BMC Bioinformatics

SOFTWARE

Open Access

BioLake: an RNA expression analysis framework for prostate cancer biomarker powered by data lakehouse

Qiaowang Li[1], Yaser Gamallat[2,3,6], Jon George Rokne[1], Tarek A. Bismar[2,3,6,7] and Reda Alhajj[1,4,5*]

*Correspondence:
alhajj@ucalgary.ca

[1] Department of Computer Science, University of Calgary, Calgary, AB, Canada
[2] Department of Pathology and Laboratory Medicine, University of Calgary, Calgary, AB, Canada
[3] Arnie Charbonneau Cancer Institute and Tom Baker Cancer Center, Calgary, AB, Canada
[4] Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey
[5] Department of Health Informatics, University of Southern Denmark, Odense, Denmark
[6] Departments of Oncology, Biochemistry and Molecular Biology, University of Calgary, Calgary, AB, Canada
[7] Prostate Cancer Centre, Calgary, AB, Canada

Abstract

Biomedical researchers must often deal with large amounts of raw data, and analysis of this data might provide significant insights. However, if the raw data size is large, it might be difficult to uncover these insights. In this paper, a data framework named BioLake is presented that provides minimalist interactive methods to help researchers conduct bioinformatics data analysis. Unlike some existing analytical tools on the market, BioLake supports a wide range of web-based bioinformatics data analysis for public datasets, while allowing researchers to analyze their private datasets instantly. The tool also significantly enhances result interpretability by providing the source code and detailed instructions. In terms of data storage design, BioLake adopts the data lakehouse architecture to provide storage scalability and analysis flexibility. To further enhance the analysis efficiency, BioLake supports online analysis for custom data, allowing researchers to upload their own data via a designed procedure without waiting for server-side approval. BioLake allows a one-time upload of custom data of up to 500 MB to ensure that researchers avoid issues with data being too large for upload. In terms of the built-in dataset, BioLake applies reactive continuous data integration, helping the analysis pipeline to get rid of most preprocessing steps. The only pre-built-in dataset of BioLake in the first public version is TCGA-PRAD mRNA expression data for prostate cancer research, which is the primary focus of the development team of BioLake. In summary, BioLake offers a lightweight online tool to facilitate bioinformatic mRNA data analysis with the support of custom online data processing.

Keywords: Parallel computing, Data lakehouse, Expression analysis, Data visualization, Prostate cancer

Introduction

Bioinformatics data analysis research is an interdisciplinary field covering computer science, statistics and biomedical science. The field has recently experienced a vast increase in the quantity of available data due to the rapid advances in sequencing technologies (Next Generation Sequencing or NGS) [1]. Numerous data sources, including DNA sequencing, RNA sequencing, electronic medical records, and time series from medical devices, have further enabled biomedical companies to collect detailed information

about patients and diseases. These data sources are often joined against public datasets such as the UK Biobank [2] which holds sequencing information and medical records for 500,000 individuals [3].

Modern research in this field can usually be divided into two main categories, namely data extraction and analysis of results. The data extraction component is mostly performed by researchers from the field of biomedicine that interact with patients and collect raw data. The analysis component is performed by researcher in bioinformatics with expertise in computational science and statistics. The challenge lies in ensuring that the data is presented in a format that is easily interpretable by all researchers.

One of the other challenges is to present vast amounts of collected data in an human-understandable form. In bioinformatics, one of the easiest ways to alleviate this problem is to collaborate with researchers in computational science or statistics who can transform the raw data into a format that can be more easily interpreted by biomedical researchers. However, this approach makes the bioinformatics research process relatively inefficient since the researchers from computational science may not be able to quickly understand the requirements of the biomedical researchers. In addition, the collaboration requires additional time and effort to discuss the requirements and to wait for the results. This paper therefore proposes an alternative way for the biomedical researchers to conduct the data analysis and interpret result in order to alleviate this problem. That is, BioLake is introduced as an interactive web-based framework powered by a data lakehouse architecture that enables the enhancement and facilitation of raw data processing and analysis. The researchers can obtain desired results by following simple instructions in BioLake, even if they have no prior data analysis experience and computer science background.

BioLake consists of a data analysis layer and a data storage layer. Figure 1 shows the overall framework design which includes a facility for users to access BioLake through

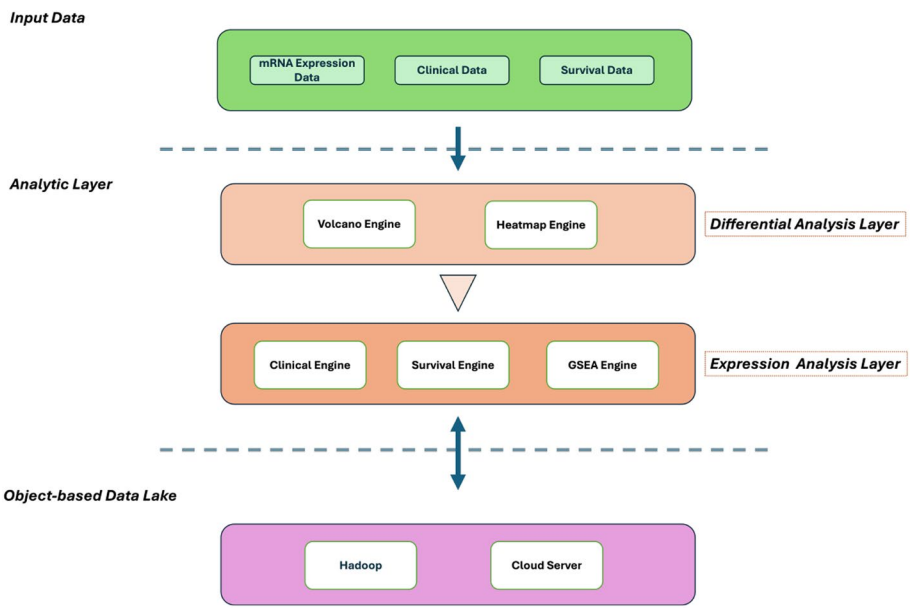

Fig. 1 Overall framework design

major browsers using a public URL. The data analysis part is handled by virtual machines with the help of several processing engines to be described in detail in Sect. 3. The data storage layer has adopted a data lakehouse architecture, a data management design based on low cost and directly-accessible storage that also provides traditional analytical database management and performance features [4]. Section 4 details this design and analyzes this design from a practical and flexible point of view. Section 5 presents a case study to demonstrate the usability of BioLake.

The primary contribution of BioLake is to offer a lightweight, scientific online web tool designed to facilitate bioinformatic mRNA data analysis. It eliminates the need for specialized expertise in software configuration, making it accessible to a broader audience. Additionally, BioLake supports custom online analyses, allowing users to process their data efficiently and without delays.

