frontiers in Genetics

TECHNOLOGY REPORT
published: 14 June 2019
doi: 10.3389/fgene.2019.00557

L1000 Viewer: A Search Engine and Web Interface for the LINCS Data Repository

Aliyu Musa[1,2], *Shailesh Tripathi*[1,3], *Matthias Dehmer*[3,4,5] *and Frank Emmert-Streib*[1,2]*

[1] Predictive Society and Data Analytics Lab, Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland, [2] Institute of Biosciences and Medical Technology, Tampere, Finland, [3] Institute for Intelligent Production, Faculty for Management, University of Applied Sciences Upper Austria, Linz, Austria, [4] Department of Mechatronics and Biomedical Computer Science, UMIT, Hall in Tyrol, Austria, [5] College of Computer and Control Engineering, Nankai University, Tianjin, China

The LINCS L1000 data repository contains almost two million gene expression profiles for thousands of small molecules and drugs. However, due to the complexity and the size of the data repository and a lack of an interoperable interface, the creation of pharmacologically meaningful workflows utilizing these data is severely hampered. In order to overcome this limitation, we developed the L1000 Viewer, a search engine and graphical web interface for the LINCS data repository. The web interface serves as an interactive platform allowing the user to select different forms of perturbation profiles, e.g., for specific cell lines, drugs, dosages, time points and combinations thereof. At its core, our method has a database we created from inferring and utilizing the intricate dependency graph structure among the data files. The L1000 Viewer is accessible via http://L1000viewer.bio-complexity.com/.

Keywords: gene expression, big data, pharmacogenomics, web application, visualization, data science

OPEN ACCESS

Edited by:
Mehdi Pirooznia,
National Heart, Lung, and Blood Institute (NHLBI), United States

Reviewed by:
Zichen Wang,
Icahn School of Medicine at Mount Sinai, United States
Marco Brandizi,
Rothamsted Research (BBSRC), United Kingdom

****Correspondence:***
Frank Emmert-Streib
v@bio-complexity.com

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Genetics

Received: *13 April 2019*
Accepted: *28 May 2019*
Published: *14 June 2019*

Citation:
Musa A, Tripathi S, Dehmer M and Emmert-Streib F (2019) L1000 Viewer: A Search Engine and Web Interface for the LINCS Data Repository. Front. Genet. 10:557. doi: 10.3389/fgene.2019.00557

1. INTRODUCTION

We are living in the era of big data that sparked the establishment of the field data science (Smith, 2006; Ma'ayan et al., 2014; Jin et al., 2015; Emmert-Streib and Dehmer, 2019). For genomics, the recent growth of high-throughput biomedical and pharmacogenomic data (Edgar et al., 2002; Barrett et al., 2013; Woo et al., 2015; Musa et al., 2017) presents opportunities and at the same time challenges for their analysis. Paramount to these problems is ensuring that comparative genomics tools keep pace with the rate at which the data are produced (Tripathi et al., 2014; Smirnov et al., 2016; Stupnikov et al., 2016). A major challenge researchers are facing practically when interacting with "big data" is that most of the relevant information requires a considerable amount of time to subset, preprocess and obtain. Therefore, novel approaches for finding, selecting and downloading specific subdata from large data repositories are required. This is particularly a problem for obtaining raw data (Musa et al., 2018).

One example for such a big data repository is the Library of Integrated Network-based Cellular Signatures (LINCS) (Subramanian et al., 2017). The LINCS L1000 data repository consists of almost two million individual files containing information about the gene expression and metadata of cell lines perturbed by chemicals of certain dosages and durations (Vempati et al., 2014). While there are several desktop or command line software tools available that are capable of processing and manually extracting subsets of large data, these tools require software installation, which can be

TABLE 1 | List of available LINCS L1000 metadata APIs.

Service (API)	Description	URL link
Cell	The cell information service provides cell line meta-information for used in the experiments.	https://clue.io/api#cells
Gene	The gene information service returns meta-information for measured and inferred genes in the LINCS dataset.	https://clue.io/api#genes
Profile	The profile information service returns meta-information for instances in the LINCS dataset.	https://clue.io/api#profiles
Pert	The pert information service returns meta-information for perturbagens in the LINCS dataset.	https://clue.io/api#perts
Plate	The plateInfo service returns plate information.	https://clue.io/api#plates

difficult and time consuming, and are only capable of processing the data locally (Duan et al., 2016; Enache et al., 2017; Fallahi-Sichani et al., 2017). Therefore, the datasets in the repository can only be analyzed if the end-user has specialized software installed. Improvements in software development but also web-based application technologies such as the Node.js and Vue.js JavaScript libraries, have led to the development of advanced web-based applications with animated and interactive features. While there are several interactive web-based tools that can access data via an application programming interface (API) (Subramanian et al., 2017), most of these tools have limited interactivity and sharing capabilities, e.g., by embedding them within web applications such as CMAP (Lamb et al., 2006). Furthermore, they are lacking an integration with biology specific analysis methods, e.g., for performing an enrichment analysis (Rahmatallah et al., 2017). Importantly, all of these tools operate on the signature level of the LINCS data, not the raw data. That means, if a user wants to select a specific subset of raw data for a dedicated analysis, there is no help available.

In order to facilitate the access and subset of raw data from the LINCS data repository we developed the L1000 Viewer. Our software is an interactive web application that does not require the user to install dedicated software, but it operates via any web browser on any operating system. Hence, it is operating system independent. Our web application provides a web interface with access to a dedicated database we created. This database utilizes the graph dependency structure between the individual data files of LINCS because *individual* does not mean *independent*. Specifically, the dependency structure is induced by the experimental conditions of the expression profiles and can be represented as a graph or network (Musa et al., 2018). In this graph, nodes correspond to data files and two data files are connected if they share experimental conditions. Our web application provides an easy-to-use interactive platform allowing the user to select subsets of raw data files that belong to specific forms of perturbation profiles, e.g., for specific cell lines, drugs, dosages and time points. This retrieval of data files is efficient and fast because of the utilization of the precomputed graph structure of the data files. In addition, we are providing software for a graphical summarization of the selected data showing various distributions of experimental parameters, e.g., sample sizes per cell line, sample sizes per concentration and sample sizes per time point. This provides valuable information for the user regarding the experimental design (Hinkelmann and Kempthorne, 2008) of follow-up computational pharmacogenomics studies based on these data.

Our paper is organized as follows. In the next section, we discuss all methods and data we use for our analysis. In the results section, we present our findings and provide results for an example application of our software. In the following sections, we discuss our results in detail. The paper finish with conclusions and an outlook.

2. METHODS

2.1. LINCS Data

The LINCS data is a vast collection of gene expression profiles that includes many experimental samples covering more than seventy human cell lines. These cell lines are populations of cells that descended from an original source cell and having the same genetic make-up. These cells have been kept alive by growing them in a culture separate from their original source (Ong et al., 2017).

Specifically, LINCS contains about $1,328,098$ gene expression profiles as a result from applying $42,553$ perturbagens ($19,811$ small molecule compounds, $18,493$ shRNAs, $3,627$ cDNAs, and 622 biologics) for a total of $476,251$ signatures (consolidating replicates) (Subramanian et al., 2017).

