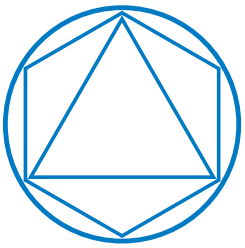
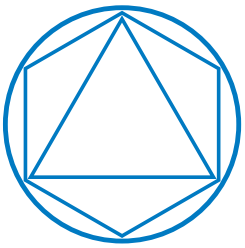


Tracking cell lineage to analyse anti-cancer drug resistance development in an agent based model

Bachelor's Thesis of Johannes Ringler

Supervisor :	Dr. Judith Perez-Velazquez
Advisor :	Prof. Dr. Christina Kuttler
Submission date :	27.11.2015



Tracking cell lineage to analyse anti-cancer drug resistance development in an agent based model

Bachelor's Thesis of Johannes Ringler

Supervisor :	Dr. Judith Perez-Velazquez
Advisor :	Prof. Dr. Christina Kuttler
Submission date :	27.11.2015

Abstract

In many medical treatments there can occur resistance to the used therapeutic agents. Unfortunately this happens to be one of the major issues dealing with cancer via chemotherapy. This of course leads to a lower medical response or even to treatment failure. In the cancer case often the resistance is not a pre-existing factor but arises due to the chemotherapeutic pressure.

In order to analyse this behaviour Gevertz et al. set up a model in [7]. They used a hybrid discrete-continuous mathematical model to describe the events in a cancer populated slice of tissue over time. The aim was to analyse the effects of the different kinds of resistance on the tumor development and finally find the causes why treatment inefficiency or failure occurred.

The objective of this thesis was to enhance the analysis of the Gevertz et al. developed model by examining clone development. To achieve this, the basic model had to be modified to allow an output capable of reconstructing the clonal evolution retrospectively. Furthermore routines were developed for processing the generated data. The outcome of the analysing routines supports the suggestion from Gevertz et al. regarding that the spatial location is significantly important for the clonal development of resistance. Tumor heterogeneity can be examined through the cell lineages. By following a single long surviving cell lineage, it is clear that there is often a fine line between death and survival of a lineage/clone.

Zusammenfassung

In vielen verschiedenen medikamentösen Behandlungen kommt es zur Resistenz des Organismus gegen das eingesetzte Arzneimittel. Unglücklicherweise ist dieses Verhalten eines der größten Probleme bei der Krebsbehandlung durch Chemotherapie und führt zu einem schlechten oder sogar erfolglosen Therapieergebnis. In der Krebstherapie ist dabei diese Resistenz oftmals ein Ergebnis der Therapie selbst und nicht ein bereits vorhandener Zustand.

Um diesen Verhalten zu analysieren haben Gevertz et al. in [7] ein hybrid diskret-kontinuierliches mathematisches Modell entwickelt um die zeitliche Entwicklung des Tumors zu beschreiben. Das Ziel dieses Modells lag in der Analyse der verschiedenen Arten von Resistenz und deren Auswirkungen auf die Entwicklung des Tumors um schließlich die Gründe für das Versagen der Therapie zu finden.

Der Zweck dieser Bachelorarbeit lag darin, die Analysemöglichkeiten des bisherigen Modells zu erweitern um die Entwicklung einzelner Zellen untersuchen zu können. Dafür musste einerseits das zugrundeliegende Modell verändert werden für einen feineren Output um die klonale Entwicklung rückwirkend genau rekonstruieren zu können, andererseits mussten zusätzlich Programme geschrieben werden um die Output Daten aufzubereiten. Die Analyse unterstützt die bereits in der Abhandlung von Gevertz et al. aufgestellten Hypothese über die Wichtigkeit der räumlichen Struktur für das Überleben mancher Klone. Ebenso kann die Heterogenisierung des Tumors durch die Beobachtung der einzelnen Zelllinien begründet werden. Bei der Betrachtung vieler langlebiger Zelllinien wird einem bewusst, dass oftmals ein schmaler Grat zwischen dem Tot und dem Überleben von Zellen oder gar ganzer Klone liegt.

Declaration

I hereby confirm that I have written the accompanying thesis by myself, without contributions from any sources other than those cited in the text.

This applies also to all graphics and images included in the thesis.

_____

Place, date, signature

Contents

List of Figures

List of Tables

1 Biological background

Cancer describes a large family of diseases which generates lots of different afflictions and symptoms. All in common is the elementary problem. Unlike normal cells, cancer cells are strictly spreading due to their anormal division control mechanisms. This leads to a steady enlarging cell lump called tumor. Additionally cancer cells have in contrast to normal cells no fix allocation and so can spread and invade neighbouring tissue which in the long run leads to a complete system breakdown of the organism. Depending on the sort of cancer cells there are many different tools for medical treatment. Some of them are surgery, chemotherapy, radiation therapy and immunotherapy. These treatments are often combined, but in this work we will focus on the most common treatment, i.e. by a chemotherapeutic agent. The model we consider uses an agent which produces DNA damage over the time until the cell dies as a consequence of the damage.

Like in many medical treatments (e.g. HIV or antibiotic medical treatment) in the chemotherapy occurs medical resistance. There are mainly three different cases. The first case is the impact of chemotherapy on healthy, sensitive, cells. They cannot stand the therapy and certainly die after a long enough period of time. Second, there are resistant cells. Here one can find two types of resistance, the pre-existing and the acquired resistance. As the naming suggests, pre-existing resistance directs to a subpopulation of the cancer cells that was already resistant before the start of the treatment. On the other hand acquired resistance stands for the observation that cells were not resistant prior to the treatment and gain a certain degree of immunity due to the chemotherapeutic pressure. From a medical point of view it is difficult to determine which sort of resistance prevails, but this would be necessary to choose the right treatment plan. Regardless of which resistance occurs, the worst case would be the treatment failure in which the cancer cells could not be erased completely and will recur more resistant and vigorous. This phenomenon was also observed by Frei and Freireich after a high dose treatment against leukemia. At first the study showed good results but at once many patients were in a much worse medical condition and a second chemotherapeutic treatment did not have any results. The problem in this case was that the cancer cells colonized the brain and that is a part of the body where chemotherapy often fails because the blood-brain barrier is a natural defense system of the body against foreign substances. [11]

This work aims to analyse chemotherapy resistance development by tracking surviving clones back on a model by Gevertz et al. [7] which involves two mechanisms of resistance.

The ability of drug resistance is most likely related to the evolution. In order to survive, mammals have evolved mechanisms to protect cells against cytotoxic compounds. These

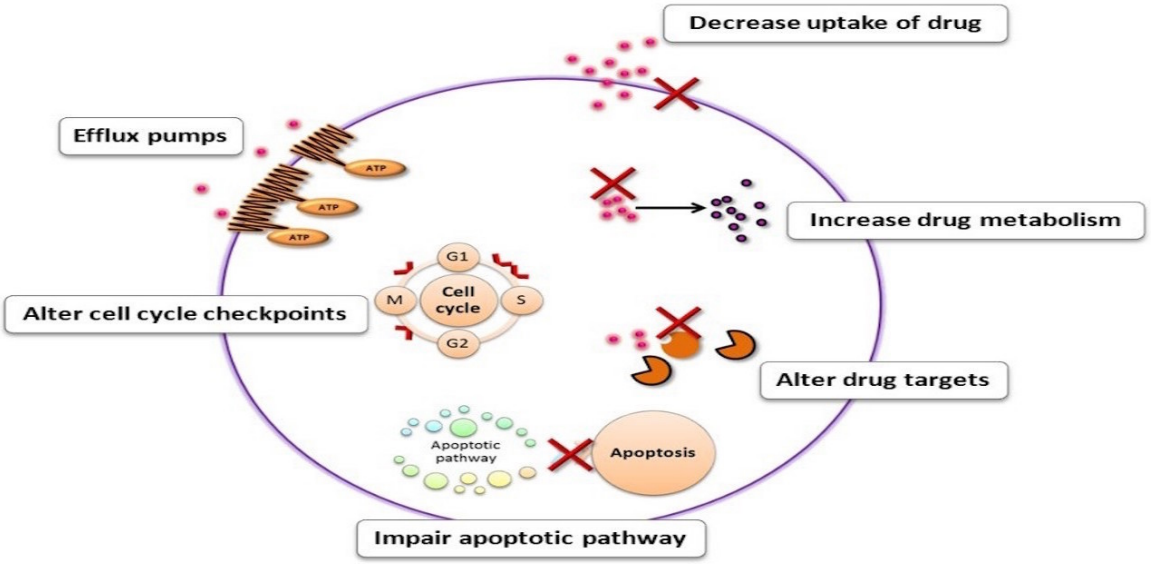


Figure 1: Some mechanisms of drug resistance [13]

mechanism often are activated as a natural protective reaction when a medical drug is administered because these are unfamiliar substances to the human body. This could be a problem when using a particular treatment and even more considering that often drugs are used in combination. Hence over coming resistance to one medicament could lead to no or less response to another treatment of a complete different disease. This phenomenon is called multi-drug resistance. So it is important to know about the mechanisms of resistance in order to switch to other therapies with other modes of action to gain a reaction.

Of course there can be many reasons why the drug cannot work properly. For example an irregular tumor vasculature which has areas not reached by the drug or because it is absorbed by the cells in between. Another reason why the medicament does not reach the cell could be areas with high fluid pressure in between which operate like a barrier without passage. Also low serum levels can occur as a result of rapid metabolism or excretion of the drug. There are more external possibilities, some intrinsic mechanisms how resistance is generated are shown in figure 1 with important implications for drug designing and therapy decisions.

One obvious way to prevent damage is not letting the drug in or dock. There are mainly three ways a cell exchanges nutrients/molecules/etc. with its environment: Diffusion of fluids across the plasma membrane, transport via receptors or transporter proteins (piggy backing) and endocytosis. Completely terminating the traffic is not possible for the cell as it also needs goods for a living. Hence the cell tries to reduce the drug input (or

even all input) through mutations or modification of the cell surface molecules or have defective endocytosis. If the drug either way managed to get into the cell there are energy depended efflux pumps. They work with a cassette of binding transporters which normally are there for moving nutrients or biological important molecules. But those can also detect different kinds of substrate (e.g. hydrophobic natural drugs) entering through the cell membrane. If it is a undesirable substance, they bind it to themselves and guide them to the extracellular space. Afterwards the transporters restore to their original structure and are operational again.

Once a anti-cancer drug achieved to get into the cell, many of them have to undergo a specific metabolic activation in order to acquire clinical efficacy. So one way of the cell to protect against the drug is a very strong enhancement of the drug metabolism to degenerate the activated drug before it can work. Another way is the reduced activation of drug and wait for its efflux without impact. A third possibility is the inactivation of the drug by modification. This could be done by partially degrading the chemical substance or complexing the drug with molecules/proteins with the result that the key-lock pair does not match any more. This method also can be used on the target side of the drug leading to a less effective or no impact.

Especially DNA/RNA damaging drugs are resting until a certain cell cycle checkpoint is reached and then interact in every run and the damage is rising step by step. The DNA damage response mechanisms can not work that fast or even do not recognize a defect and so the cell will die when a threshold is exceeded. Known mechanisms in this case are that the cell cycle is slowed down so that it is enough time for the DNA repair or the checkpoint signals are altered so that the drug does not get active. Another resistance factor in this case is simply a higher tolerance of DNA damage which mostly is pre-existing. Further, cells of a multicellular organism are an organized community also regarding cell number regulation. The process of the programmed cell death is called apoptosis (Greek for "falling off" like leaves on a tree). Some drugs aim to activate this natural process but mutations of the activation sequence or modulations of this pathway are observed techniques of the cell to protect itself from the treatment. The environment in which the cells live plays an important role. For example if the environmental conditions are bad (e.g. drug exposure) often a subpopulation of the cells are shifted in a quiescent state. This means that the cell ceases the exchange with the environment, stops proliferation and rests until better environmental conditions are reached.

Usually in current practice a mixture of drugs with different modes of action are used as a "backup" if one agent has no effect because cells are resistant to a mechanism. Otherwise independent cells can develop the different presented modes of resistance to the

treatment (in case of acquired resistance). Often cells of the same clone generate similar phenotypes. This motivated us to the analysis of lineage trees of single clones. [3, 5, 15]

2 Mathematical modelling

At first we have to point out the difference between a system, a model and a simulation. Generally speaking a system is a theoretical construct to understand processes in the world. A system consists out of two basic parts. A set of objects and the relationships between them. Additionally if it is not an isolated system; there exist interactions between the system and its environment. The delimitation of a system is given by boundary conditions which vary depending on the task to solve.

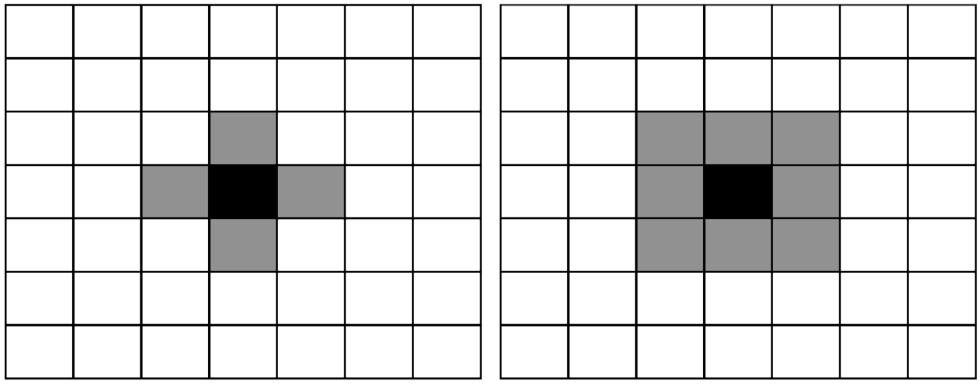
In the ancient Rome the Latin word "modulus" described a small scale replica of a building. These can be transferred to the term model. A model is a simplified representation of a real system through specifying it in some formalism. The aim is to emphasise the important aspects and neglect the unimportant features for the task by abstracting. This may reflect in loosing a certain degree of accuracy. Also, there are many possible models for a system and one cannot declare one superior to the other as its accuracy depends on the question to answer.

The motivation of setting up models for real world problems is to predict the future state of a system. This leads to the third term, the simulation. This simply means numerically or computationally running the model for a certain initial state of the system. Here the importance of neglecting aspects of no interest for the better performance may be relevant. [2]

So far we dealt with the terminology in general. As we here are interested in mathematical modelling this means that our model formalism is represented by functions and equations. In the following we introduce two types of "bottom-up" models. In contrast to the "top-down" models, which try to describe the global processes and transitions as a whole, the "bottom-up" principle proceeds from describing microscopic views which together form the macroscopic entirety.

2.1 Cellular automata

While there is a present popularity in research and application, the concept of cellular automata (abbr. CA) dates from the mid 20th century. Around 1950 the idea of CA was introduced by Stanislas Ulam, John von Neumann, and Konrad Zuse. This abstract object provides a possibility to simulate systems and processes by the discretization of



(a) Neumann (b) Moore

Figure 2: Classic neighbourhoods for CA in two dimensions [14, p.29]

time, state and space. At first it was developed for one dimensional problems but most of its applications today are in two or three dimensions. Generally a d-dimensional CA consists out of four components:

- A regular, discrete, infinite network representing the space structure. The individual parts of this network are called cells. Mathematically you can represent this by $\mathbb{Z}^d =: \Omega$.
- A finite set of elements (Ψ) representing the possible cell states.
- A finite subset $\omega \subset \Omega$ where $|\omega| = n$ for every cell in Ω. This is called the neighbourhood and is temporally and geometrically uniform.
- An update function or local transition rule $\delta : \Psi^{n+1} \to \Psi$. It is a local, deterministic and uniform function.

Assume the current timestep to be t. The transition to timestep $t + 1$ is done synchronously for all cells in Ω. As the update function is deterministic and the update is done synchronously the global evolution of the CA will always be the same provided that the initial configuration is identical. [10, 14]

Every finite subset of Ω could define a neighbourhood but the classic neighbourhoods are the nearest neighbourhoods depending on which norm is used. Let us consider a cell $z \in \Omega$. At first have a look at the 1-norm ($\|z\|_1 = \sum_{i=0}^{d} |z_i|$)[1] and its corresponding distance function $dist_1$. On the basis of that, the Von Neumann neighbourhood is defined as $N_{VN}(z) = \{x \in \Omega : dist_1(z, x) \leq 1\}$. Analogously the Moore environment is defined by

[1] d is the dimension of a general CA, $d \in \mathbb{Z}$

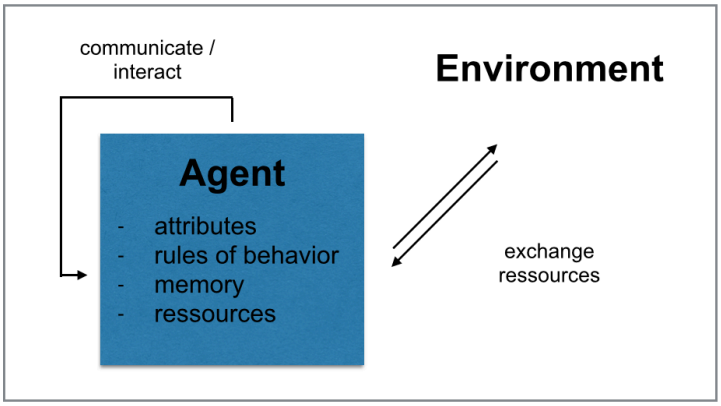


Figure 3: Overview/relationship of the terms agent and environment

the ∞-norm ($\|z\|_\infty = max\{|z_i| : i \in \{1, ..., d\}\}$) as $N_{MO}(z) = \{x \in \Omega : dist_\infty(z, x) \leq 1\}$. Both are graphically illustrated for a two dimensional CA in figure 2. [14]

CA have many applications like in traffic simulation or fluid simulation but certain capabilities are limited. As the complexity in modern times raised (e.g. robotics, social science, server-client simulation) and also the computational possibilities increased, a new type of model was necessary.

2.2 Agent-based modelling

The agent-based model (abbr. ABM) is a very young discipline in contrast to the CA. It originates from the artificial intelligence research and describes a kind of an evolution of the CA. A ABM is structured by the environment and individual entities, called agents, situated in the given environment. Now the term agent is characterized in a general way as there is no explicit definition so far.

An agent is a system in an environment. The environment is not just the spatial space the agent is located in, there can be interaction between the agent and the environment. The latter allows for the usage of the resources of the environment (e.g. input of data or goods) and consequently the manipulation of those in the environment. Further, an agent can move freely (depending on the implementation) in space and is an autonomous object. This means that at every timepoint the agent can decide which option would be the best (in the implemented sense).

