Using Cloud Computing Infrastructure with CloudBioLinux, CloudMan, and Galaxy

Enis Afgan,[1,5] Brad Chapman,[2] Margita Jadan,[3] Vedran Franke,[4] and James Taylor[5]

[1]Center for Informatics and Computing, Ruđer Bošković Institute (RBI), Zagreb, Croatia
[2]Harvard School of Public Health, Boston, Massachusetts
[3]Division of Materials Chemistry, Laboratory for Ichthyopathology–Biological Materials, Ruđer Bošković Institute (RBI), Zagreb, Croatia
[4]Department of Biology, University of Zagreb, Zagreb, Croatia
[5]Department of Biology and Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia

ABSTRACT

Cloud computing has revolutionized availability and access to computing and storage resources, making it possible to provision a large computational infrastructure with only a few clicks in a Web browser. However, those resources are typically provided in the form of low-level infrastructure components that need to be procured and configured before use. In this unit, we demonstrate how to utilize cloud computing resources to perform open-ended bioinformatic analyses, with fully automated management of the underlying cloud infrastructure. By combining three projects, CloudBioLinux, CloudMan, and Galaxy, into a cohesive unit, we have enabled researchers to gain access to more than 100 preconfigured bioinformatics tools and gigabytes of reference genomes on top of the flexible cloud computing infrastructure. The protocol demonstrates how to set up the available infrastructure and how to use the tools via a graphical desktop interface, a parallel command-line interface, and the Web-based Galaxy interface. *Curr. Protoc. Bioinform.* 38:11.9.1-11.9.20. © 2012 by John Wiley & Sons, Inc.

Keywords: accessible cloud computing • enabling bioinformatics analyses • turnkey computing system

INTRODUCTION

Biomedical research, driven by continued increases in data-generation capability, has become a data-intensive science. To process and analyze these data requires informatics support. Moreover, different types of data and analysis are likely to have different compute requirements. For example, in the context of next-generation sequencing (NGS), a de novo assembly analysis step might require vastly more memory (RAM) in a single machine compared to a BLAST search step, which is much more limited by the clock speed of the CPU. Data analysis is thus enabled through a combination of resource availability and capacity with resource configuration (i.e., installation of tools, availability of reference data, ease of use). For individual researchers and small labs, who now have access to large volumes of data, acquiring, configuring, and maintaining the required compute infrastructure is a hindrance and even a barrier to advancing research.

Fortunately, in recent years, cloud computing has emerged as a viable option to quickly and easily acquire computational resources required for an analysis. Cloud computing offers network access to computational resources where CPUs, memory, and disks are accessible in form of a virtual machine (i.e., complete operating system) that a user has

Current Protocols in Bioinformatics 11.9.1-11.9.20, June 2012
Published online June 2012 in Wiley Online Library (wileyonlinelibrary.com).
DOI: 10.1002/0471250953.bi1109s38

individual and complete control over. As a result, cloud computing has the potential to allow simple access to a variety of different types of machines, including large-memory machines, fast-CPU machines, or abundant disk space, without needing to build and later maintain the given infrastructure. There are different models for exposing cloud resources (Afgan et al., 2011a), and the one that provides the most flexibility, from the bioinformatics research standpoint, is the Infrastructure-as-a-Service (IaaS) model. Due to the low-level resources that such a model exposes, it is possible to arrange the available components in such a way as to achieve a wide variety of configurations. This feature largely removes limitations imposed by physical resource availability, and helps enable open-ended analyses.

However, such resources still need to be procured and configured to provide a complete environment. Over the years, several dedicated tools and consolidation frameworks have emerged that provide unified interfaces and access to various bioinformatics tools, most notably Galaxy (Goecks et al., 2010; Afgan et al., 2011b) and BioLinux (Field et al., 2006). In combination, these two frameworks make more than 100 current bioinformatics tools easily available and accessible to researchers while requiring no installation or configuration. Although both of the frameworks have made extensive efforts to make local installation steps very simple, the frameworks still need a computational infrastructure to run the analyses.

CloudMan (Afgan et al., 2010) and CloudBioLinux (*http://cloudbiolinux.org/*) are projects that combine these two frameworks with cloud computing resources into a solution that alleviates the issues of infrastructure provisioning and configuration. CloudBioLinux and CloudMan with Galaxy deliver completely configured instances of the respective projects with a broad range of tools, reference data, and interfaces, requiring nothing more than a Web browser to use. As a result, a researcher can simply acquire the necessary resources for a specific analysis, immediately perform the analysis, and then release the resources, without any additional management or maintenance burden. Simultaneously, if more advanced usage scenarios are desired, including command-line access, addition of custom tools or data, or sharing of complete or partial analyses, the combination of these two projects makes them possible.

CloudBioLinux instances provide an excellent environment in which to easily test and become familiar with BioLinux and cloud computing in general. In addition, CloudBioLinux makes it possible to easily test a tool or a complete analysis on various types of compute infrastructures available from a cloud provider without the need to install or configure any tools. CloudBioLinux instances alone, however, are limited to a single instance and, without additional effort, any data produced on the instance do not persist beyond the life of the instance. Coupling CloudBioLinux with CloudMan alleviates both of these concerns by automatically setting up a compute cluster and, optionally, associating persistent data storage with each cluster. Together, these two projects create a *platform* providing all the infrastructure and application components required to perform an analysis and make it accessible to a researcher.

This unit describes the functionality of CloudBioLinux and CloudMan with and without Galaxy. It is directed primarily at experimentalists and makes use of analysis tools available on the default platform of the projects. It also showcases the ability to customize the platform by adding additional tools. The unit focuses on accessibility of the solutions, and opens doors to further research using the techniques described here. The unit is divided into the following protocols: Basic Protocol 1 is an introduction to cloud computing, CloudBioLinux, and CloudMan on the Amazon Web Services (AWS) cloud; it describes how to set up a complete analysis environment on the cloud. Support Protocol 1 demonstrates how to connect to the remote instance via a graphical desktop

interface, while Support Protocol 2 describes how to connect to the created instance via the command-line method. Basic Protocol 2 shows how to perform visual analysis with a CloudBioLinux instance. Basic Protocol 3 describes how to make use of the cluster environment established by CloudMan to perform a parallel analysis. Lastly, Basic Protocol 4 shows how to create and use a private and completely configured instance of Galaxy analysis environment using CloudMan.

AN INTRODUCTION TO CLOUD COMPUTING AND ACCESS TO CLOUD RESOURCES VIA CloudBioLinux AND CloudMan

BASIC PROTOCOL 1

CloudBioLinux and CloudMan automate the process of infrastructure composition and configuration to provide a complete environment ready to perform a desired analysis; the environment is composed automatically and in a matter of minutes. This protocol presents steps required to start a personal application execution environment on AWS that is configured to perform a wide range of analyses. Overall, the process involves starting and accessing a private instance of a virtual machine image on AWS, which has been customized for use with CloudBioLinux. The process is initiated through a custom-developed Web portal (BioCloudCentral.org) that greatly simplifies the process of instance acquisition. This is because the portal automates many of the otherwise required steps (e.g., setting up firewall access rules, authentication key generation, formatting user data). Once the instance (i.e., virtual machine) starts, one can access the instance via a graphical or command-line interface and perform the desired analysis (see Support Protocols 1 and 2, respectively). This protocol provides an overview of how to start and manage CloudBioLinux and CloudMan instances; it provides details about the entire lifecycle of an instance and can be used as a reference point throughout this document. These (and other) processes are captured in webcasts available at *http://usecloudman.org/screencasts.*

Necessary Resources

Computer with Internet access and any up-to-date Web browser (Firefox, Safari, Opera, Chrome, Internet Explorer)

An AWS account with the Elastic Compute Cloud (EC2) and Simple Storage Service (S3) services enabled. To sign up for an account, visit *http://aws.amazon.com* and click the Sign Up Now link. Basic background information on the EC2 and S3 services can also be found at this page.

