Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

SPOT: Open Source framework for scientific data repository and interactive visualization

Faruk Diblen [a,*], Jisk Attema [a,*], Rena Bakhshi [a], Sascha Caron [b], Luc Hendriks [b], Bob Stienen [b]

[a] Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands
[b] Institute for Mathematics, Astro- and Particle Physics IMAPP, Radboud Universiteit, Nijmegen, The Netherlands

A R T I C L E I N F O

Article history:
Received 20 September 2018
Received in revised form 4 April 2019
Accepted 4 April 2019

Keywords:
Visualization
High-dimensional data
Theoretical models
Open data
FAIR
Particle physics

A B S T R A C T

SPOT is an open source and free visual data analytics tool for multi-dimensional data-sets. Its web-based interface enables user to do a quick and interactive analysis of complex data. Various operations on data are implemented such as aggregation and filtering. The interface supports OpenGL acceleration, which makes the generated charts very responsive. In order to have scalability, the software also supports PostgreSQL as its database. It follows FAIR principles to allow reuse and comparison of the published data-sets.

Code metadata

Current code version	0.1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_178
Legal Code License	Apache License 2.0
Code versioning system used	Git
Software code languages, tools, and services used	Javascript, Node.js, SQL, PostreSQL
Compilation requirements, operating environments & dependencies	npm
If available Link to developer documentation/manual	https://nlesc.github.io/spot-framework/
Support email for questions	f.diblen@esciencecenter.nl, j.attema@esciencecenter.nl

Software metadata

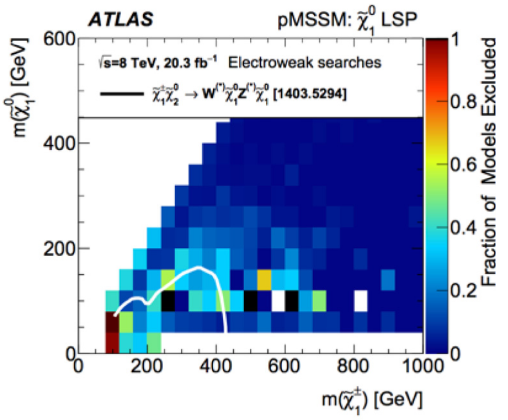
Current software version	0.1.0
Permanent link to executables of this version	https://github.com/NLeSC/spot/releases/tag/0.1.0
Legal Software License	Apache License 2.0
Computing platforms/Operating Systems	web based
Installation requirements & dependencies	Node.js
If available, link to user manual — if formally published include a reference to the publication in the reference list	https://research-software.nl/software/spot
Support email for questions	f.diblen@esciencecenter.nl, j.attema@esciencecenter.nl

1. Motivation and significance

Many scientific fields compare experimental data to complex theoretical simulations. These simulations can be based on approximations or full theoretical models and can have intrinsic complex structure consisting of multiple parameters or multiple

* Corresponding authors.
E-mail addresses: f.diblen@esciencecenter.nl (F. Diblen), j.attema@esciencecenter.nl (J. Attema), r.bakhshi@esciencecenter.nl (R. Bakhshi), scaron@nikhef.nl (S. Caron), luchendriks@gmail.com (L. Hendriks), B.Stienen@science.ru.nl (B. Stienen).

https://doi.org/10.1016/j.softx.2019.04.006



(b) Chargino–neutralino

Fig. 1. A typical scientific results from the field of High-Energy-Physics [6]. A model with 19 parameters (Supersymmetry) has been tested again the data to determine if parameter sets are excluded by the data or not. Here the fraction of model parameter sets (as color code) are plotted for two observables predicted by the model as *x*- and *y*-axis. This shows how difficult it is to visualize a 19-dimensional model on a paper and how much information is then lost . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

observables. Thus, these data-sets can be regarded as point clouds in a high-dimensional space. Due to the restrictions imposed by folding multi-dimensional structures on low-dimensional visualization, e.g. a figure in a journal, much of the information contained in the multi-dimensional graph is lost. A typical example is Fig. 1.

However, such a two dimensional (2D) representation obscures most of the correlations within the solution space. In order to encourage the publication of high-level data in the complete high-dimensional space, without restrictions, data visualization can be done via web-based tools which allow for, for example, an automatic generation of multiple histograms. The aim of SPOT [1] is to provide a flexible data visualization framework to visualize such data. SPOT, which is typically coupled to a database holding the data-sets is a tool to promote the use of *open research data* and *open science* [2]. It follows *FAIR* [3] principles to allow reuse and comparison of the published data-sets (see Fig. 2). This paper briefly introduces SPOT. General information about the software, and related information is given at [4]. The source code and the documentation is available at SPOT Github page [5].

