User Documentation

June 2018

What is EpiLog

EpiLog is a Java software for the definition, simulation and visualisation of qualitative, logical models over hexagonal grids of cells. Such models conveniently support the study of epithelial pattern formation. EpiLog defines an hexagonal cellular automaton, where the behaviour of each cell (*e.g.,* each hexagon) is governed by the associated (logical) cellular model, subject to input signals from neighbouring cells or from other positional cues. Signalling is specified through appropriate logical functions, which qualitatively handle signal ranges and synergies.

Cellular models should be provided in the SBML-qual format, being generated using *e.g.,* GINsim (`http://ginsim.org`). Internal handling of logical models is made with the support of bioLQM (`http://github.com/colomoto/biolqm`).

Where to get EpiLog EpiLog is freely available on the web at `http://epilog-tool.org`. The source code is provided in `https://github.com/epilog-tool/epilog` under a GNU General Public License v3.0.

System requirements EpiLog is OS independent (Linux, Mac OS 10.3+, Windows) and Java version $\geq$1.7 is required.

Tutorial videos EpiLog provides tutorial videos on Youtube for the specification of each characteristic of an Epithelium `https://www.youtube.com/channel/UCGGcpepqyYJkt7dhrhncQiQ`.

Getting further support Please contact us at `support@epilog-tool.org`.

Getting started

EpiLog is a project based software. A project is defined as a series of *epithelia*, each being defined by a grid configuration, cellular models associated to the cells and simulation parameters. All cellular models and epithelia of a project are displayed on the graphic user interface, which is empty if no project is loaded, as shown in Figure 1.

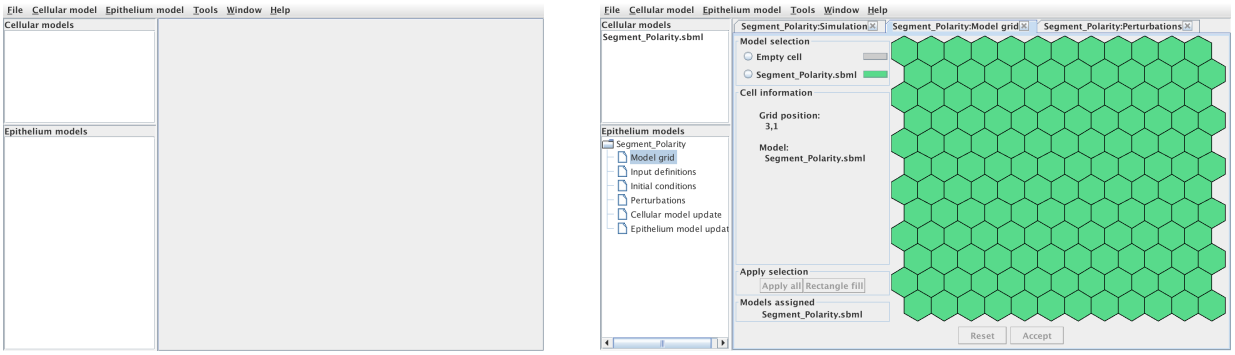

Figure 1: Main EpiLog window. Below the tool bar, the two panels on the left provide: the list of *Cellular models* stored in the project and the list of *Epithelium models*. The remaining area is reserved to the tabs related to epithelium definition and simulation operations; it is empty when starting a new project (left window) or contains tabs currently open (right window).

Load a project To load an existing project, select **`File > Load project`**. EpiLog files have the extension *peps*, each file being a compressed archive containing SBML files (cellular models) and a configuration file with the epithelium definitions. This configuration file is editable but it is highly recommended not to modify it to avoid subsequent errors.

For user convenience, recent projects are accessible by choosing **`File > Recent projects`**, which lists the latest projects opened with EpiLog.

Create a new project To create a new project, select **`File > New project`**. Once a new project is created, the first and only possible operation is to load a cellular model.

Save a project When changes are made to project, an "*" is added to the name of the project at the top bar of EpiLog. To save a project, select **`File > Save`**, or **`File > Save as`**.

> **Undo**
> EpiLog does not provide **undo/redo** functionalities; be carefully with any changes, and save your project regularly!

