TIMESPAN

Management of chronic cardiometabolic disease and treatment discontinuity in adult ADHD patients

H2020 – 965381

D6.2. – Genomic DLNNs (freely available via GitHub) (Task 4)

Dissemination level	Public
Contractual date of delivery	30. June 2022
Actual date of delivery	01. August 2022
Type	Report
Version	1
Filename	PRIME_Deliverable Report_D6.2.
Workpackage	6
Workpackage leader	Stephen Faraone (SUNY)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 965381.

Author list

Organisation	Name	Contact information
SUNY Upstate Medical University	Stephen V Faraone	sfaraone@childpsychresearch.org
SUNY Upstate Medical University	Yanli Zhang-James	Yanlizhang29@gmail.com
SUNY Upstate Medical University	Eric Barnett	Barnette@upstate.edu

Abbreviations

ML	Machine learning
DL	Deep Learning
DLNN	Deep Learning Neural Network
CNN	Convolutional Neural Network
SNP	Single Nucleotide Polymorphism

Table of Contents

1. Executive Summary

The main objective of the D6.2 is to create innovative Deep Learning Neural Network (DLNNs) using convolutional layers for genomic data. Our machine learning and deep learning framework for this objective is now complete and the codes are freely available via Github repository (https://github.com/barnettej/Genomic-CNN).

2. Deliverable report

Several methods have been developed attempting to use the power of convolutional layers with genomic data to improve prediction performance. However, most of these methods do not account for the intricacies of convolutional filters and thereby produce results that are either inaccurate or unrealistically accurate due to data leakage and overfitting. One main issue with most applications of convolutional layers to genomic data is the lack of spatial invariance in genomic data. Convolutional filters require spatial invariance, meaning a signal on one part of the two-dimensional data structure means the same thing as the same signal anywhere else in the data structure. While this holds true for images that convolutional layers were originally developed for, genomic data is not uniformly structured and identical signals at different areas of the data structure could mean vastly different things. We developed a method that combats this issue while still taking advantage of convolutional layers to minimize the number of weights in the model and incorporate genomic context through including annotations for each Single Nucleotide Polymorphism (SNP) used in a genomic dataset.

In our method we start each convolutional layer with a relatively large number of filters. These filters are all of width 1 and height equal to that of the context informed data matrix that includes all SNPs and their annotations within the same column. After the convolutional layer, the resulting tensor from each convolutional filter is fed into a global average pooling layer. This allows the model to look for consistent patterns among annotations and the related genotype that are different between cases and controls when averaged across the genome. It could also be thought of as many polygenic risk scores that each include a different data-driven combination of annotations. This method does not run into spatial invariance issues because each filter looks at a single SNP at the same time, so the filters have the same spatial meaning at any point in the data structure. It also minimizes the weights through the average pooling layer, thereby allowing many convolutional filters and a robust dense layer following the convolutional layers.

The codes for these tools and models are freely accessible via our github repository (see the link above). Sufficient documentations are included within the code files to facilitate adaptation to user-specific dataset.

All codes are written in Python, using R, Scikit-learn (Pedregosa et al., 2012), Keras (Charles, 2013) and Tensorflow libraries (Abadi et al., 2016; GoogleResearch, 2015).

Briefly, this repository contains the following files:

1. Generation of context informed data matrix: Adding genomic annotations
2. Generation of context informed data matrix: Correlation finding
3. Generation of context informed data matrix: Creating Genomic input for CNN
4. Matched pairing
5. Genomic CNN including Keras-Tuner search for selecting optimal hyperparameters

3. Conclusion

We have now completed D6.2. These models (and their supporting methods) are freely available for the scientific community and will be readily implemented in our next phase analysis.

References

Abadi, M., Barham P, Chen J, Chen, Z., Davis, A., Dean, J., . . . Brain, G. (2016, November 2–4). *TensorFlow: A system for large-scale machine learning.* Paper presented at the Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA.

Charles, P. W. D. (2013). Keras repository. *GitHub.*(https://github.com/charlespwd/project-title).

GoogleResearch. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. doi:10.1109/TIP.2003.819861.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830. doi:10.1007/s13398-014-0173-7.2