1. Visit the BioCloudCentral portal (*http://biocloudcentral.org*) with your browser and fill in the provided form (Fig. 11.9.1) as described in the following steps.
2. Provide a name for the cluster you are about to create. This can be any name you choose and will be used to distinguish this particular cluster from any others you may create. Once the name is chosen and the cluster is created, you will have the option to terminate the given cluster (i.e., release all of the cloud resources) and restart it at a later time by simply providing the same startup information. Terminated clusters preserve all their uploaded data and customizations.
3. Choose a password that will be used to connect the created cluster. The same password will be used to connect to the instance via the graphical desktop connection, CloudMan console, or SSH login. The password is required to control access to the created cluster.
4. Provide the AWS credentials for your account. These values can be obtained from the AWS portal on the account Security Credentials page.

   The credentials are needed because a deployment is a composition of multiple infrastructural resources, and their acquisition requires user credentials. The provided credentials

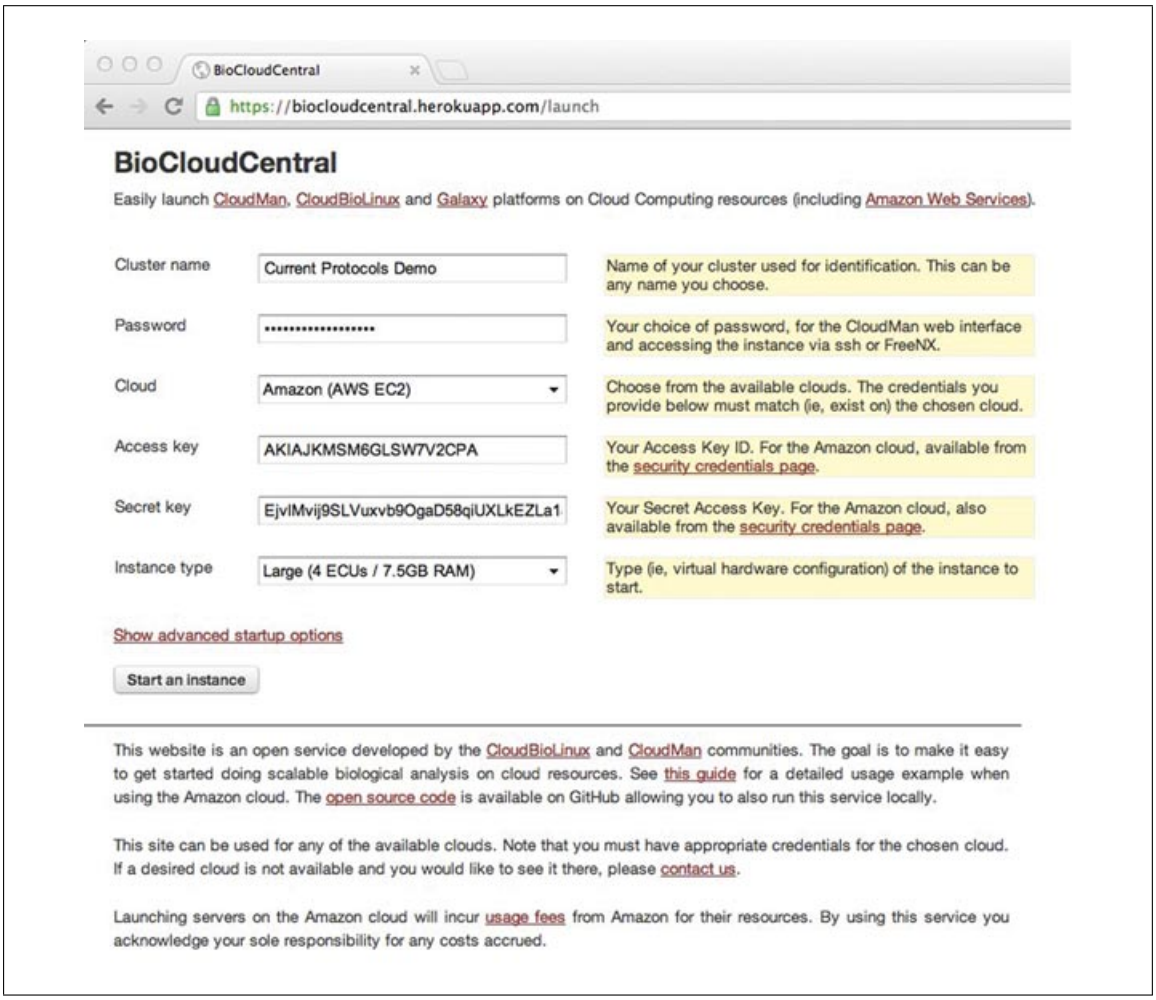

Figure 11.9.1 A snapshot of the BioCloudCentral portal showing all the form fields that are required to instantiate a CloudBioLinux and CloudMan instance.

are not permanently saved anywhere, but are instead simply cached on the server for the duration of user's browser session.

5. Choose the type of instance you wish to use. Micro instances represent an economical way to test and learn how the setup process operates, but they are not recommended for performing analyses. Large and Extra Large instances provide reasonable performance for the analyses described here.

6. Click the Submit button and, on the BioCloudCentral monitor page (Fig. 11.9.2), wait for the requested instance to enter the "running" state. This page displays some key information about the acquired instance; namely, the state of the instance, the instance IP address (used to access the instance), the command for accessing the instance via command-line interface (see Support Protocol 2), instance ID (as provided by AWS), the name of the security group (firewall rule set) associated with the instance, and the name of the key pair that can be used to access the instance via the command-line interface.

7. The instance information page provides an option to download User Data. The user data encapsulates all of the startup information provided in the initial form; it is a text file that contains information required to easily restart this same cluster at a later time directly from the AWS console. You should download and save this file to a local machine so that you may later create more clusters like this one.

