BMC Bioinformatics

SOFTWARE

Open Access

Closha 2.0: a bio-workflow design system for massive genome data analysis on high performance cluster infrastructure

Gunhwan Ko[1†], Pan-Gyu Kim[1†], Byung-Ha Yoon[1], JaeHee Kim[1], Wangho Song[1], IkSu Byeon[1], JongCheol Yoon[1], Byungwook Lee[1*] and Young-Kuk Kim[2*]

†Gunhwan Ko and Pan-Gyu Kim contributed equally.

*Correspondence: bulee@kribb.re.kr; ykim@cnu.ac.kr

[1] Korean Bioinformation Center (KOBIC), KRIBB, 125 Gwahangno, Yuseong-gu, Daejeon 34141, Korea
[2] Department of Bio-AI Convergence, Chungnam National University, Daejeon 34134, Korea

Abstract

Background: The explosive growth of next-generation sequencing data has resulted in ultra-large-scale datasets and significant computational challenges. As the cost of next-generation sequencing (NGS) has decreased, the amount of genomic data has surged globally. However, the cost and complexity of the computational resources required continue to be substantial barriers to leveraging big data. A promising solution to these computational challenges is cloud computing, which provides researchers with the necessary CPUs, memory, storage, and software tools.

Results: Here, we present Closha 2.0, a cloud computing service that offers a user-friendly platform for analyzing massive genomic datasets. Closha 2.0 is designed to provide a cloud-based environment that enables all genomic researchers, including those with limited or no programming experience, to easily analyze their genomic data. The new 2.0 version of Closha has more user-friendly features than the previous 1.0 version. Firstly, the workbench features a script editor that supports Python, R, and shell script programming, enabling users to write scripts and integrate them into their pipelines. This functionality is particularly useful for downstream analysis. Second, Closha 2.0 runs on containers, which execute each tool in an independent environment. This provides a stable environment and prevents dependency issues and version conflicts among tools. Additionally, users can execute each step of a pipeline individually, allowing them to test applications at each stage and adjust parameters to achieve the desired results. We also updated a high-speed data transmission tool called GBox that facilitates the rapid transfer of large datasets.

Conclusions: The analysis pipelines on Closha 2.0 are reproducible, with all analysis parameters and inputs being permanently recorded. Closha 2.0 simplifies multi-step analysis with drag-and-drop functionality and provides a user-friendly interface for genomic scientists to obtain accurate results from NGS data. Closha 2.0 is freely available at https://www.kobic.re.kr/closha2.

Keywords: Closha 2.0, Next-generation sequencing (NGS), Cloud computing, Bioinformatics workflow, High-performance computing (HPC), Genomic data analysis, User-friendly interface, Data transmission (GBox), Single-cell RNA sequencing (scRNA-Seq)

Background

Since the early 2000s, substantial advancements in next-generation sequencing (NGS) have led to a rapid increase in the amounts of large-scale genomic data generated [1]. The use of large sequencing datasets in research is increasing, with public repositories for raw sequencing data doubling in size every 18 months [2, 3]. Researchers need significant computational resources and complex workflows to effectively analyze extensive NGS datasets [4]. The advancement of a wide range of open-source tools fuels genomic data analysis. Many tools perform a specific task, and when used one after another, they can process large volumes of genomic data [5]. A series of bioinformatics tools that process raw sequencing data and generate interpretations from genomic data is called a pipeline or a workflow. Snakemake and Nextflow are commonly used workflow management systems that aim to reduce the complexity of creating workflows by providing a fast and user-friendly execution environment, and to create reproducible and scalable data analysis pipelines. Despite the availability of numerous computational tools and methods for genomic data analysis, genomic researchers still face challenges in installing and maintaining these tools, integrating them into workable pipelines, finding accessible computational platforms, configuring the computing environment, and performing the actual analysis [6].

To address these challenges, the cloud computing model has been suggested as a solution in genomic data analysis and continues to be widely used today [7]. Cloud computing offers genomic researchers on-demand resources by abstracting the underlying infrastructure from a service provider [8]. In the computation-as-a-service cloud model, researchers use the necessary hardware and software for data analysis as long as needed to accomplish their objectives [9]. In the field of computational biology, Galaxy is one of the most widely used free cloud computing services, offering a platform for scientific workflows, data integration, and data analysis [10]. Originally designed for genomics research, it has evolved into a versatile bioinformatics workflow management platform. Moreover, to meet the rapidly growing computational demands, commercial clouds such as Amazon Web Services (AWS), Google Cloud Platform, and Microsoft Azure are becoming indispensable for their ability to provide and manage vast amounts of computer resources [11–13]. In particular, AWS is the most widely used and well-known commercial cloud service globally, with many bioinformatics and genomics researchers using it for data analysis [14].

