R. Röhrig et al. (Eds.)

doi:10.3233/SHTI190818

MedEx – Data Analytics for Medical Domain Experts in Real-Time

Aljoscha KINDERMANN[a,b,1,*], Ekaterina STEPANOVA[c,*], Hauke HUND[b], Nicolas GEIS[b], Brandon MALONE[d] and Christoph DIETERICH[a,b,1]
[a] *Klaus Tschira Institute for Integrative Computational Cardiology*
[b] *Department of Internal Medicine III, University Hospital Heidelberg*
[c] *Center for Bioinformatics, Saarland University*
[d] *NEC Laboratories Europe*

Abstract. Translational research in the medical sector is dependent on clear communication between all participants. Visualization helps to represent data from different sources in a comprehensible way across disciplines. Existing tools for clinical data management are usually monolithic and technically challenging to set up, others require a transformation into specific data models while providing mostly non-interactive visualizations or being specialized to very particular use cases. Statistical programming languages (R, Julia) on the other hand offer great flexibility in data analytics, but are harder to access for clinicians with little to no programming expertise. Our software, the Medical Data Explorer (MedEx), aims to fill this gap as light-weight, intuitive, web-based solution with simple data import routes. We couple a modern dynamic web interface with an in-memory database solution for near real-time responsiveness. MedEx provides multiple visualization options (Scatterplot, correlation heatmap, bar chart, grouped boxplot, grouped histogram, coplot) to get an easy overview on the loaded data as well as to perform pattern discovery and elementary statistics. We demonstrate the utility of MedEx, by example, on data from the cardiology research warehouse of Heidelberg University Hospital. In summary, our tool empowers clinicians to conduct their own interactive exploratory data analysis.

Keywords. Visualization, Data Analysis, Data Exploration, Clinical Data, HiGHmed, Graphical User Interface

1. Introduction

1.1. Background

Clinical care workflows are a source of valuable data with the potential of secondary use for medical research. Sharing the data between different areas of care and medical research remains difficult. The source systems in hospitals are often distributed and heterogenous, complicating data integration. In addition to technical barriers, patient data require special care with regard to data security and can often not leave enclosed

[1] Corresponding authors: Christoph Dieterich, Aljoscha Kindermann, Analysezentrum III, Im Neuenheimer Feld 669, 69120 Heidelberg, Phone: +49 6221 56 36884, Email: christoph.dieterich@uni-heidelberg.de, aljoscha.kindermann@med.uni-heidelberg.de
[*] A. Kindermann and E. Stepanova contributed equally to this work

systems [1][2]. Finding solutions for improved data integration in the medical sector is the goal of the medical informatics initiative funded by the German Federal Ministry for Education and Research (BMBF) [3]. We are part of HiGHmed, which is one of the research consortia funded by the initiative. HiGHmed aims to enhance care and research across institutional boundaries by leveraging distributed data sources of all participating institutions [4][5].

1.2. Requirements

In this context of data collection and integration, we would like to take the medical domain experts on board. An extensive accumulation of medical data alone does not necessarily lead to improved translational research. We are certain that visualization of integrated clinical data supports the interaction across scientific disciplines. Clinical domain experts should be able work with their data and share their ideas. While a lot of clinicians are data-conscious and familiar with statistics, they mostly lack the time to get into statistical programming languages and prefer intuitive graphical user interfaces. This demand is met with modern web technology enabling expressive dynamic visualization, which is a key approach to generate new hypotheses [1][6]. Our goal is to empower clinicians to directly explore and analyze data on their own.

2. State of the art

In current medical data analysis, Microsoft Excel is and remains a popular tool to easily create data visualization [7]. However, this becomes inefficient as soon as the data set exceeds more than a million records. Examples for tools designed specifically to handle larger amounts of clinical data are tranSMART and i2b2 [8][9]. The disadvantage of these monolithic solutions is, however, that they require some technical expertise to install. Furthermore, interactive or real time visualization is not a native feature of these tools but requires further customization and installation of external plugins [10]. Other available tools are mostly specialized in terms of the presented output or the required input or are only commercially available such as the dashboard tools Tableau or Spotfire[11–15].

On the other hand, there are several statistics and visualization friendly programming languages such as R, Python or Julia, in particular if extended with libraries such as Plotly or Shiny. These tools allow a customizable visualization of all kinds of data [16–21]. They are, however, only usable by data scientists and statisticians with sufficient programming skills.

