# Heterogeneous Tumour Modeling Using PhysiCell and Its Implications in Precision Medicine

**Miloš Savić, Vladimir Kurbalija, Igor Balaz, and Mirjana Ivanović**

## 1 Introduction

### *1.1 Precision Medicine—General Introduction*

In modern society, with the constantly growing economy and demanding working and living conditions, a large portion of the population is facing stressful life that triggers chronic health problems like cancer, cardiovascular or neurological diseases. Such a situation requires a shift in healthcare towards more interdisciplinary, multidisciplinary, and holistic initiatives [6, 38].

Accordingly, the importance of the development of appropriate services to improve the quality of life of persons who suffers from chronic diseases has been widely recognized. For that, the collection of huge amounts of patients' complex data (like clinical, environmental, nutritional, everyday activities…) is needed. It is necessary to properly aggregate such data, analyze it and present it to the clinical doctors and caregivers so they can devise better recommendations for adequate treatment and actions to improve patient's health.

This approach is the essence of the emerging research called personalized medicine or precision medicine. According to the US National Research Coun-

M. Savić (✉) · V. Kurbalija · M. Ivanović
Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 3, Novi Sad, Serbia
e-mail: svc@dmi.uns.ac.rs

V. Kurbalija
e-mail: kurba@dmi.uns.ac.rs

M. Ivanović
e-mail: mira@dmi.uns.ac.rs

I. Balaz
Faculty of Agriculture, University of Novi Sad, Trg Dositeja Obradovića 8, Novi Sad, Serbia
e-mail: igor.balaz@df.uns.ac.rs

I. Balaz and A. Adamatzky (eds.), *Cancer, Complexity, Computation*,
Emergence, Complexity and Computation 46,
https://doi.org/10.1007/978-3-031-04379-6_7

cil, “personalized medicine” is an older term with a similar meaning as “precision medicine.” However, the term “personalized” could be misunderstood to imply that medical treatments are being developed uniquely for each individual.

According to the Precision Medicine Initiative [15], precision medicine is “an emerging approach for disease treatment and prevention that takes into account individual variability in genes, environment, and lifestyle for each person.” Precision medicine, where an individual patient’s molecular information is processed, also can be an adequate way to try to recognize different risk factors and help in the prevention of critical diseases like cancer [20]. Regardless of the scope of data included in consideration, the main characteristic of the approach is to obtain a more accurate prediction of which treatment and prevention strategies will have the best results for which groups of patients. It has the potential to improve the traditional symptom-driven and one-size-fits-all approach allowing earlier medical diagnostics, interventions, and tailoring better and economically personalized treatments.

Additionally in the last several decades, we are facing enormous production of biological data that causes a paradigm shift in medical research. Technological support to study molecular changes over the whole genome brings new dimensions to the concept of precision medicine. Such a comprehensive approach supported by powerful ICT tools and methods raised hope for the development of superior diagnostic and therapeutic instruments. This is especially relevant to cancer as its incidence is globally increasing.

Regardless of the promised theoretical benefits of precision medicine, its role in day-to-day healthcare is still relatively limited. However, given the enormous technological development and rapid research actions in a lot of medical-related disciplines, the expectations are that this approach will expand in medical and healthcare areas in the coming years.

### *1.2 Treatment Approaches in Precision Medicine*

Cancer treatment options are numerous. From traditional chemotherapy where chemicals are used to kill fast-growing cells (both cancerous and non-cancerous) to more precise treatments such as hormone therapy, immunotherapy and targeted therapy. All of these more precise treatments rely on performing tumour analysis for patient-specific biomarkers (genes, proteins and other tumour markers). A closely related field is Pharmacogenomics. It is still in its infancy but its main focus is on discovering how genes affect a person’s response to specific drugs. Pharmacogenomics combines the science of drugs and genomics to try to develop effective but safe medications and propose adequate doses that are tailored to variations in a person’s genes. Expectations of this approach are very optimistic for the development of tailored drugs to treat serious health problems like cancer [20], cardiovascular disease [3], and so on.

In cancer research, there are several approaches to identifying appropriate molecular targets. One approach is to identify overexpressed proteins. The very first targeted therapy, tamoxifen, made in 1962, has been developed to inhibit the growth of estrogen receptor-positive (ER+) breast cancer cells [16]. The second breakthrough happened also by identifying overexpressed proteins - Human Epidermal Growth Factor receptor 2 (HER-2). HER-2 positive tumours are sensitive to the inhibition of Her2 receptor function and to date, a large number of therapies has been developed. They include monoclonal antibodies (trastuzumab, pertuzumab, margetuximab-cmkb), antibody-drug conjugates (different conjugates with trastuzumab), pan-her inhibitors (neratinib), signal transduction inhibitors (lapatinib) and tyrosine-kinase inhibitors (tucatinib).

Another approach is to determine whether cancer cells produce mutant proteins that drive cancer progression. An example is the cell growth signalling protein BRAF, which is in many melanomas presented in altered form, BRAFV600E. Inhibitor of B-Raf enzyme, Vemurafenib, has been approved in 2011 to treat patients with inoperable or metastatic melanoma that contains mutated BRAF protein. Currently, several clinical trials investigate possibilities of combined treatments to further improve the prognosis of patients with BRAFV600E mutation [24].

Yet another approach is to search for chromosome instabilities (CIN) that are present in cancer cells but not in normal cells [35]. Such instabilities are defined as an increase in the rate at which whole chromosomes or chromosomal fragments are gained or lost, typically resulting in aneuploidy or abnormal DNA content. Such changes have been observed in many cancers but are best understood in colorectal cancer [5, 21]. Two main strategies for targeting CIN in cancer are CIN-reducing and CIN-inducing approaches. The CIN-reducing approach aims to reduce the rate of CIN (by inhibiting abnormal processes), while the CIN-inducing approach aims to increase the level of chromosome missegregation and/or DNA damage to induce cell death [1].

### *1.3 The Role of Microenvironment in the Future Development of Precision Medicine*

All the approaches described above are focused on searching for single-cell-level targets. When successful, they enable precise targeting of specific cancer phenotype/genotype thus increasing treatment efficacy and making them less harmful for a patient. However, the main limiting factor of targeting dominant tumour cells only, is the high probability of developing drug resistance. It has been shown that the tumour microenvironment plays a major role in rapidly inducing drug resistance via a cascade of signalling events that transiently protect tumour cells from apoptosis induced by therapeutic chemicals [25]. The tumour microenvironment is a spatially and functionally complex network of the vasculature, stromal cells, Cancer Stem Cells (CSC), cancer-associated fibroblasts (CAF), and tumour-associated macrophages (TAM). In the following paragraphs, we will briefly outline each of the main microenvironment constituents and highlight their importance in determining the clinical outcome of treatment and the development of treatment resistance.

To have a lasting effect a drug should reach close to 100% of tumour cells [8]. To achieve that, once it leaves blood vessels a drug should penetrate as far as possible through a tumour tissue [33, 34]. Diffusion depth depends on both the chemical properties of a drug and microenvironment properties (density of extracellular matrix, tissue pressure, interstitial fluid pressure). Moreover, tissue oxygenation strongly affects the response of tumour cells to an applied drug [41]. The maximum diffusion distance of oxygen from blood vessels is about 100 microns and cell necrosis is observable at distances of 150 microns or more from the areas supplied by blood vessels [37]. Therefore, vasculature density is strongly associated with clinical outcomes in a number of cancers [14, 17, 30].

Cancer stem cells (CSC) have been increasingly recognized as the main reason for tumour relapse and metastasis [26]. This tumour cell subpopulation is more resistant than differentiated cancer cells to most of the conventional anticancer therapies, antimitotic agents, or radiation. They also can differentiate into tumour cells of various phenotypes and are regulated by a variety of processes including Notch, Hedgehog, NF-kB, Wnt and TGF-beta pathways [9].

CAFs maintain the structural framework of a tumour and, in contrast to normal fibroblasts, have increased proliferation, enhanced extracellular matrix production and unique cytokine secretion [27]. By secreting a variety of active factors, they modulate cancer metastasis through synthesis and remodelling of the extracellular matrix (ECM), and influence angiogenesis, tumour mechanics, drug access and therapy responses [28]. Like other elements of the tumour microenvironment, CAF number and function strongly influence treatment outcomes [2, 13].

TAM’s promotion of resistance is mainly due to the secretion of a variety of cytokines that induce anti-apoptotic programs in cancer cells and stimulate tumour cell proliferation [4]. Also, activated TAMs promote metastasis by producing soluble factors [40] and can release the angiogenic molecules and express a series of enzymes involved in the regulation of angiogenesis [4].

## *1.4 Modelling in Precision Medicine*

Mathematical and computational modeling and simulations are getting more and more important in natural sciences and medical domains. The main aim is to develop and use different efficient algorithms, data structures, and visualization techniques for user-friendly and explainable communication tools for human-computer interaction. Being based on different mathematical theories and applications of ICT tools, approaches to models can take different routes, like dynamical, statistical, differential equations, game-theoretic, and so on. However, sometimes it is difficult to make a clear classification especially when modeling approaches overlap or combine producing a variety of models/systems/tools. Computer modeling and simulation of biological and medical systems encompass works with cellular subsystems, comprehensive data analyses, 2-D and 3-D visualization of the complex connections of cellular processes, and so on.

In this chapter, the focus is on tumour and drug design. Tumours are heterogeneous cellular entities, and their growth is influenced by changing microenvironment and dynamical interactions between cancerous and healthy cells. Such interactions influence changes in cell phenotypic behaviors, like proliferation, apoptosis, and migration. To obtain reliable experimental results (in spite of the fact that it enormously increases computational complexity) it is necessary to consider different spatial and temporal scales. Until recently most computational tumour models were focused on scale-specific models. As tumour growth span multiple scales, it is necessary to apply another computational approach and include modeling across different biological scales (like some of the biological spatial scales: atomic, molecular, microscopic (tissue/multicellular), and macroscopic (organ) scales). Usually, a model is considered multiscalar if considers at least two spatial scales and/or includes processes with at least two temporal scales. *"Simulating cancer behavior across multiple biological scales in space and time is increasingly being recognized as a powerful tool to refine hypotheses, focus experiments, and enable more accurate predictions."* [7].