These platforms have provided significant insights into the technical requirements for leveraging cloud computing in genomic data analysis; however, some challenges persist. For example, As the research area of genomic data utilization expands, the downstream analysis of data has become increasingly diverse. Although cloud computing services provide various bioinformatics tools, there are a wide variety of users' analytical demands. To meet these requirements, cloud computing services now necessitate the ability to code script language such as Python, R, Sell on the fly. In addition, to analyze genomic data in the cloud server, the first step is to upload their data to the server. However, as the volume of genomic data rapidly increases, uploading such large datasets is becoming gradually challenging. Therefore, it has become essential for cloud servers to provide the capability for users to quickly upload their large volumes of data.

We developed Closha 2.0, a cloud computing platform for online biological data analysis. Users can use the Closha 2.0 workbench to upload data, design workflows, choose from curated pipelines, and easily view the analysis results on their local computer. The user-friendly software environment enables users with limited Linux experience to perform biological data analysis through simple drag-and-drop actions. Moreover, the workbench features a script editor that supports scripting languages such as Python and R, enabling advanced users to code in real-time. These scripts can be integrated as nodes in a pipeline. We also updated a high-speed data transmission solution to transfer large amounts of data more stably and reliably than the previous version, KoDS. Our cloud-based workflow management system helps users run in-house pipelines or construct a series of steps in an organized manner.

Here, a workflow represents a series of bioinformatics applications being executed, with each step involving a specific action that takes an input and produces an output. A pipeline typically emphasizes the continuous flow of items through various stages or tasks, whereas a workflow refers to the progression of an item through different status changes over time. Although a slight difference in meaning between "workflow" and "pipeline" exists, we use these two terms interchangeably in the manuscript. We aimed to address these challenges by enhancing accessibility and efficiency in genomic data analysis through the development and implementation of Closha 2.0.

Methods

Closha cloud computing model

The Closha 2.0 cloud computing service model is based on a client/server architecture. The workbench is a client program that runs on a local computer, while the Closha server provides cloud computing services, such as computing power, storage, and applications, to the workbench over a network (Fig. 1). The workbench provides a GUI interface composed of several panels that display information on the projects, file explorer, workflow canvas, application parameters, curated pipelines, list of available analysis programs, and job execution history and current progress of the user (Fig. 2).

The workbench operates on Windows, Linux, and Mac operation system. Each executable file can be downloaded from the Closha website, and for the Mac version, a separate authentication process must be completed before it can be executed. Detailed execution and authentication procedures are described in detail in the manual.

The curated pipelines on the workbench are grouped into categories. Currently, two categories are available: Bioinformatics and Machine Learning. When a user selects a pipeline, it is displayed on the canvas, and its parameters are adjusted in the application parameters panel. When a user executes a tool, its output datasets are added to the execution and history panel. The colors on the canvas show the state of the running tool. Clicking on a dataset in the panel provides a lot of information, including the tool and parameter settings used to create it.

Workflow manager

The Closha workflow management system, a workflow manager, was developed to provide a framework for creating, executing, and monitoring workflows. It efficiently

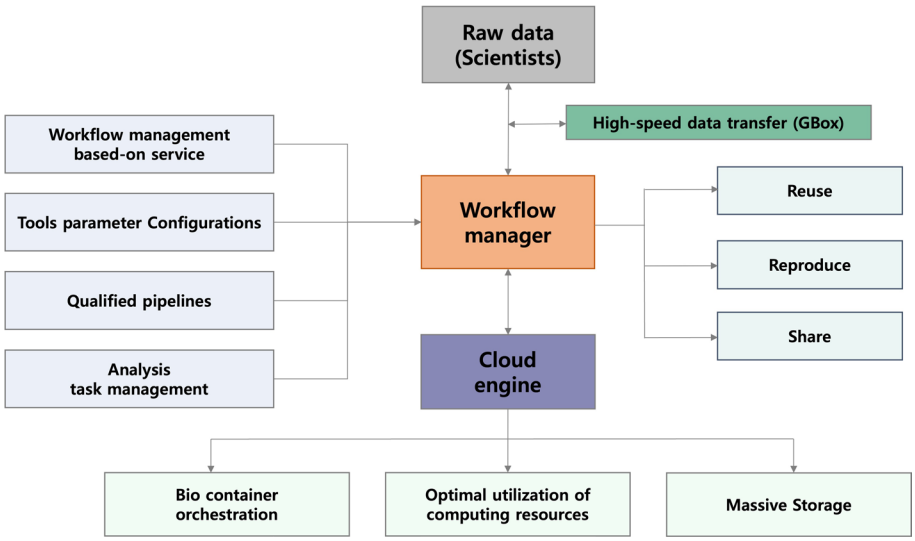

Fig. 1 The flowchart of the high-level Chosha2 cloud computing infrastructure. When users upload data to the Closha2 cloud computing server via GBox and execute a selected workflow, the workflow manager of Closha2 automatically runs the workflow. The workflow manager manages the cloud service using the cloud engine, Podman, on the Closha2 server. The completed analysis workflow is managed by the workflow manager to be stored for reuse or sharing

