BioViz *Connect*: Web application linking CyVerse science gateway resources to genomic visualization in the Integrated Genome Browser

Nowlan H Freese[1]* nfreese@uncc.edu

Karthik Raveendran[1†] kraveend@uncc.edu

Chaitanya Kintali[1†] ckintali@uncc.edu

Srishti Tiwari[1] stiwari8@uncc.edu

Pawan Bole[1] pbole@uncc.edu

Chester Dias[1] aloraine@uncc.edu

Ann E Loraine[1]* aloraine@uncc.edu

[1] Department of Bioinformatics and Genomics, University of North Carolina at Charlotte, Kannapolis, NC 28081, USA

* Corresponding authors

[†] Karthik Raveendran and Chaitanya Kintali contributed equally to this work.

Abstract

Background: Visualization of genomic data is a key step in validating methods and results. Web-based science gateways such as CyVerse provide storage and analysis tools for genomic data but often lack visualization capability. Native, desktop visualization tools like Integrated Genome Browser enable highly interactive data visualization but are difficult to deploy in science gateways due to technical constraints. Developing ways for gateways to interoperate with independently developed, native visualization applications will enhance their value to users.

Results: We developed BioViz *Connect*, a new web application that *connects* the CyVerse science gateway system with native, visualization application Integrated Genome Browser (IGB) using the CyVerse Terrain Application Programming Interface (API) and a local endpoint within IGB. Using BioViz *Connect*, users can (i) stream data from the CyVerse data store into IGB for visualization, (ii) improve the IGB user experience for themselves and others by adding IGB specific metadata to CyVerse data files, including genome version and track appearance, and (iii) run compute-intensive visual analytics functions to create new data files for visualization in IGB or other applications. To demonstrate these three core functionalities of BioViz *Connect*, we describe an example use case in which RNA-Seq data from *Arabidopsis thaliana* plants undergoing heat and desiccation stresses are interactively and visually analyzed. The example use case shows how interactive visualization helps researchers detect artifacts and sanity-check results from statistical analysis using a visual analytics function too computationally intensive to run on the user's local computer. BioViz Connect helps users analyze and visualize their CyVerse data in IGB without needing to download or upload their data into new systems. Instead, the data exit the CyVerse platform in small, genome-region-based portions when users

request them via the IGB user interface itself. BioViz *Connect* is accessible from https://bioviz.org.

Conclusions: BioViz *Connect* enhances the CyVerse gateway by adding visualization capability. BioViz Connect demonstrates a new way to integrate science gateways with native, desktop applications using APIs.

Keywords: Integrated Genome Browser, CyVerse, Terrain API, Data Visualization, Data Storage, Genomics

Background

Science gateways are Web sites that implement user-friendly interfaces to high performance computing and storage systems [1]. Science gateways assemble and curate discipline-specific tools within a single, easy-to-use interface for the benefit of researchers in that discipline. Across different fields, gateways implement similar features, such as Web-based interfaces to command-line, Unix-based tools deployed to process and analyze experimental datasets too large or too sensitive to handle on a personal computer [2-4]. In typical workflows, domain researchers upload their "raw" (unprocessed) data to the gateway site and then use the gateway's interface to develop processing and analysis pipelines customized for their data, where a pipeline is defined as tasks performed in sequence by non-interactive tools which emit and consume well-understood file types and formats. Often, the file formats emitted or consumed are indicated by the file extension used, i.e., captured in the file name. Thus, for many systems, file names convey crucial information about data formats and meaning. Common pipeline tasks in genomics include aligning RNA-Seq sequences onto a reference genome to produce BAM (binary alignment) format files [5], generating scaled RNA-Seq coverage graphs from the "BAM" files using tools such as genomeCov [6], or searching

promoter regions for sequence motifs common to sets of similarly regulated genes using tools such as DREME [7].

A science gateway aims to provide a single point of access for tools needed to process and analyze data from a research project. However, desktop visualization tools, which usually have their own graphical user interfaces, are often difficult to deploy into Web-based science gateway systems. The reasons for this are mainly technical. Science gateways provide Web browser-based interfaces to remote storage systems that are nearly always inaccessible to native desktop applications. To use a native, desktop visualization application, users must first move the data to a new location where their desktop application can access it. Similar problems exist for Web-based visualization systems. Web-based applications typically require a Web or application server configured with data storage and delivery mechanisms specialized to the application. For example, using the Web-based UCSC Genome Browser to visualize genomic data requires either deploying data into a mirror of the browser's back end database or instead deploying data in native formats to Web-accessible, UCSC Browser-specific Track Hub sites [8]. Similarly, using the JBrowse Web-based genome browser requires deploying data to systems that can provide the front end user interface with JBrowse-specific JSON formats [9]. In both cases, a back end, application-specific data delivery system is required. Integrated Genome Browser from BioViz.org {Freese, 2016 #1;Nicol, 2009 #17} and the Broad Institute's Integrative Genomics Viewer [10], which are both native, desktop applications, have full access to the user's local file system, and they are designed to read and consume standard file formats directly, without requiring translation into browser-specific formats. Even so, they also require some way to transfer data out of the gateway system and into the visualization application.

Another typical requirement for science gateways is extensibility, meaning they require a strategy for gateway developers or users to add new tools to the system to accommodate or even potentiate new directions for research. To meet this need within the CyVerse ecosystem, the designers created a mechanism for users to add new, non-interactive tools in the form of

CyVerse Apps, which are user-configured container images that run within a CyVerse-provided container environment [11]. Users create containers using Docker and then contribute their container image along with metadata specifying input parameters and accepted data types to CyVerse. Once accepted and deployed, the container is configured to run as an asynchronous "job" via a queuing system. Thus, Apps run non-interactively and therefore are not well-suited to providing interactive, exploratory visualization. However, these Apps do provide a means to create new input data for visualization, as we explore here.

In this paper, we introduce BioViz *Connect*, a Web application that overcomes limitations described above to add genome visualization capability to the CyVerse Discovery Environment, part of the CyVerse science gateway system. Previously called iPlant, the CyVerse science gateway is a United States National Science Foundation funded cyberinfrastructure project with the aim of providing computational resources for life sciences researchers [3, 4]. We chose to work with the CyVerse Discovery Environment in this study because it features a rich Application Programming Interface (API), the Terrain REST API, that supports secure computational access to CyVerse data storage and analysis resources. Using this API, we implemented a new visualization-focused interface to these resources, using the Integrated Genome Browser as the demonstration application. We selected IGB because it offers one of the richest feature sets for visual analysis in genomics (for descriptions of IGB functionality, see [12-16]) and because we are members of the core IGB development team. Therefore, we possessed insider's knowledge of the featured visualization application that allowed us to modify IGB as needed for the project.

