

1 Management, Analyses, and Distribution of the MaizeCODE Data on the Cloud

2
3 Liya Wang¹, Zhenyuan Lu¹, Melissa delaBastide¹, Peter Van Buren¹, Xiaofei Wang¹, Cornel
4 Ghiban¹, Michael Regulski¹, Jorg Drenkow¹, Xiaosa Xu¹, Carlos Ortiz-Ramirez², Cristina
5 Fernandez-Marco¹, Sara Goodwin¹, Alexander Dobin¹, Kenneth D. Birnbaum², David P.
6 Jackson¹, Robert A. Martienssen¹, William R. McCombie¹, David A. Micklos¹, Michael C.
7 Schatz^{1,3}, Doreen H. Ware^{1,4,*}, Thomas R. Gingeras^{1,*}

8
9 ¹Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, ²New York University, New York, NY,
10 ³Johns Hopkins University, Baltimore, MD. ⁴USDA-ARS Robert W. Holley Center for Agriculture
11 and Health, Ithaca, NY, United States

12
13 MaizeCODE is a project aimed at identifying and analyzing functional elements in the maize
14 genome. In its initial phase, MaizeCODE assayed up to five tissues from four maize strains
15 (B73, NC350, W22, TIL11) by RNA-Seq, Chip-Seq, RAMPAGE, and small RNA sequencing. To
16 facilitate reproducible science and provide both human and machine access to the MaizeCODE
17 data, we enhanced SciApps, a cloud-based portal, for analysis and distribution of both raw data
18 and analysis results. Based on the SciApps workflow platform, we generated new components
19 to support the complete cycle of MaizeCODE data management. These include publicly
20 accessible scientific workflows for the reproducible and shareable analysis of various functional
21 data, a RESTful API for batch processing and distribution of data and metadata, a searchable
22 data page that lists each MaizeCODE experiment as a reproducible workflow, and integrated
23 JBrowse genome browser tracks linked with workflows and metadata. The SciApps portal is a
24 flexible platform that allows the integration of new analysis tools, workflows, and genomic data
25 from multiple projects. Through metadata and a ready-to-compute cloud-based platform, the
26 portal experience improves access to the MaizeCODE data and facilitates its analysis.

27
28 **Keywords:** bioinformatics, functional annotations, cloud computing, ENCODE, workflows

1 INTRODUCTION

2

3 Maize is one of the most biologically, socially, and economically important crop plants. Following
4 the sequencing of its genome (Jiao et al., 2017; Schnable et al., 2009), the next critical step in
5 understanding maize biology will involve identifying and deciphering functional sequence
6 regions. Modeled on the Encyclopedia of DNA Elements (ENCODE) Project for the human
7 genome (ENCODE Consortium, 2004), the MaizeCODE project is an integrated and multi-
8 disciplinary project aimed at revealing the functional regions of the maize genome by identifying
9 loci that are transcribed, methylated, or bound by specific modified histones and transcription
10 factors in various tissues. In addition, MaizeCODE is designed to store, collate, display, and
11 disseminate the data to the wider community of plant biologists worldwide.

12

13 To curate, process, and distribute the ENCODE data, the ENCODE Data Coordination Center
14 (DCC) group established the ENCODE portal (Sloan et al., 2016), which relies on both rich
15 metadata and commercial cloud resources through the DNAnexus platform
16 (<https://www.dnanexus.com>). Within this platform, standard processing pipelines for human
17 genome analysis are constructed to ensure consistent and reproducible processing of primary
18 sequence data. However, both the ENCODE DCC and end users are required to cover the cost
19 of the DNAnexus service and commercial cloud resources. In order to provide a cost-free data
20 processing platform for academic users, the MaizeCODE DCC group decided to leverage two
21 NSF-funded resources, XSEDE (Towns et al., 2014) at the Texas Advanced Computing Center
22 (TACC) for computing power and the CyVerse Data Store (Goff et al., 2011) for cloud-based
23 data storage.

24

25 To automate bioinformatics analysis over both the XSEDE/TACC cloud and CyVerse Data
26 Store, we developed a bioinformatics workflow platform called SciApps (Wang et al., 2018). In

1 the work described here, we further improved SciApps by adding a RESTful API for automating
2 batch processing of the MaizeCODE data and metadata management, a searchable
3 MaizeCODE data page powered by a relational database, several analysis workflows, and
4 Genome Browser tracks automatically generated from unique workflow identifiers via the
5 RESTful API. The SciApps platform (<https://sciapps.org>) has been used to support both
6 MaizeCODE DCC and end users to process/reprocess and manage multi-omics data through
7 either the GUI or the API.

8

9 **METHODS**

10

11 **Overview of the entire MaizeCODE data management cycle**

12

13 To improve accessibility, reproducibility, reusability, and interoperability, data generated by the
14 MaizeCODE Consortium members are uploaded to a cloud-based data storage system, the
15 CyVerse Data Store (Goff et al., 2011). The Data Store, which is built on top of iRODS (a rule-
16 oriented data system) ([Moore and Rajsekar 2010](#)), supports data virtualization, sharing, bulk
17 uploads/downloads, and collaborations. Once uploaded, the experimental metadata are
18 attached to raw data files to facilitate the reuse of data, as well as submission of data to the
19 NCBI Short Read Archive (SRA). The metadata is later retrieved via the Terrain API
20 (<https://github.com/cyverse-de/terrain>) to automate batch analyses via SciApps (Wang et al.,
21 2018). SciApps provides a ready-to-compute cloud-based platform for automating complex
22 analyses constructed using modular applications (or apps). Previously, SciApps was operated
23 via a web GUI, but since that time we have developed SciApps RESTful APIs to support batch
24 processing of MaizeCODE and other data. In addition, we have integrated over 20 new apps for
25 ground-level analyses such as quality control (QC), alignment to the reference genome, filtering,
26 quantification (e.g. for gene expression), and peak calling (if needed). Replicates (and controls if

