Modular Design of Multi-scale Models

Joseph Masison[1]*, Yu Mei[1]*, Jonathan Beezly[3]*, Brian Helba[3], Henrique de Assis[1], Bandita Adhikari[1], Luis Sordo Vieira[2], Michael Grauer[3], Ning Yang[4], Yogesh Scindia[4], Will Schroeder[3], Borna Mehrad[4], Reinhard Laubenbacher[1,2]

* Joint first authors, 1 Center for Quantitative Medicine, UConn Health. 2 Jackson Laboratory for Genomic Medicine. 3 Kitware Inc. 4 Division of Pulmonary, Critical Care, and Sleep Medicine, Department of Medicine, University of Florida.

INTRODUCTION

Reproducibility, usability, and extendibility of multiscale mathematical models are limited by transparency of the model structure, level of documentation, and dependencies across model components. Coding a model from scratch is often more time efficient than adding new biology to a pre-existing implementation. High turnover rate of personnel in academic labs and the collaborative nature of science necessitate a solution if models are to be reused and added to. Here we present that solution applied to an agent-based model of the innate immune response to a fungal infection of the lung, specifically invasive aspergillosis.

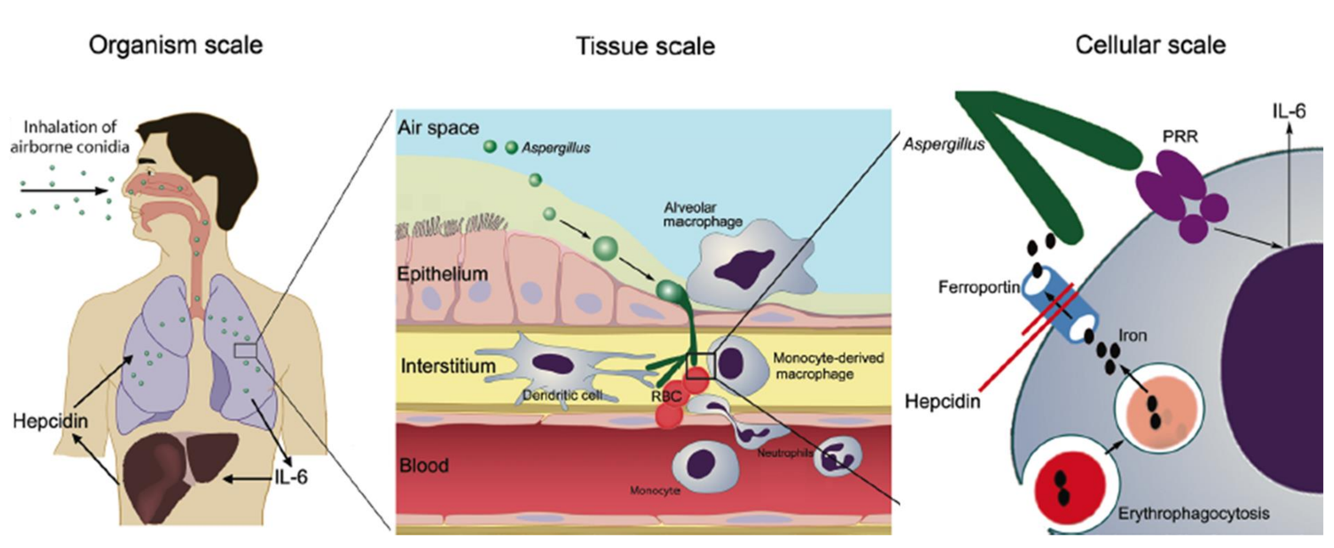

Figure 1. Invasive Aspergillosis. Pulmonary invasive aspergillosis occurs when inhaled *Aspergillus fumigatus* spores establish a foothold in the lungs. The body's initial defense consists of the alveolar epithelial cells and the innate immune system. Multiscale modeling of this disease will help determine features that can give the host an advantage in killing the fungus. Adapted from [1].

