CENTER FOR REPRODUCIBLE BIOMEDICAL MODELING

Biannual Newsletter VOL.3 JULY 2024

Outreach, Dissemination and Education

EDITED BY MCHAEL KOCHEN

Outreach

Outreach is crucial to the promotion of reproducibility and reusability to the biomedical modeling community. To that end, the center engages in a number of endeavors to 1) disseminate standards, tools and methods for the proper construction, simulation, and annotation of biomedical models, and 2) educate members of the community on the use of those resources.

Dissemination

Members of the center routinely develop and publish innovative software tools and methodologies for biomedical modeling. Perspective pieces on reproducibility and reusability are often published as well. In addition to such traditional academic routes, the center takes a number of additional steps to more broadly promote reproducibility in modeling. One of those steps is the very newsletter you are now reading. Other steps include the organization of meetings to discuss and guide the development of future computational modeling standards, and the hosting of seminars featuring speakers on a wide range of topics. These topics include, but are not limited to, tools and methods, standards and best practices, current challenges in modeling, and expert modeling of specific systems. Links to these seminars can be found on the center website and are highly recomended.

Data Use in Systems Biology: FAIR Principles and Applications

The seminar series held in the Summer of 2024 features two speakers on reproducibility and data use in systems biology. On June 28 Pedro Mendez, the director of the Center for Cell Analysis and Modeling at UConn Health, discusses the adoption of FAIR principles (findable, accessible, interoperable, and reusable) by the modeling community in a seminar entitled “Reproducibility and FAIR Principles: The Case of a Segment Polarity Network Model”. And on July 12 Paul Jonas Jost, a software Developer and Ph.D. student at Universität Bonn, discusses model calibration in a seminar entitled “pyPESTO: a modular and scalable tool for parameter estimation for dynamic models”.

Education

The seminar series work well to advance specific topics in modeling. For the more general audience, including those new to modeling, the center engages in online tutorials and in workshops that include a yearly virtual Summer school. The tutorials often cover moderately advanced topics such as Bayesian inference, spatial modeling, metabolic control analysis, and parameter fitting routines, but also provides basic instruction on highly popular modeling tools such as Tellurium and Antimony. Links to these tutorials can also be found on the center website.

Outreach, Dissemination and Education (continued)

2024 Cell Modeling Online Summer School

The virtual summer school has been conducted in some form every Summer since 2020. The course covers the basics of python programming, principles of kinetic mechanistic modeling, the SBML standard and the BioModels database, model calibration and annotation, and the modeling of various complex systems.

The 2024 edition of the course runs from July 22nd to July 26th and includes three days of intensive instruction conducted by Herbert Sauro, Lucian Smith, Joe Hellerstein, and John Gennari.

At the end of the course will be a day dedicated to a Hackathon on model credibility, and a day featuring a series of invited speakers.

2024 Hackathon

This year's Summer school hackathon will focus on the credibility of published biomedical models. Participants will use tools produced by the Sauro lab to assess properties of these models such as reproducibility, understandability, verification, and parameter sensitivity. Additionally, they will annotate models using the Antimony web editor. More on the Summer school and hackathon in the official announcements below.

Announcements

ANNUAL CELL MODELING ONLINE SUMMER SCHOOL AND HACKATHON

We are pleased to announce the 2024 Annual Cell Modeling Online Summer School and Hackathon, will take place Monday, July 22nd, through Friday, July 26th, 2024. The first three days will cover basic modeling techniques using Tellurium and Roadrunner. Day four will consist of a hackathon on model credibility and the last day is reserved for special talks. For more information and to register, go to: https://reproduciblebiomodels.org/dissemination-education/workshops/

COMPUTATIONAL MODELING IN BIOLOGY NETWORK (COMBINE) MEETING

The CRBM is active in developing and maintaining community standards. Two members of CRBM (Michael Blinov from UConn Health and Lucian Smith from U Washington) are elected SBML editors.