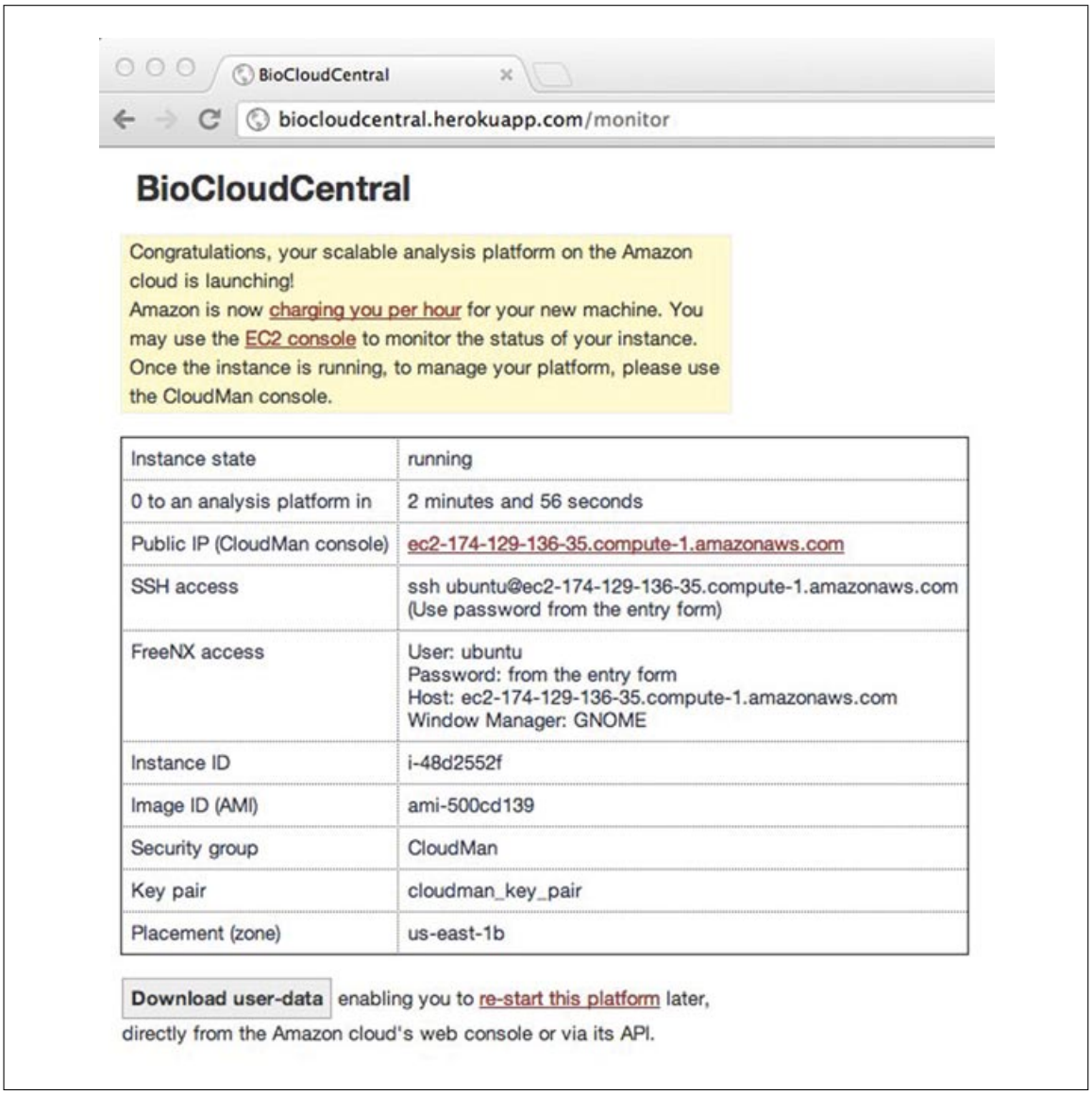

Figure 11.9.2 BioCloudCentral monitor page showing the details about the started instance. This page provides a direct link to the new instance as well as an option to download user data. These user data can be used to restart this same instance from the AWS console by uploading it in the instance wizard request form.

8. Connect to your CloudBioLinux instance and configure it via CloudMan. After the instance enters the running state, allow a few additional minutes for CloudMan to load. Then, click on the IP address of the instance where it appears in the BioCloudCentral monitor page. This will open a new browser tab showing CloudMan's Web interface (Fig. 11.9.3). This interface is used to control the newly created cluster. The first time a cluster is created, it is necessary to choose the type of cluster and, depending on the type of cluster, the amount of persistent data that wants to be associated with the cluster. The choice of these values depends on the intended usage of the cluster. CloudMan supports four types of clusters (accessible by clicking "Show more startup options" on the Initial Cluster Configuration box; Fig. 11.9.4).

   a. A "Galaxy cluster," which sets up Galaxy, available tools, reference data, a job manager (i.e., SGE), all the required services, and a persistent data volume (i.e., a file system mounted at `/export/data`).

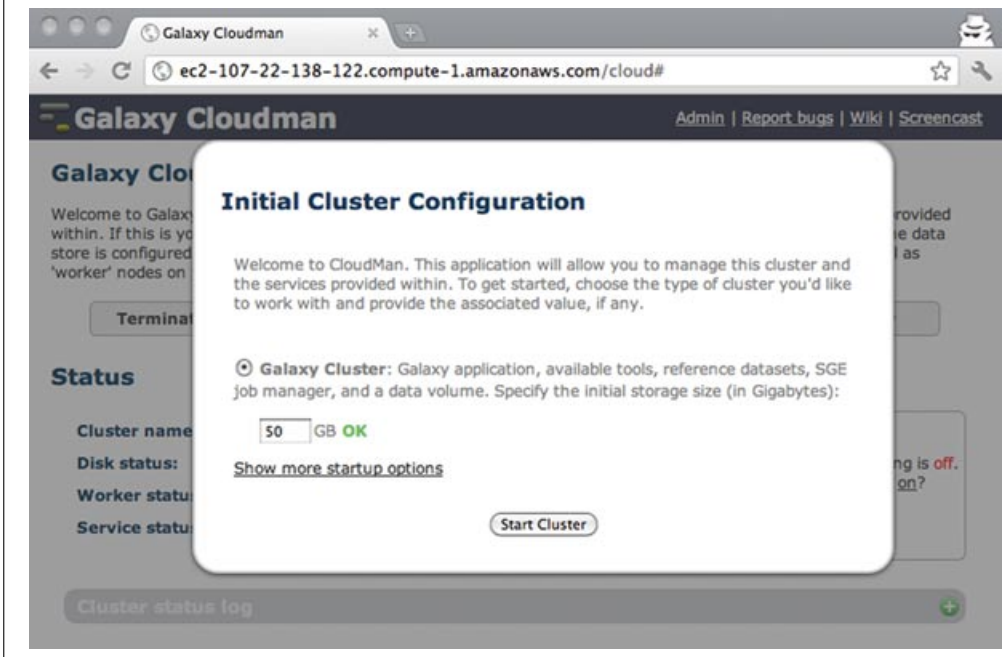

Figure 11.9.3 The CloudMan Web console used to manage the cluster.

b. A "Share-an-instance cluster," which allows you to instantiate an exact copy of someone else's cluster instance (see Basic Protocol 2).

c. A "Data cluster," which creates a complete cluster and associates a persistent data volume with the given cluster but does not set up Galaxy.

d. A "Test cluster," which sets up a complete cluster (just like the Data cluster) but does not persist any of its parts beyond termination.

The "Galaxy cluster" type is the most versatile option supported by CloudMan and, for the purposes of this protocol, should be selected. Along with the cluster type, the amount of persistent data storage associated with the cluster needs to be specified. The value should be chosen based on the intended use and the size of data to be analyzed with the cluster. Note that this value can be adjusted at cluster runtime, but the volume resizing process will stop any jobs running at the time. For use with Basic Protocol 1, specify 10 GB.

Click "Start cluster" and wait another few minutes for CloudMan to configure the cluster. The cluster is ready for use when the "Access Galaxy" button becomes active.

9. The CloudMan console is intended for management of the cluster. CloudMan's Web interface is grouped into three sections (see Fig. 11.9.5): the top third allows general control over the cluster, such as adding and removing worker nodes, easy access to the Galaxy application, and termination of the cluster. The middle section provides a general overview of the status of the cluster, such as the name of the cluster, the size of the persistent data associated with the cluster, and the global status of cluster services. The green icon next to the cluster name allows the user to package a cluster in its entirety for easy sharing with others. The disk icon next to the persistent storage status allows resizing of the data volume associated with the cluster. Clicking on either of the two icons provides more information about how to use the respective features. The bottom third of CloudMan's Web interface provides a more detailed set of log messages. In addition to the main console interface, there is the Admin interface, accessible via a link in the top right-hand corner of the

