Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Systems Biology

Machine Learning Pipeline for Feature Selection and Model Performance Analysis

Corresponding Author[1,*], Co-author[2] and Co-Author[2]

[1]Department of XXXXXXX, Address XXXX etc., [2]Department of XXXXXXX, Address XXXX etc.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary

Big data needs of specialized computational tools for its analysis. Medical data is a special kind of big data, where finding insights in the data can yield important biological knowledge about health and disease. In order to facilitate this process, a supervised machine learning pipeline with feature selection through regularization algorithms (LASSO: Least Absolute Shrinkage Selector Operator and EN: Elastic Net) and Random Forest impurity importance was built. Model building was performed by combining the selected features in 6 different and contrasting models. A Shiny App with these steps was developed with the objective to create an easy and illustrative pipeline imbued with the ability to derive insights into which data elements matter to later derive biological insights.

Availability and Implementation

This pipeline is distributed as an RShiny package that can be found at:

Contact: example@example.org
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Medical data has unique features compared with big data in other domains, the data may include administrative health data, biomarker data, biometric data (for example, from wearable technologies) and imaging, and may originate from many different sources, including EHRs, clinical registries, biobanks, the internet and patient self-reports. Medical data can also be characterized and vary by states such as (i) structured versus unstructured (for example, diagnosis codes versus free text in clinical notes); (ii) patient-care-oriented versus research-oriented (for example, hospital medical records versus biobanks); (iii) explicit versus implicit (for example, checkups versus social media), and (iv) raw versus ascertained (data without processing versus data after standardization and validation processes). In order to make sense out of it, data has to be processed and analyzed. In order to facilitate this process, the bioinformatics field is making progress in the software development domains, making open source and freely accessible tools. But this tools most of the time require of experts knowledge for access, download and correct operation. Our objective with this new tools is to make a practical, simple and effective software, with conclusive visualization and results upon which actionable decisions can be taken and useful biological insight generated. Examples of previous pipelines are:[1] ... [HBD, extra others]

2 Machine Learning Pipeline application

2.1 Local installation and data structure

We developed this pipeline for the visualization of data and generation of feature selection and model building results. It has been created through the R software (version x) , and the use of R Shiny apps.Further instructions on how to deploy the app in a local shiny server are documented in the GitHub repository. Input data can be commonly used formats to store raw – and analyzed data, such as delimited files (.txt,.csv,.xlsx) and further specifics about input data formats are detailed in the Supplementary Materials. The number of features or samples is not limited.

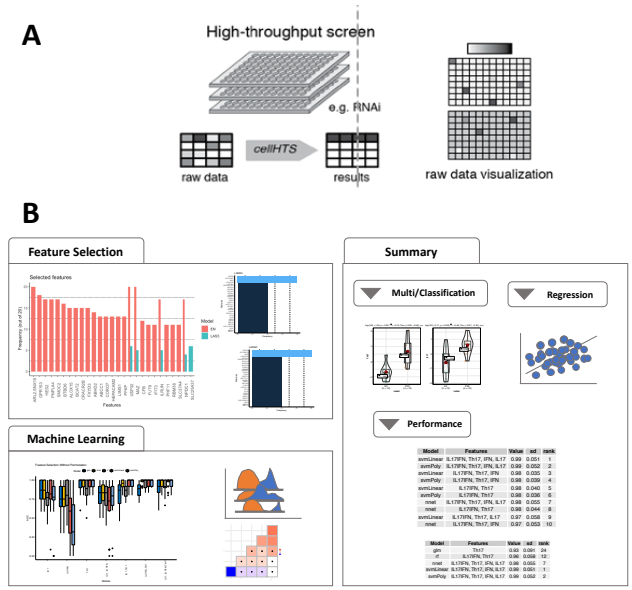

Figure 1 Modules in Shiny App. A – After raw data is introduced in the adequate format, the pipeline is run through to main modules feature selection and machine learning. These are the visualizations that are then obtained.

Figure 2 Shiny App.

2.2 Modules

2.2.1 Interactive data exploration

A simple visualization of all the data obtained is seen to start with (Conners plot) both normalized and non normalized.

2.2.2 Feature Selection

Data is then split into permuted versions of itself (bootstrap). With each of these permuted samples and after cross validation, a regularized model (both EN and LASSO) is created. Those frequency of appearance of the features in each model is quantified yielding the first plots (B1).

2.2.3 Machine Learning Models

Correlation analysis is then performed and those features with the least number of correlations and the higher number of appearances in the regularized model are then selected for the building of machine learning models. 6 different and complementary models are then used [reference] to predict the Label assigned at the start (Regression – continuous, Classification – binary and Multiclassification). The model is then created for all bootstrapped samples alongside a permuted model. Results can be seen in the second plot (B2). Finally a ranking is presented of the best models and best features with which a final univariate analysis is performed.

2.3 Conclusions

Acknowledgements

Funding

This work has been supported by the .....

Conflict of Interest: none declared.

References

1. Bravo-Merodio, L., Williams, J. A., Gkoutos, G. V. & Acharjee, A. -Omics biomarker identification pipeline for translational medicine. *J. Transl. Med.* **17**, 155 (2019).

Application Notes (up to 2 pages; this is approx. 1,300 words or 1,000 words plus one figure): Applications Notes are short descriptions of novel software or new algorithm implementations, databases and network services (web servers, and interfaces). Software or data must be freely available to non-commercial users. Availability and Implementation must be clearly stated in the article. Authors must also ensure that the software is available for a full TWO YEARS following publication. Web services must not require mandatory registration by the user. Additional Supplementary data can be published online-only by the journal. This supplementary material should be referred to in the abstract of the Application

Note. If describing software, the software should run under nearly all conditions on a wide range of machines. Web servers should not be browser specific. Application Notes must not describe trivial utilities, nor involve significant investment of time for the user to install. The name of the application should be included in the title.