Forensic Science International: Genetics Supplement Series 6 (2017) e407–e409

Contents lists available at ScienceDirect

Forensic Science International: Genetics Supplement Series

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

Kpop: A Python package for population genetics analysis

F.M. Mendes[a,*], C.C. Gontijo[b]

[a] FGA, University of Brasilia, Brazil
[b] Human Genetics Laboratory, University of Brasilia, Brazil

ARTICLE INFO

Keywords:
Population genetics
Bioinformatics
Machine learning
Python
Forensic genetics

ABSTRACT

Kpop is an open source Python package that detects population structure from biallelic data. It implements its own maximum-likelihood routine to estimate admixture coefficients and provides interfaces to run analysis from Structure [1], ADMIXTURE [2] and Plink. Kpop also simulates population dynamics, including a few different models for hybridization and genetic drift. This unified framework makes it convenient to assess how different histories of hybridization and isolation can produce certain patterns of population structure. Kpop integrates Scikit-Learn [3], and Tensorflow [4] packages that provide state-of-the-art implementations of machine learning (ML) algorithms in Python. While some techniques are widely used by the forensic genetics community (e.g., PCA, for dimensionality reduction), other techniques with similar objectives are not so common (e.g., t-distributed Stochastic Neighbor Embedding [5]).

1. Introduction

Python is a programming language that is playing an increasingly large role both in science and education. It is relatively easy to learn and use due to its clean syntax, low ceremony, and high level nature. Scientific applications can benefit from the language's high expressiveness and relative easy integration with low level C implementations to create high level interfaces with acceptable performance. Those characteristics are making Python a popular choice both in the academia and industry.

Kpop leverages some libraries from the scientific Python ecosystem to build a package for population genetics that integrates standard structure analysis with methods used by the machine learning community. While there are some tools for genetics analysis with Python, most of them seem to be focused either on genomics and molecular biology or phylogenetics. The two largest packages for population genetics, simuPOP and PyPop have very different goals than Kpop. simuPOP is mainly concerned with forward time simulations of populations under migration and selection. PyPop, on the other hand, focus on basic statistical tests for genetic data such as hypothesis test for Hardy-Weinberg equilibrium and linkage disequilibrium.

Kpop aims to provide an easy to use API that is designed to be conveniently used from an interactive session or simple scripts. It also provides a command line interface to many of its functionality so users that are not familiarized with Python can still use it. Kpop interacts with standard tools from the population genetics community such as the widely used Structure [1] and ADMIXTURE [2] programs. It can parse the output file of those programs as well as feed Kpop based populations as input data. This integration makes it easy to compare the results from standard structure analysis with other machine learning algorithms or to integrate them on a more complex data processing pipeline.

2. Project vision

While the main goal of Kpop is to build a convenient, consistent, and easy to use API, it also strives to have acceptable performance. Many performance critical parts are implemented in Cython, which is a mixed language designed to create C extensions modules for Python. This is a very standard approach on the Pydata community and it is used in many of Kpop's dependencies such as Numpy, Scikit Learn, and Tensorflow.

We assume that the main way users will interact with Kpop is through Jupyter notebooks. This is a convenient format that mixes code with text and multimedia. It is popular in the scientific Python community because of its abilities to mix an interactive shell with functionalities expected from a text editor and a multimedia platform. That said, Kpop API was designed to be easily discoverable from a notebook session and to provide useful feedback when being used from an interactive shell.

3. Underlying technologies

Numpy: efficient multidimensional arrays which are the basic data

* Corresponding author.
E-mail address: fabiomacedomendes@gmail.com (F.M. Mendes).

http://dx.doi.org/10.1016/j.fsigss.2017.09.159
Received 3 September 2017; Accepted 20 September 2017
Available online 22 September 2017

