RESEARCH ARTICLE

STENCIL: A web templating engine for visualizing and sharing life science datasets

Qi Sun[1‡], Ali Nematbakhsh[1‡], Prashant K. Kuntala[2], Gretta Kellogg[1], B. Franklin Pugh[3], William K. M. Lai[3,4]*

1 Cornell Institute of Biotechnology, Cornell University, Ithaca, New York, United States of America, **2** Department of Biochemistry and Molecular Biology, The Pennsylvania State University, University Park, Pennsylvania, United States of America, **3** Department of Molecular Biology and Genetics, Cornell University, New York, United States of America, **4** Department of Computational Biology, Cornell University, New York, United States of America

‡ These authors share first authorship on this work
* wkl29@cornell.edu

Citation: Sun Q, Nematbakhsh A, Kuntala PK, Kellogg G, Pugh BF, Lai WKM (2022) STENCIL: A web templating engine for visualizing and sharing life science datasets. PLoS Comput Biol 18(2): e1009859. https://doi.org/10.1371/journal.pcbi.1009859

Editor: Mihaela Pertea, Johns Hopkins University, UNITED STATES

Received: June 4, 2021

Accepted: January 24, 2022

Published: February 9, 2022

Data Availability Statement: Source code for STENCIL is available at https://github.com/CEGRcode/stencil (STENCIL RRID:SCR_021878) with detailed documentation and examples are available at https://CEGRcode.github.io/stencil/. Sample Galaxy tools to communicate with STENCIL are available at https://github.com/CEGRcode/stencil-galaxy_tools.

Funding: This work was supported by NIH grants [R01-ES013768] and [R01-GM125722-03S1] awarded to BFP. Additional support was provided

Abstract

The ability to aggregate experimental data analysis and results into a concise and interpretable format is a key step in evaluating the success of an experiment. This critical step determines baselines for reproducibility and is a key requirement for data dissemination. However, in practice it can be difficult to consolidate data analyses that encapsulates the broad range of datatypes available in the life sciences. We present STENCIL, a web templating engine designed to organize, visualize, and enable the sharing of interactive data visualizations. STENCIL leverages a flexible web framework for creating templates to render highly customizable visual front ends. This flexibility enables researchers to render small or large sets of experimental outcomes, producing high-quality downloadable and editable figures that retain their original relationship to the source data. REST API based back ends provide programmatic data access and supports easy data sharing. STENCIL is a lightweight tool that can stream data from Galaxy, a popular bioinformatic analysis web platform. STENCIL has been used to support the analysis and dissemination of two large scale genomic projects containing the complete data analysis for over 2,400 distinct datasets. Code and implementation details are available on GitHub: https://github.com/CEGRcode/stencil

Author summary

Efficient and scalable data visualization of analysis is a critical bottleneck in biological discovery within life sciences. The exponential growth of genomic projects, which can now generate thousands of unique datasets in a single paper, have produced organizational and logistical difficulties for biochemists and bioinformaticians. To address these challenges, we developed the STENCIL web platform to incorporate principles of project management with a strong emphasis on data reproducibility and FAIR data practices. We architected STENCIL to operate on minimal system requirements (i.e., 1 CPU, 8Gb RAM) while seamlessly organizing thousands of unique samples in an intuitive interface

by the Cornell Institute of Biotechnology's Epigenomic Core Facility, (CU Epigenomics Core Facility, RRID:SCR_021287). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: I have read the journal's policy and the authors of this manuscript have the following competing interests: BFP has a financial interest in Peconic, LLC, which utilizes the ChIP-exo technology mentioned in this study and could potentially benefit from the outcomes of this research.

for data discovery. Data reproducibility is provided by the well-proven Galaxy bioinformatic web platform. The utility of the STENCIL platform has been demonstrated in two prior publications to date. Its capabilities have now been dramatically expanded to provide native support for Galaxy integration, Federated login, local hosting, custom domain usage, and improved security protocols for analyzing controlled de-identified biomedical patient data.

Introduction

Advances in next-generation sequencing have supercharged biochemical assays into 'big data' genomic resources [1–3]. This explosion in data has been paralleled by the development of multiple quality control and data analysis tools [4–8]. The unique analysis requirements from each distinct genomic assay complicates the already diverse ecosystem of bioinformatic tools by necessitating the creation of a novel tools and algorithms to maximize biological interpretation [9–12]. Many of these tools are equipped with quantitative and qualitative metrics designed to analyze user-supplied data, generating insights into different aspects of the experiment. While many tools generate quality control (QC) reports, they do not always provide mechanisms for sharing these reports with the broader community or for generating reports composed of a diverse array of genomic experiments. Moreover, it is often not possible to programmatically access the curated data and visualized results that retain their original relationship to the source data.

Existing web-based approaches to data visualization include the R-Shiny and Dash-python web templating platforms that provide interactive analysis and plots. Shiny is an open source software platform from the R-project that provides a framework for building interactive web applications. Shiny has the advantage of providing native support for all existing R-based analysis packages. This has resulted in its rapid adoption by many research groups in the genomic field [13–17]. Dash-python is a similar web-templating package based on the Python language. Similar to Shiny's R background, Dash-python's python language heritage allows for the easy application of the myriad existing bioinformatic python packages for visualization [18]. We note that while both of these packages are open-source and the base version is freely available, many of their comparable features are advanced, such as Federated login, custom domain usage, and public data dissemination which are paid features costing users $50,000-$60,000/year [19,20]. Additionally, no current web-templating bioinformatic analysis approach provides complete analysis reproducibility and permanent session management and tracking. To address the distinct need for reproducible data analysis [21,22], we turned towards the Galaxy bioinformatic analysis platform.

The Galaxy platform is a large-scale NSF and NIH-funded initiative intended to solve many of the issues in data analysis reproducibility [23]. Galaxy is a self-contained, portable, open-source software platform that creates sharable and executable script pipelines (workflows). Galaxy tracks the exact tool parameters and computational environment for every script and tool run on its platform to enable complete bioinformatic data reproducibility [24]. Novel bioinformatic tools are easily added in the Galaxy ecosystem with over 8,000 tools currently available (as of April 2021) for download and workflow execution [25]. Critically, Galaxy provides complete programmatic API control over the system for advanced users [26].

While the Galaxy platform has tremendous capabilities for running publication-validated workflows and generating reproducible data analysis, there is no current system in place to analyze the visualizations generated from multiple workflows simultaneously. The ability to readily inspect and interpret the data, without requiring programming expertise, plays a crucial role for advancing work in the life sciences, allowing wet bench scientists to validate experimental results, develop new hypotheses, and obtain insight into the massive output of genomic data provided by the latest technologies [27–30].

As a step toward data management and visualization, we developed STENCIL, an open source web templating engine that provides a framework for creating flexible, scalable, and interactive web visualizations that adapt to the growing needs of a project as required. STENCIL is designed for efficient reporting, data analysis visualization, and data dissemination in support of recommended FAIR Data practices [21]. STENCIL provides the ability to visualize and compare large sets of samples of interest within the same web frame. STENCIL supports retaining the links to reproducible workflows and removes any file storage redundancy, resulting in direct dataset visualization. It leverages Galaxy's existing REST API interface to serve visualizations directly to STENCIL, providing complete programmatic control over all the displayed data, substantially improving a user's discovery experience. Since STENCIL presents and aggregates the visualizations and analysis results already run from Galaxy or other workflow engines and pipelines, it is extremely light weight and can be run on an individual workstation or from a single CPU virtual machine.