BioViz *Connect* enables users of Integrated Genome Browser, a highly interactive and mature desktop visualization application implemented in Java, to visually analyze their CyVerse data without having to download entire files to their local computer or migrate their data into application specific data stores. In the following sections, we describe how BioViz *Connect* is implemented, explaining the technology stack used and how BioViz Connect interacts with the

CyVerse science gateway resources via its Terrain API, a REST interface. Next, we describe how BioViz *Connect* enables flow of data into the IGB desktop software by activating a REST API endpoint residing in IGB itself. To illustrate the functionality, we describe an example use case scenario for BioViz *Connect*, focusing on visualization of an RNA-Seq data set from *Arabidopsis thaliana* plants undergoing desiccation and heat stresses. During the use case discussion, we highlight how coordinate interaction between remote and local resources enables interactive visualization, helping researchers discover biologically meaningful features in data while avoiding artifacts.

Implementation

BioViz *Connect* client and server-side design

BioViz *Connect* consists of two parts: a JavaScript-based user interface that runs in a Web browser and a server-side application that manages authentication and communication with Terrain API endpoints. The user interface code on the client-side is implemented using HTML5, CSS, Bootstrap 4.3.1, JavaScript, and jQuery 1.10.2. The server-side code is implemented in python3 using the Django web application framework (Fig. 1). The currently available production instance of BioViz *Connect* is deployed on an Ubuntu 18.04 system and hosted using the apache2 Web server software as a reverse proxy. BioViz *Connect* code is open source and available from https://bitbucket.org/lorainelab/bioviz-connect.

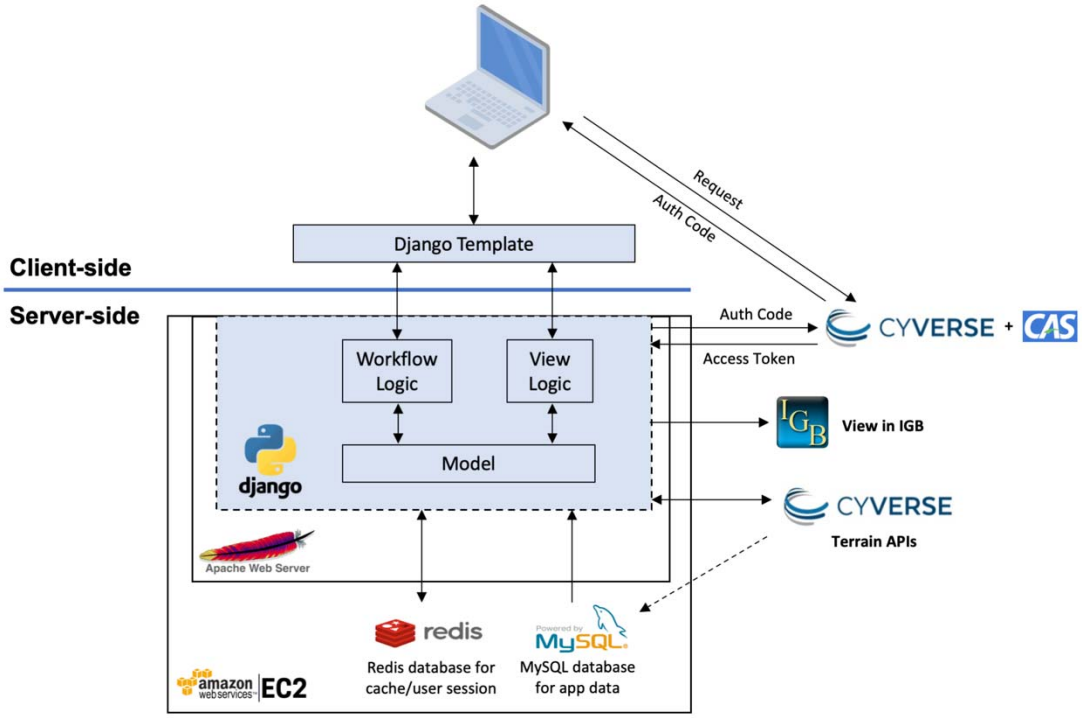


Fig. 1 Diagram illustrating client and server-side design of BioViz *Connect*.

Connecting to the CyVerse Terrain API

A free CyVerse Discovery Environment account required to access the Terrain API for testing and development is available from https://user.cyverse.org/register. For production, user authentication is handled through Central Authentication Service (CAS) OAuth hosted by CyVerse (Fig. 1). CyVerse infrastructure manages user accounts; no BioViz-specific accounts or passwords are required. Once the CAS endpoint has authenticated the user's name and password, an authorization code is returned as a parameter of the callback URL. BioViz *Connect* uses the authorization code along with the BioViz *Connect* client id and secret to obtain an access token from Terrain (Fig. 1). Other groups wishing to deploy BioViz Connect code under a non-BioViz domain should contact CyVerse support to request a client id and secret for their domain. BioViz *Connect* server-side code uses a Redis database to store a user's access token during a session, which is identified by a session id stored in the database and in the

user's Web browser. This access token can then be used to authorize calls to the Terrain API on behalf of the user.

Connect to Integrated Genome Browser

Integrated Genome Browser is a free, open-source desktop software program written in Java which users download and install on their local computer systems [12, 15]. Installers for Linux, MacOS, and Windows platforms are available at https://bioviz.org. The IGB source code resides in a git repository hosted on Atlassian's bitbucket.org site (https://bitbucket.org/lorainelab/integrated-genome-browser). IGB version 9.1.4 or greater is required for IGB to connect to BioViz *Connect*.

IGB contains a simple Web server configured to respond to REST-style queries on an IGB-specific port on the user's local computer. Clicking the "View in IGB" button in BioViz *Connect* causes JavaScript code, running in the Web browser, to send a command to IGB running on the desktop. The command includes parameters that tell IGB where to find the data on the CyVerse site along with metadata parameters including genome version and style information.

BioViz *Connect* metadata

The Terrain API represents metadata items as three values - Attribute, Value, and Unit. BioViz *Connect* uses a custom, IGB-specific Unit to retrieve and display IGB-relevant metadata for files and folders. The metadata item's Attribute attaches meaning to what the metadata contains; examples used in BioViz Connect include genome assembly version, visual style information such as foreground color and background color, or free text comments on the data provided by the user. The metadata item's Value is data appropriate for the Attribute, such as an IGB formatted genome identifier value for the genome association attribute.

