GARMEN: GRN Agent-based Reaction-diffusion Modelling Environment

Himanshu Kaul[1,2] and Peter W Zandstra[3,4]

[1]School of Engineering, University of Leicester, Leicester, UK
[2]Department of Respiratory Sciences, University of Leicester, Leicester, UK
[3]School of Biomedical Engineering, University of British Columbia, Vancouver, Canada
[4]Michael Smith Laboratories, University of British Columbia, Vancouver, Canada

This document and uploaded files accompany the following article: Kaul et al., Virtual cells in a virtual microenvironment recapitulate early development-like patterns in human pluripotent stem cell colonies, Stem Cell Reports (2022), https://doi.org/10.1016/j.stemcr.2022.10.004

INTRODUCTION

This document contains information on how to run GARMEN to simulate peri-gastrulation patterns. GARMEN represents an integration of a gene regulatory network (GRN) embedded in an agent-based model (ABM), which, in turn, is linked with a Reaction Diffusion (RD) model. The files made available here simulate a micropatterned human pluripotent stem cell colony of radius 465 µm (colony size based on in vitro measurements, number of colonies measured = 10, refer to Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004). Patterns in the simulations emerge as a result of dynamic interplay between the GRNs and signal gradients via agent (Kaul and Ventikos, 2015, https://doi.org/10.1093/bib/bbt077) activity. Model predictions were validated in vitro via signal-based and transcriptomic perturbations (Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004).

GARMEN MODULES

FLAME: The virtual nucleus and virtual cell

The virtual nucleus and virtual cell modules were simulated using the Flexible Large-scale Agent-based Modelling Environment (FLAME). Visit www.flame.ac.uk for details regarding installation and usage.

To run a model in FLAME, one needs to create an XMML scheme describing the structure of the model files (this is the uploaded *GARMEN.xml* file). Functions or rule-sets that govern agent interactions are written in *C* (this is the uploaded *final_functions.c* file). Lastly, an XMML file containing the initial conditions is required to run the model (the *0.xml* file).

FLAME Architecture

FLAME works by parsing a model XMML definition into simulation program code via the *xparser* (xparser version used in Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004 has been uploaded). Subsequently, the message board library (libmboard) is required to compile the model (libmboard version used in this study has been provided here). The libmboard MUST be in the same folder as the model.

FLAME Contents

The following files/folders have been provided:

- **Xparser** (v 0.16.2): To parse the model (in our case the GARMEN.xml file), go in the xparser folder and use the following command line
  xparser> xparser ../model/GARMEN.xml
- **Libmboard** (v 0.2.1): Copy this folder in the model folder. The model will NOT compile, if the libmboard folder is not in the same location as the model
- The **model** files: They contain
  - **GARMEN.xml**
    This file contains the model structure. It describes the *environment* that includes the width of the simulation environment, two user-defined datatypes ‘gen_net’ and ‘chem_sensing’, and the reference to the agent functions file.
    User-defined Datatypes: The datatype ‘gen_net’ enabled the incorporation of the GRN in each agent, whereas ‘chem_sensing’ enabled the agents to sense signal concentration around them. Whereas gen_net allowed us to define the name of the genes (OCT4, SOX2, etc) and their activation status (OFF, LOW, or HIGH),

chem_sensing allowed us to define the signal (BMP4, WNT3, etc) and their concentrations.

X-machines: FLAME models agents as communicating X-machines. Here, we defined only one agent (or x-machine): hpsc (human pluripotent stem cell) that had characteristics like id, type (pluripotent, ectodermal, etc), gene activation status (gen_net), signal concentration (chem_sensing), spatial coordinates, number of neighbours (monolayer and edge bonds), and cell cycle. This file also lists the various functions these agents will perform as well as the messages they can exchange with each other.

