Life Sciences Use Case: Requirements, Scenario Definitions and Initial Evaluation Report
Work Package 1 Task 1.1-1.4 Deliverable 1.3

Authors

Montagud, Arnau – BSC
Ponce de León, Miguel – BSC
Pradas, Gerard – BSC
Saxena, Gaurav – BSC
Vicente, David – BSC
Valencia, Alfonso – BSC
Atsidakou, Alexia – NCSR
Michelioudakis, Evangelos – NCSR
Artikis, Alex – NCSR
Monti, Michele – CRG/IIT
Tartaglia, Gian Gaetano – CRG/IIT

Distribution list:

Group:	Others:
WP Leader: BSC Task Leader: BSC	Internal Reviewer: NCSR Demokritos (NCSR) INFORE Management Team INFORE Project Officer

Document history:

Revision	Date	Section	Page	Modification
0.1	25/05/2020	All	All	Creation of the document and drafting of the sections
1	05/06/2020	All	All	First version completed
2	29/06/2020	All	All	Second version completed

Approvals:

First Author:	Arnau Montagud (BSC)	Date:	29 June 2020
Internal Reviewer:	Elias Alevizos (NCSR)	Date:	29 June 2020
Coordinator:	Antonios Deligiannakis (Athena)	Date:	29 June 2020

Table of contents:

1 Executive Summary

This deliverable describes the requirements and scenario definitions of the Life Sciences use case. We introduce the requirements of the multiscale model framework, termed PhysiBoSSa, and justify our choice of selection of this framework. Later, we detail the framework's components: the agent-based, the environment, the signalling network and the cell cycle components. Additionally, we explain how integrating these in a physics-based cell simulator allows us to study different aspects crucial for the development and growth of tumours and how, given the biophysical, biochemical, and biomechanical factors present, multiscale model can help identify the factors that drive a given treatment to be a success or a failure.

The Life Sciences scenario includes the study of two drug resistance-related biological models with different cell signalling networks: the first one is a study of different drug regimes using TNF and the second is a study of drug combinations using the AGS gastric cancer cell line. Using these biological models and our PhysiBoSSa, we replicate experimental data of growth profiles of cancer cells treated with different drug regimes. The ultimate goal of this use case is to provide a "virtual laboratory" for studying cancer growth and evolution by using multiscale models of tumour systems. The development of such a framework facilitates the design, test, and optimisation of cancer treatments based on combinations of different drugs and dose scheduling strategies.

These simulations are being scaled up by several orders of magnitude by parallelising the code in a hybrid OpenMP-MPI implementation, aiming to scale up simulations of cancer cell 3D spheroids up to a billion cells using high-performance computing. These scaled-up simulations using the Barcelona Supercomputing Center (BSC) MareNostrum4 will incorporate forecasting techniques for various events of interest, as well as techniques to reduce uninformative simulations. Moreover, we present a model exploration technique that allows us to study the structure and hierarchy of the model's parameters and to evaluate its sensibility to the parameters' perturbation.

All these developments will facilitate the design of different set-ups that tally cancer tumour growth conditions with increased number of cells, altered microenvironmental physical properties, different cell types, as well as, study the interaction between cancer cells and the immune system.

Lastly, we present the Initial Evaluation Report from expert users, including the results of INFORE prototype on the available data streams.

2 Introduction

Critical to the development of new cancer treatments is understanding the mechanisms underlying the emergence of cells resistant to different drug treatments, as well as discovering synergistic combinations of drugs that reduce the chance of this event. Disentangling the emergence of drug resistance is challenged by the inherent complexity of biological systems, which are characterised by a deep interplay among processes that occur at different scales (Kim *et al.*, 2018). Indeed, the emergence of drug resistance is determined by complex molecular mechanisms that ultimately lead to a reduction of the effectiveness of a particular drug, as well as various types of dynamic processes concerning large populations of cells (Brady *et al.*, 2017) Consequently, multicellular systems' dynamics such as tumour growth and evolution can only be understood by studying how the individual cells grow, divide and die, and their interactions at the population level (Anderson *et al.*, 2006). The simulation of tumour growth and evolution requires modelling multiscale processes, bridging the gap between different levels of description, and connecting events that occur at different scales (An, 2010). These simulations can address the modelling of a monolayer of cells, such as the ones observed in *in-vitro* cell growth in Petri dishes; the modelling of a sphere made up of 3000 cells surrounded by matrigel, called spheroid; and the modelling of organoids of cancer cells, such as the ones present in xenografts, with their associated array of immune cells, blood vessels and a detailed extracellular matrix. In spite of great advances in the field, in order to address organoids' simulation, effective tools that scale-up their simulations outputs and include such complex biological and physical set-ups are still much needed.

Multiscale models (MSM) are needed to model dynamics with very different timespans, as they can integrate cell signalling, cell-cell and cell-environment behaviour, which take place at different time scales and use different mathematical approaches. For instance, multiscale modelling frameworks combining agent-based and Boolean models are useful as they can bridge from genes' activity to cells' phenotypes, to physical interactions among cells or cells with their environment. Agent-based models represent cells as single agents of a cell population and account for the interactions between cells, small diffusing molecules and the environment. These agents can move, grow, divide and stick to their neighbouring cells and environment. Agent-based models have been successfully used to explore tumour spheroids and tumours boxed in ducts (Ghaffarizadeh *et al.*, 2018), to study defibrillation of a human heart in arrhythmia (Bernabeu *et al.*, 2010) and liver regeneration (Hoehme *et al.*, 2010). Boolean models, on the other hand, account for signalling pathways, cell cycle and cells' response to external signals that are integrated and can drive the cell to behave in a given manner: proliferate, migrate, divide, etc. Boolean models have been successfully used to predict mutants' effect on cancer phenotypes (Cohen *et al.*, 2015), cell sensitivity to drugs and the synergistic effects between pairs of drugs (Flobak *et al.*, 2015). Multiscale models are, thus, a promising genotype-to-phenotype mapping framework, which allows studying different kinds of variations and their effect on the cell's individual and collective behaviour.

Models have been used to study genetic variations by evaluating the effect of all knock-out mutants and over-expression of genes in cells' behaviour (Montagud *et al.*, 2017). Likewise, environment variations' effects on cells' behaviours have also been studied using models with interesting results. MSMs allow for the study of tumour growth and evolution, bridging the gap between different levels of description, and connecting events that occur at different scales (An, 2010). However, due to the uncertainties regarding the underlying biology, MSMs involve a high number of parameters that need tuning (Ozik *et al.*, 2018). Indeed, MSMs would benefit from the thorough exploration of the combination of these perturbations. To that end, high-performance computing (HPC) clusters, such as the Barcelona Supercomputing Center (BSC) MareNostrum 4 supercomputer, are ideal environments for intensive simulation of MSMs that produce extreme-scale data streams as outputs.

In the INFORE project, the Life Science use case is tasked to provide a *"virtual laboratory"* for studying cancer growth and evolution by using multiscale models of tumour systems (Trisilowati and Mallet, 2012). The goal of this use case is to facilitate the design, test, and optimization of cancer treatments based on combinations of different drugs and dosage strategies. In this deliverable we describe the multiscale model used in the Life Sciences use case, the set of applicable computational techniques, the alternative models of cell signalling and cell cycle and specific HPC implementation. We also present the different Key Performance Indicators (KPI) of this use case as well as how we have addressed them. Finally, we present an initial evaluation report that includes the results of INFORE prototype on the available data streams.

3 Requirements of the Life Sciences Use Case

To study the process of resistance acquisition by cancer cells through simulations, we use a MSM, *i.e.* a model that integrates cellular processes taking place at multiple time and space scales (*e.g.* molecular level, population level). MSM simulations represent a powerful approach to test alternative hypotheses about phenomena observed in cancer, enabling the prioritisation of optimal drug treatments (Ji *et al.*, 2017). By considering multiple scales of description in a single model, it is possible to find putative causal connections between processes acting at subcellular level (*e.g.* mutations or drug affecting the regulatory networks of a cell), and processes that occur at a population level (*e.g.* competition for resources, tumour heterogeneity). For instance, to study the relationship between tumour heterogeneity and treatment response, we require a population level description in which each individual cell exhibits certain variability in its molecular machinery (Kim *et al.*, 2018, 201). At a lower scale of description, processes taking place inside each individual cell are modelled through specific systems biology approaches, *e.g.* Boolean Logic to simulate signalling networks.

Our MSM is based on PhysiCell, a powerful lattice-free physics-based agent-based cell simulator for 2D and 3D multicellular systems (Ghaffarizadeh *et al.*, 2018) and PhysiBoSS (Letort *et al.*, 2018), a tool that merges PhysiCell with the stochastic Boolean model simulator MaBoSS (Stoll *et al.*, 2012, 2017). During the course of this project, we have refactored PhysiBoSS to bring it up to grade with the novel developments in MaBoSS and PhysiCell. This new software, termed PhysiBoSSa, for add-on, decouples the MaBoSS and the PhysiCell parts ensuring the long-time adaptability of these pieces of software. All of these tools are open-source and are released under free open-source licenses. The higher scale of description is the population level, which is modelled using the agent-based model part of PhysiBoSSa. This flexible framework is used to simulate population dynamics, while also considering spatial and physical constraints. In a PhysiBoSSa simulation, each individual agent represents a single cell that can grow, divide, or die (from necrosis or apoptosis), according to a set of physical rules and constraints, as well as inputs coming from the lower levels of description. Moreover, this agent-based framework is also able to model cell-cell and cell-environment interactions as well as the environment. This environment is represented in terms of concentration or densities of different molecules such as nutrients (e.g. O2, glucose), waste products of the cells (CO2, lactate), drugs and signalling molecules. In our MSM, the diffusion of the different molecules is governed by a system of PDEs. Moreover, each individual cell is represented by a discrete agent; the physical behaviour of the agents (e.g. movement, cell-cell contacts) is ruled by mechanical equations.

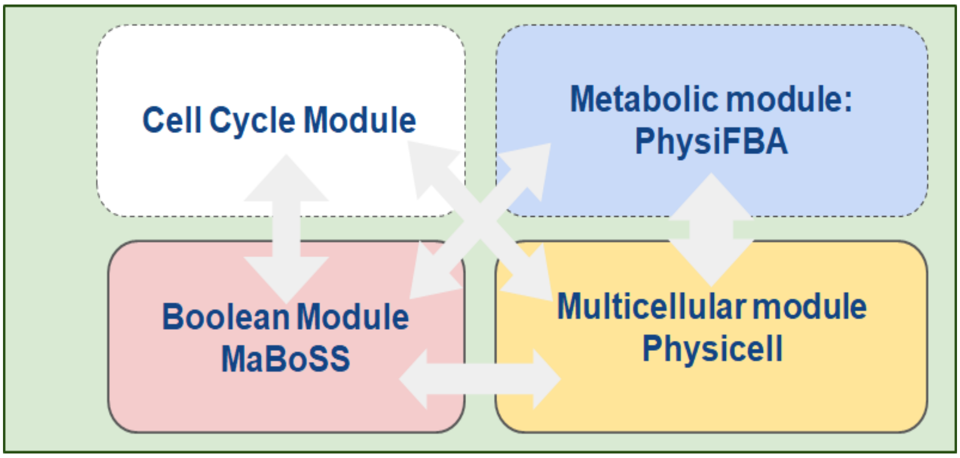
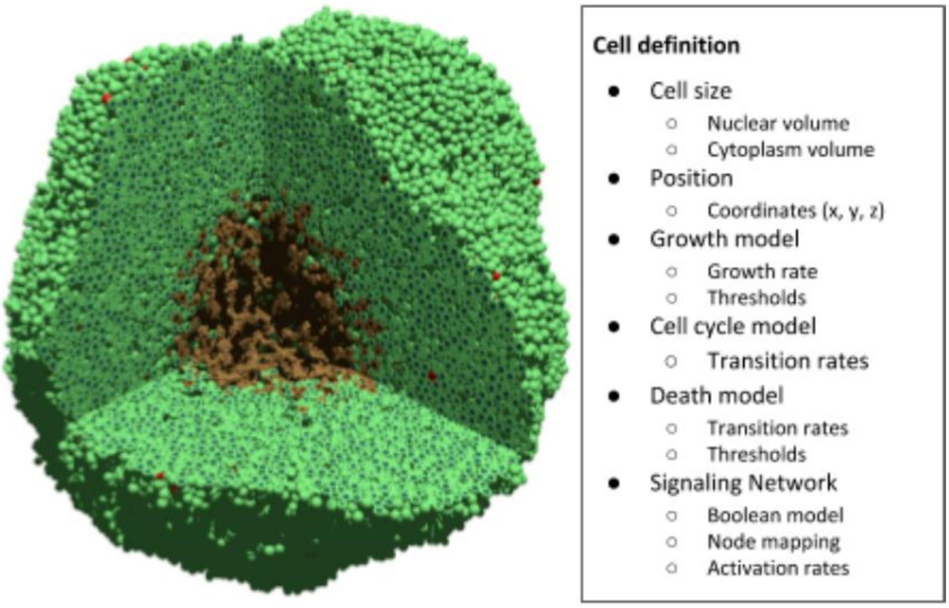


Figure 1: Scheme of the desired multiscale modelling framework. Continuous line indicates that the module has been incorporated into the framework, dotted line represents ongoing work.

This MSM allows us to connect different scales of description and to find causal relationships among them. It also allows for the introduction of different modules that simulate different parts of the MSM, like the Cell Cycle module for the cell growth, the PhysiFBA module for the metabolism, MaBoSS for the intracellular processes and PhysiCell for the environment and the agents' physics (Figure 1). In this sense, our MSM can be divided in four differentiated components: environment, agents, cell cycle and signalling. The environment component is the one that simulates all the diffusion, creation and uptake of chemical entities that roam in the environment. The agents' component is the one that takes care of the population level and simulates the cells dynamics, their growth, death, movement and overall physical behaviour among cells and among them and their surrounding environment. These first two components are simulated by PhysiCell. The signalling module is the one that takes care of the individual cells' level and simulates the behaviour of each cell in response to its environment and its neighbouring conditions. This component is simulated by PhysiBoSSa. Finally, the cell cycle module that takes care of how the cells grow and divide, is embedded into PhysiCell. An example of a typical simulation result is reported in Appendix B.

3.1 The agent-based component

The development of the agent-based model for simulating tumour growth and the responses to different drug regimens is implemented on the top of PhysiCell framework and is mostly based on its extension PhysiBoSSa. As we detail in further sections, the PhysiCell framework manages the simulation of i) the environment, ii) the cell-agent mechanics, including movement and physical interaction and iii) the basic agent behaviour by providing standard models for cell growth, division and death (**Figure 2**).



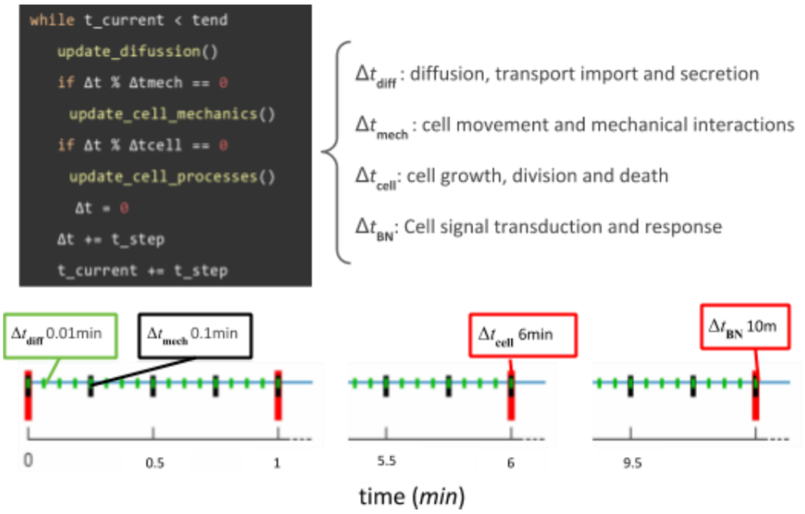
Cell definition

- Cell size
  - Nuclear volume
  - Cytoplasm volume
- Position
  - Coordinates (x, y, z)
- Growth model
  - Growth rate
  - Thresholds
- Cell cycle model
  - Transition rates
- Death model
  - Transition rates
  - Thresholds
- Signaling Network
  - Boolean model
  - Node mapping
  - Activation rates

Figure 2: Visual representation of a multiscale simulation of a population of cells. On the left side, a population of cells arranged in a 3D spheroid is depicted. Cell colours indicate cells exhibiting alternative phenotypes. For instance, the brown core at the centre of the spheroid corresponds to necrotic cells that die because of the lack of nutrients. On the right side panel, the different attributes or properties of each individual cell agent are shown.

