



# Cellular-automaton model for tumor growth dynamics: Virtualization of different scenarios

Carlos A. Valentim<sup>1</sup>, José A. Rabi<sup>1</sup>, Sergio A. David \*<sup>1</sup>

*Department of Biosystems Engineering, University of São Paulo, Pirassununga, Brazil*



## ARTICLE INFO

### Keywords:

Mathematical oncology  
Cancer dynamics  
In-silico experimentation  
Computational modeling

## ABSTRACT

Mathematical Oncology has emerged as a research field that applies either continuous or discrete models to mathematically describe cancer-related phenomena. Such methods are usually expressed in terms of differential equations, however tumor composition involves specific cellular structure and can demonstrate probabilistic nature, often requiring tailor-made approaches. In this context, cell-based models allow monitoring independent single parameters, which might vary in both time and space. By relying on extant tumor growth models in the literature, this study introduces cellular-automata simulation strategies that admit heterogeneous cell population while capturing both single-cell and cluster-cell behaviors. In this agent-based computational model, tumor cells are limited to follow four possible courses of action, namely: proliferation, migration, apoptosis or quiescence. Despite the apparent simplicity of those actions, the model can represent different complex tumor features depending on parameter settings. This study virtualized five different scenarios, showcasing model capabilities of representing tumor dynamics including alternate dormancy periods, cell death instability and cluster formation. Implementation techniques are also explored together with prospective model expansion towards deterministic features. The proposed stochastic cellular automaton model is able to effectively simulate different scenarios regarding tumor growth effectively, figuring as an interesting tool for *in silico* modeling, with promising capabilities of expansion to support research in mathematical oncology, thus improving diagnosis tools and/or personalized treatment.

## 1. Introduction

Cancer is the generic term attributed to a large group of diseases involving cells that develop characteristics allowing them to abnormally reproduce, progressively invading other tissues and unleashing several problems to the host [1]. Also called malignant tumors or neoplasms, it caused almost 10 million deaths only in 2020, indirectly being responsible for an annual cost reaching trillion dollar figures [1,2]. Although cancers have different adaptation and growth mechanisms depending on their origin in the body, they have similar traits and common genetic characteristics, generally arising from a single cancerous cell [3,4].

In the last decades, mathematical concepts have been increasingly applied to oncological phenomena not only to better understand the progression of related diseases but also to develop new methods of diagnosis and treatment, contributing to the emergence of a new research area [5–7]. Mathematical oncology comprehends the development and

application of models to phenomena ranging from neoplastic growth to personalized treatment [8]. As a strategic advantage, mathematical models can test and reproduce several scenarios either unfeasible or impossible *in vitro* experiments, which turns it into an important analysis tool as clinical tests in humans are time and resource consuming [9].

Mathematical models in oncology may be categorized into two large groups: data-driven and physics-based models. Considering the prevailing scenario of elevated difficulty (in terms of resources and feasibility) in obtaining consistent data from oncologic patients, the later category, also called phenomenological or mechanistic approach, has advanced in modeling related phenomena. Examples including ordinary and partial differential equations (ODEs and PDEs, respectively) illustrate the success of those approaches, such as strategies based on ecological models and the underlying of general avascular tumor growth [10–12]. With their relative simplicity, ODE-based approaches enable analytical solutions and have conveniences that motivate their

**Abbreviations:** CA, Cellular automata; CCT, Cell cycle time; At, time step; m, Migration potential; ODE, ordinary differential equations; p, Proliferation potential; PDE, partial differential equations;  $p_{max}$ , Maximum proliferation potential; PA, Probability of apoptosis; Pm, Probability of migration; PP, Probability of proliferation; PS, Probability of generating a stem cell; RTC, Regular tumor cells; STC, Stem tumor cells

\* Corresponding author.

E-mail addresses: [carlos.valentim@usp.br](mailto:carlos.valentim@usp.br) (C.A. Valentim), [jrabi@usp.br](mailto:jrabi@usp.br) (J.A. Rabi), [sergiодavid@usp.br](mailto:sergiодavid@usp.br) (S.A. David).

<sup>1</sup> All authors contributed equally.

<https://doi.org/10.1016/j.compbioimed.2022.106481>

Received 27 October 2022; Received in revised form 8 December 2022; Accepted 25 December 2022

Available online 28 December 2022

0010-4825/© 2022 Elsevier Ltd. This article is made available under the Elsevier license (<http://www.elsevier.com/open-access/userlicense/1.0/>).

use until today [13–15]. On the other hand, PDEs can model tumor growth into surrounding tissue [16]. Some models describe tumors as a fluid or mixture via transport equations [17], while others employ transport phenomena to model metastatic processes and beyond [15, 18].

When a model requires specific cellular structure and probabilistic nature involving cell proliferation, equation-based approaches may not suffice. In that context, Anderson et al. [19] claim that while continuum mathematical models have been successfully employed to describe several portions of matter, these portions in nature are actually particles and cells, thus discrete. In the wake of the impressive progress of biochemistry and biologist concepts on genetics, sub-cellular levels and inner works, computational-enhanced mathematical oncology faces the difficult task of transforming specific portion-sized data into complex information describing emergent higher-level multi-scale cellular phenomena. In recent years, many cell-based models have been proposed to face such challenge [20,21]. Cell-based or discrete models are organized frameworks that allow keeping track of fully independent individual parameters varying spatially and temporally, reflecting the heterogeneity and complex emergence found in cancer phenomena. Computationally, they can rely on different approaches including Monte-Carlo simulations, energy minimization techniques, volume conservation laws, and motion rules [19].

If these models follow a structural or grid organization, they are considered lattice-based models, which are categorized according to the number of cells that each lattice cell can hold [22]. Lattice-gas cellular automata models admit more than one cell per lattice (being suitable for larger systems). On the other hand, if the model admits that a single cell can occupy many spots, it is thus ideal for modeling sub-cellular systems [23]. Finally, if each cell can occupy a single lattice, it is a regular cellular automaton (CA) model [22].

Numerical simulations involving cell-based models are often referred to as in-silico modeling because of their similarity and logical extension of in vitro experimentation [24]. Concerning regular cellular automaton models, relatively simple implementations can go a long way in providing emergent complex behavior. Enderling et al. established only a basic set of rules concerning proliferation and migration rates for each type of tumor cell (regular or stem) in a CA and investigated the virtualization of very different emergent scenarios when changing these rules, including cell clustering and tumor dormancy [25]. Later, Poleszczuk and Enderling improved the model by implementing it with high-performance computational techniques. These two studies arise as the grounding basis of the automaton model herein developed.

In this context, this paper's methodology is divided into three parts: Section 2.1 presents the mathematical background and definitions, 2.2 describes biological constructs and model assumptions, and Section 2.3 reports programming aspects and computational implementation. Next, the virtualization of several case scenarios is conducted and discussed in 3. In 4, we explore potential shortcomings of this cellular automaton and how it is being developed as a grounding framework for a later-to-be-implemented hybrid model.

## 2. Methodology

### 2.1. Mathematical aspects

Since a cellular automaton is composed of a multitude of equally identifiable cells and for every cell a certain set of neighbors is used to calculate a new state, the resulting network structure of neighborhood relations is a further important characteristic of cellular automata. In order to describe the adopted method in this study, we initially revisit the definition and some concepts related to cellular automata.

#### 2.1.1. Basic concepts of cellular automata

In a simplified manner, a cellular automaton can be typically defined as a basic structure considering the quadruple  $(C, n, S, f)$  as follows [26]:

- $C$  is a set of cells, not required to be finite.
- $n : C \times C \rightarrow \{0, 1\}$  is a neighborhood function that can be seen as a relationship (usually reflexive and symmetric) between cells. This function shows which pairs of cells are neighbors, that is, the geometry of the cell organization. Furthermore,  $n$  must satisfy the neighborhood size independence condition  $|N(c_0)| = |c \in C : n(c_0, c) = 1| = N$ , which is constant for every  $c_0 \in C$ , i.e., the size of the neighborhood is the same for all cells.
- $S$  is a set of states. Each cell will have an associated state, in each moment.
- $f : S^{|N|} \rightarrow S$  is a transition function. The transition function is a core of the CA dynamics and is commonly expressed with rules that define the state of the cell in the next time moment from the state of the cell neighbors. The set of cells  $C$  with the neighborhood function  $n$  defines the structure of the cell space. The simplest CA model can have binary cells (e.g., two states: tumor or healthy) [27]. More commonly, multiple states are represented by a set of integer values  $\{0, 1, 2, \dots\}$ , each having an appropriate physical or biological interpretation (see Sections 2.2.1 and 2.2.2 for our model's neighborhood geometry and cell states).

#### 2.1.2. Stochastic cellular automata

In some situations, such as when evolution operators are stochastically approximated, the resulting states have stochastic character. A mathematical approach to random or stochastic states can happen by representing states as random variables. Stochastic cellular automata can be defined in the following way, which settles them between Bayesian networks [28] and multi-parameter stochastic processes [29]:

- $M$  is a finite set of cells.
- $\mathcal{N}$  is a neighborhood mapping  $\mathcal{N} : M \rightarrow M^k$ .
- $\mathbb{S}$  is a separable measurable Hausdorff space  $(\mathbb{S}, \mathcal{B})$  with random variables  $S_{t,m}$  from a probability space  $(\Omega, \mathcal{U}, \mathcal{P})$  to  $(\mathbb{S}, \mathcal{B})$ .
- $\mathcal{K}$  is a Markov-kernel:  $\mathbb{S}^k \times \mathcal{B} \rightarrow [0, 1]$ , such that  $\mathcal{P}(S_{t+1,m} \in \mathcal{B} | S_t, \mathcal{N}_{(m)} = s) = \mathcal{K}(s, \mathcal{B})$ .

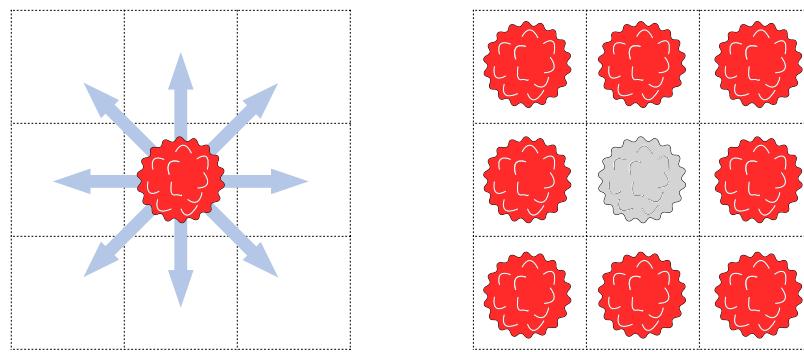
An equivalent definition of stochastic cellular automata can be formulated through filtrations of  $\sigma$ -algebras based on the graphical structure of the neighborhood relations [30]. Some direct conclusions from stochastic cellular automata include that the global stochastic process, which describes the simultaneous transition of the states of all cells, is a Markov process itself. Furthermore a Chapman–Kolmogorov like equation can be formulated for stochastic cellular automata.

In addition, stochastic processes are intrinsically connected to cellular automata modeled through multi-agent systems (also called agent-based modeling) [31]. In those models, micro-scale autonomous agents (cells) follow simple and programmable actions and cascade into different emerging processes, thus creating complex macro-scale systems (tumors) with varied behavior [32]. In this paper, we explore further our model's behaviors and stochastic characteristics in Sections 2.2.3 and 2.2.4.

### 2.2. Biological constructs and considerations

The agent-based cellular automaton model developed in this study was based on a mix of characteristics in [25,33] and follows the mathematical groundings in Section 2. It is a stochastic framework in which a single cell (agent) originates a tumor that can have several different features depending on model configurations and its parameters.

