Investigating the reproducibility of *in vitro* cell biology assays using mathematical models

Wang Jin

MSc in Mathematical Medicine and Biology

under the supervision of

Professor Matthew Simpson and Professor Scott McCue

Mathematical Sciences
Science and Engineering Faculty
Queensland University of Technology

31 July 2017

Abstract

In vitro cell biology assays are routinely used to study cancer spreading, drug design and tissue repair. However, previous studies report that results from cell biology assays are difficult to reproduce. Mathematical models have been widely used to study collective cell spreading in cell biology assays. The aim of this thesis is to analyse cell biology assays using mathematical models, to identify some of variability that affects the reproducibility. In addition, different types of mathematical models are studied to investigate their suitability. This work is presented as a thesis by published papers and consists of four related papers.

This work begins by investigating the impact of the initial degree of confluence on the reproducibility of scratch assays. Scratch assays with six different initial seeding conditions are performed. To further process the experimental images, we count individual cells and construct cell density profiles. We then calibrate the solution of the Fisher–Kolmogorov model to the data of cell density profiles to quantify the cell diffusivity D and proliferation λ for each initial seeding condition. Our results show that the cell diffusivity D varies significantly over the six initial seeding conditions. To further investigate the suitability of the Fisher–Kolmogorov model, we calibrate the solution of the Porous–Fisher model to the data. The results show that there is still a large variation in D, while the variation is smaller than that estimated from the Fisher–Kolmogorov model. Based on the results, we find that the Porous–Fisher model provides a better description of the experiment in our case, and suggest that the initial seeding condition should be reported in experimental protocols.

Our first results also show a variation in λ over different initial seeding conditions, which motivates us to investigate the cell proliferation in cell biology assays. Cell proliferation is often modelled by the logistic growth model, however, the suitability of this choice is rarely tested with experimental data. To investigate whether cells proliferate logistically, we perform both scratch assays and proliferation assays with three different initial seeding conditions. Experimental images are further processed to construct the cell density information. One feature of the logistic growth model is that the per capita growth rate is a decreasing straight line. Our per capita growth rate profiles, which are approximated from the experimental data, suggest that cells in scratch assays undergo two–phase growth: a disturbance phase in which cells do not proliferate logistically, and a logistic growth phase which takes place after the disturbance phase. For proliferation assays, there is only a logistic growth phase. Further model calibration to the data of cell density information shows that simply calibrating the solution of the logistic growth model to experimental data from scratch assays for the entire duration could result in misleading parameter estimates. Therefore, we suggest that for scratch assays the disturbance phase should be identified, and the logistic growth model should be applied only to the growth phase.

Having investigated the collective cell spreading in cell biology assays using continuum models, we then extend the proliferation mechanism in lattice–based random walks models. In these discrete models, cell proliferation is normally modelled by an unbiased nearest neighbour exclusion process. We introduce a generalised proliferation mechanism which allows cells to place daughter cells into non–nearest neighbour sites. Whether potential proliferation events are accepted depends on a crowding function which accounts for the occupancy of non–nearest neighbour sites. The continuum limit description of our generalised discrete model leads to a reaction–diffusion reaction. Comparing the solution of the continuum limit description with the averaged simulation data indicates that in general the continuum model matches the discrete model well, and the quality of the continuum–discrete match increases with the size of the template of neighbour sites.

Finally, we apply a lattice–based random walk model to mimic passaging of cell lines. In the literature, it is reported that cell proliferation changes with passage number, and seemingly contradictory effects are observed. In our discrete model, we consider both initial heterogeneity in cell proliferation and passage–induced damage on cell proliferation. Our simulation results show that due to the competition between the initial heterogeneity and the degree of the passage–induced damage, it is possible for the average proliferation rate to either increase or decrease with passage number. To investigate the impact of passage number on the reproducibility of cell biology assays, we simulate a suite of scratch assays with cells at different passage numbers. Our results show that although passage number is normally not reported in experiments, it could significantly affect the closure rate of the scratched area. Therefore we suggest that passage number should be reported in experiments to achieve better reproducibility.
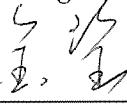
We conclude this work by outlining our main findings and contributions from both mathematical and experimental perspectives, and discussing the potential to use the combined mathematical and experimental approach to study other factors that affect the reproducibility of cell biology assays.

Declaration

I hereby declare that this submission is my own work and to
the best of my knowledge it contains no material previously
published or written by another person, nor material which to a
substantial extent has been accepted for the award of any other
degree or diploma at QUT or any other educational institution,
except where due acknowledgement is made in the thesis. Any
contribution made to the research by colleagues, with whom I
have worked at QUT or elsewhere, during my candidature, is
fully acknowledged.

I also declare that the intellectual content of this thesis is the
product of my own work, except to the extent that assistance
from others in the project's design and conception or in style,
presentation and linguistic expression is acknowledged.

Signature _____   Date ___31 / 07 / 2017___

Acknowledgements

Contents

1 Introduction

1.1 Overview

Two–dimensional *in vitro* cell biology assays are routinely used to study cancer spreading, drug design and tissue repair [2,7,58,65,76,77]. These assays provide insight into collective cell spreading, as a result of combined cell migration and cell proliferation. Generally there are two types of *in vitro* cell biology assays: (i) proliferation assays (Figure 1.1(a)), and (ii) scratch assays (Figure 1.1(b)). To perform a proliferation assay, cells grown in a cell culture are initially distributed on a two–dimensional substrate [111, 118]. Consequently, a confluent monolayer forms [111, 118]. Cells in scratch assays are initiated in the same way as proliferation assays, except that a scratch is made after cells are initially seeded. Similarly, the initially scratched area is re–colonised due to motility and proliferation events [71].

t = 0h   t = 0h

(a)   (b)
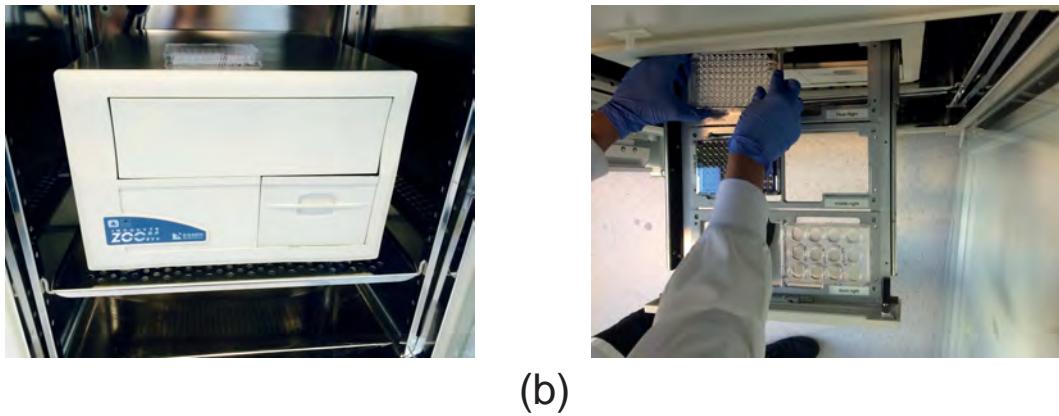
Figure 1.1: Experimental images at initial time. (a) Experimental image of a proliferation assay. (b) Experimental image of a scratch assay. The scale bar corresponds to 300 μm.

Although *in vitro* cell biology assays are popular, problems associated with reproducibility have been reported in previous studies [35,120]. Variables such as experimental equipment and environment can make reproducibility difficult [35,58,122]. For example, the type of instruments and the pressure applied when scratching could lead to significantly different sizes and shapes of the scratched area [35]. To overcome this problem, a few alternative *in*

vitro assays have been proposed to reduce sources of variability in experiments [35,58,122]. For example, IncuCyte ZOOM$^{\text{TM}}$ System (Figure 1.2) has been developed to perform both proliferation and scratch assays [29]. To address experimental limitations, IncuCyte ZOOM$^{\text{TM}}$ assays create consistent scratch areas and record high–resolution experimental images for further analysis [29]. While these alternative assays focus on minimising the variables in the experimental design [35, 58, 122], the intrinsic variables in cell lines are still overlooked.

(a) (b)

Figure 1.2: IncuCyte ZOOM$^{\text{TM}}$ assays. (a) IncuCyte ZOOM$^{\text{TM}}$ System. (b) Trays inside the IncuCyte ZOOM$^{\text{TM}}$ System.

Mathematical models have been playing an important role in understanding mechanisms in *in vitro* cell biology assays and predicting experimental observations for over two decades [14, 76, 77, 100, 104, 109, 115]. In general, there are two forms of mathematical models to describe the collective cell spreading. The first approach applies continuum reaction–diffusion equations [76, 77, 100]. In these continuum models, cell migration is modelled by a diffusion term, and the carrying–capacity limited proliferation is modelled by a logistic source term [14, 76, 77, 100]. The traditional continuum models are often of the form

$$\frac{\partial C(x,y,t)}{\partial t} = D\nabla^2 C(x,y,t) + \lambda C(x,y,t)\left(1 - \frac{C(x,y,t)}{K}\right), \tag{1.1}$$

where $C(x,y,t)$ [cells/μm^2] is the cell density, x and y are spatial coordinates [μm], t is time in [h], D [μm^2/h] is the cell diffusion coefficient, λ [/h] is the cell proliferation rate, and K [cells/μm^2] is the carrying–capacity [76, 77, 100]. In one dimension, Equation (1.1) simplifies to the Fisher–Kolmogorov model [30]. Equation (1.1) assumes that cell diffusivity is a constant, whereas other studies show that cell diffusivity depends on local cell density. Generalisations of the Fisher–Kolmogorov model, such as the Porous–Fisher model, in which cell migration is modelled by a nonlinear diffusion term, have also been applied to study cases in which cell diffusivity increases with local cell density. It is interesting to note that for proliferation assays in which cells are uniformly distributed, Equation (1.1) simplifies to the logistic growth equation [54], given by

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \lambda C(t)\left(1 - \frac{C(t)}{K}\right). \tag{1.2}$$

One feature of the logistic growth equation is that it has an exact solution, which is a sigmoid curve that monotonically increases from the initial cell density to the carrying capacity K as $t \to \infty$. The logistic growth model is often used to study carrying–capacity limited population growth, such as cell proliferation in scratch assays and proliferation assays [14, 54, 76, 77, 98, 100, 118].

The other approach to describe collective cell spreading is discrete random walk models. Based on the types of simulation domain, discrete random walk models can be further categorised into lattice–based and off–lattice models [28, 52, 80, 109]. For each type of discrete model, individual cells migrate and proliferate depending on a set of rules that are observed from experiments [80, 109]. For example, in a traditional lattice–based model, cell migration is often incorporated using an unbiased exclusion process with hard–core exclusion representing cell–to–cell crowding [109]. Cell proliferation is incorporated by placing a daughter cell into one of the surrounding sites [109]. To model crowding effects, any potential migration or proliferation events that would move a cell or place a daughter cell in an occupied site are aborted [109]. While continuum models predict results at population scale, discrete models contain details of individual cells [34, 44]. Therefore discrete models are advantageous when comparing with experimental results, such as images and time–lapse movies. However, discrete models are more computationally expensive, due to the number of realisations required to obtain statistically averaged results [52, 109, 111, 115].

While continuum and discrete models involve information at different scales, it is interesting to note that the mean–field continuum limit description of discrete models can be derived under certain circumstances [106, 107, 109]. For example, a traditional lattice–based random walk model gives rise to Equation (1.1) at the continuum limit [109]. This allows us to understand how changes in individual cells affect the collective cell behaviour at the population scale [109]. Since experimental observations often involve information at both individual and population scales, it has been of great interest to use both discrete and continuum models to study the collective cell spreading [53, 109]. However, it is worth noting that predictions from discrete models and the corresponding continuum limit descriptions do not always match well [110]. Therefore, it is important to compare the solution of the continuum limit descriptions with averaged simulation data from discrete models [110].

Recent studies have used a combination of mathematical and experimental approaches to quantify various aspects in *in vitro* cell biology assays [54, 80, 115, 118]. Using both the Fisher–Kolmogorov model and experimental data from scratch assays, Maini et al. predict that travelling wave exists in a scratch assay [76, 77]. They also find that the diffusion coefficient can be estimated based on the travelling wave solution of the Fisher–Kolmogorov model [76, 77]. Sengers et al. investigate the suitability of mathematical models for two types of cell lines, and find that the Fisher–Kolmogorov equation is suitable for only one type of the cell lines [100]. Tremel et al. quantify the *in vitro* migration and

proliferation of fibroblast cells, by using the logistic growth equation and the Fisher–Kolmogorov model to mimic proliferation assays and scratch assays, respectively [118]. Treloar et al. extract multiple types of data from circular barrier assays and apply both discrete and continuum models to quantify cell diffusivity, cell proliferation, and cell–to–cell adhesion of melanoma cells [115]. They also find that with different initial cell numbers, the estimated parameter values are not the same [115]. Johnston et al. use the Fisher–Kolmogorov model and the edge detection method to quantify the cell diffusivity, proliferation rate, and carrying capacity of prostate cancer cells from IncuCyte ZOOM$^{\text{TM}}$ assays [54]. Mort et al. apply a lattice–based discrete model to investigate the formation of diffusion clones, chimeric stripes and belly spots in a mice model [80]. Based on the simulation predictions, they conclude that a depigmented ventral belly spot in mice likely arises due to the decreased cell proliferation [80].

The combined mathematical and experimental approach has shown great capability to quantify key parameters and predict experimental observations, in order to solve problems associated with *in vitro* cell biology assays. This approach offers possibilities to look into one of the main problems in cell biology assays: reproducibility. In the following section, we identify and discuss four research questions associated with reproducibility that we will address using mathematical models.

1.2   Research questions

Although alternative *in vitro* assays have been proposed to improve experimental design and minimise experimental variability, it is reported that results from cell biology assays are difficult to reproduce [35, 120]. This suggests that there are overlooked sources of variability that affect collective cell spreading. In this thesis we analyse cell biology assays and quantify the roles of cell migration and proliferation using mathematical models, to identify and investigate some of the variability that affects reproducibility. In particular, we propose the following four questions:

1. **How does the initial degree of confluence affect the reproducibility of *in vitro* scratch assays?**

   In vitro scratch assays are routinely used to study collective cell spreading in cancers, drug design, and tissue repair [2,7,58,65,76,77]. However, it is reported that results from scratch assays are difficult to reproduce [35, 120]. This problem is partially caused by inconsistent experimental equipment and environments [35]. Alternative assays, such as IncuCyte ZOOM$^{\text{TM}}$ assays, are proposed to reduce some sources of variability in the experimental design [29]. However, some variability, such as the initial degree of confluence, is still overlooked. The initial degree of confluence describes the crowdedness of the initial cell population. Most experiments do not report any quantitative measurements of the initial degree of confluence [37,58,76,77,104]. For

those experiments in which the initial degree of confluence is reported, the measurement is merely based on a qualitative judgment [13, 58].

Previous studies have calibrated solutions of the Fisher–Kolmogorov model to experimental data of cell density profiles, to obtain the cell diffusivity and cell proliferation rate [115]. The IncuCyte ZOOM^{TM} assay can produce high–resolution experimental images with consistent scratched area, which can be further processed to produce cell density profiles [29]. Therefore, to investigate the impact of the initial degree of confluence, the combined mathematical and experimental approach allows us to obtain cell density profiles for different initial seeding conditions, and then estimate the cell diffusivity and cell proliferation rate for each initial seeding condition.

2. **Do cells in cell biology assays proliferate logistically?**

Cell proliferation is considered one of the key factors that drive collective cell spreading in *in vitro* cell biology assays [54, 71]. The continuum reaction–diffusion models that mimic certain features of cell biology assays normally include a logistic source term to model the carrying–capacity limited proliferation [100, 109, 115]. These types of models, have been calibrated to experimental data to estimate parameters, such as proliferation rate [115]. However, the suitability of using the logistic source term or the logistic growth model is rarely tested using experimental data. A few studies claim that cells do not always proliferate logistically [66, 124, 130]. For example, West et al. find that the growth of a wide range of animal models is best described by a more general model [124]. Therefore, whether cells in cell biology assays proliferate logistically still remains unclear.

The IncuCyte ZOOM^{TM} assay can produce high–resolution experimental images, which can be further processed to produce cell density information [29]. To investigate whether cells proliferate logistically, the solution of the logistic growth model can be calibrated to experimental data. Then, parameter estimates for different initial seeding conditions can be compared to test the suitability of choosing this logistic growth model.

3. **How can lattice–based discrete models incorporate a generalised proliferation mechanism in which cells do not proliferate logistically?**

Lattice–based random walk models are used to mimic cell migration and cell proliferation processes in *in vitro* cell biology assays [109, 115]. Traditional lattice–based random walk models involve a nearest–neighbour proliferation mechanism, in which potential proliferation events are aborted if the target site is occupied [109]. In addition, daughter cells can only be placed into nearest–neighbour sites [109]. However, it is biologically unrealistic to abort potential proliferation events solely depending

on whether a single, randomly chosen nearest–neighbour site is occupied, as cell–to–cell crowding plays an impotent role in cell proliferation [109]. In addition, the continuum limit description of this proliferation mechanism leads to the traditional logistic source term in reaction–diffusion equations, whereas previous studies show that cells do not always proliferate logistically [66, 124, 130]. Therefore, using the traditional lattice–based random walk models to mimic cell proliferation which does not follow the logistic growth could lead to misleading results and affect the outcomes of *in silico* experiments.

One way of proposing a more biologically realistic proliferation mechanism is to consider non–nearest neighbour proliferation events over a larger template of lattice sites. Instead of checking the occupancy of a single, randomly chosen target site, the new proliferation mechanism calculates the agent density $\hat{C}$ within the template. Whether potential proliferation events are accepted depends on a crowding function $f(\hat{C})$. The continuum limit description of this generalised discrete model can be derived, and we can compare the solution of the continuum description to the simulation data from the discrete model.

4. **How does cell proliferation change at high passage numbers? How does this change affect *in vitro* cell biology assays?**

In vitro cell culture is used to grow and supply cells for various types of experiments, such as cell biology assays [2]. Passaging is a process to further grow cell population in cell culture after cells become confluent [8, 9]. During the passaging process, cells are lifted, split into smaller proportions, and re-grown in new cell culture flasks [8, 9]. It is reported that cell characteristics change at high passage numbers, and seemingly contradictory effects have been observed in experiments of passaging cell lines [21, 31, 45, 73, 88, 95, 99]. For example, many studies report that cell proliferation decreases with the increasing passage number [31, 45, 95, 99], whereas Lin et al. find that cell proliferation increases at high passage numbers [73]. Therefore, the change of cell characteristics associated with the passage number could possibly lead to the reproducibility problem in *in vitro* experiments.

Lattice–based random walk models have been used to mimic certain features in *in vitro* cell biology assays [109]. However, previous studies of discrete simulations have not investigated the passaging process. To model the cell culture growth, we run a simulation until a certain degree of confluence is reached. To model the passaging process, a small number of cells are chosen randomly, lifted from the current simulation domain, and re-distributed randomly into an empty domain to re-grow. To model the passage–induced damage, we assume that during the passaging process, each cell's proliferation rate is decreased by a random amount. We also simulate a suite of cell biology assays using cells with different passage numbers, to investigate

how passaging affects the reproducibility of cell biology assays.

1.3   Aims and outcomes of this thesis

The primary aim of this thesis is to use mathematical models to analyse *in vitro* cell biology assays and identify some of variability that affects the reproducibility. Meanwhile, different mathematical models are studied to investigate whether they are appropriate for modelling certain features in cell biology assays.

This thesis has four objectives:

- Investigate how the initial degree of confluence affects the parameter estimates of cell diffusivity and proliferation rate in *in vitro* scratch assays.

- Investigate the cell proliferation mechanisms in *in vitro* cell biology assays, using a combined mathematical and experimental approach.

- Develop a cell proliferation mechanism in lattice–based, random walk models, which consider more biologically realistic crowding effects.

- Use lattice–based random walk models to investigate the passaging process and how passaging of cell lines affects the reproducibility of *in vitro* cell biology assays.

This thesis is presented by publication, with two papers that have been published, one paper accepted, and one paper submitted in high–quality peer–reviewed journals. The PhD candidate has contributed significantly and is the primary author of all the four papers. The work presented in this thesis fulfils the requirements for the award of thesis by published papers at Queensland University of Technology.

This thesis consists of the four following publications:

- **Jin, W.**, Shah, E. T., Penington, C. J., McCue, S. W., Chopin, L. K., & Simpson, M. J. (2016). Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection. *J. Theor. Biol.*, 390, 136–145. (Chapter 2)
- **Jin, W.**, Shah, E. T., Penington, C. J., McCue, S. W., Maini, P. K., & Simpson, M. J. (2017). Logistic proliferation of cells in scratch assays is delayed. *Bull. Math. Biol.*, 79, 1028–1050. (Chapter 3)
- **Jin, W.**, Penington, C. J., McCue, S. W., & Simpson, M. J. (2016). Stochastic simulation tools and continuum models for describing two–dimensional collective cell spreading with universal growth functions. *Phys. Biol.*, 13, 056003. (Chapter 4)

- **Jin, W.**, Penington, C. J., McCue, S. W., & Simpson, M. J. (2017). A computational modelling framework to quantify the effects of passaging cell lines. *PLOS ONE*. (Chapter 5)

In this thesis each chapter is comprised of a paper, which means that each chapter can be read individually. To make the style and layout of each chapter consistent, minor changes have been made to each of the papers, while most of the contents in Chapter 2–5 are exactly the same as the original papers that have been either published or submitted. Consequently, some overlapped contents appear in Background Section and Methods Section. Chapters 2 to 5, each contain a Background Section that introduces background knowledge and reviews relevant literature, a Methods Section demonstrating the relevant experimental and mathematical methods, a Results Section illustrating the experimental and computational results, and a Discussion Section discussing the results.

In the next section, we outline the structure of this thesis and the novel contributions of the work for each chapter.

1.4   Structure of this thesis

The structure of this thesis is as follows:

In Chapter 1 we introduce the background of *in vitro* cell biology assays and review the experimental and mathematical literature related to this work. We then address the research questions and aims of this thesis. The structure of this thesis and the outcomes are also outlined in this chapter.

In Chapter 2 we investigate how the initial degree of confluence affects the reproducibility of scratch assays. Using a combined mathematical and experimental approach, we perform scratch assays with six different initial seeding conditions. The experimental data are further processed to construct cell density profiles. We then use mathematical models, namely the Fisher–Kolmogorov model and the Porous–Fisher model, to quantify the cell diffusivity D and proliferation rate λ, and investigate whether cell migration and cell proliferation change with the initial seeding condition.

The results in Chapter 2 motivate us to further investigate the proliferation mechanisms in cell biology assays. In the literature, cell proliferation is routinely modelled by the logistic growth source term in reaction–diffusion equations or the logistic growth model. However, the suitability of this choice is rarely tested with experimental data. Therefore, it still remains unclear whether cells proliferate logistically in cell biology assays. Following the same approach as in Chapter 1, we perform scratch assays and proliferation assays with three different initial seeding conditions. The experimental images are then further processed by counting individual cells to construct cell density information. One feature of the logistic growth model is that the per capita growth rate profile is a straight line. To identify the growth profile, we also construct the per capita growth rate profiles using

the cell density information. We then calibrate the solution of the logistic growth model to the data of cell density information, to quantify the parameter of proliferation rate λ.

Chapter 4 focuses on the development of a traditional lattice–based random walk model for cell biology assays. We first generalise the proliferation mechanism by imposing a larger size of the template for neighbour sites in which non–nearest neighbour proliferation is allowed. We then develop a new mechanism for cells to sense the occupancy of their neighbour sites, and whether potential proliferation events are accepted depends on a crowding function associated with the occupancy. We also derive the continuum limit description for our generalised discrete model. In the Results section, we compare the match between the discrete model and its continuum limit description. The limitations and further work are discussed in the Discussion section.

In Chapter 5 we apply a traditional lattice–based random walk model to mimic the passaging process and investigate its impact on the reproducibility of *in vitro* cell biology assays. We first outline the novel passaging mechanism that accounts for the initial heterogeneity in cell proliferation and the passage–induced damage on cell proliferation. We then use our discrete model to investigate the seemingly contradictory effects caused by the passaging of cell lines reported in the literature. To investigate the impact of passage number on the reproducibility of cell biology assays, we simulate a suite of scratch assays with cells at different passage numbers and compare the difference in the scratch closure.

We summarise our work in Chapter 6, outlining our aims and main findings. We then discuss possible future work based on the results achieved and the limitations of our work in Chapter 2–5.

1.5   Statement of joint authorship

In this section, we outline the contribution of the PhD candidate and the co–authors to each paper. All co–authors have consented to the presentation of this material in this thesis.

Chapter 2: Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection

The associated paper for this chapter is:
Jin, W., Shah, E. T., Penington, C. J., McCue, S. W., Chopin, L. K., & Simpson, M. J. (2016). Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection. *J. Theor. Biol.*, 390, 136–145.

Abstract
Scratch assays are difficult to reproduce. Here we identify a previously overlooked source of variability which could partially explain this difficulty. We analyse a suite of scratch assays in which we vary the initial degree of confluence (initial cell density). Our results

indicate that the rate of re–colonisation is very sensitive to the initial density. To quantify the relative roles of cell migration and proliferation, we calibrate the solution of the Fisher–Kolmogorov model to cell density profiles to provide estimates of the cell diffusivity, D, and the cell proliferation rate, λ. This procedure indicates that the estimates of D and λ are very sensitive to the initial density. This dependence suggests that the Fisher–Kolmogorov model does not accurately represent the details of the collective cell spreading process, since this model assumes that D and λ are constants that ought to be independent of the initial density. Since higher initial cell density leads to enhanced spreading, we also calibrate the solution of the Porous–Fisher model to the data as this model assumes that the cell flux is an increasing function of the cell density. Estimates of D and λ associated with the Porous–Fisher model are less sensitive to the initial density, suggesting that the Porous–Fisher model provides a better description of the experiments.

