generate the Share link and Publish the History to "Published Histories."

(b) A new button "Disable Access to History via Link and Unpublish" disables this History's link so that it is not accessible and removes the History from Galaxy's Published Histories section so that it is not publicly listed or searchable.

(c) Actions for the Share links are the same as described in the above sections, **step 8c–e**.

(d) Click on "Shared Data" in the upper masthead to open the pull-down menu, and select "Published Histories."

(e) Locate your Published History by searching by keyword or publication data.

(f) Click on the button "Disable Access to History via Link and Unpublish" at any time to unshare and unpublish your History after returning to the "Share or Publish" form for the History.

10. All four of Galaxy's core objects (Histories, Workflows, Visualizations, Pages) allow Sharing and Publishing through these exact same methods, using identical "Share or Publish" interfaces.
11. Published data appears under the masthead menu "Shared Data."
    (a) All four of these same objects in **step 10** above, when in a "Published" state, will be displayed under the masthead menu "Shared Data," in the corresponding object section.
    (b) For example, "Pages" that have been published are found under the menu "Shared Data" option "Published Pages."
12. The introduction (above) for this subsection includes links to wiki and video demonstrations of Share and Publish operations in detail.

3.7 Installing a Local Galaxy Instance

You only need to download and install a local Galaxy if you plan to (1) develop it further, (2) add new tools, (3) plug in new data sources, or (4) run a local production server for your site because you have Sensitive data (e.g., clinical) or Large Datasets or processing requirements that are too big to be processed on a public server. To obtain the latest directions on running your own Galaxy instance, go to http://getgalaxy.org within your web browser. This page will also have additional information and trouble-shooting tips:

1. Open a command-line prompt (i.e., a terminal or shell).
2. Confirm that you have compatible version of Python installed (Subheadings 2.6 or 2.7) by typing "python --version" followed by the return key.

3. Confirm that you have Mercurial installed by typing "hg–version" followed by the return key. *see* **Note 10**, if you do not have Mercurial installed.
4. Download the Galaxy source code by typing "hg clone https://bitbucket.org/galaxy/galaxy-dist" followed by the return key.
5. Change your current working directory into the freshly created Galaxy root by typing "cd galaxy-dist," followed by the return key.
6. Update your Galaxy source code to the stable release branch by typing "hg update stable," followed by the return key. In the future, you can update your local Galaxy instance to the latest Galaxy version by entering "hg pull -u."
7. To start your Galaxy server, type "sh run.sh" followed by the return key. The first time that you start your Galaxy instance, it will download additional required dependencies (known as Python eggs) and automatically create local copies of several configuration files.
8. Once your Galaxy instance has started, load http://localhost:8080 within your web browser.
9. To stop the Galaxy server, use "Ctrl-C" from within the shell.
10. To add yourself as an administrator to your Galaxy instance, open the file "universe_wsgi.ini" and add your email address to the admin_users variable, for example, "admin_users = you@example.com."
11. Restart the Galaxy server to make the new Admin user active.
12. Stay current with release updates by following *Distribution News Briefs* (http://wiki.galaxyproject.org/DevNewsBriefs) and consider subscribing to the Galaxy-Dev mailing list for Galaxy community and team support (http://wiki.galaxyproject.org/MailingLists).

3.8 Running Galaxy in the Cloud

A third option for accessing Galaxy is to utilize cloud computing resources. Currently, to use Galaxy on commercial cloud resources, you will need to have an Amazon Web Services (AWS) account. A complete set of up-to-date instructions are also available at http://wiki.galaxyproject.org/CloudMan:

1. Register an account with AWS by going to http://aws.amazon.com.
2. Create a new IAM user via the AWS console.
3. Make note of the created Access Key ID and Secret Access Key.
4. Load http://usegalaxy.org/cloudlaunch in your web browser.
5. Enter your Access Key ID and your Secret Access Key into the boxes on this page.
6. Provide a name for your Galaxy cluster; this can be any value. The name should be unique for a particular AWS user, as it is

possible to have multiple CloudMan instances running at a single time and the name acts as an identifying key while running and resuming a cluster.