2.1. Objectives

We seek to fill this gap by a tool that empowers clinical domain experts to carry out their own real time visualization analyses on large amounts of patient data, while being easy to deploy, intuitive to use and provides simple ways for data import. In comparison to most established solutions, we wanted to provide a software that is interactive, requires minimal installation effort and can be used as a web application in the clinical network. In summary, we present a software prototype that tries to meet all of the aforementioned demands.

3. Concept

3.1. Software Architecture

The presented tool, the Medical Data Explorer "MedEx" is a web application programmed using the Python library Flask [22]. User input via the web interface triggers rapid data retrieval from an in-memory database. Subsequently statistical evaluations are performed with the help of Python libraries, mainly NumPy, SciPy and Scikit-learn [19][23,24]. The data aggregations are presented back to the user using JavaScript libraries Highcharts and D3 to allow dynamic and interactive charts instead of using static charts from native Python libraries [18,25,26].

We use a Redis in-memory database to store the clinical data with the goal to provide faster querying than with a classic SQL database [27]. The database comprises both numeric and categorical values. Redis databases represent the data in key-value relations. Values can be single values or sets. We set up the data field name as key, while the value is always represented in a set. Unsorted sets are used to store categorical values. To store numeric values, we employed the Redis functionality of assigning a numeric score to each item of the set. To improve the querying performance, we shifted much of the application logic such as filtering processes to the database back-end rather than subsetting in the actual statistical evaluation process.

3.2. Data Preparation

The data we use was collected from various sources of patient care within the Heidelberg University Hospital and is consolidated in a research data warehouse (RWH) [28]. Data sources comprise medical imaging results, electrophysiology results, cardiac catheterization outcomes, anamnesis data, data from quality assurance measures, laboratory results and medical diagnoses in text form. Patient names are not contained in the data; instead each patient is identified by a unique patient identifier. The data was extracted as a csv file, in which each row contains the patient identifier, datetime, parameter name and the measured value for this parameter [29].

The raw data are largely entered manually by the clinical staff. This leads to transfer errors, preventing a direct use of the RWH content in further automated steps. Therefore, the data we received from the RWH was subject to several cleaning steps. Numeric values were cleaned for incorrect decimal identifiers and obvious typos using regular expressions. Categorical fields additionally often contained various expressions identifying the same category. These fields were manually reviewed and set to one consensus category. Diagnoses were extracted from the diagnosis text fields using regular expressions for a few most important cardiology related diagnoses.

To cover temporal effects, in addition to the "one patient, one value per parameter" approach, an additional dataset was created: here, each datapoint was identified by a combination of patient identifier and the month and year of when the data was recorded. The other preprocessing steps remained the same. The data was subsequently loaded into the Redis database running on a regular desktop computer with 16 GB of RAM and a dual-core CPU of 2.30 GHz. As MedEx is supposed to run in a clinical network, the demands of the local machine are even lower, as data storage and calculations will be processed server side. Loading 22.6 Mio datapoints of 212,679 patients into the database took around 2.5 hours.

4. Implementation

The tool we developed is intended to be used by clinical domain experts who have a good knowledge about the statistical meaning of the displayed parameters, but not about data transformation and statistical programming. To make it easy for our target group, our interface is easy to navigate, does not require coding and is intuitively interactive. The average response time depends on the used visualization method but in particular on the number of patients that have a record in the database for the requested data and ranges from less than a second to up to 15 seconds.

The user can navigate between different tabs that offer various analysis and visualization methods suitable for different use cases to break down the data. These methods were selected together with cardiologists of the Heidelberg University Clinic who are regularly working on quantitative patient studies [30].

4.1. Basic Statistics & Plotting

At the **basic statistics** tab, the user can choose from a variety of the most common statistical measures to summarize the numeric variables (mean, median, standard deviation, etc.); to investigate the relationship between the values the user can choose visualization techniques from the **plotting** tab. A heatmap serves to visualize the correlation between multiple variables. To go into more detail, the user can choose to create a scatter plot of two variables. Here it is also possible to zoom in by dragging out a rectangle in the chart or inspect the measured values of each patient by mouseover. Additionally, a linear regression line is drawn, together with the regression formula to facilitate grasping a potential correlation of the variables.
For the categorical variables, the count of patients with a measured value can be displayed in the **basic statistics** tab. Changing to the **plotting** tab, the user can inspect the distribution of the categorical values of each selected field. Both, the Basic Statistics and the Plotting tab allow an export to csv or an Excel file [25,31].

4.2. Boxplot & Histogram

An important aspect of clinical research is the consideration of disease-relevant parameters grouped by patient groups. The **boxplot** function allows the user to select a numeric parameter that is displayed grouped by the values of a categorical parameter chosen by the user. As shown in Figure 1, a classic boxplot is drawn, with median, minimum, maximum and quartiles being displayed.