Statement of joint authorship

The work was divided as follows:

- **Jin, W. (Candidate)** extracted and further processed experimental data, performed data analysis, computed the solutions of mathematical models, composed all figures and supplementary material, and critically reviewed and revised the manuscript.
- Shah, E. T. performed the experiments and wrote the experimental methods.
- Penington, C. J. provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- McCue, S. W. oversaw and directed the research, provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- Chopin, L. K. provided assistance with the experiments and biological interpretations.
- Simpson, M. J. initiated the concept for this paper, oversaw and directed the research, wrote the manuscript, drafted and redrafted the manuscript, wrote the cover and revision letter, critically reviewed and revised the manuscript and supplementary material, and acted as the corresponding author.

Chapter 3: Logistic proliferation of cells in scratch assays is delayed

The associated paper for this chapter is:
Jin, W., Shah, E. T., Penington, C. J., McCue, S. W., Maini, P. K., & Simpson, M. J. (2017). Logistic proliferation of cells in scratch assays is delayed. *Bull. Math. Biol.*, 79, 1028–1050.

Abstract

Scratch assays are used to study how a population of cells re–colonises a vacant region on a two–dimensional substrate after a cell monolayer is scratched. These experiments are used in many applications including drug design for the treatment of cancer and chronic wounds. To provide insights into the mechanisms that drive scratch assays, solutions of

continuum reaction–diffusion models have been calibrated to data from scratch assays. These models typically include a logistic source term to describe carrying capacity-limited proliferation, however the choice of using a logistic source term is often made without examining whether it is valid. Here we study the proliferation of PC-3 prostate cancer cells in a scratch assay. All experimental results for the scratch assay are compared with equivalent results from a proliferation assay where the cell monolayer is not scratched. Visual inspection of the time evolution of the cell density away from the location of the scratch reveals a series of sigmoid curves that could be naively calibrated to the solution of the logistic growth model. However, careful analysis of the per capita growth rate as a function of density reveals several key differences between the proliferation of cells in scratch and proliferation assays. Our findings suggest that the logistic growth model is valid for the entire duration of the proliferation assay. On the other hand, guided by data, we suggest that there are two phases of proliferation in a scratch assay; at short time we have a *disturbance phase* where proliferation is not logistic, and this is followed by a *growth phase* where proliferation appears to be logistic. These two phases are observed across a large number of experiments performed at different initial cell densities. Overall our study shows that simply calibrating the solution of a continuum model to a scratch assay might produce misleading parameter estimates, and this issue can be resolved by making a distinction between the disturbance and growth phases. Repeating our procedure for other scratch assays will provide insight into the roles of the disturbance and growth phases for different cell lines and scratch assays performed on different substrates.

Statement of joint authorship

The work was divided as follows:

- **Jin, W. (Candidate)** extracted and further processed experimental data, performed data analysis, computed the solutions of mathematical models, composed all figures and supplementary material, wrote the manuscript and supplementary material, and critically reviewed and revised the manuscript.
- Shah, E. T. performed the experiments and wrote the experimental methods.
- Penington, C. J. provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- McCue, S. W. oversaw and directed the research, provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- Maini, P. K. provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- Simpson, M. J. initiated the concept for this paper, oversaw and directed the research, contributed to the writing of the manuscript, oversaw drafting and redrafting of the manuscript, wrote the cover and revision letter, critically reviewed and revised the manuscript and supplementary material, and acted as the corresponding author.

Chapter 4: Stochastic simulation tools and continuum models for describing two–dimensional collective cell spreading with universal growth functions

The associated paper for this chapter is:
Jin, W., Penington, C. J., McCue, S. W., & Simpson, M. J. (2016). Stochastic simulation tools and continuum models for describing two–dimensional collective cell spreading with universal growth functions. *Phys. Biol.*, 13, 056003.

Abstract

Two–dimensional collective cell migration assays are used to study cancer and tissue repair. These assays involve combined cell migration and cell proliferation processes, both of which are modulated by cell–to–cell crowding. Previous discrete models of collective cell migration assays involve a nearest–neighbour proliferation mechanism where crowding effects are incorporated by aborting potential proliferation events if the randomly chosen target site is occupied. There are two limitations of this traditional approach: (i) it seems unreasonable to abort a potential proliferation event based on the occupancy of a single, randomly chosen target site; and, (ii) the continuum limit description of this mechanism leads to the standard logistic growth function, but some experimental evidence suggests that cells do not always proliferate logistically. Motivated by these observations, we introduce a generalised proliferation mechanism which allows non–nearest neighbour proliferation events to take place over a template of $r \geq 1$ concentric rings of lattice sites. Further, the decision to abort potential proliferation events is made using a *crowding function*, $f(C)$, which accounts for the density of agents within a group of sites rather than dealing with the occupancy of a single randomly chosen site. Analysing the continuum limit description of the stochastic model shows that the standard logistic source term, $\lambda C(1-C)$, where λ is the proliferation rate, is generalised to a universal growth function, $\lambda C f(C)$. Comparing the solution of the continuum description with averaged simulation data indicates that the continuum model performs well for many choices of $f(C)$ and r. For nonlinear $f(C)$, the quality of the continuum–discrete match increases with r.

Statement of joint authorship
The work was divided as follows:

- **Jin, W. (Candidate)** implemented the modelling framework and methodology, computed the numerical simulations, composed all figures and supplementary material, wrote the manuscript and supplementary material, and critically reviewed and revised the manuscript.
- Penington, C. J. provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- McCue, S. W. oversaw and directed the research, provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- Simpson, M. J. initiated the concept for this paper, oversaw and directed the research, contributed to the writing of the manuscript, oversaw drafting and redrafting

of the manuscript, wrote the cover and revision letter, critically reviewed and revised the manuscript and supplementary material, and acted as the corresponding author.

Chapter 5: A computational modelling framework to quantify the effects of passaging cell lines

The associated paper for this chapter is:

Jin, W., Penington, C. J., McCue, S. W., & Simpson, M. J. (2017). A computational modelling framework to quantify the effects of passaging cell lines. *PLOS ONE*. (In press)

Abstract

In vitro cell culture is routinely used to grow and supply a sufficiently large number of cells for various types of cell biology experiments. Previous experimental studies report that cell characteristics evolve as the passage number increases, and various cell lines can behave differently at high passage numbers. To provide insight into the putative mechanisms that might give rise to these differences, we perform *in silico* experiments using a random walk model to mimic the *in vitro* cell culture process. Our results show that it is possible for the average proliferation rate to either increase or decrease as the passaging process takes place, and this is due to a competition between the initial heterogeneity and the degree to which passaging damages the cells. We also simulate a suite of scratch assays with cells from near–homogeneous and heterogeneous cell lines, at both high and low passage numbers. Although it is common in the literature to report experimental results without disclosing the passage number, our results show that we obtain significantly different closure rates when performing *in silico* scratch assays using cells with different passage numbers. Therefore, we suggest that the passage number should always be reported to ensure that the experiment is as reproducible as possible. Furthermore, our modelling also suggests some avenues for further experimental examination that could be used to validate or refine our simulation results.

Statement of joint authorship

The work was divided as follows:

- **Jin, W. (Candidate)** implemented the modelling framework, computed numerical the simulations, composed all figures and supplementary material, wrote the manuscript and supplementary material, and critically reviewed and revised the manuscript.
- Penington, C. J. provided technical assistance, helped interpret the results and critically reviewed the manuscript.
- McCue, S. W. oversaw and directed the research, provided technical assistance, helped interpret the results and critically reviewed the manuscript.

- Simpson, M. J. initiated the concept for this paper, oversaw and directed the research, contributed to the writing of the manuscript, oversaw drafting and redrafting of the manuscript, wrote the cover letter, critically reviewed and revised the manuscript and supplementary material, and acted as the corresponding author.

2 Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection

Abstract

Scratch assays are difficult to reproduce. Here we identify a previously overlooked source of variability which could partially explain this difficulty. We analyse a suite of scratch assays in which we vary the initial degree of confluence (initial cell density). Our results indicate that the rate of re–colonisation is very sensitive to the initial density. To quantify the relative roles of cell migration and proliferation, we calibrate the solution of the Fisher–Kolmogorov model to cell density profiles to provide estimates of the cell diffusivity, D, and the cell proliferation rate, λ. This procedure indicates that the estimates of D and λ are very sensitive to the initial density. This dependence suggests that the Fisher–Kolmogorov model does not accurately represent the details of the collective cell spreading process, since this model assumes that D and λ are constants that ought to be independent of the initial density. Since higher initial cell density leads to enhanced spreading, we also calibrate the solution of the Porous–Fisher model to the data as this model assumes that the cell flux is an increasing function of the cell density. Estimates of D and λ associated with the Porous–Fisher model are less sensitive to the initial density, suggesting that the Porous–Fisher model provides a better description of the experiments.

2.1 Background

Two–dimensional *in vitro* cell migration assays are routinely used to investigate the ability of cell populations to re–colonise an initially–vacant region. The most common type of *in vitro* cell migration assay is called a scratch assay, which is performed by: (i) growing a cell

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

16

monolayer; (ii) removing a region of the monolayer by scratching it with a sharp–tipped instrument; and (iii) making observations of the re–colonisation of the initially–vacant, scratched region [7, 65]. Comparing the rate of scratch closure in an experiment where cells are exposed to a chemical stimulus to the rate of closure in a control assay provides insight into the roles of growth factors and putative drug treatments relevant to malignant spreading and tissue repair [7, 65].

Although scratch assays remain popular, various alternative *in vitro* assays have been proposed. These alternatives, including circular barrier assays [122], circular invasion assays [58] and IncuCyte ZOOM$^{\text{TM}}$ assays [29], are often claimed to be superior because of issues associated with scratch assay reproducibility [35]. The purpose of these alternative assays is to reduce some source of variability in the experimental design. For example, standard scratch assays can be performed with various types of instruments (e.g. pipette tip, razor blade) and with varying degrees of pressure. Each of these variables is thought to have the potential to affect the results in some way. To address these limitations, the IncuCyte ZOOM$^{\text{TM}}$ real time live cell imaging assays have been developed [29]. IncuCyte ZOOM$^{\text{TM}}$ assays use a mechanical tool, called a WoundMaker$^{\text{TM}}$, to create 96 identically–sized scratches in each well of a 96–well tissue culture plate. Each WoundMaker$^{\text{TM}}$ scratch has the same dimensions, and is created with the same amount of pressure.

In this work we explore a previously overlooked source of variability that has the potential to impact the interpretation of various types of cell migration assays. While standard experimental procedures for many cell migration assays require that a sufficient amount of time is allowed for the population to become confluent before the experiment is initiated [7, 65], most experiments do not report any quantitative measurements of the initial degree of confluence [37, 58, 76, 77, 104]. Many experimental protocols simply state that the monolayer is either fully confluent [58] or 80% confluent [13] prior to making a scratch. These reports of the *degree of confluence* are typically based on a qualitative judgment rather than quantitative measurements. To investigate the significance of this, we perform a suite of IncuCyte ZOOM$^{\text{TM}}$ assays, using PC–3 prostate cancer cells [57], in which we systematically vary the initial density. A qualitative comparison of the experimental images suggests that the rate of scratch closure is extremely sensitive to the initial density. This dependence on the initial density is important since experimental results are almost always reported without any quantitative measurement of the initial density. This could explain why scratch assays are difficult to reproduce.

We make quantitative measurements of the IncuCyte ZOOM$^{\text{TM}}$ assays by extracting cell density profiles and calibrating the solution of the Fisher–Kolmogorov model [30, 64], to that data. This procedure provides an estimate of the cell diffusivity, D, and the cell proliferation rate, λ, for each initial density considered. This procedure is standard; however, typical approaches deal with just one initial density [14, 38, 76, 77, 98, 100, 104]. The appropriately calibrated solutions of the Fisher–Kolmogorov equation match our experimental observations, for each initial density, very well. However, our estimates of D and λ appear to depend upon the initial cell density, and our estimates of D are extremely

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

17

sensitive. This result has two implications. First, additional mechanisms, unaccounted
for in the Fisher–Kolmogorov model, are likely to be acting in the experimental system.
To explore this possibility we also examine the suitability of some potential extensions of
the Fisher–Kolmogorov model, such as the Porous–Fisher equation [100,104,110,125,126].
Second, our results suggest that previously reported procedures for estimating D and λ
by calibrating the solution of the Fisher–Kolmogorov equation could provide misleading
results.

2.2   Methods

We perform monolayer scratch assays using the IncuCyte ZOOM$^{\text{TM}}$ system (Essen Bio-
Science, MI USA). All experiments are performed using the PC–3 prostate cancer cell
line [57] from the American Type Culture Collection (ATCC, Manassas, USA). Cells
are propagated in RPMI 1640 medium (Life Technologies, Australia) in 10% foetal calf
serum (Sigma–Aldrich, Australia), with 100 U/mL penicillin, 100 μg/mL streptomycin
(Life Technologies), in plastic flasks (Corning Life Sciences, Asia Pacific) in 5% CO_2
and 95% air in a Panasonic incubator (VWR International) at 37°C. Cells are regularly
screened for *Mycoplasma* (Nested PCR using primers from Sigma–Aldrich).

Cells grown to approximately 80% confluence are removed from the plastic flask using
TrypLE$^{\text{TM}}$ (Life Technologies) in phosphate buffered saline (pH 7.4), resuspended in
medium and seeded at various densities in 96–well ImageLock plates (Essen BioScience).
Cells are distributed in the wells as uniformly as possible. We report results for initial
cell densities of 10,000, 12,000, 14,000, 16,000, 18,000 and 20,000 cells per well. After
seeding, cells are grown overnight to allow for attachment and some growth. We use a
WoundMaker$^{\text{TM}}$ (Essen BioScience) to create uniform, reproducible scratches in all the
wells of a 96–well plate. To ensure that as many cells are removed from the wound
region as possible, we modify the manufacturer's protocol by repeating the scratch action
20 times before lifting the WoundMaker$^{\text{TM}}$. After creating the scratch, the medium is
aspirated and the wells are washed twice with fresh medium to remove cells from the
scratched area. Following the washes, 100 μL of fresh medium is added to each well and
the plate is placed into the IncuCyte ZOOM$^{\text{TM}}$ apparatus. Images of the collective cell
spreading are recorded every two hours, for 48 hours. For each different cell density, we
perform three identically prepared experimental replicates ($n = 3$).

We obtain numerical solutions to various parabolic reaction–diffusion models using a finite
difference method [81]. The spatial domain, $0 < x < L_x$, is uniformly discretised with
grid spacing δx, and the spatial derivatives are approximated using a central–difference
approximation. Some of the models we consider involve a nonlinear diffusion term, which
is discretised with an arithmetically averaged inter–node diffusivity. For all models con-
sidered, spatial discretisation leads to a system of coupled nonlinear ordinary differen-
tial equations that are integrated through time using a backward–Euler approximation
with constant time steps of duration δt [81]. The systems of coupled nonlinear algebraic

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

18

equations are linearised using Picard (fixed–point) iteration, with absolute convergence tolerance ϵ, and solved using the Thomas algorithm [81]. For all results we choose δx, δt and ϵ so that our algorithm produces grid–independent results.

2.3 Results

2.3.1 Qualitative assessment of experiments
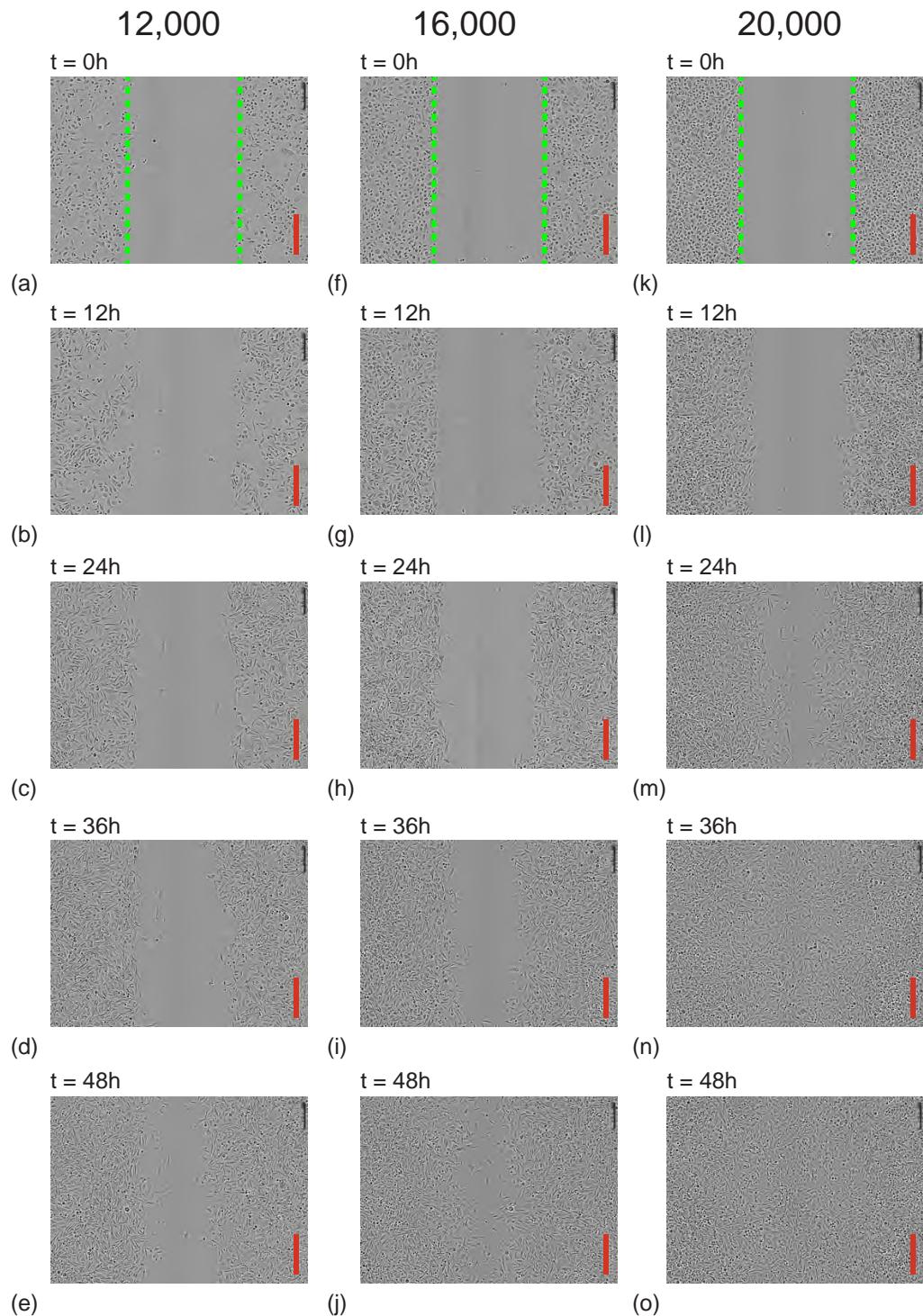
A subset of the experimental images are presented in Figure 2.1 for the experiments initiated with 12,000, 16,000 and 20,000 cells per well. Images in Figure 2.1(a), (f) and (k) show that each experiment is initiated with a clean, sharp scratch. The initial difference in cell density is visually distinct in the regions well behind the position of the scratch. The temporal progression of each experiment is shown in the columns of Figure 2.1. In each case we see evidence of combined cell migration and cell proliferation. Cells located near the edge of the scratched region move into the vacant region over time. Cells are also proliferating since we see the cell density behind the location of the scratch increasing with time. Interestingly, if we compare the final images of each experiment, in Figure 2.1(e), (j) and (o), a large portion of the initially–vacant wound space in Figure 1(e) remains uncolonised, whereas the total area imaged in Figure 2.1(o) appears to be colonised by $t = 48$ h.

Our visual interpretation of these images indicates that the ability of PC–3 cells to re–colonise the wound space is very sensitive to the initial density of cells. This observation is important because many *in vitro* experiments do not report any quantitative measurement of the initial degree of confluence [35, 37, 58, 76, 77], and our results suggest that it would be very difficult to replicate this kind of experiment unless the initial degree of confluence is measured and reported.

Although our visual interpretation of the results in Figure 2.1 implies that the rate of closure is very sensitive to the initial density, it is not obvious at this stage whether the differences in the rate of re–colonisation are caused by: (i) differences in the initial cell density alone; (ii) differences in the rate of cell migration; (iii) differences in the rate of cell proliferation; or (iv) differences in both the rates of cell migration and cell proliferation. To address this question, we now interpret our experimental results quantitatively by measuring the temporal evolution of the spatial cell density profiles and using this data to calibrate a suite of mathematical models to the data.
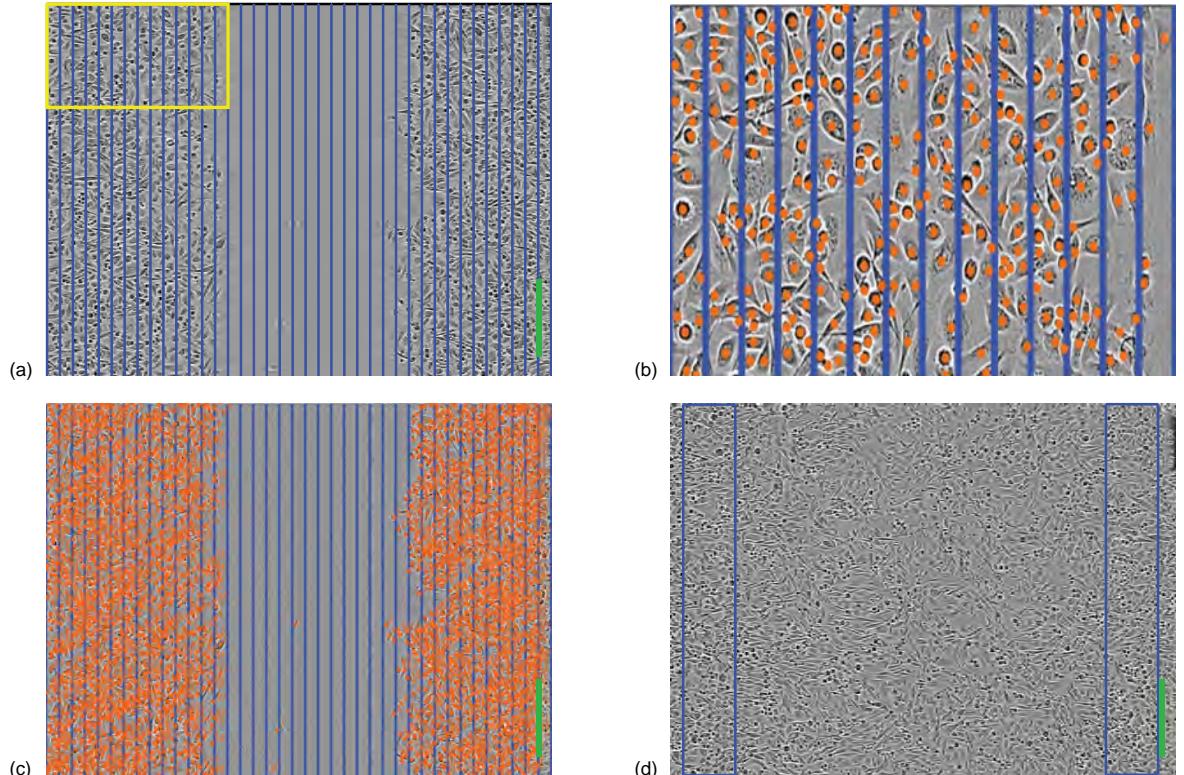
2.3.2 Quantitative assessment of experiments

We divide each image into 39 equally–spaced columns [18, 60, 61]. Each column is 50 μm wide, as shown in Figure 2.2(a). We do not use data from the right most column since the imaging system superimposes a scale bar in that column which partially obscures certain cells. We count the numbers of cells in the remaining columns, $i = 1, 2, 3, \ldots, 38$, and

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

19
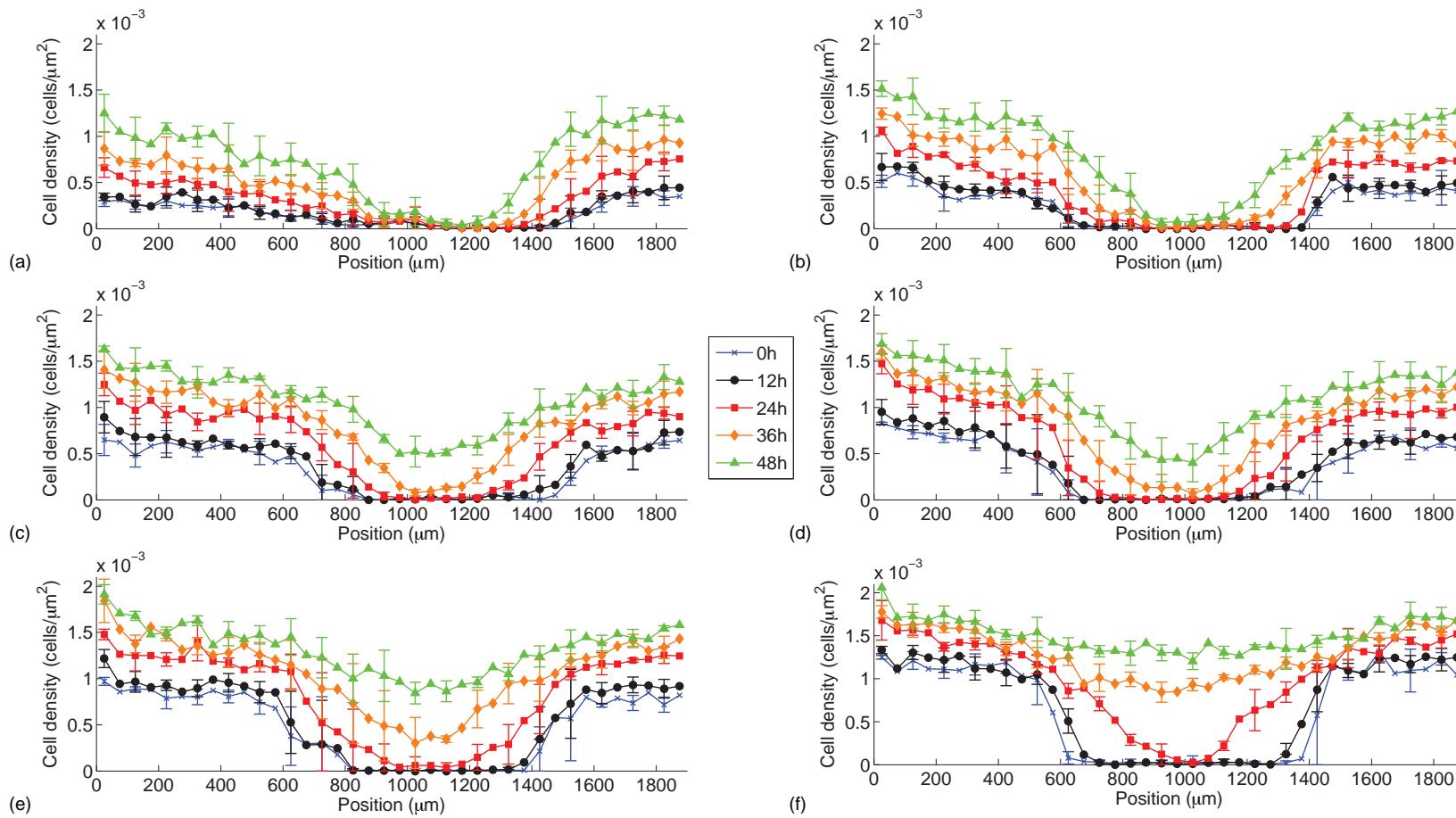
Figure 2.1: A summary of IncuCyte ZOOM^{TM} experiments. Images correspond to an experiment
initiated with: (a)–(e) 12,000; (f)–(j) 16,000; and (k)–(o) 20,000 cells per well. The time that the image
is recorded is indicated on each subfigure and the scale bar corresponds to 300 μm. The images in (a),
(f) and (k), at $t = 0$ h, show the approximate location of the position of the leading edge (dashed green).