Operations on cellular models

A cellular model is a logical model of an intra-cellular network as defined using *e.g.*, GINsim. For details on logical models, please refer to *e.g.*, Abou-Jaoudé et al. 2016, Frontiers in Genetics. Model components are classified as internal components, which are regulated, and input components, which are not regulated and account for external signals.

Load a cellular model To load a cellular model, select **`Cellular model > Load cellular model`**. The cellular models are displayed in the *Cellular models* list. If a model component appears in different cellular models, it must have the same name, the same level range (Boolean or multi-valued with the same maximal level) and the same location (input or internal component) in those models.

Rename a cellular model To rename a cellular model, select **`Cellular model > Rename cellular model`**, or right-click on the cellular model and select **`Rename cellular model`**.

Remove a cellular model To remove a cellular model that is not used in the project (*i.e.*, not associated to any cell of any epithelium of the project), select **`Cellular model > Remove cellular model`** or right-click on the cellular model and select **`Remove cellular model`**.

Save a cellular model Cellular models are saved as SBML files by selecting **`Cellular model > Save model (SBML)`**. This allows to resort to an appropriate tool (*e.g.*, GINsim) to modify the model to be then re-loaded in EpiLog.

Replace a cellular model A cellular model of an epithelium can be replaced by another cellular model stored in the project by selecting **`Cellular model > Replace cellular model`** or by right-clicking on the cellular model and selecting **`Replace cellular model`**. The user chooses the replacing cellular model and the epithelia for which this change should apply. Note that definitions regarding components no longer present in the cellular model are discarded (*e.g.*, integration functions, initial conditions).

Operations on epithelium models

Create a new epithelium When a cellular model is loaded, an epithelium can be created by choosing **`Epithelium model > New`**, which opens a panel with the specifications of the hexagonal grid, as shown in Figure 2. The customisable parameters are:

- Dimensions: Width and height of the grid. Both must be strictly positive, the default dimensions being 15×15.
- Name: a project may encompass several epithelia, each with a unique name.
- Cellular model: By default all the cells are assigned the same cellular model.
- Rollover: This accounts for grid boundary conditions, and takes one of the following values: Rectangular (square grid, horizontal and vertical borders disconnected); Cylinder (cylindrical grid, either vertical or horizontal borders connected); and Torus (toroidal grid, both horizontal and vertical borders connected). Note that dimensions of connected borders must be even.
- Topology: This corresponds to the hexagon orientation and placement at the grid corners (*pointy* or *flat*), and the offset of every other column creating *odd* and *even* variants.

If any of the parameters is not adequate (*e.g.*, an epithelium with that name already exists, or the grid is defined as a torus with odd dimensions), the text field appears in red. Once the *OK* button is pressed, the epithelium is created and it appears in the list of *Epithelium models*.

Delete an epithelium To remove an epithelium from the project, select it in the list *Epithelium models* and select **`Epithelium model > Delete`**, or right-click on the epithelium and select **`Delete`**. A confirmation is needed for such a deletion.

Edit an epithelium To edit the epithelium name, dimensions or topology, select the epithelium in the list *Epithelium models* and select **`Epithelium model > Edit`**, or right-click on the epithelium and select **`Edit`**.

Clone an epithelium To add an identical epithelium to the list of *Epithelium models*, select the epithelium to be cloned and select **`Epithelium model > Clone`**, or right-click on the epithelium and select **`Clone`**. A new epithelium is added to the project, with "clone" appended to the name.

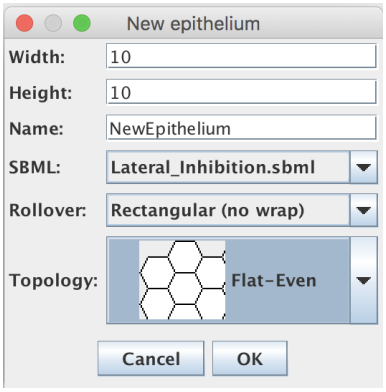

Figure 2: Pop-up window for the definition of a new epithelium. Dimensions of the grid, name of the epithelium model, selection of the cellular model to apply to all cells, rollover options (connecting the grid borders), and topology of the grid. Note that the grid dimensions must even to allow for connected borders, *e.g.*, choosing cylinder with vertical wrap requires an even vertical dimension.