The Integrated Genome Browser implements a system of genome assembly version names that allow users to view data files generated using one of these genome assemblies alongside reference sequence and gene model annotations. Reference data and gene model annotations are maintained on Web-accessible sites, called "IGB Quickload sites," which IGB accesses when it first launches. However, visualizing data files in the context of the correct reference genome assembly requires that the user knows in advance which assembly the data file belongs to. Some file formats contain this information, but many do not. To help users correctly match data files with genome versions, BioViz *Connect* uses a list of IGB-formatted genome identifiers hosted on the main IGB Quickload site to populate a menu of genome versions within the BioViz Connect IGB metadata interface. The genome version and other metadata associated with files are passed to the IGB REST endpoint when a user clicks "View in IGB". The metadata are parsed and then used by IGB to overlay the data onto the correct reference assembly and to specify how the file's data will appear.

Public data visualization

The flow of data from CyVerse into IGB depends on data files having publicly accessible URLs hosted on servers that IGB can use to retrieve data. It also depends on the server software supporting HTTP byte range requests, which allow client software to request arbitrarily small or large sections using byte addresses and thus avoid downloading entire files during the visualization session. Similar to sharing in commercial cloud storage products, users can choose to make individual files public or private through the “Manage Link” right-panel display (Fig. 2). Public links are viewable to anyone with the link, including non-CyVerse users. As a convenience for users, clicking the "View in IGB" button on private files triggers a dialog for creating the public URL for the data set.

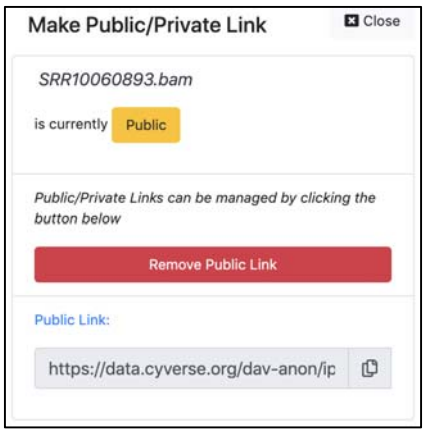


Fig. 2 Right-panel showing Public/Private options for a file. The file is currently set as public, with a public link displayed at the bottom.

BioViz *Connect* deployment

The version of BioViz *Connect* that is current as of this writing is managed using ansible roles and playbooks publicly available in a git repository from https://bitbucket.org/lorainelab/bioviz-connect-playbooks. The playbooks contain two sets of tasks. One set of tasks creates a virtual machine using the Amazon EC2 Web service. Once this host is set up and running, a second set of ansible tasks installs and configures software on the host, including an apache2 Web server, a MySQL database, and the BioViz *Connect* code base. Playbook users can specify the BioViz Connect repository and branch they wish to deploy, which facilitates rapid testing of proposed new code. During the provisioning process, a call is made to a Terrain endpoint that provides a list of all CyVerse asynchronous analysis apps that can produce output visible to IGB. These data are then used to construct the "analysis" sections of the user interface.

BioViz *Connect* interface for running visual analysis Apps

When users right-click a file name listing within BioViz *Connect*, a context menu appears containing an option labeled "Analyse." Selecting this option causes BioViz *Connect* to display a curated subset of published CyVerse Apps that accept the selected file type as input and

produce output relevant to visualization in IGB. Information about compatible Apps, the file types they can accept, and App parameters are stored in a relational database on the BioViz *Connect* host, but these data are taken directly from a Terrain REST endpoint during BioViz *Connect* deployment, as mentioned in the preceding section. When a user selects "Analyse", BioViz *Connect* queries this relational database to determine which Apps can accept the file as input and should be displayed to the user. Users can then run the App from the BioViz *Connect* interface. The BioViz *Connect* team controls which Apps appear by adding them to the IGB Community within the CyVerse Discovery Environment. Thus, as new capabilities are added to IGB, BioViz *Connect* maintainers can create new or repurpose existing CyVerse Apps to supply data for IGB.

RNA-Seq data

RNA-Seq data presented in the use case scenario are from Sequence Read Archive Bioproject PRJNA509437 [17]. The data are available in the Community folder of publicly accessible datasets, represented as a folder in the left-side panel of the BioViz *Connect* display. Files available include RNA-Seq alignment files in BAM format, together with BAM index files. BAM files were generated by aligning sequence reads onto the June 2009 release of the *Arabidopsis thaliana* genome. RNA-Seq data were aligned to the Arabidopsis June 2009 reference genome assembly using the tophat2 program [18].

Results

Example use case

To demonstrate BioViz *Connect*, we describe an example use case scenario illustrating how a user can interact with BioViz *Connect* to explore and analyze an RNA-Seq dataset. The use

case focuses on two main tasks: visually checking data quality, followed by visual analysis and confirmation of differential gene expression across sample types.

Signing in and navigating within BioViz *Connect*

To start a session with BioViz *Connect*, a user visits https://bioviz.org/connect.html and selects the link labeled "Sign in with your CyVerse ID". Clicking this link redirects the user to the CyVerse hosted Central Authentication Service where the user enters their CyVerse username and password. Once the user has logged in at CyVerse, the Web browser displays the BioViz *Connect* user interface, which consists of a browsable, sortable, paginated view of the user's CyVerse home directory and its contents (Fig. 3A).

This view of files and data resembles the interface for commercial, consumer-focused cloud storage systems, a design choice meant to leverage users' familiarity with Google Drive, the Dropbox Web interface, and others. As with these other systems, single-clicking a file or folder selects it, and double-clicking a folder opens it and displays the contents. A bread crumb display at the top of the page shows the path from the root folder to the currently opened folder, and a copy icon next to the breadcrumb allows the user to copy folder name and path. By design, the browser's URL bar reflects the currently opened folder's location, and users can use the browser's forward, back, and bookmarking functions to navigate. This design makes the interface feel more polished and user-friendly by ensuring that every user-facing detail, including the URL, mimic and reinforce how the user has organized their data within the CyVerse virtual file system.