The BioSimulations platform and related software tools COPASI, VCell and Vivarium are regularly discussed during COMBINE events. HARMONY 2023 was hosted by Herbert Sauro and the Center for Reproducible Biomedical Modeling at the University of Washington, Seattle, WA. COMBINE 2023 was hosted by Michael Blinov and Ion Moraru at the University of Connecticut Health Center, Farmington, CT. Recently, the CRBM members Michael Blinov (UConn Health), Ion Moraru (UConn Health), and Lucian Smith (UW) attended the HARMONY workshop in London (April 8-11, 2024; https://co.mbine.org/author/harmony-2024/) and presented tools developed by the CRBM. The next COMBINE meeting will take place in Stuttgart, Germany (September 1-5, 2024). Herbert Sauro and Lucian Smith are invited to give talks and chair sessions.

Announcements (continued)

COMPUTATIONAL CELL BIOLOGY WORKSHOP

The Center for Cell Analysis and Modeling (CCAM) at the University of Connecticut Health Center (UConn Health) develops and maintains several tools actively used in the BioSimulations platform: COPASI (http://copasi.org), VCell (http://vcell.org), and Vivarium (https://vivariumlab.com/). CCAM organizes annual Computational Cell Biology workshops (https://compcellbio.org/ccbworkshop) that help biologists in developing their modeling projects. The 25th Winter workshop took place online February 26-28th and had more than 40 participants. Eleven participants developed their projects while working one-on-one with software experts. The next workshop will take place July 22-24th in Farmington, CT, and last-minute applications can be considered by emailing Michael Blinov at blinov@uchc.edu.

AMERICAN SOCIETY FOR CELL BIOLOGY'S (ASCB) CELL BIO MEETING

At the American Society for Cell Biology's (ASCB) Cell Bio 2023 meeting (December 2-6, 2023, Boston, MA) CRBM member Michael Blinov and Margaret Johnson from John Hopkins University inaugurated the Special Interests Subgroup on Biophysical Modeling of the Cell. The talks described mathematical models, including those done with COPASI and VCell. The Special Interests Subgroup was a great success and will be repeated at the American Society for Cell Biology's (ASCB) Cell Bio 2024 meeting, December 14-18 in San Diego, CA.

Highlights and Updates

VIVARIUM

BY ERAN AGMON

Vivarium [1] is an innovative software platform designed to facilitate the development of integrative multiscale models in computational biology. It addresses the complexity of biological systems by allowing computational biologists to define mechanistic models, combine them with existing models, and execute them together seamlessly. This software provides a modular interface that decouples specific data types and modeling methods, supporting flexible and scalable integration. Vivarium's utility spans from sub-cellular biophysical simulations [2], to whole-cell models of E. coli [3], bacterial colonies [3], multicell models of tumors and T cells [4], and recently to ecological models of marine microbiomes. By enabling the integration of diverse modeling strategies into unified representations, Vivarium streamlines the labor-intensive process of building such models, fostering a more collaborative and interconnected computational biology landscape.

For the renewal of the Center for Reproducible Biomedical Modeling, Vivarium will be extended to enhance its capabilities further. The initial development cycle saw the creation of BioSimulations, a software ecosystem for reproducible execution and sharing of biomodels using an integrated registry of diverse simulators under a standardized API.

The next phase aims to leverage this foundation to engineer flexible, large-scale, and multi-modal computational approaches for in silico studies. These advancements will facilitate knowledge transfer between simulation domains, enabling the reuse of existing

Highlights and (continued)

simulators within a larger framework. This will be achieved through standardized composition protocols, containerized processes, and modular development, supporting multi-algorithmic and multi-scale simulations. The planned outcomes include infrastructure for the multi-scale simulation community, enhanced methods for composing simulators, and a robust framework for collaborative, reproducible, and scalable computational biology research.

1. Agmon,E. et al. (2022) Vivarium: an interface and engine for integrative multiscale modeling in computational biology. Bioinformatics, 38, 1972–1979.
2. Vegesna,K.R. et al. (2024) Comparing spatial biophysical simulations across scales and methods. Biophysical Journal, 123, 130a–131a.
3. Skalnik,C.J. et al. (2023) Whole-cell modeling of E. coli colonies enables quantification of single-cell heterogeneity in antibiotic responses. PLOS Computational Biology, 19, e1011232.
4. Hickey,J.W. et al. (2024) Integrating multiplexed imaging and multiscale modeling identifies tumor phenotype conversion as a critical component of therapeutic T cell efficacy. cels, 15, 322-338.e5.

