G&I
Genomics & Informatics

Application note

eISSN 2234-0742
Genomics Inform 2023;21(1):e14
https://doi.org/10.5808/gi.22079

Received: December 27, 2022
Revised: March 9, 2023
Accepted: March 10, 2023

*Corresponding author:
E-mail: jeongbin.park@pusan.ac.kr

PyOncoPrint: a python package for plotting OncoPrints

Jeongbin Park[1]*, Nagarajan Paramasivam[2]

[1]School of Biomedical Convergence Engineering, Pusan National University, Busan 50612, Korea
[2]Computational Oncology, Molecular Precision Oncology Program, National Center for Tumor Diseases (NCT), German Cancer Research Center (DKFZ), Heidelberg 69120, Germany

OncoPrint, the plot to visualize an overview of genetic variants in sequencing data, has been widely used in the field of cancer genomics. However, still, there have been no Python libraries capable to generate OncoPrint yet, a big hassle to plot OncoPrints within Python-based genetic variants analysis pipelines. This paper introduces a new Python package PyOncoPrint, which can be easily used to plot OncoPrints in Python. The package is based on the existing widely used scientific plotting library Matplotlib, the resulting plots are easy to be adjusted for various needs.

Keywords: OncoPrint, plotting, Python
Availability: The package is available online on Github: https://github.com/pnucolab/pyoncoprint.

Introduction

OncoPrint, the plot to visualize an overview of the genetic variants of the deposited data at cBioPortal [1,2], has become popular, especially in the field of cancer genomics [3-7]. Although OncoPrints can be easily drawn and exported at the cBioPortal website, it is difficult to generate OncoPrints via the website from the command-line-based bioinformatics workflows. To tackle this problem, there has been an R implementation [8] to plot OncoPrints, however, still there have been no Python implementations so far. We introduce a novel Python package, PyOncoPrint, for plotting OncoPrints in Python. PyOncoPrint supports various scenarios of plotting OncoPrints, such as plotting metadata, variant statistics, etc. alongside the main OncoPrint, so that it can be directly used as a figure of a paper with no modifications.

Methods

Implementation

The package is mainly based on Matplotlib [9], the de facto standard Python plotting library. The variant markers are plotted using the 'scatter' function of Matplotlib, which enables the plotting of all scatter plot marker shapes available in the Matplotlib package. In addition to conventional markers, custom markers can be designed by defining polygon coordinates. Thus, one can define as many marker shapes for printing various types of genetic variants.

In addition to the main plot (Fig. 1D), PyOncoPrint supports the plotting of subplots to provide more information. One of them is the 'annotations' plot (Fig. 1C) attached to the top of the main Oncoprint, which plots sample metadata. The annotations include