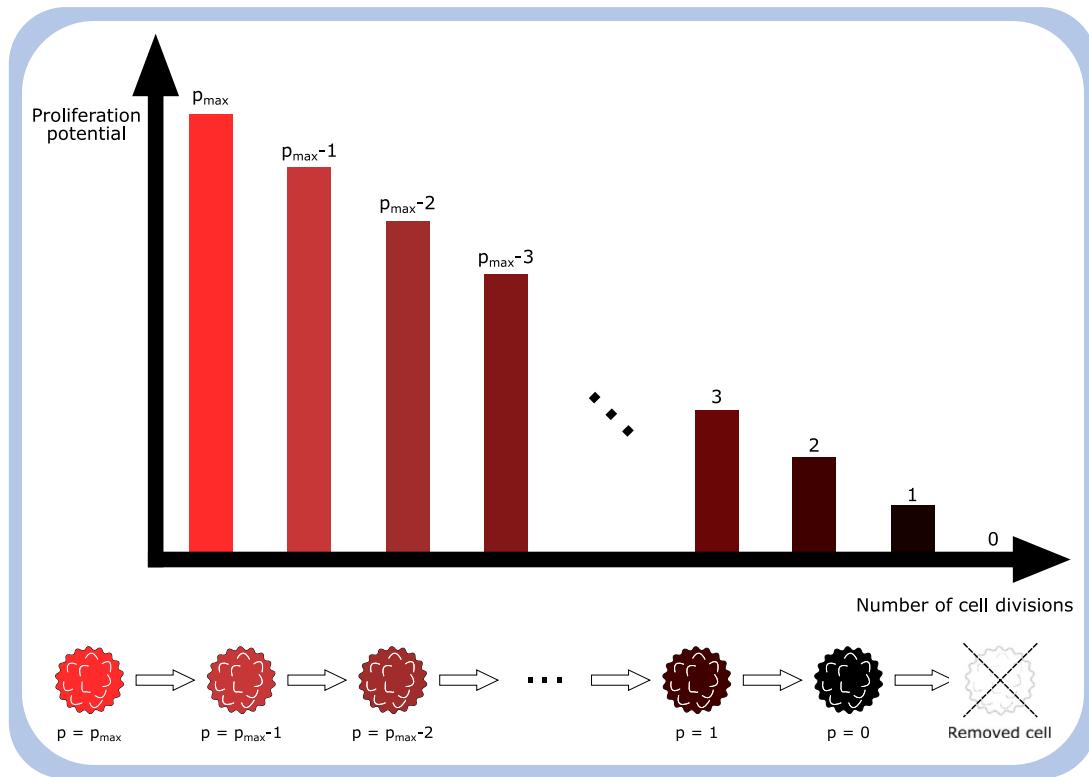
The model is discrete both in time and space. Regarding the latter, Section 2.2.1 describes lattice geometry and model neighborhood.



**(a)** If there are available spaces, the cell can move or proliferate to any of the adjacent positions.

**(b)** If there are no available spaces, the cell becomes quiescent until an adjacent cell either moves or dies.

**Fig. 1.** Representation of the computational lattice, where each space of  $100 \mu\text{m}^2$  can hold up to one cell (2D Moore).



**Fig. 2.** A graphical representation regarding the proliferation potential of RTCs.

Section 2.2.2 presents the possible states for each agent or cell in the model. The simulation advances in steps of  $\Delta t$  days, where every cell in the tumor will obligatorily present one of the behaviors described in Section 2.2.3, which also describes the main parameters of the model. Finally, as we deal with a stochastic model, the automaton results are discussed in Section 2.2.4 in terms of averages of simulation batches.

### 2.2.1. Lattice and neighborhood geometry

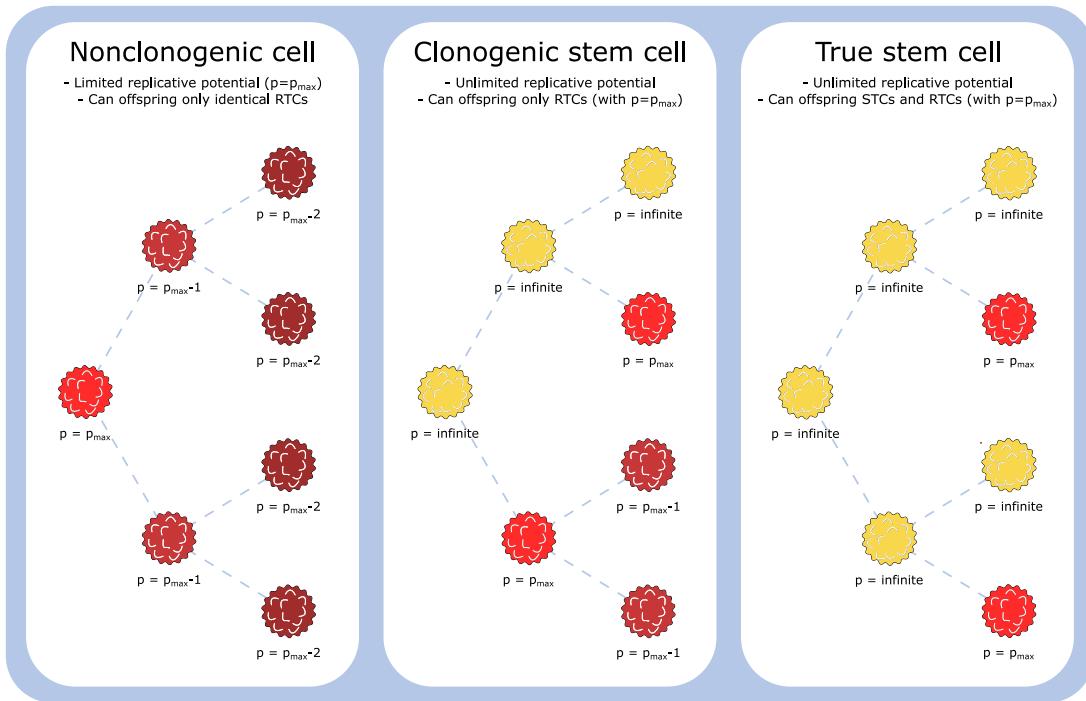
In this work, a 2D lattice is considered in which each element can hold up to one cell at a time. The length of each lattice element is  $10 \mu\text{m}$ , which is comparable to the size of a regular cell. A 2D Moore neighborhood is considered, implying that a tumor cell can move to any adjacent free place during a computational time step, as shown in Fig. 1(a). When there are no free lattice positions (i.e. empty space) surrounding the cell, it cannot move nor proliferate thus staying quiescent, as depicted in Fig. 1(b).

### 2.2.2. Cell states

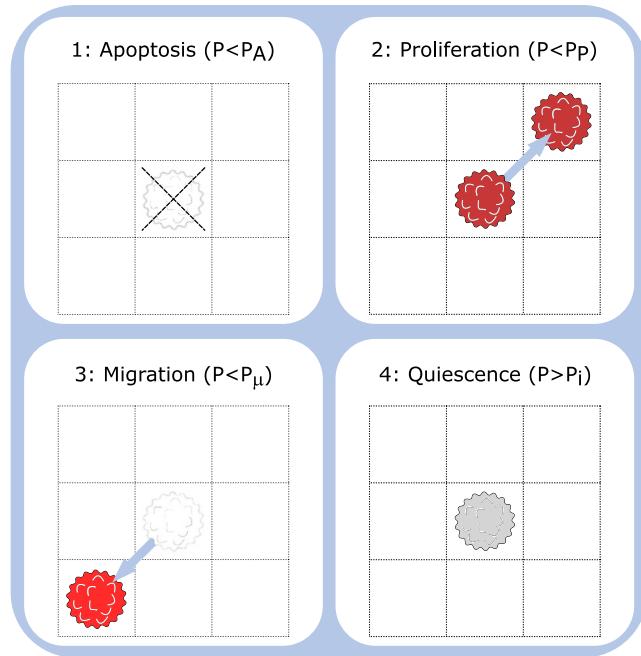
The model comprehends a heterogeneous population of tumor cells: they can either be a regular tumor cell (RTC) or a stem tumor cell (STC). Both healthy and dead cells are not accounted in this approach.

There are some crucial differences between those two types of cells. The vast majority of tumor cells will usually be RTCs, which have a maximum proliferation potential  $p_{max}$ , thus generating a finite number of offspring cells. They can only give birth to RTCs and eventually may die (either by programmed death, apoptosis, or when they reach their maximum replication potential). In those cases, the tumor cell is removed from the lattice. Also called nonclonogenic cells, RTCs are visually represented in this model in colors ranging from red (maximum proliferation potential) to black (exhausted cell). By relying on this graphical pattern, this mechanism is sketched in Fig. 2.

On the other hand, STCs completely lack internal regulatory mechanisms regarding cell death, thus being immortal. They have an infinite



**Fig. 3.** Cell populations included in the model. Outcomes of an evolving tumor will heavily depend on its original progenitor cell and if it is either nonclonogenic, clonogenic, or stem.



**Fig. 4.** During each time iteration, a tumor cell obligatorily triggers one of four events: apoptosis, proliferation, migration, or quiescence. These events are temporally exclusive (i.e., a cell will only perform one of them during a single time step).

proliferation capacity and will continue replicating independently of how many divisions they have endured. These cells can be categorized in clonogenic or true stem cells. The first group can only give birth to regular (i.e. mortal) cells with limited proliferation potential. The latter can perform an asymmetric division in which the daughter is an RTC, but might also generate an identical true stem cell. The mechanisms regarding the differences between aforementioned cells are depicted in

0	0	0	0	9	0
0	12	11	10	9	0
10	11	12	12	12	0
0	10	11	11	10	9
0	9	11	11	9	8
8	0	0	0	0	0

**Fig. 5.** An example of a coded lattice. Colors represent the convention adopted in Figs. 2 and 3.

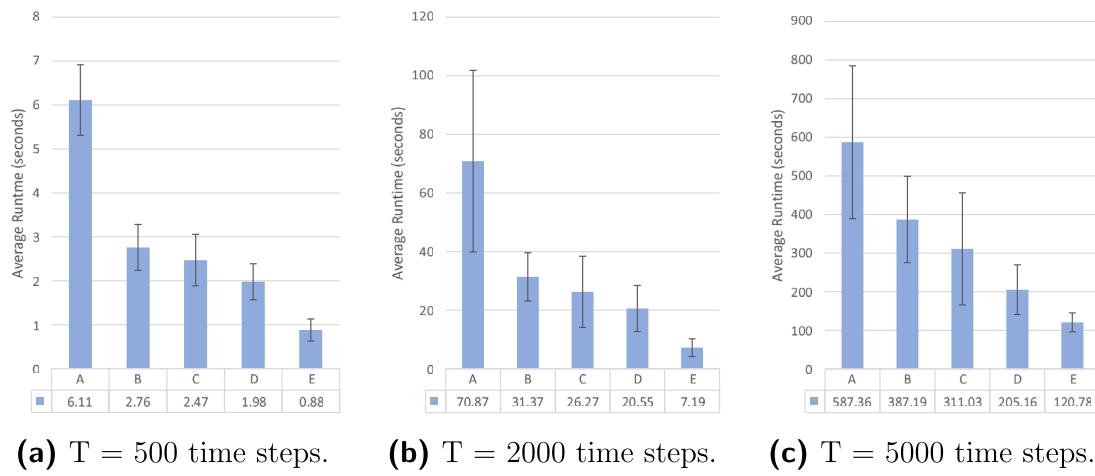
Fig. 3, in which STCs are represented in yellow and the probability of a symmetrical division from a true stem cell is  $P_S$ .