There is still no clear and specific classification of multiscale models, but since in this chapter we are concentrated on tumour and drug design, we will briefly consider several types of models appropriate for cancer diseases according to [7].

Discrete (individual-based) modeling is based on defining a set of rules for each cell. Individual cells are represented in space and time and the simulation process tracks and updates their internal states. Discrete modeling is an adequate approach for studying carcinogenesis, genetic instability, and cell-cell and cell-matrix interactions.

For modeling larger-scale systems, a better solution is the approach where tumour tissue is presented as a continuum medium instead of individual cells. This approach is known as continuum modeling (population-based) because model variables (like cell volume fractions, nutrient, oxygen, growth factors) are described as continuous fields usually using PDE equations. However, such models cannot be used to examine individual cell dynamics and discrete events.

An approach that can mitigate shortcomings of both continuous and discrete approaches is so-called hybrid modeling. This approach assumes the coupling of a continuous model with a discrete one. Hybrid modeling assumes integration and the strengths of both continuum and discrete models i.e. it assumes interaction or coupling between at least two models not based on the same formalism. Such models can couple biological phenomena from the molecular and cellular scales to those at the tumour scale. There are different definitions of hybrid modeling but generally speaking, two categories can be distinguished: composite and adaptive hybrid modeling [7]. Applications of hybrid models are numerous like: biological networks modeling, cancer growth modeling, cancer immunology and so on [32].

A tumour is in dynamic communication with the microenvironment (that shapes the cell-level behavior) through biochemical and biophysical processes. Therefore, the growth of tumour cannot be observed as an isolated process. It must be considered as an integral part of a dynamical biochemical and biophysical environment i.e. as a 3-D multicellular system [23]. According to [23] to model and perform simulations for different aspects of cancer, a 3-D multicellular simulation platform should “simulate the birth, death, and motion of tumour cells; simulate biochemical microenvironments with multiple diffusing substrates; simulate the biomechanics of cells and the extracellular matrix; simulate the evolving blood vasculature; simulate interstitial and microvascular flow; integrate the above models, along with molecular-scale models to drive cell phenotype; integrate high-throughput experimental data to calibrate and validate models; do so reproducibly, using interoperable data formats”.

A recent and very important trend in modeling and simulation in medical domains is oriented towards a specific field in precision medicine, pharmacometrics. To optimize the treatment regimens and the design of clinical trials, personalized pharmacokinetic/pharmacodynamic (PK/PD) modeling and simulations are utilized. In pharmacokinetic the goal is to find out the relation between doses and concentration while the goal of pharmacodynamic is to find out the relation between concentration and effects that produce for an individual patient. Multimodel approaches are also very welcomed in this area where several models are simultaneously fitted predominantly based on how well they fit the data [10].

Some researchers have gone a step ahead in pharmacometrics technology including Pharmacokinetic-Pharmacodynamic/Toxicodynamic (PK-PD/TD) modeling and simulation [19]. They also work on interactions between antiemetic and anticancer drugs in order to achieve an appropriate dosing scheme and a good balance between maximum drug efficacy and minimum toxicity for individual patients. Mentioned approaches are based on inter and multidisciplinary methods that, in addition to pharmacometrics also include biomathematics, pharmacology, and several ICT disciplines with expectations to evaluate the effect of individual patient factors on drug exposure and different doses of drugs.

In this chapter, we present the results of experiments performed using EvoNano PhysiCell that represents the extended version of PhysiCell system. Original PhysiCell system [12] is a physics-based 3-D multicellular systems simulator realized as a general-purpose toolkit. Multicellular systems are used in different domains including cancer (metastasis, growth, drug design for tumour cells). To achieve good results during simulation it is important to observe how individual cells grow, divide, interact, move, and die. Tissue-scale dynamics should be studied in the microenvironment as usually cells are affected by biochemical and biophysical signals. So, reliable simulation for multicellular systems should include: tissue microenvironments (the “stage”) with multiple diffusing chemical signals (oxygen, drugs, and so on); and the dynamics tissue microenvironments i.e. many interacting cells (the “players” upon the stage). *It builds upon a multi-substrate biotransport solver to link cell phenotype to multiple diffusing substrates and signaling factors. It includes biologically-driven sub-models for cell cycling, apoptosis, necrosis, solid and fluid volume changes, mechanics, and motility “out of the box”* [12].

PhysiCell is a powerful framework realized in a modular manner allowing a wide range of users to extend, rewrite, or even replace its originally implemented functions. The system also has been parallelized and supports the dynamics and interactions of even millions of cells in 3-D microenvironments, with microenvironment-dependent phenotypes.

# 2 PhysiCell Based Simulator for Precision Medicine

This section will give an overview of new concepts introduced in the original PhysiCell which are required for a high-quality precision medicine simulation. These concepts include several types of cells involved in the simulation (cancer cells, cancer stem cells, healthy cells, vascular cells, and cancer-associated fibroblast (CAF) cells) and several substrates which are added into PhysiCell microenvironment (angiogenic factor, prostaglandin, cytokine, drug substrate). The first subsection will give a high-level overview of these concepts by providing explanations about processes that take place in the simulation and by explaining all necessary parameters. The second subsection will provide low-level overview with important implementation details.

## *2.1 Functionalities and Features*

This section will provide an overview of introduced cell types and substrates in the order in which they are implemented. This is important since some later concepts require the presence of particular initial concepts.

*Cancer and cancer stem cells*

Cancer stem cells (CSC) are the first types of cells introduced in EvoNano PhysiCell extension. Together with Differentiated cancer cells (DCC) they are the main representatives of the potential tumour tissue. These two cell types share common behaviour, while some parameters (e.g. for apoptosis, necrosis, division) are different. CSCs are generally considered more resistant regarding death rate, division, resistance to drugs, etc.

DCC are part of the original PhysiCell simulator and they are always simulated in our simulator, while the simulation of CSC is controlled through parameter `cancer_stem_cells_enabled` of the Boolean type. The initial number of DCCs in the simulation is given in the parameter `random_cancer_cells`. The general behaviour of these cells is controlled through several simulation parameters:

- `random_cancer_cells` which defines the number of DCCs in the initial simulation set-up.
- `cancer_cell_persistence_time` and `cancer_stem_cell_persistence__time` define the persistence time for DCCs and CSCs respectively. For both cell types, this value is set to 15 minutes.
- `cancer_cell_migration_speed` and `cancer_stem_cell_migration_speed` define the migration velocity for DCCs and CSCs respectively. For both cell types, this value is set to 0.25 micrometers per minute.
- `cancer_cell_relative_adhesion` and `cancer_stem_cell_relative_adhesion` define the value of relative adhesion for DCCs and CSCs respesctively. This value is interpreted as the force for resistance to deformation and/or volume exclusion [29]. For both cell types, this parameter value is set to 0.05 (dimensionless).

- cancer_cell_apoptosis_rate and cancer_stem_cell_apoptosis_rate define the rate for entering the cell into apoptosis (programmed cell death). The probability of entering in the apoptosis death state is proportional to this parameter and to the delta time [12]. CSC dies very rarely at least via apoptosis death model, so the value of this parameter is set to zero (1/minute). The value for DCC is set to 4.065e-5.

The process of DCC and CSC division is more complicated and it is controlled through several parameters which will be described below. One DCC can be divided into two DCCs or into one DCC and one CSC (with some probability). One CSC can be divided into: two CSC, two DCC, and one CSC and one DCC (all outcomes with defined probability). This process of division is controlled through the following simulation parameters:

- cancer_cell_division_rate and cancer_stem_cell_division_rate define the division rate for DCCs and CSCs respectively. This value for both types of cells is set to 0.005 (dimensionless).
- CSC_probability_CC_division defines the probability for making one CSC in DCC division. This probability is set to value 0.05 (5%). Consequently, one DCC will be divided in two DCCs in 95% of cases.
- symmetric_division_probability defines the probability of splitting one CSC into two cells of the same type (two DCC or two CSC). This value is set to to 0.01 (1%) so in 99% of divisions one CSC will be divided in one DCC and one CSC.
- symmetric_division_double_CC_probability defines the probability of dividing one CSC into two DCCs when symmetric division is selected (previous item). This value is set to 0.01 (1%) so in 99% of symmetric divisions one CSC will be divided into two CSCs. Since the probability of symmetric division is 1%, and that under these circumstances the probability of creating two DCCs from one CSC is 1%, overall the probability of creating two DCCs from one CSC is just 0.01%.

The parameters cancer_stem_cell_o2_necrosis_threshold and cancer_stem_cell_o2_necrosis_max are defined only for the CSCs, and they are used to simulate higher resistance of CSCs to the lack of oxygen and for the entering necrosis process. When the level of oxygen is lower than o2_necrosis_max (PhysiCell parameter) the cell immediately becomes necrotic. When the level of oxigen is between o2_necrosis_max and o2_necrosis_threshold (also PhysiCell parameter) the necrotic process is started according to stochastic necrosis model [12]. For the values of oxygen higher than o2_necrosis_threshold, there is no necrosis. The default PhysiCell values which are also applied for DCCs are o2_necrosis_threshold=5.0 and o2_necrosis_max=2.5. For the CSCs, these values are significantly lower which simulate higher resistance to low oxygen values: cancer_stem_cell_o2_necrosis_threshold=0.1 and cancer_stem_cell_o2_necrosis_max=0.01.