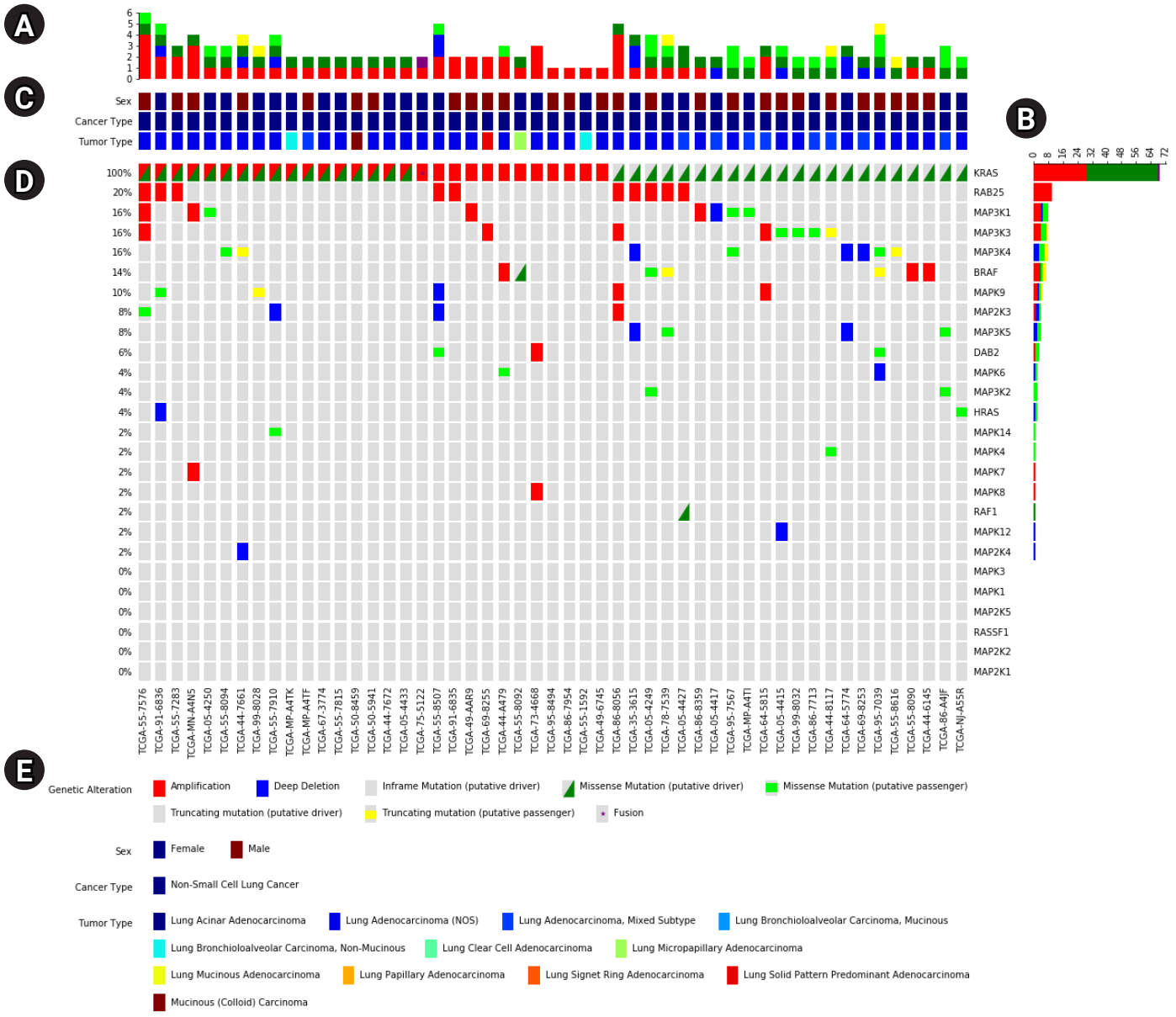

Fig. 1. An example OncoPrint based on The Cancer Genome Atlas data downloaded from cBioPortal. (A, B) The top and right plots that show the frequency of variants per type. (C) The annotations panel that displays metadata of samples. (D) The main OncoPrint. (E) The plot legend including the variant markers and annotations.

any categorical information of samples, such as sex, tumor type, etc. The annotations can be printed as a legend, and attached to the bottom of the plot (Fig. 1E). The other subplots are 'top plot' and 'right plot,' bar plots that summarize the frequency of the variants of samples (Fig. 1A) and genes (Fig. 1B).

PyOncoPrint can automatically sort the samples and genes by the frequency of each genetic variant so that one can easily overview the plotted genetic variants, and be ready to use as a figure in a research paper. The genetic variation data can be easily imported as Pandas data frame [10], as well as the CSV files exported from cBioPortal can be directly used as an input.

Basic usage

The input data to the PyOncoPrint is Pandas dataframe which contains a matrix of samples vs. genes. The input follows the format of cBioPortal's CSV exports—each element of the matrix defines variants as strings, concatenated with commas. Thus, one can either generate their own data or just convert a cBioPortal's export

https://doi.org/10.5808/gi.22079

to a Pandas dataframe, as an input to PyOncoPrint. By providing the input data with marker definitions and annotations, just one simple function 'pyoncoprint' generates the OncoPrint. A detailed example, that shows the basic usage of PyOncoPrint—including how to define the input data, markers, and annotations—is available online on our GitHub repository.

