8-4-2010

The Role of Cooperation in Pre-tumor Progression: A Cellular Population Dynamics Model

Konstantin Krepkin

The Role of Cooperation in Pre-tumor Progression:

A Cellular Population Dynamics Model

A Thesis Submitted to the
Yale University School of Medicine
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Medicine

by

Konstantin Krepkin

2010

THE ROLE OF COOPERATION IN PRE-TUMOR PROGRESSION: A CELLULAR POPULATION DYNAMICS MODEL

Konstantin Krepkin (Sponsored by Jose Costa). Department of Pathology, Yale University School of Medicine, New Haven, CT

Abstract

Competition among cells has long been recognized as an important part of the evolutionary process of tissue leading up to the development of cancer. However, the role of cellular cooperation in cancer has been largely ignored. In this work, we investigated the role of cooperation in early tumor progression using a mathematical and agent-based modeling approach. We hoped to learn whether cooperation between cells in spatially organized tissue has a significant role in hastening tumor development, and to uncover general principles governing such cooperation. We focused on the early stages of tumor development given the critical importance of this time period and since we hypothesized that cooperation will have its greatest influence during these early phases. In our model, stem cells were placed into an array of 50 x 20 cell patches, with each patch carrying a maximum of 64 cells. The stem cells' potential to replicate or leave the stem cell compartment through apoptosis or differentiation were governed by modified versions of the Lotka-Volterra equation of ecology. The cells could also acquire mutations in two oncogenes and three tumor suppressor genes. We explored two different cooperation strategies, one in which a cell could acquire the ability to send a cooperative signal that improved the fitness of its immediate neighbors, and one in which a cell could acquire the ability to take advantage of a cooperative signal already in the environment. Cooperation could be acquired through mutation or assigned in advance. We ran simulations of the model in MATLAB. We found that cooperation is a very robust property. Once a small

number of cooperative cells is introduced into a cell population, they rapidly proliferate to the point of being the major constituent of the cell population. Cooperation leads to an increased growth rate of the aggregate cell population, with the growth rate rising in parallel with the cooperative cell fraction. Interestingly, cooperation does not seem to have an effect on cell heterogeneity, counter to what we initially suspected. We also found that cooperative cells have a wider spatial influence than non-cooperating cells. The cooperative cells or their descendant are, on average, present in more patches than corresponding non-cooperative cells at each point in time. Further analysis showed that cooperation is particularly important in the very early pre-tumor stage, when tissue is morphologically and histologically normal, and during times of extensive cell death, such as when tissue experiences necrosis, repeated bouts of inflammation, or cancer treatment. In conclusion, we found that cooperation may play an important role in early tumor progression that is complementary to the competitive interactions among cells that are driven by mutations in tumor suppressors and oncogenes. Cooperation may also be a critical force during later stages of tumor progression when there is significant cell turnover. Our results have implications for cancer prevention and tumor therapeutic strategies.

Acknowledgements

I would like to thank my advisor Dr. Jose Costa for his invaluable support and guidance through every phase of this project. His enthusiasm and skill for teaching got me interested in the subject. His dedication, patience, and insight helped bring the project to fruition. I am indebted for the countless hours he spent with me in discussions and in meticulously revising this manuscript. I am also grateful to all my previous mentors, who have inspired me and taught me what it means to be a teacher, clinician, and researcher. Thank you to my family and friends; you have shaped who I am today.

Table of Contents

I. Introduction

a) Evolutionary biology of cancer

Cancer as a product of an evolutionary process has been recognized for some time now. Cancer is a hereditary process of cells, with successive acquisition of mutations being passed down to progeny cells. In essence, a cancer cell begets a cancer cell. In 1976 Nowell described cancer as arising through the process of evolution and natural selection acting upon the substrate of differential cell fitness acquired through mutation (1). He proposed that cancer cells derive from a single progenitor that undergoes successive cycles of clonal selection, expansion, and death of nonviable clones that results in the appearance of a full-fledged tumor. With the application of these ideas, cancer biology has been able to use the tools of evolutionary biology that were once restricted to the study of organisms and populations over large time scales.

A central feature of evolutionary biology is the relationship between fitness and natural selection. The fitness of an organism or a cell can be defined as the likelihood that its heritable traits will be transmitted to future generations (2). Fitness depends both on the reproductive capacity of the cell and the ability to avoid death; after all, a cell that has a large replication rate but is prone to early death will be unlikely to have a significant number of offspring. It is important to note that fitness is not an absolute quantity, but generally depends on the environment in which a cell resides and the characteristics of the other cells in the interacting population. Natural selection is the process by which traits that confer higher fitness levels become more represented in future generations due to the increased reproductive success of the cells that possess them. Natural selection in cancerous and pre-cancerous tissue is driven by the competition for space, resources, and

waste removal (3). The selective pressures generated by this competition lead to the development of clones able to survive in hypoxic conditions, induce new vessel growth through angiogenesis, and set up colonies in distant organs through metastasis (2). However, natural selection that is so crucial in shaping tumor progression would not be possible without the phenotypic variation generated by mutation. Mutations constitute any heritable variation, either genetic or epigenetic, acquired by cells (2). Epigenetic mutations, which involve changes in DNA methylation that alter gene expression, can occur at a faster rate than mutation. As a result, recent research suggest that epigenetic changes may be at least as important as genetic mutations in driving tumor progression (4).

Natural selection is not the only force that drives the evolution of populations. Many mutations that are eventually important in cancer progression do not lead to a direct increase in cell fitness (3). Genetic drift is the variation of allele frequencies due to chance, which is especially prominent in small populations. Since the development of cancer requires that most mutations occur at the immortal stem cell level, and most stem cell compartments are relatively small, genetic drift plays a significant role in the emergence of cancer. In fact, tumor suppressor gene inactivation depends heavily on genetic drift owing to the neutral nature of the inactivation of the first allele (5).

With rapid advances in the understanding of cancer, general principles that govern cancer progression are beginning to emerge. Hanahan and Weinberg have set forth six fundamental attributes that cells must acquire in order to become cancerous. These include self-sufficiency in growth signals, insensitivity to antigrowth signals, evasion of apoptosis, limitless replicative potential, sustained angiogenesis, and tissue invasion and

metastasis (6). These alterations allow cancer cells to have limitless and autonomous replicative potential that gives them a competitive advantage over normal cells, provide them with the ability to evade programmed cell death and senescence, and give them the tools to increase their nutrient availability either through new vessel growth or by migrating to a more hospitable and less competitive environment. Significantly, cells acquire nearly all of these critical attributes of cancer in the early tumor or pre-tumor stages of development, the time between when the tissue is completely normal to the time just prior to the establishment of a full-fledged tumor. Calabrese *et. al.* propose a model of colon cancer progression in which all relevant cancer mutations can arise in the pre-tumor, phenotypically normal stage without invoking increased mutation rates (7). Given the relevance of this period, we choose to focus on the early phases of tumor progression in our model. Furthermore, understanding the early phases of tumor development provides the hope of early detection and intervention to forestall the evolution to cancer.

b) Ecology and mathematical modeling of cancer

Ecology is the study of the complex interactions of organisms within communities and how their environment shapes those interactions. Both normal and cancerous tissue can be thought of as ecological systems. Parenchymal cells harboring different mutations and having different roles compete, cooperate, and interact with each other in the quest for resources and reproductive success, whether it be the success of the organism as a whole or the selfish propagation of individual cells in a tumor. All these interactions occur in a complex environment made up of stromal cells, networks of blood vessels, and chemical signaling milieu. Therefore, understanding the ecology of cells in tissue is an

important part of understanding the development of cancer.

Cells in the parenchyma of tissue are organized into compartments that shape their ecology. In most tissue, the basal compartment, or stem cell niche, consists of tissue stem cells. The compartment above consists of transit proliferating cells that serve to amplify the cellular population of tissue. The uppermost compartment consists of differentiated cells that undergo few to no cell divisions and are eventually eliminated through apoptosis (8). In the colon, for example, cells are organized into recesses called crypts. At the base of the crypts reside the stem cells that function as the reserve cells of the epithelium. As we move through higher layers, we encounter the proliferating cells in the middle portion of the crypt and the differentiated cells at the top that eventually slough off to allow the cycle to repeat. This type of tissue organization has critical implications for cancer risk. It may actually be an anticancer adaptation since the continuous replacement of cells prevents individual cells from accumulating enough cancerous mutations, thereby reducing cancer risk (9).

Disturbance is another important concept in ecology. Disturbance can be defined as a relatively sudden change in the environment that affects the dynamics of an ecosystem (10). In nature, fire, flood, and predation are examples of disturbances. Disturbance can destroy large portions of a population within a short amount of time and have drastic and interesting effects on ecosystems. Disturbance tends to create species diversity since periodic destruction prevents a single subspecies from overtaking the entire population. The effect of disturbance on species diversity is especially pronounced at intermediate levels of disturbance, an observation often references as the “intermediate disturbance hypothesis” (11, 12). The concept of disturbance has wide applications in

cancer research. Neoplastic and pre-neoplastic cells tend to face various types of disturbances that both exert selective pressure on them and and lead to massive cell death. Examples include immune surveillance, oxygen deprivation in ischemic portions of tumors, and the therapeutic barrage of radio and chemotherapy. Cell death is a mechanism that allows fitter, partially transformed cells to expand, an important step to neoplastic transformation.

Tumor heterogeneity is a salient factor shaping cancer development that is tied to disturbance. Cell heterogeneity is an important determinant of tumor robustness, which underlies a tumor's aggressiveness and ability to evade therapy. With a more heterogeneous cell population, a tumor is less likely to be completely destroyed by an environmental disturbance and more likely to find solutions for survival in adverse conditions. It is well established that different sub-clones within a tumor specialize in such important tasks as angiogensis, metastasis, and chemotherapy resistance. Heterogeneity also affords a tumor the ability to have functional redundancy distributed over dissimilar parts, thus making it less likely that it will experience common mode failure that can entirely destroy it (13). Research in the progression of Barrett's esophagus to esophageal adenocarcinoma has borne out the idea that cell heterogeneity is an important predictor of the progression to cancer (14). Tumor cell heterogeneity has been observed in various cancers, inlcuding colon, pancreatic, renal cell, and breast cancers (15-17). Tumor heterogeneity arises in part due to genetic instability of cancer cells (18), but is also related to the intrinsic spatial organization of tissue (15). Given the relevance of heterogeneity to tumor progression, clarifying the relationship between disturbance, cell heterogeneity, and cell population dynamics in pre-neoplastic and neoplastic tissue is

crucial to understanding the natural history of cancer.

The field of mathematical oncology aims to integrate the ideas of evolutionary biology, ecology, molecular biology, cell biology, statistics, applied mathematics, and physics to generate a better understanding of cancer. At the core of mathematical oncology, and mathematical biology as a whole, is the tenet that modeling a biological system by using some of its fundamental driving principles can produce unexpected insights into the workings of that system. In many ways this is the epitome of the holistic approach that is such an important part of systems biology, in that mathematical modeling can uncover emergent properties of a system that cannot simply be explained by understanding each of its constituent parts. In recent decades mathematical oncology has become an important part of cancer research, in part because of the valuable insights it has produced into the mechanisms underlying cancer progression, but also because the emerging complexity of the field of biology has made it necessary to look beyond reductionist approaches (19). Although elucidating the individual biological components and processes behind cancer is extremely important, it renders only an incomplete view of the immense complexity inherent in cancer; mathematical modeling lends another perspective to understanding cancer as a complex system. Many of the contributions to the understanding of the ecology and evolutionary biology of cancer already discussed came from the field of mathematical oncology. Mathematical oncology has also had a critical role in elucidating the epidemiology of cancer, the mechanisms of tumor development, angiogenesis, metastasis, and in tailoring tumor therapeutic strategies (20-22). Ultimately, the success of mathematical oncology, and the field of cancer research as a whole, relies on mutual understanding and collaboration among many scientific

disciplines, from applied mathematicians to experimental biologists.

c) Cellular cooperation in cancer: previous work

Cooperation in biology is characterized by mutualistic or commensal relationships between parties. Mutualism can be defined as an interaction between parties that leads to an increase in the fitness of both. Commenalism, on the other hand, is a relationship that increases the fitness of one party but has no effect on the fitness of the other (2). Much more research and attention has been paid to the role of competition in carcinogenesis than the possible role of cooperation. This is perhaps not surprising since much of our current understanding of cancer is grounded in evolutionary biology, which is founded on the idea of competition as a major driving force behind natural selection. Nonetheless, cooperation in biology is abound, from ant colonies and bee hives to mitochondria in eukaryotic cells and even the relationship between the cells in a multicellular organism. It is natural then to suppose that cooperation may also have some role in tumor development and progression.

Recent research points out that cooperation does indeed play a part in carcinogenesis, especially when it comes to epithelial and stromal cell interactions. Epithelial cells in tumors produce various chemotactic factors for fibroblasts and other stromal cells, while stromal cells aid in epithelial cell proliferation by producing growth factors and promote angiogenesis and tumor invasion by secreting proteases and angiogenic factors (23). The importance of this intimate epithelial-stromal cooperation was highlighted by work showing the ability of human prostatic carcinoma-associated fibroblasts to stimulate the growth of initiated human prostatic epithelial cells (24) and

human keratinocytes overexpressing platelet-derived growth factor (PDGF) to promote stromal cell proliferation and angiogenesis, which in turn support keratinocyte growth and tumor formation (25). Studies also reveal that stromal elements in carcinomas experience genetic alterations that accompany the genetic changes in the epithelial cells themselves, supporting the idea that epithelial and stromal cells may be co-evolving (26, 27). Furthermore, other stromal cells in tumors, such as inflammatory cells, may hasten tumor development by secreting carcinogenic factors (28).

Axelrod *et. al.* propose that cooperation between partially transformed tumor cells can occur via byproduct mutualism (29). Cells may be able to share complementary diffusible resources, such as growth factors or angiogenic factors. For example, if cells require two growth factors to attain malignant transformation, then if one cell line produces one of the growth factors and a second line the other growth factor, then by sharing the growth factors both cell lines can become fully transformed without having to attain the additional mutation to produce the second growth factor themselves. Cooperation between cells can occur in the form of commensalism as well, as long as the shareable product is in the range of diffusion and the donor cell does not incur additional cost by sharing it. Such cooperation may allow cells to survive and proliferate to eventually acquire all the hallmark mutations of cancer. The authors invoke several lines of evidence to support the possibility of cooperation among actual tumor cells. Non-uniform abundance of proteins is observed in breast and prostate cancer tissue (30, 31), suggesting that individual cells may not be able to produce all the resources necessary for a tumor. Furthermore, these studies also more directly point to possible mutualistic interactions, with cells expressing certain growth factors lying adjacent to cells

expressing the corresponding growth factor receptors, including those for TGF, PDGF, and VEGF. As previously mentioned, there is tremendous heterogeneity in cell phenotypes and genotypes in tumors, indicating the role of cell specialization and cooperative relationships in tumor masses. Finally, the inefficiency in culturing tumor cells *in vitro* suggests the importance of the tumor microenvironment in providing growth signals.

d) Aims of study and hypotheses

In this work, we investigate the role of cooperation in early tumor progression using a mathematical and computer modeling approach. In particular, we hope to establish whether cooperation between cells in spatially organized tissue has a significant role in hastening tumor development, and to perhaps uncover general principles governing such cooperation. **We hypothesize that once cooperation emerges within a cell population, it will become an established feature of the population.** We focus on the early stages of tumor development given the critical importance of this time period and since we believe that cooperation will have its greatest influence during these early phases. Different forms of cooperation are built into the interactions between normal cells, and hijacking of this machinery may be an early step in the march toward neoplasia, well before all the hallmarks of cancer are acquired. Previous published worked has focused largely on competition between cells in normal tissue and tumor, and its role in tumor development, tissue invasion, and response to chemotherapeutic agents. Cooperation has been a relatively ignored area in cancer research, and we hope to shed some light on this subject.

As discussed previously, growth rate and cell heterogeneity are two important determinants of tumor robustness. The growth rate determines a tumor's competitive advantage over nearby normal tissue. Tissue heterogeneity, on the other hand, determines how well a tumor deals with environmental stresses. **We hypothesize that cooperation increases the growth rate and cellular heterogeneity of tissue.** The effect of cooperation on growth rate and cellular heterogeneity are likely intertwined. Cooperation increases the fitness of cells that take part in the cooperative interaction, thus increasing the growth rate of those cells. We believe that this increase in fitness due to cooperation would partly offset the competitive advantage of highly fit sub-clones, preventing them from taking over the cell population. This would increase the number of different sub-clones present in the population and the heterogeneity of the tissue. Resolving how this interplay between the effects of cooperation on tissue growth rate and heterogeneity ultimately shapes out is one of the goals of this study.

In this study, we also look at the relationship between cooperation and the fitness advantage conferred by mutations in various oncogenes and tumor suppressor genes. We would like to find out whether cooperating cells are able to compete with the cells harboring such mutations or whether the mutated cells eventually come to dominate the cell population, leading to the demise of the cooperating cells. **We suspect that cooperating cells would continue to exist within the population despite the presence of cells with mutations in oncogenes or tumor suppressor genes.** Along these lines, we would also like to establish the time period in the pre-tumor stage during which cooperation plays the most significant role in driving neoplastic change.

Our final aim with this research is to investigate the effect of disturbance in

shaping early tumor progression. Judging from its role in ecology, disturbance will likely be an important force in directing tissue and tumor evolution. In particular, we would like to find out how disturbance affects cooperation, mutation, and the relationship between cooperating cells and cells with mutations in key cancer genes. **We believe that disturbance will promote cell heterogeneity and coexistence between cooperating and mutated cells.**

II. Materials and Methods

a) The model

The model is a stochastic agent-based model governed by evolutionary population dynamics and set within the spatial framework of epithelial tissue. The cellular elements of the model are all stem cells residing at the base of colonic crypts. The tissue and population structure of the colon serves as a good model for studying the development of cancer since it has a well-organized microscopic spatial structure and there is a significant amount of information already known about its cellular elements and their dynamics. However, given the general simplified modeling approach used and the fundamental nature of the cellular interactions modeled, the predictions and implications of the model would be applicable to almost any type of somatic tissue. In order to make the model as realistic as possible, the parameters of the model are determined from existing published data. We implement the model in MATLAB and run simulations to explore the evolution of the model dynamics with time.

The author, Konstantin Krepkin, was involved in creating the model, determining model parameters from literature, programming the model into MATLAB, running

simulations, analyzing the data, and interpreting the results. The author's advisor, Dr. Jose Costa, provided enormous guidance to the author, including helping to create the model, determine the important parameters, facilitate literature searches, direct the focus of the simulations, and interpret the results.

i) Spatial and population structure

We model the spatial structure of tissue using the ecologic concept of metapopulation. A metapopulation is a group of spatially separated populations that can interact through migration (32). This concept is especially attractive in studying cancerous tissue since cancerous tissue can be thought of as being separated into distinct populations of cells defined by the mutations they carry (33, 34). In fact, many tumors are a mosaic of cell clusters of different phenotypes and genotypes. Tissue, in general, can be regarded as a metapopulation of cells, with distinct patches of cells separated by stroma or other boundaries.

The spatial structure of the model is a two-dimensional grid of M x N patches, with each patch having a maximum capacity of CC number of stem cells, which we define as the carrying capacity of the patch (Figure 1). In our case, CC is equal to 64 cells from studies investigating stem cells in colonic crypts using DNA methylation patterns (35). In most of the simulations, we use $M = 50$ and $N = 20$. We set the boundary conditions such that the lower and upper boundaries of the patch grid, i.e., row 1 and row M of patches, are connected to each other, in essence forming a cylinder. This simulates the cylindrical topology of the colon. The patches form distinct cellular neighborhoods. Only cells within a patch can interact; there is no interaction between cells in different

patches. There is some communication between neighboring patches, however, as will be discussed later. We refer to the cellular neighborhoods of the model as patches rather than crypts in order to highlight the general applicability of the model's spatial structure. Most tissues in the human body are organized into clusters of cells that interact with each other on a local level. The patch grid can be thought of as representing a "volume" of tissue where a certain cell type resides. As previously described, tissue is generally organized into cellular compartments starting with stem cells at the base and developing increasing differentiation as one moves up.

In our model, the two-dimensional structure of each patch represents the stem cell niche consisting of tissue stem cells. These cells can ultimately differentiate into all the functional cellular elements of a tissue. In the colon, for example, stem cells are known to reside in the crypt base and can differentiate into all the colonic cell types, including enterocytes, goblet cells, and enteroendocrine cells (8). We choose to focus on stem cells rather than all cell types because the progression to cancer must, by definition, occur within the stem cell population. Mutations and epigenetic changes in non-stem cells will not accumulate since they lack the property of self-renewal that is so important for the propagation of a cell's characteristics across time. Barker *et. al.* showed that the deletion of APC in intestinal stem cells leads to the rapid development of microadenomas and the progression to macroscopic adenomas; the same mutation in short-lived transit-amplifying cells resulted in very rare transformation of microadenomas to macroscopic tumors (36). Furthermore, recent evidence suggests that tumors themselves are maintained by populations of cancer stem cells having properties similar to normal adult stem cells. Cancer stem cells identified in leukemia, breast cancer, and colorectal cancers

are able to recapitulate the entire phenotypic repertoire of the tumors from which they are derived (37-39). It is likely that cancer stem cells themselves arise from normal tissue stem cells, but may also derive

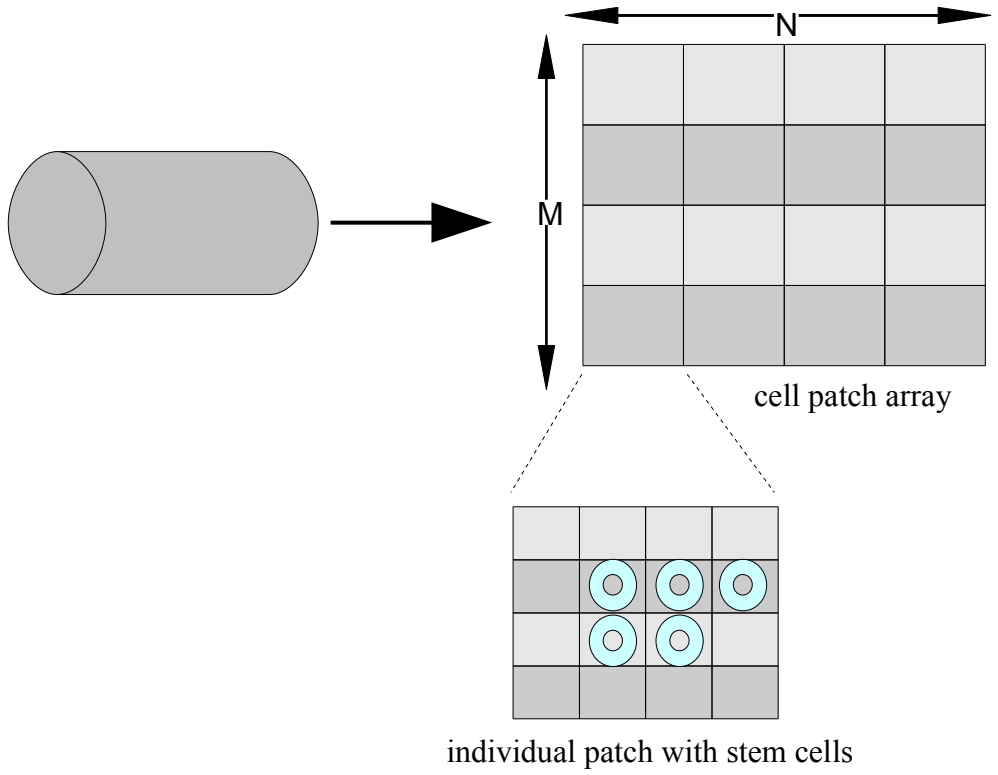

Figure 1. Spatial structure of the model. The topology of the space if that of a cylinder, which can be decomposed into an *M x N* patch grid. Each patch has cells distributed in it, and has a total carrying capacity of *CC* = 64 cells. At time *t = 0*, each patch is stochastically assigned a certain number of stem cells, which are initially centered in the middle of the patch.

from stem cell progenitors that acquire the ability to self-renew. In either case, the stem cell property is crucial to understanding the progression to tumorigenesis and resistance to therapy (see discussion). We realize that just considering stem cells in our model may prevent it from being fully generalizable to all tissue, since not all tissue has the same hierarchical vertically-organized spatial structure as described above. As a result, the stem cell niche may not always be able to be represented as planar collections of adjacent cell

patches. However, enough tissue in the body does have the vertically-organized spatial structure that the model can be widely applicable.

We initialize the model at time $t = 0$ by stochastically assigning a certain number of stem cells to each patch. The cells are centered in the middle of the patch in a contiguous fashion, without any gaps between cells (Figure 1). The probability that a given number of cells is assigned to a patch is given by the probability mass function (40):

No. of cells	7	8	9	10	11	12	13	14	15	16	17	18
Probability	0.002	0.005	0.011	0.020	0.026	0.038	0.042	0.049	0.051	0.056	0.058	0.059
No. of cells	19	20	21	22	23	24	25	26	27	28	29	30
Probability	0.057	0.055	0.053	0.053	0.051	0.050	0.047	0.045	0.043	0.042	0.039	0.038

This data is obtained from DNA methylation studies investigating the stem cells populations of human colonic crypts.

It is important to emphasize that the "empty" space surrounding the cell cluster in the patch is a potential space representing the maximum number of cells that a patch can hold, i.e., the carrying capacity of the patch. In reality, there is rarely empty physical space in a tissue. In the model, when empty space is generated by a stem cell dying or differentiating into a more mature cell it can be thought of as being occupied by an out-of-cell cycle cell (G_0 phase) or a differentiated cell. Since we are strictly concerned with modeling the stem cell compartment, and not differentiated or out-of-cycle cells, to simplify, empty space will appear in the model in such circumstances.

ii) Cell interactions: replication, apoptosis, and cooperation

Since we are dealing with an agent-based model of tissue, herein we describe the

properties of the cell as an agent. The cells in the model can replicate and undergo apoptosis. A cell's replication rate depends on whether the system is in steady state or if there is disturbance to the system that destroys nearby cell patches. This fact was illustrated elegantly in a study that used X-ray radiation to destroy a large proportion of small intestinal crypts in the mouse. Within three weeks, it was observed that the number of crypts had returned to almost normal levels (41). This and several other studies point out that the replication rate of stem cells outpaces their apoptosis rate when there are nearby empty patches. Such observations are quite intuitive since the net growth rate of cells (the difference between the replication and apoptosis rates) would need to be positive in order to return the number of cells and patches to the steady state levels.

Stem cells in tissue can have several fates. A stem cells can replicate symmetrically to produce two daughter stem cells. We define this as the intrinsic replication rate of a stem cell. A stem cell can also divide asymmetrically to produce one stem cell and one differentiated cell. Since we are not concerned with differentiated cells in the model, this situation is equivalent to the cell remaining in steady state, i.*e.* there is no loss or gain of stem cells. Finally, a stem cell can die by apoptosis or necrosis or divide symmetrically to produce two differentiated cells. In all these cases, the stem cell leaves the stem cell compartment and we define the totality of all these events as the intrinsic cell loss rate.

In the model, the maximum intrinsic replication rate of cells in a patch when there are nearby empty patches is given by R. Based on data from stem cell growth in mouse small intestinal crypts following gamma-irradiation, we use the value of 1.143 cell doublings per cell per day in the model (42). In the steady state, when there are no empty

neighboring patches, the replication rate of a cell will be equal to the cell loss rate *a*. This ensures that at steady state there is no significant net growth of cells, as would be expected in normal tissue. We set the maximum intrinsic cell loss rate *a* to the value of 25 cells lost per 1000 cells per day. This value is derived from the observation that most stem cell loss occurs from symmetric division into 2 maturing cells, and that this occurs in about 2.5% of cell divisions for intestinal stem cells that divide every day (7).

When dealing with populations of cells that reside in patches and interact with each other, the concepts of ecology become very important. In fact, the growth of cells in an environment not only depends on the cells' intrinsic replication and apoptosis rates, but also on the number of nearby cells and the types of interactions between these cells. The original model that describes these relationships in ecological systems was developed by Lotka and Volterra in the 1920's. The model focuses on the relationships between species of predators and prey and how the growth rate of each species depends on the number of members of the other species (43). In essence, a predator species' growth increases with increasing number of prey present in the environment, while the growth of prey species decreases with increasing number of predators. This model has also been applied to the growth of a single species in the face of limited resources to conclude that, as the environment becomes over-burdened with individuals, the growth rate of the species suffers.

In our model, cells are firmly anchored to their locations within a patch and cannot freely move about. When a cell replicates, its daughter cells can only occupy the free spaces immediately adjacent to it (Figure 2a). Given these spatial constraints, cellular interactions in the model occur only on the local level, i.e., the immediate 9-cell

neighborhood of a cell (including the cell itself). Taking this fact into account, we introduce a modification of the Lotka-Volterra equation to express the average number of offspring *off (j)* that cell *j* produces per day:

$$off(j) = R_j\left(1-\frac{\sum_l \alpha_{jl}}{K_j}\right) \qquad (1)$$

where R_j is the maximum intrinsic replication rate of the cell *j*, K_j is the carrying capacity of cell *j*'s immediate neighborhood, and α_{jl} is the replication interaction coefficient between cells *j* and *l*, where cell *l* is in cell *j*'s immediate neighborhood. In our case, K_j is equal to 9 to represent the immediate 9-cell neighborhood (including the slot in which cell *j* resides) in which interactions between cells can occur. The sum in equation 1 is over all the cells in this 9-cell neighborhood. In the purely competitive case, α_{jl} is equal to 1, which includes cell *j* itself. If there is cooperation, α_{jl} takes on a value in the interval -1 to 1. The actual number of offspring for cell *j* in a given day is drawn from a Poisson distribution with mean *off (j)*. The offspring of a given cell will be randomly assigned to the free neighboring sites, with each empty site having an equal likelihood of receiving a daughter cell (Figure 2a). Similarly, if there are more daughter cells than empty neighboring spots, the daughter cells will be randomly assigned to the empty spots, with each having equal probability. The rest of the daughter cells will be killed off.

With the interaction coefficients α_{jl} we introduce the idea of cooperation and competition into the model. The interaction coefficient determines whether the relationship of cell *l* to cell *j* is a competitive one or a cooperative one. In the model, α_{jl} can take on a value ranging from -1 to 1. If the coefficient is equal to 1, the relationship is competitive and cell *j*'s fitness suffers from cell *l*'s presence as cell *l* consumes valuable local resources; if the coefficient is less than 1 (or even negative), the interaction is more cooperative and cell *l*'s presence improves cell *j*'s fitness compared to the purely competitive situation. It is important to note that cell *l* may help cell *j*, but the reverse may not be true. In other words, α_{jl} and its transpose α_{lj} may not necessarily be equal or even have the same sign. The interaction coefficients are set in the beginning of each simulation. The offspring of each cell inherit their parents' interaction coefficients. However, the preset interaction coefficients only take effect in a given neighborhood if cooperation develops in the given neighborhood. Otherwise, the interaction between cells is competitive and $\alpha_{jl}=1$.

We use another modified version of the Lotka-Volterra equation to express the probability that a cell is lost from the stem cell compartment in a given day (Figure 2b). We define the probability $P_{loss}(j)$ that cell *j* is lost from the stem cell compartment in a given day as

$$P_{loss}(j) = a_j \left(1 - \frac{\sum_l \beta_{jl}}{K_j} \right) \tag{2}$$

where a_j is the maximum intrinsic cell loss rate for cell *j*, K_j is again the carrying capacity of cell *j*'s immediate neighborhood and equal to 9, and β_{jl} is the cell loss interaction coefficient between cells *j* and *l*. Similar to the replication interaction coefficient α_{jl}, the cell loss interaction coefficient β_{jl} relates the effect that cell *l* has on the likelihood that cell *j* will be lost from the stem cell compartment. β_{jl} can take on a value ranging from 0 to 1. If the coefficient is equal to 0, then cell *l* has little effect on the probability that cell *j* is lost and the interaction is competitive. For coefficients greater than 0, the relationship of cell *l* to cell *j* becomes progressively more cooperative and cell *j* benefits from cell *l*'s presence. Again, the interaction coefficients are preset in the beginning of each simulation and only take effect in a given neighborhood if cooperation develops in the neighborhood.

