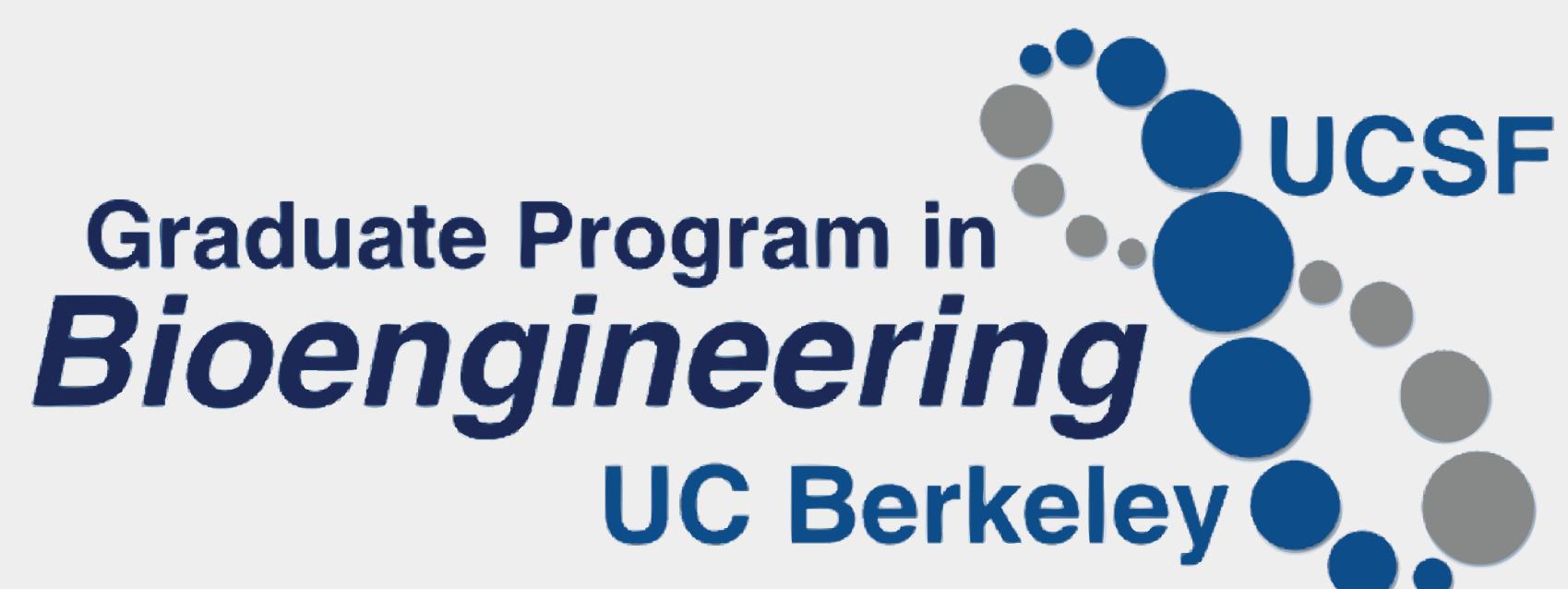


# Mechanisms as modules: developing a generic pharmacodynamic response module for use within object-oriented biological models

Brenden K. Petersen, Glen E.P. Ropella, and C. Anthony Hunt

Bioengineering and Therapeutic Sciences, University of California, San Francisco, CA



## Introduction

**Background.** Hepatocytes often respond to xenobiotic (e.g., drug) exposure by up-regulating xenobiotic metabolizing enzymes (enzyme induction). The intracellular event types are common to many pharmacodynamic responses. Elsewhere, cells often respond by down-regulating xenobiotic targets (enzyme elimination).

**Need.** Computational models of pharmacodynamic response mechanisms for use within a broad range of biological models.

**Aim.** To develop a generic, cellular pharmacodynamic response module from existing in silico liver (ISL) mechanisms. The software objects will be:

- Exchangeable—in that they can be “plugged into” or “unplugged from” a cell or exchanged for a different module without recoding the cell (Figure 2A).
- Modular—in that they can be reused by other models with minimal recoding (Figure 2B).