- **final_functions.c**
  This file details the agent functions that were specified in the .xml file. It contains the source code for how agent functions will be implemented. Refer to annotations in the .c file for more details.
- **Folder 'its' containing the initial file 0.xml**
  This initial file contains information regarding the number of agents at the starting point of the simulation and their id, type, and spatial coordinates, etc. The spatial coordinates of the agents were set randomly.
- **Multiple .txt files containing concentrations of BMP4, NOGGIN, WNT3, and DKK at various time points (12 h, 24 h, 36 h, and 48 h)**
  These files contain the spatial concentrations of BMP4 & NOGGIN (12 h) and WNT3 & DKK (24 h, 36 h, 48 h) that were computed in COMSOL. When the model was simulated for the first time, these files were generated in COMSOL after every 30 iterations (equivalent to 12 h of physical time) based on the agent phenotype and position by freezing the agent-based model (refer to Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004, for details). However, we are providing these files so that the user can execute the code fully. These .txt files MUST stay in the same folder as the model.
- **Other files**
  In this folder, we are providing WNT3/DKK concentration profiles equivalent to 12 h when differentiation is induced via WNT3 alone (i.e. the BMP4− simulation). This file (12h_wnt3_dkk_data.txt) MUST stay in the same folder as the model when simulating patterning as triggered by WNT3 (in which case do not use the .txt files specified in the preceding bullet point).

Relevant Command Lines

To parse the GARMEN.xml file: Go to the xparser folder (wherever you decide to store it),
xparser> xparser ../model/GARMEN.xml

To compile the files: Go to the model folder,
model_folder> make

To simulate the model: Stay in the model folder,
model_folder> main 180 its\0.xml
This command will simulate the model for 180 iterations (equivalent to 72 hours of physical time). This includes the first 60 iterations where agents are undergoing proliferation (equivalent to 24 hours pre-differentiation) followed by 120 iterations capturing differentiation and pattern-

emergence (equivalent to 48 hours following induction of differentiation). All output files will be stored in the folder entitled 'its'.

Additional requirements

You will need to install MinGW and change the Environment Settings to add MinGW to the path. Refer to www.flame.ac.uk/docs/install.html for instructions.

COMSOL: The virtual microenvironment

We simulated the signal gradients via the RD paradigm using Gierer-Meinhardt reaction-kinetics (Gierer and Meinhardt, 1972, https://doi.org/10.1007/BF00289234) in COMSOL Multiphysics v5.2a (COMSOL, MA, USA). We used the Coefficient Form PDEs module to simulate the RD equations. In the file, entitled the-virtual-microenvironment-nonDimensional.mph, *Analytics 1, 2, 5, and 6* represent the non-dimensional equations that were solved. Refer to Equation 9 in Supplemental Experimental Procedures (Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004). D_NOG, D_BMP, D_Wnt, and D_Dkk represent the effective diffusivities of NOGGIN, BMP4, WNT3, and DKK, respectively. They were quantified such that D_BMP:D_NOG = 0.4 and D_Wnt:D_Dkk = 0.36. Refer to Supplemental Experimental Procedures (Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004) for more details regarding the RD model.

We constructed the non-dimensional virtual twin of a single well of a microtiter plate containing multiple human pluripotent stem cell colonies. We used tetrahedral meshes of size 20 µm inside the cellular colonies and 30 µm outside to discretise the geometry. Initial BMP4, NOGGIN, and DKK concentrations were set randomly with a uniform distribution. The initial concentration of WNT3 was tied to that of BMP4 (for the cases where differentiation was initiated via BMP4). We employed zero-flux boundary conditions. Following simulations, we exported data from COMSOL, assimilated it in a .txt format, and fed it to the agent-based model (relevant files have been uploaded). Data flow between the ABM and COMSOL modules was manual. For details, refer to Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004.

COMSOL Files

We are providing the file entitled *the-virtual-microenvironment-nonDimensional.mph* that contains a module each to simulate BMP4-NOGGIN and WNT3-DKK dynamics. Model steady state is achieved equivalent to 12 h of physical time. The geometry created is non-dimensional.