Background and related work

The field of biology is experiencing an exponential growth in available data. This growth has had the unfortunate effect that the extraction of the knowledge inherent in the data is not keeping pace with the data growth. The reason for this is that processing and extracting information from the data requires large volumes of processing guided by biologists. These biologists have to be conversant with computing and computational methods so that they can use the analysis tools effectively. Unfortunately, most biologists are not experts in the disciplines of mathematics, statistics or computing [5]. The idea of BioLake was inspired during the phase when our prostate cancer team was analyzing the public TCGA-PRAD dataset, and we realized that most of the existing tools did not fully meet our expectations.

Recent studies [6–14] provide web-based data analysis tools to facilitate the analysis where users can get bioinformatics data visualization plots with a couple of clicks. The tools on the market can be roughly divided into three categories of online-based, docker-based and on-premised-based. It is worth noting that these categories are not mutually exclusive. For example, [15] supports online-based analysis through URL [6], while also offering a docker image and R-package for docker-based support, and on-premise support. Online-based web interfaces where users interact with the tools through public URL, with all the analysis done on the server side, are provided by [7, 8, 10, 12–14]. Docker-based tools such as [15] includes learning curves which can be quite steep, as this technology is unfamiliar to the vast majority of bioinformatical researchers. On-premised-based service expects users to set up their own local configuration and environment, which can be challenging. For example, [15] packs up the functionalities as a R-package, which provides excellent support for the researchers to conduct their analysis in a local environment.

In terms of functionality, Table 1 presents the supported features of selected online-based tools that provide similar functionality to that of BioLake. PcaDB offers users the ability to select a specific gene and data source, generating a box-plot for gene expression analysis, a survival analysis table, and various RNAseq plots. However, this tool lacks the capability for users to modify the method of group classification and utilize genes as differential conditions, which are typical steps in bioinformatics differential analysis. On the other hand, EnrichR provides Gene Set Enrichment Analysis (GSEA).

Table 1 Comparison of different approaches

Tool	Unit-gene	Diff. analysis	Clinical analysis	GSEA	Survival analysis	Out-of-box
PcaDB [10]	No	Partial	Partial	No	Partial	Yes
VolcaNoseR [8]	Yes	Yes	No	No	No	No
SRplot [12]	Yes	Yes	Yes	Yes	Yes	No
GEPIA [13]	Yes	Partial	Partial	No	Partial	Partial
EnrichR [7]	Yes	No	No	Yes	No	No
Linkedomics [14]	Yes	Yes	Yes	Yes	Yes	Partial
BioLake	Yes	Yes	Yes	Yes	Yes	Yes

However, it requires users to manually input sorted gene sets, significantly reducing its practicality. Moreover, VolcaNoseR and SRplot limits user inputs to maximum 200 MB. Consequently, these tools may not be able to analyze commonly used public datasets like TCGA. Lastly, while GEPIA and linkedomics offer the most comprehensive functionality, they do not support the use of clinical data for expression analysis. In terms of framework flexibility, we evaluate all tools and present the results in the last column of Table 1. We define this tool as an "out of the box" tool if it accepts user-defined hyperparameters and provide users with the last layer data for further refinement, where the last layer data refers to the data directly used for final result generation. For instance, both differential analysis engines provide the data directly utilized for generating the heatmap and volcano plot, enabling researchers to apply statistical analysis, such as identifying the similarity of top-ranking genes. The clinical and survival engine returns structured tabular data, where the label format and distribution can be adjusted by selecting the value type for candidate action. All of the last layer data is stored in CSV format to provide straightforward visualization and easy access. The linkedomics tool is defined as partial "out of the box", since it does not accept user-defined hyperparameters. And users cannot apply custom modification by linkedomics, as it does not provide the last layer data. In addition, linkedomics does not support online analysis for custom dataset provided by the user. In such cases, users need to contact the authors of linkedomics for the analysis of this type of data, which can be time consuming.

Implementation

A web-based framework designed to address all issues observed in related works has therefore been developed. Rather than duplicating tasks that have already been accomplished, BioLake is dedicated to implementing functionalities that have not yet been realized by other researchers. That is, the goal of BioLake is to provide users with a platform where they can perform a variety of bioinformatics analysis tasks in a minimalist-interactive manner. BioLake does not expect any computer science background from the user. However, since it has clear operational processes and intuitive interface design, a user can easily conduct the analysis. Users can also interact with BioLake by submitting a feedback form, which enables BioLake to update the dataset later in a smooth manner without system interruption. Section 4 presents the data analysis Layer of BioLake and Sect. 5 presents the storage strategy of BioLake.

Data analysis layer

Section 4 presents the data analytic layer of BioLake. BioLake employs dynamic analysis as shown in Fig. 2, allowing it to analyze new datasets and genes that it has never seen before. BioLake is designed to be a framework where users interact with the underlying engines through a website designed with minimalist interaction concept. The mRNA analysis layer consists of (a) Differential Analysis - Heatmap, (b) Differential Analysis - Volcano Plot, (c) Expression Analysis - Clinical, (d) Expression Analysis - Survival and e) Gene Set Enrichment Analysis.

Differential gene expression (DGE) analysis is one of the most common applications of RNA-sequencing (RNA-seq) of bioinformatics data. This process allows for the elucidation of differentially expressed genes across two or more conditions and it is widely used in many applications of RNA-seq data analysis [16]. BioLake implements two differential analysis engines as the first layer, since most of the mRNA analysis pipelines start from these two engines before a following gene set enrichment analysis can be conducted. Heatmap is one of the most popular visualizations of gaze behavior, however, increasingly voluminous streams of eye-tracking data make processing of such visualization computationally demanding [17]. BioLake's heatmap analysis starts from raw data extraction where BioLake applies the Z-score algorithm [18] for pre-cleaning. The pipeline then computes the Spearman correlation [19] between all genes and the target gene selected by the user on the starting page of BioLake. This step analyzes the monotonic relationship and applies build-in checkpoints to further filter the data, where these checkpoints are a set of small algorithms that are predefined in BioLake to enhance the readability of results.