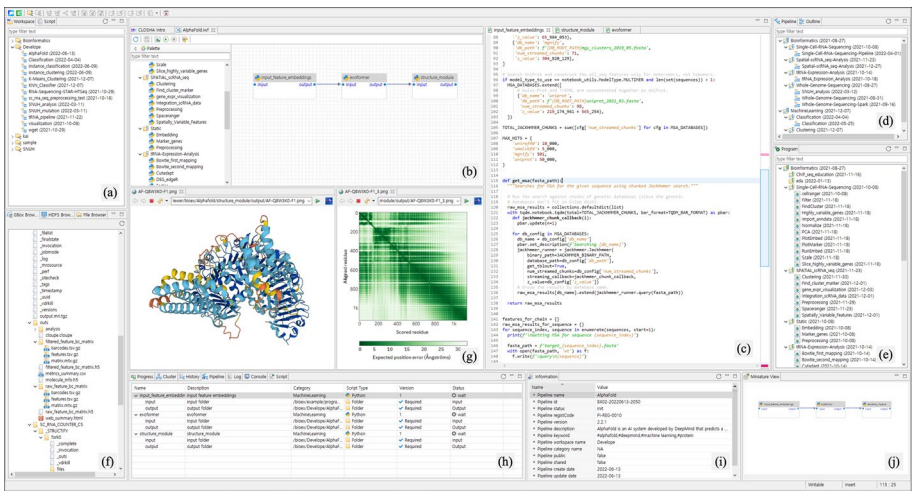

Fig. 2 Screenshot of the Closha workbench. The Closha workbench has several panels: **a** my project, **b** canvas, **c** script editor, **d** curated pipelines, **e** applications, **f** file browser, **g** viewer, **h** parameter and history, **i** miniature view

uses resources by parallelizing various steps, such as running multiple tools simultaneously, optimizing parameters, and incorporating dynamically changing reference data. The workflow manager in Closha 2.0 uses the container orchestration system Podman to automatically manage the scheduling and deployment of containers, enabling the effective utilization of the available resources. Podman is a tool used for

managing containers and images, mounting volumes into those containers, and running containers on Linux.

Workflow canvas

The workflow canvas is the main graphical user interface of the workbench, in which users can manipulate a workflow by arranging and connecting applications for biological data analysis. It provides a point-and-click interface for users to drag-and-drop tools into workflows and chain them together, enabling users without programming experience to create complex computational pipelines.

In a workflow, each step is called a node, which is classified as a start, end, or application node, based on its function. The start and end nodes are essential nodes located at the beginning and end of every workflow. Application nodes are responsible for bioinformatic analytical functions and are composed of publicly available tools provided by the Closha server. To become an application node on the workbench, a tool must first be installed on the Closha server, and then a wrapper function is attached to it to define the input and output of the tool.

The canvas provides the GUI interface for creating a new workflow with Closha nodes. Users can either use a curated pipeline or modify it to create a custom pipeline on the canvas. The canvas simplifies the creation of multi-step analyses using drag-and-drop functionality. Tool parameters can be set in the parameter panel. Workflows enable the automation and repeated running of large dataset analyses. Thus, once workflows are created, they function as tools.

Curated pipelines

The construction of a new data analysis pipeline can be challenging for beginners in computational biology. Thus, utilizing a curated or commonly used pipeline for analyzing specific types of data is recommended. The workbench offers several analysis pipelines in the curated pipeline panel, which have been tested by the Closha team. Using curated pipelines eliminates the necessity of building a new pipeline and allows researchers with no computational background to utilize optimal methods to perform bioinformatics tasks.

Coding on the workbench

Users can write scripts using languages such as Python, R, and Bash directly within the workbench. By selecting the "New Script" menu in the script panel, the workbench opens a script editor where Python, R, and Bash scripts can be entered. Users can develop and test their script, and then add it to the workflow on the fly. To register a user's script as a node, users have to enter the program's name and description, and define the input files and output directory. Programs registered as nodes can be used in the development of pipelines like Closha's public tools.