The top part of every BioViz *Connect* page also features a search bar that can be used to find and list files and folders matching a user-entered query string. Files and folders with names containing the query string are returned in a list view, in which users can sort by name, size, or date modified. Only files for which the user has read access are returned, an attribute controlled by the underlying Terrain API. On the left side of every page, BioViz *Connect* displays

icons representing shortcut links to the user's home directory, a publicly available community data folder, and other destinations of note. The "Community" folder contains data published for all CyVerse users, including the example RNA-Seq data set for the use case scenario described in the next section.

Viewing data in Integrated Genome Browser

To demonstrate BioViz *Connect* functionality, we describe the following use case scenario in which a hypothetical analyst visually analyzes data from an example RNA-Seq experiment. The example experiment investigates how heat and desiccation stresses affected gene expression in young *Arabidopsis thaliana* plants. In this experiment, young Arabidopsis plants underwent either a 3-hour, non-lethal heat stress or multi-day desiccation stress. Two post-treatment sample time points were collected for treated plants and their untreated control counterparts, with two to four replicates per sample type and 23 samples in total. This is a moderately complex experimental design, with several possible comparisons that could be done. Data from the experiment are available in the Short Read Archive with project accession SRP220157. Sample libraries were sequenced in single runs of the Illumina platform and are identified by their run identifiers in the text below. Data files discussed below are available in the Community folder in file path "BioViz / rnaseq / A_thaliana_Jun_2009 / SRP220157 / reads."

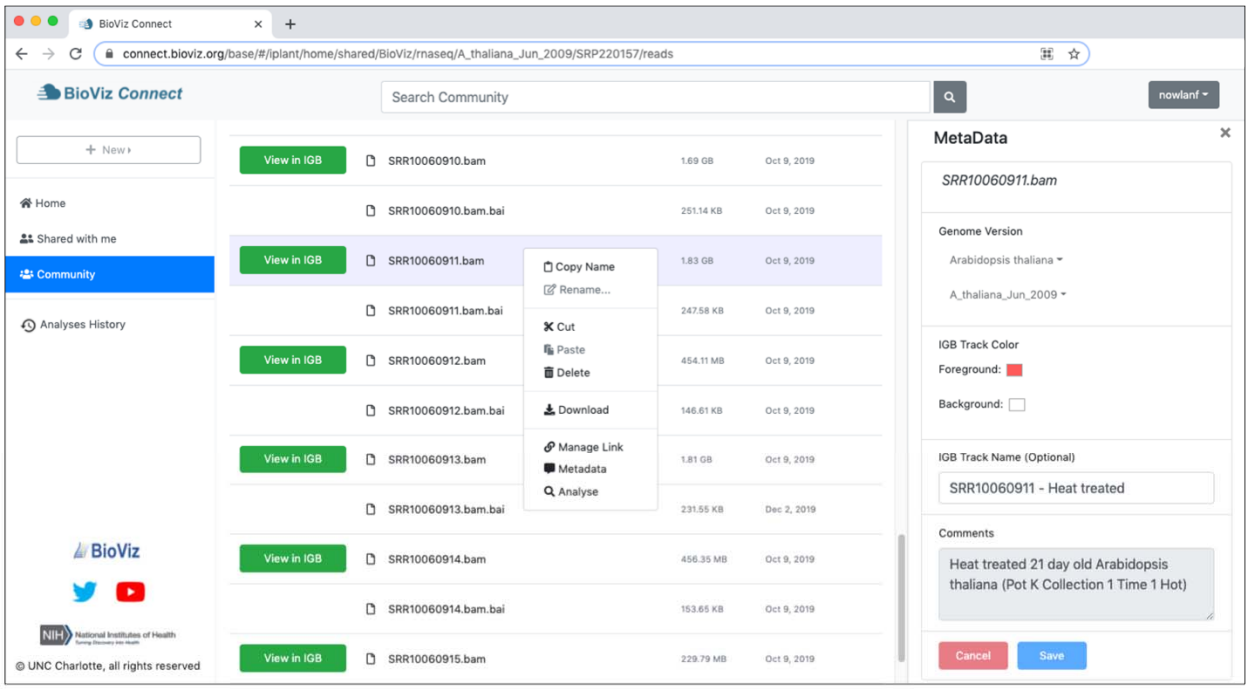
One of the first tasks an analyst would perform when working with such a dataset is to check alignment quality. BioViz Connect makes it easy to perform this task visually. To illustrate this process, we discuss RNA-Seq alignment files SRR10060893.bam and SRR10060894.bam, which are replicate control samples from time point one of the heat stress treatment. The BioViz *Connect* middle panel view shows that SRR10060893.bam has size 1.61 GB, about twice the size of SRR10060894.bam, which is 0.669 GB. According to their corresponding records in the Sequence Read Archive, the two samples were sequenced to about the same depth, and so their resulting alignment files ought to have similar sizes. The fact that the alignments files have

different sizes suggests there could be a problem with one or both samples. For example, the alignment procedure may have failed to finish, resulting in a smaller output file for sample SRR10060894. Another possibility is that library synthesis failed in some way, producing sequences that could not be aligned. There are many possible explanations, and visualization can help the analyst identify the cause of the discrepancy and determine how to handle it.

To visualize the alignments, the user first launches Integrated Genome Browser, installed on their local computer. With IGB started, the user next "opens" each data file into a new track within the IGB display by clicking each file's "View in IGB" button displayed in the BioViz *Connect* Web page. In response, IGB opens the genome version associated with the files and adds the file as a new track to the display, using graphical style information from each file's metadata to configure track appearance. Because the speed with which the file is added to IGB depends on internet speeds and can vary from file to file, the BioViz *Connect* Web page provides feedback to the user letting them know their action has succeeded.

Prior work with dataset SRP220157 and other similar datasets found that the gene SR45a, encoding an RNA-binding protein, is up-regulated by heat and desiccation stresses and so can provide a useful internal control for gene expression regulation under heat and desiccation stress [19, 20]. The user navigates to the control gene's location using IGB's simple or advanced search interfaces. To load the alignments into IGB, the user next clicks the "Load Data" button at the top right of the IGB window. Once the data are loaded into the display, a problem with SRR10060894 becomes apparent (Fig. 3B). The alignments for this sample appear in vertical stacks covering approximately 30% of the gene's exons. By contrast, the alignments for the other sample fully cover the gene body and include many spliced reads split across introns. The sparser pattern observed in SRR10060894 typically arises when the library synthesis process included too many polymerase chain reaction amplification cycles, which reduces the diversity of resulting sequence data. This pattern indicates that the user should exclude SRR10060894 from further analysis, but the other file appears to be fine.