1 available) of each assay are organized as a single experiment (or workflow with a unique ID),
2 which represents an entity that chains together raw data, analysis results, experimental
3 metadata (such as tissue, assay, etc), and computational provenance (computational metadata
4 supporting the reliable replication of scientific results, such as version of the software tools,
5 parameters used for the analysis, etc). SciApps extracts experimental metadata and attaches
6 them to a specific workflow so that users can access them directly on the SciApps portal.
7 Genome browser tracks are automatically generated and displayed within an integrated version
8 of JBrowse (Skinner et al., 2009) by looping through the list of experiments/workflows via the
9 SciApps RESTful API.

10
11 In summary, the SciApps portal and RESTful API have been used to support the management,
12 analysis, and distribution of the MaizeCODE data. Through the automation of both data and
13 metadata management, the chances of human errors in data management are greatly reduced.

14

15 **Processing and accessing the MaizeCODE data with the SciApps RESTful API**

16

17 The cloud-based architecture of SciApps (Wang et al., 2018) enables highly scalable processing
18 of MaizeCODE data on the XSEDE/TACC cloud. Both intermediate and final results are
19 archived in the CyVerse Data Store, where the raw data are also hosted. As discussed above,
20 each SciApps workflow captures experimental metadata and computational provenance along
21 with the raw and processed data. Batch processing of MaizeCODE data is supported through
22 the RESTful API via a workflow endpoint that takes a template workflow (for example
23 commands, see Supplementary file page 3); the API endpoints are provided in **Table 1**.

24

25 The analysis workflow for a specific assay is typically built interactively within the SciApps GUI
26 using one data set as a template. Once the workflow is captured, it can then be easily and

1 automatically applied to analyze other genomes and tissues. Alternatively, users may also build
2 workflows entirely programmatically with a series of analysis job IDs via the API. Experimental
3 metadata are retrieved via the CyVerse Terrain API, and then attached to the workflow via the
4 SciApps API at runtime. The API supports the MaizeCODE DCC for automatically processing a
5 large amount of data and also supports retrieval of results and metadata by end users. For
6 example, genome browser tracks can be automatically generated given a workflow ID by the
7 following steps (**Figure S1**): 1. Retrieve job IDs and inputs with the workflow endpoint, given a
8 workflow ID; 2. Retrieve the output path with the job endpoint, given a job ID; 3. Construct the
9 browser-ready link with the retrieved information. To simplify the process, the MaizeCODE DCC
10 encodes the genome, tissue, and replicate information into the input raw data file path, which is
11 also accessible through the workflow metadata endpoints. SciApps also names the output
12 filename based on the input filename with the output ID (defined by the app) as the prefix. As
13 shown in **Figure S1**, once the input filename, input path, and output path to cloud storage are
14 retrieved by calling the API, the output file path can be constructed to build the browser links.

15
16 Given a workflow ID, users can also call the API to retrieve the computational metadata (e.g.,
17 <https://sciapps.org/workflow/a14ff622-7af9-4b1f-877a-2be926dc1059>) or the experimental
18 metadata (e.g., <https://sciapps.org/workflow/a14ff622-7af9-4b1f-877a-2be926dc1059/metadata>)
19 in standard JSON format and view them in any web browsers.

20

21 **Accessing the MaizeCODE experiments as reproducible workflows**

22 The MaizeCODE data page can be accessed under 'Data' from the top navigation bar of
23 SciApps (**Figure 1**). Keyword search is supported to allow the user to narrow down the list of
24 experiments to a specific genome or tissue or assay in real time. Once an experiment is
25 selected, the user can access the metadata, workflows, and ground-level analysis results of the
26 experiments, starting from raw sequence data. With the 'Relaunch' tab, user can reproduce the

1 entire analysis with one click or apply the same analysis workflow to new data. Using the ‘Share’
2 tab, the analysis can be shared with others. Users can load the results to the History panel and
3 subject them to further analysis using the modular apps. Because all results are archived in the
4 cloud, downstream analyses can be completed quickly, e.g., differential expression analysis
5 between two tissues can be completed in a few minutes, rather than hours when starting from
6 the raw sequence data.

7

8 **Accessing the MaizeCODE data as Genome Browser tracks**

9 Once the analysis is completed, genome browser tracks are automatically generated given the
10 workflow ID by calling the SciApps API for an integrated version of JBrowse. The browser tracks
11 can be accessed under the ‘Tools’ menu within the top navigation bar. As shown in **Figure 2**,
12 tracks are organized by genome, tissue, replicate, and assay. Checking the box next to each
13 track will load it into the browser. The SciApps workflow ID is embedded, so clicking on a track
14 brings up the workflow ‘Relaunch’ interface, which can be used to reproduce the track signal if
15 needed. In this interface, the user can also check the parameters used for the analysis, as well
16 as additional results in the History panel. At the bottom of the interface, a diagram button
17 visualizes the workflow diagram, and a metadata button displays the experimental metadata
18 associated with the workflow. From the results, user can also generate additional browser track
19 links through the visualization (eye) icon. For example, this can be used to verify the signal track
20 with the alignment files (in the BAM format). As mentioned earlier, the results can also be used
21 to perform a downstream analysis on the same interface. Finally, the browser tracks are
22 available as a JSON file for integration into other platforms (e.g., the JSON file for B73 is
23 available at https://data.sciapps.org/view2/data2/B73/v4/apollo_data/trackList.json).

