Article

Not peer-reviewed version

momapy: A Python Library to Work with Molecular Maps

Adrien Rougny * , Marek Ostaszewski , Venkata Sagatopam

Posted Date: 12 January 2026

doi: 10.20944/preprints202601.0774.v1

Keywords: molecular map; sbgn; celldesigner; biological network; systems biology

 doi:10.20944/preprints202601.0774.v1

Article

momapy: A Python Library to Work with Molecular Maps

Adrien Rougny *, Marek Ostaszewski and Venkata Sagatopam

Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, Esch-sur-Alzette, Luxembourg
* Correspondence: adrienrougny@gmail.com

Abstract

Motivation: Molecular maps are graphical representations of the molecular mechanisms underlying biological systems. They are a valuable tool for curating, exchanging, and understanding biological knowledge, and may serve as a backbone for data analysis and modelling. Molecular maps are supported by a rich software ecosystem. However, there are currently no tools that support advanced programmatic analysis and processing of maps, in particular the extraction of the biological concepts they represent or their comparison. Results: We introduce *momapy*, a generic Python library to work with molecular maps efficiently. At its core, *momapy* allows users to extract and separate the data model of a map from its graphical representation, and perform a variety of base operations on them, including their manipulation and comparison. *momapy* currently supports the SBGN and CellDesigner formats, two of the main standards to represent molecular maps graphically, and can be easily extended to support additional formats and functionalities.

Keywords: molecular map; sbgn; celldesigner; biological network; systems biology

1. Introduction

Molecular maps are graphical representations of the molecular mechanisms underlying biological systems. They are detailed, curated, and exchangeable graphical knowledge bases that help to understand the mechanisms driving phenotypes of interest (e.g., diseases [1], induced cellular responses [2,3]). They may also support a variety of analyses, including analysis of omics data and dynamical modeling [4].

Molecular maps are supported by a rich software ecosystem. The available tools allow users to edit (CellDesigner™ [5], SBGN-ED [6], Newt [7]), visualise (MINERVA [8]), render (Newt, MINERVA, SBMLDiagrams [9]), format and convert (libSBGN [10], MINERVA, cd2sbgnml [11]), query (STONPy [12]) or model (SBGN2AN [13], CasQ [14]) maps. However, operations required for advanced programmatic analysis and processing are not supported by any of these tools. For example, it is currently not possible to extract the biological concepts represented by a map or to compare two maps automatically.

Here we introduce *momapy*, a generic Python library for working with molecular maps efficiently. At its core, *momapy* separates the *model* of a map from its *layout*: while the model of a map describes what biological concepts it represents, its layout describes how they are represented graphically. This distinction is borrowed from the Systems Biology Markup Language (SBML) [15], where a model defines a set of reactions and their associated mathematical equations, but may be augmented with a graphical layout using the layout and render packages [16,17]. *momapy* generalises this distinction and applies it to the Systems Biology Graphical Notation (SBGN) [18] and CellDesigner™ [5], two of the main standard languages to represent maps graphically.

 doi:10.20944/preprints202601.0774.v1

2. A *momapy* Map: A Model, a Layout, and a Layout-Model Mapping

In *momapy*, a *map* is made up of three distinct elements: a *model*; a *layout*; and a *layout-model mapping*. The *model* of a map is a structured collection of model elements that encode the biological concepts represented in the map. *momapy* defines a hierarchical data model for each type of map it supports (SBGN PD [19], SBGN AF [20], and CellDesigner™ [5], see section 3.1 for more details), formed of class and subclass relations encoding the biological concepts based on the map type and their hierarchy in the Systems Biology Ontology [21]. The encoded biological concepts include entity pools, processes and modulations for process description types of maps (SBGN PD, CellDesigner™), or activities and modulations for activity types (SBGN AF). The *layout* of a map is a structured collection of layout elements encoding the glyphs (graphical symbols) of the map. Each layout element encodes a specific shape (e.g., a circle, a rectangle with rounded corners) whose attributes define how it may be rendered (e.g., its position, dimensions, fill color). A layout element may itself contain other layout elements, giving a tree-like structure to the layout, similar to DOM-based documents such as HTML or SVG. Finally, the *layout-model mapping* associates each layout element of the map with the model element it represents.

3. *momapy*'s Features

At its core, *momapy* allows users to extract a map's model and layout, and work with them programmatically through a variety of features. These are summarised in Figure 1A, and detailed below.