Design and implementation

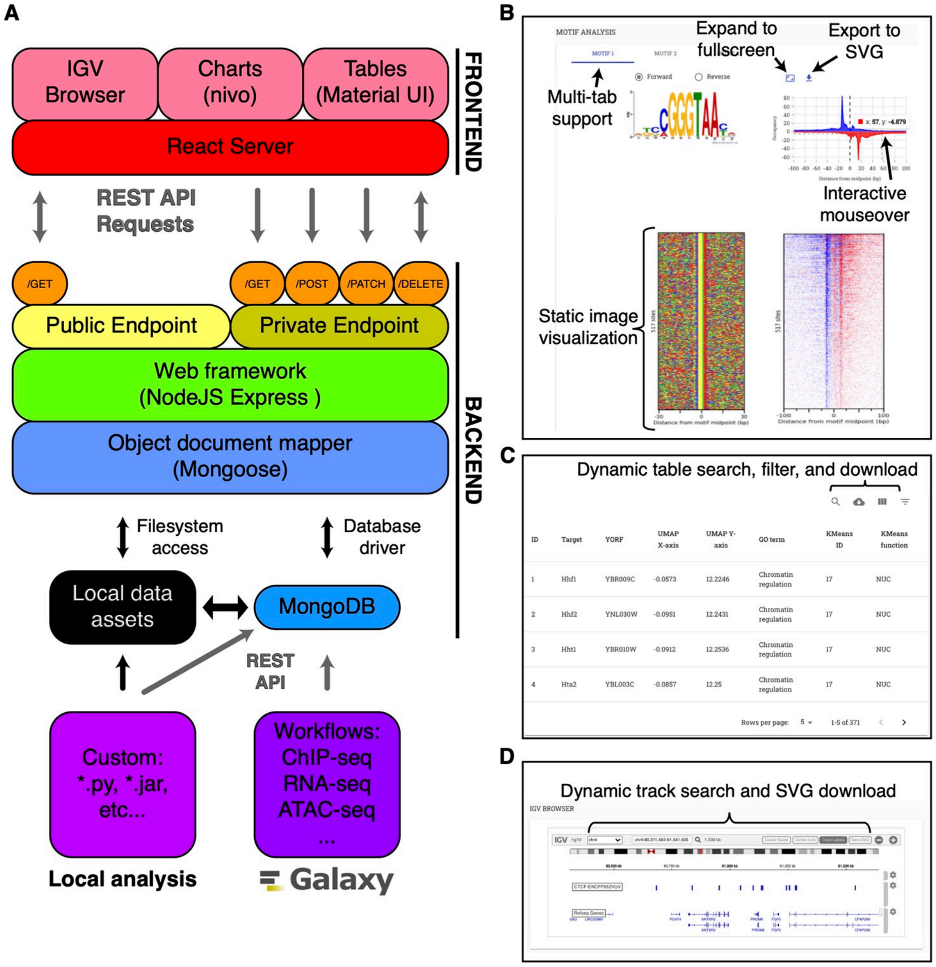
Software architecture. STENCIL is architected to aggregate data analyses from heterogeneous sources into a single reporting structure. STENCIL is composed of two primary subcomponents: Data Consuming Front End and Data Producing Back End. This design strategy dramatically simplifies horizontal scaling, allowing for the simultaneous running of multiple distinct STENCIL systems which can share the same core resources as specified. This high level of scalability ensures that STENCIL can serve thousands of unique datasets to multiple concurrent users on a simple webserver consisting of a single CPU and 8 Gb of RAM [31,32]. STENCIL also supports common deployment strategies including Docker containers and locally managed servers through automated continuous integration/continuous deployment (CI/CD) workflows that are triggered by code changes.

STENCIL architecture is centered around 'experiments' as the fundamental unit. Each experiment contains one or more 'sections' that correspond to any arbitrary combination of data analysis output including, but not limited to: static images, interactive plots, and experimental meta-information in blocks of text or tables. Conceptually these sections consist of two elements: a React component forming the display element (frontend) and a JSON object serving as the data element (backend).

Front end

STENCIL's front end is a React JavaScript application that efficiently consumes REST APIs provided by the back end (**Fig 1A**). Importantly, there is no set requirement for visualization of any experiment. The STENCIL experiment page dynamically resizes itself to visualize the data that is attributed to each sample. As more data is added (i.e., over the course of a Galaxy workflow execution), static and interactive plots as well as data tables are added into an experiment's front end in real-time.

STENCIL provides data visualization through two complementary methods. The first and simplest method of STENCIL data visualization is through the hosting of static images (PNG, JPG, SVG, etc. . .). This capability allows STENCIL to visualize the output generated directly from many common analysis pipelines that generate static images (e.g., DESeq2, EdgeR). The second method of data visualization is through interactive plots, where STENCIL pulls data



ID	Target	YORF	UMAP X-axis	UMAP Y-axis	GO term	KMeans ID	KMeans function
1	Hhf1	YBR009C	-0.0573	12.2246	Chromatin regulation	17	NUC
2	Hhf2	YNL030W	-0.0951	12.2431	Chromatin regulation	17	NUC
3	Hht1	YBR010W	-0.0912	12.2536	Chromatin regulation	17	NUC
4	Hta2	YBL003C	-0.0857	12.25	Chromatin regulation	17	NUC

Rows per page: 5 1-5 of 371

Fig 1. Overview of STENCIL architecture. (A) A ReactJS frontend server provides web-access to the data. A NodeJS server and MongoDB instance store and manage the data provided and disseminated through RESTful API calls. Data analysis and hosted URLs are provided by the Galaxy platform and communicated directly to the MongoDB. Local analysis and file-hosting outside of the Galaxy platform is also supported. (B) A sample React-served experiment section demonstrating multiple types of data. React serves static PNG/JPG/SVG images hosted remotely. Interactive charts include dynamic mouseover with additional data, the ability to expand to full screen, and export to SVG. (C) Data table analysis allows for sorting and filtering using remotely hosted data. (D) Integrated Genome Browser (igv) provides interactive track visualization in the same web frame as other advanced analysis.

https://doi.org/10.1371/journal.pcbi.1009859.g001

directly from the backend and renders and visualizes the data dynamically. The rendering of a variety of different plots is provided by a series of React JavaScript components.

While STENCIL provides a collection of default React components that effectively visualize common life science charts and images, extensibility was a primary consideration behind STENCIL's design. As a result, the modular nature of STENCIL makes it is easy to rapidly add and remove new tool sections with minimal coding. The multiple plots of each sample can be organized into sections and tabs, with multiple sections per page, and multiple tabs per section. In each tab, the plots are organized either by rows or by columns. The number of rows and columns, as well as order of the plots in the layout can be customized in the configuration file using the simple, documented syntax. Additionally, the prevalence of React in the scientific community, another key consideration in STENCIL's design, directly translates to many existing React plotting tools that can be used for data visualization. The front end uses the d3.jsderived nivo charting library to create interactive dynamic charts from the data by default [33].

STENCIL comes with several pre-defined React components that can serve as initial templates for user-development. One of the sample templates is a React component that serves static PNG, JPG, and SVG images with an associated interactive JavaScript chart. Default chart features include dynamic mouseover with additional data, the ability to expand to full-screen, and an 'export to SVG' button for further refinement in vector-based imaging software (**Fig 1B**). Static images are generated outside of STENCIL by any number of pipelines and workflows including the Galaxy platform. The use of REST APIs provides complete programmatic control over all the displayed data.

Material UI-derived mui-datatables provide dynamic filtering and sorting of text-based data in table format [33] (**Fig 1C**). Genome browser functionality is provided through the integration of the BROAD Institute's JavaScript implementation of IGV [34] (**Fig 1D**). Importantly, STENCIL is also 'plug and play' compatible with virtually all JavaScript charting libraries including the free and open source Google-charts JavaScript library.