Algorithm 1 Out-of-range checkpoint

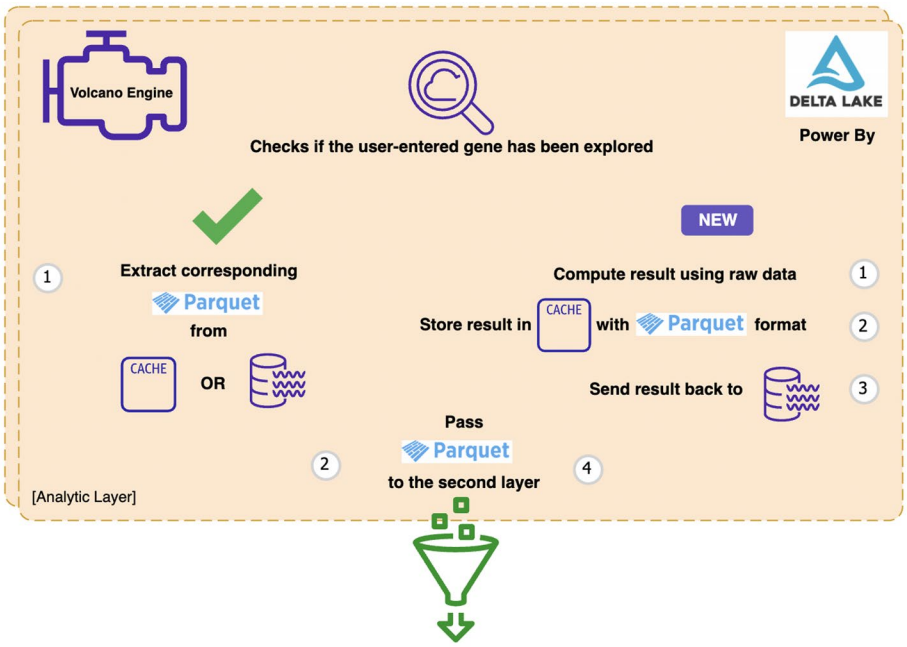

Fig. 2 Fist Layer Interface Design

Input: minValue, geneValueList (list of gene values)
1 **for** $i \leftarrow 0$ **to** $len(geneValueList) - 1$ **do**
2 **if** $geneValueList[i] < minValue$ **then**
3 $geneValueList[i] \leftarrow minValue$;

Algorithm 1 shows one of the checkpoints in the heatmap analysis pipeline which is an Out-Of-Range (OOR) checkpoints for negative heatmap, ensuring that the target gene list obtains the global minimum at the far left, preventing the situation shown in Fig. 3.

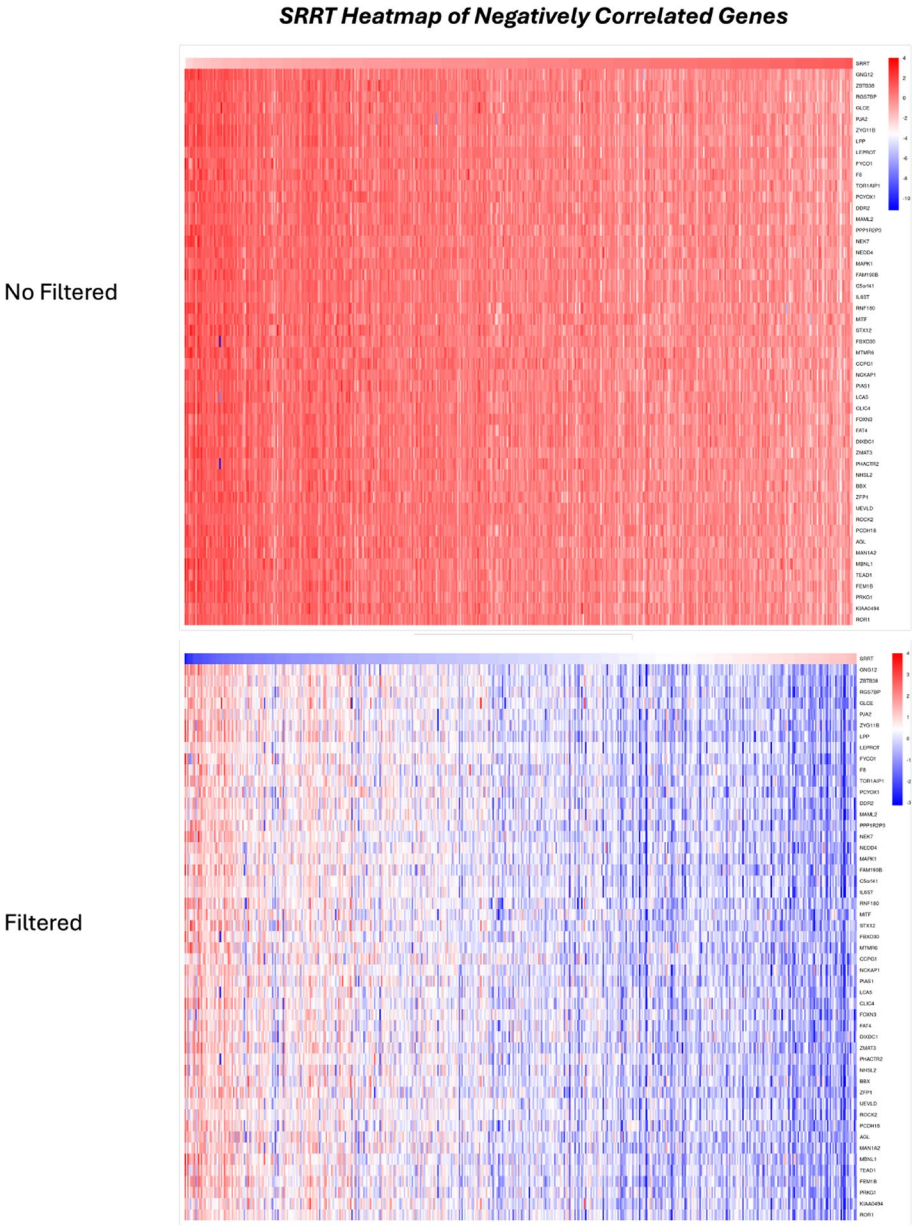

Fig. 3 The comparison between analysis with and without a checkpoint

To interact with the heatmap engine, users are expected to input the number of samples and genes involved. The output of the engine contains two plots with positive and negative correlation information and a summary table is provided below the graphs to clarify statistical information. Whenever the engine explores a new gene in a new dataset, it caches the z-score ranking information and sends it back to the data lake. It will be further used as input for GSEA. Volcano plot, supported by BioLake's volcano engine which visualizes complex datasets generated by genomic screening or proteomic approaches. It is essentially a scatter plot, in which the coordinates of data points are defined by effect size and statistical significance [8]. Similar to heatmap engine, users can interact with this engine by providing the number of P-values and fold-change values. It also caches the ranking information related to fold-change, which serves as subsequent input for GSEA.

There are three secondary engines, the clinical engine, the survival engine and the GSEA engine, implemented within BioLake. The clinical engine is designed to visualize the gene expression using a boxplot, and the survival engine is designed to visualize the survival result via KM curve. To interact with both engines, users are first expected to select a clinical factor to establish an analysis pipeline. The second step is to select how the clustering is performed, BioLake supports numeric, discrete and custom clustering modes, where numeric is set to be default mode.