There are still some more parameters that refer to DCCs and CSCs that are connected to the other concepts (vascularity, drug diffusion etc.) and they will be described in corresponding sub-sections.

*Healthy cells*

Healthy cells are introduced in EvoNano PhysiCell extension to simulate the impact of different drugs on healthy tissue. These cells are not so important for the mechanism of tumour growth and dynamics so many aspects of these cells are turned off. Healthy cells are immobile and they do not divide nor die, either via apoptosis or necrosis.

The presence of healthy cells in the simulation is controlled via parameter `healthy_cells_active` of type Boolean. Healthy cells are randomly distributed throughout the initial space. The number of healthy cells in the initial space is controlled via parameter `random_healthy_cells` (dimensionless). When the initial space needs expansion, due to the tumour growth, the number of healthy cells is controlled with parameter `random_healthy_cells_expansion_factor` (dimensionless).

*Vascular cells and an angiogenic factor*

This mechanism simulates the growth of vascular networks within and around tumour tissue which ensures an additional supply of oxygen to the growing tumour. An angiogenic factor and vascular cells mechanism can be turned on/off with parameter `vascularity_active` of type Boolean. When turned on, this mechanism introduces a new substrate into simulation: an angiogenic factor (AF). In that case, all cancer cells secrete AF which represents an activation signal to vascular cells. Vascular cells then try to improve the creation of a vascular network in the direction of a higher concentration of AF. With such a mechanism, a growing tumour indicates the higher production of vascular cells, which results in a richer vascular network, which as a consequence provides a higher supply of oxygen to tumour tissue.

The main parameters (diffusion coefficient and decay rate) for AF are taken from [11]. The amount of AF which is secreted from one cancer cell is given in the parameter `angiogen_in_cancer_cell`.

Vascular cells are initially placed randomly in the simulation space. The number of the vascular cells in the initial space is specified by a parameter `vascular_seed_points` (current value is 1). Afterward, they are dividing in the direction of high AF density if the concentration of AF is above the defined threshold value. This threshold is given by parameter `critical_angiogen_to_expand_vascularity` (current value is set to 0.055 according to [39]). In such a way, we simulate the growth of capillaries and blood supply of tumour. However, not all vascular cells in whose environment the concentration of AF is higher than the threshold value divide because that would cause the exponential growth of the vascular network. To prevent this exponential growth we introduced the probability of dividing vascular cells in the presence of high AF concentration. This value is specified in the parameter `vascular_cell_expansion_probability`, the current value is 0.01 which is determined empirically.

When the concentration of AF is too high (higher than parameter critical_angiogen_to_stop_o2_secretion) the secretion of oxygen from vascular cells stops. With this mechanisms, we simulate the death of vascular cells which happen in the deep interior of the tumour as a result of tumour necrosis (necrotic core).

In the situation when the simulation space should be expanded, the expansion of vascular network is controlled with parameter expand_vascularity_when_ _expanding_space (Boolean parameter, default value is true). The quantity of this expansion is controlled through expansion_scale_factor parameter (current value is 1.75, empirically adjusted).

When the functionality of vascular cells is defined, the process of metastasis can be also introduced. In metastasis, cancer cells break away from the original (primary) tumour, travel through the vascular system, and form a new tumour in other organs or tissues of the body. Therefore, we introduce two more parameters which control the process of detachment of CSC: CSC_detachment_probability and CSC_detachment_critical_proximity. The first parameter defines the probability of CSC which is near some vascular cell to detach from the original tumour and enter the vascular system. The threshold distance which defines what is near is defined with the second parameter. Until now, we have not experimented with the metastasis process yet, so these parameters are set to the following values: 0.0 and 1000000, respectively.

*A prostaglandin*

Prostaglandins are a family of signaling molecules that regulate the invasive behaviour of cancer cells. Instead of implementing all members of the family as separate substrates, we implement an abstract prostaglandin substrate that regulates DCC-CSC conversion. In our implementation, a prostaglandin is secreted from cells when a cell dies either by apoptosis or necrosis. The simulation of this substrate is turned off/on with Boolean parameter prostaglandin_active. The parameters for diffusion and decay rate are taken as for substrate cytokine (described below) [18] since there are no reliable experimental data for prostaglandins.

The amount of a prostaglandin that is secreted from a cell is controlled via parameter prostaglandin_in_cancer_cell. This value is set to 8.8e-7.

High values of prostaglandin trigger the conversion of some DCCs to CSCs. The DCC->CSC conversion is triggered when the concentration of prostaglandin in substrate exceeds the value of parameter cancer_cell_prostaglandin_thr. Of course, only a small amount of DCCs is converted into CSC in the presence of high prostaglandin values. The quantity of conversions is controlled with parameter prostaglandin_conversion_probability which represents the probability of DCC->CSC conversion in the presence of prostaglandin. This parameter is empirically determined to value 0.00008, in order to obtain expected tumour growth and expected number of CSCs. Also, the number of prostaglandin induced DCC->CSC conversion is tracked within the simulation as an important indicator, and it is reported in the log file.

*CAF cells and cytokine*

Cancer-Associated Fibroblasts (CAFs) are special types of tumour cells that are responsible for providing physical support for other tumour cells and for secreting cytokines [22]. These cells appear near vascular cells and it is important to preserve the expected ratio of vascular and CAF cells. This ratio is defined with parameter `CAF_vascular_ratio` (value set to 0.6, determined empirically). The simulation of these cells and accompanying substrate cytokine is turned off/on with Boolean parameter `CAF_active`.

The division of CAF cells is turned off since it is observed that they appear in a random manner near the vascular network. We simulate the same behaviour in our model. Also, the apoptosis is turned off since CAF cells are unable to undergo apoptosis. The necrosis still exists although these cells are more resistant to low oxygen levels than DCCs (while CSCs are more resistant than CAF cells). The oxygen values which indicate necrosis are defined with parameters `caf_o2_necrosis_threshold` and `caf_o2_necrosis_max` in the same fashion as with CSCs. These values are set to values 1 and 0.1 respectively, to simulate oxygen resistance between DCCs and CSCs.

As already mentioned, CAF cells secrete cytokine, another substrate that is simulated. Te cytokine diffusion coefficient and decay rate parameters are taken from [18]. The amount of cytokine which is secreted from one CAF cell can be adjusted through parameter `cytokine_in_CAF`. This value is set up on 8.8e-7 according to [18].

The cytokine affects neighboring cells in a similar manner as prostaglandin. In the presence of high cytokine levels, some DCCs convert to CSCs. The critical cytokine concentration which triggers DCC->CSC conversion is defined in parameter `cancer_cell_cytokine_resistance`. The probability of DCC->CSC conversion in the presence of high cytokine values is given in parameter `cytokine_conversion_probability`. The values of these two parameters are set up on values 1e-12 and 0.01, after extensive experiments whose goal was to replicate the desired tumour growth and desired cell type ratios.

CAF cells have one more parameter: `o2_stop_cytokine_threshold`. This parameter defines the lowest value of oxygen for which the CAF cell is capable to secrete cytokine. Bellow this oxygen value (current value is 10) CAF cell does not secrete cytokine, although it does not die (oxygen levels for necrosis are much lower: 0.1 and 1). If oxygen values later return to values higher than this threshold CAF cell “wakes up” and secretes cytokine again.

*Drug diffusion mechanism*

The mechanism of drug administration is supported on the level of the substrate. We simulate nanoparticle-based cancer therapies by simulating functionalized nanoparticles (NP), i.e. the NPs with an attached particular drug. That drug can affect tumour in different ways: CSC and DCC death, stopping the division of cancer cells, or destroying tumour vascular network. In the current simulator, we support only drugs that cause cancer cell death, but the other mechanisms will be also supported in the future.

**Table 1** NP resistance values for different cell types

| | |
|---|---|
| `healthy_cell_NP_resistance` | 0.7 |
| `cancer_cell_NP_resistance_lower_bound` | 0.3 |
| `cancer_cell_NP_resistance_upper_bound` | 0.5 |
| `cancer_stem_cell_NP_resistance_lower_bound` | 0.5 |
| `cancer_stem_cell_NP_resistance_upper_bound` | 0.7 |

The simulation of the drug mechanism can be turned off/on via the Boolean parameter `NP_active`. NP substrate is secreted from vascular cells similarly to oxygen. The amount of secreted NPs from one vascular cell can be controlled through parameter `NP_in_vascular_cell`. Since the administration of the drug does not start at the same time as the tumour creation but only after a tumour is detected and analyzed, we simulate this behaviour with parameter `cancer_cells_to_activate_NP`, which specifies the number of cancer cells when the drug administration starts.

Secreted NP has an influence on all present cells. In the current version, if the concentration of NP exceeds some threshold value, the cell dies by apoptosis. The threshold values are different for all cells since different types of cells have different resistance to a particular drug. These threshold values are controlled by several parameters. These parameters and their current values are given in Table 1. All healthy cells have the same value of NP resistance. On the other hand, each DCC and CSC acquires a unique value from the interval `[lower_bound..upper_bound]`.

Figure 1 illustrates all previously introduced concepts. It shows the state of the tumour after 15 simulation days. This condition is obtained using previously described parameters and their values which are mentioned in the text. The cell types have the following colors:

- grey: DCC
- blue: CSC
- maroon: Vascular cell
- yellow: CAF cell
- orange: Necrotic cell

Moreover, the empty space in the middle of tumour represents necrotic core, but the old necrotic cells are removed from the simulation due to efficiency issues.