Characterising an epithelium

In the list *Epithelium models*, each epithelium is displayed as a tree whose leaves correspond to the following characteristics:

- Model grid: associations of grid cells and cellular models.
- Initial conditions: initial state (internal components and positional inputs).
- Input definitions: type of inputs, and integration functions.
- Perturbations: cellular model perturbations and their associations to grid cells.
- Cellular model updating: priorities classes for updating cellular model components.
- Epithelium model updating: updating scheme of the cellular automaton.

Double clicking on any leaf creates a tab in which related characteristics can be customised. Each tab is identified by the epithelium and a characteristics name.

Saving options When a parameter is altered in a tab, this modification is registered in the project only upon confirmation by pressing the button **Accept**, otherwise a warning is issued when launching a simulation, or when quitting EpiLog, indicating unaccepted changes. If the user presses **Reset**, all unsaved modifications are discarded.

> **Project modifications**
> Accepting a modification at a tab does not permanently changes the project. A project has to be saved for tab modifications to be permanent.

Model grid If more than one cellular model is stored in the project, different cells may be assigned different cellular models (Figure 3). To identify which cellular model is assigned to a cell, each cellular model is associated with a colour. To change this colour, press the colour button and a *Colour chooser* box will appear. It is also possible to mark a cell as being "empty", meaning that no cellular model is assigned to it. The set of all cellular models present in the epithelium is shown in the *Models assigned* list.

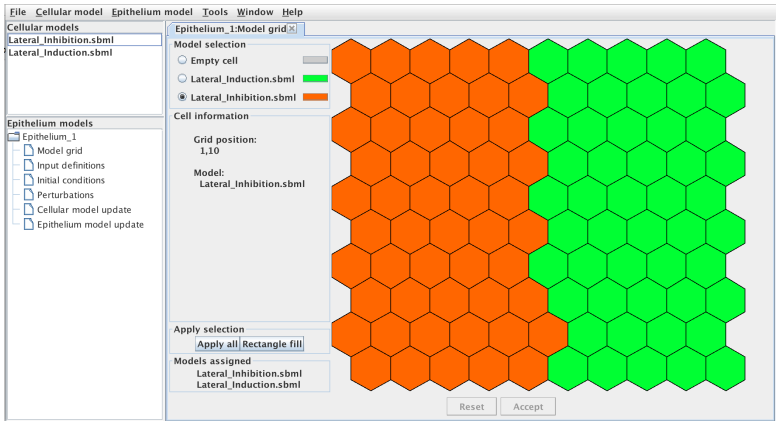

Figure 3: Model grid tab. By default, all the cells are associated with the cellular model loaded when creating the epithelium. The *Model selection* lists the cellular models stored in the project. To change the cellular model allocated to a cell, select the cellular model in the list *Model selection* and click on the cell. The cell is coloured with the corresponding model colour. A cellular model is allocated to all the cells by pressing **Apply all**. Pressing **Rectangle fill** marks all the cells inside a rectangular area defined by the diagonal between the press and release of the mouse moved on the grid. The models associated with cells of the epithelium are displayed in the list *Models assigned*.

Input definitions Selecting a cellular model in the *Model selection* combo box presents the input components of that model (Figure 4). Each input must be defined as: *Positional*, representing positional cues (*e.g.*, a morphogen discrete gradient), which are constant, and by default at level zero; *Integration*, representing signals from neighbouring cells. A logical rule, called *integration function*, must be specified for each level of an integration input (see Appendix: Grammar for the definition of integration functions). If the function is not valid, the text field is red, otherwise it is white.
Initial conditions tutorial video available: `https://www.youtube.com/watch?v=Y9sWimuFT-E`.