Reentrancy function

Since computational workflows have multiple steps, any errors or manual intervention that interrupt the operation of a pipeline require it to be restarted. This leads to the

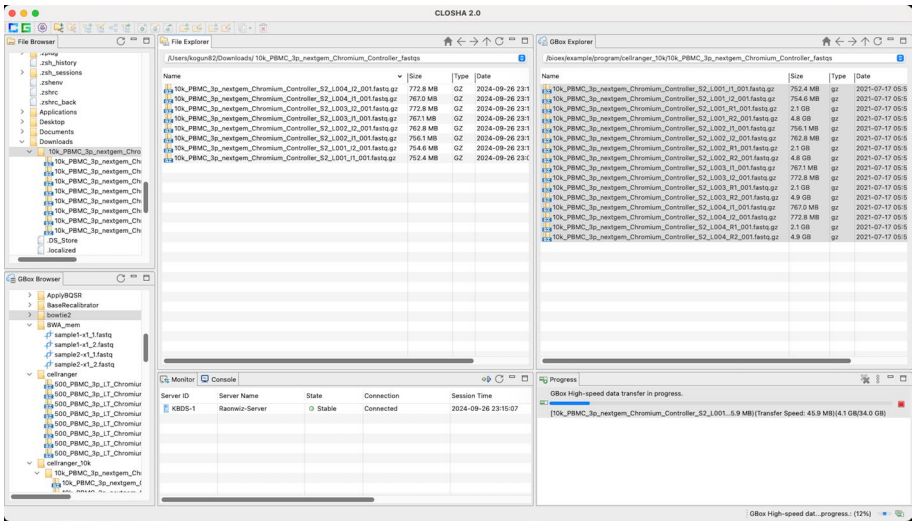

Fig. 3 Screenshot of the GBox. GBox enables quick uploads and downloads of large-scale data, ranging from millions of small files to very large datasets

Table 1 Comparison of FTP/HTTP and GBox

Features	FTP/HTTP	GBox
Transport protocols	Data transfer based on standard protocols	Optimized data transfer using a multi-channel dedicated protocol
Bandwidth management	Potential network congestion due to insufficient bandwidth management	Minimize network congestion through efficient bandwidth utilization
Transfer speed	Relatively slow speed when transferring large data volumes	High-speed transfer with an optimized transmission algorithm

re-computation of already computed results and the unnecessary use of computational resources.

The Closha workflow manager allows for reentrancy by handling such events. This enables users to resume a pipeline from the last step that was successfully executed, rather than starting the entire process all over again, in case of disruption. Reentrancy also reduces the necessity of recalculating data processing tasks. To accomplish this, the workflow manager stores the intermediate results and data files.

Furthermore, the workflow manager enables users to independently execute each step within the workflow. This allows the verification of whether each step can produce the desired result before running the entire workflow. It also enables the execution of some parts of the entire workflow. Although reentrancy could lead to increased storage demands when generating intermediate files, it saves considerable time and computational resources.

GBox for high-speed data transfer

To effectively utilize the analysis service on Closha 2.0, users need to transfer the large sequence data files from their local computer to the remote Closha storage space. To support this, we developed GBox, a big data transfer tool capable of high-speed and

large-scale data file transfer (Fig. 3). GBox has features that distinguish it from conventional FTP/HTTP protocols in terms of transmission protocol, bandwidth management, and transfer speed (Table 1). These features enable reliable file transfers that are roughly ten times faster than traditional FTP or HTTP methods, all accomplished using an easy-to-use interface. Users can upload sequence data files or download the resulting files with a size of up to 10 TB to their Closha storage using GBox.

GBox has advantages over KoDS of Closha 1.0. First, we upgraded the transmission engine, which dynamically allocates network resources according to real-time conditions when multiple users are transmitting data, enabling simultaneous data transfers for numerous users. Second, GBox shows real-time transmission status information, which helps avoid network congestion during data transfers. Third, the transmission engine employed by GBox enhances data security by offering secure file transfer capabilities through data encryption. Lastly, GBox includes several convenient functions for file management, such as moving, copying, and deleting. It also allows users to set minimum and maximum network resource allocations during data transfers, thereby optimizing the use of network resources according to the user's network environment.

Cloud computing architecture and environment

The Closha pipeline operates on a cluster of 33 high-performance computing nodes, each equipped with 1,188 CPU cores and 12.4 TB of memory. The primary storage system is constructed using the Lustre parallel file system to efficiently manage and store extensive amounts of data, providing fast throughput and enabling the handling of numerous analytical tasks simultaneously with great scalability. Each analysis pipeline is allocated 6 CPU cores and 64 GB of memory by default. Users can also select a GPU server for running a machine learning pipeline or a large memory server with 12 TB of memory. Each user can utilize up to 10 TB of storage space. We are able to provide additional hardware upon request for users who require more computing hardware.

Users can run multiple analysis pipelines in parallel using the Closha cloud computing environment. Furthermore, Podman offers a separate execution environment to coordinate the different execution environments required for various analytical tools, allowing users to conduct data analysis in a uniform environment.