Back end

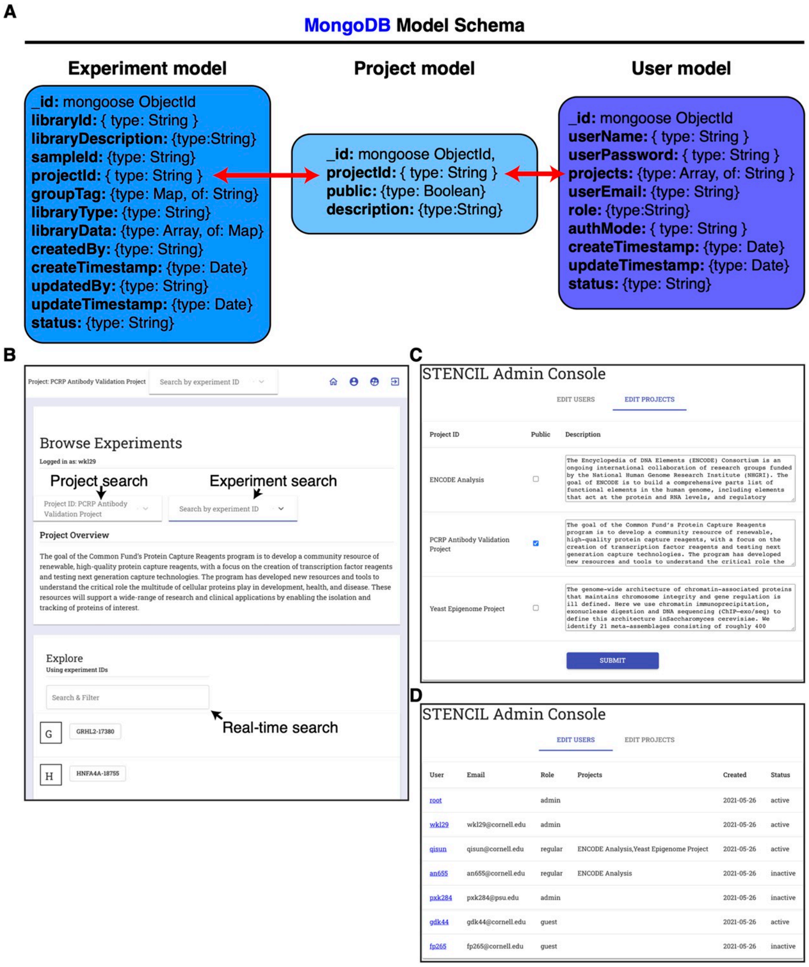
STENCIL's back end consists of a NodeJS Express application connected to a MongoDB database engine (**Fig 1A**). The MongoDB data models (**Fig 2A**) store and organize all the relevant meta-information associated with each experiment for display in the frontend, and an array of plot identifier and metadata associated with each plot. STENCIL's backend database is designed to easily allow many-to-many relationships between experiments and projects, and between projects and users. The sample metadata include two required fields: project identifier and sample identifier; these are used to identify each experiment. The metadata for each plot include four required fields: layout identifier, tab identifier, plot identifier, and URL. The URL can be a link to the remote server or a local file path. The link can be either an image file or a JSON file with actual data values. If the link is a JSON file, the front-end plotting tool will be used to create a dynamic graph.

Sample and analysis tracking is accomplished by the MongoDB backend at two distinct layers. The first layer is the 'Project' layer which contains any number of sample pages. The use of MongoDB's many-to-many relationships allows for samples to be assigned to multiple projects without data duplication. The second layer is at the 'Experiment' layer which can contain any arbitrary number of distinct workflow analyses. Again, the nature of our backend design allows for the same analysis to be associated with any number of 'Experiments' across multiple 'Projects' simultaneously without data duplication. This is accomplished by a workflow submitting multiple POST requests, with each request directing the results to a different 'Experiment' and 'Project'.

STENCIL's front end seamlessly parses these relationships and provides a user interface capable of navigating and modifying many of these relationships. The main landing page after login provides a user-searchable list of permission-assigned available projects (**Fig 2B**). Since each project can contain anywhere from a few dozen to a few thousand experiments [31,32], the landing page also offers a real-time auto-complete searchable interface to find experiments within the project.

Data security and single Sign-On integration

STENCIL provides Role Based Access Control to authorize appropriate access to the datasets and visualizations and provides more granular project-specific access control as well [35]. The application backend includes a permission authorization table in the MongoDB. Each ID


A
MongoDB Model Schema
Experiment model
Project model
User model
_id: mongoose ObjectId
libraryId: { type: String }
libraryDescription: {type:String}
sampleId: {type: String}
projectId: { type: String }
groupTag: {type: Map, of: String}
libraryType: {type: String}
libraryData: {type: Array, of: Map}
createdBy: {type: String}
createTimestamp: {type: Date}
updatedBy: {type: String}
updateTimestamp: {type: Date}
status: {type: String}
_id: mongoose ObjectId,
projectId: { type: String }
public: {type: Boolean}
description: {type:String}
_id: mongoose ObjectId
userName: { type: String }
userPassword: { type: String }
projects: {type: Array, of: String }
userEmail: {type: String}
role: {type:String}
authMode: { type: String }
createTimestamp: {type: Date}
updateTimestamp: {type: Date}
status: {type: String}
B
C
D
Project search
Experiment search
Real-time search
EDIT USERS
EDIT PROJECTS
SUBMIT
Search by experiment ID
Search & Filter
G
H

Project: PCRP Antibody Validation Project

Browse Experiments

Logged in as: wkl29

Project ID: PCRP Antibody Validation Project

Project Overview

The goal of the Common Fund's Protein Capture Reagents program is to develop a community resource of renewable, high-quality protein capture reagents, with a focus on the creation of transcription factor reagents and testing next generation capture technologies. The program has developed new resources and tools to understand the critical role the multitude of cellular proteins play in development, health, and disease. These resources will support a wide-range of research and clinical applications by enabling the isolation and tracking of proteins of interest.

Explore

Using experiment IDs

GRHL2-17380

HNFA4A-18755

STENCIL Admin Console

Project ID	Public	Description
ENCODE Analysis	☐	The Encyclopedia of DNA Elements (ENCODE) Consortium is an ongoing international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI). The goal of ENCODE is to build a comprehensive parts list of functional elements in the human genome, including elements that act at the protein and RNA levels, and regulatory
PCRP Antibody Validation Project	☑	The goal of the Common Fund's Protein Capture Reagents program is to develop a community resource of renewable, high-quality protein capture reagents, with a focus on the creation of transcription factor reagents and testing next generation capture technologies. The program has developed new resources and tools to understand the critical role the
Yeast Epigenome Project	☐	The genome-wide architecture of chromatin-associated proteins that maintains chromosome integrity and gene regulation is ill defined. Here we use chromatin immunoprecipitation, exonuclease digestion and DNA sequencing (ChIP–exo/seq) to define this architecture inSaccharomyces cerevisiae. We identify 21 meta-assemblages consisting of roughly 400

STENCIL Admin Console

User	Email	Role	Projects	Created	Status
root		admin		2021-05-26	active
wkl29	wkl29@cornell.edu	admin		2021-05-26	active
qisun	qisun@cornell.edu	regular	ENCODE Analysis,Yeast Epigenome Project	2021-05-26	active
an655	an655@cornell.edu	regular	ENCODE Analysis	2021-05-26	inactive
pxk284	pxk284@psu.edu	admin		2021-05-26	inactive
gdk44	gdk44@cornell.edu	guest		2021-05-26	active
fp265	fp265@cornell.edu	guest		2021-05-26	inactive

Fig 2. STENCIL backend enables precise project organization and authorization (A) MongoDB model schema of STENCIL backend. URL references to experiment analysis are stored in the Experiment model. Experiments are linked to the Project model through a common projectId string. Users gain access to models and underlying experimental data through assignment to a variable number of projectIds. (B) Experiment navigation screen of STENCIL. STENCIL supports users accessing multiple public and private projects as defined by the backend database. Real-time search enables quick filtering and access of experiments in large projects. (C) STENCIL Admin console is available to STENCIL administrators and allows for project public/private control and project summary edits. (D) STENCIL Admin console provides control of user access to projects and defining user roles.

https://doi.org/10.1371/journal.pcbi.1009859.g002

is assigned a level of access to the application. Defined roles include Admin, Regular, and Guest.