#### 2.2.3. Model mechanics and cell behavior

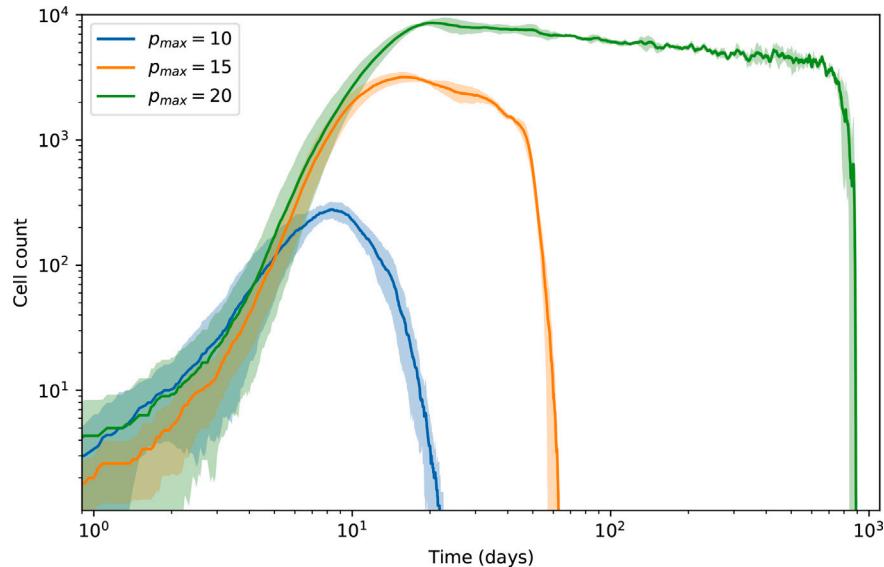
Every cell in the model will obligatorily present one of the following behaviors during each time step. At first, every cell has a chance  $P_A$  of undergoing apoptosis. This is usually a very low rate, since tumor cells can generally activate a number of processes to avoid cell death [4]. Naturally,  $P_A = 0$  for every STC since they are allegedly immortal.

If the cell endures, it has a proliferation probability  $P_P = CCT \Delta t / 24$ , where  $CCT$  is cell cycle time. This probability suggests that a cell will replicate more or less according to its natural cycle. If the original cell is a true STC, it will have a low, but essential, chance  $P_S$  of generating another identical STC.

If the cell neither dies nor replicates, it has a chance  $P_\mu = \mu \Delta t$  of migrating according to its displacement capacity  $\mu$ . In case the cell



**Fig. 6.** Speed performance tests for representative code versions and time steps ( $N = 10$  replicates).



**Fig. 7.** First scenario: Population dynamics of a tumor originated by a nonclonogenic cell with different proliferation potentials (average of 5 simulations with  $p_{max} = 10$ ,  $p_{max} = 15$ , and  $p_{max} = 20$ ).

Table 1	
Description of each main code version evaluated in the speed performance test.	
Code version	Main characteristics (comparatively)
'A'	Sparse matrix to represent tumor cells.
'B'	Use of large lattice with no expanding domain.
'C'	Coordinate system with an expanding domain; Use of the Random library.
'D'	Similar to 'C', but using the Numpy-Random library.
'E'	Similar to 'D', but using <i>shuffle</i> instead of <i>choice</i> .

fails upon all these probabilities within the same time step, it stays quiescent or dormant. This state of dormancy is also achieved if the cell is completely surrounded and lacks free space for either proliferating or moving.

Fig. 4 sketches the four possible mechanisms for each cell during any time step, along with their probabilities. By presenting the  $\Delta t$  term, every probability is adjusted to the respective time scale used in the simulation. One exception is the chance for symmetrical division by stem cells because the time scale has already been considered when calculating its proliferation chance.

The flowchart in Fig. 17 of the Appendix summarizes the logic developed for this model and implemented in the algorithm discussed

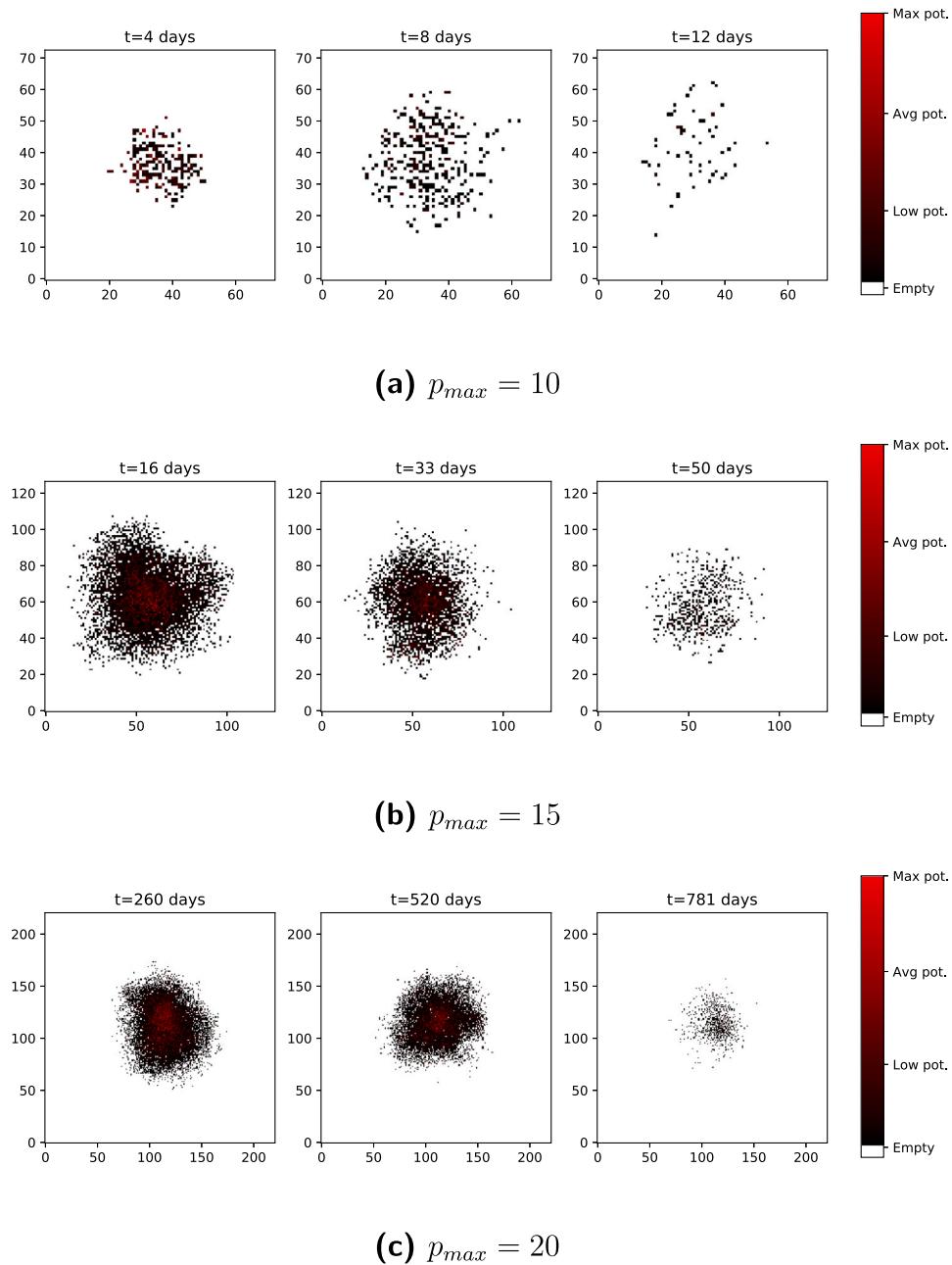
**Table 2**  
Common parameters and probabilities for all studied scenarios.

Parameter	Value	Unit
Time step ( $\Delta t$ )	1/24	day
Cell cycle time (CCT)	24	hours
Proliferation probability	4.17%	(per time step)
Lattice cell width	10	$\mu\text{m}$

**Table 3**  
Average results for the first scenario. Tumor growth from a cell without clonogenic potential.

Proliferation potential	Max. RTC count	Final RTC count
$p_{max} = 10$	$280 \pm 43$	0
$p_{max} = 15$	$3191 \pm 306$	0
$p_{max} = 20$	$8657 \pm 574$	0

in Section 2.3. While this figure provides a general overview of how each cell can act during a simulation time step, some specific details are addressed in Section 2.3, such as the approach regarding domain/lattice extension.



**Fig. 8.** First scenario: Spatial evolution of a representative tumor originated by a nonclonogenic cell with different proliferation potentials.

**Table 4**

Average results for the second scenario. Tumor growth from a clonogenic cell.

Proliferation potential	Max. RTC count	Final RTC count
$p_{max} = 10$	$2049 \pm 241$	$1699 \pm 522$
$p_{max} = 15$	$6676 \pm 1070$	$4442 \pm 872$
$p_{max} = 20$	$13,012 \pm 1525$	$8912 \pm 1643$

#### 2.2.4. The stochastic process

Cell behavior described in Section 2.2.3 is modeled through probabilities so that the cellular automaton herein considered is a stochastic process. Therefore, a batch of simulations must be carried out for any investigation (study) using the model.

Some authors argue that for general trends and ideas a low number of simulations – such as five – is enough [25] whereas for more

delicate testing (or performance evaluation) a number from 50 to 100 should suffice [33]. This necessity for replications casts a light on the importance of developing a simulation code with as low computational effort (in terms of CPU time and memory) as possible, which is a programming goal discussed in Section 2.3. An alternative to decrease the number of simulations is to implement some deterministic characteristics to the model, thus transforming it into a hybrid approach. For more information on the latter, please refer to Section 4.

In order to express overall results from the batch of simulations, one can use averages and standard deviations to describe cells population dynamics, with RTC and STC populations being described separately. On the other hand, tumor development in space cannot be mathematically described using averages. Therefore, in order to spatially represent a virtualized tumor one must elect the most representative replicate

from the simulation batches (i.e. one whose average diffusivity, RTC and STC counts are closer to average values).

### 2.3. Computational implementation

The cellular automaton herein described was coded in Python, chosen as programming language due to its versatility and accessible learning curve. Even though other languages such as Fortran, C and C++ could be faster to tackle procedures such as array swapping and random events, Python has a number of libraries that greatly improve both code accessibility and readability without jeopardizing its speed.

In [33], some high-performance techniques are presented towards the development of a cellular automaton for tumor growth. According to their tests regarding speed and memory, an improved code containing their suggestions would greatly improve performance in comparison to a “naive” code. Overall, the authors defend the use of specific libraries of the target language instead of trying to build customized code (e.g. use of C++ Standard Template Library, STL, to tackle random events). Many of their suggestions have been applied in the present automaton, with a few occasional caveats and adaptations since the programming languages are different (e.g., *numpy* was used instead of STL). Those suggestions are briefly discussed in this section along with some details regarding code implementation. The final code used for simulations in Section 3 has been shared on a public GitHub repository [34].

#### 2.3.1. The coded lattice

A squared matrix represents the 2D-lattice in the model described in Section 2.2.1. Therefore, each tumor starts as an empty matrix with a single cancerous cell at its center. The *numpy* library then provides the data structure for the matrix, namely a 2D array with only integer values.

In the tumor matrix, a zero represents an empty space while a nonzero element is a tumor cell. Although one could simply use a matrix of zeros and ones, Poleszczuk and Enderling [33] suggest using a so-called coded lattice, in which each element value in the tumor matrix not only indicates the presence of a cell but also provides some additional information without relying on extra memory. In this case, each element value represents the proliferation potential of the corresponding cell positioned at that site. Following this approach, after each successful replication of a nonclonogenic cell, its current proliferation potential is updated by simply subtracting one. Besides, by looking at each element value in the matrix, one can quickly grasp how many divisions that corresponding cell can still endure.