2.2. Metadata and Data Standards

LINCS provides an API to annotations and perturbational signatures in the L1000 data repository via a collection of HTTP-based RESTful web services. An example of such a services is the Cell Service which is a service that describes meta-information for cell lines. **Table 1** lists all the API services provided by LINCS for querying the L1000 metadata. These services support complex queries via simple HTTP GET requests that can be executed in a web browser or within most programming languages.

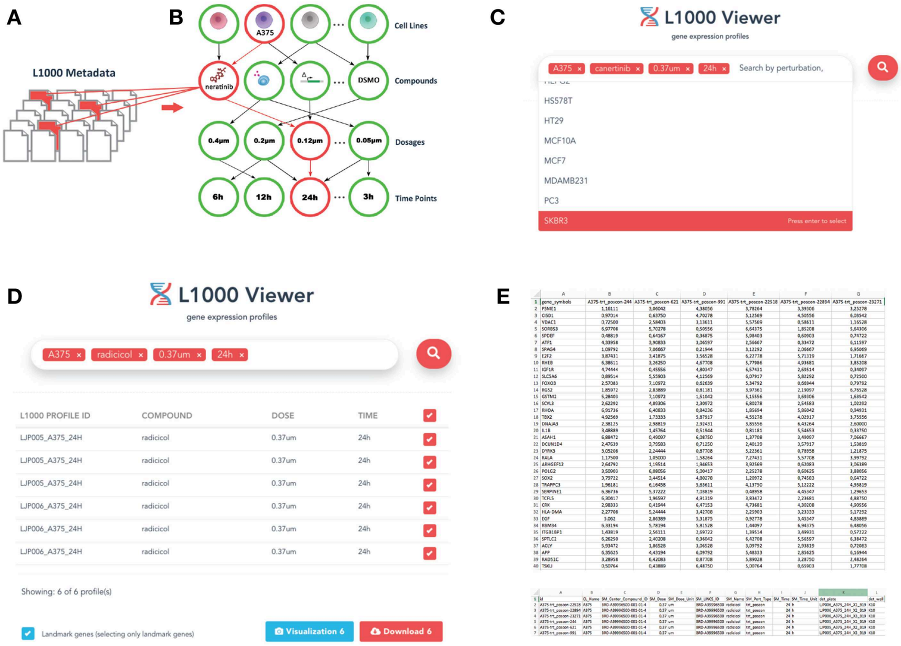
2.3. Development of the Web Application

L1000 Viewer, the web application we have developed, consists of three main parts namely; (I) the database, (II) back-end, and (III) front-end implementations.

First, in order to store the data in the back-end, we use a MongoDB database. We convert and store all the raw data into a json object structure to enable identifier reference to each profile sample in the database. This enables the data to be stored as a document-oriented structure that allows fast user queries. The document-oriented model maps to the data objects in the application code in the back-end, making the data easy to work with. The MongoDB is a distributed database at its core, therefore, it enables a horizontal scaling, high availability and faster access.

The specific document structure is constructed from the experimental conditions of the individual data profiles within the LINCS data repository. As a result we obtained a relational representation of the documents using Mongoose schema. Mongoose provides a straight-forward, schema-based solution to model json object data into relationships. It includes built-in type casting, validation, query building, and logic hooks environment that wraps the Node.js native driver. This is visualized in **Figures 1A,B**. By (I) identifying and (II) utilizing this structure, our L1000 Viewer is able to efficiently provide a list of result profiles corresponding to an user-defined query. For instance, querying for the cell line A375, the drug neratinib, a dosage of 0.12μ and a duration of 24 h (see **Figure 1C**) results in 5565 files (see **Figure 1D**) that match the query list. That means the L1000 Viewer is a search interface that represent a relational structure from the underlying individual profiles corresponding to the instances in the database collection and allows by this an efficient querying of these profiles.

Second, for the back-end component, we decided to use Node.js for the server side architecture. A Node.js server



L1000 PROFILE ID	COMPOUND	DOSE	TIME	☑
LJP005_A375_24H	radicicol	0.37um	24h	☑
LJP005_A375_24H	radicicol	0.37um	24h	☑
LJP005_A375_24H	radicicol	0.37um	24h	☑
LJP006_A375_24H	radicicol	0.37um	24h	☑
LJP006_A375_24H	radicicol	0.37um	24h	☑
LJP006_A375_24H	radicicol	0.37um	24h	☑

Showing: 6 of 6 profile(s)

☑ Landmark genes (selecting only landmark genes)

FIGURE 1 | Overview of the L1000 Viewer. **(A)** Individual files forming the LINCS data. **(B)** Our graph database inferred from the relations among the individual profiles. **(C)** The interface allowing to enter queries to search the graph database. Shown is an example query discussed in the text. **(D)** The result of a query leads to a downloadable list. **(E)** Gene expression and metadata csv files.

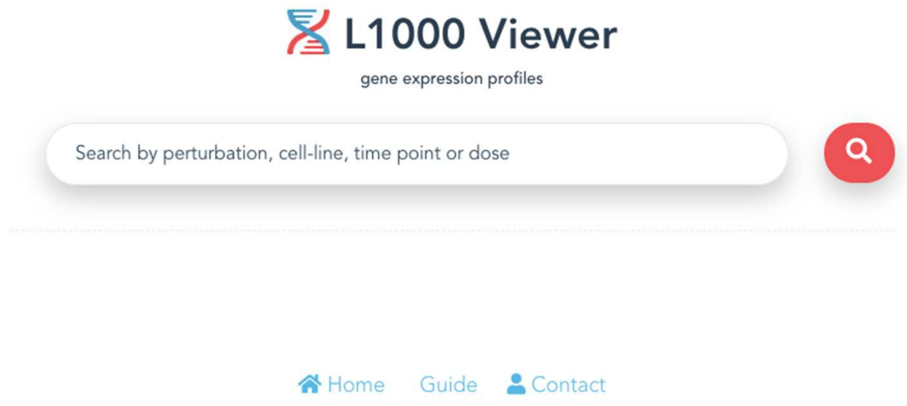


FIGURE 2 | The entry point of the L1000 Viewer application.

environment was utilized to interact with the database through custom object-data modeling (ODM) calls adopted from pseudo relational database representation in Mongoose API. The main benefit of using this model is that you can define schemas for your collections which are then enforced at the ODM layer by architecture. It also has utilities for simplifying Node's callback patterns that make it easier to work with than the standard MongoDB driver alone. In general, this approach makes it even easier to use MongoDB with Node.js. Node.js is a web application development framework that uses convention over configuration. This means it can be efficiently used to spin a back-end development environment and also allows users to quickly understand the source code and contribute to development. It also supports a rich database of user-contributed libraries called packages that ease many complicated tasks, e.g., in handling downloading and archiving requests on the server side. We use packages such as backbone.js, archiver.js, underscore.js etc. to build the back-end. The L1000 Viewer was deployed on a Linux operating system supported by the Node.js runtime library. It is deployed on an Nginx server using Linode node.

Third, for the work-flow designer on the front-end we used javascript. Specifically, we use Vue.js to created the front-end representation. Vue.js is a widely used javascript framework and the L1000 Viewer uses it for handling all client side user interactions. The connections between the components of the interface are implemented using Vue.js plugins. It provides a mechanism to display and render the structural components from HTML tags. To interactively display the large collection of drug-induced profiles, the HTML5 elements were used to layout the profiles systematically.