MODULAR PRINCIPLE

1. Separate model dynamic processes from each other into self-contained modules
2. Allow process communication through a global shared state variable, not directly process to process

New software design components:

1. **Global Shared State variable**
   - Contains only simulation data and parameters
2. **Modules**
   - Contain a sub-model of one dynamic process used to update the global state

RESULTS

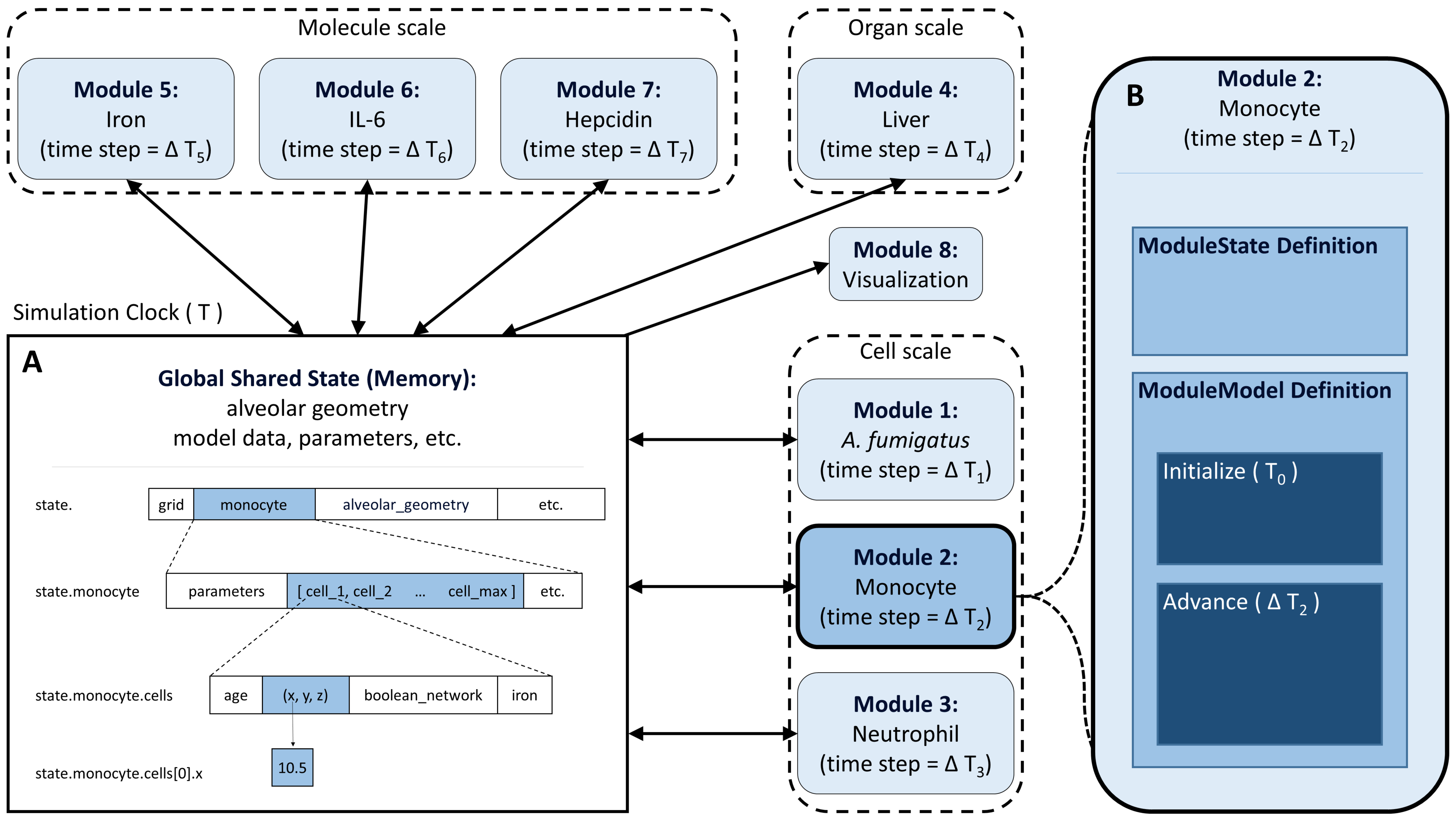

Figure 2. Modular Principle. A. We implement simulation data (Global Shared State) as blocks of memory ("state") shared amongst modules. Agent data and model parameters (monocyte data) are appended to this memory ("state.monocyte"). **B.** An implemented 'module' consists of two parts, a state definition (ModuleState) and the sub-model definition (ModuleModel). Within the state definition is the template for memory allocation for data appended to the state as in **A**, independent of the values. The module initializes these data based on a flexible user configuration, and alters the data at each time step according to its sub-model.