2. Software description

SPOT provides users with an interactive data exploration environment for high-dimensional data-sets. The focus is on scientific use, with the aim of facilitating open science, data sharing and reuse (see Fig. 2). It is ideal for numerical data, but categorical (labeled) and temporal data is also supported.

Built on a number of concepts from the field of information visualization, it allows a user to create multiple coordinated views called *charts*, showing the data from different perspectives. All charts allow direct manipulation (i.e., selecting and zooming) of the data, and provide visual clues or *animations* when data changes due to user interaction.

2.1. Software architecture

The software consists of three components: a *framework*, a *frontend*, and a *server*. A brief description for each of these components follows:

SPOT-*framework*. The framework provides classes for data-sets, data views, partitions, aggregation and filtering. A *data-set* consists of a number of items (or rows), and each item has a set of *facets*. Facets can be used to partition the data, or they can be aggregated (counted). For numerical, data more complex operations are possible, namely, summation, averaging, extremes and standard deviation. One or more facets make up a *filter*, and all filters combined together form the *data view*. The user interacts with the framework by setting ranges or selections for the filters, and by adding or removing filters from the data view. After filtering, the partitioned and aggregated data is available as a simple array, which can be plotted or further processed. All filters in a data view are linked, and a change in one filter triggers an update of the whole data view.

Frontend. The frontend is a web-based application with separate pages where a user can upload and define data-sets, a dashboard page that provides the main interaction, and a page where analyses can be downloaded or shared. The frontend is built on a number of open-source JavaScript packages, such as Chart.js, Vis.js, and Sigma.js for animated plots, and Ampersand.js and Material Design Lite for the interface. The filtering is implemented using asynchronous functions, so updates can start as soon as the first results become available. The visualization libraries are HTML Canvas based, and offer OpenGL accelerated rendering where available.

Server. The server processes requests for data and applies the necessary filtering and aggregation. When data becomes available, it is pushed to the client which can then update the charts. We currently have two different implementations for the server component. The first one, which is included in SPOT-framework, is based on Crossfilter.js and runs in the user's web browser without requiring any further resources or even internet access. The second implementation (SPOT-server), provides a bridge to an external PostgreSQL database for scalability. Database queries are run in parallel, and make use of indices for extra performance. Connections to other datastores, like MySQL or MongoDB, can easily be achieved by extending the server component.

2.2. Software functionality

Data import and database connection. There are two options to import a new data-set. In the first option, users can upload data available on their own system. The software supports most common data formats CSV and JSON, to make data import process easy for different scientific domains. After the import, the data is checked to automatically detect data types, such as integers and strings. Users can then fix auto detection issues. In the second option, the data is imported from SPOT-server. The meta data for a data-set (e.g. the name and description) can easily be set in a configuration file stored on the server.

Dashboard. SPOT has eight ready-to-use chart types, namely, horizontal and vertical histograms, line chart, pie chart, bubble chart, 3-d scatter chart, radar chart and network chart. Charts are added to the dashboard by clicking the chart icon. The chart's filter requires one or more facets to partition over, and can take up to 4 facets to aggregate. Charts show their configuration pane by default. A visual feature of the chart can be linked to a specific facet by dragging a facet from the top of the screen and dropping it on a slot in the configuration pane. A Partition or Aggregation can be further configured by clicking on its name on the configuration pane.

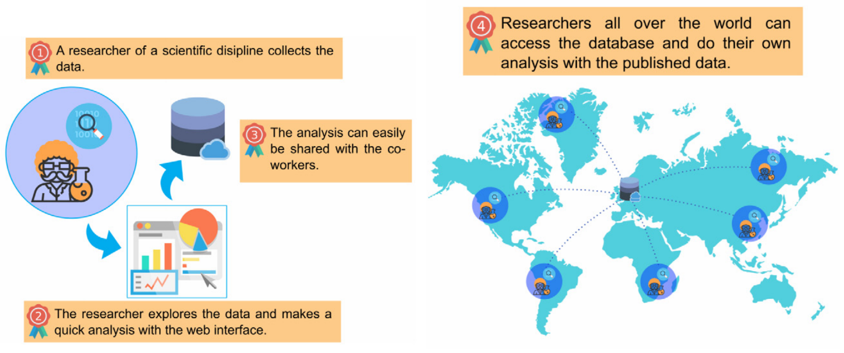

① A researcher of a scientific disipline collects the data.
② The researcher explores the data and makes a quick analysis with the web interface.
③ The analysis can easily be shared with the co-workers.
④ Researchers all over the world can access the database and do their own analysis with the published data.