The analysis service of Closha 2.0 uses a NoSQL database that combines Apache HBase and Phoenix to efficiently store and manage large-scale logs and metadata generated during analysis execution. This combination allows for efficient metadata storage and allows users to easily make changes to the data model by storing data in JSON format. A real-time data search and the monitoring of the execution status of the analysis pipeline are included, enabling quick retrieval and analysis of the data collected (Fig. 4).

Results

Comparison between Closha 1.0 and Closha 2.0

The Closha 2.0 workbench includes more user-friendly features than the previous 1.0 version (Table 2). Firstly, it operates based on containers. Container technology offers process-level isolation, minimizing interference between applications. Thus, the container-based environment allows the execution of bioinformatics tools in an independent environment, thereby preventing dependency issues and version

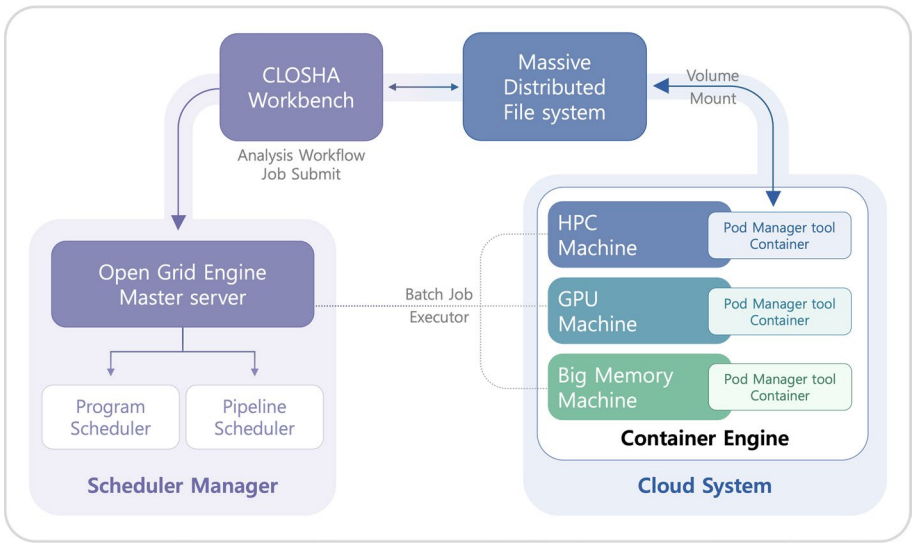

Fig. 4 Overview of the container processing step. In the Closha 2.0 cloud service, we support a container-based secure, and independent analysis environment to easily support analysis tools and user script code execution environments. Users can perform analysis and verify the results using registered Podman images

Table 2 Comparison of Closha 1.0 and Closha 2.0

	Closha 1.0	Closha 2.0
Architecture	Operation-based environment	Container-based virtualization
Script editing	X	O
File transfer tool	KoDS	GBox
Reentrancy function	X	O
Providing HW	Only one CPU core	Users can choose between one or six CPU cores. Users also select a GPU server or a large memory server

conflicts among analytical tools. We can address the issue of conflicts between different versions of the same tool. Compared to Closha 1.0, Closha 2.0 does not support Hadoop-based bioinformatics tools because most of them are no longer updated. Second, the workbench includes a script editor that supports Python, R, and shell script programs. This enables users to directly write scripts and integrate them into pipelines for data analysis. Third, Closha 2.0 provides a reentrancy function that allows users to execute each step of a pipeline individually. Therefore, users test applications in each step and adjust the parameters for each application to obtain the desired results. Fourth, we updated a high-speed file transfer tool from KoDS to GBox. It supports more stable and secure file transmission. Lastly, users can choose between one or six CPU cores. In Closha 1.0, only one CPU core was allocated for each user's job. Users also select a GPU server for running a machine learning pipeline or a large memory server for assembling genome sequencing reads.

Using Closha workbench

To use the Closha cloud computing service, users must create an account on the Closha homepage. Users can download the compressed file of the Closha workbench from the homepage and run it without having to go through an installation process. Closha workbench is available for free and is open-source software licensed under the GNU General Public License (GPL) version 3 or later. The Closha workbench is a multi-platform software that can be operated on Microsoft Windows (Windows 10/11), Macintosh OSX (OSX 14.2.1), and Linux (Ubuntu 20.3, Linux Rocky) operating systems. Using a local computer with the following specifications is recommended: a CPU of 2.8 GHz or higher, 2 cores, and a RAM of 8 GB or higher.

Single-cell RNA sequencing pipeline

Closha workbench 2.0 provides a complete pipeline for analyzing large-scale single-cell RNA sequencing (scRNA-Seq) data in the curated pipeline panel. This pipeline enables the analysis of the gene expression of individual cells, making it easier to investigate cellular differences and gain a better understanding of cellular roles and activities. Figure 5 shows an overview of the scRNA-Seq analysis pipeline.