BIOMODELS SED-ML UPDATE

The BioModels database contains 1057 curated literature derived models. These models are reconstructed from publications and put into a standard SBML file if such a file does not already exist. Results from the paper, such as select figures, are then reproduced and notes regarding the curation process are submitted. However, faithful and thorough reproduction of model simulation results requires extensive documentation of computational experimental design that is absent from a majority of the entries in the BioModels database. Recent work has begun to address this issue.

The Simulation Experiment Description Markup Language (SED-ML) provides a standard for in-silico experimental design. Most of the entries in the BioModels database contain no such file and for many of those that do the file is insufficient for general reproducibility. Thus the Center for Reproducible Biomedical Modeling has teamed with the European Bioinformatics Institute (EBI) to address this deficiency.

The goal of this work is to update or implement SED-ML files for every curated model in the BioModels database and use them to demonstrate reproducibility on multiple biosimulation platforms. Existing SED-ML files, and SED-ML files produced from an existing COPASI save file, were carefully examined for correctness and reproducibility. Numerous errors were found and corrected. For those models with neither SED-ML or COPASI save files a template SED-ML file describing a simple time-course experiment was created. While this template SED_ML file does not reproduce results from the associated papers it does enable comparison across biosimulator platforms.

SED-ML wrappers were created for five well known biosimulators: COPASI, VCell, Tellurium, pySCeS, and Amici. These wrappers, residing on biosimulators.org, translate the experiment instructions from a SED-ML file into API calls for each biosimulator and enable a comparison of results across platforms. They include support for time course and steady state analyses. As a by-product of wrapper development, numerous updates to the SED-ML specification were made leading to the release of SED-ML Level 1 Version 4 in September of 2021. 1035 of the 1057 curated models were able to run on Tellurium and 828 (531 models with full SED-ML files and 297 with template SED-ML) were capable of running on at least two biosimulators. Work is ongoing to further improve reproducibility across platforms.

Highlights and Updates (continued)

SBMLNETWORK AND ALCUIN

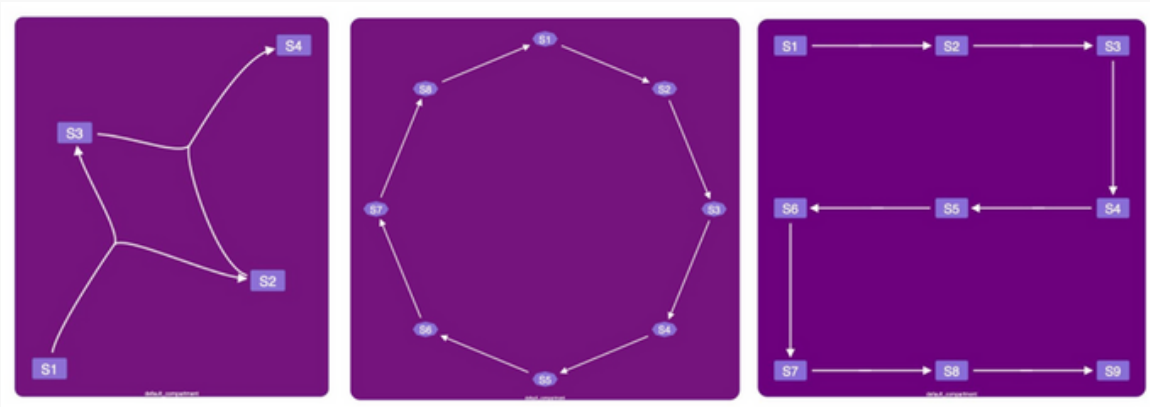
BY ADEL HEYDARABADIPOUR

The visualization of computational models of biological processes is an important aspect of systems biology research. It provides the researchers with intuitive maps that elucidate the intricate interactions and dynamics within biological models. By visually depicting these interactions, systems biologists can observe patterns, analyze system dynamics, and simulate hypothetical scenarios to predict biological behaviors. Beyond aiding in comprehension, it can significantly facilitate collaboration by enabling researchers to effectively communicate their findings, insights, and innovations with others.

