German Medical Data Sciences: Shaping Change – Creative Solutions for Innovative Medicine
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI190817

A Visualisation and Extraction Tool for Time Series in the MIMIC III Database

Sven FESTAG[a,1] and Cord SPRECKELSEN[a]
[a] *Department of Medical Informatics, Medical Faculty, RWTH Aachen University*

Abstract. We developed a tool based on the KNIME analytics platform for the extraction and visualisation of medical time series stored in the Medical Information Mart for Intensive Care III (MIMIC III) and the related MIMIC-III Waveform Database Matched Subset. The large number of data points and the free accessibility make these data sets an attractive source for data-driven projects in the medical domain. The problem that we tackled with our tool was the lack of an easy and extensible way of selecting, reading, and visualising stored time series. Especially the fact that medical data science projects are often conducted by interdisciplinary teams called for a software solution that can be utilised by medical practitioners without programming experiences and that still offers enough flexibility for data scientists.

Keywords. Data Science, Interdisciplinary Research, Data Visualization, Software

1. Introduction

1.1. Background

The *Medical Information Mart for Intensive Care* (MIMIC) is a large database (DB) containing data of patients who were treated at intensive care units of the Beth Israel Deaconess Medical Center (Boston, Massachusetts, USA). The full database underwent an automatic de-identification to protect patients' privacy. As full anonymity cannot be guaranteed, researchers who want to use the data have to pass an online course and accept a data use agreement before they are granted access by the curators [1]. However, there is no further restriction on the usage, which makes the database a valuable source for reproducible data-driven studies in the biomedical domain.

The third version MIMIC III comprises clinical measurements, administrative information, laboratory test results, (pharmacological) therapy descriptions, and caregiver notes related to more than 40 000 patients [1]. During the generation process between 2001 and 2012, (electronic) medical records and accounting documents as well as free texts were compiled into this database.

The *MIMIC-III Waveform Database Matched Subset* (WAVEFORM) is an extension to this collection and offers recordings of physiological signals as well as time series of vital signs recorded by bedside monitors. It contains high-resolution sequence data of more than 10 000 patients also represented in the MIMIC DB. The

[1] Corresponding Author, Sven Festag, Department of Medical Informatics, Medical Faculty, RWTH Aachen University, Pauwelsstr. 30, 52074 Aachen, Germany; E-mail: sven.festag@rwth-aachen.de

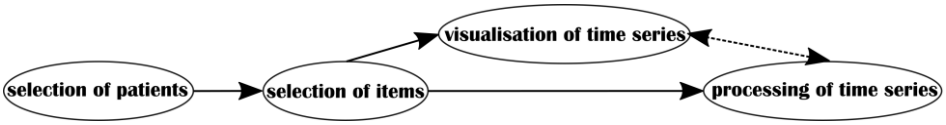

Figure 1. Processing pipeline supported by the new tool.

waveforms were sampled at a frequency of 125 Hz with at least 8-bit resolution, while the time series have a frequency of 1 Hz [2].

Both DBs are accessible as PhysioBank DBs administered by the PhysioNet project [3]. In contrast to MIMIC III, the WAVEFORM database is even publicly available without any formal request. The raw data are stored as comma-separated (MIMIC III) or binary (WAVEFORM) files, respectively. Thus, software tools are needed to enable ergonomic work. Especially if the data are used by interdisciplinary teams or groups with small programming skills, systems for visualisation and extraction of information are needed.

We developed such a system that allows users to easily examine available data and to extract required subsets[2]. A key characteristic of this tool is its extensibility. The intended use is the collaboration between medically trained researchers and data scientists.

1.2. Requirements

We defined the requirements in the context of an interdisciplinary project between a medical researcher and a data scientist. These requirements were repeatedly updated in an agile fashion in order to meet all needs. Moreover, they were generalised with respect to relevant literature presented in section 2.

