Computer Methods and Programs in Biomedicine 212 (2021) 106496

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

FIMED: Flexible management of biomedical data

Sandro Hurtado[a,*], José García-Nieto[a,b,c], Ismael Navas-Delgado[a,b,c], José F. Aldana-Montes[a,b,c]

[a] Khaos Research, ITIS Software, Universidad de Málaga, Arquitecto Francisco Peñalosa 18, Málaga, 29071, Spain
[b] Biomedical Research Institute of Málaga (IBIMA), Universidad de Málaga, Málaga, Spain
[c] Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain

ARTICLE INFO

Article history:
Received 4 May 2021
Accepted 18 October 2021

Keywords:
Clinical trial management systems
Clinical research
NoSQL database
MongoDB
Gene expression data analysis
Melanoma disease

ABSTRACT

Background and objectives: In the last decade, clinical trial management systems have become an essential support tool for data management and analysis in clinical research. However, these clinical tools have design limitations, since they are currently not able to cover the needs of adaptation to the continuous changes in the practice of the trials due to the heterogeneous and dynamic nature of the clinical research data. These systems are usually proprietary solutions provided by vendors for specific tasks. In this work, we propose FIMED, a software solution for the flexible management of clinical data from multiple trials, moving towards personalized medicine, which can contribute positively by improving clinical researchers quality and ease in clinical trials.

Methods: This tool allows a dynamic and incremental design of patients' profiles in the context of clinical trials, providing a flexible user interface that hides the complexity of using databases. Clinical researchers will be able to define personalized data schemas according to their needs and clinical study specifications. Thus, FIMED allows the incorporation of separate clinical data analysis from multiple trials.

Results: The efficiency of the software has been demonstrated by a real-world use case for a clinical assay in Melanoma disease, which has been indeed anonymized to provide a user demonstration. FIMED currently provides three data analysis and visualization components, guaranteeing a clinical exploration for gene expression data: heatmap visualization, clusterheatmap visualization, as well as gene regulatory network inference and visualization. An instance of this tool is freely available on the web at https://khaos.uma.es/fimed. It can be accessed with a demo user account, *"researcher"*, using the password *"demo"*.

Conclusion: This paper shows FIMED as a flexible and user-friendly way of managing multidimensional clinical research data. Hence, without loss of generality, FIMED is flexible enough to be used in the context of any other disease where clinical data and assays are involved.

1. Introduction

Current advances in Next-Generation Sequencing (NGS) together with the consequent fast-growth and availability of biological data [1,2], enable practitioners to combine these data with other clinical and personal information of patients, such as: electronic health records, habits, inheritance and environmental factors; and therefore perform deeper analyses [3]. This is promoting the development of sophisticated tools for data management and analysis in clinical research and personalized medicine [4–6].

Managing clinical data involved in NGS studies is a challenging task [7], given the continuous obstacles encountered in the system maintenance during patient enrollment, the acquisition process of clinical study samples and the different steps for the preparation of processing pipelines of clinical data. Most of these difficulties are indeed produced due to the dynamic and heterogeneous nature of clinical data [8]. The variability of clinical data with respect to the type of data requires special attention in data management sys-

* Corresponding author.
E-mail address: sandrohr@uma.es (S. Hurtado).

https://doi.org/10.1016/j.cmpb.2021.106496

tems, since a large volumes of heterogeneous data are integrated from multiple sources with different structures and data formats.

There is a myriad of research efforts implementing software applications focused on the management of clinical information, which traditionally relied on the use of relational database management systems, such as MySQL, Oracle, or Microsoft SQL Server [9,10]. Although the relational data model is the longest established approach to carry out data management, it introduces certain limitations when dealing with clinical data [6]. In this sense, since relational databases require the schema design to be set up before introducing data, this demands Software Engineers to know the structure of the data that will be stored and the characteristics they possess in advance [11]. Later modifications in the schema, once the users are introducing data through an application, are complex as they need to be done by engineers and can have consequences, such as data lost and data inconsistencies [12]. However, data collection could produce cases where new clinical variables need to be considered [13]. For this reason, using relational databases to store clinical research data would cause dispersed tables with empty fields as a result of schema changes. In consequence, we can identify several features that would be of interest in clinical research tools [14]:

- Dynamically storing the clinical data from multiple clinical trials;
- Allowing to expand their functionalities;
- Integrating data from different clinical operations in multiple systems;
- Transferring data to different types of samples to target different analysis;
- Being adequate to the special characteristics of clinical data;
- Using a database schema that grants sufficient adaptability to face the continuous changes in the practice of clinical trials;
- Enabling ways to secure patients' information.

There exist different software platforms available devoted to store and process large volume of heterogeneous data from multiple data sources, which are also focused on performing computation on encrypted data, in different domains of application. Some prominent examples in this sense are: TrajMesa [15], a holistic distributed NoSQL trajectory storage engine. TrajMesa can manage a extremely large number of trajectories, and support plenty of query types efficiently; PERSIST [16], is a middleware architecture that externalizes the complexity of a federated cloud, storage architecture and the complex storage logic from the SaaS application to storage policies. This platform also allows tenants to enforce different storage- and privacy related requirements at a fine-grained level, and supports the dynamic (re)configurability of the underlying federated cloud storage architecture. PERSIST offers support for run-time cross-provider polyglot persistence and the confidentiality of sensitive data through encryption; a third solution is CryptDICE [17], a distributed data protection system that provides built-in support for a number of different data encryption schemes, supports making appropriate trade-offs and execution of these encryption decisions at diverse levels of data granularity, and integrates a lightweight service that performs dynamic deployment of User Defined Functions (UDF), without performing any alteration directly in the database engine for heterogeneous NoSQL databases. This leads to realize low-latency aggregate queries and also to avoid expensive data shuffling. Finally, SecureNoSQL [18] aims at covering not only data confidentiality, but also the integrity of the datasets residing on a cloud server. In this last platform, a secure proxy carries out the required transformations and the cloud server is not modified. The construction is applicable to all NoSQL data models, specially those oriented to document-store data model.

Similarly, in the specific case of clinical data management, there are many software packages already developed, some of them freely available to clinicians [19–26]. Although the goal of this paper is not to produce a complete review on this topic, we mention here a set of those with similar features to FIMED.

In this regard, OpenClinica [21] is one of the most prominent tools designed to capture clinical trial data. This web-based tool allows to design electronic Case Report Forms (eCRF), firstly building them in any spreadsheet program and uploading them via the user interface. However, the forms must be uploaded again in the tool if users want to modify or update them in OpenClinica. For this reason, the users will be hindered as they will have to constantly load the forms into the tool due to the heterogeneity of the clinical data and the changes that may occur in the different trials.