In the coded lattice, STCs are characterized by the smallest integer value above the maximum proliferation potential for that tumor setting. For instance, if a tumor with  $p_{max} = 10$  is simulated, new RTCs will be represented by the value 11 and will decrease this value by one for every successful division (generating a cell with an identical value). Then, in this simulation STCs would be represented by the value 12. In the algorithm, one would have to establish that if a element of the tumor matrix is higher than 11 then it is a stem cell and normal rules do not apply to it (such as apoptosis). Fig. 5 illustrates that process.

#### 2.3.2. Loops and array operations

When using a numerical matrix as lattice, one needs to sweep every row and column in order to access every element. One usual way to program such a routine is using nested loops, which is a very slow computational task in Python. An alternative is to use array operations, which perform, for instance, algebraic calculations on entire arrays and matrices.

Nevertheless, these operations do not fit very well with the systematic time-incremental procedure that a cellular automaton generally follows, particularly because the code will need to rely on random events to determine how a cell will act. Therefore, as a middle-ground solution, one can opt for using the matrix-specific *numpy* command

*matrix.nonzero*, which will quickly scan the matrix and return row and column indexes related to every tumor cell (i.e., all elements greater than zero).

Next, a single array pertaining the coordinates from all tumor cells in the tissue will be created. This will enable a single iterative loop to check the content in the main tumor matrix according to the coordinates in the array. This approach is generally faster than relying on nested loops to find tumor cells in this matrix.

#### 2.3.3. Random ordering and neighbor selection

There are multiple ways of having access to random numbers using Python. One usual way is through the package *random* and drawing a pseudo-random number between 0 and 1 each time the chance of a cell fulfill some action is tested. Another approach is to use the *numpy* library to draw a full array of pseudo-random numbers at once. The latter can be faster than the former when the length of the referred array is sufficiently large.

In our model, one always knows the number of tumor cells at the beginning of each time iteration. Therefore, one can use *numpy.random* to generate an entire array of random numbers, thus generating random chances for apoptosis, proliferation, and migration for each cell at the beginning of each time iteration. Those chances are stored in three separate arrays and compared to the respective set probabilities when the behavior of each cell is tested and decided.

For a model to describe different tumor geometries, random arrays can also be applied when considering the direction along which the tumor will effectively grow. In this cellular automaton, such direction is considered twice, first when the coordinate array is swept and behavior is attributed to each cell. If any specific direction is arbitrarily chosen, it will interfere in growth orientation (i.e. left to right). Therefore, the coordinates regarding cells can be randomly accessed during each time step by means of *numpy.random.shuffle* command, guaranteeing that no geometric shape will be favored during tumor evolution.

The second time occurs when checking for free adjacent spots (or neighbors) each time a cell might proliferate or move. An array containing the directions for each of the 8 possible neighbors (see Fig. 1(a)) is shuffled every time a cell behavior is activated. Then, tumor matrix is checked for each possible location, returning the content when the first free lattice is matched. If all adjacent spots are occupied, cell quiescence command is passed.

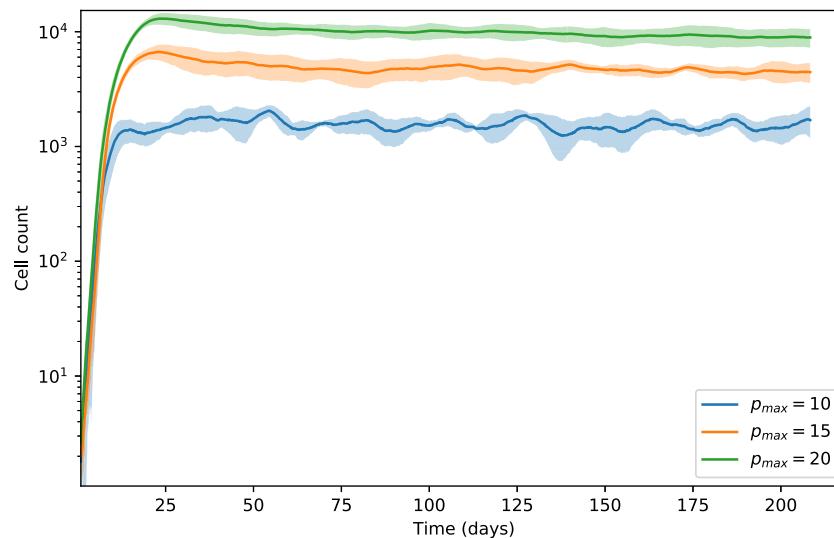
It is worth mentioning that although this approach of not checking every neighbor before acting is faster according to [33], it may be problematic when extending the stochastic cellular automaton model to a hybrid model. In this case one would need to have information on all neighbors surrounding a tumor cell, since it may “prefer” to migrate to a place with a higher nutrient availability.

#### 2.3.4. Dynamically growing domain

A frequent problem when modeling tumor growth is that one does not know a priori the final size of the neoplastic mass. One would have to know beforehand properties such as cell density and diffusivity besides emergent complex dynamics. Therefore, it may be difficult to establish tumor matrix size in the cellular automaton.

Poleszczuk and Enderling [33] suggest that a dynamically growing domain could be used as a workaround for such problem. Therefore, the present model starts every simulation with a small  $11 \times 11$  matrix whose center hosts the progenitor tumor cell. After every new proliferation or migration, the model keeps track of how close every tumor cell is from this domain/matrix borders. The automaton is coded to add a few rows and columns to tumor matrix and re-center the old lattice to the new one every time that a tumor cell reaches the penultimate empty spot in every lattice direction. This approach allows the model to effectively deal with different types of tumor configuration by relying on very large matrices only when really necessary.

Such approach has an important caveat. As this automaton was developed aiming at a future expansion to a hybrid model, dealing



**Fig. 9.** Second scenario: Population dynamics of a tumor originated by a clonogenic STC for different proliferation potentials (average of 5 simulations with  $p_{max} = 10$ ,  $p_{max} = 15$ , and  $p_{max} = 20$ ).

with expanding matrices lacking border definitions would be a problem when synchronously considering PDEs (whose borders are really important). Nevertheless, we decided to maintain this approach, relying on linear transformations and scaling tools to handle the future expansion.

### 2.3.5. Dense vs. sparse matrices

Even when using dynamically growing domains, the tumor matrix customarily has a high number of zero elements (characterizing free space). Therefore, it is natural to think of employing sparse matrices since those are commonly used in simulations of mathematical problems (such as PDE solving). However, when sparse matrices were implemented in this model, they lagged considerably behind of other approaches during empirical speed tests (check Section 2.3.6 for more information). Accordingly, in [33] there is no mention to using sparse matrices to increase speed performance.

One way to justify such speed disadvantage is that, depending on the input parameters, this model will often describe dense tumors, whose matrices will constitute in its majority nonzero elements. Additionally, this cellular automaton was designed initially with dense matrices in mind and the version using sparse ones was developed later. Therefore, it is here acknowledged that a similar automaton could be conceived using sparse matrices.

In terms of memory saving, sparse matrices are indeed much more efficient. When running several different tests, such as ones in Section 3, our model stores every snapshot of the lattice for every time step and every replicate of a simulation. This allows revisiting any instance of tumor evolution in order to visualize or post process any needed detail. Nevertheless, the variables produced are very long lists of large matrices. Converting each matrix to its sparse correspondent is a workaround to save considerable disk space (thus improving speed during variable writing and reading).

### 2.3.6. Speed tests

Over 40 different versions of this cellular automaton have been coded during the development of this model. Some versions were incomplete, some were too slow, and others were not very readable or practical. Since the simulation of in silico models tend to become slower as cell populations increase, runtime speed was a main aspect considered during code development.

In this vein, the most recent version of the code has been continuously improved to become faster and more efficient, mainly in line with implementation approaches previously discussed. The impact of some of those implementation techniques on average runtime is presented in

Fig. 6, which compares performance of five representative versions of the code during development. Main characteristics of each code version are described in Table 1. Each test consisted of a batch of 10 simulations using the same parameters and with compatible cell population outcomes. Averages and standard deviations were presented. Code 'E' is the most stable and fast version of the cellular automaton thus far developed and was used for all subsequent simulations in Section 3.

## 3. Results: Case scenarios and virtualizations

In this section, some of the complex scenarios that can be generated from the stochastic cellular automaton are discussed by reproducing the cases studied by Enderling et al. [25]. Firstly, some general parameters and details are presented in Section 3.1. Different cases pertaining dormancy periods, stem cells, and diffusive/dense tumors are then exposed and commented in subsequent sections.

### 3.1. Parameters and details

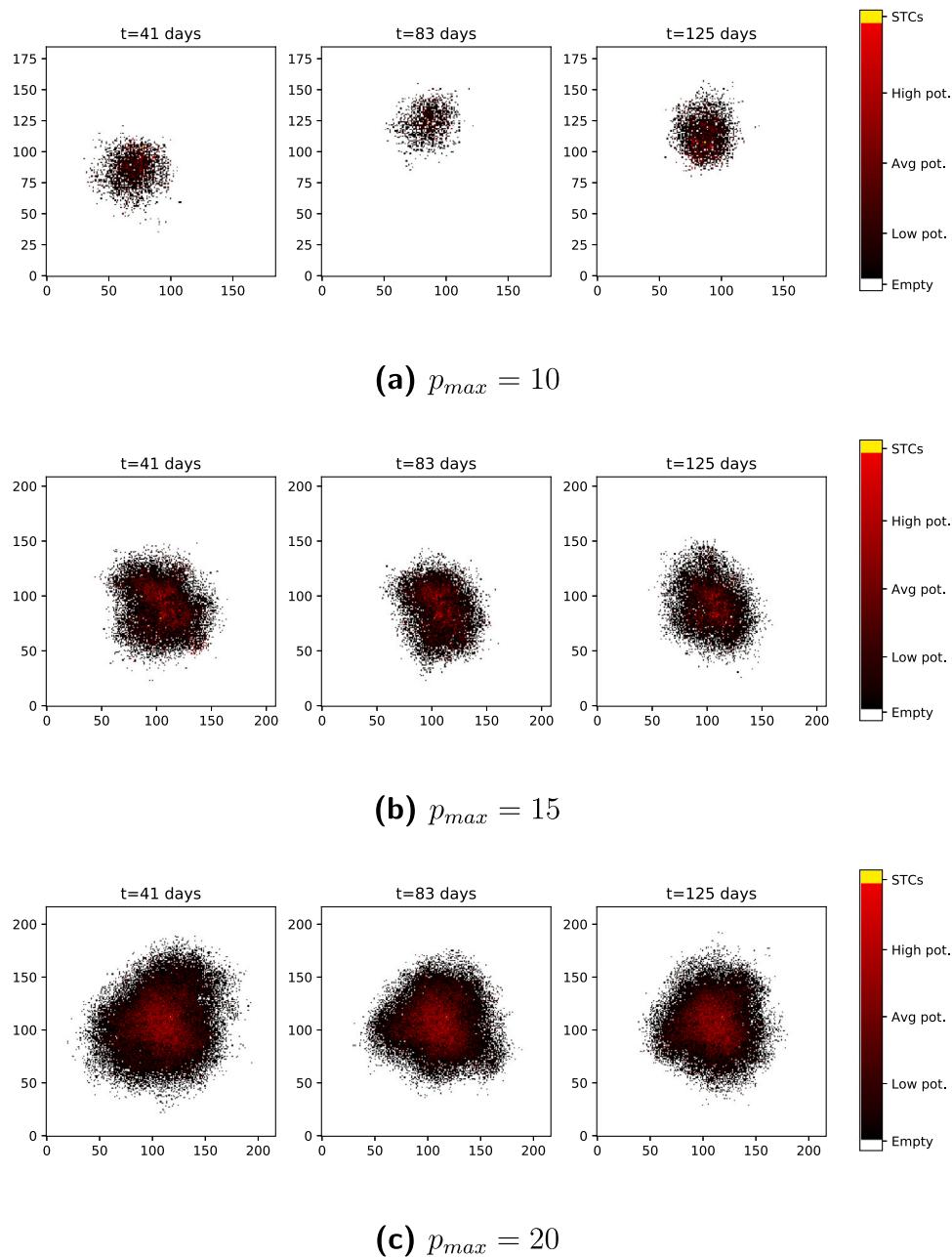
Although the scenarios studied in this section were taken from [25], it is worth highlighting that this cellular automaton model also presents mixed attributes from the one detailed in a more recent paper [33]. Therefore, actual input values used in the simulations were from the latter. Nevertheless, the qualitative behavior and magnitude of results are the same.

