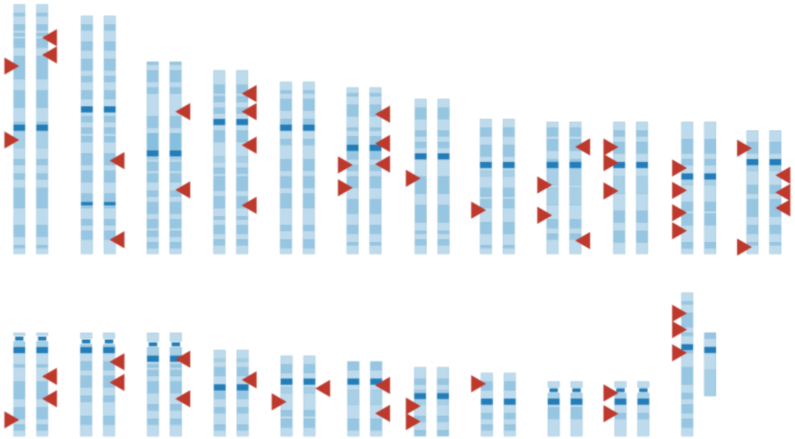
Reusing & combining simulations with BioSimulators, BioSimulations & Vivarium

E. Agmon, J. Karr, B. Shaikh & R. Spangler

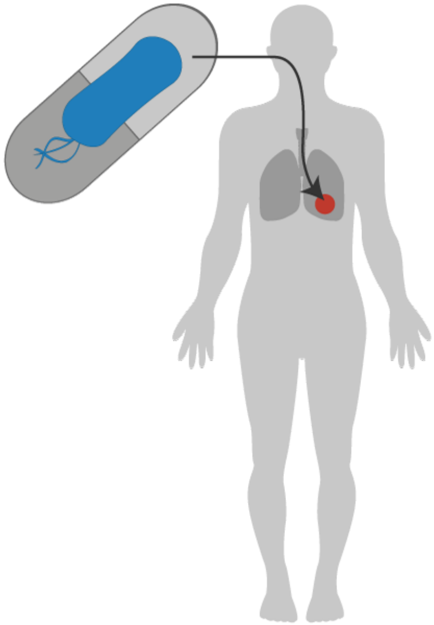
BioSimulators.org BioSimulations.org, Vivarium-Collective.github.io