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

20

divide the total number of cells per column by the column area to give an estimate of the cell density in each column. We repeat this process for each replicate and calculate the sample mean and sample standard deviation of the cell density in each column for $t = 0, 12, 24, 36$ and 48 h (Online Supplementary Material). Plots of the mean cell density, with error bars indicating the sample deviation, are given in Figure 2.3. In general, we see that the variability between the different experimental replicates for each initial condition is small (Online Supplementary Material). In contrast, the evolution of the cell density profiles with different initial conditions gives vastly different results. For example, the cell density profiles in Figure 2.3(a), shows that the initial location of the scratch, where the local density is approximately zero, is evident after $t = 48$ h. In contrast, the profiles in Figure 2.3(f) indicate that the population becomes almost uniformly confluent after $t = 48$ h. Overall, the general trends in Figure 2.3 show that the experiments with a higher initial density leads to a more rapid re–colonisation.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

21

Figure 2.2: To quantify the cell density profile, each image is divided into vertical columns of width 50 μm, as shown in (a), from which manual cell counting is used to estimate the cell density at positions $x = 25, 75, 125, \ldots, 1925$ μm. To count and record the locations of individual cells we zoom in to focus on certain subregions, such as the subregion shown in (b), which corresponds to the rectangle highlighted in (a). Using the counting features in Adobe Photoshop, we identify individual cells and place a unique marker on each cell (red disk) as shown in (b). After each image is processed in this way we have then identified the location and total number of cells in each image, as shown in (c). The average cell density profile is calculated by averaging results for three identically–prepared experimental replicates. To quantify the carrying capacity density, K, we count the cell density in two regions at $t = 48$ h. The locations of the two regions are indicated in (d). These two regions are located well behind the initial position of the leading edge. Estimates of K are obtained using three identically–prepared experimental replicates and the results are averaged to give $K = 1.7 \times 10^{-3}$ cells/μm^2. The scale bar in each image corresponds to 300 μm. These particular experimental images correspond to one of the replicates from an experiment initialised with 20,000 cells per well. All parameter estimates reported in this caption are given to two significant figures.

Figure 2.3: Cell density profiles. Results in (a)–(f) correspond to experiments initiated with 10,000, 12,000, 14,000, 16,000, 18,000 and 20,000 cells per well, respectively. For each different initial condition, experimental cell density profiles are shown at $t = 0, 12, 24, 36$ and 48 h. Each experiment was repeated $n = 3$ times. The average density corresponds to the sample mean and the error bars indicate the variability, corresponding to the sample standard deviation.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

23

2.3.3   Measuring the carrying capacity density K

The carrying capacity density, $K > 0$, is the maximum density for a monolayer of cells. Comparing various cell density profiles in Figure 2.3, we see that the experiments initialised with 20,000 cells become more confluent by $t = 48$ h than the experiments initialised with smaller numbers of cells. Therefore, to estimate K we focus on the data from experiments initialised with 20,000 cells at $t = 48$ h, to ensure that our estimate of K corresponds to a confluent population. We estimate K by calculating the cell density in two subregions, each of width 200 μm, located well–behind the initial position of the front, as indicated in Figure 2.2(d). We count the total number of cells in these two subregions and divide by the total area. Repeating this procedure for our three identically–prepared experimental replicates gives: $K = 1.71 \times 10^{-3}, 1.74 \times 10^{-3}$ and 1.69×10^{-3} cells/μm^2. Therefore, our estimate is $K = 1.7 \times 10^{-3} \pm 2.5 \times 10^{-5}$ cells/μm^2, where the variability is indicated by the sample standard deviation. Since the variability is very small, with the coefficient of variation just 1.5%, our estimate is precise. Our estimate is consistent with previously research. For example, Cai et al. report $K = 1.0 \times 10^{-3}$ cells/μm^2 for fibroblasts, whereas Treloar et al. report $K = 1.6 \times 10^{-3}$ cells/μm^2 for 3T3 fibroblast cells in a circular barrier assay.

Our estimate of K is consistent with certain features of the cell density profiles in Figure 2.3 since our measurements of $C(x, t)$ appear to approach K after a sufficiently large amount of time has elapsed. This is evident in Figure 2.3(e)–(f) where the population becomes almost uniformly confluent by $t = 48$ h and we have $C(x, t) \to K^-$ as t becomes sufficiently large. Since our experimental data contains fluctuations (Online Supplementary Material). and our estimate of K represents an average, we do occasionally observe estimates of cell density that exceed our estimate of K.

2.3.4   Fisher–Kolmogorov model

The Fisher–Kolmogorov model has been used to describe collective cell spreading in both *in vitro* [76, 77, 100, 104] and *in vivo* [47, 70, 93] contexts. Calibrating the solution of the Fisher–Kolmogorov model to experimental data can be used to quantify the roles of cell migration and cell proliferation [76, 77, 100]. Given our estimates of the average cell density profiles in Figure 2.3 together with our estimate of K, we will calibrate the solution of the Fisher–Kolmogorov model to our data. In this work we apply

$$\frac{\partial C}{\partial t} = \mathcal{D}\frac{\partial^2 C}{\partial x^2} + \lambda C \left[1 - \left(\frac{C}{K}\right)\right], \tag{2.1}$$

on $0 \leq x \leq 1900$ μm, where $\mathcal{D} = D$ is the cell diffusivity and λ is the cell proliferation rate. The cell proliferation rate is related to the cell doubling time, $t_d = \ln(2)/\lambda$. To determine the appropriate boundary conditions for Equation (2.1), we recall that the diameter of the 96–well plate is 9000 μm whereas the width of view of the experimental images is just 1950 μm. Therefore, the actual distribution of cells in each well extends far beyond the vertical

Chapter 2. Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection

24

boundaries of the images in Figure 2.1. Since the cells are initially distributed uniformly, and the distribution of cells appears to remain approximate spatially uniform far away from the edges of the scratch for the duration of the experiment, we impose zero net flux boundary conditions at both $x = 0$ μm and $x = 1900$ μm [11]. The initial condition for Equation (2.1) is specified using the mean cell density data obtained by direct cell counting at $t = 0$ h. The data in Figure 2.3 at $t = 0$ h gives the average cell density at the center of each column in Figure 2.2(a), corresponding to $x = 25, 75, 125, \ldots, 1925$ μm, for the six different experiments with different initial densities. We specify the continuous initial condition, $C(x, 0)$, for the numerical solution of Equation (2.1) by linearly interpolating these measurements at $t = 0$ h.

While our experiments are genuinely two–dimensional, we always consider a special class of problems where the initial density is, on average, independent of the vertical location in the field of view. Under these conditions it is standard to quantify the cell density profiles by dividing the experimental image into many equally–spaced columns and to report the cell density using a one–dimensional column–averaged cell density and to calibrate the solution of a one–dimensional reaction–diffusion equation to that data [18, 60, 61]. Some of our previous analysis has shown that, in general, there can be an error introduced by approximating a two–dimensional nonlinear reaction–diffusion equation by a vertically–averaged one–dimensional reaction–diffusion equation [107, 108]. However, when we consider a special initial condition where the density is independent of vertical location (Figure 2.1), the averaging error vanishes [108].

To estimate D and λ we minimise a least–squares error describing the discrepancy between the solution of Equation (2.1) and the average cell density profiles. The least–squares error is given by

$$E(D, \lambda) = \sum_{i=1}^{38} \sum_{j=1}^{4} \left[C^{\text{model}}(x_i, t_j) - C^{\text{data}}(x_i, t_j) \right]^2, \tag{2.2}$$

where $C^{\text{model}}(x, t)$ is the numerical solution of Equation (2.1) and $C^{\text{data}}(x, t)$ is the average cell density data. The index i indicates the position along the x coordinate where the cell density is measured so that $i = 1, 2, 3, \ldots, 38$. The index j indicates the time so that $j = 1, 2, 3$ and 4 corresponds to $t = 12, 24, 36$ and 48 h. We denote the least–squares estimate of D and λ as $\bar{D}$ and $\bar{\lambda}$, respectively, such that $E(\bar{D}, \bar{\lambda})$ is a minimum.

In the literature, cell diffusivities are reported to be in the range of 10 to 3000 μm^2/h [14, 76, 77, 100, 122]. Similarly, cell doubling times are typically reported to be within the range of 10–30 h [53]. Therefore, we begin our parameter estimation by conservatively limiting our search to within $0 \leq D \leq 4000$ μm^2/h and $0 \leq \lambda \leq 0.1$ /h. Using numerical solutions of Equation (2.1), we plot $E(D, \lambda)$ for each set of experimental data in Figure 2.4(a)–(f). Each subfigure in Figure 2.4(a)–(f) shows $E(D, \lambda)$ constructed by considering a grid of 100 equally–spaced intervals of D, and 50 equally–spaced values of λ. Visual inspection of $E(D, \lambda)$ in Figure 2.4(a)–(f) confirms that each error surface appears to contain a well–defined minimum, from which we estimate $\bar{D}$ and $\bar{\lambda}$. After we have obtained our initial

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

25
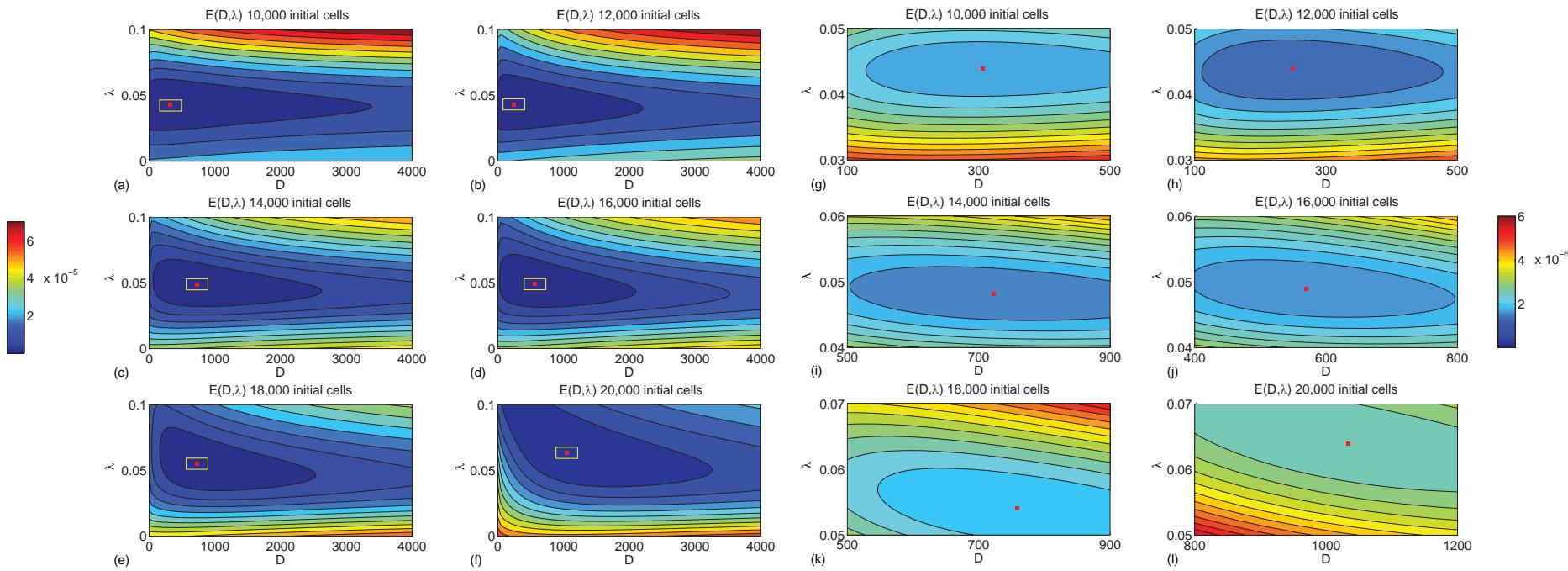
estimates using the surfaces in Figure 2.4(a)–(f), we refine our estimates by identifying a subregion surrounding the point $(\bar{D}, \bar{\lambda})$, shown in Figure 2.4(a)–(f) as a white rectangle, and we refine our search within this subregion. Results in Figure 2.4(g)–(l) show a refined plot of $E(D, \lambda)$ constructed using a grid of 100 equally–spaced intervals of D and 50 equally–spaced values of λ within the subregions identified in Figure 2.4(a)–(f). The refined plots of $E(D, \lambda)$ in Figure 2.4(g)–(l) focus on the subregion with $\bar{D} - 200 \leq D \leq \bar{D} + 200$ μm^2/h and $\bar{\lambda} - 0.01 \leq \lambda \leq \bar{\lambda} + 0.01$ /h. Each refined plot of $E(D, \lambda)$ in Figure 2.4(g)–(l) also appears to contain a well–defined minimum from which we can identify refined estimates of $(\bar{D}, \bar{\lambda})$. Using this approach we found that our refined estimates of $\bar{D}$ and $\bar{\lambda}$ obtained from Figures 2.4(g)–(l) are not very different from the original estimates identified in Figures 2.4(a)–(f). Therefore, we make no additional refinements, and a summary of our refined estimates is given in Table 2.1.

To provide an additional check on our estimates of $\bar{D}$ and $\bar{\lambda}$ in Table 2.1, we also used these estimates, together with our measurements K, as an initial guess for a MATLAB Levenberg Marquardt–based calibration algorithm [20]. We found that applying the MATLAB routine produced parameter estimates that are very similar to those in Table 2.1, so we do not report any additional results from the MATLAB algorithm here. Instead, we prefer to manually explore the error surface. This allows us to visualise the shape of the error surface, providing us with additional qualitative and quantitative information about the quality of the match between the mathematical model and the experimental data.

Table 2.1: Estimates of $\bar{D}$ and $\bar{\lambda}$ for the Fisher–Kolmogorov model obtained by calibrating the solution of Equation (2.1) to the average cell density profiles for all six different initial conditions. The right–most column gives the doubling time, $t_d = \ln(2)/\bar{\lambda}$. E_{min} represents the minimum least-squares error. All parameter estimates are given to two significant figures.
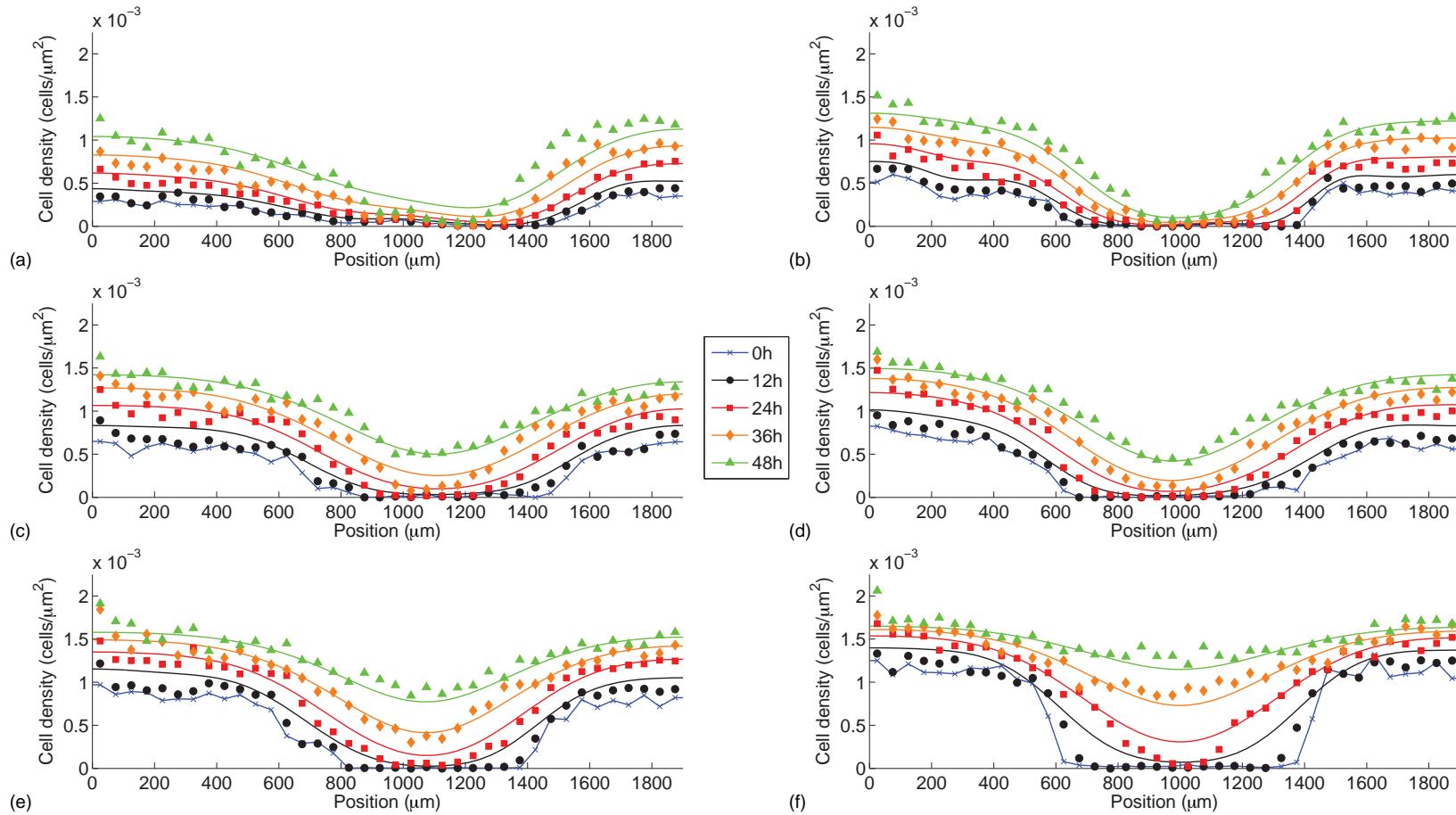
Initial number of cells	$\bar{D}$ (μm^2/h)	$\bar{\lambda}$ (/h)	t_d (h)	E_{min} ((μm^2/h)^2)
10,000	310 ± 130	0.044 ± 0.005	15.75	1.8×10^{-6}
12,000	250 ± 140	0.044 ± 0.002	15.75	1.3×10^{-6}
14,000	720 ± 60	0.048 ± 0.001	14.44	1.5×10^{-6}
16,000	570 ± 250	0.049 ± 0.003	14.15	1.6×10^{-6}
18,000	760 ± 80	0.054 ± 0.003	12.84	1.9×10^{-6}
20,000	1030 ± 200	0.064 ± 0.001	10.83	2.5×10^{-6}

Figure 2.4: (a)–(f) Error surface profiles, $E(D, \lambda)$, for the Fisher–Kolmogorov model. The error surface profiles, $E(D, \lambda)$, are constructed using 100 equally–spaced values of D in $0 \leq D \leq 4000$ μm^2/h, and 50 equally–spaced values of λ in $0 \leq \lambda \leq 0.1$ /h. The value of $E(D, \lambda)$ in (a)–(f) is shown on the left–most colour bar. (g)–(l) Refined estimates of $E(D, \lambda)$ centered about the minima identified in (a)–(f) (white rectangle). The value of $E(D, \lambda)$ in (g)–(l) is shown on the right–most colour bar. The location of $\bar{D}$ and $\bar{\lambda}$ in each case is shown as a red square. Estimates of $(\bar{D}, \bar{\lambda})$ correspond to: (a) $(320, 0.043)$, (b) $(240, 0.043)$, (c) $(730, 0.049)$, (d) $(570, 0.049)$, (e) $(730, 0.055)$, (f) $(1050, 0.063)$, (g) $(310, 0.044)$, (h) $(250, 0.044)$, (i) $(720, 0.048)$, (j) $(570, 0.049)$, (k) $(760, 0.054)$, (l) $(1030, 0.064)$. All parameter estimates reported in this caption are given to two significant figures.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

27

Our estimates of D and λ in Table 2.1 show several interesting trends, the most obvious being that we obtain very different estimates of $\bar{D}$ and $\bar{\lambda}$ for each initial condition. In general, both $\bar{D}$ and $\bar{\lambda}$ increase with the initial density of cells. In particular, we have an approximately 45% variation in $\bar{\lambda}$ and an approximately 310 % variation in $\bar{D}$ across the different experiments with different initial conditions. Therefore, our estimates of $\bar{D}$ appear to be more sensitive to variations in the initial density rather than $\bar{\lambda}$. We note that the estimates of $\bar{D}$ and $\bar{\lambda}$ are obtained by calibrating the solution of Equation (2.1) to the average cell density profiles shown in Figure 2.3. In addition, we also calibrated the solution of Equation (2.1) to each of the three individual cell density profiles from each experimental replicate to provide three additional estimates of the cell diffusivity and the cell proliferation rate for each initial density of cells. By calculating the sample standard deviation from these additional estimates we also report, in Table 2.1, an estimate of the variability in our estimates of $\bar{D}$ and $\bar{\lambda}$ between the three experimental replicates for each different initial cell density. The variability in our estimates of $\bar{D}$ and $\bar{\lambda}$ between each experimental replicate with the same initial density is much smaller than the variability we observe between different experiments with different initial cell density.

To demonstrate the quality of match between the experimental data and the calibrated solution the Fisher–Kolmogorov model, we superimpose, in Figure 2.5, the experimental data and the calibrated solution of Equation (2.1) for each initial condition. For each of the six different initial conditions we solve Equation (2.1) numerically setting $D = \bar{D}$ and $\lambda = \bar{\lambda}$ using the values reported in Table 2.1. The quality of match between the calibrated Fisher–Kolmogorov equation and the experimental data is excellent.

Figure 2.5: Calibrated solutions of Equation (2.1) to the averaged cell density profiles. Results in (a)–(f) show the average cell density for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equation (2.1) with the least squares estimates of $\bar{D}$ and $\bar{\lambda}$ from Table 2.1. All results correspond to $K = 1.7 \times 10^{-3}$ cells/μm^2 and the numerical solution of Equation (2.1) is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

29

Our approach of calibrating the Fisher–Kolmogorov model to each data set for the various
initial conditions separately reveals several important insights. For example, had we
followed a standard approach and studied just one initial condition (e.g. [76, 77, 100]), we
would have identified estimates of D and λ for which our numerical solution of Equation
(2.1) would match the experimental data extremely well. However, had another researcher
attempted to repeat our experiments and parameter estimation procedure, it is likely
that a small change in the initial density could lead to a large change in the estimates
of $\bar{D}$ and $\bar{\lambda}$. This is an important limitation since many cell migration assays neglect to
measure the initial density (e.g. [37, 76, 77, 104]). Therefore, given the extreme sensitivity
of the experimental outcomes to the initial density, it is essential that the initial degree
of confluence ought to be measured and reported if the experiment is to be reproductive.

Our finding, that $\bar{D}$ and $\bar{\lambda}$ appear to depend on the initial density, implies that the
Fisher–Kolmogorov model does not apply to our experimental system. Since the Fisher–
Kolmogorov model has constant coefficients, an implicit assumption in the application
of this model is that D and λ are independent of the cell density. However, we find
that $\bar{D}$ and $\bar{\lambda}$ are very sensitive to the initial density. This motivates us to consider
what kind of additional mechanisms might be relevant. One particular criticism of the
Fisher–Kolmogorov model is that the linear diffusion term neglects to account for any
cell–to–cell adhesion effects [1, 25, 60, 61, 83]. It is of interest to note that our experimental
results imply that higher initial density leads to enhanced cell spreading (Figure 2.1).
This is inconsistent with the idea that cell–to–cell adhesion plays an important role in
our experiments, since larger initial cell densities would lead to more frequent cell–to–cell
collisions and adhesion would lead to a reduced collective spreading of the population. To
capture potential adhesive effects, many continuum models take the form of a nonlinear
diffusion mechanism where the cell diffusivity, $\mathcal{D}$, is a decreasing function of C [1, 25, 60,
61, 83]. In contrast, our results imply that larger initial cell density is associated with
increased cell spreading. This motivates us to consider a model in which $\mathcal{D}$ is an increasing
function of C.