Level 3 of Systems Biology Markup Language (SBML), the de facto standard to reinforce the exchange and reproducibility of biological models, is introduced with Layout and Render packages that enable storing graphical representations of biological models alongside their mathematical descriptions. The Layout extension describes the positions and dimensions of graphical objects representing the biological model and the Render extension specifies their styles and how they appear on the canvas.

Sauro Lab at University of Washington has a rich background in utilizing the functionalities of SBML Layout and Render extensions to create software tools that aid the systems biology community in visualizing computational models. Among them are PathwayDesigner, SBMLSupportLayout[1], SBWAutoLayout[1], and SBMLDiagrams[2]. Leveraging our past achievements, we are currently engaged in the development of the following projects:

SBMLNetwork: A library designed to enable software developers and systems biologists to interact with the graphical representation of SBML models. It makes use of SBML Layout and Render extensions and provides the following features: (i) a built-in high-performance autolayout algorithm that automatically generates and adds graphical representation data to SBML models (ii) a robust API that provides the users and developers with seamless access to the graphical representation attributes of SBML models (iii) a drawing tool to render an image of the graphical representation of SBML models using their Layout and Render data. SBMLNetwork is distributed as both a python package, which can be installed using the 'pip install sbmlnetwork' command, and as a shared library, which can be downloaded from https://github.com/adelhpour/SBMLNetwork/releases.



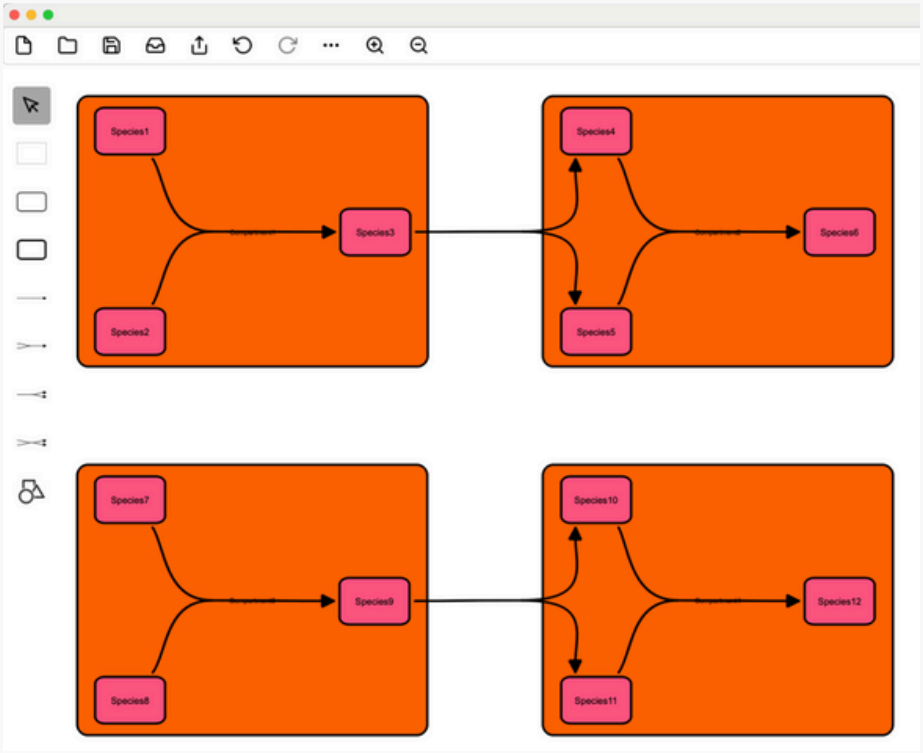
Rendered images of SBML models using SBMLNetwork (left) the combination of a uni-bi reaction and a bi-uni reaction (middle) a series of reactions aligned in a circular pattern (right) a series of reactions aligned on a grid.