Figure 4 illustrates an example showcasing BioLake's clustering functionalities. In numeric mode, each distinct value identified in the target column is treated as an individual cluster. In contrast, the discrete mode in BioLake partitions the complete dataset into three clusters by utilizing the values from the target column. These clusters are denoted as "high expression", "stable expression" and "low expression", representing the respective clinical factor levels. It is worth mentioning that the clinical engine does not implement any pre-filtering of clinical data, which implies that users can select all features present in submitted raw clinical data. To enhance the readability, a checkpoint is implemented within this engine. That is, if user selects a clinical factor, where at least one corresponding value of this factor is not numeric, the discrete mode will become invalid even if selected. In such cases, the numeric mode will be automatically applied instead. Custom mode is a unique feature provided exclusively by BioLake, which allows users to create clusters the way they want. When custom mode is selected, users are able to create up to five groups for clustering. Custom mode also supports custom A/B testing, if users do not assign all items from the top side table to the groups they create,

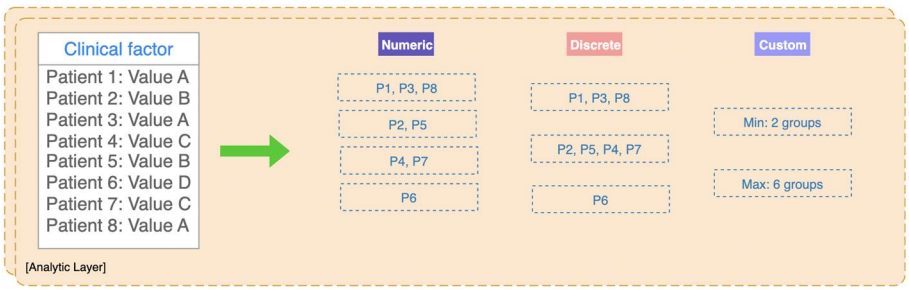

Fig. 4 Clustering mode

a complementary group called "Other" will be generated to store the remaining items. BioLake's clinical engine and survival engines employ this clustering method to facilitate the clinical and survival analysis. In terms of minor adjustments to the result, BioLake provides last-layer data in all cases for users to apply custom modification. Let L be the gene set input by the user and S the gene set in the pathway provided by the annotated database such as GO [20] and KEGG [21]. The goal of the GSEA engine is then to determine whether the members of S are randomly distributed throughout L or primarily found at the top or bottom [22]. BioLake invokes methods from clusterprofiler, a R package implemented by [23], to conduct GSEA. By using BioLake, users do not need to input the gene set manually but directly select the data type they want, and the GSEA engine will extract the correct gene set according to the user input.

Data storage layer

The data storage layer of the BioLake framework is now presented. It applies data lakehouse architecture implemented with Hadoop File System(HDFS), Amazon S3 and Delta Lake [3] to enhance the flexibility and practicality of the framework. Formally, a data lakehouse is a data management system design that combines the key benefits of data lakes and data warehouses which are low-cost storage in an open format accessible by a variety of systems from the data lake to powerful management and optimization features from the data warehouse [4]. Unlike OLTP-type data, such as banking data, bioinformatics data is similar to streaming data. For example, a banking data item such as a user's account balance can be frequently updated, but we mainly care about the latest value. However, in terms of bioinformatics data, the expression value of a gene at different time points is equally important as bioinformatics data does not lose its value over time. Cloud object stores such as Amazon S3 are some of the largest and most cost-effective storage systems on the planet, making them an attractive target for storing large data warehouses and data lakes [3]. However, storing the data in the cloud may lead to some privacy issues. Thus, in BioLake, both Amazon S3 and Hadoop have been chosen to be the data lake component. There is no restriction on the data format, as the framework does not expect S3 and Hadoop to interpret the information. This enhances BioLake's ability to accept a wide range of data formats, which significantly increases its flexibility. Delta Lake [3] is then applied to manage the data lake, which provides traditional analytical DBMS management and performance features to the processing engine. This data storage strategy brings great scalability to the framework as the storage capacity can be easily expanded through horizontal scaling to accommodate growing data volumes. Other than that, the framework is highly portable as it applies analysis-storage separation, which means that the storage can quickly be migrated to other cloud vendors or on-premises servers without updating any data processing logic.

In BioLake, Delta [3] is employed as the preferred storage format to optimize computational performance. Figure 5 provides a detailed comparison between Delta and CSV for all analysis pipelines in BioLake. The result denotes that Delta markedly overperforms CSV, which demonstrates Delta's superior capability in handling large datasets, ensuring faster data retrieval and processing, thereby facilitating more effective data analysis within the BioLake framework. The substantial difference in processing speeds can be attributed to the data storage format. Although traditional bioinformatics tools have

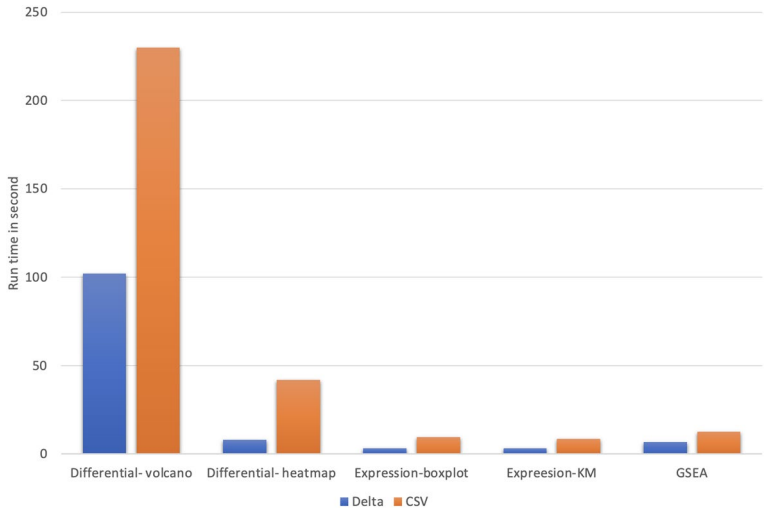

Fig. 5 Runtime comparison

used custom data formats such as SAM, BAM and VCF [24, 25], many organizations are now storing this data in data lake formats such as Parquet. This approach was pioneered by The Big Data Genomics project [3, 26]. In BioLake, Delta is the default storage format, where expression data is stored in Parquet, an efficient, structured, column-oriented (also called columnar storage), com-pressed, binary file format [27]. The difference between row-oriented storage and column-oriented storage was discussed above and in greater detail in [28]. In simple terms, column-oriented storage is ideal when we need to frequently update the existing data (OLTP), which is not the typical scenario in bioinformatics analysis. In fact, Delta is a prevalent storage format for bioinformatics data. In 2019, an open-source toolkit for genomics data named Glow was released by Databricks and Regeneron [29]. This toolkit uses Delta for storage.

Reactive continuous data integration

Data warehouse systems normally have static structures for their schemas and the relationships between data, and they are therefore not able to support any dynamism in their structure and content. Their data is only periodically updated because they are not prepared for continuous data integration [30]. In other words, this approach expects a large amount of fixed-format data injection in the early stage of the launching of an application which reduces the flexibility for the input data. The reactive continuous data integration mechanism for the BioLake framework, which enables greater flexibility, is therefore presented now.