REDCap [23] is a research electronic data capture tool where clinical researchers declare the fields in a spreadsheet using metadata and send it to the REDCap team. The computer scientists design the tables in the database and deploy the web application for the specific case. However, additional modifications in the data structure need to be approved by the REDCap team. Thus, some changes are not allowed due to the database limitations and this limits the flexibility of the system.

There are many other software tools for the management of clinical data (such as [20,22,24,25]), with more or less similar functionalities as described in some surveys [27–29]. In this regard, the survey presented in [30] indicates that most of the clinical data management systems are web-based platforms based on the needs of a specific clinical trial in the shortest possible time. Therefore, these systems do not fully support the process of clinical data management and lack of flexibility and extensibility in terms of development. Similarly, as argued in [31], the systems being used to collect study data are often operated redundantly to systems used in patient care. As a consequence, the data collection in studies is inefficient and data quality may suffer from unsynchronized datasets, non-normalized database scenarios and manually executed data transfers. A solution proposed in [31] consists in OpenCampus Research, an open adoption software (OAS) solution, which provides a standard environment for state-of-the-art research database management.

However, practically none of these tools include the possibility of analyzing the clinical data of patients in terms of disease exploration.

With the motivation of approaching all these features, this paper presents FIMED, a tool for the flexible management and analysis of clinical research information. It is a "*do-it-yourself*" tool that allows users to build their forms in a simple, incremental and dynamic way to facilitate multiple source data collection. FIMED offers an advance in the functionalities offered by these tools providing users with an easy to use tool for the flexible design (including later modifications on it) of eCRFs according to the clinician's needs.

FIMED has been developed using MongoDB to alleviate some of the limitations imposed by relational databases. MongoDB is a non-relational database (NoSQL). This document oriented database offers a schema-less approach to database design [32], where database schema does not require to be defined entirely beforehand, and the data structure can change over time without needing to update previous database entries. Thus, any new data entry can introduce schema changes without declaring them at the schema level. This allows a flexible design of the databases at the data load phase. FIMED provides a web user interface in which users focus on inserting the data they are collecting in clinical trials. As soon as they detect new fields to be added, they are just included in the latest data inserted in the database, so the model is updated. MongoDB has been designed to operate using a cluster configuration, making it a great choice if scalability is required.

In clinical trials, data collection/management is normally separated from data analytics. FIMED also provides analysis tools for

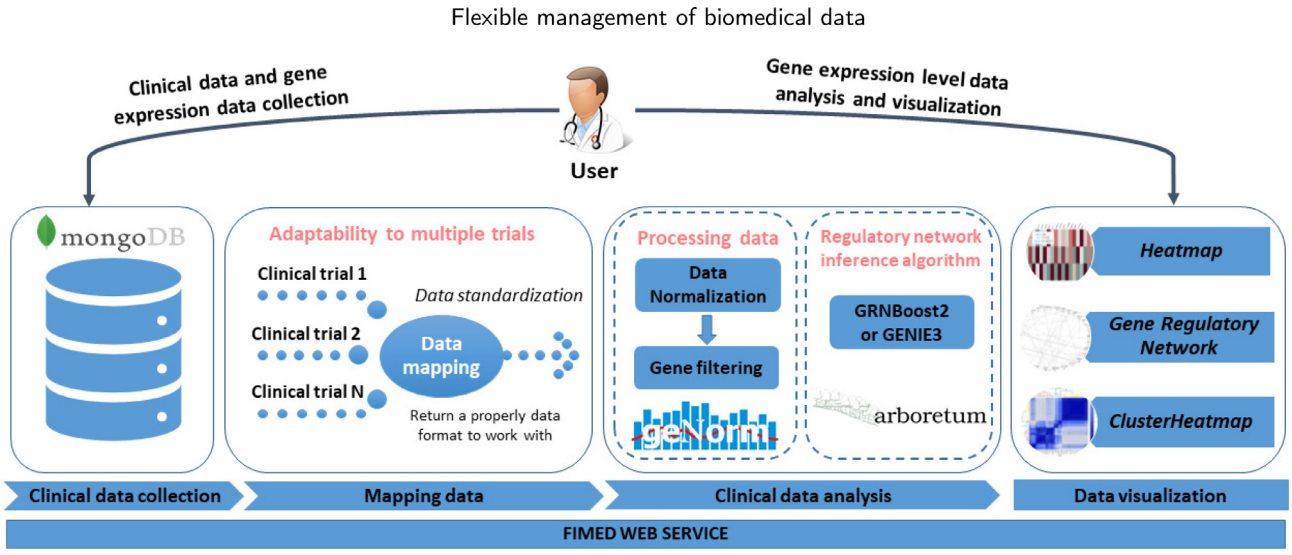

Fig. 1. FIMED Workflow. (I) Data collection, (II) Mapping data, (III) Data analysis and (IV) Data Visualization.

clinical trials in order to minimises bias. The current version enables tools for gene expression data analysis in a semi-automatic and straightforward way using heat maps, cluster heat maps and gene regulatory networks inferred from inserted data. This functionality provides practitioners with early insights into the gene expression samples of a patient (discovering changes in their gene expression levels) or sets of patients (discovering patient clusters with a possible clinical correlation).

In order to evaluate FIMED, a practical use case has been conducted with real expression data taken from metastatic Melanoma patients used in a previous work, VIGLA-M [33]. This use case replicates this work from data acquisition to the integration and analysis based on advanced visualizations. FIMED has shown to ease the process of flexible and dynamic collecting patient's clinical information without the need of a database re-engineering process.

2. System architecture

FIMED internally implements a workflow as depicted in Fig. 1, which consist in several phases: data collection, integration, analysis and visualization. Thanks to the web interface, the user is guided through this workflow, so internal data mappings and adaptations are automatically conducted.

2.1. Data collection

For the integration of clinical information and other related information, such as gene expression data, we have designed a core MongoDB schema. This schema can be observed in the JSON Code Snipped 1, as a single collection of users, each one of them corresponding to a MongoDB document in the database. Each entry (user) in this collection contains the list of patients who have undergone clinical trials with this user (clinician). The user can store clinical trial information of each patient (e.g. name, gender, date of birth, medical records, medication, diagnosis date, disease's progress, etc.) and associated files obtained from, for example, gene expression assays in different formats. Moreover, the user could attach files associated to the patient as additional information (e.g. reports in PDF, scanner images, signed informed consent, etc.).