Highlights and Updates (continued)

Alcuin: An open-source cross-platform Qt/C++ portable library that provides developers with an embeddable widget with the required features for a network viewer/editor software tool. With minimal effort, a developer can embed this widget into their own software tool and provide their users with a user interface to view, build, and edit a network consisting of nodes and edges. Alcuin is shipped with the fundamental features of a network viewer/editor tool, including adding/removing nodes to its graphics scene, adding/removing edges (with optional arrowheads) between nodes, nesting one node into another node, modifying the properties of network elements (nodes, edges, and arrowheads), selecting, moving, copying, cutting, pasting, aligning, dragging and dropping network elements, zooming and panning on its graphics scene, and undoing and redoing the user's actions. In addition, Alcuin is equipped with a plugin interface that lets developers use Python scripts to customize it to meet specific user requirements.

As a use case, we show how its plugin interface can be implemented to customize this widget into a tool for viewing, building, and editing biochemical reaction networks formatted using SBML Layout and Render packages. Using this customized tool, a user can create, load, save or edit biochemical reaction networks. Alcuin is distributed in the form of a shared library and an executable app that can be downloaded from https://github.com/adelhpour/Alcuin/releases. The name Alcuin refers to the 8th century English scholar and teacher who was headmaster in 778 AD at the cathedral school of York, England and later became master of the palace school of Charlemagne at Aachen in 782 AD.

1. Deckard A, Bergmann FT, Sauro HM. Supporting the SBML layout extension. Bioinformatics. 2006 Dec 1;22(23):2966-7.
2. Xu J, Jiang J, Sauro HM. SBMLDiagrams: a python package to process and visualize SBML layout and render. Bioinformatics. 2023 Jan 1;39(1):btac730.



A screenshot of the Alcuin window. A biological model is created using Alcuin drawing tools.

Quick Tutorials

BNGLVIZ VISUALIZATION OF RULE-BASED MODELS

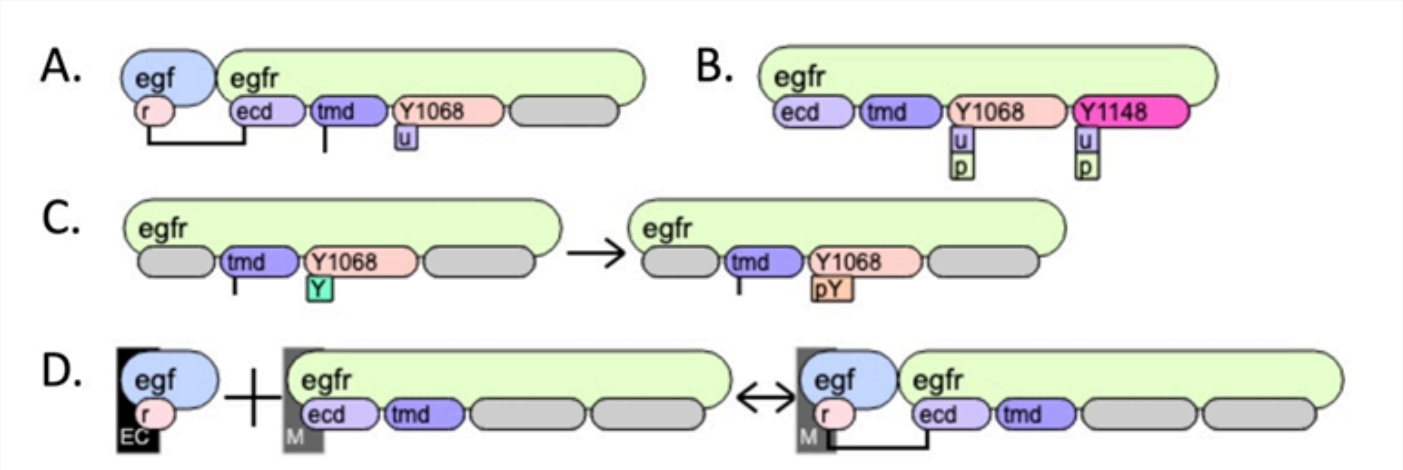
BY MICHAEL BLINOV

The rule-based modeling approach provides a compact description of molecules, molecular interactions, and their effects in the form of templates for possible chemical species and reactions. These templates are based on the description of biomolecules as containers that have multiple sites (e.g., molecular binding sites such as SH2 domains or tyrosine residues), their states (such as phosphorylated on unphosphorylated states of a tyrosine), and connectivity of molecules through explicit binding among molecular binding sites. Thus, the rule-based modeling approach enables the representation of intricate details in biochemical processes and provides high-level expressiveness in describing molecular interactions and reactions. The actual chemical species and reactions are generated either deterministically, using BioNetGen simulation engine (http://bionetgen.org), or stochastically, using agent-based modeling with NFSim (Network-Free stochastic simulator).