Figure 6 presents the design of the reactive continuous data integration mechanism. The design framework only pre-stores raw data in the data lake. If a user establishes a pipeline, the pipeline first searches if the required data can be found in the server cache. If not, the pipeline requests the data from lake and appends the data to the data in the cache. A vital point of this mechanism is that the pipelines only directly access raw data when necessary, since raw data cannot be used before filtering and checkpointing. For example, if a user enables a differential analysis, the most ideal scenario is that the pipeline can find directly usable data in the cache, which implies

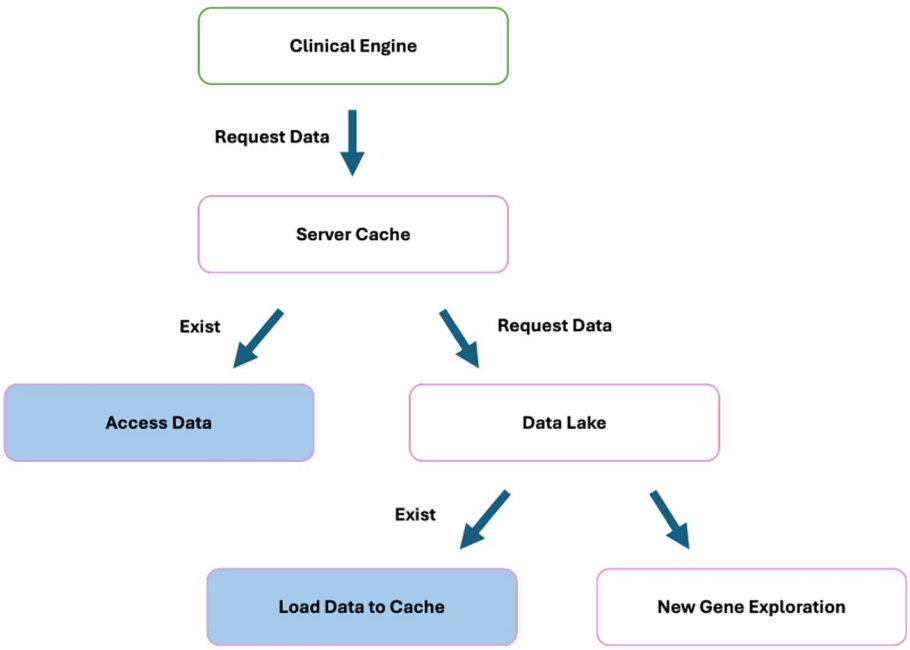

Fig. 6 Reactive continuous data integration

that there is another user who set up a same analysis right before the current user and leaves the intermediate data in the cache. The main advantage of this design is that it significantly reduces duplication of work.

The working of this mechanism is now presented by an example. Suppose Alex is the first user accessing the framework and he selects *SETD*2 as the target gene. Then Alex completes the configuration step and establishes a heatmap differential analysis pipeline. The pipeline then checks if there is a usable data set A in the cache that can be used for the first step of the mechanism. In this case, A does not exist in the cache since this is the first heatmap pipeline in the framework. The pipeline then tries to access the data in data lake. Again, it cannot get the expected data as the framework only pre-stores raw data in data lake, which is not directly usable. Since the pipeline fails to retrieve the data A from either the cache or the data lake, the analysis pipeline starts to generate the required data from scratch. For this process it fetches the raw data from data lake and builds the data following the differential analysis pipeline described in Sect. 3.1. Once completed, the intermediate data will be sent back to data lake with configuration tagging, so that the later analysis pipeline can detect the usable data by checking the configuration tagging. Suppose later that Bob accesses the framework and selects the same gene of interest, which in this case is *SETD*2. Then the analysis pipeline he submits will get the data without creating it from scratch since this gene has been analyzed by other users.

This mechanism is an implementation of the idea of space time tradeoffs, where we increase the use of space while reducing pipeline processing time. Having said that, the framework does not keep all the intermediate data generated by the framework. Only high-frequency and relatively general data can be permanently stored in BioLake.

Custom resource management

The cashing approach used for the resource management is now presented. Caching techniques are widely used in today's computing infrastructure from virtual memory management to server caches and memory caches [31]. As noted in Section 4.1, most of the intermediate data generated by the pipeline is placed in the server cache for reuse purpose. Thus, the framework applies two resource management mechanisms to prevent the infinite growth of the cache size, since the server capacity is limited.

The first mechanism for this approach is for the general custom intermediate data where general custom intermediate data is data constructed without the participation of a custom configuration. For example, the foldchange table of a specific gene, is defined as general custom intermediate data in the BioLake framework. The primary feature of this kind of data is the repeated usage by multiple pipelines. Thus, the Least Recently Used (LRU) policy is applied, which always replaces the least-recently-used page in the memory cache [32] with the new data. This policy is applied to the server cache, to remove the least recently used data. That is, when the cache capacity exceeds a preset limit, and a pipeline tries to place new data that doesn't exist in the cache and the framework starts removing the least recently used data in the cache to ensure the cache capacity stays below our preset limit. The second mechanism is for the fully custom intermediate data where fully custom intermediate data is defined as data constructed with the participation of all the custom configurations. Since this kind of data has a low reuse rate, it is assigned a browser ID to each data item, where the browser ID corresponds to the browser the user is using. When a user accesses the BioLake framework, its browser will be assigned a unique ID and the framework will append this ID to all the fully custom intermediate data generated by this browser. Once the user closes this browser, all fully custom intermediate data assigned with the corresponding browser ID will be deleted immediately.

Case study

A case study to illustrate BioLake's usability and flexibility is now presented. This study briefly introduces the basic functionalities of BioLake and demonstrates some sample results. To begin, all the data configurations are reset and the server cache is cleared and we apply *SETD*2 analysis on the *TCGA_PRAD* dataset.

Figures 7 and 8 present the heatmap analysis where users are able to decide the number of top-ranked genes and where these selected top-ranked genes will be stored for GSEA. To conduct this analysis, users are expected to input the number of genes involved, which determines the vertical extent. It is worth mentioning that BioLake does not pre-analyze the user-provided dataset until the actual execution is triggered by the user, which implies that the user's input range is not limited. However, if the input range is beyond the scope of the applied dataset, BioLake will apply the maximum range to the output instead.