Core JSON Schema. It constitutes the initial document structure from which the database is incrementally adding new elements and updating existing ones.

```
{   "_id": <ObjectId()>,
    "Name": <String>,
    "Surname": <String>,
    "Password": <String>,
    "Patients": [{
            "_id": <ObjectId()>,
            "_patientInformation": <Object>,
            "_files": [
            {
                "filename": <String>,
                "metadata": <Object>,
                "gridFS": <Object>
            }],
            "_clinicalSamples": [
            {
                "sample_name": <String>,
                "metadata": <Object>,
                "gridFS": <Object>
            }]
        }],
    "Form": <Object>,
    "Analysis_results": [

        {
            "name_analysis": <String>,
            "results": <String>
        }]
}
```

This database is supported with a web based front-end to facilitate the data collection, which is dynamically adapting user needs along with the database scheme. The data insertion process starts with an initial form with fields according to the core JSON schema. The user can add new fields dynamically, just indicating the field name and inserting data for this field in the current patient. The fields used in previous records are automatically shown when adding information for a new patient. The system also learns from the fields introduced by any user, so when a new field is cre-

ated, its use is available to other users, which could fill it from now on (when required). The system processes the data in real-time, and so for each field in which the user introduces information, it is automatically linked with the corresponding clinician records when applicable.

Patients' information is secured in the whole database. For this propose FIMED uses the Advanced Encryption Standard (AES). AES is an encryption algorithm [34], which employs the same secret key to encrypt and decrypt the data. FIMED uses the 256-bit keys which is the longest allowed by AES and also recommended to achieve strong data security. Hence, the encryption key (secret key) is a random combination of a suitable length of 256 bits that is generated on the server side during the registration process each time a user is registered in FIMED.

In a first instance, this encryption key is used to encrypt the user password and is also used as secret key to encrypt patients' information in the database and to decrypt the data when retrieving it by querying the database. Thus, we avoid that the clinicians registered in the application can access to information that does not correspond to them. Hence, FIMED protects the sensitive information of users and their patients.

The proper management of cryptographic keys is essential to the effective use of encryption products. Loss or corruption of these keys can lead to loss of access to systems and data, as well as making a system completely unusable unless it is reformatted and reinstalled. For this reason, FIMED saved the cryptographic keys to a MinIO[1] cluster. This cluster is an internal network only accessible from the server where the APIs are allocated. This enables to have a fine grain track on the access to these keys, replication of the information to avoid losing keys and an additional security level provided by MinIO.

2.2. Gene expression data mapping

A series of mapping processes have been developed to translate gene expression data in a format suitable for their processing and analysis. This enables the tool to perform the import of gene expression data in different NGS file formats from different providers (Nanostring, Affymetrix, etc.). In this sense, we first perform a gene expression data parsing process, since different brands of machines for gene expression profiling will produce different formats. The goal of the parsing process is to extract the gene names, class names and gene expression values to obtain a gene expression matrix. These data provide us with the level of expression of each gene in the patient's temporary samples to be later pre-processed and analyzed by the tools offered by FIMED, as observed in Fig. 2.

FIMED currently supports gene expression samples in RCC (*Reporter Code Count*) format. It has been tested with RCC examples, and also with real samples from the Immune Profiling Panel Nanostring^{TM}. Each RCC file contains the count for each target mRNA molecule in a sample. From each RCC file, we can extract Code Class, Gene Name, Accession and Count (see Fig. 2 A), essential lane attributes to subsequently carry out the normalization process and finally obtain the gene expression matrix. At the end of this step, the system has uniform gene expression data and additional transformations are done to ensure high quality results.

2.3. Gene expression pre-processing

In this step of the workflow, the gene expression files previously generated are pre-processed, since the variation of gene expression data is the aggregation of biological variations that could include possible bias or noise produced during the gene sequencing process. FIMED focuses on Normalization in this stage.

A standard normalization process is carried out in order to reduce technical variations from experiments within the different files, so the remaining variance can be attributed to the underlying biology of the system under study. It is worth noting that the most common of these variations are originated in the sample or in the platform. Thus, the normalization of this variability is essential, since the precision and accuracy of the analysis techniques in gene expression assays depend on it [35]. In this sense, normalization allows users to directly compare gene expression samples.

Samples include quality control flags: positive and negative control genes. The positive control linearity ensures that the samples maintain a certain linear relationship. Background correction is achieved with the use of negative control samples. A certain threshold is calculated as two standard deviations of the negative control values over the geometric mean of reference genes. A filtering process is carried out to filter less expressed genes. Those most stable reference (*housekeeping*) genes will be identified, using the algorithm *geNorm*[2]~ [35].These genes will be used to calculate the scale factors for the rest of the sample. In this way, we can calculate the specific normalization factors for each sample. As a result of this normalization and filtering process, the gene expression matrix is obtained (as illustrated Fig. 2 B), which will be used in the data analysis and visualization processes.

2.4. Gene expression data analysis

After the pre-processing step, the use of series of gene expression data enables the possibility of exploring how different genes are connected (through gene interactions). The analysis of these interactions is useful to produce networks of interactions focused on transcription factors (TFs). In order to infer possible interactions between them, two distinguished algorithms are provided in the *arboretum* Python package [3], which are integrated in FIMED:

- GENIE3: it decomposes the prediction of the network in different regression problems. In each problem, the expression pattern of one target gene is predicted from the expression patterns of all the input genes. GENIE3 is considered the popular classic Gene Regulatory Network Inference algorithm based on tree-based ensemble methods: Random Forests (RF) or Extra-Trees (ET) [36].
- GRNBoost2: it is an efficient algorithm for regulatory network inference using Gradient Boosting Machine regression with early-stopping regularization, based on the GENIE3 architecture [37].

The result of the gene inference analysis algorithms is a collection of regulatory interactions between transcription factors and their target genes. An example can be observed in Fig. 2C, where the importance value is the strength of the interaction. The resulting links are then used to create and visualize the gene regulatory network, which can be plotted as network graphs with different layouts (see Fig. 2F).

2.5. Visualization

After the pre-processing, the most variable genes are used to perform a variety of visualizations to provide users with a rich set of tools for validating targets, through the comparison of different patient samples or patients in the same disease stage. Hence, a set of analytic functionalities are used to discover patterns in the change of the gene expression levels. In concrete, FIMED provides three main visualizations, as represented in Fig. 2 (D,E,F):

[1] https://min.io/.

[2] https://genorm.cmgg.be/.

[3] https://arboretum.readthedocs.io/en/latest/.