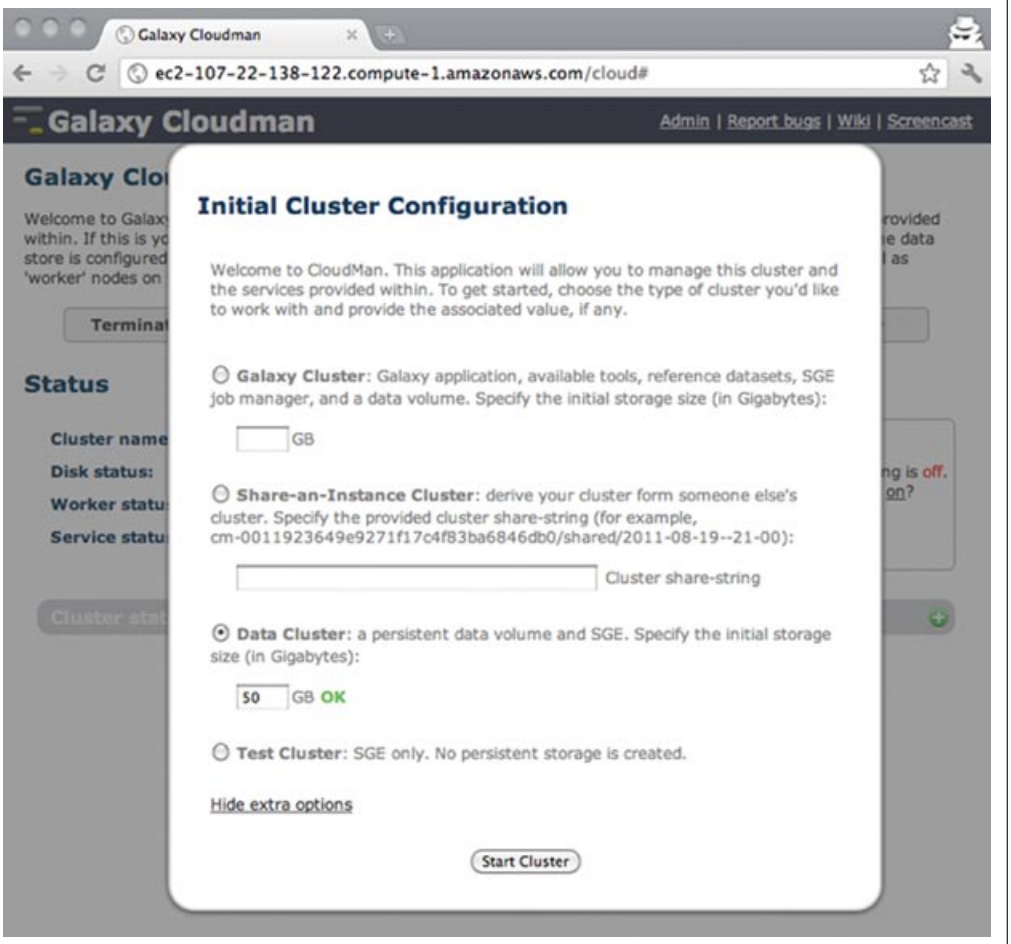

Figure 11.9.4 The initial CloudMan cluster configuration box. Here, it is possible to choose from the different cluster types supported by CloudMan. Depending on the cluster type, input may be required.

console. The Admin interface provides a more detailed overview and control over various services and features supported by CloudMan. The Admin page can be used to control or troubleshoot available services (e.g., if a service does not start, try restarting it via the Admin interface).

10. Once an analysis cluster is no longer needed, it should be terminated; otherwise, in the case of AWS, you will continue to be charged for as long as the acquired resources are running. To terminate your cluster, on the CloudMan console, click the "Terminate cluster" button in the top third of the interface. You will be given an option to terminate the master instance as well as delete the complete cluster. Simply terminating the cluster without deleting it will allow you to easily recreate the same cluster with all of the data at a later time. To do so, simply start a new instance as described in steps 1 to 5 and provide the same values in the presented form. It is also possible to start the same instance from the AWS Management Console and using the user data downloaded in steps 6 to 7. If you choose to delete the cluster, all of the data will be permanently deleted and cannot be recovered.

    Note that if a cluster is not deleted, a persistent data volume will be kept in your AWS account. This volume holds all the data that was uploaded to and analyzed on the instance. Because this volume is associated with your account, your account will incur charges for the amount of provisioned data storage as long as the volume exists.

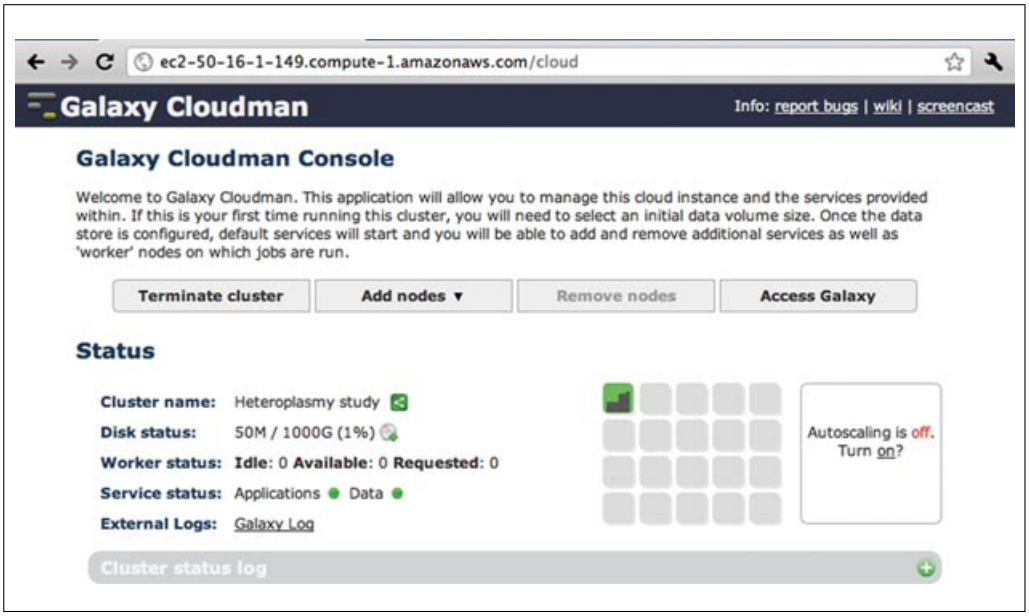

Figure 11.9.5 The main CloudMan interface used to control and manage the cloud cluster. Through this interface, it is possible to add and remove nodes from the cluster, monitor the status of cluster services, and manage cluster features such as auto-scaling and instance sharing. For the color version of this figure go to *http://www.currentprotocols.com/protocol/bi1109*.

SUPPORT PROTOCOL 1

ACCESS YOUR CloudBioLinux INSTANCE USING GRAPHICAL DESKTOP INTERFACE

After the instance setup is complete, the instance can be accessed via a graphical desktop interface. This interface acts like a remote desktop and provides all the features of a typical desktop.

Necessary Resources

Computer with Internet access and any up-to-date Web browser (Firefox, Safari, Opera, Chrome, Internet Explorer)

An AWS account with the Elastic Compute Cloud (EC2) and Simple Storage Service (S3) services enabled. To sign up for an account, visit *http://aws.amazon.com* and click the Sign Up Now link. Basic background information on the EC2 and S3 services can also be found at this page.

NX Client installed on the local machine; this software allows you to establish a graphical connection to remote computers. We recommend the free clients from NoMachine (*http://www.nomachine.com/download*).

1. The graphical desktop interface can be reached via an NX client from your local machine.

   In the case of FreeNX Client, start a new connection wizard, provide a name for your current session and the IP address of your instance (available from the instance monitor page on the BioCloudCentral portal or the AWS Management Console), and choose Unix GNOME desktop (see Fig. 11.9.6). Once the wizard is complete, establish a connection to the instance by specifying `ubuntu` user and providing the same password as you provided in the instance request from earlier (see Basic Protocol 1, step 3). After the connection is established, you will see the CloudBioLinux desktop interface with which you can normally work (see Fig. 11.9.7).

2. To end the session, simply close the connection window.

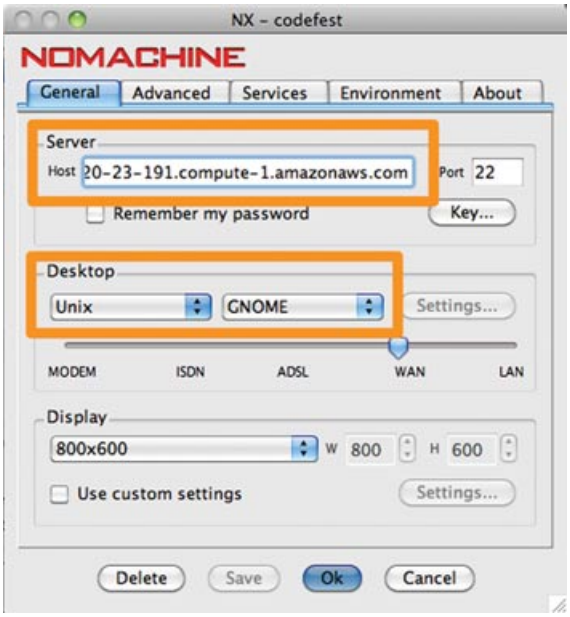

Figure 11.9.6 The NX client properties box specifying the IP address of the instance and the choice of GNOME desktop—both are required to establish a successful connection.

Figure 11.9.7 The remote CloudBioLinux graphical interface. Via this interface, it is possible to interact with the system as if it was a local workstation; standard Ubuntu menus and tools are available via the point-and-click interface.