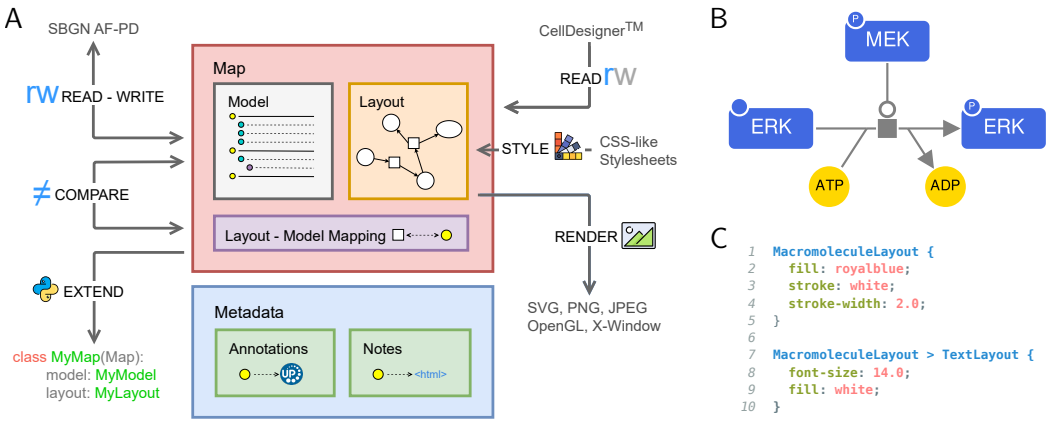

Figure 1. Overview of *momapy*'s features. A. A graphical summary of *momapy*'s features, showing how a map is formed of a model, a layout, and a layout-model mapping. **B**. An example of an SBGN PD map styled and rendered using *momapy*. **C**. An excerpt of the CSS-like style sheet used to style the map represented in B.

3.1. Supported Map Formats

momapy currently supports SBGN (PD and AF) and CellDesigner™, two of the main languages used to draw molecular maps. SBGN maps may be read and written from/to SBGN-ML, and CellDesigner™ maps read from the CellDesigner™ exchange format (based on SBML).

3.2. Exploring and Manipulating Maps

momapy elements (maps, models, layouts, and their sub-elements) are made readily available as Python objects, that can be easily explored and manipulated programmatically. Notes and annotations (as defined by SBML) are made accessible outside of the map they describe, complying with the FAIR principles on the separation of metadata from data [22].

3.3. Comparing Maps

momapy elements are implemented in a way they can be easily compared programmatically. This implementation also allows users to check whether a *momapy* element belongs to a given set of objects

 doi:10.20944/preprints202601.0774.v1

efficiently, which is crucial for some applications where one needs to compare large sets of maps (e.g., sets of SBGN bricks instances [23]).

3.4. Styling and Style Sheets

Using *momapy*, the graphical style of layout elements may be extensively customised. *momapy* supports the modification of common presentation attributes such as stroke and fill colors or line width, but also the application of advanced graphical effects such as shadows. These custom styles may be applied directly to the individual layout element objects, or using user-defined CSS-like style sheets (see Figure 1B, C). *momapy* also includes a set of predefined style sheets, that mimic the graphical style of several common map editors such as Newt [7], SBGN-ED [6], or CellDesigner™ [5].

3.5. Rendering of Maps

momapy may be used to render layouts of maps to images (e.g., SVG, PNG, JPEG, PDF files) or directly to the screen (e.g., OpenGL windows). The rendering is done offline using different alternative common backends (SVG-native, Skia (https://skia.org/), Cairo (https://www.cairographics.org/)), enabling the embedding of *momapy* in other applications on all platforms. Rendering with *momapy* supports all the aforementioned styles, including advanced graphical effects (depending on the backend).

3.6. Extending momapy *with New Formats and Functionalities*

momapy offers a set of abstract and concrete classes that may be easily extended for the fast development of new data models and glyphs. An example of such an extension is *momapy-bel* (available at https://github.com/adrienrougny/momapy_bel), which adds support for BEL models [24] to *momapy*. Since *momapy* is built as a library, it can be freely used by third-party tools to work with molecular maps. Examples of such tools include *momapy-kb* (available at https://github.com/adrienrougny/momapy_kb), which allows users to integrate molecular maps into a graph database, and *momapy-draw* (available at https://github.com/adrienrougny/momapy_draw), a tool based on *momapy*'s rendering capabilities to draw SBGN and CellDesigner™ maps programmatically.

4. Software Implementation and Availability

momapy is implemented in Python (RRID:SCR_008394). Its code is freely available from https://github.com/adrienrougny/momapy under a GPLv3 license. Complete documentation and a user manual are available at https://adrienrougny.github.io/momapy.