A guest is a public user that may only view and download experimental data for projects that are designated as public. STENCIL assigns each experiment a unique URL that can be publicly shared as needed. A Regular user is an authenticated user via a local login or Federated Single Sign-on. Regular users are only able to view and download public library pages until an administrator designates which project(s) they can work with on the website. As all Projects and their associated Experiment pages define URLs for which users are authorized (Fig 2A), STENCIL's authentication system prevents unauthorized users from gaining access to Experiment and Project pages even if they possess the URL to the analysis. This is a critical feature for properly storing and analyzing biomedical data.

Only STENCIL administrators have access to the Admin Console that enables them to designate project and role authorizations across the platform. Administrators can make projects globally public, as well as edit the project summary displayed for each project (**Fig 2C**). The Admin role can define the level of access to a user ID; they can designate users to allow them to have appropriate access to view or edit the information. The Admin role also manages the projects, and assigns projects for each user, and can assign a Status to a User as 'active' or 'inactive' in order to more carefully control access to data files and experiment pages (**Fig 2D**).

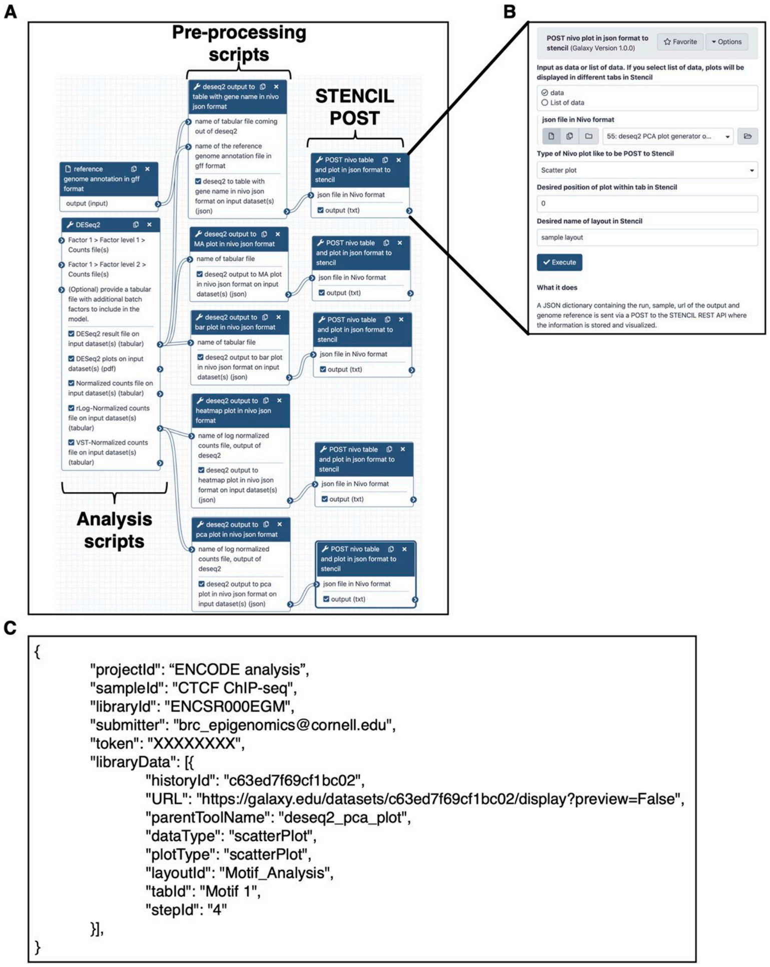
STENCIL provides two different authentication mechanisms: stand-alone user authentication and Shibboleth Single Sign-On (SSO). Shibboleth is a Single Sign-On solution for institutions within the InCommon Federation with optional integration to Two-Factor Authentication. Upon installation of STENCIL, the default initial role of site administrator can set up either or both authentication options for the website. SSO provides the ability for users to authenticate to STENCIL using their university or institutional account without requiring additional management overhead for ID and password creation. The application stores the login information in a cookie and allows for SSO login for an extended defined period. This approach eases the cost of adoption of the software tool, particularly for smaller labs and individual PIs. For stand-alone user authentication, STENCIL stores the user ID and encrypted password within its backend MongoDB and authenticates the user by matching the stored password. The site administrator can reset local passwords and user IDs as needed in this model.

Galaxy integration

STENCIL was conceived of as a system to aggregate the downloaded results of multiple Galaxy workflows. This matured into its current iteration of hosting direct links to Galaxy datasets. In addition to the practical benefits of removing the need for data duplication (data already exists within Galaxy), it has the major benefit of maintaining the auditable and reproducible analysis generated within the Galaxy platform [24]. The datasets hosted on Galaxy enable STENCIL users to leverage Galaxy's API and data dissemination features to achieve FAIR data standards compliance, all the way through final figure generation for publications. Since 2018, Galaxy has provided native support for hosting protected biomedical information through its compliance with the European Union's GDPR mandate [36]. This enables users of STENCIL to directly benefit from existing approaches to privacy law compliance.

Galaxy integration with STENCIL is enabled by two classes of custom Galaxy tools available here: (https://github.com/CEGRcode/galaxy_tools_for_stencil). The first class of tool is a pre-processing script that converts the output of an analysis tool into a JSON payload. Additional pre-processing scripts will be available to support common analysis algorithms (i.e., DESeq2, Cuffdiff, MEME). The second tool is a script whose sole purpose is to POST the plain-text JSON file to a STENCIL webserver (**Fig 3A**). The simplicity of the second tool provides enormous flexibility in its usage. The user pre-defines in a workflow (recommended), or using Galaxy's real-time interface, whether the data being sent is a static image (PNG/JPG/SVG), a table, or a nivo chart, to be dynamically rendered (**Fig 3B**). The current dynamic chart options supported are: Line Plot, Scatter Plot, Bar Plot, Heat Map, and Data Table and are selectable using a simple dropdown menu in the Galaxy interface. Additional chart options will be supported to accommodate a wider variety of supported analyses (https://usegalaxy.org/workflows/list_published).

The JSON payload sent from Galaxy contains the minimal amount of information required for STENCIL to properly visualize the data as well as provide a mechanism (Galaxy historyID) to trace the analysis back to Galaxy (**Fig 3C**). The small payload enables the STENCIL database


A
Pre-processing scripts
STENCIL POST
reference genome annotation in gff format
output (input)
DESeq2
Factor 1 > Factor level 1 > Counts file(s)
Factor 1 > Factor level 2 > Counts file(s)
(Optional) provide a tabular file with additional batch factors to include in the model.
☑ DESeq2 result file on input dataset(s) (tabular)
☑ DESeq2 plots on input dataset(s) (pdf)
☑ Normalized counts file on input dataset(s) (tabular)
☑ rLog-Normalized counts file on input dataset(s) (tabular)
☑ VST-Normalized counts file on input dataset(s) (tabular)
Analysis scripts
deseq2 output to table with gene name in nivo json format
name of tabular file coming out of deseq2
name of the reference genome annotation file in gff format
☑ deseq2 to table with gene name in nivo json format on input dataset(s) (json)
deseq2 output to MA plot in nivo json format
name of tabular file
☑ deseq2 output to MA plot in nivo json format on input dataset(s) (json)
deseq2 output to bar plot in nivo json format
name of tabular file
☑ deseq2 output to bar plot in nivo json format on input dataset(s) (json)
deseq2 output to heatmap plot in nivo json format
name of log normalized counts file, output of deseq2
☑ deseq2 output to heatmap plot in nivo json format on input dataset(s) (json)
deseq2 output to pca plot in nivo json format
name of log normalized counts file, output of deseq2
☑ deseq2 output to pca plot in nivo json format on input dataset(s) (json)
POST nivo table and plot in json format to stencil
json file in Nivo format
☑ output (txt)
B
Favorite
Options
Execute
C

POST nivo plot in json format to stencil (Galaxy Version 1.0.0)

Input as data or list of data. If you select list of data, plots will be displayed in different tabs in Stencil

● data
○ List of data

json file in Nivo format

55: deseq2 PCA plot generator o...