As there are usually many agents in an ABM there arise two questions. Are the single agents different from each other and do they have relationships? For the first question the most important fact is that the agents are diverse and heterogeneous. So primarily you can say every agent is a unique entity but on the other hand there can exist more agents

with the same or partly the same properties. Those form groups and are (normally) important for observations concerning e.g. the dominance of different types of agents. The second question depends on the implementation of a model. The theory allows every possible scenario, from no relationship to interactions and exchange of goods with other agents. A graphical summary can be seen in figure 3. Every agent has a limited point of view, they do not see the whole system, only a part of it in their periphery. If in an agent based model the agents are computed asynchronous, the simulation of such a model can generate different outputs in different runs although the initial conditions were the same. [4, 8]

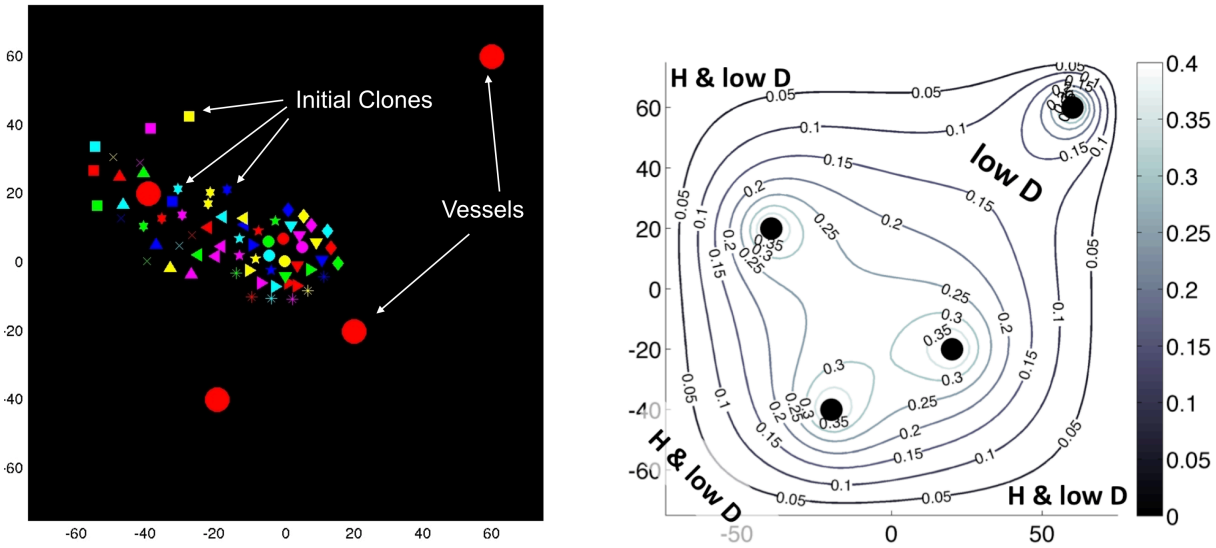
2.3 Cellular automata vs. agent-based models

After a short introduction in the two types of models we continue by a comparison in order to work out their important differences. This could also be helpful for choosing one of the two models for a given task.

So first starting with the environment. The CA always has to be structured in a grid-like way (transferable on $\mathbb{Z}^d$) whereas the ABM does not need any given grid structure. Second the cells in a CA are locally fixed and only can attain one state. A movement can only be modelled by the "movement" of a state from cell to cell. This leads also to a limited movement possibility (usually 1 unit). In contrary, agents can move in the environment freely and can carry a lot of information / data.

Evolution of the model in a simulation is a time-dependent process. The CA dictates that this is a discrete step-by-step process following fixed rules which are applied simultaneously. This may lead to a deterministic behaviour. On the other hand in an ABM it is no specified if the time is considered discrete or continuous. However one has to mention that as the model has to be computed it is only a virtually continuous time scale. The state transition in this case depends on many factors, just to mention a few like spatial positions, agent properties, etc.. Because of the asynchronous computation of the agents in the model this cannot be deterministic.

In general you can work out two "CA vs. ABM statements" describing the main differences pretty well: standardised strict rules vs. degrees of freedom, simplicity vs. adequate accuracy. The main questions of interest in both models are the same. Will a specific model form (agents types or cell states) dominate in the future? Will simulations have a stable mix of agents in the future, or respectively, will there be a steady state in a CA? So in the end it is all a question about complexity which model you will choose for a appropriate representation of a system.



(a) Initial configuration

(b) Drug gradient

Figure 4: In (a) one can see a exemplary starting configuration where the 65 different initial cells are marked with different symbols to graphical track the different clonal evolution. In (b) one can see an exemplary drug gradient with the described niche. [7, p.31]

2.4 Agent-based hybrid model

After some theoretical presentation of possible ways for modelling systems, now the hybrid discrete-continuous mathematical model used in this thesis is introduced. It was developed and presented by Gevertz et al. in [7] and aimed to model cancer growth processes while treatment with a DNA damaging chemotherapeutic agent. Thereby, the interaction between the cells and the different types of resistance is regarded. In the following the model under study is abbreviated by WhAM.

WhAM is a two-dimensional ABM where the environment is a small slice of tissue with four fixed points to represent the blood vessels. These are necessary for the oxygen and drug supply. Their location is fix in every simulation to allow us to draw conclusions about the different resistance types or drug protocols. Otherwise results could be associated to the different positioning of the vessels. Also always the same amount of 65 initial cells are observed, only the mode of resistance and their initial location can be changed in the case of pre-existing resistance. Then, resistant clones can be placed either very near the vessels for good oxygen and drug supply or at a intermediate distance. It is also possible to place them in a position between the vessels to investigate niche formation. The spatial configuration can be seen in figure 4. All other influences around the tissue and all other tissue components are neglected for simplicity.

The model is declared as a hybrid model because it has two types of modelling compo-

nents. The ABM with discrete time steps and variables and parallel the system's supply of oxygen and drug is modelled by a reaction-diffusion partial-differential equation. The resulting continuous gradient is evaluated at the discrete time points during the simulation.

Previous to the presentation of the model equations we shortly mention here the main results gained from the model simulations. In the case of pre-existing resistance it is observed that with a low DNA damage repair term the tumor is eradicated, with a medium DNA damage repair term resistant clones survive whereas sensitive clones die and with a high DNA damage repair term all clones survived regardless their phenotype. Similarly in the acquired resistance case. With slow increase of the death threshold all cells died; with an intermediate increase some of the clones died but a resistant subpopulation emerges and with high increase the treatment fails as (nearly) all clones survive. Also spatial dynamics are observed. Cells near the vessels are killed very fast otherwise cells far away from the vessels or in drug/oxygen niches survive much longer or rest in quiescent states and emerge later in a resistant tumor.

2.4.1 Model equations

From now on let $x = (x, y)$ denote the location of continuous variables and (X, Y) defines positions of discrete objects.

Oxygen kinetics

As declared above all nutrients including oxygen are delivered by the vasculature V_j[2] with a constant rate of S_ξ. The oxygen concentration ξ at location $x = (x, y)$ at timepoint t depends on the supply by the vasculature if it is close enough (see equation (1)) as well as the diffusional inflow with diffusion coefficient $\mathcal{D}_\xi$. Additionally we consider the cellular uptake by the tumor cells C_k [3] at this location (see equation (2)) with rate ρ_ξ. All this is unified in the following equation:

$$\frac{\partial \xi(x,t)}{\partial t} = \underbrace{\mathcal{D}_\xi \Delta \xi(x,t)}_{diffusion} \underbrace{- min\left(\xi(x,t), \rho_\xi \sum_k \chi_{C_k}(x,t)\right)}_{uptake\ by\ the\ cells} + \underbrace{S_\xi \sum_j \chi_{V_j}(x,t)}_{supply}.$$

In this equation appear two characteristic functions for determining the neighbourhood which are defined below. R_C and R_V thereby stand for the fixed cell radius respectively

[2] j indexing over the the vessels $V_j^{(X,Y)}$

[3] k indexing over the cancer cells $C_k^{(X,Y)}$

vessel radius.

$$\chi_{C_k}(x,t) = \begin{cases} 1 \; if \; \|x - C_k^{(X,Y)}(t)\| < R_C, \\ 0 \; otherwise. \end{cases} \tag{1}$$

$$\chi_{V_j}(x,t) = \begin{cases} 1 \; if \; \|x - V_j^{(X,Y)}(t)\| < R_V, \\ 0 \; otherwise. \end{cases} \tag{2}$$

In order to get a unique oxygen gradient one has to define boundary conditions. So in this case sink-like boundary conditions are used: $\forall x \in \partial\Omega : \frac{\partial\xi(x,t)}{\partial n} = -\varpi\xi(x,t)$ [4]. The initial oxygen concentration $\xi(x,t_0)$ was determined to get a stable gradient for a healthy tissue or a tissue with cancer population but no chemotherapeutic treatment.

Drug kinetics

The modelling of the drug supply equation is pretty similar to the oxygen kinetics. Like the drug concentration γ at location $x = (x,y)$ and timepoint t firstly depends on the inflow of near vessels and the diffusion. Therefore the diffusion coefficient is depicted by $\mathcal{D}_\gamma$ and the supply rate by $S_\gamma(t)$. Here can be remarked the first difference. The supply rate of the vessels is not necessarily constant in order to have the possibility to test drug protocols with no constant supply over time. Secondly we have to consider the efflux term. We have a cellular uptake with a rate ρ_γ but also a proportion of the chemotherapeutic agent decays represented by the decay rate d_γ.

$$\frac{\partial\gamma(x,t)}{\partial t} = \underbrace{\mathcal{D}_\gamma\Delta\gamma(x,t)}_{diffusion} - \underbrace{d_\gamma\gamma(x,t)}_{decay} - \underbrace{min\Big(\gamma(x,t), \rho_\gamma \sum_k \chi_{C_k}(x,t)\Big)}_{uptake\ by\ the\ cells} + \underbrace{S_\gamma(t)\sum_j \chi_{V_j}(x,t)}_{supply}. \tag{3}$$

The boundary conditions for the drug equation are the same as for the oxygen and the initial condition is the start of the treatment which means that drug is only found at the blood vessels. This leads to: $\forall x \in \Omega \setminus V_k : \gamma(x,t_0) = 0, \; \forall V_k^{(X,Y)} : \gamma(x,t_0) = S_\gamma(t_0)$.

Cell mechanics

In WhAM a cell is represented by its center $C_k^{(X,Y)}(t)$ and a fixed cell radius R_C. Cells do not have a fixed position, when cells collide [5], two repulsive forces $f_{i,j}$ and $-f_{i,j}$ are applied to cells $C_i^{(X,Y)}(t)$ and $C_j^{(X,Y)}(t)$ to move the cells apart. If cells leave the tissue

[4] n inpointing normal, Ω the whole model domain

[5] For i,j cell indices: $\|C_i^{(X,Y)}(t) - C_j^{(X,Y)}(t)\| < 2R_C$

$C_k^{(X,Y)}(t)$	cell position	$C_k^{age}(t)$	cell age
C_k^{mat}	cell maturation age	$C_k^{\xi}(t)$	level of sensed oxygen
$C_k^{\gamma}(t)$	level of accumulated drug	$C_k^{exp}(t)$	time high drug exposure
$C_k^{dam}(t)$	level of accumulated DNA damage	$C_k^{death}(t)$	death threshold
$C_k^{(ID_c,ID_m)}$	(host cell index, mother cell index)		

Table 1: Legend for (5). Values are observed at timepoint t if it is a time dependent variable.

domain [6], they disappear in the model. To simplify notation we let $X_i := C_i^{(X,Y)}(t)$ and $\mathcal{F}$ denotes the constant spring stiffness. Then the most simple case is about just two cells:

$$f_{i,j} = \begin{cases} \mathcal{F}(2R_C - \|X_i - X_j\|)\frac{X_i - X_j}{\|X_i - X_j\|} \ if \ \|X_i - X_j\| < 2R_C, \\ 0 \ otherwise. \end{cases} \tag{4}$$

But nearly always there are more cells involved. So let N be the set of all indices where the associated cells are in the neighbourhood of X_i, then the resulting force F_i acting on X_i results out of simple vector addition of the forces betwen X_i and X_j $\forall j \in N \setminus i$.

$$F_i = \sum_{j \in N} f_{i,j}.$$

The whole mechanic scheme can be seen as a system of overdamped springs connecting the neighbouring cells. The Newtonian equations control the cells dynamics. This leads to a representation of the force with a damping constant v and a location tracking within a small interval Δt:

$$F_i = -v\frac{dX_i}{dt} \ \ and \ \ X_i(t + \Delta t) = X_i(t) - \frac{1}{v}\Delta t F_i.$$

The tumor would achieve a stable state the moment when the neighbourhood of every cell is empty. According to equation (4) every sub forces equal to zero and consequently the resulting force equals zero, too.

2.4.2 Model dynamics

$$C_k(t) = \left\{ C_k^{(X,Y)}(t), C_k^{age}(t), C_k^{mat}, C_k^{\xi}(t), C_k^{\gamma}(t), C_k^{exp}(t), C_k^{dam}(t), C_k^{death}(t), C_k^{(ID_c,ID_m)} \right\}. \tag{5}$$

In equation (5) you see all the attributes/values of the cell with index k at timepoint t. Speaking in terms of "sensed" oxygen or drug, the neighbourhood of C_k is defined as

[6] nucleus once crosses the domain border

n_k[7]. In the following the update cycle of a cell (see figure 5) is described and the update of the cell attributes is defined. This cycle is repeated for every cell in every timestep in a random order as the agents are computed asynchronous and no cell should have advantage/disadvantage by positioning.

First step is to determine the oxygen and drug uptake. Thereby the oxygen level is equal to the oxygen uptake because all of it is used in every iteration. This simply leads to the summation over the oxygen concentrations in the neighbourhood n_k. The drug uptake is slightly more complicated. First one ensures that the drug does not decline. Then for every location x in the neighbourhood n_k, the uptake is the difference between the maximal possible uptake and the drug decay. If the uptake sum overall x leads to a negative uptake this is set to zero in order to hold the steady rising drug level condition.

$$C_k^{\xi}(t+\Delta t) = \underbrace{\sum_x \xi(x,t)}_{sensed\ \&\ used}, \quad C_k^{\gamma}(t+\Delta t) = C_k^{\gamma}(t) + \left[max\left(0, \sum_x \underbrace{min(\gamma(x,t), \rho_\gamma)}_{uptake} - \underbrace{d_\gamma C_k^{\gamma}(t)}_{decay}\right)\right] \Delta t.$$

These two values decide the further course. The C^{ξ} regulates if the cell oxygenation falls under a threshold and consequently changes into a quiscent state which inhibits cell division. The change of the C^{γ} value in combination with the duration of drug exposure determines the DNA damage. This is calculated under consideration of the repair rate as follows:

$$C_k^{dam}(t+\Delta t) = C_k^{dam}(t) + \left[max\left(0, \sum_x min(\gamma(x,t), \rho_\gamma) - d_\gamma C_k^{\gamma}(t)\right)\right] \Delta t - \underbrace{p C_k^{dam}(t)}_{repair}. \quad (6)$$

If the DNA damage exceeds a certain fixed death threshold (C^{death}) the cell dies. In the case of pre-existing resistance the death threshold of some cells is higher by a factor compared to the others, but this value cannot change over time. In the case of acquired resistance this value is not fixed. Under given conditions the threshold can be computed as follows: [8]

$$C_k^{exp}(t+\Delta t) = \begin{cases} C_k^{exp}(t) + \Delta t \ if\ C_k^{\gamma}(t) > \gamma_{exp}, \\ C_k^{deat}(t)\ otherwise. \end{cases}$$

[7] $n_k = \left\{ x : \|x - C_k^{(X,Y)}\| < R_C \right\}$

[8] t_{exp}: minimal time of high drug exposure for more resistance, γ_{exp}: barrier of high drug exposure

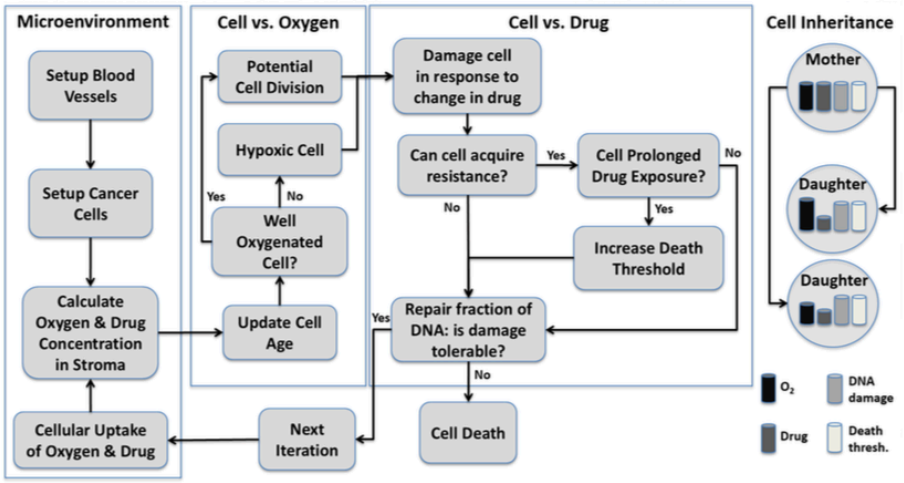


Figure 5: Update cycle for a cell [7, p.8]

$$C_k^{death}(t+\Delta t) = \begin{cases} C_k^{death}(t) + \Delta_{death} \; if \; C_k^{exp}(t) > t_{exp}, \\ 0 \; otherwise. \end{cases}$$

The cell age is updated respectively by the timestep if it does not die. So there is a last case to discuss, the cell division. This happens if the cell reached its mature age, it is well oxygenated and not overcrowded. The latter means that only a certain amount of cells can be located in a surrounding of the potentially dividing cell. If these conditions are fulfilled, the cell will undergo mitosis and will place one daughter cell at the location of the mother cell and the other at a random angle around the mother cell. At this point the cell mechanics start working to stabilize the tumor. In this proliferation process the mother passes the DNA damage and the death threshold but naturally the amount of drug is bisected.

Some last words about the initial configuration of the cells which is set as follows:

$$C_k(t_0) = \left\{ (X_k, Y_k), 0, M_k, \sum_k \xi(x, t_0), 0, 0, 0, T_k, (k, 0) \right\}$$

Going through the values, the cells have a predefined location (X_k, Y_k) and start with age zero. The maturation age M_k is drawn from a uniform distribution $[0.5 \times Age, 1.5 \times Age]$ with Age being the average maturation age. The initial sensed oxygen level is set to the cell neighbourhood of the stable gradient and all drug concerning attributes are set to zero. All cells start with the same death threshold T_k and the unique cell index is set to $(k, 0)$ as the mothers of the initial cells are unknown.