The **histogram** function allows the same grouped representation of a numeric parameter as the boxplot does, however using a different visualization style. Through the display of the numeric values sorted in bins, the user can get a better understanding of underlying distributions (Figure 2). Both, the boxplot and the histogram allow an interaction with the plotted data. A zoom function allows to inspect areas by drawing a rectangle around them. It is also possible to hide groups by clicking on their name in the legend and displaying more details by mouseover on the respective plot area.

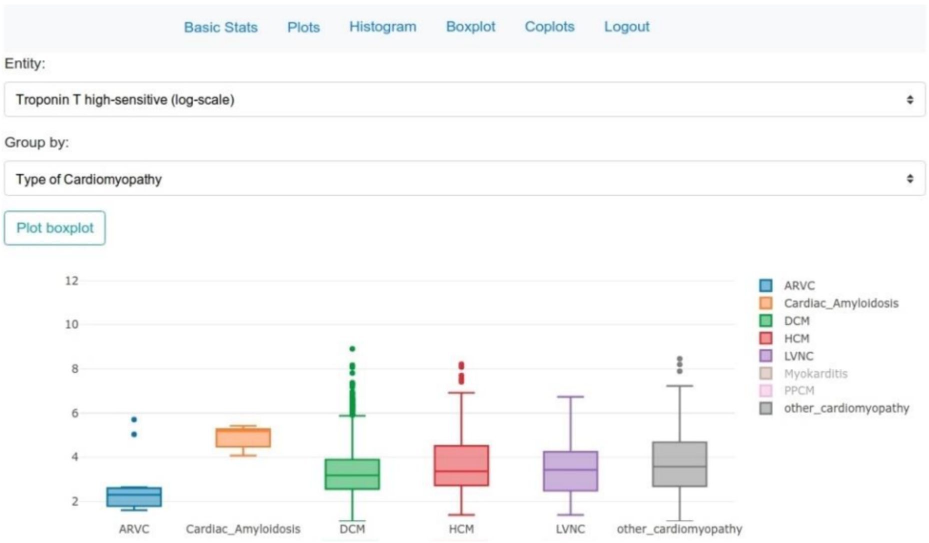

Figure 1. Boxplot of Troponin T levels stratified by type of cardiomyopathy. In this example, we see that the Cardiac Amyloidosis group seems to have generally higher Troponin values. PPCM and Myocarditis were hidden by clicking in the legend. Generating the plot took about five seconds on a desktop computer (22.6 Mio entries in the database).

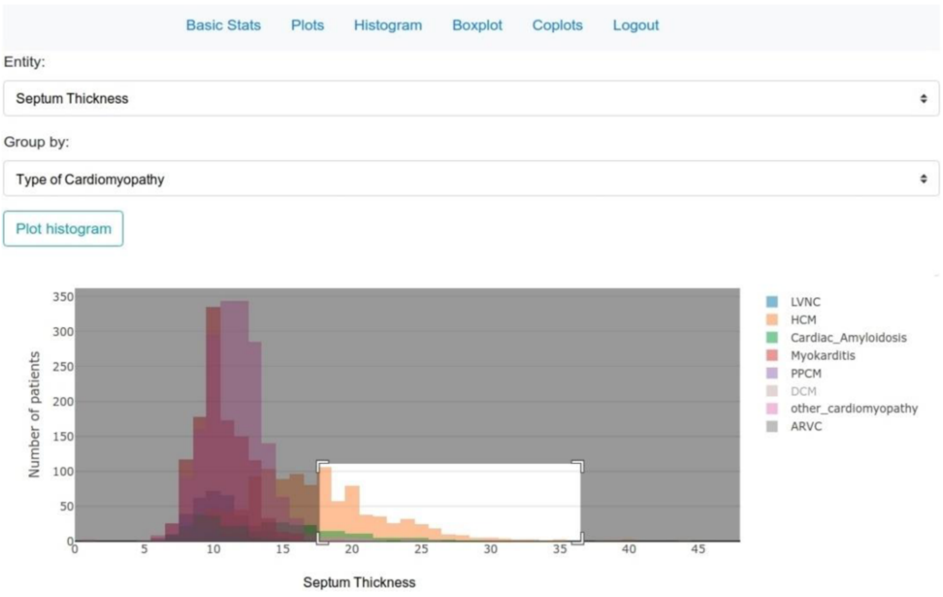

Figure 2. Histogram of septum thickness stratified by Type of cardiomyopathy. In this example, we observe a higher septum thickness occurring in specific forms of cardiomyopathy. Generating the plot took about three seconds on a desktop computer (22.6 Mio entries in the database).