Type of Nivo plot like to be POST to Stencil

Scatter plot

Desired position of plot within tab in Stencil

0

Desired name of layout in Stencil

sample layout

What it does

A JSON dictionary containing the run, sample, url of the output and genome reference is sent via a POST to the STENCIL REST API where the information is stored and visualized.

```
{
        "projectId": “ENCODE analysis”,
        "sampleId": "CTCF ChIP-seq",
        "libraryId": "ENCSR000EGM",
        "submitter": "brc_epigenomics@cornell.edu",
        "token": "XXXXXXXX",
        "libraryData": [{
                "historyId": "c63ed7f69cf1bc02",
                "URL": "https://galaxy.edu/datasets/c63ed7f69cf1bc02/display?preview=False",
                "parentToolName": "deseq2_pca_plot",
                "dataType": "scatterPlot",
                "plotType": "scatterPlot",
                "layoutId": "Motif_Analysis",
                "tabId": "Motif 1",
                "stepId": "4"
        }],
}
```

Fig 3. Galaxy integration with STENCIL. (A) A custom python script provides the mechanism through which a Galaxy tool (i.e., DESeq2) can POST its output to STENCIL (B) The STENCIL-Galaxy communication tool allows the user to specify the type of data being transmitted to STENCIL (C) A sample JSON payload containing the minimum amount of information needed by STENCIL to correctly place the analysis within an experiment and associated project.

https://doi.org/10.1371/journal.pcbi.1009859.g003

to scale tremendously in size without dramatically increasing its disk space footprint. Additionally, the decision to use a document-based non-relational database (MongoDB) allows the user to trivially track any arbitrary additional sample meta-information such as genome ID, timestamp, workflow ID, etc. The design of this payloads allows for multiple Galaxy workflows to send their analysis results to the same STENCIL project and experiment. This structure fundamentally allows STENCIL to aggregate data from multiple discrete workflows into a single reporting structure.

Alternative workflow support and local hosting

STENCIL is primarily designed to complement existing Galaxy functionality. However, our architecture also supports integration with any analysis workflow that can execute a POST request. The POST must contain a JSON file detailing relevant sample information to the STENCIL backend. Alternative workflow engines such as Pegasus as well as standard shell-script based analyses are supported [37]. STENCIL can seamlessly support file hosting from any arbitrary webserver (NGINX, Apache, etc.), locally installed, or cloud hosted.

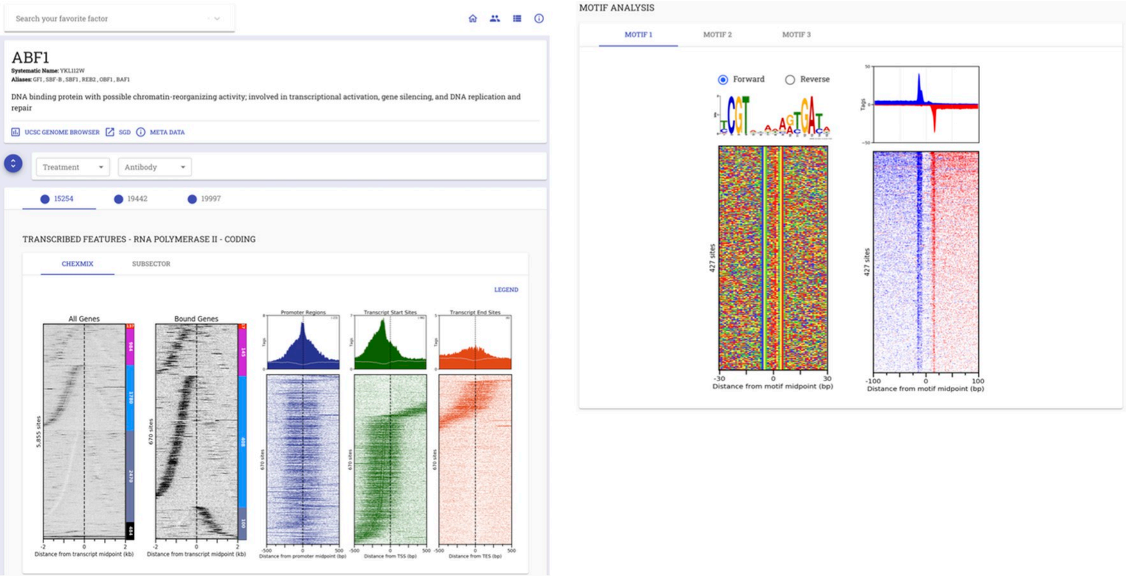
In the event a separate webserver is not available for hosting data or local hosting is preferred due to biomedical privacy constraints, the included React backend server of STENCIL is also configured to allow for local file hosting. The POST request will result in STENCIL visualizing the data as posted to the MongoDB instance referencing the local files. Critically, this functionality provides support for highly custom local analyses that often arise during algorithmic development and may not necessarily have existing support in the Galaxy toolshed. Although this process does not benefit from the significant bioinformatic tracking support provided by the Galaxy platform, it does allow bioinformaticians to rapidly iterate and develop novel algorithms, leveraging STENCIL's visualization capabilities to develop robust and reproducible workflows.

Results

To date, STENCIL has been used to generate web applications to support analysis and dissemination of two large scale genomic projects each containing the data analysis for over 2,400 distinct genomic datasets. STENCIL currently operates on a webserver with a single CPU and 8Gb and easily hosts and disseminates all available data to multiple concurrent users.

Protein capture reagents program validation

The Protein Capture Reagent Program (PCRP) was a NIH Common Fund project to generate and validate a renewable source of immunoreagents [38]. STENCIL was used to collate and provide a mechanism for the project's data dissemination as per the NIH project requirements (http://www.pcrpvalidation.org) (**Fig 4**) [32]. Data was generated using a custom analysis pipeline (https://github.com/CEGRcode/PCRPpipeline) and uploaded to the STENCIL React backend. While this workflow enabled fast and accessible data analysis inspection, it lacked the full range of data reproducibility offered by the Galaxy platform.



ABF1

Systematic Name: YKL112W

Aliases: GF1 , SBF-B , SBF1 , REB2 , OBF1 , BAF1

DNA binding protein with possible chromatin-reorganizing activity; involved in transcriptional activation, gene silencing, and DNA replication and repair

Fig 4. STENCIL application in PCRP. Application of STENCIL in the PCRP was an NIH-initiated project to screen 800 antibodies in ChIP-exo. NIH project requirements included providing the generated data in a community-accessible manner.

https://doi.org/10.1371/journal.pcbi.1009859.g004

Yeast epigenome project

As part of our continued STENCIL development after the PCRP project, we incorporated Galaxy as a means of enabling completely reproducible data analysis. The Yeast Epigenome Project (http://www.yeastepigenome.org) was an ambitious endeavor to comprehensively map locations of protein-DNA interaction in *S. cerevisiae* [31]. All data for the Yeast Epigenome Project was generated using a Galaxy workflow available here: (https://github.com/CEGRcode/cegr-yep-qcviz/tree/master/exportedWorkflows). We generated over 1,200 unique genomic datasets and dozens of affiliated analyses per dataset that were all created using a completely reproducible Galaxy workflow. The resulting data was then uploaded to the React backend. STENCIL was used in this project to provide an interface for users to examine both quality control metrics as well as interrogate the baseline biology of each unique factor's binding in the genome (**Fig 5**).