2.5 Comparison with an evolutionary model

The model we will analyse has been introduced in detail, now follows a short overview over another approach dealing with the problem of drug resistance. A model dealing with evolutionary dynamics of cancer in response to targeted combination therapy is presented in [6]. Unlike than in WhAM the model neglects the spatial structure and environmental influence factors and focuses on the tumor size and the treatment with more than one drug (to target different pathways). It is a multitype branching process in continuous time based on empirical obtained probabilities. In comparison to the WhAM model, which focuses on empirical values (e.g. biological data), probabilities (e.g. the order the agent update is executed) and fixed cell rules, the evolutionary model only stands on probabilities. For the sake of convenience the cross-resistance phenomenon and the loosing of resistance by mutation is neglected.

We shortly summarize the major results for a dual treatment. Comparison with the WhAM model will be neglected as this does not focus on therapy with more than one agent. Even though in medical practise for economic or medical reasons a sequential therapy is used, the model precludes any chance of cure even in the case that there is no possible mutation that confers cross resistance. One would assume that the simultaneous therapy would deliver a promising outcome but only if there is no mutation the dual simultaneous therapy will eradicate the tumor. Even if there is only one possible mutation conferring cross resistance there is nearly no chance for tumor eradication. If still this unlikely eradication happens, the remission is only shortly living. Furthermore simulations with three therapeutic agents led to the same result.

Model

The possible combinations of resistance in the system with D different ordered drugs are modelled with a binary string of length D. 0 stands for sensitivity and 1 for resistance. So let $m \leq D$ define the quantity of resistant drugs for a cell and let $D_r = \{i_1, ..., i_m : 1 \leq i_1 \leq ... \leq i_m \leq D\}$ be the set of the indexing drugs a cell is resistant to (but not to the other $D - m$). Then for all sets $s \in P(D \setminus D_r)$ the n_s describes the number of point mutations that have the potential to confer resistance to all drugs in s and u is the point mutation rate. A point mutation is a single nucleotide base change, insertion, or deletion of the genetic material and the point mutation rate is a measure for the frequency how often a point mutation occurs over time. Point mutations happen during the cell division and the model assumes that the resistant mutation occurs in one daughter cell and the other stays in the same state as the mother cell. If both daughter cells should have the possibility of mutation, in the model the mutation factor simply is doubled.

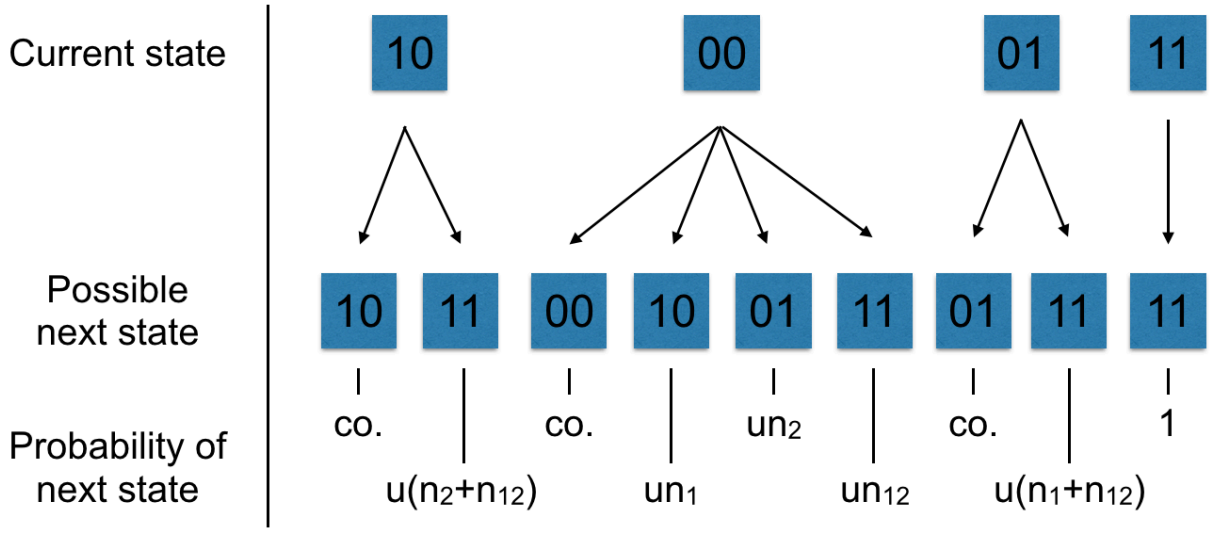


Figure 6: Illustration for the 4 possible resistance state proceedings in a model concerning two different drugs. co. describes the complement of 1 and the sum of all the other probabilities of the possible next state for a current state.

The model contains two stages. Stage 1 is the pretreatment where the model is initialized with only sensitive cells ($s = (0, ..., 0)$) in place and the tumor expands at rate $r = b - d$[9]. The second stage starts the treatment when the cells reached a number of M. In this stage fully resistant cells (those with resistance profile (1,...,1)) continue with rates b respectively d, all the other get new rates $r' = b' - d' < 0$. The paper continues with analysing the different branches, especially concerning the existence possibilities for a total multi resistant clone (resistance profile (1,...,1)) which is equal to the treatment failure.

3 Analysis and results

Treatment (or also without pressure) leads to the development of new cell features, for example resistance or the possibility of spreading by metastatic behaviour. This may result in differentiated cells with new phenotypes. Ultimately, this causes a heterogeneous cell population and consequently difficulties in the therapy. The phenotype diversity causes different responses of the subpopulations to the chemotherapeutic pressure. Our aim is to understand the multiple cell lineages and to analyse how the surviving subpopulations should have been treated. [1]

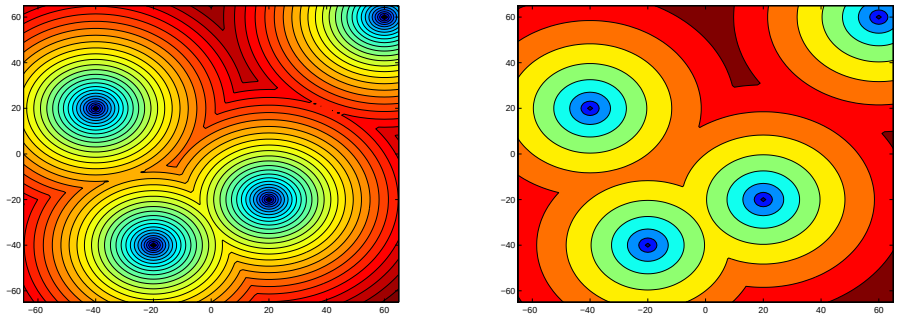
The aim of this work is to develop a routine analysing the output of WhAM to generate

[9] b defines the cell division rate and d the death rate

lineage trees. This hierarchical structure is often used in biology and medicine to encode cell division events over time. The function generates among others the complete lineage trees of the tumor initiating cells (TIC). Every single branch of a tree describes one specific lineage. In contrast to the structures displaying the complete tumor growth, the partition into the single TIC for illustration and analyse gives the possibility to distinguish between the several subpopulations (phenotypes) and their behaviour. Particularly, in this work, the advantage of analysing the lineage trees is to draw spatial conclusions why this specific clone behaved as simulated. To compare cell attributes one can define a domain partition. The key aspect is to partition the tissue space in spatial levels describing areas with similar supply of oxygen and drug to analyse the impact of niches or far distances to vessels on the clonal evolution. There are of course more possible distance functions assigning to each location in the tissue a value for the just mentioned purpose. So let d be the euclidean norm in two dimensions, $X = (X, Y)$ an arbitrary location in the tissue and let $V = \{v_1, ...v_4\}$ be the set of the locations of the four vessels. Then the used distance function $dist$ is as follows:

$$dist : \underbrace{[-65, 65] \times [-65, 65]}_{tissue\ \ space} \ \rightarrow \ \mathbb{R}, \qquad X \ \mapsto \ \min_{v \in V} \left(\sqrt{d(X, v)} \right). \tag{7}$$

The reason why the square root of the Euclidean distances (or the minimal distance) between a point and the vessels are taken is because in a healthy tissue the oxygen/drug firstly spreads in an area of a circle, consequently the drug/oxygen is distributed quadratic concerning the radius. From now on the term distance describes the outcome of $dist$ for the position of a cell in the tissue. This holds in the text as well as in the illustrations. This distribution of the distance function is illustrated in figure 7. As the maximal value that can result out of the $dist$ function (this value is taken as the tissue is represented with a compact set) is 7.9774, for the analyse the domain is partitioned in eight different intervals respectively defining the eight categories cells are ordered to. Except of the last interval, all of them evenly cover 1 distance unit (meaning [0,1) refers to category 1, [1,2) refers to category 2 and so on). The higher is the distance category (depends on in which interval the distance value (equation (7)) is located) of a cell, the lower is the drug/oxygen uptake and so the cell is located in a spatial niche. The higher the total population around the vessels, the lower is the category number already touched by the niche phenomenon as the drug/oxygen is broadly used by the cells near the vessels.



(a) Distance distribution

(b) Used distance levels

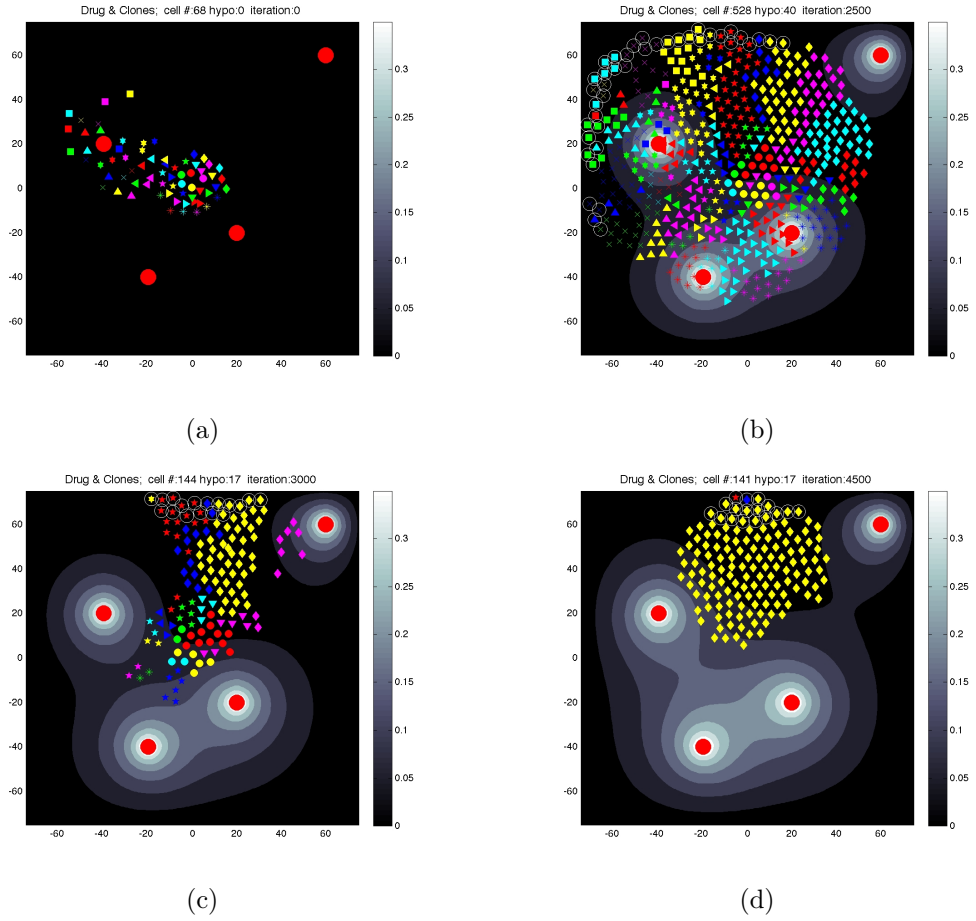
Figure 7: The figures are generated with the use of the implementation of function *dist* (see appendix). (a) Graphical illustration of the *dist* function. (*dist*(30)) (b) The used levels of distance (in the sense of equation (7)) in this paper to categorize cells. It is a partition of the domain in nearly 8 equidistant intervals. (*dist*(8))

3.1 No drug resistance

In order to analyse the behaviour of the tumor evolution in resistant systems during the therapy with a chemotherapeutic agent one also has to document the case of no resistance. Otherwise no conclusions can be drawn of the impact of the different types of resistance on the clonal evolution with anti-cancer medication. Therefore WhAM was run in the no resistance mode and with a DNA repair rate of 0.015% (same repair rate in all cases). The simulation in this and all other cases is considered until iteration 20000.

In figure 8 you can see shots of the tumor tissue at certain time points. In figure 8(a) you can see the distribution after the first pass of the loop. So in the beginning the tumor grows steady with many different clones (see illustration 8(b)) until the cells near the vessels are killed and the cells in the drug niche for now survive (compare figure 8(c)). The tumor is diminished further until one strong clone (clone 25) arises from the drug niche and starts spreading which can be seen in figure 8(d). But in the end this clone dies out as well because there is no resistance to the drug which is equal to the complete tumor eradication (iteration 6607).

At the beginning the tumor can grow quickly because it is well oxygenated (see figure 5: the cell has to be non-hypoxic in order to proliferate, in the WhAM tissue plots hypoxic cells are represented by a white circle around them) and drug has not reached all cells. The latter takes more time in average if the tumor, like in this case, grows fast because the drug is distributed over more and more cells. The drug is indeed halved but as the rise of damage depends on the change in drug and the current DNA damage there is no gain from the drug halving (compare equation (6)). In figure 9(a) we can see that the



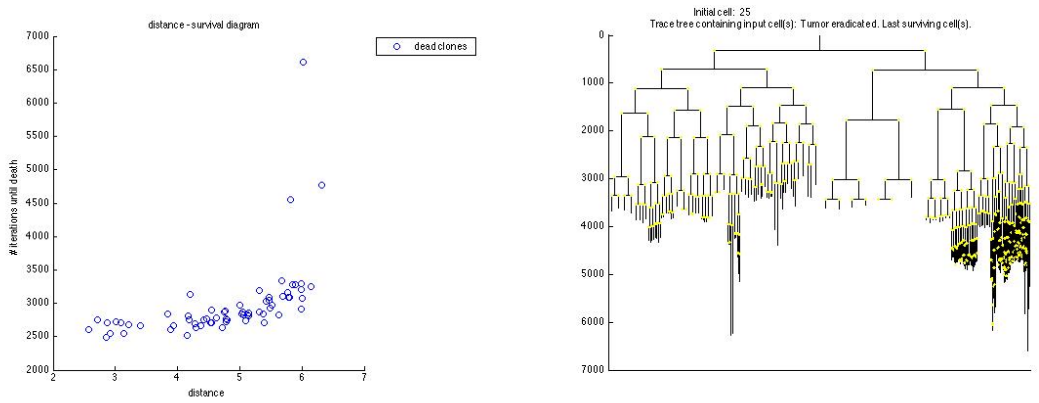
(a) (b)

(c) (d)

Figure 8: No resistance: Illustration of the tumor evolution.

first clones are dying out around iteration 2500. In this plot there is a noticeable trend. The higher the distance to the nearest vessel, the more likely the clone survives longer. The initially surviving of the cells in the niche refers to the less drug uptake but why the tumor can recover and grow again and why are they eradicated later? The reason is the diminished number of cells. This leads to a further diffusion of the drug because it is not used by cells in between. As a result the area of low drug supply shrinks and the incremental term of the DNA damage equation essentially overpowers the repair term.

Now have a closer look on the clone 25 as the longest surviving. Its lineage tree (or trace tree) can be seen in figure 9(b) (and also its symbol in the other plots). There may be three reasons why this clone survived that long. The first is the optimal initial position concerning the complete tissue In figure 8(a) we can see that the initial the cell 25 is placed right in the middle of the niche between the four vessels. This is important for the primary surviving of the cell and for the growth of the clone. The second reason is the protection of the clone by the clones represented via the blue and pink diamonds in



(a) (b)

Figure 9: No resistance: (a) Illustration of the relationship between the distance of the initial cell (in the sense of equation 7) of a clone and the extinction of the complete clone. (b) Lineage tree for the initial cell 25. It is the longest surviving before tumor eradication.

the WhAM tissue images 8. They broadly absorbed the drug before it could invade in the niche and reach clone 25. The last point is coincidence. The new cells and consequently the movement of the tumor by the cell mechanics lead cell 25 completely in the niche. Afterwards, approximately from iteration 3000 on, the clone strongly increased which can be seen in the trace tree. This is a result of the cease of the other clones so there is much less overcrowding. Nevertheless this only prolongs the survival of clone 25. The cell protection is lost, the drug is invading in the niche and the number of cells is not rising fast enough.

3.2 Pre-existing resistance

As already addressed in section 2.4, the pre-existing resistance in WhAM is modelled so that two of the 65 initial clones have a higher death threshold (five times higher) than the others. This ability is inherited and during the treatment the death threshold cannot be heightened. Moreover there are three possible initial configurations (near, intermediate and far away from the vessels) where the resistant clones are located at. Since in all the three simulated initial configurations the same progress and result was observed, only the case of the resistant cells near the vessels is presented.

At first all the clones evolve like in the case of no resistance. They spread and remain in the niche until they die except for the two resistant clones. Those do not response to the drug treatment and eventually populate the complete tissue space. We present a similar analysis as in the case of no resistance, both in terms of resistant and non-resistant cells evolution.