Nonetheless, most of the behaviours of the cell agents are governed by complex rules that are not part of the basic agent-based mechanisms, as is the signalling transduction pathway, the intracellular machinery that integrate and process external and internal stimulus and transduce them into cell fate decision, e.g. start replication cycle or commit into apoptosis. In order to capture part of the intracellular processes and their dynamics, a model of cell signalling can be integrated within each individual cell-agent. This extension adds another layer to the multiscale model allowing for a more detailed description of the signal transduction process which, for instance, can be used to simulate the effect of drugs in a more realistic way. This motivated the development of PhysiBoSS (Letort *et al.*, 2018), a framework which combines PhysiCell with MaBoSS and its refactored version PhysiBoSSa.

PhysiBoSSa extends the functionalities of PhysiCell by allowing it to model and simulate the cell signalling network which processes inputs and dictates the cell fate (e.g. proliferate, commit apoptosis). Importantly, this extension adds a fourth time-scale in the model, the Boolean network time to the original three time-scales from PhysiCell: the diffusion, the mechanics and the cell division (Figure 3). At each time-scale, the engine queries the specific component and evaluates if it needs to update any of the component. The diffusion Δt_{diff}, typically 0.01 minutes, is the time where changes in the environmental entities are considered: their diffusion, reactions and transport. The mechanics Δt_{mech}, typically 0.1 minutes, is the time where changes in the cell physical behaviour are considered: cell movement, cell-cell attachment, cell-environment attachment, etc.. The cell division Δt_{cell}, typically 6 minutes, is the time where PhysiCell queries if there is any change in the cell growth, cell division and the different death modes considered (Apoptosis and Necrosis). Finally, the Boolean network Δt_{BN}, typically 10 minutes, is the time where PhysiCell queries if there is any change in the cells' behaviour as a result of the signalling pathway activation.



```
while t_current < tend
    update_difussion()
    if Δt % Δtmech == 0
      update_cell_mechanics()
    if Δt % Δtcell == 0
      update_cell_processes()
       Δt = 0
    Δt += t_step
    t_current += t_step
```

Δt_{diff} : diffusion, transport import and secretion
Δt_{mech} : cell movement and mechanical interactions
Δt_{cell}: cell growth, division and death
Δt_{BN}: Cell signal transduction and response
Δt_{diff} 0.01min
Δt_{mech} 0.1min
Δt_{cell} 6min
Δt_{BN} 10m
0
0.5
1
5.5
6
9.5
time (*min*)

Figure 3: Multiscale simulation main loop pseudo-code and the different time scales.

Thus, the population dynamics allow studying different physical properties' variations (cell-cell adhesions, cell-matrix adhesions) under different microenvironmental conditions (presence of oxygen, signalling molecules or extracellular matrix). This agent-based engine has been connected to MaBoSS to simulate the signalling network of each individual agent following the structure of PhysiBoSSa. This allows us to have a stochastic Boolean model simulator (MaBoSS) that predicts cell fates embedded in a flexible agent-based model (PhysiCell) that simulates multicellular systems. In a typical MSM simulation, each single sphere corresponds to a cell agent and the colour code indicates the cellular phenotype (Figure 2). For instance, the brown-coloured cells at the centre of the structure correspond to necrotic cells, *i.e.* cells that die as a consequence of the lack of nutrients and oxygen. On the right-hand side of Figure 2, some properties of the individual agents are listed. Notice that each individual agent holds its own signalling network, which is used to update the phenotype of the cell at each time step.

The coupling between the different scales is conducted by using output of one model as input of the other. Specifically, the availability of nutrients or the presence of drugs around a cell agent are used as inputs of the signalling model of that cell. Subsequently, the output of the signalling model is used to update the behaviour of the cell agent. For example, during a simulation, a drug pulse is introduced in the environment; when the drug reaches a particular agent, it affects the function of its protein target, which corresponds to a node within the cell agent signalling network; then, the state of the signalling network is updated to assess the effect of the drug perturbation by running the Boolean stochastic simulator of PhysiBoSSa. The new state of the signalling network is used to update the phenotype of the cell agent, which for instance might enter into apoptosis due to the drug effect, closing the loop.

3.2 The environment component

The aforementioned agents are not standing in the void, instead they are communicating with a rich surrounding environment Figure 4) with chemical entities governed by reaction-diffusion equations (Figure 4b). This environment is a dynamic one, where these entities are created, diffuse and can be uptaken dynamically by the agents or can be added from fountains or drained from sinks. For this, PhysiCell uses BioFVM, a Finite Volume Method (FVM) based simulation software, to simulate the chemical microenvironment with a vector of reaction-diffusion Partial Differential Equations (PDEs) with both bulk source/sinks and cell-centred sources and sinks (Ghaffarizadeh *et al.*, 2016).

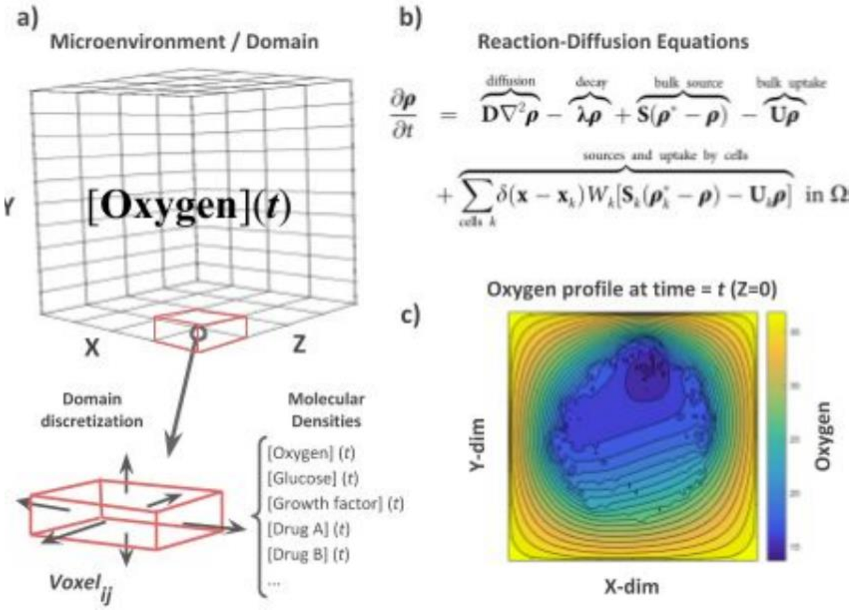

a)
Microenvironment / Domain
[Oxygen](*t*)
Y
X
Z
Domain discretization
Molecular Densities
[Oxygen] (*t*)
[Glucose] (*t*)
[Growth factor] (*t*)
[Drug A] (*t*)
[Drug B] (*t*)
...
Voxel$_{ij}$
b)
Reaction-Diffusion Equations
$\frac{\partial \boldsymbol{\rho}}{\partial t} = \overbrace{\mathbf{D}\nabla^2\boldsymbol{\rho}}^{\text{diffusion}} - \overbrace{\boldsymbol{\lambda\rho}}^{\text{decay}} + \overbrace{\mathbf{S}(\boldsymbol{\rho}^* - \boldsymbol{\rho})}^{\text{bulk source}} - \overbrace{\mathbf{U}\boldsymbol{\rho}}^{\text{bulk uptake}}$
$+ \overbrace{\sum_{\text{cells } k} \delta(\mathbf{x} - \mathbf{x}_k) W_k [\mathbf{S}_k(\boldsymbol{\rho}_k^* - \boldsymbol{\rho}) - \mathbf{U}_k\boldsymbol{\rho}]}^{\text{sources and uptake by cells}} \text{ in } \Omega$
c)
Oxygen profile at time = *t* (Z=0)
Y-dim
X-dim
Oxygen

Figure 4: Domain definition, representation and simulation of the molecular densities that define the microenvironment. a) The domain is discretised in voxels and each voxel stores the amount of each molecular density in this space point and at a given time step. b) the systems of partial differential equations that govern the time evolution of the different densities. c) an example of the visual representation of the profile of a given density (oxygen) at the final time step for a domain of size 10003. Bright yellow areas are substrate-producing areas and dark blue areas are areas with less substrate concentration.

In practical terms, this means that our MSM can include diffusible chemicals such as oxygen, nutrients and drugs, as well as patches of dense static cells that impede the advancement of otherwise migrating cells. These elements allow setting up complex microenvironments that enable modellers to tackle real-life scenarios like the ones found in cancer growth and migration. For instance, we can set up a scenario with different substrate-producing areas in the environment and dynamically track the substrate diffusion (Figure 4c).

3.3 The signalling network component

The signal transduction machinery of a cell is composed of a network of molecular components (*e.g.* protein complexes, small molecules), which allows the cell to decode different signals and adjust its internal state to respond to different stimuli (Tyson *et al.*, 2003). The interaction between the different molecular components can result in activation or inhibition and be wired in such a way that these pathways can include feedback and/or feedforward loops. Thus, a signalling pathway consists of sets of proteins and small compounds used by the cell to process different signals and modify its state in response to certain stimuli.

In Figure 5, a schematic representation of different signalling pathways involved in prostate cancer is depicted. Distinct signalling pathways are meant to process different classes of external and internal signals, such as the availability of nutrients and the space to proliferate, the level of DNA damage, and others. The transduction of a signal induces the cell to respond by changing its internal state or the phenotype of the cell.

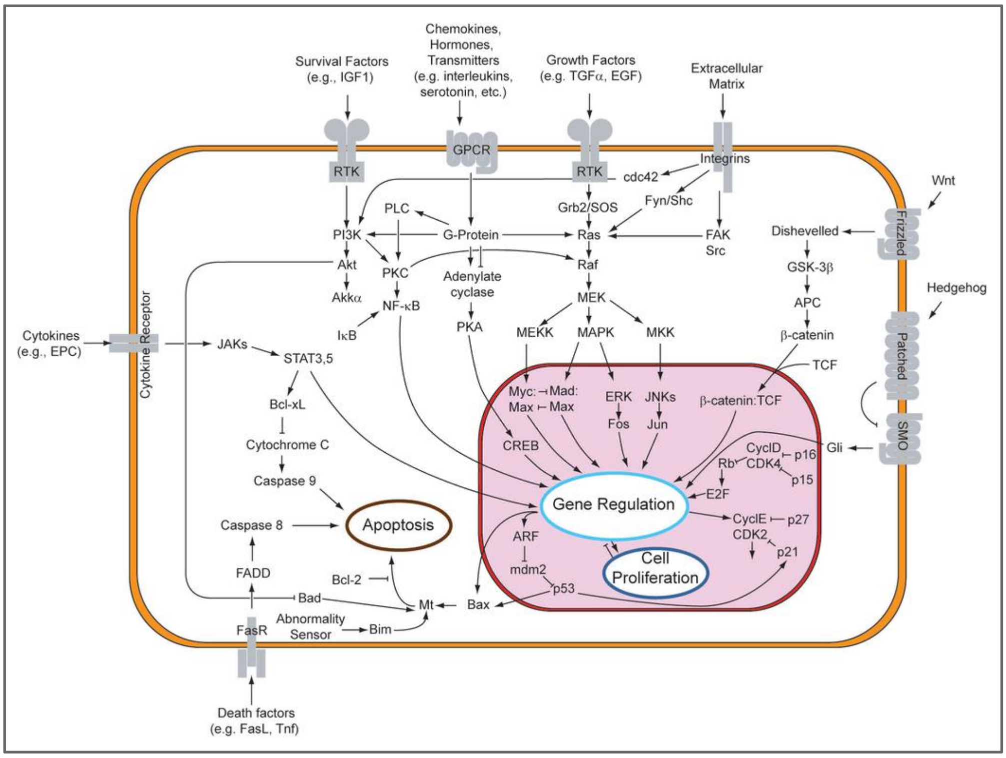


Figure 5: Schematic representation of a signalling network which includes different signalling pathways found in most human tissues. Downstream the signal transduction pathways, different effectors change the cell phase or phenotype (e.g. proliferation, apoptosis) to respond to different stimuli. Orange and red rounded rectangles represent the cell membrane and nucleus, respectively. Labels indicate protein complexes whereas arrows and T-bar lines represent activation and inhibitory interactions, respectively. Taken from https://en.wikipedia.org/wiki/File:Signal_transduction_pathways.png

Living systems tend to accumulate mutations in the DNA during their lifespan. While some mutations may not alter the molecular functions of the cell, others can cause the malfunction or dysregulation of particular cellular processes. For instance, mutations affecting pathways involved in the cell cycle can lead to undesired cell phenotypes, such as uncontrolled cell growth, *i.e.* cancer (Hanahan and Weinberg, 2011). For this reason, targeting dysfunctional signalling pathways is one of the main focus in the development of novel target therapies. Signalling networks are complex systems that exhibit non-trivial behaviour. Thus, mathematical and computational models are needed to rationalise and understand their functioning and predict novel therapies.

From a modelling point of view, a signalling network is a dynamical system governed by a set of differential equations. The structure of systems is described as a network where the nodes, representing proteins and complexes, are connected through directed signed interactions, indicating activation or inhibition (Calzone *et al.*, 2018). Additionally, each node can be found in one of two states (active or inactive). The state at a given time point is computed through an activation function which integrates the different inputs of the node. The activation function operates as a logical gate that integrates the incoming inputs of a node (from a previous time point) and gives the updated node state. It is important to clarify the difference between the signalling network model and the approach used to simulate its behaviour: the signalling network can change in different cell types, while the same signalling model can be simulated with alternative approaches.

Different mathematical approaches have been developed to simulate and analyse signalling network models. Most common approaches are based on ordinary differential equations (Wittmann *et al.*, 2009), Boolean logic (Calzone *et al.*, 2010), or stochastic approaches such as MaBoSS (Stoll *et al.*, 2012, 2017). Such simulation frameworks are used to study the dynamics of signalling networks and, in the case of cancer research, to predict the effect of single drugs and combined therapies. In this project, we simulate and analyse signalling models using the stochastic Boolean approach implemented in PhysiBoSSa, as it allows to compute the probabilities associated to the proliferative and apoptotic phenotypes.

The signalling network is subject to the physiological role performed by the cell, thus cells belonging to different tissues exhibit differences in their signalling pathways. Differences can also be found between individual cells, and even between cells belonging to the same tumour. For this reason, models of signalling networks should be calibrated or adjusted to represent a particular cell type (Béal *et al.*, 2019b, 2019a). In this project, we use a curated signalling model of the adenocarcinoma cancer cell line AGS (Flobak *et al.*, 2015) (Figure 6). The model depicted in Figure 6 accounts for most of the signalling pathways known to play a relevant role in this type of cancer (*e.g.* MAPK, PI3K/AKT/mTOR, Wnt/β-catenin and NF-κB). This model was successfully used to predict synergistic effects of different pairs of drugs that were validated experimentally (Flobak *et al.*, 2015).