A key demand on the new tool is that the user can get informative visualisations of data in combination with a representation that allows further processing. By this, an early participation of team members without programming skills is enabled, what in turn accelerates and simplifies the research. Another requirement is that the software is easily extensible even for the medically trained researchers such that representations can be customised without the help of someone who adapts source code. The focus of our project lay on the visualisation and extraction of time series data like the development of a patient's blood pressure over time. Figure 1 represents a usual workflow that the developed tool supports. In a first step, the user selects patients and medically relevant items that he wants to extract from the data source. Afterwards, the retrieved time series are visualised and additionally processed as part of the user's project.

The aim of our project was to develop a tool that offers both, easy screening and extraction of available data (MIMIC III and WAVEFORM) as well as fast adaptability and extensibility of underlying functionality.

[2] The code is available at https://bitbucket.org/svefes/mimic_viewer

2. State of the art

There have been several endeavours to make data from PhysioBank sources more easily accessible to researchers. The web service *PhysioBank Automated Teller Machine* (ATM) offered by PhysioNet is a general tool for the visualisation and extraction of such records [4]. In a first step, the ATM asks users to select one of the publicly available PhysioBank DBs (like the WAVEFORM DB). Afterwards, a record that corresponds to a single patient can be chosen. In the last step, the user specifies a format of the output, i.e. the time interval and the (physiological) signals that are included as well as the kind of presentation. Then the ATM either provides a file to download or it computes and displays a visual representation (histogram, raw waveform etc.) of the data. The first problem of this method is that it only offers the extraction of data from a single patient at a time.

PhysioBank ATM enables interactive visualisation of patient data. By using buttons, the user can navigate from one visualised interval to the next. The shortcoming of this approach is that every time the navigation buttons are used, a new query has to be processed and a new PNG image of the requested record and format is computed before it is displayed. This static behaviour hampers the exploration of unseen data. Moreover, the DBs with restricted access (like MIMIC III) cannot be accessed with the help of the ATM. However, this online service can be used without any prior knowledge and to get a quick overview of available PhysioBank records and data.

A similar tool is LightWAVE. It offers nearly the same functionality as the previously described ATM software. An advantage of LightWAVE is that it also handles the editing of annotations stored in combination with the waveforms [5]. Still, only single records can be viewed. Moreover, an experiment with the Chrome browser revealed that different waveforms of a patient begin to overlap if the user zooms in too much. That means, the scale of abscissa and ordinate cannot be changed independently.

Lee et al. developed a web-based tool for the visualisation of data from the MIMIC II DB [6]. Their aim was to allow researchers fast and easy data exploration of patient cohorts without the need for downloading data and learning SQL. The tool is specialised at visualising distributions of features in a selected patient cohort and thought of as a starting point for larger research studies [6].

Many other tools are aimed at users who are familiar with programming and/or query languages. A large collection of such tools is contained in the WaveForm DataBase (WFDB) software package [7]. It is written in C and available for many operating systems. It allows to process, view, automatically analyse, and even create PhysioBank records. There are also wrappers or re-implementations of WFDB for the use with Java, Python and Matlab [7]. Again, these packages cannot be used to access the MIMIC III DB directly due to the previously mentioned access restrictions.

However, there are lots of software solutions for the use with MIMIC III that originate from projects conducted by individual MIMIC users who contributed their code to a centrally curated repository [8]. All these projects aim at different outcomes. There are, for example, scripts that enable the user to create a local copy of the MIMIC DB in a database management system of his choice. Other scripts help to summarise or extract high-level information that is implicitly stored in MIMIC III.

A recent publication by Unberath et al. deals with the visualisation of time series data, extracted from a patient data management system [9]. Their tool is integrated into a clinical decision support platform and can only be used in combination with such

software. Although the service allows easy customisability for practitioners, it does not support extensions by researchers without programming skills.

