

# DNAPlotlib: standardized visualization of genetic constructs, libraries and associated data

Thomas E. Gorochoowski<sup>1</sup>, Bryan Der<sup>1</sup>, Emerson Glassey<sup>1</sup>, D. Benjamin Gordon<sup>1</sup> and Christopher A. Voigt<sup>1,\*</sup>

<sup>1</sup>Department of Biological Engineering, Synthetic Biology Center, Massachusetts Institute of Technology, USA.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXXX

## ABSTRACT

**Summary:** DNAPlotlib is a computational toolkit that enables highly customizable visualization of single genetic constructs and libraries of design variants. Publication quality vector-based output is produced and all aspects of the rendering process can be easily customized or extended by the user. DNAPlotlib is capable of outputting SBOL Visual compliant diagrams, in addition to a trace-based format that is able to better illustrate the precise location and length of each genetic part. This alternative visualization method enables direct comparison with nucleotide-level data such as RNA-seq read depth. While it is envisaged that access will be predominantly via the programming interface, command-line and web-based front-ends are also provided to support broader usage.

**Availability:** DNAPlotlib is cross-platform and open-source software developed using Python and released under the OSI recognized NPOSL-3.0 licence. Source code, documentation and a web front-end are available at the project website: <http://www.dnaplotlib.org>.

**Contact:** [cavoigt@gmail.com](mailto:cavoigt@gmail.com)

Engineering disciplines rely on standardized pictorial representations of parts and their interconnections to clearly communicate how these should be pieced together to form a design and to allow for the reliable construction of large complex systems. In biology, DNA sequences are often engineered to create genetic constructs that probe or perturb the function of natural systems, or more recently, create novel capabilities in what has been termed “synthetic biology”. Unlike traditional engineering fields, the way that these designs are visually represented varies significantly between labs and areas of the field, leading to ambiguities that can hinder understanding and the effective reuse of this research. The Synthetic Biology Open Language (SBOL) Visual initiative was started to help alleviate this problem by defining a set of agreed symbols for commonly used genetic elements (Quinn *et al.*, 2013). However, so far this standard has seen limited uptake due to a lack of accessible tools that can be directly integrated into existing design and analysis workflows.

To tackle this issue we developed DNAPlotlib, a computational toolkit that facilitates the allows for the

Some attempts to simplify the creation of standard-compliant diagrams have been made. The most prominent of these is PigeonCAD (Bhatia & Densmore, 2013), a web-based tool that interprets a text-based syntax for specifying a genetic design and automatically transforms this into a visual representation. While this tool is Unfortunately, this tool is hampered by several limitations. The biggest limitation of these approaches so far though is an inability to integrate with existing scientific computing visualization and analysis tools. This is essential as

Another growing area of research that is not well supported by such existing tools is the push towards automated design procedures that harness the potential to construct huge libraries of design variants (Smanski *et al.*, 2014; Bilitchenko *et al.*, 2011). Under these scenarios, automation of visualization tasks is essential to ensure clear communication of these large and diverse design spaces. No existing tools are available to support this need.

As synthetic biology moves ever closer to automated design procedures that harness the potential to construct huge libraries of design variants (Smanski *et al.*, 2014; Bilitchenko *et al.*, 2011), supporting visualization tools will become important to ensure clear communication of these large and diverse designs. The SBOL Visual initiative was started to aid in this task with the definition of agreed pictorial representations of genetic elements (Quinn *et al.*, 2013). However, so far it has seen limited uptake due to a lack a tools that are able to be easily integrated into existing analysis work-flows. DNAPlotlib fills this gap by providing a computational toolkit that both adheres to this standard while also enabling highly customizable visualizations that are most appropriate for the specific needs of the laboratory. It is hoped that by providing researchers with this flexibility that they will be more willing to apply such standards to their work, improving general uptake of SBOL Visual across the field.

To address these needs DNAPlotlib has been built using matplotlib (Hunter, 2007), a highly portable Python-based 2D graphics library that allows for graphical output in the form of vector based PDFs or rasterized PNG or JPEG images. A major advantage oFurthermore, it enables the rendering of genetic designs to be easily integrated into existing analysis scripts. Python was chosen due to its increasing use for the analysis of biological data (Cock *et al.*, 2009) and its availability across all major operating systems.

A key consideration in the design of DNAPlotlib was ensuring all aspects of the rendering process could be customized to specific

\*to whom correspondence should be addressed

user requirements. As fields such as synthetic biology are still evolving, the precise way in which such tools will be used and the discovery of new types of genetic part that need to be included, but may not yet have a standardized representation. Enabling custom elements to be easily added and refined over time will ensure such elements are captured and also contribute to the standardization process.

dictionary that contains core attributes corresponding to part type and orientation, in addition to other attributes that influence the style of the part. Attributes that are not used by a particular renderer are ignored, meaning that it is easy to swap these functions without fear of breaking the entire rendering pipeline.

To generate a In addition to direct library access through Python analysis scripts, we also provide two scripts to enable input in the form of text files

To ensure broadest application of these tools by non-programmers, a web-based interface was also developed. This allows for the visualization of single designs or the . Reduces the requirement of a user to have a working Python installation locally on their machine. that allows text files describing part styles and DNA designs to be uploaded and processed. These files uploaded to the server, processed by the scripts described previously, and generated images are returned to the user. This ensures broadest use of the library and ensures that members across a lab can share the same styling of diagrams to ensure clear communication of their designs.