A



B

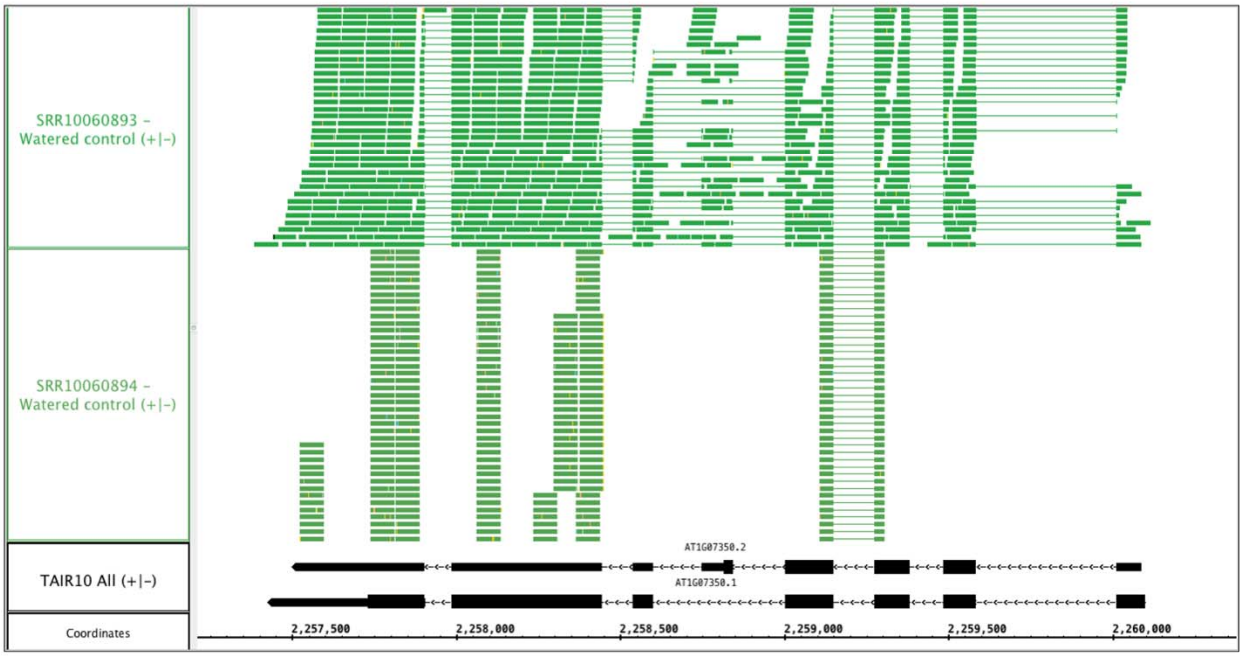


Fig. 3 Example BioViz *Connect* main page and data visualization. (A) BioViz *Connect* main page. The left panel shows shortcuts to home, shared, community folders. The middle panel

lists files and folders. The right panel shows the selected file's metadata. (B) SRR10060893.bam and SRR10060894.bam files viewed in IGB overlapping the SR45a gene of *Arabidopsis thaliana*.

IGB visual analytics to compare gene expression in RNA-Seq data

Repeating the preceding process with other samples in the dataset, the user identifies another potentially problematic pair of files. The alignments file SRR10060911.bam is 1.83 Gb, but SRR10060912.bam is only 0.454 Gb. Opening and viewing the alignment files in IGB, the user discovers that each contains diverse pattern of alignments, but one file appears to contain more data than the other (Fig. 4A). By clicking-dragging the mouse over each track within the IGB display, the user can select and count the number of alignments per sample. Sample SRR10060912 contains 1,925 alignments, and sample SRR10060911 produced 10,867 alignments, nearly five times as many. By operating the vertical zoom control with IGB, the user can interactively stretch the display vertically to reveal more or less detail about the alignments (Fig. 4B and 4C). From this, the user can tentatively conclude that the overall diversity of alignments is similar, but the depth of sequencing was greater in SRR1006911. To confirm the finding, the user applies a visual analytics function (called a "Track Operation" within IGB) that creates coverage or depth graphs using data from the read alignment tracks (Fig. 4D). To make a coverage graph, the user right-clicks a track label for a read alignment track and chooses option "Track Operations > Depth Graph (All)". This generates a new track showing a graph in which the y-axis indicates the number of sequences aligned per position indicated on the x-axis, corresponding to base pair positions in the genome. After modifying the y-axis lower and upper boundary values (using controls in IGB's Graph tab), the user again can observe that the pattern of alignments is similar between the two samples, but the overall level of sequencing was different. Thus, the file size difference most likely reflects different amounts of sequencing rather than a problem with the library synthesis, as was the case in the previous example.