Fig. 2. Workflow of SPOT.

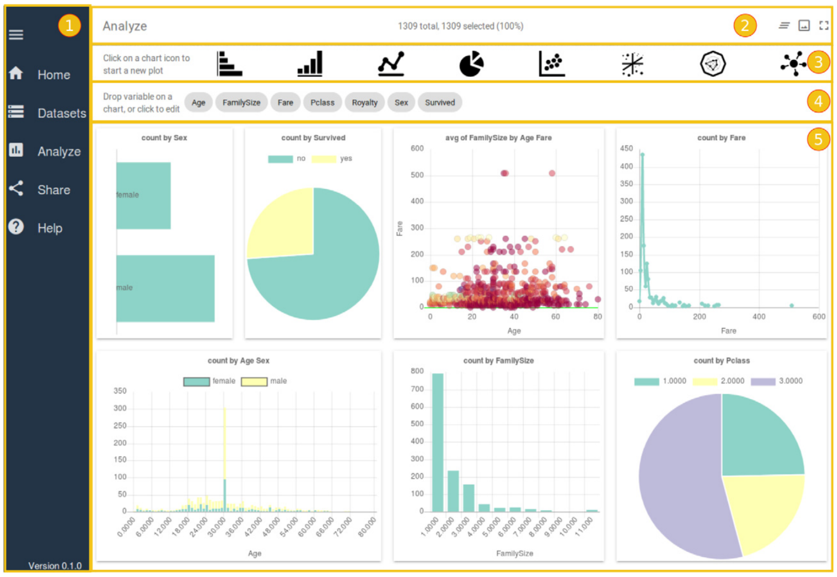


Fig. 3. Web-interface of SPOT. (1) Main menu: to navigate between different pages. Users can upload data (Datasets page), analyze it (Analyze page), share selection and charts as a *session* (Share page), and read help documentation (Help page); (2) Information bar: provides basic statistics about selected data; (3) Chart bar: in order to add available charts to the dashboard; (4) Variable bar: lists available variables within the data-set; (5) The dashboard.

Download and share. The dashboard generated by the user can be saved as a single file, a session file, in JSON format. This file contains aggregated data and settings of the dashboard such as existing charts, existing filters. The session file then can be used to restore the analysis. In addition, the session file can be uploaded to cloud storage or file-sharing platform where it can be shared.

3. Illustrative examples

We show the applicability and the features of the SPOT software on two examples of data-set: (1) the Titanic data-set [7], a well-known data-set in data science, and (2) a high-dimensional data-set containing models for dark matter (see [8,9] for more information).

3.1. Titanic data-set

Visualization plots of the Titanic data-set are provided in Fig. 3; the data can be viewed interactively on the idark survey website by clicking on Demo link [10].

The top of the figure shows the chart types, each of which can be selected to make a new chart. Directly below that are the data facets, which can be dragged-and-dropped into the empty charts to create any visualization of any parameter(s).

3.2. Example from high-energy physics

A real world example where SPOT can help the scientific community is visualizing, for example, models of high-energy physics. The data in this field is typically high-dimensional and even though different models have different theoretical parameters, they share the same observables. SPOT allows comparison of these observables from different data-sets, providing the user an unprecedented ability to compare the model space. In Fig. 4, two data-sets for models predicting Dark Matter are compared for three observables: the dark matter mass, a annihilation probability σv and a cosmic density Ωh^2.

4. Impact

The high-dimensional space can be stored in the SPOT database so that the data-set can be published along with the paper. While the paper still contains the most relevant 2D plots, a researcher can plot different variables using SPOT for further research. By

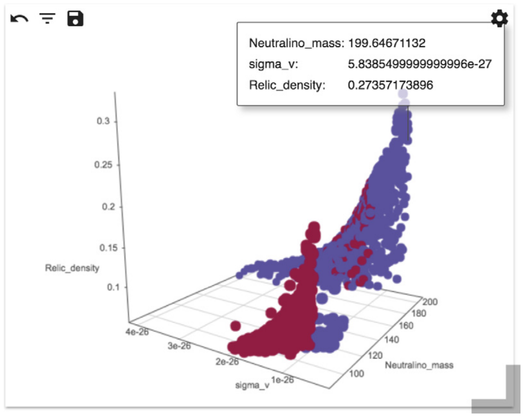

Neutralino_mass: 199.64671132
sigma_v: 5.8385499999999996e-27
Relic_density: 0.27357173896
Relic_density
sigma_v
Neutralino_mass

Fig. 4. An example 3D visualization of a comparison between two dark matter model data-sets. There are three shared observables. When hovering over a data point, a pop-up tooltip will show the actual values of the model point.

intuitively making cuts in histograms, researchers can investigate the high-dimensional space in an unprecedented way. In addition, comparison between data-sets on shared observables is likely to lead to new research questions.