4.3. Coplot Tab

As a solution to show the relationships between two numeric parameters and up to two categorical parameters, R provides the coplot function [16]. This function creates scatter plots grouped by multiple categories, which our tool emulates by multiple scatter plots grouped by two categorical variables.

5. Lessons learned (Discussion)

We developed MedEx to empower clinicians to analyze their data in real-time without having to care about programming or time-consuming data transformations.

5.1. Exploratory Data Analysis Functionality

MedEx is particularly intended for exploratory data analysis tasks. Its basic statistics features and the basic plotting functionalities quickly provide an overview on the imported data. This is particular helpful to detect outliers, imbalanced data or systematic errors early in the analysis.

An even more important feature is MedEx's ability to detect correlations between numeric variables or numeric and categorical variables. In contrast to the all-purpose tool Excel, MedEx is capable of handling millions of datapoints and has per-group boxplot and histogram functionality available on a few clicks. When showing live sessions and video presentations of the tool to clinicians, we received a positive feedback on the intuitive interface; the selected visualizations were regarded as relevant. In contrast to monolithic solutions such as tranSMART the analyses are interactive by default. An additional advantage comes to bear here in contrast to more specialized tools: MedEx does not require the data to be of a specific domain or to be formatted in a complicated common data model.

5.2. Query Time

The Redis database provides generally quick querying times due to its in-memory implementation. Still we experienced long querying times in early versions of our software. We found out, that this was due to filtering steps that were carried out during the statistical analysis in Python instead of using the Redis queries themselves to filter the data. After changing that we could improve the querying times from several minutes to a few seconds. In future versions we plan to extend this optimization process to be able to handle even larger datasets.

5.3. Temporal Analyses

In clinical research, development of patient status over time plays an important role. Comparing disease signs and symptoms of patients between baseline and one or more follow-up visits is a fundamental principle of clinical studies. In our first approach, we covered the time aspect by grouping the data of each patient according to monthly time windows. One of our high priority development tasks is to add actual 1:N time series representation to MedEx. This is particularly important in the HiGHmed project, where heart failure patients will be monitored longitudinally using wearables and apps. Likewise, cardiologists, to whom MedEx was shown, were requesting the addition of a time series representation as one of the most important functionalities.

5.4. Outlier Detection and Filtering

A big pitfall in data analysis lies in data quality. MedEx's ability to give an overview on the range of the data, especially with the help of scatter- or boxplots can help to identify "erroneous" data points. Here again, the ability to interactively select plot areas of interest stands out against non-interactive visualization tools. In the future, we plan to extend the filtering capabilities so that for example users can identify outliers in one analysis and keep them excluded when proceeding to other visualization styles. Deciding which data is interesting for the user will remain their choice, we do not intend to filter out data in advance. However, general preprocessing and loading steps that were run on scripts for the first prototype, will be automated in future versions. Generally, we expect a stronger focus on data quality in medical data science of the future through projects like HiGHmed which might reduce the need of extensive data cleaning.

5.5. Software Deployment

Our goal in the longer run is to make this tool available to clinicians within the Heidelberg University Hospital as an easy interface to the research data warehouse. After evaluating feedback from the respective domain experts, we plan on providing the software to the public as an open source tool for easy data analysis. We intend to provide the packaged Python software directly as well as via container-based software distribution systems.

6. Conclusion

To extract most value from clinical data, it is important to empower respective the domain experts to do data analysis. Tools that are intuitive to use, such as MedEx, can help to achieve this goal. It is important to consider that these tools must find ways to also handle an ever-rising amount of data as well as having solutions to handle data of heterogenous sources and with varying quality levels.

Conflict of Interest

The authors state that they have no conflict of interests.

Acknowledgements

We want to thank Dr. Thiago Britto Borges for his support on Flask.

References

[1] W. Raghupathi et al., Big data analytics in healthcare: promise and potential., *Heal. Inf. Sci. Syst.* **2** (2014) 3. doi:10.1186/2047-2501-2-3.

[2] P. Knaup et al., Implementation of a National Framework to Promote Health Data Sharing, *Yearb. Med. Inform.* **27** (2018) 302–304. doi:10.1055/s-0038-1641210.

[3] Shared data, shared benefits | Medical Informatics Initiative, (n.d.). https://www.medizininformatik-initiative.de/en/start (accessed March 6, 2019).

[4] B. Haarbrandt et al., HiGHmed – An Open Platform Approach to Enhance Care and Research across Institutional Boundaries, *Methods Inf. Med.* **57** (2018) e66–e81. doi:10.3414/ME18-02-0002.