24

1 Users can also locate a gene on the JBrowse by pasting the gene ID (e.g. Zm00001d02723)
2 into the address field, as shown in Figure S3.

3 **Accessing the raw reads on CyVerse Data Store**

4 The raw sequence data is deposited into the CyVerse Data Store via iCommands
5 (<https://docs.irods.org/4.2.1/icommands/user/>), with metadata attached before submission to the
6 NCBI SRA. From there, users can access the raw data in several ways. Within SciApps, the
7 input file node of the graphic diagram for a workflow/experiment is linked to the raw sequence
8 file. Clicking on the input node will open the CyVerse Data Common landing page in a web
9 browser. The metadata attached to the raw sequence file is also displayed on the same page.
10 The user can further navigate through all released raw data from the landing page
11 (<http://datacommons.cyverse.org/browse/iplant/home/shared/maizecode/released/>); the
12 SciApps workflow ID is attached as metadata to the raw data files if it has been processed. The
13 user can use the ID to load the workflow on the SciApps portal. For batch downloading of raw
14 sequence files through the GUI or the command line, we recommend CyberDuck
15 (<https://cyberduck.io/>) or iCommands, respectively.

16 **Analysis with reproducible workflows**

17 As previously described (Wang et al., 2018), bioinformatics applications (or apps) are
18 integrated into SciApps as modular components that can be chained with other apps into an
19 automated workflow. Individual apps are built with Singularity images (Kurtzer et al., 2017) from
20 BioConda recipes (Grüning et al., 2018) or directly from Dockerfiles to ensure reproducibility
21 across different cloud resources. To support the analysis of MaizeCODE data, over 20 software
22 tools are integrated. **Figure 3** shows two publicly accessible workflows for differential
23 expression analysis and cytosine methylation analysis, building on the popular STAR (Dobin et
24 al., 2013)/RSEM (Li and Dewey, 2011)/StringTie (Pertea et al., 2015)/ Ballgown(Frazer et al.,

1 2015) and Bismark (Krueger and Andrews, 2011) pipelines, respectively. These workflows can
2 be constructed either with the SciApps GUI or through the API. The user can retrieve the inputs,
3 metadata, results, and provenance of the software used in the analysis with a unique workflow
4 ID. The interactive graph, along with the platform guide ([https://cyverse-sciapps-
5 guide.readthedocs-hosted.com/en/latest/index.html](https://cyverse-sciapps-guide.readthedocs-hosted.com/en/latest/index.html)), helps users to understand how multiple
6 apps are used together to analyze a specific assay. For MaizeCODE data, the graph is also
7 helpful for visually inspecting the input–output relationship. Additionally, the user can check the
8 input data (through the input file node) and relaunch each individual step of the analysis, or
9 even the entire analysis, via the web interface or API.

10

11 **RESULTS AND DISCUSSION**

12 A large variety of software is needed to process the MaizeCODE data. For each experiment
13 (consisting of two replicates), a workflow with a unique ID is provided via the SciApps platform.
14 One major goal of SciApps is to empower anyone in the community to easily repeat an entire
15 analysis, or use a workflow with alternative parameters for each step if so desired. A second
16 major goal is to empower community members to process and combine their own comparable
17 data sets if they are generated with similar protocols used by the MaizeCode project, available
18 at <https://datacommons.cyverse.org/browse/iplant/home/shared/maizecode/released/protocols>.
19 In the following sections, we describe how RNA-seq data is processed, how the results can be
20 visualized, and how the primary analysis results can be used for differential expression analysis.

21 **Processing the RNA-seq data**

22 Besides the UCSC genome analysis tool bedGraphToBigWig (for format conversion and
23 generating browser track signals), the major software used in MaizeCODE RNA-seq data

1 analysis are bbdduk (<https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbdduk-guide/>)
2 for trimming low quality reads and adapter sequences, FastQC and MultiQC (Ewels et al., 2016)
3 for visually checking read quality, STAR (Dobin et al., 2013) for alignment, RSEM (Li and
4 Dewey, 2011) for quantifying gene expression, and StringTie (Pertea et al., 2015) for
5 transcriptome assembly. All tools are integrated into SciApps individually as separate apps, and
6 also combined as a single app, MCrna, for rapid batch processing of RNA-seq data without
7 requiring intermediate results to be transferred between the TACC and CyVerse cloud. **Figure 4**
8 shows the relationship among these analysis tools within the MCrna app, which is used to
9 process both replicates of an experiment in parallel.

10

11 For each replicate, the MultiQC software outputs a quality report for the sequence data, before
12 and after trimming, in an interactive HTML format. This report can be accessed via the
13 visualization (eye) icon in the History panel (next to each loaded replicate, as shown in **Figure**
14 **1**). As with the HTML format, text, image, and other web browser-compatible files can be
15 visualized by clicking the icon. For files that can be displayed on a Genome Browser (e.g., BAM,
16 bigwig, etc), the user can also generate browser-ready links by clicking the same icon. These
17 links address the cloud storage system from the CyVerse project, so they can be displayed on
18 Genome Browsers hosted by different portals. If the user clicks on the output file name (from the
19 History panel), they will be directed to the CyVerse Data Commons landing page, where they
20 can preview or download the results. For files over a few GBs, we recommend that the user
21 download their data using either iCommands or CyberDuck, using the CyVerse Data Store path
22 available in the file URL.