October 13, 2021

Goal: understand, control, and design biology



Precision diagnosis
and therapy



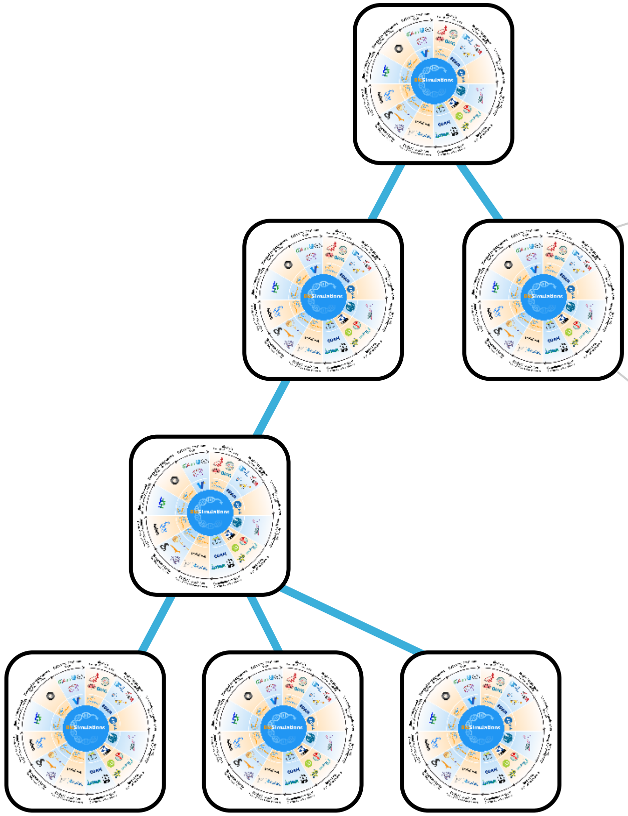
Synthetic cells
and organisms

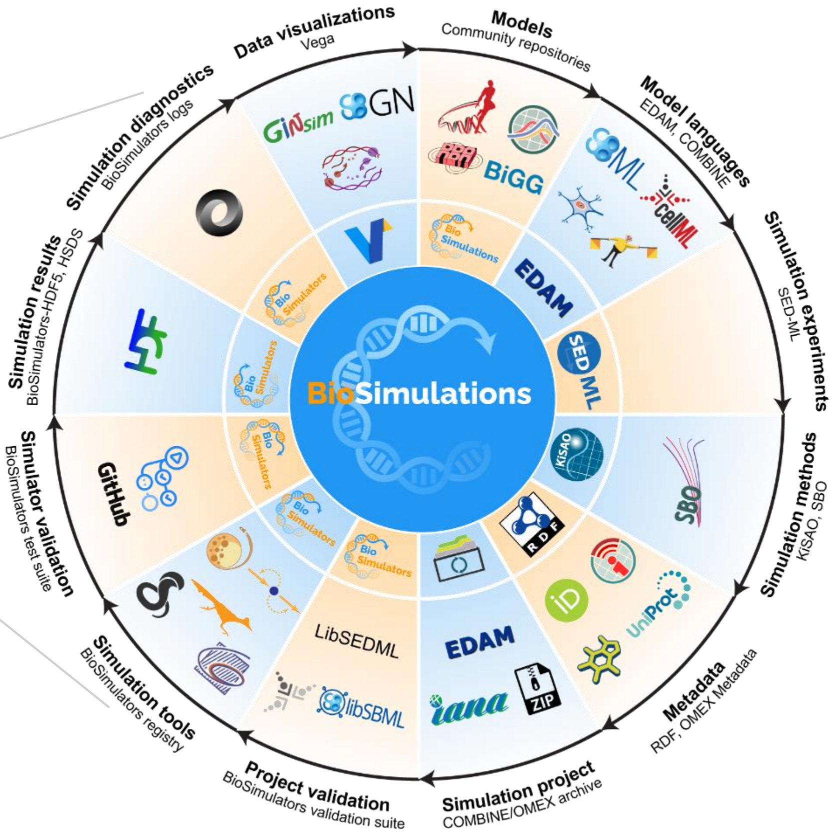
Vivarium

Framework for composition

BioSimulators/BioSimulations

Reusable models and simulation tools





Today's agenda

- **14:00: Introduction**
- **14:05: BioSimulators/BioSimulations**
  - Overview of concepts
  - Major progress since HARMONY 2021
  - Demos
    - Browsing published simulations online
    - Reusing simulations with standardized Python APIs
  - Questions and discussion
- **15:00: Vivarium**
  - Overview of concepts
  - Demo of a DFBA simulation
  - Questions and discussion

Week's agenda

- **Wednesday** 14:00 UTC
  - – Introduction: Goals, concepts and demos
- **Thursday** 14:00 UTC
  - – Hacking: work with the developers
- **Friday** 16:00 UTC
  - – Discussion: Community and input

Collaborate with us!

Reach out to work together

info@biosimulations.org

eagmon@stanford.edu

github.com/biosimulations/biosimulations

github.com/vivarium-collective/vivarium-core

Participate in weekly meetings

BioSimulations: Thu 11 am EDT

https://bit.ly/biosimulations-team-meeting

Online resources

BioSimulators

https://biosimulators.org

BioSimulations

https://biosimulations.org

Vivarium

https://vivarium-collective.github.io

BioSimulators/BioSimulations

Mount Sinai

Acknowledgements: BioSimulators/BioSimulations

Core team

Michael Blinov
Andrew Freiburger
Jonathan Karr
Gnaneswara Marupilla
Ion Moraru
Bilal Shaikh
Herbert Sauro
Lucian Smith
Mike Wilson

Funding

Center for Reproducible Biomedical Modeling
National Institutes of Health
National Science Foundation