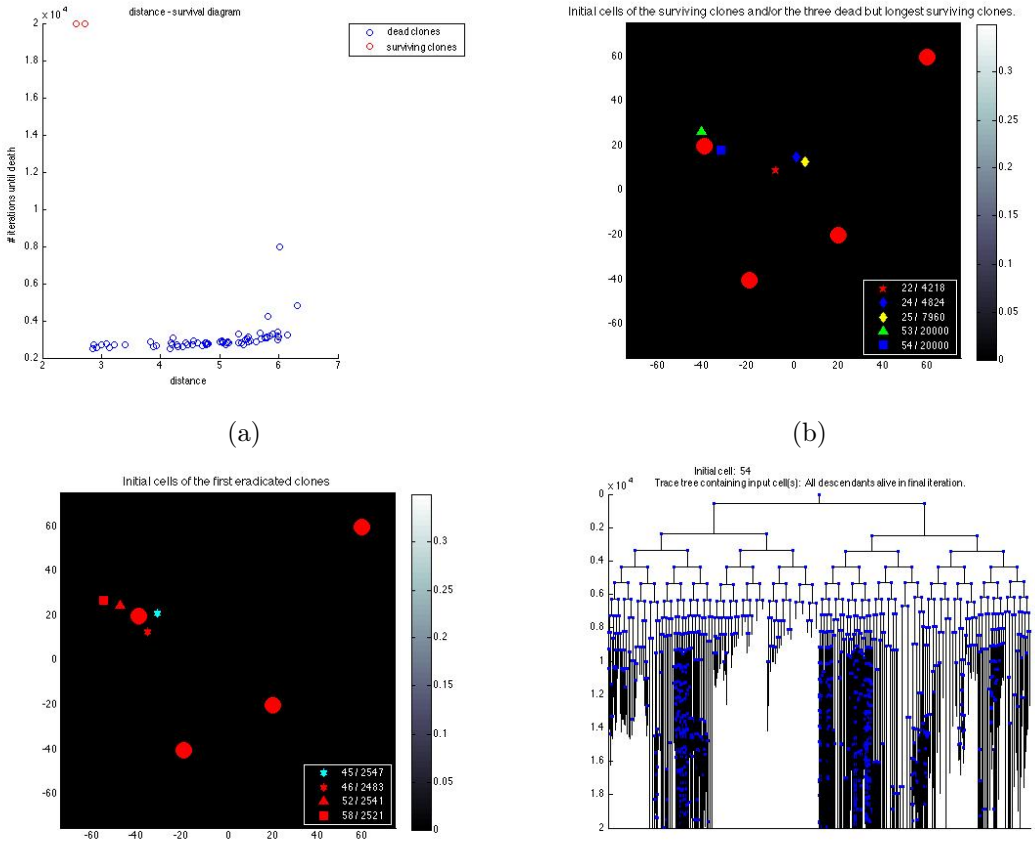


Figure 10: Pre-existing resistance: (a) Illustration of the relationship between the distance of the initial cell (in the sense of equation (7)) of a clone and the extinction of the complete clone. (b) Location of the initial cells of the long surviving clones. (c) Location of the initial cells of the short living clones. (d) Trace tree of resistant cell 54.

The same picture of the relationship between the distance to the nearest vessel and the clonal survival is observed. In figure 10(a) one can see the exact same trend as in the non-resistant case but the two resistant clones (those with the lowest distance to the vessels) survive until the end of the simulation. In the tissue space one can see in figure 10(b) the location of the longest surviving clones and in 10(c) the shortest surviving clones. The clones dying first are located at the vessel (see figure 8(a)). The trace tree of one of the resistant surviving clones, namely clone 54, can be seen in. 10(d). The first sign of resistance was the rather constant iteration (except generation 1 due to display problems) until proliferation and no cell death until about iteration 8000. This speaks for the division right after the maturation age is reached. Afterwards the cell division gets irregular and also some cells "disappear". The term is set in quotes because most of the cells do not die but are not considered in the trace tree any more because the cells

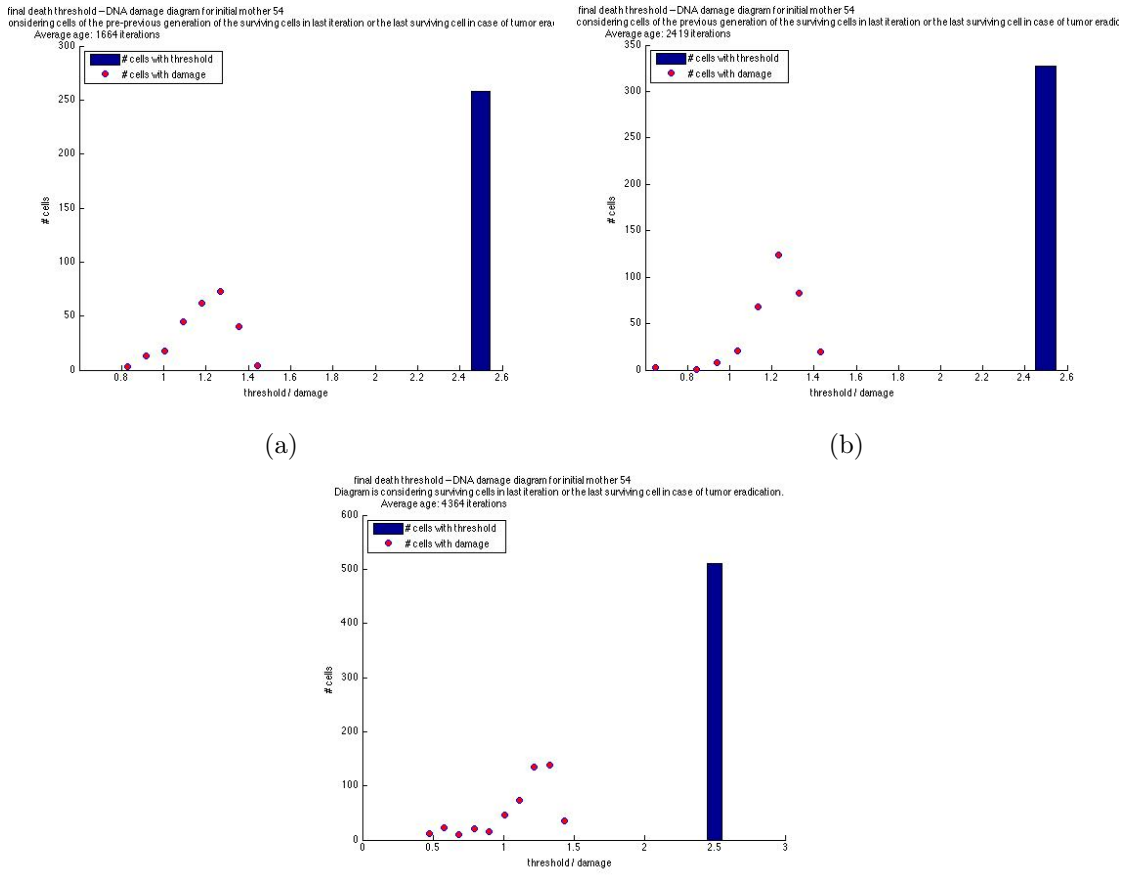


Figure 11: Pre-existing resistance: Digram plotting the number of cells in the DNA damage levels against the death threshold levels of the cells in the last three generations of cell 54. (a) pre-previous iteration; average cell age: 1664 iterations. (b) previous iteration; average cell age: 2419 iterations. (c) generation in last iteration; average cell age: 4364 iterations.

are pushed out of the watched domain. In the introduction of WhAM this was equal to the death of the cell. This explains the ending of some lineages. There are two possible explanations why the reproduction cycle concerning the number of iterations has become unrhythmic. Both of them are related to the rapidly growing population. On the one hand at the margin of the tissue the cells slip in a quiescent state as result of the low oxygen supply (according to figure 5 this blocks the potential cell division), on the other hand the overcrowding of the space. In figure 11 the distribution of the cells according to their DNA damage level in final iterations is presented together with two ancestor generations. So for example in figure 11(b) are considered only the ancestor cells of the cells alive in final iteration. Speaking of the generations in time we see a rising generation age and also a small shift of the damage level towards the death threshold. But even if

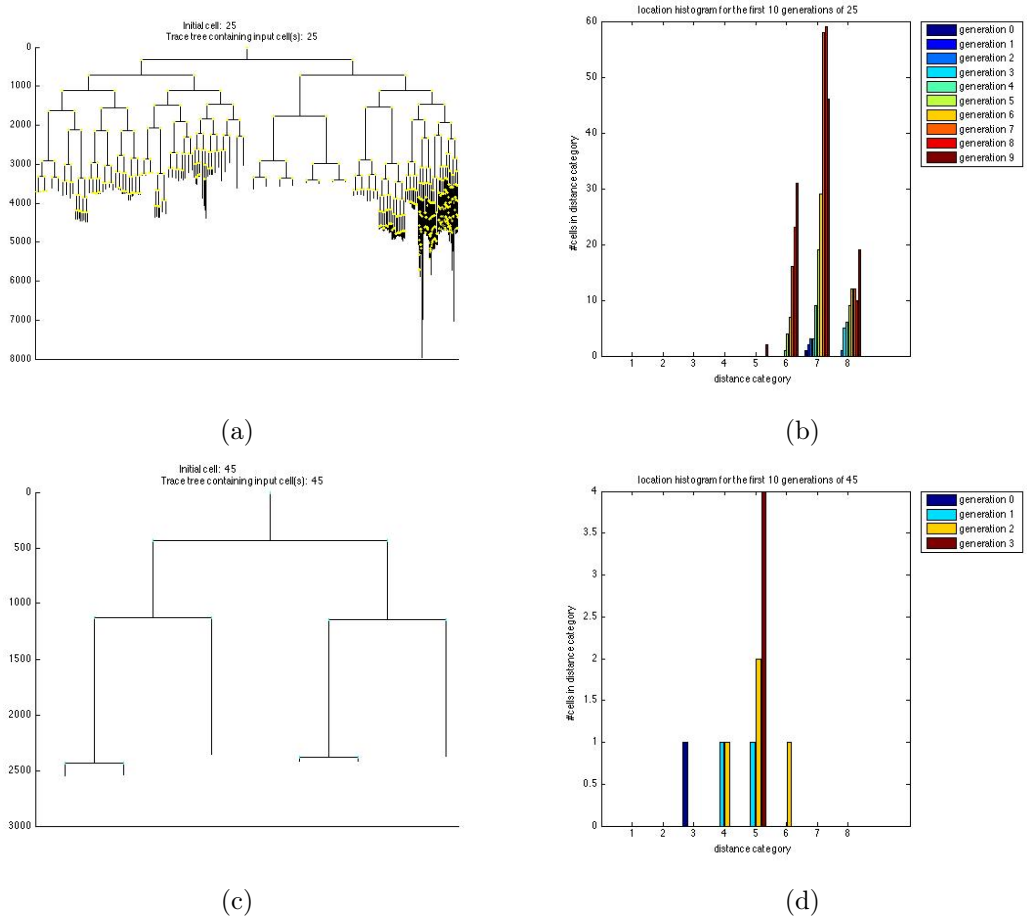


Figure 12: Pre-existing resistance: (a) Lineage tree of longest surviving clone without resistance, cell 25. (b) Histogram of the first 10 generations of cell 25 according their participation in the distance levels 1-8 presented in figure 7(b). (c) Lineage tree of cell 45. (d) Generation distance level histogram for cell 45.

the DNA damage comes near the death threshold level or even exceeds it, the dying cells would be immediately replaced by new cells. As a result of the high population number, there will be cells which are slightly or not at all reached by the drug and so cells with a low DNA damage will be generated. Consequently the pre-existing resistance results in a complete treatment failure.

Now comparing this resistant tree to two non resistant clones, the longest non resistant surviving (clone 25) and a short living (clone 45). In figure 12 the lineage trees are illustrated. Initially focus on the less complex tree of cell 45 (figure 12(c)). The cell starts dividing but all the cells of the clone die nearly at the same time. This shows that the cells of the clone got the same amount of drug and because of no resistance they died very early. Now the more interesting lineage tree of 25 (figure 12(a)). Here in the beginning a regular reproduction is observed but earlier than the resistant clone 54 (about iteration

1000) this order disappears. This results because clone 25 was in a spatial niche and also the surrounding clones are long living (see figure 10(b)). The low drug supply in the niche generates a low death rate and leads to an early overcrowding. Until there is more space again, the process of cell division is inhibited. In the early iterations of tree 54 this behaviour is not observable. As the initial cell is placed near the vessels and the non resistant clones around it die very early (figure 10(c)) space is free and the dividing is continued. The first cells of clone 25 are dying from iteration 3000 which is identical to the non resistant case. This fact as well as nearly the complete lineage tree is so similar to the non resistant case because clone 25 never came into contact with the resistant ones and died before they spread over the tissue.

Additionally to the previous observations in subsection 3.1 of clone 25 and clone 45 we want to point out the spatial "movement" of the generations. For clone 54 this consideration is not that interesting, because of the resistance, the clone survives anyway. The diagram plots for the first 10 generations (if it survives that long) of the spatial distribution of the generation in 8 distance levels (according to figure 7(b)) is shown in figures 12(b) and 12(d). In the simple case of clone 45, as a representative of fast dying clones, we see in figure 12(d) that the initial cell is located relatively near the vessels, namely in distance level 3. In the following two generations the cells move in higher distance levels (4,5,6) but this was not fast enough and the clones could not escape the high drug supply. The last generation is located in level 5 and the clone perishes there. Now comparing to figure 12(b) which displays the same plot for cell 25. As we already know it starts in the niche (level 7) and in the early beginning (since generation 2) already populates the highest distance level 8 and never really reaches level 5 or lower. As one can see the cell explosion after generation 5.

3.3 Acquired resistance

Acquired resistance, the second possible kind of resistance implemented in WhAM, was modelled as follows. The death threshold level (the DNA damage a cell can manage) is not like in the other case fixed, but a variable (once gained resistance cannot be lost for example by another mutation). The rise is attached to the condition that the exposure time with drug higher than a level exceeds a certain duration. For the sake of convenience the step the threshold rises is at all times and for all cells the same constant Δ_{death}. Now we discuss two cases, differing in the amount of threshold gained per step, as they deliver completely different results.

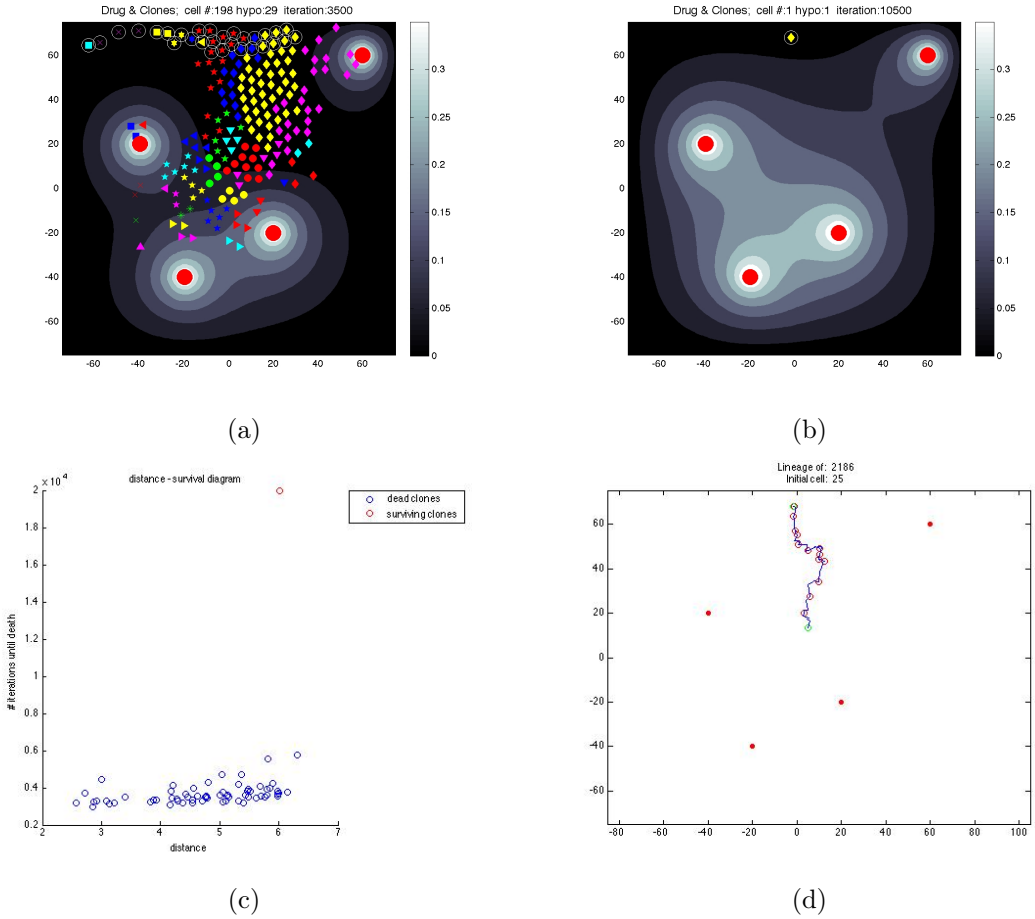
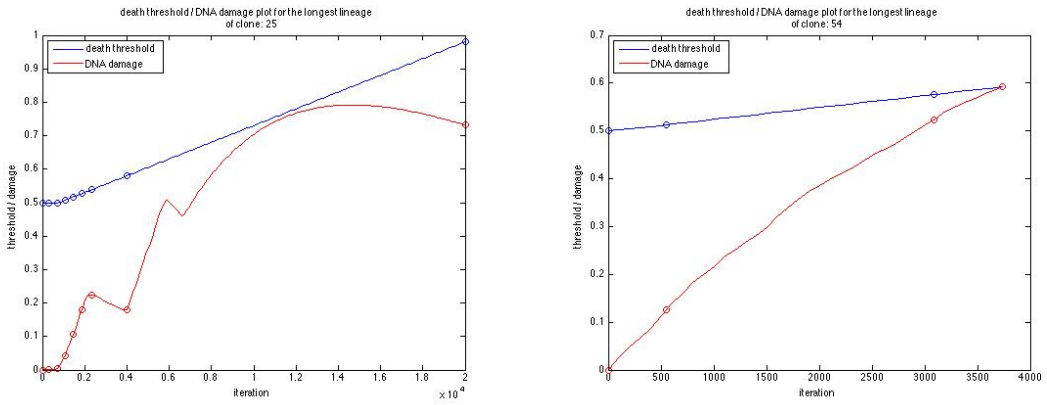


Figure 13: Acquired resistance case 1: (a),(b) Tissue image at iteration 3500 respectively iteration 10500. (c) Illustration of the relationship between the distance of the initial cell (in the sense of equation (7)) of a clone and the extinction of the complete clone. (d) Spatial movement of the longest lineage of 25. The first green circle stands for the beginning location, every further red circle is the position after 500 iterations later. The second green circle stands for the end position.

Case 1: $\Delta_{death} = 0.000025$

Comparing the figure 13(a) to the illustration 8(c) in case of no resistance the timepoint when the first wave of death kicks in delays about 500 iterations. Occasional some cells survive also in the drug exposed regions at first. But gradually the same course is pursued which means that step by step all but one clone dies out, clone 25 recovers temporarily but in the end also is diminished further and further. The main difference so far is the timescale which is prolonged. This also reflects in the distance-survival plot (see illustration 13(c)). The survival trend concerning the distance (in terms of equation (7)) and also the level of first deaths nearly stays the same as in the non-resistant case, but there is a wider distribution regarding the surviving iterations. In the simulation the clone



(a) (b)

Figure 14: Acquired resistance case 1: (a) Death threshold plotted against DNA damage for the longest surviving lineage of clone 25. (b) Death threshold plotted against DNA damage for the longest surviving lineage of clone 54.