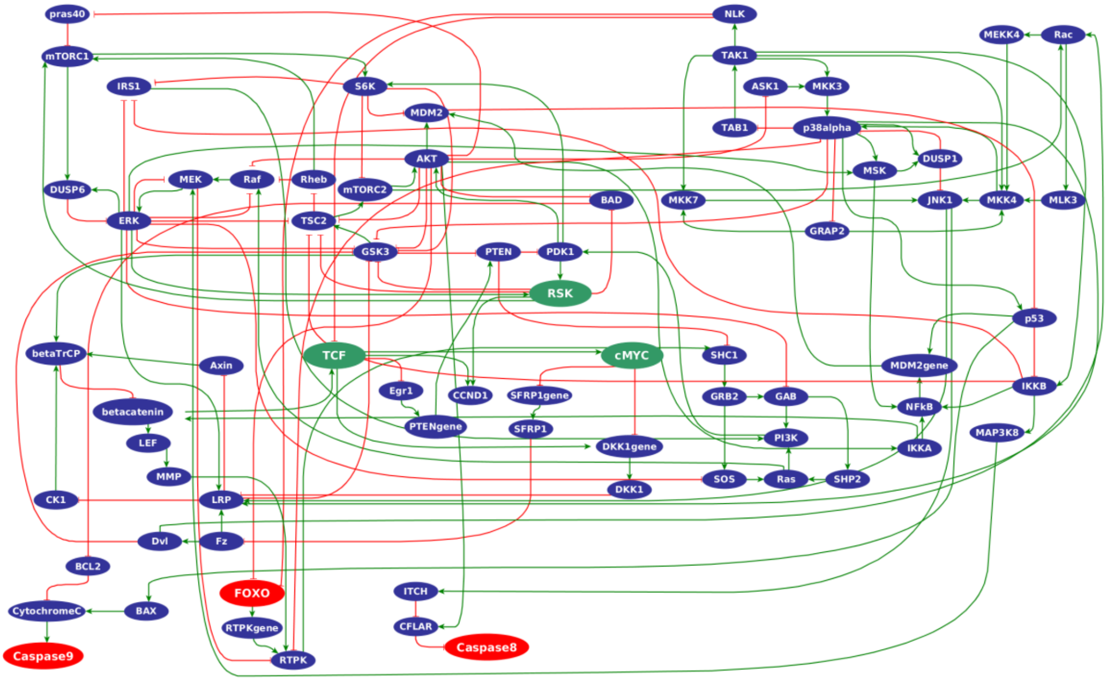


Figure 6: Network model of the AGS cell line signalling. Nodes correspond to proteins or complexes of proteins. Red and green arrows correspond to inhibitory and activation interactions, respectively. Red and green nodes indicate those nodes associated with anti-survival and pro-survival phenotypes, respectively.

The transduction of signals like the availability of nutrients, the presence of DNA damage, etc. will induce the cell to respond by changing its internal state or the phenotype of the cell like moving, growing, dividing, etc.

These transductions of signals are described with signalling networks that are complex systems that exhibit non-trivial behaviours. These networks are oftentimes studied with mathematical and computational models where the nodes, representing proteins can be active or inactive and are connected through signed interactions, indicating activation or inhibition (Calzone *et al.*, 2018). The node state at a given time point is computed through an activation function, which integrates the different inputs of the node. The model state is computed by aggregating all the node states at a given time. In this project, we simulate and analyse signalling models using the stochastic Boolean approach implemented in MaBoSS, as it allows us to compute the probabilities associated with the different phenotypes.

3.4 The cell cycle component

PhysiBoSSa include the description of 6 different models of cell cycles, these are, listed from less to more detailed: the "Live" model, the "Cycling-Quiescent", the "Ki-67 Basic", the "Ki-67 Advanced", the "Flow Cytometry" and the

"Flow Cytometry Separated". Nevertheless, all these models are based on ODEs and work more or less independently from the signalling or the metabolic components.

In this section we present the implementation of an alternative model of the cell cycle inside the simulation architecture of PhysiBoSSa. This model differs from the rest of models already present in PhysiBoSSa as it is more detailed, more realistic and its different phases have different cellular phenotypes that could be modelled using agents. First of all it is worthy to remember that nearly all the tumours have some kind of dysfunctionality related to the cell cycle. Indeed, this is the biochemical time regulator and it orchestrates the cellular phases during the growth. In turn, it is tuned on or off in respect to the external and internal cues. Thus, being able to correctly regulate the activity of the cell cycle is a critical feature for cancer development.

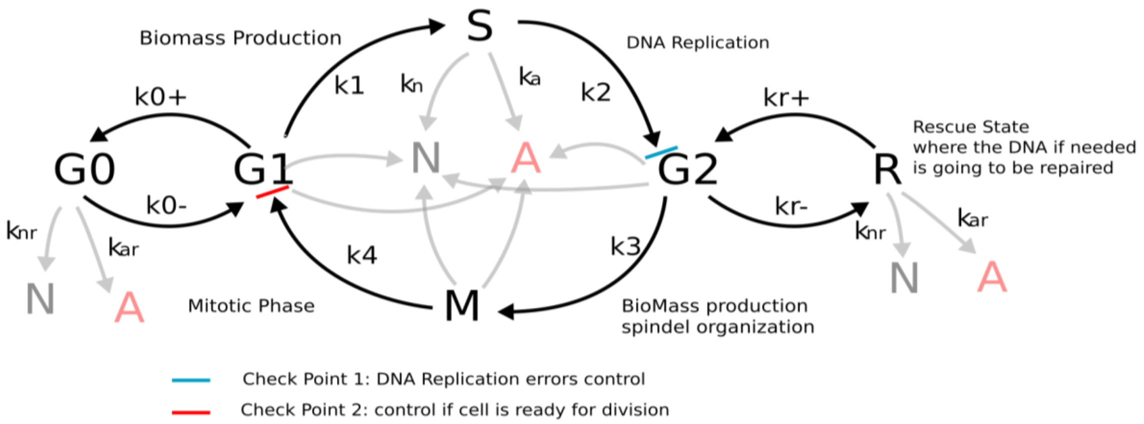


Figure 7: Cell cycle module currently considered in PhysiBoSS.

Currently, the cell cycle model implemented in PhysiBoSSa is based on a set of transitions among the different cell states G1, S, G2 and M (Figure 7). Each of these phases is responsible for a certain task of the growth, like biomass production, DNA replication or mitotic division. At the moment, there is no quantitative connection with the growth, indeed the cell growth rate is independent of its cellular phase.

Furthermore, one of the operative definitions of the cell phase is the set of genes that are expressed in that time. This means that each phase of the cell cycle can drive the metabolism in different directions that in turn allows the cell to grow or not. In order to make the model more realistic together with the implementation of metabolic models relying on flux balance analysis (FBA) approximations, we are developing a cell cycle dynamic that can drive the genes to an oscillatory behaviour, as it happens in a real cell cycle that in turn drives the metabolism.

The model here describes the oscillatory behaviour of the gene expressions and its relative transcripts during the cell cycle. First, we consider a central core of genes that are responsible for these behaviours and their checkpoints, as well as their connection with the rest of the genome. These networks have been taken from experimental and computational models from literature (Yang *et al.*, 2018; Gérard and Goldbeter, 2014) and they regulate the oscillation of the genome and the checkpoint dynamic of the cell cycle. We implemented a stochastic version of this dynamic network (Figure 8).

We aim to connect this stochastic dynamic with the genes relative to each phase and in turn connect it to the FBA module. This connection allows us to turn on and off the metabolism for different phases of the cell cycle building up a more realistic model of the growth dynamic.

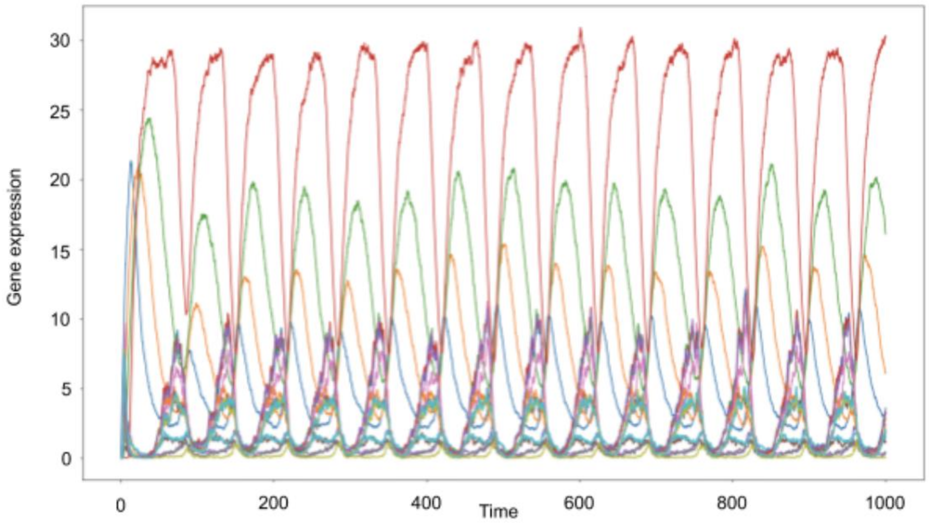


Figure 8: Gene expression levels oscillations driven by the cell cycle are reproduced using stochastic simulations coupled to a cell cycle model.

4 Scenario Definition

4.1 Life Sciences use case scenario

The Life Sciences use case explores different mechanistic explanations to help understand how cancer cells acquire resistance to a particular treatment (Holohan *et al.*, 2013). In this context, a mechanistic explanation is defined by a set of assumptions and a set of model parameters, which need to be validated by contrasting the simulation outputs against experimental results. For instance, there are alternative models to explain the emergence of a resistant cell (Shaffer *et al.*, 2017). In one of them, the resistant cells are already present in a population and when the drug is provided they are selected. In an alternative model, the stress induced to cells by the introduction of the drug triggers cells' responses by changing the expression of the genes in many ways, and some of these allow cells to avoid the drug effect. In addition, the acquisition of resistance to a particular drug can be reversible or irreversible, meaning that it can occur at the level of the phenotype or the genotype. These models are being implemented and tested in the multiscale framework, to test which one reproduces experimental data.

In addition to the exploration of alternative resistance acquisition models, we study how the emergence of resistant cells can be mitigated by using combinatorial therapies based on pairs of drugs exhibiting a synergistic effect (Flobak *et al.*, 2015) as well as different drug scheduling schemes, such as intermittent pulses and drug alternation (Barros de Andrade E Sousa *et al.*, 2015), to test which strategies show better results.

PhysiBoSSa software allows to perform powerful simulations and to explore in more detail complex processes that can play a critical role in the emergence of resistant cells. Moreover, by considering the intracellular cell processes it is also possible to model phenomena that take place at the molecular level, such as the presence of synergy between different drug combinations. Consequently, the possibility to perform an online monitoring of the simulation is of great interest in the detection of patterns, which allow anticipating or predicting future events of interest, without the need to wait until the end of the simulation. However, due to its complexity, the simulations with this software are very computationally demanding.

We study two drug resistance-related scenarios using alternative models of cell signalling: the first one is a study of different drug regimes using TNF and the second is a study of drug combinations using the AGS gastric cancer cell line. In Figure 9, a schematic representation of the workflow to run MSM simulation is depicted. The components of the multiscale framework are used to run a simulation (Figure 9a-b), generating the simulation outcomes for several *in silico* TNF experiments from (Letort *et al.*, 2018) (Figure 9c): in each of the three experiments a different drug dose schedule was evaluated. The authors found that the drug dose provided in pulses at intervals of 300 minutes proved to be the best strategy (Letort *et al.*, 2018).

For the first scenario, we have considered the cell fate model from (Calzone *et al.*, 2010) and included in PhysiBoSS (Letort *et al.*, 2018) and a study on the best interval to deliver a drug that promotes cell death as well as survival. For the second scenario, we have considered an AGS Boolean model based on (Flobak *et al.*, 2015) (Figure 6). We have identified how to include techniques to forecast various events of interest of these scenarios, as well as interactive learning techniques to assist the calibration of the parameters of our models.

We explore two alternative geometries for cell arrangement: one-cell-thick 2D monolayers and 3D spheroids. The 2D monolayer is used to simulate cells growing in a plate, which correspond to the setting in which drug experiments where performed. The 3D spheroids are more suitable to simulate the growth of an *in vivo* tumour and are used to simulate real tumours and to explore different treatment strategies.

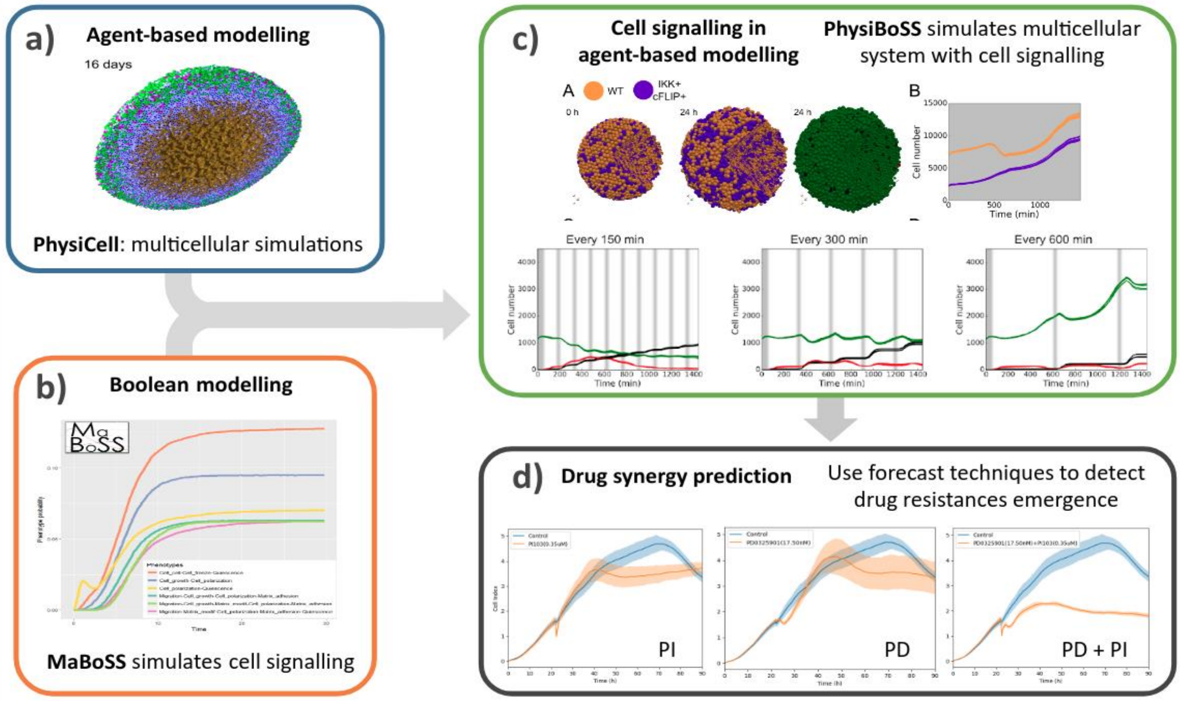

a) Agent-based modelling
16 days
PhysiCell: multicellular simulations
b) Boolean modelling
MaBoSS
Time
MaBoSS simulates cell signalling
c) Cell signalling in agent-based modelling
PhysiBoSS simulates multicellular system with cell signalling
A WT IKK+ cFLIP+
0 h
24 h
24 h
B
15000
10000
5000
0
Cell number
500
1000
Time (min)
Every 150 min
Every 300 min
Every 600 min
Cell number
Time (min)
d) Drug synergy prediction
Use forecast techniques to detect drug resistances emergence
Control
PI
PD
PD + PI
Time (h)

Figure 9: Representation of the MSM framework. MSM combines simulations of cell populations and the environment using an agent-based model approach (a), and the internal cells' state using a stochastic Boolean simulator (b). Results of the application of PhysiBoSSa to investigate drug regime treatments are shown (c) (Letort et al, 2018). Experimental growth curves of cells treated with different drugs dosages (PI or PD, as examples of drugs) or combinations of drugs (PD+PI) in orange are shown (d) (untreated cells are shown in blue).

4.2 Key Performance Indicators

The design and implementation of the Life Sciences use case will allow us to reach the Key Performance Indicators (KPIs) described in the Grant Agreement. Life Sciences use case had three different KPIs that addressed different parts of the architecture.

4.2.1 KPI 1: Increase the scale of the simulations up to a billion cells

To reach this first KPI, we are using the high-performance computing (HPC) at BSC's MareNostrum4 as it will allow scaling-up the simulations from thousands of cells up to a billion of cells. In this regard, our modelling framework already allows to use OpenMP (https://www.openmp.org/) to fully exploit the multicore architecture of each node of MareNostrum4.