Our model explores two different types of cooperation strategies. The first one involves a cell acquiring the ability to send out a cooperative signal that can be picked up by its immediate neighbors; we call this the "give cooperation" model. We allow for autocrine signaling to occur such that the cell can benefit from its own cooperative signal. Again, the cells that can benefit from the cooperative signal are the one's in the cooperative cell's immediate 9-cell neighborhood. Physically this cooperative signal can be thought of as a diffusible substance or a signal transmitted via cell to cell contact. The assumption is that all cells in the environment have the tools to receive and benefit from this cooperative signal. The other counterpart cooperation strategy involves a cell gaining the ability to take advantage of a cooperative signal that is already in the environment; we call this the "receive cooperation" model. Also allowing for autocrine cooperation, the cooperative signal can only be received from the cell's 9-cell neighborhood. For both

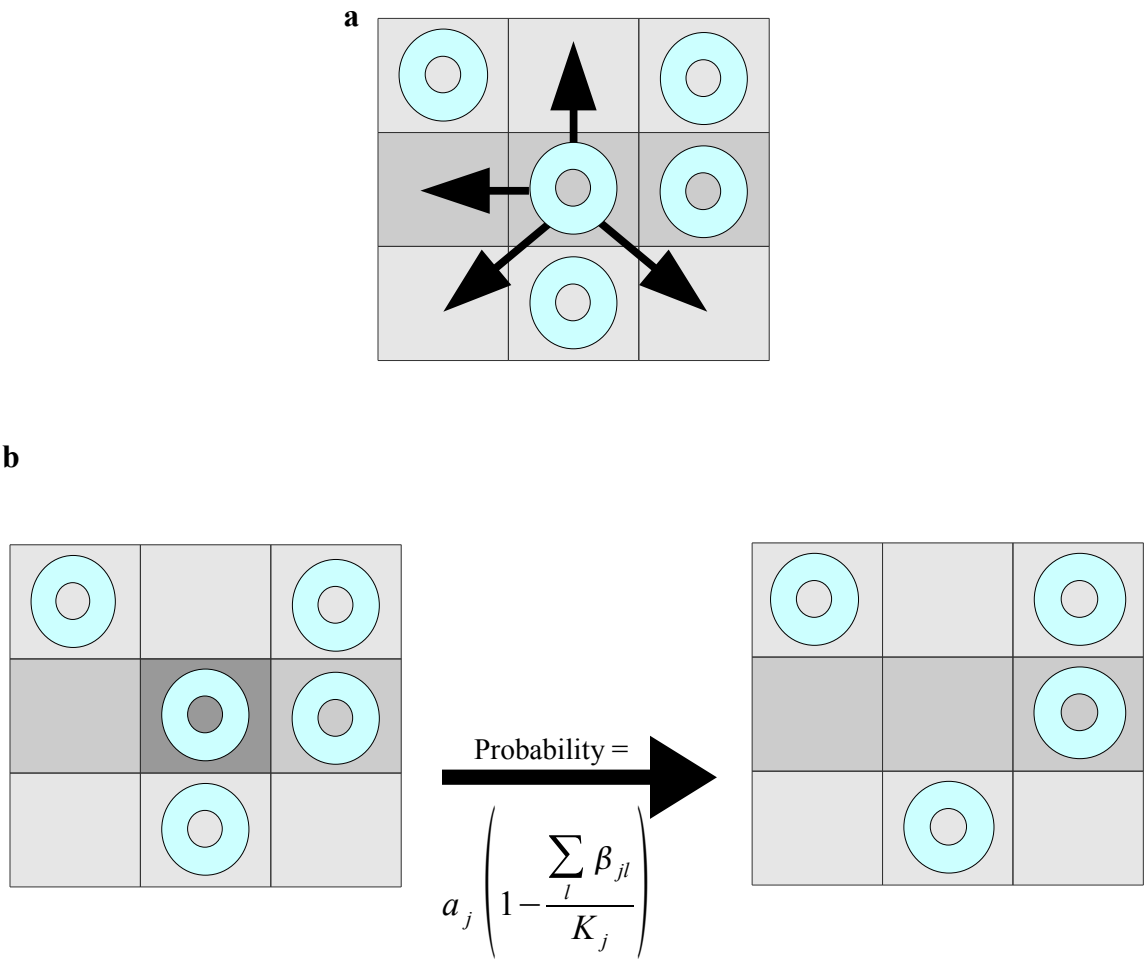

Figure 2. Cell replication and cell loss. (a) When a cell replicates, its daughter cells can occupy only the empty slots that are immediately adjacent to it. Each empty slot has an equal likelihood of receiving the daughter cell. (b) A cell disappearing from the stem cell compartment, along with the probability of this transition, is depicted.

types of cooperation strategies, the ability to either send or receive a cooperative signal can be acquired in the model through mutation or assigned in advance. If a cell acquires the ability to send a cooperative signal, its replication interaction coefficient becomes the preset value that is drawn from a uniform distribution on the interval -1 to 1 in the beginning of each simulation and its cell loss interaction coefficient becomes the preset value drawn from a uniform distribution on the interval 0 to 1. These interaction coefficients would then be the same in relation to all its immediate neighbors, *i.e.*, the

cooperative cell would have the same effect on all its neighbors. Thus, the interaction coefficients α_{jl} and β_{jl} in equations 1 and 2 would reduce to α_l and β_l, respectively. On the other hand, if a cell acquires the ability to a receive a cooperative signal, the interaction coefficients for its neighbors become the preset values independently drawn from the same intervals in the beginning of each simulation. As already stated, these interaction coefficients remain with the cells as they move around and are passed on to offspring. The interaction coefficients of a given cell are the same in relation to any cell in its immediate neighborhood that can take advantage of the cooperative signal. If a cell cannot take advantage of the cooperative signal, then the replication interaction coefficient α_{jl} between the cells and its neighbors would be equal to 1 and the cell loss interaction coefficient β_{jl} would be equal to 0.

Although by themselves these cooperation strategies are more akin to commensalism, where one cell benefits in an interaction without any expense or benefit to the other cell, if more than one cell that is able to send or receive a cooperative signal is present in a neighborhood then true cooperation develops. In such a situation, the relationship between the two cells is of byproduct mutualism, a form of cooperation where both cells are able to send and receive from each other a cooperative signal that results in a mutual increase in fitness.

ii) Patch dynamics

Given the parallel between patches in our model and intestinal crypts, we use another concept in intestinal dynamics to define how patches behave. It has been observed for some time now that crypts in the small and large intestines can divide by a

process called crypt fission (44). This mechanism of crypt reproduction entails either an indentation forming at the base of a crypt and advancing vertically until two crypts are formed or through small buds forming along the parent crypt's vertical axis. This process is most evident in the postnatal period or after insults to the intestinal mucosa, such as from irradiation or chemotherapy, when a large number of new crypts need to be formed to populate an intestine lacking in crypts (41, 44). Furthermore, the crypts most likely to undergo fission are the ones at the upper end of the crypt size distribution and have acquired a sufficient number of stem cells (8). These studies suggest that a crypt tends to undergo fission when there is nearby "empty space" and when it has reached a threshold stem cell capacity. Both of these considerations are important factors in the dynamics of patches in our model.

In the model, we create a rule that a cell patch undergoes fission if it has reached its stem cell carrying capacity, i.e., it has 64 stem cells, and if there is at least one immediately adjacent patch without any stem cells. If a patch undergoes fission, half of its cells will remain in the original patch and half will be transferred to the previously empty adjacent patch (Figure 3).

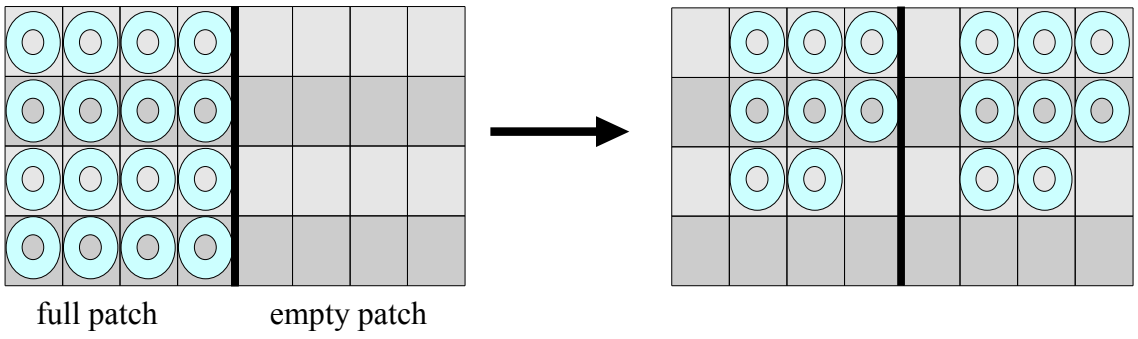

Figure 3. Patch fission. A patch undergoes fission in the model if the number of cells it contains is equal to the carrying capacity of the patch, and there is an adjacent patch without any cells. If this occurs, half of the patch's cells remain in the original patch and half are transferred to the empty neighboring patch. The cells in the recipient and donor patches are re-centered in the middle of each patch.

The cells in the recipient and donor patches will be re-centered in the middle of each patch. If the above conditions are met, a patch will always undergo fission. In the case that there are multiple empty neighboring patches, it will be probabilistically determined which patch will be the recipient of the splitting cells, with each empty patch having equal likelihood of being the recipient.

iv) Mutation

Each cell in the model can undergo mutation within its genome. For the majority of simulations we use the gene mutation rate 1.82×10^{-7} mutations per gene per cell division (45), which is an estimation of the baseline gene mutation rate. However, to explore the effects of increased mutation rate on the dynamics of the model, we also use the mutation rate 1.82×10^{-5} mutations per gene per cell division for several of the simulations. This is in line with evidence that suggests that loss of DNA repair mechanisms can increase the mutation rate by a factor of 10 to 10^4, creating a "mutator phenotype" (46). We focus on mutations within several cancer genes, a hypothetical gene responsible for cooperation, and a cell's essential genes. The cancer genes incorporated into the model include three tumor suppressors and two oncogenes. In general, tumor suppressor genes are thought to be recessive genes, while oncogenes are considered dominant. However, given that we parametrize the phenotypic effects of mutations in the tumor suppressor genes and oncogenes in the model from experimental data on several important tumor suppressors and oncogenes in cancer, these general rules do not necessarily apply in all cases.

We assume that a mutation in a single gene, which we call the "cooperation gene,"

can confer the cooperative property to the cell. This can be either the ability to receive or send the cooperative signal, depending on the cooperation strategy we specify. Furthermore, we assume that this is a dominant mutations, such that a mutation in only one allele is necessary to confer the phenotype.

An essential gene is defined as a gene that is required for an organism to survive. We assume in the model that a mutation in any of a cell's essential genes results in the cell committing apoptosis. Gene deletion studies in yeast indicate that approximately 17.8% of genes in the yeast genome are essential (47). Given the similarity of the yeast and human genomes, we assume that the human genome contains the same proportion of essential genes. We use the value of 25,000 as the number of genes in the human genome (48). Using this value, the gene mutation rate, and the proportion of essential genes, we can calculate the probability that a cell dies in a given replications cycle due to a mutation in one of its essential genes. For the mutation rate 1.82×10^{-7} mutations per gene per cell division, the probability of death per cell division is 8.09×10^{-4}.

A mutation in each of the oncogenes or tumor suppressor genes used in the model will have a distinct effect on the cells' intrinsic replication and apoptosis rates. Let R_{factor} be the multiplicative effect of a mutation in a cancer gene's allele on a cell's intrinsic replication rate R and a_{factor} the multiplicative effect of a mutation on a cell's intrinsic cell loss rate a. In other words, if a homozygous mutation in APC produces an R_{factor} of 0.2 and an a_{factor} of 3, then the cell's new intrinsic replication and apoptosis rates will be $0.2R$ and $3a$, respectively. Table 1 lists the R_{factor} and a_{factor} values for a mutation in each allele of the cancer genes used in the model.

Gene mutation	R_{factor}	a_{factor}
TS1 +/- (49)	2.02	1
TS1 -/- (50)	3.96	4.40
TS2 +/-	1	1
TS2 -/- (51)	1	0.36
TS3 +/- (15, 52)	1.40	1
TS3 -/- (15, 52)	1.01	1
ONCO1 +/- (53)	13.83	3.17
ONCO1 -/-	1	1
ONCO2 +/- (54)	1.40	0.24
ONCO2 -/-	1	1

+/-, heterozygous mutation; -/-, homozygous mutation

TS, tumor suppresor; ONCO, oncogene

R_{factor} , multiplicative effect on replication rate

a_{factor} , multiplicative effect on cell loss rate

Table 1. Multiplicative effects of mutations in each allele of cancer genes used in model on intrinsic replication and cell loss rates.

v) Disturbance

We incorporate disturbance into the model as follows. At time $t = 0$, some of the simulations will experience a perturbation in the system. The perturbation acts by adversely affecting the intrinsic replication and cell loss rates of each cell. The multiplicative effect $R_{perturb}$ of the perturbation on the intrinsic replication rate R is a number drawn from a normal distribution with mean $Rm_{perturb}$ and standard deviation

$R\sigma_{perturb}$. Similarly, the multiplicative effect $a_{perturb}$ on the intrinsic cell loss rate is a number drawn from a normal distribution with mean $am_{perturb}$ and standard deviation $a\sigma_{perturb}$. The new intrinsic replication and cell loss rates will then be $R_{perturb} \times R$ and $a_{perturb} \times a$, respectively. If $R_{perturb} > 1$ or $a_{perturb} < 1$, then $R_{perturb}$ or $a_{perturb}$ will be set to 1, since perturbation can at best have a neutral effect on the cells' reproductive fitness. Using data on the effect of the chemopreventive compound sulforaphane on a human colon carcinoma cell line, we set $Rm_{perturb}$ to 0.974, $R\sigma_{perturb}$ to 0.183, $am_{perturb}$ to 2.341, and $a\sigma_{perturb}$ to 0.610 (55).

The perturbation has a different effect on each cell because each cell has a varying ability to cope with the perturbation and the spatial distribution of the perturbation may be random. As a result, it creates diversity in the cells' fitness levels. The perturbation does not quite qualify as a disturbance since it does not immediately kill any cells, but can be thought of as a mild version of a disturbance. On top of the perturbation, some of the simulations will also experience a disturbance. Disturbance kills a certain fraction of cells p_{kill} at discrete time points. All cells will have equal likelihood of being killed. The times at which the disturbance acts will be randomly distributed during the run-time of the simulation at a certain density *d* of disturbance events per total run-time. We are able to assign the variables p_{kill} and *d*.

b) Model parameters

Table 2 recapitulates the parameters used in the model and their corresponding values.

Parameter	Value
Patch carrying capacity (*CC)*	64
Patch grid dimensions (*M* x *N*)	50 x 20
Maximum intrinsic replication rate (*R*)	1.143
Maximum intrinsic cell loss rate (*a*)	0.025
Baseline gene mutation rate (mutations/gene/cell division)	1.82×10^{-7}
Percentage of essential genes in genome	17.8%
Number of genes in genome	25,000
Probability of death from essential gene mutation per cell division (using baseline mutation rate)	8.09×10^{-4}
Mean multiplicative effect of perturbation on replication rate ($Rm_{perturb}$)	0.974
Standard deviation of multiplicative effect of perturbation on replication rate ($R\sigma_{perturb}$)	0.183
Mean multiplicative effect of perturbation on cell loss rate ($am_{perturb}$)	2.341
Standard deviation of multiplicative effect of perturbation on cell loss rate ($am_{perturb}$)	0.610

Table 2. Model parameters and their values

III. Results

a) How does the number of cooperating cells in the environment vary with time?

We ran simulations of the model with $M = 50$ X $N = 20$ patches using the baseline gene mutation rate of 1.82×10^{-7} mutations per gene per cell division. We wanted to explore whether the introduction of various numbers of cells capable of cooperation in the beginning of each simulation would lead to the proliferation of these cooperative cells or whether they would reach extinction. To this end, we ran simulations with an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population, all distributed randomly throughout the entire space of patches. All the simulations experienced a perturbation at time $t = 0$, but no disturbance (see materials and methods, disturbance section). We ran simulations both for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself, and the "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment. Each simulation was 2000 days long, with data recorded every 100 days. In all simulations, cells could acquire mutations in the cooperative gene if they do not already have cooperative capabilities, which would allow them to either send or receive a cooperative signal depending on the type of model used, or in the tumor suppressors and oncogenes that are built into the model. Since each simulation operates on 1000 contiguous patches, this allows for statistical analysis to be performed on each simulation. We realize that because of patch fission, most patches are not independent of each other. However, due to the spatial constraints, enough groups of patches are

independent that each simulation produces a significant number of independent data points. We approximate the standard error of the mean by treating each of the patches as independent of each other. This technique results in an S.E.M. that is smaller than the actual S.E.M. Nonetheless, it still allows for an appreciation of the statistical significance of the results.

Figure 4 shows the cooperative cell fraction of the total cell population as a function of time for an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population. Both for the "give cooperation" model (Figure 4a) and the "receive cooperation" model (Figure 4b), the cooperative cells introduced in the beginning of each simulation do not become extinct, but rather steadily proliferate with time. In fact, for the simulations with initial cooperative cell fractions of 5% and 16%, the cooperative cells virtually take over the entire cell population by the end of the simulations. Given the trend for the 1% cooperative cell fraction, this would also likely occur with additional time. It is important to point out that there were very few mutations acquired in oncogenes and tumor suppressor genes in the run-time of the simulation. Many simulations did not have any mutations, while the maximum number of mutations was three (in three separate cells) for one cancer gene. The cells harboring mutations in oncogenes or tumor suppressors showed only transient existence, never reaching more than 0.2% of the total cell population, and generally becoming extinct by the end of the simulation. This is most likely due to a low mutation rate and relatively short run-time of the simulations, and will be addressed in further sections.

Figure 5 shows the average number of cells per patch as a function of time for all the simulations for both the "give cooperation" and "receive cooperation" models. The

number of cells per patch steadily increases for initial cooperative cell fractions of 5% and 16%, such that by the end of the simulation most patches come close to the patch carrying capacity of 64 cells. Even though there is a general upward trend in the average number of cells per patch for the simulations with initial cooperative cell fraction of 1%, the increase is quite gradual. Furthermore, it is much slower than the rise in the cooperative cell fraction.

Figures 6 and 7 depict the cooperative cell fraction within each patch for initial fraction of cooperative cells of 1% and 5%, respectively, at four different times during the simulation (t = 100, 700, 1300, and 2000 days); all depicted simulations are for the "give cooperation" model. The number of patches only occupied by cooperative cells (pure white ares in figures), steadily increases with time. The pace at which this occurs is greater for the simulations with a larger number of initially cooperative cells. Furthermore, we can see the dynamics by which the cooperative cells take over the space; first, they proliferate and take over the patches in which they reside, and then spread to adjacent patches by patch fission when the adjacent patches lose their cell complement and become "empty." Thus, the patches filled with cooperative cells tend to cluster together even if the cooperative cells are initially randomly distributed in the space.

b) How do cooperative cells cluster in the environment?

We wanted to determine how cooperative cells become distributed in the space with time, whether there is a tight clustering of cooperative cells or if they disperse sparsely. If cooperative cells tended to cluster together, this would mean that true mutualistic cooperation would be established, since all the cooperative cells in an

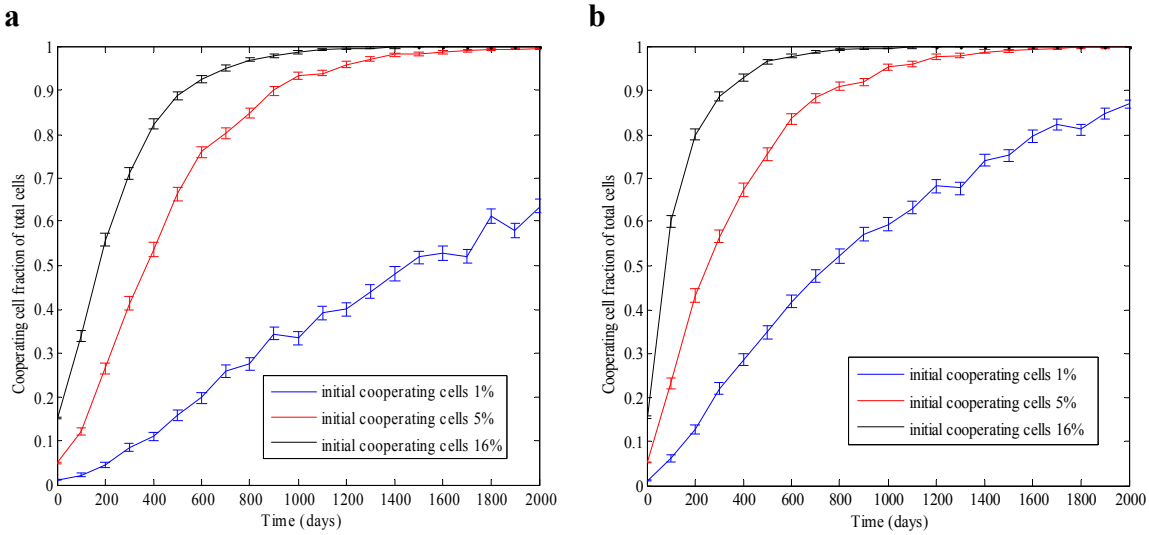

Figure 4. Cooperative cell fraction of total cell population as a function of time. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Each curve represents one simulation. Results are depicted for (a) "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

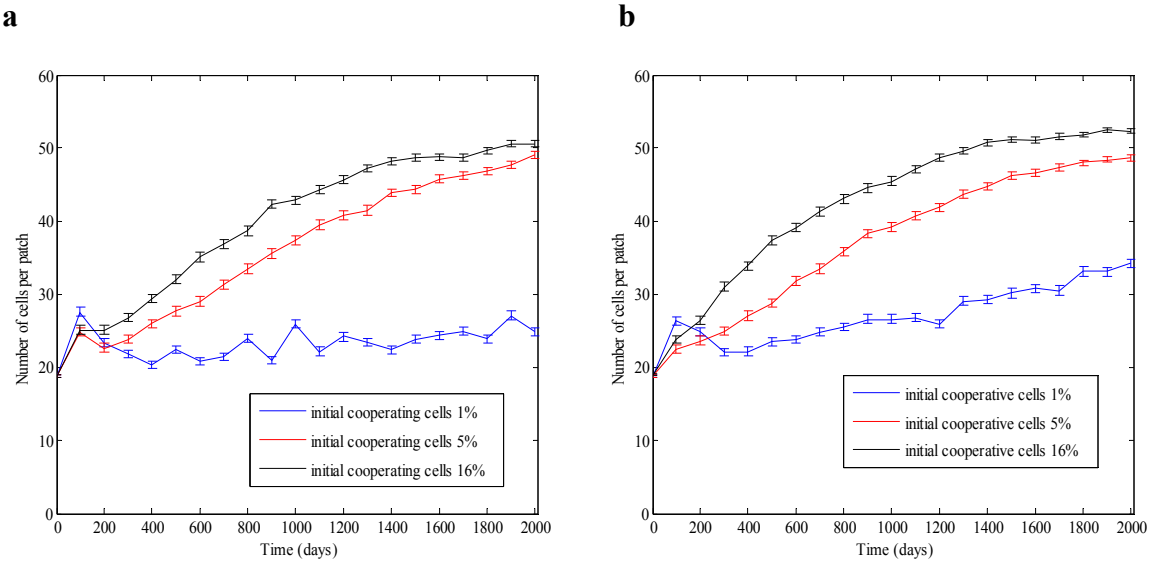

Figure 5. Average number of cells per patch as a function of time. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Each curve represents one simulation. (a) "Give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

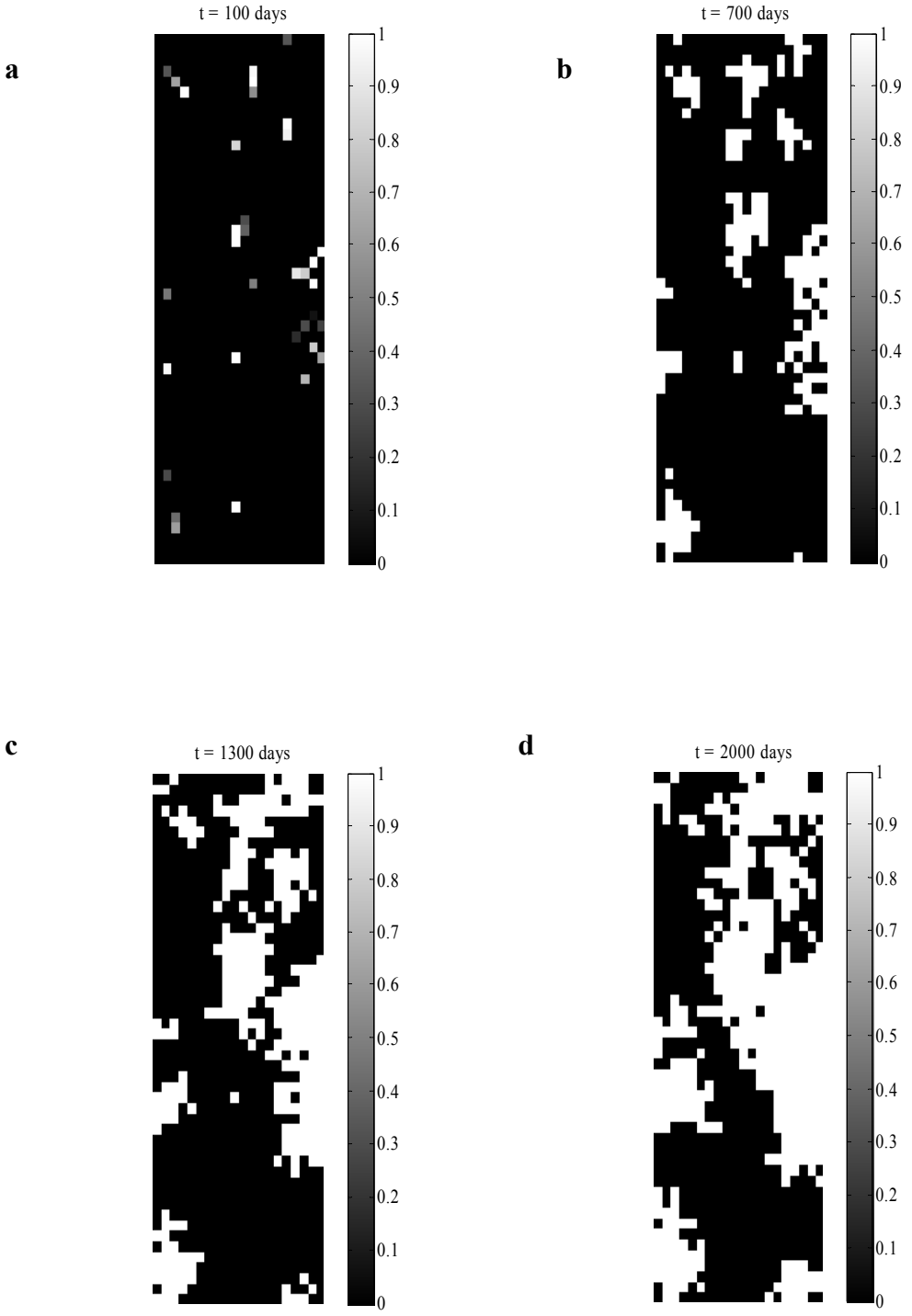

Figure 6. Cooperative cell fraction of total cells in each patch for simulation with initial cooperative cell fraction of 1% and "give cooperation" model. Each pixel represents a patch, and the grayscale intensity represents the fraction of cells that are cooperative, with black being 0 and white being 1. Four time points in the simulation are depicted: (a) t = 100 days, (b) t = 700 days, (c) t = 1300 days, and (d) t = 2000, which is the final time point of the simulation.

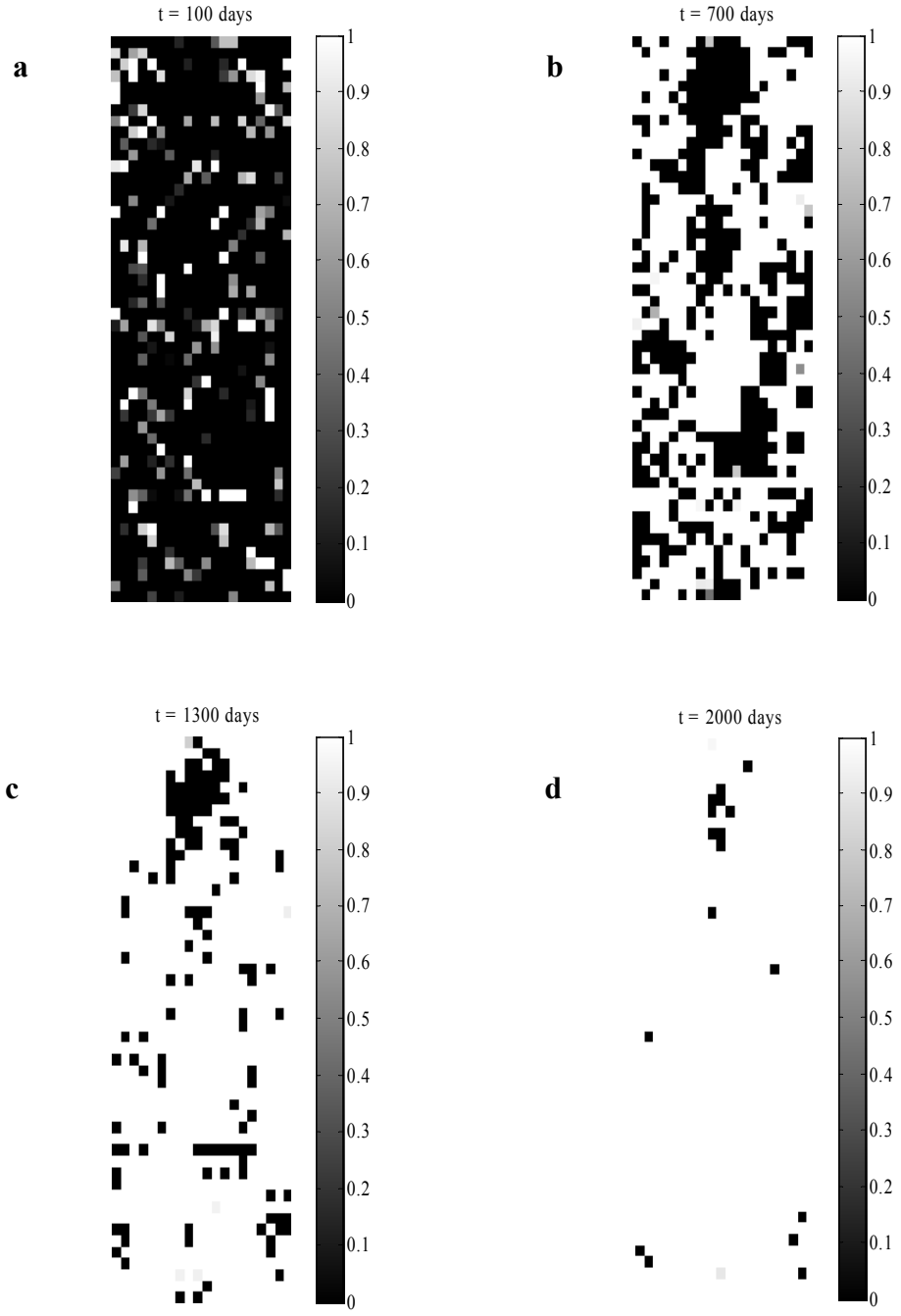

Figure 7. Cooperative cell fraction of total cells in each patch for simulation with initial cooperative cell fraction of 5% and "give cooperation" model. Each pixel represents a patch, and the grayscale intensity represents the fraction of cells that are cooperative, with black being 0 and white being 1. Four time points in the simulation are depicted: (a) t = 100 days, (b) t = 700 days, (c) t = 1300 days, and (d) t = 2000, which is the final time point of the simulation.

immediate neighborhood would be able to benefit from the cooperative signal that the other cooperative cells in the neighborhood emit. Using the same simulations as above, we tallied the number of cooperative cells in the immediate neighborhood of each cooperative cell. Hence, if there were no other cooperative cells in the immediate neighborhood of a given cooperative cell, then this would be a one-cooperative cell cluster (the minimum); if there were eight cooperative cells in a given cooperative cell's neighborhood, then this would be a nine-cooperative cell cluster (the maximum). Figure 8 shows the fraction of such clusters that have at least two cooperative cells as a function of time for an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population; both the “give cooperation” and “receive cooperation” models are depicted. We can see that for all values of initially cooperative cells, the fraction of two-cooperative cell clusters becomes nearly identical and essentially 100% after the first 100 days, even though each simulation started with very different initial fractions. This result indicates that an environment of mutualistic cooperation does indeed become established very quickly, even though initially the cooperative cells are distributed randomly in space.