25 survives until the end but only with one hypoxic cell which is located in the highest distance class and consequently deeply in a drug niche. In figure 13(b) you already can see the final tissue overview already at iteration 10500. Because in this class of resistance the death threshold additionally varies to the DNA damage it is interesting to have a look at the relationship between them for some certain lineages. Maybe this also could give clues why clone 25 has not died so far.

In figure 14 the information is shown for the surviving lineage of 25 and the longest surviving lineage of a dying cell 54. The circles in the plot stand for the initial cell at the beginning, a proliferation within or death at the end of the cell lineage. The graph is generated out of the given data from WhAM. In the used simulation the data was saved to portrait the current state of the tissue and cells every 100 iterations. Between these data points every 100 iterations the graph is linearised. This accuracy is absolute adequate for this reflection, if not the simulation has to be done with more saving steps. Naturally this leads to a worse runtime of the simulation and the analysis. First discuss the plot for the clone 54 (see figure 14(b)) as a representative for a clone with a initial position near the vessel. As a result of this, the clone (and so the cells of the considered lineage) from the beginning and the complete lifetime is under high drug exposure. This leads to a sustained death threshold increase right from the beginning. The needed drug exposure level of 0.01 is overrun right away and at a high level, so that halving after proliferation does not decrease the level. This increase is not fast enough to escape death and so the lineage and the clone dies. Interesting is that the slope of the DNA damage graph slowly falls within time in 14(b). The earliest decrease after the first proliferation

is related to the rising number of cells and so less drug can be sensed by the lineage. The second decrease after the second proliferation is the result of a greater minimal distance to vessels, although there are less cells around the considered cell. This also is an indicator that location (respectively distance to the nearest vessels) is the most important factor for drug supply and DNA damage. In the end, clone 54 had an advantage over other cells as it is located slightly further for longer survival, this is because the instant resistance gain was superior to the faster DNA damage which led to slightly longer survival.

Now have a look at the longest lineage of clone 25 (see illustration 14(a)). In the early time the death threshold curve as well as the DNA damage curve do not join. This is justified by the fact, that the drug needs time to diffuse to the cells in the niche. Important to notice is the earlier (but minimal) rise of the DNA damage because the drug exposure must be high and long enough to develop resistance. After the drug has accumulated strong enough, the lineage over the whole time linearly acquires resistance. In the early segment also a frequent proliferation can be seen accompanied by a strongly rising DNA damage. Both stops after iteration 2000 where the DNA damage further declines. The cell is surrounded by other cells/clones and so shielded from the drug. On one side, the cell cannot divide due to overcrowding, on the other side the fraction of repaired DNA damage is greater than the gained drug in this period of time. Remarkable is that the drug input is not zero, in this scenario the DNA damage would fall stronger. This trend finds an abrupt end (near iteration 4000) when the first death wave clears the tissue around the niches. The protective shield is destroyed and immediately the cell can divide as there is enough space and the maturation age of course already was reached. The death wave of the other clones leads to more drug input which instantly leads to an increase of DNA damage. This same process can be seen later around iteration 6000 but only with cells of clone 25 as all the others are eradicated already. This second decline stops at after 500 iterations for the simple reason that there are not enough cells as a shield. At this point the cell already reached its final position which can be seen in the plot of the spatial movement of the watched longest cell line (see figure 13(d)). Here it is also clear that the longest cell line walked through the entire niche in the most distant area. The cell from now acts in a hypoxic state which prohibits the proliferation. After this timepoint one indirectly sees a stabilisation of the drug gradient. At iteration 11000, the drug uptake nearly was sufficient enough to kill the cell and at iteration 14000 the fraction of repaired DNA damage was higher than the sensed and used drug in the process of stabilization. At the end of the process the drug gradient will have approximated the same state as if there would be no cells in the tissue (see illustration 4) and not more drug will arrive to the cell because of the drug decay (metabolism). The cell is going to regenerate to a

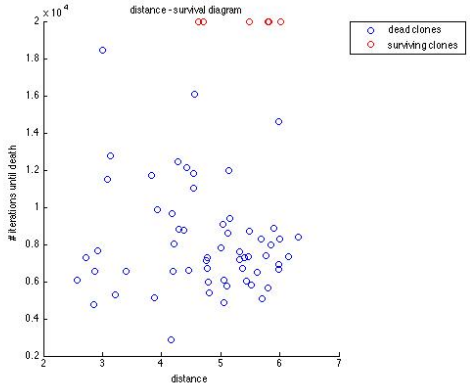


Figure 15: Acquired resistance case 2: Illustration of the relationship between the distance of the initial cell (in the sense of equation (7)) of a clone and the extinction of the complete clone.

level that the repaired fraction of the current damage is equal to the constant sensed drug uptake.

As only one cancer cell is left, it appears that the treatment was successful (remission) but as soon as more oxygen in this tissue area would be supplied a very highly resistant clone will spread and lead to a very dangerous recrudescence.

But how could the last cell have been eradicated? First idea is to diminish the resistance gain. This would only be possible by reducing the drug supply to a level that the cells would not even get in danger to die. This would lead to a worse result. The second option is a higher drug supply after the other clones were eradicated. This would lead to a further diffusion (if the decay rate would stay the same) and so the drug uptake (and consequently the DNA damage) would rise and the death threshold would be exceeded for a successful treatment. In order to protect the organism, a high dosage would only be used at the end for a short period of time. This short high dose treatment for example would be sufficient in iteration 10500 as the cell nearly died.

Again as in the other cases three factors were responsible for the survival: initial position, protection by surrounding cells and the spatial niche.

Case 2: $\Delta_{death} = 0.000059$

As the heading of the paragraph already notes we discuss in this segment the case of acquired resistance but with a higher amount of resistance gained in each step. In this mode of resistance we know that the initial distance (see equation (7)) does not play the same role as in the other modes (see 15). Nevertheless all the last surviving clones started in the upper half of the complete distance interval. Explicit in 16(c) and 16(d) one can

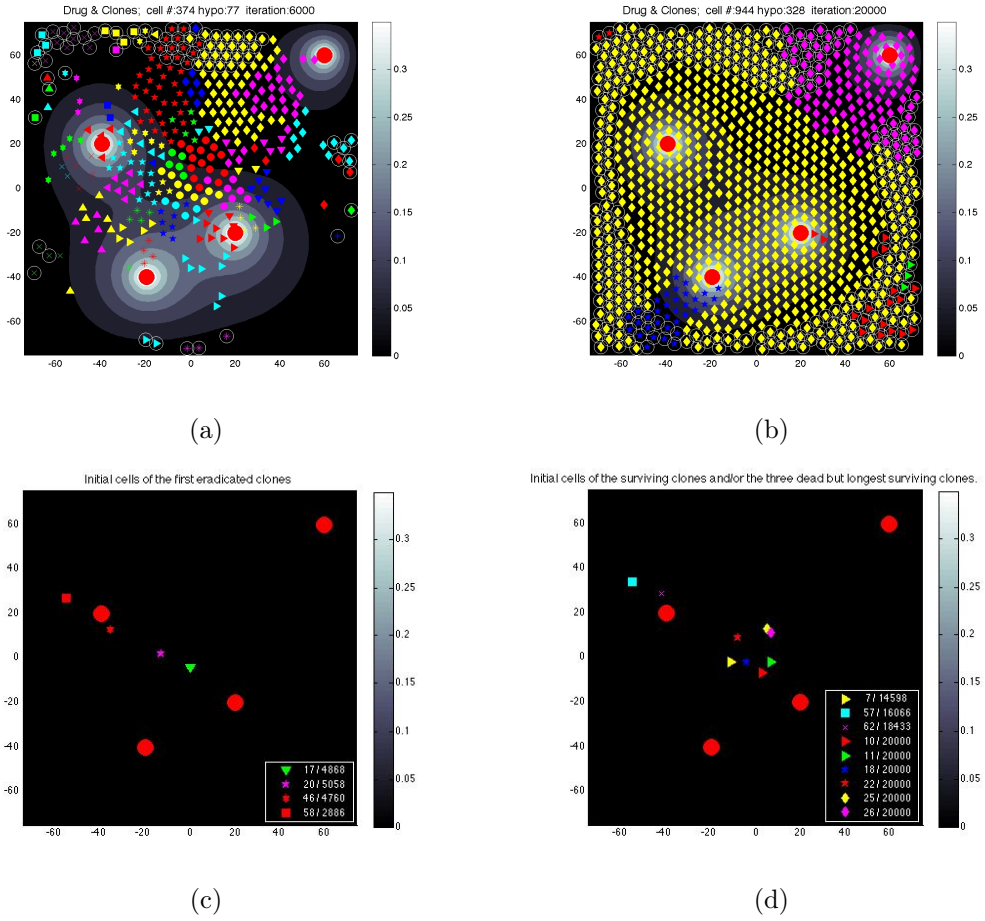


Figure 16: Acquired resistance case 2: Tissue condition over time.

have a look at the first dying and the longest surviving clones. Here one sees how both close and far distant initial clones die in the early stage of the simulation. Of course the first deaths still survived longer than in all other modes. The same in the case of long surviving clones whereas all the final living clones started around the niche. In illustration 16(a) we can see the results of the first stronger death wave. This could not strike as in the other cases as a result of the stronger death threshold gain. In the further course a very strong heterogenization took place which lead to the final picture of the simulation in iteration 20000 in figure 16(b).

At first the longest lineage of clone 46 (see illustration 17(a)) is compared to that in case 1 with longest lineage of clone 54 (see figure 14(b)). The clones nearly have the same initial position and both hardly moved in the tissue. This results in a similar DNA damage and death threshold graph. Here one can nicely see that the stronger death threshold gain resulted in a longer survival of about 1000 iterations. The second presented lineage (and also the longest lineage of the clone) also shows more clones but are "killed" from the

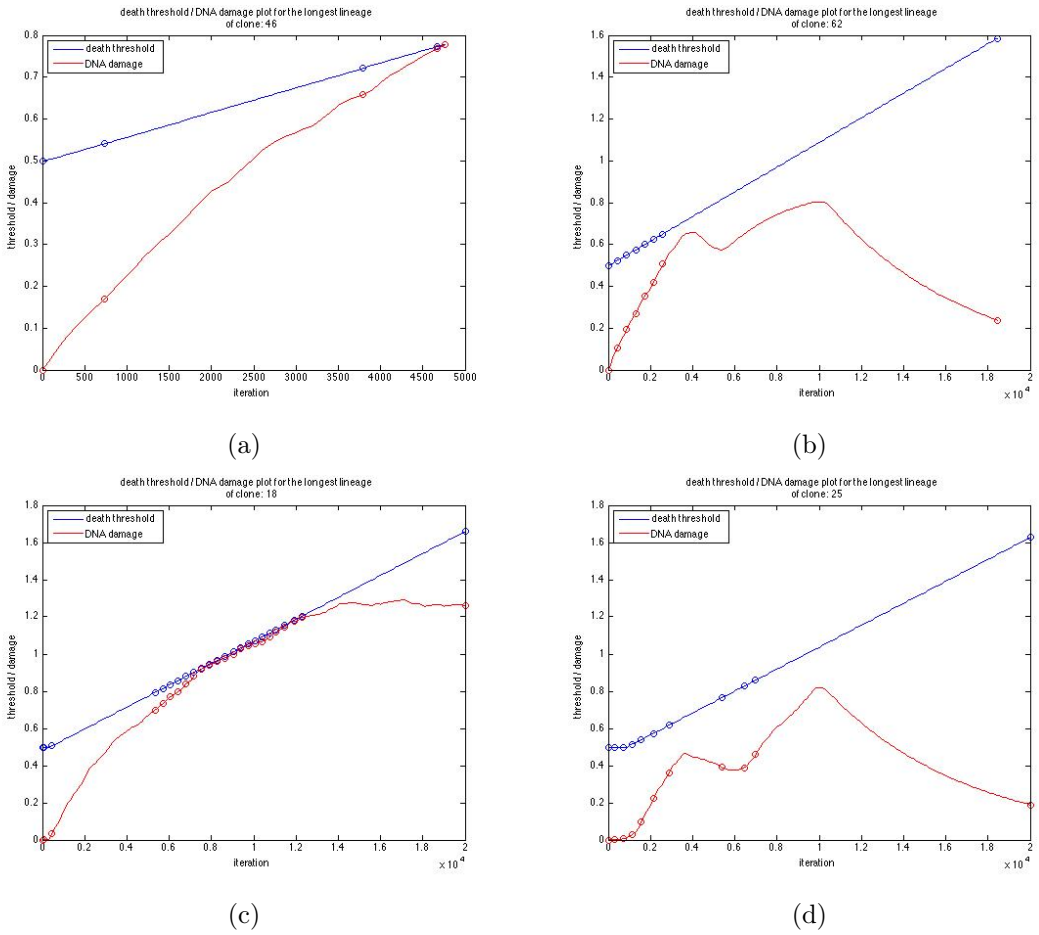
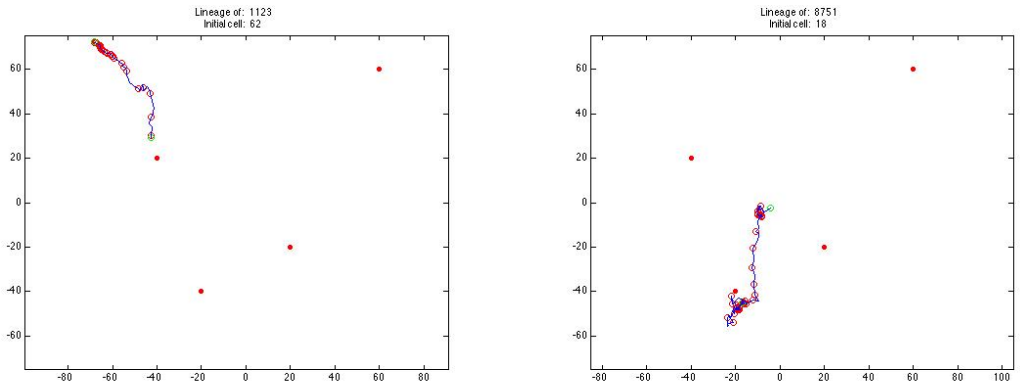


Figure 17: Acquired resistance case 2: Death threshold plotted against the DNA damage graph for lineages for one longest (or the longest) lineage of the clone.

simulation. Therefore from figure 17(b) and 18(a) one can view that the DNA damage is far away from the threshold level. The reason can be found in the movement plot of the lineage. There you can see that the lineage is slightly pushed out of the watched domain and consequently is no longer considered in the simulation. This happens not for a single clone but several, which means the tumor is much more aggressive from what the simulation shows.

Now an example for a weakly surviving clone can be found in illustration 17(c). We neglect the early behaviour as this was already described with a sample clone before. The interesting part is the period of the impending death between iteration 6000 to 12000. In this period the cell is located in the wild area south of the vessel in the spatial plot 18(b). The lineage "fights" for survival and finally escapes the almost sure death because the highly proliferative clone 25 pushes just a little bit further from the vessel. This was sufficient enough to stop the rise of the DNA damage and at last generates a location



(a) (b)

Figure 18: Acquired resistance case 2: Movement of explicit lineages of clone 62 respectively clone 18.

where the DNA damage and the repair is balanced. The reason why this lineage survived is the high proliferation rate, so the drug is partitioned every time in more clones. This rapid division bisects every time the drug level but never deceeds the level for death threshold gain. The constant reproduction of the cells of clone 18 in the late phase delays the process of killing by the drug until clone 25 is saved. In this critical phase hardly a lineage survived. Just consider the respective trace tree in figure 19(a). Little before iteration 6000 it is clear that nearly all lineages die out. As the cells are not scattered, they nearly have the same DNA damage - death threshold plots but in the critical phase most (especially the edging and less distant) were not affected.

Now let us consider the strongest clone in the tissue, clone 25. As in all the other simulations this clone invades the niche and from then on completely spreads through the tissue. In figure 19(c) one can see between iteration 6000 to 14000 a population explosion. From then on the proliferation rate is much less. Only if the tissue is not overcrowded due to the death of other clones, mechanical movement or when scattered cells die, the cells divide to fill up the tissue again. In figure 17(d) one can see the diagram for a lineage completely moving through the niche and staying in the highest distance level. One can recognize the often observed "wave phenomenon" due to overcrowded protection and the drug niche. The lineage never was in danger of dying because of how its death threshold advance against the DNA damage gained in the niche before population spread.