## *2.2 EvoNano PhysiCell Implementation*

The implementation of the EvoNano PhysiCell simulator for precision medicine is organized into several custom C/C++ modules (modules put in `custom_modules` folder of PhysiCell) and the main module realizing necessary initialization and the main loop of the simulator. The custom modules are:

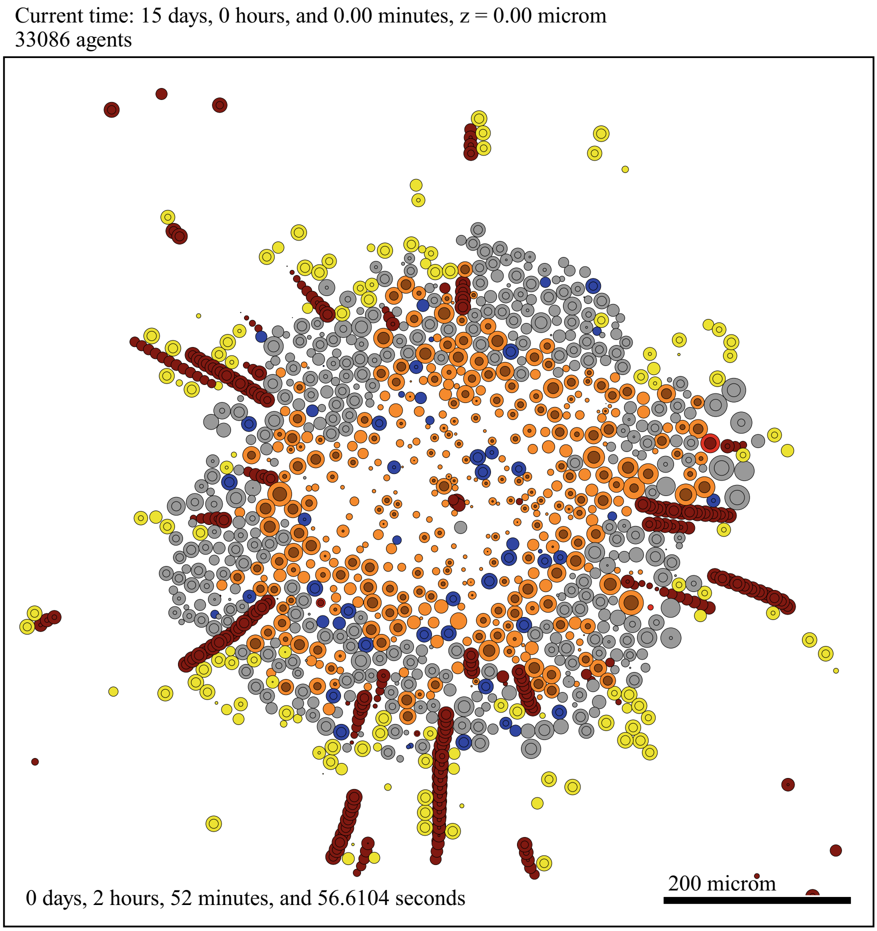

**Fig. 1** An example of tumour after 15 simulation days

1. `cancc` that contains functionalities related to cancer cells supported by the simulator,
2. `csvlog` that defines functions for emitting log messages in CSV (comma separated values) log files,
3. `evo_nano_3D` that defines functions for creating cell types, initializing cell definitions, initializing the microenvironment and tissue and realizing specific rules for cancer cell creation and division,
4. `healthyc` that implements normal (non-cancer, healthy) cells,
5. `spexp` that implements the microenviroment with dynamic space expansion capabilities,
6. `utils` that contains various utility functions for sampling random numbers,

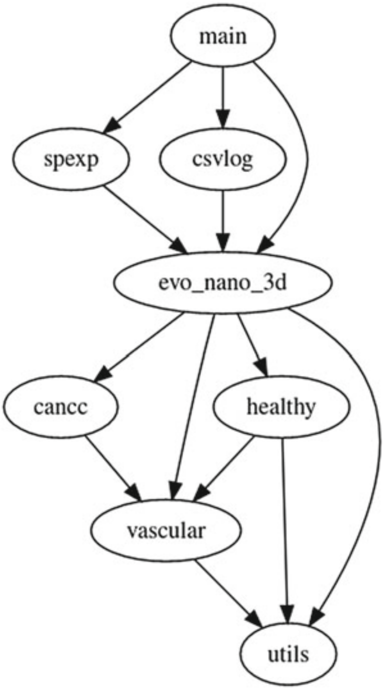

**Fig. 2** Dependencies between modules of the EvoNano PhysiCell simulator

7. `vascular` that implements functionalities related to vascular cells and vascular network growth.

As usually for C/C++ modules, each module of our simulator is realized in two files, i.e. one header file exposing the interface of the module to other modules and one cpp file containing the full implementation of the specified functionalities. Figure 2 shows dependencies between the modules of the simulator including also the main module.

*Utils module*
The `utils` module is located at the bottom of the module dependency graph and it is the only module that does not depend on other EvoNano PhysiCell modules. This module implements two routines for sampling real-valued random numbers from uniform distributions: one for sampling a real number from the interval $[A, B]$ and one for sampling a real number from $[A, B]$ with a given probability $p$ or from the interval $[C, D]$ with 1 - $p$ probability ($A$, $B$, $C$, $D$ and $p$ are input parameters). This module also defines two methods returning the borders of the simulation space inside which are sampled coordinates of:

1. initially created cancer and health cells (cells that are formed when the simulation starts), and
2. healthy and vascular cells after extending the simulation space.

*Vascular module*

In EvoNano PhysiCell, blood vessels are implemented as chains of vascular cells. This means that one capillary is composed of adjoined vascular cells, while the complete vascular network is a set of capillaries (which are not necessarily mutually inter-connected). The `vascular` module defines various routines related to vascular cells and their functions. This module defines 4 important global variables:

1. `bool vascularity_active`—a boolean variable indicating whether the vascular network is simulated,
2. `bool NP_simul`—a boolean variable indicating whether NP (nano-particles or drugs) is simulated,
3. `bool NP_active`—a boolean variable indicating whether NP diffusion is active when NP is simulated (vascular cells may not start diffusing NP immediately when the simulation starts, but after some time specified in the configuration file; NP diffusion may also stop and later resume in more complex simulation scenarios), and
4. `std::vector<Cell*>* vascular_cells`—a pointer to vector containing pointers to all vascular cells (each cell PhysiCell is an instance of the `Cell` class).

The module exposes the following functions to other EvoNano PhysiCell modules:

1. `void init_vascular_network()`—creates the initial vasculature when the simulation starts,
2. `void expand_vascular_network(double sm)`—creates new vascular cells in a new space region after the simulation space dynamically expands (`sm` is the seed multiplier),
3. `void init_vascular_cells()`—initializes vascularity configuration parameters and creates a PhysiCell cell definition object for vascular cells,
4. `Cell* create_vascular_cell(double x, double y, double z)`—creates a new vascular cell at the given position $(x, y, z)$ and returns the pointer to it, and
5. `void divide_vascular_cell(Cell* c)`—"divides" the vascular cell $c$ by creating a new vascular cell in the immediate neighborhood of $c$.

The initial vascular network is formed by creating $k$ vascular seed cells at the beginning of the simulation, where $k$ is one of the EvoNano PhysiCell configuration parameters. The coordinates of each seed cell are randomly sampled. The total amount of $k\dot{s}$ vascular seed cells are also created when the simulation space is expanded, where $s$ is the seed multiplier. Their coordinates are also randomly sampled but considering only the newly created regions of space.

Vascular cells are defined as non-divisible, immortal, and non-motile cells of constant volume as shown by the program code fragment given below. Two custom variables are also associated to each vascular cell: `num_expansions` and `flagged_for_division`. The value of the first variable reflects how many times a vascular cell was ”divided”. The second variable indicates whether the cell is designated to be ”divided” in the next simulation cycle. The vascular cell can be ”divided” exactly once, i.e. if num_expansions is equal to 0 then the cell can be flagged for division and after division num_expansions is increased by 1.

**Program code fragment: initialization of vascular cell definition**

```
Cell_Definition vascular_cell_def;

void create_vascular_cell_definition() {
    ...
    vascular_cell_def = cell_defaults;

    // turn off motility
    vascular_cell_def.phenotype.motility.is_motile = false;

    // turn off cell division
    int s = live.find_phase_index(PhysiCell_constants::live);
    int e = live.find_phase_index(PhysiCell_constants::live);
    vascular_cell_def.phenotype.
        cycle.data.transition_rate(s, e) = 0.0;

    // turn off cell death
    int apoptosis = cell_defaults.phenotype.
        death.find_death_model_index("Apoptosis");
    vascular_cell_def.phenotype.death.rates[apoptosis] = 0.0;
    int necrosis = cell_defaults.phenotype.
        death.find_death_model_index("Necrosis");
    vascular_cell_def.phenotype.death.rates[necrosis] = 0.0;

    // turn off volume updates
    vascular_cell_def.functions.volume_update_function = NULL;

    // custom variables
    vascular_cell_def.custom_data.add_variable
        ("num_expansions", "dimensionless", 0);
    vascular_cell_def.custom_data.add_variable
        ("flagged_for_division", "dimensionless", 0);

    // set function performing phenotype updates
    vascular_cell_def.functions.update_phenotype =
        vascular_cell_update_phenotype;
    ...
}
```

The function for creating vascular cells reduces to the PhysiCell function for creating cells (also called PhysiCell agents) with the vascular cell definition defined in the `vascular` module. This function also adds the newly created vascular cell to the vector of all vascular cells.

The function for updating the phenotype of vascular cells defines how vascular cells react to the angiogen. Let $a$ denotes the concentration of the angiogen near a vascular cell $v$. The impact of $a$ to $v$ is defined by the following rules implemented in the function for handling the angiogen:

- If $a$ is higher than the angiogen threshold to stop oxygen secretion then $c$ immediately stops releasing oxygen. When $a$ drops below this threshold then $c$ reactivates its oxygen release.
- If $a$ is higher than the critical angiogen threshold to expand the vascular network and $c$ was not previously expanded then it is considered as the candidate for ”division”. The cell is marked for division with a certain probability that is one of EvoNano PhysiCell parameters controlling vascular network growth.