We have designed different ways to further optimise the parallelisation process to multiple nodes using MPI. We estimate that the optimised parallelisation techniques will scale the volume of the data stream from 100 MB/min to 100 GB/min by increasing the number of agents and small molecules considered, as well as the time resolution at which the simulation outputs results. This MPI implementation allows us to address bigger, more complex simulation set-ups with cancer cell spheroids of up to a billion cells.

More on how we are working to reach this KPI in the following sections.

4.2.2 KPI 2: Forecasting up to five biological events

To reach this second KPI, we will use different forecasting techniques that help us detect events of biological interest. These techniques have been developed by the consortium in the scope of this project and are being currently tested in the TNF scenario.

Specifically, we aim at identifying drug dosage (in terms of concentration and frequency) that reduces the tumour growth; uncovering combination strategies that help stabilise the tumour and non-trivial patterns that can be used to predict the early emergence of resistant cells.

More on how we are working to reach this KPI in the following sections.

4.2.3 KPI 3: Dynamic modification of the simulation

The forecasting of these events will reduce uninformative simulations and avoid wasting computational resources while addressing highly demanding simulations such as those using billions of cell agents.

This third KPI aims at dynamically modifying the simulation runs to reduce uninformative simulations by 10%, avoiding wasting computational resources while addressing highly demanding simulations such as those using billions of cell agents. To achieve this, we need to have a definition of what is a desired simulation, let it be because we have labelled them as such or because the framework has rules that are based on that labelling, so that it can assign new simulations as desired or undesired.

Examples of undesired simulations are cases that are unfeasible from a biological point of view, cases that are uninformative, as their read-outs greatly differ from relevant biological data, cases with no remaining living cells and cases of drug effect saturation, where more drug will not further affect the simulation behaviour.

More on how we are working to reach this KPI in the following sections.

4.3 Addressing KPI 1: Multiscale Model HPC Implementation

We are currently working in the deployment of the MSM simulation framework PhysiBoSSa in BSC's supercomputer, the MareNostrum 4. This involves the parallelisation process using different technologies such as OpenMP and MPI to scale-up our simulations in terms of numbers of cells considered. We decided to first address the most basic level of simulation: the environment part and its dedicated engine, BioFVM. Our current work aims to parallelise the core kernels of BioFVM to support distributed parallelism using Message Passing Interface (MPI) enabling one to solve much larger problems with greater resolution and limited time. This work has been described in detail in Deliverable 1.2 and we present here a summary.

BioFVM (Ghaffarizadeh *et al.*, 2016) is a Finite Volume Method (FVM) based simulation software for solving PDEs (Partial Differential Equations) that model complex processes like uptake, release and diffusion etc., of substrates for multicellular organisms. BioFVM is capable of handling multiple substrates and can simulate the biological processes such as decay, diffusion using both cell and bulk sources. The design and implementation of BioFVM is such that it is scalable within a compute node, enjoys a minimum dependency on external libraries, is capable of running a multi-million cell simulation on desktops, and supports both 2D/3D simulations, among others. The software has been implemented in C++ and uses Open Multiprocessing (OpenMP) (OpenMP Architecture Review Board, 2018) to support shared memory parallelisation.

The code takes advantage of extensive vectorisation and the diffusion equation is solved using a fast direct algorithm called the Thomas algorithm for solving a tridiagonal system of linear equations. Multiple instances of such linear systems are solved simultaneously by multiple threads that also take advantage of extensive vectorisation. Such multiple instances of linear systems and solutions are made possible by splitting a higher dimensional PDE into multiple related 1-dimensional PDEs. The method that makes this splitting possible is called the locally 1-dimensional method or lod for short (Ghaffarizadeh *et al.*, 2018, 2016). Though BioFVM exhibits efficiency, reduced dependencies, ease of use, its greatest limitation is that it is limited to a single node only i.e. it is not capable of running on multiple nodes of an HPC cluster to solve a single, coherent problem. Thus, the simulation problem size is limited by the memory of a single node. Our current work aims to parallelise the core kernels of BioFVM to support distributed parallelism using Message Passing Interface (MPI) (Message Passing Interface Forum, 2015) enabling one to solve much larger problems with greater resolution and to reduce the time to solution.

We are motivated by the fact that BioFVM forms a core component of PhysiCell (Ghaffarizadeh *et al.*, 2018) and is used when updating the environment component. Thus, it is imperative to parallelise the core kernels of BioFVM to support distributed parallelism before PhysiCell can be parallelised. Nevertheless, BioFVM as a stand-alone software is useful enough to warrant parallelisation.

4.3.1 Design and parallelisation

The first step in parallelisation is domain decomposition or domain partitioning (Saxena *et al.*, 2016; Saxena, 2018; Saxena *et al.*, 2018) where the domain is divided into multiple smaller sub-domains and assigned to a specific MPI process. This approach of dividing the space was preferred to the alternative approach of dividing the total work instead of dividing the sub-domain as it was found difficult to quantify the total work in such a way (and since this approach leans towards a master-slave design pattern that is generally not scalable).

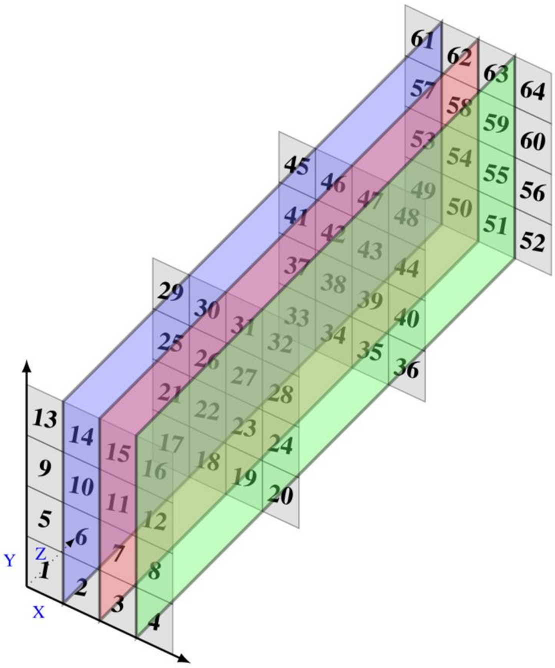


Figure 10: A 3D domain of dimensions 4×4×4 can be visualised as four 2D plates of dimension 4×4 arranged one after the other. A pure 1D domain partition of 4 MPI processes in the x-direction divides the voxels numbered from 1 to 64 into 4 parts. Rank 0 process contains voxel IDs 4n+1, rank 1 process contains voxel IDs 4n+2, rank 2 process contains voxel IDs 4n+3 and rank 3 process contains voxel IDs 4n+4, where n = 0,1,2,...,15. Data is contiguous in the x-direction and the distance between 2 consecutive elements in the y and z direction is 4 and 16, respectively.

Furthermore, a pure x-decomposition refers to the division of the x-direction of BioFVM among multiple processes. It can now be noted (according to our assumptions) that creating a pure x-decomposition requires creating a 1D MPI topology that has processes only in the y-direction. Thus, if P is the total number of MPI processes, dims[3] represents MPI processes in the x, y and z direction, respectively, we set dims[0]=1, dims[2]=1 and dims[1]=P to obtain a pure x-decomposition in BioFVM. The impact of a good domain decomposition on the performance of the diffusion solver in BioFVM cannot be emphasised enough and it's further detailed hereafter. For simplicity, we use a 1D MPI Cartesian topology of processes instead of a 3D topology to divide the physical domain into sub-domains although the mapping functions between MPI ranks, voxel indexes and points in 3D space apply to general 3D decompositions in 3D space (Figure 10).

Further, each process in our design maintains the local and global values of the number of voxels, local voxel indexes (mesh_index), global mesh index of the local voxels and the centre of each local voxel's global coordinates. First the global dimensions of the domain and the voxel dimensions are used to decide the number of local voxels. This is followed by the calculation of the global coordinates of the centres if voxels. Further, since each voxel must maintain the local and global mesh index, a local start of the global mesh index is calculated on each process, which then is used to assign the global mesh index to each voxel.

Apart from the domain decomposition, a list of the immediate directional neighbours of each voxel is also maintained. Such a scheme must accommodate for the cases when there is no local left/right neighbour or when the process is aligned to the physical boundary of the domain. In the serial BioFVM, a list for the Moore neighbourhood is also built for each voxel but since we do not find any examples that utilised this neighbourhood, we abstain from parallelising this routine. Note that the Moore neighbourhood equates to a 9-point stencil in 2D and a 27-point stencil in 3D (Kamil *et al.*, 2010).

BioFVM uses the Thomas solver (Thomas, 1949) for solving a tridiagonal system of linear equations that result from the FVM discretisation of diffusion PDEs. This solver is inherently serial and hence cannot be fully parallelised. However, there do exist parallel algorithms capable of solving tridiagonal systems of linear equations but with an increase in operation count and significant complexity of implementation (László, 2016). Thus, we perform the domain decomposition in only 1 direction. This makes the solver completely parallel in two directions but sequential in the third direction.

Additionally, the pure OpenMP version of BioFVM supports only the serial writing of the result data such as the concentration of the substrate after a specified time interval in a .mat file. In our Hybrid implementation, instead of gathering data at the root process, we use MPI-IO so that processes can view their portion of the file and write their part of the data simultaneously.

For all our experiments, we use the MareNostrum IV (MN4) supercomputer at the Barcelona Supercomputing Center (BSC). There are a total of 3456 nodes where each node has two Intel Xeon Platinum 8160 processors with a base frequency of 2.1 GHz. Each of the processors have 24 cores and the cores in each processor share a main memory of 48 GB. The computer nodes are interconnected using the Intel Omni-Path (OPA). The Operating System that the cluster uses is the SUSE Linux Enterprise Server 12 SP2. Further, we use GCC 8.1 and OpenMPI 3.1.1 as our compiler and MPI implementation respectively. We chose GCC as the compiler because the user documentation of BioFVM takes the GCC as the "gold standard".

4.3.2 Discussion

BioFVM enables the simulation of biological processes such as secretion, uptake, and diffusion for multicellular organisms. Internally it sets-up a microenvironment and uses PDEs to represent the biological processes. After discretisation using the Finite Volume Method (FVM), these PDEs are numerically solved using a direct solver for the tridiagonal system of linear equations. Currently, BioFVM suffers from a serious limitation as it only supports shared memory parallelisation using OpenMP and thus the size of the problems that it can solve is limited by the memory of a single node. With the aim to remove this limitation, we restructured the base data-structures and functions of BioFVM to add support for distributed parallelism using MPI.

To provide a proof of concept, we present the parallelisation of a chosen example and in the process, successfully parallelise the key kernels of BioFVM. For instance, we implement a pure 1D x-direction MPI Cartesian Topology and assign sub-domains to individual processes, generate basic agents that represent cells on the root process and map the positions to processes that these basic agents belong to, and parallelise the writing of result files among many other changes.

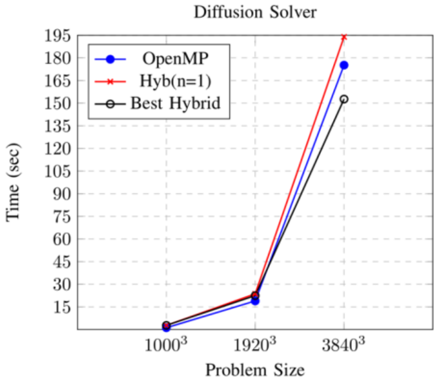


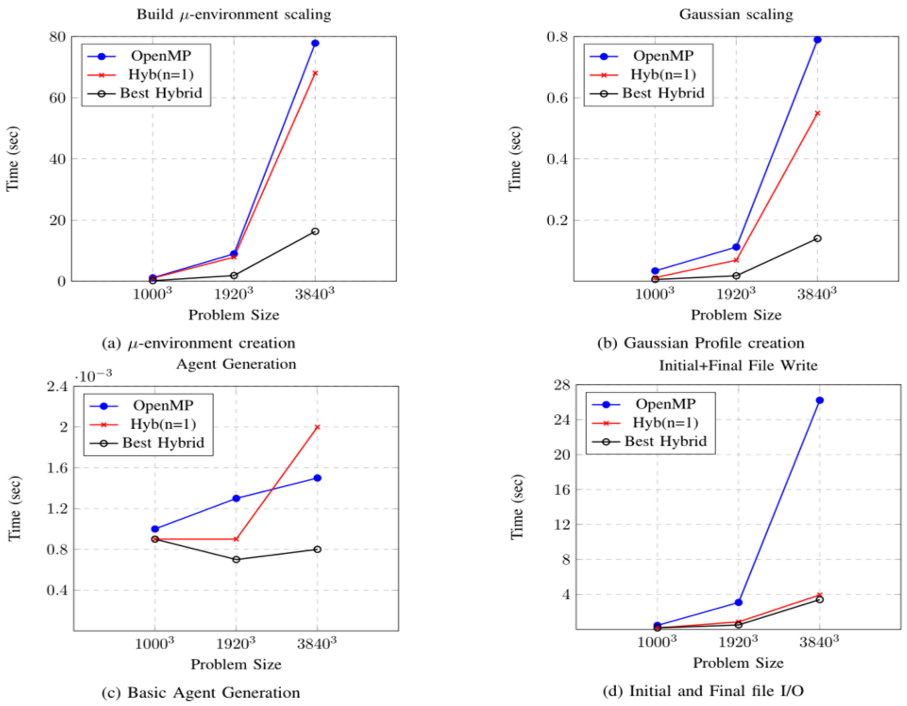
Figure 11: Execution time for the Thomas algorithm using pure OpenMP and Hybrid implementation. The Hybrid solver is completely parallel in the y and z direction but only on-node parallel in the x-direction.

Despite the fact that the solver is only 2/3 parallel and communication bound, we see a gain of about 4.21% over the pure OpenMP implementation with Hyb (n=2) for a (small) domain of size 1000^3 and with 1000 Basic Agents (Figure 11). Note that it is very difficult to beat the performance of the pure OpenMP implementation on a small problem size. As seen in Figure 12, one of the reasons for this gain is the excellent scaling of the phases: (1) creating a microenvironment, (2) generating a Gaussian profile and (3) the agent generation phase. Furthermore, using MPI-IO we substantially reduced the time for writing the files. With a domain of size 1920^3 and 1000 basic agents we see a performance gain of 13.24% with Hyb (n=2) and this gain approximately grows to 33% for a domain of size 3840^3. A domain of size 7680^3 cannot be executed on a single node as the application runs out of memory but was successfully executed using 4 and 8 nodes (Table 1). As an alternate test, we increased the number of agents to $2{\times}10^6$ on a domain of size 1000^3 and observed a performance gain of 26.12% over the pure OpenMP version.

Table 1: Time in seconds of the execution for the OpenMP version and the Hybrid versions of tutorial1 of BioFVM with a pure 1D x-decomposition. The total voxels and resolution are ≈ 500 million. The successful runs with 8 nodes and a minimum of 4 nodes use 384 and 192 threads respectively.

7680x7680x7680	OpenMP	Hyb(n=8)	Hyb(n=4)
Build μ-environment	-	67.81	141.98
Gaussian Profile	-	0.448	0.916
Initial File Write	-	4.1	2.56
Agent generation	-	0.0023	0.1060
Source/Sink/Diffusion	-	1210.41	1109.69
Final File Write	-	3.32	4.83
Total Time	-	1286.1	1260

We see a high performance gain in Basic Agent generation, building the microenvironment, file I/O (Figure 11) but a sub-optimal gain in the diffusion solver as the solver remains partially parallelised (Figure 12). Most importantly, we expose the structure of BioFVM to evaluate parallelisation schemes and make it possible to simulate larger and more complex problems i.e. BioFVM simulations can use multiple nodes and are not in any way limited to the memory of a single node.