ACCESS YOUR CloudBioLinux INSTANCE USING THE COMMAND-LINE METHOD

In addition to accessing your instance through the graphical interface, it is possible to access the instance via the command-line interface. This method is suitable for text-only environments, such as composing and running custom scripts or running parallel jobs.

Necessary Resources

Computer with Internet access and any up-to-date Web browser (Firefox, Safari, Opera, Chrome, Internet Explorer)

An AWS account with the Elastic Compute Cloud (EC2) and Simple Storage Service (S3) services enabled. To sign up for an account, visit *http://aws.amazon.com* and click the Sign Up Now link. Basic background information on the EC2 and S3 services can also be found at this page.

SSH client application available on the local machine (e.g., OpenSSH, Putty, Terminal)

1. Connect to your instance via SSH. The monitor page on the BioCloudCentral portal provides the command used to connect to the instance. Copy that command in the SSH client application on your local machine and hit Enter:

   ```
   ssh ubuntu@ec2-107-20-23-191.compute-1.amazonaws.com
   ```

   The text after the `@` symbol is the IP address of your running instance; the IP address shown in the example above is for illustration only. When prompted for the password, provide the password you selected when launching the instance on the BioCloudPortal.

   The BioCloudCentral portal provides an option to save a key pair that might have been generated at instance request time. A key pair (named `cloudman_key_pair`*) will be created in your AWS account if a key pair of the same name does not already exist. If the key was created, the monitor page allows downloading the private part of the key by clicking the key name. Note that downloading of the key is possible only following key creation (i.e., if another instance is created, and the key already exists, it will simply be used but it will not be possible to download the private portion of the key). As a result, it is important to download and save the key to a safe location the first time it is created. Once saved, it is necessary to set restrictive permissions on the key file so that only the current user can read it (e.g.,* `0400`*). If you did not save the private key file initially, simply delete the key pair from the AWS console and another one with the same name will be created the next time an instance is requested.*

   Access to the private part of the key pair allows an alternative method for accessing the instance, and the only method if the instance was started through the AWS console. In the case of using the key pair to access the instance, enter the following command in the SSH client application on your local machine:

   ```
   ssh -i path_to_key_file ubuntu@ec2-107-20-23-191.
   compute-1.amazonaws.com
   ```

   Again, the IP address shown above is for illustration only and should be replaced with the correct address of the instance.

2. Once logged in, the provided environment acts just like any other cluster environment. While logged in as the `ubuntu` user, any data stored to `/export/data` will persist beyond the life of the instance; all other modifications or customizations of the instance will be lost after the instance is terminated.

3. In addition to the `ubuntu` user that was used to logged in to the instance, it is possible to use the `root` and `galaxy` accounts. To access the `root` account, simply type `su -s`. To access the `galaxy` account, type `sudo su galaxy`. The `root` user

should be used sparingly, if at all. When logged in as the `galaxy` user, persistent data need to be stored under `/mnt/galaxyData` file system.

4. The Galaxy cluster type (see Basic Protocol 1, step 8) makes two additional file systems accessible, one for all the tools used by Galaxy and one for the reference genomes. These file systems are accessible at `/mnt/galaxyTools` and `/mnt/galaxyIndices`, respectively. If any changes are performed on either of these two file systems and one wishes them to be preserved beyond the life of the instance, they need to be persisted. This is possible via the Admin page on the CloudMan console. Note that if any changes to the file systems are performed from the CloudMan console, no users should be logged into the instance.

5. To exit the SSH session, simply type `exit` at the command prompt or press `ctrl-D`. This logs you out and terminates the connection, but leaves the instance running. When terminating the instance itself (see Basic Protocol 1, step 10), no users should be logged in to the instance.

PERFORMING VISUAL ANALYSIS WITH THE CloudBioLinux GRAPHICAL USER INTERFACE

BASIC PROTOCOL 2

This protocol presents an example of phylogenetic analysis using bioinformatics tools available under CloudBioLinux, and shows you how to add a new tool needed in the analysis that is not prepackaged with the distribution. To explore these features, we present a data analysis example: exploring the phylogenetic relationships of brown trout haplotypes using Bayesian inference.

Necessary Resources

You must have completed Basic Protocol 1 (steps 1 to 9) and Support Protocol 1 before advancing to this protocol

1. Launch the NX client and connect to the remote desktop as described in Support Protocol 1.

2. On the remote desktop within the NX client window, open the preinstalled Firefox Web browser (Applications → Internet → Firefox Web Browser) and download the input files to be used throughout this protocol.

   All of the sample data used in this protocol can be downloaded from an S3 bucket accessible at http://cpbi.s3-website-us-east-1.amazonaws.com/. Download the sequences file (`SP1_file1.txt`*) and the MrBayes block file (*`SP1_file2.nxs`*). Save those files on the remote instance to folder* `/export/data`.

3. Run ClustalX tool (Larkin et al., 2007) using `SP1_file1.txt` from step 1 as the input file.

   Start ClustalX from the Applications → Bioinformatics → clustalx menu on the remote desktop (Figs. 11.9.7 and 11.9.8).

4. Make sure to check Nexus format (`nsx`) in the Output Format Section under the Alignment menu, as this output file will be needed in the following steps:

   Choose Do Complete Alignment under the Alignment menu.

   Save the results of this step to a file called `SP1_file1.nxs`.

5. Although CloudBioLinux comes prepackaged with a large number of tools that are ready to use, there may be cases where a desired tool is not available. To showcase the flexibility of the system, this step describes how to add a new tool to your instance. This demonstrates the ability to customize an instance so that it meets your needs.

Figure 11.9.8 ClustalX application on the remote instance with the sample dataset loaded.

Customization will be done by installing the jModelTest tool (Guindon and Gascuel, 2003; Posada, 2008), which will be used to determine priors for Bayesian inference. For this example, we will determine priors by the Akaike Information Criteria (AIC) model and use them as the MrBayes block in step 6.

a. Visit *http://darwin.uvigo.es/software/jmodeltest.html*, register, and download the tool. Open the file with the Archive Manager and copy the contents of directory `jModelTest 0.1 package` to `/export/data`. Open the copied directory, right click the `jModelTest.jar`, and chose Properties. Under the Permissions tab, check the box that allows execution of the file.

b. Open the terminal application (Applications → Accessories → Terminal), change into the `jModelTest directory` (`cd /export/data/jModelTest/0.1/; package`), and start the tool: `java -jar jModelTest.jar`.

c. Load the alignment file that was generated in step 2 of this protocol and, from the Analysis menu, choose "Compute likelihood scores."

d. Select AIC from the Analysis menu. Check "Write PAUP* block" and start the computation. Note that this computation will also take several minutes to complete.

6. Compose an input file that will be fed to MrBayes. This file is a composition of the initially downloaded `SP1_file2.nxs` file and the results from steps 1, 4, and 5d of this protocol.

   a. Open the `SP1_file1.nxs` file and paste the contents of the MrBayes block (file `SP1_file2.nxs`, Fig. 11.9.9) at the bottom of it. Then, adjust the values in this block with the values obtained in step 5. Save the file as `SP1_file3.nxs`.

7. Run the MrBayes tool (Huelsenbeck and Ronquist, 2001) using the `SP1_file3.nxs` as the execution file.

Figure 11.9.9 A snapshot of the MyBayes block (file `SP1_file2.nxs`) that needs to be manually adjusted with the results from step 5 of Basic Protocol 2. Append the edited block to the end of file `SP1_file1.nxs` and save the resulting file as `SP1_file3.nxs`.

a. At the MrBayes prompt (>), type `exe SP1_file3.nxs`, which will execute the commands available in the specified file. The result will be a consensus phylogenetic tree with the posterior probabilities for each clade.

b. Output from the MyBayes tools (`SP1_file4.*` files) is saved to user' home directory (`/home/ubuntu`). In order to persist these files beyond the life of this cluster instance, it is necessary to copy the files to the `/export/data` directory.

8. At this point, the analysis is complete and the cluster can be terminated (see Basic Protocol 1, step 10). Alternatively, this same cluster may be left alive and utilized for Basic Protocol 4.