Figures 9 and 10 depict the relative abundance of clusters with a number of cooperative cells ranging from one to nine. The results are shown for initially cooperating cell fractions of 1% and 5%, respectively, for the “give cooperation” model at four different times during the simulation (t = 0, 200, 700, and 2000 days). We can see that even after a short period of time, clusters with multiple cooperative cells become abundant within the space. Furthermore, the distribution of clusters that becomes established fairly early remains relatively unchanged with time. The peaks at 6 and 9

cooperative cells per cluster is likely due to the tendency of cooperative cells to cluster together and displace non-cooperative cells. If a cooperative cell is away form the edges of a patch and all its neighbors are also cooperative cells, then we have a 9-cell cooperative cluster. If a cooperative cell is at the edge of a patch and all its neighbors are cooperative, then we have a 6-cell cooperative cluster. These results confirm that there is a strong tendency for the cooperative cells to grow in a fashion that establishes a tight clustering of cooperative cells.

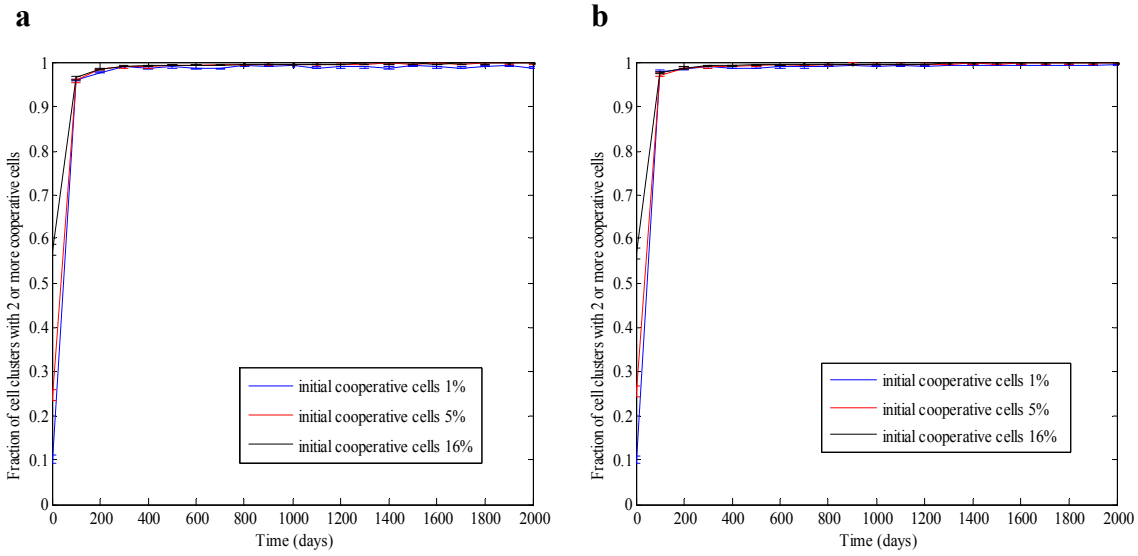

Figure 8. Fraction of cooperative cell clusters with at least two cooperative cells as a function of time. A cooperative cell cluster is defined as the cells immediately adjacent to a cooperative cell, and includes the cooperative cell itself. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Each curve represents one simulation. Results are depicted for (a) "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

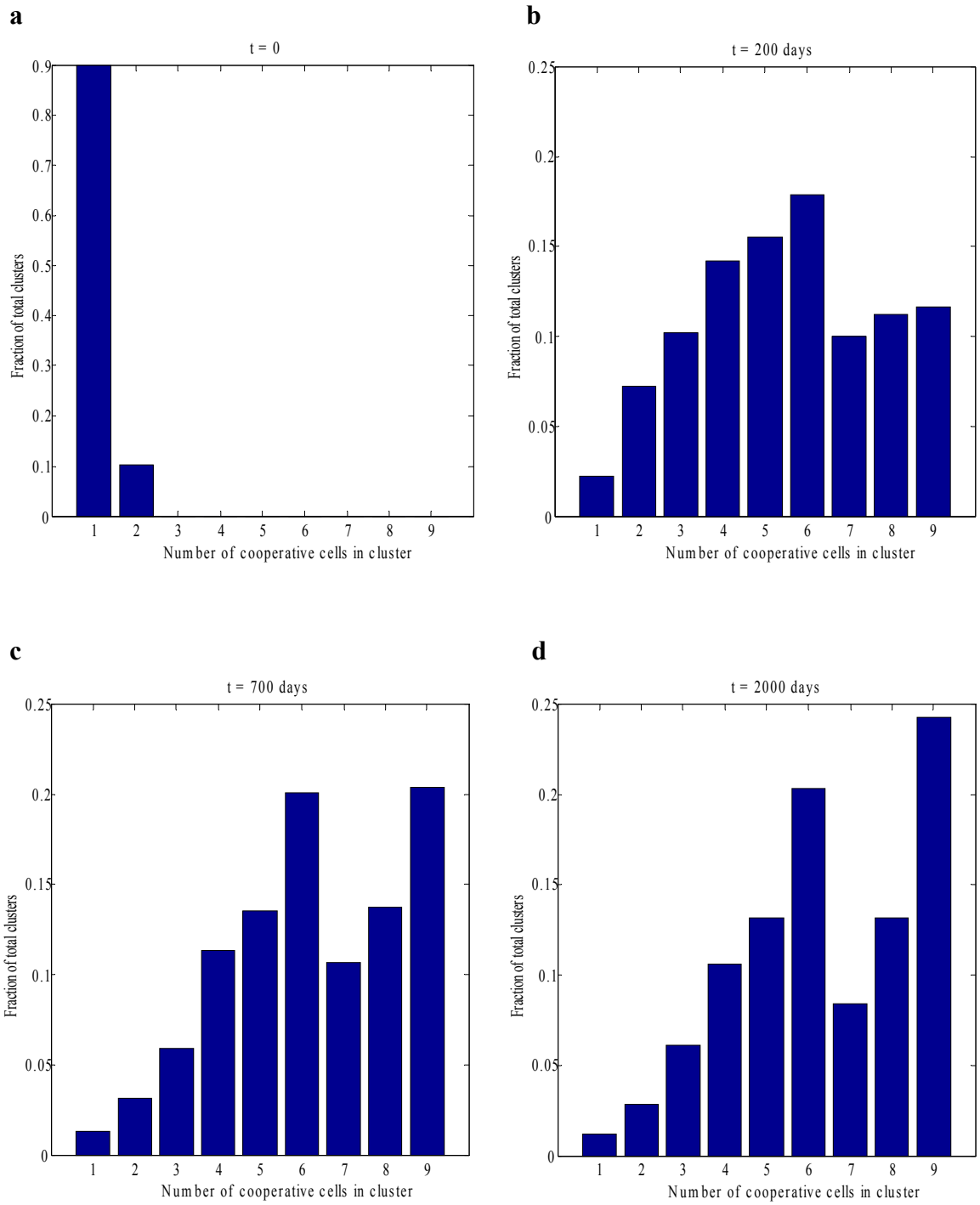

Figure 9. Relative abundance of cooperative cells in cooperative clusters for simulation with initial cooperative cell fraction of 1% and "give cooperation" model. A cooperative cell cluster is defined as the cells immediately adjacent to a cooperative cell, and includes the cooperative cell itself. Four time points in the simulation are depicted: (a) t = 100 days, (b) t = 700 days, (c) t = 1300 days, and (d) t = 2000, which is the final time point of the simulation.

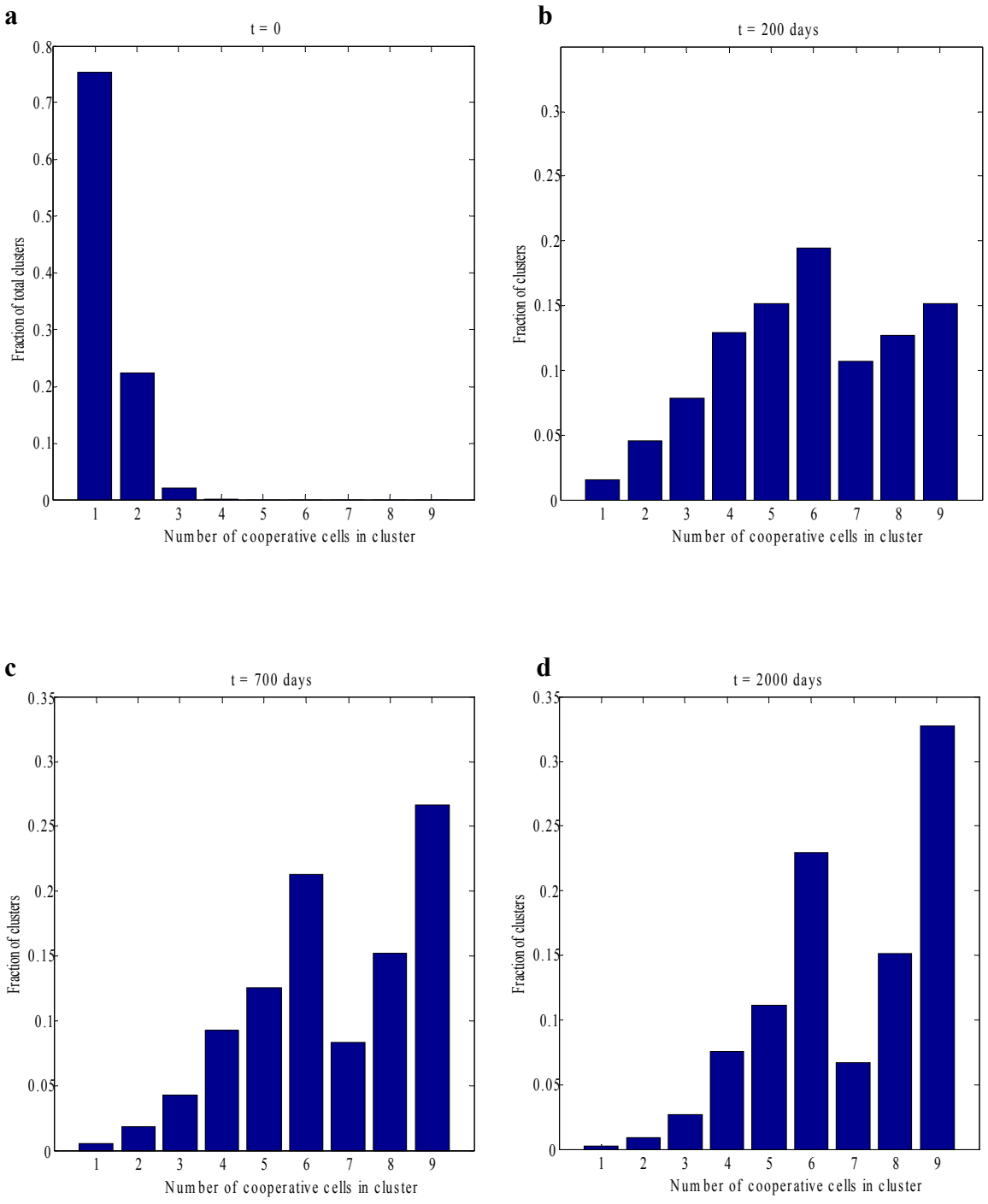

Figure 10. Relative abundance of cooperative cells in cooperative clusters for simulation with initial cooperative cell fraction of 5% and "give cooperation" model. A cooperative cell cluster is defined as the cells immediately adjacent to a cooperative cell, and includes the cooperative cell itself. Four time points in the simulation are depicted: (a) t = 100 days, (b) t = 700 days, (c) t = 1300 days, and (d) t = 2000, which is the final time point of the simulation.

c) How does cooperation affect the patch growth rate?

Since growth rate is such an important property of pre-cancerous and cancerous tissue, we wanted to determine if cooperation increases the patch growth rate in our model. Using the same simulations as above, we calculated the average growth rate of the patches for each simulation. The average is weighted based on the number of cells in each patch. We define the growth rate of a patch as the sum of the individual growth rates of its cells. The growth rate of each cell was calculated as the difference between the replication and cell loss rates, with the replication and cell loss rates determined by equations 1 and 2, respectively (see materials and methods). Figure 11 shows the average patch growth rate as a function of time for the simulations with an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population; both the "give cooperation" and "receive cooperation" models are depicted. The growth rate is greatest for the simulations with initial fractions of cooperative cells of 5% and 16%, and least for the simulation with 1% initial fraction of cooperative cells. The increase in the patch growth rate parallels the rise in the fraction of cooperative cells in the environment; in fact, the curves look very similar (figures 4 and 11). The crossing of the growth rate curves in the latter part of the simulations for initial cooperative cell fractions of 5% and 16% is likely due to the attainment of similar cooperative cell numbers in the two simulations. We can also see that the rate of increase in the patch growth rate is greater than the rate of increase in patch cell numbers (see figure 5). Therefore, cooperation confers a significant growth advantage apart from simply generating more cells that can grow.

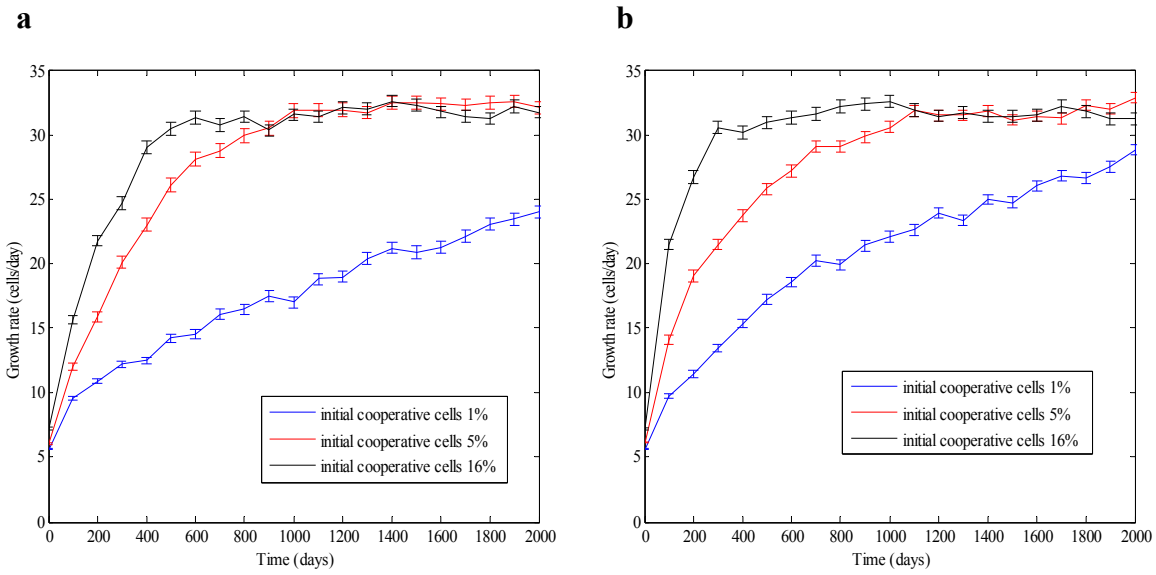

Figure 11. Patch growth rate as a function of time. Average patch growth rates ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Each curve represents one simulation. Results are depicted for (a) "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

d) How does cooperation affect cell heterogeneity?

As previously discussed, cell heterogeneity is an important determinant of a tumor's robustness, and has a strong influence on a tumor's aggressiveness and ability yo evade therapy. In our model, perturbation creates cell heterogeneity by randomizing intrinsic cell fitness levels in the beginning of each simulation (see materials and methods, disturbance section). We use intrinsic cell fitness (or intrinsic replication rate, cell loss rate pairs, which are independent of additive fitness from cooperation) as a surrogate for cell heterogeneity in the model. As a result, if cells have different intrinsic fitness levels, we consider them to be different cell types. Cells with the same fitness levels, such as a given cell's progeny, are of the same cell type.

We use the same simulations as above to determine the effect of cooperation on cell heterogeneity in our model. Figure 12 shows the average number of different cell

types per patch as a function of time for the simulations with an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population; both the "give cooperation" and "receive cooperation" models are depicted. We can see that there is a rather rapid decline in the average cell heterogeneity of the patches, with most patches becoming dominated by a single cell type after 200 days. Furthermore, there is no significant difference in the cell heterogeneity between the simulations with different initial fractions of cooperative cells. As a result, cooperation does not seem to affect cell heterogeneity in the model.

Figure 13 shows the total number of different cell types in the entire space for each of the simulations starting at 100 days. At $t = 0$, the number of different cell types is essentially equal to the number of initial cells, ranging from 18,000 to 19,000. Again, there is a rapid decline in the total cell heterogeneity, with little difference in total cell heterogeneity between the simulations. However, the space does not become completely homogeneous. Even though most patches may be dominated by a single cell type, because of the spatial constraints of the model there is a slower spread of cells across patches. Nonetheless, towards the end of each simulations, the number of different cell types is less than the total number of patches, indicating that certain cell types are spreading and taking over multiple patches.

e) Do individual cooperating cells have a wider spatial influence over time than non-cooperating cells?

In order to further evaluate the effect of cooperation on tissue, we wanted to determine if cooperative cells had a wider spatial influence over time than non-

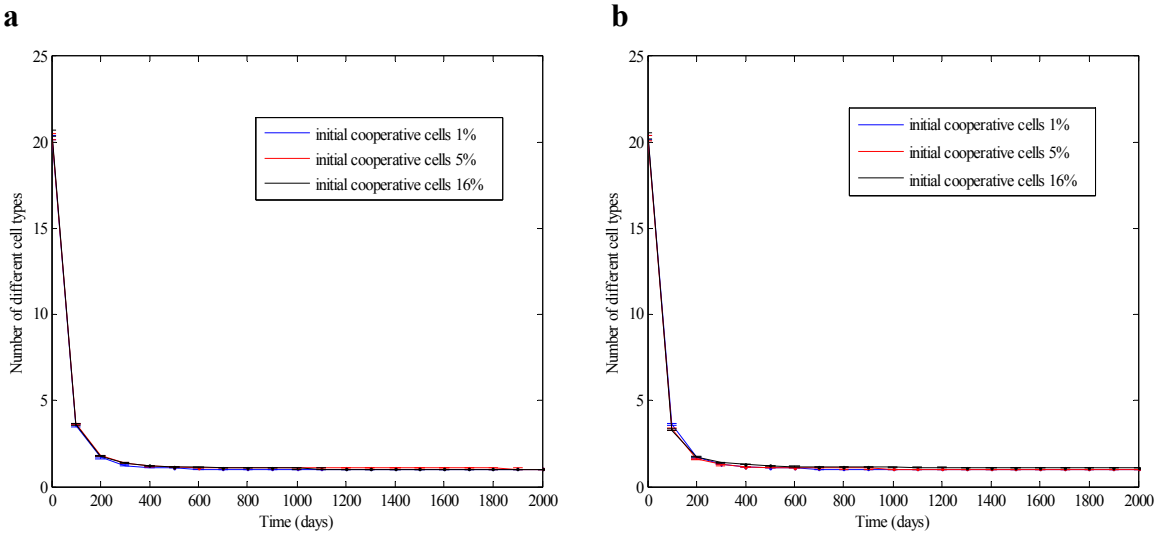

Figure 12. Cell heterogeneity per patch as a function of time. Average number of different cell types per patch ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Cell fitness is used as a surrogate for cell heterogeneity, with cells of different fitness levels being of different cell types. Each curve represents one simulation. Results are depicted for (a) "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

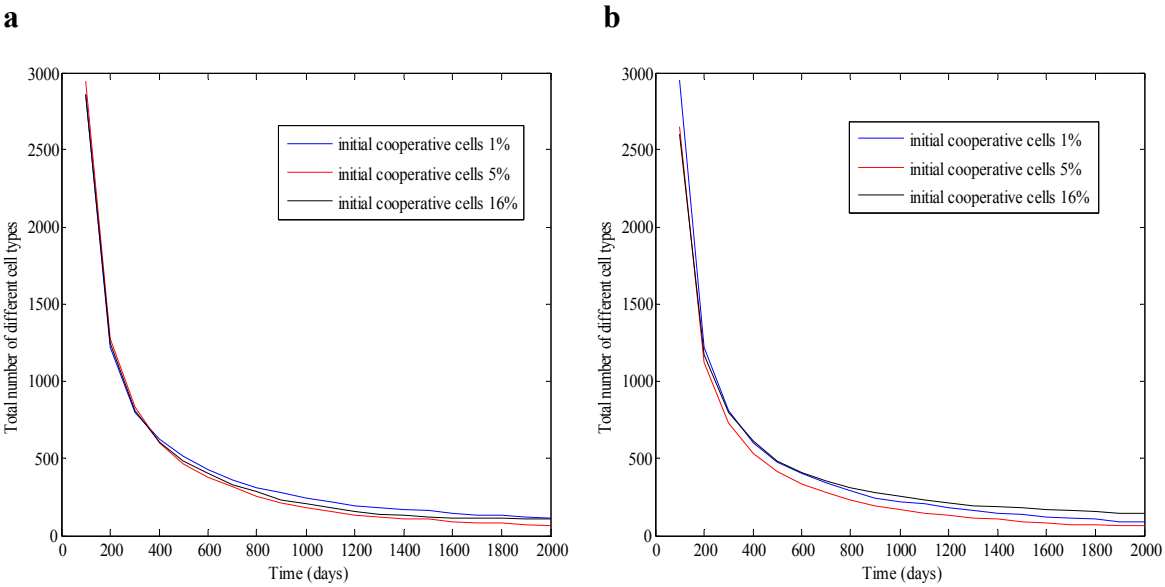

Figure 13. Total cell heterogeneity as a function of time. Total number of different cell types in the entire space for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. The value at $t = 0$ is omitted since it is essentially equal to the total initial number of cells, in the range 18,000 to 19,000. Cell fitness is used as a surrogate for cell heterogeneity, with cells of different fitness levels being of different cell types. Each curve represents one simulation. Results are depicted for (a) "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

cooperating cells. If so, then the cooperative property would help to partially overcome the spatial constraint of tissue. In order to do this, we matched all the initially cooperative cells in the above simulations to an equal number of randomly distributed non-cooperative cells. We then tracked the number of patches in which each of the cells and their descendants were present at each point in time. For example, for an initially cooperative cell fraction of 1%, or about 200 total cooperative cells, we tracked the 200 initial cooperative cells and their descendants and an equal number of non-cooperative cells randomly chosen in the beginning of each simulation and their descendants.

Figure 14 shows the average and maximum number of patches in which the tracked cooperative and non-cooperative cells, and their descendants, are present as a function of time. The average is taken across the cohort of cooperative and non-cooperative cells, respectively, while the maximum refers to the progeny of a cooperative or non-cooperative that is present in the most patches at a given time. If a cell and its progeny die before the end of the simulation, the cell will be present in zero patches from that time onward, and this will be factored into the average. The results are depicted for the simulations with an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population for both the “give cooperation” and “receive cooperation” models. We can see that there is a significant difference between the cooperative and non-cooperative cells in the average number of patches in which they are present, with the cooperative cells being present in more patches on average than their non-cooperative counterparts after $t = 100$ days. From time $t = 0$ to $t = 100$ days, there is significant attrition of both the cooperative and non-cooperative tracked cells across all the simulations. However, during this time a few very competitive cooperative cells begin

to grow and take over their respective patches, such that after $t = 100$ days they begin to spread and invade neighboring patches. The spatial extent of these very competitive cooperative cells is evidenced by the maximum curves, which indicate that toward the end of the simulations some of these cells are present in over 10% of the total patches, or greater than 10 times more than their non-cooperative counterparts. There is also a significant difference in the average number of patches in which the cooperative cells are present between the simulations with different numbers of initially cooperative cells, which is evident later in the run-time for the "give cooperation" model and through most of the run-time for the "receive cooperation" model. This is likely due to the fact that only a small number of the initially cooperative cells are very competitive and become dominant within and across patches, and the rest die out. Hence, for a larger number of initial cooperative cells, a smaller fraction of the initial cooperative cells survive (figure 15).

f) How does eliminating perturbation affect the dynamics of the system?

We defined perturbation in the model as an initial event that occurs at time $t = 0$ that adversely affects the intrinsic replication and cell loss rates of each cell (see materials and methods, disturbance section). Perturbation has a different effect on each cell, such that some cells suffer from it more than others. As a result, it serves to randomize cell fitness levels. So far, the simulations we have discussed have included a perturbation in the beginning that created a heterogeneous cell population by randomizing the cell fitness levels. However, we also wanted to find out how having a homogeneous cell population, with all cells having the same fitness levels, would affect the system. We ran simulations

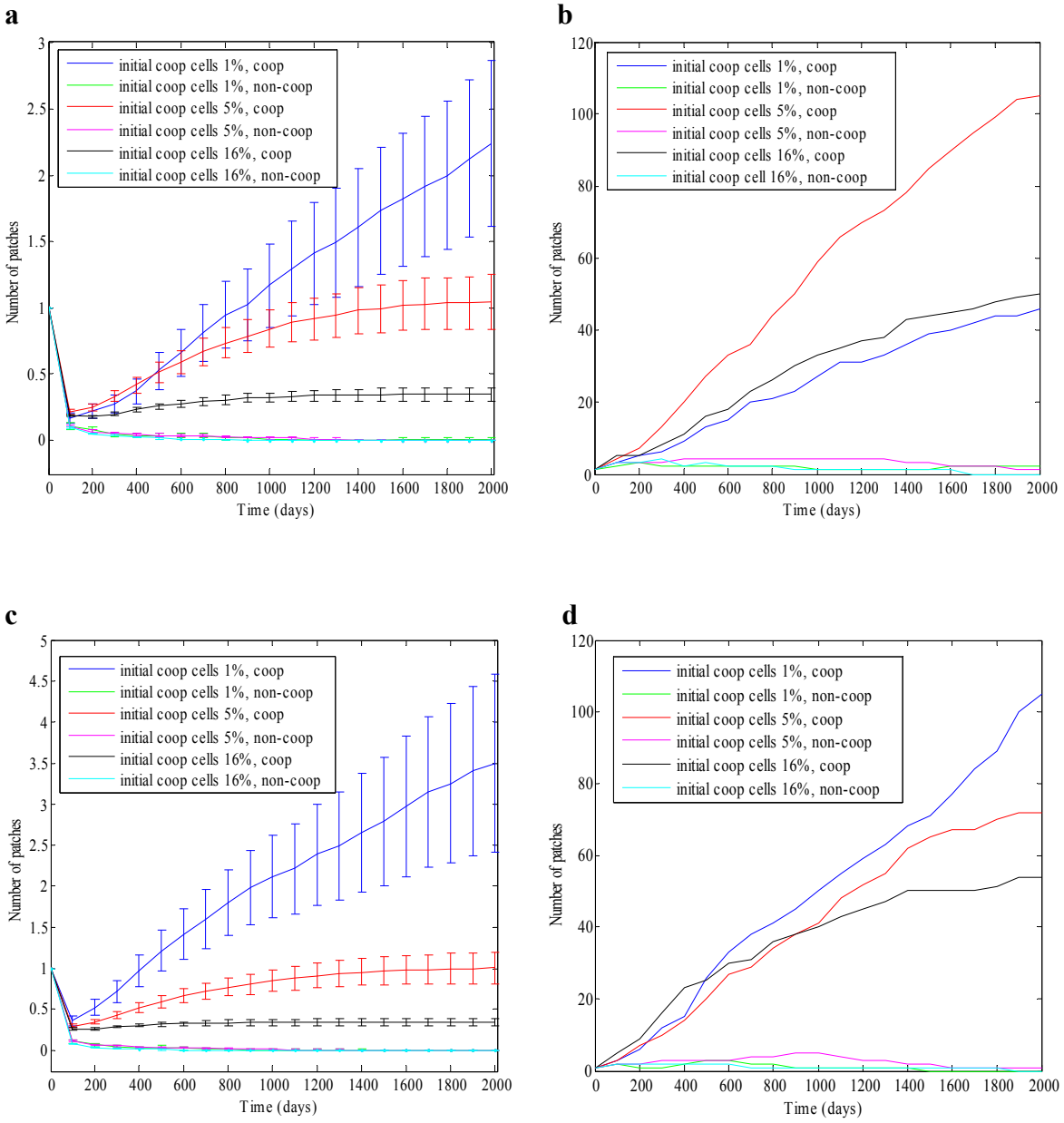

Figure 14. Average and maximum number of patches in which cooperative and non-cooperative cells and their progeny are present as a function of time. Simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Coop refers to the initial cooperative cells and non-coop to the non-cooperative cells. (a) Average number of patches in which the cell cohorts are present ± S.E.M for "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. (b) Maximum number of patches in which the cell cohorts are present for "give cooperation" model. (c) Average number of patches in which the cell cohorts are present ± S.E.M for "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment. (d) Maximum number of patches in which the cell cohorts are present for "receive cooperation" model.

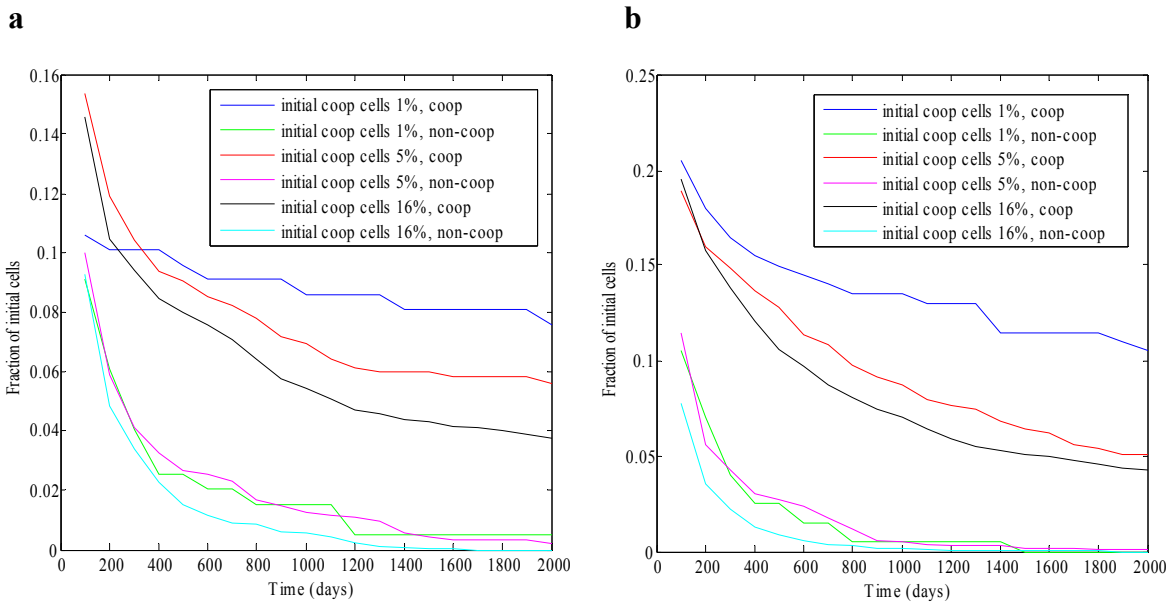

Figure 15. Fraction of initial cooperative and non-cooperative cells or their progeny present as a function of time. Simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown. Coop refers to the initial cooperative cells and non-coop to the non-cooperative cells. The value at $t = 0$ is omitted since it is always equal to 1. Results are depicted for (a) "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself and (b) "receive cooperation" model, where a cooperative cell has the ability to take advantage of a cooperative signal that is already in the environment.

with an initial fraction of cooperative cells of 1%, 5%, and 16% of the total initial cell population, but without perturbation. We ran simulations only for the "give cooperation" model. From now on, we choose to focus only on the "give cooperation" model, since it represents a more natural and common form of cooperation, with a cell acquiring the ability to help its neighbors. Furthermore, as we have already seen, the dynamics of the "give cooperation" and "receive cooperation" models are very similar. The rest of the specifications for the simulations were the same as before. Each simulation was 2000 days long, with data recorded every 100 days, and cells could acquire mutations in the cooperative gene, if they do not already have cooperative capabilities, or in the tumor suppressors and oncogenes that are built into the model. We used $M = 50$ X $N = 20$ patches and the baseline gene mutation rate of 1.82×10^{-7} mutations per gene per cell

division.