(a) μ-environment creation

(b) Gaussian Profile creation

(c) Basic Agent Generation

(d) Initial and Final file I/O

Figure 12: Pure OpenMP execution times with increasing problem size Vs Hybrid MPI times for the same. Hyb(n=1) represents the time when a single node with 2 MPI processes and 24 threads is used. The Best Hybrid represents the least time for that kernel for any number of experimental nodes considered. The kernels considered are: (a) Creating the microenvironment (b) Building the initial Gaussian profile (c) Generation of Basic Agents (d) Sum of the initial and final I/O file write.

BioFVM constitutes the core at the centre of PhysiCell and its parallelisation is a required milestone to address the parallelisation of PhysiCell. Next, we plan to work towards optimally parallelising PhysiCell using an object-oriented framework.

Furthermore, since the solver is only parallelised using OpenMP, we aim to make it fully parallel by exploring parallel tridiagonal solver algorithms such as recursive doubling etc. As PhysiCell allows the cells to move across subdomains, a strategy is needed to asynchronously track the movement of cells across subdomains. This poses an additional challenge in terms of choosing an appropriate design pattern and a possibility of correctly using the MPI Remote Memory Access (RMA) operations. Future implementations can use 3D MPI Cartesian topologies instead of a 1D topology that we used in the current work, if there is a significant performance gain. A plethora of literature advocates the three dimensional decomposition for 3D decompositions can reduce the total volume of data that is exchanged but there exists literature that refutes these claims (Saxena *et al.*, 2018).

The MPI-ready BioFVM code is freely available from https://gitlab.bsc.es/gsaxena/physicell_x and the ongoing efforts to have an MPI-ready PhysiCell, also termed DistPhy, are available from https://gitlab.bsc.es/gsaxena/distphy.

4.4 Addressing KPI 2 and 3: model exploration with optimisation algorithms

Modern simulation-based application studies consist of large numbers of simulations with many possible variations in their parameters. Simulations are run with different parameters sets, resulting from an automated model parameter optimisation, classification, or, more generally, a model exploration (ME). Constructing software to run such studies at HPC scale is often unnecessarily time-consuming, as the resulting software artefacts are typically cluster-specific (Ozik *et al.*, 2016). Applying ME to MSMs requires an iterative workflow where simulations are run across a high dimensional parameter space and their initial conditions change to explore alternative areas of the parameter space.

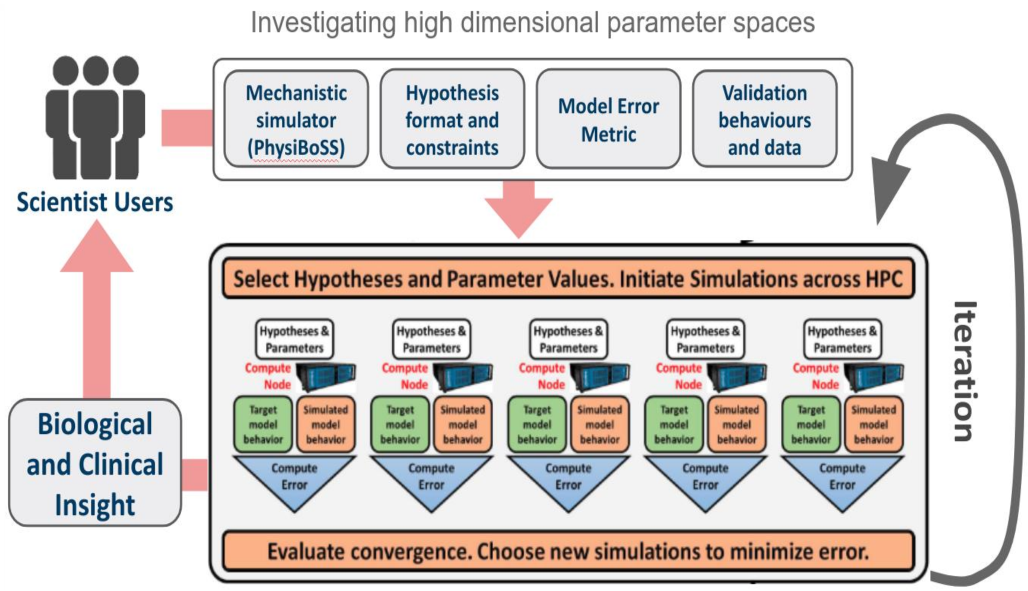

Investigating high dimensional parameter spaces
Scientist Users
Mechanistic simulator (PhysiBoSS)
Hypothesis format and constraints
Model Error Metric
Validation behaviours and data
Select Hypotheses and Parameter Values. Initiate Simulations across HPC
Hypotheses & Parameters
Compute Node
Target model behavior
Simulated model behavior
Compute Error
Evaluate convergence. Choose new simulations to minimize error.
Iteration
Biological and Clinical Insight

Figure 13: Model Exploration workflow adapted from (Ozik *et al.*, 2018)**.**

In a typical ME workflow, simulations' outputs from a set of experiments are evaluated against some predetermined metric, which informs the next iteration of simulation experiments (Figure 13). This metric is problem-specific and can range from a linear fit to a reinforced learning algorithm. In fact, the ME process can be enhanced with the use of complex event forecasting techniques, which can be used to improve the parameter space exploration.

Here, we present preliminary studies on the use of different metrics to find sets of parameters that reproduce a desired simulation's behaviour. This work has been described in detail in Deliverable 1.2 and 6.2.

4.4.1 Model exploration of simulation parameters

The first part of the model exploration strategy focuses on performing different simulations with slightly different parameters using the EMEWS framework. These different parameter sets are obtained by sweeping their values according to parameter-specific ranges. Extreme-scale Model Exploration with Swift/T (EMEWS) uses the general-purpose parallel scripting language Swift (Armstrong *et al.*, 2014) to generate highly concurrent simulation workflows. These workflows enable the integration of external ME algorithms to coordinate the running and evaluation of large numbers of simulations. The general-purpose nature of the programming model allows the user to supplement the workflows with additional analysis and post-processing (Ozik *et al.*, 2016). EMEWS framework and its high-throughput hypothesis testing has already been applied to the complex problem of tumour-immune interactions integrating it with PhysiCell and BioFVM (Ozik *et al.*, 2018, 2019).

We adapted the EMEWS framework to work with PhysiBoSSa and to study TNF-dependent behaviour. This study uses a signalling pathway model consisting of a network of 31 nodes that describes the behaviour of cells in response to different TNF regimes. In response to TNF presence, the cells can Proliferate, die by Apoptosis or Necrosis or remain in a Naive state of survival.

This study was already performed in PhysiBoSS (Letort *et al.*, 2018), however we wanted to reproduce it in the upgraded version of PhysiBoSSa, where many of the functions, parameters and their implementation have changed, notably the internalisation of environmental chemical entities into the agents and how they trigger effects on the signalling pathways. Also, we wanted to study the heterogeneity of the parameter space by finding the sets of parameters that evolve in the same way; that is, different sets of parameters that have the same global simulation output. Thus, we focused on finding a proper set of values for these modified parameters (Table) that allow for the reproduction of the simulations in which proliferative cells are depleted upon the addition of TNF each 150 minutes (Figure 14).

Table 2: Parameter space of the workflow for the TNF-dependent behaviour example.

Parameter description	Set of parameters	Min	Max	Increment	Dimensions
Concentration of TNF	NCSR	0	100	5	ng / mL
Frequency of TNF injections	NCSR	0	1440	10	minutes
Time point at which TNF is removed from the system	NCSR	0	1440	10	minutes
Duration of the TNF injections	NCSR	1	200	10	minutes
Oxygen sensing threshold	NCSR	0.1	0.5	0.01	dimensionless
TNF sensing threshold	BSC	0.1	0.5	0.01	dimensionless
TNF uptake rate	BSC	0	2	0.01	ng/mL/voxel/min
TNF secretion rate	BSC	0.01	1	0.01	ng/mL/voxel/min
Cell cycle rate	BSC	0.0001	0.01	0.0001	min^{-1}
Time of signaling model evaluation	BSC	5	20	0.5	minutes
Duration of initial injection of TNF	BSC	1	25	1	minutes

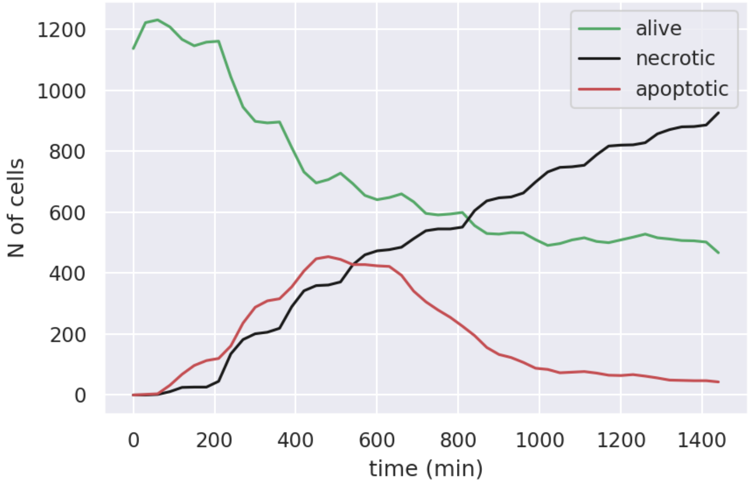


Figure 14: Spheroid's dynamical changes to pulses of TNF injection (0.5 ng/mL during 10 minutes) each 150 minutes. Green, Proliferative cells; red, Apoptosis; black, Necrosis. Initial spheroid radius is 100 mm.

4.4.2 Use of different optimisation metrics to find proper sets of parameters

The second part of the model exploration strategy involves the use of an optimisation metric that evaluates and ranks the performance of the different parameter combinations. As the set of parameters used are specific to the problem at hand, this metric needs also to be problem-specific. Several optimisation algorithms can be used for this; in fact, in a previous work (Ozik *et al.*, 2019), authors used a genetic algorithm to find optimal parameter sets that produced simulations with the lowest final mean tumour cell counts as well as an active learning algorithm to build surrogate models for characterising the parameter space structure based on different viability thresholds.

In our case, we have used a genetic algorithm and a random forest strategy to find a proper set of values for the parameters in PhysiBoSSa (Table) that allow for the reproduction of the results from the simulation with a pulsating regime of TNF each 150 minutes. In this section we detail some characteristics of these algorithms that have proved useful for our goals, but more in-depth information on the use of these algorithms can be found in Deliverable 1.2 and 6.2. The genetic algorithm from python's DEAP library was used with two sets of settings (Table) and following a classic GA strategy using BSC set (Figure 15) and a combined strategy using NCSR set (Figure 18).

Table 3: Sets of parameters used in the Genetic Algorithm studies.

	BSC set	NCSR set
Crossover probability	0.5	0.7
Mutation probability	0.2	0.5
Individuals' characteristics	7 parameters	4 parameters
Population	150	20
Number of generations	30	20

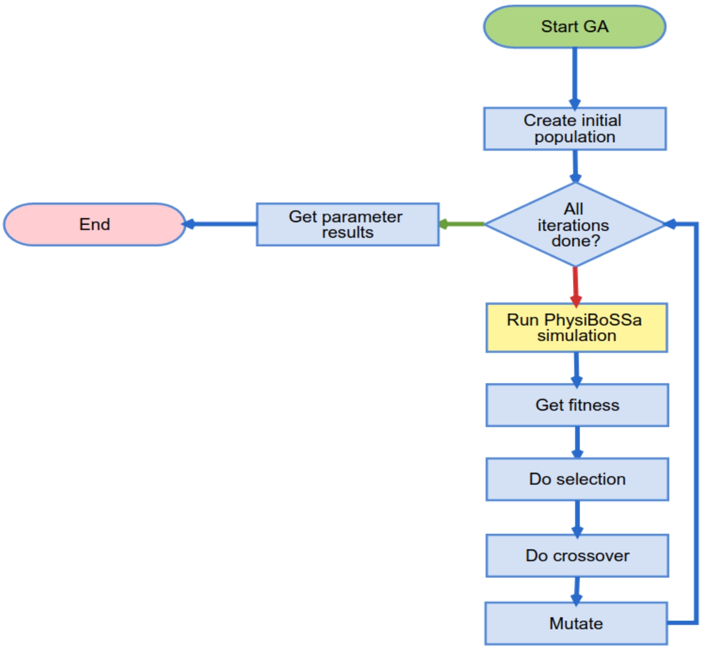


Figure 15: Flowchart of the genetic algorithm used on PhysiBoSSa on BSC set of parameters.

Additionally, we used a random forest algorithm from python's scikit-learn library (Pedregosa *et al.*, 2011) to study the structure of the parameter space. While genetic algorithms can be very efficient in discovering optimal solutions in large spaces, they are not sufficient for estimating the structure of complex parameter spaces, a task where random forests can provide useful insight.

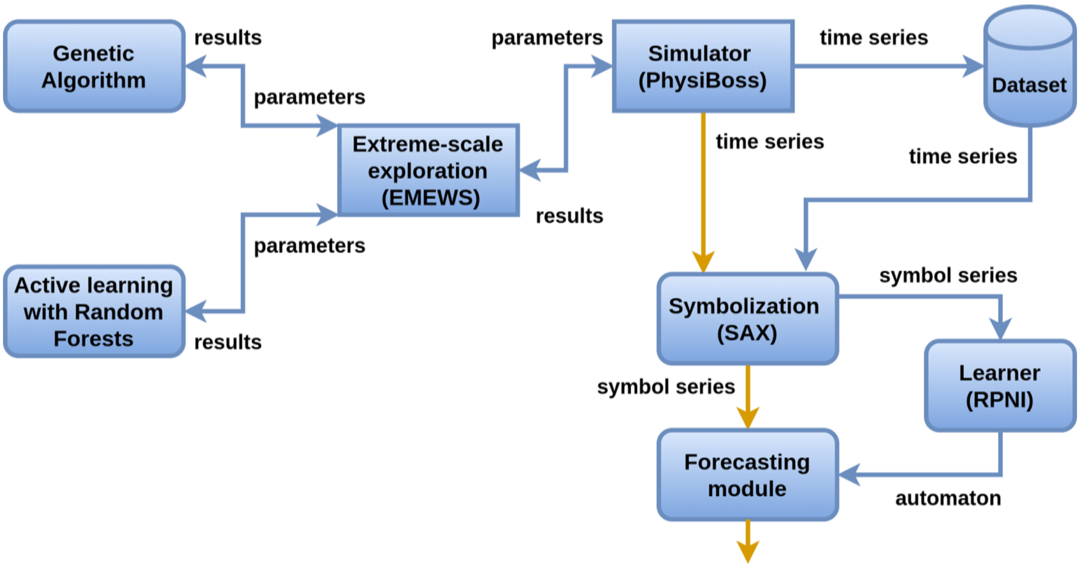


Figure 16: Flowchart of the diverse algorithms used on PhysiBoSS using NCSR set of parameters.

The training dataset used is a set of simulations classified via 1-NN and known cases as interesting or non-interesting and constitutes the initial random forest seed. From here, the algorithm iterates over a set of steps in which it constructs a random forest; it creates a fine grid in the parameter space and classifies its points using the new random forest; then computes the class uncertainty and keeps the points with the highest uncertainty values, as candidate points; and if the number of candidate points is bigger than a threshold (k), it clusters the candidate points into k groups. Finally, the algorithm selects one point from each cluster at random, evaluates the selected points by running a PhysiBoSSa simulation and using a 1-NN classifier and adds these points to the dataset.

4.4.3 Preliminary results

These ongoing efforts allow us to define the structure and hierarchy of the model's parameters and to evaluate its sensibility to the parameters' perturbation and are necessary to ensure the adaptability of present multiscale modelling to other models and use cases. We have found sets of data that capture the desired behaviours of our model as well as sets of data that fail at capturing these behaviours. We hereby present some of the preliminary results (**Figure 17** and **Figure 18**) using the aforementioned algorithms to find ranges of parameters that cause the model to behave like in **Figure 14**.