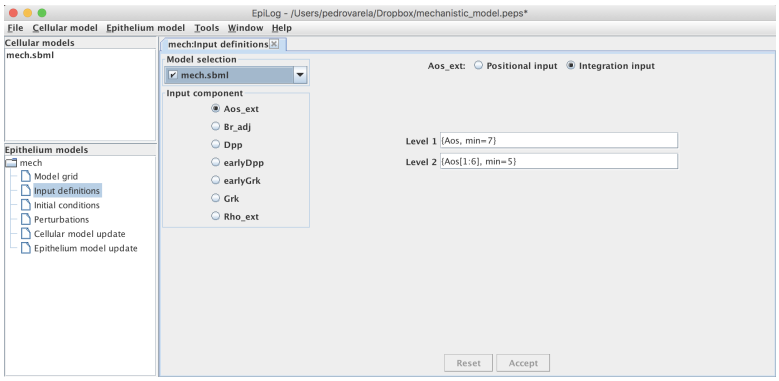

Figure 4: Input definition tab. By default all inputs are set as positional inputs. Select an input from the list *Input component*, and choose the role of that input (Positional or Integration). If it is an integration input, the integration function must be defined for each level above 0 (see Appendix: Grammar for the definition of integration functions). Note that displayed input components are those of the cellular model selected in the list *Model selection*.

Copy/Paste
On Mac OS instead of using ⌘-C/⌘-V to copy/paste, use CTRL-C/CTRL-V.

Initial conditions Selecting a cellular model in the *Model selection* combo box presents the internal components and positional input components of that model (Figure 5). To define component initial levels (zero by default), use the level selection boxes and then choose the cells for which these initial levels apply. To change the colour assigned to a component, press the colour button for a *Colour*

chooser box to appear. Note that only cells associated with the selected cellular model can be chosen. A *random initial state* can be defined to all the components or only to the checked components. By moving the mouse over the grid, cell data is displayed in the panel *Cell information*, showing which model is assigned to that cell, the levels of the components and associated perturbations if any. Information about empty cells is also displayed.
Initial conditions tutorial video available: `https://www.youtube.com/watch?v=u4PKCpUAq-w`.

Random initial state
A random initial state is applied to all cellular models on the epithelium, independently of being selected for visualisation.

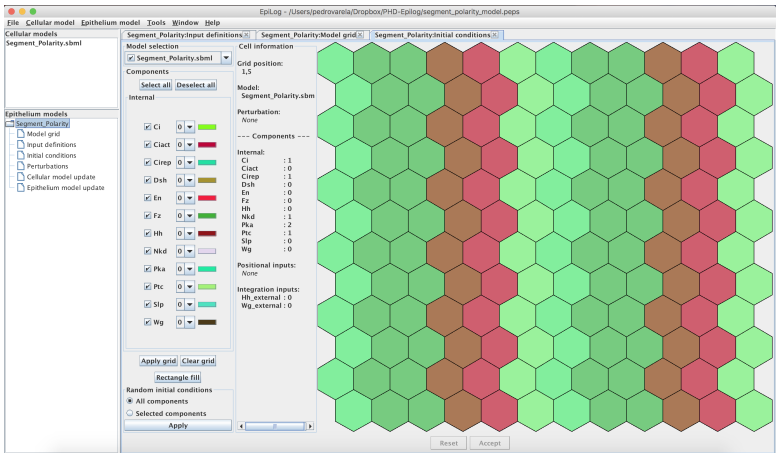

Figure 5: Initial conditions tab. By default, all the cells have their components at level zero. Clicking on a cell assigns the checked components with the selected levels. The colour of a cell is the combination of the colours of all checked components. If the chosen assignment (checked components and levels) is to be applied to all the cells, press **Apply grid**. Pressing **Rectangle fill** marks all the cells inside a rectangular area defined by the diagonal between press and release of the mouse on the grid. To set the levels of all the components to 0, press **Clear grid**. If a component is multi-valued, its colour is the highest value colour, lower levels being light toned.

Component colour
Changing a component colour applies to all epithelia, *i.e.*, the new colour applies to all initial conditions and simulations tabs of the epithelia encompassing this component.

Perturbations Perturbations can be applied to internal components of a model selected in the list *Model selection* (Figure 6). A perturbation amounts to set a fixed level or a range of levels to a component selected in the *Component* combo box *e.g.*, a knock-out corresponds to fixing a component level to 0. Single perturbations are combined to defined multiple perturbations. Each perturbation is associated with a colour, which can be changed by pressing the colour button for a *Colour chooser* box to appear. By moving the mouse over the grid, cell data is displayed in the *Cell information* panel, showing which model is assigned to that cell, the initial levels of the components and the defined perturbations. Empty cells are indicated, and perturbations can only be applied to cells associated with the selected cellular model. Choosing the cells for which a perturbation applies is done similarly to the definition of the initial conditions.
Perturbations tutorial video available: `https://www.youtube.com/watch?v=oIeTjOjqBs4`.