We now present a comparison between the location of the first 10 generations for clone 18 and 25 (histograms see figure 19). Both clones start at a similar position, 6 respectively 7. We already see for the strongest surviving clone little advantage. The most important proliferations are the first ones, they strongly decide in which direction the clone will

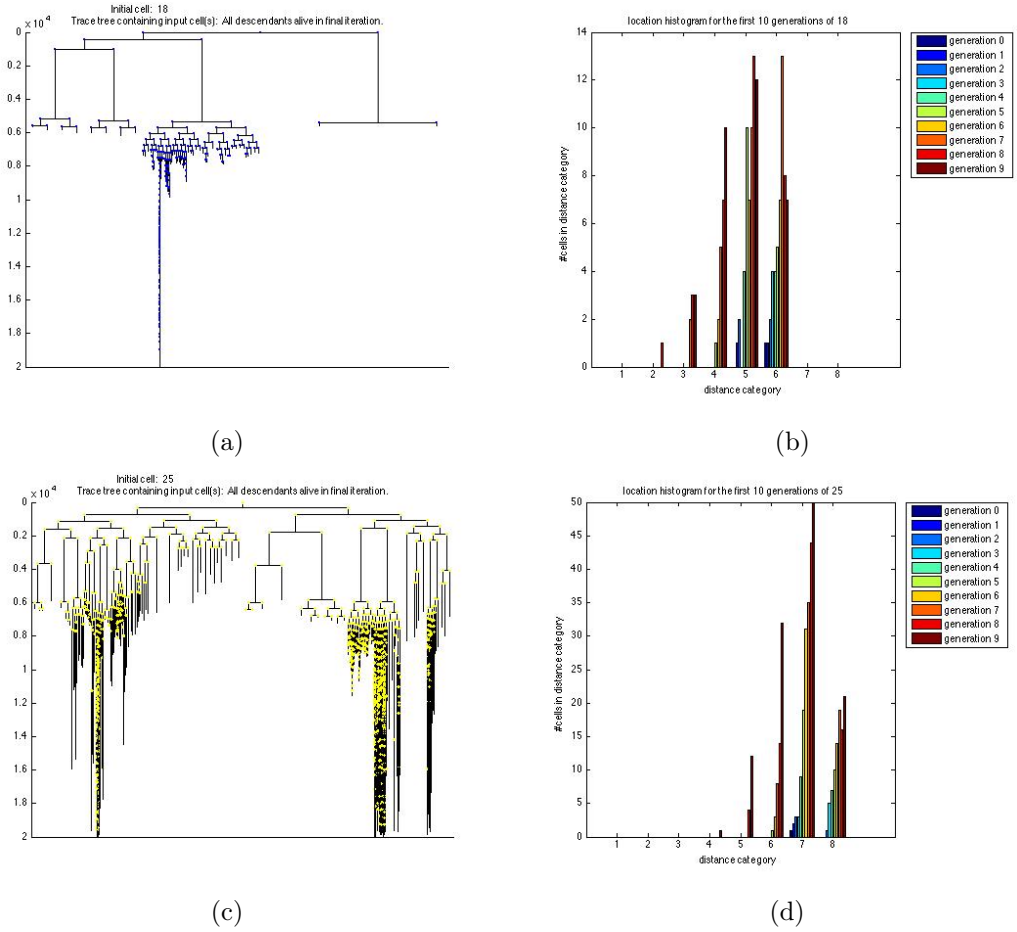


Figure 19: Acquired resistance case 2: (a) Lineage tree of clone 18. (b) Histogram of the first 10 generations of clone 18 according their participation in the distance levels 1-8 presented in figure 7(b). (c) Lineage tree of clone 25. (d) Histogram of the first 10 generations of cell 25 according their participation in the distance levels 1-8 presented in figure 7(b).

evolve. As we see in figure 19(d) already in generation 1 clone 25 settles the highest distance level and not until generation 5 levels lower than 7 are colonized. As we would extend this histogram for all generations, all the levels would be touched by the strongly dominating clone. In the weakly surviving clone 18 we see in figure 19(b) the exact contrary movement in the early stadium. Already in generation 1 the movement towards the vessels begins with the advance in level 5 and in generation 5 in level 4. In the higher levels (5-6) since iteration 8 the number of cells decline. If we would further protocol these histograms over generations we would observe a further shift of the histogram to the lower left side.

3.4 Discussion

In this work clonal and lineage analysis were performed on a modified output of model WhAM. The idea was to challenge that mostly all the cells of a clone behave the same way. Simulations were performed and evaluated. All of them differ in the combination between the mode of resistance, the DNA damage repair rate, the initial positioning in case of pre-existing resistance and the gain of death threshold in case of acquired resistance. In the presented variations all possible final outcomes occurred. The successful treatment in case of no resistance, the nearly successful treatment in case of weak acquired resistance and the treatment failure in the pre-existing and strong acquired resistance case.

We found that clone 25 is quite strong, indeed we compared its lineage trees in the different cases. The trees can be partitioned in three segments. In a first one all trees are nearly identical in the early phase. In the mid phase they differ concerning the proliferation rate or survival time. The end phase is the most different. Graphically this means more dense cell points/lineages, but in general the proportions of the trees preserve in the different cases. The first segment is mainly controlled from the initial positioning as in every case the cells can tolerate some DNA damage. Since this is always the same, the early stage results in nearly identical trees. The second segment is driven by ability. The ability (the type of resistance) additionally to the early positioning leads to a different survival in the most critical mid phase. In the late phase we only recognize the manifestation of the actions in the critical phase. Important is that this is not a clone 25 exclusive observation.

A strong focus was the local positioning of the initial clones with respect to the clonal evolution. The outcome showed different key points depending on the mode of resistance. In the pre-existing case, the position of the initial clone plays no role. All the clones except of the two resistant ones behave the same way as in the case of no resistance. So the local movements and positioning do not affect the result because in the no resistance case all clones died. The opposite occurs in the acquired resistance case. Initial positioning and movement is important for the survival and the strength of the surviving clones. The death threshold gain plays a key role.

Two general statements that at more opportunities came up. The first one is the more likely survival of high proliferative clones. To point out is that this is not owed to the drug halving during division. It does not matter how high the drug level is already. The second annotation is, that in all the observed lineages in the case of pre-existing resistance, once the death threshold rise started it never stops. This is founded on the fact that the needed drug level is so minimal and even in the niche more drug arrives than the halving could get rid of.

4 Prospect

In this thesis lineage trees were generated based on model WhAM and these trees were analysed. The quantity of trees one can compare is strongly limited and secondly human activity is error-prone. Another problem is the displaying of complex trees, one would need really large surfaces to spot details. In order to solve problems it would be great to transfer the analysing of lineage trees on a machine routine. This would give the possibility to compare much more and complex lineage trees without mistakes (if the routine is implemented correctly) and thereby can recognize patterns. The main difficulty is to develop a proceeding to compare lineage trees. One possible solution is presented in [12] by introducing a metric and in the end remains one ratio classifying the similarity/difference between two lineage trees. We now describe briefly the method developed in that paper.

First step was to define the space of labelled but unordered trees ($\mathcal{T}$) for the algorithm. Therefore an unordered but labelled tree $T = (V, E, \nu, \sigma, \Sigma)$ was defined whereas V *and* E describe the nodes (cells) respectively, the directed edges. The function $\nu : V \to P_2(V)$ allocates every node its descendants. Σ is the set of all possible labels and $\sigma : V \to \Sigma$ assigns every node its label (cell attributes). Second was to define a metric $MaxSimilarity$ (computation of maximal similarity common subtree) on $\mathcal{T}$ to obtain a metric space $(\mathcal{T}, MaxSimilarity)$. In the paper they experimented with different types of metrics. After underlying conditions are formalized the tree clustering algorithm is reviewed by using a k-mean/k-median algorithm. The aim of this algorithm is dividing a dataset $A = \{a_1, ..., a_n\}$ into non-empty and disjoint subsets B_i ($i \in \{1, ..., k\}$) with $\cup_i B_i = A$. Accompanying to the subsets there is a set of centroids c_i. This partition is done by minimizing the squared sum of the distances between the $a_j \in B_i$ and its belonging centroid c_i.

This strategy could be transferred to lineage trees produced here. One could for example define the centroids as trees with a different long survival or different high proliferation rate and then let the algorithm work on the set of all initial cell trees. What we get as a result are clusters with the same properties and so afterwards conclusions can be drawn for example if the spatial structure is the reason for the resulting development of cells or the type of resistance or both. A key role also would be that the conclusions we draw from the lineage trees are more representative as a result of the much higher amount of analysed data.

In the paper and also in the analysis of WhAM in [7] so far was only dealt with a continuous drug supply of one single therapeutic agent. We saw that in nearly all cases that the treatment with the drug develops resistance and gives rise to a resistant

subpopulation and the tumor cannot be eradicated. In section 2.5 a model approach was presented trying to avoid these problems by the usage of more than one drug with different pathways, so if the cells gain resistance to one drug the other can still kill the cells. The results of this model were not promising either. Once there is one single possibility for a mutation generating cross-resistance to all drugs, there is nearly no chance for the treatment success. Beside the number of different drugs the second possibility of change is the time treatment protocol. The model WhAM allows different drug treatment protocols with respect to time and dosage of the drug. This can be seen in equation (3) because the drug supply rate is time dependent. Who is further interested in this kind of modification of WhAM should have a look at [9].

References

[1] Stuart K. Calderwood. Tumor heterogeneity, clonal evolution, and therapy resistance: An opportunity for multitargeting therapy. *Discor Med., 15(82): 188-194*, 2014.

[2] Stefan Pfenninger Dieter M. Imboden. *Introduction to Systems Analysis: Mathematically Modeling Natural Systems.* Springer, 2013.

[3] Sarah Heerboth-Karolina Lapinska Mckenna Longacre-Nicole Snyder Genevieve Housman, Shannon Byler and Sibaji Sarkar. Drug resistance in cancer: An overview. *cancers, 6(3): 1769 - 1792*, 2014.

[4] Terna Gilbert. How to build and usa agents-based models in social science. *Mind and Society, 1:57-72*, 2000.

[5] Michael M. Gottesman. Mechanisms of cancer drug resistance. *Annual Reviews, 53:615-627*, 2002.

[6] B. Allen-T. Antal K. Chatterjee-P. Shah Y. S. Moon A. Yaqubie N. Kelly D.T. Le E.J. Lipson P.B. Chapman L.A. Diaz B. Vogelstein M.A. Nowak I. Bozic, J.G. Reiter. Evolutionary dynamics of cancer in response to targeted combination therapy. *eLife, 2:1-15*, 2013.

[7] Kerri-Ann Norton Judith Perez-Velazquez Alexandria Volkening Jana L. Gevertz, Zahra Aminzare and Katarzyna A. Rejniak. Emergence of anti-cancer drug resistance: Exploring the importance of the microenvironmental niche via a spatial model. 2012.

[8] Wooldrige Jennings, Sycara. *A Roadmap of Agent Research and Development (Autonomous Agents and Multi- Agent Systems).* Kluwer Academic Publishers, 1998.

[9] Olga Khrutska. Exploring the anti-cancer drug resistance via an agent-based hybrid model. Master's thesis, Technische Universität München, 2015.

[10] J.Mazoyer M. Delorme, editor. *Cellular Automata A Parallel Model.* Springer Science+Business Media Dordrecht, 1999.

[11] S. Mukherjee. The emperor of all maladies: a biography of cancer. *New York: Scribner*, 2012.

[12] Nina Hubig-Claudia Plant Carsten Marr-Michael A. Rieger Timm Schroeder Fabian J. Theis Valeriy Khakhutskyy, Michael Schwarzfischer. Centroid clustering of cellular lineage trees. *Information Technology in Bio- and Medical Informatics, p.15-29*, September 2014.

[13] Daniel Weber. Circumvention of multi-drug resistance of cancer cells by chinese herbal medicines. *Chin Med. 2010; 5: 26*, 2010.

[14] Dieter A. Wold-Gladrow. *Lattice - Gas, Cellular Automata and Lattice Boltzman Models*. Springer, 2000.

[15] Hiba Zahreddine and Katherine L. B. Borden. Mechanisms and insights into drug resistance in cancer. *Frontiers in Pharmacology, 4:28*, 2013.