KNOWN ISSUES

Data exchange: The simulation will slow down at 61st, 91st, 121st, and 151st iterations because information from the .txt COMSOL files is being passed on to agents. The total execution time of the simulation, in the case where all 4 .txt data files are used, is ~60 minutes. Providing the .txt files means the simulation run will be continuous and not require intervention by the user. However, the link between the agent-based and RD models is currently manual. This means that the agent-based code will not trigger COMSOL automatically to simulate and access the signal gradients. This will need to be done manually.

FLAME_environment_variable_substrate_width: In certain compilers, this variable may get flagged as undefined during compiling. If this happens, change FLAME_environment_variable_substrate_width to substrate_width *throughout* the final_functions.c file.

Warnings: You may see some warnings when the code compiles. They will not affect the simulation, so ignore them.

Using the COMSOL data: Depending on which data files you are feeding to the model (see *Simulating Different Conditions* below), you will need to edit line 232 in final_functions.c file. For example, if you are only feeding in BMP4 and NOGGIN data at 12 h, then the line should read:

if(get_counter()==60) {

If you are only feeding BMP4/NOGGIN data at 12 h and WNT3/DKK data at 24 h, then the line should read:

if(get_counter()==60 || get_counter()==90) {

And, so on. In lines 222-229, we have defined the files that will be read at specific iteration numbers, but not all data files will be needed for certain perturbations. The latter include Patterns I, II, III, IV, XII, and Dynamic OCT4 inhibition).

SIMULATING DIFFERENT CONDITIONS

The GRN model embedded in the ABM layer allows one to simulate multiple perturbations (refer to Figure 1E in Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004) without making any alterations to the GRN.

However, we assumed that these perturbations will affect signalling in the following manner.

Pattern 0: Control, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern I: BMP4−, Only the WNT pathway is active (ensure 12 h WNT3/DKK data file is copied in the model folder)

Pattern II: WNT3−, Only the BMP4 pathway is active (ensure 12 h BMP4/NOGGIN data file is copied in the model folder)

Pattern III: BMP4− and WNT3−, no active pathways (ensure no data files are copied in the model folder and lines 205-334 in the function *signal_sensing* in final_functions.c are commented out)

Pattern IV: OCT4−, Only the BMP4 pathway is active (ensure 12 h BMP4/NOGGIN data file is copied in the model folder)

Pattern V: TBXT−, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern VI: SOX2++, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern VII: SOX17−, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern VIII: SOX17++, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern IX: SOX2−, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern X: OCT4++, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern XI: CDX2−, Both BMP and WNT pathways are active (ensure 12 h BMP4/NOGGIN and 24-48 h WNT3/DKK data files are copied in the model folder)

Pattern XII: CDX2++, Only the BMP4 pathway is active (ensure 12 h BMP4/NOGGIN data file is copied in the model folder)

BMP4 inhibition at 24 h: We took 120.xml file (equivalent to 24 h post-differentiation of physical time) from the control (Pattern 0) simulation as the initial file for this simulation. No data files were run with this simulation

Dynamic OCT4 inhibition: We assumed WNT activity was inhibited after 24 h (ensure only 12 h BMP4/NOGGIN and 24 h WNT3/DKK data files are copied in the model folder)

Refer to Figures 1E and 4H in Kaul et al., 2022, https://doi.org/10.1016/j.stemcr.2022.10.004, for pattern outcomes.

SUPPORT

Contact Himanshu Kaul at himanshu.kaul@leicester.ac.uk for queries related to this model
FLAME related queries should be directed to flame-dev@googlegroups.com

LICENSE

REFERENCES

Gierer, A., and Meinhardt, H. (1972). A theory of biological pattern formation. Kybernetik 12, 30–39. https://doi.org/10.1007/BF00289234

Kaul, H., and Ventikos, Y. (2015). Investigating biocomplexity through the agent-based paradigm. Brief. Bioinform. 16, 137–152. https://doi.org/10.1093/bib/bbt077

Kaul, H., et al. (2022). Virtual cells in a virtual microenvironment recapitulate early development-like patterns in human pluripotent stem cell colonies, Stem Cell Reports. https://doi.org/10.1016/j.stemcr.2022.10.004