

# 1 DuneCopasi: A multi-compartment reaction-diffusion 2 simulator for systems biology

3 **Santiago Ospina De Los Ríos** <sup>1,2</sup>, **Peter Bastian**<sup>1</sup>, **Liam Keegan** <sup>3</sup>, **Sven**  
4 **Sahle** <sup>4</sup>, **Dylan Vermoortele** <sup>6</sup>, and **Lilija Wehling** <sup>4,5</sup>

5 **1** Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Germany **2** Heidelberg  
6 Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp),  
7 Heidelberg University, Germany **3** Scientific Software Center (SSC), Heidelberg University, Germany **4**  
8 BioQuant, Centre for Organismal Studies (COS), Heidelberg University, Germany **5** Institute of  
9 Pathology, University Hospital Heidelberg, Germany **6** Cardiovascular Imaging and Dynamics, KU  
10 Leuven, Belgium

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

## Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

## License

Authors of papers retain copyright  
and release the work under a  
Creative Commons Attribution 4.0  
International License ([CC BY 4.0](#))

## 11 Summary

12 DuneCopasi is a C++ library designed to simulate how chemical reactions and diffusion  
13 processes occur in space, which is crucial for understanding many biological systems. It allows  
14 users to model these processes in both simple and complex environments, using grids that  
15 represent one, two, or three dimensions. Optimized for modern computers, DuneCopasi can  
16 be used as a standalone command-line application, accessed through tools like Docker or a  
17 web-based terminal, as a C++ library, or integrated into a graphical interface. One such  
18 interface is the **Spatial Model Editor (SME)** ([Keegan et al., 2023](#)), which helps users create  
19 and manipulate biological reaction models. The project was initiated to provide the numerical  
20 foundation for a spatial modelling tool for biochemical reaction networks, complementing the  
21 **Complex PATHway SIMulator (COPASI)** software ([Hoops et al., 2006](#)). This development  
22 resulted from a collaboration between the COPASI team and the **Distributed and Unified**  
23 **Numerics Environment (DUNE)** team ([Bastian et al., 2021](#)).

## 24 Background

25 In the context of cell biology, computational modelling has become an essential technique  
26 for understanding and discovering biological processes. By comparing model simulations with  
27 experimental data, researchers can set up models, validate them, and test hypotheses about  
28 these processes. Traditionally, most biological systems studied this way have been spatially  
29 homogeneous. That is, the data are typically time-resolved concentrations or quantities of  
30 biochemical species, modelled using **Ordinary Differential Equations (ODEs)**.

31 However, recent advances in live-cell imaging technology have made more detailed spatio-  
32 temporal data available, leading to a growing need for models that capture not only the time  
33 dynamics but also the spatial distribution of these biochemical species. This has prompted a  
34 shift toward spatially resolved models.

## 35 Model Problem

36 A natural extension of spatially homogeneous ODE models is to incorporate spatio-temporal  
37 dynamics by formulating the problem as a system of **Partial Differential Equations (PDEs)**  
38 across multiple compartments. In such models, each compartment's PDE represents the  
39 reaction-diffusion processes of biochemical species in a specific physical domain (e.g., the

40 cytosol), while boundary conditions between compartments (e.g., membrane fluxes) govern  
41 the interactions between them.

42 Specifically, our program solves the mass balance equation for the species  $u_{ik}$  in the  $k$ -th  
43 compartment  $\Omega_k$  for every  $k \in K$  and  $i \in N_k$ . Each mass balance equation is given by

$$\begin{aligned} \partial_t(\phi_{ik}u_{ik}) &= \nabla \cdot \mathbf{j}_{ik} + \mathcal{R}_{ik}(\mathbf{u}) && \text{in } \Omega_k \subset \mathbb{R}^d, \\ u_{ik} &= u_{ik}^{(0)} && \text{on } \Gamma_k^D \subseteq \partial\Omega_k, \\ \mathbf{j}_{ik} \cdot \mathbf{n}_k &= \sum_{l \in T_k} \mathcal{T}_{ikl}(\mathbf{u}) && \text{on } \partial\Omega_k \setminus \Gamma_k^D, \end{aligned} \quad \text{with } \mathbf{j}_{ik} := \sum_{j \in N_k} D_{ijk} \nabla u_{jk}.$$