**Context.** These methods are aimed at object-oriented biological models. Such models are often multi-scale and designed to have a long lifetime. Mechanistic hypotheses are tested by performing in silico experiments, which are then validated (or falsified) against wet-lab experimental data. Models are then iteratively refined to meet similarity criteria for an increasing set of targeted attributes.

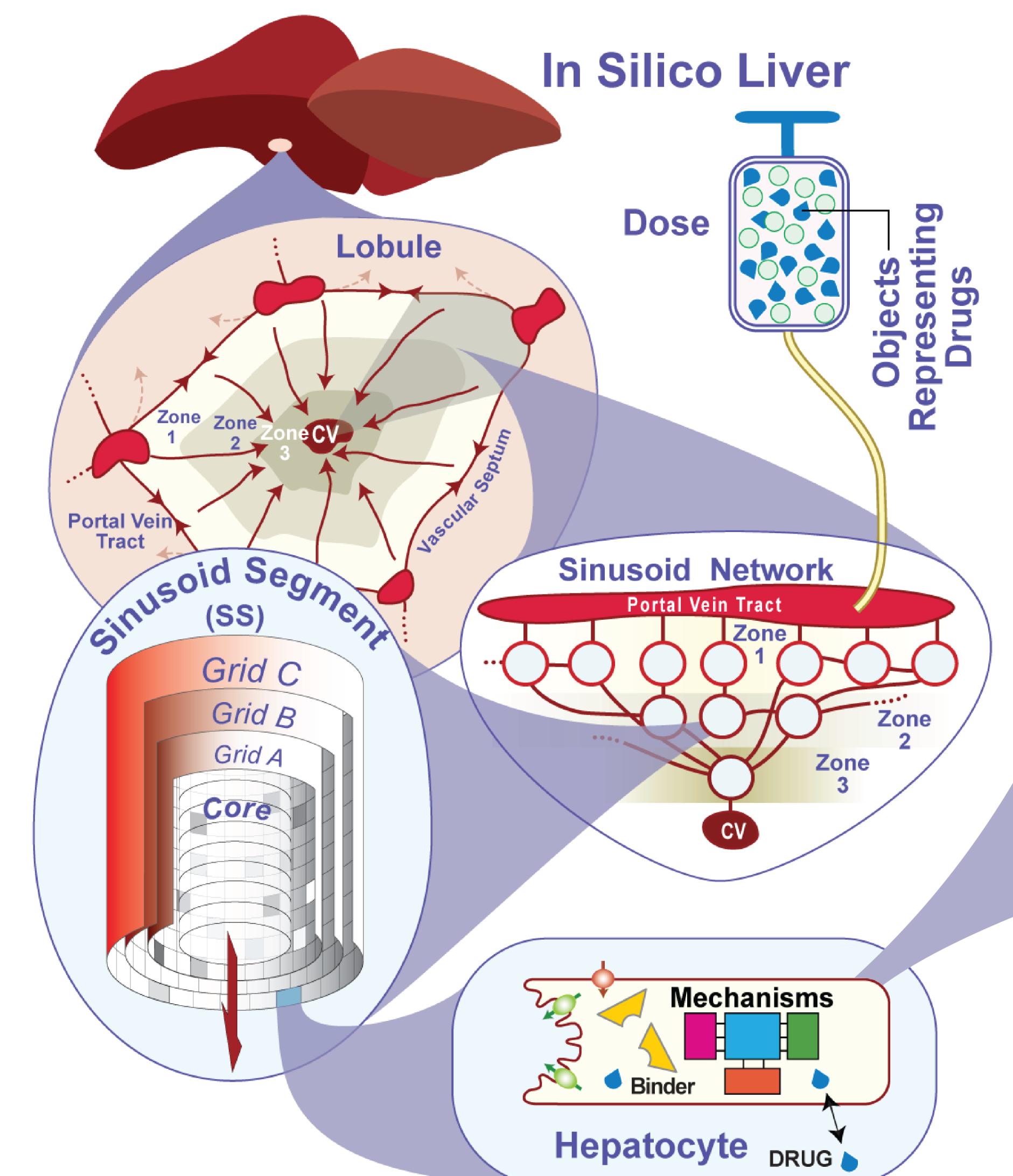