Overall, the model-view-controller (MVC) software architecture was used to integrate the front-end, back-end and the database. The MVC pattern of design describes the behavior of the application's data, logic, rules, and generates an output based on changes to the application. The advantage of this is it helps in focusing on a specific part of the application name, the ways information is presented to and accepted from, the user.

2.4. Graphical Summary

In addition, we provide a functionality for an interactive visualization for viewing the selected profiles on the web. A user can click on the visualization button from the search results to visualize the selected profiles in different plots (e.g., boxplot representation of the profiles etc.). The metadata information of the selected profiles are also displayed. We provide R scripts for further metadata visualizations. Specifically, we provide scripts that allow the user to generate graphical summary statistics of their metadata query results. From the download function, the user can immediately download the profiles and use the R scripts on the subset of the data that was retrieved.

3. RESULTS

We start this section by describing the basic functionality of the L1000 viewer web application we developed. Then we discuss its specific components in detail and provide an example.

3.1. General Overview of the L1000 Viewer

The L1000 Viewer has an interface allowing the user to enter queries in a disjunctive normal form (DNF), i.e., one can search for the simultaneous presence of search terms in the form,

$$\text{term}_1 \text{ AND } \text{term}_2 \text{ AND } \cdots \text{ AND } \text{term}_n \quad (1)$$

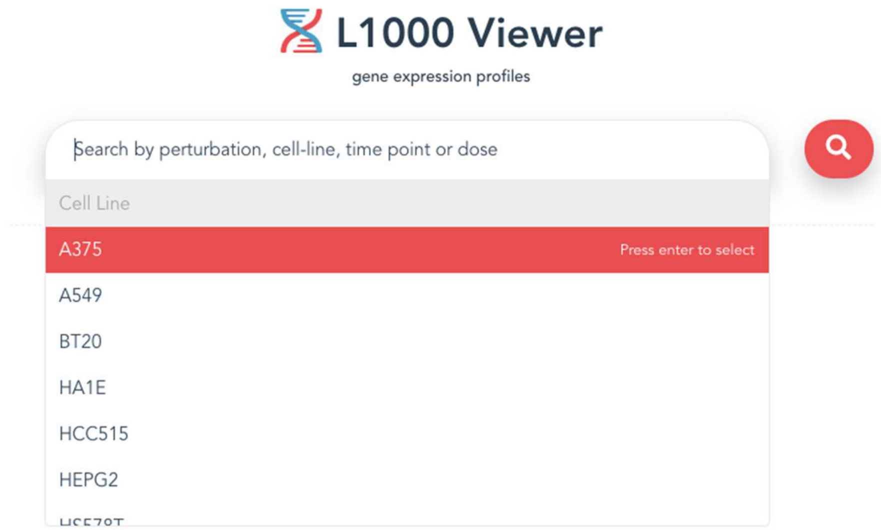


FIGURE 3 | An example of a search term that can be used to search for gene expression profiles.

For instance, in **Figure 1C** we entered the cell line A375, the drug neratinib, a dosage of 0.12μ and a duration of 24 h resulting in all profiles that are simultaneously indexed by cell line A375 AND drug neratinib AND a dosage of 0.12μ AND a duration of 24 h. The user can obtain a comprehensive list of available options directly from the L1000 interface by selecting the search field with a mouse click. This will open a pull-down menu that lists all available options that can be used as a search term in the query. Overall, the major categories for a query are cell lines, drugs and small compounds, dosages and time points.

A query finds any entity that exists among the treatment and control profiles. All queries will return a table of profiles listing unique ID numbers (e.g., LINCS profile ID, Compound), and if selected, a listing of metadata associated with the experiment will also be included in the download link. The interface is the data access point into the L1000 data repository.

The result from a query may be downloaded as a matrix of gene expression profiles. The array contains, for every gene, a binary vector representing the probe signal from the gene expression experiments (Subramanian et al., 2017). We converted the probe IDs to gene symbols for global representation. The L1000 Viewer allows the user to download complete matrices in either .csv or .csv.gz format, conferring flexibility to choose among alternative software analysis packages with optimal criteria and easy matrix subseting.

3.2. Constituting Components

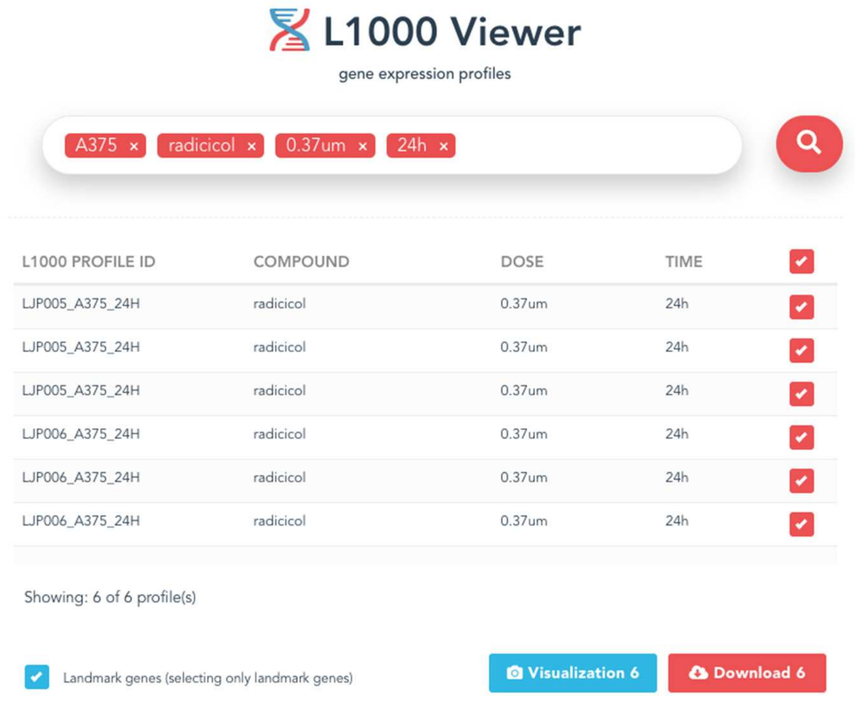
3.2.1. Data Available for Download

A large collection of almost two million L1000 gene expression profile data can be downloaded from the web interface, including the aforementioned GSE70138 from the LJP, CPC and CPD data repositories. Our application provides an easy-to-use and user-friendly interface to query the data repository, simply by searching for the desired experimental conditions.

From the L1000 Viewer web interface, metadata attributes can be used as input keyword to query the data repository. Any metadata associated with the input search can be entered in the search box. By default, the section provides four input fields for metadata: Cell, Perturbation, Dosage, and Time Point (**Figure 1**). Users can add new search terms for specific types of metadata by typing in the search box or remove one by clicking the close (x) sign on the right hand side of each keyword. The tag field is used to enter the keywords which are most descriptive of the input metadata.