The function for phenotype updates of vascular cells also handles NP. If NP is simulated and active then vascular cells release a certain concentration of NP into the environment.

As already emphasized, the division of a vascular cell is actually the creation of a new vascular cell. The new cell is created to touch the cell marked for division in the opposite direction of the angiogen gradient. The function for dividing vascular cells also increases `num_expansions` variable to prevent further divisions and ensure that vascular cells form a chain-like structure.

*Healthy module*

Healthy (normal, non-cancer) cells are implemented in the `healthy` module. In reality, normal cells are divisible and mortal. However, their division rates are negligible compared to cancer cells. Thus, normal cells are in EvoNano PhysiCell implemented as non-divisible cells. Additionally, a normal cell can die by entering apoptosis only if it is exposed to a high concentration of NP.

The module defines the following two global variables:

- `bool healthy_cells_active`—a boolean flag indicating whether healthy cells are simulated, and
- `int NP_HC_deaths`—the total number of normal cell deaths due to critical exposure to NP.

The public functionalities of the module are given by the following functions:

- `void init_healthy_cells()`—initializes configuration parameters and creates the definition object for healthy cells.
- `Cell* create_healthy_cell(double x, double y, double z)`—creates and returns a new healthy cell at given coordinates.
- `void expand_healthy_cells(int m)`—creates new healthy cells in the new region of space after the simulation space is dynamically expanded ($m$ is a

multiplier factor, i.e. the total number of newly created normal cells is $k\dot{m}$, where $k$ is the number of normal cells created at the start of the simulation).

The function defining how healthy cells react to NP is called by the function performing phenotype updates at regular time intervals. This function checks whether the concentration of the NP near a healthy cell is higher than critical. If it is then the apoptotic death of the cell is activated (see the program code fragment given below).

**Program code fragment: function for handling NP by healthy cells**

```
void healthy_cell_handle_NP(Cell* c) {
    // is NP simulated?
    if (!NP_simul)
        return;

    // is NP active?
    if (!NP_active)
        return;

    int NP = microenvironment.find_density_index("NP");
    double NP_near = microenvironment.
        nearest_density_vector(c->position)[NP];

    if (NP_near > HC_NP_resistance) {
        int apoptosis =  cell_defaults.phenotype.
            death.find_death_model_index("Apoptosis");
        c->start_death(apoptosis);
        ++NP_HC_deaths;
    }
}
```

*Cancc module*

The functionalities of three different types of cancer cells, DCC, CSC, and CAF, are implemented in the `cancc` module. This module defines 6 global variables. The first three global variables are boolean flags that can be set to enable or disable certain functionalities related to cancer cells:

1. `bool cancer_stem_cells_enabled`—CSC cells are simulated if this flag is set to true, otherwise CSC cells are not created during the simulation.
2. `bool prostaglandin_simul`—if this flag is set to true then DCC and CSC cells release prostaglandin during their death cycle.
3. `bool cytokine_caf_simul`—this boolean flag indicates whether CAF cancer cells releasing cytokine are simulated or not.

The next three global variables are counters:

1. NP_CC_deaths—the number of cancer cell deaths caused by NP.
2. prostaglandin_conversions—the number of DCC converted to CSC due to critical exposure to prostaglandin.
3. cytokine_conversions—the number of DCC converted to CSC due to critical exposure to cytokine.

The module exports the following functions:

- void init_cancer_cells()—initializes configuration parameters related to cancer cells and creates definition objects for all three cancer cell types.
- Cell* create_cancer_cell(double x, double y, double z, bool s)—creates a new DCC cell (if *s* is false) or a new CSC cell (if *s* is true) at the given position.
- Cell* create_CAF_cell(double x, double y, double z)—creates a new CAF cell at the given position.
- void divide_cancer_cell(Cell* c)—divides a DCC cell into two cancer cells.
- void divide_cancer_stem_cell(Cell* c)—divides a CSC cell into two cancer cells.

The function init_cancer_cells creates three different instances of the Cell_Definition class, one object per cancer cell type. These objects reflect the main differences between different cancer cell types, i.e. each cell type defines its cell division rate, motility parameters, adhesion strengths, apoptosis rates, oxygen thresholds to enter necrosis, secretion, and uptake parameters for relevant substrates (all those parameters are specified in the configuration file). Two custom data variables are associated with each cancer cell:

- NP_resistance—the critical concentration of NP to start apoptotic cell death.
- NP_prev—the concentration of NP near the cell in the previous iteration of the simulation. This variable enables quantifying temporal change of NP near the cell and to adjust cell division rate accordingly.

All cancer cells update their base phenotype parameters according to the oxygen phenotype update model defined by PhysiCell. However, each cancer cell type has its own phenotype update function that additionally defines how cancer cells react with substrates and release them. The cancc module implements customized phenotype update functions relying on the standard oxygen phenotype update model that additionally defines how cancer cells release and interact with substrates. The phenotype update function for DCC is shown in the code inset given below.

**Program code fragment: phenotype update function for DCC.**

```
void cancer_cell_phenotype_update(
    Cell* c, Phenotype& phenotype, double dt)
{
    // dying cancer cells release prostaglandin
    if (c->phenotype.death.dead && prostaglandin_simul) {
        int p = microenvironment.
            find_density_index("prostaglandin");
        microenvironment.
            nearest_density_vector(c->position)[p] =
            PROSTAGLANDIN_IN_CANCER;

        return;
    }

    // standard oxygen-based phenotype update model
    update_cell_and_death_parameters_O2_based(
        c, phenotype, dt);

    // phenotype updates based on NP
    cancer_cell_handle_NP(c);

    if (prostaglandin_simul) {
        // check for prostaglandin-based DCC->CSC conversion
        cancer_cell_handle_prostaglandin(c);
    }

    if (cytokine_caf_simul) {
        // check for cytokine-based DCC->CSC conversion
        cancer_cell_handle_cytokine(c);
    }
}
```

The function for handling NP operates according to the following rules. If the concentration of NP near the cancer cell is higher than the NP resistance threshold for that cell then the cell enters apoptotic death. The concentration of NP near the cancer cell also affect the cell division rate: if it is close to zero then the cell division rate is returned to the original value from the configuration file, otherwise, the cell division rate is divided by the increase of NP concentration (i.e., more NP implies slower division of cancer cells).

Functions for handling prostaglandin and cytokine are structurally similar. If the concentration of prostaglandin (resp. cytokine) is higher than critical then the DCC cell is converted into a CSC cell with a given probability that controls the conversion rate.

**Program code fragment: function for handling prostaglandin.**

```
void cancer_cell_handle_prostaglandin(Cell* c) {
    int p = microenvironment.
        find_density_index("prostaglandin");
    double pnear = microenvironment.
        nearest_density_vector(c->position)[p];

    double prob = UniformRandom();
    if (prostaglandin_near > cc_prostaglandin_thr &&
        prob <= prost_conv_prob)
    {
        // DCC -> CSC conversion
        c->convert_to_cell_definition(cancer_stem_cell_def);
        prostaglandin_conversions++;
    }
}
```

Cancer cells have heterogeneous resistances to NP. This means the resistance to NP is randomly sampled from a uniform distribution where each cell type has its own upper and lower bounds of the distribution. The NP threshold above which a cell enters apoptotic death is determined upon cell creation as it can be seen in the following program code fragment.

**Program code fragment: function for creating DCC and CSC cells**

```
Cell* create_cancer_cell(
    double x, double y, double z, bool stem)
{
    Cell* c;

    if (stem)
        c = create_cell(cancer_stem_cell_def);
    else
        c = create_cell(cancer_cell_def);

    c->assign_position(x, y, z);
```

```
        // determine the NP resistance level
        if (NP_simul) {
            int ind = c->custom_data.
                find_variable_index("NP_resistance");

            double u = UniformRandom();
            if (stem)
                c->custom_data[ind] = CSC_NP_LOWER +
                    u * (CSC_NP_UPPER - CSC_NP_LOWER);
            else
                c->custom_data[ind] = CC_NP_LOWER +
                    u * (CC_NP_UPPER - CC_NP_LOWER);
        }

        return c;
    }
```

EvoNano PhysiCell functions for dividing cancer cells rely on the `divide()` method implemented in the PhysiCell `Cell` class. This method called on an arbitrary cell *c* returns a new cell *n* that is placed near *c*. The new cell *n* has the same characteristic as *c* and the volume of both cells is equal to half of the volume of *c* prior to division. The function for dividing DCC always creates a new DCC if CSC cells are not simulated. If CSC cells are simulated then a new DCC is converted to a CSC with a certain probability (see the program code fragment given below). The division of CSC cells is realized similarly by following appropriate rules for symmetric and asymmetric CSC division.

**Program code fragment: function for dividing DCC cells**

```
    void divide_cancer_cell(Cell* c) {
        if (!cancer_stem_cells_enabled) {
            Cell* new_cell = c->divide();
            return;
        }

        double pCSC = UniformRandom();
        Cell* new_cell = c->divide();

        // CSCP: the probability to convert to a CSC
        if (pCSC <= CSCP) {
            new_cell->convert_to_cell_definition(
                cancer_stem_cell_def);
        }
    }
```

*Evo_nano_3d module*

The evo_nano_3d module implements the main rules of the simulator (cell division and detachment rules) and various setup functions that are called from the main module. This module also implements a function providing a periodic report about all cells and a custom cell coloring function for visualizations produced by PhysiCell. This module initializes the default cell definition object that is reused to build definitions of all supported cell types. The default cell definition object specifies the default cell cycle model and oxygen uptake and secretion parameters.

The module defines the following two setup functions:

- void setup_microenvironment() to initialize the BioFVM microenvironment (please recall that BioFVM is a library used by PhysiCell to simulate diffusion)
- void setup_tissue() to setup the vascular network and create initial DCC and healthy cells that are randomly placed in the environment. The number of initially created DCC and healthy cells is specified in the configuration file.