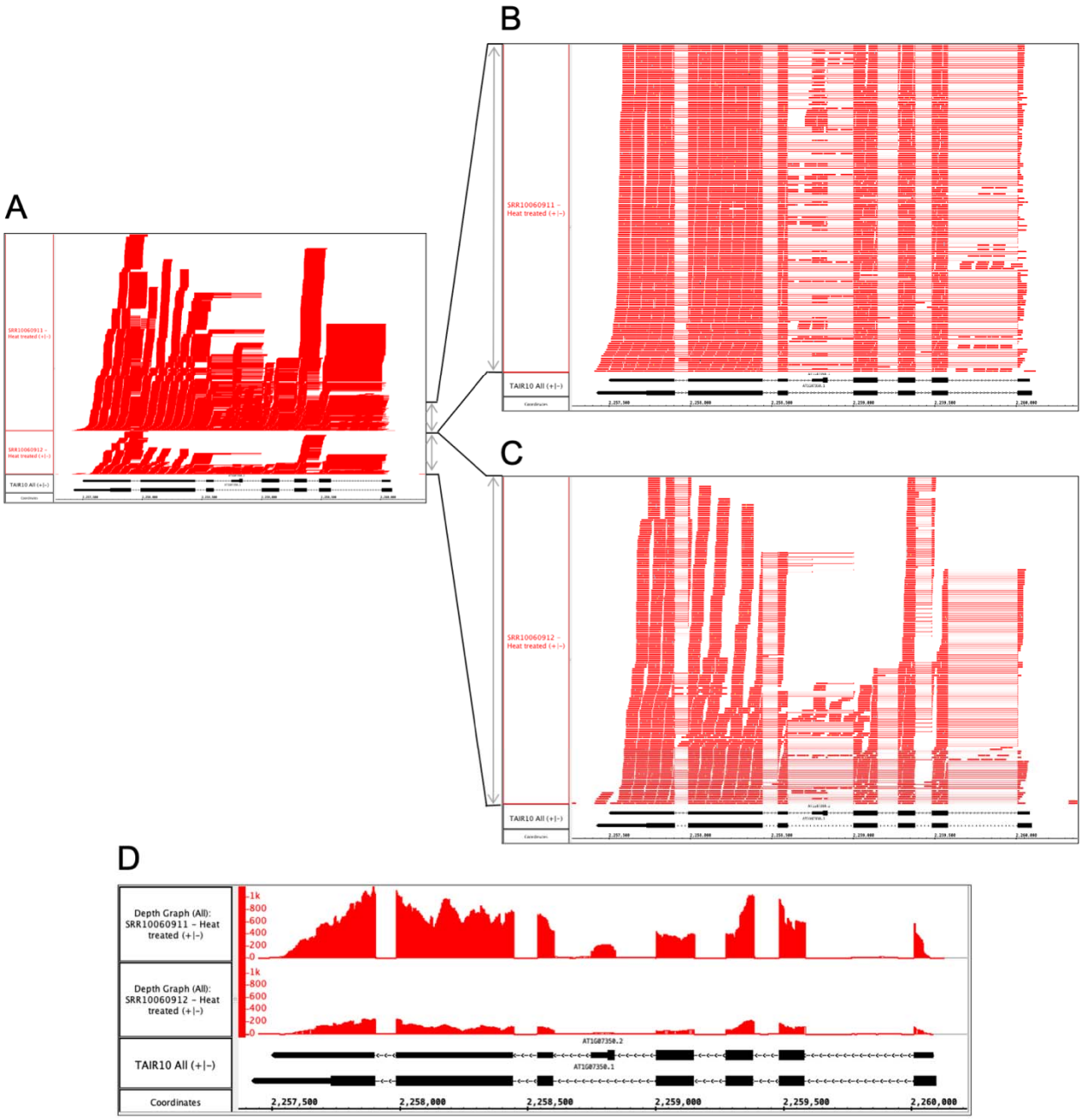


Fig. 4 Heat treated samples viewed in IGB. (A) Vertical dimension is compressed to show all alignments. (B) SR10060912 and (C) SR10060911 tracks stretched vertically to reveal alignment patterns in more detail. (D) Alignment coverage graphs calculated within IGB using alignments from A. The y-axis values represent the number of aligned sequences per base pair

position indicated on the coordinates track. The track labeled TAIR10 mRNA shows SR45a gene models AT1G07350.1 and AT1G07350.2.

Analyze data using CyVerse resources via BioViz *Connect*

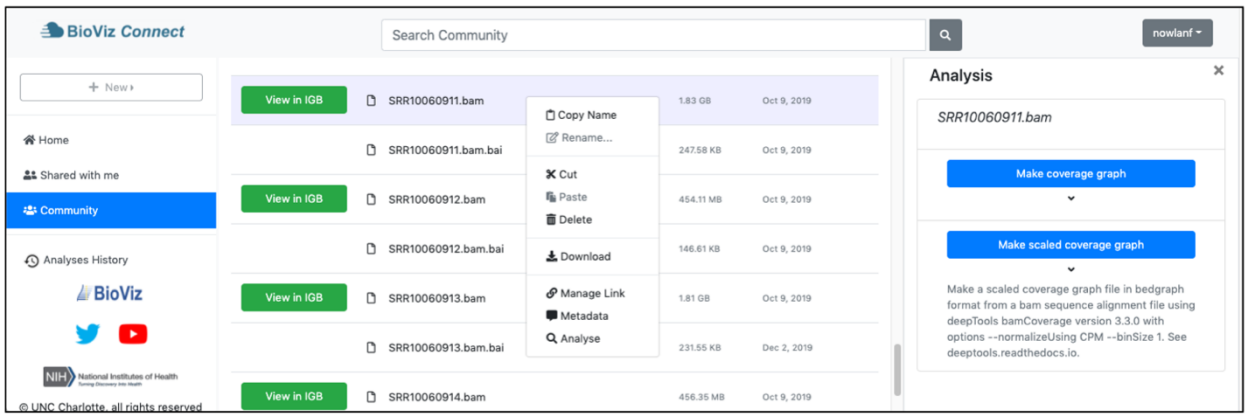
Summarizing read alignments with coverage graphs can provide a fast way to compare expression across sample types, but only if the libraries were sequenced to approximately the same depth. To properly use coverage graphs to assess expression, the user needs to scale or normalize the graphs to account for differences in sequencing depth. Performing this transformation within IGB itself is impractical, however, as it would require downloading, reading, and processing the entire bam-format alignments file. A better approach is to off-load computationally intensive visual analytics tasks to CyVerse cloud computing resources. To demonstrate the value of this strategy, we deployed the deepTools genomeCoverage command line tool from the deepTools suite as a new IGB-friendly CyVerse App [6].

To create a scaled coverage graph, the user returns to BioViz *Connect*, right-clicks a bam format file, and chooses "Analyse." This opens the Analysis right-panel display, which lists CyVerse Apps that can accept the selected file type as input (Fig. 5A). Selecting "Make scaled coverage graph" opens a form with options for creating the graph using the genomeCoverage algorithm (Fig. 5B). The interface includes a place for the user to enter names for the analysis and also for the output file that will be produced. After filling out the form as required, the user then clicks Run Analysis button, which submits a job request to run the App via the CyVerse analysis API, using CyVerse computing resources. The job runs asynchronously, and the user can check its status by looking it up in the Analyses History (Fig. 5C). Analyses are listed as Queued, Running, Failed, or Completed. The CyVerse infrastructure sends an email once the analysis has finished. When the job completes, its output appears in the analyses folder in the

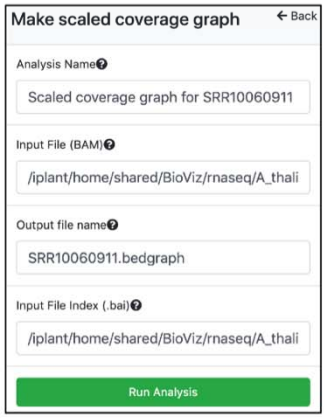
user's home directory. Clicking the analysis name in the Analyses History opens a shortcut to the folder where the output data are stored.