3.3. Search Input

The entry point for our L1000 Viewer is to input a search term or a list of metadata query terms in the search box (see **Figure 2**) or paste a symbol (see **Figure 3**) into the search box. In order to provide guidance for setting search parameters, a query term is a list of cell lines, drug compounds, dosages or time points. The search button will only become enabled when the text box



L1000 PROFILE ID	COMPOUND	DOSE	TIME	✓
LJP005_A375_24H	radicicol	0.37um	24h	✓
LJP005_A375_24H	radicicol	0.37um	24h	✓
LJP005_A375_24H	radicicol	0.37um	24h	✓
LJP006_A375_24H	radicicol	0.37um	24h	✓
LJP006_A375_24H	radicicol	0.37um	24h	✓
LJP006_A375_24H	radicicol	0.37um	24h	✓

Showing: 6 of 6 profile(s)

☑ Landmark genes (selecting only landmark genes)

FIGURE 4 | An example of the displayed search results using all four metadata information.

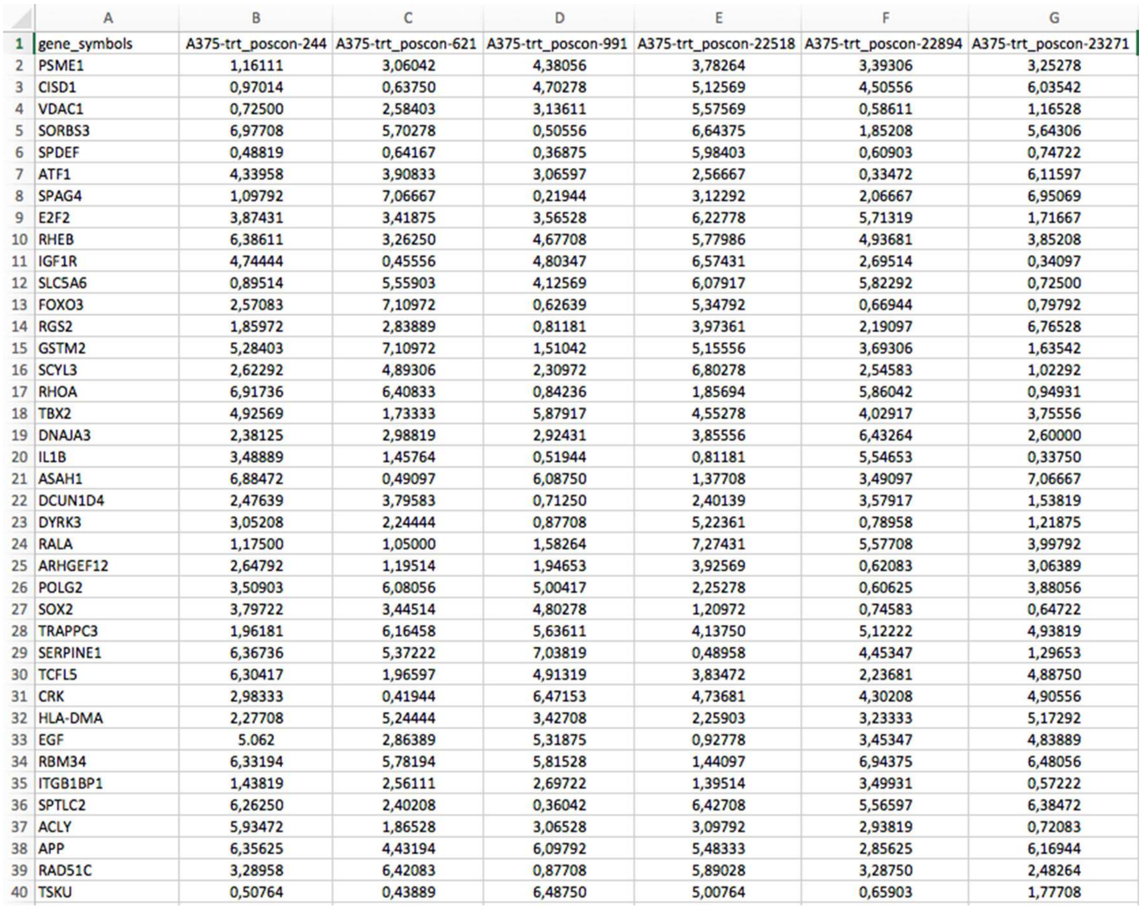
is filled with a search term, or when the text box is filled with a selection from the drop down list. By clicking the search button, the information for the top 50 samples will be displayed in a table below the search box. The interface provides the user with a user-friendly scrolling functionality for displaying more than 50 results.

3.4. Search Results

When a user successfully submits a query, the application will search and retrieve the corresponding profiles that match the user's input term and display the results. The performance of the search results will depend on the user-defined input terms. However, for any given query the application will guarantee fast results within milliseconds. In contrast, when the data is manually processed and retrieved directly from the LINCS data repositories a similar process can consume up to one day.

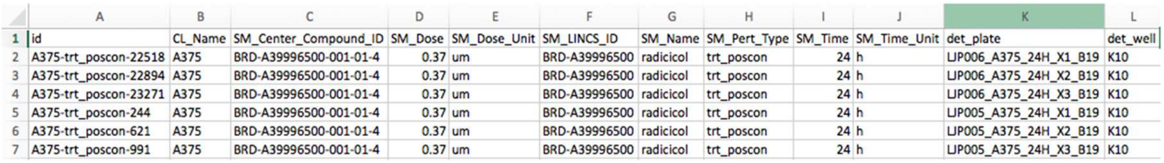
3.5. Download View

After the search results are displayed, the user can select individual profiles in the search results using the check box to fine-tune the results or decide to download all results by checking the first selection box. Then a download button will appear at the bottom of the page in the right corner. Clicking on this button will bring up the download view. The download button will generate and download gene expression profiles and signatures