Author Contributions: Conceptualisation: A.R, M.O, V.S. Software: A.R. Writing - original draft: A.R. Writing - review & editing: M.O., V.S. Project administration: M.O., V.S. Funding acquisition: M.O., V.S. All authors have read and agreed to the final version of the manuscript.

Funding: This work was supported by the COMMUTE project, funded by the European Union under the grant agreement number 101136957.

Acknowledgments: This work was supported by the COMMUTE project. The COMMUTE project is funded by the European Union under the grant agreement number 101136957. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them. The authors would also like to thank the coaches of the PPC team at LCSB for checking the manuscript before submission.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

 doi:10.20944/preprints202601.0774.v1

SBGN Systems Biology Graphical Notation
SBML Systems Biology Markup Language
BEL Biological Expression Language

References

1. Ostaszewski, M.; Gebel, S.; Kuperstein, I.; Mazein, A.; Zinovyev, A.; Dogrusoz, U.; Hasenauer, J.; Fleming, R.M.T.; Le Novère, N.; Gawron, P.; et al. Community-driven roadmap for integrated disease maps. *Briefings in Bioinformatics* **2019**, *20*, 659–670. https://doi.org/10.1093/bib/bby024.
2. Oda, K.; Matsuoka, Y.; Funahashi, A.; Kitano, H. A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology* **2005**, *1*, 2005.0010. Publisher: John Wiley & Sons, Ltd, https://doi.org/10.1038/msb4100014.
3. Calzone, L.; Gelay, A.; Zinovyev, A.; Radvanyi, F.; Barillot, E. A comprehensive modular map of molecular interactions in RB/E2F pathway. *Molecular Systems Biology* **2008**, *4*. Publisher: EMBO Press.
4. Niarakis, A.; Ostaszewski, M.; Mazein, A.; Kuperstein, I.; Kutmon, M.; Gillespie, M.E.; Funahashi, A.; Acencio, M.L.; Hemedan, A.; Aichem, M.; et al. Drug-target identification in COVID-19 disease mechanisms using computational systems biology approaches. *Frontiers in Immunology* **2024**, *14*. Publisher: Frontiers, https://doi.org/10.3389/fimmu.2023.1282859.
5. Funahashi, A.; Matsuoka, Y.; Jouraku, A.; Morohashi, M.; Kikuchi, N.; Kitano, H. CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proceedings of the IEEE* **2008**, *96*, 1254–1265. Publisher: IEEE.
6. Czauderna, T.; Klukas, C.; Schreiber, F. Editing, validating and translating of SBGN maps. *Bioinformatics* **2010**, *26*, 2340–2341. https://doi.org/10.1093/bioinformatics/btq407.
7. Balci, H.; Siper, M.C.; Saleh, N.; Safarli, I.; Roy, L.; Kilicarslan, M.; Ozaydin, R.; Mazein, A.; Auffray, C.; Babur, Ã.; et al. Newt: a comprehensive web-based tool for viewing, constructing and analyzing biological maps. *Bioinformatics* **2021**, *37*, 1475–1477. https://doi.org/10.1093/bioinformatics/btaa850.
8. Gawron, P.; Ostaszewski, M.; Satagopam, V.; Gebel, S.; Mazein, A.; Kuzma, M.; Zorzan, S.; McGee, F.; Otjacques, B.; Balling, R.; et al. MINERVA: a platform for visualization and curation of molecular interaction networks. *npj Systems Biology and Applications* **2016**, *2*, 16020. Publisher: Nature Publishing Group, https://doi.org/10.1038/npjsba.2016.20.
9. Xu, J.; Jiang, J.; Sauro, H.M. SBMLDiagrams: a python package to process and visualize SBML layout and render. *Bioinformatics* **2023**, *39*, btac730. https://doi.org/10.1093/bioinformatics/btac730.
10. van Iersel, M.P.; Villéger, A.C.; Czauderna, T.; Boyd, S.E.; Bergmann, F.T.; Luna, A.; Demir, E.; Sorokin, A.; Dogrusoz, U.; Matsuoka, Y.; et al. Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics* **2012**, *28*, 2016–2021. https://doi.org/10.1093/bioinformatics/bts270.
11. Balaur, I.; Roy, L.; Mazein, A.; Karaca, S.G.; Dogrusoz, U.; Barillot, E.; Zinovyev, A. cd2sbgnml: bidirectional conversion between CellDesigner and SBGN formats. *Bioinformatics* **2020**, *36*, 2620–2622. Publisher: Oxford University Press.
12. Rougny, A.; Balaur, I.; Luna, A.; Mazein, A. StonPy: a tool to parse and query collections of SBGN maps in a graph database. *Bioinformatics* **2023**, *39*, btad100. https://doi.org/10.1093/bioinformatics/btad100.
13. Rougny, A.; Froidevaux, C.; Calzone, L.; Paulevé, L. Qualitative dynamics semantics for SBGN process description. *BMC Systems Biology* **2016**, *10*, 42. https://doi.org/10.1186/s12918-016-0285-0.
14. Aghamiri, S.S.; Singh, V.; Naldi, A.; Helikar, T.; Soliman, S.; Niarakis, A. Automated inference of Boolean models from molecular interaction maps using CaSQ. *Bioinformatics* **2020**, *36*, 4473–4482. Publisher: Oxford University Press (OUP), https://doi.org/10.1093/bioinformatics/btaa484.
15. Hucka, M.; Finney, A.; Sauro, H.M.; Bolouri, H.; Doyle, J.C.; Kitano, H.; Arkin, A.P.; Bornstein, B.J.; Bray, D.; Cornish-Bowden, A.; et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **2003**, *19*, 524–531. Publisher: Oxford Univ Press.
16. Gauges, R.; Rost, U.; Sahle, S.; Wengler, K.; Bergmann, F.T. The Systems Biology Markup Language (SBML) Level 3 Package: Layout, Version 1 Core. *Journal of Integrative Bioinformatics* **2015**, *12*, 550–602. Publisher: De Gruyter, https://doi.org/10.1515/jib-2015-267.
17. Bergmann, F.T.; Keating, S.M.; Gauges, R.; Sahle, S.; Wengler, K. SBML Level 3 package: Render, Version 1, Release 1. *Journal of Integrative Bioinformatics* **2018**, *15*. Publisher: De Gruyter, https://doi.org/10.1515/jib-2017-0078.
18. Le Novère, N.; Hucka, M.; Mi, H.; Moodie, S.; Schreiber, F.; Sorokin, A.; Demir, E.; Wegner, K.; Aladjem, M.I.; Wimalaratne, S.M.; et al. The systems biology graphical notation. *Nature Biotechnology* **2009**, *27*, 735–741. Publisher: Nature Publishing Group.