The function for making cell report iterates through the vector of all cells (this vector is maintained by PhysiCell) and makes the distribution of cells per cell type. This function also counts how many cells entered the death cycle.

The function defining EvoNano PhysiCell cell division rules is an umbrella function: it iterates through the vector of cells marked for division and depending on the cell type calls the appropriate cell division function defined in other EvoNano PhysiCell modules. Additionally, this function takes care of CAF cells by creating them near vascular cells as the vascular network grows such that the ratio of CAF to vascular cells never exceeds the bound specified in the configuration file.

**Program code fragment: function implementing cell division rules.**

```
void evonano_cell_division(
    Cell_Container* cell_container, double t)
{
    for (int i = 0;
         i < cell_container->cells_ready_to_divide.size();
         i++)
    {
        Cell* c = cell_container->cells_ready_to_divide[i];
        int type = c->type;
```

```
          if (type == CANCER_CELL_TYPE)
              divide_cancer_cell(c);
          else if (type == CANCER_STEM_CELL_TYPE)
              divide_cancer_stem_cell(c);
          else if (type == VASCULAR_CELL_TYPE) {
              divide_vascular_cell(c);
              if (cytokine_caf_simul)
                  create_CAF_near_vascular(c);
          } else
              c->divide();
      }

      cell_container->cells_ready_to_divide.clear();
  }
```

The module also defines the CSC detachment rule. Namely, if a CSC cell is close to some vascular cell then it can go to the vascular network with a certain probability. The number of detached CSC cells is a proxy indicator of the degree of tumour metastasis. Both the critical distance and the detachment probability are specified in the configuration file. Also, the user can disable CSC detachments.

*Spexp module*
The BioFVM environment in PhysiCell has constant dimensions. For a large tumour it is necessary to allocate large dimensions of the simulation space prior to starting the simulation. If the tumour evolves slowly then BioFVM most of the time simulates diffusion through empty space regions located far from actual cancer cells, which in turn may drastically slow down the simulation. Thus, we decided to make a general-purpose extension of PhysiCell that enables dynamic space expansion in run time. With this capability, it is possible to start the simulation in a relatively small space that is appropriately expanded as the tumour grows. Consequently, BioFVM does not simulate diffusion through empty space regions which significantly improves the execution time for large tumours.

Dynamic space expansion functionalities are implemented in the `spexp` module. This module defines two public functions called from the `main` module:

1. `bool check_conditions_to_expand()`
2. `void expand_microenvironment(double voxel_size)`

The first function checks if there is a cell close to the space borders. If there is a cell such that its distance to the closest border is smaller than some threshold then the first function returns true indicating that the space should be expanded by calling the second function. The distance threshold is one of the simulation parameters and it is specified in the configuration file.

The second function for expanding the microenvironment first makes its copy, i.e. the density of each substrate in each voxel is saved in a temporary 3D matrix. Then, a new bigger microenvironment is created and substrate densities are restored from the temporary matrix. The increase along all space axes is specified in the configuration file. The new microenvironment is registered to each cell by calling the appropriate method. Finally, all substrate gradient vectors and the BioFVM diffusion solver are reinitialized.

*Csvlog module*
The `csvlog` module is the simplest EvoNano PhysiCell module. This module logs the state of the simulation at regular time intervals by exporting relevant information for all cells (cell type, position, and volume) into a csv file. The concentration of oxygen near each active cell is also recorded.

*Main module*
The `main` module defines the main function which starts the simulation and implements the main simulation loop that executes for the specified maximum simulation time. Prior to entering the main simulation loop, all necessary initialization steps are performed:

1. the configuration file is parsed and loaded into an internal data structures,
2. the random number generator is initialized,
3. cell type definition objects are instantiated, and
4. the BioFVM microenvironment is created and populated with initially created cells.

The main loop takes care of events occurring at different time scales: diffusion of substrates (0.1 min scale), cell mechanics updates (1 min scale) and cell processes (10–100 min scale). Each iteration of the main loop corresponds to one event at the lowest time scale (substrate diffusion), whereas events at higher time scales are triggered by checking appropriate time step sizes. In each iteration of the main loop the following actions are performed:

1. BioFVM functions are executed to simulate the diffusion of the specified substrates and update the microenvrionment accordingly.
2. It is checked whether the simulation space should be expanded (by calling the function from the `spexp` module). If the criterion for expanding space is satisfied then the microenviroment, the vascular network, and the set of healthy (normal) cells are expanded.
3. It is checked whether NP should be activated in the case that it is simulated. The diffusion of NP activates either after the specified number of days or after the tumour reaches some critical mass.
4. If it is time for cell events then the position, volume, and phenotype of each cell is updated and the routines from the `evo_nano_3d` module implementing custom EvoNano cell division and detachment rules are called.
5. It is checked whether the simulation state should be logged and appropriate log operations are performed.

## 3 Results

The configuration parameters of EvoNano PhysiCell could be calibrated such that the growth of a simulated tumour is aligned with some theoretical model. To set parameters of the simulator corresponding to normal physiological conditions we use the Gompertzian growth model [31]. The Gompertzian growth is given by the following equation

$$V(t) = V_0 e^{\frac{\alpha}{\beta}(1-e^{-\beta t})},$$

where $V(t)$ is the total tumour volume at time $t$, $V_0$ is the initial tumour volume, and $\alpha$ and $\beta$ are constants regulating the growth rate. In our experiments we use $\alpha = 0.58$ and $\beta = 0.072$ according to the recent study by Vaghi et al. [36]. In our simulations, we always start from 100 regular cancer cells (DCCs) that are randomly placed within the initial simulation space of a small dimension ($x = y = z = 200\mu$m). Then, we simulate 30 days of tumour growth with dynamic space expansion capabilities described in the previous section. Since the volume of a single cell when crated by the simulator is 0.000002494 mm$^3$, the volume $V_0 = 0.0002494$ mm$^3$. Figure 3 shows the growth of a simulated tumour after calibrating the simulator's parameters according to the previously described Gompertzian growth model. The volume of the tumour is computed as the total volume of all cancer cells and it is averaged over 10 simulations. It can be seen that only in the first 10 days there are small deviations from the theoretical model and after that initial period the volume of the simulated tumour is almost identical to the predictions by the theoretical model.

The evolution of the total number of live cancer cells, dying cancer cells and vascular cells is shown in Fig. 4. One randomly place vascular cell is created at the beginning of each simulation in order to initialize the vascular network. This network

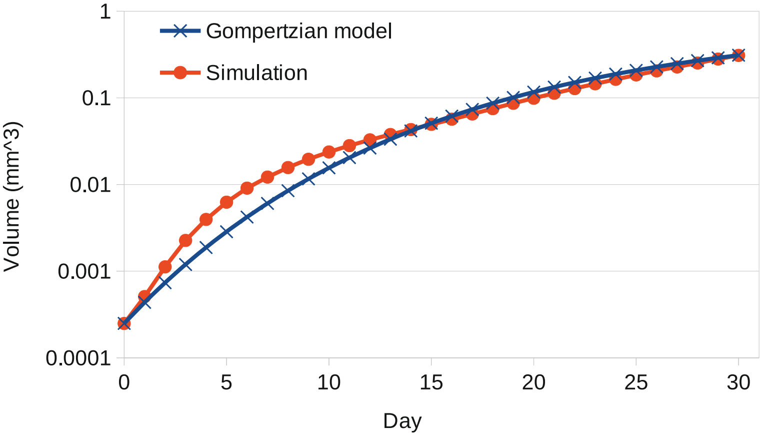

**Fig. 3** The daily evolution of the volume of a tumour simulated by EvoNano PhysiCell compared to the volume determined by the Gompertzian model

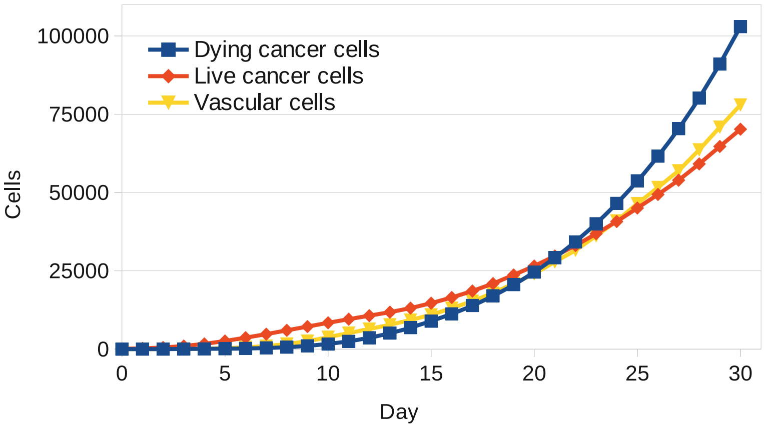

**Fig. 4** The daily evolution of the total number of live cancer cells, dying cancer cells and vascular cells in the simulated tumour

is then expanded according to nearby concentrations of the angiogen produced by cancer cells. It can be seen that the total number of vascular cells closely follows the total number of live cancer cells. On the other side, the total number of dying cancer cells is close to 0 up to day 10. On day 10, the necrotic core of the tumour emerges and the number of dying cancer cells grows at a faster rate than live cancer cells and vascular cancer cells.

Simulated cancer cells can enter either the necrotic or the apoptotic death cycle. Figure 5 shows the evolution of the percentages of necrotic and apoptotic cancer cells. It can be observed that there is a phase transition that corresponds to the emergence of the necrotic core happening at day 10: in the first days cancer cells mostly die by apoptosis and later, due to low oxygen conditions, they die dominantly by necrosis.