23 **Automated differential expression analysis**

1 One of the key advantages of distributing the MaizeCODE data through SciApps is that it
2 facilitates downstream analysis. In this section, we will show how differential expression
3 analysis can be performed, on either the gene or isoform levels, using the primary analysis
4 results discussed in the last section.

5

6 As shown in **Figure 5**, after loading the results for two experiments from the MaizeCODE data
7 page (**Figure 1**), one for ear tissue and the other for the root tissue, we can launch the
8 RSEM_de app from the 'Comparison' category (or through searching the app panel). For the
9 analysis, users drag and drop output files with names starting with 'rsem_' into the input field for
10 both replicates of each sample. The analysis job can then be submitted to the cloud for running,
11 and the results (i.e., the differentially expressed genes) will be available within a few minutes.
12 Note that the app is flexible in handling different numbers of replicates per sample. Additional
13 input fields can be added using the '+ Insert' button. For the MaizeCODE project, most data sets
14 are generated with two replicates.

15

16 Users can check the results through the History panel or the list of jobs (under the 'Workflow'
17 tab from the top menu). Users can also select jobs from the History panel and save them as
18 new workflows to organize the analysis and/or share it with others. Given that the XSEDE/TACC
19 cloud is a shared resource, and jobs may be queued for several minutes to several hours, we
20 have also established a local cluster (Wang et al., 2015) to quickly process small jobs requiring
21 less than an hour to complete. Powered by the Agave API (Dooley et al., 2012), SciApps treats
22 both the XSEDE/TACC cloud and the local cluster as virtual execution systems, allowing a
23 scientific app to be configured to run on either cloud. As such, by using the CyVerse Data Store
24 as a central storage hub, SciApps workflows can be seamlessly executed on a mixture of
25 XSEDE/TACC cloud and local clusters for efficient yet scalable processing and consumption of
26 MaizeCODE data.

1
2 To perform differential expression on the isoform level, users can launch the workflow diagram
3 shown in **Figure 3**. From the diagram and the relaunched app forms, the StringTie_Merge app
4 is used to merge assembled transcripts from all replicates to generate a new annotation file.
5 The annotation file is then passed again to the StringTie app to compute gene expression
6 quantification results, which are then input to the Ballgown app (Frazee et al., 2015) to compute
7 differentially expressed isoforms. Again, the user can drag and drop the alignment files and
8 assembled transcripts from the MaizeCODE primary analysis results without repeating the time-
9 consuming alignment and quantification steps. Given that all results are accessible through web
10 URLs, users can also retrieve the data directly to their local server for further interactive
11 analysis. For example, SciApps users can inspect the quantification results of each pair of
12 replicates to confirm that they are consistent, and if they are, proceed with the analysis of
13 differentially expressed genes.

14 **CONCLUSION AND FUTURE WORK**

15 SciApps is a cloud-based platform that provides a data management infrastructure for the
16 MaizeCODE data. The platform supports the management of experimental metadata and
17 computational provenance; provides a collection of analysis apps covering analysis of multi-
18 omics data sets; and provides both web browser and API access to data, results, metadata, and
19 computational provenance of software tools through a unique workflow ID. Genome browser
20 tracks are also provided to enable visualization of the results using an integrated version of
21 JBrowse.

22
23 SciApps has been designed to integrate cloud resources for scaling and long-term stability.
24 Currently, SciApps has been integrated with the NSF-funded XSEDE/TACC cloud and the
25 CyVerse Data Store to provide academic users with cost-free data storage and computing

1 services. Both resources are integrated as virtual systems via the Agave API, which also
2 supports the integration with commercial cloud platforms like Amazon EC2/S3 and Microsoft
3 Azure. Therefore, SciApps can be scaled for large-scale data management and analysis if
4 needed. Additionally, SciApps can also seamlessly integrate local data and computing
5 resources to complement cloud resources. The successful utilization of our local system
6 suggests that SciApps can facilitate collaborative projects across different institutes for joint data
7 production, analysis, and management with multiple local systems, thereby avoiding high
8 financial costs.

9
10 To process the data sets that are continually being generated by the MaizeCODE project,
11 several new analysis tools have been integrated into SciApps. Future goals include developing
12 new analysis workflows, supporting sophisticated queries against metadata, reanalyzing and
13 distributing published sequencing data sets from raw data, and conducting training and
14 community outreach.

15

16 **IMPLEMENTATION**

17 As previously described, the major components of the SciApps analysis portal include a web
18 browser user interface, a MySQL database, a workflow engine, and a newly developed web
19 API. The web API is written in Perl to complement the web browser user interface, especially
20 for batch processing and metadata handling. Analysis jobs are submitted to the cloud, and in
21 addition all released raw data, processed results, metadata, and workflows, are made publicly
22 accessible through both the GUI and API with no authentication needed.

23

24 To support MaizeCODE data management and enhance SciApps functionalities, several
25 searchable pages are added, including the user workflow page, user job page, and MaizeCODE

1 data page. On these pages, users can relaunch the workflow/job, visualize the workflow
2 diagram, load the job history and results into the History panel, share the workflow with others,
3 and check the metadata associated with the workflow. The RESTful API is designed to facilitate
4 batch processing of the MaizeCODE data through template workflows. More details and other
5 updates to the SciApps platform are described in the Supplementary file section 1 (page 2). The
6 overall cycle of MaizeCODE data management and processing is described in Figure S2.