Results of RNA-seq and ATAC-seq pipelines in STENCIL

We continued to refine our use of STENCIL to remove data duplication resulting from downloading data from Galaxy and uploading to STENCIL, although we maintain that functionality to support all possible analysis workflows. The recommended usage of STENCIL is to host data directly from Galaxy. We provide a template RNA-seq analysis workflow that POSTs the data URL directly from Galaxy instance to a STENCIL webserver (**Fig 6A**). Since the data is stored and hosted from Galaxy, there is no unnecessary data duplication. As an additional

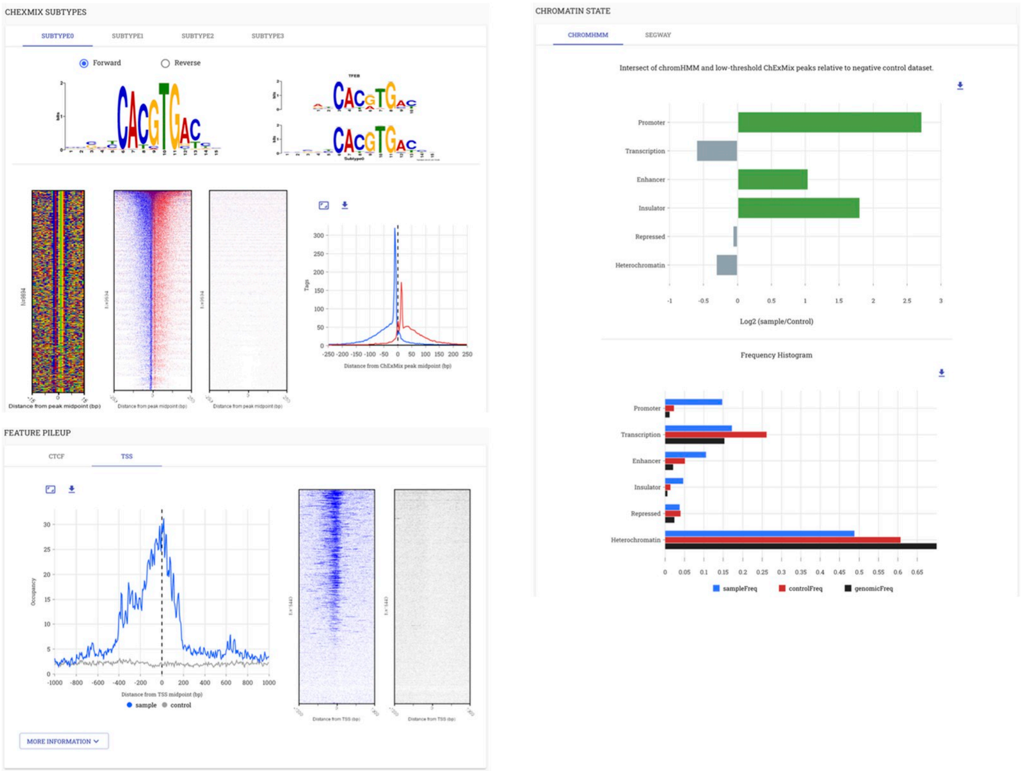
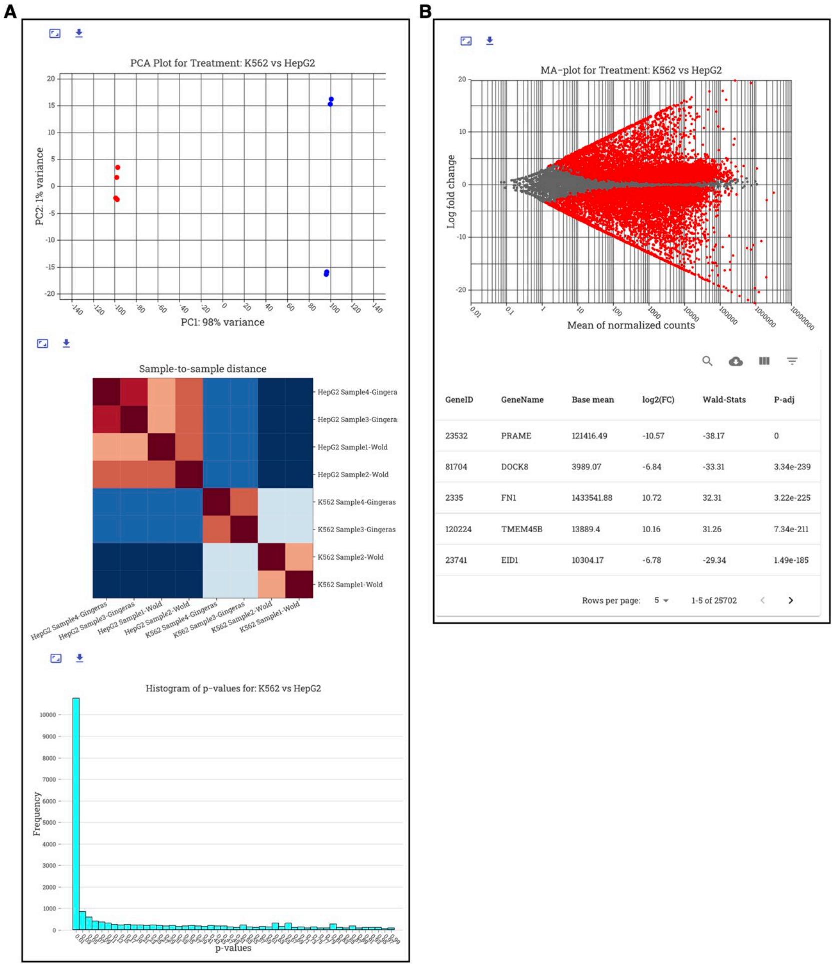


Fig 5. Application of STENCIL in the Yeast Epigenome Project. A comprehensive map of protein binding in *S. cerevisiae* required the development of a Galaxy workflow to both generate quality control metrics as well as provide baseline biological insight. Full analysis with high-resolution figures is available at http://www.yeastepigenome.org/yep/factor/ABF1.

https://doi.org/10.1371/journal.pcbi.1009859.g005



GeneID	GeneName	Base mean	log2(FC)	Wald-Stats	P-adj
23532	PRAME	121416.49	-10.57	-38.17	0
81704	DOCK8	3989.07	-6.84	-33.31	3.34e-239
2335	FN1	1433541.88	10.72	32.31	3.22e-225
120224	TMEM45B	13889.4	10.16	31.26	7.34e-211
23741	EID1	10304.17	-6.78	-29.34	1.49e-185

Rows per page: 5 ▾ 1-5 of 25702 ‹ ›

Fig 6. DESeq2 RNA-seq analysis visualized in STENCIL. (A) DESeq2 charts are visualized as dynamic charts in STENCIL. All data is generated and hosted directly from Galaxy. (B) The nivo charting library allows for on-the-fly generated of interactive plots containing tens of thousands of unique datapoint in seconds.

https://doi.org/10.1371/journal.pcbi.1009859.g006

advantage, in this configuration, STENCIL now maintains a link to the unique Galaxy history and data. This provides a continuous link from initial data input all the way through figure generation in line with FAIR data practices.

In addition to being able to host thousands of concurrent datasets, STENCIL can visualize tens of thousands of datapoints simultaneously and interactively in the same web frame (Fig 6B). We examined the ability of STENCIL to render large sets of datapoints into interactive interfaces by calculating rendering time using Firefox's native website render timer. 14,000 datapoints in an interactive MA scatterplot from a DESeq2 RNA-seq analysis were loaded into 3 distinct STENCIL Experiment pages where the plot was rendered: 1, 5, or 10 times respectively for a max rendering of 140,000 distinct interactive datapoint (Table 1). We performed three timing replicates across two distinct machines to examine the effect of loading time based on STENCIL's client-side rendering architecture. Timing was calculated using Firefox's native page rendering tool. We found that even while rendering 140,000 datapoints on an older (10+ year) computer, the page loaded in under 6 seconds. We have also the applied the same workflow using ATAC-seq data demonstrating its utility across multiple genomic assays (S1 Fig).