Figure 16 shows the cooperative cell fraction of the total cell population and the average number of cells per patch as a function of time. The results are very similar to the situation with perturbation (see figures 4 and 5). There is a steady increase in the cooperative cell fraction with time, with the cooperative cells virtually taking over the entire cell population in the simulations with initial cooperative cell fractions of 5% and 16%. There is also an upward trend in the average number of cells per patch, with the rise being slower than the rise in the cooperative cell fraction, as seen in the simulations with perturbation. The only difference between the simulations with and without perturbation is that the simulation with an initial cooperative cell fraction of 1% and no perturbation has a slightly more rapid increase in both the cooperative cell fraction and average cells per patch than its counterpart with perturbation. As before, the cells that acquired mutations in tumor suppressors and oncogenes became extinct very rapidly.

Figure 17 shows the fraction of cell clusters with at least two cooperative cells and the average patch growth rate as a function of time. The curves depicting the fraction of two-cooperative cell clusters with perturbation and without look essentially identical (see figure 8); the fraction of two-cooperative cell clusters become nearly 100% after the first 100 days for all simulations. The average patch growth rates are also very similar for the situations with and without perturbation (see figure 11). There is a steady increase in the patch growth rate that parallels the rise in the fraction of cooperative cells in the environment. The rate of increase in the patch growth rate is also greater than the rate of increase in the patch cell numbers.

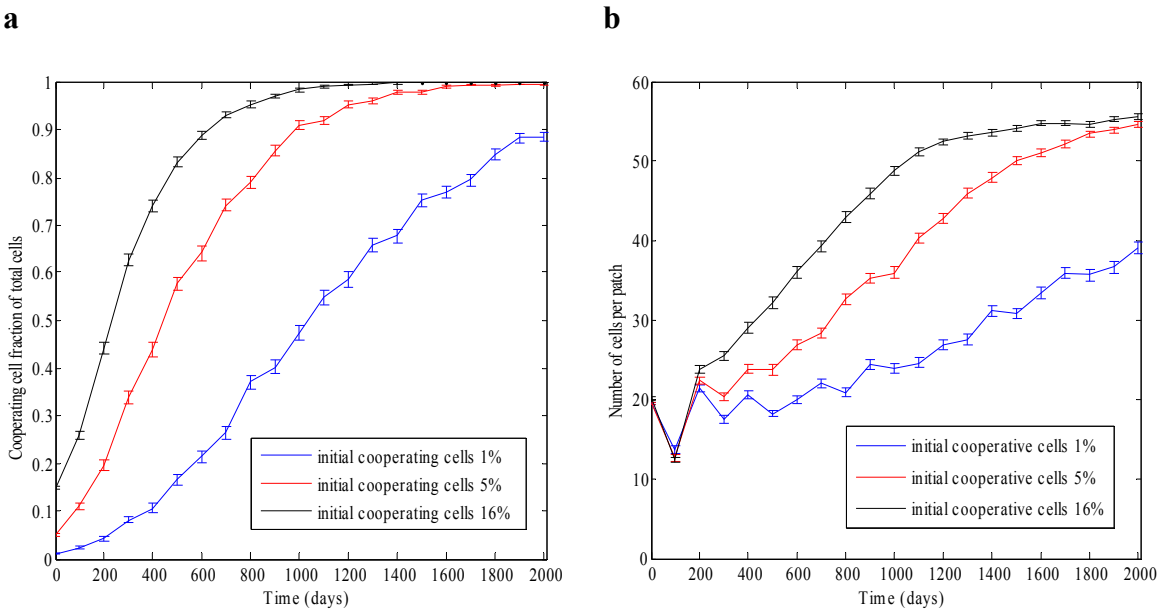

Figure 16. Cooperative cell fraction of total cell population and average number of cells per patch as a function of time for simulations without perturbation. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Cooperative cell fraction of total cell population. (b) Average number of cells per patch.

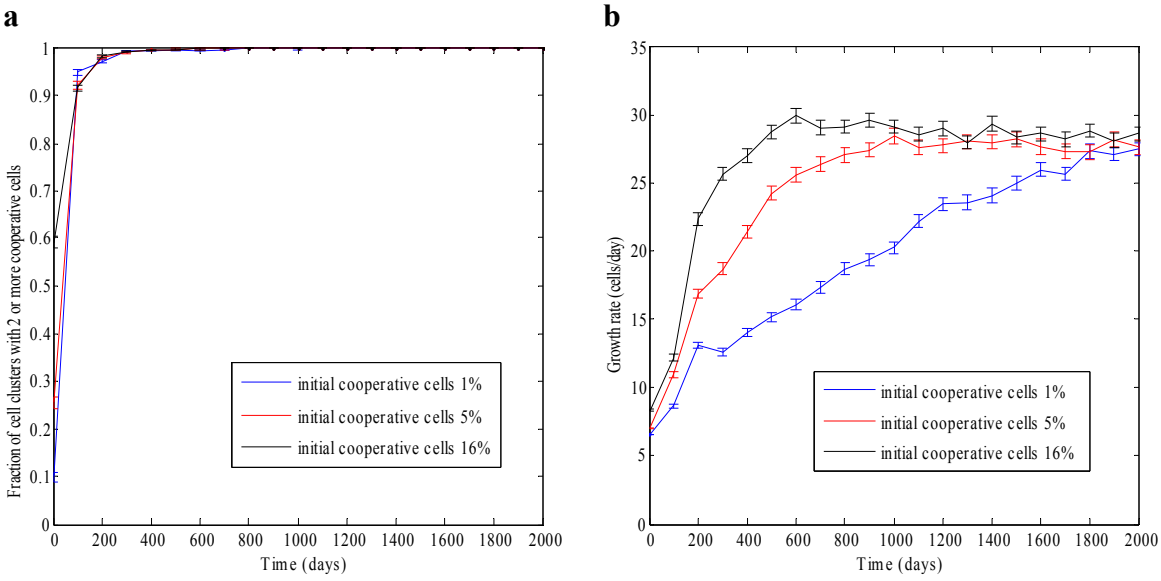

Figure 17. Fraction of cooperative cell clusters with at least two cooperative cells and average patch growth rate as a function of time for simulations without perturbation. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Fraction of cooperative cell clusters with at least two cooperative cells. (b) Average patch growth rate.

As before, we also tracked the initial cooperative cells and their descendants, as well as a matched number of randomly distributed non-cooperative cells and their descendants. Figure 18 shows the average and maximum number of patches in which the tracked cooperative and non-cooperative cells, and their descendants, are present as a function of time. The results look very similar to the situation with perturbation (see figure 14). The data on cell heterogeneity is not shown since all cells have the same intrinsic fitness level when there is no perturbation.

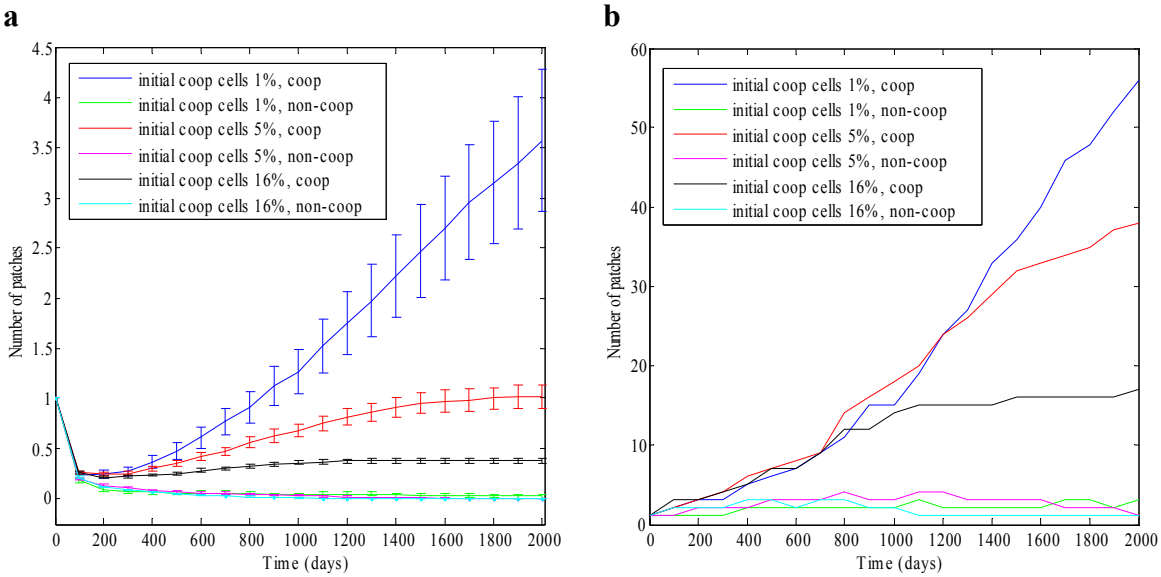

Figure 18. Average and maximum number of patches in which cooperative and non-cooperative cells and their progeny are present as a function of time for simulations without perturbation. Simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Coop refers to the initial cooperative cells and non-coop to the non-cooperative cells. (a) Average number of patches in which the cell cohorts are present ± S.E.M. (b) Maximum number of patches in which the cell cohorts are present.

g) How does the addition of disturbance affect the dynamics of the system?

We wanted to explore how the addition of disturbance to our model would affect the system; we defined disturbance as an event that kills a certain fraction of cells at

discrete time points. To this end, we ran simulations with disturbance occurring at an average density *d = 1 event/100 days* that would kill on average p_{kill} = 10% of all cells (see materials and methods, disturbance section). We ran simulations with initial cooperative cell fractions of 1%, 5%, and 16% of the total initial cell population. We ran simulations only for the "give cooperation" model. The rest of the specifications for the simulations were the same as before. Each simulation was 2000 days long, with data recorded every 100 days, and cells could acquire mutations in the cooperative gene, if they do not already have cooperative capabilities, or in the tumor suppressors and oncogenes that are built into the model. We used *M = 50* X *N =20* patches and the baseline gene mutation rate of 1.82×10^{-7} mutations per gene per cell division.

Figure 19 shows the cooperative cell fraction of the total cell population and the average number of cells per patch as a function of time. There is no substantial difference between these results and the corresponding ones for the simulations without disturbance, including with and without perturbation. Even the average number of cells in each patch does not seem to suffer despite disturbance wiping out a significant portion of cells twenty times during the run-time of each simulation. Similarly, the fraction of cell clusters with at least two cooperative cells (figure 20a), the average patch growth rate (figure 20b), the average number of different cell types per patch (figure 21a), the total number of different cell types in the entire space (figure 21b), and the average number of patches in which the tracked cooperative and non-cooperative cells, and their descendants, are present as a function of time (figure 22a) show no substantial difference between the simulations with and without disturbance. The maximum number of patches in which the tracked cooperative cells, and their descendants, are present as a function of

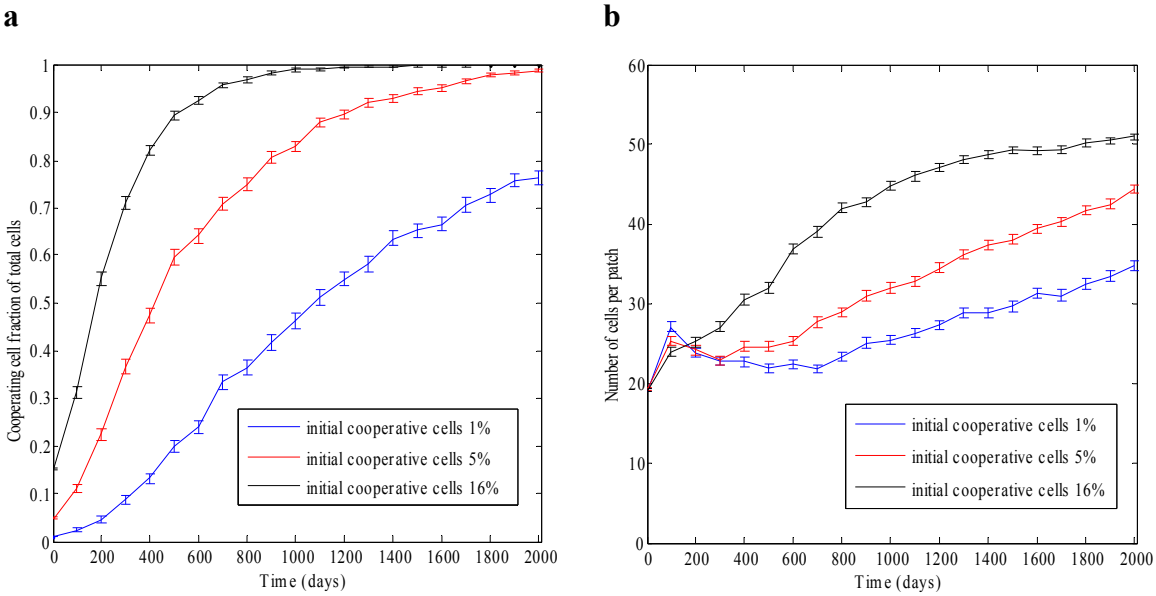

Figure 19. Cooperative cell fraction of total cell population and average number of cells per patch as a function of time for simulations with disturbance (*d = 1 event/100 days* and p_{kill} = 10%). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Cooperative cell fraction of total cell population. (b) Average number of cells per patch.

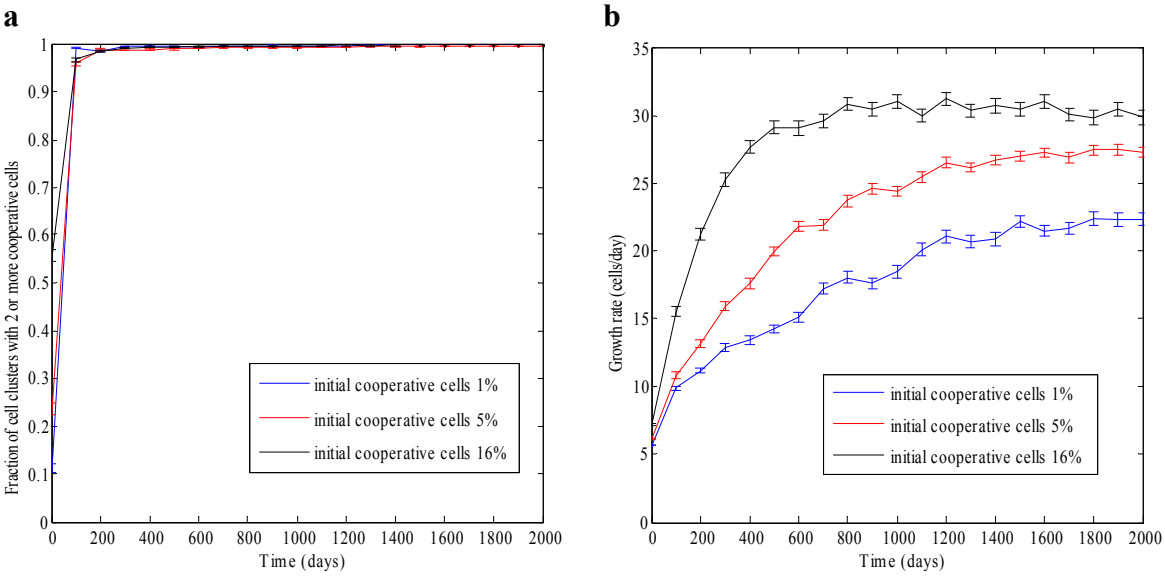

Figure 20. Fraction of cooperative cell clusters with at least two cooperative cells and average patch growth rate as a function of time for simulations with disturbance (*d = 1 event/100 days* and p_{kill} = 10%). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Fraction of cooperative cell clusters with at least two cooperative cells. (b) Average patch growth rate.

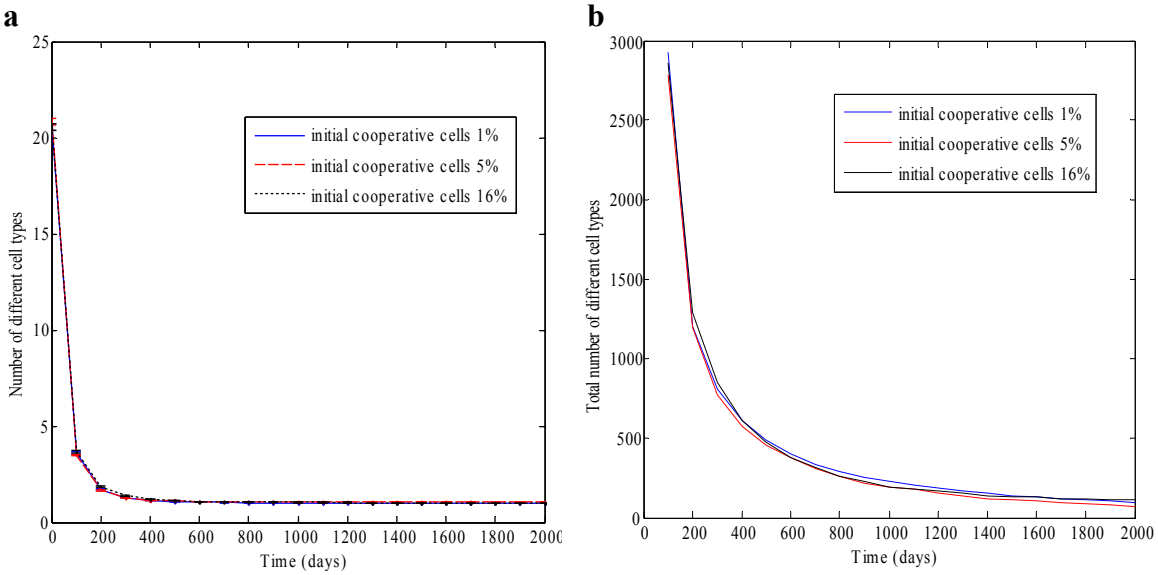

Figure 21. Average number of different cell types per patch and total number of different cell types in the entire space as a function of time for simulations with disturbance (*d = 1 event/100 days* and p_{kill} = 10%). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Average number of different cell types per patch. (b) Total number of different cell types in the entire space.

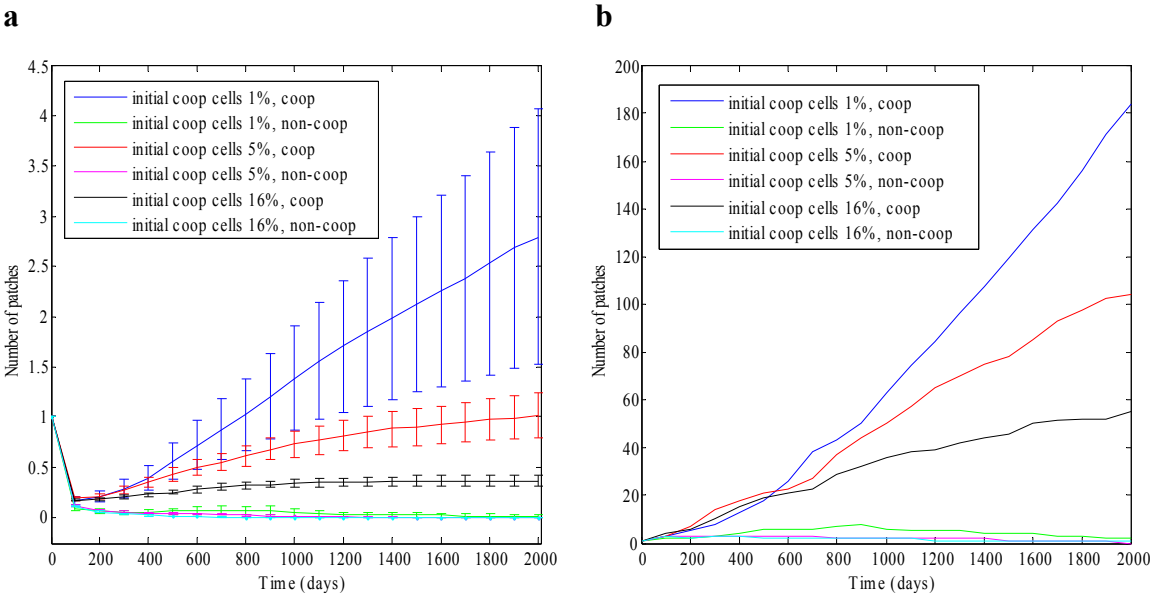

Figure 22. Average and maximum number of patches in which cooperative and non-cooperative cells and their progeny are present as a function of time for simulations with disturbance (*d = 1 event/100 days* and p_{kill} = 10%). Simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Coop refers to the initial cooperative cells and non-coop to the non-cooperative cells. (a) Average number of patches in which the cell cohorts are present ± S.E.M. (b) Maximum number of patches in which the cell cohorts are present.

time is substantially higher for the simulations with disturbance compared to those without disturbance (figure 22b). This may occur because disturbance kills many of the more fit cooperative cells, allowing other cooperative cells to proliferate without as much competition. Finally, as we have seen before, the cells that acquired mutations in tumor suppressors and oncogenes were not able to compete with the cooperative cells since there were so few mutations.

h) How does increasing the mutation rate affect the dynamics of the system?

So far none of the simulations we discussed have had more than a few cells acquire mutations in tumor suppressor genes and oncogenes. However, we wanted to determine how the presence of cells harboring mutations in these genes would alter the effects of cooperation on the system. To achieve this, we increased the mutation rate 100-fold to 1.82×10^{-5} mutations per gene per cell division. We first ran simulations with initial cooperative cell fractions of 1% and 5% of the total initial cell population for the "give cooperation model." These simulations experienced a perturbation in the beginning, but did not experience any disturbance. The rest of the specifications were the same as described before.

Figure 23 shows the cooperative cell fraction of the total cell population and the average number of cells per patch as a function of time. There is no substantial difference compared to the previous simulations (those with normal mutation rate, with/without perturbation, and with disturbance), except for a slightly slower increase in the cooperative cell fraction for the simulation with an initial cooperative cell fraction of 1%.

The major difference compared to the previous simulations, however, is that now the cells harboring mutations in tumor suppressor genes and oncogenes begin playing a more important role (figure 24a, which shows the fraction of cells harboring mutations in any of the tumor suppressors or oncogenes in the model). There is a steady increase in the number of such cells, albeit at a slower rate than the number of cooperative cells; the cooperative cells do get a head start, however, since a certain number is introduced in the beginning of the simulations. It is important to note that the vast majority of the cells with mutations in the tumor suppressors or oncogenes have a mutation in oncogene 2 (see materials and methods, mutation section), which provides the greatest reduction in apoptosis and the highest ratio of replication increase to apoptosis reduction.

Figure 24b shows the fraction of cells that have both the cooperative ability and have a mutation in a cancer gene (tumor suppressor or oncogene). For initial cooperative cell fraction of 5%, most of the cells with mutations in cancer genes also have the cooperative ability. On the other hand, for initial cooperative cell fraction of 1%, most of the cells with mutations in cancer genes do not have the cooperative ability, and are able to proliferate faster than their counterparts in the simulation with 5% initial cooperative cell fraction. This shows that the cells with cancer gene mutations are able to compete with cooperative cells. However, if the cooperative cells gain a significant head start, the only way the cells with cancer gene mutations can survive is by piggybacking on the cooperative gene. In either case, there does seem to be some added growth benefit from having more cells with mutations in cancer genes (figure 25b).

The fraction of cell clusters with at least two cooperative cells (figure 25a), the average number of different cell types per patch (figure 26a), the total number of different

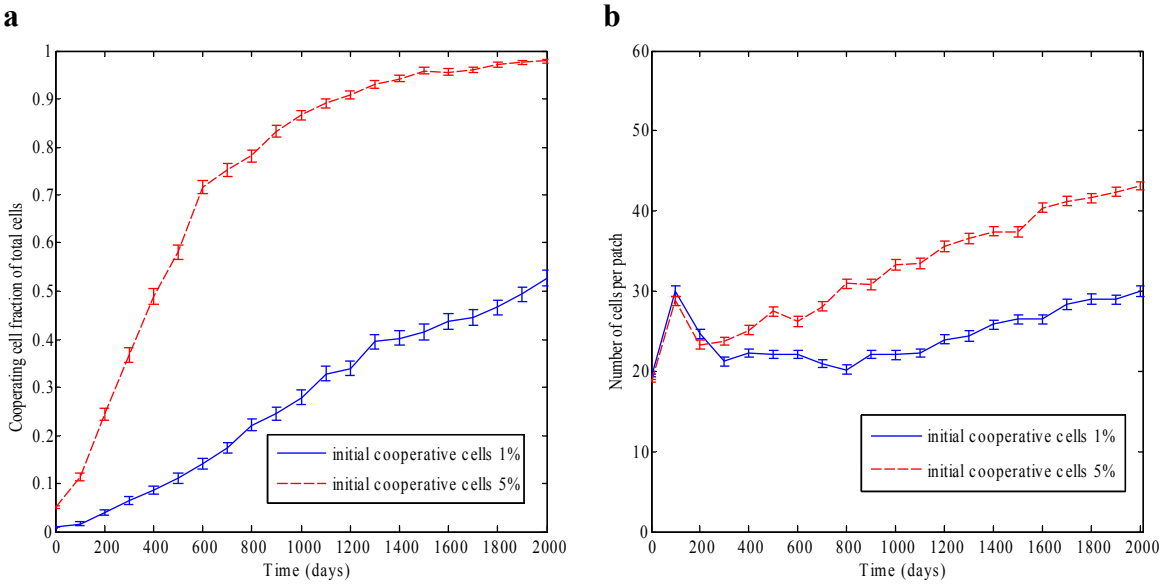

Figure 23. Cooperative cell fraction of total cell population and average number of cells per patch as a function of time for simulations with 100-fold increased mutation rate (with perturbation, and without disturbance). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1% and 5% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Cooperative cell fraction of total cell population. (b) Average number of cells per patch.

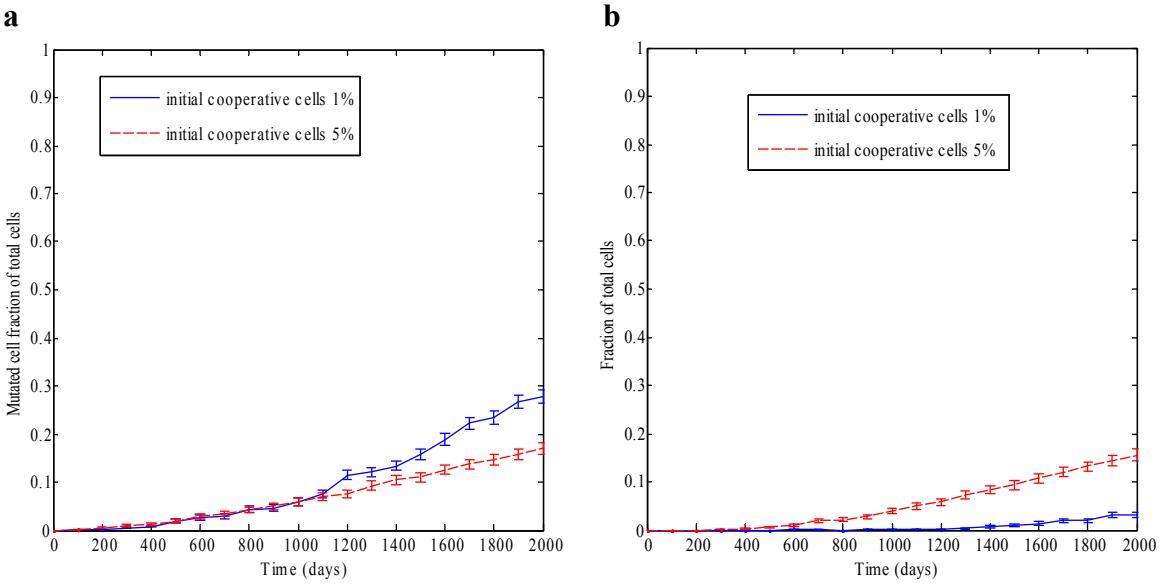

Figure 24. Fraction of cells with mutations in tumor suppressors or oncogenes and with both mutations in cancer genes and cooperative ability as a function of time for simulations with 100-fold increased mutation rate (with perturbation, and without disturbance). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1% and 5% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Fraction of cells with mutations in tumor suppressors or oncogenes. (b) Fraction of cells with both mutations in cancer genes and cooperative ability.

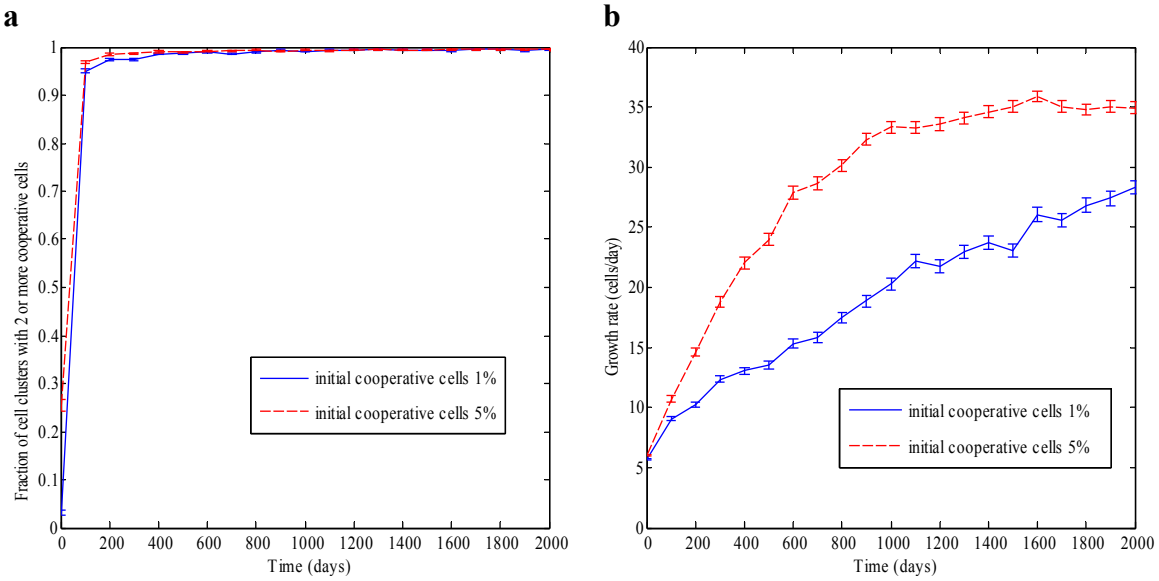

Figure 25. Fraction of cooperative cell clusters with at least two cooperative cells and average patch growth rate as a function of time for simulations with 100-fold increased mutation rate (with perturbation, and without disturbance). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1% and 5% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Fraction of cooperative cell clusters with at least two cooperative cells. (b) Average patch growth rate.