	A	B	C	D	E	F	G
1	gene_symbols	A375-trt_poscon-244	A375-trt_poscon-621	A375-trt_poscon-991	A375-trt_poscon-22518	A375-trt_poscon-22894	A375-trt_poscon-23271
2	PSME1	1,16111	3,06042	4,38056	3,78264	3,39306	3,25278
3	CISD1	0,97014	0,63750	4,70278	5,12569	4,50556	6,03542
4	VDAC1	0,72500	2,58403	3,13611	5,57569	0,58611	1,16528
5	SORBS3	6,97708	5,70278	0,50556	6,64375	1,85208	5,64306
6	SPDEF	0,48819	0,64167	0,36875	5,98403	0,60903	0,74722
7	ATF1	4,33958	3,90833	3,06597	2,56667	0,33472	6,11597
8	SPAG4	1,09792	7,06667	0,21944	3,12292	2,06667	6,95069
9	E2F2	3,87431	3,41875	3,56528	6,22778	5,71319	1,71667
10	RHEB	6,38611	3,26250	4,67708	5,77986	4,93681	3,85208
11	IGF1R	4,74444	0,45556	4,80347	6,57431	2,69514	0,34097
12	SLC5A6	0,89514	5,55903	4,12569	6,07917	5,82292	0,72500
13	FOXO3	2,57083	7,10972	0,62639	5,34792	0,66944	0,79792
14	RGS2	1,85972	2,83889	0,81181	3,97361	2,19097	6,76528
15	GSTM2	5,28403	7,10972	1,51042	5,15556	3,69306	1,63542
16	SCYL3	2,62292	4,89306	2,30972	6,80278	2,54583	1,02292
17	RHOA	6,91736	6,40833	0,84236	1,85694	5,86042	0,94931
18	TBX2	4,92569	1,73333	5,87917	4,55278	4,02917	3,75556
19	DNAJA3	2,38125	2,98819	2,92431	3,85556	6,43264	2,60000
20	IL1B	3,48889	1,45764	0,51944	0,81181	5,54653	0,33750
21	ASAH1	6,88472	0,49097	6,08750	1,37708	3,49097	7,06667
22	DCUN1D4	2,47639	3,79583	0,71250	2,40139	3,57917	1,53819
23	DYRK3	3,05208	2,24444	0,87708	5,22361	0,78958	1,21875
24	RALA	1,17500	1,05000	1,58264	7,27431	5,57708	3,99792
25	ARHGEF12	2,64792	1,19514	1,94653	3,92569	0,62083	3,06389
26	POLG2	3,50903	6,08056	5,00417	2,25278	0,60625	3,88056
27	SOX2	3,79722	3,44514	4,80278	1,20972	0,74583	0,64722
28	TRAPPC3	1,96181	6,16458	5,63611	4,13750	5,12222	4,93819
29	SERPINE1	6,36736	5,37222	7,03819	0,48958	4,45347	1,29653
30	TCFL5	6,30417	1,96597	4,91319	3,83472	2,23681	4,88750
31	CRK	2,98333	0,41944	6,47153	4,73681	4,30208	4,90556
32	HLA-DMA	2,27708	5,24444	3,42708	2,25903	3,23333	5,17292
33	EGF	5.062	2,86389	5,31875	0,92778	3,45347	4,83889
34	RBM34	6,33194	5,78194	5,81528	1,44097	6,94375	6,48056
35	ITGB1BP1	1,43819	2,56111	2,69722	1,39514	3,49931	0,57222
36	SPTLC2	6,26250	2,40208	0,36042	6,42708	5,56597	6,38472
37	ACLY	5,93472	1,86528	3,06528	3,09792	2,93819	0,72083
38	APP	6,35625	4,43194	6,09792	5,48333	2,85625	6,16944
39	RAD51C	3,28958	6,42083	0,87708	5,89028	3,28750	2,48264
40	TSKU	0,50764	0,43889	6,48750	5,00764	0,65903	1,77708

FIGURE 5 | Example for a gene expression profile matrix (matrix.csv) available in the download zip folder.



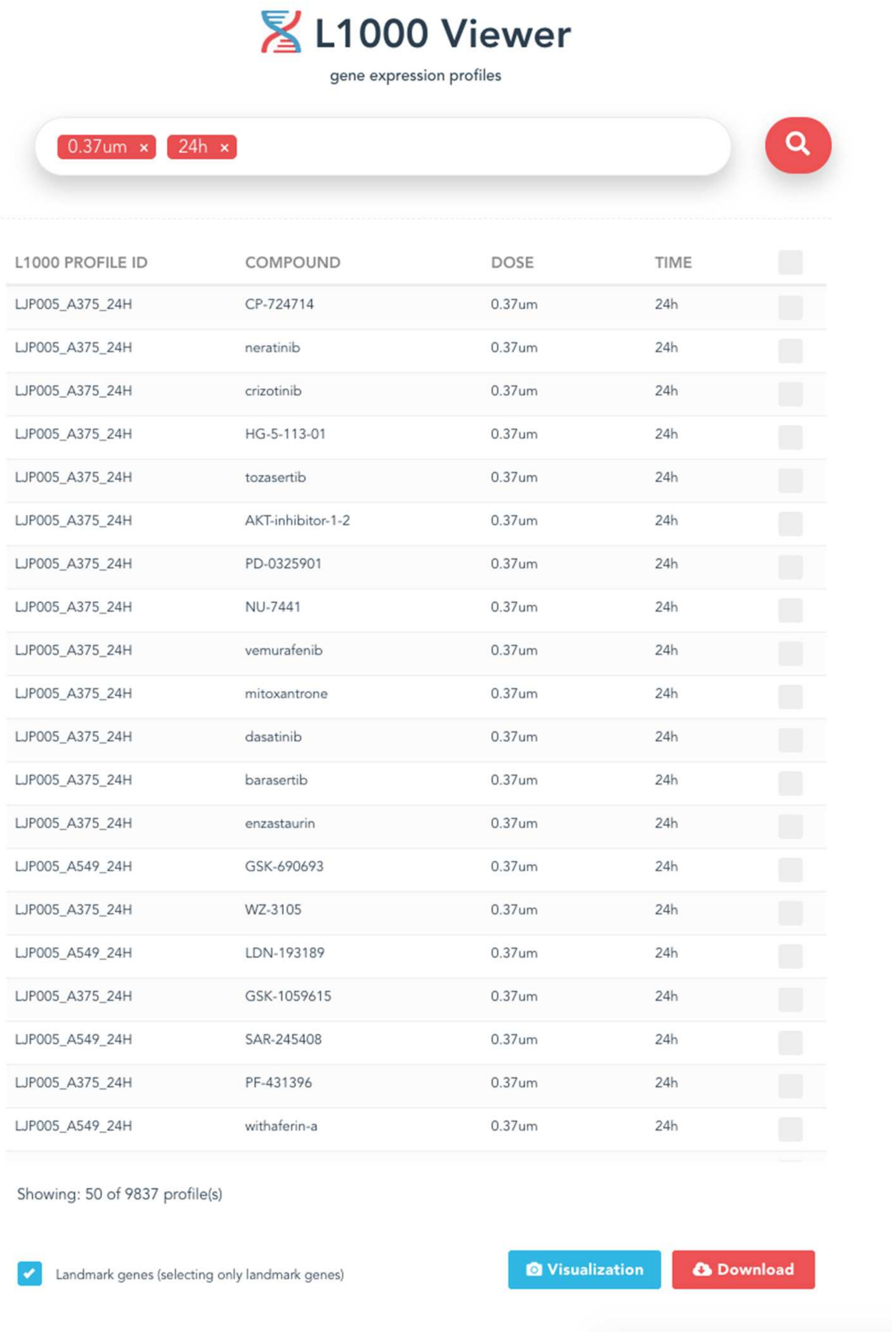
	A	B	C	D	E	F	G	H	I	J	K	L
1	id	CL_Name	SM_Center_Compound_ID	SM_Dose	SM_Dose_Unit	SM_LINCS_ID	SM_Name	SM_Pert_Type	SM_Time	SM_Time_Unit	det_plate	det_well
2	A375-trt_poscon-22518	A375	BRD-A39996500-001-01-4	0.37	um	BRD-A39996500	radicicol	trt_poscon	24	h	LJP006_A375_24H_X1_B19	K10
3	A375-trt_poscon-22894	A375	BRD-A39996500-001-01-4	0.37	um	BRD-A39996500	radicicol	trt_poscon	24	h	LJP006_A375_24H_X2_B19	K10
4	A375-trt_poscon-23271	A375	BRD-A39996500-001-01-4	0.37	um	BRD-A39996500	radicicol	trt_poscon	24	h	LJP006_A375_24H_X3_B19	K10
5	A375-trt_poscon-244	A375	BRD-A39996500-001-01-4	0.37	um	BRD-A39996500	radicicol	trt_poscon	24	h	LJP005_A375_24H_X1_B19	K10
6	A375-trt_poscon-621	A375	BRD-A39996500-001-01-4	0.37	um	BRD-A39996500	radicicol	trt_poscon	24	h	LJP005_A375_24H_X2_B19	K10
7	A375-trt_poscon-991	A375	BRD-A39996500-001-01-4	0.37	um	BRD-A39996500	radicicol	trt_poscon	24	h	LJP005_A375_24H_X3_B19	K10

FIGURE 6 | Example for a metadata matrix (metadata.csv) available in the download zip folder.

selected within the search results as .csv files, and will also include the metadata information associated with the profiles in a zipped folder. an example is shown in **Figure 4**.