We see a key feature of ensuring adoption as

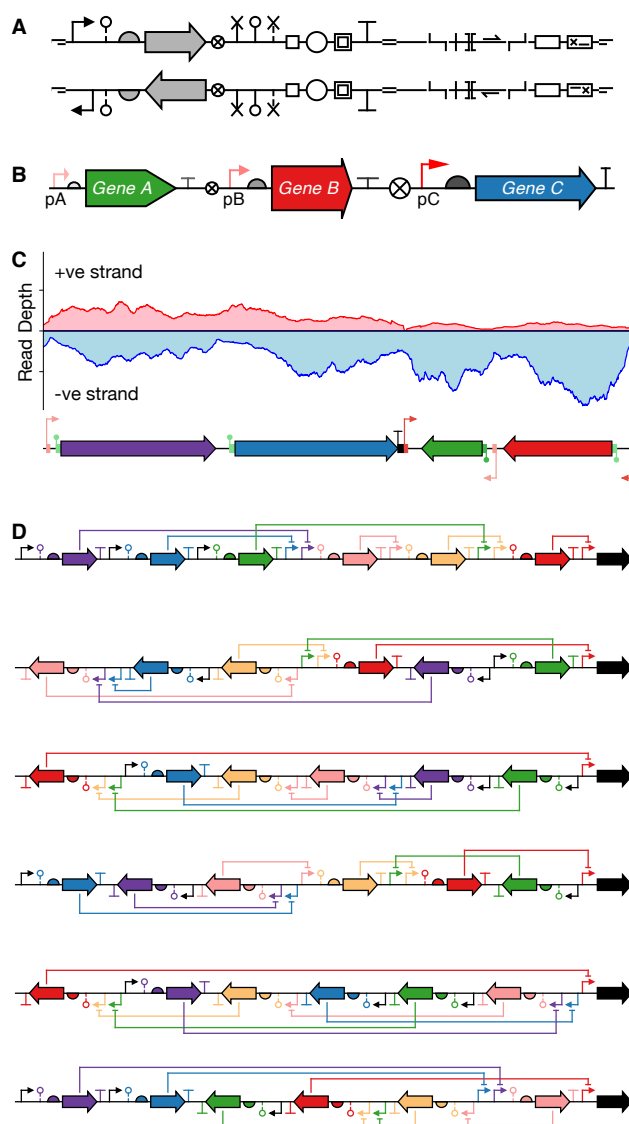
DNAPlotlib is under continual development with a current focus on broadening the types of genetic element covered to include new synthetic biological parts. The project welcomes contributions from others within the community through the project website and public development repository: <http://www.dnaplotlib.org>.

## ACKNOWLEDGEMENTS

T.E.G., B.D., E.G., D.B.G. and C.A.V. were supported by...

## REFERENCES

- Bhatia, S. and Densmore, D. (2013). Pigeon: A Design Visualizer for Synthetic Biology, *ACS Synth. Biol.*, **2**, 348-350.
- Smanski, M.J., Swapnil, B., Park, Y.J., Zhao, D., Giannoukos, G., Ciulla, D., Busby, M., Calderon, J., Nicol, R., Gordon, D.B., Densmore, D. and Voigt, C.A. (2014) Combinatorial design and assembly of refactored gene clusters, *Nat. Biotech.*, **?**, ???-???
- Hunter, J.D. (2007). Matplotlib: A 2D graphics environment, *Computing in Science & Engineering*, **9**, 90-95.
- Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, and de Hoon MJ. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422-1422.
- Bilitchenko, L., Liu, A., Cheung, S., Weeding, E., Xia, B., Leguia, M., Anderson, J.C., Densmore, D. (2011) Eugene A Domain Specific Language for Specifying and Constraining Synthetic Biological Parts, Devices, and Systems. *PLoS ONE*, **6**, e18882.
- Quinn, J., Beal, J., Bhatia, S., Cai, P., Chen, J., Clancy, K., Hillson, N., Galdzicki, M., Maheshwari, A.P., Umesh; P., Matthew; R.C.; Stan, G.-B., Endy, D. (2013) "Synthetic Biology Open Language Visual (SBOL Visual), version 1.0.0."



**Fig. 1.** Overview of core DNAPlotlib functionality. All visualizations were generated using DNAPlotlib. (A) The complete set of SBOL Visual parts that capture the majority of widely used genetic elements are available for use in both forward and reverse orientations. (B) The size, color, shape and labeling of all elements can be easily customized allowing for additional information to be communicated e.g., promoter strengths or spacer lengths. Users can further supply their own functions to draw parts in a non-standard way or to represent new types of genetic element yet to be incorporated into SBOL Visual. (C) To allow for direct comparison between a genetic design and associated nucleotide-level data, such as RNA-seq read depths, trace-based renderers are also provided. These use the standard promoter and terminator symbols and a small filled circle to represent an RBS. Indicators of actual part widths (arrow length for CDSs and small filled rectangles for promoters and RBSs) are displayed on the DNA backbone to enable a clear visual alignment of design information with trace data. (D) Visualization of a library of genetic design variants implementing the same 3-input (black promoters), 1-output (black CDS) device. Colors have been used to link repressor genes to their cognate promoters.