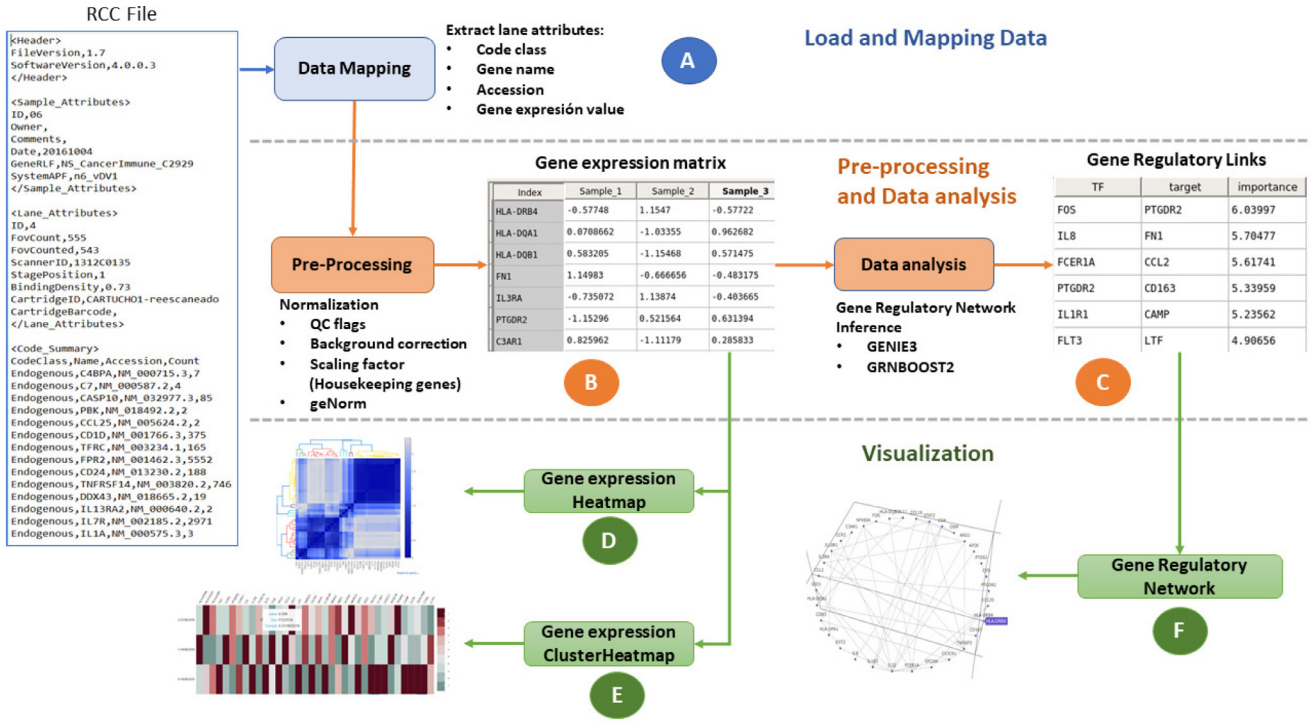

Fig. 2. FIMED provides functions for loading, mapping, pre-processing, analyzing and visualizing data from different profiling panels.

- Heatmaps. A heatmap is a graphical representation of a matrix of data, where the cell values are represented with different colors depending on their values. This is useful to visually discover relationships between elements.
- Cluster heatmaps. They follow the same principles as heatmaps, but re-ordering the matrix of data to aggregate those submatrices with similar values.
- Gene Interaction Network. The interactions between different genes can be presented as a graph where nodes represent genes and arcs the interactions between them. The arcs can represent the strength of the interaction by means of the arc shape or length.

3. Use cases

In order to enable users to explore FIMED functionalities, an instance has been deployed in our servers. In this instance users can freely manage their patient data or test it with sample data using the demo user provided[4] that contains anonymized patient's data. This sample data enables new users to explore an example of how their databases could be developed. However, to make an actual use of the tool, the user should create a new free account. Thus, users will have their own independent workspaces, where each one can only access its owns patients' data. After logging in, users can use different options in the main page (Fig. 3).

Option **Form design** (Fig. 3A) allows the user to define new data fields to include any kind of patient's clinical information. Thanks to the flexibility provided by MongoDB, the initial database schema can be then increased in a personalized way. Users can create dynamically new fields of any simple type *String, Number, Date, Boolean*. It is also possible to define compound fields with nested sub-fields, hence constituting a hierarchical organization. Once the user has designed the form (for inserting patient's data), the new fields will be stored in the database as *Attributes* (Keys in JSON and MongoDB terminology), so the database scheme is incrementally designed. It is worth noting that users could adapt the database schema to any case of study in the handling of data in clinical trials.

At this point, the process of inserting patient's information into the application is performed through option **Add patient (s)** (Fig. 3B), which enables different kind of data to be stored in the database. First, patients' clinical information is introduced in the existing data fields previously declared on the form. The user interface extracts the schema for the data insertion dynamically from the database. In this way, whenever a new field is added, it will appear in the user interface automatically. In addition, new fields are recommended to other users for future forms. Second, files containing gene expression assays associated to the patient can be loaded using the browse functionality. Accordingly, new meta-data fields could be added to the gene expression files to provide additional information to the samples. Depending on the file type, it could be used in different analyses. Third, it is also possible to insert additional files to guarantee the complete patient information (E.g. doctor's reports, test results, scanned images, signed inform consents, etc.).

Similarly, FIMED provides the user with a search engine with the purpose of helping clinicians to retrieve their patient's information. The option **Search patient (s)** (Fig. 3C) offers a dynamic interface to facilitate this functionality, as well as filter options according to existing data fields to enhance the search process. The search tool provides access not only to view the data, but also to modify them. In this sense, the patient's information can be updated at any moment. In this operation, new fields could also be created when required.

A last main option comprises the data analysis and visualization, which is offered by clicking on **Gene expression level analysis** (Fig. 3D). As commented before, FIMED currently enables three main analysis with gene expression levels that focus on: heatmaps, clusterheatmaps and gene regulatory networks. Depending on data availability, these analyses could be performed for one single patient, as well as for aggregated data from several patients, hence

[4] Demo user grants: username *"researcher"* and password *"demo"*.

Fig. 3. Main panel of the FIMED web application. The first main option is the form design, which enables the user to create its own fields with corresponding attributes in the database.

allowing comparisons among different individuals, some of them acting as control samples.

With the aim of showing the potentials of using FIMED, we have tested the tool in use cases conducted with actual sequence data from metastatic Melanoma patients [33]. Thus, we have validated the management and analytical functionalities generating indicative analysis and visualization in cancer research.

In these use cases, we have inserted clinical information of three Melanoma patients through the FIMED web service. Firstly, we have designed the form for this clinical assay and then, we have inserted the clinical information of the patients in the tool. This clinical case has a set of 5 simple fields and 1 composed field as can be seen in Code Snippet 2.

Code Snippet 2: Data Schema in Melanoma use cases.

```
{   "Form":
    {
        "_id":<ObjectId()>,
        "Attributes":
        {
            "Patient Code": <Number>,
            "Sex": <String>,
            "Birth Date": <Date>,
            "Blood pressure": <String>,
            "Observations": <String>,
            "Hospital admission":
            {
                "Hospital name": <String>,
                "Hospital address": <String>
            }
        }
    }
}
```

Moreover, we have used gene expression data using the Immune Profiling Panel Nanostring^{TM} (770 genes). This panel has been specifically designed for cancer projects studying immune aspects of the disease. The panel includes 24 different immune cell types, common checkpoint inhibitors, CT antigens, and genes covering both, the adaptive and innate immune response. For this case, the analysis component works with RCC files[5], starting from the data normalization with a housekeeping based method. This platform can analyze 12 samples in each cartridge, so it provides 12 RCC files with the gene counts for each of the gene panels. These files are stored in FIMED associated to the patient's code, the sample collection date and the experiment date.