To summarise, there are many approaches for the extraction or visualisation of data stored in medical databases, but they all aim at different target groups and fields of applications. None of the available tools met all the requirements we identified for software that supports interdisciplinary data-driven research based on the MIMIC III and its extension WAVEFORM.

3. Concept

To be able to access MIMIC III, a local copy of this DB was set up in MySQL on a server accessible by all team members. Only members that were granted access to these data by the PhysioNet team, were also given the rights to read on the local server. At this point in time, these researchers would have been able to access raw data of MIMIC III via SQL queries and the web-based WAVEFORM DB via the WFDB package. For users unfamiliar with reading in relational databases and with writing code, this basic setup was not sufficient. Moreover, the fact that both DBs had to be accessed with different tools rendered this preliminary system useless for all users.

The basis for our implementation is the KNIME Analytics Platform. It offers modular and graphic implementation of data processing pipelines. The core modules of KNIME for I/O, manipulation, analysis and visualisation of data are capsuled by nodes [10]. Every node handles a single task and its output(s) can be used as an input for following nodes. Data are passed as so-called DataTables containing one row per instance and one column per feature. In total, nodes and their connections define a pipeline. During implementation, the user places boxes representing these nodes on a graphic workspace. Additionally, lines are drawn between boxes to define the data flow. Every node has a dialogue window where parameters can be specified. After the pipeline has been defined, the user can execute every node individually or several nodes at once. However, the execution of nodes must be postponed until all predecessors have computed their outputs. The graphic implementation implicitly documents its own behaviour and allows KNIME users without any programming experiences to quickly understand the data processing pipeline and even to adapt it according to their needs. Besides this characteristic, the fact that the software is an open source project, its scalability, and its platform independence led to our decision for KNIME.

Many basic building blocks for data processing and even more advanced routines like the training of multilayer perceptrons or the execution of hypothesis tests are available in KNIME. Many optional extensions complement the set of nodes. To enable as much flexibility as possible for users, KNIME also offers scripting nodes that can execute fully customisable Python or Java code. Moreover, sub-pipelines consisting of several nodes can be summarised into meta-nodes. This transparent representation keeps the graphic workflow clear and concise while preserving modularity. For an extensive survey of available tools for life science research in KNIME see [11] by Fillbrunn et al.

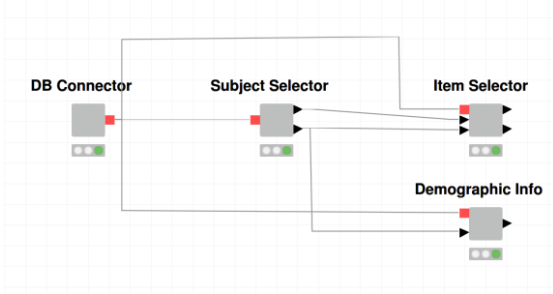

Figure 2. Graphic Workflow as presented in KNIME.

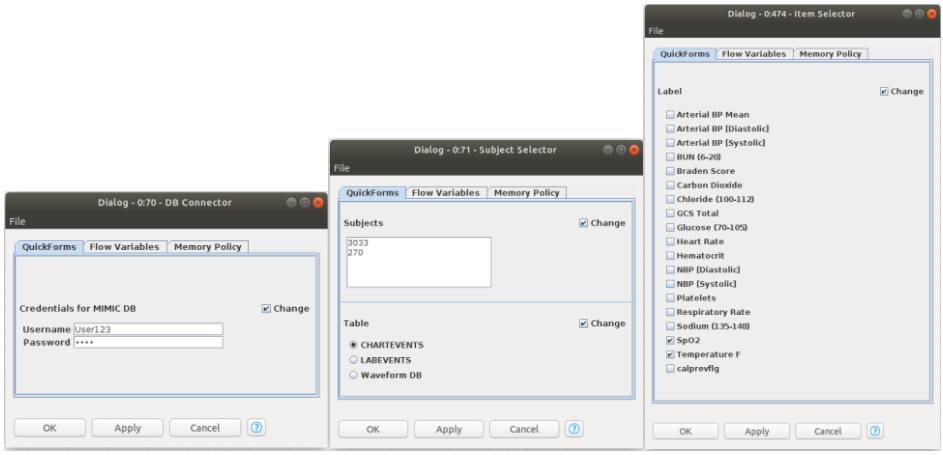

Figure 3. Dialogue windows of the DB Connector, Subject Selector and Item Selector.