44 Here,  $\mathbf{u}_k := (u_{1k}, \dots, u_{\dim(N_k)k})$  represents the vector of species concentrations in compart-  
45 ment  $k$ , while the full vector of species concentrations across all compartments is denoted as  
46  $\mathbf{u} := (\mathbf{u}_1, \dots, \mathbf{u}_{\dim(K)})$ . The unit outer normal vector on the boundary  $\partial\Omega_k$  is represented by  
47  $\mathbf{n}_k$ , and the set of neighbouring compartments to  $k$  is given by  $T_k := \{l \in K : \partial\Omega_k \cap \bar{\Omega}_l \neq \emptyset\}$ ,  
48 indicating the compartments that share a boundary with  $\Omega_k$ .

49 The reaction operator  $\mathcal{R}_{ik}(\mathbf{u})$  governs the local reaction dynamics within  $\Omega_k$ , while the storage  
50 terms  $\phi_{ik}$  account for species accumulation in the compartment. Cross-diffusion terms  $D_{ijk}$   
51 describe how species diffuse between different species within the same compartment. The non-  
52 linear transmission conditions  $\mathcal{T}_{ikl}(\mathbf{u})$  represent the outflow of species  $u_{ik}$  from compartment  
53  $\Omega_k$ , where the outflow can either move to a neighbouring compartment  $l$  (if  $l \neq k$ ) or exit  
54 the system. Likewise, Dirichlet boundary conditions  $u_{ik}^{(0)}$  are imposed on the subset  $\Gamma_k^D$  of the  
55 boundary  $\partial\Omega_k$ , specifying the fixed concentrations of species on that portion of the boundary.

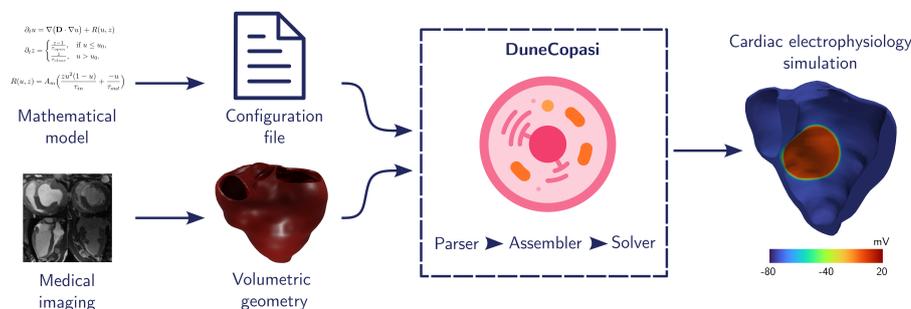
56 The parameters governing these equations are fully configurable at run-time, either through  
57 the command line or via a configuration file. By allowing full control over these parameters,  
58 users can adapt the software to simulate a wide variety of biochemical processes, from simple  
59 reactions in homogeneous environments to complex, multi-compartmental systems with intricate  
60 boundary conditions and interactions.

## 61 Capabilities

62 Many features of DuneCopasi have been designed and developed with the specific case of  
63 systems biology in mind. Thus, a substantial effort has been made to have a library that is  
64 interoperable with systems biology data assets and requirements needed by their practitioners.  
65 Among others, include:

- 66 ■ a single executable configurable at run-time,
- 67 ■ a run-time mathematical expression parser that understands the Systems Biology Markup  
68 Language SBML (Hucka et al., 2003) specification,
- 69 ■ the input of custom grid data and image files in the TIFF format for initial spatial  
70 concentrations and other parameters,
- 71 ■ a powerful, yet simple, boundary/transmission condition specification for each compart-  
72 ment to account for generic trans-membrane fluxes,
- 73 ■ a (non-linear) cross-diffusion specification that allows any species to cross-diffuse into  
74 other mass balance equations,
- 75 ■ a specification to compare results with user-defined objective functions,
- 76 ■ an in-place function interpolator that reduces the computational cost of evaluating  
77 common expensive reaction operations, and
- 78 ■ an embedded random field generator (Kempf et al., 2023) to represent statistical spatial  
79 variations on the domain.

80 Furthermore, DuneCopasi is a stand-alone multi-compartment reaction-diffusion solver that  
81 may easily be used for many other fields of research and engineering fitting our model problem  
82 (e.g. [Figure 1](#)).