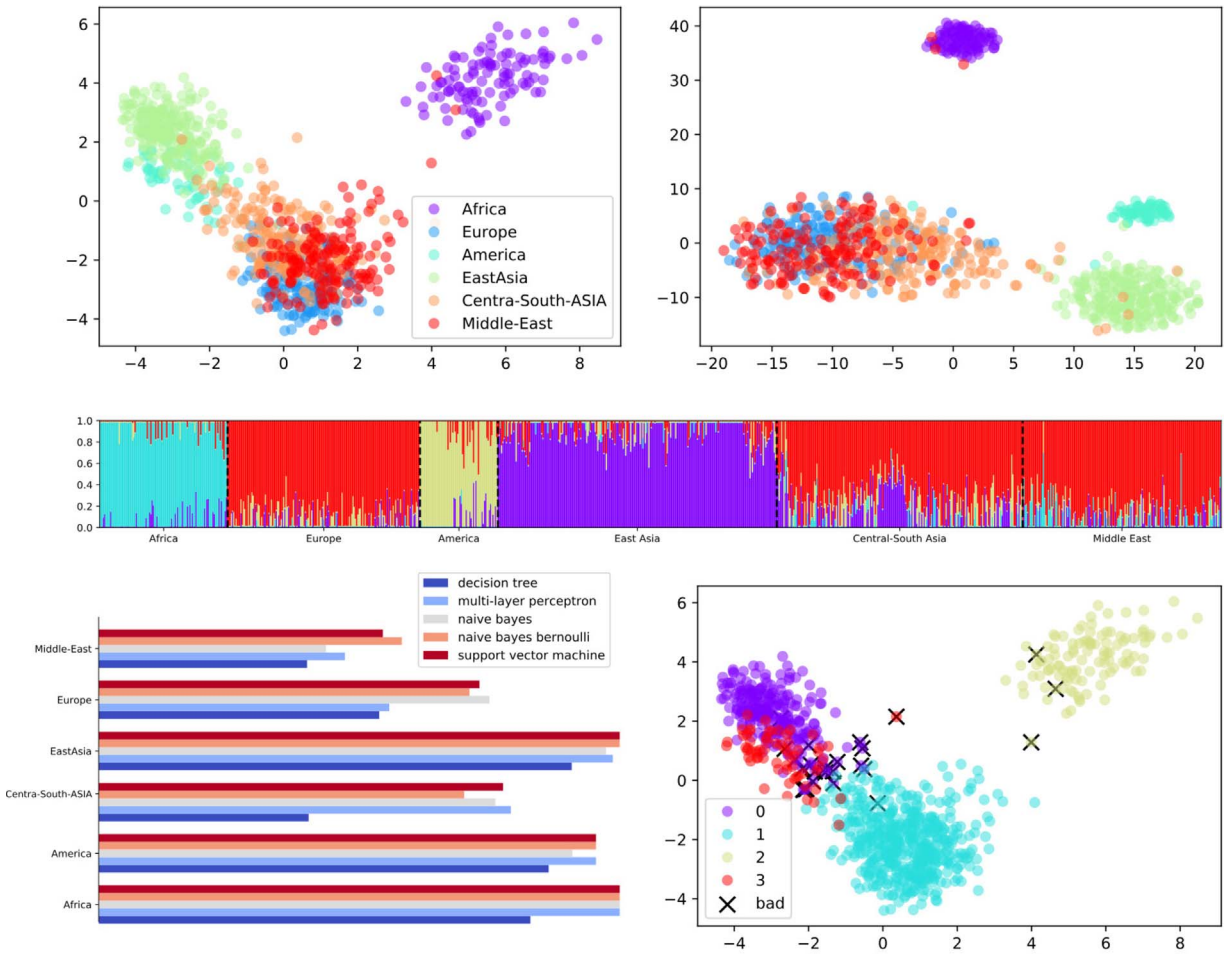

Fig. 1. Main results of the analysis of a human genetic data of 921 individuals in Kpop.

structure in Kpop.

Cython: a language that combines commands and constructs from both C and Python.

Scikit-learn [3]: a library with many state-of-the-art implementations of machine learning algorithms.

Structure: performs structure analyses using a Bayesian Gibbs sampling algorithm that infers both admixture coefficients and allelic frequencies of each sub-population. Kpop can call the Structure executable, if it is available.

ADMIXTURE: similar analysis as Structure, but it uses a different algorithm based on maximum likelihood.

Tensorflow [4]: provides tools and implements machine learning algorithms with the focus on neural networks.

4. Kpop architecture and objects

Kpop exposes most of its functionality through 'kpop.Population' instances. A population is modelled as a collection of individuals and possesses many methods and attributes that provide basic statistics, visualization, and more sophisticated analysis. They also have limited resources for simulating basic scenarios such as genetic drift and breeding.

Population objects use inner namespaces to organize functionality in a searchable way. For instance, 'population.plot' contains many functions for visualization while 'population.cluster' contains all functions related to clustering. This makes the API more compact and organized and more discoverable in an interactive session.