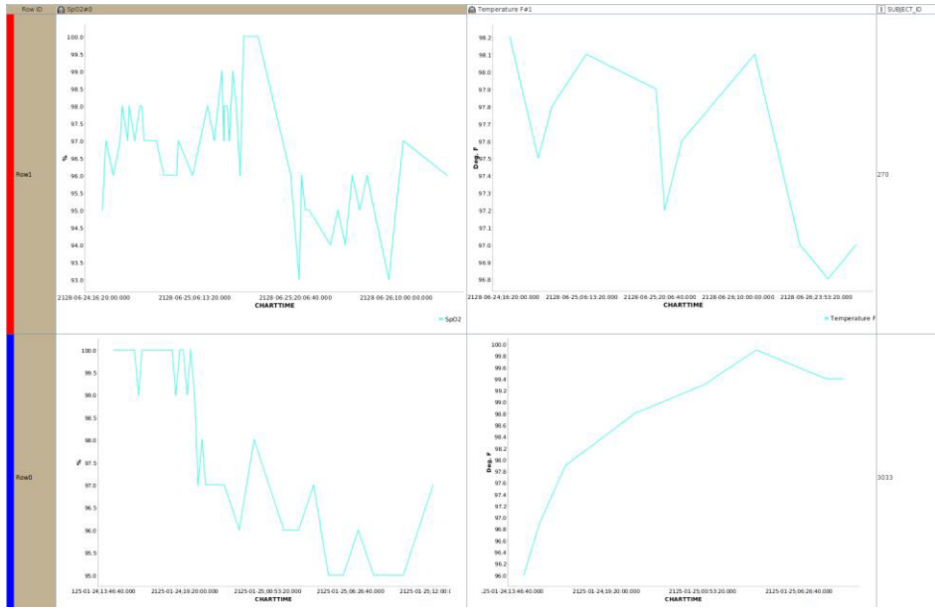

Figure 4. First output of the Item Selector: Line plots of time series.

4. Implementation

Our workflow consists of four main components that are implemented as meta-nodes. The graphic representation of them in KNIME is depicted in figure 2. Except for the *DB Connector*, all nodes depend on the output of their left neighbours. Thus, they also must be executed in this ordering. Figure 3 shows the configuration windows of the upper three nodes during a sample run.

The first node (*DB Connector*) handles the access to our local copy of MIMIC III. The corresponding dialogue asks the user to enter his credentials. The established connection is needed for all following DB accesses.

The *Subject Selector* is the first node that is specialised for our purposes. Its dialogue asks the user to enter one or more Subject IDs that uniquely describe patients whose treatments are documented in MIMIC III or the WAVEFORM. Moreover, 3 radio buttons enable the choice between data sources. The first two options correspond to two tables of the MIMIC III DB, namely CHARTEVENTS and LABEVENTS. The CHARTEVENTS table contains all data that were collected from electronic patient charts like physiological measurements and caregiver observations. LABEVENTS comprises laboratory measurements for patients [1]. The two tables partially overlap with the result that certain events are documented in both. Since our tool was developed for the extraction and visualisation of original MIMIC data, this overlap was not dealt with. The user must decide which data source should be used as the basis for his query and how to handle the redundancy in these sets. If access to further tables is needed in subsequent projects, they can be added as well. The third and last option available as a radio button is the WAVEFORM DB. The exclusive choice for one source ensures acceptable runtimes and enables a clear representation of (intermediate) results. When the node is executed, either a database query is sent to the local MIMIC server or a request is submitted to the WAVEFORM server, depending on the activated radio button. In the first case, the *Database Reader* node contained in KNIME Core is used, while the latter request is handled via functionalities of the WFDB Python package wrapped in the *Python Script* node taken from the KNIME Python Integration. If CHARTEVENTS or LABEVENTS are chosen, the internal workflow of *Subject Selector* determines all items that are stored in the corresponding table for all selected subjects. Items are used by MIMIC to uniquely describe concepts like heart rate and blood pressure. The more subjects are specified the fewer items are found that are monitored for all patients. Similarly, if WAVEFORM is selected, all signals that are available in the corresponding numeric records of all subjects are selected and made available at the first output port. The second port just forwards the selected subjects.

