supported by the German Federal Ministry of Education and Research (BMBF) within the framework of the e:Med research and funding concept (grant no. 01ZX1912C).

Author contributions

M.B.G. and J.Q conceived the project. D.W., C.M.L.-R, R.B., R.T.P., K.P., M.S., M.F., D.S., K.G., M.A., M.L.K. and J.P. contributed use cases for the collection of notebooks. All authors edited and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-022-01479-2.

Peer review information *Nature Methods* thanks Laurence Calzone and Michael Reich for their contribution to the peer review of this work.

Check for updates

The Simularium Viewer: an interactive online tool for sharing spatiotemporal biological models

To the Editor — We present the Simularium Viewer, a user-friendly, open-source application that makes it easy to share and interrogate interactive three-dimensional (3D) visualizations of biological simulation trajectories directly in a web browser at https://simularium.allencell.org. The primary goal of the Simularium project is to facilitate collaborations among experimental and computational biologists by removing challenges to sharing, accessing and comparing simulation results. As a new arrow in the modeling community's quiver, the Simularium Viewer provides a platform to share simulation outputs in an easy-to-use interface that requires no computational expertise from end users. With a relatively small effort, any modeling researcher can use our conversion package to save their data as a Simularium file and generate a link that anyone can use to interactively investigate their simulation trajectories and related plots immediately in a browser, instead of spending time downloading, installing and learning to use tools specific to any given model.

Spatial simulations are a powerful tool for investigating biological phenomena at different scales;[1,2] however, by surveying researchers, educators and students in the field of cell biology, we have identified several pain points that restrict their utility and adoption. In general, simulation tools are often difficult for non-computational biologists to use, provide limited options for the easy sharing of interactive visualizations, and must be downloaded and run as independent desktop software—often with challenging installation processes. Many require users to write code or to run software from a terminal window and offer only a limited user interface or no graphical user interface. Most visualization tools for spatial modeling are either specialized to support atomic coordinates and/or image data or built to support visualization for one particular modeling engine (Supplementary Table 1), making it challenging to compare models that were created using different engines. The common practice of developing new visualization software from scratch often results in redundant efforts to solve common challenges that take developer time and expertise away from addressing unsolved problems in the field.

To make simulated models more accessible and easier for broader audiences to examine, we have developed the Simularium Viewer as an online 3D viewport with a graphical user interface that enables interactive exploration of dynamic spatial data, alongside plots of calculated metrics, directly in a web browser (Fig. 1). It uses advanced rendering techniques[3] to enable meaningful interpretations of spatial relationships[4,5] among thousands of moving components in 3D. The viewer allows a user to interact with precalculated simulation results, including their own results converted into the Simularium file format (JSON or binary). Files can be loaded from local disk storage or shared publicly by generating URLs formatted as https://simularium.allencell.org/viewer?trajUrl=[link to file]. These links enable one-click access to an interactive simulation window. We currently support public links to Dropbox, Google Drive and Amazon S3 files and will continue to add support for other cloud storage providers. URL access allows any user to share their own library of simulations publicly, and these links can be included in publications to provide easy access for readers to explore interactive supplementary figures, as demonstrated in a recent article that includes links to models produced using the SpringSaLaD engine[6]. The Simularium Viewer website also provides a library of example models that were generated by different simulation platforms and selected to represent a broad variety of simulation types, visual complexity and spatiotemporal scales. Examples include a Cytosim model of clathrin-mediated endocytosis[7], a PhysiCell model of SARS-CoV-2 dynamics in human lung epithelium[8], a Smoldyn model of the *Escherichia coli* Min1 system[9], and a published model of a membrane wrapping a nanoparticle[10] (Supplementary Table 2).

The Simularium Viewer is the first component released from a larger Simularium platform in development. It is freely available at https://simularium.allencell.org. We welcome community involvement in the development process, and Simularium code repositories are available at https://github.com/simularium with documentation and an open-source license. The Help menu on the Simularium Viewer website includes tutorials, sample Python code and Jupyter Notebooks to guide the conversion of custom data into the format consumed by the viewer, as well as instructions for sharing your own Simularium files online. We provide Python Jupyter notebooks for converting the outputs of several community-developed spatial simulation engines (Supplementary Table 3) and plan to continue working with engine developers to write directly to the Simularium file format and support more engines. We are also working to transcode native files output from supported engines automatically upon loading, similarly to other streaming services. We have forums and issue boards to collect feedback and will continue to work with Simularium users to add visualization features that will make the Simularium Viewer an even more powerful open-source tool for exploring dynamic 3D biological data.

Reporting Summary

Further information on research design is available in the Nature Research Reporting Summary linked to this article.