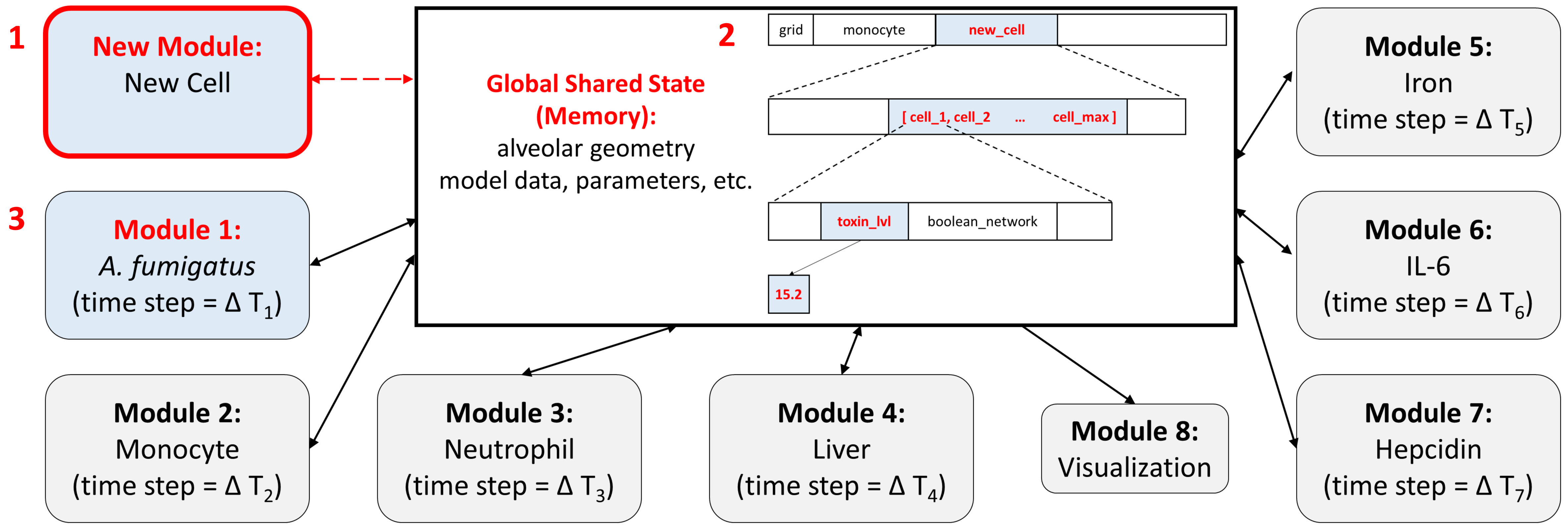

Figure 3. Adding a Module. The addition of biology corresponds to addition or alteration of a module. To add a module, a developer must simply follow the components marked in red to 1) code their new biological model (e.g. a new cell that produces a toxin to the fungus at cell_N location) 2) define new state data related to the new cell and 3) update any modules to reflect the new biology (e.g. *A. fumigatus* reads data to detect toxin and lower health).