5 Appendix

This section contains the generated MATLAB code for the analysis of the lineage trees. Therefore also the model WhAM was modified by the introduction of two new data storage matrices containing the history of important attributes by cell which are saved in the end. The following functions work on this new data and the anyway existing output of model WhAM. Also the code of the function generating the illustration for the distance levels/gradient is added.

```
function draw_cell_trace_my_version_2( cells, pathname, fig_mod, ...
    fig_mod_2, save_mod, text_mod, doc_mod )

% Function for analysing the output of the modified model WhAM
% ("modelWhAM_updated_my_version.m") regarding the clonal evolution. The
% function delivers the lineage trees containing the input cells and
% generate further data of interest.
%
% Input arguments:
%  - cells      1xn vector with cell ID's of the cells to be tracked
%               If a empty vector is given the surviving clones are used.
%               default     []      All cells alive in final iteration.
%  - pathname   path of the output folder of
%               "modelWhAM_udated_my_version.m"
%  - fig_mod    0       every tree has its own figure
%               1       one figure containing all trees
%               default     0
%  - fig_mod_2  0       not
%               1       tree edges have same color as in model
%               default 1
%  - save_mod   in case of fig_mod = 0
%               0       normal
%               1       x - axis stretched
%               2       y - axis stretched
%               3       grand
%               default     0
%  - text_mod   0       off (text and colored input values)
%               1       all cellID's
%               2       cellID's, threshold, damage of the last branch cell
%               3       both 1 and 2
%               default     0
%  - doc_mod    document showing data for last three generations
```

```
33 %                and overview for important inital cell values
34 %                0       off
35 %                1       on
36 %                default     1
37
38
39 % set defaults
40 if( ~exist('fig_mod','var'))
41     fig_mod     = 0;
42 end
43 if( ~exist('fig_mod_2','var'))
44     fig_mod_2     = 1;
45 end
46 if( ~exist('save_mod','var'))
47     save_mod    = 0;
48 end
49 if( ~exist('text_mod','var'))
50     text_mod    = 0;
51 end
52 if( ~exist('doc_mod','var'))
53     doc_mod    = 1;
54 end
55 if( ~exist('doc_mod_2','var'))
56     doc_mod    = 1;
57 end
58
59
60 % cells a row vector
61 if( ~isrow(cells) )
62     cells = cells';
63 end
64
65
66 % get constants
67 ncc             =   load([pathname,'/number_cancer_cells.txt']);
68 Niter           =   size(ncc,2)-1;
69 paramInt        =   load([pathname, '/paramInt.txt']);
70 num_init_cells  =   paramInt(5);
71
72
73 % loading files
74 bdh             =   load([pathname, '/birth_death_history.txt']);
75 tdh             =   load([pathname, '/threshold_damage_history.txt']);
```

```
76 cells_mother    =   load([pathname, '/cellsMotherID_', num2str(Niter), ...
77                         '.txt']);
78
79 % cells default
80 which_cells = 0;            % information for naming the plot
81 if( isempty(cells))
82     % cells alive in last iteration
83     cells = find(bdh(3,:) == 1);
84     which_cells = 1;
85     if( isempty(cells))
86         % tumor is eradicated: last surviving cell(s)
87         cells = find(bdh(2,:) == Niter);
88         which_cells = 2;
89     end
90 end
91
92
93 % get initial mother for every cell ID
94 init_mothers        = find_init_mothers( cells, cells_mother );
95 init_mothers_unique = sort(unique(init_mothers));
96
97
98 % main
99 if(fig_mod == 1)
100     m = floor(sqrt(size(init_mothers_unique,2)));
101     n = ceil(sqrt(size(init_mothers_unique,2)));
102     subplot(m,n,1);
103
104     for i = 1:size(init_mothers_unique,2)
105         if(which_cells == 0)
106             help_vec = cells(init_mothers_unique(i) == init_mothers);
107             help_vec_str = sprintf('%.0f,' , help_vec);
108             help_vec_str = help_vec_str(1:end-1);   % strip final comma
109         elseif(which_cells == 1)
110             help_vec_str = 'All descendants alive in final iteration.';
111         else
112             help_vec_str = 'Tumor eradicated. Last surviving cell(s).';
113         end
114         help_string = ['Initial cell:  ', ...
115             num2str(init_mothers_unique(i)), ...
116             '\n Trace tree containing input cell(s):  ', help_vec_str];
117
118         subplot(m,n,i)
```

```
119         title(sprintf(help_string))
120         hold on
121 
122         [data, pointer_storage] = tree_gen(init_mothers_unique(i), ...
123             bdh , cells_mother , Niter);
124         plot_tree(data, pointer_storage, text_mod,tdh, fig_mod_2, ...
125             init_mothers_unique(i));
126     end
127     clear help_string;
128     clear help_vec;
129     clear help_vec_str;
130 
131     path_result = save_data_1(pathname);
132 
133     % create document
134     if(doc_mod == 1)
135         doc_gen(path_result, init_mothers_unique, bdh, tdh, ...
136             cells_mother, Niter);
137         DrawTissue(bdh, path_result,Niter, pathname);
138     end
139 else
140     for i = 1:size(init_mothers_unique,2)
141         if(which_cells == 0)
142             help_vec = cells(init_mothers_unique(i) == init_mothers);
143             help_vec_str = sprintf('%.0f,' , help_vec);
144             help_vec_str = help_vec_str(1:end-1);   % strip final comma
145         elseif(which_cells == 1)
146             help_vec_str = 'All descendants alive in final iteration.';
147         else
148             help_vec_str = 'Tumor eradicated. Last surviving cell(s).';
149         end
150         help_string = ['Initial cell:  ', ...
151             num2str(init_mothers_unique(i)), ...
152             '\n Trace tree containing input cell(s):  ', help_vec_str];
153 
154         figure_storage(init_mothers_unique(i)) = figure;
155         title(sprintf(help_string))
156         hold on
157         [data, pointer_storage] = tree_gen(init_mothers_unique(i), ...
158             bdh , cells_mother , Niter);
159         plot_tree(data, pointer_storage, text_mod, tdh, fig_mod_2, ...
160             init_mothers_unique(i));
161     end
```

```
162
163     clear help_string;
164     clear help_vec;
165     clear help_vec_str;
166
167     path_result = save_data_2(init_mothers_unique, figure_storage, ...
168         pathname, save_mod);
169
170     % create document
171     if(doc_mod == 1)
172         doc_gen(path_result,init_mothers_unique, bdh, tdh, ...
173             cells_mother, Niter);
174         DrawTissue(bdh, path_result,Niter, pathname);
175     end
176 end
177
178 end     % end main
179
180
181
182 % defined help functions
183
184 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
185 % function creating and saving a document for all trees in order     %
186 % to display data for every tree                                     %
187 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
188 function doc_gen( path_result, init_mothers_unique, bdh, tdh, ...
189     cells_mother, Niter )
190
191     final_data_matrix = [];
192     % open a file for writing
193     fid = fopen([path_result,'/doc_file.txt'], 'w');
194
195     % title
196     s1 = 'Document summing up important data for the cells in last ';
197     s2 = 'generation and its two progenitors in every trace tree.\nThe ';
198     s3 = 'data is formatted by the following pattern:\n\ncell ID\nmother ';
199     s4 = 'cell ID\nage\nfinal threshold\nincrease of threshold\nfinal ';
200     s5 = 'damage\nincrease of damage\n\n';
201     fprintf(fid, [s1,s2,s3,s4,s5]);
202     clear s1
203     clear s2
204     clear s3
```

```
205     clear s4
206     clear s5
207
208     % write data for every initial mother cell
209     for i = init_mothers_unique
210         fprintf(fid, ['Data for trace tree with initial cell ', ...
211             num2str(i), ':\n\n']);
212
213         last_cells =  [];
214         stack = i;
215         while(~isempty(stack))
216             if(bdh(3,stack(1))~=2)
217                 last_cells(end+1) = stack(1);
218                 stack(1)=[];
219             else
220                 stack(end+1:end+2) = [bdh(4,stack(1)), bdh(4,stack(1))+1];
221                 stack(1)=[];
222             end
223         end
224
225         %{
226         % In case you want to have the last cells over all initial mothers
227         last_cells = find(bdh(2,:) == Niter);   % tumor died out
228         if(isempty(last_cells))
229             last_cells = find(bdh(3,:) == 1);   % cells survived
230         end
231         %}
232         for j = last_cells
233             data =      zeros(3,3);
234             data_2 =    zeros(4,3);
235             help = j;
236             for jj = 1:3
237                 if(bdh(2,help) == -1)
238                     data(:,jj) = [help;cells_mother(help); ...
239                         Niter-bdh(1,help)];
240                 else
241                     data(:,jj) = [help;cells_mother(help); ...
242                         bdh(2,help)-bdh(1,help)];
243                 end
244                 data_2(:,jj) = [tdh(1,help);tdh(3,help)-tdh(1,help); ...
245                     tdh(2,help);tdh(4,help)-tdh(2,help)];
246                 help = cells_mother(help);
247             end
```

```
248             fprintf(fid,'%d\t\t\t\t%d\t\t\t\t\t%d\n',data');
249             fprintf(fid,'%.4f\t\t\t\t%.4f\t\t\t\t\t%.4f\n',data_2');
250             fprintf(fid,'\n');
251             final_data_matrix(end+1:end+7,1:3) = [data;data_2];
252         end
253         % seperation between initial mother cells
254         final_data_matrix(end+1:end+7,1:3) = ones(7,3)*-1;
255         fprintf(fid,'\n\n');
256
257
258         % generate and save hist for threshold/damage diagram considering
259         % surviving cells
260         cells_diag_1 = last_cells(bdh(3,last_cells) == 1);
261         if(isempty(cells_diag_1))
262             cells_diag_1 = last_cells(bdh(2,last_cells) == max(bdh(2,last_cells)));
263         end
264         one = sum(Niter - bdh(1,cells_diag_1(bdh(2,cells_diag_1) == -1)));
265         two = sum(bdh(2,cells_diag_1(bdh(2,cells_diag_1) ~= -1)) ...
266             - bdh(1,cells_diag_1(bdh(2,cells_diag_1) ~= -1)));
267         average_age = round((one+two)/numel(cells_diag_1));
268         diagram1 = figure;
269         s1 = 'final death threshold -- DNA damage diagram for initial mother ';
270         s2 = '\nDiagram is considering surviving cells in last iteration or ';
271         s3 = 'the last surviving cell in case of tumor eradication.\nAverage age: ';
272         s4 = ' iterations';
273         title(sprintf([s1, num2str(i),s2,s3, ...
274             num2str(average_age),s4]));
275         clear s1
276         clear s2
277         clear s3
278         clear s4
279         hold on
280         fin_thr = tdh(3,cells_diag_1);
281         fin_dam = tdh(4,cells_diag_1);
282         xval = linspace(min([fin_thr,fin_dam]),max([fin_thr,fin_dam]), 20);
283         hist(fin_thr, xval);
284         plotval = histc(fin_dam, xval);
285         xval = xval(plotval ~= 0);
286         plotval = plotval(plotval ~= 0);
287         scatter(xval,plotval, 'MarkerFaceColor','r');
288         xlabel('threshold / damage');
289         ylabel('# cells');
290         legend('# cells with threshold','# cells with damage', ...
```

```
291             'Location','Northwest');
292         hold off
293 
294         average_thr_dam = zeros(2,3);
295         average_thr_dam(:,1) = [sum(fin_thr),sum(fin_dam)]/numel(fin_thr);
296 
297         cells_diag_2 = unique(cells_mother(cells_diag_1));
298         one = sum(Niter - bdh(1,cells_diag_2(bdh(2,cells_diag_2) == -1)));
299         two = sum(bdh(2,cells_diag_2(bdh(2,cells_diag_2) ~= -1)) ...
300             - bdh(1,cells_diag_2(bdh(2,cells_diag_2) ~= -1)));
301         average_age = round((one+two)/numel(cells_diag_2));
302         diagram2 = figure;
303         s1 = 'final death threshold -- DNA damage diagram for initial mother ';
304         s2 = '\nDiagram is considering cells of the previous generation of ';
305         s3 = 'the surviving cells in last iteration or the last surviving cell ';
306         s4 = 'in case of tumor eradication.\nAverage age: ';
307         s5 = ' iterations';
308         title(sprintf([s1, num2str(i),s2,s3,s4, ...
309             num2str(average_age),s5]));
310         clear s1
311         clear s2
312         clear s3
313         clear s4
314         clear s5
315         hold on
316         fin_thr = tdh(3,cells_diag_2);
317         fin_dam = tdh(4,cells_diag_2);
318         xval = linspace(min([fin_thr,fin_dam]),max([fin_thr,fin_dam]), 20);
319         hist(fin_thr, xval);
320         plotval = histc(fin_dam, xval);
321         xval = xval(plotval ~= 0);
322         plotval = plotval(plotval ~= 0);
323         scatter(xval,plotval, 'MarkerFaceColor','r');
324         xlabel('threshold / damage');
325         ylabel('# cells');
326         legend('# cells with threshold','# cells with damage', ...
327             'Location','Northwest');
328         hold off
329 
330         average_thr_dam(:,2) = [sum(fin_thr),sum(fin_dam)]/numel(fin_thr);
331 
332         cells_diag_3 = unique(cells_mother(cells_diag_2));
333         one = sum(Niter - bdh(1,cells_diag_3(bdh(2,cells_diag_3) == -1)));
```

```
334         two = sum(bdh(2,cells_diag_3(bdh(2,cells_diag_3) ~= -1)) ...
335             - bdh(1,cells_diag_3(bdh(2,cells_diag_3) ~= -1)));
336         average_age = round((one+two)/numel(cells_diag_3));
337         diagram3 = figure;
338         s1 = 'final death threshold -- DNA damage diagram for initial mother ';
339         s2 = '\nDiagram is considering cells of the pre-previous generation ';
340         s3 = 'of the surviving cells in last iteration or the last surviving ';
341         s4 = 'cell in case of tumor eradication.\nAverage age: ';
342         s5 = ' iterations';
343         title(sprintf([s1, num2str(i),s2,s3,s4, ...
344             num2str(average_age),s5]));
345         clear s1
346         clear s2
347         clear s3
348         clear s4
349         clear s5
350         hold on
351         fin_thr = tdh(3,cells_diag_3);
352         fin_dam = tdh(4,cells_diag_3);
353         xval = linspace(min([fin_thr,fin_dam]),max([fin_thr,fin_dam]), 20);
354         hist(fin_thr, xval);
355         plotval = histc(fin_dam, xval);
356         xval = xval(plotval ~= 0);
357         plotval = plotval(plotval ~= 0);
358         scatter(xval,plotval, 'MarkerFaceColor','r');
359         xlabel('threshold / damage');
360         ylabel('# cells');
361         legend('# cells with threshold','# cells with damage', ...
362             'Location','Northwest');
363         hold off
364
365         average_thr_dam(:,3) = [sum(fin_thr),sum(fin_dam)]/numel(fin_thr);
366
367         diagram4 = figure;
368         s1 = 'Average threshold/damage of the last three generations for ';
369         s2 = 'surviving cells.';
370         title([s1,s2]);
371         clear s1
372         clear s2
373         hold on
374         xval = 1:3;
375         plot(xval,average_thr_dam(1,3:-1:1),'r');
376         plot(xval,average_thr_dam(2,3:-1:1),'b');
```

```
377         legend('avg threshold','avg damage');
378         set(gca, 'XTick',1:3, 'XTickLabel', ...
379             {'pre-previous','previous','last'});
380         xlabel('generation');
381         ylabel('threshold / damage');
382         hold off
383 
384 
385         fname = [path_result,'/threshold_damage_hist_',num2str(i),'_1'];
386         saveas(diagram1 ,fname,'epsc')
387         saveas(diagram1 ,fname,'fig')
388         fname = [path_result,'/threshold_damage_hist_',num2str(i),'_2'];
389         saveas(diagram2 ,fname,'epsc')
390         saveas(diagram2 ,fname,'fig')
391         fname = [path_result,'/threshold_damage_hist_',num2str(i),'_3'];
392         saveas(diagram3 ,fname,'epsc')
393         saveas(diagram3 ,fname,'fig')
394         fname = [path_result,'/average_generation_threshold_damage_', ...
395             num2str(i)];
396         saveas(diagram4 ,fname,'epsc')
397         saveas(diagram4 ,fname,'fig')
398 
399     end
400 
401     % close the file
402     fclose(fid);
403 
404     % save matrix
405     save([path_result,'/doc_matrix.mat'],'final_data_matrix');
406 
407 end % doc_gen
408 
409 
410 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
411 % function plotting the data generated by tree_gen                        %
412 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
413 function plot_tree( data, pointer_storage, text_mod, tdh, ...
414     fig_mod_2, mother )
415 
416     % determining color/symbol
417     Ncolors=6;
418     colors=[0,0,1;1,1,0;1,0,1;0,1,1;1,0,0;0,1,0];
419     Nsymbols=12;
```

```
420     symbols=['o','>','v','p','d','*','<','h','^','s','x','+'];
421
422     ccol=1+mod(mother,Ncolors);
423     ssym=1+(mother-mod(mother,Ncolors))/Ncolors;
424
425     for cell = unique(data.cellNr)
426         rootpath = [];
427         help_cell = cell;
428         while( help_cell > 0 )
429             rootpath = [help_cell rootpath];
430             help_cell = floor(help_cell/2);
431         end
432         clear help_cell;
433
434         % determining x position
435         x = 0;
436         for i=2:numel(rootpath)
437             step = 1/2^(i-1);
438             if mod(rootpath(i),2)==0
439                 % even number, left branch
440                 x = x - step;
441             else
442                 % odd number, right branch
443                 x = x + step;
444             end
445         end
446
447         % determining y position
448         time = data.timepoint(data.cellNr==cell);
449         ypos = [min(time),max(time)];
450
451         %  plot life current mother
452         line([x x], ypos,'color','k','linewidth',0.1);
453
454         % plot color/symbol
455         if(fig_mod_2 == 1)
456             plot(x,ypos(1),symbols(ssym),'MarkerFaceColor', ...
457                 colors(ccol,1:3), 'MarkerEdgeColor',colors(ccol,1:3), ...
458                 'MarkerSize', 3)
459         end
460
461         % plot connection if there are children
462         c1 = cell*2;
```

```
463         c2 = cell*2+1;
464         if (numel(find(data.cellNr==c1))>0 || ...
465                 numel(find(data.cellNr==c2))>0)
466             x1 = x+(1/2^floor(log2(cell)+1));
467             x2 = x-(1/2^floor(log2(cell)+1));
468             line([x1 x2], [max(time) max(time)],'color', ...
469                 'k','linewidth',0.1);
470 
471             if(text_mod == 1 || text_mod == 3)
472                 text(x2,max(time), sprintf('%d', ...
473                     find(pointer_storage == cell*2)),'VerticalAlignment', ...
474                     'baseline','HorizontalAlignment','left',...
475                     'color','r','FontWeight','light','FontSize',10);
476                 text(x1,max(time), sprintf('%d', ...
477                     find(pointer_storage == cell*2+1)),'VerticalAlignment', ...
478                     'baseline','HorizontalAlignment','right',...
479                     'color','r','FontWeight','light', 'FontSize',10);
480             end
481         else
482             current_cellID = find(pointer_storage == cell);
483             if(text_mod == 2 || text_mod == 3)
484                 s1 = 'cellID: %d \n final threshold: %d \n final damage: ';
485                 s2 = '%d \n threshold increase: %d \n damage incease: %d ';
486                 text(x,max(time) + 10,...
487                 sprintf([s1,s2],...
488                 current_cellID,tdh(3,current_cellID), ...
489                 tdh(4,current_cellID),tdh(3,current_cellID)- ...
490                 tdh(1,current_cellID),tdh(4,current_cellID)- ...
491                 tdh(2,current_cellID)),'EdgeColor','r', ...
492                 'VerticalAlignment','top','HorizontalAlignment','center',...
493                 'color','k','FontWeight','light', 'FontSize',10);
494                 clear s1
495                 clear s2
496             end
497         end
498 
499     end
500 
501     set(gca,'YDir','reverse');  % reverse timescale
502     set(gca,'XTick',[])         % remove x-axis
503 
504 end % plot_tree
505 
```

```
506
507 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
508 % function finding the initial mother cell for a vector of cell ID's %
509 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
510 function init_mothers = find_init_mothers( cells, cells_mother )
511
512     init_mothers = cells;
513     help_cells = cells;
514
515     % init_cells are the mother cells of help_cells
516     init_mothers = cells_mother(init_mothers);
517     while(sum(help_cells == init_mothers) < size(init_mothers,2))
518         help_cells = init_mothers;
519         init_mothers = cells_mother(init_mothers);
520     end
521
522 end % find_init_mothers
523
524
525 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
526 % function saving the data                                                    %
527 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
528 function path_result = save_data_1 ( pathname )
529
530     path_result = [pathname,'/trace_trees_',datestr(now,'HHMMSS')];
531     mkdir(path_result)
532     fname=[path_result, '/trace_trees'];
533
534     % set suitable size
535     number_subplots = size(get(gcf,'Children'),1);
536     m = floor(sqrt(number_subplots));       % #rows
537     n = ceil(sqrt(number_subplots));        % #columns
538     set(gcf, 'PaperUnits', 'inches');
539     x_width=n*8;
540     y_width=m*5;
541     set(gcf, 'PaperPosition', [0 0 x_width y_width]);
542
543     saveas(gcf ,fname,'epsc')
544     saveas(gcf ,fname,'fig')
545
546 end % save_data_1
547
548
```

```
549 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