Modeling formats

BNGL
CellML
COMBINE
NeuroML
LEMS
OMEX Meta
RBA XML
SBML
SED-ML
Smoldyn
XPP ODE

Model format libraries

libCellML
libCOMBINE
libNeuroML
libOMEXMeta
libSED-ML
pyBioNetGen
pyNeuroML
pyLEMS

Simulation tools

AMICI
BioNetGen
CBMPy
COBRApy
COPASI
GillesPy2
GINsim
iBioSim
LibSBMLSim
MASSpy
NEURON
NetPyNe
OpenCOR
pyNeuroML
PySCeS
RBApy
Smoldyn
tellurium
VCell
XPP

Model repositories

BiGG
BioModels
ModelDB
JWS Online
Physiome
RBA Bacteria
RuleHub

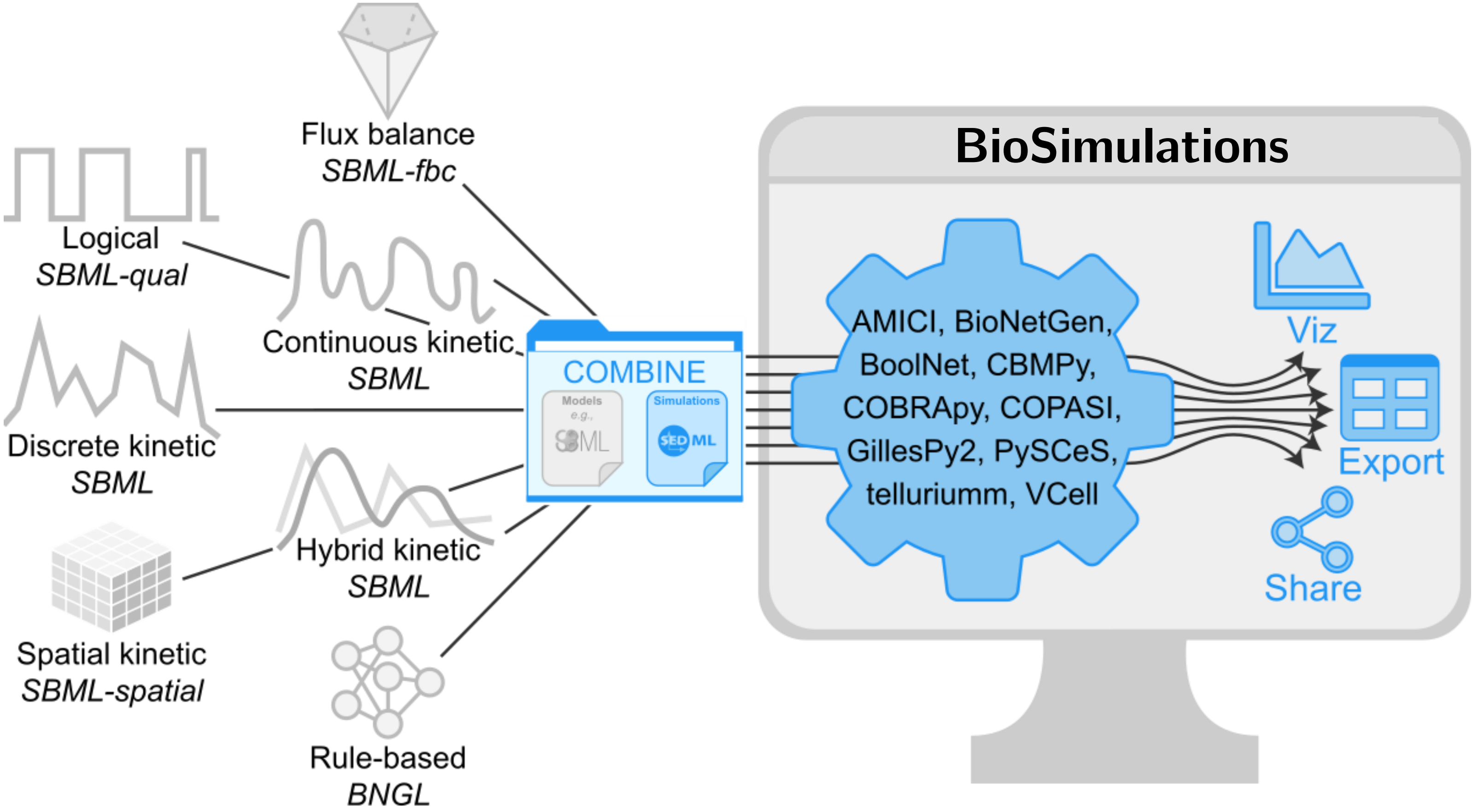
Ontologies

EDAM
Identifiers.org
KiSAO
SBO
SIO

Other

CoLoMoTo
HDF Group
Vivarium

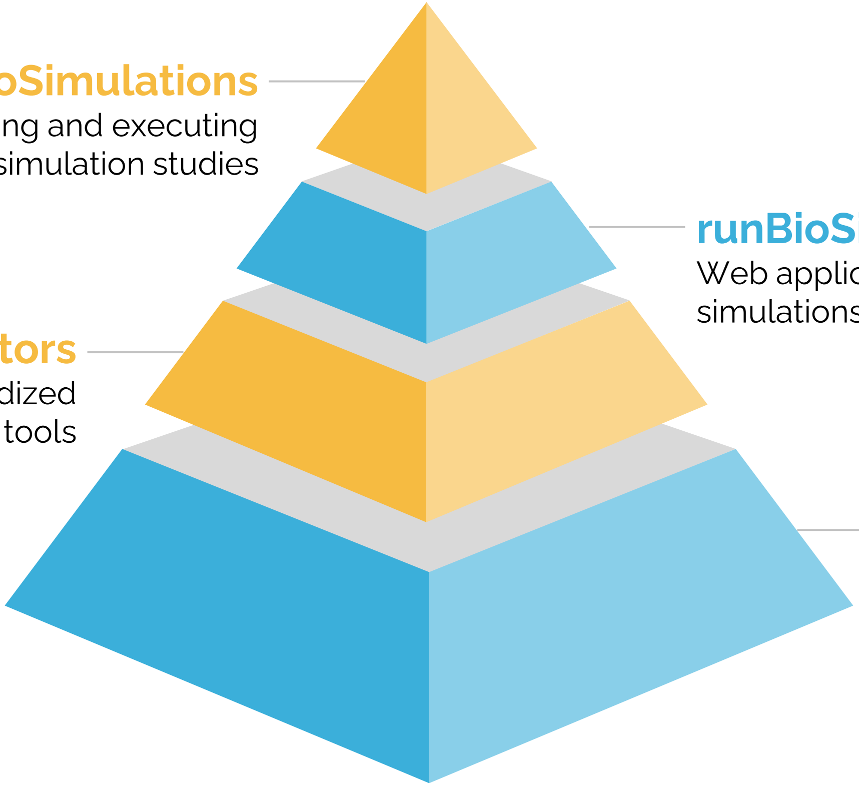
Goal: reuse models, simulations & simulation tools



Goal: reuse models, simulations & simulation tools

- Entire modeling projects: models, simulations, simulation tools, simulation results, visualizations, metadata
  - Broad range of modeling format and methods
  - Small and large models
  - Basic and domain-specific charts
- Just work for users: simple, up to date, and high-quality
  - Automated recommendations of methods and tools
  - Recommended tools must be available
  - Seamless transitions between creating, publishing, and reusing models
  - High-quality simulations: empower users, peer reviewers to validate work
  - Have up to date tools: Easy for developers, curators to maintain tools
  - Have old tools for old models
- Fully transparent: execution and curation
- Leverage existing work
- Support existing and emerging modeling communities of all sizes