3.1. Use case 1: heatmap clustering

At this point, users are enabled to perform a first main analysis based on the generation of heatmaps and hierarchical clusters with dendograms. As illustrated in Fig. 4, users can select gene expression samples from one or more patients to constitute the gene expression dataset to be analysed. In this process, a sliding element is provided to set a parameter for extracting only those most variable gene expression levels, as a percentage of the total number of genes in the panel. Therefore, in a given session, a series of different analysis can be generated according to this parameter. Thus, resulting clusters and heatmaps can be visually compared and inspected by a simple click-and-drag feature to zoom in, and a click-once feature to zoom out. This is a new functionality that was not present in our previous work VIGLA-M [33].

The long term goal would be to identify unexpected relationships between genes expressing in a similar way that would help identifying new drug targets or identifying new biomarkers of the patient expected evolution in their treatment. This is an ongoing work in collaboration with the regional hospital through the biomedical institute of Málaga (*Hospitales Univesitarios Regional y Virgen de la Victoria de Málaga, Instituto de Investigaciones Biomédicas - IBIMA*) using FIMED.

3.2. Use case 2: gene regulatory network

Another interesting analysis comprises the inference of gen regulatory networks, which can be now extracted from the gene expression levels previously stored in FIMED. A gene regulatory network consists of a set of genes (acting as transcription factors) that regulate (activate or inhibit) each other's expression. The nodes are the genes themselves and the connections between them represent the regulatory mechanisms of their genetic expression, such that two genes are connected if one regulates, positively or negatively, the expression of the other.

Figure 5 shows the selection panel offered in FIMED to generate and visualize gene regulatory networks. Similarly to the previous functionality, a sliding parameter is used to extract only those most variable gene expression levels, as a percentage of the total number of genes in the panel. In addition, a statistical cutoff parameter is provided to limit the maximum number of links in the network, which is useful to enhance visualization, as it just centers on most important genes and their relationships. Nonetheless, this interactive graph functionality allows the user to manually move the network and explore the connectivity between the nodes and hence, to clearly inspect the topology of network. In this regard, users can select different layouts for network representation: Force-directed layout or Circular layout.

An interesting experiment consists in inferring a set of different networks, which are obtained using different random seeds, although using the same parameters of percentage of total number of genes in the panel to 5%, and the maximum number of allowed links to 10. This way, it is possible to discover those genes that, with a high frequency, are attractors of multiple links (interactions) with other genes. These attractors are then considered as hubs in transcriptional regulatory network, which are usually identified to be used as diagnostic and prognostic markers and possibly for targeted therapy. In the case of the sample Melanoma data stored in FIMED, inferred networks are frequently generated with hubs in genes: ARG1, IL18RAP, CD163 and FCER1A. These genes are usually identified to have an adaptive resistance on immune regulatory factors in pathology [38], so this could support the clinician with new useful information for adjusting the treatment process.

4. Prototype implementation

FIMED has been developed in *JAVA, JSP*, and *JavaScript* languages and follows a Model-View-Controller (MVC) software design pattern to manage the MongoDB database. The user interface is served through a standard Tomcat 9 Web application service. FIMED provides a user-friendly web application freely available on the web at http://khaos.uma.es/fimed, with all major browsers supported. The web interface has been designed for guiding the user in the tasks of clinical data collection and database organization in an transparent and straightforward way. In addition, this tool provides gene expression data analysis by means of the visualization of clusters, anomalies, changes in patterns, etc., with open source libraries (*Plotly*[6] and *Bokeh*[7]).

FIMED also provides a Open Source version for being installed on clinicians own servers to secure patients' information. This also enables the extension of FIMED by external developers providing new functionalities. Additionally, the Model-View-Controller (MVC) software design pattern has been implemented by means of an API connecting the web user interface with MongoDB.

5. Results

Additionally, we evaluated the performance of FIMED by means of locust[8], an open source Python-based user load testing tool. We

[5] See https://khaos.uma.es/fimedRCC for examples copied from https://github.com/hbc/sen-Nanostring and so licensed under MIT License.

[6] https://plot.ly/.

[7] https://bokeh.pydata.org/en/latest/.

[8] https://locust.io/.

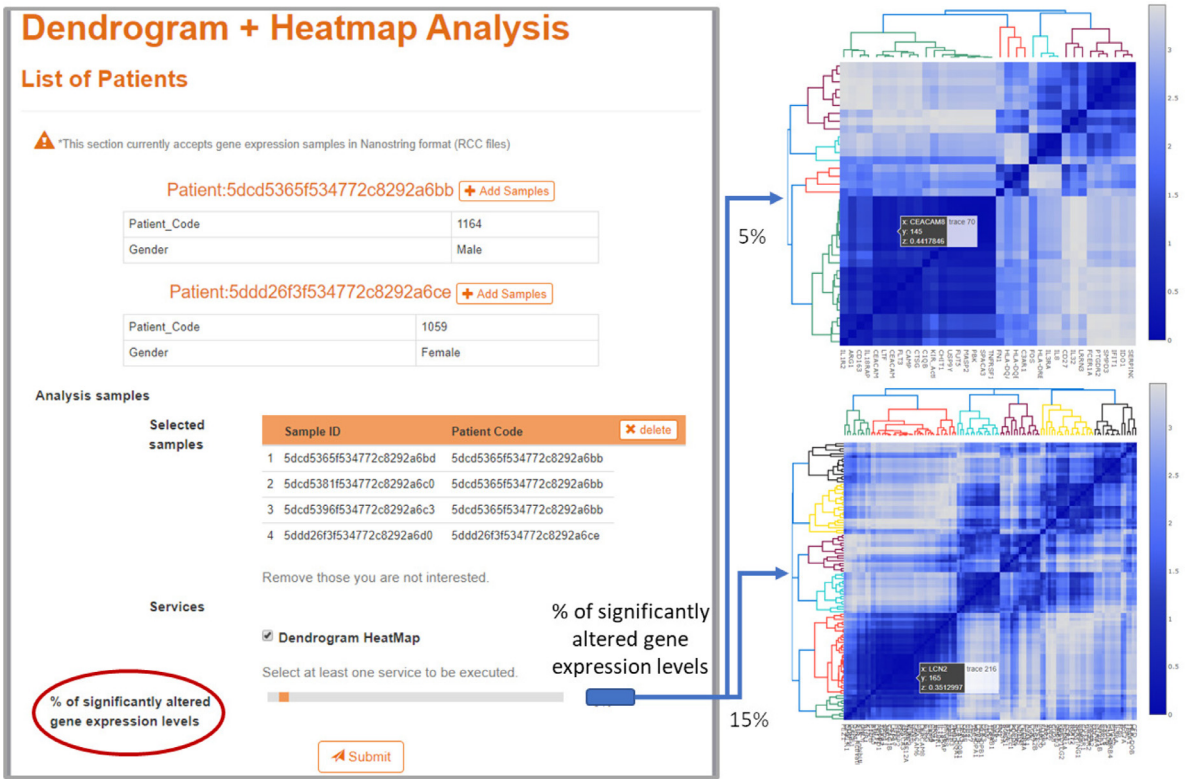

Fig. 4. Selection panel of gene expression files and visualization of resulting Cluster heatmaps according to different percentages of significantly altered gene expression levels. In this example is observed the results for four samples (three from first patient and one from the second one) with two filtering percentages. Thus, the result on the right-up side shows a case with only the 5% of the most representative genes.