7 **AUTHOR CONTRIBUTIONS**

8 LW and ZL designed, implemented, and tested the software. MD and LW managed metadata
9 and submission of raw data to NCBI SRA. LW and PVB designed and maintained the local
10 system consisting of a web server, a data server, and several computing servers. LW, ZL, and
11 XW developed the scientific workflows and participated in testing through the web interface. AD
12 and TG helped in developing the workflow. XW and LW processed the data with the automated
13 workflows. MR, JD, XX, COR, KB, DJ, RM, SG, and WM generated the raw data. CG developed
14 the Perl code for interacting with Agave API. CFM, CG, and DM managed the MaizeCODE
15 website. LW, DW, MS, and TG wrote the manuscript. All the authors read and approved the
16 final manuscript.

17

18 **AVAILABILITY AND REQUIREMENTS**

19

20 Project name: SciApps

21 Project home page: <https://sciapps.org/>

22 Source code repository: <https://github.com/warelab/sciapps>

23 Tutorial: <https://cyverse-sciapps-guide.readthedocs-hosted.com/en/latest/maizecode.html>

24 Software Design Document: <https://github.com/warelab/sciapps/blob/master/doc/SDD.pdf>

25 MaizeCODE data page: <https://www.sciapps.org/data/MaizeCODE>

- 1 Operating system(s): Platform independent
- 2 Programming language: JavaScript, Perl
- 3 License: MIT
- 4 Any restrictions to use the data: Toronto Agreement

5

6 **FUNDING**

- 7 This work is supported by NSF grant IOS 1445025 for MaizeCODE. SciApps has also been
- 8 supported by NSF grant DBI-1265383, and USDA-ARS (1907-21000-030-00D).

9

1 **References**

- 2 Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., et al. (2013). STAR:
3 ultrafast universal RNA-seq aligner. *Bioinformatics* 29, 15–21.
- 4 Dooley, R., Vaughn, M., Stanzione D, T. S., and Skidmore, E. (2012). Software-as-a-service:
5 the iPlant Foundation API. in *5th IEEE Workshop on Many-Task Computing on Grids and*
6 *Supercomputers (MTAGS)*.
- 7 ENCODE Consortium (2004). The ENCODE (ENCyclopedia Of DNA Elements) Project.
8 *Science* 306, 636–640.
- 9 Ewels, P., Magnusson, M., Lundin, S., and Käller, M. (2016). MultiQC: summarize analysis
10 results for multiple tools and samples in a single report. *Bioinformatics* 32, 3047–3048.
- 11 Frazee, A. C., Perte, G., Jaffe, A. E., Langmead, B., Salzberg, S. L., and Leek, J. T. (2015).
12 Ballgown bridges the gap between transcriptome assembly and expression analysis. *Nat.*
13 *Biotechnol.* 33, 243–246.
- 14 Goff, S. A., Vaughn, M., McKay, S., Lyons, E., Stapleton, A. E., Gessler, D., et al. (2011). The
15 iPlant Collaborative: Cyberinfrastructure for Plant Biology. *Front. Plant Sci.* 2, 34.
- 16 Grüning, B., The Bioconda Team, Dale, R., Sjödin, A., Chapman, B. A., Rowe, J., et al. (2018).
17 Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nature*
18 *Methods* 15, 475–476. doi:10.1038/s41592-018-0046-7.
- 19 Jiao, Y., Peluso, P., Shi, J., Liang, T., Stitzer, M. C., Wang, B., et al. (2017). Improved maize
20 reference genome with single-molecule technologies. *Nature* 546, 524–527.
- 21 Krueger, F., and Andrews, S. R. (2011). Bismark: a flexible aligner and methylation caller for
22 Bisulfite-Seq applications. *Bioinformatics* 27, 1571–1572. doi:10.1093/bioinformatics/btr167.

