IRootLab: a free and open-source MATLAB toolbox for vibrational biospectroscopy data analysis

Júlio Trevisan\textsuperscript{a,b,*}, Plamen P. Angelov\textsuperscript{a,b}, Andrew D. Scott\textsuperscript{c}, Paul L. Carmichael\textsuperscript{c} and Francis L. Martin\textsuperscript{b}

\textsuperscript{a}School of Computing and Communications, Infolab21, Lancaster University, Lancaster LA1 4WA, UK
\textsuperscript{b}Centre for Biophotonics, Lancaster Environment Centre, Lancaster University, Lancaster LA1 4YQ, UK
\textsuperscript{c}Safety and Environmental Assurance Centre, Unilever Colworth Science Park, Bedfordshire, MK44 1LQ, UK

ABSTRACT

Summary: IRootLab is a free and open-source MATLAB toolbox for vibrational biospectroscopy (VBS) data analysis. It offers an object-oriented programming class library, graphical user interfaces (GUIs), and automatic MATLAB code generation. The class library contains a large number of methods, concepts and visualizations for VBS data analysis, some of which are introduced in the toolbox. The GUIs provide an interface to the class library, including a module to merge several spectral files into a dataset. Automatic code allows developers to quickly write VBS data analysis scripts, and is a unique resource among tools for VBS. Documentation includes a manual, tutorials, Doxygen-generated reference, and a demonstration showcase. IRootLab can handle some of the most popular file formats used in VBS.

License: GNU-LGPL.

Availability: Official website: \url{http://irootlab.googlecode.com/}.

Contact: juliotrevisan@gmail.com

1 INTRODUCTION

Vibrational biospectroscopy (VBS) is the application of infrared (IR) and Raman spectroscopy to biological studies and biomedical applications (Martin \textit{et al.}, 2010). In the past decades, vibrational spectroscopy data processing was developed on the grounds of signal processing, statistics, and pattern recognition, resulting in a large number of computational methods that are routinely applied in the field. However, the field needs further development and establishment of data analysis strategies, partially due to a continuous push from instrumental advancements, but also motivated by challenges in pre-processing, biomarker extraction and standardization of inter-experimental datasets, and inspired by the increasing power of computers (Trevisan \textit{et al.}, 2012).

VBS data analysis is largely carried out by commercial softwares such as CytoSpec (\url{www.cytospec.com}), Neurodeveloper (\url{www.neurodeveloper.com}), Unscrambler (\url{www.camo.com}), Pirouette (\url{www.inforometrix.com}), OPUS (\url{www.bruker.com}) and Wire (\url{www.renishaw.com}). Popularly used commercial MATLAB toolboxes include the Neural Network, and the Bioinformatics Toolbox. Others are free to use, but closed-source, such as PRTools (Duin \textit{et al.}, 2007), and GA_ORS (Nikulin \textit{et al.}, 1998). While these are all high-quality software, the use of closed-source software contradicts the multidisciplinary aspect of the field, since researchers looking for new analysis strategies remain dependant from the software manufacturers. On the other hand, open-source softwares available are largely restricted to a given task, such as LibSVM (Chang and Lin, 2011), Functional Data Analysis toolbox (Ramsay \textit{et al.}, 2009), and various other resources available online.

IRootLab is a free/libre and open-source (FOSS) MATLAB toolbox created to fulfill the need for FOSS software encompassing every stage of VBS data analysis. It is a highly modular software that was created based on existing theory of data analysis (Kuncheva, 2004; Guyon \textit{et al.}, 2006; Hastie \textit{et al.}, 2007; Duda \textit{et al.}, 2001; Alpaydın, 2004; Bishop, 2006), and as such, applied to VBS (Somorjai, 2009; Griffiths and Haseth, 2007), bringing together families of methods such as outlier removal, preprocessing, feature extraction, feature selection, feature construction, classification, classifier aggregation, clustering; concepts such as cross-validation, random sub-sampling, peak detection, biomarker identification; a set of unique visualization options (Supplementary Figure S1); and a MATLAB code generator. The basic modular design is very simple, which makes IRootLab a flexible and intuitive resource for vibrational spectroscopy data analysis developers.