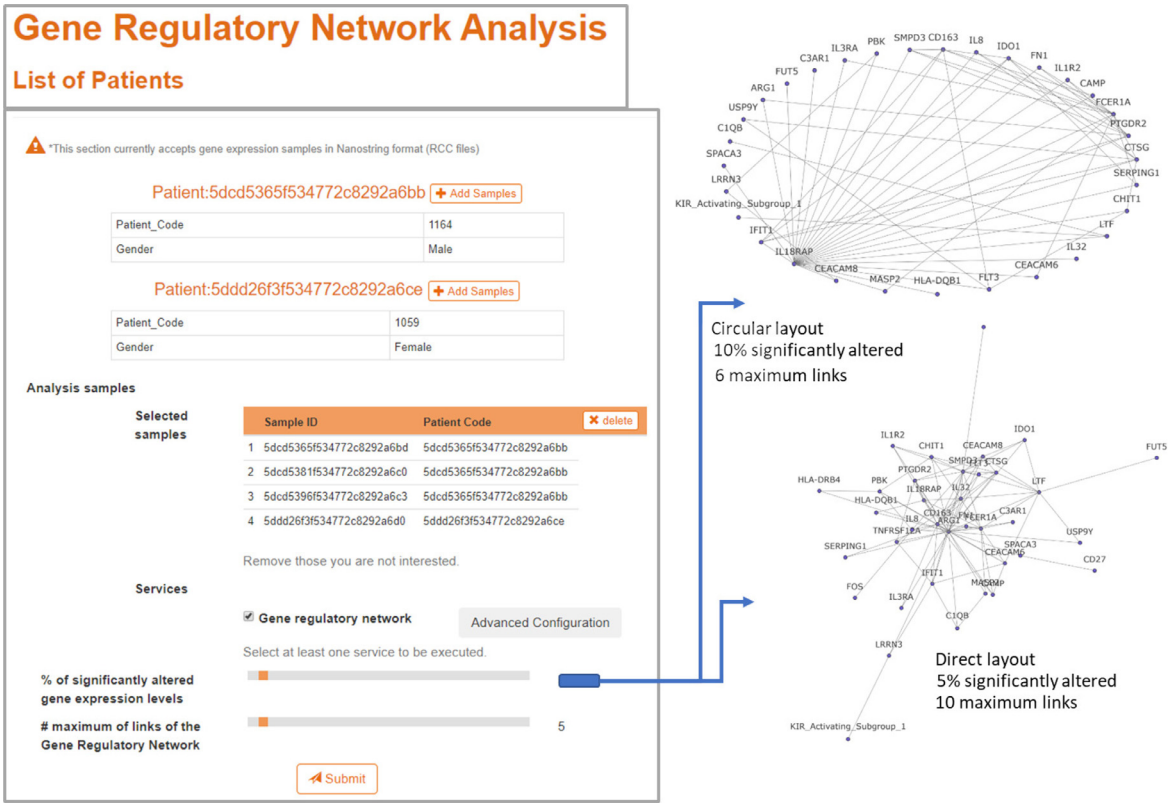

Fig. 5. Selection panel of gene expression files and visualization of resulting gene regulatory networks according to different percentages of of significantly altered gene expression levels.

Table 1
FIMED in comparison.

	OpenClinica [21]	REDCap [23]	TrialDB [20]	Phoenix [24]	Progmatic [22]	Dados-P. [25]	openCDMS [19]	PhOsCo [26]	FIMED
1	✔	✔	✔	✔	✔	✔	✔	✔	✔
2	✔	✔	✔	✔	✔	✔	✔	✔	✔
3	✔	✔	✔	✔	✔	✔	✔	✔	✔
4	✔	✔	✔	✔	✔	✔	✔	✔	✔
5	✔	✔	✔	✔	✔	✔	✔	✔	✔
6	✔	✔	-	✔	✔	-	✔	✔	✔
7	✔	✔	✔	✔	✔	✔	✔	✔	✔
8	✖	✖	-	✔	-	✔	✔	✔	✔
9	✖	✖	-	-	✖	✖	✖	-	✔
10	✔	-	-	-	✔	✔	✖	-	✔
11	✔	✔	-	✔	-	-	✔	-	✖
12	✖	✖	✖	✖	✖	✖	✖	✖	✔
13	✔	✖	✖	✖	✖	✖	-	✖	✔
14	✖	✖	✖	✖	✖	✖	✖	✖	✔
15	✖	✖	✖	✖	✖	✖	✖	✖	✔
16	✖	✖	✖	✖	✖	✖	✖	✖	✔

have used locust tool to perform different data loads and operations, stressing the API of FIMED with a series of queries. The performance of the time response of FIMED has been studied for data insertion, deletion and retrieval. For this propose, We have configured the locust API to simulate up to 5000 users interacting with the FIMED API with a Spawn rate of 50 users (users spawned/second = 50). We have defined the behaviour of the users in the locust API with Python code as follow: each user will make a POST request to create his/her form, then (s)he will fill in that form with his/her patient's information, then (s)he will search for the patient and finally (s)he will delete it.

FIMED performs properly as the number of simultaneous users increases, with an average time response of 74 (ms), 81 (ms), 42 (ms) and 50 (ms) respectively, for each of the request presented before. When we reach approximately 4500 users, the FIMED API starts to perform moderately in some requests and the time response starts to grow speedily. Probably, this could be explained by the fact that FIMED is served through a standard Tomcat 9 Web application service and more simultaneous requests are received than can be handled by the currently available request processing threads.

6. Discussion

Despite the availability of many electronic Case Report Form (eCRF) tools designed to capture clinical trial data, most of them lack a flexible integration of clinical information since the clinicians are not able to design and modify the forms according to their needs. Moreover, we have observed that the majority of these tools required a significant time investment to create CRFs and a thorough study making their use complicated for small-scale investigator.