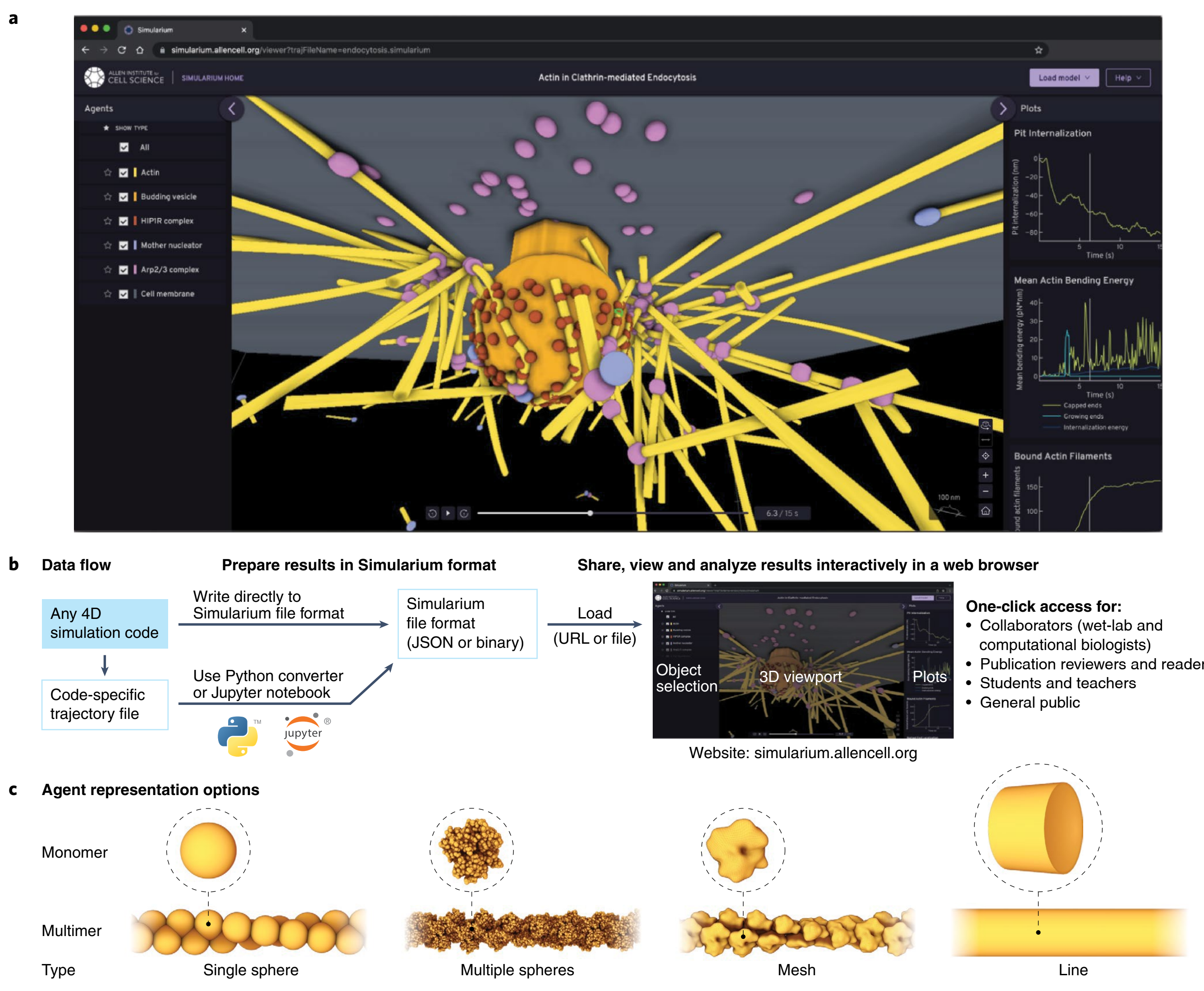
a
Simularium
simularium.allencell.org/viewer?trajFileName=endocytosis.simularium
ALLEN INSTITUTE for CELL SCIENCE
SIMULARIUM HOME
Actin in Clathrin-mediated Endocytosis
Load model
Help
Agents
SHOW TYPE
All
Actin
Budding vesicle
HIPIR complex
Mother nucleator
Arp2/3 complex
Cell membrane
Plots
Pit Internalization
Mean Actin Bending Energy
Bound Actin Filaments
Time (s)
Capped ends
Growing ends
Internalization energy
100 nm
6.3 / 15 s
b
Data flow
Prepare results in Simularium format
Share, view and analyze results interactively in a web browser
Any 4D simulation code
Code-specific trajectory file
Write directly to Simularium file format
Use Python converter or Jupyter notebook
Simularium file format (JSON or binary)
Load (URL or file)
Object selection
3D viewport
Plots
Website: simularium.allencell.org
One-click access for:
• Collaborators (wet-lab and computational biologists)
• Publication reviewers and readers
• Students and teachers
• General public
c
Agent representation options
Monomer
Multimer
Type
Single sphere
Multiple spheres
Mesh
Line