The next meta-node is called *Item Selector* and directly depends on the outputs of the *Subject Selector*. In the configuration dialogue the new node displays all items or signals found by the *Subject Selector* in combination with checkboxes. The user now decides which items or signals are of interest and ticks the checkboxes accordingly. On execution, the *Item Selector* works similarly to the *Subject Selector* during the first steps. In contrast to the previous node, however, it does not extract meta information but fetches all CHARTEVENTS, LABEVENTS, or WAVEFORM records that correspond to the selected subjects and the chosen items/signals. The second part of the internal pipeline handles the visualisation and representation of the results. For every pair of selected subject and signal, one line-plot is generated with the help of *Line Plot (JavaScript)*. This node plots signal values against corresponding time points of the measurements (CHARTTIME). The generated graphs are stored as vector graphics in a

DataTable that is handed to the first output port of the Item Selector. An example table corresponding to the sample run defined in figure 3 is presented in figure 4. The *Line Plot (JavaScript)* node also offers an interactive view that allows zooming and simplified reading of single values. However, there is no possibility of displaying a varying number of plots in an interactive view. Hence, we decided to output a DataTable that can handle arbitrarily many rows (subjects) and columns (signals) containing only the static representations of the line plots as depicted in figure 4. To integrate the convenience of interactive plots, a python script executed within another *Python Script* node was added to the workflow. This script mainly uses the interactive visualisation library *Bokeh* that offers a variety of data representations displayed in a web browser. On execution, the node generates line plots and displays them in a new browser tab. The order of plots is the same as in the previously described DataTable. Every row corresponds to a subject, while the columns refer to the chosen signals. In contrast to the static table, the depictions in the browser are interactive. Especially for long time series this is advantageous, as regions of interest can be enlarged and isolated.

The second output of the Item Selector is a DataTable containing all raw measurements and their timestamps. The output of raw data is needed to allow extensions and to ensure a shared interface for both, researchers implementing data-driven approaches and medical researchers advising on them.

The last component of the workflow is called *Demographic Info*. It sends queries to the MIMIC DB to derive age, gender, and admission time for every selected subject.

The usage of the workflow presented in figure 2 can be summarised as follows. At first, the user has to pass his credentials to the *DB Connector* (figure 3, left). Afterwards, the subjects of interest are specified in the dialogue window of the *Subject Selector* (figure 3, middle). Moreover, one of the data sources CHARTEVENTS, LABEVENTS, and WAVEFORM has to be chosen. After *Subject Selector* has finished its processing, the *Item Selector* can be configured (figure 3, right). For this purpose, the user must choose all signals or measurements that are plotted afterwards.

The *Demographic Info* node does not need user input but only the outputs of *DB Connector* and *Subject Selector* in order to extract relevant demographic information.

5. Discussion and Outlook

The key features of the presented workflow are a self-explanatory, graphic user interface and the possibility for project members with small or no programming skills to engage actively without the help of researchers trained in computer science. Furthermore, the software is scalable and allows direct extensions to generate elaborate data processing pipelines for collaborative projects.