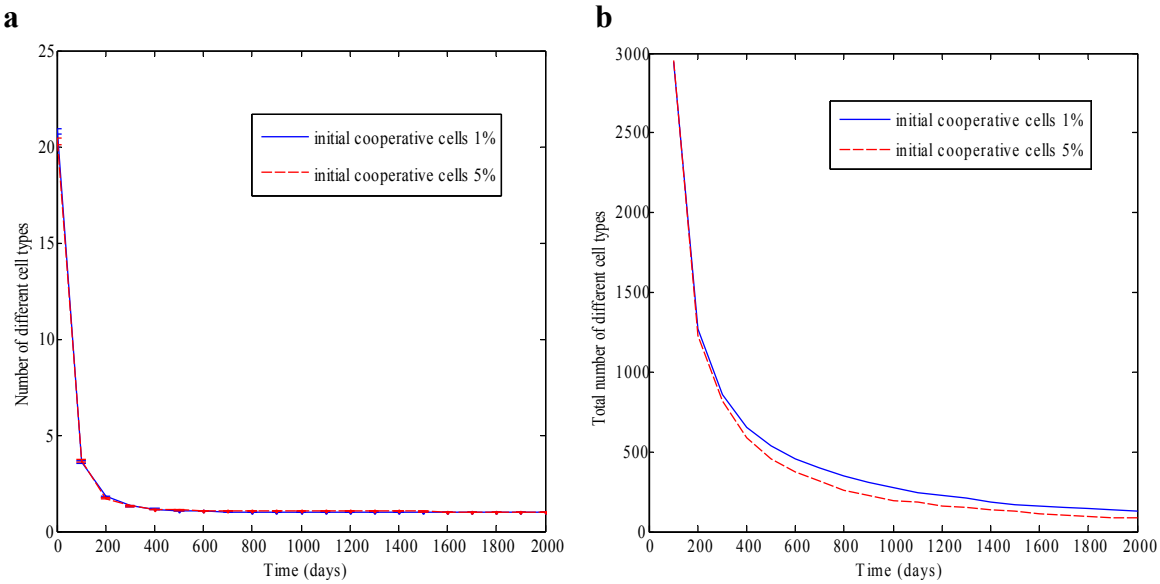

Figure 26. Average number of different cell types per patch and total number of different cell types in the entire space as a function of time for simulations with 100-fold increased mutation rate (with perturbation, and without disturbance). The results ± S.E.M. for simulations with initial cooperative cell fractions of 1% and 5% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Average number of different cell types per patch. (b) Total number of different cell types in the entire space.

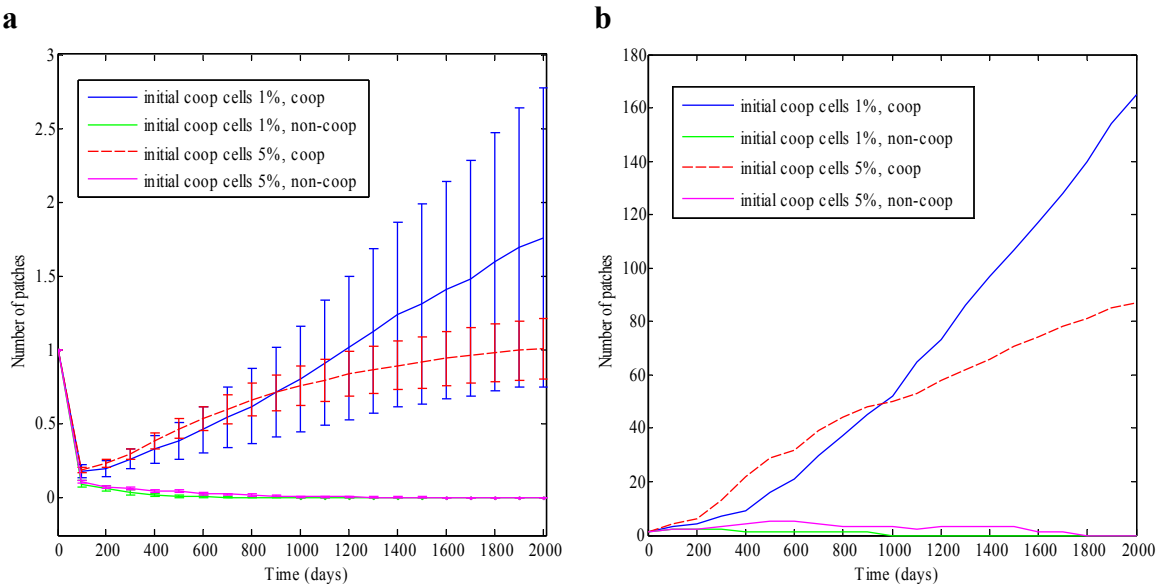

Figure 27. Average and maximum number of patches in which cooperative and non-cooperative cells and their progeny are present as a function of time for simulations with 100-fold increased mutation rate (with perturbation and without disturbance). Simulations with initial cooperative cell fractions of 1% and 5% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Coop refers to the initial cooperative cells and non-coop to the non-cooperative cells. (a) Average number of patches in which the cell cohorts are present ± S.E.M. (b) Maximum number of patches in which the cell cohorts are present.

cell types in the entire space (figure 26b), and the average and maximum number of patches in which the tracked cooperative and non-cooperative cells, and their descendants, are present as a function of time (figure 27) show no substantial difference compared to all the other simulations already discussed.

Since in all the simulations so far the cooperative cells had a head start over the cells with mutations in tumor suppressors and oncogenes, we wanted to see what would happen to the relative abundance of cooperative cells and cells with cancer gene mutations if both started out on an even playing field. As a result, we ran simulations without any initially cooperative cells or cells with cancer gene mutations for the following three situations: with perturbation and no disturbance, no perturbation and no disturbance, and with perturbation and disturbance having $d = 1$ *event/100 days* and p_{kill}

= 10%. The cells can then acquire mutations in the tumor suppressor genes, oncogenes, and the cooperation gene, which would give them the cooperative ability. We used the increase mutations rate 1.82×10^{-5} mutations per gene per cell division to allow for both types of cells to emerge fairly rapidly.

Figure 28a shows the cooperative cell fraction and figure 28b shows the fraction of cells harboring mutations in any of the tumor suppressors or oncogenes in the model. We can see that for the situation with perturbation and no disturbance, the cells with mutations in cancer genes outcompete the cooperative cells; the cells with cancer gene mutations ultimately reach about 50% of the total cell population, while the cooperative cells reach only 2% of the cell population and show very stagnant growth. For the situation with no perturbation and no disturbance, the cells with cancer gene mutations and the cooperative cells are able to coexist; they both reach about 10% of the total cell population. Finally in the situation with perturbation and with disturbance, the cooperative cells outcompete the cells with cancer gene mutations; the cooperative cells reach about 7% of the total cell population, while the cells with mutations in cancer genes almost completely disappear. This also demonstrates that, although the cooperative cells outcompete the cells with cancer gene mutations, the presence of periodic disturbance does limit the cooperative cell expansion. As before, the vast majority of the cells with mutations in the tumor suppressors or oncogenes have a mutation in oncogene 2

Figure 28c shows the fraction of cells that have both the cooperative ability and a mutation in a cancer gene. For the cases with no perturbation and no disturbance and with perturbation and with disturbance, there are virtually no cells that have both properties. On the other hand, for the case with perturbation and without disturbance, nearly all the

cooperative cells at the end also have a mutation in a cancer gene. This points to the difficulty that the cooperative cells have in competing outright with the cells harboring cancer gene mutations under those conditions.

We wanted to know whether the ability of the cooperative cells to coexist with (as in the case without perturbation and without disturbance) or outcompete (as in the case with perturbation and with disturbance) the cells with cancer gene mutations was a result of their increased competitiveness under those conditions or whether the environmental conditions simply stifled the growth of the cells with cancer gene mutations. To investigate this, we completely eliminated cooperation, such that the cells could not acquire a mutation in the cooperative gene, and then ran simulations for the following two scenarios: without perturbation and without disturbance, and with perturbation and with disturbance. We again used the 100-fold increased mutation rate and for the simulations with disturbance we used $d = 1\ event/100\ days$ and $p_{kill} = 10\%$. Figure 28d shows the fraction of cells with mutations in any of the tumor suppressors or oncogenes for these simulations. We can see that in the case with perturbation and with disturbance, the environmental conditions are prohibitive to the growth of the cells with mutations in cancer genes, rather than direct competition from the cooperative cells. However, in the case without perturbation and without disturbance, the cells with mutations in cancer genes demonstrate increased proliferative capacity in the absence of cooperative cells. This indicates that the environmental in this case likely improve the competitiveness of the cooperative cells.

Figure 29 shows the average number of different cell types per patch and the average patch growth rate for the same simulations as above in which the cells could only

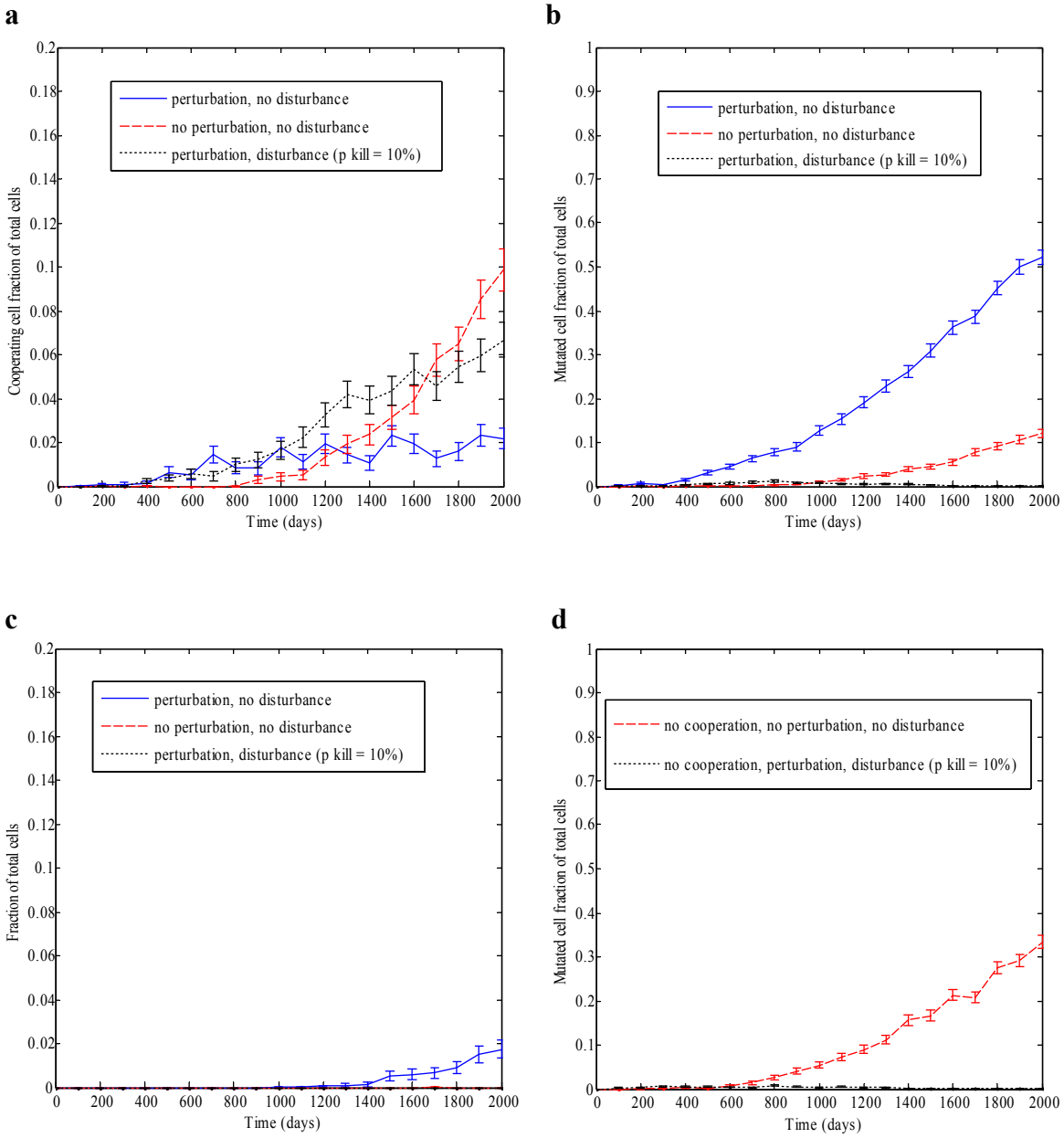

Figure 28. Simulations with 0 initial cooperative cells, 0 initial cells with mutations in tumor suppressors or oncogenes, and 100-fold increased mutation rate. Results ± S.E.M. are shown. Each curves represents one simulation. (a) Cooperative cell fraction of total cell population. (b) Fraction of cells with mutations in tumor suppressors or oncogenes. (c) Fraction of cells with both mutations in cancer genes and cooperative ability. (d) Fraction of cells with mutations in tumor suppressors and oncogenes for simulations without cooperation.

attain cooperative ability through mutation. Both the average cells per patch and the patch growth rate are reduced compared to the simulations in which cooperative cells were introduced initially. This testifies to the ability of the cooperative cells to confer a significant growth advantage to tissue.

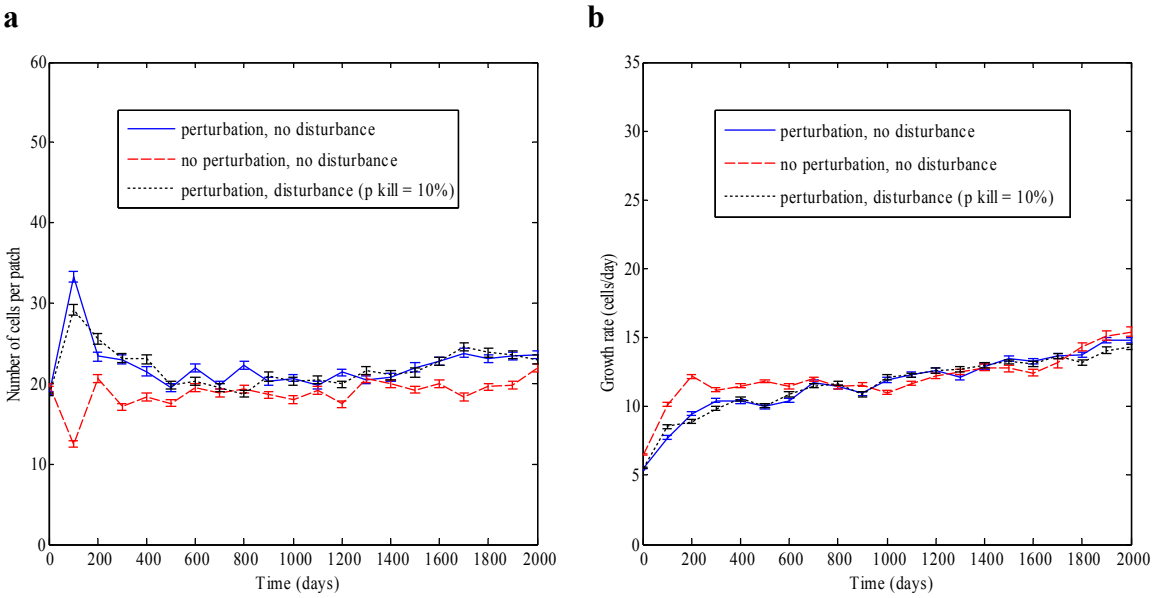

Figure 29. Average number of cells per patch and average patch growth rate as a function of time for simulations with 0 initial cooperative cells, 0 initial cells with mutations in tumor suppressors or oncogenes, and 100-fold increased mutation rate. Results ± S.E.M. are shown. Each curves represents one simulation. (a) Average number of cells per patch. (b) Average patch growth rate.

i) How does the elimination of autocrine cooperation affect the dynamics of the system?

In all the simulations so far discussed, each cooperative cell had the ability to increase its own fitness through autocrine cooperation. As a result, the dynamics observed could be a result of the autocrine effects rather than true cooperation. To eliminate this confounding factor, we ran simulations of the model without autocrine cooperation. We ran simulations with an initial fraction of cooperative cells of 1%, 5%, and 16% of the

total initial cell population, but without perturbation. We ran simulations only for the "give cooperation" model. Each simulation was 2000 days long, with data recorded every 100 days, and cells could acquire mutations in the cooperative gene, if they do not already have cooperative capabilities, or in the tumor suppressors and oncogenes that are built into the model. We used $M = 50$ X $N = 20$ patches and the baseline gene mutation rate of 1.82×10^{-7} mutations per gene per cell division. Each simulation experienced perturbation but no disturbance.

We see that the cooperative cell fraction of the total cell population (figure 30a) and the average number of cells per patch as a function of time (figure 30b) are very similar to the cases with autocrine cooperation. In addition, the fraction of cell clusters with at least two cooperative cells (figure 31a), the average patch growth rate (figure 31b), the average number of different cell types per patch (figure 32a), the total number of different cell types in the entire space (figure 32b), and the average and maximum number of patches in which the tracked cooperative and non-cooperative cells, and their descendants, are present as a function of time (figure 33) all are very similar to the cases with autocrine cooperation. It appears that the dynamics observed are largely a result of cooperation rather than simple selection of autocrine behavior.

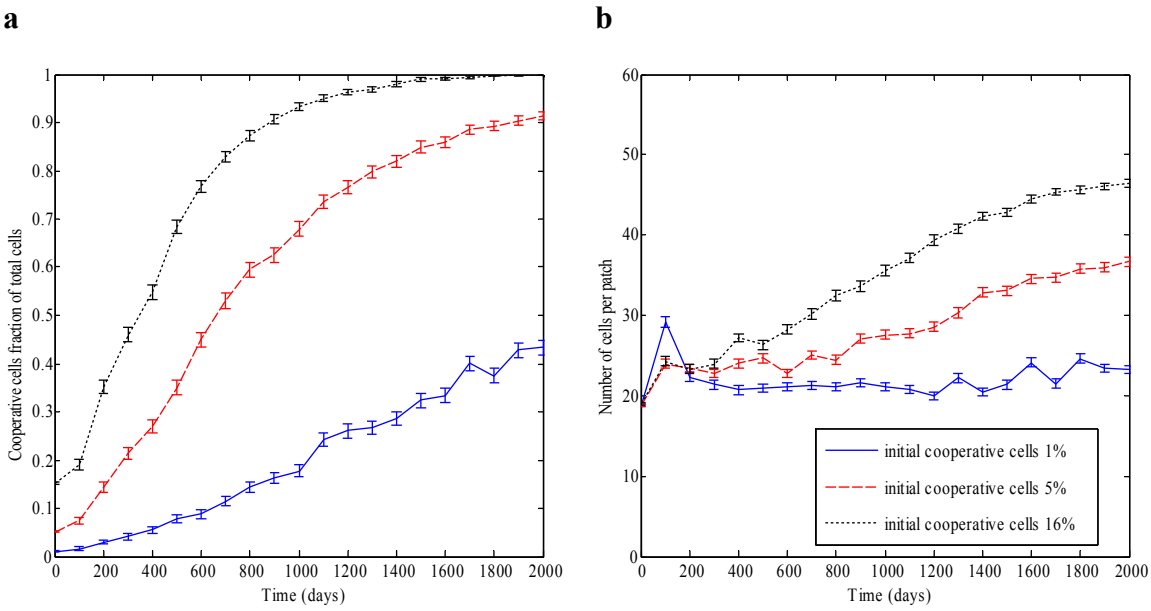

Figure 30. Cooperative cell fraction of total cell population and average number of cells per patch as a function of time for simulations without autocrine cooperation. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Cooperative cell fraction of total cell population. (b) Average number of cells per patch.

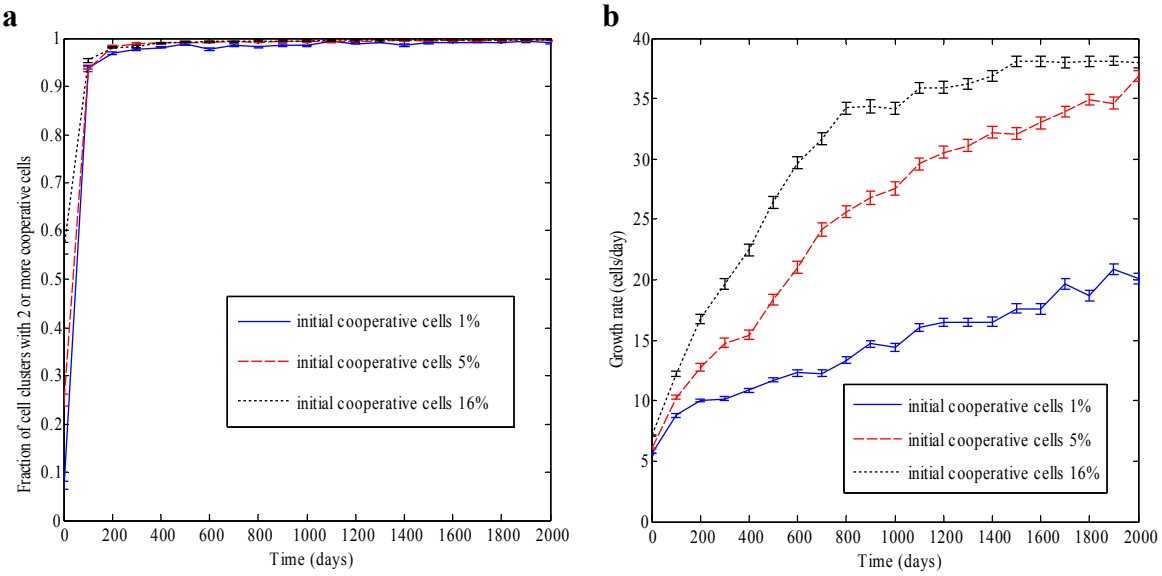

Figure 31. Fraction of cooperative cell clusters with at least two cooperative cells and average patch growth rate as a function of time for simulations without autocrine cooperation. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Fraction of cooperative cell clusters with at least two cooperative cells. (b) Average patch growth rate.

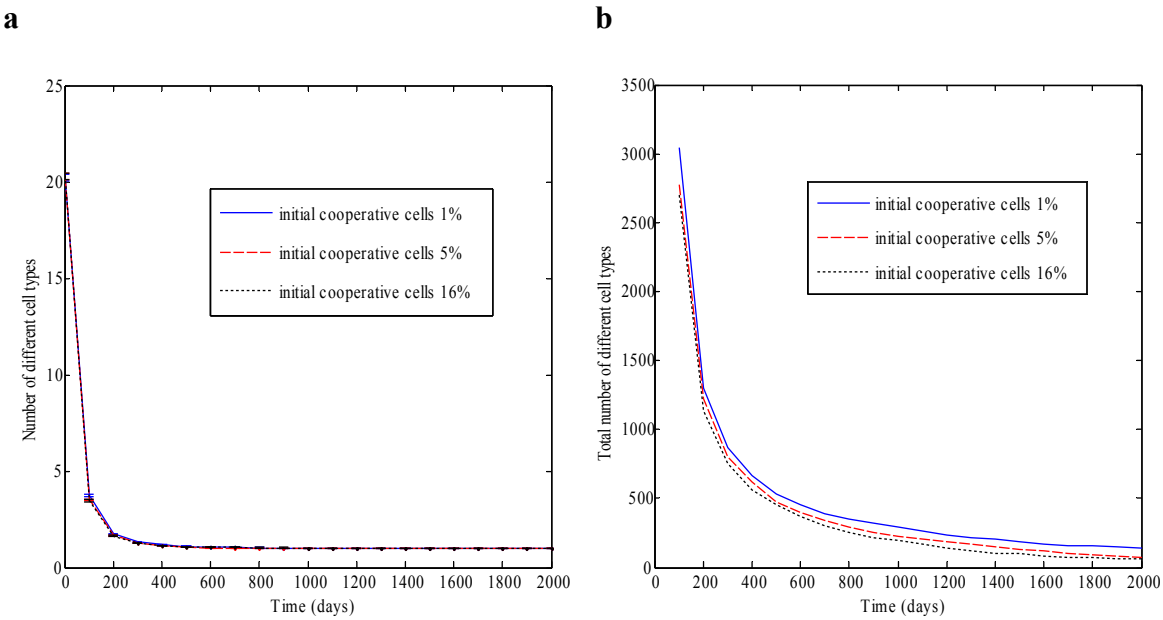

Figure 32. Average number of different cell types per patch and total number of different cell types in the entire space as a function of time for simulations without autocrine cooperation. The results ± S.E.M. for simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Each curve represents one simulation. (a) Average number of different cell types per patch. (b) Total number of different cell types in the entire space.

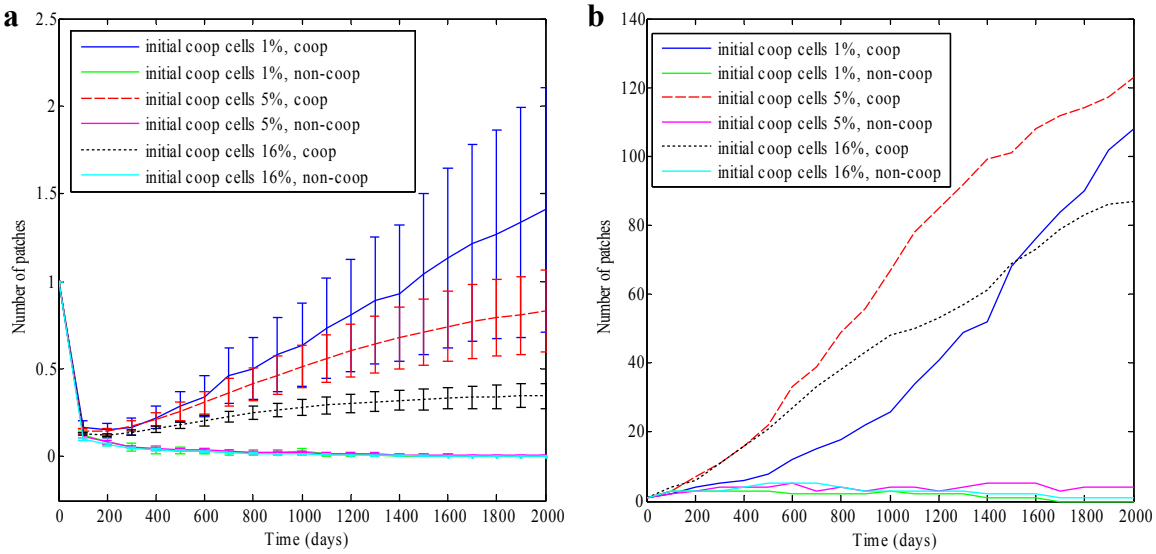

Figure 33. Average and maximum number of patches in which cooperative and non-cooperative cells and their progeny are present as a function of time for simulations without autocrine cooperation. Simulations with initial cooperative cell fractions of 1%, 5%, and 16% are shown for the "give cooperation" model, in which a cooperative cell can send out a cooperative signal that can be picked up by its immediate neighbors and itself. Coop refers to the initial cooperative cells and non-coop to the non-cooperative cells. (a) Average number of patches in which the cell cohorts are present ± S.E.M. (b) Maximum number of patches in which the cell cohorts are present.

IV. Discussion

a) Conclusions

A major aim of this study was to investigate the potential role of cooperation in early tumor progression and to understand some of the general principles underlying cooperation. We have shown that cooperation is a very robust property. Once a small number of cooperative cells is introduced into a cell population, even as little as 1% of the total cell population, they rapidly proliferate to the point of being the major constituent of the cell population. Furthermore, in the model cooperation leads to an increased growth rate of the entire cell population, with the growth rate rising in parallel with the cooperative cell fraction. Even though the presence of cooperation leads to an increase in the total number of cells in the environment, the rate of increase in the growth rate is still greater than the rate of increase in the total cell number. Therefore, cooperation confers a significant growth advantage apart from simply generating more cells that can grow. Interestingly, cooperation does not seem to have an effect on cell heterogeneity, counter to what we initially suspected. We hypothesized that since cooperation increases the fitness of all the cells within range of the cooperative signal, different cells would be able to coexist in local neighborhoods. However, cooperative cells are so robust that they tend to outcompete the cells in their neighborhood, thus homogenizing the space. We also observed a dose-response relationship between the number of initially cooperative cells and some of the important variables; the greater the number of cooperative cells introduced in the beginning, the faster the rise in the cooperative cell fraction, growth rate, and total cell number.

We were also able to make several key observations about the dynamics of

cooperation within the model. Cooperative cells tend to cluster together with time, such that each cooperative cell generally has at least one other cooperative cells as its immediate neighbor. This holds true even when the cooperative cells are initially randomly distributed throughout the available space. In fact, with enough time, most cooperative cells reside in cell clusters that have multiple cooperative cells. This creates an environment of mutualistic cooperation, in that virtually every cooperative cell has a cooperative neighbor with which it can engage in a mutually beneficial interaction. We also found that cooperative cells have a wider spatial influence than non-cooperating cells. The cooperative cells or their descendant are, on average, present in more patches than corresponding non-cooperative cells at each point in time. The disparity is even larger when we compare the most competitive cooperative and non-cooperative cells.

The robustness of the cooperative property is further evident by the fact that all the above observations hold true in multiple conditions. There is a similar pattern when all the cells have the same intrinsic fitness levels (in the absence of perturbation), when the cells all have different intrinsic fitness levels (in the presence of perturbation), and even when all the cells have different intrinsic fitness levels in the presence of disturbance that kills a certain fraction of the cells in the environment at different time points. Furthermore, in the case with cells having different intrinsic fitness levels, we observed essentially the same results for both types of cooperation strategies investigated in the study, *i.e.* whether cooperation entails a cell acquiring the ability to send or the ability to receive a cooperative signal. The model also produces similar results whether or not we allow for autocrine cooperation, *i.e.,* a cell being able to increase its own fitness through the cooperative signal it sends out. The fact that the above dynamics hold true

even without autocrine cooperation shows that the cooperative property itself, and not simply autocrine effects, is largely responsible for the observed behavior.

We were also interested in understanding the relationship between cooperative cells and cells harboring mutations in tumor suppressor genes or oncogenes in various environments. We accomplished this by running simulations with 100-fold increased mutation rate, thus creating a "mutator phenotype," which allowed a substantial number of new mutations to arise in tumor suppressor genes and oncogenes, as well as the cooperative gene. When cooperative cells are present in the environment in the beginning, the cooperative cells are able to coexist with the cells that eventually acquire mutations in tumor suppressors and oncogenes; there is a steady increase in both the fraction of cooperative cells and the fraction of cells harboring mutations in tumor suppressors or oncogenes. However, once the fraction of initially cooperative cells reaches a certain threshold, mutations in tumor suppressors and oncogenes can survive and spread only if those mutations occur in cells with cooperative ability. In essence, if the cooperative cells gain a significant head start, the only way the cells with cancer gene mutations can survive is by piggybacking on the cooperative gene. As expected, the genotype (tumor suppressor or oncogene mutation) that confers the greatest competitive advantage is the one that provides the highest ratio of replication increase to apoptosis reduction.

In the scenario when there are no initial cooperative cells, such that the cooperative cells and cells with cancer gene mutations start out on an even playing field, the balance between these cell types is determined by the presence or absence of perturbation and disturbance. In the case when all the cells have the same intrinsic fitness

levels (in the absence of perturbation), the cooperative cells and the cells with cancer gene mutations are able to coexist. If the cells all have different intrinsic fitness levels (in the presence of perturbation), the cells with cancer gene mutations outcompete the cooperative cells and become the dominant cell group in the population. Finally, in the case when all the cells have different fitness levels and there is periodic disturbance that kills a certain fraction of cells, the cooperative cells fare much better than the cells with cancer gene mutations; while the cooperative cells survive and grow, the cells with cancer gene mutations virtually disappear. Interestingly, in the scenarios in which the cooperative cells are able to either coexist with or outcompete the cells with cancer gene mutations, the specific reason why this occur also depends on the environmental conditions. For example, in the case of all cells having uniform intrinsic fitness levels, the cooperative cells are able to coexist with the cells harboring cancer gene mutations because the uniform fitness landscape renders the cooperative cells more competitive. On the other hand, in the case when the cells have different fitness levels and there is periodic disturbance, the environmental conditions themselves are prohibitive to the growth of the cells with mutations in cancer genes, rather than direct competition from the cooperative cells. The cooperative cells are able to better deal with disturbance and can survive in this type of environment.