Results

We demonstrated visualization of OncoPrint of The Cancer Genome Atlas lung adenocarcinoma data using PyOncoPrint. The data was obtained from the Oncoprinter at cBioPortal, containing 24 genes and 996 patient samples as a tab-delimited format.

The downloaded data was then loaded as a Pandas dataframe object using the "read_csv" function of Pandas. The patients having no mutations in the 24 genes were truncated, resulting in 463 remaining patients having at least one mutation.

Next, the marker types for each mutation pattern were defined. For demonstration purpose, three different marker types for the mutation types were defined as following: (1) fill patterns, which fills the marker with a specified color and given height; (2) an asterisk symbol (*); and (3) a custom triangle pattern defined using "Polygon" class available in Matplotlib. All of the markers were defined with different colors so that the mutation types could be distinguishable from each other.

Finally, the plot was generated using "oncoprint" method of PyOncoPrint (Supplementary Fig. 1), which shows the mutational landscape of the lung adenocarcinoma patients.

Conclusion

We developed a novel Python package, PyOncoPrint, which provides an easy way to plot OncoPrints using Python. Thanks to its simple usage and easy-to-use interface, the package can be easily adapted to various Python-based command-line pipelines. The source code is freely available on our GitHub repository (https://github.com/pnucolab/pyoncoprint).

ORCID

Jeongbin Park: https://orcid.org/0000-0002-9064-4912
Nagarajan Paramasivam: https://orcid.org/0000-0002-7126-8472

Authors' Contribution

Conceptualization: JP. Data curation: JP, NP. Formal analysis: JP. Funding acquisition: JP. Methodology: JP. Validation: JP, NP. Writing - original draft: JP. Writing - review & editing: JP, NP.

Conflicts of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This work was supported by a 2-Year Research Grant of Pusan National University.

Supplementary Materials

Supplementary data can be found with this article online at http://www.genominfo.org.

References

1. Gao J, Aksoy BA, Dogrusoz U, Dresdner G, Gross B, Sumer SO, et al. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. Sci Signal 2013;6:pl1.
2. Cerami E, Gao J, Dogrusoz U, Gross BE, Sumer SO, Aksoy BA, et al. The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. Cancer Discov 2012;2:401-404.
3. Chudasama P, Mughal SS, Sanders MA, Hubschmann D, Chung I, Deeg KI, et al. Integrative genomic and transcriptomic analysis of leiomyosarcoma. Nat Commun 2018;9:144.
4. Paramasivam N, Hubschmann D, Toprak UH, Ishaque N, Neidert M, Schrimpf D, et al. Mutational patterns and regulatory networks in epigenetic subgroups of meningioma. Acta Neuropathol 2019;138:295-308.
5. Ishaque N, Abba ML, Hauser C, Patil N, Paramasivam N, Huebschmann D, et al. Whole genome sequencing puts forward hypotheses on metastasis evolution and therapy in colorectal cancer. Nat Commun 2018;9:4782.
6. Lopez C, Kleinheinz K, Aukema SM, Rohde M, Bernhart SH, Hubschmann D, et al. Genomic and transcriptomic changes complement each other in the pathogenesis of sporadic Burkitt lymphoma. Nat Commun 2019;10:1459.
7. Northcott PA, Buchhalter I, Morrissy AS, Hovestadt V, Weischenfeldt J, Ehrenberger T, et al. The whole-genome landscape of medulloblastoma subtypes. Nature 2017;547:311-317.
8. Gu Z, Eils R, Schlesner M. Complex heatmaps reveal patterns

and correlations in multidimensional genomic data. Bioinformatics 2016;32:2847-2849.
9. Hunter JD. Matplotlib: a 2D graphics environment. Comput Sci Eng 2007;9:90-95.
10. McKinney W. Data structures for statistical computing in Python. In: Proceedings of the 9th Python in Science Conference (SciPy 2010) (van der Walt S, Millan J, eds.). Austin: SciPy, 2010. pp. 56-61.

https://doi.org/10.5808/gi.22079