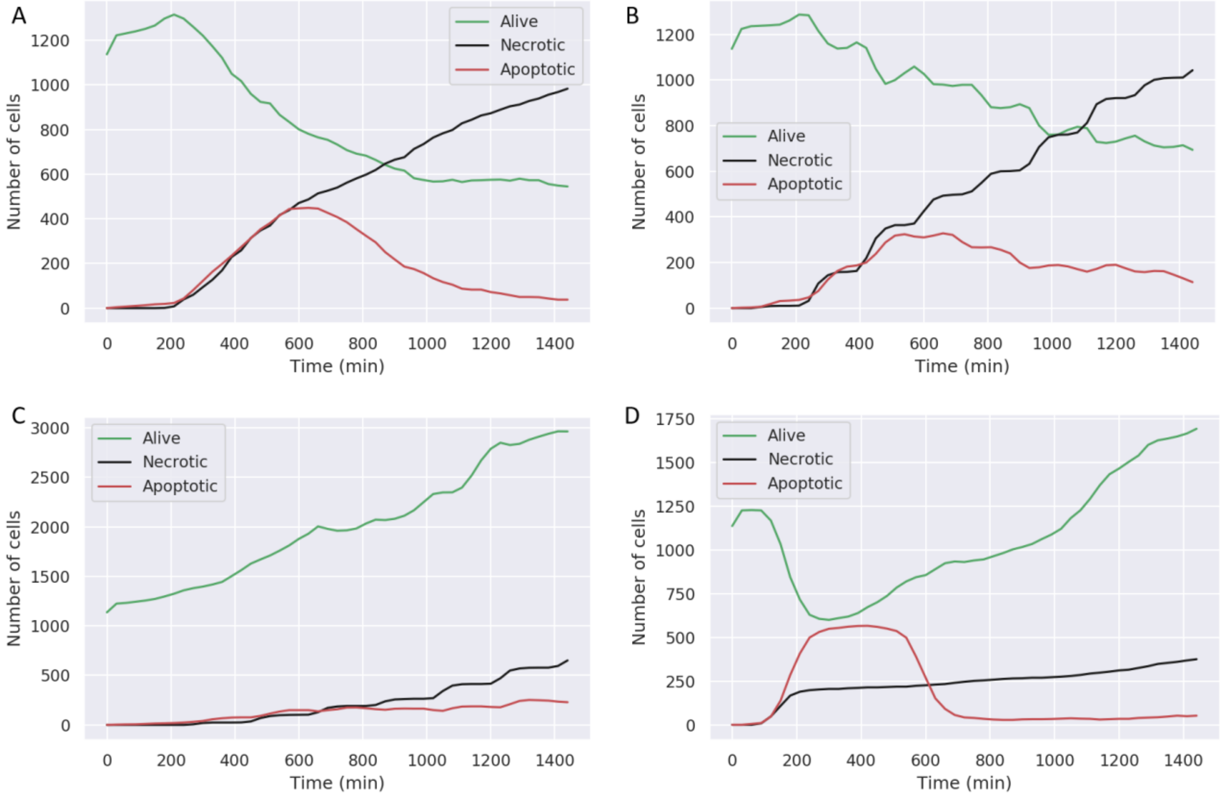


Figure 17: Some representative simulations performed by the model exploration framework on NCSR parameter set. (A and B) Positive examples of parameter sets that tally the behaviour of Figure 14 (C and D) Negative examples of parameter sets that do not tally the behaviour of Figure 14.

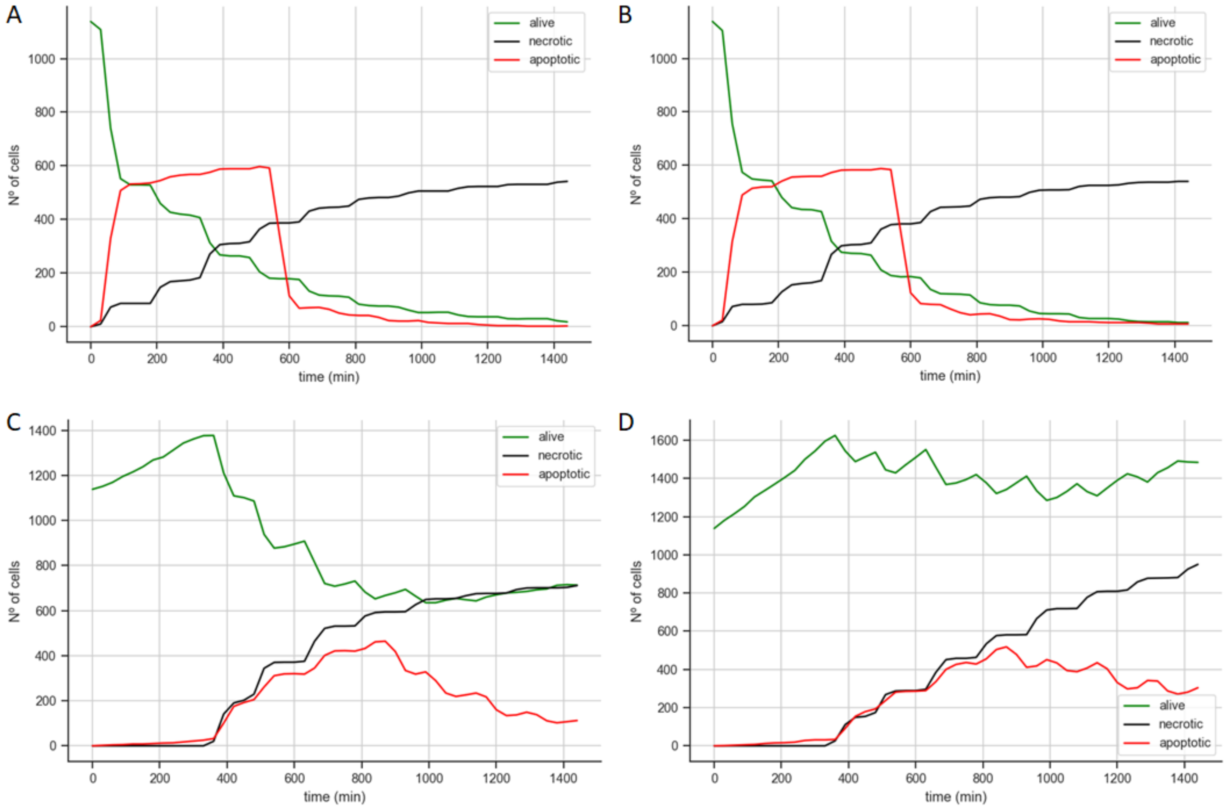


Figure 18: Some representative simulations performed by the model exploration framework on BSC parameter set. (A and B) Positive examples of parameter sets that tally the behaviour of Figure 14 (C and D) Negative examples of parameter sets that do not tally the behaviour of Figure 14.

4.5 Set of Applicable Computational Techniques

- **Boolean model solver:** To study Boolean asymptotic solutions, we use GINsim software (Naldi *et al.*, 2018).
- **Stochastic Boolean solver:** To study the transitions of the model when reaching asymptotic solutions, we use MaBoSS (Stoll *et al.*, 2012, 2017). This tool, based on Monte-Carlo kinetic algorithm, performs stochastic simulations on logical models. This strategy allows to calculate probabilities of each phenotype enabling a semi-quantitative approach to study the model outcomes and their response to different perturbations such as drugs. MaBoSS results are analysed using tailored scripts in Python and R.
- **Multiscale model:** Our MSM merges the stochastic Boolean model of MaBoSS with the agent-based PhysiCell software (Ghaffarizadeh *et al.*, 2018) into a tool named PhysiBoSS (Letort *et al.*, 2018) and refactored into PhysiBoSSa. PhysiCell as well as PhysiBoSS and PhysiBoSSa can be used with OpenMP and thus are suitable to be deployed in multi-thread computers. We are currently migrating PhysiCell to a hybrid OpenMP-MPI implementation to scale-up the Life Sciences use case simulations, allowing us to reach KPI 2.
- **Visualisation of simulation results:** We generally use Paraview software (https://www.paraview.org/) to visualise PhysiBoSSa results. In addition, other software is used for general data analyses such as PovRay, R or Python.
- **Model exploration:** Extreme-scale Model Exploration with Swift/T (EMEWS) was used to generate highly concurrent simulation workflows.
- **Optimisation techniques:** Genetic algorithms and random forest are being used together with EMEWS to select the simulation that tally a given desired output, allowing us to reach KPI 3.

5 Initial Evaluation Report

5.1 Expert Users and their Questionnaires

We contacted three researchers from public European research and high-education centres to be our Life Science's use case expert users. They are experienced researchers that cover different aspects in Systems Biology, modelling, omics data processing and pathway-based data analyses.

All three researchers were contacted and consented to participate in our questionnaire (Appendix C) and their anonymised responses can be found in Appendix D. From the interviews and questionnaires, we captured their interest in using data-streams, even though they are not prevalent in Life Sciences, and their lack of tools to use them.

All expert users agreed on the importance of the present project and the impact its outcomes, deliverables and KPIs would have on their current workflows. The models and know-how of these experts are a reflection of the state of the art on the field of biological modelling and their opinions are crucial as they are the ones in charge of exploring and incorporating in their future developments real-time data and forecasting techniques developed in the present project. In fact, according to their questionnaires and interviews they were willing to incorporate these tools and data on their day-to-day work and were very positive on the impact these would have in their fields.

They had high interest on being able to have a framework that could facilitate the development of forecasting techniques on real-time data. In addition, they agreed on the usefulness of having a graphical user interface to be able to program analyses with little coding knowledge and their will to incorporate their own tools to such a software.

6 Conclusions and Future Work

In this deliverable, we describe the Life Sciences use case, its requirements and a detailed description of the scenario, with different cell signalling and cell cycle modes. We have presented results on our currently ongoing HPC implementation and model exploration using BSC’s MareNostrum4.

We have detailed the Multiscale model that is being used in the Life Sciences use case, its parameters and how the different components are connected and build up our MSM. These components consist of:

- the environment component that simulates all the diffusion, creation and uptake of chemical entities that roam in the environment;
- the agent-based component that takes care of the population level and simulates the cells dynamics, their growth, death, movement and overall physical behaviour among cells and among them and their surrounding environment;
- the signalling network module that takes care of the individual cells’ level and simulates the behavior of each cell in response to its environment and its neighbouring conditions; and
- the cell cycle module that takes care of how the cells grow and divide.

We have explained why we need these components and their different scale of time to simulate complex cancerous behaviour, as well as how these components are connected and what are their relationships and dependencies.

Then, we have also described the scenario that are considered in this use case, its KPIs and how they are being addressed. We consider two different biological models that study drug resistance mechanisms in cancer: TNF pulsating regimes and drug combinations in a gastric cell line. These are well-studied models that allow us to take advantage of our multiscale modelling to uncover mechanisms in the interface of the cell signalling and environment effect that promote drug resistances.

Three KPIs have been established for this use case. KPI 1 requires the increase in scale of the simulations. We are addressing this KPI by migrating PhysiCell core code to use a hybrid OpenMP-MPI strategy in which simulations can now be simulated across nodes in MareNostrum4 cluster, as can be seen in Section 4.3 and in detail in D1.2. We have started by parallelising environment component and its Thomas solver used to simulate the diffusion of all chemical entities by using domain partitioning. We present benchmarks on different environmental domains that show a speed-up in the simulations and, more importantly, that the parallelization is possible. Currently, we are parallelizing the agents’ component using the same domain partition strategy. This will allow to have simulations of a billion of cells starting from few seeding cells.

KPI 2 requires the forecasting of five biological events and KPI3 requires to reduce uninformative simulations by 10%. We are addressing both of these KPI by using a simulation-based optimisation using a framework of model exploration, termed EMEWS. This EMEWS framework allows us to find the structure and hierarchy of the model’s parameters and to evaluate its sensibility to the parameters’ perturbations. EMEWs can also reduce many uninformative simulations and paves the way to have downstream forecasting techniques (Figure 18).

Finally, we have included the initial evaluation report, including the results of INFORE prototype on the available data streams in the Life Sciences use case.

7 References

An,G. (2010) Closing the Scientific Loop: Bridging Correlation and Causality in the Petaflop Age. *Sci. Transl. Med.*, **2**, 41ps34-41ps34.

Anderson,A.R.A. *et al.* (2006) Tumor Morphology and Phenotypic Evolution Driven by Selective Pressure from the Microenvironment. *Cell*, **127**, 905–915.

Armstrong,T.G. *et al.* (2014) Compiler Techniques for Massively Scalable Implicit Task Parallelism. In, *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, New Orleans, LA, USA, pp. 299–310.

Barros de Andrade E Sousa,L.C. *et al.* (2015) Dosage and Dose Schedule Screening of Drug Combinations in Agent-Based Models Reveals Hidden Synergies. *Front. Physiol.*, **6**, 398.

Béal,J. *et al.* (2019a) Framework for high-throughput personalization of logical models using multi-omics data. In, Kuperstein,I. and Barillot,E. (eds), *Computational systems biology approaches in cancer research*, Chapman & Hall/CRC mathematical & comput. CRC Press, Boca Raton.

Béal,J. *et al.* (2019b) Personalization of logical models with multi-omics data allows clinical stratification of patients. *Front. Physiol.*, **9**, 1965.

Bernabeu,M.O. *et al.* (2010) Shock-induced arrhythmogenesis in the human heart: A computational modelling study. *Conf. Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, **2010**, 760–763.

Brady,S.W. *et al.* (2017) Combating subclonal evolution of resistant cancer phenotypes. *Nat. Commun.*, **8**, 1231.

Calzone,L. *et al.* (2018) Logical versus kinetic modeling of biological networks: applications in cancer research. *Curr. Opin. Chem. Eng.*, **21**, 22–31.

Calzone,L. *et al.* (2010) Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput. Biol.*, **6**, e1000702.

Cohen,D.P.A. *et al.* (2015) Mathematical Modelling of Molecular Pathways Enabling Tumour Cell Invasion and Migration. *PLoS Comput Biol*, **11**, e1004571.

Flobak,Å. *et al.* (2015) Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling. *PLOS Comput. Biol.*, **11**, e1004426.

Gérard,C. and Goldbeter,A. (2014) The balance between cell cycle arrest and cell proliferation: control by the extracellular matrix and by contact inhibition. *Interface Focus*, **4**, 20130075.

Ghaffarizadeh,A. *et al.* (2016) BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations. *Bioinformatics*, **32**, 1256–1258.

Ghaffarizadeh,A. *et al.* (2018) PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems. *PLOS Comput. Biol.*, **14**, e1005991.

Hanahan,D. and Weinberg,R.A. (2011) Hallmarks of cancer: the next generation. *Cell*, **144**, 646–674.

Hoehme,S. *et al.* (2010) Prediction and validation of cell alignment along microvessels as order principle to restore tissue architecture in liver regeneration. *Proc. Natl. Acad. Sci.*, **107**, 10371–10376.

Holohan,C. *et al.* (2013) Cancer drug resistance: an evolving paradigm. *Nat. Rev. Cancer*, **13**, 714–726.

Ji,Z. *et al.* (2017) Predicting the impact of combined therapies on myeloma cell growth using a hybrid multi-scale agent-based model. *Oncotarget*, **8**, 7647–7665.

Kamil,S. *et al.* (2010) An auto-tuning framework for parallel multicore stencil computations. In, *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*., pp. 1–12.

Kim,E. *et al.* (2018) Cell signaling heterogeneity is modulated by both cell-intrinsic and -extrinsic mechanisms: An integrated approach to understanding targeted therapy. *PLoS Biol.*, **16**, e2002930.

László,E. (2016) Parallelization of numerical methods on parallel processor architectures.

Letort,G. *et al.* (2018) PhysiBoSS: a multi-scale agent-based modelling framework integrating physical dimension and cell signalling. *Bioinformatics*, bty766.

Message Passing Interface Forum (2015) MPI: A message-passing interface standard version 3.1.

Montagud,A. *et al.* (2017) Conceptual and computational framework for logical modelling of biological networks deregulated in diseases. *Brief. Bioinform.*, bbx163.

Naldi,A. *et al.* (2018) Logical Modeling and Analysis of Cellular Regulatory Networks With GINsim 3.0. *Front. Physiol.*, **9**, 646.