**Figure 1:** DuneCopasi is a highly flexible simulator that is run-time configurable using dedicated configuration files. DuneCopasi can autonomously interpret mathematical equations, assemble the computational problems and solve these equations based on the configuration file. Here we illustrate the integration of DuneCopasi within a computational electrophysiology pipeline allowing to solve the electrophysiological mono-domain equations with Mitchell-Schaefer cell model using a medical imaging derived biventricular geometry.

## 83 Statement of Need

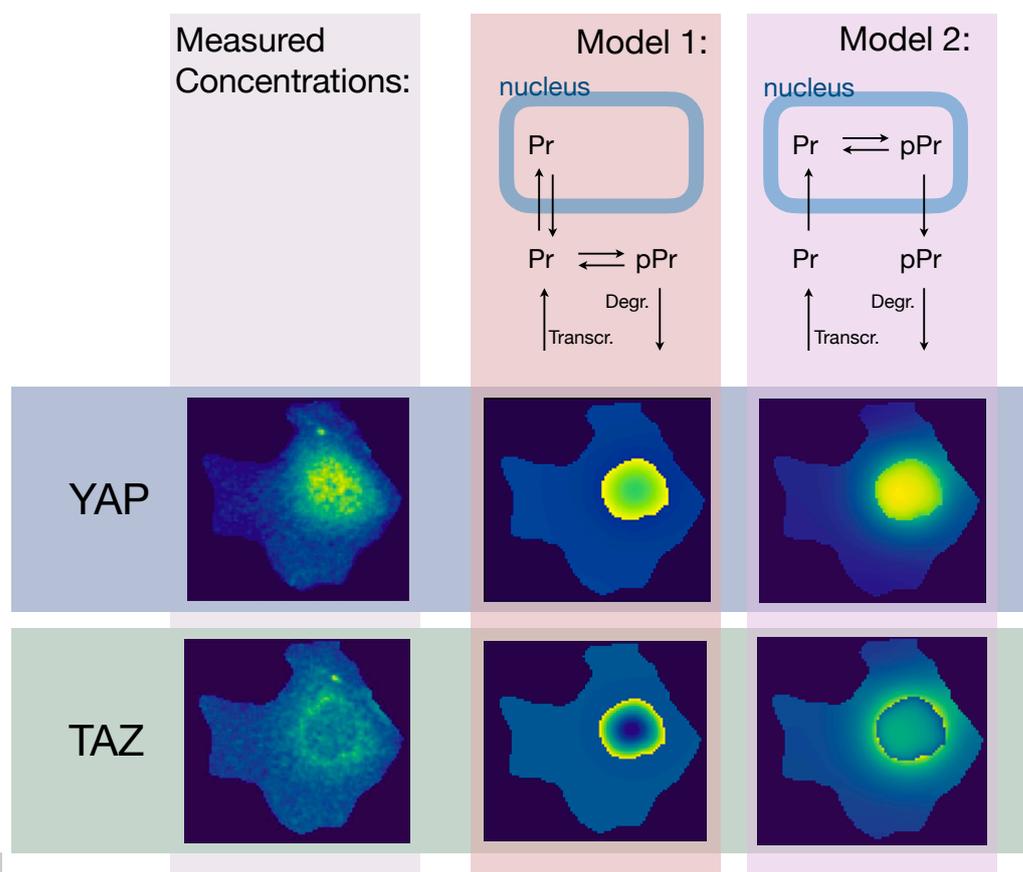
84 Finite element frameworks like DUNE ([Bastian et al., 2008](#)), Deal.II ([Arndt et al., 2023](#)),  
85 Necktar++ ([Cantwell et al., 2015](#)), or FreeFem++ ([Hecht, 2012](#)) are too generic by design and  
86 don't address the specific requirements of the computational modelling practices in systems  
87 biology out of the box. In particular, most finite element frameworks don't consider the  
88 multi-compartment systems in their design, resulting, if at all possible, in inefficient simulations  
89 or obscure tricks to force this feature. Here, we extended DUNE-PDELab ([Bastian et al., 2010](#);  
90 [Müthing, 2015](#)) with efficient data structures especially tailored for this task.