2.3.5 Porous–Fisher model

Although many studies have examined various theoretical aspects of extensions of the
traditional constant coefficient Fisher–Kolmogorov model (e.g. [22, 40, 43, 125, 126]), less
attention has been devoted to determining whether collective cell spreading is best mod-
elled using the constant coefficient Fisher–Kolmogorov model or a variable coefficient
generalisation of the Fisher–Kolmogorov model. We will now briefly survey the most rel-
evant studies that have begun to explore this question. Maini, McElwain and Leavesley
model the motion of the position of the cell front in an *in vitro* scratch assay using the
Fisher–Kolmogorov equation [76, 77]. Although they find that the Fisher–Kolmogorov
model is consistent with their experimental observations, they conclude by suggesting
that their analysis could be improved by considering the Porous–Fisher model in which

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

30

the cell flux is governed by a nonlinear diffusion term $\mathcal{D} = D(C/K)$, so that the diffusivity increases with C. Different with the Fisher–Kolmogorov model, the Porous–Fisher model has a characteristic sharp front, which is suitable for scratch assays in which the vacant area is clearly defined. Sherratt and Murray model the closure of a radial wound using both the Fisher–Kolmogorov equation and the Porous–Fisher model [104]. Sherratt and Murray find that both reaction–diffusion models can be used to replicate their experimental observations [104]. Similarly, Sengers et al. collect data from two different cell types in a series of *in vitro* cell spreading experiments and calibrate the solutions of the Fisher–Kolmogorov and Porous–Fisher equations to their data [100]. Sengers et al. find that the Fisher–Kolmogorov model fits the experimental data for one particular cell line very well, whereas the Porous–Fisher model fits the experimental data for the other cell line very well. This suggests that some cell lines might be best described by the Fisher–Kolmogorov model while others might be best described by the Porous–Fisher model. While these four previous studies are unique in that they directly compare the relative performance of the Fisher–Kolmogorov and Porous–Fisher models, we note that they did not consider the impact of varying the initial density of cells in their experimental system [76, 77, 100, 104].

We now calibrate the solution of the Porous–Fisher model to our data. For our work we consider

$$\frac{\partial C}{\partial t} = D \frac{\partial}{\partial x}\left[\left(\frac{C}{K}\right)\frac{\partial C}{\partial x}\right] + \lambda C \left[1 - \left(\frac{C}{K}\right)\right], \tag{2.3}$$

on $0 \leq x \leq 1900\,\mu\text{m}$. We impose the same initial and boundary conditions used in Section 3.4. The only difference between Equations (2.1) and (2.3) is the nonlinear diffusion mechanism in Equation (2.3), where $\mathcal{D} = D(C/K)$. In particular, both models contain the same two unknown parameters: D and λ.

To estimate D and λ in Equation (2.3) we followed a similar approach that we use when working with Equation (2.1) by plotting $E(D, \lambda)$, given by Equation (2.2), and restricting the range of D and λ to be the same as before. The error surfaces in Figures 2.4 and 2.6 appear to be different. For example, the error surface in Figure 2.6(a) has a less well defined minimum than the surface in Figure 2.4(a). The error surface in Figure 2.6(a) contains a valley–shaped region, parallel to the D–axis, whereas the surface in Figure 2.4(a) does not. Regardless of this difference, we follow the same procedure used previously to identify $\bar{D}$ and $\bar{\lambda}$ in Figure 2.6(a)–(f). We also refine our estimates by identifying a subregion about the point $(\bar{D}, \bar{\lambda})$ in Figure 2.6(a)–(f) and plotting $E(D, \lambda)$ centered about this point in Figure 2.6(g)–(l). Estimates of $\bar{D}$ and $\bar{\lambda}$ from the refined subregion in Figure 2.6(g)–(l) are given in Table 2.2.
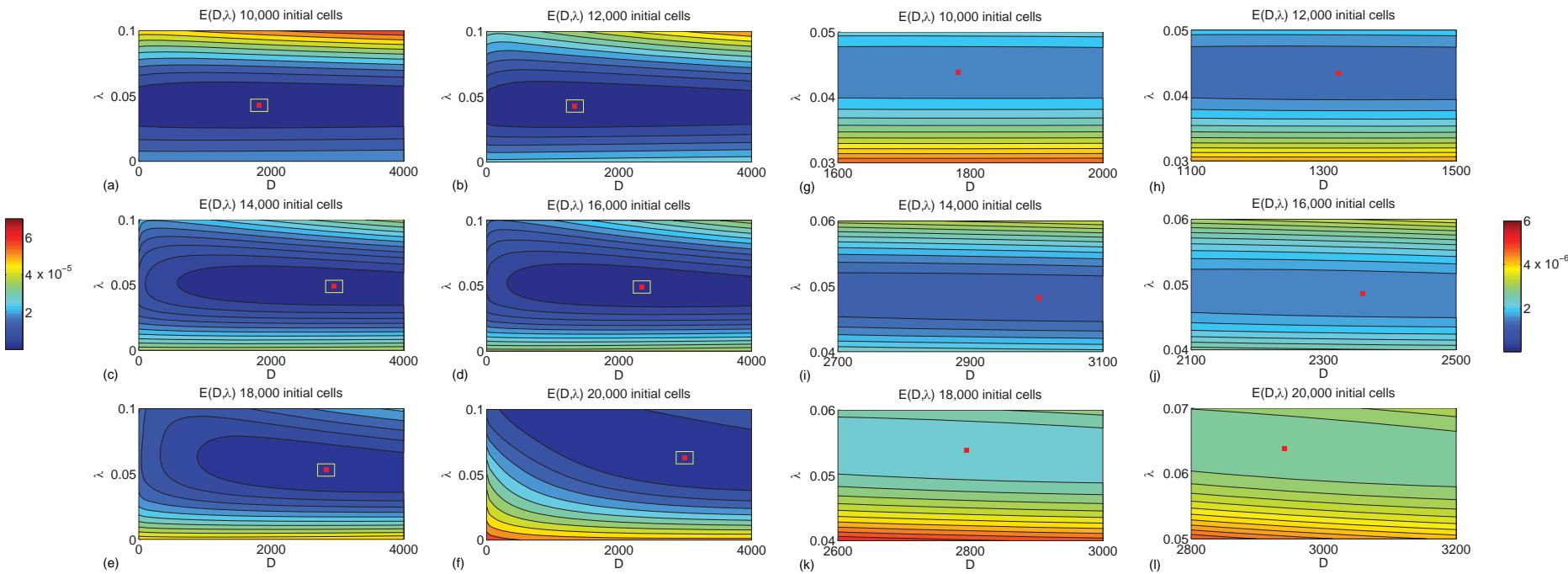
Our estimates of $\bar{\lambda}$ for the Porous–Fisher model are very similar to our estimates for the Fisher–Kolmogorov model, and we observe a variation in $\bar{\lambda}$ of approximately 49% between the different initial conditions. We also observe a variation in our estimates of $\bar{D}$ for the Porous–Fisher model of approximately 130%, which is much smaller than the variation in $\bar{D}$ for the Fisher–Kolmogorov model. To demonstrate the quality of match between the

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

31

experimental data and the solution the Porous–Fisher model we superimpose, in Figure 2.7, the experimental data and the solution of Equation (2.3) for each initial condition, and in each case we solve Equation (2.3) numerically with $D = \bar{D}$ and $\lambda = \bar{\lambda}$ reported in Table 2.2. The quality of match between the calibrated Porous–Fisher equation and the experimental data is excellent.
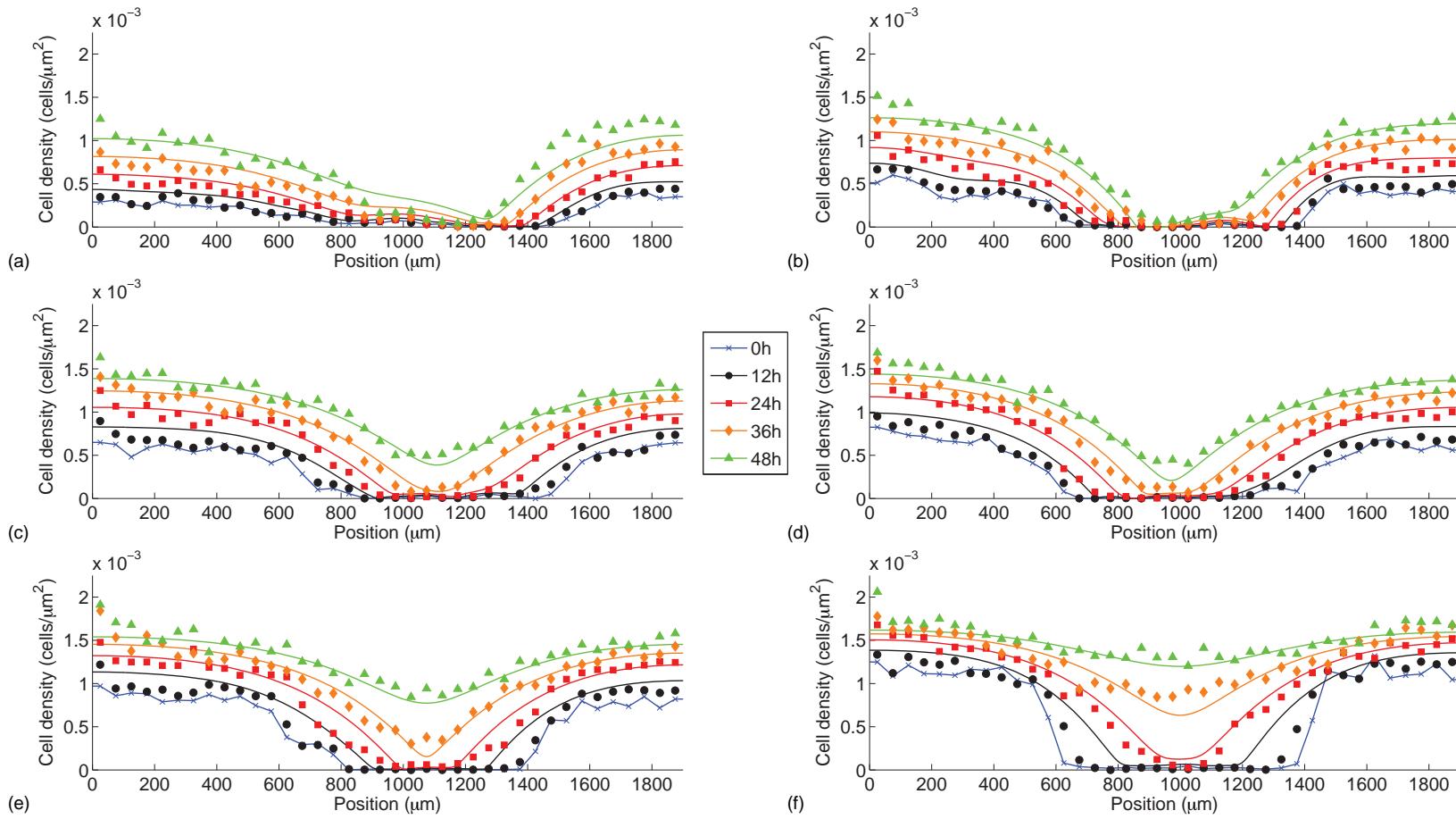
Table 2.2: Estimates of $\bar{D}$ and $\bar{\lambda}$ for the Porous–Fisher model obtained by calibrating the solution of Equation (2.3) to the average cell density profiles for all six different initial conditions. The right–most column gives the doubling time, $t_d = \ln(2)/\bar{\lambda}$. E_{min} represents the minimum least-squares error. All parameter estimates are given to two significant figures.

Initial number of cells	$\bar{D}$ (μm^2/h)	$\bar{\lambda}$ (/h)	t_d (h)	E_{min} ((μm^2/h)^2)
10,000	1800 ± 1050	0.044 ± 0.005	15.75	1.6×10^{-6}
12,000	1300 ± 930	0.043 ± 0.002	16.12	1.4×10^{-6}
14,000	3000 ± 190	0.048 ± 0.001	14.44	1.1×10^{-6}
16,000	2400 ± 990	0.049 ± 0.003	14.15	1.5×10^{-6}
18,000	2800 ± 100	0.054 ± 0.004	12.84	2.3×10^{-6}
20,000	2900 ± 620	0.064 ± 0.002	10.83	2.7×10^{-6}

Figure 2.6: (a)–(f) Error surface profiles, $E(D, \lambda)$, for the Porous–Fisher model. The error surface profiles, $E(D, \lambda)$, are constructed using 100 equally–spaced values of D in $0 \leq D \leq 4000$ μm^2/h and 50 equally–spaced values of λ in $0 \leq \lambda \leq 0.1$ /h. The value of $E(D, \lambda)$ in (a)–(f) is shown on the left–most colour bar. (g)–(l) Refined estimates of $E(D, \lambda)$ centered about the minimum identified in (a)–(f) (white rectangle). The value of $E(D, \lambda)$ in (g)–(l) is shown on the right–most colour bar. The location of $\bar{D}$ and $\bar{\lambda}$ in each case is shown as a red square. Estimates of $(\bar{D}, \bar{\lambda})$ correspond to: (a) $(1800, 0.043)$, (b) $(1300, 0.043)$, (c) $(2900, 0.049)$, (d) $(2300, 0.049)$, (e) $(2800, 0.053)$, (f) $(3000, 0.063)$, (g) $(1800, 0.044)$, (h) $(1300, 0.043)$, (i) $(3000, 0.048)$, (j) $(2400, 0.049)$, (k) $(2800, 0.054)$, (l) $(2900, 0.064)$. All parameter estimates reported in this caption are given to two significant figures.

Figure 2.7: Calibrated solutions of Equation (2.3) to the averaged cell density profiles. Results in (a)–(f) show the average cell density (circles) for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equation (2.3) with the least squares estimates of D and λ from Table 2.2. All results correspond to $K = 1.7 \times 10^{-3}$ cells/μm^2 and the numerical solution of Equation (2.3) is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

34

The question of whether our experimental data is best described by the Fisher–Kolmogorov or the Porous–Fisher model is a delicate one. Since our estimates of $\bar{D}$ and $\bar{\lambda}$ for the Porous–Fisher model are less sensitive to the initial density than for the Fisher–Kolmogorov model, it appears that the Porous–Fisher model provides a better description of our experimental data. Although the Porous–Fisher model is preferable in this regard, our parameter estimation procedure still implies that $\bar{D}$ and $\bar{\lambda}$ appear to depend on the initial density of cells for the Porous–Fisher model, suggesting that further model refinements could be warranted. Some previous studies suggest a further generalisation of the Porous–Fisher model where the nonlinear diffusivity function is generalised to $\mathcal{D} = D(C/K)^m$ [43,104]. This means that setting $m = 0$ in the generalised Porous–Fisher model corresponds to the Fisher–Kolmogorov model. We also calibrate solutions of the Porous–Fisher model to our data with $m = 1/2, 2, 3$ and 4. The calibrated model provides a poor match to the data with $m = 2, 3$ and 4, whereas we observe a good match with $m = 1/2$ and $m = 1$. The variation in $\bar{D}$ and $\bar{\lambda}$ is smaller for $m = 1$ than for $m = 1/2$, and therefore we conclude that the standard Porous–Fisher model with $m = 1$ outperforms the generalised Porous–Fisher model with the other values of m that we consider (Supplementary material). We also consider calibrating a model of chemokinesis to our data (Supplementary material) and demonstrate that this kind of detailed, coupled model, also has the potential to describe our experimental data.

2.4 Discussion

In this work we explore a previously overlooked source of variability which affects the reproducibility of scratch assays. In a standard experiment, the initial degree of confluence is neither measured or varied [7,65]. To explore the significance of this, we perform a suite of cell migration assays in which we deliberately vary the initial cell density. Our results show that the rate of re–colonisation is extremely sensitive to the initial density. Therefore, if a scratch assay is to be reproducible, the initial density must be measured and reported.

To quantify the rate of cell migration and proliferation we use a cell counting procedure to estimate the average cell density profiles and the carrying capacity density, K. Estimates of $\bar{D}$ and $\bar{\lambda}$ are obtained by calibrating the Fisher–Kolmogorov model to the data for each initial condition. The calibrated solutions of the Fisher–Kolmogorov model match the experimental cell density profiles, for each initial condition, very well. However, comparing estimates of $\bar{D}$ and $\bar{\lambda}$ for different initial conditions shows that different initial conditions lead to very different estimates of $\bar{D}$ and $\bar{\lambda}$. In particular, $\bar{\lambda}$ varies by approximately 45% whereas $\bar{D}$ varies by approximately 310% across our six different initial conditions. This result is novel since previous studies have considered just one initial condition, and our results show that this standard approach can give very misleading results.

Since our data implies that larger initial cell densities lead to enhanced migration, we calibrate the Porous–Fisher model, with $\mathcal{D}(C) = D(C/K)$, to our data. This exercise

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

35

shows that the Porous–Fisher model matches the evolution of each of our experiments
with estimates of $\bar{D}$ and $\bar{\lambda}$ that are less variable between experiments with different initial
conditions than the standard Fisher–Kolmogorov model, suggesting that the Porous–
Fisher model provides an improved match to our experimental data. To explore further
potential refinements, we also calibrated other reaction–diffusion models to the data,
but we found that these extensions offered no advantage over the Porous–Fisher model
(Supplementary material). Another feature of our analysis is that the error surfaces are
more sensitive to variations in λ than D. This could be associated with the way that
we have imaged and analysed the experiments. Most of the cells in the field of view are
associated with the dense unscratched regions where the dynamics of the experiments
are largely associated with proliferation rather than migration. This means that we
characterise D using data from the scratched region where the density is, by definition,
low, and hence more variable. This could explain why the error surfaces tend to be more
dependent on λ than D.

All of the extensions of the Fisher–Kolmogorov model that we consider deal with intro-
ducing more detail into the flux term of the reaction–diffusion model. We deliberately
choose to focus on the flux term because our estimates of $\bar{D}$ are far more sensitive to vari-
ations in the initial condition than $\bar{\lambda}$ in the proliferation term. In principle, it is possible
to examine other extensions where the proliferation term, $\lambda C(1 - C/K)$, is generalised to
$\lambda C^{\alpha}(1 - [C/K]^{\beta})^{\gamma}$, where α, β and γ are real positive constants [119]. We do not pursue
this idea here because the variation in $\bar{\lambda}$ for the six different initial conditions is much
smaller compared to the variation in $\bar{D}$.

In summary, our study reveals insights that have both experimental and mathematical
significance. From an experimental point of view, our results give a partial explanation
about why scratch assays are difficult to reproduce. We recommend that measurements of
the initial cell density must be reported when describing the outcomes of a scratch assay.
From a mathematical point of view we show that a standard approach of calibrating the
solution of the Fisher–Kolmogorov model to a scratch assay with a single initial condition
can produce misleading estimates of $\bar{D}$ and $\bar{\lambda}$. Since our estimates of $\bar{D}$ and $\bar{\lambda}$ appear
to depend on the initial cell density, we suggest that a refined version of this constant
coefficient model is warranted and we show that calibrating the Porous–Fisher model to
our data leads to an improved outcome in this regard.

The results in this study also raise a question, such that whether the information contained
within experiments is sufficient to accurately estimate the parameters that best represent
the cell characteristics. For example, starting with a very high initial cell density and
performing the experiment for a short period of time can lead to less accurate parameter
estimates. Therefore, when calibrating models to experiment data, we suggest that cell
density should be measured at smaller spatial and temporal intervals to obtain more
accurate parameter estimates. In addition, data of individual cell tracking with a small
temporal interval could provide more details of individual cell migration, which helps
identify the cell motility as a function of cell density and any bias in migration direction.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

36

2.5   Supplementary material
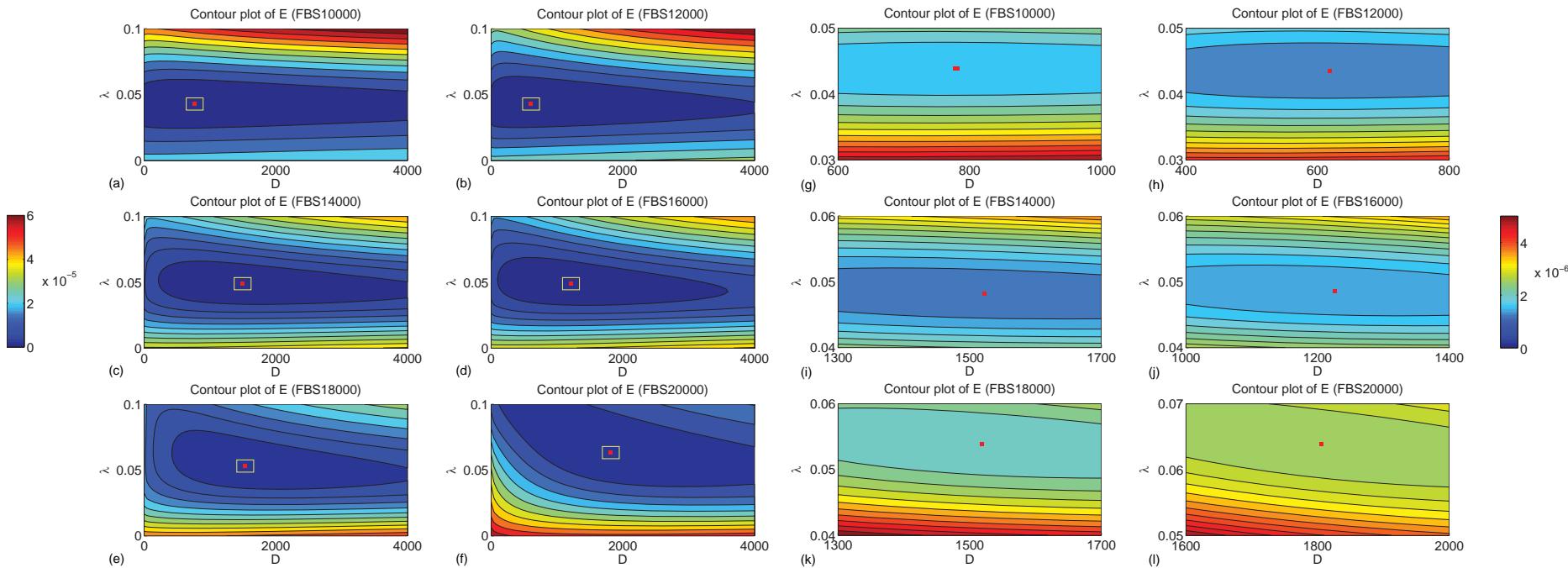
2.5.1   Generalised Porous–Fisher model

In addition to the standard Porous–Fisher model, we also consider a generalisation of the Porous–Fisher model that can be written as,

$$\frac{\partial C}{\partial t} = D\frac{\partial}{\partial x}\left[\left(\frac{C}{K}\right)^m \frac{\partial C}{\partial x}\right] + \lambda C\left[1 - \left(\frac{C}{K}\right)\right], \tag{2.4}$$