## Modular mechanisms

**Requirements.** In a modular biological model, exchanging of one version of a mechanism for another requires no recoding of the cell object. Similarly, using a completely different cell type with the same mechanisms only requires defining the appropriate groups of state information.

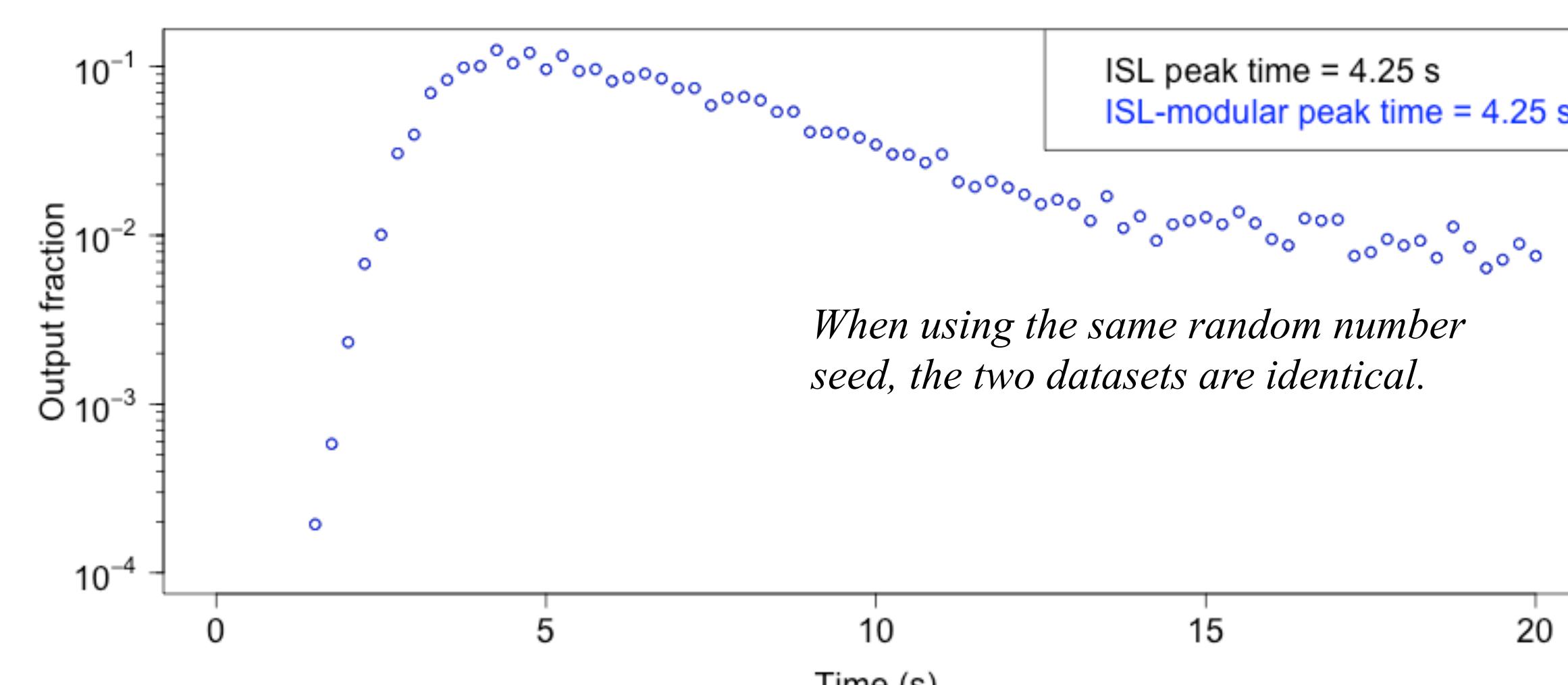
### Benefits of modularity:

- Modules and components can be used without *a priori* knowledge of each other.
- Modules enable concrete component-to-biological counterpart mappings.
- Modularity facilitates version control, exchangeability, and adaptation of past/future experimental designs.
- Modularity facilitates scaling and changing granularity to represent different systems (e.g. *in vitro* to animal, normal to diseased).
- Modularity enables validation of individual components: testing in isolation or within the context of the rest of the model.

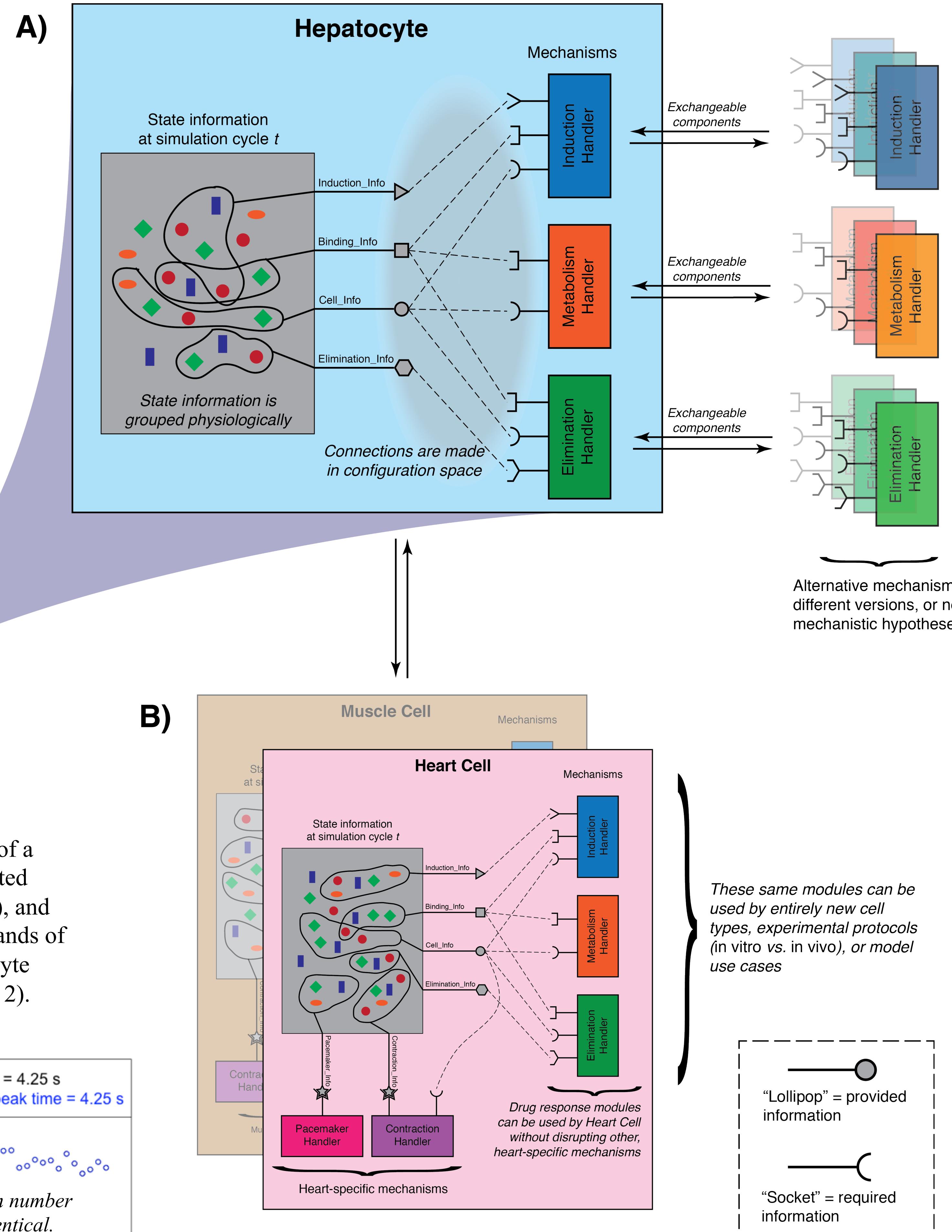
## Results and Schematics



**Figure 1.** Components of the in silico liver (ISL). One portion of a liver lobule is represented as a sinusoid network of interconnected nodes and edges. Each node represents a sinusoid segment (SS), and edges represent direction of fluid flow. Each SS contains thousands of Hepatocyte analogs contained within distinct layers. A Hepatocyte contains state information and mechanism modules (see Figure 2).



**Figure 3.** Typical outflow profile of the ISL before and after modularization. The results of the two experiments are identical (blue ISL-modular points completely overlap black ISL points), verifying that the modularization process does not affect computational results. Each experiment is an average of 8 Monte Carlo trials. Both experiments use the same random number seed.



**Figure 2.** Component diagram of the Hepatocyte software analog. A) State information is grouped physiologically and exposed to mechanism modules as interfaces. These mechanism modules are easily exchangeable with alternative mechanisms, different versions, or new mechanistic hypotheses. B) Alternatively, these same mechanisms can be used by entirely new cell types, experimental protocols (*in vitro* vs. *in vivo*), or model use cases. The only requirement is that the “client” cell (e.g. Heart Cell) exposes the appropriate groups of state information.

## Methods

**Existing in silico liver (ISL) model.** The existing ISL is a discrete-event, multi-scale, agent-based model [1]. It is written in Java, utilizing a multi-agent simulation toolkit called MASON.