USING A CloudMan CLUSTER TO PERFORM A PARALLEL ANALYSIS

BASIC PROTOCOL 3

CloudBioLinux and CloudMan provide a framework for creating and sharing parallel analysis pipelines. This section shows the utility of this approach by demonstrating a pipeline for processing next-generation sequencing data. It takes advantage of CloudMan shared instances, the SGE cluster with a shared file system, and custom Galaxy integration.

The parallel analysis pipeline starts with next-generation sequencing data in FASTQ format. A custom Galaxy front end runs an automated pipeline that:

- Aligns the sequences with BWA (Li and Durbin, 2009)
- Manages and sorts the resulting files with Picard (*http://picard.sourceforge.net*)
- Performs recalibration, realignment and variant calling with the Genome Analysis Toolkit (DePristo et al., 2011)
- Prepares quality metrics with FastQC (*http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/*)
- Uploads results into the Galaxy framework for visualization and further analysis

All components of the pipeline are open source and designed for customization. The goal of this section is both to present a useful analysis and also to demonstrate the capabilities of CloudBioLinux and CloudMan for building your own pipelines.

Additional documentation for this workflow is also available along with webcasts of the process: *http://bcbio.wordpress.com/2011/11/29/making-next-generation-sequencing-analysis-pipelines-easier-with-biocloudcentral-and-galaxy-integration/*.

In order to follow along with the protocol below, one should have completed Basic Protocol 1 and possess a knowledge of navigating the UNIX command line.

Necessary Resources

Computer with Internet access and any up-to-date Web browser (Firefox, Safari, Opera, Chrome, Internet Explorer)

An AWS account with the Elastic Compute Cloud (EC2) and Simple Storage Service (S3) services enabled. To sign up for an account, visit *http://aws.amazon.com* and click the Sign Up Now link. Basic background information on the EC2 and S3 services can also be found at this page.

Cluster startup

1. Replicate the steps for starting an EC2 instance from Basic Protocol 1, steps 1 to 7 only. Use either the BioCloudCentral Web site or the Amazon EC2 console with a correctly defined set of user data. The goal is to start a CloudMan instance and have access to the CloudMan Web interface but not to choose the type of CloudMan cluster.

2. Enter your CloudMan password in the Web interface, and click on "Show more startup options" to start a "Share-an-instance cluster" containing next-generation sequencing data (see Fig. 11. 9.4). The identifier of the shared cluster is:

   `cm-b53c6f1223f966914df347687f6fc818/shared/2011-11-29--01-44`

3. You can retrieve the most recent identifier from the on-line documentation available at *http://j.mp/uNXZY6*.

4. Allow the boot process to complete by monitoring the console in the CloudMan Web interface.

5. Use the Add Nodes button in the CloudMan Web interface to add additional nodes to your cluster. You will want to add at least one new node. Monitor the CloudMan console until the node has started.

6. Click on the Access Galaxy button to go to a customized Galaxy instance running on your cluster.

Run analysis

7. Log in to the Galaxy instance with the example user:

   username: `example@example.com`

   password: `example`.

8. Import example data into your active history. Click on Shared Data/Data Libraries, then select "Exome example." Select two of the FASTQ files plus `baits.bed` and `targets.bed`, and click "Import to current history."

9. Click on Analyze Data and view the input files in your data history. The FASTQ files contain next-generation sequencing read data, and the BED files contain lists of target and bait files used in hybrid selection.

10. Click on Pipeline to view a custom interface designed for running the analysis workflow. The Wizard interface walks through each step of the process:

    a. Select the "Variant calling" analysis.

    b. Drag and drop the reads as the read and read pair to process.

    c. Select target and bait BED files. Other parameters are also available for adjustment if desired, but the defaults work well with the provided sample data.

d. Review the input parameters and press "process" to start the analysis. This page also contains the Data Library name where the pipeline uploads finalized results.

This starts the distributed processing pipeline. This can take hours to run with next-generation datasets; the example dataset takes 5 hr with a cluster of two large instances. With your own datasets, you can monitor the process directly on the CloudBioLinux cluster using the detailed instructions below. With the example data, you do not need to wait, as results from a previously run pipeline are available.

View results in Galaxy

11. View the data library with results by clicking on Shared Data/Data Libraries, then selecting "example@example.com" and the output directory name you noted in step 10d of running the analysis. There are also pre-populated directories available to use.

12. Import files into your Galaxy history for visualization. Check the BigWig, summary PDF, and variant VCF files, and click on "Import to current history."

13. Click "Analyze data" to return to your Galaxy analysis history to view the imported files:

    a. Use the 'eye' icon to view the VCF file, which is a text file of called variants with locations, dbSNP identifiers, and annotations.
    b. View the PDF summary file with the 'eye' icon. This contains a summary table to variants called along with diagnostic plots of read and alignment quality.
    c. Click on the BigWig filename, then select 'display at UCSC.' This will take you directly to the UCSC Genome Browser with the read coverage information included as a custom track. Choosing the dense coverage option for viewing the track will display coverage as peaks. The example dataset is on chromosome 22, so narrowing to a gene region on "chr22" will show coverage information localized to exon regions.

Monitoring the process

14. To monitor a custom analysis or better understand the parallel pipeline, directly log into your cluster with SSH or an NX client following the instructions in step 1 of Support Protocol 1.

15. On the instance, change to the base work directory, `/export/data/work`. This will have a working directory for your submission matching the output directory name in step 10d of running the analysis. There is also an example directory (`111121_V5ARGC`) corresponding to the precomputed output data library.

16. The directory contains analysis files from the pipeline run. These include intermediate files and temporary directories produced as part of the processing. The most useful files for monitoring an in-progress pipeline are the SGE log files, which start with `nextgen_analysis_server.py.o`. Each file contains the in-depth log information on processing for one of the distributed nodes. Viewing these will provide details about the stage of processing.

17. The CloudMan SGE manages job running, so standard SGE commands provide detailed insight into cluster usage and job status. `qstat -f` will display available nodes along with running and scheduled jobs.

USING A PRIVATE, SCALABLE GALAXY ANALYSIS ENVIRONMENT ON TOP OF CloudMan

BASIC PROTOCOL 4

As indicated in Basic Protocol 1, step 8, CloudMan supports creation of different types of cloud clusters. By choosing the Galaxy cluster type when setting up a CloudMan cluster, CloudMan will provision necessary resources and make the Galaxy application, many

reference genome datasets, a range of tools, and persistent data available to the user. The user can then use the familiar Galaxy Web interface to perform complete analyses without need to set up any of the underlying components.

To follow along with this protocol it is necessary to have completed steps 1 to 8 of Basic Protocol 1.

Necessary Resources

Computer with Internet access and any up-to-date Web browser (Firefox, Safari, Opera, Chrome, Internet Explorer)

An AWS account with the Elastic Compute Cloud (EC2) and Simple Storage Service (S3) services enabled. To sign up for an account, visit *http://aws.amazon.com* and click the Sign Up Now link. Basic background information on the EC2 and S3 services can also be found at this page.

1. After cluster initialization by CloudMan is complete, indicated by the button Access Galaxy becoming active, click on it and a new tab will open with your own instance of Galaxy. Galaxy is now configured and ready to be used. It is desirable to register a user account on this new Galaxy instance, so any analysis steps will be preserved even if the cluster is terminated.

RNA seq data import

We will use the *Drosophila melanogaster* RNA-seq data from two stages during embryonic development produced by the modENCODE project. The files have already been mapped using the TopHat spliced aligner with multi-mapping allowed.

2. Click on "Get Data/modENCODE modMIne" server; select the Data tab and open "See all fly modENCODE submissions" in a new tab in the browser. In the upper part of the list you will find "Dev Timecourse Embryo 0-2h RNA-Seq multimapped reads"; click on the link and go to "External links → Download data submitted to DCC" (found by the right margin of the window). Select `1_accepted_hits.DCCheader.sam.gz.bam.sorted.bam` using your mouse cursor and copy the path to the clipboard. Return to the Galaxy tab, click "Get Data/Upload File," and paste the saved link in the URL/Text text box. Set the genome to "D. melanogaster Apr. 2006 build," and click execute. Repeat the process, but this time select the first sorted `bam` file in the `Dev Timecourse Embryo 2-4h RNA-Seq multimapped reads` folder.