where all terms are the same as in Section 2.3.5 except that we have an additional parameter m, which we take to be a positive constant [43]. Results in Section 2.3.5 describe fitting the Porous–Fisher equation to the experimental data for $m = 1$. Since previous studies have applied the Porous–Fisher model to experimental data for $m = 4$ [104], we repeat the model calibration procedure outlined in the main paper for four additional cases: $m = 1/2, 2, 3$ and 4.
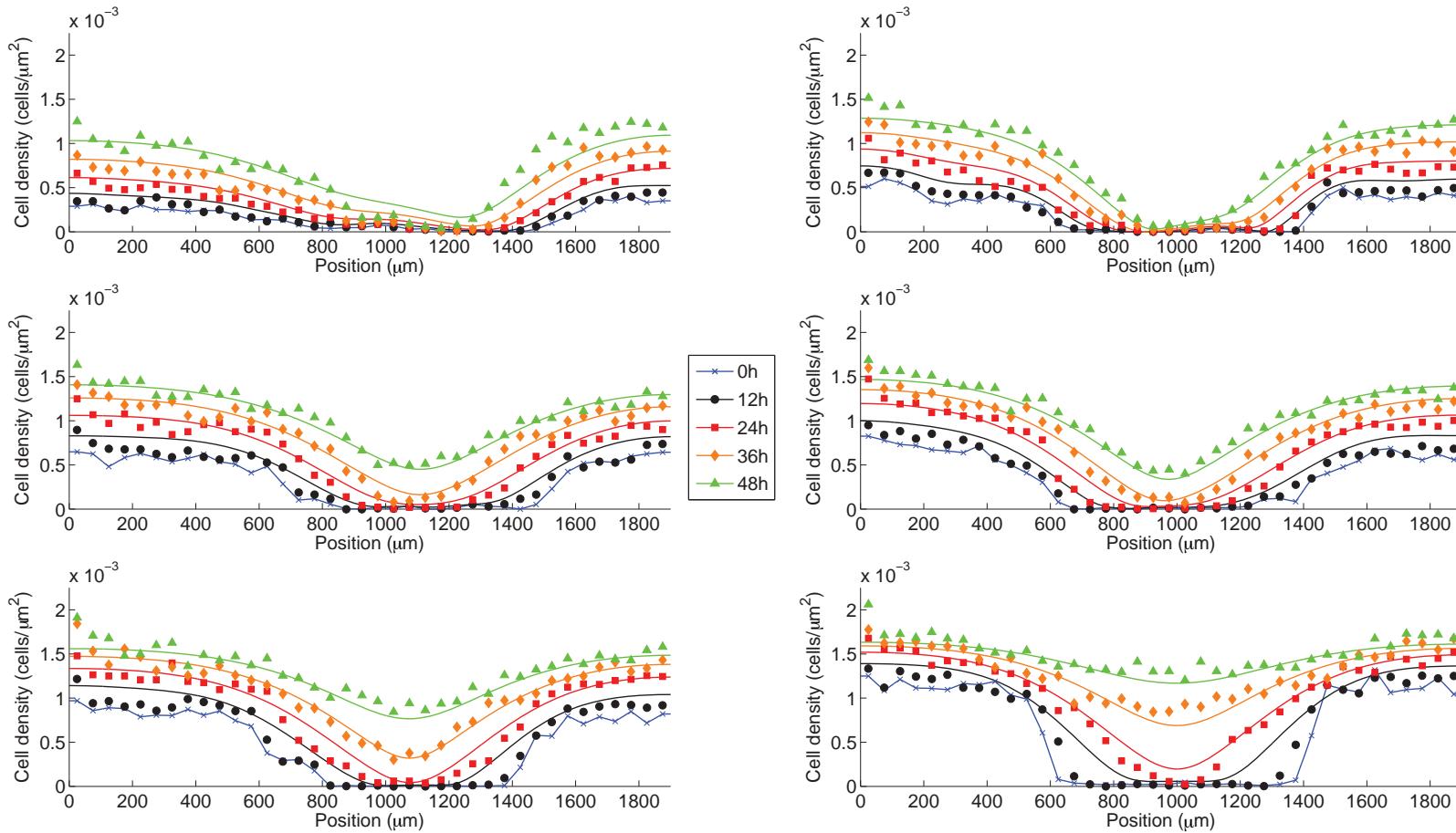
Here, we treat the cases of $m = 1/2, 2, 3$ and 4 separately. Results from the model calibration procedure are presented in Tables 2.3–2.6 and Figures 1 - 8, for $m = 1/2, 2, 3$ and 4, respectively. In general we find that increased values of m lead to a poorer quality of match to the experimental data. Comparing our estimates of $\bar{D}$ for $m = 0$ and 1 (Section 2.3.4 and 2.3.5) with the additional results here for the intermediate value of $m = 1/2$ shows that our estimates of $\bar{D}$ vary by approximately 310% for the Fisher–Kolmogorov model Section 2.3.4, approximately 130% for the Porous–Fisher model with $m = 1$ Section 2.3.5 and by approximately 190% for the generalized Porous–Fisher model with $m = 1/2$. Therefore, from this point of view, the standard Porous–Fisher model with $m = 1$ outperforms both the standard Fisher–Kolmogorov model with $m = 0$ and the generalised Porous–Fisher model with $m = 1/2$. We do not consider any further values of m.
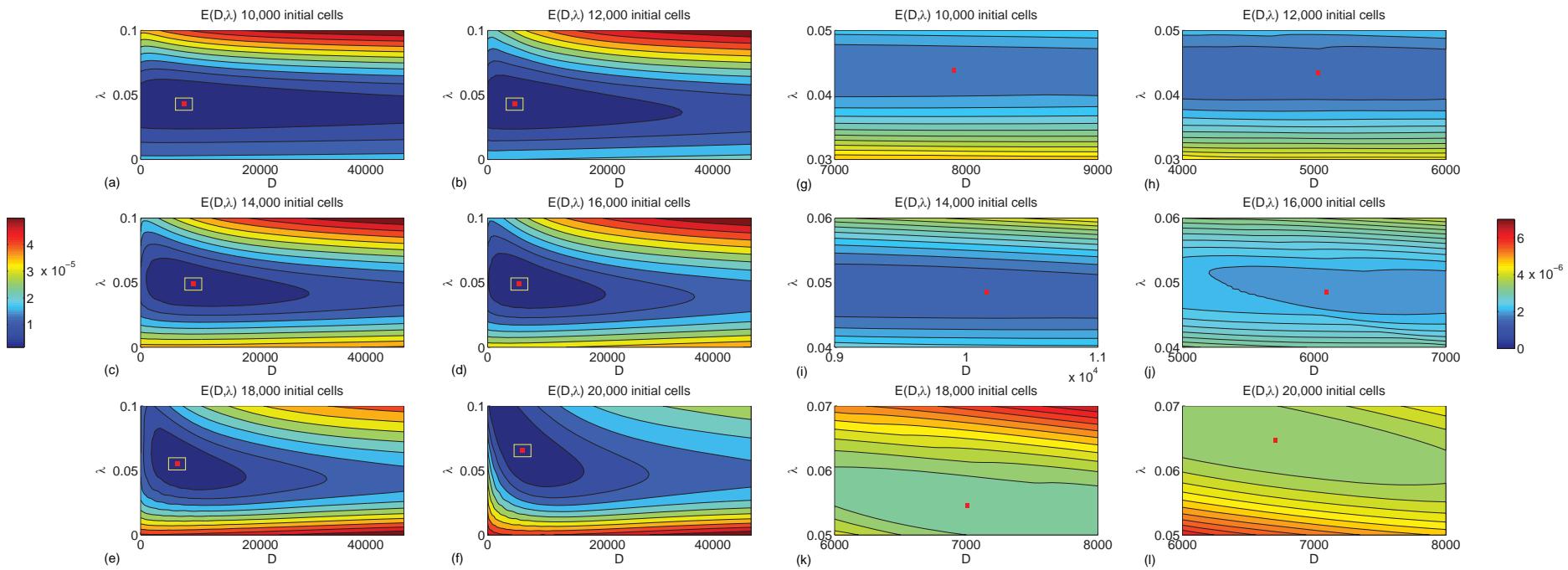
Figure 2.8: (a)–(f) Error surface profiles, $E(D, \lambda)$, for the generalised Porous–Fisher model with $m = 1/2$. The error surface profiles, $E(D, \lambda)$, are constructed using 100 equally–spaced values of D in $0 \leq D \leq 4000$ μm^2/h and 50 equally–spaced values of λ in $0 \leq \lambda \leq 0.1$ /h. The value of $E(D, \lambda)$ in (a)–(f) is shown on the left–most colour bar. (g)–(l) Refined estimates of $E(D, \lambda)$ centered about the minimum identified in (a)–(f) (white rectangle). The value of $E(D, \lambda)$ in (g)–(l) is shown on the right–most colour bar. The location of $\bar{D}$ and $\bar{\lambda}$ in each case is shown as a red square. Estimates of $(\bar{D}, \bar{\lambda})$ correspond to: (a) $(770, 0.043)$, (b) $(610, 0.043)$, (c) $(1500, 0.049)$, (d) $(1200, 0.049)$, (e) $(1500, 0.053)$, (f) $(1800, 0.063)$, (g) $(780, 0.044)$, (h) $(620, 0.043)$, (i) $(1500, 0.048)$, (j) $(1200, 0.049)$, (k) $(1500, 0.054)$, (l) $(1800, 0.064)$. All parameter estimates reported in this caption are given to two significant figures.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

38

Table 2.3: Least–squares estimates of $\bar{D}$ and $\bar{\lambda}$ for the generalised Porous–Fisher model with $m = 1/2$. All parameter estimates are given to two significant figures.

Initial number of cells	$\bar{D}$ (μm^2/h)	$\bar{\lambda}$ (/h)	t_d (h)
10,000	780 ± 340	0.044 ± 0.005	15.75
12,000	620 ± 410	0.043 ± 0.002	16.12
14,000	1500 ± 110	0.048 ± 0.002	14.44
16,000	1200 ± 530	0.049 ± 0.003	14.15
18,000	1500 ± 110	0.054 ± 0.004	12.84
20,000	1800 ± 370	0.064 ± 0.002	10.83
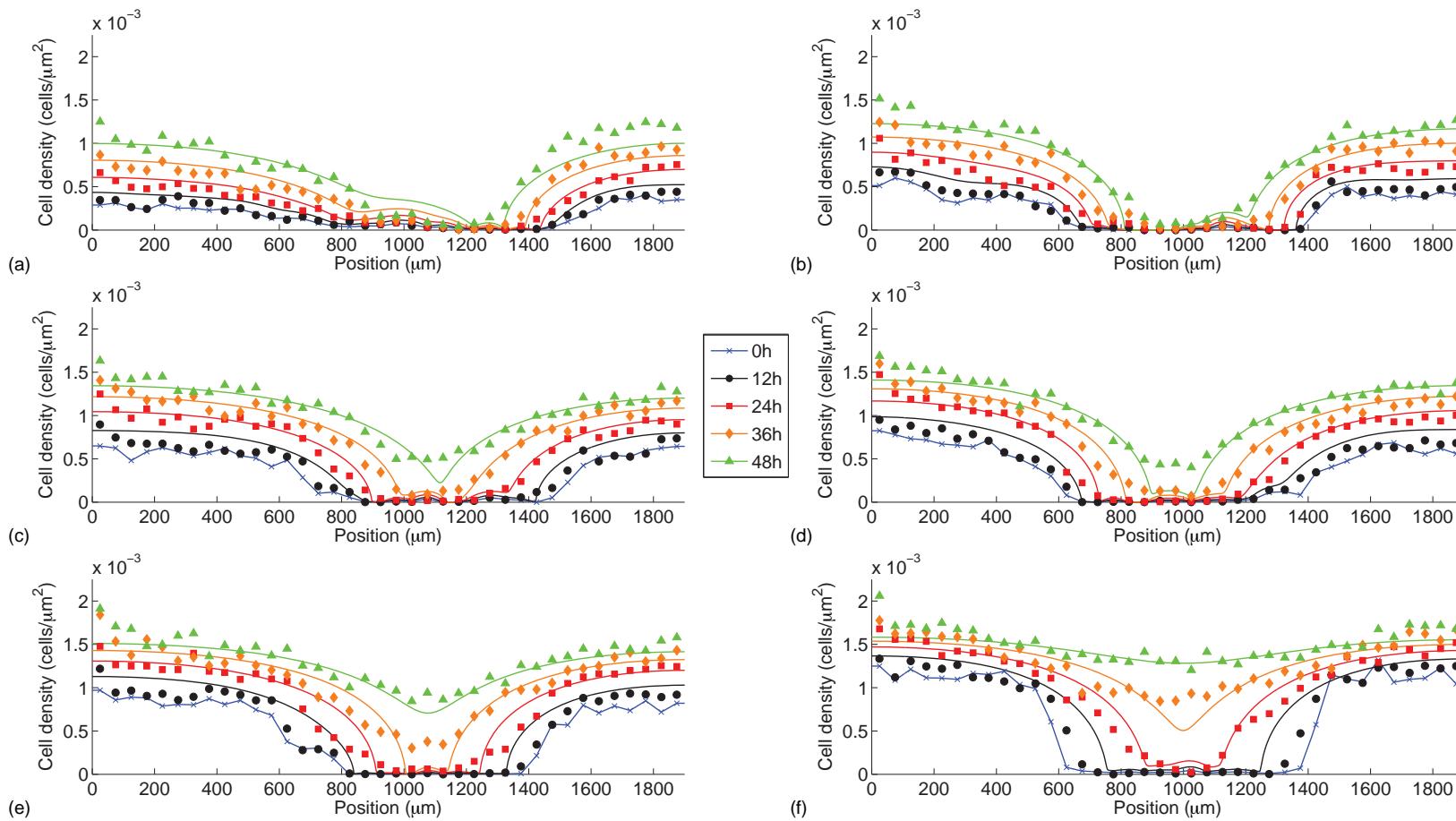
Figure 2.9: Calibrated solutions of Equation (2.4) ($m = 1/2$) to the averaged cell density profiles. Results in (a)–(f) show the average cell density for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equation (2.4) ($m = 1/2$) with the least squares estimates of D and λ from Table 2.3. All results correspond to $K = 1.7 \times 10^{-3}$ cells/μm^2 and the numerical solution is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.
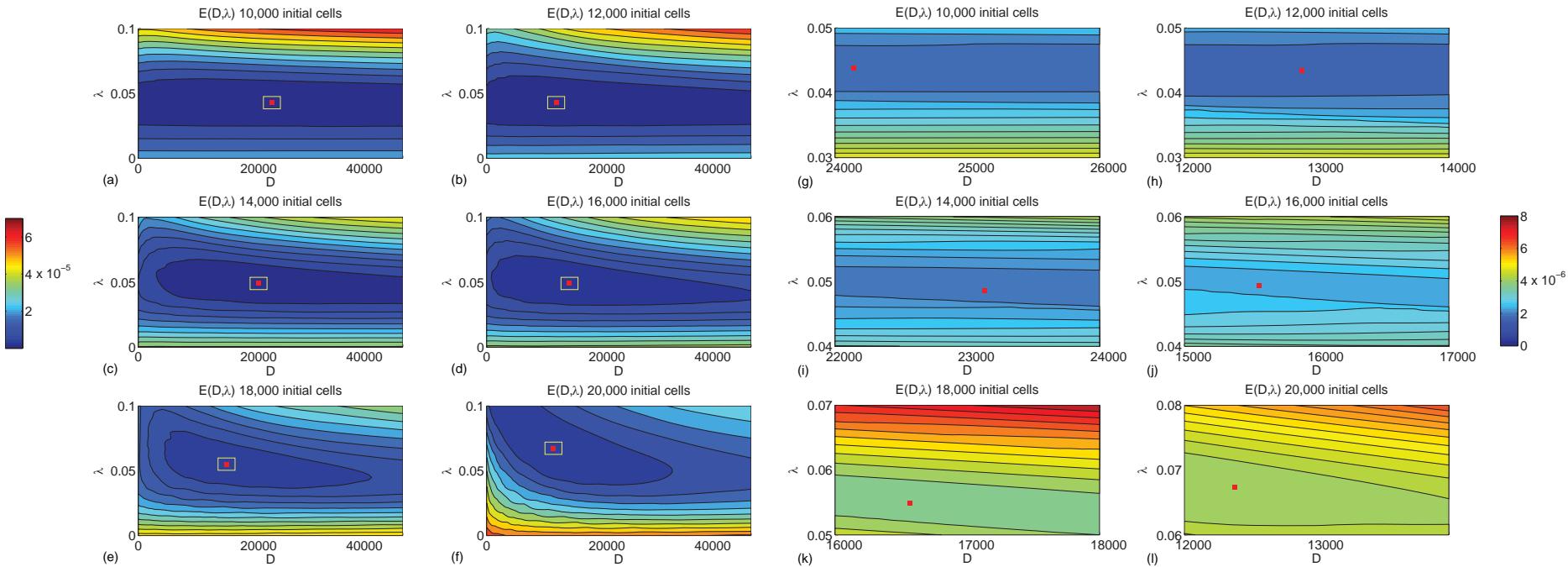
Figure 2.10: (a)–(f) Error surface profiles, $E(D, \lambda)$, for the generalised Porous–Fisher model with $m = 2$. The error surface profiles, $E(D, \lambda)$, are constructed using 100 equally–spaced values of D in $0 \leq D \leq 50000$ $\mu m^2/h$, and 50 equally–spaced values of λ in $0 \leq \lambda \leq 0.1$ /h. The value of $E(D, \lambda)$ in (a)–(f) is shown on the left–most colour bar. (g)–(l) Refined estimates of $E(D, \lambda)$ centered about the minima identified in (a)–(f) (white rectangle). The value of $E(D, \lambda)$ in (g)–(l) is shown on the right–most colour bar. The location of $\bar{D}$ and $\bar{\lambda}$ in each case is shown as a red square. Estimates of $(\bar{D}, \bar{\lambda})$ correspond to: (a) $(8200, 0.043)$, (b) $(5100, 0.043)$, (c) $(10000, 0.049)$, (d) $6000, 0.049)$, (e) $(6900, 0.055)$, (f) $(6600, 0.065)$, (g) $(7900, 0.044)$, (h) $(5000, 0.043)$, (i) $(10000, 0.049)$, (j) $(6100, 0.049)$, (k) $(7000, 0.054)$, (l) $(6700, 0.065)$. All parameter estimates reported in this caption are given to two significant figures.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

41

Table 2.4: Least–squares estimates of $\bar{D}$ and $\bar{\lambda}$ for the generalised Porous–Fisher model with $m = 2$. All parameter estimates are given to two significant figures.

Initial number of cells	$\bar{D}$ (μm^2/h)	$\bar{\lambda}$ (/h)	t_d (h)
10,000	7900 ± 4600	0.044 ± 0.005	15.75
12,000	5000 ± 3000	0.043 ± 0.002	16.12
14,000	10000 ± 1000	0.049 ± 0.001	14.15
16,000	6100 ± 3000	0.049 ± 0.003	14.15
18,000	7000 ± 500	0.054 ± 0.003	12.84
20,000	6700 ± 1700	0.065 ± 0.003	10.66
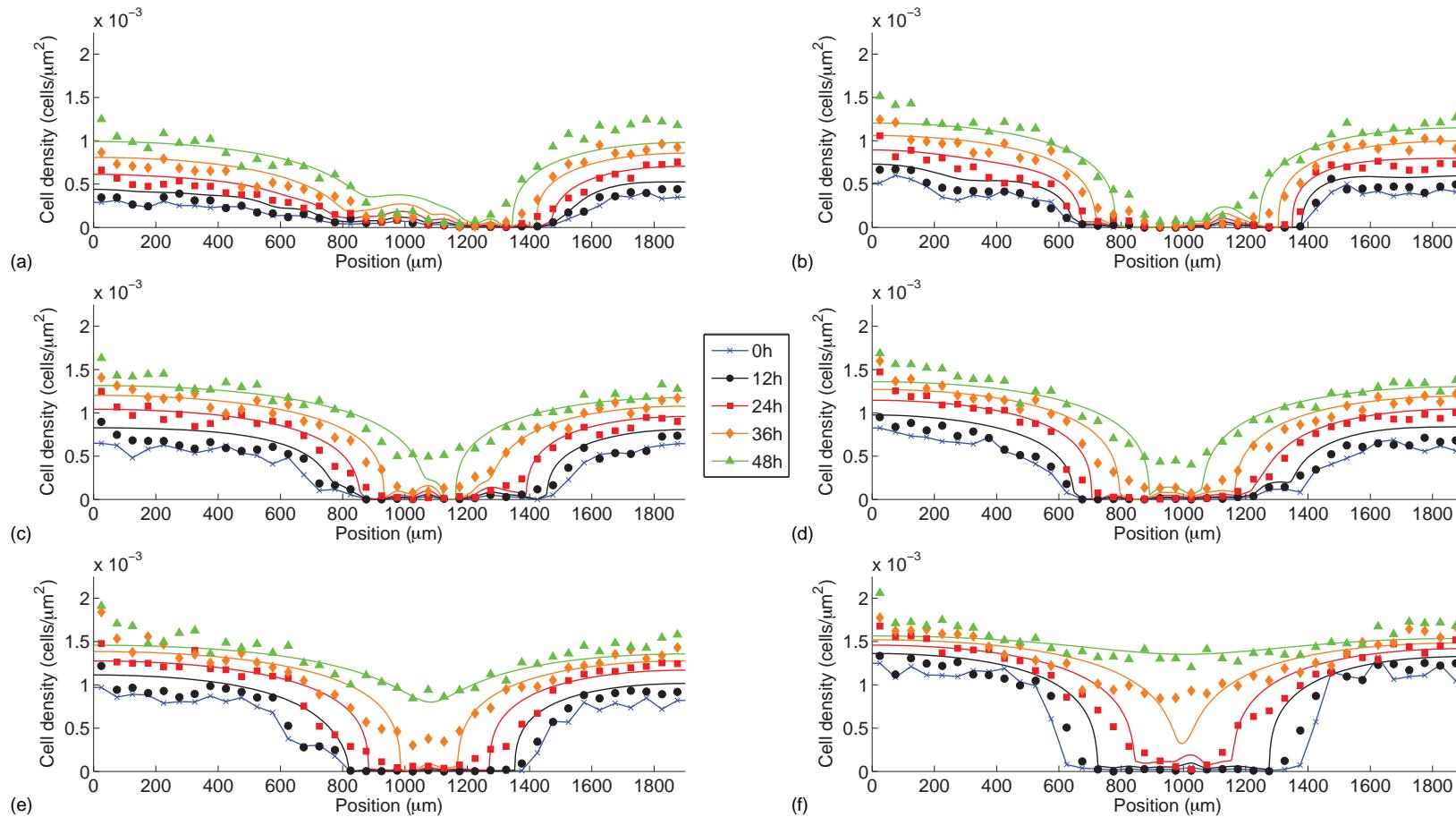
Figure 2.11: Calibrated solutions of Equation (2.4) ($m = 2$) to the averaged cell density profiles. Results in (a)–(f) show the average cell density for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equation (2.4) ($m = 2$) with the least squares estimates of D and λ from Table 2.4. All results correspond to $K = 1.7 \times 10^{-3}$ cells/μm^2 and the numerical solution is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.
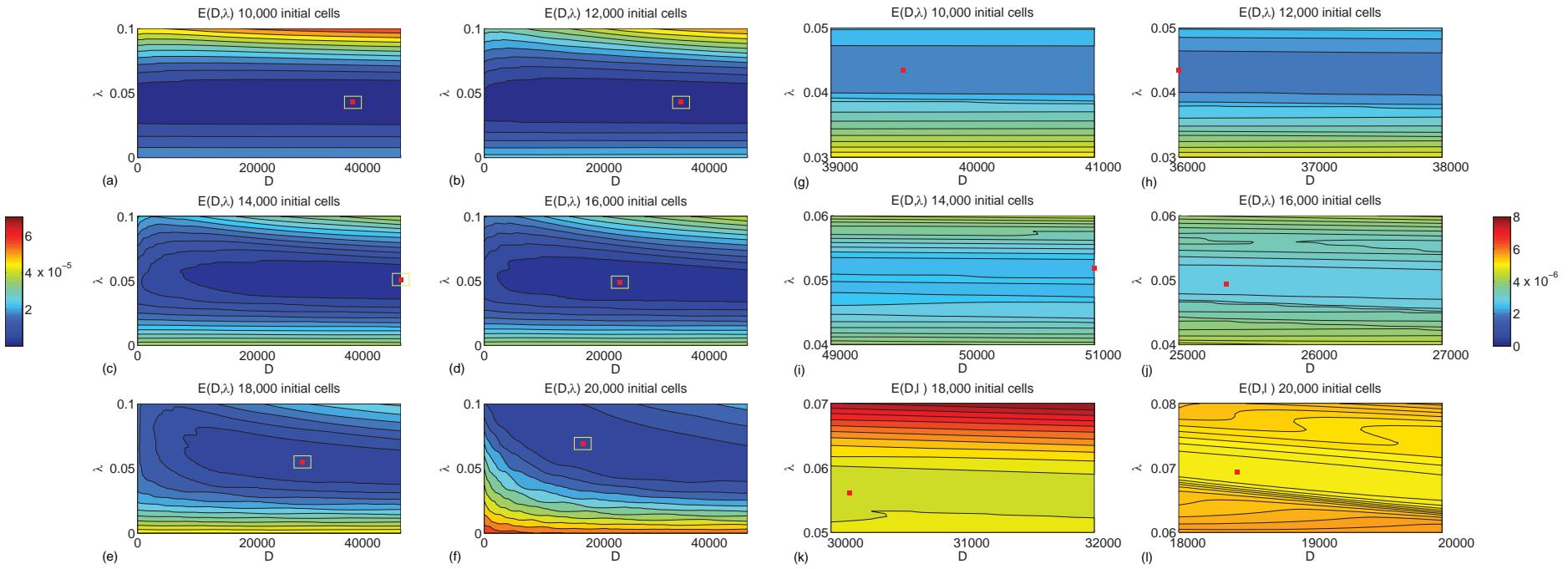
Figure 2.12: (a)–(f) Error surface profiles, $E(D, \lambda)$, for the generalised Porous–Fisher model with $m = 3$. The error surface profiles, $E(D, \lambda)$, are constructed using 100 equally–spaced values of D in $0 \leq D \leq 50000\ \mu\text{m}^2/\text{h}$ and 50 equally–spaced values of λ in $0 \leq \lambda \leq 0.1$ /h. The value of $E(D, \lambda)$ in (a)–(f) is shown on the left–most colour bar. (g)–(l) Refined estimates of $E(D, \lambda)$ centered about the minimum identified in (a)–(f) (white rectangle). The value of $E(D, \lambda)$ in (g)–(l) is shown on the right–most colour bar. The location of $\bar{D}$ and $\bar{\lambda}$ in each case is shown as a red square. Estimates of $(\bar{D}, \bar{\lambda})$ correspond to: (a) $(25000, 0.043)$, (b) $(13000, 0.043)$, (c) $(23000, 0.049)$, (d) $(16000, 0.049)$, (e) $(17000, 0.055)$, (f) $(13000, 0.067)$, (g) $(24000, 0.044)$, (h) $(13000, 0.043)$, (i) $(23000, 0.049)$, (j) $(16000, 0.049)$, (k) $(17000, 0.055)$, (l) $(12000, 0.067)$. All parameter estimates reported in this caption are given to two significant figures.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

44

Table 2.5: Least–squares estimates of $\bar{D}$ and $\bar{\lambda}$ for the generalised Porous–Fisher model with $m = 3$. All parameter estimates are given to two significant figures.

| Initial number of cells	$\bar{D}$ (μm^2/h)	$\bar{\lambda}$ (/h)	t_d (h)
10,000	24000 ± 18000	0.044 ± 0.005	15.75
12,000	13000 ± 11000	0.043 ± 0.002	16.12
14,000	23000 ± 6700	0.049 ± 0.001	14.15
16,000	16000 ± 8100	0.049 ± 0.004	14.15
18,000	17000 ± 1300	0.055 ± 0.004	12.60
20,000	12000 ± 3800	0.067 ± 0.003	10.35
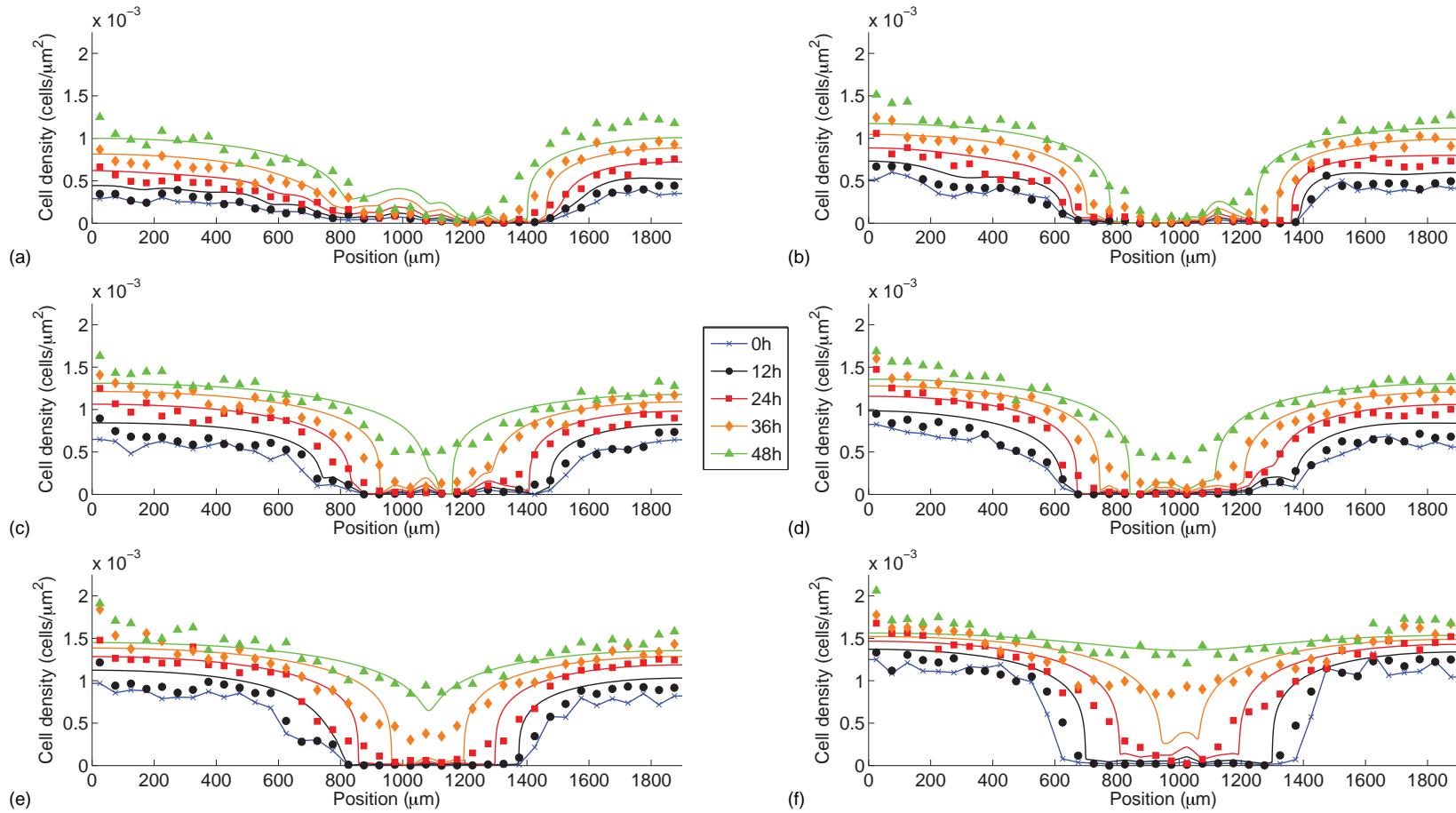
Figure 2.13: Calibrated solutions of Equation (2.4) ($m = 3$) to the averaged cell density profiles. Results in (a)–(f) show the average cell density for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equation (2.4) ($m = 3$) with the least squares estimates of D and λ from Table 2.5. All results correspond to $K = 1.7 \times 10^{-3}$ cells/μm^2 and the numerical solution is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.