In all simulations some parameters were kept constant, namely time step  $\Delta t$ , cell cycle time CCT proliferation probability  $P_p$ , and lattice cell width, whose values are presented in Table 2. Other variables, such as migration potential  $\mu$ , maximum simulation steps  $T$ , maximum proliferation potential  $p_{max}$ , apoptosis rate  $P_A$ , and probability of symmetrical stem division  $P_S$ , were set in view of the tested scenario.

All tumors originated from a single progenitor cell. All simulations were replicated five times. The population dynamics in every scenario considered averages and standard deviations of these replicates. The spatial snapshots of evolving tumors considered the one whose number of cells were closer to the calculated average.

### 3.2. First scenario: Tumor growth from a cell without clonogenic potential

The first scenario refers to an original nonclonogenic cell, i.e. a tumor cell with limited replication capacity. In this scenario, the migration potential is  $\mu = 10$  cell width/day, the probability of apoptosis



**Fig. 10.** Second scenario: Spatial evolution of a representative tumor originated by a clonogenic STC with different proliferation potentials.

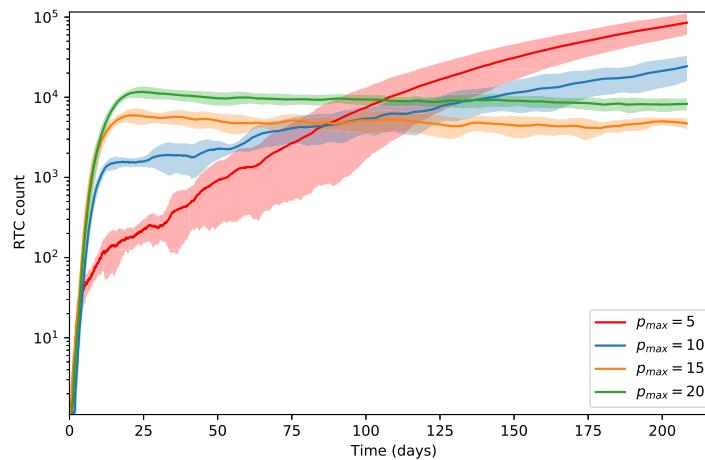
is null  $P_A = 0$  and tumor cells only die after they expire their maximum proliferation potential, which can be  $p_{max} = 10$ ,  $p_{max} = 15$ , or  $p_{max} = 20$ . For each of these cases, the population dynamics is shown in Fig. 7 and a representative spatial snapshot of tumor evolution is pictured in Fig. 8.

Table 3 lists the maximum and final cell count for each of the modeled proliferation potentials. From these results, it is noticeable that, regardless of how many times cells are able to divide, they can never generate a long-lasting tumor with every cell dying after sufficient time is passed. The tumor size is also limited, with cells still capable of proliferating in inner parts of neoplastic tissue while exhausted/dying ones concentrate on the extremities. These results are in accordance with those obtained in [25] and indicate that tumors originating from a nonclonogenic cell will eventually perish.

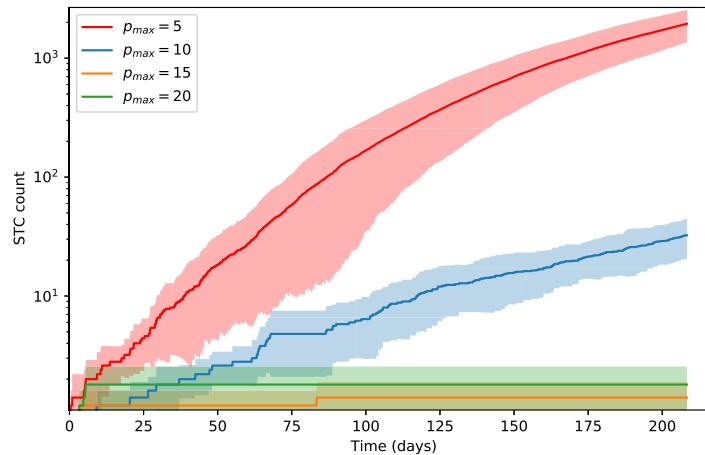
### 3.3. Second scenario: Tumor growth from a clonogenic cell

In the second scenario, the modeled tumor is generated from a clonogenic cell, i.e. a tumor stem cell that though immortal can only generate daughter cells with limited capabilities. Migration potential, probability of apoptosis and maximum proliferation potential are the same as in the first scenario, i.e.,  $\mu = 10$  cell width/day,  $P_A = 0$ , and  $p_{max} = 10$ ,  $p_{max} = 15$  or  $p_{max} = 20$ . The population dynamics is shown in Fig. 9 and a representative spatial snapshot of tumor evolution is presented in Fig. 10. Table 4 lists the maximum and final cell counts for each simulated case.

Differently from the previous scenario, the tumor does not disappear regardless of how many days pass, but reaches a stable size indefinitely maintained. This new feature is due to the clonogenic STC in neoplasm center, which creates a new RTC with full proliferation capacity every time it replicates. Besides, the clonogenic cell never dies, thus replenishing RTC population as those cells become exhausted. The maximum



(a) Evolution of regular tumor cells (RTCs).



(b) Evolution of stem tumor cells (STCs).

**Fig. 11.** Third scenario: Population dynamics of a tumor originated by a true STC with different proliferation potentials (average of 5 simulations with  $p_{max} = 5$ ,  $p_{max} = 10$ ,  $p_{max} = 15$ , and  $p_{max} = 20$ ).

possible tumor size is indicated by cells migration and proliferation potentials, with higher values generating larger masses. It is noticeable from spatial projections in Fig. 10 that, regardless of  $p_{max}$ , tumor shape is always circular (spherical if extended to three dimensions), with new cells in the inside and exhausted cells on the outside. Results from this scenario also match those in [25], where authors claim this outcome can likely be used to describe some types of benign tumors, which live in patients for up to decades and never reach a dangerous size nor become malignant.

#### 3.4. Third scenario: Tumor growth from a true stem cell

The third scenario presents a tumor originated from a true stem cell, i.e., an immortal tumor cell with probability  $P_S$  of generating an identical daughter cell, as indicated in the third frame of Fig. 3. Migration potential is  $\mu = 10$  cell width/day and the probability of apoptosis is null  $P_A = 0$  (same as scenarios 1 and 2). Tumor cells only die after they expire their maximum proliferation potential, which can be  $p_{max} = 10$ ,  $p_{max} = 15$ , or  $p_{max} = 20$ . The chance for symmetrical STC replication is  $P_S = 5\%$ . The population dynamics is shown in Fig. 11 and a representative spatial snapshot of tumor evolution is pictured in Fig. 12. Table 5 lists both RTC and STC final counts for each simulated case.

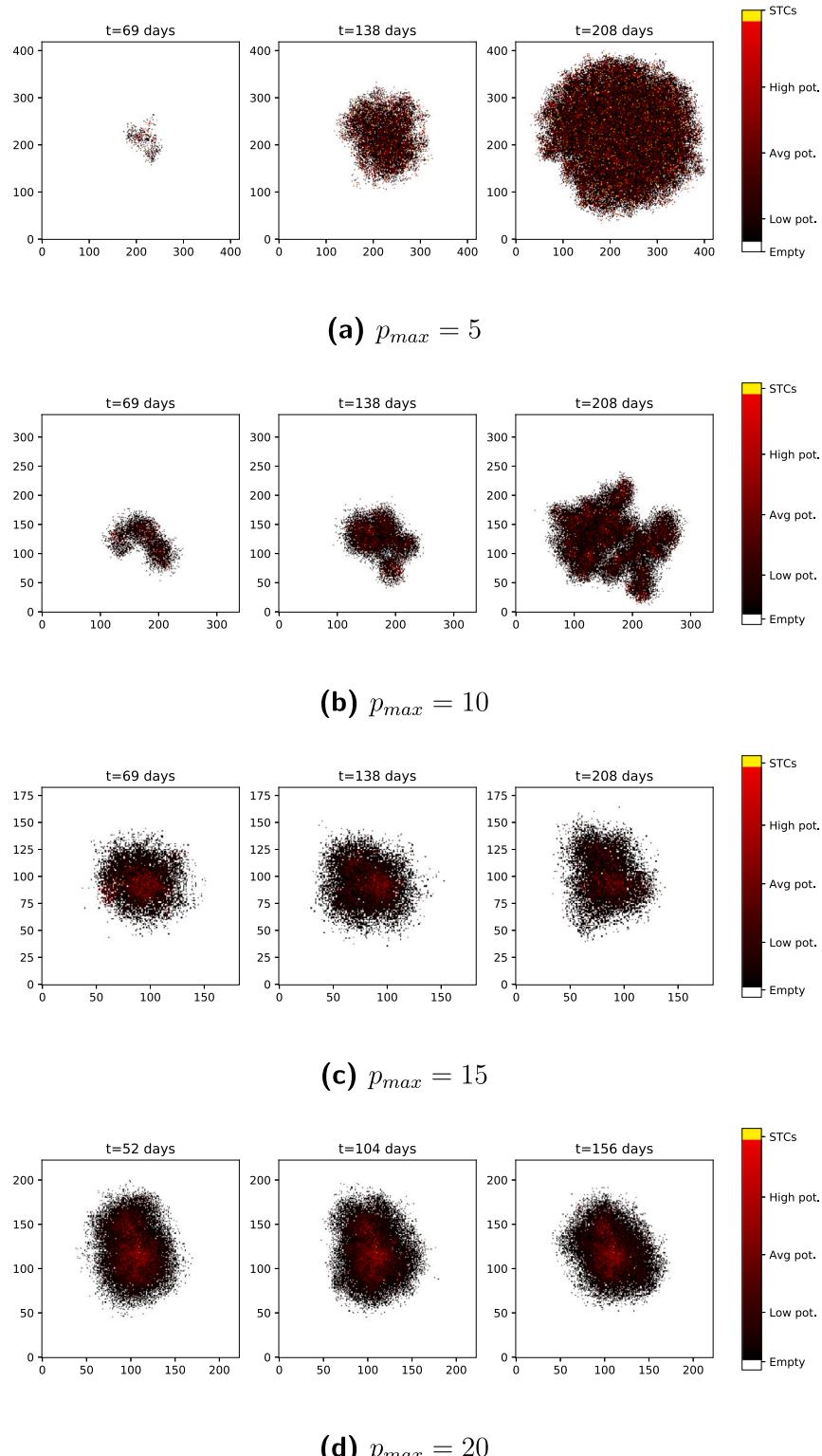
**Table 5**

Average results for the third scenario. Tumor growth from a true stem cell.

Proliferation potential	Final RTC count	Final STC count
$p_{max} = 05$	$85,167 \pm 25,076$	$1947 \pm 594$
$p_{max} = 10$	$24,386 \pm 8479$	$32 \pm 12$
$p_{max} = 15$	$4731 \pm 746$	$1.4 \pm 0.5$
$p_{max} = 20$	$8273 \pm 1385$	$1.8 \pm 0.7$

This scenario is the first one in this paper where tumors can quickly reach large dangerous sizes. As discussed in [25], stem cells population apparently dictates tumor size and endurance. This becomes clearer from Table 5, suggesting that neoplasms with higher RTC count are those presenting more STCs.