 doi:10.20944/preprints202601.0774.v1

19. Rougny, A.; Touré, V.; Moodie, S.; Balaur, I.; Czauderna, T.; Borlinghaus, H.; Dogrusoz, U.; Mazein, A.; Dräger, A.; Blinov, M.L.; et al. Systems Biology Graphical Notation: Process Description language Level 1 Version 2.0. *Journal of Integrative Bioinformatics* **2019**, *16*. Publisher: De Gruyter, https://doi.org/10.1515/jib-2019-0022.
20. Mi, H.; Schreiber, F.; Moodie, S.; Czauderna, T.; Demir, E.; Haw, R.; Luna, A.; Le Novère, N.; Sorokin, A.; Villéger, A. Systems biology graphical notation: activity flow language level 1 version 1.2. *Journal of integrative bioinformatics* **2015**, *12*, 340–381. Publisher: De Gruyter.
21. Juty, N.; le Novère, N. Systems biology ontology. *Encyclopedia of Systems Biology* **2013**, pp. 2063–2063. Publisher: Springer.
22. Balaur, I.; Welter, D.; Rougny, A.; Inau, E.T.; Mazein, A.; Ghosh, S.; Schneider, R.; Waltemath, D.; Ostaszewski, M.; Satagopam, V. FAIR assessment of Disease Maps fosters open science and scientific crowdsourcing in systems biomedicine. *Scientific Data* **2025**, *12*, 851. Publisher: Nature Publishing Group, https://doi.org/10.1038/s41597-025-05147-w.
23. Rougny, A.; Touré, V.; Albanese, J.; Waltemath, D.; Shirshov, D.; Sorokin, A.; Bader, G.D.; Blinov, M.L.; Mazein, A. SBGN Bricks Ontology as a tool to describe recurring concepts in molecular networks. *Briefings in Bioinformatics* **2021**, *22*, bbab049. https://doi.org/10.1093/bib/bbab049.
24. Hoyt, C.T.; Domingo-Fernández, D.; Hofmann-Apitius, M. BEL Commons: an environment for exploration and analysis of networks encoded in Biological Expression Language. *Database* **2018**, *2018*, bay126. https://doi.org/10.1093/database/bay126.