7. Set a password for your new Galaxy cluster. This password is the CloudMan console password and is only used to restrict access to the CloudMan administration interface.
8. Click Submit to launch your Galaxy CloudMan-managed cloud instance.
9. After a few minutes, the private Galaxy cloud instance will start and can be accessed from the provided link, with the form of "While it may take a few moments to boot, you will be able to access the cloud control panel at ec2-75-101-202-210.compute-1.amazonaws.com/cloud."
10. At the authentication prompt, enter the password that was specified in **step 7**. The username field can be left blank.
11. Since we are starting a new Galaxy cluster, you can accept the default settings and click "Choose platform type." It is also possible to specify a Dataset storage size different from the default (10 GB) depending upon the needs of the analysis; this size can later be changed via the CloudMan interface, but the Galaxy instance will be inaccessible while it is resized.
12. Within the CloudMan Console, the cloud cluster can be terminated, additional worker nodes can be added, and a link to Galaxy can be accessed. Additionally, the CloudMan Admin panel can be accessed by clicking on "Admin" at the top right of the masthead.
13. Add a new Galaxy Admin user by entering your email address into the "Set Galaxy admin users" textbox in the CloudMan Admin panel.
14. Click "Access Galaxy" to load the Galaxy Analysis interface.
15. You can now register a new Galaxy user, including the one that corresponds to the Admin user that was just created.
16. When you are finished, be sure to terminate the Galaxy cluster from the CloudMan interface, or else you will continue to be charged for usage. When you are completely finished with the cluster, be sure to check the box next to "Also delete this cluster"; this will delete the EBS volumes and S3 buckets that have been created for use in the cluster and allowed the cluster to be persisted without requiring compute nodes to be constantly running.
17. Consider subscribing to the Galaxy-Dev mailing list for Galaxy community and team support (http://wiki.galaxyproject.org/MailingLists).

3.9 Installing New Tools via the Galaxy ToolShed

The Galaxy ToolShed (http://usegalaxy.org/toolshed) enables sharing of Galaxy tools across the Galaxy community.

It is a software distribution hub for biomedical software that supports versioning, dependency, and datatype management, as well as Workflow and data integration. The ToolShed supports a wide array of tool types allowing nearly any software utility (written in any programming language), ranging from simple scripts written in interpreted languages such as Python to complex software packages distributed as source code that require compilation and installation as well as external dependencies, to be automatically installed into a Galaxy instance with a few clicks.

Here, we will install an example tool, the FreeBayes variant detector. Video examples of ToolShed tool installations into a CloudMan Galaxy can be viewed at http://vimeo.com/channels/galaxytoolshed:

1. Log in to a Galaxy instance where you are an administrator (*see* Subheadings 3.7 and 3.8).
2. Access the Administrator interface by clicking the "Admin" link in the masthead.
3. In the left-hand pane, click "Search and browse tool sheds." The ToolShed selection screen will appear in the main pane.
4. Click on the button labeled "Galaxy main tool shed." The primary public Galaxy ToolShed will load in the pane.
5. In the search box at the top left of the page, search for "freebayes." As the ToolShed is a community resource, several different versions of the FreeBayes tool may be available.
6. Choose the FreeBayes tool repository created by the Galaxy development team by clicking on the button "freebayes" that has the owner listed as "devteam." If a pop-up appears, click "Preview and Install."
7. In the top left-hand corner, click "Install to Galaxy."
8. Take note of the warning at the top of the Page, indicating that the ToolShed is a public resource where community members can add code and as such not all of this external code has been verified by the Galaxy development team nor the community-based team tasked with approving community contributions, known as the Intergalactic Utilities Commission (IUC). We are installing a tool added by the official Galaxy development team ("devteam"), so we can be rather sure of it being non-malicious.
9. Be sure that "Handle tool dependencies" is checked. This will download and install a local copy of the versioned FreeBayes binaries for Galaxy to run.

10. Select a tool section to install the new tool. In this case, we will create a new section, "NGS: Variant Detection," by entering that text into the box labeled "Add new tool panel section."
11. Click "Install." Galaxy will download and install the new tool. The installation progress can be monitored in real time as it advances through the various stages (e.g., new, downloading, installing dependencies, installed).
12. Switch back to the Galaxy Analysis interface and confirm that the tool has been installed into the new tool section.

4 Notes

1. The FASTQ format has become the de facto standard format for the representation of sequencing reads [12]. There are several different FASTQ variants in use, with the Sanger variant being the preferred form. In the fastqsanger format, the quality scores are Phred-scaled and encoded using ASCII characters, with one character used per base. The score value is equal to the ordinal value of the ASCII character subtracted by 33.
2. The Galaxy API provides a programmatic interface to communicate with a Galaxy instance when directly interacting with the web-based GUI is not practical or desired. For example, external programs that want to access features of Galaxy can make use of the RESTful API.
3. It is important to use meaningful names for Histories. History names serve as a straightforward method of keeping multiple Histories organized. It is a good idea to use a separate History for each analysis or logical portion of each analysis. Access all account Histories using the "History List" menu option "Saved Histories."
4. Galaxy's datatypes utilize a hierarchical system where more restricted datatypes are children of less defined types. This allows tools to function on specific data formats or on types of data. For example, a tool which utilizes the generic "fastq" datatype will accept "fastqsanger" and "fastqillumina" as input. However, a tool which accepts "fastq" for input will not accept generic "text" Datasets as input, despite "fastq" being a child of "text."
5. Tools can sometimes be difficult to locate. You can use the Tool Search to search for tools based upon name and key words. Enter a word or phrase into the "search tools" text box at the top of the Tool pane on the left-hand side.
6. When uploading a file, it is possible to manually declare a datatype instead of utilizing the auto-detect functionality. This can be helpful to select a specific sub-datatype during upload;