Method: layers of abstraction via standards



Vivarium
Framework for composition

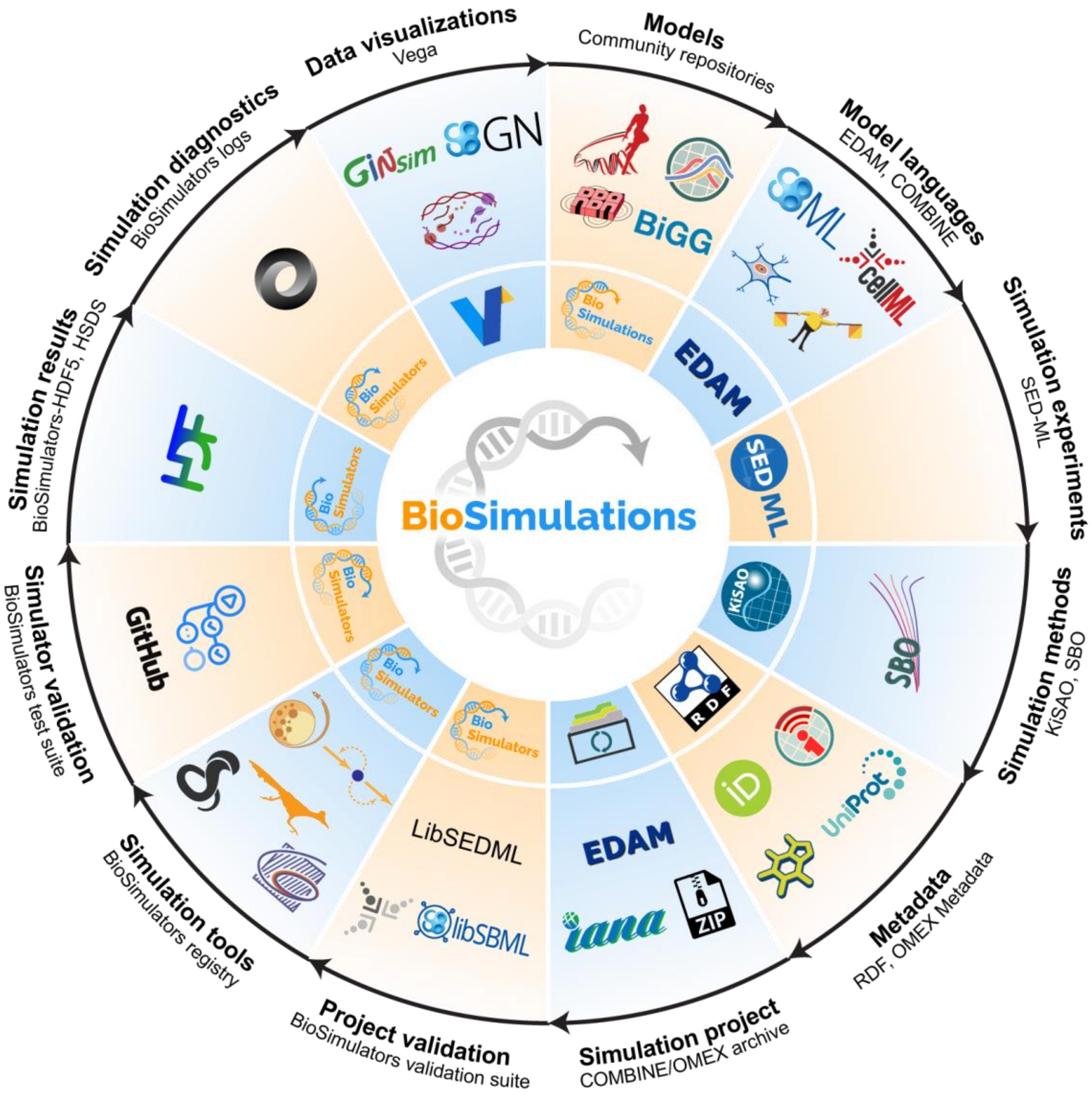
BioSimulations
Platform for sharing and executing simulation studies

runBioSimulations
Web application for executing simulations

BioSimulators
Registry of standardized simulation tools

Foundational formats and validation tools
Models, simulation experiments, simulation methods, simulation tools