3.6. Files

There are two files generated that are available in the zipped folder. The first is a comma-separated data matrix file (.csv) named "matrix.csv." It contains the gene expression profiles of the downloaded dataset as shown in **Figure 5**. The rows in the file correspond to all the gene symbol annotations for each profile and the columns correspond to the samples. A second file contains the meta description of the profiles. It is also a comma-separated file named "metadata.csv." This file contains the meta-information of the experiment of each profile, such as



L1000 PROFILE ID	COMPOUND	DOSE	TIME	□
LJP005_A375_24H	CP-724714	0.37um	24h	□
LJP005_A375_24H	neratinib	0.37um	24h	□
LJP005_A375_24H	crizotinib	0.37um	24h	□
LJP005_A375_24H	HG-5-113-01	0.37um	24h	□
LJP005_A375_24H	tozasertib	0.37um	24h	□
LJP005_A375_24H	AKT-inhibitor-1-2	0.37um	24h	□
LJP005_A375_24H	PD-0325901	0.37um	24h	□
LJP005_A375_24H	NU-7441	0.37um	24h	□
LJP005_A375_24H	vemurafenib	0.37um	24h	□
LJP005_A375_24H	mitoxantrone	0.37um	24h	□
LJP005_A375_24H	dasatinib	0.37um	24h	□
LJP005_A375_24H	barasertib	0.37um	24h	□
LJP005_A375_24H	enzastaurin	0.37um	24h	□
LJP005_A549_24H	GSK-690693	0.37um	24h	□
LJP005_A375_24H	WZ-3105	0.37um	24h	□
LJP005_A549_24H	LDN-193189	0.37um	24h	□
LJP005_A375_24H	GSK-1059615	0.37um	24h	□
LJP005_A549_24H	SAR-245408	0.37um	24h	□
LJP005_A375_24H	PF-431396	0.37um	24h	□
LJP005_A549_24H	withaferin-a	0.37um	24h	□

Showing: 50 of 9837 profile(s)

☑ Landmark genes (selecting only landmark genes)

FIGURE 7 | Sample query displaying drug profiles that are treated on different cell lines with 0.37 um concentration and 24 h time point. In total, 9,837 profiles have been retrieved.

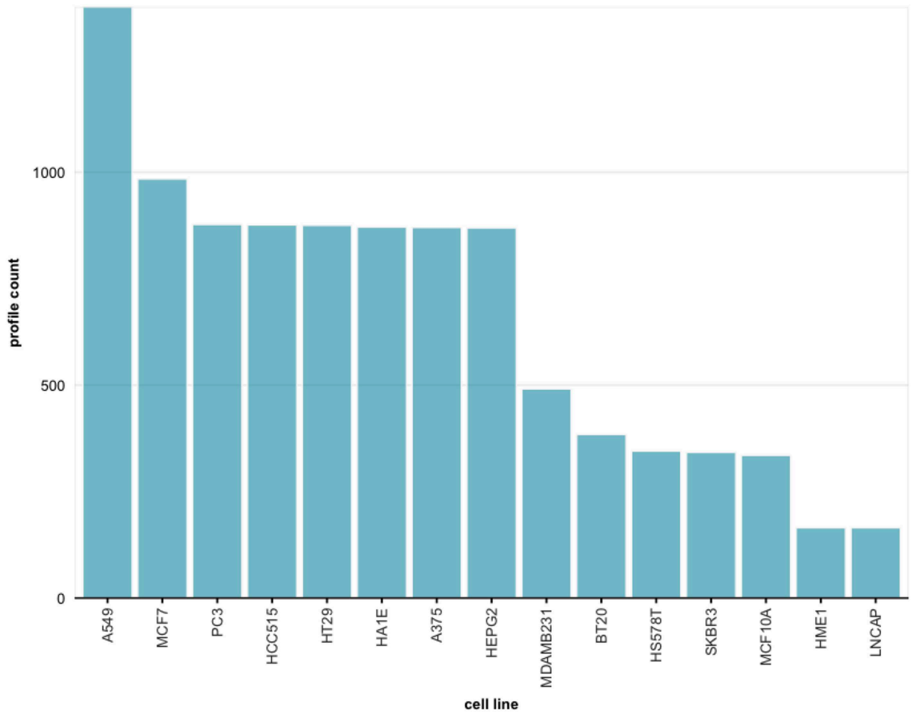


FIGURE 8 | Frequency distribution for cell lines across all experiments retrieved from the query in **Figure 7**.

time points, dosages, profile IDs, etc. The content of the file is shown in **Figure 6**.

3.7. Data Visualization: An Example

In addition to the above search and downloading capability of our L1000 viewer, described above, we provide a graphical summarization of the selected files. Specifically, we provide code that can be used to plot (in an R environment) the statistical distributions of cell lines, dosage concentrations or time points. A user can make use of the scripts to visualize the data obtained directly from a specified query.

For instance, from the query shown in **Figure 7**, setting the concentration to 0.37um and the time points to 24h, 9, 837 profiles are obtained. In **Figure 8** we show the distribution of these 9, 837 profiles over 15 cell lines. Here we leverage the metadata annotations downloaded along with the expression profiles obtained from the Cell Service API to show the distribution of each cell line.

eFor the same query we obtain the distribution of different concentrations of small molecule perturbagens, shown in **Figure 9A**. One can see that there are more than 9 different concentrations available in this data set. The compound information for small molecule perturbagens was retrieved using the Pert service API to identify unique and common compounds used in the L1000 data.

Finally, in **Figure 9B** we show the distribution of available time points in the data set. The R code and guidelines are provided from the web interface in order to subset and visualize the L1000 dataset using user specific query.

3.8. L1000 Viewer Accessibility

Access to the data indexed by the L1000 Viewer is provided through our web interface via http://L1000viewer.bio-complexity.com/. It enhances the biomedical data repository by providing a simple and fast access to LINCS raw data and allows to easily generate subsets of data. In this way, users of the web interface can extract knowledge more efficiently when interfacing with LINCS data.

3.9. Code Availability

All code associated with the L1000 Viewer project is open source. The code is available from the BitBuket repository (https://bitbucket.org/aliocee/devcrew/src/master/). The L1000 Viewer libraries are versioned according to the Semantic Versioning 2.0.0 guidelines (http://semver.org/).

4. DISCUSSION

Advances in experimental and computational methods in biomedical research are now producing large volumes of digital data objects that are rapidly accumulating. At the same time, a variety of bioinformatics tools to handle the analysis of all this data are promptly being developed and published. However, systematic linking of digital data entities for easy access are currently lacking most especially for the LINCS L1000 raw data.