Figure 2.14: (a)–(f) Error surface profiles, $E(D, \lambda)$, for the generalised Porous–Fisher model with $m = 4$. The error surface profiles, $E(D, \lambda)$, are constructed using 100 equally–spaced values of D in $0 \leq D \leq 50000$ $\mu\text{m}^2/\text{h}$ and 50 equally–spaced values of λ in $0 \leq \lambda \leq 0.1$ /h. The value of $E(D, \lambda)$ in (a)–(f) is shown on the left–most colour bar. (g)–(l) Refined estimates of $E(D, \lambda)$ centered about the minimum identified in (a)–(f) (white rectangle). The value of $E(D, \lambda)$ in (g)–(l) is shown on the right–most colour bar. The location of $\bar{D}$ and $\bar{\lambda}$ in each case is shown as a red square. Estimates of $(\bar{D}, \bar{\lambda})$ correspond to: (a) $(41000, 0.043)$, (b) $(37000, 0.043)$, (c) $(50000, 0.051)$, (d) $(26000, 0.049)$, (e) $(31000, 0.055)$, (f) $(19000, 0.069)$, (g) $(40000, 0.043)$, (h) $(36000, 0.043)$, (i) $(51000, 0.052)$, (j) $(25000, 0.049)$, (k) $(30000, 0.056)$, (l) $(18000, 0.069)$. All parameter estimates reported in this caption are given to two significant figures.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

47

Table 2.6: Least–squares estimates of $\bar{D}$ and $\bar{\lambda}$ for the generalised Porous–Fisher model with $m = 4$. All parameter estimates are given to two significant figures.

| Initial number of cells	$\bar{D}$ (μm^2/h)	$\bar{\lambda}$ (/h)	t_d (h)
10,000	40000 ± 26000	0.043 ± 0.005	16.12
12,000	36000 ± 22000	0.043 ± 0.001	16.12
14,000	51000 ± 7200	0.052 ± 0.002	13.33
16,000	25000 ± 21000	0.049 ± 0.004	14.15
18,000	30000 ± 4700	0.056 ± 0.004	12.38
20,000	18000 ± 6200	0.069 ± 0.003	10.05

Figure 2.15: Calibrated solutions of Equation (2.4) ($m = 4$) to the averaged cell density profiles. Results in (a)–(f) show the average cell density for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equation (2.4) ($m = 4$) with the least squares estimates of D and λ from Table 2.6. All results correspond to $K = 1.7 \times 10^{-3}$ cells/μm^2 and the numerical solution is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

49

2.5.2 Chemokinesis model

Our experimental results suggest that larger initial numbers of cells lead to enhanced cell spreading, and one way to model this kind of observation is to propose a more detailed reaction–diffusion model whereby the migration of cells is coupled to the availability of a growth factor which is produced by the cells. To demonstrate this kind of coupling we consider the following multispecies chemokinesis model,

$$\frac{\partial C}{\partial t} = D\frac{\partial}{\partial x}\left[\left(\frac{G}{G_{\max}}\right)\frac{\partial C}{\partial x}\right] + \lambda C\left[1 - \left(\frac{C}{K}\right)\right], \tag{2.5}$$
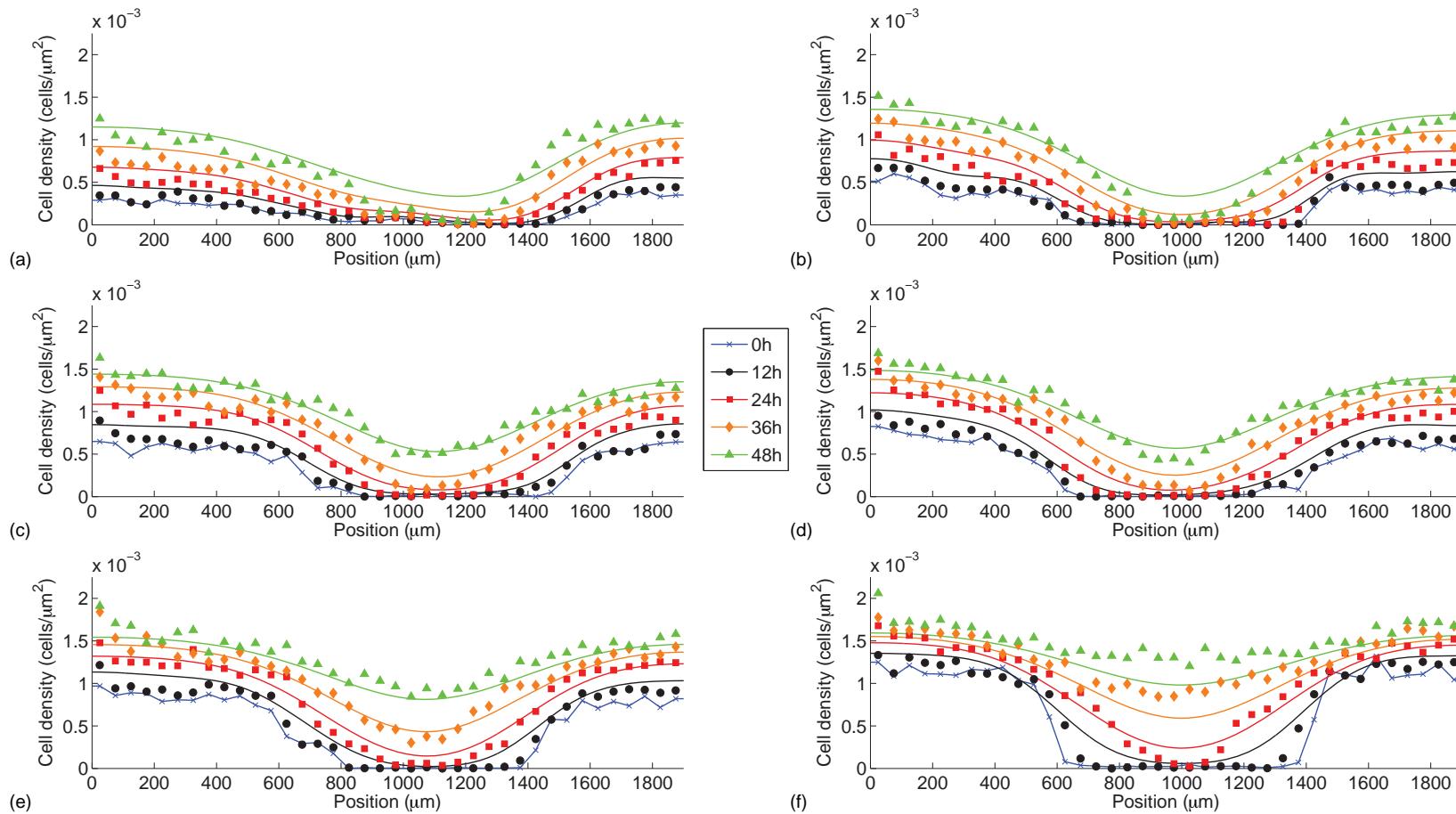
$$\frac{\partial G}{\partial t} = D_g\frac{\partial^2 G}{\partial x^2} + \alpha C - \beta G, \tag{2.6}$$

where $G(x, t)$ is the density of the growth factor, $G_{\max} = \alpha K/\beta$ is the maximum value of the growth factor concentration, D_g is the diffusivity of the growth factor, α is the rate at which cells produce growth factor, and β is the decay rate of growth factor. In this model the diffusivity is an increasing function of the growth factor concentration, $\mathcal{D} = D(G/G_{\max})$. Since the monolayer of cells is very shallow, the concentration $G(x, t)$ is taken to be two–dimensional with dimensions of ng/μm^2, which implies that the dimensions of α are ng/(cell h). Considering the cell monolayer is shallow comparing with the dimension of the well plate, in this model we neglect the diffusion of the growth factor along z axis.

The key difference between Equations (2.5)–(2.6) and Equation (2.4) is that in the chemokinesis model the flux of cells is proportional to the concentration of the growth factor whereas in the Porous–Fisher model the flux of cells is proportional to the density of cells. Unfortunately, calibrating the solution of Equations (2.5)–(2.6) to the experimental observations in Figure 2.3 is far more challenging than dealing with the Fisher–Kolmogorov or Porous–Fisher models, since the chemokinesis model contains five unknown parameters whereas the Fisher–Kolmogorov and Porous–Fisher models involve just two. The difficulties associated with identifying the appropriate parameters in Equations (2.5)–(2.6) is further complicated by the fact that standard IncuCyte ZOOM$^{\text{TM}}$ assays do not make any measurements regarding the spatial or temporal distribution of putative growth factors [29]. Therefore, our ability to meaningfully calibrate this more complicated model to our data is limited by the experimental protocol. Nonetheless, we will attempt to demonstrate how the solution of Equations (2.5)–(2.6) could be relevant to our experiment by comparing solutions of this model to our experimental data. To proceed we do not attempt to calibrate the solution of Equations (2.5)–(2.6) to our data, nor do we attempt to construct high–dimensional error surface plots of $E(D, \lambda, D_g, \alpha, \beta)$. Instead, we proceed by making some fairly crude assumptions about the likely values of the parameters. For example we set $K = 1.7 \times 10^{-3}$ cells/μm^2 using our previous measurements, and set $\lambda = 0.05$ /h by taking an average of the estimates of λ from Table 2.1. Previous measurements of diffusivity values for growth factors suggest that $D_g = 5 \times 10^5$ μm^2/h [84] and previous measurements of growth factor decay rates suggest that $\beta = 0.069$ /h [113]. Therefore, with these values we attempt, by trial and error, to

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

50

find estimates of D and α for which the solution of Equations (2.5)–(2.6) matches all the data in Figure 2.3 for all six initial conditions.

We solve Equations (2.5)–(2.6) on $0 < x < 1900$ μm, with zero flux boundary conditions for both $C(x,t)$ and $G(x,t)$ at both $x = 0$ and $x = 1900$ μm. The initial condition for Equation (2.5) was taken to be a linear interpolation of the data in Figure 2.3 at $t = 0$ h. Since we have no measurements of $G(x,t)$ at any point in space or time, we must also estimate $G(x,0)$. Since the diffusivity of growth factor is much faster than the diffusivity of cells, we assume that the growth factor becomes steady much faster than the cell density profile and we set $G(x,0) = \alpha C(x,0)/\beta$ for the initial condition. Results in Figure 2.16 compare the solution of Equations (2.5)–(2.6) with $D = 2530$ μm^2 /h and $\alpha = 1.78$ ng/(cell h).

Figure 2.16: Solutions of the chemokinesis model, Equations (2.5)–(2.6), superimposed on the averaged cell density profiles. Results in (a)–(f) show the average cell density for experiments initiated with 10000, 12000, 14000, 16000, 18000 and 20000 cells per well. In each experiment profiles at $t = 0, 12, 24, 36$ and 48 h are shown. The solid lines correspond to the solution of Equations (2.5)–(2.6) with $K = 1.7 \times 10^{-3}$ cells/μm^2, $D = 2530$ μm^2 /h, $\lambda = 0.05$ /h, $D_g = 5 \times 10^5$ μm^2/h, $\beta = 0.0693$ /h and $\alpha = 1.78$ ng/(cell h). The numerical solution of Equations (2.5)–(2.6) is obtained with $\delta x = 0.25$ μm, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.

Chapter 2. Reproducibility of scratch assays is affected by the initial degree
of confluence: Experiments, modelling and model selection

52

The match between the solution of Equations (2.5)–(2.6) and the experimental data is reasonable given that we have attempted to fit all the experimental data for all six different initial conditions using a single reaction–diffusion model with constant coefficients. Since standard IncuCyte ZOOM$^{\text{TM}}$ protocols make no attempt to measure the presence or spatial distribution of any growth factors present in the experiment [29], it is not reasonable for us to attempt to perform a detailed parameter estimation for Equations (2.5)–(2.6). Instead, as a proof–of–concept, we can demonstrate that it is possible, in principle, to select parameter values in Equations (2.5)–(2.6) for which this more complicated reaction–diffusion model appears to provide a reasonable match to the experimental data for all initial conditions. However, instead of claiming that Equations (2.5)–(2.6) is a superior model to the simpler Fisher–Kolmogorov and Porous–Fisher models, it is worthwhile to recall that the chemokinesis model involves five unknown parameters meaning that to use this kind of model with certainty we would, ideally, prefer to have additional data describing some aspects of the growth factor dynamics. Unfortunately this kind of additional data is never collected when using IncuCyte ZOOM$^{\text{TM}}$ zoom assays [29].

2.5.3 Relationship between the Fisher–Kolmogorov, Porous–Fisher and Chemokinesis models

Although we have presented the Fisher–Kolmogorov, Porous–Fisher and Chemokinesis models as being completely unrelated, these three models are, in fact, related to each other and it is worthwhile to point out these relationships. The chemokinesis model, given by Equations (2.5)–(2.6), is related to the standard Fisher–Kolmogorov model in the limit of $D_g/D \gg (\beta L^2)/D \gg 1$, where L is the width of the experimental window. Additional numerical simulations (not shown) support this finding which suggests that we can interpret the Fisher–Kolmogorov model as a limiting case the chemokinesis model. Furthermore, the standard Porous–Fisher model, Equation (2.4) with $m = 1$, can also be thought of as a limiting case of the chemokinesis model when $(\beta L^2)/D \gg D_g/D \gg 1$. Again, additional numerical simulations (not shown) support this argument. In addition, we could use the three models to generate surrogate data with some random noise, to investigate whether the estimated parameters can be fitted back for various initial cell densities.

3 Logistic proliferation of cells in scratch assays is delayed

Abstract

Scratch assays are used to study how a population of cells re–colonises a vacant region on a two–dimensional substrate after a cell monolayer is scratched. These experiments are used in many applications including drug design for the treatment of cancer and chronic wounds. To provide insights into the mechanisms that drive scratch assays, solutions of continuum reaction–diffusion models have been calibrated to data from scratch assays. These models typically include a logistic source term to describe carrying capacity-limited proliferation, however the choice of using a logistic source term is often made without examining whether it is valid. Here we study the proliferation of PC-3 prostate cancer cells in a scratch assay. All experimental results for the scratch assay are compared with equivalent results from a proliferation assay where the cell monolayer is not scratched. Visual inspection of the time evolution of the cell density away from the location of the scratch reveals a series of sigmoid curves that could be naively calibrated to the solution of the logistic growth model. However, careful analysis of the per capita growth rate as a function of density reveals several key differences between the proliferation of cells in scratch and proliferation assays. Our findings suggest that the logistic growth model is valid for the entire duration of the proliferation assay. On the other hand, guided by data, we suggest that there are two phases of proliferation in a scratch assay; at short time we have a _disturbance phase_ where proliferation is not logistic, and this is followed by a _growth phase_ where proliferation appears to be logistic. These two phases are observed across a large number of experiments performed at different initial cell densities. Overall our study shows that simply calibrating the solution of a continuum model to a scratch assay might produce misleading parameter estimates, and this issue can be resolved by making a distinction between the disturbance and growth phases. Repeating our procedure for

other scratch assays will provide insight into the roles of the disturbance and growth phases for different cell lines and scratch assays performed on different substrates.

3.1 Background

Understanding population dynamics is a fundamental question that has wide relevance to many biological and ecological processes. For example, the rate of spatial spreading of invasive species through different ecosystems is driven, in part, by the population dynamics and rates of growth of the invasive species [69, 82, 123]. Population dynamics and population growth are also central to understanding the spread of infectious diseases. For example, the spread of *Wolbachia* into wild mosquito populations is thought to reduce a wide range of diseases, and the spatial spreading of the mosquito population is partly driven by the population dynamics of the mosquito population [16]. Similar ideas also apply to the spreading of tumour cells and the progression of cancer, which is related to the rates of proliferation of invasive cancer cells [4, 78, 92]. Therefore, improving our understanding of population dynamics by calibrating mathematical models to experimental observations of population dynamics is of great interest.

In vitro scratch assays are routinely used to study the ability of cell populations to re–colonise an initially–vacant region [65, 71, 115, 118]. This re–colonisation occurs as a result of the combination of cell migration and cell proliferation, and gives rise to moving fronts of cells that re–colonise the vacant region. Scratch assays provide insights into both cancer spreading and tissue repair processes [65, 76, 77]. In general, performing a scratch assay involves three steps: (i) growing a monolayer of cells on a two–dimensional substrate; (ii) creating a vacant region in the monolayer by scratching it with a sharp–tipped instrument; and, (iii) imaging the re–colonisation of the scratched region [65, 71]. Another type of *in vitro* assay, called a proliferation assay, is performed using the exact same procedure as a scratch assay, except that the monolayer of cells is not scratched [56, 111, 118]. Cell proliferation assays allow experimentalists to measure the increase in cell numbers over time due to proliferation [118].

In the applied mathematics literature, scratch assays have been modelled using continuum reaction–diffusion equations [14, 49, 54, 76, 77, 98, 100, 102, 103, 106]. In these models, cell migration is represented by a diffusion term, and carrying–capacity limited proliferation is represented by a logistic source term. For proliferation assays in which the cell population is uniformly distributed and no scratch is made, the continuum reaction–diffusion equation simplifies to the logistic growth equation [14, 54, 111], given by

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \lambda C(t)\left(1 - \frac{C(t)}{K}\right), \tag{3.1}$$

where $C(t) > 0$ is the density of cells, t is time, $\lambda > 0$ is the proliferation rate, and $K > 0$ is the carrying capacity density.

It is interesting to note that a logistic growth term is often used when modelling scratch assays or proliferation assays [14,49,54,76,77,98,100,118], yet the suitability of this choice is rarely, if ever, tested using experimental data. In fact, several studies argue that the logistic growth equation does not always match experimental data [66,97,124,130]. For example, Laird examines *in vivo* tumour growth data and shows that the standard logistic model does not match experimental data [66]. Similarly, Sarapata and de Pillis find that the logistic growth model does not always match experimental tumour growth data [97]. West and coworkers investigate the growth patterns of a wide range of animal models [124]. By comparing experimental data with model predictions, they suggest that the growth is not logistic, and is better described by a more general model. In addition, the results from our previous study, focusing on scratch assays, suggest that when calibrating solutions of a logistic–type reaction–diffusion equation to experimental data with varying initial cell density, there appears to be no unique value of λ for which the logistic growth equation matches the entire data set for all initial cell densities [49]. One way of interpreting this result is that the cells in the scratch assay do not proliferate logistically.

In the present work, we use a combined experimental and mathematical approach to investigate whether the proliferation of cells in a scratch assay can be modelled with the classical logistic equation. Our approach involves performing a series of proliferation assays to act as a control so that we can examine whether the process of scratching the monolayer affects the way that cells proliferate. While many experimental studies implicitly assume that scratching the monolayer does not affect cell proliferation, others suggest the process of scratching can trigger certain signalling pathways that may have some effects on the way that cells proliferate [85,86]. To investigate these questions, we perform a suite of scratch assays and proliferation assays using the IncuCyte ZOOM$^{\text{TM}}$ system [54]. For both types of assays, we use the PC–3 prostate cancer cell line [57], and we consider varying the initial seeding condition so that we can examine the influence of varying the initial cell density.

To quantitatively test the suitability of the logistic growth model, we extract cell density information from the experimental images and then estimate the per capita growth rates from the data for both the scratch assays and the proliferation assays. Our results show that the evolution in cell density in the proliferation assays appears to be logistic for the entire duration of the experiment. In contrast, the variation in cell density in the scratch assays is very different. We observe two phases in the scratch assays: (i) a disturbance phase at early time, in which the proliferation of cells is not logistic; and, (ii) a classic logistic growth phase for the remainder of the experiment. These two phases are observed in all of our experiments, across a wide range of initial cell densities. The differences how cells proliferate in the scratch assay and the proliferation assay is surprising because we are making observations well away from the location of the scratch. This finding that we have two phases of proliferation in scratch assays is significant because many mathematical studies implicitly assume that cells in scratch assays proliferate logistically for the entire duration of the experiment [14, 49, 54, 76, 77, 98, 100, 118]. However, our

finding is that cells located far away from the scratch proliferate very differently to cells in the proliferation assay.

This manuscript is organised in the following way. First, we describe the experimental methods, including how we process the experimental images to obtain cell density information. We then outline the logistic growth model and the least–squares method for calibrating the model to our data. By presenting information about the evolution of the cell density and the per capita growth rate, we identify two phases of proliferation in the scratch assays. These phases are identified by focusing on regions of the scratch assay that are located well behind the location of the scratch. After calibrating the solution of the logistic model to the cell density information, our results suggest that the logistic equation is relevant for the proliferation assays but only for the later phase in the scratch assays. We conclude this study by discussing some of the limitations, and we outline some extensions for future work.

3.2   Methods

3.2.1   Experimental Methods
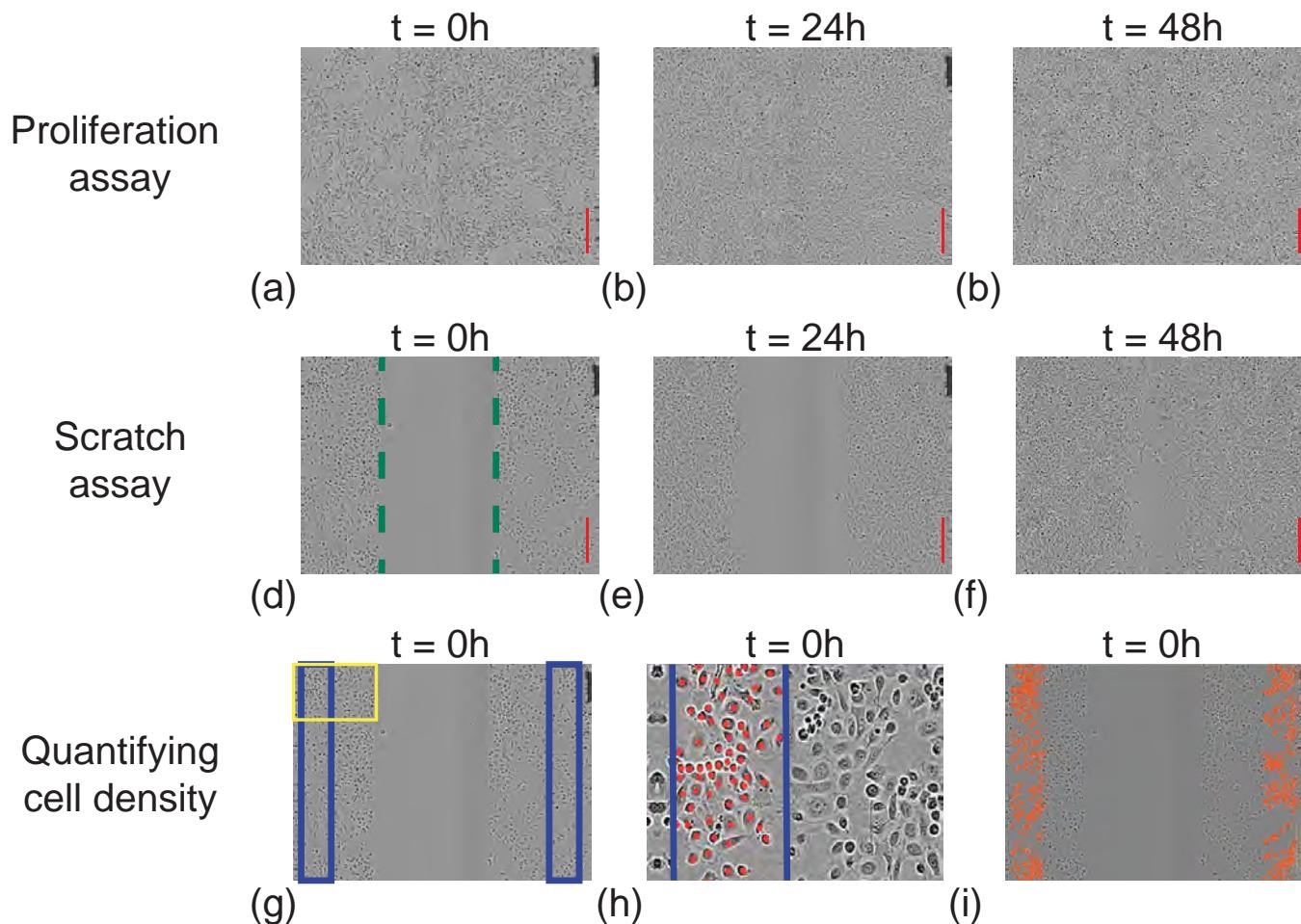
We perform scratch assays and proliferation assays using the IncuCyte ZOOM$^{\text{TM}}$ live cell imaging system (Essen BioScience, MI USA). All experiments are performed using the PC–3 prostate cancer cell line [57]. These cells, originally purchased from American Type Culture Collection (Manassas, VA, USA), are a gift from Lisa Chopin (April, 2016). The cell line is used according to the National Health and Medical Research Council (NHMRC) National statement on ethical conduct in human research with ethics approval for Queensland University of Technology Human Research Ethics Committee (QUT HREC 59644, Chopin). Cells are propagated in RPMI 1640 medium (Life Technologies, Australia) with 10% foetal calf serum (Sigma–Aldrich, Australia), 100 U/mL penicillin, and 100 μg/mL streptomycin (Life Technologies), in plastic tissue culture flasks (Corning Life Sciences, Asia Pacific). Cells are cultured in 5% CO_2 and 95% air in a Panasonic incubator (VWR International) at 37 ºC. Cells are regularly screened for *Mycoplasma* (Nested PCR using primers from Sigma–Aldrich).