Figure 9 shows the overall procedure of volcano differential analysis for gene *SETD*2, where users can generate the volcano plots in the way they want, by adjusting adjust P value and fold-change values. Since BioLake employs a reactive continuous data integration as described in Sect. 4.1, this differential analysis becomes the prerequisite for

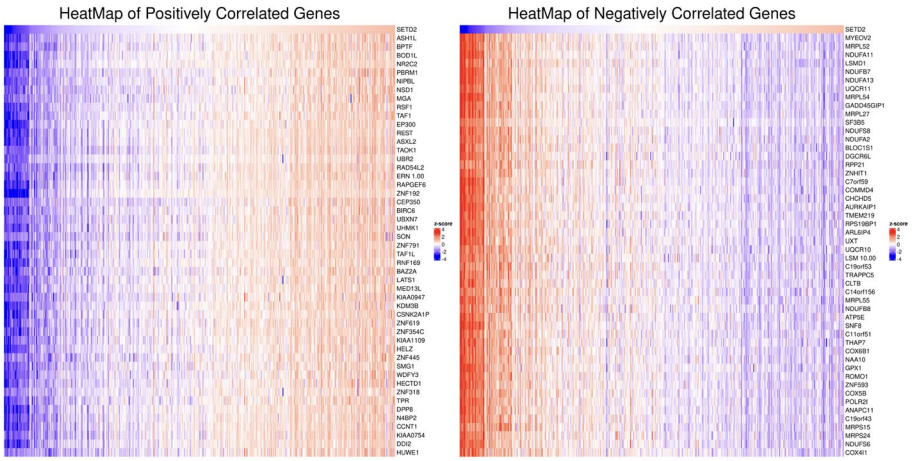

Fig. 7 Heatmap analysis for top 50 genes in TCGA_PRAD, where target gene is set to *SETD2*

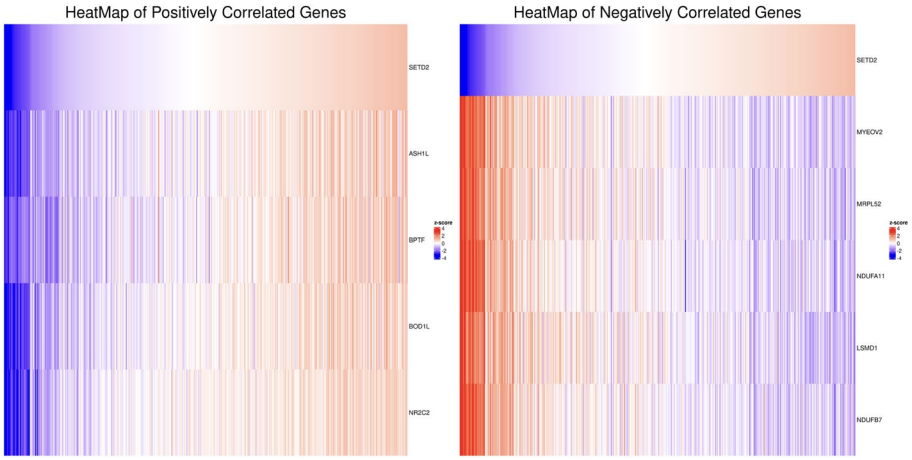

Fig. 8 Heatmap analysis for top 5 genes in TCGA_PRAD, where target gene is set to *SETD2*

GSEA if BioLake has not previously interpreted the applied dataset. However, once this differential analysis is performed for the first time, the intermediate data will be cached, and publicly accessible instantly for further use.

Figure 10 presents sample results of BioLake's clinical analysis, where *gleason_score* was used as the splitting metric. The left figure displays the results in numeric mode, with each distinct stage's gene expression represented by a single boxplot. The right figure shows the expression results in custom mode, where a predefined group was created to store specific *gleason_score* stages (in this example, we picked stage 6, 7, and 8). Recall that if not all stages are assigned to these predefined groups, BioLake creates an additional *other* group to include any unassigned stages. In this example, the *other* group contained all samples with *gleason_score* stages not equal to 6, 7, or 8. This design enables users to conduct A/B testing by assigning a set of the target stages to *group*0 while leaving the remaining stages into the *other* group. Figure 11 shows the sample result of BioLake's survival analysis, where the level of gene expression was used to split the samples into four groups

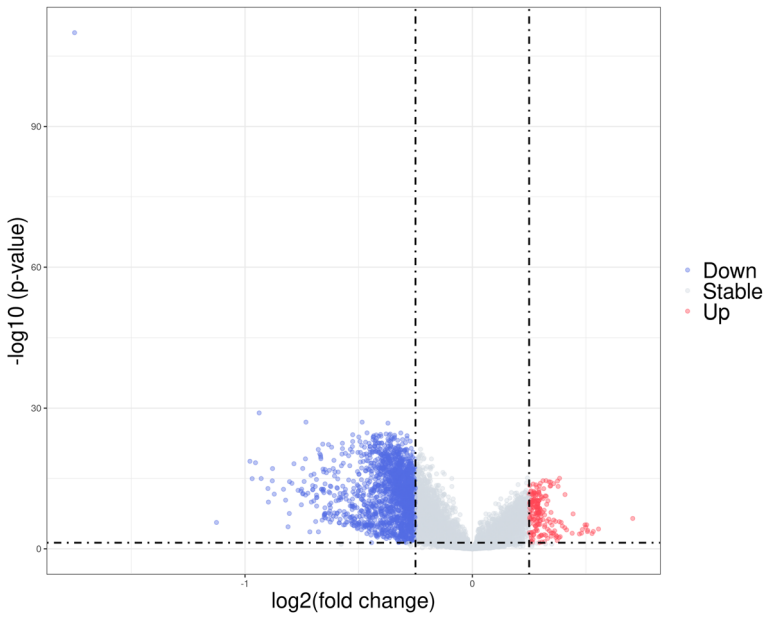

Fig. 9 Volcano analysis with adjust P value = 0.05 and foldchange = 0.25

The users can apply the intermediate data to conduct the corresponding GSEA when at least one of the two differential analyses has been done. Figure 12 presents the GESA analysis using correlation as the ranking metric, where the correlation value was generated by the previous heatmap analysis pipeline. There are three KEGG plots shown in Fig. 12, where the left side was generated using the top 700 genes, the middle was generated using the top 7000 genes, and the last one was generated using all available genes.

Conclusion

BioLake provides a web-based framework for bioinformatic researchers, accessible at https://biolake.ucalgary.ca. The hope is that BioLake will allow researchers to focus more on scientific research rather than tool usage. BioLake does not apply manual inspection for any incoming data provided by the user to enhance its flexibility. And users can decide about the public visibility of provided data. In the case where the permission of public visibility is granted, BioLake will credit the provider and open the data to public access. However, denying public visibility cannot guarantee the security of provided data, as the current version of BioLake does not apply strict security protection. In term of usability, BioLake returns the last layer data in each analysis, which can be directly used in modern machine learning algorithms, or critical feature selection for clinical perdition [33]. Lasting, while BioLake was designed to tackle prostate cancer data analysis, the capability goes way beyond this scope, and we expect to append more datasets gradually for other cancers if applicable in the future to fulfill more analysis requirements.