Figure 5D shows scaled coverage graphs created for the RNA-Seq samples described above. Examining the scaled coverage graphs for SR45a shows that both heat-treated samples had similar expression levels post-normalization (Fig. 5D). This indicates that the difference observed in the unscaled coverage graph was likely an artifact of lower sequencing depth rather than depressed expression of the gene in one replicate versus the other. Figure 5D also shows a scaled coverage graph from a control sample that received no heat stress treatment. The control sample's scaled coverage graph exhibits approximately 10-fold lower expression (measured as y-axis values) across the spliced regions of the gene, visually demonstrating that the heat stress treatment increased SR45a transcript abundance during stress.

A



B



C

Name	App Name	Start Date	End Date	Status
Scaled coverage graph for SRR10...	Make scaled coverage graph	05-01-2020 19:07:57	null	Submitted
Scaled coverage graph for SRR10...	Make scaled coverage graph	05-01-2020 19:07:11	null	Running
Scaled Coverage Graph for SRR1...	Create Scaled Coverage Graph	04-28-2020 19:47:26	04-28-2020 19:58:40	Completed
Scaled Coverage Graph for SRR1...	Create Scaled Coverage Graph	04-28-2020 19:46:28	04-28-2020 19:58:40	Completed
Not public read access		10-16-2019 20:55:51	10-16-2019 21:13:28	Completed
Not Public Read Access 3	FastQC 0.11.5 (multi-file)	10-16-2019 20:58:34	10-16-2019 21:00:19	Completed
Not Public Read Access 2	FastQC 0.11.5 (multi-file)	10-16-2019 20:56:20	10-16-2019 20:57:39	Failed
PK_Unzip_analysis1	PK Unzip	02-25-2015 18:53:57	02-25-2015 19:03:48	Completed
Uncompress_with_bunzip2_analys...	Uncompress with bunzip2	02-25-2015 18:50:52	02-25-2015 18:52:15	Failed
Uncompress_files_with_gunzip_an...	Uncompress files with gunzip	02-25-2015 18:47:24	02-25-2015 18:48:24	Failed
Cuffmerge2_analysis1	Cuffmerge2	01-30-2015 14:33:32	01-30-2015 14:40:08	Failed
Cufflinks2_analysis1	Cufflinks2	01-29-2015 20:28:54	01-29-2015 21:23:53	Completed
TopHat2-SE_analysis1	TopHat2-SE	01-29-2015 18:14:44	01-29-2015 19:23:06	Failed

D

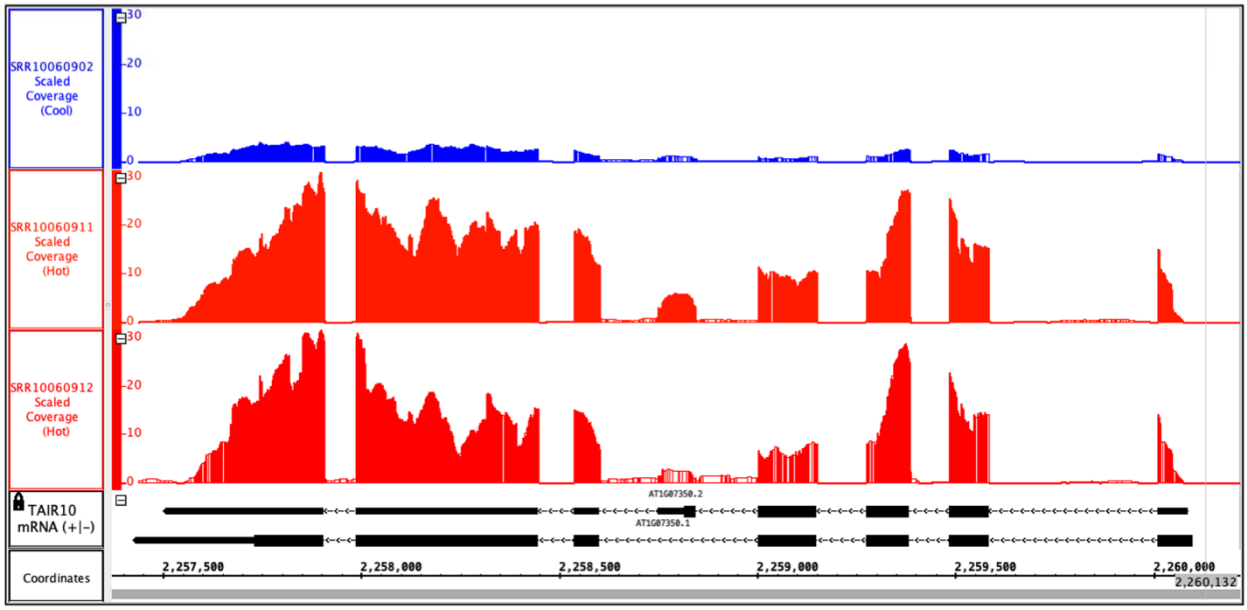


Fig. 5 Example analysis in BioViz *Connect* with output visualized in IGB. (A) BioViz *Connect* main page with analysis right panel open. (B) Scaled coverage graph analysis options for naming the analysis, selecting input file, output file name, and index file selection. (C) Analyses History showing the status of current and previous jobs. (D) SRR10060902 (control), SRR10060911 (heat treated), and SRR10060912 (heat treated) scaled bedgraph files viewed in IGB overlapping the SR45a gene of *Arabidopsis thaliana*.

Conclusions

BioViz *Connect* introduces and demonstrates several innovations in the field of science gateway development and research, while providing useful functionality for researchers seeking to understand and visualize genomic data. First, BioViz *Connect* enables users of the CyVerse science gateway to visualize genomic data files from their CyVerse accounts using Integrated Genome Browser, a desktop application. To our knowledge, BioViz *Connect* is the first and only resource that enables CyVerse users to visualize data in a genome browser or other desktop tool. Second, we implemented BioViz *Connect* using the CyVerse Terrain API, a collection of REST endpoints that form a comprehensive computational interface to CyVerse resources. BioViz *Connect* is the first application developed using the Terrain API by a group outside the CyVerse development team and using an entirely different visual interface. BioViz *Connect* demonstrates to the larger community of biologists, developers, and funders that modern, feature-rich REST interfaces to powerful computational resources stimulate and enable innovation and progress. Third, BioViz *Connect* introduces a new breed of science gateway specialized for visualization on the desktop. Rather than re-implement visualization tooling within the gateway, we instead created a new kind of gateway that manages and coordinates the flow of data into an-already mature and robust desktop application with a pre-existing userbase. Fourth, BioViz Connect demonstrates using powerful remote resources to perform

compute-intensive visual analytics functions, as in the genome-wide coverage graph example described in preceding sections of the paper. This example is straightforward and uses a pre-existing algorithm, but it shows how developers can harness a more powerful gateway system to develop and deploy all-new visualizations of genomic data.