Cellular model update Priority classes can be defined for updating the internal components of the cellular model selected in the list *Model selection*. Components with the lowest ranked components

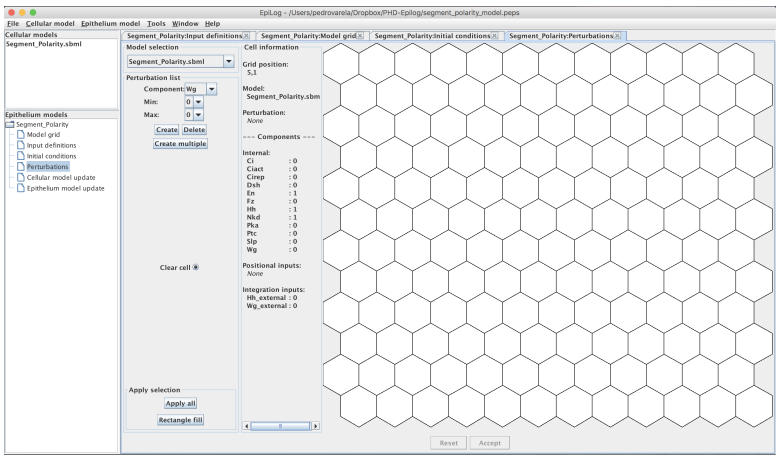

Figure 6: Perturbations tab. To create a perturbation select a component in the *Perturbation list*, set its minimum and maximum levels, and press **Create**. The perturbation is added to the *Perturbation list*. Having defined several perturbations, these can be combined: check chosen perturbations and press **Create multiple**. To delete a perturbation, check that perturbation and press **Delete**. To mark a perturbation on the grid, select the radio button next to that perturbation and press the cell where it is to be applied. If the perturbation is to be applied to all cells, press **Apply all**. Pressing **Rectangle fill** marks all cells inside a rectangular area defined by the by the diagonal between the press and release of the mouse on the grid. Selecting *Clear cell* removes all the perturbations defined for the selected cell. Perturbed cells are displayed with thicker edges.

are first considered for updating, *i.e.*, higher ranked components are updated only if higher ranked components are stable (Figure 7). Priorities can also be defined according to the type of update (level increase or decrease), by "splitting" the component, with "[+]" (increasing update), and "[-]" (decreasing update).
Priorities tutorial video available: `https://www.youtube.com/watch?v=eNqtwZklFP4`.

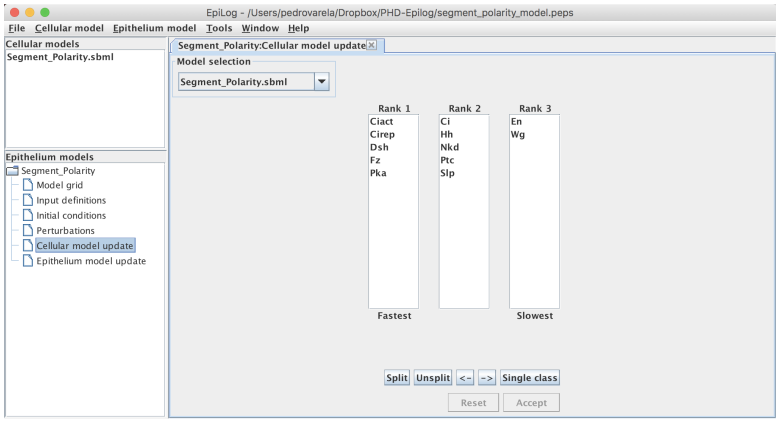

Figure 7: Model updating tab By default all the components update synchronously. To move a component to a lower/higher rank, select the component and press the **right/left arrow**. A component is split by selecting it and pressing **Split**, and unsplit by pressing any of the split version of the component and pressing **Unsplit**. Pressing **Single class** removes all the priorities.