The above results have implications for the time frames during tumor progression that cooperation plays the most significant role. Each environmental condition is supposed to mimic different stages in tissue life and in tumor progression. The scenario in which all cells have uniform fitness levels models young tissue, before significant mutations have accumulated that would differentiate cell fitness levels. As a result, if

cooperation evolves in this stage, the cooperative cells would be able to compete with cells that develop mutations in cancer genes at this very early pre-tumor stage. The scenario in which the cells have different fitness levels models slightly older tissue, one that has acquired enough mutations to make the cells slightly distinct. The model predicts that cooperation is unlikely to first arise at this stage due to the more intense competition from the cells with cancer gene mutations in this environment. However, if cooperative cells are already present at this stage, *i.e.* they got a head start by arising at an earlier more favorable stage, then they are able to coexist alongside the cells with cancer gene mutations. Cooperation, much like other factors involved in cancer, depends on the interplay of genetics, environment, and chance.

Although our model focuses mainly on the early tumor stages, the scenarios with disturbance can be extended to events that occur in later stages of tumor progression, such as the necrosis that occur in rapidly expanding tumors or the massive cell death in tumors undergoing treatment. If we make this extrapolation, then the model predicts that cooperation plays a very important role at these later stages of tumor progression. The cooperative cells seem to be more robust than the cells that have mutations in tumor suppressor genes or oncogenes, and may be key players in getting a tumor past these difficult times in its life history. Furthermore, disturbance arises in many other instances related to cancer, such as inflammation and chronic toxic exposure that leads to cancer. Consequently, cooperation may play a contributory role in colon cancer arising in inflammatory bowel disease, hepatocellular carcinoma arising in a cirrhotic liver or lung cancer arising in a smoker.

b) Clinical applications: cancer prevention and therapeutic strategies

The ultimate aim of any modeling endeavor in medicine or biology, including this project, is to generate insight into disease processes that will help guide future basic science and clinical applications. The insight into tumor development resulting from the current work bears implications for two important clinical issues, that of cancer prevention and treatment strategies.

Early detection for the sake of preventing advanced cancer holds much promise for revolutionizing this onerous disease. Despite significant research and advancements in cancer therapy, the survival rates for those diagnosed with advanced cancer have not significantly improved over the last few decades (56). However, it is almost universally true that if cancer is detected at an early localized stage, the chance for long-term survival is extremely good, with 5-year survival rates exceeding 90% in many cancer types. As a result, being able to diagnose tumors when they are localized, or better yet before they are even apparent grossly or histologically, can have a drastic impact on the long-term burden of cancer on individuals and society as a whole. The effect of Pap smear screening on reducing the mortality burden from cervical cancer is one of the most striking examples.

A recent study on the early detection of ovarian cancer, one of the cancers that currently offers little in the way of early detection, showed that early-stage (up to stage II) serous ovarian tumors have a median diameter less than 3 mm (57). This news is particularly sobering since most of the studies evaluating ovarian cancer biomarkers have focused on clinically apparent tumors (58-60), which are typically an order of magnitude larger than the average early-stage ovarian tumor. The necessity for reliable biomarkers

associated with very early cancer stages is pressing.

As we have seen, one of the predictions of our study is that cooperation could potentially play a significant role in the early pre-tumor stage, at a time when tissue looks grossly, or even histologically, normal. This means that if there were a reliable way to detect the cooperative signal emitted by the cooperative cells, then this could form the basis for a very early cancer detection strategy, one that can detect pre-cancerous states, and allow for cancer prevention. However, there are several barriers than first need to be overcome. First, the role of cooperation in cancer would need to be firmly established through experimental data (we will explore this in more detail in the next section). Second, there would need to be an accurate detection method that is able to pick up cooperation-associated biomarkers, potentially at very low biomarker levels. Third, there would need to be a detectable difference in the cooperation-associated biomarkers between normal subjects and those with pre-cancerous lesions. Finally, cooperation-associated biomarkers specific to each pre-cancerous lesion would need to be identified.

The search for effective cancer treatment serves a complementary role to cancer prevention in the battle against cancer. One of the challenges of developing potent cancer therapy is the inherent ability of cancer cell populations to evolve in the face of selective pressure imposed by treatment. A cancer therapy that does not kill all cancer cells simply creates a selective pressure for resistant cell populations to evolve (22). Research into tumor robustness concludes that a potentially effective therapeutic strategy against cancer is to use combination therapy that targets different robust mechanisms employed by tumors (13). This strategy aims to induce fragility in a system by specifically targeting some of its robust features. Our research suggests that cooperation may be a crucial

robust mechanism used by pre-cancerous or cancerous tissue to overcome disturbance, or periods of massive cell death. Treatment is an important example of disturbance. Hence, using a therapy that disrupts the cooperative ability can weaken a tumor's ability to deal with disturbance and make the entire cancer cell population more vulnerable to other cytotoxic agents.

We have shown that cooperative cells are able to outcompete cells with mutations in oncogenes and tumor suppressors in the presence of disturbance. An interesting treatment strategy that evolves from this result is the concept of using therapeutic cooperative cells to outcompete cancer cells at times of high disturbance, such as when a tumor outgrows its blood supply or is undergoing treatment by a cytotoxic agent. A similar idea was first introduced by Deisboeck and Wang. They found through mathematical modeling and simulations that a group of fast-growing therapeutic cells can lead to the extinction of tumor cells with a lower proliferative rate (61). This ecologically-driven concept is attractive since is it turns a tumor's aggressiveness against itself. However, it may also lead to potential problems since the fast-growing therapeutic cells may be difficult to control. To avoid this possible complication, using cooperative cells, rather than simply fast-growing cells, may be a better alternative. Cooperative cells can be engineered to have cooperative ability that can be easily turned on and off. Thus, after completing their job of vanquishing the tumor cells, the cells' cooperation can be silenced and their competitiveness drastically reduced.

c) Future directions

The modeling approach to biological systems is a very powerful tool in

understanding the underlying dynamics of the systems. However, it certainly has its limitations. Modeling involves making certain assumptions about the structure of a system that is modeled. We have tried to make these assumptions as accurate and biologically relevant as possible by using known data and facts about the structure of tissue, cell characteristics, and cell interactions to build our model. Unfortunately, our knowledge of these variables is not perfect. Modeling also requires the determination of various parameters that are built into the model. We have tried to use existing experimental data to parametrize our model. However, again we are limited by the data that is currently available. Our model predicts that cooperation can be a strong force in driving tumor progression, but cannot say for certain whether this happens in reality. Cooperation may represent an optimal or near-optimal strategy. However, it does not mean that pre-cancerous or cancerous tissue actually uses such a strategy. Finally, our modeling is also constrained by the available computing power. As a result, we were only able to run a single simulation for each of the situations modeled. To allow for statistical analysis to be performed on each simulation, we assumed that the 1000 patches in each simulation are independent and thus were able to calculate the standard deviation and standard error of the mean. We realize that because of patch fission this assumption is not correct. However, due to the spatial constraints, enough groups of patches are independent that each simulation produces a significant number of independent data points. Even though our statistical technique results in an estimate of the S.E.M. that is smaller than the actual S.E.M., it still allows for an appreciation of the statistical significance of the results. The instance where the reliance on single simulations runs poses the biggest challenge is in the simulations with 100-fold increased mutation rate, in

which there are no initially cooperative cells or cells with cancer gene mutations. In those cases, the dynamics are driven by relatively rare mutations in a small number of cells, and are thus subject to variability between different simulation runs.

Our simulations did not show any phase transitions, or abrupt non-linear changes, in any of the variables studied; all the variables changed continuously and smoothly with time. Such phase transitions would have indicated a pre-malignant tissue's transformation to a full-fledged tumor. This speaks to another limitation of our study. Since we focused on the early stages of tumor progression, we omitted many important components that determine cancer progression, such as angiogenesis, tissue invasion, and metastasis. We realize that because of this, our model cannot make general predictions about cancer as a whole. It is our hope that the inclusion of other important components of cancer into future modeling endeavors would allow for a more complete understanding of this disease. The rest of the time, we will discuss some of the experimental evidence that suggests the existence of cooperation between epithelial cells in partially transformed tissue and offer some direction for future research.

As already mentioned in the introduction, there is strong evidence for cooperative relationships between epithelial and stromal cells in cancer. However, there is also some experimental evidence for cooperation between partially transformed epithelial cells, which is the the subject of our work. Miller *et. al.* showed that different mouse tumor cell populations have the potential to augment each others' growth when injected into opposite flanks of a mouse (62). They further showed that this is most likely immunologically mediated. Jouanneau *et. al.* demonstrated that a tumor cell line able to produce acidic fibroblast growth factor can confer tumorigenicity and metastatic ability

to a cell line that does not produce the growth factor but has receptors for it (63).

Despite the already existing evidence of cooperation between epithelial cells in cancer, much more experimental research still needs to be done to better characterize such cooperative interactions. To start, we would need more confirmation of the ubiquity of cooperation in cancer. If cooperation is indeed ubiquitous, then several important questions would need to be answered. Is cooperation mediated through a diffusible substance, cell to cell contact, or some other means? Is the cooperative mechanism the same or similar across different cancer types? What are the genetic or epigenetic changes that are responsible for cooperation? Does cooperation arise when cancer cells coopt already established cellular circuits designed to carry out a biologically relevant cooperation program (similar to the situation with tumor suppressor genes and oncogenes)? Finally, the predictions made by out model would need to be confirmed experimentally as well. This is by no means an exhaustive list of questions that need to be answered or issues that need to be addressed. We hope this is only the beginning of the investigation into the role of cooperation in cancer. Gaining a better understanding of cell interactions in cancer holds the promise of more rational drug design, improved cancer diagnosis, and a deeper appreciation of the complexities of this inexorable disease.