Intuitive data visualization is still in its infancy, especially in the field of high energy physics. Although advanced analysis frameworks were developed for this field, such as ROOT [11], they still require expert knowledge. SPOT aims to be the first tool to provide an intuitive interface for visualizing high-dimensional data-sets and as a place for researcher to store their high-dimensional data.

Thus, there are two categories of applications that are related to SPOT: online sharing services and visualization libraries. The most relevant to SPOT are Microsoft Power BI [12], Spotfire [13] and Tableau [14] but these are commercial products. The most popular sharing services include data sharing platform Zenodo [15], digital repository for sharing Figshare [16], and high-energy physics specific platform Hepdata [17]. In comparison with SPOT, however, these services basically provide only storage that allow researchers to upload and publicly store their data and figures, but not compare different visualizations from different articles. On the other hand, visualization frameworks such as Dash [18] is a Python framework that gives users possibility of interactive visualization, but also it requires users to have an advance knowledge to create a desired dashboard.

5. Conclusions

The software has been written to provide a free and open state-of-the-art platform for researchers in many scientific domains. It helps researchers to publish, share their data-sets, and collaborate by comparing their data-sets to identify the differences. This makes SPOT a perfect candidate for FAIR data platforms.

CRediT authorship contribution statement

Luc Hendriks: Validation, Writing - review & editing.

Acknowledgments

This work is supported by the Netherlands eScience Center under the project iDark: The intelligent Dark Matter Survey.

References

[1] Attema J, Diblen F. Nlesc/spot: version 0.1.0. 2017, http://dx.doi.org/10.5281/zenodo.1003346.
[2] Machado J. Open data and open science. Open Sci. 2015;189.
[3] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, da Silva Santos LB, Bourne PE, et al. The fair guiding principles for scientific data management and stewardship. Sci. Data 2016;3.
[4] Spot research software directory page. 2018, https://research-software.nl/software/spot.
[5] Spot github page. 2018, https://github.com/NLeSC/spot.
[6] Aad G, et al., ATLAS Collaboration Collaboration Summary of the atlas experiment's sensitivity to supersymmetry after lhc run 1 interpreted in the phenomenological MSSM. J High Energy Phys 2015;10:134. http://dx.doi.org/10.1007/JHEP10(2015)134, arXiv:1508.06608.
[7] Kagglecom. Titanic: machine learning form disaster. 2018, https://www.kaggle.com/c/titanic/data. Online. (accessed 01 June 2018).
[8] Achterberg A, van Beekveld M, Caron S, Gómez-Vargas GA, Hendriks L, de Austri RR. Implications of the fermi-lat pass 8 galactic center excess on supersymmetric dark matter. J Cosmol Astropart Phys 2017;2017(12):040.
[9] Achterberg A, Amoroso S, Caron S, Nikhef A, Hendriks L, Austri RRd, Weniger C. A description of the galactic center excess in the minimal supersymmetric standard model. J Cosmol Astropart Phys 2015;2015(08). http://dx.doi.org/10.1088/1475-7516/2015/08/006.
[10] Idak survey webpage. 2018, http://www.idarksurvey.com.
[11] Brun R, Rademakers F. Root—an object oriented data analysis framework. Nucl Instrum Methods Phys Res A 1997;389(1–2):81–6.
[12] Lachev T, Price E. Applied microsoft power BI: bring your data to life!. 3rd edition. Prologika Press; 2018.
[13] Ahlberg C. Spotfire: an information exploration environment. SIGMOD Rec. 1996;25(4):25–9.
[14] Heer J, Mackinlay J, Stolte C, Agrawala M. Graphical histories for visualization: supporting analysis, communication, and evaluation. IEEE Trans. Vis. Comput. Graph. 2008;14(6).
[15] OpenAIRE CERN. Zenodo. 2018, https://zenodo.org/.
[16] DScience. Figshare. 2018, http://figshare.com.
[17] Maguire E, Heinrich L, Watt G. Hepdata: a repository for high energy physics data. In: Journal of Physics: Conference Series, vol. 898, IOP Publishing; 2017, p. 102006.
[18] Plotly. Dash. 2018, http://dash.plot.ly.