Epithelium model update The epithelium update scheme defines the percentage of cells that are updated at each iteration (Figure 8). The alpha value varies between 0 and 1, related to a percentage of cells considered for update per iteration, where the synchronous update is defined by alpha equals to 1. By definition, if alpha is 0, a single cell is updated at each iteration, corresponding to a random asynchronous update. The percentage of cells may be defined over all the cells, or over the subset of cells that are called to update *i.e.*, only non-stable cells (at least component of the associated cellular models is called to update). Choosing a fixed simulation seed ensures that repeated simulations follow the same path, *i.e.*, generate the same patterns. Otherwise, the user can choose a random seed.

Alpha parameter tutorial video available: `https://www.youtube.com/watch?v=eSlOa7waWDk`.

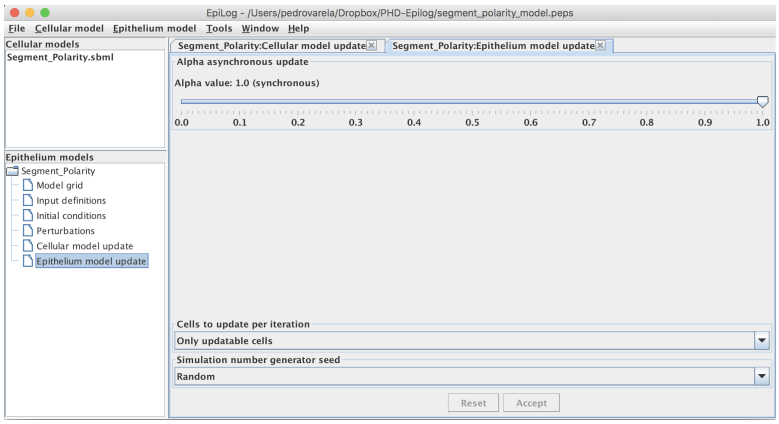

Figure 8: Epithelium updating tab Changing the value of alpha amounts to change the percentage of cells to update. If only non-stable cells are to be updated, select **Only updatable cells**, otherwise select **All cells**. To run the simulation with a fixed seed select **Fixed**, otherwise select **Random**.

Simulation

Given an epithelium the simulation allows the user to visualise the dynamics of expression patterns over the grid and to assess if a stable pattern is reached. Note that EpiLog only identifies stable states (when all the cellular model components are stable), so that oscillations are no identified as attractors.

> **Iteration**
> Each iteration is performed in two steps: updates of integration inputs followed by updates of internal components.

Run a simulation To run a simulation, select an epithelium and press **`Tools > Simulation`**, which opens the *Simulation* tab (Figure 9). Components of cellular models checked in the list *Model selection* are shown in the *Components* list. Next to each component name is the colour assigned to that component. To change the colour, press the colour button for a *Colour chooser* box to appear. Checked components are displayed on the grid, where the colour of each cell is a combination of the checked components colours. By moving the mouse over the grid, cell data is displayed on the *Cell information* panel, showing the model assigned to that cell, the levels of the components and if the applied perturbations if any. Information about empty cells is also displayed.
Simulation tutorial video available: `https://www.youtube.com/watch?v=cfk3p3tfx1I`.

Iteration navigation The navigation arrows allow the user to navigate through the simulation iterations, with a single iteration backward/forward navigation, or a burst of iterations, which by default is set to 30. The user can return to the initial state with the fast backward button. The iteration number is displayed next to the navigation buttons. When a stable pattern is found, the message "Stable grid" is displayed bellow the iteration number and no more forward iterations are allowed.

> **Terminal cyclic attractor**
> If the cellular update mode is synchronous (alpha = 1), terminal cycles are identified, displaying the message "Terminal cyclic attractor" followed by the length of the cycle (*i.e.*, the number of iterations between identical patterns).

Clone At any iteration of a simulation, a clone of the epithelium with the current state of the grid defined as initial condition can be created by pressing the *Clone* button (Figure 9). The new epithelium appears as the last epithelium in the *Epithelium models* list. The original and the cloned epithelia only differ in their initial conditions.

Saving images At any iteration, an image of the grid can be saved by pressing the *camera* button (Figure 9). A dialogue box appears to ask the preferred name and save location for the PNG file. Alternatively, all the grid states generated up to that iteration can be saved, by pressing the *double camera* button, which also opens a dialogue box and a *.zip* file is created with the images of the grid states of all the simulation iterations.