91 In the space of systems biology, two well-known software packages also provide similar features  
92 to DuneCopasi, namely Morpheus ([Starruß et al., 2014](#)) and VCell ([Schaff et al., 1997](#)).  
93 Morpheus is a computational tool for multi-cellular systems which follows a cellular automata  
94 design to set rules of interactions between individual cells. It focuses on modelling approaches  
95 like cellular Potts models combined with gradients modelled by PDEs, rather than multi-  
96 compartment PDE models. Furthermore, due to its design, it falls short for moderate and big  
97 PDE computations since explicit solvers are severely limited by the required time steps. On the  
98 other hand, VCell provides a fully implicit finite volume solver on structured grids that can very  
99 well manage the multi-compartment case in addition to membrane unknowns. In comparison,  
100 our solution aims to resolve the geometry and the transmission conditions directly in the weak  
101 formulation of the problem while also being designed to provide implicit and monolithic solvers  
102 as well as tailored preconditioners for the underlying linear solvers.

## 103 Research and Use Cases

104 The principal purpose of our project has been to bridge the gap between systems biology  
105 and scientific computing by providing researchers with an accessible and reproducible spatial  
106 simulator. An illustration of this is the study depicted in [Figure 2](#) ([Wehling et al., 2022](#)), where  
107 simulations with the SME ([Keegan et al., 2023](#)) aided in assessing and better understanding  
108 the mechanisms of the YAP and TAZ proteins that regulate cell proliferation in liver cells.  
109 Further examples that fit our model problem in this context have been presented by others

110 (Eliaš & Clairambault, 2014). Additionally, DuneCopasi stands as a versatile tool capable of  
 111 accommodating diverse computational needs beyond its initial focus on systems biology. For  
 112 example, the general purpose nature allows for seamless integration into electrophysiology  
 113 simulations, without requiring any modifications. The emerging concept of identifying patients  
 114 based on personalized cardiac electrophysiology simulations (Arevalo et al., 2016) underscores  
 115 the demand for easy-to-use and efficient simulators. Figure 1 demonstrates how DuneCopasi  
 116 emerges as a flexible solver poised to meet these evolving needs.



**Figure 2:** Mathematical modelling predicts that nuclear phosphorylation controls spatial localization of Hippo signalling pathway components YAP and TAZ. Here, we compare two model topologies - “Model 1” (canonical) and “Model 2” (alternative model) - concerning the intracellular distribution of YAP and TAZ proteins (Pr) and their phosphorylated counterparts (pPr). If the phosphorylation of YAP/TAZ takes place exclusively outside the nucleus, as shown in “Model 1”, PDE simulation indicates low spatial accordance with the experimentally measured subcellular localization of YAP/TAZ. Whereas, “Model 2” describes YAP/TAZ protein phosphorylation and dephosphorylation in the nucleus. The simulation of “Model 2” agrees with experimentally measured subcellular distribution of YAP/TAZ proteins, as reported in (Wehling et al., 2022).

## 117 Acknowledgements

118 We want to thank all the contributions that aided in the development and deployment of the  
 119 package. Special thanks to Ursula Kummer from the COPASI team for valuable guidance.  
 120 This work has been funded and supported by the German Federal Ministry of Education and  
 121 Research (BMBF) FKZ 031L0158.