As already mentioned, three different types of cancer cells are currently supported by the EvoNano PhysiCell simulator: DCC, CSC, and CAF cancer cells. The percentages of those three types of cancer cells considering the whole tumour (both live and dying cancer cells) are shown in Fig. 6. It can be seen that CSC cells constitute less than 6% of the tumour. More specifically, their percentage slowly grows from 2.94% on day 1 to 5.65% on day 30. On the other hand, DCC and CAF cells have a more complex dynamic with a phase transition. It can be seen that the percentage of CAF cells increases in the first 12 days and then starts to decline. The dynamic of DCC cells is exactly the opposite: the percentage of DCC cells first decline and after day 12 this percentage starts to grow. This phase transition can be also explained by the emergence of the necrotic core. After the emergence of the necrotic core, the number of dying cancer cells, which are mostly DCC cells, grows at a faster rate than the number of vascular cells. Consequently, DCC cells grow faster than CAF cells whose growth rate is always lower than the growth rate of vascular cells.

One of the important aspects of cancer simulation with the previously described setup is the count and origin of CSCs. These cells do not die in apoptosis, and they

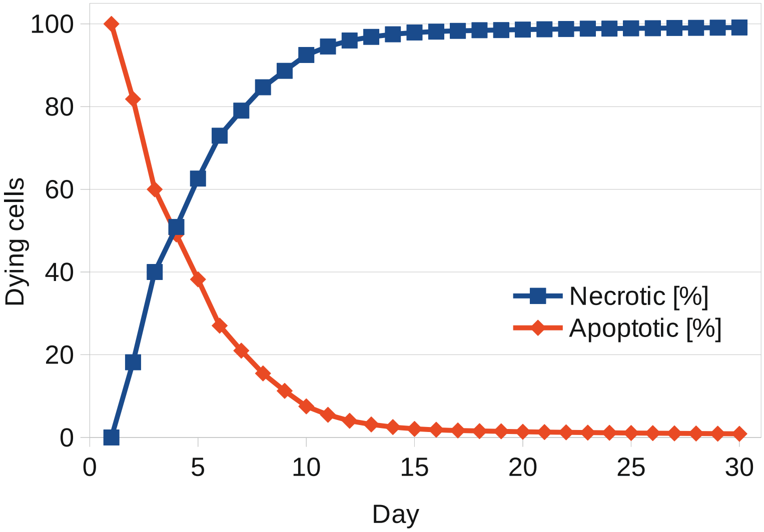

**Fig. 5** The percentage of necrotic and apoptotic cells in the total number of dying cancer cells

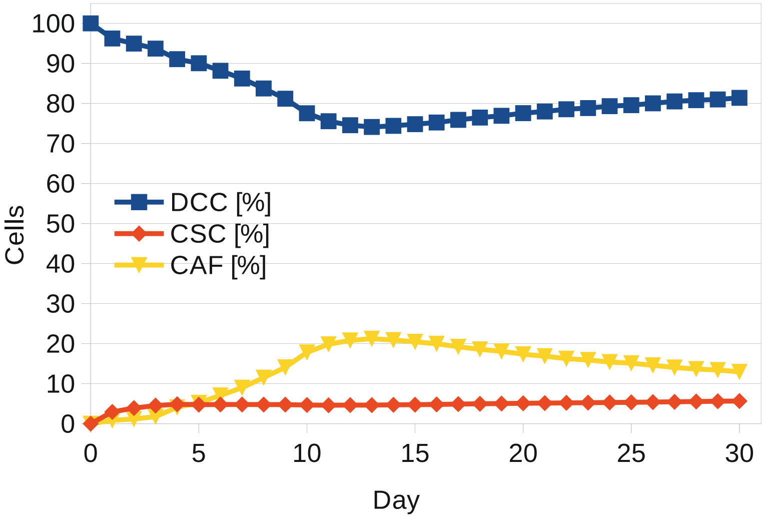

**Fig. 6** The percentage of DCC, CSC and CAF cancer cells in the simulated tumour

are more resistant to low oxygen levels. Therefore, they have an important influence on cancer growth. Furthermore, the emergence of these cells is three-folded:

- some CSCs appear from DCCs in the DCC->CSC conversion process under the influence of high levels of cytokine

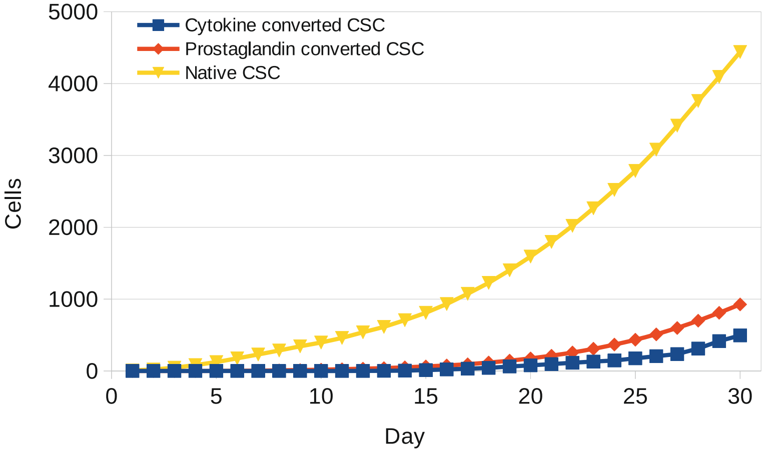

**Fig. 7** The daily count of CSCs and their origin

- some CSCs appear from DCCs in the DCC->CSC conversion process under the influence of high levels of prostaglandine
- all other CSCs are created as a result of DCCs or other CSCs division.

The number of created CSCs according to their origin is shown in Fig. 7. The distribution of prostaglandin converted and cytokine converted CSCs is not empirically known (from clinical research) but it is assumed that the number of converted CSCs is significantly smaller. The number of CSCs and their origin is controlled through numerous parameters as already described. The situation shown in Fig. 7 is a result of many experiments and parameter tuning, all for the purpose of obtaining Gompertzian growth.

In order to demonstrate the influence of all introduced concepts, we performed several simulations with different setups. We compared the simulations where all parameter values are the same (as described earlier) and where:

- prostaglandin and CAF/cytokine mechanisms are turned off
- only prostaglandin mechanism is turned off
- only CAF/cytokine mechanism is turned off
- all mechanisms are turned on.

The daily volumes of tumours with these setups are given in Fig. 8. It is evident that the simulation without prostaglandin behaves almost exactly as with all mechanisms turned on. Also, the simulation without cytokine/CAF behaves similarly as without both mechanisms. However, as seen in Fig. 7 the amount of prostaglandin conversions is higher than cytokine conversions. That may imply that prostaglandin will have a higher influence on cancer growth. However, this is not the case (as seen in Fig. 8) because cytokine/CAF mechanisms also influence the propagation of CAF cells, and

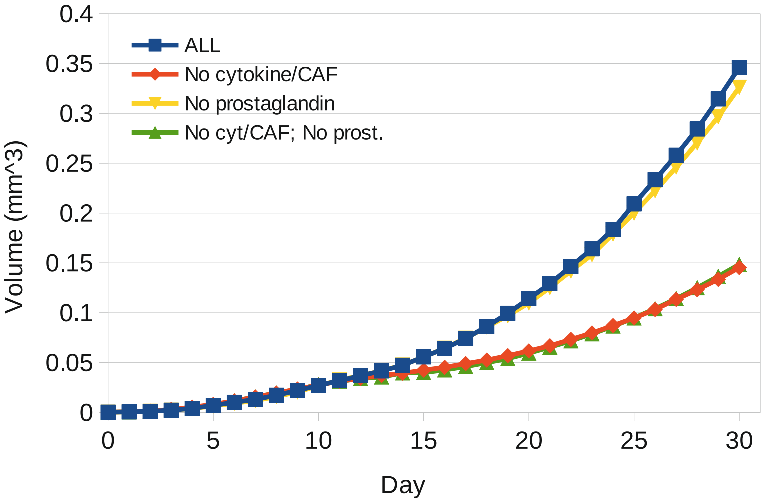

**Fig. 8** The daily evolution of the volume of a tumour simulated with different simulation setups

the amount of them is not negligible (see Fig. 6). All in all, the simulation with all mechanisms nicely follows Gompertzian growth with all specified parameters. If any of the mechanisms are turned off, the re-tuning of parameters is inevitable.

## 4 Conclusions

By intensive experiments, we simulate nanoparticle-based cancer therapies and the existence of NPs with an attached particular drug. That drug can affect cancer in different ways but at the moment we support only drugs that affect cancer cell death.

As we mentioned, for the experimental part we used the EvoNano cancer simulator. It is implemented in C++ relying on the PhysiCell library. The simulator realizes various types of cancer cells (differentiated cancer cells, cancer stem cells, and CAFs), healthy cells, and vascular cells as building blocks for the vascular network. Various substrates are supported (oxygen, angiogen, cytokine, prostaglandin) including also special nano-particle substrates simulating drug diffusion. The main technical feature of the simulator is dynamic space expansion that improves runtime efficiency and enables simulations of larger tumours compared to regular PhysiCell-based simulators without this feature.

The results of conducted experiments show that the large space of simulator parameters can be calibrated to be in line with the Gompertzian tumour growth model. In our simulations, we have observed the emergence of tumour necrotic cores and related phase transitions. We also empirically examined the dynamics of cancer cell conversions based on cytokine and prostaglandin, showing that re-tuning of simulator parameters is required if any of cancer cell conversion mechanisms are turned off or changed. In short, our results demonstrate high sensitivity of tumour development on the interplay of microenvironment signalling factors. Since this aspect of tumour physiology is often neglected in cancer modeling, we believe that our results will help in highlighting the importance of cell-communication signalling in analyzing tumour growth and treatment response.

