

## What is *insilicoML*

### ***insilicoML* (ver0.1alpha)**

The dynamics of biophysical functions usually can be described by a set of ordinary or partial differential equations or IF-THEN rules. Although it is difficult to archive biological functions because they are phenomena in time domain, once they are modeled by these equations, it is possible to store them in database since models are mathematical, *i.e.* symbolic, descriptions.

To receive the full benefit of modeling, those models should include not only equations but also meta-information such as author name, model descriptions, information of the original paper in which the model was proposed, and so on to increase the availability of the database. We have been developing a markup language called "*insilicoML*" to describe mathematical equations of models as well as meta-information. There are other languages, such as CellML and SBML that are also developed to describe mathematical models of biophysical objects. The format of *insilicoML* is defined according to XML by W3C.

A target biological organism, either its structure or dynamics is represented as a **module** which is characterized basically by unique ID number throughout the system, name, edges that represent the connections among modules, states representing dynamical variables, parameters, functions, update-rules which define the dynamics how states evolve in time and geometry that includes morphological information of shape, angle and position, among others. Another important idea on *insilico* modeling was called encapsulation that is a way to explicitly realize the multi-level modeling on the platform. Relationship between modules are represented by **edges**. We consider two type of relationship, *i.e.* structural relation, and functional relation.

### **Module**

A module represents either an element or a certain set of biological organisms to be modeled, *i.e.* a set of modules can be bundled into one group which can also be considered as a module. This nested structure is a realization of a way to model the biophysical structure with multilevel structure. Thus a model is composed of many modules. A module has several properties to characterize the module, such as physiological quantities like states, parameters, and definitions of update-rules. User can make and edit it on the platform. The quantities such as states and parameters, can be transmitted by **edges** through **ports**.

### **Physiological quantities**

The states represent dynamical variables in ordinary/partial differential equations or IF-THEN rules. If a module represents dynamically varying physical quantity, the module may possess the states. The functions express user-defined mathematical functions and can include nested IF-THEN rules. The parameters express constant values used in functions. A module can possess several of these attributes at the same time. All type attributes has following member variables, name, unit, data type, description. It is

important information that user can should input those member variables. If it is described correctly and minutely, user can easily understand what the *module* is and reuse it.

### **Port**

Two types of *ports* are existed, *i.e. input and output ports*. When module-A transmits its state value to module-B, the information come out from the output port of module-A and goes into the input port of module-B. Each input port should be bound to a parameter, which are used in functions or update rules.

### **Edge**

The *edges* represent the structural and functional relationships among modules. The structural relationships defined in the system are “*include*” and “*constituent*” representing hierarchical or parent-child relationship, and “*attachment*” to represent two or more modules are glued each other. The relationship of “*include*” is defined in case that an object is included by other objects physically or conceptually as hierarchical relation. For example, cell membrane includes intracellular liquid and sarcoplasmic reticulum. The relation of “*constituent*” is used in case that an object is made up of other objects as hierarchical relation. For example cell membrane is consisted by lipid and several proteins.

### **Encapsulation**

A single or a set of modules may form a module by operating class methods, *encapsulation* and *input/output ports* through which two or more modules are *functionally related* with each other by the *functional links*. *Ports* of an encapsulated module are packed in a class object which has properties describing information necessary enough for the use of the module. Once the modules are constructed using concept of *encapsulation* and *ports*, it can be reused without considering lower hierarchical systems. It allows users to use the modules which were made by other modelers easily. When such the modules are connected, *interface links* are used. *The states* are responsible for dynamics of the module. The *states* may evolve or be updated as time based on the property specified in the *dynamic-rules* which utilize the *states* of other modules that influence the state update of the module through the *functional links*.

### **Geometry and Morphology**

*Geometry* defines local angle and local position and also morphological information of the *module*. The *morphology* property of a module specifies the geometrical shape of the *module*. Since a model is composed of many *modules*, morphology of the entire model also can be built as ensemble of all modules. User can easily built large structure models with morphometric characteristics on *insilico* IDE. The morphological data can be specified by several data format such as VRML and VCAD volume data. It can be

used merely for 3D visualization, but may also essentially be used for dynamic simulation. Using these 3D data, it is able to visualize *models* such as a musculoskeletal system and diffusion of ions in any structures on the *insilico* IDE. VCAD is a software developed by Riken to reformat the object surface data used in CAD softwares into the assembly of lattice points. *insilico* IDE can import the VCAD file format (.vcar file), and can allocate update rules to each cell. The morphometric characteristics play an important role in Agent-Based Models and spatially distributed systems such as systems described by partial differential equations. We can perform simulation with visualization of resultant time series by considering geometric and morphometric properties in modules.