Appendix – Model Code in MATLAB

```
function [crypts_pos, alpha, give_coop, mutation, mutation2, num_mut,
num_mut2, num_mut_coop, T_kill] = Model_give_coop(num_coop_start)

%MATLAB code for tumor development model that incorporates cooperation
and
%colonic spatial structure
%Cell can give the cooperative signal to other nearby cells only if it
%has acquired an appropriate mutation

%num_coop_start = # of cells that are able to give coop signal at the
%start of simulation

%global variables
global M N dim R a stem_cells

CC = 64;   %carrying capacity of crypt (Yatabe, Shibata. PNAS, 2001)
R = 1.14286;  %basal replication rate when nearby crypts are empty
(Potten, Taylor. Int. J. Radiat. Biol. (1998):54)
a = 0.025;  %basal apoptosis rate in units 1/day (Calabrese, Shibata.
American Journal of Pathology, 4/2004, Vol.164/No.4)
b = 5.6;  %weight assigned to diversity in determining robustness-
scalar, no dimensions (Maley. Nature Genetics, 4/2006, Vol.38/No.4)
c = 1.9;  %weight assigned to net growth in determining robustness
(Maley. Nature Genetics, 4/2006, Vol.38/No.4, figure 3)
total_t = 2001;  %total run-time of simulation
dt = 1;  %sampling time interval
t = dt:dt:total_t;  %time vector
num_empty = 0;  %# of empty neighboring cell sites

%disturbance matrix; first row will be times of onset of each
disturbance,
%second row will be duration of each disturbance, third row the
intensity
%of each disturbance, and the fourth row will indicate if the
disturbance
%is mutagenic (1 if yes, 0 if no)
%disturb = [0;total_t;1;0];
disturb = [0;0;1;0];

%time at which a certain fraction of cells will be killed
%T_kill = round(total_t * rand(floor(total_t/20),1));
%T_kill = unique(sort(T_kill));
T_kill = 0;

%probability that certain cell will be killed (or fraction of cells to
be
%killed)
p_kill = 0.1;

%percentage of essential genes in genome; will be used to determine if
%mutation should trigger apoptosis/necrosis
%any mutation in an essential gene will trigger apoptosis/necrosis
%(Tong, Evangelista. Science (2001): 294)
```

```
p_essential = 0.17742;

%dimensions of crypt locations
M = 50;  %x dimension
N = 20;  %y dimension

%row and col dimension of each crypt plus 2 placeholder spots
dim = floor(sqrt(CC)) + 2;

%probability distribution for the # of stem cells in each crypt (from
%Shibata)
%first row is the # of stem cells & second row is the probaility of
having
%that stem cell # (probability density) (Nicolas, Shibata. PLoS
Computational Bilogy, 3/07, Vol.3/No.3)
stem_cells = [6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30;
                0.000143 0.002 0.00486 0.01114 0.02 0.02571 0.038
0.04229 0.04857 0.05143 0.056 0.05771 0.05886 0.05657 ...
                0.05486 0.05314 0.05343 0.05086 0.04971 0.04743 0.04486
0.04314 0.04229 0.03886 0.03829];

%state variables that determine whether disturbance is new or on-going
and
%give index of current (if disturbance is present) or next disturbance
%d_state1 = 1 if disturbance is on-going, 0 if it is new or no current
%disturbance; d_state2 gives index of current disturbance or next
%disturbance
d_state1 = zeros(M,N+2,dim,dim);
d_state2 = ones(M,N+2,dim,dim);

%dose-dependent gene mutation rate
%columns correpsond to disturbance intensity
%first column corresponds to baseline (intensity = 0)
%(Spencer, Gerety. PLoS Computational Biology (2006), Vol.2/No.8)
%gene_mut = 1/(5.5 * 10^6);
gene_mut = 1/(5.5 * 10^4);

num_genes = 25000;  %# of genes in human genome
mut_rate = gene_mut * num_genes;  %mutation probability per replication

%dose-dependent mean distrubance effect on replication rate
(distribution
%of this factor will be a normal distribution with the given mean)
%(Pappa. Mol. Nutr. Food Res. (2007): 51, table 1, disturbance doses
correspond to 6, 12, 24, and 48 hours)
C_rmean = [0.97382 0.8377 0.8377 0.75393];
%dose-dependent standard deviation of effect of disturbance on
replication
%rate
sd_r = [0.18325 0.1623 0.0733 0.18325];

%dose-dependent mean disturbance effect on apoptosis rate (distribution
of
%this factor will be a normal distribution w/ given mean for each
%disturbance intensity)
C_amean = [2.34146 6.7561 9.80488 11.09756];
```

```
%dose-dependent s.d. of effect of disturbance on apoptosis rate
sd_a = [0.60976 2.14634 2.07317 1.2439];

%initialize matrix of replication and apoptosis multiplicative factors
C_r = zeros(M,N+2,dim,dim);
C_a = zeros(M,N+2,dim,dim);

%multiplicative effect of a mutation in a given gene on replication
rate;
%the genes are indexed in the order APC, KRAS, P53, TGFBR2, and
%beta-catenin; row vectors
%APC +/-: Katdare. Annals NY Acadaemy of Sciences (2001): 952.
%APC -/-: Herrero-Jimenez. Mutation Research (1998): 400.
%ras: Arends. American Journal of Pathology (1994): Vol.144/No.5.
%TGFBR2: Costa. PNAS (2002): Vol.99/No.20.; Bismas. Cancer Research
(2004): 64.
%beta-catenin: Wagenaar, Crawford. Cancer Research (2001): 61.
R_gene = [2.02041 13.83495 1 1.399 1.4];
R_gene2 = [3.95959 1 1 1.01 1];  %effect of mutation in second allele
of each gene on replication

%multiplicative effect of a mutation in a given gene on apoptosis rate;
%the genes are indexed in the order APC, KRAS, P53, TGFBR2, and
%beta-catenin
%P53: Fazeli. PNAS (1997): 94.
a_gene = [1 3.16667 1 1 0.24324];
a_gene2 = [4.4 1 0.35714 1 1];  %effect of mutation in second allele of
each gene on apoptosis

%matrix that represents the presence or absence of mutations in the
first
%allele of each gene under question -- 1 if mutation is present, 0 if
absent;
%first two indices represent the crypts, third index represents
%the gene involved, in same order as above, last two indices the
positions
%within crypt
mut = zeros(M,N+2,5,dim,dim);

%matrix that represents whether there is mutation in second allele of
each
%gene
mut2 = zeros(M,N+2,5,dim,dim);

%matrix that contains whether there is a mutation in first allele of
each gene
%at a given time (recorded every 100 days)
mutation = zeros((floor(total_t/100)+1),M,N+2,5,dim,dim);

%matrix that contains whether there is a mutation in second allele of
each gene
%at a given time (recorded every 100 days)
mutation2 = zeros((floor(total_t/100)+1),M,N+2,5,dim,dim);

%counter for how many new mutations arise in each gene allele
new_mut = zeros(5,1);
new_mut2 = zeros(5,1);
```

```
%recording of how many new mutations arise in each gene allele by each
%time
num_mut = zeros((floor(total_t/100)+1),5);
num_mut2 = zeros((floor(total_t/100)+1),5);

%initialize matrices representing # of cells in each crypt and their
%positions
crypts_ncells = zeros(M,N+2); %matrix representing # of cells in each
crypt

%matrix representing positions of cells in each crypt; first index
denotes the time,
%in units of 100 days, second two indices represent the crypts, last
two indices represent the
%position w/in each crypt plus 2 placeholder rows and columns to
simplify code
%(limited by the carrying capacity), middle index denotes the
replication and
%apoptosis rate, respectively, at each location (if replication and
apoptosis
%rates are both 0, then there is no cell at that location)
crypts_pos = zeros((floor(total_t/100)+1),M,N+2,2,dim,dim);

%represents positions of the cells in each crypt for the current and
previous time period
crypts_pos_old = zeros(M,N+2,2,dim,dim);

%represents positions of the cells in each crypt for the next time
period
crypts_pos_new = zeros(M,N+2,2,dim,dim);

%matrix that represents min and max row and column dimension extent of
cell
%occupancy within a crypt -- first two indices
%represent the crypts, 3rd represents row_min, row_max, col_min,
col_max in
%that order
crypts_min_max = zeros(M,N+2,4);

%potential interaction coefficients that can arise if a cell cell
acquires a
%mutation in the gene responsible for giving the cooperative signal;
%the assumption is that cells are able to receive the cooperative
signals by
%default and it takes a mutation for a cell to provide the cooperative
signal
%the interaction coefficents are uniformly distributed
%the first index represents time in units of 100 days, the next two
indices
%represent the crypts (with two additional columns on the right and
left edges
%as placeholders to simplify code), middle index denotes the
replication and
%apoptosis coefficients, repsectively, and the last two the cell's
position
%within a crypt (top, bottom, left, and right rows and columns are
```

```
placeholders
%and will have coefficients of 0) if there is no cells at a site, the
coefficient
%is 0 at that location
alpha = zeros((floor(total_t/100)+1),M,N+2,2,dim,dim);

%represents interaction coefficients of the cells in each crypt for the
current and
%previous time period
alpha_old = zeros(M,N+2,2,dim,dim);

%represents interaction coefficients of the cells in each crypt for the
next time period
alpha_new = zeros(M,N+2,2,dim,dim);

%fill crypts with cells from Shibata distribution
%length-wise (in the x-dimensions), the right and left edges will
simply be
%place holders to simplify code; the top and bottom edges will be
%considered connected in order to model the colon as a tube
[crypts_ncells, crypts_pos ,crypts_pos_old, crypts_min_max,alpha,
alpha_old] = ...
    initialize_crypts(crypts_ncells, crypts_pos, crypts_pos_old,
crypts_min_max, alpha, alpha_old);

crypts_pos_new = crypts_pos_old;
alpha_new = alpha_old;

%matrix that represents whether a cell has acquired the mutation to give
%cooperative signal-- 0 if no, 1 if yes; if no cell at site, default is
0
give_coop = zeros((floor(total_t/100)+1),M,N+2,dim,dim);

%represents give_coop status of the cells in each crypt for the current
and previous
%time period
give_coop_old = zeros(M,N+2,dim,dim);

%represents give_coop status of the cells in each crypt for the next
time period
give_coop_new = zeros(M,N+2,dim,dim);

%counter for how many new mutations arise in coop gene
new_mut_coop = 0;

%recording of how many new mutations arise in coop gene by given time
num_mut_coop = zeros((floor(total_t/100)+1),1);

%matrix that holds mutation profiles for daughter cells that are vying
to
%fill a given location within a crypt; this procedure allows all cells
to
%have equal opportunity to occupy a vacant spot within a crypt
candidates = cell(dim,dim);

%# of offspring potentially produced by each parent cell in a given
location w/in crypt
```

```
off = zeros(dim,dim);

%initialize some cells to cooperate from beginning of simulation
if num_coop_start > 0
    num_coop_start1 = num_coop_start;
    alpha1 = reshape(alpha_old(:,:,1,:,:),M,N+2,dim,dim);
    cells = find(alpha1);
    if num_coop_start > length(cells)
        num_coop_start1 = length(cells);
    end
    for i = 1:num_coop_start1
        random_cell = ceil(length(cells) .* rand);
        give_coop_old(cells(random_cell)) = 1;
        cells(random_cell) = [];
    end
    give_coop(1,:,:,:,:) = give_coop_old;
    give_coop_new = give_coop_old;
end


%simulation
for i = 1:length(t)
    for j = 1:M
        for k = 2:N+1
            if crypts_ncells(j,k) > 0

                %check if carrying capacity of crypt is full
                if crypts_ncells(j,k) >= CC
                    %split crypt if there are empty neighboring crypts
                    [crypts_ncells, crypts_pos_old(:,:,:,2:(dim-1),2:
(dim-1)), crypts_pos_new(:,:,:,2:(dim-1),2:(dim-1)), ...
                        alpha_old(:,:,:,2:(dim-1),2:(dim-1)),
alpha_new(:,:,:,2:(dim-1),2:(dim-1)), crypts_min_max, ...
                        mut(:,:,:,2:(dim-1),2:(dim-1)), mut2(:,:,:,2:
(dim-1),2:(dim-1)),give_coop_old(:,:,2:(dim-1),2:(dim-1)), ...
                        give_coop_new(:,:,2:(dim-1),2:(dim-1))] = ...
                        split_crypt(crypts_ncells,
crypts_pos_old(:,:,:,2:(dim-1),2:(dim-1)), crypts_pos_new(:,:,:,2:(dim-
1),2:(dim-1)), ...
                        alpha_old(:,:,:,2:(dim-1),2:(dim-1)),
alpha_new(:,:,:,2:(dim-1),2:(dim-1)), crypts_min_max, ...
                        mut(:,:,:,2:(dim-1),2:(dim-1)), mut2(:,:,:,2:
(dim-1),2:(dim-1)),give_coop_old(:,:,2:(dim-1),2:(dim-1)), ...
                        give_coop_new(:,:,2:(dim-1),2:(dim-1)),j,k);
                end

                %loop over cell sites within the crypt
                for l = crypts_min_max(j,k,1):crypts_min_max(j,k,2)
                    for m = crypts_min_max(j,k,3):crypts_min_max(j,k,4)

                        %check for presence of new disturbance
                        if disturb(1,d_state2(j,k,l,m)) <= t(i) <=
(disturb(1,d_state2(j,k,l,m)) + disturb(2,d_state2(j,k,l,m)))...
                            && d_state1(j,k,l,m) == 0  %new disturbance
                            C_r(j,k,l,m) =
C_rmean(disturb(3,d_state2(j,k,l,m))) +
sd_r(disturb(3,d_state2(j,k,l,m))) * randn;
```

```
                            C_a(j,k,l,m) =
C_amean(disturb(3,d_state2(j,k,l,m))) +
sd_a(disturb(3,d_state2(j,k,l,m))) * randn;
                            C_r(find(C_r>1)) = 1;  %disturbance can at
best have neutral effect on replication
                            C_r(find(C_r<=0)) = 0.001;  %if C_r is 0 or
less, reset it to an arbitrarily small #
                            C_a(find(C_a<1)) = 1;  %disturbance can at
best have neutral effect on apoptosis
                            crypts_pos_old(j,k,1,l,m) = C_r(j,k,l,m) *
crypts_pos_old(j,k,1,l,m);
                            crypts_pos_old(j,k,2,l,m) = C_a(j,k,l,m) *
crypts_pos_old(j,k,2,l,m);
                            d_state1(j,k,l,m) = 1;  %disturbance
becomes on-going
                        elseif t(i) > (disturb(1,d_state2(j,k,l,m)) +
disturb(2,d_state2(j,k,l,m))) && d_state1(j,k,l,m) == 1
                            d_state1(j,k,l,m) = 0;  %no disturbance
                            d_state2(j,k,l,m) = d_state2(j,k,l,m) + 1;
%look for next disturbance
                            %change fitness of cells to pre-disturbance
levels
                            if C_r(j,k,l,m) ~= 0 && C_a(j,k,l,m) ~= 0
                                crypts_pos_old(j,k,1,l,m) =
crypts_pos_old(j,k,1,l,m) / C_r(j,k,l,m);
                                crypts_pos_old(j,k,2,l,m) =
crypts_pos_old(j,k,2,l,m) / C_a(j,k,l,m);
                            end
                            C_r(j,k,l,m) = 0;  %return matrix to pre-
disturbance values
                            C_a(j,k,l,m) = 0;
                        elseif d_state1(j,k,l,m) == 1  %disturbance is
on-going
                            %set replication and apoptosis rates for new
                            %cells introduced in previous time period
                            if C_r(j,k,l,m) == 0
                                C_r(j,k,l,m) =
C_rmean(disturb(3,d_state2(j,k,l,m))) +
sd_r(disturb(3,d_state2(j,k,l,m))) * randn;
                                C_r(find(C_r>1)) = 1;
                                C_r(find(C_r<=0)) = 0.001;
                                crypts_pos_old(j,k,1,l,m) =
C_r(j,k,l,m) * crypts_pos_old(j,k,1,l,m);
                            end
                            if C_a(j,k,l,m) == 0
                                C_a(j,k,l,m) =
C_amean(disturb(3,d_state2(j,k,l,m))) +
sd_a(disturb(3,d_state2(j,k,l,m))) * randn;
                                C_a(find(C_a<1)) = 1;
                                crypts_pos_old(j,k,2,l,m) =
C_a(j,k,l,m) * crypts_pos_old(j,k,2,l,m);
                            end
                        end

                        %kill certain percentage of cells at an
appropriate
                        %time
```

```
                        if ismember(t(i), T_kill)
                            kill = rand;
                            if kill < p_kill &&
crypts_pos_old(j,k,1,l,m) > 0
                                %delete cell
                                crypts_pos_new(j,k,1,l,m) = 0;
                                crypts_pos_new(j,k,2,l,m) = 0;
                                alpha_new(j,k,1,l,m) = 0;
                                alpha_new(j,k,2,l,m) = 0;
                                mut(j,k,:,l,m) = 0;
                                mut2(j,k,:,l,m) = 0;
                                give_coop_new(j,k,l,m) = 0;
                            end
                        end


                        %check if nearby crypts are empty
                        if crypts_ncells(j,k-1) == 0 ||
crypts_ncells(j,k+1) == 0 || ...
                            (j ~= 1 && j~=M && (crypts_ncells(j-1,k-1)
== 0 || crypts_ncells(j-1,k) == 0 || ...
                            crypts_ncells(j-1,k+1) == 0 ||
crypts_ncells(j+1,k-1) == 0 || ...
                            crypts_ncells(j+1,k) == 0 ||
crypts_ncells(j+1,k+1) == 0)) || ...
                            (j == 1 && (crypts_ncells(M,k-1) == 0 ||
crypts_ncells(M,k) == 0 || ...
                            crypts_ncells(M,k+1) == 0 ||
crypts_ncells(j+1,k-1) == 0 || ...
                            crypts_ncells(j+1,k) == 0 ||
crypts_ncells(j+1,k+1) == 0)) || ...
                            (j==M && (crypts_ncells(j-1,k-1) == 0 ||
crypts_ncells(j-1,k) == 0 || ...
                            crypts_ncells(j-1,k+1) == 0 ||
crypts_ncells(1,k-1) == 0 || ...
                            crypts_ncells(1,k) == 0 ||
crypts_ncells(1,k+1) == 0))

                            %cell replication stage
                            %see if there are any empty neighboring
cell sites
                            empty_sites = find(crypts_pos_old(j,k,1,(l-
1):(l+1),(m-1):(m+1)) == 0);
                            num_empty = length(empty_sites);  %# of
empty neighboring sites
                            %draw # of offspring from Poisson
distribution
                            if num_empty > 0 &&
crypts_pos_old(j,k,1,l,m) > 0
                                dN = crypts_pos_old(j,k,1,l,m) * (1 -
...
                                    sum(sum(reshape(alpha_old(j,k,1,(l-
1):(l+1),(m-1):(m+1)),1,1,3,3) .* ...
                                    give_coop_old(j,k,(l-1):(l+1),(m-
1):(m+1)) + ...
                                    ceil(abs(reshape(alpha_old(j,k,1,
(l-1):(l+1),(m-1):(m+1)),1,1,3,3))) .* ...
```

```
                                    (1 - give_coop_old(j,k,(l-1):(l+1),
(m-1):(m+1)))))/9);
                                N_off = poissrnd(dN*dt);
                            end
                            if crypts_pos_old(j,k,1,l,m) > 0 && N_off >
0
                                minimum = min([num_empty N_off]);
                                %randomly permute the empty sites
                                empty_sites =
empty_sites(randperm(num_empty));
                                empty_sites = empty_sites(1:minimum);
%normalize empty sites

                                %determine if mutations in any of the
genes in
                                %original cell or daughter cells have
occurred
                                %see if there is a mutation in an
essential gene for any daughter cell
                                %if so, cell dies
                                mut_essential =
rand(length(empty_sites),1);
                                loc_essential = find(mut_essential <
0.5*p_essential*mut_rate);
                                empty_sites(loc_essential) = [];
%delete cell that has mutated essential gene
                                off(l,m) = length(empty_sites);
%record potential # of offpsring

                                if ~isempty(empty_sites)
                                    draw_mut =
rand(5,length(empty_sites));  %draw mutations for daughter cells
                                    draw_mut2 =
rand(5,length(empty_sites));  %draw mutations in second alleles
                                    %genes and sites where mutations
have taken place
                                    mut_loc = find(draw_mut <
repmat(0.5*gene_mut,5,length(empty_sites)));
                                    mut2_loc = find(draw_mut2 <
repmat(0.5*gene_mut,5,length(empty_sites)));
                                    draw_mut =
zeros(5,length(empty_sites));  %zero draw_mut
                                    draw_mut(mut_loc) = 1;  %set
corresponding values to 1 if mutations have taken place
                                    draw_mut2 =
zeros(5,length(empty_sites));
                                    draw_mut2(mut2_loc) = 1;
                                    [row,col] = ind2sub([3
3],empty_sites);
                                    row = row + l - 2;  %adjust to fit
larger cell site matrix
                                    col = col + m - 2;

                                    %determine if cells have acquired
mutation that would allow them to take advantage of
                                    %cooperative signal (1 gene
involved)
```

```
                                    coop_mut =
rand(1,length(empty_sites));
                                    coop_loc = find(coop_mut <
0.5*gene_mut);
                                    coop_mut =
zeros(1,length(empty_sites));
                                    coop_mut(coop_loc) = 1;  %set to 1
places where mutations have taken place


                                    lin_index = sub2ind([dim dim],l,m);
%convert l,m to linear index

                                    for n = 1:length(row)
                                        %each matrix in candidate cell
array will have first row giving the linear
                                        %index of parent cell (l,m
linear), rows 2-6 will be mutation vector for 5
                                        %genes under consideration,
rows 7-11 the mutation vector for second allele
                                        %row 12 will be coop_mut
                                        candidates{row(n),col(n)} =
[candidates{row(n),col(n)}, ...
                                            [lin_index; draw_mut(:,n);
draw_mut2(:,n); coop_mut(1,n)]];
                                    end
                                end
                            end

                        else  %no empty neighboring crypts
                            %cell replication stage
                            %see if there are any empty neighboring
cell sites
                            empty_sites = find(crypts_pos_old(j,k,1,(l-
1):(l+1),(m-1):(m+1)) == 0);
                            num_empty = length(empty_sites);  %# of
empty neighboring sites
                            %draw # of offspring from Poisson
distribution
                            if num_empty > 0 &&
crypts_pos_old(j,k,1,l,m) > 0
                                dN = a * (1 - ...
                                sum(sum(reshape(alpha_old(j,k,1,(l-
1):(l+1),(m-1):(m+1)),1,1,3,3) .* ...
                                give_coop_old(j,k,(l-1):(l+1),(m-
1):(m+1)) + ...
                                ceil(abs(reshape(alpha_old(j,k,1,
(l-1):(l+1),(m-1):(m+1)),1,1,3,3))) .* ...
                                (1 - give_coop_old(j,k,(l-1):(l+1),
(m-1):(m+1)))))/9);
                                N_off = poissrnd(dN*dt);
                            end
                            if crypts_pos_old(j,k,1,l,m) > 0 && N_off >
0
                                minimum = min([num_empty N_off]);
                                %randomly permute the empty sites
                                empty_sites =
```

```
empty_sites(randperm(num_empty));
                                empty_sites = empty_sites(1:minimum);
%normalize empty sites

                                %determine if mutations in any of the
genes in
                                %original cell or daughter cells have
occurred
                                %see if there is a mutation in an
essential gene for any daughter cell
                                %if so, cell dies
                                mut_essential =
rand(length(empty_sites),1);
                                loc_essential = find(mut_essential <
0.5*p_essential*mut_rate);
                                empty_sites(loc_essential) = [];
%delete cell that has mutated essential gene
                                off(l,m) = length(empty_sites);
%record potential # of offpsring

                                if ~isempty(empty_sites)
                                    draw_mut =
rand(5,length(empty_sites));  %draw mutations for daughter cells
                                    draw_mut2 =
rand(5,length(empty_sites));  %draw mutations in second alleles
                                    %genes and sites where mutations
have taken place
                                    mut_loc = find(draw_mut <
repmat(0.5*gene_mut,5,length(empty_sites)));
                                    mut2_loc = find(draw_mut2 <
repmat(0.5*gene_mut,5,length(empty_sites)));
                                    draw_mut =
zeros(5,length(empty_sites));  %zero draw_mut
                                    draw_mut(mut_loc) = 1;  %set
corresponding values to 1 if mutations have taken place
                                    draw_mut2 =
zeros(5,length(empty_sites));
                                    draw_mut2(mut2_loc) = 1;
                                    [row,col] = ind2sub([3
3],empty_sites);
                                    row = row + l - 2;  %adjust to fit
larger cell site matrix
                                    col = col + m - 2;

                                    %determine if cells have acquired
mutation that would allow them give
                                    %cooperative signal (1 gene
involved)
                                    coop_mut =
rand(1,length(empty_sites));
                                    coop_loc = find(coop_mut <
0.5*gene_mut);
                                    coop_mut =
zeros(1,length(empty_sites));
                                    coop_mut(coop_loc) = 1;  %set to 1
places where mutations have taken place
```

```
                                    lin_index = sub2ind([dim dim],l,m);
%convert l,m to linear index

                                    for n = 1:length(row)
                                        %each matrix in candidate cell
array will have first row giving the linear
                                        %index of parent cell (l,m
linear), rows 2-6 will be mutation vector for 5
                                        %genes under consideration,
rows 7-11 the mutation vector for second allele
                                        %row 12 will be coop_mut
                                        candidates{row(n),col(n)} =
[candidates{row(n),col(n)}, ...
                                            [lin_index; draw_mut(:,n);
draw_mut2(:,n); coop_mut(1,n)]];
                                    end
                                end
                            end

                        end
                    end
                end


                %choose which daughter cells will occupy empty sites
                for r = (crypts_min_max(j,k,1)-1):
(crypts_min_max(j,k,2)+1)
                    for s = (crypts_min_max(j,k,3)-1):
(crypts_min_max(j,k,4)+1)
                        if ~isempty(candidates{r,s})
                            %select random cell from candidates to
occupy given location
                            candidates{r,s} = candidates{r,s}
(:,randperm(size(candidates{r,s},2)));
                            mut(j,k,:,r,s) =
ceil((mut(j,k,:,candidates{r,s}(1,1)) + reshape(candidates{r,s}
(2:6,1),1,1,5))/2);
                            new_mut = new_mut +
reshape(mut(j,k,:,r,s),5,1) - reshape(mut(j,k,:,candidates{r,s}
(1,1)),5,1);
                            %make sure that mutation in 2nd allele only
                            %occurs if there is a mutation in 1st allele
                            mut2(j,k,:,r,s) = mut(j,k,:,r,s) .* ...
                                ceil((mut2(j,k,:,candidates{r,s}(1,1))
+ reshape(candidates{r,s}(7:11,1),1,1,5))/2);
                            new_mut2 = new_mut2 +
reshape(mut2(j,k,:,r,s),5,1) - reshape(mut2(j,k,:,candidates{r,s}
(1,1)),5,1);
                            give_coop_new(j,k,r,s) =
ceil((give_coop_old(j,k,candidates{r,s}(1,1)) + candidates{r,s}
(12,1))/2);
                            new_mut_coop = new_mut_coop +
give_coop_new(j,k,r,s) - give_coop_old(j,k,candidates{r,s}(1,1));

                            %find replication and apoptosis factors
that reflect new mutations
```

```
                            R_factors = reshape((mut(j,k,:,r,s) -
mut(j,k,:,candidates{r,s}(1,1))),1,5) .* R_gene;
                            R2_factors = reshape((mut2(j,k,:,r,s) -
mut2(j,k,:,candidates{r,s}(1,1))),1,5) .* R_gene2;
                            R_factors = [R_factors R2_factors];
                            R_factors(find(R_factors == 0)) = 1;  %set
0's to 1's to allow for multiplication
                            a_factors = reshape((mut(j,k,:,r,s) -
mut(j,k,:,candidates{r,s}(1,1))),1,5) .* a_gene;
                            a2_factors = reshape((mut2(j,k,:,r,s) -
mut2(j,k,:,candidates{r,s}(1,1))),1,5) .* a_gene2;
                            a_factors = [a_factors a2_factors];
                            a_factors(find(a_factors == 0)) = 1;
%allow for multiplication
                            R_factors = prod(R_factors);
                            a_factors = prod(a_factors);
                            crypts_pos_new(j,k,1,r,s) =
crypts_pos_old(j,k,1,candidates{r,s}(1,1)) * R_factors;
                            crypts_pos_new(j,k,2,r,s) =
crypts_pos_old(j,k,2,candidates{r,s}(1,1)) * a_factors;
                            alpha_new(j,k,1,r,s) =
alpha_old(j,k,1,candidates{r,s}(1,1));
                            alpha_new(j,k,2,r,s) =
alpha_old(j,k,2,candidates{r,s}(1,1));

                            %the potential daughter cells that were not
selected for reproduction
                            %reduce the # of offspring for the parent
cell
                            if size(candidates{r,s},2) > 1
                                off(candidates{r,s}(1,2:end)) =
off(candidates{r,s}(1,2:end)) - 1;
                            end
                        end
                    end
                end


                %draw mutations for each parent cell
                for v = crypts_min_max(j,k,1):crypts_min_max(j,k,2)
                    for w = crypts_min_max(j,k,3):crypts_min_max(j,k,4)

                        if off(v,w) > 0
                            parent_essent = rand(1,off(v,w));  %draw
mutations in essential genes
                            parent_essent = min(parent_essent);
                            if parent_essent < (0.5 * p_essential *
mut_rate)
                                %parent cell dies
                                crypts_pos_new(j,k,:,v,w) = 0;
                                alpha_new(j,k,:,v,w) = 0;
                                mut(j,k,:,v,w) = 0;
                                mut2(j,k,:,v,w) = 0;
                                give_coop_new(j,k,v,w) = 0;
                                off(v,w) = 0;
                            else  %continue drawing mutations for
parent cell
```

```
                                mut_parent = rand(5,off(v,w));
                                mut_parent = min(mut_parent,[],2);
                                mut2_parent = rand(5,off(v,w));
                                mut2_parent = min(mut2_parent,[],2);
                                parent_loc = find(mut_parent <
repmat(0.5*gene_mut,5,1));
                                parent2_loc = find(mut2_parent <
repmat(0.5*gene_mut,5,1));
                                mut_parent = zeros(5,1);  %zero
mut_parent
                                mut_parent(parent_loc) = 1;
                                mut2_parent = zeros(5,1);
                                mut2_parent(parent2_loc) = 1;
                                mut_old = mut(j,k,:,v,w);
                                mut(j,k,:,v,w) = ceil((mut(j,k,:,v,w) +
reshape(mut_parent,1,1,5))/2);
                                new_mut = new_mut +
reshape(mut(j,k,:,v,w),5,1) - reshape(mut_old,5,1);
                                mut2_old = mut2(j,k,:,v,w);
                                mut2(j,k,:,v,w) = mut(j,k,:,v,w) .* ...
                                    ceil((mut2(j,k,:,v,w) +
reshape(mut2_parent,1,1,5))/2);
                                new_mut2 = new_mut2 +
reshape(mut2(j,k,:,v,w),5,1) - reshape(mut2_old,5,1);

                                %determine if parent cell has acquired
coop signal mutation
                                coop_parent = rand(1,off(v,w));
                                coop_parent = min(coop_parent);
                                Pcoop_parent = find(coop_parent <
0.5*gene_mut);
                                coop_parent = 0;
                                coop_parent(Pcoop_parent) = 1;
                                give_coop_new(j,k,v,w) =
ceil((give_coop_old(j,k,v,w) + coop_parent)/2);
                                new_mut_coop = new_mut_coop +
give_coop_new(j,k,v,w) - give_coop_old(j,k,v,w);

                                %adjust apoptosis and replication rates
for parent cell
                                PR_factors = reshape((mut(j,k,:,v,w) -
mut_old),1,5) .* R_gene;
                                PR2_factors = reshape((mut2(j,k,:,v,w)
- mut2_old),1,5) .* R_gene;
                                PR_factors = [PR_factors PR2_factors];
                                PR_factors(find(PR_factors == 0)) = 1;
%set 0's to 1's to allow for multiplication
                                Pa_factors = reshape((mut(j,k,:,v,w) -
mut_old),1,5) .* a_gene;
                                Pa2_factors = reshape((mut2(j,k,:,v,w)
- mut2_old),1,5) .* a_gene;
                                Pa_factors = [Pa_factors Pa2_factors];
                                Pa_factors(find(Pa_factors == 0)) = 1;
                                crypts_pos_new(j,k,1,v,w) =
crypts_pos_old(j,k,1,v,w) * prod(PR_factors);
                                crypts_pos_new(j,k,2,v,w) =
crypts_pos_old(j,k,2,v,w) * prod(Pa_factors);
```

```
                                alpha_new(j,k,1,v,w) =
alpha_old(j,k,1,v,w);
                                alpha_new(j,k,2,v,w) =
alpha_old(j,k,2,v,w);

                                off(v,w) = 0;
                            end
                        elseif crypts_pos_old(j,k,1,v,w) > 0  %set all
variables to previous values
                            crypts_pos_new(j,k,1,v,w) =
crypts_pos_old(j,k,1,v,w);
                            crypts_pos_new(j,k,2,v,w) =
crypts_pos_old(j,k,2,v,w);
                            alpha_new(j,k,1,v,w) = alpha_old(j,k,1,v,w);
                            alpha_new(j,k,2,v,w) = alpha_old(j,k,2,v,w);
                            give_coop_new(j,k,v,w) =
give_coop_old(j,k,v,w);
                            give_coop_new(j,k,v,w) =
give_coop_old(j,k,v,w);
                        end

                        %determine if cell should commit apoptosis
                        apop_rate = crypts_pos_old(j,k,2,v,w) * (1 - ...
                            sum(sum(reshape(alpha_old(j,k,2,(v-1):
(v+1),(w-1):(w+1)),1,1,3,3) .* ...
                            give_coop_old(j,k,(v-1):(v+1),(w-1):
(w+1))))/9);
                        p_apop = apop_rate * dt;  %probability of
apoptosis
                        if rand < p_apop
                            %cell commits apoptosis (reflected in next
time period)
                            crypts_pos_new(j,k,1,v,w) = 0;
                            crypts_pos_new(j,k,2,v,w) = 0;
                            alpha_new(j,k,1,v,w) = 0;
                            alpha_new(j,k,2,v,w) = 0;
                            mut(j,k,:,v,w) = 0;
                            mut2(j,k,:,v,w) = 0;
                            give_coop_new(j,k,v,w) = 0;
                        end

                    end
                end

                %clear candidates for next round
                candidates = cell(dim,dim);

                %update crypts_pos_old, alpha_old, and give_coop_old
                crypts_pos_old = crypts_pos_new;
                alpha_old = alpha_new;
                give_coop_new = zeros(M,N+2,dim,dim);  %variant with no
cooperation
                give_coop_old = give_coop_new;

                %record crypts_pos, alpha, and give_coop if i+1 is a
multiple of 100 days
                if rem(i-1,100) == 0
```

```
                    crypts_pos((((i-1)/100)+1),:,:,:,:,:) =
crypts_pos_new;
                    alpha((((i-1)/100)+1),:,:,:,:,:) = alpha_new;
                    give_coop((((i-1)/100)+1),:,:,:,:) = give_coop_new;
                    mutation((((i-1)/100)+1),:,:,:,:,:) = mut;
                    mutation2((((i-1)/100)+1),:,:,:,:,:) = mut2;
                    num_mut((((i-1)/100)+1),:) = new_mut;
                    num_mut2((((i-1)/100)+1),:) = new_mut2;
                    num_mut_coop((((i-1)/100)+1)) = new_mut_coop;
                end

                %adjust crypts_min_max and crypts_ncells
                crypts_ncells(j,k) =
length(find(crypts_pos_new(j,k,1,:,:) > 0));
                linear_index = find(crypts_pos_new(j,k,1,:,:) > 0);
                if ~isempty(linear_index)
                    [rows, cols] = ind2sub([dim, dim],linear_index);
                    row_min = min(rows);
                    row_max = max(rows);
                    col_min = min(cols);
                    col_max = max(cols);
                    crypts_min_max(j,k,:) = [row_min, row_max, col_min,
col_max];
                else  %no cells left in crypt
                    crypts_min_max(j,k,:) = [0, 0, 0, 0];
                end
            end
        end
    end
end


return
```

```
function [crypts_ncells, crypts_pos, crypts_pos_old,
crypts_min_max,alpha, alpha_old] = ...
    initialize_crypts(crypts_ncells, crypts_pos, crypts_pos_old,
crypts_min_max,alpha, alpha_old)

%function that initializes the crypts in the model by filling them with
%stem cells from the Shibata distribution; all cells are placed in the
%middle of the disc representing the carrying capacity
%refer to the main function Model.m for a description of the function
%inputs, as they are the same

%global variables
global M N dim R a stem_cells

for i = 1:M
    for j = 1:N+2
        l = rand;
        cum_prob = cumsum(stem_cells(2,:)); %cumulative probability fxn
of the Shibata distribution
        cum_prob(25) = 1;
        k = find(l<=cum_prob); %find where on prob distribution l falls
        crypts_ncells(i,j) = stem_cells(1,k(1));

        %determine positions in which to put cells
        d1 = ceil(sqrt(stem_cells(1,k(1))));
        d2 = floor(sqrt(stem_cells(1,k(1))));

        row_min = floor((dim-d1)/2) + 1;
        row_max = row_min + d1 - 1;
        col_min = floor((dim-d2)/2) + 1;
        col_max = col_min + d2 - 1 + ceil((stem_cells(1,k(1))-
d1*d2)/d1);
        crypts_min_max(i,j,1) = row_min;
        crypts_min_max(i,j,2) = row_max;
        crypts_min_max(i,j,3) = col_min;
        crypts_min_max(i,j,4) = col_max;

        %row of the last cell in the last column
        row_last = row_max + stem_cells(1,k(1)) - (row_max-row_min+1) *
(col_max-col_min+1);

        %initialize the cell replication and apoptosis rates to the
        %basal rates
        if j == 1 || j == N+2
            crypts_pos(:,i,j,1,row_min:row_max,col_min:(col_max-1)) = R;
            crypts_pos(:,i,j,1,row_min:row_last,col_max) = R;
            crypts_pos(:,i,j,2,row_min:row_max,col_min:(col_max-1)) = a;
            crypts_pos(:,i,j,2,row_min:row_last,col_max) = a;
            crypts_ncells(i,j) = stem_cells(1,k(1));
            crypts_pos(:,i,j,[1 2],1,:) = -1;  %placeholders
            crypts_pos(:,i,j,[1 2],dim,:) = -1;
            crypts_pos(:,i,j,[1 2],:,1) = -1;
            crypts_pos(:,i,j,[1 2],:,dim) = -1;
        else
            crypts_pos(1,i,j,1,row_min:row_max,col_min:(col_max-1)) = R;
            crypts_pos(1,i,j,1,row_min:row_last,col_max) = R;
            crypts_pos(1,i,j,2,row_min:row_max,col_min:(col_max-1)) = a;
```

```
            crypts_pos(1,i,j,2,row_min:row_last,col_max) = a;
            crypts_pos(:,i,j,[1 2],1,:) = -1;  %placeholders
            crypts_pos(:,i,j,[1 2],dim,:) = -1;
            crypts_pos(:,i,j,[1 2],:,1) = -1;
            crypts_pos(:,i,j,[1 2],:,dim) = -1;
            r_random1 = rand((row_max-row_min+1),(col_max-col_min));
            r_random2 = rand((row_last-row_min+1),1);
            a_random1 = rand((row_max-row_min+1),(col_max-col_min));
            a_random2 = rand((row_last-row_min+1),1);
            alpha(1,i,j,1,row_min:row_max,col_min:(col_max-1)) = -1 + 2
* r_random1;
            alpha(1,i,j,1,row_min:row_last,col_max) = -1 + 2 *
r_random2;
            alpha(1,i,j,2,row_min:row_max,col_min:(col_max-1)) =
a_random1;
            alpha(1,i,j,2,row_min:row_last,col_max) = a_random2;
            alpha_old(i,j,1,row_min:row_max,col_min:(col_max-1)) = -1 +
2 * r_random1;
            alpha_old(i,j,1,row_min:row_last,col_max) = -1 + 2 *
r_random2;
            alpha_old(i,j,2,row_min:row_max,col_min:(col_max-1)) =
a_random1;
            alpha_old(i,j,2,row_min:row_last,col_max) = a_random2;
        end

        crypts_pos_old(i,j,1,row_min:row_max,col_min:(col_max-1)) = R;
        crypts_pos_old(i,j,1,row_min:row_last,col_max) = R;
        crypts_pos_old(i,j,2,row_min:row_max,col_min:(col_max-1)) = a;
        crypts_pos_old(i,j,2,row_min:row_last,col_max) = a;
        crypts_pos_old(i,j,[1 2],1,:) = -1;  %placeholders
        crypts_pos_old(i,j,[1 2],dim,:) = -1;
        crypts_pos_old(i,j,[1 2],:,1) = -1;
        crypts_pos_old(i,j,[1 2],:,dim) = -1;

    end
end

return
```

```
function [crypts_ncells, crypts_pos_old, crypts_pos_new, alpha_old,
alpha_new, crypts_min_max, mut, mut2,receive_coop_old,
receive_coop_new] = ...
                        split_crypt(crypts_ncells, crypts_pos_old,
crypts_pos_new, alpha_old, alpha_new, crypts_min_max, mut, mut2, ...
                        receive_coop_old, receive_coop_new, j, k)


%function that splits crypt filled to carrying capacity down the middle
%crypt splits only if there are empty neighboring crypts

global M dim

empty_crypts = [];  %matrix of empty neighboring crypts

%compile matrix of empty neighboring crypts
%look at each case
if j ~= 1 && j ~= M
    empty_loc = find(crypts_ncells((j-1):(j+1),(k-1):(k+1)) == 0);
    if ~isempty(empty_loc)
        [row, col] = ind2sub([3 3], empty_loc);
        row = row + j - 2;  %adjust to fit larger crypt matrix
        col = col + k - 2;
        %add crypts to empty crypts matrix
        empty_crypts = [empty_crypts; [row, col]];
    end
end

if j == 1
    empty_loc = find(crypts_ncells(1:(j+1),(k-1):(k+1)) == 0);
    if ~isempty(empty_loc)
        [row, col] = ind2sub([2 3], empty_loc);
        row = row + j - 1;  %adjust to fit larger crypt matrix
        col = col + k - 2;
        empty_crypts = [empty_crypts; [row, col]];
    end
    if crypts_ncells(M,k-1) == 0
        empty_crypts = [empty_crypts; [M, k-1]];
    end
    if crypts_ncells(M,k) == 0
        empty_crypts = [empty_crypts; [M, k]];
    end
    if crypts_ncells(M,k+1) == 0
        empty_crypts = [empty_crypts; [M, k+1]];
    end
end

if j == M
   empty_loc = find(crypts_ncells((j-1):M,(k-1):(k+1)) == 0);
    if ~isempty(empty_loc)
        [row, col] = ind2sub([2 3], empty_loc);
        row = row + j - 2;  %adjust to fit larger crypt matrix
        col = col + k - 2;
        empty_crypts = [empty_crypts; [row, col]];
    end
    if crypts_ncells(1,k-1) == 0
        empty_crypts = [empty_crypts; [1, k-1]];
```

```
    end
    if crypts_ncells(1,k) == 0
        empty_crypts = [empty_crypts; [1, k]];
    end
    if crypts_ncells(1,k+1) == 0
        empty_crypts = [empty_crypts; [1, k+1]];
    end
end

if ~isempty(empty_crypts)

    %choose random crypt from empty crypts matrix to split cells into
    empty_crypts = empty_crypts(randperm(size(empty_crypts,1)),:);
    chosen_crypt = empty_crypts(1,:);

    %split the crypts
    crypts_ncells(j,k) = floor(crypts_ncells(j,k)/2);
    crypts_ncells(chosen_crypt(1),chosen_crypt(2)) = crypts_ncells(j,k);

    %determine positions in which to put cells
    d1 = ceil(sqrt(crypts_ncells(j,k)));
    d2 = floor(sqrt(crypts_ncells(j,k)));
    row_min = floor((dim-2-d1)/2) + 1;
    row_max = row_min + d1 - 1;
    col_min = floor((dim-2-d2)/2) + 1;
    col_max = col_min + d2 - 1 + ceil((crypts_ncells(j,k)-d1*d2)/d1);
    crypts_min_max(j,k,1) = row_min + 1;
    crypts_min_max(j,k,2) = row_max + 1;
    crypts_min_max(j,k,3) = col_min + 1;
    crypts_min_max(j,k,4) = col_max + 1;
    crypts_min_max(chosen_crypt(1),chosen_crypt(2),1) = row_min + 1;
    crypts_min_max(chosen_crypt(1),chosen_crypt(2),2) = row_max + 1;
    crypts_min_max(chosen_crypt(1),chosen_crypt(2),3) = col_min + 1;
    crypts_min_max(chosen_crypt(1),chosen_crypt(2),4) = col_max + 1;

    %row of the last cell in the last column
    row_last = row_max + crypts_ncells(j,k) - (row_max-row_min+1) *
(col_max-col_min+1);
    %set the cell replication and apoptosis rates
    crypts_pos2 = crypts_pos_old(j,k,:,:,:);  %store old values of
crypts_pos matrix
    alpha2 = alpha_old(j,k,:,:,:);
    mut_old = mut(j,k,:,:,:);
    mut2_old = mut2(j,k,:,:,:);
    receive_coop2 = receive_coop_old(j,k,:,:);

    %convert rows to linear index
    rows = repmat(row_min:row_max,1,(col_max-col_min));
    cols = reshape(repmat(col_min:(col_max-1),(row_max-row_min+1),1),1,
(row_max-row_min+1) * (col_max-col_min));
    ind = sub2ind([dim-2 dim-2],rows,cols);
    rows2 = row_min:row_last;
    cols2 = repmat(col_max,1,(row_last-row_min+1));
    ind2 = sub2ind([dim-2 dim-2],rows2,cols2);

    %original crypt
    crypts_pos_old(j,k,:,:,:) = 0;  %zero previous values
```

```
    crypts_pos_new(j,k,:,:,:) = 0;
    alpha_old(j,k,:,:,:) = 0;
    alpha_new(j,k,:,:,:) = 0;
    mut(j,k,:,:,:) = 0;
    mut2(j,k,:,:,:) = 0;
    receive_coop_old(j,k,:,:) = 0;
    receive_coop_new(j,k,:,:) = 0;
    crypts_pos_old(j,k,1,ind) = crypts_pos2(1,1,1,1:(row_max-row_min+1)
* (col_max-col_min));
    crypts_pos_old(j,k,1,ind2) = ...
        crypts_pos2(1,1,1,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));
    crypts_pos_old(j,k,2,ind) = crypts_pos2(1,1,2,1:(row_max-row_min+1)
* (col_max-col_min));
    crypts_pos_old(j,k,2,ind2) = ...
        crypts_pos2(1,1,2,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));
    alpha_old(j,k,1,ind) = alpha2(1,1,1,1:(row_max-row_min+1) *
(col_max-col_min));
    alpha_old(j,k,1,ind2) = ...
        alpha2(1,1,1,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));
    alpha_old(j,k,2,ind) = alpha2(1,1,2,1:(row_max-row_min+1) *
(col_max-col_min));
    alpha_old(j,k,2,ind2) = ...
        alpha2(1,1,2,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));
    mut(j,k,:,ind) = mut_old(1,1,:,1:(row_max-row_min+1) * (col_max-
col_min));
    mut(j,k,:,ind2) = ...
        mut_old(1,1,:,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));
    mut2(j,k,:,ind) = mut2_old(1,1,:,1:(row_max-row_min+1) * (col_max-
col_min));
    mut2(j,k,:,ind2) = ...
        mut2_old(1,1,:,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));
    receive_coop_old(j,k,ind) = receive_coop2(1,1,1:(row_max-row_min+1)
* (col_max-col_min));
    receive_coop_old(j,k,ind2) = ...
        receive_coop2(1,1,((row_max-row_min+1) * (col_max-col_min) +
1):crypts_ncells(j,k));

    %chosen crypt to receive half of splitting cells
    crypts_pos_old(chosen_crypt(1),chosen_crypt(2),1,ind) = ...
        crypts_pos2(1,1,1,(crypts_ncells(j,k) + 1):((row_max-row_min+1)
* (col_max-col_min) + crypts_ncells(j,k)));
    crypts_pos_old(chosen_crypt(1),chosen_crypt(2),1,ind2) = ...
        crypts_pos2(1,1,1,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));
    crypts_pos_old(chosen_crypt(1),chosen_crypt(2),2,ind) = ...
        crypts_pos2(1,1,2,(crypts_ncells(j,k) + 1):((row_max-row_min+1)
* (col_max-col_min) + crypts_ncells(j,k)));
    crypts_pos_old(chosen_crypt(1),chosen_crypt(2),2,ind2) = ...
        crypts_pos2(1,1,2,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));
    alpha_old(chosen_crypt(1),chosen_crypt(2),1,ind) = ...
```

```
        alpha2(1,1,1,(crypts_ncells(j,k) + 1):((row_max-row_min+1) *
(col_max-col_min) + crypts_ncells(j,k)));
    alpha_old(chosen_crypt(1),chosen_crypt(2),1,ind2) = ...
        alpha2(1,1,1,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));
    alpha_old(chosen_crypt(1),chosen_crypt(2),2,ind) = ...
        alpha2(1,1,2,(crypts_ncells(j,k) + 1):((row_max-row_min+1) *
(col_max-col_min) + crypts_ncells(j,k)));
    alpha_old(chosen_crypt(1),chosen_crypt(2),2,ind2) = ...
        alpha2(1,1,2,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));
    mut(chosen_crypt(1),chosen_crypt(2),:,ind) = ...
        mut_old(1,1,:,(crypts_ncells(j,k) + 1):((row_max-row_min+1) *
(col_max-col_min) + crypts_ncells(j,k)));
    mut(chosen_crypt(1),chosen_crypt(2),:,ind2) = ...
        mut_old(1,1,:,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));
    mut2(chosen_crypt(1),chosen_crypt(2),:,ind) = ...
        mut2_old(1,1,:,(crypts_ncells(j,k) + 1):((row_max-row_min+1) *
(col_max-col_min) + crypts_ncells(j,k)));
    mut2(chosen_crypt(1),chosen_crypt(2),:,ind2) = ...
        mut2_old(1,1,:,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));
    receive_coop_old(chosen_crypt(1),chosen_crypt(2),ind) = ...
        receive_coop2(1,1,(crypts_ncells(j,k) + 1):((row_max-row_min+1)
* (col_max-col_min) + crypts_ncells(j,k)));
    receive_coop_old(chosen_crypt(1),chosen_crypt(2),ind2) = ...
        receive_coop2(1,1,((row_max-row_min+1) * (col_max-col_min) +
crypts_ncells(j,k) + 1):(2 * crypts_ncells(j,k)));

    crypts_pos_new(j,k,1,ind) = crypts_pos_old(j,k,1,ind);
    crypts_pos_new(j,k,2,ind) = crypts_pos_old(j,k,2,ind);
    crypts_pos_new(j,k,1,ind2) = crypts_pos_old(j,k,1,ind2);
    crypts_pos_new(j,k,2,ind2) = crypts_pos_old(j,k,2,ind2);
    crypts_pos_new(chosen_crypt(1),chosen_crypt(2),1,ind) =
crypts_pos_old(chosen_crypt(1),chosen_crypt(2),1,ind);
    crypts_pos_new(chosen_crypt(1),chosen_crypt(2),2,ind) =
crypts_pos_old(chosen_crypt(1),chosen_crypt(2),2,ind);
    crypts_pos_new(chosen_crypt(1),chosen_crypt(2),1,ind2) =
crypts_pos_old(chosen_crypt(1),chosen_crypt(2),1,ind2);
    crypts_pos_new(chosen_crypt(1),chosen_crypt(2),2,ind2) =
crypts_pos_old(chosen_crypt(1),chosen_crypt(2),2,ind2);
    alpha_new(j,k,1,ind) = alpha_old(j,k,1,ind);
    alpha_new(j,k,2,ind) = alpha_old(j,k,2,ind);
    alpha_new(j,k,1,ind2) = alpha_old(j,k,1,ind2);
    alpha_new(j,k,2,ind2) = alpha_old(j,k,2,ind2);
    alpha_new(chosen_crypt(1),chosen_crypt(2),1,ind) =
alpha_old(chosen_crypt(1),chosen_crypt(2),1,ind);
    alpha_new(chosen_crypt(1),chosen_crypt(2),2,ind) =
alpha_old(chosen_crypt(1),chosen_crypt(2),2,ind);
    alpha_new(chosen_crypt(1),chosen_crypt(2),1,ind2) =
alpha_old(chosen_crypt(1),chosen_crypt(2),1,ind2);
    alpha_new(chosen_crypt(1),chosen_crypt(2),2,ind2) =
alpha_old(chosen_crypt(1),chosen_crypt(2),2,ind2);
    receive_coop_new(j,k,ind) = receive_coop_old(j,k,ind);
    receive_coop_new(j,k,ind2) = receive_coop_old(j,k,ind2);
    receive_coop_new(chosen_crypt(1),chosen_crypt(2),ind) =
```

```
receive_coop_old(chosen_crypt(1),chosen_crypt(2),ind);
    receive_coop_new(chosen_crypt(1),chosen_crypt(2),ind2) =
receive_coop_old(chosen_crypt(1),chosen_crypt(2),ind2);

end


return
```

```
function [corr_cluster, p_cluster, corr_crypt, p_crypt, num_cells,
num_coop_cells, crypts_coop, crypts_non_coop, ...
    growth_cluster, growth_crypt, growth_space, diversity_cluster,
diversity_crypt, diversity_space, avg_growth_coop, avg_growth_noncoop,
...
    num_cells_crypt, num_coopcells_crypt, num_mut_cells,
num_mut_cells2, num_mut_cells_crypt, num_mut_cells_crypt2] = ...
    data_analysis(crypts_pos,alpha,coop,mutation,mutation2,coop_type)

%function that analyzes the data from tumor model simulations

%crypts_pos --
%matrix representing positions of cells in each crypt; first index
%denotes the time, in units of 100 days, second two indices represent
the crypts,
%last two indices represent the position w/in each crypt plus 2
placeholder rows
%and columns to simplify code (limited by the carrying capacity),
middle index denotes
%the replication and apoptosis rate, respectively, at each location (if
replication
%and apoptosis rates are both 0, then there is no cell at that location)

%alpha --
%interaction coefficients that a cell can acquire if it is able to
receive
%cooperation
%the first index represents time in units of 100 days, the next two
indices
%represent the crypts (with two additional columns on the right and
left edges
%as placeholders to simplify code), middle index denotes the
replication and
%apoptosis coefficients, repsectively, and the last two the cell's
position
%within a crypt (top, bottom, left, and right rows and columns are
placeholders
%and will have coefficients of 0) if there is no cells at a site, the
coefficient
%is 0 at that location remember to include these in the code

%coop --
%matrix that represents whether cell is able to give or receive a coop
%signal (depending on model sub-type) -- 0 if no, 1 if yes; if no cell
at site,
%default is 0

%coop_type -- 0 for receive_coop, 1 for give_coop
%specifies model type -- give_coop vs receive_coop


[t,M,N1,dim1,dim2] = size(coop);
%t = number of time points (each time point is 100 days apart)

%initialize vectors carrying total number of cells and cooperating cells
%for entire space
num_cells = zeros(t,1);
```

```
num_coop_cells = zeros(t,1);

%initialze matrices carrying total number of cells and cooperating cells
%for each crypt
num_cells_crypt = zeros(t,M,N1-2);
num_coopcells_crypt = zeros(t,M,N1-2);

%initilialize matrices carrying total # of mutated cells for given gene
num_mut_cells = zeros(t,5);
num_mut_cells2 = zeros(t,5);

%initialize matrices carrying total # of mutated cells for given each
for
%each crypt
num_mut_cells_crypt = zeros(t,M,N1-2,5);
num_mut_cells_crypt2 = zeros(t,M,N1-2,5);

%initialize vectors carrying average growth rates for cooperating and
non-cooperating
%cells that are indedpenent of any surrounding cells, i.e. R-a
avg_growth_coop = zeros(t,1);
avg_growth_noncoop = zeros(t,1);

%initialize matrices containing average growth rate per 9-cell cluster,
%crypt, and entire space at each time point
growth_cluster = cell(t,M,N1-2);
growth_crypt = zeros(t,M,N1-2);
growth_space = zeros(t,1);

%initialize matrices containing cellular diversity score for 9-cell
%cluster, crypt, and entire space at each time point
%diversity score the number of different cell types in a region as
defined
%by different replication rate, apoptosis rate pairs
diversity_cluster = cell(t,M,N1-2);
diversity_crypt = zeros(t,M,N1-2);
diversity_space = zeros(t,1);

%initialize cellular array containing number of cooperating cells in a
%cluster and total cells in a cluster
coop_cluster = cell(t,M,N1-2);  %cooperating cells
cells_cluster = cell(t,M,N1-2);  %total cells

%initialize replication rate matrix
rep_rate = zeros(dim1,dim2);

%initialize correlation matrix between growth rate and fraction of
%cooperating cells (1st collumn) and diversity and cooperating cells
(2nd
%column) for 9-cell clusters
corr_cluster = zeros(t,2);
p_cluster = zeros(t,2);  %correpsonding p values

%initialize correlation matrix between growth rate and proportion of
%cooperating cells (1st collumn) and diversity and cooperating cells
(2nd
%column) for crypts
```

```
corr_crypt = zeros(t,2);
p_crypt = zeros(t,2);  %correpsonding p values


%track the number of crypts certain cells (cooperating and non-
cooperating)
%visit through the entire time of the simulation
%number of cells to be tracked within each category (cooperating and
%non-cooperating)
coop_init = nnz(coop(1,:,:,:,:));
non_coop_init = length(find(crypts_pos(1,:,(2:N1-1),1,:,:) > 0))-
nnz(coop(1,:,:,:,:));
num_cells_track = min(coop_init, non_coop_init);
%pick random cells to be tracked
new_crypts_pos = reshape(crypts_pos(1,:,(2:N1-1),1,:,:),M,(N1-
2),dim1,dim2);
new_coop = reshape(coop(1,:,(2:N1-1),:,:),M,(N1-2),dim1,dim2);
coop_cells = find(new_coop);
new_crypts_pos(coop_cells) = 0;
non_coop_cells = find(new_crypts_pos > 0);
coop_cells = coop_cells(randperm(length(coop_cells)));
non_coop_cells = non_coop_cells(randperm(length(non_coop_cells)));
coop_cells = coop_cells(1:num_cells_track);  %positions of coop cells
to be tracked
non_coop_cells = non_coop_cells(1:num_cells_track); %pos of non-coop
cells to be tracked
alpha_init1 = reshape(alpha(1,:,(2:N1-1),1,:,:),M,(N1-2),dim1,dim2);
%initial alpha for replication
alpha_init2 = reshape(alpha(1,:,(2:N1-1),2,:,:),M,(N1-2),dim1,dim2);
%initial alpha for apoptosis

%initialize tallying matrix for number of crypts a cell's progeny are
present in at a
%given time; if value is 0, cell no longer exists
crypts_coop = zeros(t,num_cells_track);  %number of crypts visited by
cooperating cells
crypts_non_coop = zeros(t,num_cells_track);  %number of crypts visited
by non-cooperating cells


%calculate number of cooperating cells and total number of cells at
each time point
%determine number of crypts visited by selected cooperating and
%non-cooperating cells
for i = 1:t
    num_cells(i) = length(find(crypts_pos(i,:,(2:N1-1),1,:,:) > 0));
%total # of cells
    num_coop_cells(i) = nnz(coop(i,:,:,:,:));   %number of cooperating
cells

    %calculate # of cells with a mutation in each gene
    for v = 1:5
        num_mut_cells(i,v) = nnz(mutation(i,:,:,v,:,:));
        num_mut_cells2(i,v) = nnz(mutation2(i,:,:,v,:,:));
    end

    for j = 1:M
```

```
        for k = 1:N1-2
            %total # of cells in each crypt at given time
            num_cells_crypt(i,j,k) = length(find(crypts_pos(i,j,
(k+1),1,:,:) > 0));
            %number of cooperating cells in each crypt at a given time
            num_coopcells_crypt(i,j,k) = nnz(coop(i,j,(k+1),:,:));

            for w = 1:5
                num_mut_cells_crypt(i,j,k,w) = nnz(mutation(i,j,
(k+1),w,:,:));
                num_mut_cells_crypt2(i,j,k,w) = nnz(mutation2(i,j,
(k+1),w,:,:));
            end


            %build growth rate and diversity matrices
            crypt_cells = find(crypts_pos(i,j,(k+1),1,:,:) > 0);
            if ~isempty(crypt_cells)
                [rows, cols] = ind2sub([dim1, dim2],crypt_cells);
                row_min = min(rows);
                row_max = max(rows);
                col_min = min(cols);
                col_max = max(cols);
                for m = row_min:row_max
                    for n = col_min:col_max
                        empty_sites = find(crypts_pos(i,j,(k+1),1,(m-
1):(m+1),(n-1):(n+1)) == 0);
                        num_empty = length(empty_sites);  %# of empty
neighboring sites
                        %calculate replication rates for each cell
                        if num_empty > 0 && crypts_pos(i,j,(k+1),1,m,n)
> 0
                            if coop(i,j,(k+1),m,n) == 1 && coop_type ==
0
                                dN = crypts_pos(i,j,(k+1),1,m,n) * (1 -
...
                                    sum(sum(alpha(i,j,(k+1),1,(m-1):
(m+1),(n-1):(n+1))))/9);
                                apop = crypts_pos(i,j,(k+1),2,m,n) * (1
- ...
                                    sum(sum(alpha(i,j,(k+1),2,(m-1):
(m+1),(n-1):(n+1))))/9);
                                rep_rate(m,n) = dN - apop;
                            elseif coop_type == 1
                                dN = crypts_pos(i,j,(k+1),1,m,n) * (1 -
...
                                    sum(sum(reshape(alpha(i,j,(k+1),1,
(m-1):(m+1),(n-1):(n+1)),1,1,1,3,3) .* ...
                                    coop(i,j,(k+1),(m-1):(m+1),(n-1):
(n+1)) + ...
                                    ceil(abs(reshape(alpha(i,j,(k+1),1,
(m-1):(m+1),(n-1):(n+1)),1,1,1,3,3))) .* ...
                                    (1 - coop(i,j,(k+1),(m-1):(m+1),(n-
1):(n+1)))))/9);
                                apop = crypts_pos(i,j,(k+1),2,m,n) * (1
- ...
                                    sum(sum(reshape(alpha(i,j,(k+1),2,
```

```
(m-1):(m+1),(n-1):(n+1)),1,1,1,3,3) .* ...
                                        coop(i,j,(k+1),(m-1):(m+1),(n-1):
(n+1))))/9);
                                    rep_rate(m,n) = dN - apop;
                                else  %no coop mutation and coop_receive
                                    dN = crypts_pos(i,j,(k+1),1,m,n) * (1 -
...
                                        sum(sum(ceil(abs(alpha(i,j,(k+1),1,
(m-1):(m+1),(n-1):(n+1))))))/9);
                                    apop = crypts_pos(i,j,(k+1),2,m,n);
                                    rep_rate(m,n) = dN - apop;
                                end
                            end
                        end
                    end

                    for m = row_min:row_max
                        for n = col_min:col_max
                            if crypts_pos(i,j,(k+1),1,m,n) > 0
                                rep_cluster = sum(sum(rep_rate((m-1):(m+1),
(n-1):(n+1))));
                                growth_cluster{i,j,k} =
[growth_cluster{i,j,k}; rep_cluster];
                                %find number of cooperating cells in a
cluster
                                num_coop = nnz(coop(i,j,(k+1),(m-1):(m+1),
(n-1):(n+1)));
                                coop_cluster{i,j,k} = [coop_cluster{i,j,k};
num_coop];
                                num_cells_cluster =
length(find(crypts_pos(i,j,(k+1),1,(m-1):(m+1),(n-1):(n+1)) > 0));
                                cells_cluster{i,j,k} =
[cells_cluster{i,j,k}; num_cells_cluster];
                                %find unique cells within cluster
                                pos_cluster = reshape(crypts_pos(i,j,
(k+1),:,(m-1):(m+1),(n-1):(n+1)),2,9);
                                pos_cluster = pos_cluster';
                                cluster_unique = unique(pos_cluster,'rows');
                                diversity_score = size(cluster_unique,1);
                                if ~isempty(find(cluster_unique == -1,1))
                                    diversity_score = diversity_score - 1;
                                end
                                if ~isempty(find(cluster_unique == 0,1))
                                    diversity_score = diversity_score - 1;
                                end
                                diversity_cluster{i,j,k} =
[diversity_cluster{i,j,k}; diversity_score];
                            end
                        end
                    end
                    growth_crypt(i,j,k) = sum(sum(rep_rate));
                    rep_rate = zeros(dim1,dim2);
                    %find unique cells within crypt
                    crypts_pos2 = reshape(crypts_pos(i,j,(k+1),:,2:(dim1-
1),2:(dim2-1)),2,(dim1-2)*(dim2-2));
                    crypts_pos2 = crypts_pos2';
                    crypt_unique = unique(crypts_pos2,'rows');
```

```
                diversity_crypt(i,j,k) = size(crypt_unique,1);
                if ~isempty(find(crypt_unique == 0,1))
                    diversity_crypt(i,j,k) = diversity_crypt(i,j,k) - 1;
                end

            end
        end
    end

    %compute growth rate and diversity index for entire space
    growth_space(i) = sum(sum(growth_crypt(i,:,:)));
    pos_space = permute(crypts_pos, [1 2 3 5 6 4]);
    pos_space2 = reshape(pos_space(i,:,2:(N1-1),2:(dim1-1),2:(dim2-
1),:),M*(N1-2)*(dim1-2)*(dim2-2),2);
    space_unique = unique(pos_space2,'rows');
    diversity_space(i) = size(space_unique,1);
    if ~isempty(find(space_unique == 0,1))
        diversity_space(i) = diversity_space(i) - 1;
    end

    %compute correlations between growth rate and fraction of
cooperating
    %cells and diversity and cooperating cells for 9-cell clusters
    %growth rate
    coop_cluster2 = cat(1,coop_cluster{i,:,:});
    cells_cluster2 = cat(1,cells_cluster{i,:,:});
    coop_fraction_cluster = coop_cluster2./cells_cluster2;  %fraction
of coop cells in cluster
    coop_fraction_cluster(isnan(coop_fraction_cluster)) = 0;
    growth_cluster2 = cat(1,growth_cluster{i,:,:});
    [corr_cluster(i,1), p_cluster(i,1)] = corr(growth_cluster2,
coop_fraction_cluster);
    %diversity
    diversity_cluster2 = cat(1,diversity_cluster{i,:,:});
    [corr_cluster(i,2), p_cluster(i,2)] = corr(diversity_cluster2,
coop_fraction_cluster);

    %compute correlations between growth rate and fraction of
cooperating
    %cells and diversity and cooperating cells for crypts
    %growth rate
    coop_crypt = reshape(num_coopcells_crypt(i,:,:),M*(N1-2),1);
    cells_crypt = reshape(num_cells_crypt(i,:,:),M*(N1-2),1);
    coop_fraction_crypt = coop_crypt./cells_crypt;
    coop_fraction_crypt(isnan(coop_fraction_crypt)) = 0;
    growth_crypt2 = reshape(growth_crypt(i,:,:),M*(N1-2),1);
    [corr_crypt(i,1), p_crypt(i,1)] = corr(growth_crypt2,
coop_fraction_crypt);
    %diversity
    diversity_crypt2 = reshape(diversity_crypt(i,:,:),M*(N1-2),1);
    [corr_crypt(i,2), p_crypt(i,2)] = corr(diversity_crypt2,
coop_fraction_crypt);


    %determine average growth rate for cooperating and non-cooperating
    %cells that is indedpenent of any surrounding cells
    crypts_pos_new1 = reshape(crypts_pos(i,:,2:(N1-1),1,2:(dim1-1),2:
```

```
(dim2-1)),1,M,(N1-2),(dim1-2),(dim2-2));
    crypts_pos_new2 = reshape(crypts_pos(i,:,2:(N1-1),2,2:(dim1-1),2:
(dim2-1)),1,M,(N1-2),(dim1-2),(dim2-2));
    coop_R = crypts_pos_new1 .* coop(i,:,2:(N1-1),2:(dim1-1),2:(dim2-
1));
    coop_a = crypts_pos_new2 .* coop(i,:,2:(N1-1),2:(dim1-1),2:(dim2-
1));
    non_coop_R = crypts_pos_new1 .* (ones(1,M,(N1-2),(dim1-2),(dim2-2))
- coop(i,:,2:(N1-1),2:(dim1-1),2:(dim2-1)));
    non_coop_a = crypts_pos_new2 .* (ones(1,M,(N1-2),(dim1-2),(dim2-2))
- coop(i,:,2:(N1-1),2:(dim1-1),2:(dim2-1)));
    avg_R_coop = sum(sum(sum(sum(coop_R))))/nnz(coop_R);
    if isnan(avg_R_coop)
        avg_R_coop = 0;
    end
    avg_a_coop = sum(sum(sum(sum(coop_a))))/nnz(coop_a);
    if isnan(avg_a_coop)
        avg_a_coop = 0;
    end
    avg_R_noncoop = sum(sum(sum(sum(non_coop_R))))/nnz(non_coop_R);
    if isnan(avg_R_noncoop)
        avg_R_noncoop = 0;
    end
    avg_a_noncoop = sum(sum(sum(sum(non_coop_a))))/nnz(non_coop_a);
    if isnan(avg_a_noncoop)
        avg_a_noncoop = 0;
    end
    avg_growth_coop(i) = avg_R_coop - avg_a_coop;
    avg_growth_noncoop(i) = avg_R_noncoop - avg_a_noncoop;


    %determine # of crypts a given cooperating and non-cooperating
cell's
    %progeny are present in at a given time
    alpha1 = reshape(alpha(i,:,2:(N1-1),1,:,:),M,(N1-2),dim1,dim2);
%alpha at given time
    alpha2 = reshape(alpha(i,:,2:(N1-1),2,:,:),M,(N1-2),dim1,dim2);
    for l = 1:num_cells_track
        %cooperating cells
        %find if cell is present at current time
        coop_intersect1 = ismember(alpha1,alpha_init1(coop_cells(l)));
        coop_intersect2 = ismember(alpha2,alpha_init2(coop_cells(l)));
        %make sure both replication and apoptosis coefficients are the
same
        coop_intersect = coop_intersect1 .* coop_intersect2;
        %locations of cells that survived to current time
        coop_cell_pos = find(coop_intersect);
        if ~isempty(coop_cell_pos)
            [row1, col1, R1, S1] = ind2sub([M, N1-2, dim1, dim2],
coop_cell_pos);
            pos1 = [row1 col1];
            %update tallying matrix for # of crypts visited by cell
            crypts_coop(i,l) = size(unique(pos1,'rows'),1);
        end

        %non_cooperating cells
        %find if cell is present at current time
```

```
        non_coop_intersect1 =
ismember(alpha1,alpha_init1(non_coop_cells(l)));
        non_coop_intersect2 =
ismember(alpha2,alpha_init2(non_coop_cells(l)));
        %make sure both replication and apoptosis coefficients are the
same
        non_coop_intersect = non_coop_intersect1 .* non_coop_intersect2;
        %locations of cells that survived to current time
        non_coop_cell_pos = find(non_coop_intersect);
        if ~isempty(non_coop_cell_pos)
            [row2, col2, R2, S2] = ind2sub([M, N1-2, dim1, dim2],
non_coop_cell_pos);
            pos2 = [row2 col2];
            %update tallying matrix for # of crypts visited by cell
            crypts_non_coop(i,l) = size(unique(pos2,'rows'),1);
        end
    end

end

return
```