The scRNA-Seq data analysis pipeline consists of four main steps: data preprocessing, dimensionality reduction, clustering analysis, and visualization. In the preprocessing step, cells with low gene expression or insufficient reads are filtered out, and then the data is standardized before further analysis. The high-dimensional scRNA-Seq data is then subjected to dimensionality reduction using PCA, which minimizes information loss [15]. The PCA-reduced data is clustered using the Louvain algorithm, a graph-based method that groups cells according to their similarities. Finally, the clustered data is depicted in a lower-dimensional space through nonlinear dimensionality reduction methods, such as UMAP or t-SNE [16].

In the visualization step, the most important factor is to maintain the complex organization of the data while improving its visual clarity. To achieve this, the

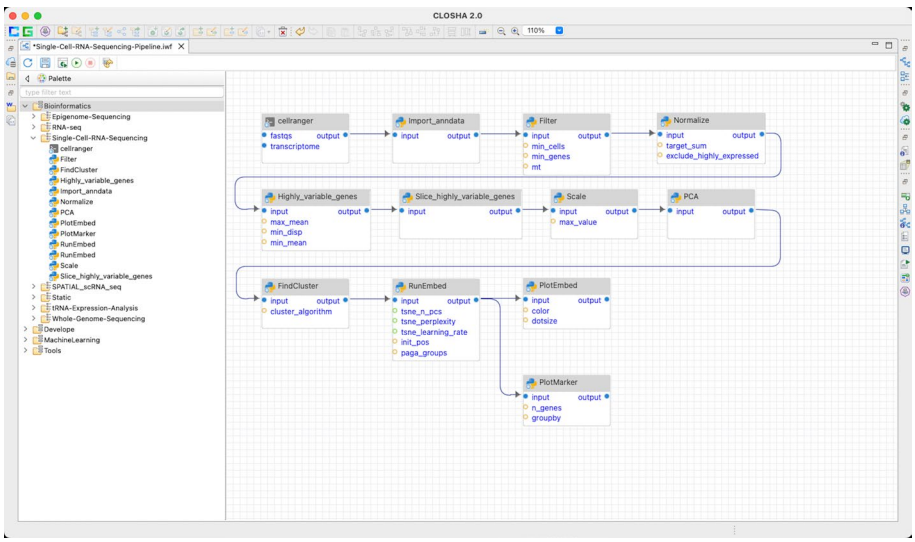

Fig. 5 The scRNA-Seq analysis pipeline implemented on the Closha workbench canvas

scRNA-Seq pipeline includes visualizations using the UMAP, t-SNE, and FIt-SNE algorithms for comparative analysis to identify similarities and differences between cells from different algorithmic viewpoints. UMAP successfully preserves the general organization and clustering accuracy of the data, whereas t-SNE focuses on the specific structures within the data. FIt-SNE, an enhanced version of t-SNE, maintains its accuracy while also improving computational efficiency [17].

To evaluate the performance of the scRNA-Seq pipeline in the Closha workbench, we used scRNA-Seq data from 49,384 cells extracted from human lung tissue provided by the Human Cell Atlas project's Mould-Human-10 × 3pv3, totaling approximately 357 GB of data (https://explore.data.humancellatlas.org/projects/272b7602-66cd-4b02-a86b-2b7c9c51a9ea). The execution of the scRNA-Seq pipeline provided the baseline runtime speed. Closha assigned six CPU cores and 64 GB of memory for a single scRNA-Seq job. Ultimately, it took approximately 47 h to complete the data analysis, with most of the time spent in the Cellranger analysis stage.

Creating a new pipeline

The Closha workbench allows users to design custom pipelines for analyzing their data directly on the canvas. Users can initiate the creation of a new analysis pipeline in Closha by selecting the "New Pipeline" option from the top menu, providing a name and description for the pipeline, and choosing the desired type of analysis pipeline.

When selecting "new analysis pipeline design" in the project type, users will only see the [Start] and [End] nodes on the canvas. Users can drag and drop the applications from the list of available applications located on the right of the canvas. After placing an application on the canvas, when the user hovers the mouse over the edge of the application node icon, a connection mark that can be drawn to the next node is displayed. Starting from the mark, the connector must be dragged until the icon of the next node is connected, after which it becomes translucent. Users can use this method to connect to the start node, the application nodes, and the end node to conduct the analysis.

Next, users can adjust the parameter values by selecting the "Set Parameters" button on the toolbar before running the pipeline project. Default parameter values are automatically assigned upon the creation of a new project. Users can adjust these to meet the necessary conditions for configuring and examining their input data. The user can link their files to Closha by clicking on the "File Selection" icon, which opens a window where they can choose an input file, either personal or commonly used data, from the list. The output file path is automatically set as a sub-path of the project when configuring the input data. Finally, the analysis pipeline is run, with a notification indicating that the analysis has begun. The progress of the project is shown in real-time through three modes: "finished," "in progress," and "pending."