Before developing FIMED, we have made an exhaustive study of general features in this kind of systems and we outlined several of these features that are essential and are shared in almost all the systems. In order to alleviate some of the limitations encountered in the literature, we present other features in our tool that we have not observed in the systems found in the scientific literature but are crucial for the collection, management and analysis of the clinical information of the study subjects. These features are presented below:

1. Enabling ways to secure patient's information.
2. Ensuring that retrieved data regarding each subject is only attributable to that subject.
3. Creating clinical research forms (eCRF).
4. Providing support for several types of fields (such as dates, text, numerical values) and in various formats/ support for all basic field data types).
5. Supporting Web-based interfaces.
6. Providing software to be hosted locally to protect sensitive data.
7. Being Open source.
8. Providing user-friendly interfaces so that users can create CRFs and enter data directly on the interface.
9. CRFs should be easy to modify once created.
10. Should be able to contain non-traditional fields such clinical images, samples, etc.
11. Exporting to formats such as Excel, Pdf, Xml, Html, and CSV.
12. Dynamically storing the clinical data from multiple clinical trials.
13. Allowing to extend their functionalities.
14. Transferring data to different types of samples to target different analysis.
15. Using a database schema that grants enough adaptability to face the continuous changes in the practice of clinical trials.
16. Integration of analysis tools in order to examine the data to understand a disease.

Table 1 shows a comparison between FIMED and a set of related tools found in the literature, according to the list criteria exposed above. As can be observed, desirable features related to dynamism in integration phase, adaptability, scalability and advanced analytic are covered by FIMED, which represent an advantage with regards to these compared tools.

7. Conclusion

FIMED is a software tool for clinical data collections allowing clinicians without programming skills flexible management of clinical research information. It provides many functionalities in order to facilitate data management by clinicians, such a (I) personalized form design ("do-it-yourself") dynamically adapting to each of the patients entries in the application; (II) browse functionality to store gene expression assays associated to the patient with metadata to grant additional information to the samples; (III) the modification and the update of the data over the time; and (IV) a search tool to provide direct access to the data with different filter options. Additionally, FIMED integrates analysis tools for clinical trials to allow clinicians to perform different types of analysis towards a deeper comprehension of the molecular mechanisms in a particular disease through the interpretation of results. Moreover, FIMED offers some mechanisms to extend the software with new components in order to expand its functionalities.

Current version incorporates algorithms for gene expression data analysis and offers visualization tools for the exploration of these data: Heatmaps, Cluster Heatmaps and Gene Regulatory Networks. FIMED has taken the experience acquired with the development of VIGLA-M [33] in the analysis of gene expression data, and has been tested with use cases conducted with actual sequence data from metastatic Melanoma patients. This previous work provided also relevant needs from the clinical assay data management from the clinician point of view, as clinicians found limitations in improving or extending the data collected during the process. Its usability in this real scenario has been validated, since we have obtained first real clinical insights. In this sense, it has been evident how this tool can be easily integrated into different use cases, making FIMED a powerful clinical research tool for data management, analysis and visualization in the practice of clinical assays in different studied diseases. Apart from the public instance provided, the project can be deployed by IT administrators in any health information system, ensuring higher protection of the clinical data.

As a matter of future work, we plan to update it and ensure future compatibility with more use cases, such a adaptability to more gene expression files formats, other diseases and integration with other analytical tools or algorithms (real time sensor data analysis, clinical image analysis). FIMED is also being used in the melanoma clinical case to find new biomarkers for predicting the patient evolution during the treatment, which results will be relevant for the medical community.

Funding

This work has been partially funded by the Spanish Ministry of Science and Innovation via Grant PID2020-112540RB-C41 (AEI/FEDER, UE) and Andalusian PAIDI program with grant P18-RT-2799.

Declaration of Competing Interest

The authors: Mr. Sandro Hurtado, Dr. José Garcḍa-Nieto, Dr. Ismael Navas-Delgado and Dr. José Aldana-Montes. Declare that there is no conflict of interest regarding the publication of this article with title: *FIMED: Flexible management of biomedical data*

CRediT authorship contribution statement

Sandro Hurtado: Formal analysis, Software, Writing – original draft. **José García-Nieto:** Conceptualization, Methodology, Writing – original draft. **Ismael Navas-Delgado:** Conceptualization, Resources, Writing – original draft. **José F. Aldana-Montes:** Supervision.

Acknowledgements

We would like to thank to Doctor Miguel Ángel Berciano Guerrero (from Hospitales Univesitarios Regional y Virgen de la Victoria de Málaga and Instituto de Investigaciones Biomédicas - IBIMA) for his advice and guidance in user's requirements.