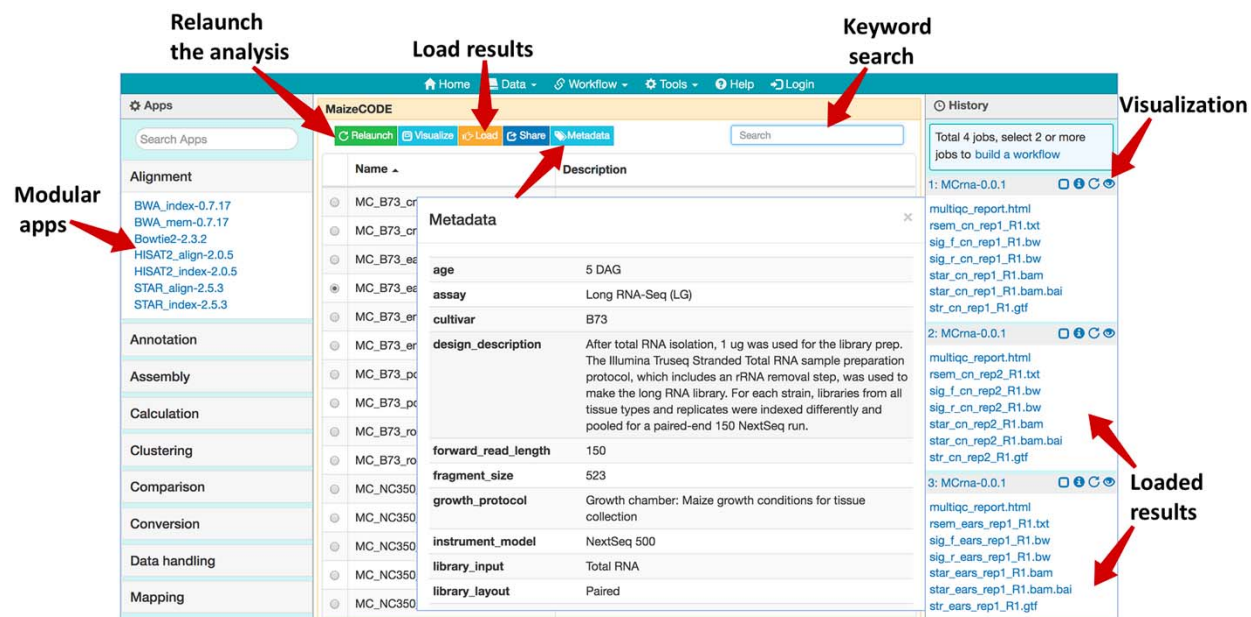
- 1 Kurtzer, G. M., Sochat, V., and Bauer, M. W. (2017). Singularity: Scientific containers for
2 mobility of compute. *PLoS One* 12, e0177459.
- 3 Li, B., and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data
4 with or without a reference genome. *BMC Bioinformatics* 12, 323.
- 5 Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T.-C., Mendell, J. T., and Salzberg, S. L.
6 (2015). StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat.*
7 *Biotechnol.* 33, 290–295.
- 8 Schnable, P. S., Ware, D., Fulton, R. S., Stein, J. C., Wei, F., Pasternak, S., et al. (2009). The
9 B73 maize genome: complexity, diversity, and dynamics. *Science* 326, 1112–1115.
- 10 Skinner, M. E., Uzilov, A. V., Stein, L. D., Mungall, C. J., and Holmes, I. H. (2009). JBrowse: A
11 next-generation genome browser. *Genome Research* 19, 1630–1638.
12 doi:10.1101/gr.094607.109.
- 13 Sloan, C. A., Chan, E. T., Davidson, J. M., Malladi, V. S., Strattan, J. S., Hitz, B. C., et al.
14 (2016). ENCODE data at the ENCODE portal. *Nucleic Acids Res.* 44, D726–32.
- 15 Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., et al. (2014). XSEDE:
16 Accelerating Scientific Discovery. in *Computing in Science & Engineering* 5., 62–74.
- 17 Wang, L., Lu, Z., Van Buren, P., and Ware, D. (2018). SciApps: a cloud-based platform for
18 reproducible bioinformatics workflows. *Bioinformatics* 34, 3917–3920.
- 19 Wang, L., Van Buren, P., and Ware, D. (2015). Architecting a distributed bioinformatics platform
20 with iRODS and iPlant Agave API. in *International Conference on Computational Science and*
21 *Computational Intelligence (CSCI)*, 420–423.

22

1 **Table 1.** SciApps release 1.0 RESTful API

Endpoint	Method	Description
/job	GET	List all jobs
/job/new/{id}	POST	Run a new job
/workflow/build	POST	Build a workflow from jobs
/workflowJob/new	POST	Generate a workflow JSON
/workflow/new	POST	Save a new workflow
/job/{id}	GET	Return the job JSON
/job/{id}/delete	GET	Delete the job
/workflowJob/run/{id}	GET	Run a new workflow
/workflow/{id}/metadata	GET	Get the workflow metadata
/workflow/{id}/update	POST	Update the workflow with metadata etc
/workflow	GET	List all workflows
/apps/{id}	GET	Return the application JSON
/workflow/{id}/delete	GET	Delete the workflow
/workflow/{id}	GET	Return the workflow JSON
/apps	GET	List all integrated apps

2



1

2 **Figure 1.** Web browser interface of the MaizeCODE data page. In the middle panel page, a list

3 of workflows/experiments is presented. Above the list, several action buttons are available:

4 'Relaunch' the analysis, 'Visualize' the graphic diagram of the workflow (with URLs for the raw

5 sequence files from the input file node), 'Load' the results to the History panel, 'Share' the

6 analysis with others, and display the experimental 'Metadata'. User can perform a keyword

7 search for a specific dataset (e.g., B73 ears RNA-Seq). In the right panel, SciApps displays the

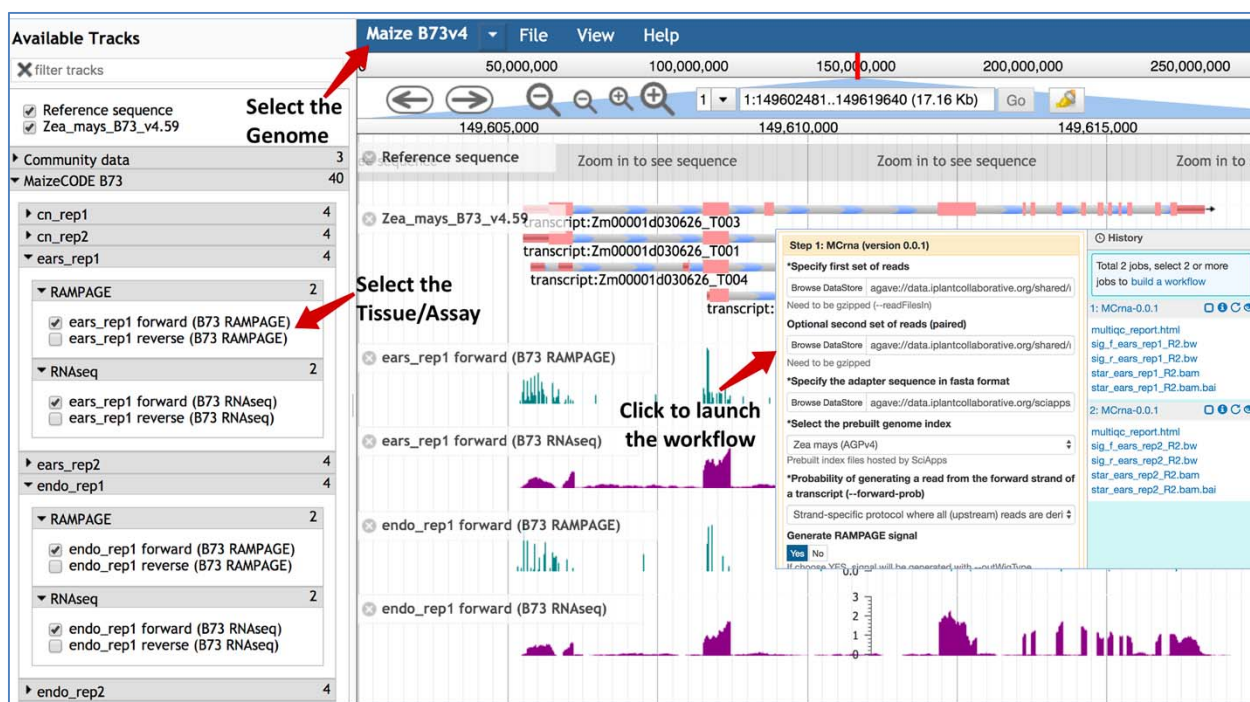
8 history of the selected datasets; the visualization (eye) icon opens a panel where users can