Users can view the resulting files by selecting the "result" icon in the menu bar, and then downloading them to their local computer by clicking the "Download" button located in the bottom menu. This enables GBox to be utilized for efficient high-speed transmission. Closha enables users to access files in different formats, such as text, HTML, and PNG, directly on the screen without the need to download them.

Discussion

The significant decrease in the cost of NGS techniques over the past decade has dramatically reshaped genome research and led to its rapid adoption in biological research. Nowadays, a massive amount of data can be quickly generated using NGS platforms. With the exponential increase in the volume and complexity of the NGS data generated, cluster or high-performance computing (HPC) systems are essential for the analysis of large amounts of NGS data. However, the associated costs, including the infrastructure itself and the maintenance personnel, are usually prohibitive for small institutions or laboratories.

Cloud-based applications and resources have been specifically developed to address the computational challenges of working with very large volumes of data generated using NGS technology. Cloud computing has changed how we manage computational resources. Moreover, it is also increasingly changing how large computational resources are organized and how genomics scientists collaborate and deal with vast genome data sets.

In the field of biological data analysis, the need for cloud computing has emerged among both novice and experienced researchers. Beginners frequently lack knowledge about which tools or analysis pipelines are the most adequate for examining their data. Experienced data analysis experts not only set up workspaces but also prefer to write code directly within the workspace for real-time data analysis.

In addition, as the types of NGS applications increase, the specific goals of an experiment become increasingly diverse. Although various bioinformatics tools were developed for processing genomic data, there are still a wide variety of users' analytical demands to produce a publishable result. To meet these requirements, tools like Jupyter notebook or RStudio can be integrated into Galaxy and provide running ad hoc scripts within the Galaxy instance.

Closha 2.0 has a user-friendly feature that allows for easy script writing within the workbench to meet these requirements. Without the need for integration with external script editors such as Jupyter or Rstudio, Closha 2.0 has directly implemented this functionality within the workbench. Researchers can write scripts in languages such as Python, R, and Bash on-the-fly for downstream analysis, as well as using the bioinformatics tools. This function will provide significant flexibility for the analysis of genomic data.

With the advancement of data analysis technologies, many researchers are developing a variety of analysis pipelines tailored to their own data and research objectives, and are sharing them publically. In this flood of analysis pipelines, it has become increasingly challenging for beginners to select an appropriate pipeline for their studies. In the face of these practical challenges, we are developing best practice pipelines for different data types and curating them for optimal use. Furthermore, we keep installing public bioinformatics tools in the Closha server that will enable researchers to easily construct pipelines using the provided tools.

Since many users transfer their data to the cloud server and analyze it, security becomes crucial in cloud environments. Storing users' applications and data in the cloud is more secure than on-premises, especially when considering sensitive data such as personal information. Closha 2.0 offers enhanced security by rigorously isolating users' data

and analytical tasks. Each user performs analyses in an independent working environment, with strict restrictions on data access from other users. Furthermore, to ensure the secure transmission of sensitive data, all user data is encrypted during transit and access rights are minimized to the owning user.

We developed Closha cloud computing services. Closha allows users to create multi-step analyses using drag-and-drop functionality and to modify the parameters of pipeline tools. Closha provides an easy-to-use cloud workbench that reflects these diverse demands and will continue to develop and provide curated pipelines.

Conclusions

Closha 2.0 represents a significant advancement in cloud-based bioinformatics, addressing the computational challenges faced by the explosive growth of NGS data generated. By offering a user-friendly interface with drag-and-drop functionality, an integrated script editor, and container tool management, Closha 2.0 empowers both novice and experienced researchers to efficiently analyze their large-scale genomic datasets. The ability of the platform to execute and adjust each pipeline step individually, coupled with the high-speed data transmission solution GBox, ensures robust and flexible data processing. As genomic research continues to evolve, Closha 2.0 stands out as a powerful tool for simplifying complex bioinformatics workflows, making advanced genomic analysis more accessible and reproducible in the scientific community. This platform not only simplifies complex workflows but also ensures reproducibility and scalability, underscoring its relevance and importance in modern biological research. The source code, system architecture, and design information for Closha 2.0 have been made publicly available on GitHub (https://github.com/kobic-dev/closha.git). The GitHub repository includes the source code along with documentation that provides a detailed explanation of Closha 2.0's core features and architecture.

Availability and requirements

Project name: Closha 2.0

Project home page: https://www.kobic.re.kr/closha2

Operating system(s): Platform independent.

Programming language: Java.

Other requirements: The Closha workbench is operated on Microsoft Windows (Windows 10/11), Macintosh OSX (OSX 14.2.1), and Linux (Ubuntu 20.3, Linux Rocky) operating systems.