Table 1. Evaluation of STENCIL load times.

Test platform	Number of distinct plots	Data points per plot	Total data points	Avg load time (millisec)	Std dev load time (millisec)
macOS Catalina 10.15.7, 4.2 GHz Quad-Core Intel Core i7, 32 GB 2400 MHz DDR4 *Web browser*: *Firefox 93.0*	1	14,000	14,000	173.0	40.0
macOS Catalina 10.15.7, 4.2 GHz Quad-Core Intel Core i7, 32 GB 2400 MHz DDR4. *Web browser*: *Firefox 93.0*	5	14,000	70,000	1,230.7	62.9
macOS Catalina 10.15.7, 4.2 GHz Quad-Core Intel Core i7, 32 GB 2400 MHz DDR4. *Web browser*: *Firefox 93.0*	10	14,000	140,000	5,109.7	321.1
macOS High Sierra 10.13.18, 2.7 GHz Intel Core i7 8 GB 1333 MHz DDR3 *Web browser*: *Firefox 93.0*	1	14,000	14,000	233.3	17.1
macOS High Sierra 10.13.18, 2.7 GHz Intel Core i7 8 GB 1333 MHz DDR3 *Web browser*: *Firefox 93.0*	5	14,000	70,000	1,277.7	30.4
macOS High Sierra 10.13.18, 2.7 GHz Intel Core i7 8 GB 1333 MHz DDR3 *Web browser*: *Firefox 93.0*	10	14,000	140,000	5,493.0	254.4

https://doi.org/10.1371/journal.pcbi.1009859.t001

Discussion

One of the main bottlenecks in analysis of genomic data is efficient and scalable visualization approaches. To address this challenge, we developed a graphical reporting interface (STENCIL) designed to integrate data generated by the Galaxy platform or in combination with custom lab workflows. STENCIL is capable of dynamically composing interactive graphical reports on biological results from multiple genomic assays. In its Galaxy integration option, STENCIL displays its data directly from the Galaxy instance, removing the need for local duplication of data files. STENCIL also enables aggregate data from multiple assays and workflows from any hosted Galaxy instance into a single reporting structure. The data generated by Galaxy is visualized interactively and provides capability for user-downloads of pre-packaged datasets and publication-quality figures for further analysis. Users can alternatively download STENCIL-analyzed data in a format compatible with analysis programs such as PRISM and Microsoft Excel. While STENCIL supports data visualization and analysis from any number of heterogenous sources, we strongly recommend using STENCIL in conjunction with a Galaxy server due to its history of supporting robust reproducible research and existing support for GDPR-compatible privacy.

Future development of STENCIL, in addition to generating automated reports for high-throughput experimental pipelines, will provide additional options for users to interactively work with the data to create customized reports based on user-selected samples and user-selected options for further analyses without having to leave the front-end web browser environment. A simple webform will enable users to select datasets of interest, or to upload other datasets, then select compatible analysis workflows. In addition, options for overlaying results of different samples which have been already analyzed through the sample workflow will be provided. Future improvement of STENCIL includes analyses will then be performed on remote HPC systems to generate and subsequently visualize the data and results on the fly

[39]. STENCIL will be further expanded to include quality control metrics and biological discovery workflows associated others popular genomic experiments (i.e., scRNA-seq, GWAS).

The authors welcome and encourage support and contributions by the larger life sciences community, particularly the involvement of the active Galaxy.org open-source community. Future development includes plans for sharing out this tool and conducting training to the Galaxy and other HPC research communities.

STENCIL is designed to efficiently report data analysis, generate high-quality visualizations for figures, and disseminate the corresponding data while ensuring FAIR Data practices [21]. The ability to visualize and compare samples of interest from multiple distinct workflows within the same web frame assists researchers' interpretation of results from inter-relating large-scale datasets. This coalescence of analysis should result in better evaluation of signal-versus-noise (experimental quality control) when visualizing the results. The multiple mechanisms that STENCIL provide for reporting and visualization will facilitate sharing and utilization of reproducible life science data. Reports produced by STENCIL should allow the researchers to gauge the underlying biology from the experiment in an efficient and scalable manner.

Supporting information

S1 Fig. DESeq2 ATAC-seq analysis visualized in STENCIL. (A) DESeq2 charts are visualized as dynamic charts in STENCIL. All data is generated and hosted directly from Galaxy. (B) The nivo charting library allows for on-the-fly generated of interactive plots containing hundreds of thousands of unique datapoint in seconds.
(TIFF)

Acknowledgments

The authors thank the current and former members of Pugh lab for helpful discussions and user feedback. The authors thank the bioHPC group for their generous support of the project.

Author Contributions

Conceptualization: Prashant K. Kuntala, B. Franklin Pugh, William K. M. Lai.

Data curation: Ali Nematbakhsh.

Funding acquisition: Gretta Kellogg, B. Franklin Pugh.

Methodology: Qi Sun, Ali Nematbakhsh, Prashant K. Kuntala, William K. M. Lai.

Project administration: Gretta Kellogg.

Software: Qi Sun, Ali Nematbakhsh, Prashant K. Kuntala, William K. M. Lai.

Supervision: Gretta Kellogg, B. Franklin Pugh, William K. M. Lai.

Visualization: Qi Sun, Ali Nematbakhsh, Prashant K. Kuntala, William K. M. Lai.

Writing – original draft: Qi Sun, Ali Nematbakhsh, Prashant K. Kuntala, Gretta Kellogg, William K. M. Lai.

Writing – review & editing: William K. M. Lai.

References

1. ENCODE. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012; 489 (7414):57–74. doi: 10.1038/nature11247. PMID: 22955616; PubMed Central PMCID: PMC3439153.

2. GTEx. The Genotype-Tissue Expression (GTEx) project. Nat Genet. 2013; 45(6):580–5. Epub 2013/05/30. doi: 10.1038/ng.2653. PMID: 23715323; PubMed Central PMCID: PMC4010069.

3. Roadmap Epigenomics C, Kundaje A, Meuleman W, Ernst J, Bilenky M, Yen A, et al. Integrative analysis of 111 reference human epigenomes. Nature. 2015; 518(7539):317–30. Epub 2015/02/20. doi: 10.1038/nature14248. PMID: 25693563; PubMed Central PMCID: PMC4530010.

4. Andrews S. FastQC: a quality control tool for high throughput sequence data. http://wwwbioinformaticsbabrahamacuk/projects/fastqc. 2010.

5. Wang L, Wang S, Li W. RSeQC: quality control of RNA-seq experiments. Bioinformatics. 2012; 28(16):2184–5. Epub 2012/06/30. doi: 10.1093/bioinformatics/bts356. PMID: 22743226.

6. Okonechnikov K, Conesa A, Garcia-Alcalde F. Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. Bioinformatics. 2016; 32(2):292–4. Epub 2015/10/03. doi: 10.1093/bioinformatics/btv566. PMID: 26428292; PubMed Central PMCID: PMC4708105.

7. Ward CM, To TH, Pederson SM. ngsReports: a Bioconductor package for managing FastQC reports and other NGS related log files. Bioinformatics. 2020; 36(8):2587–8. Epub 2019/12/17. doi: 10.1093/bioinformatics/btz937. PMID: 31841127.

8. Brown J, Pirrung M, McCue LA. FQC Dashboard: integrates FastQC results into a web-based, interactive, and extensible FASTQ quality control tool. Bioinformatics. 2017; 33(19):3137–9. Epub 2017/06/13. doi: 10.1093/bioinformatics/btx373. PMID: 28605449; PubMed Central PMCID: PMC5870778.

9. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, et al. Model-based analysis of ChIP-Seq (MACS). Genome Biol. 2008; 9(9):R137. Epub 2008/09/19. doi: 10.1186/gb-2008-9-9-r137. PMID: 18798982; PubMed Central PMCID: PMC2592715.