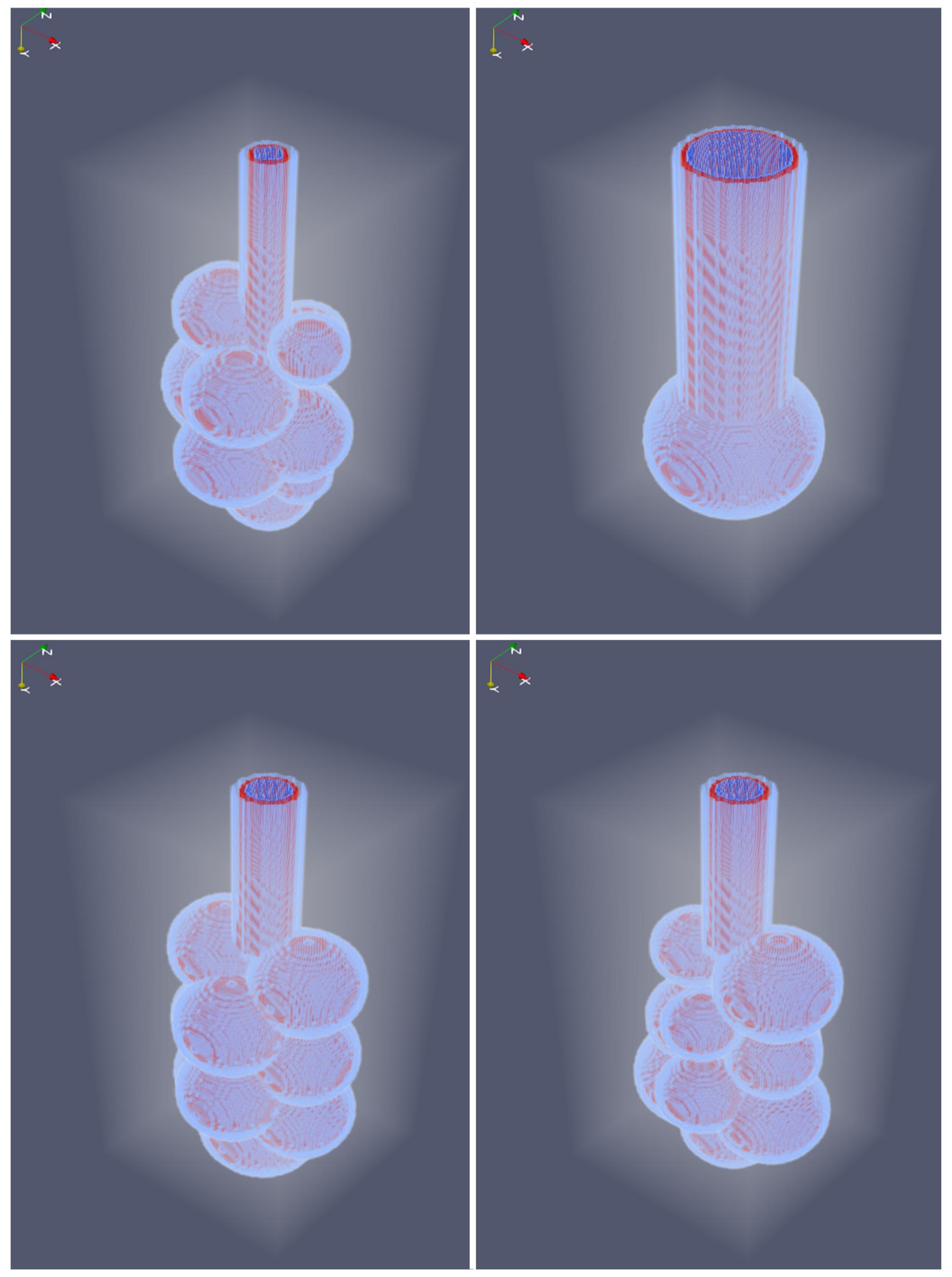

Figure 4. Versatile Simulation Space Geometry. There is no direct dependence of the modules on the data their sub-models use, including the alveolar geometry. As a result, many geometries can be created and tested without altering any of the modules.

DISCUSSION

Major advantages of the modular implementation:

- Minimizes sub-model interconnections
- Makes model and data dependencies transparent
- Relocates functional properties from tissue geometry into modules
- Makes adding, deleting, or replacing modules and changing tissue geometry easy
- Allows simulation of multiple biological scenarios
- Wide range of applications in modeling disease
- Allows for crowdsourcing of modeling efforts

TEAM

UCONN HEALTH · UF UNIVERSITY of FLORIDA · Kitware · The Jackson Laboratory

FUNDING

1. NIBIB 1U01EB024501
2. NSF EAGER #1750183
3. NIAID 1R01AI135128

1. Park, S. J., & Mehrad, B. (2009). Innate immunity to Aspergillus species. *Clinical microbiology reviews*, *22*(4), 535-551.