License: GNU General Public License (GPL) version 3 or later.

Any restrictions to use by non-academics: none.

Abbreviations

AWS Amazon Web Services
GPL General Public License
HPC High-Performance Computing
NGS Next-generation sequencing

Acknowledgements

We thank the Korea BioData Station (K-BDS) at Korean Bioinformation Center (KOBIC) for sharing their data (ID: KAP230705).

Author contributions
GK and PK launched the Closha project and developed the cloud computing service. BY, JK, WS, IB, and JY were responsible for development of the web interface and the back-end cloud system. BL and KY supervised the project. GK and BL wrote the draft of the manuscript. All authors read and approved the final manuscript.

Funding
This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No. RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University)]. Additionally, publication costs were funded by the Korean Ministry of Science and Technology (under Grant No. 2020M3A9I6A01036057).

Availability of data and materials
The Closha 2.0 program is freely available at https://www.kobic.re.kr/closha2. The Cloahs2 program includes the GBox tool. The source code, system architecture, and design information for Closha 2.0 can be downloaded from GitHub (https://github.com/kobic-dev/closha.git). The GitHub repository includes the source code along with documentation that provides a detailed explanation of Closha 2.0's core features and architecture. The scRNA-Seq data provided by the Human Cell Atlas project's Mould-Human-10×3pv3 is used to evaluate the performance of the scRNA-Seq pipeline in the Closha workbench. The dataset is available at the Human Cell Atlas Data Explorer (https://explore.data.humancellatlas.org/projects/272b7602-66cd-4b02-a86b-2b7c9c51a9ea).

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 24 June 2024 Accepted: 21 October 2024
Published online: 12 November 2024

References

1. Satam H, Joshi K, Mangrolia U, Waghoo S, Zaidi G, Rawool S, Thakare RP, Banday S, Mishra AK, Das G, et al. Next-generation sequencing technology: current trends and advancements. Biology (Basel). 2023;12(7):997.
2. Qian T, Zhu S, Hoshida Y. Use of big data in drug development for precision medicine: an update. Expert Rev Precis Med Drug Dev. 2019;4(3):189–200.
3. Ko G, Lee JH, Sim YM, Song W, Yoon BH, Byeon I, Lee BH, Kim SO, Choi J, Jang I, et al. KoNA: Korean nucleotide archive as a new data repository for nucleotide sequence data. Genom Proteom Bioinform. 2024;22(1):qzae017.
4. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol. 2019;15(6): e8746.
5. Ji F, Sadreyev RI. RNA-seq: basic bioinformatics analysis. Curr Protoc Mol Biol. 2018;124(1): e68.
6. Ko G, Kim PG, Cho Y, Jeong S, Kim JY, Kim KH, Lee HY, Han J, Yu N, Ham S, et al. Bioinformatics services for analyzing massive genomic datasets. Genomics Inform. 2020;18(1): e8.
7. Stein LD. The case for cloud computing in genome informatics. Genome Biol. 2010;11(5):207.
8. Afgan E, Baker D, Coraor N, Goto H, Paul IM, Makova KD, Nekrutenko A, Taylor J. Harnessing cloud computing with Galaxy Cloud. Nat Biotechnol. 2011;29(11):972–4.
9. Mrozek D. A review of Cloud computing technologies for comprehensive microRNA analyses. Comput Biol Chem. 2020;88: 107365.
10. The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update. *Nucleic Acids Res* 2024.
11. Mora-Márquez F, Vázquez-Poletti JL. López de Heredia U: NGScloud2: optimized bioinformatic analysis using Amazon Web Services. PeerJ. 2021;9: e11237.
12. Sivagnanam S, Gorman W, Doherty D, Neymotin SA, Fang S, Hovhannisyan H, Lytton WW, Dura-Bernal S. Simulating large-scale models of brain neuronal circuits using google cloud platform. Pearc20. 2020;2020:505–9.
13. Truong L, Ayora F, D'Orsogna L, Martinez P, De Santis D. Nanopore sequencing data analysis using Microsoft Azure cloud computing service. PLoS ONE. 2022;17(12): e0278609.
14. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nat Methods. 2021;18(10):1161–8.
15. Linderman GC. Dimensionality reduction of single-cell RNA-Seq data. Methods Mol Biol. 2021;2284:331–42.
16. Ujas TA, Obregon-Perko V, Stowe AM. A guide on analyzing flow cytometry data using clustering methods and nonlinear dimensionality reduction (tSNE or UMAP). Methods Mol Biol. 2023;2616:231–49.
17. Kobak D, Berens P. The art of using t-SNE for single-cell transcriptomics. Nat Commun. 2019;10(1):5416.

Publisher's Note