OpenMP Architecture Review Board (2018) OpenMP application program interface version 5.0.

Ozik,J. *et al.* (2016) From desktop to large-scale model exploration with SWIFT/T. *Proc. Winter Simul. Conf. Winter Simul. Conf.*, **2016**, 206–220.

Ozik,J. *et al.* (2018) High-throughput cancer hypothesis testing with an integrated PhysiCell-EMEWS workflow. *BMC Bioinformatics*, **19**, 483.

Ozik,J. *et al.* (2019) Learning-accelerated discovery of immune-tumour interactions. *Mol. Syst. Des. Eng.*, **4**, 747–760.

Pedregosa,F. *et al.* (2011) Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Saxena,G. *et al.* (2016) A cache-aware approach to domain decomposition for stencil-based codes. In, *2016 International Conference on High Performance Computing Simulation (HPCS)*., pp. 875–885.

Saxena,G. *et al.* (2018) A quasi-cache-aware model for optimal domain partitioning in parallel geometric multigrid. *Concurr. Comput. Pract. Exp.*, **30**, e4328.

Saxena,G. (2018) Efficient Domain Partitioning for Stencil-based Parallel Operators.

Shaffer,S.M. *et al.* (2017) Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance. *Nature*, **546**, 431–435.

Stoll,G. *et al.* (2012) Continuous time Boolean modeling for biological signaling: application of Gillespie algorithm. *BMC Syst. Biol.*, **6**, 116.

Stoll,G. *et al.* (2017) MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics*, **33**, 2226–2228.

Thomas,L. (1949) Elliptic problems in linear differential equations over a network: Watson scientific computing laboratory Columbia University, New York, NY.

Trisilowati and Mallet,D.G. (2012) *In Silico* Experimental Modeling of Cancer Treatment. *ISRN Oncol.*, **2012**, 1–8.

Tyson,J.J. *et al.* (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.*, **15**, 221–231.

Wittmann,D.M. *et al.* (2009) Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst. Biol.*, **3**, 98.

Yang,M. *et al.* (2018) Linking drug target and pathway activation for effective therapy using multi-task learning. *Sci. Rep.*, **8**, 1–10.

8 Glossary

- **AGS:** is the name of a cell line derived from gastric adenocarcinoma used as a laboratory model to study the aforementioned disease (https://web.expasy.org/cellosaurus/CVCL_0139).
- **Apoptosis:** or programmed cell death that occur in multicellular organisms.
- **Cell cycle:** or cell-division cycle, is the series of events that take place in a cell leading to duplication of its DNA (DNA replication) and division of cytoplasm and organelles to produce two daughter cells (https://en.wikipedia.org/wiki/Cell_cycle).
- **Drug synergy:** positive interaction between two or more drugs that causes the total effect of the drugs to be greater than the sum of the individual effects of each drug.
- **Necrosis:** traumatic cell death caused by acute cellular injury or starvation (lack of nutrients).
- **Resistant cell:** cell that become immune to a specific drug or treatment.
- **Signalling pathway:** is a set of proteins in a cell that work together to translate external signals and to control one or more cell functions, such as cell division or cell death (https://en.wikipedia.org/wiki/Cell_signaling#Signaling_pathways). The emergence of cancer cell is closely related to the deregulation of malfunctioning of specific signalling pathway.

Appendices

Appendix A: Experimental Data for Model Calibration

The use of drugs and mutants to study cell growth is an established strategy to uncover mechanistic associations in the molecular biology of cell growth, as drugs and mutants can hamper the functioning of a vital pathway diminishing the cells' viability to grow. If the target of the drug or the function of the gene is known, researchers can study the robustness of the different signalling pathways against a given perturbation. In our modelling framework these drugs' hampering effects can be simulated by inhibiting the activity of a given node that correspond to a protein or pathway. This same strategy can be followed to study the effects of different mutants on cell growth, such as knock-outs, over-expressed genes or knock-downs.

We plan to use growth experiments on a gastric cell line (AGS) to calibrate and test the predictions of the MSM simulations in 2D monolayer and 3D spheroids. AGS is a cell line derived from gastric adenocarcinoma and is a model system to study this type of cancer. We have gathered experimental datasets for cells growing under different treatments conditions, which include single drug experiments at different doses as well as studies of pairs of drugs (see Table A.1 in Appendix A for further details regarding drug and dose used). These works have been published by collaborators at the Norwegian University of Science and Technology (NTNU) (Flobak *et al.*, 2015).

Cell growth measurements were done in real-time using the xCELLigence RTCA SP growth assay (Roche Applied Science), an experimental system device to monitor cells in real-time that allows a non-invasive measurement of cell viability. Experiments were done in four replicates and controls (untreated cells) were included. This system uses culture plates with gold electrode arrays at the bottom of each well. The electrodes are used to perform real-time measurements of the impedance across the gold arrays. These measurements are reported in the dimensionless unit of cell index, which is taken to correspond to the number of cells (for further details see (Flobak *et al.*, 2015)).

Examples of growth under a single drug, combination of two drugs and their controls are depicted in Figure 9d. Cells grow exponentially until nutrients are consumed and cells die of starvation in the control experiment (Figure 9d, blue lines), while the population stops growing and, in some cases, decreases upon treatment with drugs (Figure 9d, orange lines), i.e., drugs hamper the functioning of a vital pathway and cells grow less. We plan to use part of these experiments to calibrate parameters of the MSM model in 2D monolayer and 3D spheroids – to use the same experimental conditions to simulate the model and reproduce the experimental results. Once this is done, we will use the calibrated model to propose other experiments not included during the parameter calibration (or learning) step and predict their outcome.

Additionally, we have another dataset of unpublished gene expression RNA-seq data for the same AGS cell line under five drug treatments from our collaborators at NTNU. We plan to use these gene expression data to, first, constraint the Boolean AGS signalling network and, afterwards, to use it to refine the cell cycle model from CRG that uses the actual cell cycle state (t) of a cell to infer the gene expression profile at time t+1.

Finally, we also have access to drug treatment experiments on other cell lines, which include 19 different drugs and their combinations on eight different cell lines of different cancer types (see Appendix A). For this dataset, we have access to the cell-line-specific IC50 values for the different drugs, a measure of the effectiveness of a substance in inhibiting a specific biological or biochemical function. On a later phase of the project, we plan to use the calibrated multiscale model to predict these IC50 values. Additional datasets can be retrieved from public repositories of cell-line-specific drug sensitivities data such as Genomics of Drug Sensitivity in Cancer (GDSC). This database covers the effect of over 300 drugs on over 500 cell lines, AGS being among them.

We have experimental data for single and combined drug screening in the AGS cell lines. Table A.1. describes the names of the different drugs as well their molecular target. For those drugs indicated with (*) we have time course experiments, i.e. the total number of cells at each time point (before and after the drug supply). For the other drugs we have experimental measurements of the IC50 for different concentrations of a single drug as well as combinations of two of drugs. While the time course experiments were only conducted on the AGS cell lines, the IC50 experiments were conducted in many different cell lines, listed in Table A.2.

Table A.1. Relation between drugs used in the experiments and the biological entity (protein/complex) target.

Abbreviation	Drug Name	Drug Effect	Protein/Complex Target
5Z	*(5Z)-7-oxozeaenol	inhibits	MAP3K7
AK	* AKTi-1,2	inhibits	AKT_f
BI	BIRB0796	inhibits	MAPK14
CT	CT99021	inhibits	GSK3_f
PD	*PD0325901	inhibits	MEK_f
PI	*PI103	inhibits	PIK3CA
PK	PKF118-310	inhibits	CTNNB1
JN	JNK Inhibitor XVI, JNK-IN-8	inhibits	JNK_f
D1	BI-D1870	inhibits	RSK_f
60	BI605906 (BIX02514)	inhibits	IKBKB
SB	SB-505124	inhibits	TGFBR1, ACVR1
RU	Ruxolitinib (INCB18424)	inhibits	JAK_f
D4	D4476	inhibits	CK1_f
F4	10058-F4	inhibits	MYC
SF	SF1670	inhibits	PTEN
ST	Stattic	inhibits	STAT3
G2	GSK2334470	inhibits	PDPK1
G4	GSK-429286	inhibits	ROCK1
P5	P 505-15 (PRT 062607)	inhibits	SYK

Table A.2. Cell line descriptors in which drug experiments were conducted and for which we have IC50 values available.

Cell line name	Cell line descriptor	Link
A498	Renal cell carcinoma	CVCL_0139
AGS	Gastric adenocarcinoma	CVCL_1056
COLO205	Colon adenocarcinoma	CVCL_0218
DU145	Prostate carcinoma	CVCL_0105
MDA-MB-468	Breast adenocarcinoma	CVCL_0419
SF295	Colon adenocarcinoma	CVCL_1690
SW620	Melanoma	CVCL_0547
UACC62	Melanoma	CVCL_1780

For the experiments performed on the AGS cell lines we also have gene expression level measured before (control) and after the drug/s supply. Figure A.1 shows a heatmap representing the expression level of those protein coding genes that are included in the AGS signalling network model. The clustering on the left-hand side indicates the replicates of the experiments are coherent. The information on gene expression can be used to further calibrate the MSM and, in particular, it can be used to inform the cell cycle model.

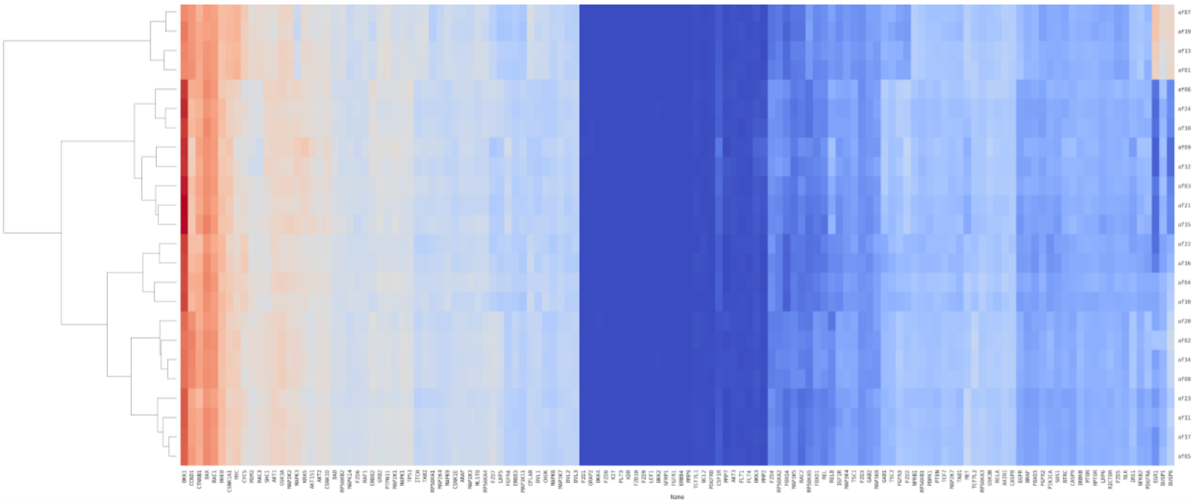


Figure A.1. Expression profile for those genes included in the signalling model corresponding to five different drug treatments and the control. Experiments were done in four replicates. Samples were clustered using a standard Hierarchical Clustering algorithm to check that they are clustered by their replicates, as indicated by the dendrogram on the left.

Appendix B: Example of Simulation Results

For each run, PhysiBoSS builds one file per time point with the results of each of the studied variables. This semicolon-separated file has a header row with the names of the variables and one row for each agent (cells, in this case).

Example of output file for cells at time 0:

```
Time;ID;x;y;z;radius;volume_total;radius_nuclear;contact_ECM;freezer;polarized_fraction;motility;cell_line;Cell_cell;phase;Cycle;NFkB

0;0;-46.758;-10.7294;-85.806;10.0174;4210.69;6.02332;0;0;0.1;0.01;0;2.65594;0;0;-1

0;1;-46.5751;7.86895;-82.8054;9.81311;3958.3;5.90048;0;0;0.1;0.01;0;3.66865;0;0;-1

0;2;-31.2033;-37.4872;-84.2829;9.5;3591.36;5.71221;0;0;0.1;0.01;0;2.99975;1;0;-1

0;3;-30.9612;-19.2273;-82.8856;10.6359;5039.78;6.39521;0;0;0.1;0.01;0;4.65177;0;0;-1

0;4;-33.8193;-0.642819;-82.4852;9.78174;3920.46;5.88162;0;0;0.1;0.01;0;5.55423;0;0;-1

…

0;945;39.239;-19.5593;86.5458;10.4203;4739.46;6.26558;0;0;0.1;0.01;0;4.97028;0;0;-1

0;946;42.6192;-2.5138;88.2562;11.4447;6279.16;6.88153;0;0;0.1;0.01;0;4.00144;0;0;-1

0;947;42.7765;18.3806;88.1489;9.64008;3752.58;5.79644;0;0;0.1;0.01;0;2.5922;0;0;-1
```

Similarly, time-specific semicolon-separated files are built for the different densities (free-roaming molecules on the extracellular space, such as oxygen, signalling molecules, microenvironment density, etc). This are named after the density they represent.

An Example of an output file for microenvironment density at time 0 is given below. Note that there is no header row. The first three columns correspond to spatial coordinates and the fourth to the value of the density:

```
-417.5;-492.5;-492.5;0.0630239

-357.5;-492.5;-492.5;0.0630185

-267.5;-492.5;-492.5;0.0630086

-252.5;-492.5;-492.5;0.0630072

-237.5;-492.5;-492.5;0.0630059

-117.5;-492.5;-492.5;0.0630041
```

Appendix C: Expert User Questionnaires, Life Sciences Use Case: Original questions

1. **User background information**
   a. Company/organisation
   b. Professional position [and years of experience]
   c. Domain of expertise
   d. Background studies (university degree major, etc.)
   e. What are your main job tasks?
   f. To whom are you responsible for performing these tasks (role, not name)?

2. **Existing workflow**
   a. Please describe the different kinds of data sources that you use in your day-to-day tasks and the tools that you use to process them

Kind of data sources	Volume of data (approx. MB, GB)	Purpose (task involved that generated the data)	Tools used to process the data*	Data analysis is automatic/ manual/ semi-automatic	Data gathering is historical/ real-time (online)	If real-time, update frequency

*If custom programs are used to process the data, please mention the programming language.

   b. Which is the aim of the analyses you perform (what kind of insights do you try to find)?
   c. What data processing challenges do you experience in your day-to-day tasks (e.g., fusion of heterogeneous sources, performance, analytics, scale of simulations, volume of simulation outputs, etc)?
   d. What problems do you run into in your day-to-day work when performing your data analysis?
      i. Why is this a problem?
      ii. How do you currently solve the problem? Is there a standard way of solving this?
      iii. How would you ideally like to solve the problem?
   e. On what kind of infrastructure do you usually run your analyses (e.g. personal laptop, high spec workstation, cloud server, cluster, HPC, etc)

3. **Expected benefits from using INFORE**

   a. Do you work with real-time data? If not, would you like to work with this kind of data?
      i. Are you tools and workflows able to work with such data?
      ii. Would you be interested in using a data processing workflow that would allow using real-time data?
      iii. Would using this data allow you to address different problems than the ones you are currently addressing?
   b. Do you use runtime analysis or do you wait until the end of the analysis to study its results?
      i. Would you be interested in using techniques that provide runtime results in your analyses?
      ii. Even if the provided output was an accurate approximation of the correct result?
   c. Do you currently use forecasting techniques? Are there specific events that you would like to forecast in real-time, which you currently cannot forecast?

4. **Specific aspects of the Life Sciences use case**

   a. Have you ever worked with biological models? With multiscale agent-based models?
   b. To what extent do multiscale agent-based simulations of tumoural cell growth relate to your projects/work?

c. Rate the following components of the INFORE Life Sciences use case according to your interest on the foreseen results (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

  i. multiscale models.
  ii. Boolean models.
  iii. cell cycle dynamics.
  iv. drug synergies inferences.
  v. real-size tumour simulations.
  vi. real-time/online data processing.

d. Rate the following Key Performance Indicators according to your interest on the foreseen results. (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

  i. KPI 1: Increase the scale of the simulations up to a billion cells.
  ii. KPI 2: Forecasting up to five biological events.
  iii. KPI 3: Dynamic modification of the simulation, such as killing simulations that are deviating from the targeted behaviour.