Fig. 1 | Simularium enables one-click access to the interactive visual analysis of spatiotemporal simulations. a, A screenshot of the Simularium Viewer loaded with an example model[7] has a 3D viewport with an interactive timeline in the center and two collapsible panels on the sides holding entity selection on the left and trajectory related plots on the right. **b**, Data from simulation engines must be converted to the Simularium file format, using either direct output from an engine or postprocessing with Python converters. Since the viewer is web-based, a link can be shared with collaborators, paper reviewers and readers, or even broader audiences of students and the general public. **c**, Rendering options for agents in the trajectory viewport currently include single spheres; collections of spheres with internal structure specified by, for example, Protein Data Bank structure files; mesh surfaces; and 3D lines. We plan to add volume rendering for PDE or RDME-based data.

Code availability

The software described here is available under a modified BSD license at https://github.com/simularium and is free for non-commercial use.

Blair Lyons[1], Eric Isaac[1], Na Hyung Choi[1], Thao P. Do[1], Justin Domingus[1], Janet Iwasa[2], Andrew Leonard[1], Megan Riel-Mehan[1], Emily Rodgers[1], Lisa Schaefbauer[1], Daniel Toloudis[1], Olivia Waltner[1], Lyndsay Wilhelm[1] and Graham T. Johnson[1] ✉

[1]*Allen Institute for Cell Science, Animated Cell, Seattle, WA, USA.* [2]*University of Utah, Biochemistry, Salt Lake City, UT, USA.*
✉*e-mail: grahamj@alleninstitute.org*

Published online: 4 April 2022
https://doi.org/10.1038/s41592-022-01442-1

References

1. Osborne, J. M., Fletcher, A. G., Pitt-Francis, J. M., Maini, P. K. & Gavaghan, D. J. *PLOS Comput. Biol.* **13**, e1005387 (2017).
2. Johnson, M. E. et al. *Mol. Biol. Cell* **32**, 186–210 (2021).
3. Miao, H. et al. *J. Mol. Biol.* **431**, 1049–1070 (2019).
4. Langer, M. S. & Bülthoff, H. H. *Perception* **29**, 649–660 (2000).
5. Johnson, G. T., Autin, L., Goodsell, D. S., Sanner, M. F. & Olson, A. J. *Structure* **19**, 293–303 (2011).
6. Chattaraj, A., Blinov, M. L. & Loew, L. M. *eLife* **10**, e67176 (2021).
7. Akamatsu, M. et al. *eLife* **9**, e49840 (2020).
8. Getz, M. et al. Preprint at *bioRxiv* https://doi.org/10.1101/2020.04.02.019075 (2021).
9. Andrews, S. S., Addy, N. J., Brent, R. & Arkin, A. P. *PLOS Comput. Biol.* **6**, e1000705 (2010).
10. Sadeghi, M., Weikl, T. R. & Noé, F. *J. Chem. Phys.* **148**, 044901 (2018).

Acknowledgements

We thank Johannes Schöneberg for help building example models and discussion; Moritz Hoffmann, Christoph Fröhner and Frank Noé for training and development with ReaDDy; Adam Husar, Tom Bartol and Jacob Czech for training and collaboration using MCell; James Faeder for training in BioNetGen and BNGL; François Nédélec for training and development with dynamic cytoskeletal fibers and membranes in Cytosim and Matthew Akamatsu and Daniel Serwas for insight, testing and example data; Leslie Loew, Michael Blinov, Ann Cowen, Ion Moraru, Aniruddha Chattaraj, Steve Andrews, Garegin Papoian, Paul Macklin, Randy Heiland, Ryan Spangler and Eran Agmon for collaboration to integrate their simulators and example data; Ludovic Autin and Ivan Viola for visualization inspiration and discussion; Jonathan Alberts for extensive help testing the prototype viewer with large trajectories; and Rick Horwitz for suggestions. Researchers from the Allen Institute for Cell Science wish to thank the Allen Institute founder, Paul G. Allen, for his vision, encouragement and support. Please see the Supplementary Acknowledgements to help us thank a long list of additional contributors who helped with user testing, scientific and design feedback, implementation testing and more.