In contrast to the PhysioBank ATM, our tool allows to extract and visualise data of several patients at once, what simplifies comparisons. Moreover, the dynamic display of graphs in the web browser is easier to handle than the comparable function offered by the ATM. The visualisations used in the “MIMIC II Visualization” tool by Lee et al. like pie charts, boxplots, and histograms are geared to cohort data. They do not contain any graphs representing time series data of single patients.

The current version of our tool requires a local installation of the KNIME platform on every participant’s machine. Moreover, possible extensions must be distributed manually between all collaborators. To overcome these difficulties, we will extend the tool with KNIME Server that allows offering the pipeline as a web service.

Due to the sensitive data stored in MIMIC III and the consequent protection regulations, the tool requires a local copy of the DB to which it must be connected before the first usage. Moreover, only team members who are granted access to MIMIC III by the PhysioNet curators can be given access to our full KNIME pipeline. Another weakness of our tool is that the web queries sent to the WAVEFORM DB sometimes lead to long runtimes of the workflow. For a future version we plan to store this data set on the local server as well to improve processing times. Moreover, an enhanced requirement analysis with the help of additional use cases developed in cooperation with current users is in progress.

6. Conclusion

We developed an extendable KNIME workflow for the visualisation and extraction of time series to simplify interdisciplinary, data-driven, medical projects. The data sources are MIMIC III and the corresponding matched WAVEFORM DB.

Our tool is a good starting point for larger projects, as it circumvents redundant setups of the MIMIC database and offers a shared basis for all participants.

Conflict of Interest

The authors state that they have no conflict of interests.

References

[1] A.E. Johnson, T.J. Pollard, L. Shen, L.-w.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, *Scientific Data* **3** (2016), 160035. doi:10.1038/sdata.2016.35.

[2] B. Moody, M. Craig, A. Johnson, T. Kyaw, G. Moody, M. Saeed, M. Villarroel, *The MIMIC-III Waveform Database Matched Subset*, physionet.org, 2017. doi:10.13026/C2294B

[3] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet, *Circulation* **101** (2000), e215-e220. doi:10.1161/01.CIR.101.23.e215.

[4] PhysioNet, *PhysioBank ATM* [internet]. Available from: https://www.physionet.org/cgi-bin/atm/ATM.

[5] G.B. Moody, LightWAVE: Waveform and Annotation Viewing and Editing in a Web Browser, *Comput. Cardiol.* **40** (2013), 17–20.

[6] J. Lee, E. Ribey, J.R. Wallace, A web-based data visualization tool for the MIMIC-II database, *BMC medical informatics and decision making* **16** (2016). doi:10.1186/s12911-016-0256-9.

[7] G.B. Moody, *WFDB Programmer's Guide*. 10th ed., 2018.

[8] A. Johnson, T. Pollard, J. Blundell, erinhong, P. Nicolas, A. Zhang, et al., MIT-LCP/mimic-code: MIMIC Code v1.4.1, 2018. doi:10.5281/zenodo.1411319.

[9] P. Unberath, H.-U. Prokosch, S. Kraus, A REST Service for the Visualization of Clinical Time Series Data in the Context of Clinical Decision Support, *Stud. Health Technol. Inform.* **258** (2019), 26–30. doi:10.3233/978-1-61499-959-1-26.

[10] M.R. Berthold, N. Cebron, F. Dill, T.R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel, KNIME - the Konstanz Information Miner: Version 2.0 and Beyond, *SIGKDD Explor. Newsl.* **11** (2009), 26–31. doi:10.1145/1656274.1656280.

[11] A. Fillbrunn, C. Dietz, J. Pfeuffer, R. Rahn, G.A. Landrum, M.R. Berthold, KNIME for reproducible cross-domain analysis of life science data, *Journal of Biotechnology* **261** (2017), 149–156. doi:10.1016/j.jbiotec.2017.07.028.