**Developing a pharmacodynamic response module.** A software entity’s state information is partitioned into physiologically relevant groupings. For example, a Hepatocyte’s Binding\_Info contains a list of intracellular drug objects, enzyme objects, and a map of bound drug-enzyme pairs. Each group is then exposed as a software interface, which is recognized by internal mechanisms (e.g. Metabolism Handler) or external objects (e.g. neighboring Hepatocyte) without *a priori* knowledge of the entity.

## Future milestones

**Publish/subscribe pattern (PSP).** Under this framework, objects publish properties that others may need to know about. Another object that wants to access or react to that information will subscribe by name to any published properties. This allows any two model components to interact without *a priori* knowledge of each other, bypassing the need to expose groups of state information as interfaces.

**Analog based knowledge repository.** Mechanism modules such as the ones presented here will be added to a repository containing all accumulated mechanistic knowledge obtained from computational models. The repository contains annotated records of models (current and falsified) and their mechanisms, along with records of in silico experiments. To be both useful and productive, model components within the repository should be modular. [2]

## References

1. Ropella GEP, S Park, CA Hunt. 2008. Evaluating an hepatic enzyme induction mechanism through coarse- and fine-grained measurements of an in silico liver. Complexity. 14(6):28-34.
2. Hunt CA, R Kennedy, S Kim, GEP Ropella. 2013. Agent-based modeling: a systematic assessment of uses and requirements for enhancing pharmaceutical research and development productivity. WIREs Sys Bio Med. (in review)

## Useful links

- The Virtual Liver Network. [www.virtual-liver.de](http://www.virtual-liver.de)
- The EPA Virtual Liver Project. [www.epa.gov/ncct/virtual\\_liver](http://www.epa.gov/ncct/virtual_liver)
- The Multiscale Modeling Consortium, Interagency Modeling and Analysis Group. [www.imagwiki.nibib.nih.gov](http://www.imagwiki.nibib.nih.gov)