## References

- 122
- 123 Arevalo, H. J., Vadakkumpadan, F., Guallar, E., Jebb, A., Malamas, P., Wu, K. C., &  
124 Trayanova, N. A. (2016). Arrhythmia risk stratification of patients after myocardial  
125 infarction using personalized heart models. *Nature Communications*, 7(1), 11437. <https://doi.org/10.1038/ncomms11437>  
126
- 127 Arndt, D., Bangerth, W., Bergbauer, M., Feder, M., Fehling, M., Heinz, J., Heister, T., Heltai,  
128 L., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.-P., Turcksin, B., Wells, D., &  
129 Zampini, S. (2023). The deal.II library, version 9.5. *Journal of Numerical Mathematics*,  
130 31(3), 231–246. <https://doi.org/10.1515/jnma-2023-0089>
- 131 Bastian, P., Blatt, M., Dedner, A., Dreier, N.-A., Engwer, C., Fritze, R., Gräser, C., Grüninger,  
132 C., Kempf, D., Klöforn, R., Ohlberger, M., & Sander, O. (2021). The dune framework:  
133 Basic concepts and recent developments. *Computers & Mathematics with Applications*,  
134 81, 75–112. <https://doi.org/10.1016/j.camwa.2020.06.007>
- 135 Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöforn, R., Ohlberger, M., & Sander, O.  
136 (2008). A generic grid interface for parallel and adaptive scientific computing. Part i: Ab-  
137 stract framework. *Computing*, 82, 103–119. <https://doi.org/10.1007/s00607-008-0003-x>
- 138 Bastian, P., Heimann, F., & Marnach, S. (2010). Generic implementation of finite element  
139 methods in the Distributed and Unified Numerics Environment (DUNE). *Kybernetika*,  
140 46(2), 294–315. [dml.cz/dmlcz/140745](http://dml.cz/dmlcz/140745)
- 141 Cantwell, C. D., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., De Grazia, D.,  
142 Yakovlev, S., Lombard, J.-E., Ekelschot, D., Jordi, B., Xu, H., Mohamied, Y., Eskilsson,  
143 C., Nelson, B., Vos, P., Biotto, C., Kirby, R. M., & Sherwin, S. J. (2015). Nektar++:  
144 An open-source spectral/hp element framework. *Computer Physics Communications*, 192,  
145 205–219. <https://doi.org/10.1016/j.cpc.2015.02.008>
- 146 Eliaš, J., & Clairambault, J. (2014). Reaction–diffusion systems for spatio-temporal intracellular  
147 protein networks: A beginner’s guide with two examples. *Computational and Structural*  
148 *Biotechnology Journal*, 10(16), 12–22. <https://doi.org/10.1016/j.csbj.2014.05.007>
- 149 Hecht, F. (2012). New development in FreeFem++. *J. Numer. Math.*, 20(3-4), 251–265.  
150 <https://doi.org/10.1515/jnum-2012-0013>
- 151 Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes,  
152 P., & Kummer, U. (2006). COPASI—a COmplex Pathway Simulator. *Bioinformatics*,  
153 22(24), 3067–3074. <https://doi.org/10.1093/bioinformatics/btl485>
- 154 Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P.,  
155 Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D.,  
156 Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., ...  
157 SBML Forum, the rest of the. (2003). The systems biology markup language (SBML): a  
158 medium for representation and exchange of biochemical network models. *Bioinformatics*,  
159 19(4), 524–531. <https://doi.org/10.1093/bioinformatics/btg015>
- 160 Keegan, L., Andriushchenko, P., Schreiner, H., Caramizaru, H., & Patel, H. (2023). *Spatial-*  
161 *model-editor/spatial-model-editor: 1.5.0* (Version 1.5.0). Zenodo. [https://doi.org/10.](https://doi.org/10.5281/zenodo.10246531)  
162 [5281/zenodo.10246531](https://doi.org/10.5281/zenodo.10246531)
- 163 Kempf, D., Klein, O., Kutri, R., Scheichl, R., & Bastian, P. (2023). Parafields: A generator  
164 for distributed, stationary gaussian processes. *Journal of Open Source Software*, 8(92),  
165 5735. <https://doi.org/10.21105/joss.05735>
- 166 Müthing, S. (2015). *A flexible framework for multi physics and multi domain PDE simulations*  
167 [PhD thesis, Universität Stuttgart]. <https://doi.org/10.18419/opus-3620>

- 168 Schaff, J., Fink, C. C., Slepchenko, B., Carson, J. H., & Loew, L. M. (1997). A general  
169 computational framework for modeling cellular structure and function. *Biophys. J.*, 73(3),  
170 1135–1146. [https://doi.org/10.1016/s0006-3495\(97\)78146-3](https://doi.org/10.1016/s0006-3495(97)78146-3)
- 171 Starruß, J., Back, W. de, Brusch, L., & Deutsch, A. (2014). Morpheus: a user-friendly  
172 modeling environment for multiscale and multicellular systems biology. *Bioinformatics*,  
173 30(9), 1331–1332. <https://doi.org/10.1093/bioinformatics/btt772>
- 174 Wehling, L., Keegan, L., Fernández-Palanca, P., Hassan, R., Ghallab, A., Schmitt, J., Tang, Y.,  
175 Le Marois, M., Roessler, S., Schirmacher, P., & others. (2022). Spatial modeling reveals  
176 nuclear phosphorylation and subcellular shuttling of YAP upon drug-induced liver injury.  
177 *Elife*, 11, e78540. <https://doi.org/10.7554/eLife.78540>

DRAFT