for example, all variants of the FASTQ format are detected as the base "fastq" class; however, selecting the specific FASTQ variant (e.g., "fastqsanger") during upload will allow a user to bypass the grooming step if they are sure of the correctness and encoding type of their uploaded FASTQ file.

7. It is possible to change the declared datatype of a Dataset after it exists in a History by clicking the pencil icon and selecting a new datatype from the drop-down list. This will not alter the actual Dataset content, just the way that Galaxy will interpret the Dataset. To convert Dataset content, select the "Convert Format" tab or search for a stand-alone tool in the Tool pane that will convert the Dataset to different formats.
8. There are several different variations to the FASTQ format, depending upon the value of and the encoding method used for base quality scores. The datatype "fastqsanger" is the preferred variant datatype to use for most tools. The FASTQ Grooming tool can be used to convert between the different variants. Color-space reads will use the "fastqcssanger" datatype.
9. Workflows can be configured to hide intermediate analysis Datasets; however, all Datasets are still present in the result History. The "History list" options that make hidden datasets accessible are (a) "Include Hidden Datasets" to temporarily unhide and rehide or individually permanently unhide and (b) "Unhide Hidden Datasets" to permanently unhide all hidden Datasets at once. Hidden Datasets are created by Workflows only and a Dataset that has permanently been marked as unhidden cannot return to hidden status.
10. Mercurial is a source-code revision control system. It can often be installed by the packaging system used by your operating system; for example, in Ubuntu, typing "sudo apt-get install mercurial" into a shell will perform the installation. If it is not possible to use Mercurial to obtain the Galaxy source code, a downloadable archive is also available from https://bitbucket.org/galaxy/galaxy-dist. Using Mercurial is the preferred method as it will greatly simplify updating Galaxy to the newest versions.

Acknowledgments

Efforts of the Galaxy team (E. Afgan, D. Baker, D.B., D. Bouvier, M. Cech, D. Clements, N. Coraor, C. Eberhard, D. Francheteau, J. Goecks, S. Guerler, J.J., G. Von Kuster, R. Lazarus, Anton Nekrutenko, and James Taylor) were instrumental in making this work possible. We extend a special thank you to the Galaxy community for their continuing contributions, both inspirational and technical.

References

1. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15(10):1451–1455. doi:10.1101/gr.4086505, gr.4086505 [pii]
2. Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J (2010) Galaxy: a web-based genome analysis tool for experimentalists. Curr Protoc Mol Biol. Chapter 19: Unit 19 10 11-21. doi: 10.1002/0471142727.mb1910s89
3. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11(8):R86. doi:10.1186/gb-2010-11-8-r86, gb-2010-11-8-r86 [pii]
4. Kasprzyk A (2011) BioMart: driving a paradigm change in biological data management. Database (Oxford) 2011:bar049. doi:10.1093/database/bar049 bar049 [pii]
5. Karolchik D, Hinrichs AS, Furey TS, Roskin KM, Sugnet CW, Haussler D, Kent WJ (2004) The UCSC Table Browser data retrieval tool. Nucleic Acids Res 32(Database issue):D493–D496. doi:10.1093/nar/gkh103 32/suppl_1/D493 [pii]
6. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol 14(4):R36. doi:10.1186/gb-2013-14-4-r36, gb-2013-14-4-r36 [pii]
7. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28(5):511–515. doi:10.1038/nbt.1621 nbt.1621 [pii]
8. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25(14):1754–1760. doi:10.1093/bioinformatics/btp324 btp324 [pii]
9. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10(3):R25. doi:10.1186/gb-2009-10-3-r25 gb-2009-10-3-r25 [pii]
10. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS (2008) Model-based analysis of ChIP-Seq (MACS). Genome Biol 9(9):R137. doi:10.1186/gb-2008-9-9-r137 gb-2008-9-9-r137 [pii]
11. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D (2002) The human genome browser at UCSC. Genome Res 12(6):996–1006. doi:10.1101/gr.229102
12. Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38(6):1767–1771. doi:10.1093/nar/gkp1137 gkp1137 [pii]