Future work

The future development of this platform will be directed toward two critical aspects. The first aspect focuses on enhancing the security and confidentiality of user-provided data. To achieve this, we will explore the implementation of advanced security technologies,

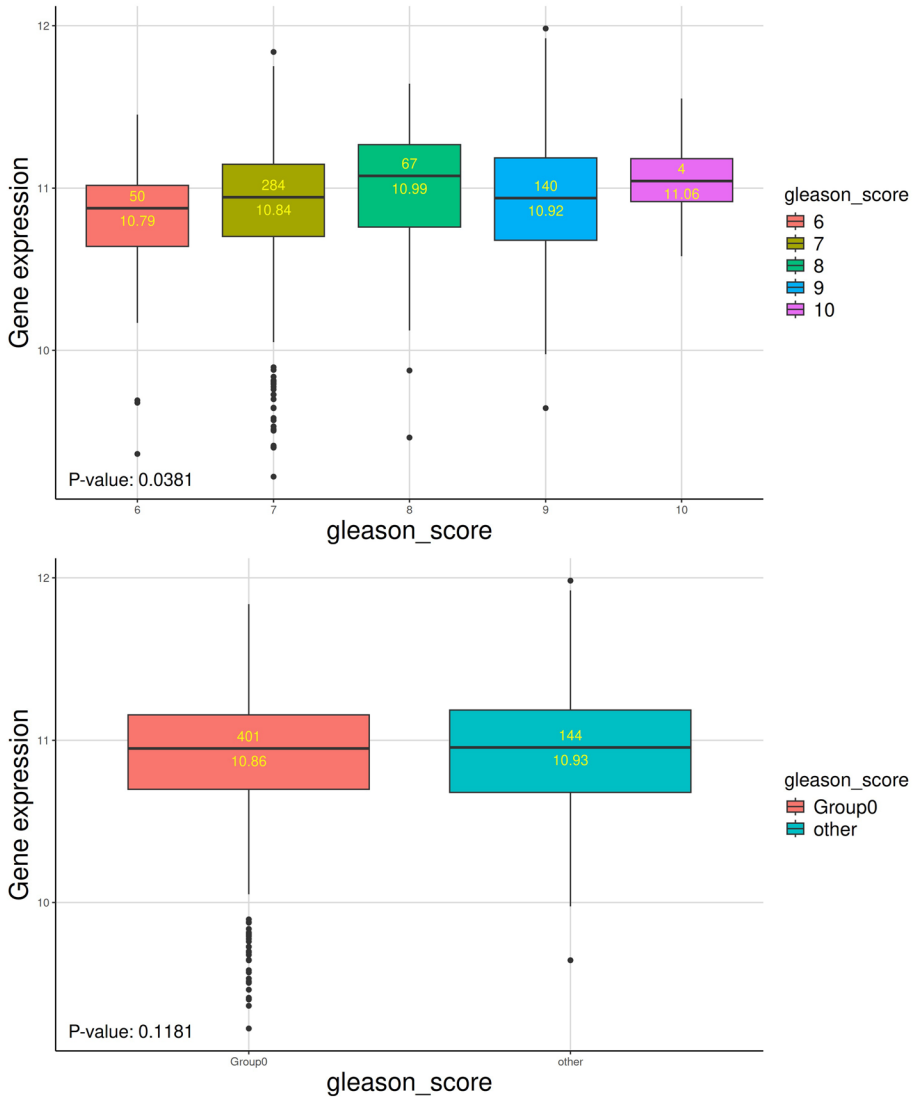

Fig. 10 Clinical analysis for expression level using gleason score as splitting metric

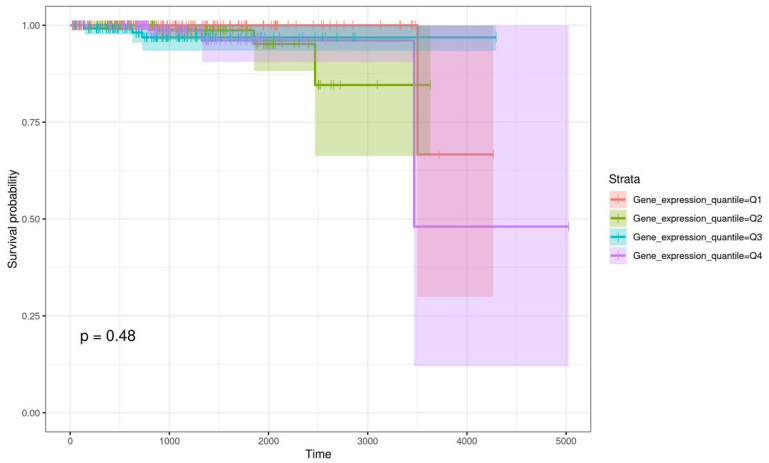

Fig. 11 KM analysis where groups are split by expression level

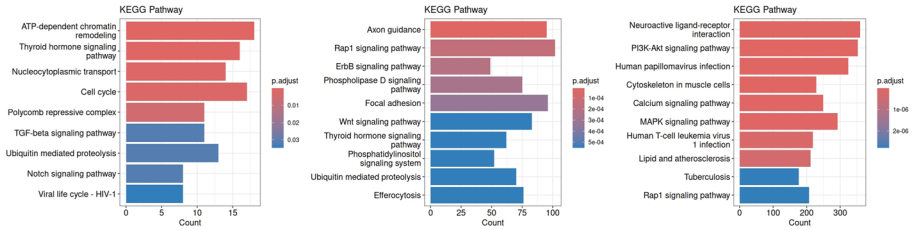

Fig. 12 GSEA analysis ranked by correlation

such as blockchain frameworks [34] and RNA-based encryption methodologies [35], to safeguard the integrity and privacy of custom data during analysis.

The second aspect involves improving the scalability and versatility of the platform. This will be accomplished by incorporating additional analytical engines and expanding the range of supported datasets. Such advancements will enable the platform to facilitate the analysis of various cancer types beyond prostate cancer, thereby broadening its applicability and impact in the field of bioinformatics research.

Abbreviations

NGS	Next generation sequencing
DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
mRNA	Messenger ribonucleic acid
URL	Uniform resource locator
TCGA	The cancer genome atlas
PRAD	Prostate adenocarcinoma
DGE	Differential gene expression
GSEA	Gene set enrichment analysis
HDFS	Hadoop file system
OLTP	Online transaction processing
CSV	Comma-separated values
SAM	Sequence alignment/Map
BAM	Binary alignment/Map
VCF	Variant call format
LRU	Least recently used

Author Contributions

Qiaowang worked on the software development under the supervision of Dr. Reda and Dr. Jon. Dr. Yaser and Dr. Tarek provided clinical and bioinformatics suggestions. All authors proofread the paper.

Funding

This project does not receive external funding.

