

# PhysiBoSS 2.0 Intracellular model

Vincent Noël<sup>1</sup>

<sup>1</sup>Computational Systems Biology of Cancer,  
Institut Curie, INSERM U900, Université PSL, Paris

September 16, 2020

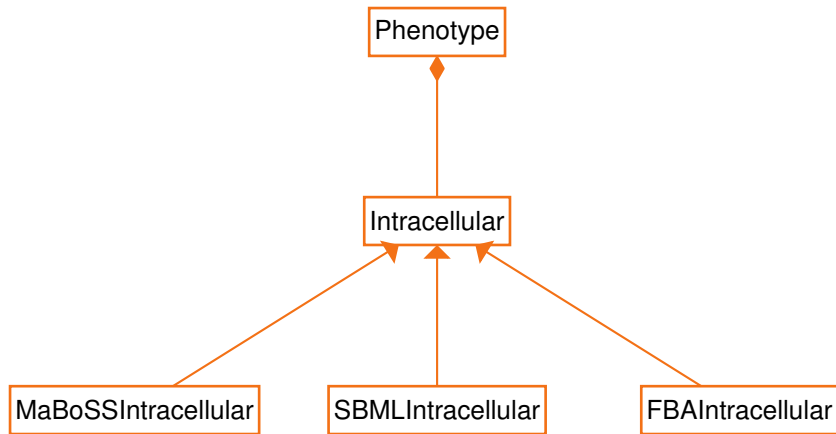
# Introduction

- › We want a cell to have an intracellular model
- › Intracellular models can be of different types : SBML, MaBoSS, FBA, ...
- › We want to be able to describe them in the XML Cell Definitions

# Introduction

```
<cell_definitions>
  <cell_definition name="default" ID="0">
    <phenotype>
      <intracellular type="maboss">
        <bnd_filename>./config/boolean_network/model_0.bnd</bnd_filename>
        <cfg_filename>./config/boolean_network/model.cfg</cfg_filename>
        <time_step>1</time_step>
        <initial_values>
          <initial_value node="A">1</initial_value>
          <initial_value node="C">0</initial_value>
        </initial_values>
      </intracellular>
    </phenotype>
    <custom_data/>
  </cell_definition>
  <cell_definition name="other" parent_type="default" ID="1">
    <phenotype>
      <intracellular type="maboss">
        <bnd_filename>./config/boolean_network/model_1.bnd</bnd_filename>
      </intracellular>
    </phenotype>
    <custom_data/>
  </cell_definition>
```

# Implementation



# Implementation

## › Phenotype

```
class Phenotype
{
private:
public:
    bool flagged_for_division;
    bool flagged_for_removal;

    Cycle cycle;
    Death death;
    Volume volume;
    Geometry geometry;
    Mechanics mechanics;
    Motility motility;
    Secretion secretion;

    Molecular molecular;

    // We need it to be a pointer to allow polymorphism
    // then this object could be a MaBoSSIntracellular, or a SBMLIntracellular
    Intracellular* intracellular;

    Phenotype(); // done

    void operator=(const Phenotype &p );
    void operator=(Phenotype &p );
    void sync_to_functions( Cell_Functions& functions ); // done
}
```

# Implementation

## › Intracellular virtual class

```
class Intracellular
{
private:
public:
    std::string type;

    // This function returns a copy of the intracellular object
    virtual Intracellular* clone() = 0;

    // This function parse the xml cell definition
    virtual void initialize_intracellular_from_pugixml(pugi::xml_node& node) = 0;

    // This function initialize the model, needs to be called on each cell once created
    virtual void start() = 0;

    // This function update the model for the time_step defined in the xml definition
    virtual void update() = 0;

    // This function checks if it's time to update the model
    virtual bool need_update() = 0;

    // This function checks if a node exists
    virtual bool has_node(std::string name) = 0;

    // Access value for boolean model
    virtual bool get_boolean_node_value(std::string name) = 0;

    // Set value for boolean model
    virtual void set_boolean_node_value(std::string name, bool value) = 0;
```

You, 4 months ago • Added virtual methods for getting and s

# Implementation

## › MaBoSSIntracellular implementation

```
class MaBoSSIntracellular : public PhysiCell::Intracellular {
private:
public:

    static long counter;

    std::string bnd_filename;
    std::string cfg_filename;

    double time_step = 12;
    bool discrete_time = false;
    double time_tick = 0.5;
    double scaling = 1.0;

    std::map<std::string, double> initial_values;
    std::map<std::string, double> mutations;
    std::map<std::string, double> parameters;

    MaBoSSNetwork maboss;

    double next_physiboss_run = 0;

    MaBoSSIntracellular();

    MaBoSSIntracellular(pugi::xml_node& node);

    MaBoSSIntracellular(MaBoSSIntracellular* copy);
```

# Implementation

## › MaBoSSIntracellular implementation

```
Intracellular* clone() {  
    return static_cast<Intracellular*>(new MaBoSSIntracellular(this));  
}  
Intracellular* getIntracellularModel() {  
    return static_cast<Intracellular*>(this);  
}  
  
void initialize_intracellular_from_pugixml(pugi::xml_node& node);  
  
void start() {  
    this->maboss.restart_node_values();  
}  
  
void update() {  
    this->maboss.run_simulation();  
    this->next_physiboss_run += this->maboss.get_time_to_update();  
}  
  
bool need_update() {  
    return PhysiCell::PhysiCell_globals.current_time >= this->next_physiboss_run;  
}  
  
bool has_node(std::string name) {  
    return this->maboss.has_node(name);  
}  
  
bool get_boolean_node_value(std::string name) {  
    return this->maboss.get_node_value(name);  
}
```



# Implementation

## › XML Cell definition parsing

```
// intracellular You, 4 months ago • Added maboss model to phenotype, extended XML

node = cd_node.child( "phenotype" );
node = node.child( "intracellular" );
if( node )
{
    std::string model_type = node.attribute( "type" ).value();

#ifdef ADDON_PHYSIBOSS
    if (model_type == "maboss") {
        // If it has already be copied
        if (pParent != NULL && pParent->phenotype.intracellular != NULL) {
            pCD->phenotype.intracellular->initialize_intracellular_from_pugixml(node);

            // Otherwise we need to create a new one
        } else {
            MaBoSSIntracellular* pIntra = new MaBoSSIntracellular(node);
            pCD->phenotype.intracellular = pIntra->getIntracellularModel();
        }
    }
#endif
}
```

# Implementation

## › Usage

```
// Creating a new intracellular model
pC = create_cell(get_cell_definition("default"));

// Updating an intracellular model
if (pCell->phenotype.intracellular->need_update())
{
    set_input_nodes(pCell);

    pCell->phenotype.intracellular->update();

    from_nodes_to_cell(pCell, phenotype, dt);
}

// Get node value
bool value = pCell->phenotype.intracellular->get_boolean_node_value("Apoptosis");

// Modify node value
pCell->phenotype.intracellular->set_boolean_node_value("TNF", 1);
```

---

# Conclusion

- › Developed with Master student Marco Ruscone from University of Turin
- › Available at <https://github.com/sysbio-curie/PhysiBoSSa>, sample project : physiboss-cell-lines
- › We might want several intracellular models within a cell (and they might communicate)
- › We also added PhysiBoSS specific components in the Jupyter Notebook GUI