Figs. 11 and 12 also highlight a counter-intuitive idea: cells with lower maximum proliferation potential tend to grow much larger tumors. The underlying reason is that as RTCs with low  $p_{max}$  die faster they make available space for STCs to replicate, thus rising chances for symmetrical division and consequently increasing STC population. When RTCs can divide many times before exhausting, they make no room for the very small initial STC population to reproduce, drastically reducing the chances for tumor grow over a stability point. This is what



**Fig. 12.** Third scenario: Spatial evolution of a representative tumor originated by a true STC with different proliferation potentials.

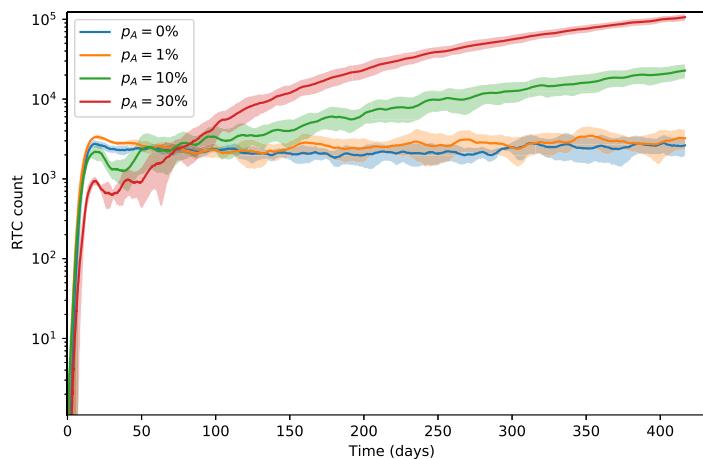
happens in the cases where  $p_{max} = 15$  and  $p_{max} = 20$ , as shown in aforementioned figures.

### 3.5. Fourth scenario: Tumor growth with different apoptosis rates

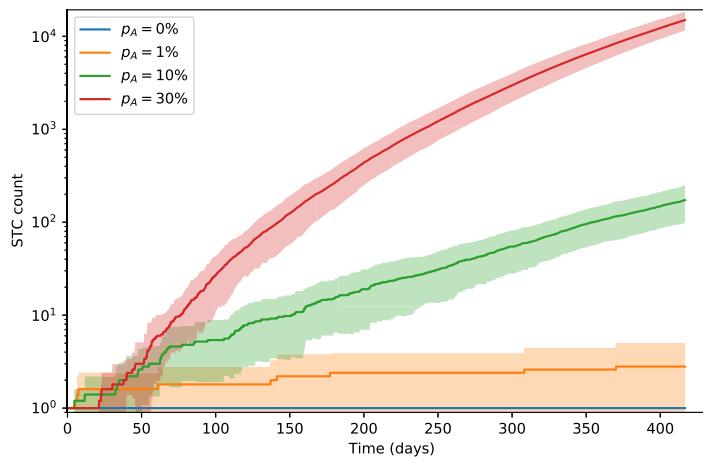
In the fourth scenario, the nonzero apoptosis condition is explored. A tumor originated by a true STC is simulated with  $p_{max} = 10$ ,  $\mu = 10$ ,

and  $P_S = 1\%$  for four different apoptosis rates. For each outcome, the population dynamics is shown in Fig. 13 and a representative spatial snapshot of tumor evolution is presented in Fig. 14. Table 6 lists the final counts pertaining RTCs and STCs for each simulated case.

As thoroughly discussed in [25], outcomes represented in Figs. 13 and 14 are probably the most interesting and revealing among analyzed scenarios. As the chance of programmed cell death increases,



(a) Evolution of regular tumor cells (RTCs).



(b) Evolution of stem tumor cells (STCs).

**Fig. 13.** Fourth scenario: Population dynamics of a tumor originated by a true STC with different apoptosis rates (average and standard deviation of 5 simulations with  $P_A = 0\%$ ,  $P_A = 1\%$ ,  $P_A = 10\%$ , and  $P_A = 30\%$  a day).

**Table 6**  
Average results for the fourth scenario. Tumor growth with different apoptosis rates.

Apoptosis chance	Final RTC count	Final STC count
$P_A = 0$	$2628 \pm 677$	$1.0 \pm 0.0$
$P_A = 1\text{/day}$	$3241 \pm 872$	$2.8 \pm 2.2$
$P_A = 10\text{/day}$	$22,771 \pm 4443$	$173 \pm 75$
$P_A = 30\text{/day}$	$106,296 \pm 9969$	$14,958 \pm 3400$

the overall tumor cell populations also dramatically rise. The extreme case of a 30% daily chance of apoptosis yields a large neoplastic mass with an average of over 120 thousand cells after 400 days, from which around 12% are stem cells. The elevated number of STCs is actually what justifies such large tumors. In fact, as RTCs can die at an increased rate in this scenario, there is much more free space left for STCs to create identical daughter cells (whose chance of apoptosis is always zero).

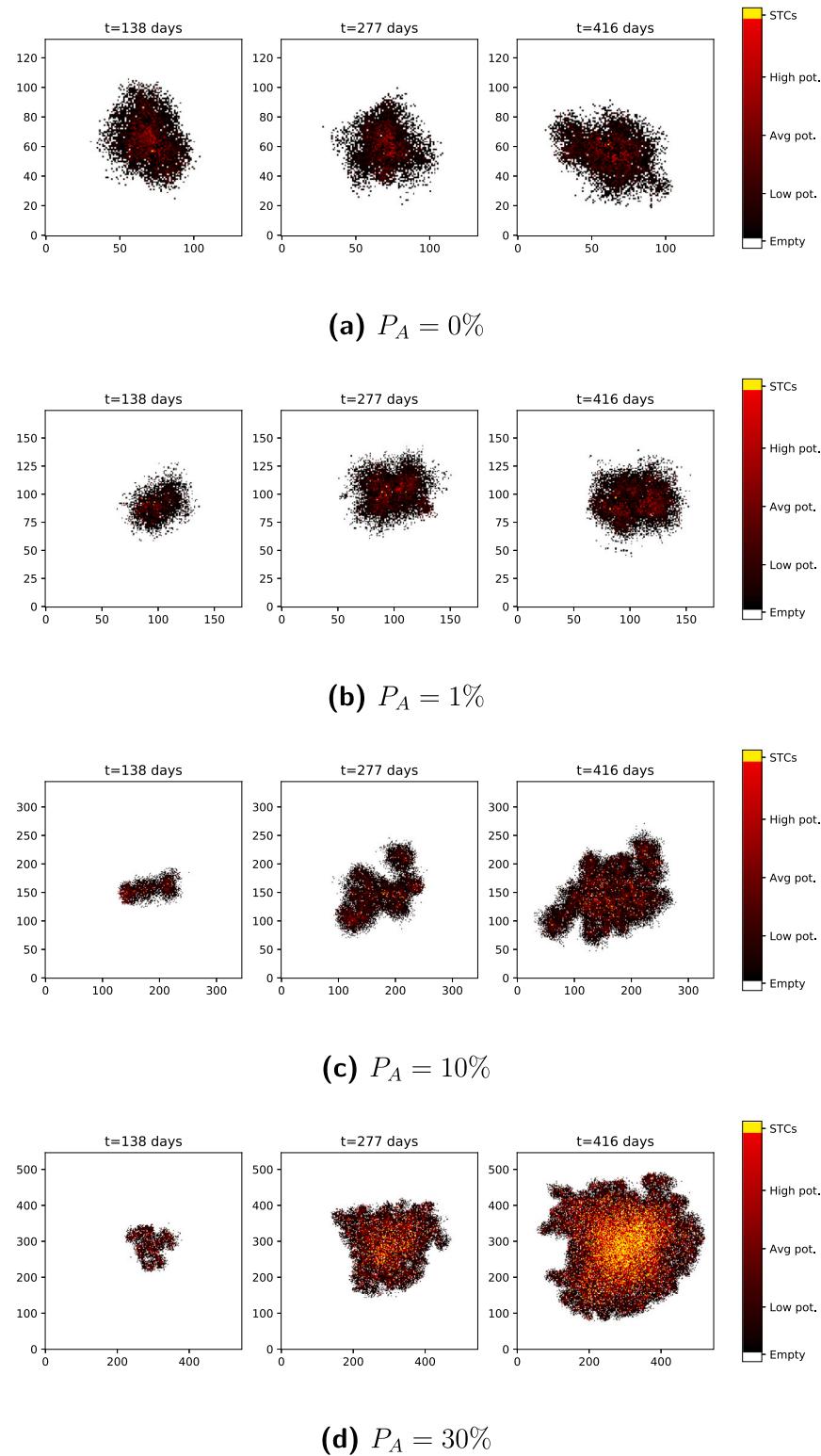
Tumor shapes for cases with higher cell death start to present protrusions around their extremities, a morphological sign of malignant advanced tumors (refer to Figs. 14(c) and 14(d)). Furthermore, when  $P_A$  is low, but nonzero, there is still a chance of a tumor with

morphology and size as illustrated in Fig. 14(b) to become similar to the one represented in Fig. 14(d). The start of this process is depicted in Figs. 13(a) and 13(b), as the population of stem cells plotted by the orange curve slowly rises. Given enough time, overall population will increase, thus creating potentially large and malignant tumors. This phenomenon of very slow growth with low apoptosis rate can be characterized as some types of dormancy periods seen in many cancers.

### 3.6. Fifth scenario: influence of migration potential and stem symmetrical proliferation

The fifth and final scenario accounts for how the migration potential of cells and the chance of symmetrical replication of STCs affect tumor size and shape. In all cases analyzed in this scenario,  $p_{max} = 10$  and  $P_A = 1\%$ , but  $\mu$  and  $P_S$  varies from 1 to 10 cell width/day and 1% to 10%, respectively. In each case, population dynamics is presented in Fig. 15 and possible spatial forms regarding tumor evolution are represented in Fig. 16. Table 7 lists both RTC and STC final counts for each case combination.

The main interesting aspect from the outcomes of the fifth scenario is that every combination can lead to potentially large and malignant cancers (Figs. 15(c) and 15(d) show growing STC populations in all cases — albeit some much faster than others). Growth speed, on the

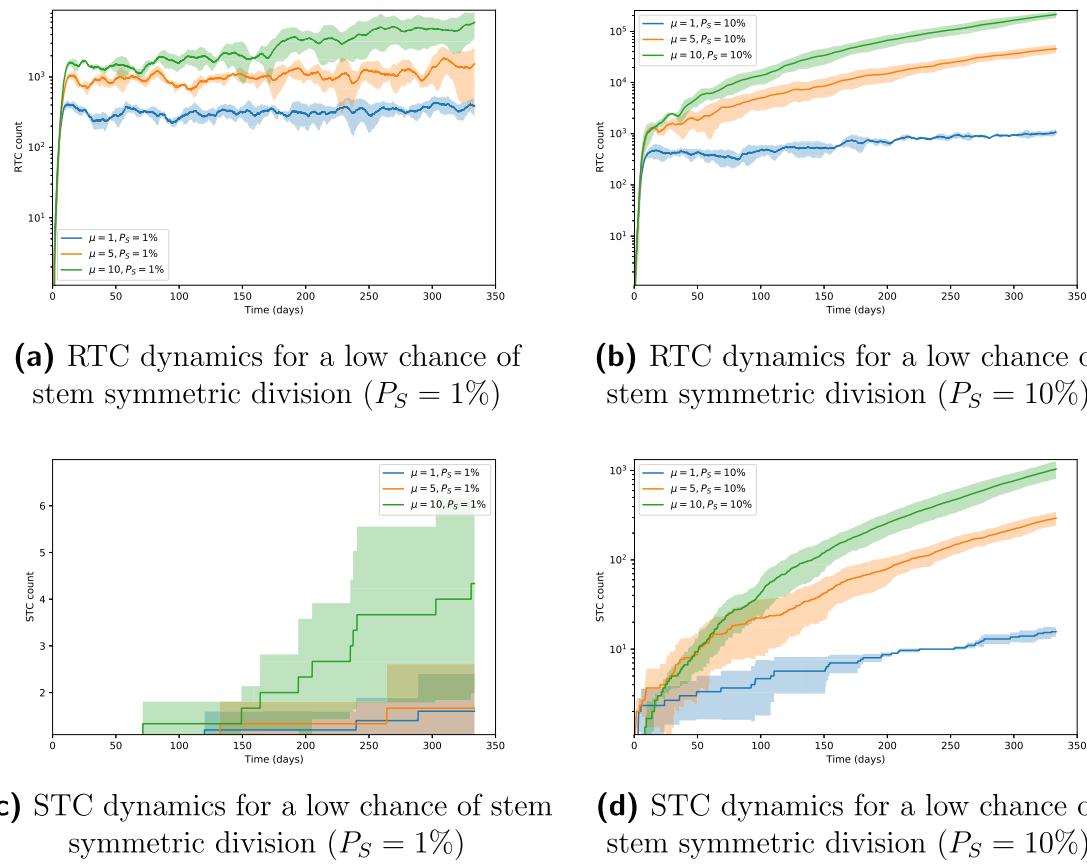


**Fig. 14.** Fourth scenario: Spatial evolution of a representative tumor originated by a true STC with different apoptosis rates.