9 generate links to visualize the results in a web browser (e.g., a QC report) or genome browser

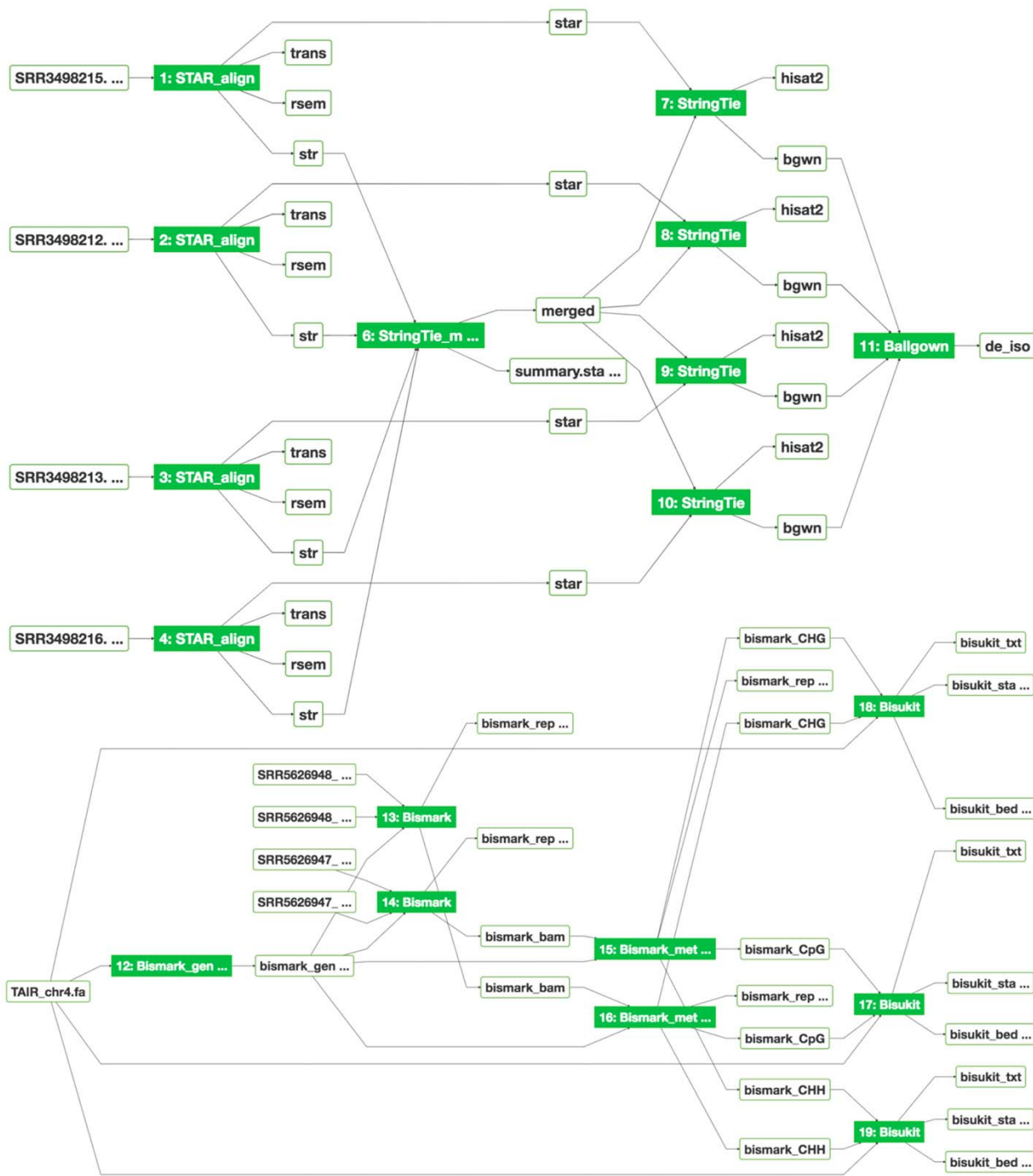
10 (e.g., alignments or signal tracks). The left panel shows a list of modular apps that can be

11 launched to perform a variety of downstream analyses with the loaded results.