Kpop uses Python's pickle binary format for persistence and it also defines a human readable CSV (comma separated values) format. Population files can be converted to other formats such as Plink's PED or the data format used by Structure. This unified interface makes it a convenient tool to compare the results of different programs such as Structure and ADMIXTURE with machine learning algorithms.

5. Machine learning algorithms in genetics

The population genetics and machine learning communities have discovered and used independently many popular algorithms and there are some stories of cross fertilization. Kpop benefits from this since many important algorithms in population genetics have high quality implementations aimed at machine learning applications that are ready for use. We should mention notable algorithms such as PCA, LDA (the algorithm used by Pritchard et. al. in the STRUCTURE package), MDS (also known as PCoA) and Naive Bayes.

However, many popular machine learning algorithms are still at best only sparsely used in the population genetics community and can be proven to be useful for data analysis.

5.1. Visualization and dimensionality reduction

The main tool for dimensionality reduction and visualization of genetic data is Principal Component Analysis (PCA). While it is very useful and there are very efficient implementations, PCA has some limitations. It is not particularly good for detecting structure in some data sets and can present spurious superpositions specially if the first two principal axis do not explain most of the variance. In order to overcome this limitations, it is often necessary to perform non-linear transformations of data in order to prevent occlusion, thus revealing

clusters present in the high dimension space.

The most common non-linear dimensionality reduction algorithm in population genetics is MDS (Multidimensional Scaling), but now some authors are starting to explore more advanced algorithms [5]. Kpop integrates all manifold learning methods of Scikit Learn. Not all of those algorithms are equally useful in the context of genetic data analysis, and here we list only the most promising ones.

5.2. Clustering and classification

A classification problem consists of assigning labels to samples. In the context of machine learning, a labeled data set is used to train the computer to reproduce the labels trying to find patterns that allow it to generalize the classifier to yet unseen data points. Classification can be used in phenotype attribution and population attribution.

Traditional approaches include admixture analysis and classification using Naive Bayes in which each individual receives a single parental label. While those principled approaches work successfully in many cases, there are situations that violate some of the underlying assumptions (HWE) and could be handled better by more generic and powerful methods.

Kpop integrates clustering, classification, and regression algorithms from Scikit Learn and Tensorflow. Many of those algorithms are generally regarded as having a better performance than popular solutions such as e.g. Naive Bayes.

6. Results

Fig. 1 summarizes the most important methods implemented in Kpop. It uses data from CEPH of 921 individuals for the 46 IndelPlex system. The upper left plot is a PCA visualization of the data set, followed by a more advanced visualization using t-SNE on the right. In the middle, we show an admixture plot for 4 parental populations that was computed using a variational Bayes methods implemented in Kpop. Finally, the lower right plot shows cluster data using the *k*-means algorithm, where it was reported to have only 2.4% of misclassifications. The lower left plot shows the classification performance for finding the original parental populations using a few different algorithms. African populations provide a benchmark in which most algorithms classified all individuals correctly.

7. Conclusion

Kpop is a Python library designed to integrate tools from the population genetics community with machine learning algorithms (most of them implemented by the Scikit learn package). It is designed to be easy to use for novice programmers, while being flexible, extensible, and convenient to use by more advanced users.

Kpop shares the vision that machine learning algorithms have a larger role to play in the population genetics community. We try to provide an easy to use and convenient bridge between those two worlds.

References

[1] J.K. Pritchard, M. Stephens, P. Donnelly, Inference of population structure using multilocus genotype data, Genetics 155 (2000) 945–959, http://dx.doi.org/10.1111/j.1471-8286.2007.01758.x arXiv:0208024.

[2] D.H. Alexander, J. Novembre, K. Lange, Fast model-based estimation of ancestry in unrelated individuals, Genome Res. 19 (9) (2009) 1655–1664, http://dx.doi.org/10.1101/gr.094052.109.

[3] F. Pedregosa, G. Varoquaux, Scikit-learn: Machine Learning in Python vol. 12, (2011), http://dx.doi.org/10.1007/s13398-014-0173-7.2 arXiv:1201.0490v2. http://dl.acm.org/citation.cfm?id=2078195.

[4] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, G. Brain, TensorFlow: a system for large-scale machine learning, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), (2016), pp. 265–284 arXiv:1605.08695. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[5] W. Li, J.E. Cerise, Y. Yang, H. Han, Application of t-SNE to human genetic data, J. Bioinform. Comput. Biol. (2017) 1750017, http://dx.doi.org/10.1142/S0219720017500172.