## References

1. Bakhoum, S.F., Compton, D.A.: Chromosomal instability and cancer: a complex relationship with therapeutic potential. J. Clin. Investig. **122**, 1138–1143 (2012). https://doi.org/10.1172/JCI59954
2. Calon, A., et al.: Stromal gene expression defines poor-prognosis subtypes in colorectal cancer. Nat. Genet. **47**, 320–329 (2015)
3. Cavallari, L.H., Mason, D.L.: Cardiovascular pharmacogenomics-implications for patients with CKD. Adv. Chronic Kidney Dis. **23**(2), 82–90 (2016)
4. Chen, Y., Song, Y., Du, W., et al.: Tumor-associated macrophages: an accomplice in solid tumor progression. J. Biomed. Sci. **26**, 78 (2019). https://doi.org/10.1186/s12929-019-0568-z
5. Cisyk, A.L., Penner-Goeke, S., Lichtensztejn, Z., Nugent, Z., Wightman, R.H., Singh, H., McManus, K.J.: Characterizing the prevalence of chromosome instability in interval colorectal cancer. Neoplasia **17**, 306–316 (2015). https://doi.org/10.1016/j.neo.2015.02.001
6. Claeys, A., Vialatte, J.S.: Advances in genetics: towards a Precision Medicine? Technological, social and ethical scientific issues of personalised medicine [Les progrès de la génétique: vers une médecine de précision? Les enjeux scientifiques, technologiques, sociaux et éthiques de la médecine personnalisée] (2014)
7. Deisboeck, T.S., Wang, Z., Macklin, P., Cristini, V.: Multiscale cancer modeling. Annu. Rev. Biomed. Eng. **15**(13), 127–155 (2011)
8. Dewhirst, M.W., Secomb, T.W.: Transport of drugs from blood vessels to tumour tissue. Nat. Rev. Cancer **17**(12), 738–750 (2017). https://doi.org/10.1038/nrc.2017.93
9. Dreesen, O., Brivanlou, A.H.: Signaling pathways in cancer and embryonic stem cells. Stem Cell Rev. **3**(1), 7–17 (2007). https://doi.org/10.1007/s12015-007-0004-8. PMID: 17873377
10. Feng, K., Leary, R.H.: Toward personalized medicine with physiologically based pharmacokinetic modeling. Int. J. Pharmacokinet. **2**(1), 1–4 (2017)
11. Ghaffarizadeh, A., Friedman, S.H., Macklin, P.: BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations. Bioinformatics **32**(8), 1256–1258 (2016). https://doi.org/10.1093/bioinformatics/btv730. Epub (2015), PMID: 26656933; PMCID: PMC4824128
12. Ghaffarizadeh, A., Heiland, R., Friedman, S.H., Mumenthaler, S.M., Macklin, P.: PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems. PLoS Comput. Biol. **14**(2), e1005991 (2018). https://doi.org/10.1371/journal.pcbi.1005991
13. Guinney, J., et al.: The consensus molecular subtypes of colorectal cancer. Nat. Med. **21**, 1350–1356 (2015)
14. Gujam, F.J., Going, J.J., Edwards, J., et al.: The role of lymphatic and blood vessel invasion in predicting survival and methods of detection in patients with primary operable breast cancer. Crit. Rev. Oncol. Hematol. **89**, 231–241 (2014)
15. https://medlineplus.gov/genetics/understanding/precisionmedicine/definition/
16. Jordan, V.C.: Tamoxifen: catalyst for the change to targeted therapy. Eur. J. Cancer **44**(1), 30–38 (2008)
17. Kahn, et al.: J. Clin. Invest. **131**(2), e136655 (2021). https://doi.org/10.1172/JCI136655

18. Kim, B.J., Hannanta-anan, P., Chau, M., Kim, Y.S., Swartz, M.A., et al.: Cooperative roles of SDF-1a and EGF gradients on tumor cell migration revealed by a robust 3D microfluidic model. PLoS ONE **8**(7), e68422 (2013). https://doi.org/10.1371/journal.pone.0068422
19. Kobuchi, S., Shimizu, R., Ito, Y.: Semi-mechanism-based pharmacokinetic-toxicodynamic model of oxaliplatin-induced acute and chronic neuropathy. Pharmaceutics **12**(2), 125 (2020)
20. Lahiri, C., Pawar, S., Mishra, R.: Precision medicine and future of cancer treatment. Precis. Cancer Med. **2**, 33 (2019). AME Publishing
21. Lengauer, C., Kinzler, K.W., Vogelstein, B.: Genetic instability in colorectal cancers. Nature **386**, 623–627 (1997). https://doi.org/10.1038/386623a0
22. Liu, T., Han, C., Wang, S., et al.: Cancer-associated fibroblasts: an emerging target of anti-cancer immunotherapy. J. Hematol. Oncol. **12**, 86 (2019). https://doi.org/10.1186/s13045-019-0770-1
23. Macklin, P., Frieboes, H.B., Sparks, J.L., Ghaffarizadeh, A., Friedman, S.H., Juarez, E.F., Jonckheere, E., Mumenthaler, S.M.: Progress towards computational 3-D multicellular systems biology. Adv. Exp. Med. Biol. **936**, 225–246 (2016). https://doi.org/10.1007/978-3-319-42023-3-12
24. Mauri, G., Bonazzina, E., Amatu, A., Tosi, F., Bencardino, K., Gori, V., Massihnia, D., Cipani, T., Spina, F., Ghezzi, S., Siena, S., Sartore-Bianchi, A.: The evolutionary landscape of treatment for BRAFV600E mutant metastatic colorectal cancer. Cancers (Basel) **13**(1), 137 (2021). https://doi.org/10.3390/cancers13010137. PMID: 33406649; PMCID: PMC7795863
25. Meads, M.B., Gatenby, R.A., Dalton, W.S.: Environment-mediated drug resistance: a major contributor to minimal residual disease. Nat. Rev. Cancer **9**(9), 665–674 (2009). https://doi.org/10.1038/nrc2714
26. Pattabiraman, D.R., Weinberg, R.A.: Tackling the cancer stem cells - what challenges do they pose? Nat. Rev. Drug Discov. **13**(7), 497–512 (2014). https://doi.org/10.1038/nrd4253. PMID: 24981363; PMCID: PMC4234172
27. Ping, Q., Yan, R., Cheng, X., et al.: Cancer-associated fibroblasts: overview, progress, challenges, and directions. Cancer Gene Ther. **28**, 984–999 (2021). https://doi.org/10.1038/s41417-021-00318-4
28. Sahai, E., Astsaturov, I., Cukierman, E., et al.: A framework for advancing our understanding of cancer-associated fibroblasts. Nat. Rev. Cancer **20**, 174–186 (2020). https://doi.org/10.1038/s41568-019-0238-1
29. Simpson, M.J., Towne, C., McElwain, D.L.S., Upton, Z.: Migration of breast cancer cells: understanding the roles of volume exclusion and cell-to-cell adhesion. Phys. Rev. E. **82**, 041901 (2010). https://doi.org/10.1103/PhysRevE.82.041901
30. Sleeman, J.P., Thiele, W.: Tumor metastasis and the lymphatic vasculature. Int. J. Cancer **125**, 2747–2756 (2009)
31. Speer, J.F., Petrosky, V.E., Retsky, M.W., Wardwell, R.H.: A stochastic numerical model of breast cancer growth that simulates clinical data. Cancer Res. **44**(9), 4124–4130 (1984)
32. Stéphanou, A., Volpert, V.: Hybrid modelling in biology: a classification review. Math. Model. Nat. Phenomena **11**(1), 37–48 (2016)
33. Stillman, N.R., Kovacevic, M., Balaz, I., Hauert, S.: In silico modelling of cancer nanomedicine, across scales and transport barriers. NPJ Comput. Mater. **6**, 92 (2020). https://doi.org/10.1038/s41524-020-00366-8
34. Stillman, N.R., Balaz, I., Tsompanas, M.A., et al.: Evolutionary computational platform for the automatic discovery of nanocarriers for cancer treatment. NPJ Comput. Mater. **7**, 150 (2021). https://doi.org/10.1038/s41524-021-00614-5
35. Thompson, L.L., Jeusset, L.M., Lepage, C.C., McManus, K.J.: Evolving therapeutic strategies to exploit chromosome instability in cancer. Cancers (Basel) **9**(11), 151 (2017). https://doi.org/10.3390/cancers9110151. PMID: 29104272; PMCID: PMC5704169
36. Vaghi, C., Rodallec, A., Fanciullino, R., Ciccolini, J., Mochel, J.P., et al.: Population modeling of tumor growth curves and the reduced Gompertz model improve prediction of the age of experimental tumors. PLoS Comput. Biol. **16**(2), e1007178 (2020). https://doi.org/10.1371/journal.pcbi.1007178

37. Vaupel, P., Mayer, A.: A hypoxia in cancer significance and impact on clinical outcome. Cancer Metastasis Rev. **26**, 225–239 (2007)
38. Venne, J., Busshoff, U., Poschadel, S., Menschel, R., Evangelatos, N., Vysyaraju, K., Brand, A.: International consortium for personalized medicine: an international survey about the future of personalized medicine. Pers. Med. **17**(2), 89–100 (2020)
39. Vilanova, G., Colominas, I., Gomez, H.: Capillary networks in tumor angiogenesis: from discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis. Int. J. Numer. Method Biomed. Eng. **29**(10), 1015–1037 (2013). https://doi.org/10.1002/cnm.2552. Epub (2013). PMID: 23653256
40. Wei, Y., Zhao, Q., Gao, Z., Lao, X.M., Lin, W.M., Chen, D.P., et al.: The local immune landscape determines tumor PD-L1 heterogeneity and sensitivity to therapy. J. Clin. Invest. **129**(8), 3347–3360 (2019). https://doi.org/10.1172/JCI127726
41. Wilson, W.R., Hay, M.P.: Targeting hypoxia in cancer therapy. Nat. Rev. Cancer **11**, 393–410 (2011)