General preferences

General preferences are available by choosing **`File > Edit preferences`**. The user may choose to: display the percentages over the grid of cells with each component level, list the components in the order originally defined in the SBML file, or alphabetically (Figure 10).

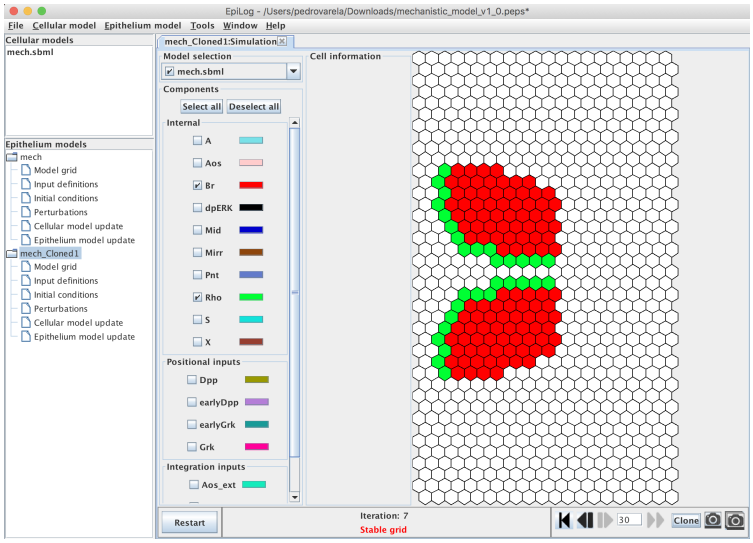

Figure 9: Simulation tab. Pressing **Select all** on the *Components* list checks all the components, *i.e.*, they are all displayed on the grid, and **Deselect all** unchecks all the components. If a stable grid is found, a message appears below the iteration number and the simulation stops. Pressing **Restart** deletes the simulation, starting a new simulation. If at any point a change is made (and saved), the user is warned but may still continue with the current simulation, until the tab is closed, or **Restart** is pressed.

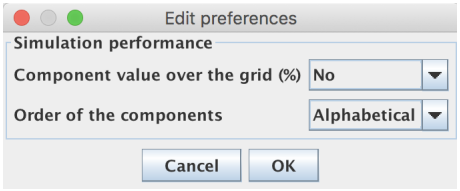

Figure 10: Preferences pop-up window: i) In the simulation and initial conditions tabs, if the "node percentage" option is selected, the percentage of cells expressing a component (for all its levels) is shown next to the component; ii) components are displayed by alphabetical order, or by the order retrieved from the cellular model.

Appendix: Grammar for the definition of integration functions

Cell-cell communication is defined by signals emitted by specific internal components, which correspond to *e.g.* secreted proteins. These signals are received by specific input components, the *integration inputs* of neighbouring cells.

An integration input is a mapping of the influences of the internal components of neighbouring cells, em i.e., cells within a specified distance. The level of an integration input is thus defined by a logical *integration function* that properly combines signals emitted by neighbouring cells.

Each signal, henceforth called *integration signal*, must be emitted from a given distance by an internal component at a given level. A *signalling function* combines integration signals, since the integration function may depend on signals from different internal components, and/or from the same component but at different levels/distances.

The specification of the integration functions must comply a grammar described below.

Integration signal

An integration signal is defined by: the emitting internal component, g; its minimum level $v > 0$; and the minimum and maximum signalling distances d_{min} and d_{max} ($d_{min}, d_{max} \geq 0$). The specification

of an integration signal receiving signal from an internal component g of neighbouring cells is of the form:

$$g : v[d_{min} : d_{max}]$$

When defining the distance range, if only one parameter is specified then $d_{min} = d_{max}$, *i.e.*, cells at exactly that distance are considered as neighbours. However, if only one parameter is specified and if it is followed by a colon, that parameter is a superior/inferior threshold (either d_{max} or d_{min}), *i.e.*, cells at a distance below/above that threshold are considered as neighbours. Note that an integration signal defines a set of neighbouring cells with internal component g at a minimum level v, within a specific signalling range.