Cell counting is performed using a Neubauer–improved haemocytometer (ProSciTech, Australia). Cells, grown to approximately 80% confluence, are removed from the flask using TrypLE$^{\text{TM}}$ (Life Technologies) in phosphate buffered saline (pH 7.4) and resuspended in culture medium ensuring that they are thoroughly mixed. After resuspension, an aliquot of 10 μL is quickly removed before the cells start to settle. A 1:1 mixture of cell suspension and 0.4% trypan blue solution (Sigma–Aldrich; a blue stain that is only absorbed by dead cells) is prepared and 10 μL of the solution is loaded onto the counting chamber of a clean Neubauer–improved haemocytometer. The counting chamber of a haemocytometer is delineated by grid lines that identify four chamber areas to be used in cell counting. The volume of the chamber area is 1×10^4 mL. Using a microscope, each

chamber area is viewed, and the live cells that are not coloured in blue are counted. The cell density is calculated by taking the average of the four readings and multiplying it by 10^4 and the dilution factor, to obtain the approximate number of cells per mL of the cell suspension [74].

For the proliferation assays, the cell count is determined and the cells are seeded at various densities in 96–well ImageLock plates [29]. Cells are distributed in the wells of the tissue culture plate as uniformly as possible. We report results for initial seeding densities of approximately 12,000, 16,000 and 20,000 cells per well. After seeding, cells are grown overnight to allow for attachment and some subsequent growth. The plate is placed into the IncuCyte ZOOM$^{\text{TM}}$ apparatus, and images are recorded every two hours for a total duration of 48 hours. An example of a set of experimental images from a proliferation assay is shown in Figure 1(a)–(c). For each initial seeding condition we perform 16 identically prepared experimental replicates ($n = 16$).

For the scratch assays, the cell count is determined and the cells are seeded at various densities in 96–well ImageLock plates different with the proliferation assays [29]. Cells are distributed in the wells of the tissue culture plate as uniformly as possible. We report results for initial seeding densities of approximately 12,000, 16,000 and 20,000 cells per well. After seeding, cells are grown overnight to allow for attachment and some subsequent growth. We use a WoundMaker$^{\text{TM}}$ to create uniform scratches in each well of a 96–well ImageLock plate [29]. To ensure that all cells are removed from the scratched region, a modification is made to the manufacturer's protocol, where the scratching motion is repeated 20 times over a short duration before lifting the WoundMaker$^{\text{TM}}$. After creating the scratch, the medium is aspirated and the wells are washed twice with fresh medium to remove any cells from the scratched area. Following the washes, 100 μL fresh medium is added to each well and the plate is placed into the IncuCyte ZOOM$^{\text{TM}}$ apparatus. Images of the collective cell spreading are recorded every two hours for a total duration of 48 hours. An example of a set of experimental images taken from a scratch assay is shown in Figure 3.1(d)–(f). For each initial seeding condition we perform 16 identically prepared experiments in different wells of the tissue culture plate ($n = 16$). Throughout this work we will refer to these identically prepared experiments in different wells as different *replicates*.

Figure 3.1: Experimental images. (a)–(c) A summary of IncuCyte ZOOM^{TM} experiments for proliferation assays. (d)–(f) A summary of IncuCyte ZOOM^{TM} experiments for scratch assays. Images show both types of experiments initiated with 16,000 cells per well. The time at which the image is recorded is indicated on each subfigure, and the scale bar (red line) corresponds to 300μm. The image in (d), at $t = 0$ hours, shows the approximate location of the position of the leading edges (dashed green). (g)–(i) To quantify the cell density profile, two rectangles of width 200μm, are superimposed on the experimental image as shown in (g). Manual cell counting is used to estimate the number of cells in each subregion, and these estimates are converted into an estimate of cell density in these regions at two–hour intervals during the first 18 hours of the experiment, and then at six–hour intervals during the remaining 30 hours of the experiment. To count individual cells we zoom in to focus on certain subregions, such as shown in (h), which corresponds to the yellow rectangle highlighted in (g). Using the counting features in Adobe Photoshop [3], we identify individual cells and place a unique marker on each cell (red disk), as shown in (h). After each image is processed in this way we have identified the total number of cells in the two subregions in the image, as shown in (i), and then we convert these estimates of cell numbers into an estimate of cell density.

3.2.2 Experimental Image Processing

New experimental data are generated for this study. To obtain cell density information from the experimental images, we count the number of cells in two identically sized subregions that are well behind the location of the scratch, as shown in Figure 3.1(g). The positions of the two subregions are located about 400 μm behind the scratch, and each subregion has dimensions 1430 μm $\times$ 200 μm. Throughout this work, we refer to the subregion to the left of the image as subregion 1, and the subregion to the right of the image as subregion 2. Because the subregions are located well away from the scratched region, we are able to invoke a simplifying assumption that the dynamic changes in cell density in these subregions is due to cell proliferation alone (Supplementary material) [54]. We do not use data that are directly adjacent to the left or right sides of the images since this corresponds to the boundary of the field of view. Cells in each subregion are counted in Photoshop using the 'Count Tool' [3]. After counting the number of cells in each subregion, we divide the total number of cells by the total area to give an estimate of the cell density. We repeat this process for each replicate and calculate the sample mean of the cell density at two–hour intervals during the first 18 hours of the experiment where the most rapid temporal changes take place. Then, during the last 30 hours of the experiment, we count cells at six–hour intervals.

One of the assumptions we make when analysing data from the scratch assay is that the two subregions are sufficiently far away from the edges of the scratch so that there are no spatial variations in cell density at these locations for the entire duration of the experiment. This assumption allows us to attribute any changes in cell density in the subregions to be a result of cell proliferation [54]. Quantitative evidence to support this assumption is provided in the Supplementary material.

3.2.3 Mathematical Methods

The logistic growth equation, given by Equation (3.1), has an exact solution

$$C(t) = \frac{KC(0)}{(K - C(0))\, e^{-\lambda t} + C(0)}, \tag{3.2}$$

which is a sigmoid curve that monotonically increases from the initial density $C(0)$ to K as $t \to \infty$. An important feature of the logistic growth model that we will make use of in this study is that the per capita growth rate, $(1/C)(dC/dt) = \lambda(1 - C/K)$, decreases linearly with C.

We estimate the two parameters in the logistic growth model, λ and K, by minimising a least–squares measure of the discrepancy between the solution of the logistic growth

equation and the average cell density information in our subregions that are located far away from the scratched region. The least–squares error is given by

$$E(\lambda, K) = \sum_{i=1}^{I} \left[C^{\text{model}}(t_i) - C^{\text{data}}(t_i) \right]^2, \tag{3.3}$$

where i is an index that indicates the number of time points used from the experimental data sets and I is the total number of time points used in the calibration procedure. We calibrate the solution of the logistic growth equation to the average cell density information using the MATLAB function `lsqcurvefit` [79] that is based on the Levenberg–Marquardt algorithm. For notational simplicity we denote the minimum least–squares error as $E_{\min} = E\left(\bar{\lambda}, \bar{K}\right)$. Each time we use the MATLAB function `lsqcurvefit`, we always check that the least–squares estimates of $\bar{\lambda}$ and $\bar{K}$ are independent of the initial estimate that is required for the iterative algorithm to converge.

3.3 Results

3.3.1 Quantitative assessment of experiments

Initial cell density

Many previous studies that calibrate solutions of mathematical models to experimental data from proliferation or scratch assays make use of just one initial density of cells [14,76,77,118]. To provide a more thorough investigation of the suitability of various mathematical models, we calibrate mathematical models to a suite of experimental data where the initial density of cells is intentionally varied [49]. To achieve this, our experimental procedure involves placing a different number of cells into each well of the tissue culture plate. We describe this as varying the *initial seeding condition*. In this work we consider three different initial seeding conditions that correspond to placing either: (i) 12,000; (ii) 16,000; or, (iii) 20,000 cells per well. For brevity, we refer to these three conditions as initial seeding conditions 1, 2 and 3, respectively.

After a particular number of cells are placed into the tissue culture plate, the cells are incubated overnight to allow them to attach to the plate and begin to proliferate. The experiments are then performed on the following day. Since the cell density changes overnight, we will refer to the initial density of cells at the beginning of the experiment on the following day, as the *initial cell density*. Therefore the cell population in the experimental images at $t = 0$ is associated with the initial cell density. Intuitively, we expect that the initial cell density in proliferation assays will be greater than the cell density associated with the initial seeding condition, because the cells have had a period of time to attach and begin to proliferate.
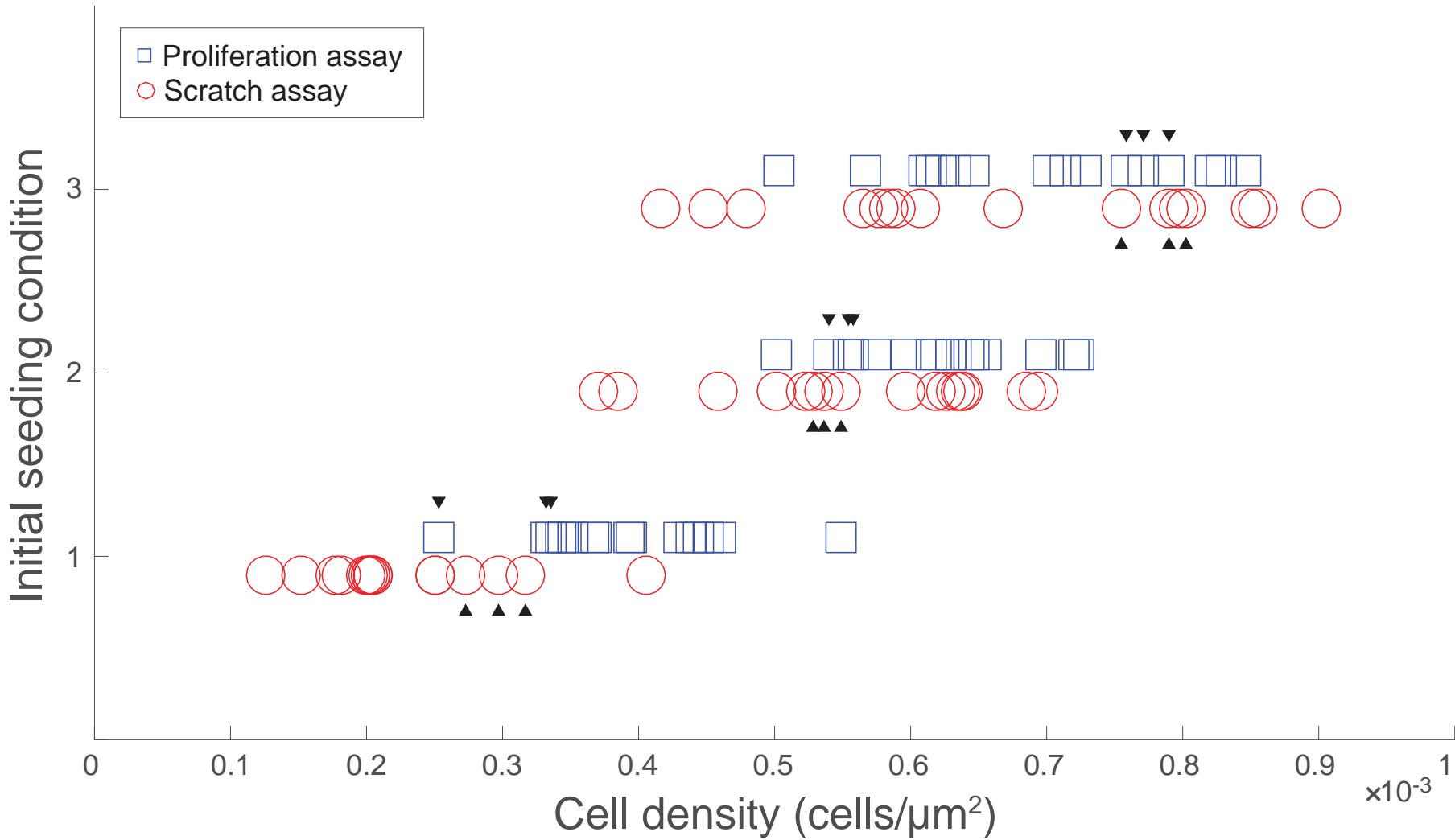
Before we examine the temporal evolution of cell density in our experiments, we first examine the variability in the initial cell densities amongst our various experimental replicates.

This is essential, since the process of placing either 12,000, 16,000 or 20,000 cells in each well of the tissue culture place is, at best, an approximation. To quantify the variability in the initial cell density, we count the number of cells in the two subregions, as shown in Figure 3.1(g), and convert these counts into an estimate of the initial cell density, $C(0)$. We repeat this procedure for both the proliferation and scratch assays, giving a total of 96 individual estimates of the initial cell density. These 96 estimates of the initial cell density are reported in Figure 3.2, revealing three features:

1. In general, those experiments initiated with a higher number of cells per well lead to a higher initial cell density after the overnight attachment and proliferation has taken place;

2. Within each initial seeding condition, the variability in initial cell density for the proliferation assays is very similar to the variability in initial cell density for the scratch assays; and,

3. There is a large variation in the initial cell density within each initial seeding condition.

Of these three features, the variation in the initial cell density within each initial seeding condition is very important. For example, the highest recorded initial cell density for initial seeding condition 1 (12,000 cells per well) is greater than the smallest recorded initial cell density for initial seeding condition 3 (20,000 cells per well). This means that we ought to take great care when selecting particular experimental replicates from the 96 data sets in Figure 3.2, otherwise our results could be misleading when we try to examine how the results depend on the initial cell density.

We select three replicates from each initial seeding condition for both the proliferation and scratch assays so that the initial cell density for the initial seeding condition 3 is greater than the initial cell density for the initial seeding condition 2, which is greater than the initial cell density for the initial seeding condition 1. Furthermore, we select three replicates for both the proliferation and scratch assays from each initial seeding condition. These choices are made so that the initial cell density for each type of assay is approximately the same within each seeding condition. To satisfy these constraints we choose three replicates from each set of 16 experimental replicates. The selected replicates are indicated in Figure 3.2.
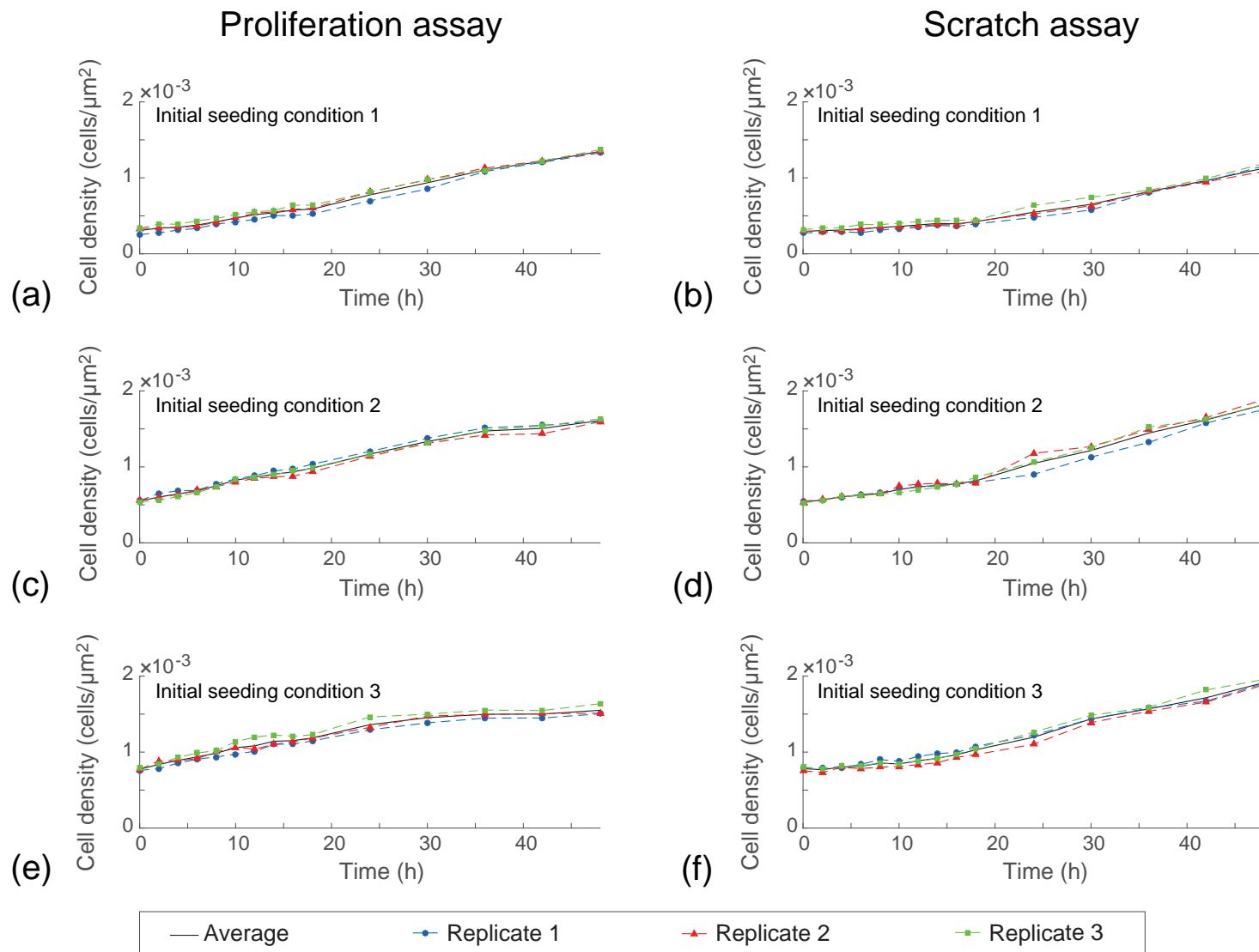
Figure 3.2: Variation of initial cell densities. Initial cell densities in the 96 replicates of proliferation and scratch assays. Results for the three different initial seeding conditions are shown. Initial seeding condition 1 corresponds to 12,000 cells per well; initial seeding condition 2 corresponds to 16,000 cell per well; and initial seeding condition 3 corresponds to 20,000 cells per well. For each initial seeding condition, each experiment is repeated $n = 16$ times and the variation in initial cell density is illustrated by comparing the spread of estimates of cell density on the horizontal axis. Each blue square represents an individual replicate of the proliferation assay, and each red circle represents an individual replicate of the scratch assay. The 18 black triangles indicate the individual replicates chosen to construct the cell density information.

Cell density information

Using the previously identified three experimental replicates for each type of assay and each initial seeding condition (Figure 3.2), we plot the evolution of the cell density as a function of time for each experimental replicate, as shown in Figure 3.3. We also superimpose, in Figure 3.3, the evolution of the average cell density for each type of assay and each initial seeding condition. We see that the differences in initial density between the proliferation assay and the scratch assay are minimal. The most obvious trend in the data is that the cell density in both the proliferation assay and the scratch assay increases dramatically with time, regardless of the initial condition. It is worth emphasizing that seeding condition 1 involves a relatively small initial cell density, whereas seeding conditions 2 and 3 are not particularly small. For example, the initial cell density for initial seeding condition 3 is approximately equal to the cell density for initial seeding condition 1 after a period of 24 hours has elapsed. Therefore our experimental design allows us to make a clear distinction between the effects of small cell density, which would appear more strongly and for longer in initial seeding condition 1 than initial seeding condition 3, and the effects of early time, which would appear equally in all three initial seeding conditions. We return to this issue later. It is also worth noting that for *in vitro* experiments it is difficult to maintain cells for a long period of time [71]. Therefore, in Figure 3.3 the cell density does not reach a constant level within the time frame of the experiment, especially for those with lower initial cell densities.

We note that it could be possible to calibrate the solution of Equation (3.1) to any of the density curves in Figure 3.3, and this approach has been widely used [14, 111, 116, 118]. However, there is no guarantee that simply fitting the solution of the logistic equation to this kind of data means that the logistic model describes the underlying mechanism [112]. To provide further insight into whether the logistic model applies to these data, we re-interpret the data in terms of the per capita growth rate.

Figure 3.3: Temporal evolution of cell density. Results in (a)–(f) correspond to proliferation and scratch assays initiated with 12,000 (initial seeding condition 1); 16,000 (initial seeding condition 2); and, 20,000 (initial seeding condition 3) cells per well, as indicated. Cell density profiles are shown at two–hour intervals during the first 18 hours, and at six–hour intervals during the remaining 30 hours of the experiment. For each experiment, we report results for three identically prepared experimental replicates, and the average of these three data sets is also shown.

3.3.2 Per capita growth rate

To estimate the per capita growth rate, $(1/C)(\mathrm{d}C/\mathrm{d}t)$, we use the cell density data in Figure 3.3 to estimate $\mathrm{d}C/\mathrm{d}t$ using a finite difference approximation. Our estimate of $\mathrm{d}C/\mathrm{d}t$ at the first and last time points is obtained using a forward and backward difference approximation, respectively, while our estimates at all other time points are obtained using an appropriate central difference approximation [17]. With these estimates, we plot the per capita growth rate as a function of the density in Figure 3.4. Results are shown for both proliferation and scratch assays, for the three initial densities considered.

To interpret our results, it is instructive to recall that the data in Figure 3.3 show that the cell density, in each type of experiment for all three initial densities of cells, increases with time. Therefore, when we interpret each plot showing the per capita growth rate as a function of density in Figure 3.4, it is useful to recall how the data in these plots vary with time during the experiment. Data for smaller values of C in each subfigure in Figure 3.4 correspond to the early part of the experiment, and hence small t. In contrast, data for larger values of C in each subfigure in Figure 3.4 correspond to the latter part of the experiment, and hence larger t.

If the logistic growth model is valid, then we expect that the per capita growth rate will be a linearly decreasing function of the density. In contrast, other kinds of carrying-capacity limited growth models, such as the Gompertz law, imply a non-linear relationship [66]. Visual inspection the per capita growth rate data in Figure 3.4 reveals several trends:

1. The relationship between the per capita growth rate and the density in the proliferation assay is very different to the relationship between the per capita growth rate and the density in the scratch assay;
2. The relationship between the per capita growth rate for each proliferation assay, at each initial seeding condition, appears to be reasonably well approximated by a linearly decreasing function of density; and,
3. The relationship between the per capita growth rate for each scratch assay is more complicated, with the per capita growth rate increasing with density when the density is small, and then decreasing with density when the density is sufficiently large.