[5] Home | HiGHmed, (n.d.). http://www.highmed.org/ (accessed March 6, 2019).

[6] J. Heer et al., A tour through the visualization zoo, *Commun. ACM.* **53** (2010) 59. doi:10.1145/1743546.1743567.

[7] A.C. Elliott et al., Preparing data for analysis using microsoft Excel., *J. Investig. Med.* **54** (2006) 334–41. doi:10.2310/6650.2006.05038.

[8] B.D. Athey et al., tranSMART: An Open Source and Community-Driven Informatics and Data Sharing Platform for Clinical and Translational Research., *AMIA Jt. Summits Transl. Sci. Proceedings. AMIA Jt. Summits Transl. Sci.* **2013** (2013) 6–8. http://www.ncbi.nlm.nih.gov/pubmed/24303286 (accessed March 14, 2019).

[9] S.N. Murphy et al., Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2), *J. Am. Med. Informatics Assoc.* **17** (2010) 124–130. doi:10.1136/jamia.2009.000893.

[10] S. Herzinger et al., SmartR: an open-source platform for interactive visual analytics for translational research data, *Bioinformatics*. **33** (2017) 2229–2231. doi:10.1093/bioinformatics/btx137.

[11] B.K. Kuntal et al., Igloo-Plot: A tool for visualization of multidimensional datasets, *Genomics*. **103** (2014) 11–20. doi:10.1016/j.ygeno.2014.01.004.

[12] J. Gao et al., Abstract 4271: The cBioPortal for Cancer Genomics as a clinical decision support tool, *Cancer Res.* **74** (2014) 4271–4271. doi:10.1158/1538-7445.AM2014-4271.

[13] H. Ding et al., iGPSe: A visual analytic system for integrative genomic based cancer patient stratification, *BMC Bioinformatics*. **15** (2014) 203. doi:10.1186/1471-2105-15-203.

[14] TIBCO Spotfire Data Visualization and Analytics Software | TIBCO Software, (n.d.). https://www.tibco.com/de/products/tibco-spotfire (accessed March 14, 2019).

[15] Healthcare Analytics: Data Discoveries With Tableau, (n.d.). https://www.tableau.com/solutions/healthcare-analytics (accessed June 6, 2019).

[16] R Core Team, R: A Language and Environment for Statistical Computing, (2018). https://www.r-project.org/.

[17] G. van Rossum, Python tutorial, Amsterdam, 1995.

[18] P.T. Inc., Collaborative data science, (2015). https://plot.ly.

[19] T. Oliphant, NumPy: A guide to NumPy, (n.d.). http://www.numpy.org/.

[20] W. Chang et al., shiny: Web Application Framework for R, (2018). https://cran.r-project.org/package=shiny.

[21] J. Bezanson et al., Julia: A Fast Dynamic Language for Technical Computing, (2012). http://arxiv.org/abs/1209.5145 (accessed March 20, 2019).

[22] Welcome | Flask (A Python Microframework), (n.d.). http://flask.pocoo.org/ (accessed March 6, 2019).

[23] E. Jones et al., SciPy: Open source scientific tools for Python, *Http://Www.Scipy.Org/.* (2001).

[24] F. Pedregosa et al., Scikit-learn: Machine Learning in Python, *J. Mach. Learn. Res.* **12** (2011) 2825–2830.

[25] Interactive JavaScript charts for your webpage | Highcharts, (n.d.). https://www.highcharts.com/ (accessed March 18, 2019).

[26] M. Bostock et al., D^3 Data-Driven Documents, *IEEE Trans. Vis. Comput. Graph.* **17** (2011) 2301–2309. doi:10.1109/TVCG.2011.185.

[27] Redis, (n.d.). https://redis.io/ (accessed March 20, 2019).

[28] H. Hund et al., Erschließung von klinischen Routinedaten für die medizinische Forschung, *Proc. 56. Jahrestagung Der GMDS*. (2011). doi:10.3205/11gmds466.

[29] H. Hund et al., Longitudinal data driven study design., *Stud. Health Technol. Inform.* **205** (2014) 373–7. http://www.ncbi.nlm.nih.gov/pubmed/25160209 (accessed March 19, 2019).

[30] C. Seyler et al., TranslatiOnal Registry for CardiomyopatHies (TORCH) - rationale and first results, *ESC Hear. Fail.* **4** (2017) 209–215. doi:10.1002/ehf2.12145.

[31] rainabba/jquery-table2excel: jQuery Plugin to export HTML tabled to Excel Spreadsheet Compatible Files, (n.d.). https://github.com/rainabba/jquery-table2excel (accessed June 4, 2019).