550 % function saving the data                                                 %
551 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
552 function path_result = save_data_2 ( init_mothers_unique, ...
553     figure_storage, pathname, save_mod )
554 
555     path_result = [pathname,'/trace_trees_',datestr(now,'HHMMSS')];
556     mkdir(path_result)
557     for i=1:size(init_mothers_unique,2)
558 
559         % set suitable size
560         set(figure_storage(init_mothers_unique(i)), 'PaperUnits', 'inches');
561         switch save_mod
562             case 0
563                 set(figure_storage(init_mothers_unique(i)), ...
564                     'PaperPosition', [0 0 8 6]);
565             case 1
566                 set(figure_storage(init_mothers_unique(i)), ...
567                     'PaperPosition', [0 0 40 6]);
568             case 2
569                 set(figure_storage(init_mothers_unique(i)), ...
570                     'PaperPosition', [0 0 8 40]);
571             case 3
572                 set(figure_storage(init_mothers_unique(i)), ...
573                     'PaperPosition', [0 0 40 30]);
574         end
575 
576         fname=[path_result, '/trace_tree_',num2str(init_mothers_unique(i))];
577         saveas(figure_storage(init_mothers_unique(i)),fname,'epsc')
578         saveas(figure_storage(init_mothers_unique(i)),fname,'fig')
579     end
580 
581 end % save_data_2
582 
583 
584 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
585 % function generating data structure for the plot_tree function      %
586 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
587 function [tree, pointer_storage] = tree_gen( mother, bdh , cells_mother , Niter)
588 
589     % stack contains cellsID's you havent't considered so far
590     stack   = mother;
591     pointer_storage = [];       % contains which "plotID" every cellID is
```

```
592     i       = 1;                % pointer
593
594     % initialize tree
595     tree.cellNr     = [];
596     tree.timepoint  = [];
597     tree.identifier = ['trace tree with mother ',num2str(mother)];
598
599
600     while(isempty(stack) == 0)
601         if(bdh(1,stack(1)) == -1)           % #iterations alive
602             num_entries = bdh(2,stack(1))+1;
603         else
604             if(bdh(2,stack(1)) == -1)
605                 num_entries = Niter-bdh(1,stack(1))+1;
606             else
607                 num_entries = bdh(2,stack(1))-bdh(1,stack(1))+1;
608             end
609         end
610
611         pointer_storage(stack(1)) = i;
612
613         cellNr      = ones(1,num_entries)*i;
614         if(bdh(2,stack(1))==-1)
615             timepoint   = Niter-num_entries+1 : Niter;
616         else
617             timepoint   = bdh(2,stack(1))-num_entries+1 : bdh(2,stack(1));
618         end
619         tree.cellNr(end+1:end+num_entries) = cellNr;
620         tree.timepoint(end+1:end+num_entries) = timepoint;
621
622         % add children to stack
623         if(bdh(4,stack(1)) ~= 0)
624             % add children to stack
625             stack(end+1:end+2) = bdh(4,stack(1)):bdh(4,stack(1))+1;
626         end
627
628          % update pointer for next iteration if it isn't the last one
629         if(size(stack,2) > 1)
630             if(cells_mother(stack(1)) == cells_mother(stack(2)))
631                 i = i+1;
632             else
633                 i = pointer_storage(cells_mother(stack(2)))*2;
634             end
```

```
635         end
636
637         stack(1)    = [];           % clear processed cell off stack
638
639     end
640
641 end % tree_gen
642
643
644 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
645 % function drawing initial cells and analysing their clonal survival %
646 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
647 function DrawTissue(bdh,path_result,Niter, pathname)
648
649     % define draw input values
650     cellSize=8;     % size of the cell marker
651
652     xmin=-75;  xmax= 75;
653     ymin=xmin; ymax=xmax;
654     hb              =  2;
655     Source_drug=1;
656     Ngx=1+floor((xmax-xmin)/hb); % number of grid points - x axis
657     Ngy=1+floor((ymax-ymin)/hb); % number of grid points - y axis
658     xgg=xmin:hb:xmax;            % data for drawing
659     ygg=ymin:hb:ymax;            % data for drawing
660     drugDom=zeros(Ngx,Ngy);
661
662     vessel=[-20,-40;-40,20;20,-20;60,60];
663     cell_xy=[0,0;5,5;-5,2;-1,7;-5,6;-10,5;-11,-2;-7,-6;-3,-7;3,-7;7,-2;...
664       11,1;9,6;4,8;2,11;3,-1;0,-4;-4,-2;-9,1;-13,2;-13,7;-8,9;-3,12;...
665       1,15;5,13;7,11;12,9;13,4;15,0;11,-4;6,-8;2,-10;-4,-10;-10,-10;...
666       -14,-3;-11.9366,10.9859;-12.1479,12.4648;-18.9085,4.4366;-18.2746,...
667       13.3099;-22.7113,10.1408;-25.4577,2.3239;-17.0070,21.3380;-22.5000,...
668       17.5352;-30.1056,13.9437;-31.5845,21.1268;-36.0211,12.8873;-41.5141,...
669       10.5634;-37.7113,5.0704;-33.6972,-1.6901;-27.5704,-3.3803;-47.4296,...
670       16.4789;-48.4859,24.9296;-41.5141,26.4085;-32.8521,18.1690;-28.2042,...
671       42.4648;-39.4014,39.0845;-55.4577,33.8028;-55.8803,26.8310;-55.0352,...
672       16.4789;-47.8521,13.3099;-50.1761,30.6338;-42.3592,28.7324;-30.7394,...
673       5.0704;-27.5704,7.8169;-40.4577,0.4225];
674
675     Ncolors=6;
676     colors=[0,0,1;1,1,0;1,0,1;0,1,1;1,0,0;0,1,0];
677     Nsymbols=12;
```

```
678     symbols=['o','>','v','p','d','*','<','h','^','s','x','+'];
679
680     % Determine the initial clones to draw
681     % 1st row:  cellID
682     % 2nd row:  clone death iteration
683     %           if -1 stll alive
684     init_clones_death = find(bdh(1,:)==-1);
685     for i=init_clones_death
686         stack1 = i;
687         stack2 = i;
688         while(isempty(stack1)==0)
689             if(bdh(4,stack1(1))~=0)
690                 stack2(end+1:end+2)=[bdh(4,stack1(1)), bdh(4,stack1(1))+1];
691                 stack1(end+1:end+2)=[bdh(4,stack1(1)), bdh(4,stack1(1))+1];
692             end
693             stack1(1)=[];
694         end
695         if(min(bdh(2,stack2))==-1)
696             init_clones_death(2,i)= -Niter;
697         else
698             init_clones_death(2,i)= max(bdh(2,stack2));
699         end
700         distances = [norm(vessel(1,:)-cell_xy(i,:)),norm(vessel(2,:)- ...
701             cell_xy(i,:)),norm(vessel(3,:)- cell_xy(i,:)),norm(vessel(4,:)- ...
702             cell_xy(i,:))];
703         % item describing if a position lies in a niche
704         init_clones_vessel_distance(i)=min(distances.^(1/2));
705     end
706
707     sort_iterations = unique(sort(init_clones_death(2,:)));
708     last_iterations = sort_iterations(end-2:end);
709     sort_iterations = sort_iterations(sort_iterations>=0);
710     first_iterations = sort_iterations(1:4);
711
712     last_init_cells = init_clones_death(:,...
713         init_clones_death(2,:)==last_iterations(1) | ...
714         init_clones_death(2,:)==last_iterations(2) | ...
715         init_clones_death(2,:)==last_iterations(3) | ...
716         init_clones_death(2,:)==-Niter);
717     last_init_cells = sortrows(last_init_cells',2)';
718     last_init_cells = [last_init_cells(:,end-2:end), ...
719         last_init_cells(:,1:end-3)];
720
```

```
721     first_init_cells = init_clones_death(:,...
722         init_clones_death(2,:)==first_iterations(1) | ...
723         init_clones_death(2,:)==first_iterations(2) | ...
724         init_clones_death(2,:)==first_iterations(3) | ...
725         init_clones_death(2,:)==first_iterations(4));
726 
727     overview_last = figure;
728     axis([xmin,xmax,ymin,ymax])
729     axis equal
730     hold on
731 
732     contourf(xgg,ygg,drugDom',[0:0.05:Source_drug],'edgecolor','none');
733     colormap(bone)
734     caxis([0,0.35*Source_drug])
735     colorbar
736 
737     for z = last_init_cells(1,:)
738         ccol=1+mod(z,Ncolors);
739         ssym=1+(z-mod(z,Ncolors))/Ncolors;
740         plot(cell_xy(z,1), cell_xy(z,2),symbols(ssym), ...
741           'MarkerFaceColor',colors(ccol,1:3),'MarkerEdgeColor', ...
742           colors(ccol,1:3),'MarkerSize',cellSize)
743     end
744     set(gca,'Color','k');
745     h = legend(strcat(num2str(last_init_cells(1,:)'),{' / '}, ...
746         num2str(abs(last_init_cells(2,:)'))),'Location','southeast');
747     set(h,'TextColor', 'w')
748     set(h,'EdgeColor', 'w')
749 
750     % plot vessels
751     plot(vessel(:,1),vessel(:,2),'ro','MarkerFaceColor','r','MarkerSize',...
752          2*cellSize,'LineWidth',2)
753 
754     % title plot
755     s1 = 'Initial cells of the surviving clones and/or the three dead but ';
756     s2 = 'longest surviving clones.';
757     title(sprintf([s1,s2]) ,'FontSize',12)
758     clear s1
759     clear s2
760 
761     saveas(overview_last,[path_result,'/long_initial_cells'],'epsc')
762     saveas(overview_last,[path_result,'/long_initial_cells'],'fig')
763 
```

```
764     overview_first = figure;
765     axis([xmin,xmax,ymin,ymax])
766     axis equal
767     axis([xmin,xmax,ymin,ymax])
768     hold on
769
770     contourf(xgg,ygg,drugDom',[0:0.05:Source_drug],'edgecolor','none');
771     colormap(bone)
772     caxis([0,0.35*Source_drug])
773     colorbar
774
775     for z = first_init_cells(1,:)
776         ccol=1+mod(z,Ncolors);
777         ssym=1+(z-mod(z,Ncolors))/Ncolors;
778         plot(cell_xy(z,1), cell_xy(z,2),symbols(ssym), ...
779           'MarkerFaceColor',colors(ccol,1:3),'MarkerEdgeColor', ...
780           colors(ccol,1:3),'MarkerSize',cellSize)
781     end
782     set(gca,'Color','k');
783     h = legend(strcat(num2str(first_init_cells(1,:)'),{' / '}, ...
784         num2str(first_init_cells(2,:)')),'Location','southeast');
785     set(h,'TextColor', 'w')
786     set(h,'EdgeColor', 'w')
787
788     % plot vessels
789     plot(vessel(:,1),vessel(:,2),'ro','MarkerFaceColor','r','MarkerSize',...
790          2*cellSize,'LineWidth',2)
791
792     % title plot
793     s1 = 'Initial cells of the first eradicated clones';
794     title(sprintf(s1) ,'FontSize',12)
795     clear s1
796
797     saveas(overview_first,[path_result,'/first_initial_cells'],'epsc')
798     saveas(overview_first,[path_result,'/first_initial_cells'],'fig')
799
800     % plot distance - death iteration
801     overview2 = figure;
802     hold on
803     bol = init_clones_death(2,:)<0;
804     scatter(init_clones_vessel_distance(~bol), init_clones_death(2,~bol))
805     scatter(init_clones_vessel_distance(bol), init_clones_death(2,bol).*(-1),'r')
806     ylabel('# iterations until death')
```

```
807     xlabel('distance')
808     title('distance - survival diagram')
809     legend('dead clones','surviving clones','Location','northeastoutside')
810
811     saveas(overview2,[path_result,'/dist_survival'],'epsc')
812     saveas(overview2,[path_result,'/dist_survival'],'fig')
813
814     cell_xy_history = load([pathname,'/cell_xy_history.txt']);
815     % categorizing the cells in 8 distance zones
816     % nearly the value 8 is gained out of the maximum of the distance
817     % function over the complete tissue space
818     bins = 0:1:8;
819     % spatial histograms per clone
820     for i = first_init_cells(1,:)
821         stack1 = i;     % ID of current generation
822         stack2 = [];    % ID of next generation
823         for j = 1:10
824             if(isempty(stack1)==1)
825                 break
826             end
827             d1 = cell_xy_history(stack1,:) - repmat(vessel(1,:),size(stack1,2),1);
828             d2 = cell_xy_history(stack1,:) - repmat(vessel(2,:),size(stack1,2),1);
829             d3 = cell_xy_history(stack1,:) - repmat(vessel(3,:),size(stack1,2),1);
830             d4 = cell_xy_history(stack1,:) - repmat(vessel(4,:),size(stack1,2),1);
831             for z=1:size(d1,1)
832                 d11(z) = norm(d1(z,:));
833                 d22(z) = norm(d2(z,:));
834                 d33(z) = norm(d3(z,:));
835                 d44(z) = norm(d4(z,:));
836             end
837             gen_dist = min([d11;d22;d33;d44]).^(1/2);
838             % columns describe generations
839             bincounts(:,j) = histc(gen_dist,bins);
840             % update neue generation
841             while(isempty(stack1)==0)
842                 if(bdh(4,stack1(1))~=0)
843                     stack2(end+1:end+2)=[bdh(4,stack1(1)),bdh(4,stack1(1))+1];
844                     stack1(1)=[];
845                 else
846                     stack1(1)=[];
847                 end
848             end
849             stack1 = stack2;
```

```
850             stack2 = [];
851         end
852 
853         % plot histogram
854         figure
855         bar(bincounts(1:end-1,:))
856         legend('generation 0','generation 1','generation 2','generation 3', ...
857             'generation 4','generation 5','generation 6', ...
858             'generation 7','generation 8','generation 9', 'Location', ...
859             'northeastoutside')
860         title(['location histogram for the first 10 generations of ', ...
861             num2str(i)])
862         ylabel('#cells in distance category')
863         xlabel('distance category')
864 
865         saveas(gcf,[path_result,'/hist_first_',num2str(i)],'epsc')
866         saveas(gcf,[path_result,'/hist_first_',num2str(i)],'fig')
867         hold off
868 
869         clear d1
870         clear d2
871         clear d3
872         clear d4
873         clear d11
874         clear d22
875         clear d33
876         clear d44
877         clear bincounts
878     end
879 
880     for i = last_init_cells(1,:)
881         stack1 = i;     % ID of current generation
882         stack2 = [];    % ID of next generation
883 
884         for j = 1:10
885             if(isempty(stack1)==1)
886                 break
887             end
888             d1 = cell_xy_history(stack1,:) - repmat(vessel(1,:),size(stack1,2),1);
889             d2 = cell_xy_history(stack1,:) - repmat(vessel(2,:),size(stack1,2),1);
890             d3 = cell_xy_history(stack1,:) - repmat(vessel(3,:),size(stack1,2),1);
891             d4 = cell_xy_history(stack1,:) - repmat(vessel(4,:),size(stack1,2),1);
892             for z=1:size(d1,1)
```

```
                d11(z) = norm(d1(z,:));
                d22(z) = norm(d2(z,:));
                d33(z) = norm(d3(z,:));
                d44(z) = norm(d4(z,:));
            end
            gen_dist = min([d11;d22;d33;d44]).^(1/2);
            % columns describe generations
            bincounts(:,j) = histc(gen_dist,bins);
            % update neue generation
            while(isempty(stack1)==0)
                if(bdh(4,stack1(1))~=0)
                    stack2(end+1:end+2)=[bdh(4,stack1(1)),bdh(4,stack1(1))+1];
                    stack1(1)=[];
                else
                    stack1(1)=[];
                end
            end
            stack1 = stack2;
            stack2 = [];
        end

        % plot histogram
        figure
        bar(bincounts(1:end-1,:))
        legend('generation 0','generation 1','generation 2','generation 3', ...
            'generation 4','generation 5','generation 6', ...
            'generation 7','generation 8','generation 9', 'Location', ...
            'northeastoutside')
        title(['location histogram for the first 10 generations of ', ...
            num2str(i)])
        ylabel('#cells in distance category')
        xlabel('distance category')

        saveas(gcf,[path_result,'/hist_last_',num2str(i)],'epsc')
        saveas(gcf,[path_result,'/hist_last_',num2str(i)],'fig')
        hold off

        clear d1
        clear d2
        clear d3
        clear d4
        clear d11
        clear d22
```

```
936         clear d33
937         clear d44
938         clear bincounts
939     end
940 
941 end % DrawTissue
```

```
1  function  thres_dam_lineage_2( pathname, cellID, final, step)
2  % function determining and plotting the death threshold against
3  % the DNA damage for a longest lineage of a initital cell cellID or a exact
4  % defined lineages for not eradicating tumors
5  %
6  % pathname: path of the saved data from WhAM
7  % cellID:   ID of a initial mother cell
8  %               => output: arbitrary longest surviving lineage
9  %           ID of a cell in final iteration
10 %               => output: exact the lineage with final of cellID
11 % final:    final iteration
12 % step:     save step out of WhAM
13 
14 bdh = load([pathname,'/birth_death_history.txt']);
15 tdh = load([pathname,'/threshold_damage_history.txt']);
16 cellsMotherID = load([pathname,'/cellsMotherID_',num2str(final),'.txt']);
17 
18 if(sum(cellID == 1:65)==1)
19     % determine all cells of clone
20     stack1 = cellID;
21     stack2 = cellID;
22     while(isempty(stack1)==0)
23         if(bdh(4,stack1(1))~=0)
24             stack2(end+1:end+2)=[bdh(4,stack1(1)), bdh(4,stack1(1))+1];
25             stack1(end+1:end+2)=[bdh(4,stack1(1)), bdh(4,stack1(1))+1];
26         end
27         stack1(1)=[];
28     end
29 
30     % determine longest lineages and select one lineage
31     if(min(bdh(2,stack2))~=-1)
32         last = find(bdh(2,:)==max(bdh(2,stack2)));
33     else
34         last = find(bdh(2,:)==-1);
35     end
```

```
36      while(sum(stack2 == last(1))==0)
37          last(1) = [];
38      end
39      last = last(1);
40  else
41      last = cellID;
42  end
43  % data2 is a storage matrix. In the first row the death threshold is stored,
44  % in the second the DNA damage of the current lineage. data1 parallel
45  % stores the attendant number of iteration. Generally this is only the data
46  % at specific timepoints, when a cell dies or proliferates.
47  help = last
48  if(bdh(2,help)==-1)
49      data1 = final;
50  else
51      data1 = bdh(2,help);
52  end
53  data2= tdh(3:4,help);
54  while(cellsMotherID(help)~=help)
55      help = cellsMotherID(help);
56      data1(end+1) = bdh(2,help);
57      data2(:,end+1) = tdh(3:4,help);
58  end
59  data1(end+1) = 0;
60  data2(:,end+1) = tdh(1:2,help);
61
62  % data describes the development of the death threshold and damage in
63  % the steps of the output of WhAM.
64  % 1st row:  death threshold
65  % 2nd row:  DNA damage
66  help = last;
67  if(bdh(2,help)==-1)
68      data = [];
69      for i = final:-step:0
70          cell_damage = load([pathname,'/cell_damage_',num2str(i),'.txt']);
71          cell_death  = load([pathname,'/cell_death_',num2str(i),'.txt']);
72          cell_ID     = load([pathname,'/cell_ID_',num2str(i),'.txt']);
73
74          if(sum(cell_ID(1,:) == help) ~= 1)
75              help = cellsMotherID(help);
76          end
77
78          data(1:2,end+1) = [cell_death(cell_ID(1,:)==help),...
```

```
            cell_damage(cell_ID(1,:)==help)];
    end
    iterations = 0:step:final;
else
    data=[];
    data(1:2,end+1) = tdh(3:4,help);
    final_help = floor(bdh(2,help)/100)*100;
    for i = final_help:-step:0
        cell_damage = load([pathname,'/cell_damage_',num2str(i),'.txt']);
        cell_death  = load([pathname,'/cell_death_',num2str(i),'.txt']);
        cell_ID     = load([pathname,'/cell_ID_',num2str(i),'.txt']);

        if(sum(cell_ID(1,:) == help) ~= 1)
            help = cellsMotherID(help);
        end

        data(1:2,end+1) = [cell_death(cell_ID(1,:)==help),...
            cell_damage(cell_ID(1,:)==help)];
    end
    iterations = 0:step:final_help;
    iterations(end+1)=bdh(2,last);
end

x=figure;
plot(iterations, data(1,end:-1:1), 'b')
hold on
plot(iterations, data(2,end:-1:1), 'r')
scatter(data1(end:-1:1), data2(1,end:-1:1), 'b')
scatter(data1(end:-1:1), data2(2,end:-1:1), 'r')

legend('death threshold','DNA damage','Location','northwest')
xlabel('iteration')
ylabel('threshold / damage')
title(sprintf(['death threshold / DNA damage plot for the longest lineage', ...
    '\n of clone:\t',num2str(cellID)]))

end
```

```
function spatial_lineage_track( pathname, cellID, final, step )
% Function tracking the movement of a lineage in time.
% pathname: path of the saved data from WhAM
% cellID:   ID of a cell in final iteration
```

```
% final:    final iteration
% step:     save step out of WhAM

vessel=[-20,-40;-40,20;20,-20;60,60];
xmin=-75;  xmax= 75;
ymin=xmin; ymax=xmax;

cellsMotherID   = load([pathname,'/cellsMotherID_',num2str(final),'.txt']);
bdh             = load([pathname,'/birth_death_history.txt']);

help_cell = cellID;
while(cellsMotherID(help_cell)~=help_cell)
    help_cell = cellsMotherID(help_cell);
end

help = cellID;
data = [];
if(bdh(2,help)==-1)
    final_help = 20000;
else
    final_help = floor(bdh(2,help)/100)*100;
end

for i = final_help:-step:0
    cell_xy     = load([pathname,'/cell_xy_',num2str(i),'.txt']);
    cell_ID     = load([pathname,'/cell_ID_',num2str(i),'.txt']);

    if(sum(cell_ID(1,:) == help) ~= 1)
        help = cellsMotherID(help);
    end

    data(end+1,1:2) = cell_xy(cell_ID(1,:)==help,:);
end
iterations = 0:step:final_help;


x = figure;
plot(vessel(:,1),vessel(:,2),'ro','MarkerFaceColor','r','MarkerSize',...
         2*2,'LineWidth',2)
hold on
scatter(data(1:5:end,1),data(1:5:end,2), 'r')
scatter(data([1,end],1),data([1,end],2), 'g', 'ro')
plot(data(:,1),data(:,2), 'b')
```

```
48 axis([xmin,xmax,ymin,ymax])
49 axis equal
50 title(sprintf(['Lineage of: \t',num2str(cellID),'\nInitial cell: \t',...
51     num2str(help_cell)]))
52 end
```

```
1  function dist(plot_accuracy)
2  % Function generating figure for the distance gradient as presented in the
3  % paper. plot_accuracy defines in how many levels the plot should be
4  % presented.
5
6  vessel  = [-20,-40;-40,20;20,-20;60,60];
7  xdata   = repmat([-65:65],131,1);
8  ydata   = repmat([65:-1:-65]',1,131);
9  data    = zeros(131,131);
10
11 for i = 1:(131*131)
12     co = [xdata(i),ydata(i)];
13     d1 = norm(co-vessel(1,:));
14     d2 = norm(co-vessel(2,:));
15     d3 = norm(co-vessel(3,:));
16     d4 = norm(co-vessel(4,:));
17     data(i) = min([d1,d2,d3,d4].^(1/2));
18 end
19 data(1:end,:)=data(end:-1:1,:);
20
21 grad = figure;
22 contourf([-65:65],[-65:65],data,plot_accuracy);
23 end
```