Availability and requirements

Project name: BioViz *Connect*

Project home page: https://bitbucket.org/lorainelab/bioviz-connect

Operating system(s): Linux-based systems for the BioViz *Connect* server. BioViz Connect web app tested on Ubuntu, Windows 10, MacOS Mojave operating systems and Chrome, Firefox, and Edge web browsers.

Programming language: HTML, JavaScript, CSS, and python3

Other requirements: A free CyVerse account is required to login: https://user.cyverse.org/register. Integrated Genome Browser 9.1.4 or greater required for visualizing data using BioViz *Connect*.

License: Common Public License Version 1.0

Any restrictions to use by non-academics: No

Abbreviations

API: Application Programming Interface; CSS: Cascading Style Sheets; HTML: HyperText Markup Language; IGB: Integrated Genome Browser; REST: REpresentational State Transfer

Declarations

Availability of data and Materials

RNA-Seq data files are available from https://data.cyverse.org/dav-anon/iplant/home/shared/BioViz/rnaseq.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing Interests

The authors declare that they have no competing interests.

Funding

Research reported in this publication was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number 5R01GM121927-02. Funding was used to plan, design, and develop the software reported in the article.

Authors' Contributions

NHF and AEL conceived of and supervised the project. KR, CK, ST, and PB planned and developed BioViz *Connect*. NHF, AEL, KR, CK, ST, PB tested and debugged BioViz *Connect*. NHF, KR, CK, and AEL wrote the draft manuscript. All authors read and approved the final manuscript.

Acknowledgements

We thank Paul Sarando, Sarah Roberts, Sriram Srinivasan, Ian McEwen, Ramona Walls, and Reetu Tuteja for their assistance with the Terrain API and publishing CyVerse apps, which was made possible through CyVerse's External Collaborative Partnership program.

REFERENCES CITED

1. Wilkins-Diehr N, Gannon D, Klimeck G, Oster S, Pamidighantam S: **TeraGrid Science Gateways and Their Impact on Science**. *Computer* 2008, **41**(11):32-41.
2. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J *et al*: **Galaxy: a platform for interactive large-scale genome analysis**. *Genome research* 2005, **15**(10):1451-1455.
3. Goff SA, Vaughn M, McKay S, Lyons E, Stapleton AE, Gessler D, Matasci N, Wang L, Hanlon M, Lenards A *et al*: **The iPlant Collaborative: Cyberinfrastructure for Plant Biology**. *Front Plant Sci* 2011, **2**:34-34.
4. Merchant N, Lyons E, Goff S, Vaughn M, Ware D, Micklos D, Antin P: **The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences**. *PLoS Biol* 2016, **14**(1):e1002342-e1002342.
5. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R: **The Sequence Alignment/Map format and SAMtools**. *Bioinformatics* 2009, **25**(16):2078-2079.
6. Ramirez F, Ryan DP, Gruning B, Bhardwaj V, Kilpert F, Richter AS, Heyne S, Dundar F, Manke T: **deepTools2: a next generation web server for deep-sequencing data analysis**. *Nucleic Acids Res* 2016, **44**(W1):W160-165.
7. Bailey TL: **DREME: motif discovery in transcription factor ChIP-seq data**. *Bioinformatics* 2011, **27**(12):1653-1659.
8. Raney BJ, Dreszer TR, Barber GP, Clawson H, Fujita PA, Wang T, Nguyen N, Paten B, Zweig AS, Karolchik D *et al*: **Track data hubs enable visualization of user-defined genome-wide annotations on the UCSC Genome Browser**. *Bioinformatics* 2014, **30**(7):1003-1005.
9. Buels R, Yao E, Diesh CM, Hayes RD, Munoz-Torres M, Helt G, Goodstein DM, Elsik CG, Lewis SE, Stein L *et al*: **JBrowse: a dynamic web platform for genome visualization and analysis**. *Genome biology* 2016, **17**(1):66.
10. Robinson JT, Thorvaldsdottir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP: **Integrative genomics viewer**. *Nat Biotechnol* 2011, **29**(1):24-26.
11. Devisetty UK, Kennedy K, Sarando P, Merchant N, Lyons E: **Bringing your tools to CyVerse Discovery Environment using Docker**. *F1000Res* 2016, **5**:1442.
12. Nicol JW, Helt GA, Blanchard SG, Jr., Raja A, Loraine AE: **The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets**. *Bioinformatics* 2009, **25**(20):2730-2731.

13. Gulledge AA, Vora H, Patel K, Loraine AE: **A protocol for visual analysis of alternative splicing in RNA-Seq data using integrated genome browser**. *Methods Mol Biol* 2014, **1158**:123-137.
14. Loraine AE, Blakley IC, Jagadeesan S, Harper J, Miller G, Firon N: **Analysis and visualization of RNA-Seq expression data using RStudio, Bioconductor, and Integrated Genome Browser**. *Methods Mol Biol* 2015, **1284**:481-501.
15. Freese NH, Norris DC, Loraine AE: **Integrated genome browser: visual analytics platform for genomics**. *Bioinformatics* 2016, **32**(14):2089-2095.
16. Mall T, Eckstein J, Norris D, Vora H, Freese NH, Loraine AE: **ProtAnnot: an App for Integrated Genome Browser to display how alternative splicing and transcription affect proteins**. *Bioinformatics* 2016, **32**(16):2499-2501.
17. Leinonen R, Sugawara H, Shumway M: **The sequence read archive**. *Nucleic Acids Res* 2011, **39**(Database issue):D19-21.
18. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL: **TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions**. *Genome biology* 2013, **14**(4):R36.
19. Gulledge AA, Roberts AD, Vora H, Patel K, Loraine AE: **Mining Arabidopsis thaliana RNA-seq data with Integrated Genome Browser reveals stress-induced alternative splicing of the putative splicing regulator SR45a**. *American journal of botany* 2012, **99**(2):219-231.
20. Yoshimura K, Mori T, Yokoyama K, Koike Y, Tanabe N, Sato N, Takahashi H, Maruta T, Shigeoka S: **Identification of alternative splicing events regulated by an Arabidopsis serine/arginine-like protein, atSR45a, in response to high-light stress using a tiling array**. *Plant & cell physiology* 2011, **52**(10):1786-1805.