**Table 7**

Average results for the fifth scenario. Influence of migration potential and stem symmetrical proliferation.

Migration pot.	$P_S = 1\%$		$P_S = 10\%$	
	Final RTC count	Final STC count	Final RTC count	Final STC count
$\mu = 1$	$383 \pm 110$	$1.6 \pm 0.8$	$1063 \pm 148$	$16 \pm 2$
$\mu = 5$	$1541 \pm 1002$	$1.7 \pm 0.9$	$45,494 \pm 8729$	$294 \pm 50$
$\mu = 10$	$5971 \pm 2483$	$4.3 \pm 2.4$	$212,795 \pm 32,244$	$1043 \pm 227$



**Fig. 15.** Fifth scenario: Population dynamics of a tumor originated by a true STC with different migration potentials and probabilities of stem symmetrical division (average and standard deviation of 5 simulations).

other hand, seems to be positively connected to  $\mu$  and  $P_S$  values: the higher their values, the more sprawling the resulting neoplasms. As clusters appearance is also identified, some combination of parameters could yield a scenario in which distant metastasis can be virtualized.

In case of highly migrating cells with low potential of STC creation, cell clusters are created (Fig. 16(e)) and chances of a resulting invasive diffusive tumor are higher. If cells are movable and still generate a high number of STCs then the resulting neoplasms will be denser but equally prone to invasion and with a much larger overall mass (Fig. 16(f)).

#### 4. Model limitations and future expansion

The automaton model discussed in this paper is not only capable of describing several different scenarios, as shown in Section 3, but also is ready to be expanded upon. By coupling PDE systems to the automaton, it is possible to improve the solely stochastic characteristics of the CA, enabling deterministic features to compensate some shortcomings, thus transforming the former into a hybrid model.

Hybrid models are a recent category in which continuum characteristics are incorporated into discrete models. Advantages of such approach are clear for modeling multi-scale phenomena since the discrete part can focus on the scale of cell movements while the continuum methods can model events on larger scales [35]. This capacity of being able to bridge scale gaps while communicating aspects of different magnitudes across the model makes hybrid approaches very interesting for describing several aspects of cancer phenomena [36].

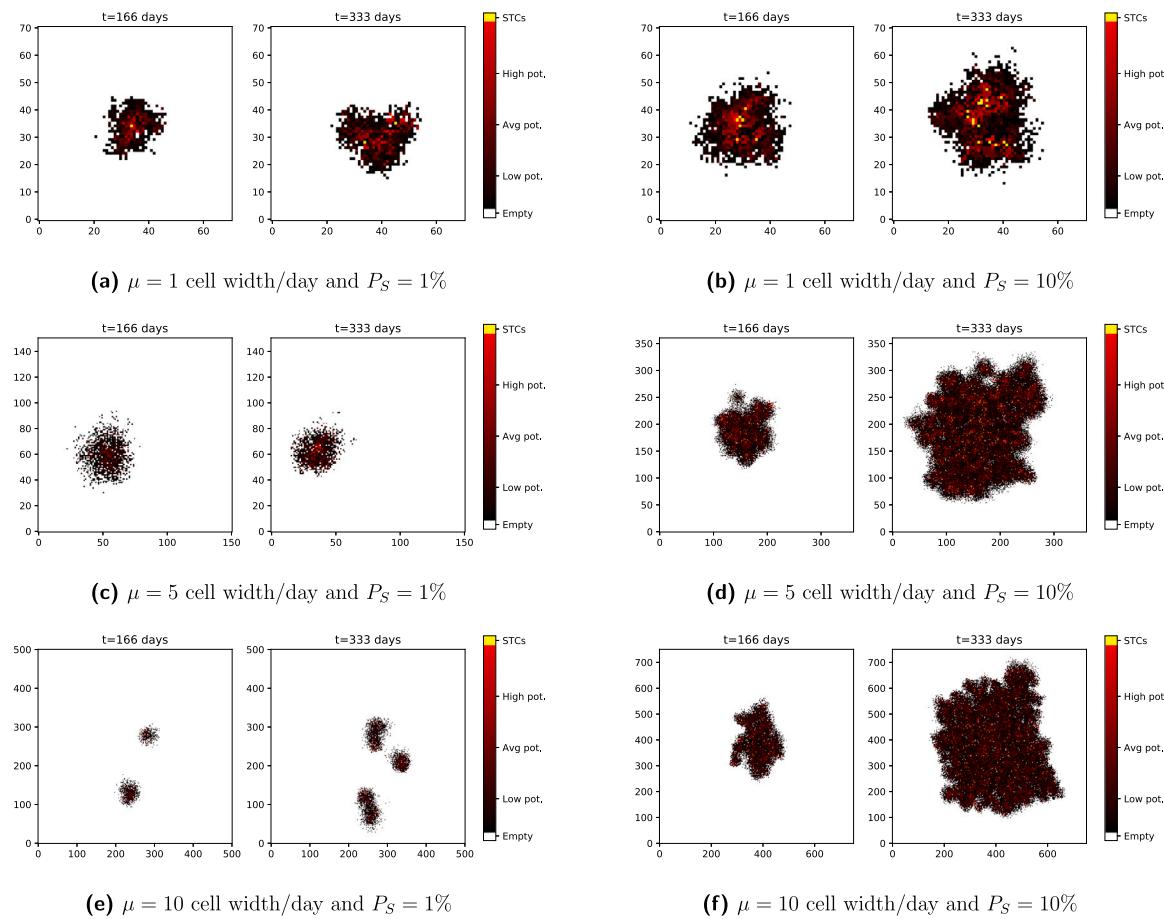
Accordingly, Anderson et al. [19] proposed a hybrid model constituting of discrete methods to deal with tumor cells while considering continuous methods to model micro-environment factors such as host tissue, matrix-degradative enzymes and oxygens. Their model focused

on the micro-scale level to produce simulations of tumor at tissue-scale and could be easily implemented to incorporate other range of scales (such as sub-cellular). Consequently, many other hybrid models emerged, each with their own characteristics and often involving different discrete and continuum tools [37,38].

There are some stochastic CA model's main shortcomings that could be directly compensated. Firstly, by disregarding dead cells, the CA model also dismisses the remains of those cells, which could cause toxicity unbalance in tumor micro-environment. Secondly, the CA model does not take into account the nutrient availability in the tissue where the tumor grows. It is known that tumors can react very differently depending on oxygen lack or abundance. For instance, some cells can effectively change the biomechanical characteristics in order to migrate if they are at a oxygen-deprived environment [39]. A diffusion PDE coupled to the CA could be used to tackle both these problems.

Therefore, the diffusion equation could be an important tool to model the micro-environment surrounding the tumor. It could correctly describe oxygen (or other nutrient) being diffused throughout the tissue in which the tumor grows as modeled by the CA. Simultaneously, it could also account for the toxicity of cell remains in the environment. Following transport laws, this part of the model would be completely deterministic while also depending on outcomes of the stochastic CA (e.g., if a cell replicates, it will increase the nutrient consumption in that lattice area, interfering with the diffusion equation). On the other hand, the deterministic portion of the model solved at each time step would also interfere with probabilities generated by the CA (e.g., if nutrient availability is very low, the chance of a RTC undergoing apoptosis in that area is higher).

Another problem with the CA model in its current form is that it does not take healthy cells into consideration, not establishing any stress relation between cells and the surrounding extracellular matrix,



**Fig. 16.** Fifth scenario: Spatial evolution of a representative tumor originated by a true stem cell with different migration potentials and stem replication probabilities.

fact shown to be important in tumor progression [21,40]. As an attempt to improve this characteristic on the hybrid model, a differential equation to model tumor viscoelasticity and its surrounding tissue may well be a very useful strategy.

A hybrid model could potentially contain at least two equations describing tumor micro-environment: one dealing with nutrient diffusivity and the other tackling tissue stresses. In short, the central framework of the model would be the stochastic CA previously described, but it would influence and be influenced by the coupled deterministic models. The equations would be simultaneously solved and updated along with the time steps of the CA. Boundary conditions for the equations would be correctly selected according to the phenomena being considered.

In this context, Fractional calculus can be an interesting tool to model both diffusive and viscoelastic aspects of those systems [41]. Fractional models have been applied to vast number of different areas [42], including in health sciences and biomathematics [43], in which it has provided a singular perspective on mathematical oncology and cancer modeling [44–47]. Furthermore, fractional viscoelastic models have been widely explored in the literature, show promising features to describe tissue heterogeneity [48,49] while non-integer operators are well known for extending the capabilities of diffusive models [50,51].

## 5. Final remarks

In this paper we have introduced and discussed the development and implementation of a stochastic cellular automaton model for tumor dynamics. The model has been created based on main characteristics of the frameworks presented in [25,33].

Although designed from a relatively simple set of agent-based rules, the stochastic cellular automaton model can simulate several different scenarios regarding tumor growth such as dormancy periods, instability caused by cell-death/competition and invasion — effectively capturing the emergency and complexity inherent to oncological phenomena.

Future expansion is also explored, opening the path to transforming the model into a hybrid framework featuring deterministic characteristics from differential equations capable of potentially mitigate model shortcomings.

The framework herein discussed is an interesting tool for *in silico* modeling, with promising capabilities and possibilities to support further research in mathematical oncology, thus improving diagnosis tools and/or personalized treatment.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This study is financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance code 001.

## Appendix

See Fig. 17.