References

1. Nowell, P.C. 1976. The clonal evolution of tumor cell populations. *Science*. 194: 23-28.

2. Merlo, L.M., Pepper, J.W., Reid, B.J., and Maley, C.C. 2006. Cancer as an evolutionary and ecological process. *Nat. Rev. Cancer.* 6: 924-935.

3. Crespi, B.J., and Summers, K. 2005. Evolutionary biology of cancer. *Trends Ecol. Evol.* 20: 545-552.

4. Feinberg, A.P., Ohlsson, R, and Henikoff, S. 2006. The epigenetic progenitor origin of human cancer. *Nat. Rev. Genet*. 7: 21-33.

5. Nowak, M.A., Michor, F, Komarova, N.L., and Iwasa, Y. 2004. Evolutionary dynamics of tumor suppressor gene inactivation. *Proc. Natl. Acad. Sci. U.S.A.* 101: 10635-10638.

6. Hanahan, D, and Weinberg, R.A. 2000. The hallmarks of cancer. *Cell*. 100: 57-70.

7. Calabrese, P, Tavare, S, and Shibata, D. 2004. Pretumor progression: clonal evolution of human stem cell populations. *Am. J. Pathol.* 164: 1337-1346.

8. McDonald, S.A., Graham, M, Potten, C.S., Wright, N.A., Tomlinson, I.P., *et. al.* 2006. Intestinal stem cells and the development of colorectal neoplasia. In *Tissue Stem Cells*. C. Potten, R. Clarke, J. Wilson, and A. Renehan, editors. New York: Taylor & Francis. 229-270.

9. Nowak, M.A., Michor, F, Iwasa, Y. 2003. The linear process of somatic evolution. *Proc. Natl. Acad. Sci. U.S.A.* 100: 14966-14969.

10. Fraterrigo J.M., and Rusak, J.A. 2008. Disturbance-driven changes in the variability of ecological patterns and processes. *Ecol. Lett.* 11: 756-770.

11. Connell, J.H. 1978. Diversity in tropical rain forests and coral reefs. *Science*. 199: 1302-1310.

12. Svensson, J.R., Lindegarth, M, Siccha, M, Lenz, M, Molis, M, *et. al.* 2007. Maximum species richness at intermediate frequencies of disturbance: consistency among levels of productivity. *Ecology*. 88: 830-838.

13. Kitano, H. 2004. Cancer as a robust system: implications for anticancer therapy. *Nat. Rev. Cancer.* 4: 227-235.

14. Maley, C.C., Galipeau, P.C., Finley, J.C., Wongsurawat, V.J., Li, X, *et. al.* 2006. Genetic clonal diversity predicts progression to esophageal adenocarcinoma. *Nat. Genet.* 38: 468-473.

15. Gonzalez-Garcia, I, Sole, R.V., Costa, J. 2002. Metapopulation dynamics and spatial heterogeneity in cancer. *Proc. Natl. Acad. Sci. U.S.A.* 99: 13085-13089.

16. Frigyesi, A, Gisselsson, D, Mitelman, F, Hoglund, M. 2003. Power law distribution of chromosome aberrations in cancer. *Cancer Res.* 63: 7094-7097.

17. Gorunova, L, Hoglund, M, Andren-Sandberg, A, Dawiskiba, S, Jin, Y, *et. al.* 1998. Cytogenetic analysis of pancreatic carcinomas: intratumor heterogeneity and nonrandom pattern of chromosome aberrations. *Genes Chromosomes Cancer.* 23: 81-99.

18. Lengauer, C, Kinzler, K.W., and Vogelstein, B. 1998. Genetic instabilities in human cancers. *Nature*. 396: 643-649.

19. Anderson, A.R., and Quaranta, V. 2008. Integrative mathematical oncology. *Nat. Rev. Cancer.* 8: 227-234.

20. Komarova, N.L. 2005. Mathematical modeling of tumorigenesis: mission possible. *Curr Opin Oncol*. 17: 39-43.

21. Gatenby, R.A., and Vincent, T.L. 2003. An evolutionary model of carcinogenesis. *Cancer Res.* 63: 6212-6220.

22. Gatenby, R.A., and Vincent, T.L. 2003. Application of quantitative models from population biology and evolutionary game theory to tumor therapeutic strategies. *Mol. Cancer Ther.* 2: 919-927.

23. Mueller, M.M., and Fusenig, N.E. 2004. Friends or foes – bipolar effects of the tumour stroma in cancer. *Nat. Rev. Cancer.* 4: 839-849.

24. Olumi, A.F., Grossfeld, G.D., Hayward, S.W., Carroll, P.R., Tlsty, T.D., *et. al.* 1999. Carcinoma-associated fibroblasts direct tumor progression of initiated human prostatic epithelium. *Cancer Res*. 59: 5002-5011.

25. Skobe, M, and Fusenig, N.E. 1998. Tumorigenic conversion of immortal human keratinocytes through stromal cell activation. *Proc. Natl. Acad. Sci. U.S.A.* 95: 1050-1055.

26. Ishiguro, K, Yoshida, T, Yagishita, H, Numata, Y, and Okayasu, T. 2006. Epithelial and stromal instability contributes to genesis of colorectal adenomas. *Gut*. 55: 695-702.

27. Fukino, K, Shen, L, Matsumoto, S, Morrison, C.D., Mutter, G.L., *et. al.* 2004. Combined total genome loss of heterozygosity scan of breast cancer stroma and epithelium reveals multiplicity of stromal targets. *Cancer Res*. 64: 7231-7236.

28. Cordon-Cardo, C, and Prives, C. 1999. At the crossroads of inflammation and tumorigenesis. *J. Exp. Med*. 190: 1367-1370.

29. Axelrod, R, Axelrod, D.E., and Pienta, K.J. 2006. Evolution of cooperation among tumor cells. *Proc. Natl. Acad. Sci. U.S.A.* 103: 13474-13479.

30. De Jong, J.S., Van Diest, P.J., Van Der Valk, P, and Baak, J.P. 1998. Expression of growth factors, growth-inhibiting factors, and their receptors in invasive breast cancer. II: Correlations with proliferation and angiogenesis. *J. Pathol.* 184: 53-57.

31. Royuela, M, Ricote, M, Parsons, M.S., Garcia-Tunon, I, Paniagua, R, *et. al.* 2004. Immunohistochemical analysis of the IL-6 family of cytokines and their receptors in benign, hyperplastic, and malignant human prostate. *J. Pathol*. 202: 41-49.

32. Hanski, I. 1998. Metapopulation dynamics. *Nature*. 396: 41-49.

33. Sole, R.V., Gonzalez Garcia, I, and Costa, J. 2006. Spatial dynamics in cancer. In *Complex Systems Science in Biomedicine*. T. Deisboeck and J. Kresh, editors. New York: Springer. 557-572.

34. Tarafa, G, Tuck, D, Ladner, D, Topazian, M, Brand, R, *et. al.* 2008. Mutational load distribution analysis yields metrics reflecting genetic instability during pancreatic carcinogenesis. *Proc. Natl. Acad. Sci. U.S.A.* 105: 4306-4311.

35. Yatabe, Y, Tavare, S, and Shibata, D. 2001. Investigating stem cells in human colon by using methylation patterns. *Proc. Natl. Acad. Sci. U.S.A.* 98: 10839-10844.

36. Barker, N, Ridgway, R.A., van Es, J.H., van de Wetering, M, Begthel, H, *et. al.* 2009. Crypt stem cells as the cells-of-origin of intestinal cancer. *Nature*. 457: 608-612.

37. Tan, B.T., Park, C.Y., Ailles, L.E., and Weissman, I.L. 2006. The cancer stem cell hypothesis: a work in progress. *Lab. Invest*. 86: 1203-1207.

38. Dalerba, P, Dylla, S.J., Park, I.K., Liu, R, Wang, X, *et. al*. 2007. Phenotypic characterization of human colorectal cancer stem cells. *Proc. Natl. Acad. Sci. U.S.A.* 104: 10158-10163.

39. Al-Hajj, M, Micha, M.S., Benito-Hernandez, A, Morrison, S.J., and Clarke, M.F. 2003. Prospective identification of tumorigenic breast cancer cells. *Proc. Natl. Acad. Sci. U.S.A.* 100: 3983-3988.

40. Nicolas, P, Kim, K.M., Shibata, D, and Tavare, S. 2007. The stem cell population of the human colon crypt: analysis via methylation patterns. *PloS Comput. Biol*. 3: 364-374.

41. Cairnie, A.B., and Millen, B.H. 1975. Fission of crypts in the small intestine of the irradiated mouse. *Cell Tissue Kinet*. 8: 189-196.

42. Potten, C.S., Taylor, Y, and Hendry, J.H. 1988. The doubling time of regenerating clonogenic cells in the crypts of the irradiated mouse small intestine. *Int. J. Radiat. Biol.*

54: 1041-1051.

43. Lotka, A.J. 1920. Undamped oscillations derived from the law of mass action. *J. Am. Chem. Soc.* 42: 1595-1599.

44. St Clair, W.H., and Osborne, J.W. 1985. Crypt fission and crypt number in the small and large bowel of postnatal rats. *Cell Tissue Kinet*. 18: 255-262.

45. Spencer, S.L., Gerety, R.A., Pienta, K.J., and Forrest, S. 2006. Modeling somatic evolution in tumorigenesis. *PloS Comput. Biol*. 2: 939-947.

46. Tomlinson, I.P., Novelli, M.R., and Bodmer, W.F. 1996. The mutation rate and cancer. *Proc. Natl. Acad. Sci. U.S.A.* 93: 14800-14803.

47. Tong, A.H., Evangelista, M, Parsons, A.B., Xu, H, Bader, G.D., *et. al*. 2001. Systematic genetic analysis of with ordered arrays of yeast deletion mutants. *Science*. 294: 2364-2368.

48. International Human Genome Sequencing Consortium. 2004. Finishing the euchromatic sequence of the human genome. *Nature*. 431: 931-945.

49. Katdare, M, Kopelovich, L, and Telang, N. 2001. Chemopreventive agents inhibit aberrant proliferation of the aneuploid phenotype in a colon epithelial cell line established from Apc 1638N [+/-] mouse. *Ann. N. Y. Acad. Sci.* 952: 169-174.

50. Herrero-Jimenez, P, Thilly, G, Southam, P.J., Tomita-Mitchell, A, Morgenthaler, S, *et. al.* 1998. Mutation, cell kinetics, and subpopulations at risk for colon cancer in the United States. *Mutat. Res*. 400: 553-578.

51. Fazeli, A, Steen, R.G., Dickinson, S.L., Bautista, D, Dietrich, W.F., *et. al*. 1997. Effects of p53 mutations on apoptosis in mouse in mouse intestinal and human colonic adenomas. *Proc. Natl. Acad. Sci. U.S.A.* 94: 10199-10204.

52. Biswas, S, Chytil, A, Washington, K, Romero-Gallo, J, Gorska, A.E., *et. al.* 2004. Transforming growth factor β receptor type II inactivation promotes the establishment and progression of colon cancer. *Cancer Res.* 64: 4687-4692.

53. Arends, M.J., McGregor, A.H., and Wyllie, A.H. 1994. Apoptosis is inversely related to necrosis and determines net growth in tumors bearing constitutively expressed myc, ras, and HPV oncogenes. *Am. J. Pathol*. 144: 1045-1057.

54. Wagenaar, R.A., Crawford, H.C., Matrisian, L.M. 2001. Stabilized β-catenin immortalizes colonic epithelial cells. *Cancer Res*. 61: 2097-2104.

55. Pappa, G, Bartsch, H, and Gerhauser, C. 2007. Biphasic modulation of cell proliferation by sulforaphane at physiologically relevant exposure times in a human colon

cancer cell line. *Mol Nutr Food Res*. 51: 977-984.

56. Etzioni, R, Urban, N, Ramsey, S, McIntosh, M, Schwartz, S, *et. al*. 2003. The case for early detection. *Nat. Rev. Cancer.* 3: 243-252.

57. Brown, P.O., and Palmer, C. 2009. The preclinical natural history of serous ovarian cancer: defining the target for early detection. *PloS Med.* 6: e1000114.

58. Mor, G, Visintin, I, Lai, Y, Zhao, H, Schwartz, P, *et. al*. 2005. Serum protein markers for early detection of ovarian cancer. *Proc. Natl. Acad. Sci. U.S.A*. 102: 7677-7682.

59. Visintin, I, Feng, Z, Longton, G, Ward, D.C., Alvero, A.B., *et. al*. 2008. Diagnostic markers for early detection of ovarian cancer. *Clin. Cancer Res.* 14: 1065-1072.

60. McIntosh, M.W., Liu, Y, Drescher, C, Urban, N, Diamandis, E.P. 2007. Validation and characterization of human kallikrein 11 as a serum marker for diagnosis of ovarian cancer. *Clin. Cancer Res*. 13: 4422-4428.

61. Deisboeck, T.S., and Wang, Z. 2008. A new concept for cancer therapy: out-competing the aggressor. *Cancer Cell Int.* 8: 19.

62. Miller, B.E., Miller, F.R., Leith, J, and Heppner, G.H. 1980. Growth interaction in vivo between tumor subpopulations derived from a single mouse mammary tumor. *Cancer Res.* 40: 3977-3981.

63. Jouanneau, J, Moens, G, Bourgeois, Y, Poupon, M.F., and Thiery, J.P. 1994. A minority of carcinoma cells producing acidic fibroblast growth factor induces a community effect for tumor progression. *Proc. Natl. Acad. Sci. U.S.A*. 91: 286-290.