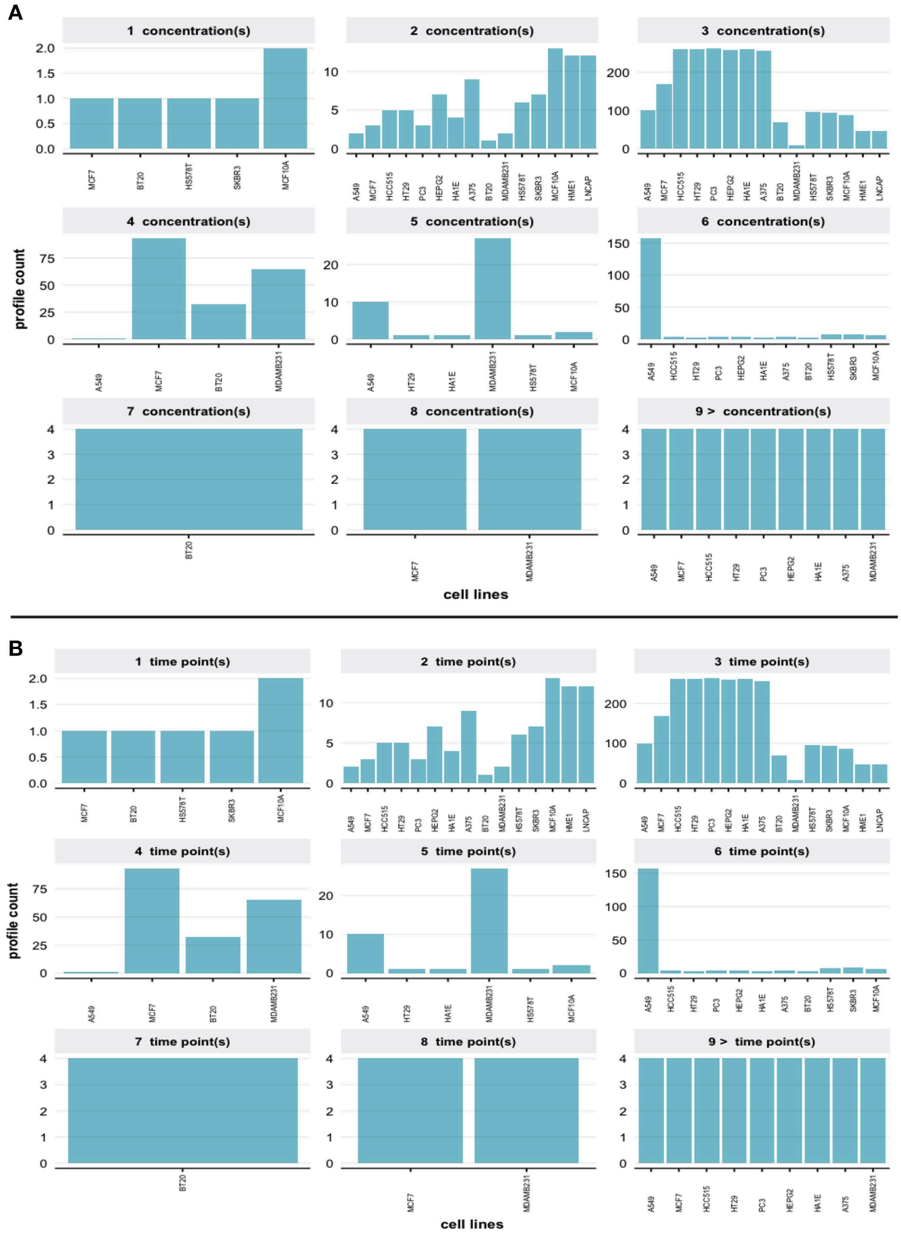


FIGURE 9 | (A) Distributions of different dosages (concentrations) of small molecules for the query in **Figure 7**. **(B)** Distributions of different time points for the query in **Figure 7**.

That means there is a gap between the data availability and how much of it can be employed in applications for extracting useful knowledge.

Previous attempts to build gene expression content-based databases have provided new support for perturbational data accessibility (Subramanian et al., 2017; Wang et al., 2018). The data within these databases is structured, and thus suitable for data access; however, most attempts to represent such data only succeeded in accomplishing this in a complex representation. For example, web-based platforms such as the CLUE Platform (Li et al., 2019), LINCS Data Portal (Koleti et al., 2017), L1000FWD (Wang et al., 2018) or iLINCS (Keenan et al., 2018) provide information about signature profiles and metadata, but there are no easy-to-use resources that enable the user to access selected raw data. Specifically, the CLUE Platform is one of the most comprehensive resources for collective knowledge about the LINCS project and L1000 data, aggregating information from over 20, 000 perturbagens and 400, 000 signature profiles. However, the CLUE Platform is very complex and does not provide direct access to the raw data.

Instead, it provides an open and free API for accessing metadata. Moreover, most of these platforms operate on metadata like annotated cell lines, proteins, and small molecules. Still they lack the simplicity and interactivity for users to access the data (Vempati et al., 2014). In comparison, our L1000 Viewer provides an easy-to-use interface for searching and downloading raw data.

The L1000 Viewer web application will enable the user to easily search the LINCS L1000 raw data via an interactive web interface. The L1000 Viewer is built using Javascript libraries, and is deployed as a Node.js application (Tilkov and Vinoski, 2010) in order to provide quick access. Its front end interface utilizes the core Vue.js libraries (You, 2017) and all gene expression and metadata are stored in a MongoDB database. Furthermore, we developed and integrated an API in our application that enables users to search the LINCS data repository and to automatically generate data for download.

In contrast to stand-alone software that needs to be installed locally on a computer, our L1000 Viewer is a web application that can be accessed via any web browser without the need of installing software on a computer locally. This makes it not only easy to access but ensures also an operating system independent functioning.

5. CONCLUSION

In this paper, we introduced the L1000 Viewer (http://L1000viewer.bio-complexity.com/), a search engine and graphical web interface for the LINCS data repository. The core of our L1000 Viewer is a database that utilizes the intricate dependency structure among the files in the LINCS data. This resulted in a reorganization of the files and enables efficient search capabilities based on graph-oriented operations.

Overall, the L1000 Viewer provides a useful tool for efficiently accessing exclusive information from the LINCS data repository that can be utilized for computational pharmacogenomics studies (Hopkins, 2008; Davis and Chawla, 2011; Emmert-Streib et al., 2013; Himmelstein et al., 2017), e.g., for drug repurposing and cancer therapeutics, as well as for understanding the composition and relationships between genes, drugs and diseases.

DATA AVAILABILITY

Publicly available datasets were analyzed in this study. This data can be found here: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70138.

AUTHOR CONTRIBUTIONS

FE-S conceived this study. AM and ST performed the analysis. AM, ST, MD, and FE-S wrote the paper and approved the final version of the manuscript.

FUNDING

MD thanks the Austrian Science Funds for supporting this work (project P 30031).

ACKNOWLEDGMENTS

We would like to thank Ricardo de Matos Simoes for fruitful discussions.