Author contributions

B.L., J.I., and G.T.J. conceived the project; B.L., E.I., J.D., M.R.-M., and D.T. designed the software architecture; B.L., E.I., N.H.C., J.D., M.R.-M. and D.T. wrote the software and A.L. helped with debugging; L.S. designed user interfaces and conducted user tests; T.P.D. contributed illustrations and managed analytics data; T.P.D. and L.W. refined user interface designs; E.R. and O.W. helped test the software; G.T.J. supervised the project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-022-01442-1.

Peer review information *Nature Methods* thanks Zaida Luthey-Schulten and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Viv: multiscale visualization of high-resolution multiplexed bioimaging data on the web

To the Editor — Advances in highly multiplexed imaging have enabled the comprehensive profiling of complex tissues in healthy and diseased states, facilitating the study of fundamental biology and human disease at spatially resolved, subcellular resolution[1,2]. Although the rapid innovation of biological imaging brings significant scientific value, the proliferation of technologies without unification of interoperable standards has created challenges that limit the analysis and sharing of results. The adoption of community-designed next-generation file formats (NGFF) is a proposed solution to promote bioimaging interoperability at scale[3]. Here we introduce Viv (https://github.com/hms-dbmi/viv), an open-source bioimaging visualization library that supports OME-TIFF[4] and OME-NGFF[3] directly on the web. Viv addresses a critical limitation of most web-based bioimaging viewers by removing a dependency on server-side rendering, offering a flexible toolkit for browsing multi-terabyte datasets on both mobile and desktop devices—without software installation.

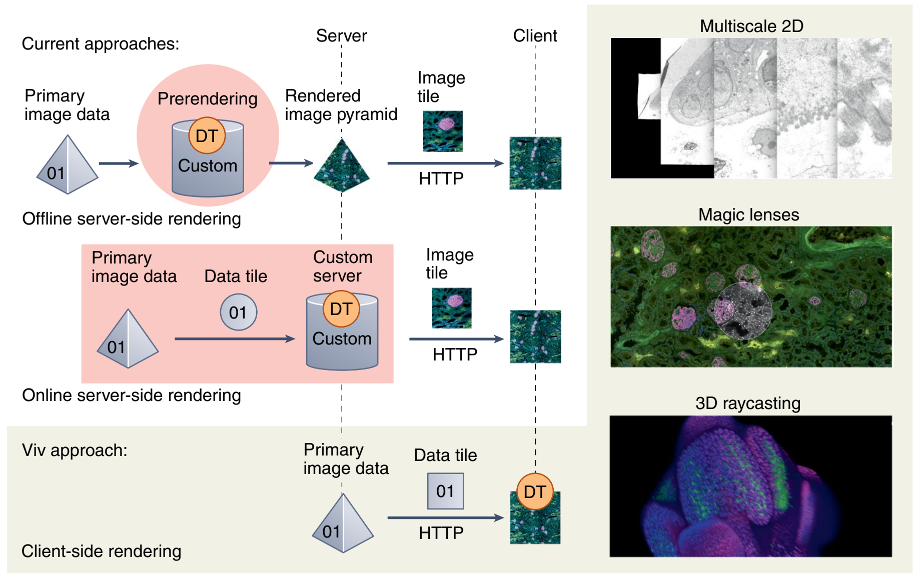

Fig. 1 | Overview of data flow and rendering approaches for web-based bioimage data visualization and Viv features. DT indicates the location where data is transformed into an image. The right column displays a subset of Viv's flexible client-side rendering, including multiscale 2D pyramids, magic lenses and 3D volumes via raycasting.

Viv functions more similarly to popular desktop bioimaging applications than to corresponding web alternatives (Supplementary Note 1). Most web viewers require the pre-translation of large binary data files into rendered images (PNG or JPEG) for display in a browser client. Two existing approaches perform this step (server-side rendering) but differ with regard to when rendering occurs and how much of the binary data is transformed at one time (Fig. 1). The offline option performs all rendering before an application is deployed to users, meaning that all channel groupings and data transformations are fixed and cannot be adjusted via the user interface. The online option supports on-demand, user-defined rendering but introduces latency when exploring data transformations and requires active maintenance of complex server infrastructure. Neither approach provides a flexible solution to directly view datasets saved in open formats from large-scale public data repositories, and the transient data representation introduced by server-side rendering inhibits interoperability with other visualization and analysis software.

Viv implements purely client-side rendering to decouple the browser from the server and still offer the flexibility of on-demand multichannel rendering. Existing web viewers also leverage graphical processing unit (GPU)-accelerated rendering but are generally tailored toward single-channel volumetric datasets and, crucially, lack the ability to reuse and compose features for existing or novel applications[5,6]. In contrast, Viv's modularity allows core functionality to be repurposed and extended. The library consists of two primary components: (i) data-loading