3. Select the Edit Attributes pencil icon on the corresponding history item and rename the files as desired (e.g., `0-2h` and `2-4h`; Fig. 11.9.10).

Transcriptome reconstruction and transcript expression quantification

4. Next, construct the gene model using the Cufflinks transcriptome assembler (Trapnell et al., 2010). Within Galaxy, click "NGS: RNA Analysis," and select "Cufflinks transcript assembly." In the "SAM or BAM file of aligned RNA-Seq reads" dropdown menu, select one file and execute the analysis using the default parameters. Repeat the process for the second file. Format of the output files is briefly explained in the help text beneath the Execute button of the Cufflinks tool, while the complete tool manual can be found at: *http://cufflinks.cbcb.umd.edu/manual.html.*

Comparison of gene models with reference annotation

5. Go to "Get Data/UCSC" Main and select clade: "Insect, genome D. melanogaster, assembly: Apr. 2006, group: Genes and Gene Prediction tracks, track: FlyBase Genes." Under the output format, select the "GTF – gene transfer" format and mark "Send output to Galaxy." Click "get output."

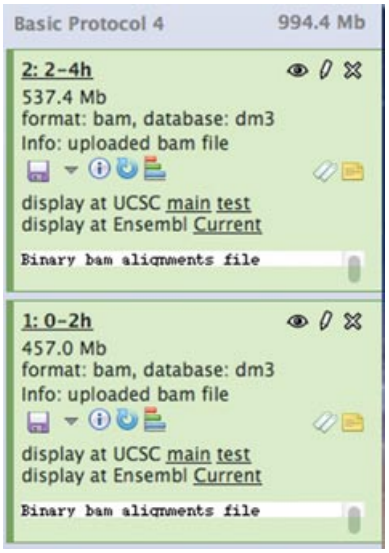

Figure 11.9.10 Galaxy history view with the two RNA datasets transferred from modENCODE.

6. Because there is a discrepancy between the chromosome names in the files containing mapped reads and the reference annotation, we need to remove the "chr" prefix from the names in the first column of the table. Click on "Text Manipulation → Trim." Under "Trim this column only," state column `1`, and under "Trim from the beginning to this position," state number `4`. After you click "execute," you will get a new reference file that contains properly formatted chromosome names for the downstream analysis. Change the name of the dataset to "Reference annotation" (by clicking on the pencil icon).

7. To compare the newly assembled transcripts from each developmental stage with the acquired reference transcriptome, use the Cuffcompare tool within Galaxy. Go to "NGS: RNA Analysis" and click Cuffcompare. Under the "GTF file produced by Cufflinks" drop-down menu, select one of the transcriptomes from one of the previous stages and add the other one by clicking on the "Add new Additional GTF Input files." Select "yes" under Use Reference Annotation and make sure the "Reference annotation" dataset is selected (Fig. 11.9.11). Start the execution of the application.

Differential expression calculation

8. Using the completed gene models and the expression estimation for each sample, test whether some of the genes are differentially expressed in a certain condition. This is done using Cuffdiff. Go to "NGS:RNA Analysis," click on the Cuffdiff link, and under the drop-down menu, select the GTF file produced by Cuffcompare. Now select both of the files that contain the mapped reads under the SAM or BAM file of aligned RNA-Seq reads. Change the default parameter: "Perform quartile normalization" from No to Yes and start the analysis.

Functional annotation of differentially enriched genes

9. When we get a list of differentially expressed genes, one of the most informative steps in downstream analysis is to find out what those genes actually do. We can get insight into the possible function of our genes by finding the statistically over-represented functional categories associated with the gene set. For this we will use the DAVID tool (Huang et al., 2009a,b). Before using the DAVID tool, we first need to map the identifiers in the third column of the table that contains the differentially

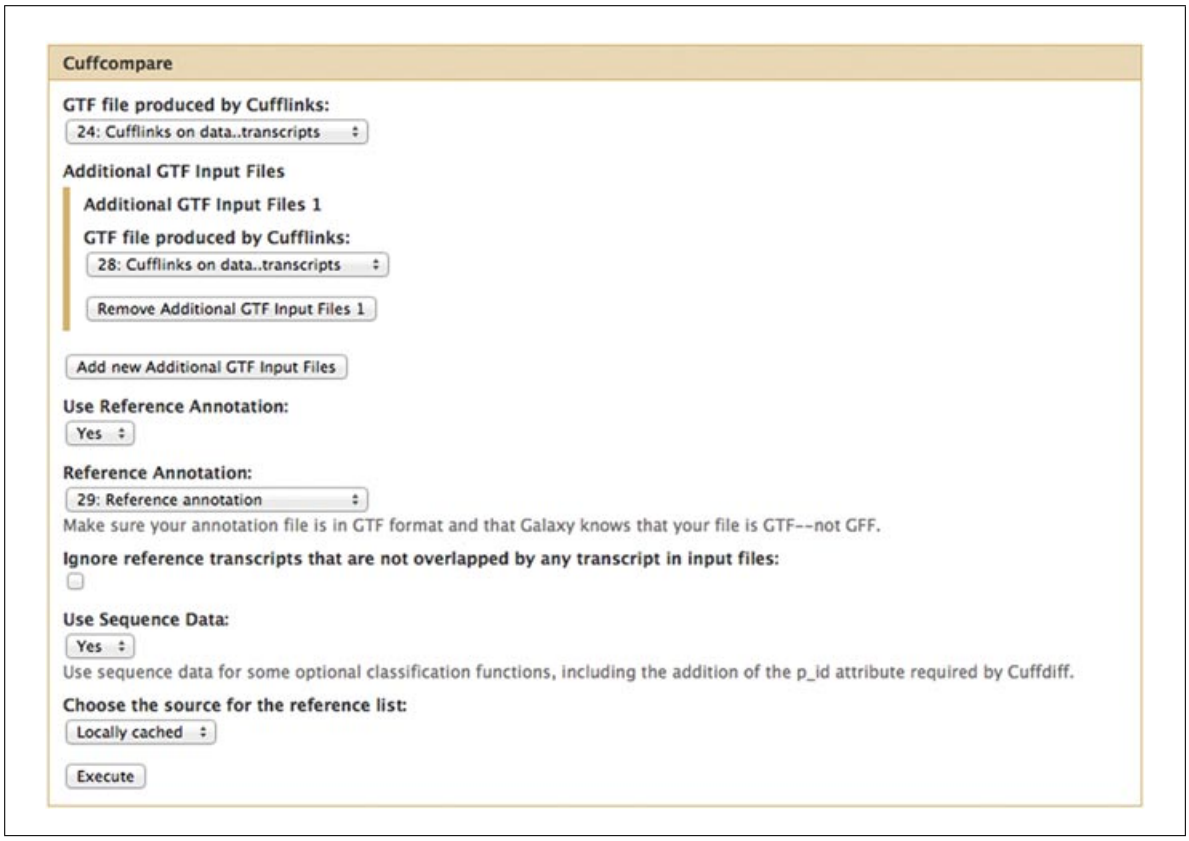

Figure 11.9.11 The Cuffcompare tool interface within Galaxy with all the options described in Basic Protocol 4, step 7, selected.

expressed genes to identifiers that can be used by the DAVID tool. For this, we will use RefSeq ids. Go to Get Data/UCSC main, select the same attributes as in step 5, but under "output format" state selected fields from primary and related data sets. Click the "get data" button. The screen will now split into two parts. In the lower part (named Linked Tables), mark flyBaseToRefSeq, and click on "Allow Selection From checked tables." Under the dm3.flyBaseToRefSeq fields, mark the two boxes and click on "done" with selection → Send Output to Galaxy.

10. Now that you have a table of mappings from the Berkeley Drosophila Genome Project gene ID to refseq IDs, filter the identifiers of the genes that were differentially expressed. Go to Human Genome Variation/David, select the output of the mapping table, and mark the column containing the subset of Refseq identifiers. Identifier type is "Refseq mrna." Run the analysis. The results will be a link to the DAVID Web application. By clicking on the link, a new tab will open.