5. After the live demo of the biological scenario:

a. Would you be able to integrate these new developments to your current workflows? How relevant are they for your analyses?

  i. the use of online data stream.
  ii. the multiscale model distributed implementation.
  iii. the model exploration with optimisation.
  iv. the use of a graphical user interface such as RapidMiner that wraps it all up.

b. Would you be interested in incorporating the tools from your workflow in RapidMiner (if dedicated documentation is available)?

  i. If not, what else would be needed?

c. Rate the usefulness on the variety of characteristics of INFORE methods. (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

  i. How much are you satisfied by the variety of complex events that can be forecasted?
  ii. How much would you trust the output of the INFORE platform with respect to accuracy?
  iii. Is usability a key option for those developments?
  iv. Is the use of a graphical user interface such as RapidMiner is a good solution to incorporate new users?

d. Are there specific events that you would like to forecast in real-time, which we currently cannot forecast? If so, please explain.

e. Rate the following objectives of INFORE, based on how useful they may be at YOUR data analysis. (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

  i. Ability to design data processing workflows with no code required
  ii. Ability to change algorithm parameters graphically
  iii. Ability to receive quick approximate answers instead of 100% accurate, but long running queries
  iv. Ability to interactively explore the data in order to detect patterns/features of interest
  v. Ability to accurately forecast events of interest
  vi. Ability to optimise automatically your data analysis task over different data processing platforms (HPC, Big Data Platforms, etc).

f. What is the expected benefit from INFORE for you and your corporation? Rate from 1 (low priority) to 5 (high priority).

  i. Performance
  ii. Ability to handle multiple sources
  iii. Timely forecast of events
  iv. Interactive and efficient optimisable workflows

v. Automation of processes will reduce human error

g. Overall evaluation

i. How likely would your organisation be to use the INFORE platform operationally in the future?

ii. How useful are the methods used in INFORE?

iii. How useful are the biological models used in INFORE?

h. Is there anything else that you would like to add to your answers?

Appendix D: Expert User Questionnaires, Life Sciences Use Case: Expert users' anonymised answers

1. User background information

a. Company/organisation

Expert user 1: Public university

Expert user 2: Public research center

Expert user 3: Research center, private and publicly funded

b. Professional position [and years of experience]

Expert user 1: Professor (30 years)

Expert user 2: Group leader (12 years post PhD, group started 2018)

Expert user 3: Research engineer (15 years of experience)

c. Domain of expertise

Expert user 1: Systems biology/medicine, functional genomics,molecular biology, cancer biology

Expert user 2: Epigenetics, Logical modelling, immuno-oncology

Expert user 3: Systems Biology / Mathematical modelling

d. Background studies (university degree major, etc.)

Expert user 1: PhD in Biochemistry

Expert user 2: Degree Physics (2004) PhD Earth Sciences (2008)

Expert user 3: PhD in Biology, Masters in Mathematics

e. What are your main job tasks?

Expert user 1: Research group leader / University

Expert user 2: We are running projects simulating the interactions between different cells in the tumour micro-environment

Expert user 3: Construction of mathematical models / Analyses of models / Participation to the development of appropriate tools / Writing grants / Writing articles

f. To whom are you responsible for performing these tasks (role, not name)?

Expert user 1: Department head / The University

Expert user 2: Research centre director

Expert user 3: Department head

2. **Existing workflow**
   a. Please describe the different kinds of data sources that you use in your day-to-day tasks and the tools that you use to process them

Kind of data sources	Volume of data (approx. MB, GB)	Purpose (task involved that generated the data)	Tools used to process the data*	Data analysis is automatic/ manual/ semi-automatic	Data gathering is historical/ real-time (online)	If real-time, update frequency
Outputs of simulations	MB	Model analysis / Mutant simulations / Drug effect predictions	Logical modelling (GINsim, MaBoSS, PhysiBoSS)	Manual	historical	
In lab and public data on cancer cell line cellular and molecular (proteomic, transcriptomic) responses to drug treatment (single and combination); Knowledge bases with data on molecules, interaction, and function (e.g. Uniprot, Intact, reactome, SIGNOR, Gene ontology annotation; etc)	In the high MBs	Observe cancer cell drug response for testing of predictions from simulations with computational models	Statistical tools (SPSS, Matlab, R-tools) Functional overrepresentation (e.g. BINGO, DAVID) PARADIGM (infer protein state from transcriptomics and genomics)	semi-automatic	historical	
Cellular responses are mainly cell viability and are measured mainly in robotic high throughput systems; Molecular data are both large scale (NGS, MS) and small	In the high MBs	Molecular data: biomarkers for configuring computational models; testing of molecular hypotheses generated by models	Statistical tools (SPSS, Matlab, R-tools) Functional overrepresentation (e.g. BINGO, DAVID) PARADIGM (infer protein state from transcriptomics and genomics)	semi-automatic	historical	

scale (e.g. western blot)						
Outputs of simulations	MB	Model analysis / Mutant simulations / Cell-cell interactions / Drug effect predictions	Logical modelling (GINsim, MaBoSS, PhysiBoSS) Custom scripts, BooleanNet	Manual, semi-automatic	historical	

*If custom programs are used to process the data, please mention the programming language.

b. Which is the aim of the analyses you perform (what kind of insights do you try to find)?

Expert user 1: Insights that can help us build computational modes predictive of individual cancer patient drug response and resistance

Expert user 2: We try to understand what are the most important processes influencing the state of the tumour microenvironment

Expert user 3: Reproduce experimental data, use model outputs to stratify cancer patients, predict outcomes of drug treatments, etc.

c. What data processing challenges do you experience in your day-to-day tasks (e.g., fusion of heterogeneous sources, performance, analytics, scale of simulations, volume of simulation outputs, etc)?

Expert user 1: Integration data from heterogeneous sources; Inference of state of one type of molecular entities in the cell (proteins) from other types of molecular entities in the cells (transcripts, genome)

Expert user 2: Parameter search, expansion of logical model though genetic algorithms, integration with data

Expert user 3: Performance: computation time spent on laptop too long; Parameter fitting is an important aspect of our research and the current tools do not allow a comprehensive search of parameter search

d. What problems do you run into in your day-to-day work when performing your data analysis?

Expert user 1: Difficult to A) integrate large scale molecular data from heterogeneous sources; B) infer protein state from transcriptomics and genomics

Expert user 2: Increasing complexity of logical model

Expert user 3: Time for each simulation

i. Why is this a problem?

Expert user 1: A) standardization of data format not optimal across data sources; B) science lacks sufficient insight into relationship between protein-transcript-gene and problem B) may be impossible to solve without additional data

Expert user 2: It will hit execution time limits when trying to run enough realizations of the simulations

Expert user 3: The model we use are bigger and bigger (with high number of nodes, which requires more time for simulations)

ii. How do you currently solve the problem? Is there a standard way of solving this?

Expert user 1: A) extra work (partially manual) to align data; B) available computational tool (PARADIGM)

Expert user 2: It will reduce number of simulations or increase patience

Expert user 3: We use the clusters made available by the institute. No. I usually wait for the simulations to be done on my computer.

iii. How would you ideally like to solve the problem?

Expert user 1: Standardized data format and interconversions interlinkages between data and knowledge sources; improved computational tools for inferring protein states from transcriptomics and genomics, possibly by taking additional data and knowledge into consideration.

Expert user 2: More efficient coding, explore simulations in real time

Expert user 3: Not sure

e. On what kind of infrastructure do you usually run your analyses (e.g. personal laptop, high spec workstation, cloud server, cluster, HPC, etc)

Expert user 1: Personal laptops, high specs workstations, work servers.

Expert user 2: Currently high-spec workstation, clusters for parameter search

Expert user 3: Laptop / workstation / cluster

3. Expected benefits from using INFORE

a. Do you work with real-time data? If not, would you like to work with this kind of data?

Expert user 1: Yes, I work with growth curves of cancer cells.

Expert user 2: No. Yes, I would like to work with it.

Expert user 3: No, I do not work with real-time data. Yes, I would like to work with it.

iv. Are you tools and workflows able to work with such data?

Expert user 1: Yes, but the analysis are quite simple.

Expert user 2: Yes, from imaging data.

Expert user 3: Not yet

v. Would you be interested in using a data processing workflow that would allow using real-time data?

Expert user 1: Yes.

Expert user 2: Yes.

Expert user 3: Of course

vi. Would using this data allow you to address different problems than the ones you are currently addressing?

Expert user 1: Yes. Namely with an improved workflow.

Expert user 2: Yes.

Expert user 3: No, but it would address more accurately the problems we are working on.

b. Do you use runtime analysis or do you wait until the end of the analysis to study its results?

Expert user 1: We wait.

Expert user 2: We wait.

Expert user 3: We wait.

i. Would you be interested in using techniques that provide runtime results in your analyses?

Expert user 1: Yes.

Expert user 2: Yes.

Expert user 3: Yes, definitively.

ii. Even if the provided output was an accurate approximation of the correct result?

Expert user 1: Yes.

Expert user 2: Yes.

Expert user 3: Yes.

c. Do you currently use forecasting techniques? Are there specific events that you would like to forecast in real-time, which you currently cannot forecast?

Expert user 1: Not forecasting specifically, but we work with predictions. Yes, we would like to forecast different events regarding drug interaction and growth curves.

Expert user 2: Yes.

Expert user 3: No, but it would be very useful as it would save time in the parameter search.

4. Specific aspects of the Life Sciences use case

a. Have you ever worked with biological models? With multiscale agent-based models?

Expert user 1: Yes. Not agent-based models, but Boolean models.

Expert user 2: Yes. Yes.

Expert user 3: Yes. Yes.

b. To what extent do multiscale agent-based simulations of tumoural cell growth relate to your projects/work?

Expert user 1: Very relevant for drug response.

Expert user 2: A lot, 100%.

Expert user 3: Completely

c. Rate the following components of the INFORE Life Sciences use case according to your interest on the foreseen results (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

vii. multiscale models.

Expert user 1: 5

Expert user 2: 5

Expert user 3: 5

viii. Boolean models.

Expert user 1: 5

Expert user 2: 5

Expert user 3: 5

ix. cell cycle dynamics.

Expert user 1: 5

Expert user 2: 3

Expert user 3: 5

x. drug synergies inferences.

Expert user 1: 5

Expert user 2: 3

Expert user 3: 5

xi. real-size tumour simulations.

Expert user 1: 5

Expert user 2: 5

Expert user 3: 5

xii. real-time/online data processing.

Expert user 1: 5

Expert user 2: 4

Expert user 3: 5

d. Rate the following Key Performance Indicators according to your interest on the foreseen results. (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

iv. KPI 1: Increase the scale of the simulations up to a billion cells.

Expert user 1: 5

Expert user 2: 4

Expert user 3: 5

v. KPI 2: Forecasting up to five biological events.

Expert user 1: 5

Expert user 2: 5

Expert user 3: 5

vi. KPI 3: Dynamic modification of the simulation, such as killing simulations that are deviating from the targeted behaviour.

Expert user 1: 5

Expert user 2: 3

Expert user 3: 5

5. After the live demo of the biological scenario:

a. Would you be able to integrate these new developments to your current workflows? How relevant are they for your analyses?

Expert user 1: Yes.

Expert user 2: Yes. Very much relevant.

Expert user 3: Yes. A lot.

i. the use of online data stream.

Expert user 1: 5

Expert user 2: Yes.

Expert user 3: Yes.

ii. the multiscale model distributed implementation.

Expert user 1: 5, with proper specialists

Expert user 2: Yes.

Expert user 3: Yes.

iii. the model exploration with optimisation.

Expert user 1: Yes.

Expert user 2: Yes.

Expert user 3: Yes.

iv. the use of a graphical user interface such as RapidMiner that wraps it all up.

Expert user 1: Yes.

Expert user 2: Yes.

Expert user 3: Yes.

b. Would you be interested in incorporating the tools from your workflow in RapidMiner (if dedicated documentation is available)?

Expert user 1: Yes.

Expert user 2: Yes.

Expert user 3: Yes

i. If not, what else would be needed?

c. Rate the usefulness on the variety of characteristics of INFORE methods. (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

i. How much are you satisfied by the variety of complex events that can be forecasted?

Expert user 1: 5

Expert user 2: 5

Expert user 3: 4

ii. How much would you trust the output of the INFORE platform with respect to accuracy?

Expert user 1: 4

Expert user 2: 5

Expert user 3: I would first us the platform as a test. For instance, I would like to see examples like cell division and different solvers to check for numerical instability. For a later explorative phase, then 4.

iii. Is usability a key option for those developments?

Expert user 1: 5

Expert user 2: 5

Expert user 3: No, but it may come to be.

iv. Is the use of a graphical user interface such as RapidMiner is a good solution to incorporate new users?

Expert user 1: 5, from the looks of it. We would need to use it to evaluate it further.

Expert user 2: 5

Expert user 3: Yes.

d. Are there specific events that you would like to forecast in real-time, which we currently cannot forecast? If so, please explain.

Expert user 1: Yes, sure. Response of patient-derived cell culture to combinations of drugs, where time is of the essence. In addition, it would be interesting to simulate bigger and complex patients' tumours in real time.

Expert user 2: I would like to forecast spatiotemporal patterns.

Expert user 3: I would like to have forecasts on the search for parameters.

e. Rate the following objectives of INFORE, based on how useful they may be at YOUR data analysis. (1: Not useful, 2: Of some use, 3: Average Use, 4: Quite useful, 5: Very useful)

vii. Ability to design data processing workflows with no code required

Expert user 1: 5

Expert user 2: 2

Expert user 3: 4

viii. Ability to change algorithm parameters graphically

Expert user 1: 5

Expert user 2: 3

Expert user 3: 4

ix. Ability to receive quick approximate answers instead of 100% accurate, but long running queries

Expert user 1: 5

Expert user 2: 5

Expert user 3: We have two different purposes. If it's to build models, then 2; once the model is built, then 4.

x. Ability to interactively explore the data in order to detect patterns/features of interest

Expert user 1: 5

Expert user 2: 5

Expert user 3: 4

xi. Ability to accurately forecast events of interest

Expert user 1: 5

Expert user 2: 5

Expert user 3: 4

xii. Ability to optimise automatically your data analysis task over different data processing platforms (HPC, Big Data Platforms, etc).

Expert user 1: 5

Expert user 2: 5

Expert user 3: 4

f. What is the expected benefit from INFORE for you and your corporation? Rate from 1 (low priority) to 5 (high priority).

i. Performance

Expert user 1: 5

Expert user 2: 5

Expert user 3: 5

ii. Ability to handle multiple sources

Expert user 1: 5

Expert user 2: 5

Expert user 3: 2

iii. Timely forecast of events

Expert user 1: 5

Expert user 2: 3

Expert user 3: 4

iv. Interactive and efficient optimisable workflows

Expert user 1: 5

Expert user 2: 5

Expert user 3: 4

v. Automation of processes will reduce human error

Expert user 1: 5

Expert user 2: 4

Expert user 3: 3, but needs to be transparent.

g. Overall evaluation

i. How likely would your organisation be to use the INFORE platform operationally in the future?

Expert user 1: 5, with proper personnel and funding

Expert user 2: Very likely.

Expert user 3: Very high.

ii. How useful are the methods used in INFORE?

Expert user 1: 5

Expert user 2: Very useful.

Expert user 3: Very useful. We routinely use them.

iii. How useful are the biological models used in INFORE?

Expert user 1: 5

Expert user 2: Very useful.

Expert user 3: Very useful. We routinely use them.

h. Is there anything else that you would like to add to your answers?

Expert user 1: No.

Expert user 2: Addressing spatiotemporal patterns would be of high interest.

Expert user 3: No