The web tool bnglViz enables online graphical visualization of rule-based models. Any BNGL model can be loaded and fully visualized in a colorful cartoon-style way, following graphical notations introduced within a popular modeling and simulation framework Virtual Cell (http://vcell.org). The generated visualizations can be used as supplemental figures in publications or as a way to annotate BNGL models on web repositories. More than 30 published models implemented in BioNetGen are available on our website at https://bnglviz.github.io/examples.html as both BNGL code and bnglViz-generated visualization

A model encoded in BNGL (such as taken from publication's supplemental material or designed in VCell) can be loaded at the website, and then all molecules, species, reaction rules, and observables will be displayed on the same page.

Liguori-Bills, N., & Blinov, M. L. (2024). bnglViz: Online visualization of rule-based models. Bioinformatics, 40(6), btae351; https://bnglviz.github.io/



A. Description of a molecular template. Two molecules "egf" (epidermal growth factor) and "egfr" (for epidermal growth factor receptor) are connected at the two sites "r" (receptor-bunding domain) and "ecd" (extra-cellular domain) respectively. The transmembrane domain ("tmd" site) of "egfr" must be bound to some other molecule, indicated by a vertical line. "Y1068" site of "egfr" is unphosphorylated (denoted by a square with a letter "u") and unbound, while the state of "Y1148" is not defined and this site may be bound or unbound.

B. The full description of an "egfr" molecule. It has four sites: extracellular "ecd", transmembrane "tmd" and two tyrosine "Y1068" and "Y1148". Each of the two tyrosine can be in two possible states: unphosphorylated ("u") and phosphorylated ("p").

Quick Tutorials (continued)

C. The simple rule of phosphorylation of “Y1069” residue of “egfr” receptor by another receptor bound to it. The trans-membrane domain of egfr (“tmd” site) must be bound for the state of “Y1068” to change from unphosphorylated (“u”) to phosphorylated (“p”). This rule corresponds to a potentially infinite number of individual interactions among molecular complexes, because other sites (shown in gray) may be bound to other molecules not shown in the rule definition. However, all interactions corresponding to the single rule are parameterized by the same on- and off-rate constants. Note that rules are not limited to binary site-site interactions but can be extended to encode arbitrary levels of complexities for interaction among multivalent molecules; for example, the binding strength of one molecular site may depend on the binding status of other sites of the same molecule to capture allosteric or cooperativity effects.

D. Adding locations to rule-based description.“egf” ligand is located in a volumetric extracellular compartment (“EC”), while “egfr” receptor is located on a membrane “M”. The resulting complex is located on a membrane “M". For the reaction to proceed, the trans-membrane domain (“tmd”) must be unbound.