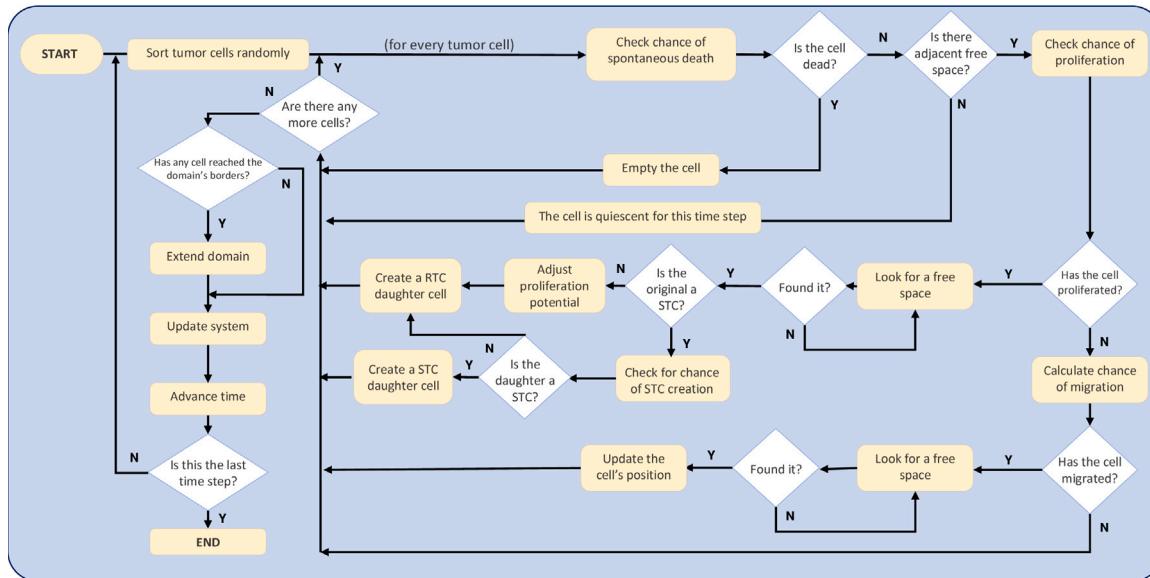


Fig. 17. Logic flowchart for the CA algorithm.

## References

- [1] W.H. Organization, Breast cancer: Fact sheet, 2021.
- [2] International Agency for Research on Cancer (Lyon), Global cancer observatory: Cancer today, 2020.
- [3] D. Hanahan, R.A. Weinberg, The hallmarks of cancer, *Cell* 100 (1) (2000) 57–70.
- [4] D. Hanahan, R.A. Weinberg, Hallmarks of cancer: The next generation, *Cell* 144 (5) (2011) 646–674.
- [5] R.C. Rockne, J.G. Scott, Introduction to mathematical oncology, *JCO Clin. Cancer Inf.* (3) (2019) 1–4.
- [6] S. Hamis, G.G. Powathil, M.A. Chaplain, Blackboard to bedside: A mathematical modeling bottom-up approach toward personalized cancer treatments, *JCO Clin. Cancer Inf.* (3) (2019) 1–11.
- [7] H.M. Byrne, Dissecting cancer through mathematics: From the cell to the animal model, *Nat. Rev. Cancer* 10 (3) (2010) 221–230.
- [8] T. Jackson, N. Komarova, K. Swanson, Mathematical oncology: Using mathematics to enable cancer discoveries, *Amer. Math. Monthly* 121 (9) (2014) 840–856.
- [9] F. Moradi Kashkooli, M. Soltani, M. Souri, C. Meaney, M. Kohandel, Nexus between in silico and in vivo models to enhance clinical translation of nanomedicine, *Nano Today* 36 (2021) 101057.
- [10] M.A. Savageau, Growth equations: A general equation and a survey of special cases, *Math. Biosci.* 48 (3–4) (1980) 267–278.
- [11] R. Sachs, L. Hlatky, P. Hahnfeldt, Simple ODE models of tumor growth and anti-angiogenic or radiation treatment, *Math. Comput. Modelling* 33 (12–13) (2001) 1297–1305.
- [12] E.A. Sarapata, L.G. de Pillis, A comparison and catalog of intrinsic tumor growth models, *Bull. Math. Biol.* 76 (8) (2014) 2010–2024.
- [13] D. Wodarz, N.L. Komarova, Dynamics of Cancer, World Scientific, Singapore, 2014, p. 563.
- [14] S. Benzekry, C. Lamont, A. Beheshti, A. Tracz, J.M. Ebos, L. Hlatky, P. Hahnfeldt, Classical mathematical models for description and prediction of experimental tumor growth, *PLoS Comput. Biol.* 10 (8) (2014).
- [15] N. Hartung, S. Millard, D. Barbolosi, A. Benabdallah, G. Chapuisat, G. Henry, S. Giacometti, A. Iliadis, J. Ciccolini, C. Faivre, F. Hubert, Mathematical modeling of tumor growth and metastatic spreading: Validation in tumor-bearing mice, *Cancer Res.* 74 (22) (2014) 6397–6407.
- [16] M.V. Polovinkina, A. Debboche, I.P. Polovinkin, S.A. David, Stability of stationary solutions for the glioma growth equations with radial or axial symmetries, *Math. Methods Appl. Sci.* (2021) 1–14.
- [17] H. Byrne, L. Preziosi, Modelling solid tumour growth using the theory of mixtures, *Math. Med. Biol.* 20 (4) (2003) 341–366.
- [18] J. Xu, G. Vilanova, H. Gomez, A mathematical model coupling tumor growth and angiogenesis, *PLOS ONE* 11 (2) (2016) e0149422.
- [19] A.R.A. Anderson, M.A. Chaplain, K.A. Rejniak, Single-Cell-Based Models in Biology and Medicine, first ed, Birkhäuser, Basel, Switzerland, 2007, p. 400.
- [20] A. Deutsch, J.M. Nava-Sedeño, S. Syga, H. Hatzikirou, BIO-LGCA: A cellular automaton modelling class for analysing collective cell migration, *PLoS Comput. Biol.* 17 (6) (2021) e1009066.
- [21] H.N. Weerasinghe, P.M. Burrage, K. Burrage, D.V. Nicolau, Mathematical models of cancer cell plasticity, *J. Oncol.* 2019 (2019).
- [22] J. Metzcar, Y. Wang, R. Heiland, P. Macklin, A review of cell-based computational modeling in cancer biology, *JCO Clin. Cancer Inf.* (3) (2019) 1–13.
- [23] Y. Jamali, M. Azimi, M.R. Mofrad, A sub-cellular viscoelastic model for cell population mechanics, *PLoS ONE* 5 (8) (2010).
- [24] F. Jeanquartier, C. Jean-Quartier, D. Cemernek, A. Holzinger, In silico modeling for tumor growth visualization, *BMC Syst. Biol.* 10 (1) (2016) 1–15.
- [25] H. Enderling, A.R. Anderson, M.A. Chaplain, A. Beheshti, L. Hlatky, P. Hahnfeldt, Paradoxical dependencies of tumor dormancy and progression on basic cell kinetics, *Cancer Res.* 69 (22) (2009) 8814–8821.
- [26] A. Deutsch, S. Domm, Mathematical modeling of biological pattern formation, in: Cellular Automaton Modeling of Biological Pattern Formation, Birkhäuser Boston, Boston, MA, 2005, pp. 45–56.
- [27] R. Hu, X. Ruan, A simple cellular automaton model for tumor-immunity system, in: International Conference on Robotics, Intelligent Systems and Signal Processing Vol. 2, IEEE, Changsha, China, 2003, pp. 1031–1035.
- [28] N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using bayesian networks to analyze expression data, *J. Comput. Biol.* 7 (3–4) (2000) 601–620.
- [29] D. Khoshnevisan, Multiparameter Processes, in: Springer Monographs in Mathematics, Springer New York, New York, NY, 2002.
- [30] G. Schneckenreither, Developing mathematical formalisms for cellular automata in modelling and simulation, 2014, p. 144.
- [31] A.T. Crooks, A.J. Heppenstall, Introduction to agent-based modelling, in: A.J. Heppenstall, A.T. Crooks, L.M. See, M. Batty (Eds.), Agent-Based Models of Geographical Systems, Springer Netherlands, Dordrecht, 2012, pp. 85–105.
- [32] Z. Wang, J.D. Butner, R. Kerketta, V. Cristini, T.S. Deisboeck, Simulating cancer growth with multiscale agent-based modeling, *Sem. Cancer Biol.* 30 (2015) 70–78.
- [33] J. Poleszczuk, H. Enderling, A high-performance cellular automaton model of tumor growth with dynamically growing domains, *Appl. Math.* 05 (01) (2014) 144–152.
- [34] C.A. Valentim, Code: Stochastic agent-based cellular automaton model for tumor dynamics, 2022, Made available on December 4, 2022 [https://github.com/carlosvalentim/phd-published/blob/main/CBM-2022\\_Valentim-et-al\\_Cellular-automaton-model.py](https://github.com/carlosvalentim/phd-published/blob/main/CBM-2022_Valentim-et-al_Cellular-automaton-model.py).
- [35] K.A. Rejniak, A.R. Anderson, Hybrid models of tumor growth, *Wiley Interdiscip. Rev.: Syst. Biol. Med.* 3 (1) (2011) 115–125.
- [36] A.R. Anderson, K.A. Rejniak, P. Gerlee, V. Quaranta, Modelling of cancer growth, evolution and invasion: Bridging scales and models, *Math. Model. Nat. Phenom.* 2 (3) (2007) 1–29.
- [37] R. Ibrahim, Z. Moghaddasi, H. Jalab, R. Noor, Fractional differential texture descriptors based on the machado entropy for image splicing detection, *Entropy* 17 (7) (2015) 4775–4785.
- [38] D. Alemani, F. Pappalardo, M. Pennisi, S. Motta, V. Brusic, Combining cellular automata and lattice Boltzmann method to model multiscale avascular tumor growth coupled with nutrient diffusion and immune competition, *J. Immunol. Methods* 376 (1–2) (2012) 55–68.
- [39] D.A. Hormuth, C.M. Phillips, C. Wu, E.A.B.F. Lima, G. Lorenzo, P.K. Jha, A.M. Jarrett, J.T. Oden, T.E. Yankeelov, Biologically-based mathematical modeling of tumor vasculature and angiogenesis via time-resolved imaging data, *Cancers* 13 (12) (2021) 3008.

- [40] A. Taloni, M. Ben Amar, S. Zapperi, C.A. La Porta, The role of pressure in cancer growth, *Eur. Phys. J. Plus* 130 (11) (2015) 224.
- [41] B.J. West, Fractional calculus and the future of science, *Entropy* 23 (12) (2021) 1566.
- [42] D. Baleanu, R.P. Agarwal, Fractional calculus in the sky, *Adv. Difference Equ.* 2021 (1) (2021) 117.
- [43] D. Kumar, J. Singh, Fractional Calculus in Medical and Health Science, first ed, CRC Press, Boca Raton, FL, 2020.
- [44] B.J. West, The fractal tapestry of life: II entailment of fractional oncology by physiology networks, *Front. Netw. Physiol.* 2 (2022) 845495.
- [45] C.A. Valentim, J.A. Rabi, S.A. David, Fractional mathematical oncology: On the potential of non-integer order calculus applied to interdisciplinary models, *Biosystems* 204 (2021) 104377.
- [46] C.A. Valentim, J.A. Rabi, S.A. David, J.A. Tenreiro Machado, On multistep tumor growth models of fractional variable-order, *Biosystems* 199 (2021) 104294.
- [47] C.A. Valentim, N.A. Oliveira, J.A. Rabi, S.A. David, Can fractional calculus help improve tumor growth models? *J. Comput. Appl. Math.* 379 (112964) (2020) 112964.
- [48] M. Ghita, D. Copot, C.M. Ionescu, Lung cancer dynamics using fractional order impedance modeling on a mimicked lung tumor setup, *J. Adv. Res.* 32 (2021) 61–71.
- [49] B. Carmichael, H. Babahosseini, S.N. Mahmoodi, M. Agah, The fractional viscoelastic response of human breast tissue cells, *Phys. Biol.* 12 (4) (2015).
- [50] A. Debbouche, M.V. Polovinkina, I.P. Polovinkin, C.A. Valentim, S.A. David, On the stability of stationary solutions in diffusion models of oncological processes, *Eur. Phys. J. Plus* 136 (1) (2021) 131.
- [51] R.L. Magin, H. Karani, S. Wang, Y. Liang, Fractional order complexity model of the diffusion signal decay in MRI, *Mathematics* 7 (4) (2019) 348.