Most often, integration signals account for signals with $v = 1$ and direct neighbours ($d_{min} = d_{max} = 1$). Hence the following default values: if no minimum value is defined then $v = 1$, and if no distance is defined then $d_{min} = d_{max} = 1$. To define an autocrine signal, $d_{min} = 0$ must be explicitly set up. Examples are provided in Table 1.

g	set of cells at distance 1 (direct neighbours), with g at minimum level 1
g:2 [1:2]	set of cells at distance 1 or 2, with g at minimum level 2
g [2]	set of cells at distance 2, with g at minimum level 1
g [:3]	set of cells at maximum distance 3, with g at minimum level 1
g:2 [2:]	set of cells at minimum distance 2, with g at minimum level 2
g:2 [0:1]	set of cells at distance 1 including the proper cell with g at minimum level 2

Table 1: Examples of integration signals specified according to the grammar (left column) and their descriptions (right column).

Note that, depending on the grid borders and the position of the cells, these have not necessarily the same number of neighbours.

Signalling term

Because the level of an integration input may depend on several signals, *signalling terms* are defined as logical functions over sets of integration signals. An example is provided in Table 2.

$g_1 \mid g_2$	set of cells at distance 1, with g_1 at minimum level 1 or with g_2 at minimum level 1
$g_1[1:2] \mid g_2$:2 [2:]	set of cells at distance 1 or 2 with g_1 at minimum level 1, or cells at distance at least 2 with g_2 at minimum level of 2

Table 2: Example of signalling terms written according to the grammar (left column), and its description (right column).

Cardinality constraints

A cardinality constraint (CC) defines the minimum and/or maximum number of cells for which a signalling term must be satisfied, and is specified as follows,

$$CC = \{\text{signaling term, min, max}\}$$

where the CC is *true* if the number of cells satisfying the signalling term is within min and max values ($min, max \geq 0$); otherwise it is *false*.

If no values are specified, then min=1, and there is no maximum; if only the value of min is defined, there is no maximum; if only max is defined, there is no minimum. Note that a CC must be defined in curly brackets. Examples are provided in Table 3.

Caution note on the values of min

If the value of min in a cardinality constraint (CC) is greater than the number of neighbouring cells, then CC is *false*. If min=0 then CC is *true*.

$\{g\}$	at-least-1 cell at distance 1 (direct neighbour) with g at minimum level 1
$\{g, min = 2, max = 5\}$	at-least-2 and at-most-5 cells at distance 1 with g at minimum level 1
$\{g, min = 2\}$	at-least-2 cells at distance 1 with g at minimum level 1
$\{g_1 \mid g_2\ , max = 2\}$	at-most-2 cells at distance 1 with g_1 at minimum level 1, or cells at distance 1 with g_2 at minimum level 1

Table 3: Examples of cardinality constraints written according to the grammar (left column), and their descriptions (right column).

Integration function

For each level $v \neq 0$ of an integration input g_{int}, an integration function specifies the conditions under which g_{int} takes this level. It is defined as a logical function f_v combining cardinality constraints CCs using the logical connectors & (AND), | (OR) and ! (NOT):

$$g_{int} = v \quad \text{if } f_v(CCs) \text{ is true, 0 otherwise}$$

The first integration function to be evaluated is that of the highest level v_{max}. If it evaluates to *true*, g_{int} is called to update to this level; otherwise, the integration function $f_{v_{max}-1}$ of the next level is evaluated and so on. If no functions evaluate to *true* then g_{int} is is called to update to level zero. An integration function can be set to *true* or *false* by using the keywords TRUE or FALSE.

Examples of integration functions are provided in Table 4.

TRUE	the integration function is *true*
$\{g_1\}\&\{g_2\}$	at-least-1 cell at distance 1 with g_1 at minimum level 1 and at-least-1 cell at distance 1 with g_2 at minimum level 1
$(\{g_1\}\&\{g_2 : 2, \max = 4\})$ $\mid\{g_1 : 2[2:], \min = 4\}$	at-least-1 cell at distance 1 with g_1 at minimum level 1 and at-most-4 cells at distance 1 with g_2 at minimum level 2, or at-least-4 cells at distance at least 2 with g_1 at minimum level 2

Table 4: Examples of integration functions written according to the grammar (left column), and their descriptions (right column).