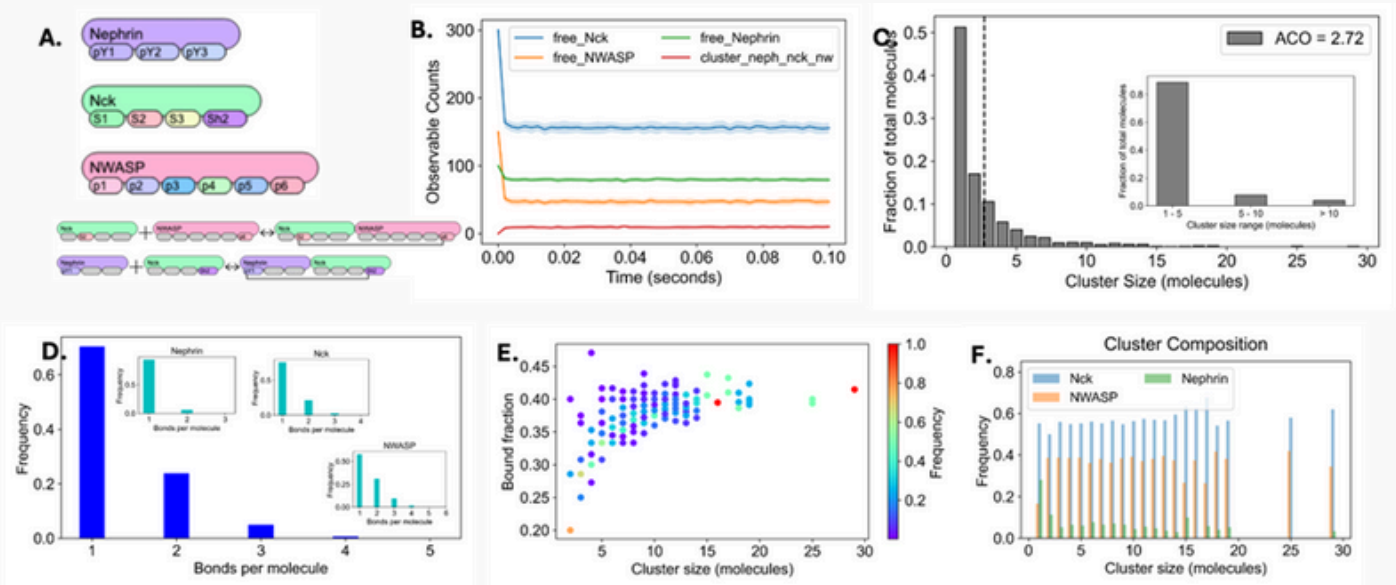
MOLCLUSTPY: CHARACTERIZATION OF MULTIVALENT BIOMOLECULAR CLUSTERS

BY MICHAEL BLINOV

Chattaraj, A., Nalagandla, I., Loew, L. M., & Blinov, M. L. (2023). MolClustPy: a Python package to characterize multivalent biomolecular clusters. Bioinformatics, 39(6), btad385. https://molclustpy.github.io/

Rule-based modeling allows the simulation of large multimolecular assemblies (clusters).

When the affinities of the individual molecular interactions are relatively weak, multivalent clusters maintain their integrity but allow various molecular compositions, so multiple simulation runs are required to determine the average behavior of such bimolecular system. MolClustPy is a Python package to perform multiple stochastic simulation runs using NFsim (Network-Free stochastic simulator) and characterize distribution of cluster sizes, molecular composition, and bonds across molecular clusters and individual molecules of different types. NFsim is a non-spatial simulator, so it does not account for excluded volume and non-physical crosslinking when generating molecular complexes.



Quick Tutorials (continued)

Interactions between Nephrin, Nck, and NWASP molecules can be visualized using bnglViz https://bnglviz.github.io/examples/Chattaraj_2021.html (Figure 1A). To stochastically simulate rule-based systems with a potentially infinite number of species and interactions, NolClustPy uses NFsim

```
simObj = BNG_multiTrials(bng_file, t_end=0.02, steps=20, numRuns=20)
```

NFsim simulation outputs consist of two major parts—observables and final molecular species. Observables are predefined global properties of the biological system whose concentrations are reported as a time course during the simulation, for example, concentrations of free molecules of a certain type. The final molecular species encompass the set of molecular complexes that exist at the end of each simulation run.

MolClustPy analyzes both observables and molecular clusters across multiple simulation runs.

```
nfsObj.process_gdatfiles()
nfsObj.process_speciesfiles(molecules, counts, numSite)
```

MolClustPy can plot mean trajectory and standard deviation (as a fluctuation envelope) time courses for multiple observables (Figure 1B):

```
plotTimeCourse(outpath, obsList=[2,4,6])
```

The cluster size distribution can be plotted where each bar corresponds to the fraction of total molecules in each cluster size (Figure 1C):

```
plotClusterDist(outpath)
```

For a large cluster size range, a binned histogram might be helpful, as shown in the inset.

```
plotClusterDist(outpath, sizeRange=[1,5,15])
```

The molecular crosslinking (the average number of bonds per molecule) can be plotted in Figure 1D:

```
plotBondsPerMolecule(outpath)
```

Inspecting bond distribution of individual molecular types (Nick, inset) informs on the degree of bound saturation for a given affinity and stoichiometry:

```
plotBondCounts(outpath, molecules=['Nck'])
```