10. Liu T, Ortiz JA, Taing L, Meyer CA, Lee B, Zhang Y, et al. Cistrome: an integrative platform for transcriptional regulation studies. Genome Biol. 2011; 12(8):R83. Epub 2011/08/24. doi: 10.1186/gb-2011-12-8-r83. PMID: 21859476; PubMed Central PMCID: PMC3245621.

11. Costa-Silva J, Domingues D, Lopes FM. RNA-Seq differential expression analysis: An extended review and a software tool. PLoS One. 2017; 12(12):e0190152. Epub 2017/12/22. doi: 10.1371/journal.pone.0190152. PMID: 29267363; PubMed Central PMCID: PMC5739479.

12. Chen H, Lareau C, Andreani T, Vinyard ME, Garcia SP, Clement K, et al. Assessment of computational methods for the analysis of single-cell ATAC-seq data. Genome Biol. 2019; 20(1):241. Epub 2019/11/20. doi: 10.1186/s13059-019-1854-5. PMID: 31739806; PubMed Central PMCID: PMC6859644.

13. Yu Y, Ouyang Y, Yao W. shinyCircos: an R/Shiny application for interactive creation of Circos plot. Bioinformatics. 2018; 34(7):1229–31. Epub 2017/12/01. doi: 10.1093/bioinformatics/btx763. PMID: 29186362.

14. Su W, Sun J, Shimizu K, Kadota K. TCC-GUI: a Shiny-based application for differential expression analysis of RNA-Seq count data. BMC Res Notes. 2019; 12(1):133. Epub 2019/03/15. doi: 10.1186/s13104-019-4179-2. PMID: 30867032; PubMed Central PMCID: PMC6417217.

15. Kim J, Yoon S, Nam D. netGO: R-Shiny package for network-integrated pathway enrichment analysis. Bioinformatics. 2020; 36(10):3283–5. Epub 2020/02/23. doi: 10.1093/bioinformatics/btaa077. PMID: 32083639.

16. Ouyang JF, Kamaraj US, Cao EY, Rackham OJL. ShinyCell: Simple and sharable visualisation of single-cell gene expression data. Bioinformatics. 2021. Epub 2021/03/29. doi: 10.1093/bioinformatics/btab209. PMID: 33774659.

17. Zhao Y, Federico A, Faits T, Manimaran S, Segre D, Monti S, et al. animalcules: interactive microbiome analytics and visualization in R. Microbiome. 2021; 9(1):76. Epub 2021/03/30. doi: 10.1186/s40168-021-01013-0. PMID: 33775256; PubMed Central PMCID: PMC8006385.

18. Hossain S. Visualization of Bioinformatics Data with Dash Bio. 2019:126–33. https://doi.org/10.25080/Majora-7ddc1dd1-012

19. R-studio. Pricing 2021. Available from: https://www.rstudio.com/pricing/.

20. Plotly. Pricing 2021. Available from: https://plotly.com/get-pricing/.

21. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016; 3:160018. Epub 2016/03/16. doi: 10.1038/sdata.2016.18. PMID: 26978244; PubMed Central PMCID: PMC4792175.

22. Gronemeyer H, Souren NY. Big Data: The good, the bad and the ugly. Int J Cancer. 2021; 148(12):2870–1. Epub 2021/02/03. doi: 10.1002/ijc.33466. PMID: 33529345.

23. Goecks J, Nekrutenko A, Taylor J, Galaxy T. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol. 2010; 11(8):R86. Epub 2010/08/27. doi: 10.1186/gb-2010-11-8-r86. PMID: 20738864; PubMed Central PMCID: PMC2945788.

24. Gruning B, Chilton J, Koster J, Dale R, Soranzo N, van den Beek M, et al. Practical Computational Reproducibility in the Life Sciences. Cell Syst. 2018; 6(6):631–5. Epub 2018/06/29. doi: 10.1016/j.cels.2018.03.014. PMID: 29953862; PubMed Central PMCID: PMC6263957.

25. Blankenberg D, Von Kuster G, Bouvier E, Baker D, Afgan E, Stoler N, et al. Dissemination of scientific software with Galaxy ToolShed. Genome Biol. 2014; 15(2):403. Epub 2014/07/09. doi: 10.1186/gb4161. PMID: 25001293; PubMed Central PMCID: PMC4038738.

26. Sloggett C, Goonasekera N, Afgan E. BioBlend: automating pipeline analyses within Galaxy and CloudMan. Bioinformatics. 2013; 29(13):1685–6. Epub 2013/05/01. doi: 10.1093/bioinformatics/btt199. PMID: 23630176; PubMed Central PMCID: PMC4288140.

27. Tao Y, Liu Y, Friedman C, Lussier YA. Information Visualization Techniques in Bioinformatics during the Postgenomic Era. Drug Discov Today Biosilico. 2004; 2(6):237–45. Epub 2004/11/01. https://doi.org/10.1016/S1741-8364(04)02423-0 PMID: 20976032; PubMed Central PMCID: PMC2957900.

28. Nusrat S, Harbig T, Gehlenborg N. Tasks, Techniques, and Tools for Genomic Data Visualization. Comput Graph Forum. 2019; 38(3):781–805. Epub 2019/11/27. doi: 10.1111/cgf.13727. PMID: 31768085; PubMed Central PMCID: PMC6876635.

29. Harrison KJ, Crecy-Lagard V, Zallot R. Gene Graphics: a genomic neighborhood data visualization web application. Bioinformatics. 2018; 34(8):1406–8. Epub 2017/12/12. doi: 10.1093/bioinformatics/btx793. PMID: 29228171; PubMed Central PMCID: PMC5905594.

30. Qu Z, Lau CW, Nguyen QV, Zhou Y, Catchpoole DR. Visual Analytics of Genomic and Cancer Data: A Systematic Review. Cancer Inform. 2019; 18:1176935119835546. Epub 2019/03/21. doi: 10.1177/1176935119835546. PMID: 30890859; PubMed Central PMCID: PMC6416684.

31. Rossi MJ, Kuntala PK, Lai WKM, Yamada N, Badjatia N, Mittal C, et al. A high-resolution protein architecture of the budding yeast genome. Nature. 2021; 592(7853):309–14. Epub 2021/03/12. doi: 10.1038/s41586-021-03314-8. PMID: 33692541; PubMed Central PMCID: PMC8035251.

32. Lai WKM, Mariani L, Rothschild G, Smith ER, Venters BJ, Blanda TR, et al. A ChIP-exo screen of 887 PCRP transcription factor antibodies in human cells. bioRxiv. 2021:2020.06.08.140046. https://doi.org/10.1101/gr.275472.121 PMID: 34426512

33. mui-datatables. https://githubcom/gregnb/mui-datatables. 2021.

34. Robinson JT, Thorvaldsdóttir H, Turner D, Mesirov JP. igv.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV). bioRxiv. 2020:2020.05.03.075499. https://doi.org/10.1101/2020.05.03.075499

35. Sandhu R, Coynek E, Feinsteink H, Youmank C. Role-Based Access Control Models. IEEE Computer. 1996; 29(2):38–47.

36. Galaxy T. GDPR Compliance Documentation 2021. Available from: https://usegalaxy-eu.github.io/gdpr/.

37. Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling P, et al. Pegasus, a workflow management system for science automation. Future Generation Computer Systems. 2015; 46:17–35. https://doi.org/10.1016/j.future.2014.10.008.

38. Venkataraman A, Yang K, Irizarry J, Mackiewicz M, Mita P, Kuang Z, et al. A toolbox of immunoprecipitation-grade monoclonal antibodies to human transcription factors. Nat Methods. 2018. doi: 10.1038/nmeth.4632. PMID: 29638227.

39. Towns J, Cockerill T, Dahan M, Foster I, Gaither K, Grimshaw A, et al. XSEDE: Accelerating Scientific Discovery. Computing in Science & Engineering. 2014; 16(5):62–74.