BioSimulations: platform for sharing projects



Key integrations

- **Creating shareable simulation projects**
  - Tools for generating SED-ML, COMBINE archives from models
  - Seamless connection between work and publication: use the same simulation tools locally as BioSimulations
  - Project validation, simulation logs
- **Publishing projects and tools that others can reproduce**
  - Empower authors to validate their own work: review what others will see
- **Reusing simulations across tools**
  - Consistent interfaces to tools: image, CLI, Python
  - Algorithm and simulator recommendation using EDAM, KiSAO, and simulator specifications
  - Automated algorithm substitution

Key integrations

- **Extensibility to additional languages, algorithms & simulation tools**
  - Require minimal modular contributions (e.g., run single simulation)
  - Use SED-ML, KiSAO, OMEX Metadata, COMBINE archives to abstract projects
  - Use OCI images to abstract environments for tools
  - Manage license keys
- **Keeping model repositories and tools up to date**
  - Automated tasks to pull, test, and submit projects
  - Automated simulator submission and validation
- **Building visualizations:** utilities for converting visualizations to Vega
- **Simple management:** public cloud infrastructure
  - GitHub repositories, teams
  - OCI images, GitHub Container Registry
  - Unit testing and GitHub actions

Results

- **Users:** simple reuse
  - Central places to discover and use projects and tools
  - Concrete provenance, permanent archival
  - Services for recommending methods and tools
  - Consistent OCI images, command-line programs, and Python APIs
- **Modelers:** tools to make work reusable, ability to preview how others will see your work
  - Tools for validating and executing projects
  - Detailed information about the simulation capabilities of tools
  - Diagnostic simulation logs
  - Tools for painting complex data visualizations
- **Tool and standards developers:** expanded user access to tools
  - Enable more focused, modular tools
  - Platform where users can use and combine tools and share projects
  - Platform for automatically testing, releasing, and archiving tools
  - Improved formats and libraries, expanded ontologies
  - Transparent curation

Results

- **Peer reviewers and curators:** quick review
  - Authors responsible for making projects executable
  - Web application for quickly exploring simulation results
  - Easy to review a broad range of projects

Anticipated impacts

- Enabling **co-simulation** (e.g., Vivarium)
- Enabling more **complex workflows**
  - Multiple simulation tools
  - Parallel execution
- Raise expectations for **publishing reusable projects**
  - Reviewers can reasonably request a higher degree of reusability
- Lowering the barrier to **new resources and communities**
  - Ready platform for publishing projects
  - Ready platform for executing simulations online
  - Ability to leverage existing tools enables focus on domain problems

Demo

- **Web applications**
  - BioSimulations: https://biosimulations.org
  - runBioSimulations: https://run.biosimulations.org
  - BioSimulators: https://biosimulators.org
- **Docker image, command-line, Python tools**
  - https://tutorial.biosimulators.org
  - Executing simulation projects
  - Executing individual simulations
  - Introspecting models

Questions/discussion

Week's agenda

- **Wednesday** 2pm UTC
  - – Introduction: Goals, concepts and demos
- **Thursday** 2pm UTC
  - – Hacking: work with the developers
- **Friday** 4pm UTC
  - – Discussion: Community and input

Collaborate with us!

Reach out to work together

info@biosimulations.org

eagmon@stanford.edu

github.com/biosimulations/biosimulations

github.com/vivarium-collective/vivarium-core

Participate in weekly meetings

BioSimulations: Thu 11 am EDT

https://bit.ly/biosimulations-team-meeting

Vivarium

Questions/discussion

Week's agenda

- **Wednesday** 2pm UTC
  - Introduction: Goals, concepts and demos
- **Thursday** 2pm UTC
  - Hacking: work with the developers
- **Friday** 4pm UTC
  - Discussion: Community and input

Collaborate with us!

Reach out to work together

info@biosimulations.org

eagmon@stanford.edu

github.com/biosimulations/biosimulations

github.com/vivarium-collective/vivarium-core

Participate in weekly meetings

BioSimulations: Thu 11 am EDT

https://bit.ly/biosimulations-team-meeting