References

[1] J. Xuan, Y. Yu, T. Qing, L. Guo, L. Shi, Next-generation sequencing in the clinic: promises and challenges, Cancer Lett. 340 (2) (2013) 284–295.
[2] E.D. Green, M.S. Guyer, Charting a course for genomic medicine from base pairs to bedside, Nature 470 (7333) (2011) 204–213.
[3] M. Ou, R. Ma, J. Cheung, K. Lo, P. Yee, T. Luo, T. Chan, C.H. Au, A. Kwong, R. Luo, et al., Database. bio: a web application for interpreting human variations, Bioinformatics 31 (24) (2015) 4035–4037.
[4] J.S. Beckmann, D. Lew, Reconciling evidence-based medicine and precision medicine in the era of big data: challenges and opportunities, Genome Med. 8 (134) (2016) 2–11.
[5] S.K. Gill, A.F. Christopher, V. Gupta, P. Bansal, Emerging role of bioinformatics tools and software in evolution of clinical research, Perspect. Clin. Res. 7 (3) (2016) 115.
[6] W.L. Schulz, B.G. Nelson, D.K. Felker, T.J. Durant, R. Torres, Evaluation of relational and NoSQL database architectures to manage genomic annotations, J. Biomed. Inf. 64 (2016) 288–295.
[7] V. Bianchi, A. Ceol, A.G.E. Ogier, S. de Pretis, E. Galeota, K. Kishore, P. Bora, O. Croci, S. Campaner, B. Amati, M.J. Morelli, M. Pelizzola, Integrated systems for NGS data management and analysis: open issues and available solutions, Front. Genet. 7 (2016) 75, doi:10.3389/fgene.2016.00075.
[8] G.M. Weber, K.D. Mandl, I.S. Kohane, Finding the missing link for big biomedical data, Jama 311 (24) (2014) 2479–2480.
[9] W.J. Kent, C.W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, D. Haussler, The human genome browser at UCSC, Genome Res. 12 (6) (2002) 996–1006.
[10] M.M. Astrahan, M.W. Blasgen, D.D. Chamberlin, K.P. Eswaran, J.N. Gray, P.P. Griffiths, W.F. King, R.A. Lorie, P.R. McJones, J.W. Mehl, et al., System R: relational approach to database management, ACM Trans. Database Syst. (TODS) 1 (2) (1976) 97–137.
[11] R. Elmasri, Fundamentals of database systems, 2017.
[12] E. Rahm, P.A. Bernstein, An online bibliography on schema evolution, SIGMOD Rec. 35 (4) (2006) 30–31, doi:10.1145/1228268.1228273.
[13] P. Payne, A.W. Greaves, T.J. Kipps, CRC clinical trials management system (CTMS): an integrated information management solution for collaborative clinical research, AMIA... Annual Symposium proceedings. AMIA Symposium, vol. 2003, American Medical Informatics Association, 2003. 967–967
[14] H. Leroux, S. McBride, S. Gibson, On selecting a clinical trial management system for large scale, multi-centre, multi-modal clinical research study, in: HIC, 2011, pp. 89–95.
[15] R. Li, H. He, R. Wang, S. Ruan, T. He, J. Bao, J. Zhang, L. Hong, Y. Zheng, TrajMesa: A distributed NoSQL-based trajectory data management system, IEEE Trans. Knowl. Data Eng. (2021), doi:10.1109/TKDE.2021.3079880. 1–1
[16] A. Rafique, D. Van Landuyt, W. Joosen, PERSIST: policy-based data management middleware for multi-tenant SaaS leveraging federated cloud storage, J. Grid Comput. (16) (2018) 165–194, doi:10.1007/s10723-018-9434-6.
[17] A. Rafique, D. Van Landuyt, E. Heydari Beni, B. Lagaisse, W. Joosen, CryptDICE: distributed data protection system for secure cloud data storage and computation, Inf. Syst. 96 (2021) 101671, doi:10.1016/j.is.2020.101671.
[18] M. Ahmadian, F. Plochan, Z. Roessler, D.C. Marinescu, SecureNoSQL: an approach for secure search of encrypted NoSQL databases in the public cloud, Int. J. Inf. Manage. 37 (2) (2017) 63–74, doi:10.1016/j.ijinfomgt.2016.11.005.
[19] J. Ainsworth, R. Harper, The PsyGrid experience: using web services in the study of schizophrenia, Int. J. Healthcare Inf.Syst. Inf. (IJHISI) 2 (2) (2007) 1–20.
[20] C. Brandt, A.M. Deshpande, C. Lu, G. Ananth, K. Sun, R. Gadagkar, R. Morse, C. Rodriguez, P.L. Miller, P.M. Nadkarni, TrialDB: a web-based clinical study data management system, in: AMIA Annual Symposium Proceedings, vol. 2003, American Medical Informatics Association, 2003. pp. 794–794
[21] M. Cavelaars, J. Rousseau, C. Parlayan, S. de Ridder, A. Verburg, R. Ross, G.R. Visser, A. Rotte, R. Azevedo, J.-W. Boiten, et al., Openclinica, in: Journal of Clinical Bioinformatics, vol. 5, Springer, 2015, p. S2.
[22] P. Cramon, Å.K. Rasmussen, S.J. Bonnema, J.B. Bjorner, U. Feldt-Rasmussen, M. Groenvold, L. Hegedüs, T. Watt, Development and implementation of PROgmatic: a clinical trial management system for pragmatic multi-centre trials, optimised for electronic data capture and patient-reported outcomes, Clin. Trials 11 (3) (2014) 344–354.
[23] P.A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, J.G. Conde, Research electronic data capture (redcap)a metadata-driven methodology and workflow process for providing translational research informatics support, J. Biomed. Inf. 42 (2) (2009) 377–381.
[24] R. Krenn, Design and development of a web-based clinical trial management system, 2014 Ph.D. thesis. 10.13140/RG.2.1.4306.2723/1
[25] L. Nguyen, A. Shah, M. Harker, H. Martins, M. McCready, A. Menezes, D.O. Jacobs, R. Pietrobon, DADOS-prospective: an open source application for web-based prospective data collection, Source Code Biol. Med. 1 (1) (2006) 7.
[26] D. Venizeleas, M. Linzbach, C. Ohmann, PhOSCo (Pharma Open Source Community): Open Source für klinische Studien, Dtsch Arztebl Int. 101 (19) (2004) 19.
[27] C.A. Brandt, S. Argraves, R. Money, G. Ananth, N.M. Trocky, P.M. Nadkarni, Informatics tools to improve clinical research study implementation, Contemp. Clin. Trials 27 (2) (2006) 112–122.
[28] S. Kaur, I. Singh, et al., Artificial intelligence based clinical data management systems: a review, Inf. Med. Unlocked 9 (2017) 219–229.
[29] J. Shah, D. Rajgor, S. Pradhan, M. McCready, A. Zaveri, R. Pietrobon, Electronic data capture for registries and clinical trials in orthopaedic surgery: open source versus commercial systems, Clin. Orthop. Relat.Research® 468 (10) (2010) 2664–2671.
[30] A. Nourani, H. Ayatollahi, M.S. Dodaran, Clinical trial data management software: a review of the technical features, Rev. Recent Clin. Trials 14 (3) (2019) 1–10, doi:10.2174/1574887114666190207151500.
[31] J. Muller, K. Heiss, R. Oberhoffer, Implementation of an open adoption research data management system for clinical studies, BMC Res. Notes 10 (252) (2017) 1–10, doi:10.1186/s13104-017-2566-0.
[32] K. Chodorow, MongoDB: The Definitive Guide: Powerful and Scalable Data Storage, O'Reilly Media, Inc., 2013, p. 193.

[33] I. Navas-Delgado, J. García-Nieto, E. López-Camacho, M. Rybinski, R. Lavado, M.Á.B. Guerrero, J.F. Aldana-Montes, VIGLA-M: visual gene expression data analytics, BMC Bioinf. 20 (4) (2019) 150.
[34] J. Daemen, V. Rijmen, The Design of Rijndael: AES-The Advanced Encryption Standard, Springer Science & Business Media, Belgium, 2013.
[35] P. Mestdagh, P. Van Vlierberghe, A. De Weer, D. Muth, F. Westermann, F. Speleman, J. Vandesompele, A novel and universal method for microRNA RT-qPCR data normalization, Genome Biol. 10 (6) (2009) R64.
[36] A. Irrthum, L. Wehenkel, P. Geurts, et al., Inferring regulatory networks from expression data using tree-based methods, PloS one 5 (9) (2010) e12776.
[37] T. Moerman, S. Aibar Santos, C. Bravo González-Blas, J. Simm, Y. Moreau, J. Aerts, S. Aerts, GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks, Bioinformatics 35 (12) (2018) 2159–2161.
[38] C.A. Torres-Cabala, J.L. Curry, Genetics of Melanoma, Springer-Verlag, 2016.