REFERENCES

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., et al. (2013). Ncbi geo: archive for functional genomics data sets—update. *Nucleic Acids Res.* 41, D991–D995. doi: 10.1093/nar/gks1193

Davis, D. A., and Chawla, N. V. (2011). Exploring and exploiting disease interactions from multi-relational gene and phenotype networks. *PLoS ONE* 6:e22670. doi: 10.1371/journal.pone.0022670

Duan, Q., Reid, S. P., Clark, N. R., Wang, Z., Fernandez, N. F., Rouillard, A. D., et al. (2016). L1000cds2: lincs l1000 characteristic direction signatures search engine. *npj Syst. Biol. Appl.* 2:16015. doi: 10.1038/npjsba.2016.15

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* 30, 207–210. doi: 10.1093/nar/30.1.207

Emmert-Streib, F., and Dehmer, M. (2019). Defining data science by a data-driven quantification of the community. *Mach. Learn. Knowledge Extract.* 1, 235–251. doi: 10.3390/make1010015

Emmert-Streib, F., Tripathi, S., de Matos Simoes, R., Hawwa, A., and Dehmer, M. (2013). The human disease network: opportunities for classification, diagnosis and prediction of disorders and disease genes. *Syst. Biomed.* 1, 20–28. doi: 10.4161/sysb.22816

Enache, O. M., Lahr, D. L., Natoli, T. E., Litichevskiy, L., Wadden, D., Flynn, C., et al. (2017). The gctx format and cmapPy, R, M packages: resources for the optimized storage and integrated traversal of dense matrices of data and annotations. *Bioinformatics* 35, 1427–1429 doi: 10.1093/bioinformatics/bty784

Fallahi-Sichani, M., Becker, V., Izar, B., Baker, G. J., Lin, J. R., Boswell, S. A., et al. (2017). Adaptive resistance of melanoma cells to raf inhibition via reversible induction of a slowly dividing de-differentiated state. *Mol. Syst. Biol.* 13:905. doi: 10.15252/msb.20166796

Himmelstein, D. S., Lizee, A., Hessler, C., Brueggeman, L., Chen, S. L., Hadley, D., et al. (2017). Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife* 6:e26726. doi: 10.7554/eLife.26726

Hinkelmann, K., and Kempthorne, O. (2008). *Design and Analysis of Experiments: Introduction to Experimental Design*. Chichester: Wiley-Interscience.

Hopkins, A. L. (2008). Network pharmacology: the next paradigm in drug discovery. *Nat. Chem. Biol.* 4, 682–690. doi: 10.1038/nchembio.118

Jin, X., Wah, B. W., Cheng, X., and Wang, Y. (2015). Significance and challenges of big data research. *Big Data Res.* 2, 59–64. doi: 10.1016/j.bdr.2015.01.006

Keenan, A. B., Jenkins, S. L., Jagodnik, K. M., Koplev, S., He, E., Torre, D., et al. (2018). The library of integrated network-based cellular signatures nih program: system-level cataloging of human cells response to perturbations. *Cell Syst.* 6, 13–24. doi: 10.1016/j.cels.2017.11.001

Koleti, A., Terryn, R., Stathias, V., Chung, C., Cooper, D. J., Turner, J. P., et al. (2017). Data portal for the library of integrated network-based cellular signatures (lincs) program: integrated access to diverse large-scale cellular perturbation response data. *Nucleic Acids Res.* 46, D558–D566. doi: 10.1093/nar/gkx1063

Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., et al. (2006). The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science* 313, 1929–1935. doi: 10.1126/science.1132939

Li, A., Lu, X., Natoli, T., Bittker, J., Sipes, N. S., Subramanian, A., et al. (2019). The carcinogenome project: *in vitro* gene expression profiling of chemical perturbations to predict long-term carcinogenicity. *Environ. Health Perspect.* 127:047002. doi: 10.1289/EHP3986

Ma'ayan, A., Rouillard, A. D., Clark, N. R., Wang, Z., Duan, Q., and Kou, Y. (2014). Lean big data integration in systems biology and systems pharmacology. *Trends Pharmacol. Sci.* 35, 450–60. doi: 10.1016/j.tips.2014.07.001

Musa, A., Dehmer, M., Yli-Harja, O., and Emmert-Streib, F. (2018). Exploiting genomic relations in big data repositories by graph-based search methods. *Mach. Learn. Knowl. Extr.* 1, 205–210. doi: 10.3390/make1010012

Musa, A., Ghoraie, L. S., Zhang, S.-D., Glazko, G., Yli-Harja, O., Dehmer, M., et al. (2017). A review of connectivity map and computational approaches in pharmacogenomics. *Brief. Bioinform.* 18:903. doi: 10.1093/bib/bbx023

Ong, E., Xie, J., Ni, Z., Liu, Q., Sarntivijai, S., Lin, Y., et al. (2017). Ontological representation, integration, and analysis of lincs cell line cells and their cellular responses. *BMC Bioinformatics* 18:556. doi: 10.1186/s12859-017-1981-5

Rahmatallah, Y., Zybailov, B., Emmert-Streib, F., and Glazko, G. (2017). GSAR: Bioconductor package for gene set analysis in R. *BMC Bioinform.* 18:61. doi: 10.1186/s12859-017-1482-6

Smirnov, P., Safikhani, Z., El-Hachem, N., Wang, D., She, A., and Olsen, C. (2016). PharmacoGx: an R package for analysis of large pharmacogenomic datasets. *Bioinformatics* 32, 1244–1246. doi: 10.1093/bioinformatics/btv723

Smith, F. J. (2006). Data science as an academic discipline. *Data Sci. J.* 5, 163–164. Available online at: https://www.jstage.jst.go.jp/article/dsj/5/0/5_0_163/_article/-char/ja/

Stupnikov, A., Tripathi, S., de Matos Simoes, R., McArt, D., Salto-Tellez, M., Glazko, G., et al. (2016). samExploreR: exploring reproducibility and robustness of RNA-seq results based on SAM files. *Bioinformatics* 32, 3345–3347. doi: 10.1093/bioinformatics/btw475

Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., et al. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* 171, 1437–1452.e17. doi: 10.1016/j.cell.2017.10.049

Tilkov, S., and Vinoski, S. (2010). Node.js: using javascript to build high-performance network programs. *IEEE Int. Comput.* 14, 80–83. doi: 10.1109/MIC.2010.145

Tripathi, S., Dehmer, M., and Emmert-Streib, F. (2014). NetBioV: an R package for visualizing large-scale data in network biology. *Bioinformatics* 30, 2834–2836. doi: 10.1093/bioinformatics/btu384

Vempati, U. D., Chung, C., Mader, C., Koleti, A., Datar, N., Vidović, D., et al. (2014). Metadata standard and data exchange specifications to describe, model, and integrate complex and diverse high-throughput screening data from the library of integrated network-based cellular signatures (lincs). *J. Biomol. Screen.* 19, 803–816. doi: 10.1177/1087057114522514

Wang, Z., Lachmann, A., Keenan, A. B., and Ma'ayan, A. (2018). L1000fwd: fireworks visualization of drug-induced transcriptomic signatures. *Bioinformatics* 34, 2150–2152

Woo, J. H., Shimoni, Y., Yang, W. S., Subramaniam, P., Iyer, A., Nicoletti, P., et al. (2015). Elucidating compound mechanism of action by network perturbation analysis. *Cell* 162, 441–451. doi: 10.1016/j.cell.2015.05.056

You, E. (2017). *Vuejs Javascript Framework*. Available online at: https://vuejs.org/

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.