11. In this step, we will select the functional groups that will be tested for over-representation. Click on the Clear All button found in the middle of the screen. Open the Gene Ontology set and mark the three red-colored terms. This will select GO super sets Biological Process, Molecular Function, and Cellular Component. By clicking on the functional annotation chart, you will get a list of Gene Ontology terms found over-represented in the set you submitted, ordered by their corresponding *p* values (Fig. 11.9.12). With this step, the basic RNA-seq analysis is done. However, note that a rigorous analysis of differential expression always needs to include more than one sample per condition (the use of biological replicates will improve both the sensitivity and the specificity of the procedure), and for you to get the optimal

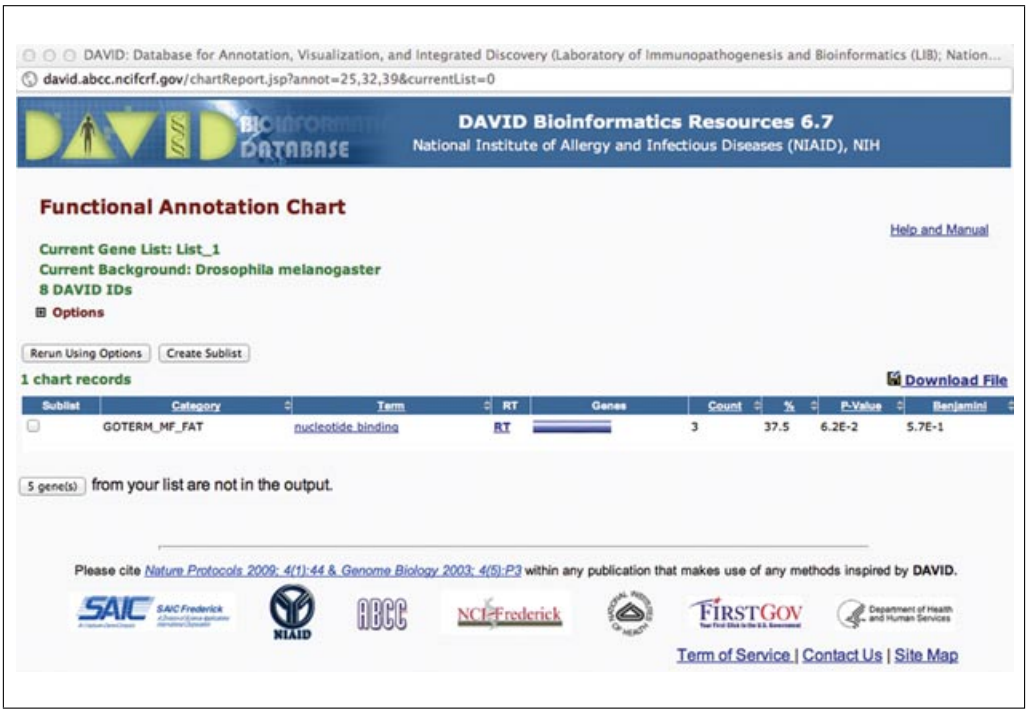

Figure 11.9.12 List of Gene Ontology terms found over-represented in the submitted dataset, ordered by their corresponding *p* values, as returned by the DAVID tool. For the color version of this figure go to *http://www.currentprotocols.com/protocol/bi1109*.

results for your experiment, you will need to spend some time learning and playing with parameters of all of the tools used in the analysis.

12. At this point, the analysis is complete and the cluster can be terminated as shown in Basic Protocol 1, step 10.

COMMENTARY

CloudBioLinux and CloudMan successfully bridge the gap between the flexibility and accessibility that cloud computing offers and the high-level functionality a user desires. The combination of the two projects allows researchers and small groups to easily, quickly, and relatively cheaply gain access to a functional and configured compute infrastructure. This infrastructure can be configured in a matter of minutes, provides convenient access to an extensive set of bioinformatics tools, and can be terminated when no longer needed. As a result, it is possible to use the infrastructure periodically (depending on the data volume), as a test platform (to test a tool on exotic resources), or in addition to an existing infrastructure (as an overflow valve).

Users do not need to delve into the components of the underlying infrastructure or platform configuration, but can instead focus on performing their analyses through a point-and-click interface. If additional functionality is required, users can access the system at the command-line level, install new software, and otherwise customize the system to meet their needs. Any changes or customizations can be persisted and shared with others, thus creating and supporting an evolvable collaborative research environment. All of the code and documentation associated with these projects is open source and freely available. The configuration of machine images used in the process is also automated, lending itself to complete transparency and reproducibility. Such an approach facilitates extensions of the projects, local development, and growth of the community.

The protocols shown here act as introductory material to get one familiarized with the process, possibilities, and functionality offered through these projects. Open-ended analyses are supported and new functionality continuously added. There are active user communities surrounding the projects, so help or advice can be obtained

through mailing lists or otherwise existing documentation (see *http://www.cloudbiolinux.org/* and *http://usecloudman.org/*).

Acknowledgments

This project was supported by American Recovery and Reinvestment Act (ARRA) funds through grant number RC2 HG005542 from the National Genome Research Institute, National Institutes of Health.

Literature Cited

Afgan, E., Baker, D., Coraor, N., Chapman, B., Nekrutenko, A., and Taylor, J. 2010. Galaxy CloudMan: Delivering cloud compute clusters. *BMC Bioinformatics* 11:S4.

Afgan, E., Baker, D., Nekrutenko, A., and Taylor, J. 2011a. A Reference Model for Deploying Applications in Virtualized Environments. Concurrency and Computation: Practice and Experience. John Wiley & Sons, Hoboken, New Jersey.

Afgan, E., Goecks, J., Baker, D., Coraor, N., Nekrutenko, A., and Taylor, J. 2011b. Galaxy: A gateway to tools in e-Science. *In* Guide to e-Science: Next Generation Scientific Research and Discovery (X. Yang, L. Wang, and W. Xie, eds.) pp. 145-177. Springer, New York.

DePristo, M., Banks, E., Poplin, R., Garimella, K., Maguire, J., Hartl, C., Philippakis, A.A., del Angel, G., Rivas, M.A., Hanna, M., McKenna, A., Fennell, T.J., Kernytsky, A.M., Sivachenko, A.Y., Cibulskis, K., Gabriel, S.B., Altshuler, D., and Daly, M.J. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* 43:491-498.

Field, D., Tiwari, B., Booth, T., Houten, S., Swan, D., Bertrand, N., and Thurston, M. 2006. Open software for biologists: From famine to feast. *Nat. Biotechnol.* 24:801-803.

Goecks, J., Nekrutenko, A., and Taylor, J. 2010. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* 11:R86.

Guindon, S. and Gascuel, O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.* 52:696-704.

Huang, D.W., Sherman, B.T., and Lempicki, R.A. 2009a. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* 37:1-13.

Huang, D.W., Sherman, B.T., and Lempicki, R.A. 2009b. Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. *Nat. Protoc*. 4:44-57.

Huelsenbeck, J. and Ronquist, F. 2001. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* 17:754-755.

Larkin, M., Blackshields, G., Brown, N., Chenna, R., McGettigan, P., McWilliam, H., Valentin, F., Wallace, I.M., Wilm, A., Lopez, R., Thompson, J.D., Gibson, T.J., and Higgins, D.G. 2007. Clustal W and Clustal X version 2.0. *Bioinformatics* 23:2947-2948.

Li, H. and Durbin, R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:1754-1760.

Posada, D. 2008. jModelTest: Phylogenetic model averaging. *Mol. Biol. Evol.* 25:1253-1256.

Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J., and Pachter, L. 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28:511-515.

Key Reference

Afgan, E., Baker, D., Coraor, N., Goto, H., Paul, I.M., Makova, K.D., Nekrutenko, A., and Taylor, J. 2011. Harnessing cloud computing with Galaxy cloud. *Nat. Biotechnol.* 29:972-974.

This article gives more detailed background and description as to the available features and perceived functionality when trying to use functionality described within this unit.