Availability of data and materials

The public data we use can be found at https://biolake.ucalgary.ca/. **Project name:** BioLake; **Project home page:** e.g. https://biolake.ucalgary.ca/; **Operating system(s):** Linux; **Programming language:** Python; **Other requirements:** N/A as BioLake is a web-based software; **License:** Creative Commons Attribution-NonCommercial (CC BY-NC); **Any restrictions to use by non-academics:** Limited to academic and non-commercial use

Code availability

The R source code used to generate the result images can be found at https://github.com/qiaowangli/bioLake_r_source_code.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Materials availability

Not applicable.

Competing interests
Not applicable.

Received: 22 October 2024 Accepted: 14 January 2025
Published online: 04 February 2025

References

1. Smith S, Hogan J, Rittenbruch M, Johnson D, Brereton M. Visual analytics for large-scale bioinformatic data sets. In Proceedings of the annual meeting of the australian special interest group for computer human interaction. 2015;603–7.
2. Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, Downey P, Elliott P, Green J, Landray M, et al. UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Med. 2015;12(3):1001779.
3. Armbrust M, Das T, Sun L, Yavuz B, Zhu S, Murthy M, Torres J, Hovell H, Ionescu A, Łuszczak A, et al. Delta lake: high-performance acid table storage over cloud object stores. Proc VLDB Endow. 2020;13(12):3411–24.
4. Armbrust M, Ghodsi A, Xin R, Zaharia M. Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. Proc CIDR. 2021;8:28.
5. Segal T, Barnard R. Let the shoemaker make the shoes-an abstraction layer is needed between bioinformatic analysis, tools, data, and equipment: an agenda for the next 5 years. In: Proceedings of the first Asia-pacific bioinformatics conference on bioinformatics 2003. 2003;19:215–8.
6. BioinformaticsFMRP: TCGAbiolinksGUI. http://tcgabiolinks.fmrp.usp.br:3838/. Accessed 8 April 2023
7. Chen EY, Tan CM, Kou Y, Duan Q, Wang Z, Meirelles GV, Clark NR, Ma'ayan A. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. BMC Bioinform. 2013;14:1–14.
8. Goedhart J, Luijsterburg MS. Volcanoser is a web app for creating, exploring, labeling and sharing volcano plots. Sci Rep. 2020;10(1):20560.
9. Goldman M, Craft B, Hastie M, Repečka K, McDade F, Kamath A, Banerjee A, Luo Y, Rogers D, Brooks AN, et al. The UCSC Xena platform for public and private cancer genomics data visualization and interpretation. biorxiv, 2018;326470.
10. Li R, Zhu J, Zhong W-D, Jia Z. Pcadb-a comprehensive and interactive database for transcriptomes from prostate cancer population cohorts. BioRxiv. 2021;2021–06.
11. Mahi NA, Najafabadi MF, Pilarczyk M, Kouril M, Medvedovic M. GREIN: an interactive web platform for re-analyzing GEO RNA-seq data. Sci Rep. 2019;9(1):7580.
12. Bioinformatics—SRPlot. https://www.bioinformatics.com.cn/srplot. Accessed 26 June 2023
13. Tang Z, Li C, Kang B, Gao G, Li C, Zhang Z. Gepia: a web server for cancer and normal gene expression profiling and interactive analyses. Nucl Acids Res. 2017;45(W1):98–102.
14. Vasaikar SV, Straub P, Wang J, Zhang B. Linkedomics: analyzing multi-omics data within and across 32 cancer types. Nucl Acids Res. 2018;46(D1):956–63.
15. Colaprico A, Silva TC, Olsen C, Garofano L, Cava C, Garolini D, Sabedot TS, Malta TM, Pagnotta SM, Castiglioni I, et al. Tcgabiolinks: an R/Bioconductor package for integrative analysis of TCGA data. Nucl Acids Res. 2016;44(8):71–71.
16. McDermaid A, Monier B, Zhao J, Liu B, Ma Q. Interpretation of differential gene expression results of RNA-seq data: review and integration. Brief Bioinform. 2019;20(6):2044–54.
17. Dao T-C, Bednarik R, Vrzakova H. Heatmap rendering from large-scale distributed datasets using cloud computing. In: Proceedings of the symposium on eye tracking research and applications. 2014;215–8.
18. Abdi H. Z-scores. Encycl Meas Stat. 2007;3:1055–8.
19. Spearman C. The proof and measurement of association between two things. 1961.
20. Consortium GO. The gene ontology (go) database and informatics resource. Nucl Acids Res. 2004;32(suppl-1):258–61.
21. Kanehisa M. The kegg database. In: 'In Silico'simulation of biological processes: novartis foundation symposium 2002;247, vol. 247, pp. 91–103. Wiley Online Library
22. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci. 2005;102(43):15545–50.
23. Yu G, Wang L-G, Han Y, He Q-Y. clusterprofiler: an r package for comparing biological themes among gene clusters. Omics J Integrat Biol. 2012;16(5):284–7.
24. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, et al. The variant call format and VCFtools. Bioinformatics. 2011;27(15):2156–8.
25. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Subgroup GPDP. The sequence alignment/map format and SAMtools. bioinformatics. 2009;25(16):2078–9.
26. Nothaft FA, Massie M, Danford T, Zhang Z, Laserson U, Yeksigian C, Kottalam J, Ahuja A, Hammerbacher J, Linderman M, et al. Rethinking data-intensive science using scalable analytics systems. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. 2015;631–46.
27. Vohra D, Vohra D. Apache parquet. Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools. 2016;325–35.
28. Abadi DJ, Madden SR, Hachem N. Column-stores versus row-stores: how different are they really? In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008;pp. 967–980.
29. Glow: An open-source toolkit for large-scale genomic analysis. https://projectglow.io
30. Santos RJ, Bernardino J. Real-time data warehouse loading methodology. In: Proceedings of the 2008 International Symposium on Database Engineering & Applications, 2008;pp. 49–58

31. Li P, Pronovost C, Wilson W, Tait B, Zhou J, Ding C, Criswell J. Beating opt with statistical clairvoyance and variable size caching. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019;pp. 243–256.
32. Megiddo N, Modha DS. Outperforming LRU with an adaptive replacement cache algorithm. Computer. 2004;37(4):58–65.
33. Khan MA, Mazhar T, Mateen Yaqoob M, Badruddin Khan M, Jilani Saudagar AK, Ghadi YY, Khattak UF, Shahid M. Optimal feature selection for heart disease prediction using modified Artificial Bee colony (M-ABC) and K-nearest neighbors (KNN). Sci Rep. 2024;14(1):26241.
34. Mazhar T, Shah SFA, Inam SA, Awotunde JB, Saeed MM, Hamam H. Analysis of integration of IoMT with blockchain: issues, challenges and solutions. Discover Internet of Things. 2024;4(1):1–36.
35. Aoun M, Mazhar T, Nadeem MA, Shahzad T, Rehman SU, Rehman AU. A novel encryption scheme for secure communication based on rna. CSI Transactions on ICT, 2024;1–10.

Publisher's Note