GdMetriX – A NetworkX Extension For Graph Drawing Metrics

Martin Nöllenburg
Algorithms and Complexity Group, TU Wien,Austria

Sebastian Röder
TU Wien, Austria

Markus Wallinger
Chair for Efficient Algorithms, Technical University of Munich, Germany

Abstract

networkX is a well-established Python library for network analysis. With *gdMetriX*, we aim to extend the functionality of *networkX* and provide common quality metrics used in the field of graph drawing, such as the number of crossings or the angular resolution. In addition, the package provides easy-to-use access to the graph datasets provided by the 'Graph Layout Benchmark Datasets' project from the Northeastern University Visualization Lab.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases Graph Drawing Metrics

Digital Object Identifier 10.4230/LIPIcs.GD.2024.45

Category Poster Abstract

Supplementary Material *Text (Documentation)*: `https://livus.github.io/gdMetriX/`

Funding *Martin Nöllenburg*: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].
Markus Wallinger: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

1 Introduction

networkX is a well-established and commonly used Python package for working with graph structures [1]. In their own words, *networkX* is intended for "the creation, manipulation, and study of the structure, dynamics, and functions of complex networks" [1]. It supports various data structures and algorithms useful in graph theory [4]. *networkX* also supports some basic graph drawing algorithms, such as for drawing planar graphs or spring embedding. Its focus however does not lie on graph drawing, but rather on graph analytics.

With the *gdMetriX* package, we aim to extend the functionality of *networkX* and provide common quality metrics used in graph drawing, such as the number of crossings or the angular resolution. In addition, the package provides easy-to-use access to the graph datasets provided by the "Graph Layout Benchmark Datasets" project from the Northeastern University Visualization Lab [3].

The project is published on the Python packaging index (see `https://pypi.org/project/gdMetriX/`). More information about all implemented metrics and additional features can be found at the project homepage (see `https://livus.github.io/gdMetriX/`) or the GitHub repository (see `https://github.com/livus/gdMetriX`).

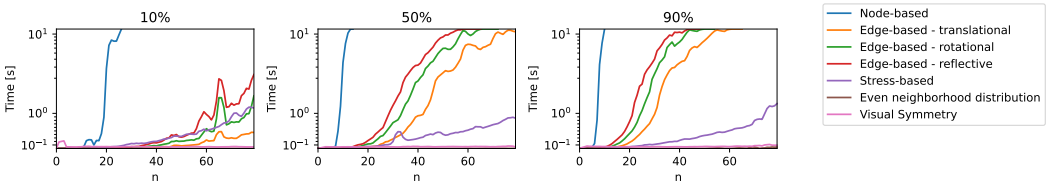

Figure 1 Runtime of symmetry metrics of random graphs with edge densities ranging from $10\% - 90\%$.

Table 1 Overview of all implemented graph drawing metrics.

Category	Metrics
Crossings	Crossing number, percentage of crossings compared to maximum number of crossings, crossing angles, crossing angular resolution
Area and boundary	Area, tight area, height, width, aspect ratio
Node distribution	Center of mass, closest pair of points, closest pair of elements, concentration, homogeneity, horizontal balance, vertical balance, node orthogonality, Gabriel ratio
Edge directions	Angular resolution, average flow, upwards flow, coherence to average flow, edge orthogonality
Symmetry	Node-based symmetry, edge-based symmetry, stress, even neighborhood distribution, visual symmetry

2 Graph Drawing Quality Metrics

Graph drawing quality measures are a practical and important tool to evaluate the aesthetics and readability of existing drawings and guide the creation of new drawings as well as the development of new drawing algorithms. However, for a large amount of metrics, no publicly accessible Python implementation is available. This increases implementation efforts for new projects and impedes standardization. Indispensable for collecting a set of suitable metrics were the publications by Taylor and Rodgers [7], Purchase [6], Bennett et al. [2], as well as the recent paper by Mooney et al. [5]. A list of all metrics can be found in Table 1.

The framework aims to provide efficient implementations for all metrics. Especially the symmetry metrics offer measures with similar properties but vastly different runtimes (see Figure 1). As some symmetry metrics are infeasible to compute for larger instances, we provide a novel metric – called "visual symmetry" – with linear runtime, aimed to quickly asses the symmetry of a drawing. The metric works by creating a constant-size image of a graph drawing. From this image, rotated and mirrored versions are subtracted and the ink left in the drawing is summed together. The less ink is left, the more symmetric the graph drawing is assumed to be. A more in-depth analysis can be found on the project homepage.

3 Access to Common Benchmark Datasets

Di Bartolomeo et al. [3] collected commonly used benchmark datasets for evaluating graph layout algorithms. To ease the usage of the datasets and make the collection more visible, we have implemented a module to download and parse the provided datasets automatically with a single line of code. The module downloads requested datasets, caches them for reuse, and parses the data for use with networkX.

References

1 networkX homepage. https://networkx.org/. Accessed: 2024-07-20.

2 Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The Aesthetics of Graph Visualization. *Computational Aesthetics in Graphics*, 2007. doi:10.2312/COMPAESTH/COMPAESTH07/057-064.

3 Sara Di Bartolomeo, Eduardo Puerta, Connor Wilson, Tarik Crnovrsanin, and Cody Dunne. A Collection of Benchmark Datasets for Evaluating Graph Layout Algorithms. In *Graph Drawing and Network Visualization (GD'23)*, volume 14466 of *LNCS*, pages 251–252. Springer, 2023. doi:10.31219/osf.io/yftju.

4 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11–15, 2008.

5 Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, and Stephen G. Kobourov. The Multi-Dimensional Landscape of Graph Drawing Metrics. In *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*, pages 122–131. IEEE, 2024. doi:10.1109/PacificVis60374.2024.00022.

6 Helen C. Purchase. Metrics for Graph Drawing Aesthetics. *Journal of Visual Languages & Computing*, 13(5):501–516, 2002. doi:10.1006/jvlc.2002.0232.

7 Martyn Taylor and Peter Rodgers. Applying Graphical Design Techniques to Graph Visualisation. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 651–656, London, England, 2005. IEEE. doi:10.1109/IV.2005.19.