Figure 1E demonstrates bound fraction (the ratio of bound sites over total sites present in that cluster). The color bar shows the relative frequencies of a given configuration.

```
plotBoundFraction(outpath)
```

Figure 1F demonstrates the molecular composition of the clusters, giving the relative fraction of each molecular type within a given cluster size:

```
plotClusterComposition(outpath, specialClusters=[2, 4, 10], width=0.15, alpha=0.7)
```

Publications

Xu, J., Geng, G., Nguyen, N. D., Perena-Cortes, C., Samuels, C., and Sauro, H. M. (2023) SBcoyote: An extensible Python-based reaction editor and viewer. Biosystems 232, 105001.

Xu, J. (2023) SBMLKinetics: a tool for annotation-independent classification of reaction kinetics for SBML models. BMC Bioinformatics 24, 1–16.

Mendes, P. (2023) Reproducibility and FAIR principles: the case of a segment polarity network model. Front. Cell Dev. Biol. 11.

Bodeit, O., Ben Samir, I., Karr, J. R., Goelzer, A., and Liebermeister, W. (2023) RBAtools: a programming interface for Resource Balance Analysis models. Bioinformatics Advances 3, vbad056.

Porubsky, V. L., and Sauro, H. M. (2023) A Practical Guide to Reproducible Modeling for Biochemical Networks, in Computational Modeling of Signaling Networks (Nguyen, L. K., Ed.), pp 107–138. Springer US, New York, NY.

Shin, J., Porubsky, V., Carothers, J., and Sauro, H. M. (2023) Standards, dissemination, and best practices in systems biology. Current Opinion in Biotechnology 81, 102922.

Tatka, L. T., Smith, L. P., Hellerstein, J. L., and Sauro, H. M. (2023, January 14) Adapting Modeling and Simulation Credibility Standards to Computational Systems Biology. arXiv.

Welsh, C., Xu, J., Smith, L., König, M., Choi, K., and Sauro, H. M. (2023) libRoadRunner 2.0: a high performance SBML simulation and analysis library. Bioinformatics 39, btac770.

Xu, J., Jiang, J., and Sauro, H. M. (2023) SBMLDiagrams: a python package to process and visualize SBML layout and render. Bioinformatics 39, btac730.

Kochen, M. A., Wiley, H. S., Feng, S., and Sauro, H. M. (2022) SBbadger: biochemical reaction networks with definable degree distributions. Bioinformatics 38, 5064–5072.

Shaikh, B., Smith, L. P., Vasilescu, D., Marupilla, G., Wilson, M., Agmon, E., Agnew, H., Andrews, S. S., Anwar, A., Beber, M. E., Bergmann, F. T., Brooks, D., Brusch, L., Calzone, L., Choi, K., Cooper, J., Detloff, J., Drawert, B., Dumontier, M., Ermentrout, G. B., Faeder, J. R., Freiburger, A. P., Fröhlich, F., Funahashi, A., Garny, A., Gennari, J. H., Gleeson, P., Goelzer, A., Haiman, Z., Hasenauer, J., Hellerstein, J. L., Hermjakob, H., Hoops, S., Ison, J. C., Jahn, D., Jakubowski, H. V., Jordan, R., Kalaš, M., König, M., Liebermeister, W., Sheriff, R. S. M., Mandal, S., McDougal, R., Medley, J. K., Mendes, P., Müller, R., Myers, C. J., Naldi, A., Nguyen, T. V. N., Nickerson, D. P., Olivier, B. G., Patoliya, D., Paulevé, L., Petzold, L. R., Priya, A., Rampadarath, A. K., Rohwer, J. M., Saglam, A. S., Singh, D., Sinha, A., Snoep, J., Sorby, H., Spangler, R., Starruß, J., Thomas, P. J., van Niekerk, D., Weindl, D., Zhang, F., Zhukova, A., Goldberg, A. P., Schaff, J. C., Blinov, M. L., Sauro, H. M., Moraru, I. I., and Karr, J. R. (2022) BioSimulators: a central registry of simulation engines and services for recommending specific tools. Nucleic Acids Research 50, W108–W114.