2 CLASS LIBRARY

The majority of IRootLab code lines are attributed to its hierarchically organized object-oriented programming (OOP) classes. The basic framework is constituted of three branches: datasets, blocks, and logs. A dataset is an object representing point spectra (with their respective classes, patient names etc), image maps, clustering data, or classifier estimations. A block is the basis of data manipulation. It is an object that allows mainly two operations: training and using. Training involves the use of training data to modify a block (according to its specific idiosyncrasy). Using consists of inputting data into a block to get various results, including output data, estimation data, figures, and HTML reports. Finally, logs represent the output of special blocks called analysis sessions, which perform complex analyses and generate output that does not fit into the usual dataset class.

Additionally, there are a few complementary branches, from which the most important are sub-dataset generation specs (SGS), feature subset grader (FSG), and peak detector. SGS is an abstraction of the concept of random sub-sampling (e.g., cross-validation, repeated random sub-sampling); FSG is used to evaluate sub-sets of features in feature selection tasks, freeing the feature selection algorithm from having to implement its own evaluation; peak detection (Coombes \textit{et al.}, 2003) is used in visualizations and biomarker identification.

\textsuperscript{*}To whom correspondence should be addressed.
Blocks can be created and combined recursively (e.g., a special block called “block cascade” is composed of other blocks, which in turn can be used as a component of a higher-level block cascade) to build analyses as complex as desired.

Many complex operations are already offered as sub-classes of a special class of block called analysis session, including grid search optimization (Hsu et al., 2010), repeated train-test using an SGS object to obtain performance estimation of a classifier, and cross-calculation of LDA scores (Riding et al., 2012).

3 GRAPHICAL USER INTERFACES (GUIs)

Figure 1 shows the interaction of IRootLab GUIs with the class library and their surrounding environment. IRootLab offers a flexible and intuitive GUI called objtool that is an interface to the class library. Objtool allows datasets and other objects present in the MATLAB workspace to be browsed and manipulated using new or existing blocks. Objtool can handle three types of TXT files; OPUS image maps; and a native IRootLab MAT format.

Mergetool is another GUI that allows a collection of single-spectrum files to be merged together to form a dataset. Currently, mergetool can import three different types of single-spectrum files: Pirouette .dat, OPUS single-spectrum files, and Wire TXT files. Finally, the SHEload GUI accesses an online MySQL database that is part of a chemical database project (Trevisan et al., 2010), importing datasets from there.

4 CODE GENERATION

One of the major features of IRootLab is the ability of objtool to generate MATLAB code (similar to macro recording in Microsoft Word). This is a highly efficient resource for MATLAB scripting. In fact, a significant part of IRootLab was written using this resource. Code generation also keeps developers from having to consult the library documentation too often.

5 DOCUMENTATION

IRootLab has a clear documentation project. The official documentation (available at http://bioph.lancs.ac.uk/irootlabdoc) is generated directly from the source code using Doxygen software. Apart from this, IRootLab has a manual, and a series of tutorials (available at downloadable area of the official website) that together provide a friendly start for new users. The official reference can be accessed from MATLAB in two different occasions: from the command line, by using the help2 command; and from the GUIs, by pressing the F1 key (context-sensitive help). Additionally, a demonstration showcase was prepared (which can be opened by typing the command browse_demos at MATLAB command line), and a number of sample datasets are shipped with the toolbox.

6 CONCLUSION

Although the most valuable contributions of IRootLab are its framework and its code generation, there are currently more than 200 implemented OOP classes representing methods, algorithms, concepts, and visualizations. However, it is not guaranteed that any user will find their desired method(s) among the existing set thereof. However, being FOSS software means that any user can expand the toolbox to contain new methods necessary to their own analysis, and then share these new additions, creating a synergic effort among the scientific community. Expansion is facilitated by IRootLab modular structure, which allows developers to target or inherit a specific OOP class. IRootLab is a unique and innovative contribution to the interdisciplinary research field of VBS, providing researchers with a valuable tool for the development of VBS data analysis.

ACKNOWLEDGEMENT

IRootLab uses a number of 3rd-party packages, whose developers we would like to thank: Chih-Chung Chang, Chih-Jen Lin, Jacob Filik, James O. Ramsay, Michael Nasse, Paul Bassan, and Yannick Maret. We would also like to thank our colleagues Abdullah A. Ahmadzai, Imran I. Patel, Jemma G. Kerns, and Valon Llbjani for their participation in the conception. The project was partly funded by Unilever.

REFERENCES