12

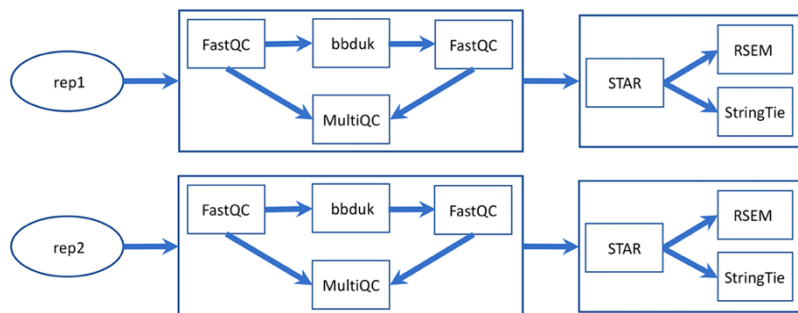


- 1
- 2 **Figure 2.** Genome browser tracks for the MaizeCODE data. JBrowse is used to hold the
- 3 MaizeCODE signal tracks, which are organized in the following order: genome, tissue, replicate,
- 4 and assay. Clicking on each track brings up the workflow 'Relaunch' interface.
- 5

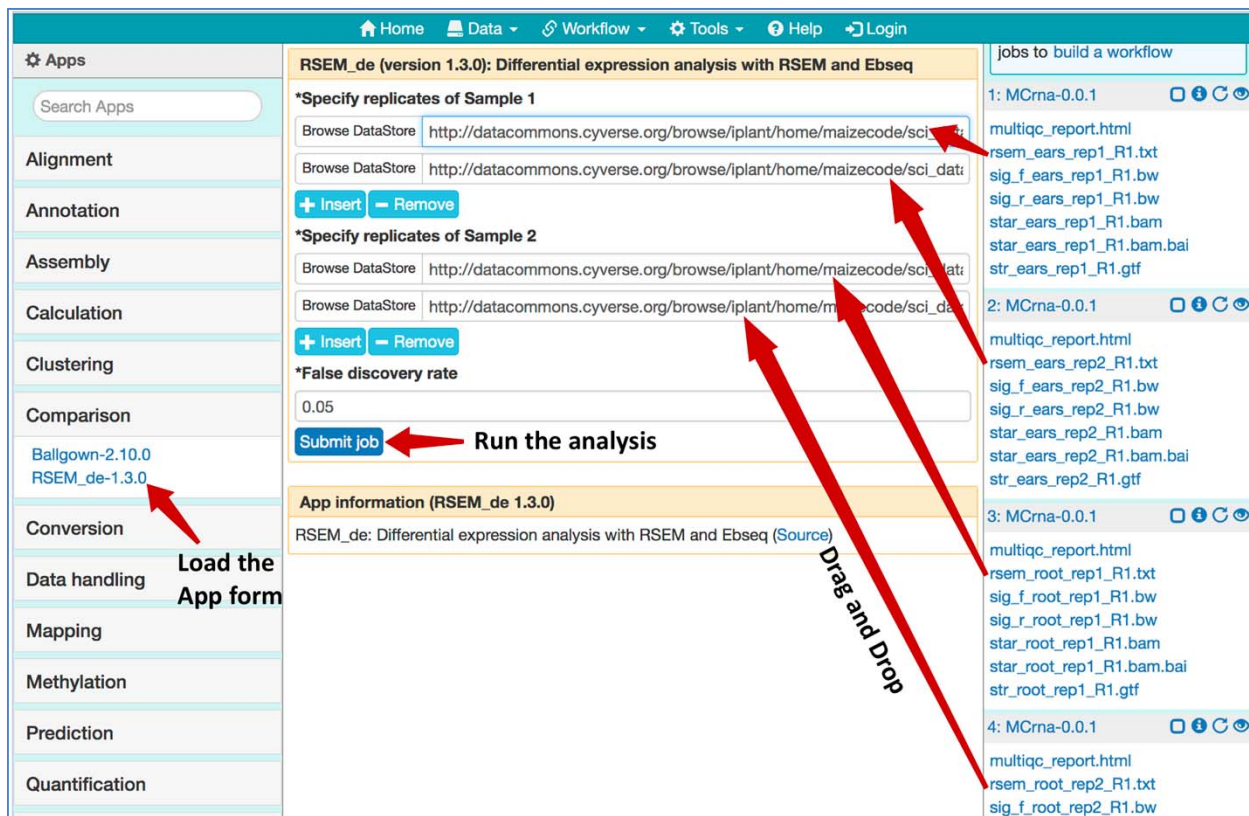


1
2 **Figure 3.** Graphical workflow diagrams for differential expression analysis (top) and MethyC-
3 seq analysis (bottom). The interactive graph demonstrates the relationships among input–output
4 files, displays provenance of the software tools, and provides real-time job status updates of

- 1 new analyses via the color of the app node (green: completed; blue: running; yellow: pending;
- 2 red: failed).
- 3



- 1
- 2 **Figure 4.** MaizeCODE MCrna app for processing RNA-seq data from two replicates. The
- 3 MCrna app wraps six tools, FastQC, bbduk, MultiQC, STAR, RSEM, and StringTie, together for
- 4 QC and quantification of each replicate.
- 5



- 1
- 2 **Figure 5.** Using the RSEM_de app for gene-level differential expression analysis. Job histories
- 3 are “Loaded” from the MaizeCODE data page (<https://www.sciapps.org/data/MaizeCODE>);
- 4 Clicking the RSEM_de-1.3.0 from the left App panel brings up the app form in the main panel;
- 5 Dragging and dropping the gene quantification result files starting with “rsem_” into the input
- 6 field, then clicking the “Submit job” button to run the differential expression analysis.