These observations suggest that the proliferation of cells in the scratch assay is very different to the proliferation of cells in a proliferation assay. Because we are examining the proliferation of cells that are located well away from the scratch, this result implies that the process of scratching the monolayer can induce non-local effects.
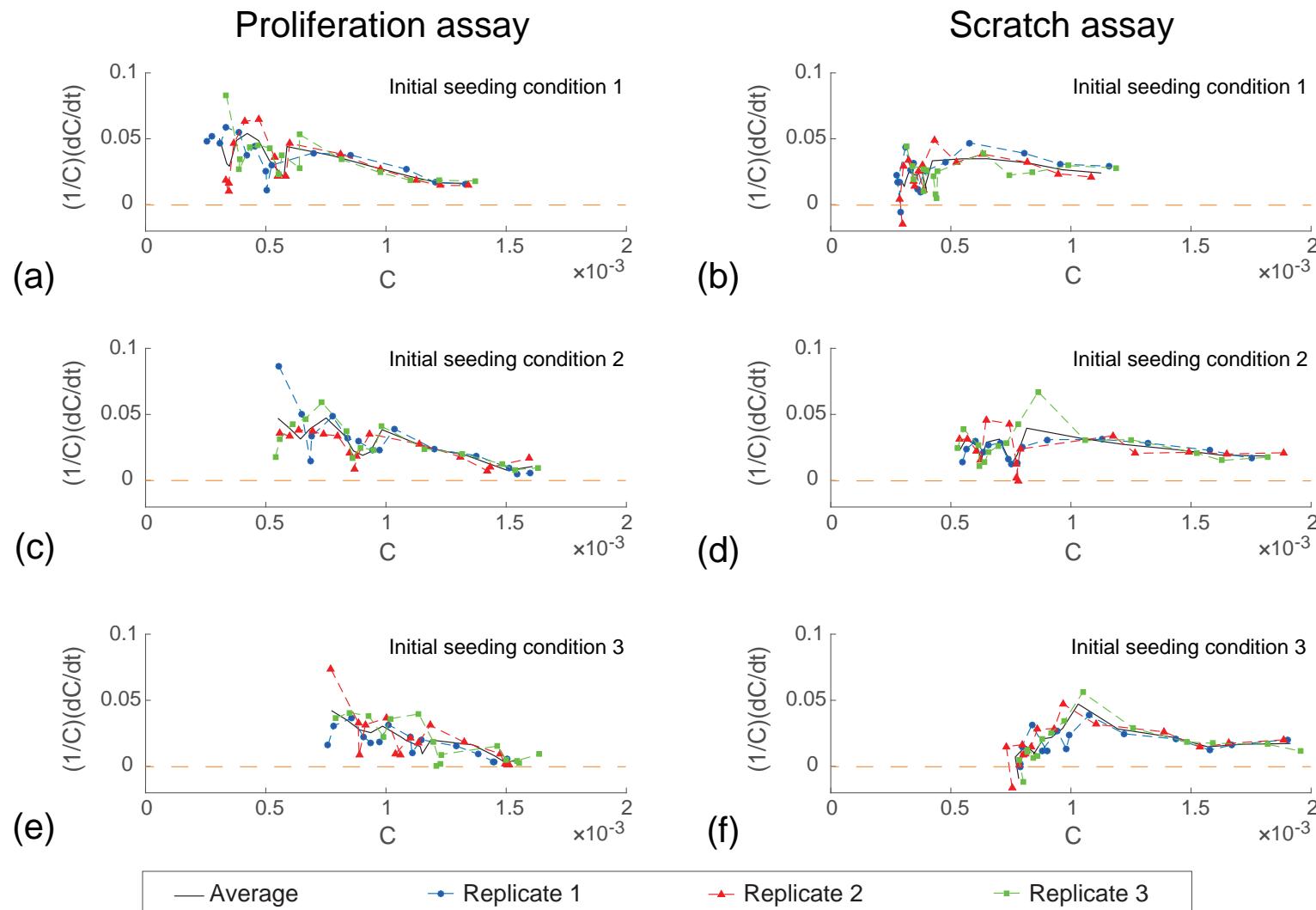
Instead of relying on visual interpretation alone, we now attempt to match the per capita growth rate data and the logistic growth model by fitting a series of straight lines to the averaged per capita growth rate data using `lsqcurvefit` [79]. Results in Figure 3.5 show the least–squares straight line and the coefficient of determination for each

data set. Results for the proliferation assay (Figure 3.5(a), (c) and (e)) suggest that the putative linear relationship is reasonable since our straight lines have negative slope and the coefficient of determination is reasonably high ($R^2 = 0.50 - 0.87$). In contrast, results for the scratch assay (Figure 3.5(b), (d) and (f)) show that the least–squares linear regression is a poor match to the data with a very low coefficient of determination ($R^2 = 0.04 - 0.16$). Indeed, the least–squares straight lines in Figure 3.5(b) and (f) are particularly troublesome since they have a positive slope which is biologically unrealistic, suggesting that the quantity λ/K is negative. Therefore, it is clear that the per capita growth data for the scratch assays does not follow a linearly decreasing straight line for the entire duration of the experiment, and the commonly–invoked logistic model does not appear to match these data at all.
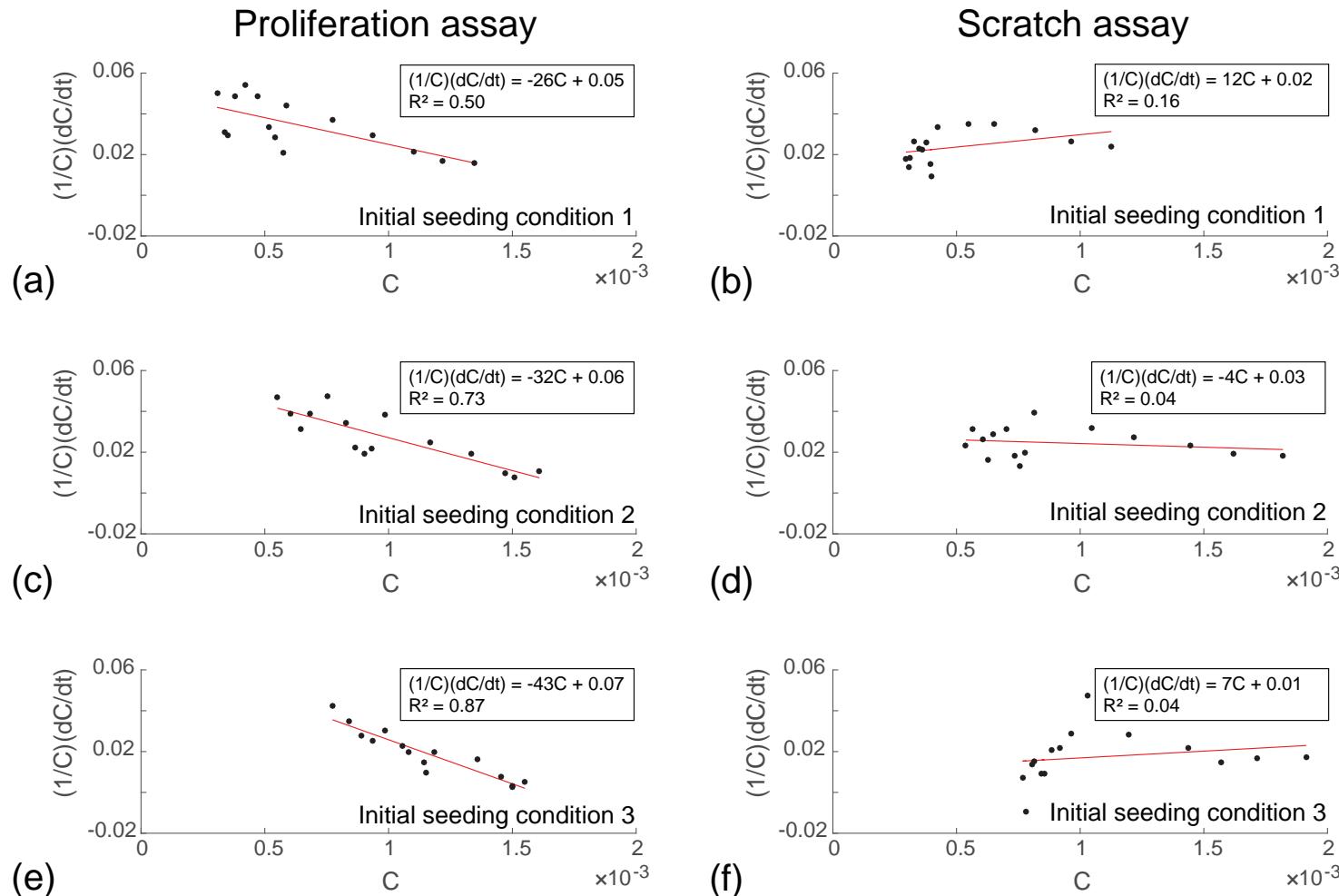
The fact that we observe two very different trends in the per capita growth rate data for the scratch assay motivates us to conjecture that the proliferation of cells in the scratch assay, far away from the location of the scratch, takes place in two phases. The first phase, which occurs at early time, involves the per capita growth rate increasing with density. This trend is the opposite of what we expect if the logistic growth model is valid and not what we observe in the proliferation assay. The second phase, which occurs at later time, involves the per capita growth rate decreasing with the density. These two phases occur consistently across all three initial seeding conditions (Figure 3.4(b), (d) and (f)). A schematic illustration of the differences observed between the per capita growth rate in the scratch assay and the proliferation assay is given in Figure 3.6. In this schematic, we refer to the first phase in the scratch assay as the *disturbance phase*, and the second phase in the scratch assay as the *growth phase*. The per capita growth data in the proliferation assay appear to be similar to the growth phase of the scratch assay for the entire duration of the experiment.

In the schematic (Figure 3.6), we suggest that the relationship between the per capita growth rate and the density during the growth phase is a linearly decreasing function, which is consistent with the logistic model. To quantitatively examine whether this assumption is valid for our data set we now construct a series of least–squares straight lines to our averaged per capita growth rate data during the growth phase. To examine this question, we need to quantitatively distinguish between the end of the first phase and the beginning of the second phase. We separate the data in Figure 3.5(b), (d) and (f) into two groups, the disturbance phase for $t < 18$ hours, and the growth phase for $t > 18$ 18 hours. To examine whether the data in the growth phase appear to be logistic we determine the least–squares linear relationship using `lsqcurvefit` [79] for the data in the growth phase. This least–squares straight line is superimposed on the averaged data for $t \geq 18$ hours in Figure 3.7(b), (d) and (f). Again, a visual comparison of the match between the linear regression and the data in the growth phase, and the much higher values of coefficient of determination ($R^2 = 0.75 - 0.91$) suggest that the putative linear relationship is reasonable.
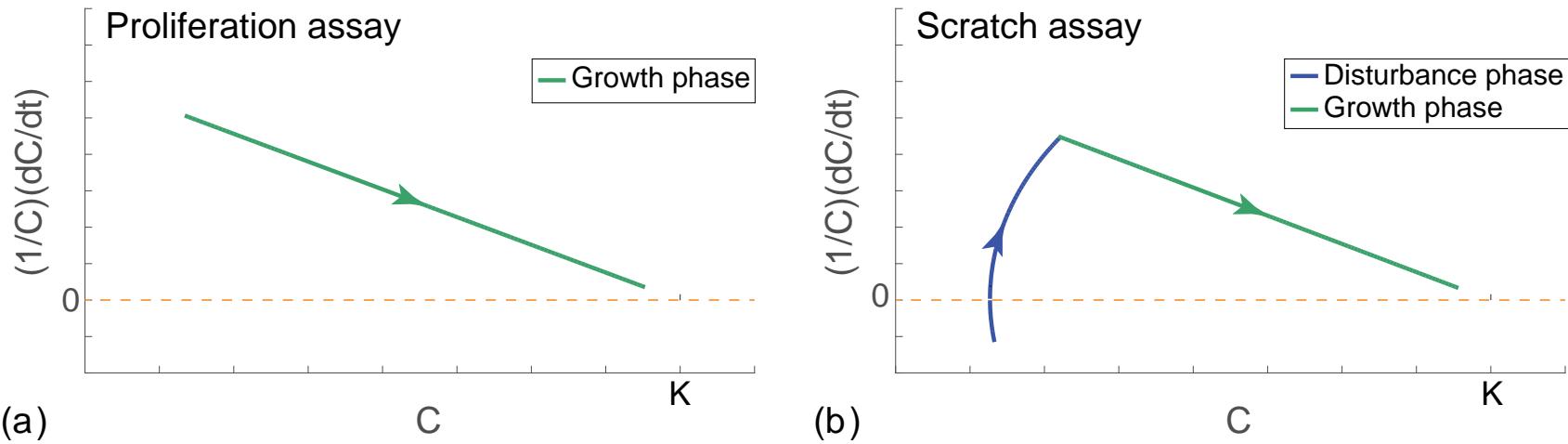
In summary, we have used the per capita growth rate information in Figure 3.4 to make a distinction between the disturbance phase and the growth phase in the scratch assay. These differences are highlighted in the schematic in Figure 3.6. Furthermore, guided by the observed relationship between the per capita growth rate and the density in the proliferation assay we assume that the logistic growth model applies and fit a straight line to the per capita growth rate data and find that the match between the data and the straight line appears to be reasonable. Similarly, we assume that the logistic growth model applies to the growth phase in the scratch assay, for $t \geq 18$ hours. Fitting a straight line to the per capita growth data suggests that the logistic growth model is reasonable in the growth phase for the scratch assay. Now that we have used the per capita growth rate data to identify the disturbance and growth phases in the scratch assay, as well as providing evidence that cells proliferate logistically in the growth phase, we re–examine the cell density profiles with a view to estimating λ and K.
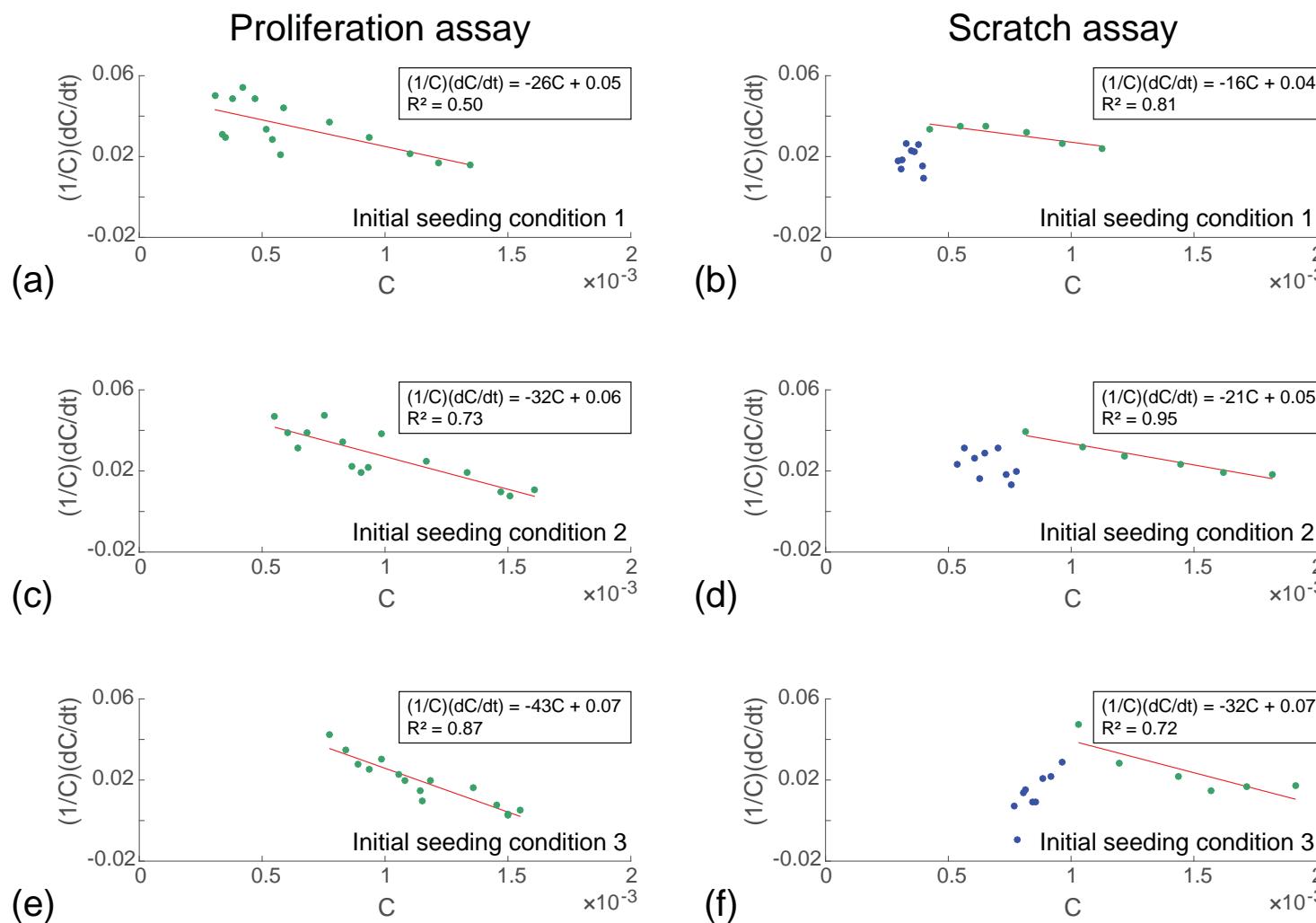
Figure 3.4: Per capita growth rates as a function of cell density. Results in (a)–(f) correspond to proliferation and scratch assays initiated with 12,000 (initial seeding condition 1); 16,000 (initial seeding condition 2); and, 20,000 (initial seeding condition 3) cells per well, as indicated. Per capita growth rate data is calculated using the data in Figure 3.3. For each experiment, we report results for three identically prepared experimental replicates, and the average of these three experimental replicates is also shown.

Figure 3.5: Least–squares straight line fit to per capita growth rate data for the entire duration of the experiment. Results in show the average per capita growth rate data as a function of density for both proliferation and scratch assays initiated with 12,000 (initial seeding condition 1); 16,000 (initial seeding condition 2); and, 20,000 (initial seeding condition 3) cells per well, as indicated. Black dots correspond to averaged data for the entire duration of the experiment. Solid lines show the least–squares linear relationship between the averaged per capita growth rate and averaged density, with R^2 indicating the coefficient of determination.

Figure 3.6: Schematic illustration of the differences between the proliferation and scratch assays. (a) Schematic showing the per capita growth rate as a function of density for the proliferation assays. (b) Schematic of the per capita growth rate for the scratch assays illustrating two phases of proliferation. The solid blue line indicates the disturbance phase in the scratch assay, and the solid green line indicates the growth phase in both the proliferation and scratch assays. The arrow heads indicate the direction of increasing time.

Figure 3.7: Straight line fit to per capita growth rates in the growth phase. Results in (a)–(f) show the average per capita growth rate data as a function of density for both proliferation and scratch assays initiated with 12,000 (initial seeding condition 1); 16,000 (initial seeding condition 2); and, 20,000 (initial seeding condition 3) cells per well, as indicated. Green dots correspond to averaged data in the growth phase (Figure 3.5), and blue dots correspond to averaged data in the disturbance phase (Figure 3.5). The solid lines show the least–squares linear relationship between the averaged per capita growth rate and averaged density, with R^2 indicating the coefficient of determination. The least–squares straight line is constructed using data from 0–48 hours in the proliferation assays, and using data from 18–48 hours in the scratch assays.

3.3.3 The logistic growth model

To calibrate the logistic growth model to our data from the proliferation assay, we match the solution of Equation (3.1) to the averaged data in Figure 3.3(a), (c) and (e) over the entire duration of the experiment, $0 \leq t \leq 48$ hours. To calibrate the logistic growth model to our data from the scratch assay, accounting for the differences in the disturbance and growth phases, we match the solution of Equation (3.1) to the averaged data in Figure 3.3(b), (d) and (f) during the growth phase only, $18 \leq t \leq 48$ hours. This provides us with six estimates of $\bar{\lambda}$ and $\bar{K}$. To demonstrate the quality of the match between the experimental data and the calibrated logistic model, we superimpose the experimental data and Equation (3.2) with $\lambda = \bar{\lambda}$ and $K = \bar{K}$, for each initial seeding condition and for both assays in Figure 3.8. These results show that the quality of match between the solution of the calibrated model and the experimental data is excellent. Our estimates of λ and K are summarised in Tables 3.1 and 3.2 for the proliferation assay and the scratch assay, respectively. In summary, our estimates of λ vary within the range $\lambda = 0.048 - 0.067$ h^{-1}, and our estimates of K vary within the range $K = 1.6 - 2.5 \times 10^{-3}$ cells/μm^2. Strictly speaking, since λ and K are supposed to be constants in Equation (3.1), the fact that we see only a relatively small variation in our estimates of these parameters is encouraging. In particular, we also report, in Tables 3.1 and 3.2, the sample standard deviation showing the variability of our estimates. Overall, we find that the coefficient of variation is approximately 10%, which is relatively small when dealing with this kind of biological data [122].

Table 3.1: Estimates of $\bar{\lambda}$ and $\bar{K}$ for the proliferation assay using data from $0 \leq t \leq 48$ hours. All parameter estimates are given to two significant figures. Results are reported as the sample mean and the uncertainty is quantified in terms of the sample standard deviation.

Initial seeding condition	$\bar{\lambda}$ (/h)	$\bar{K}$ (cells/μm^2)
1	0.052 ± 0.004	$2.0 \times 10^{-3} \pm 8 \times 10^{-5}$
2	0.059 ± 0.006	$1.8 \times 10^{-3} \pm 6 \times 10^{-5}$
3	0.067 ± 0.009	$1.6 \times 10^{-3} \pm 2 \times 10^{-5}$
Average	0.059 ± 0.008	$1.8 \times 10^{-3} \pm 2 \times 10^{-4}$

We now explore how our estimates of λ and K are sensitive to whether or not we account for the differences in the disturbance and growth phases in the scratch assay. We repeat the same calibration process as described for the results in Figure 3.8, except now we take the standard, naive approach and calibrate the solution of Equation (3.1) to the averaged data in Figure 3.3(b), (d) and (f) over the entire duration of the scratch assay, $0 \leq t \leq 48$ hours. This procedure provides us with three additional estimates of $\bar{\lambda}$ and $\bar{K}$ for the scratch assay, as summarised in Table 3.3.

Table 3.2: Estimates of $\bar{\lambda}$ and $\bar{K}$ for the scratch assay using data from $18 \leq t \leq 48$ hours. All parameter estimates are given to two significant figures. Results are reported as the sample mean and the uncertainty is quantified in terms of the sample standard deviation.

Initial seeding condition	$\bar{\lambda}$ (/h)	$\bar{K}$ (cells/μm^2)
1	0.051 ± 0.009	$2.1 \times 10^{-3} \pm 2 \times 10^{-3}$
2	0.059 ± 0.02	$2.4 \times 10^{-3} \pm 1 \times 10^{-3}$
3	0.048 ± 0.008	$2.5 \times 10^{-3} \pm 2 \times 10^{-4}$
Average	0.053 ± 0.005	$2.3 \times 10^{-3} \pm 2 \times 10^{-4}$
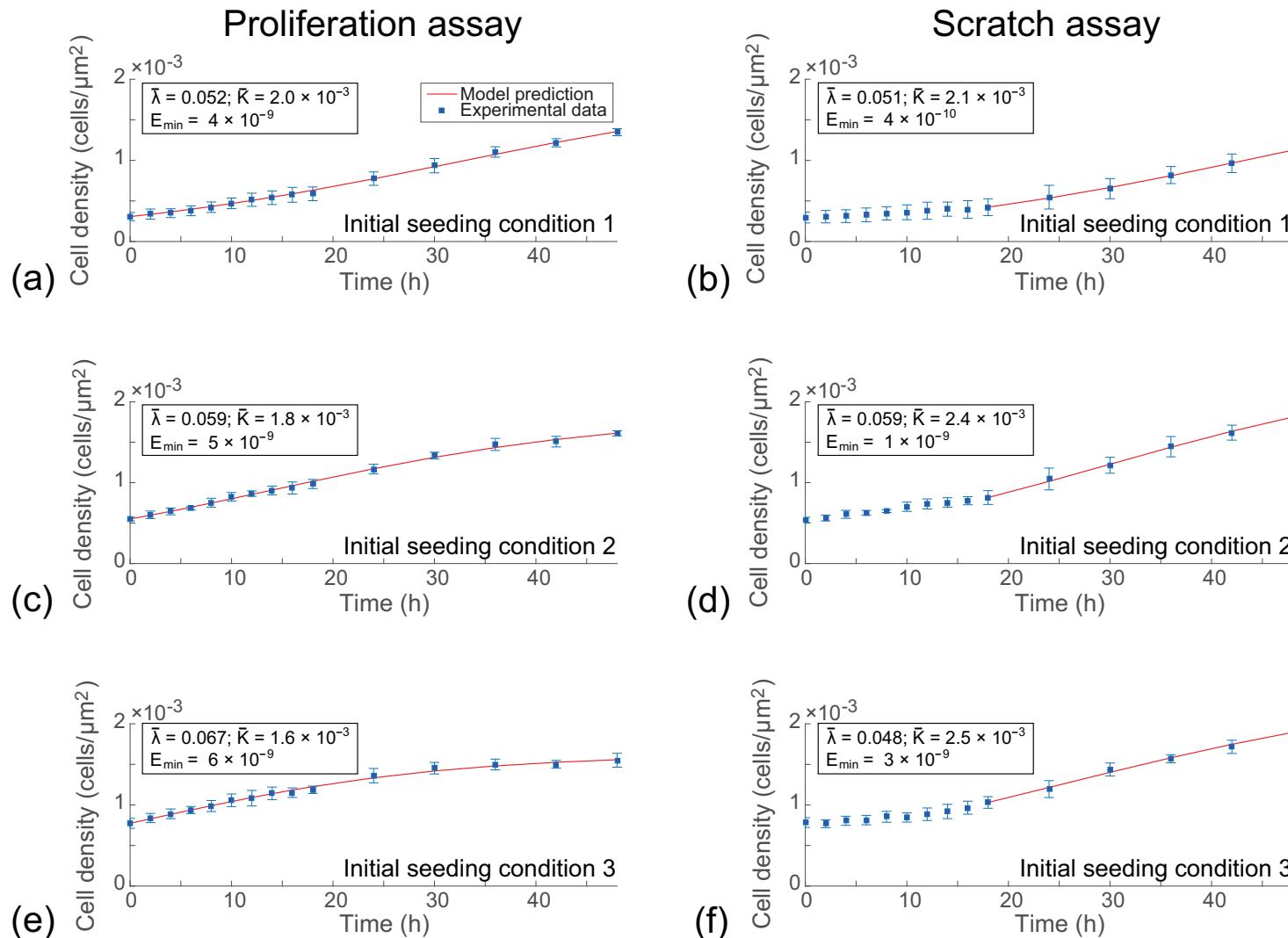
To demonstrate the quality of the match between the experimental data and the calibrated logistic model, we superimpose the experimental data and Equation (3.2) with $\lambda = \bar{\lambda}$ and $K = \bar{K}$, for each initial seeding condition and for both assays in Figure 3.9. When we visually compare the quality of the match between the experimental data in Figure 3.8 and Figure 3.9, and the corresponding calibrated solution of the logistic equation, there does not appear to be any significant difference at all. It is worth noting that the values of $E_{\min}$ in Figure 3.9(b), (d), and (f) are an order of magnitude greater than the corresponding values in Figure 3.8(b), (d) and (f). This implies that the match between the logistic model and the experimental data is improved when we ignore that data during the disturbance phase. However, at first glance, these differences are visually indistinguishable when we compare the results in Figure 3.8 and 3.9. In contrast, when we examine the estimates of $\bar{K}$ and $\bar{\lambda}$ in Table 3.3, the importance of properly accounting for the disturbance phase in the scratch assay becomes strikingly obvious. For example, taking this latter approach, our estimates of the carrying capacity vary within the range $K = 1.6 \times 10^{-3} - 2.8 \times 10^7$ cells/μm^2, and our estimates of the proliferation rate vary within the range $\lambda = 0.019 - 0.067$ h^{-1}. We recall that λ and K are supposed to be constants in Equation (3.1), and so the fact that this naive calibration process suggests that the least–squares estimate of the carrying capacity density varies of many order of magnitude provides a clear illustration that this standard approach to calibrating the logistic equation to our experimental data is problematic. We note that the results of the Levenberg–Marquardt algorithm are robust, returning the same least–squares estimates of $\bar{\lambda}$ and $\bar{K}$ for any positive initial estimate of K and λ in the iterative algorithm [79].

Comparing the ranges of estimates for λ and K in Tables 3.2 and 3.3 shows that the model calibration procedure is extremely sensitive. For example, our range of estimates of K when we account for the disturbance phase is smaller than a factor of two amongst the six estimates. In contrast, when we neglect the disturbance phase, our estimates of K vary across more than ten orders of magnitude amongst the six estimates. Similarly, our range of estimates of λ when we account for the disturbance phase is smaller than a factor of 1.5 among the six estimates. Again, in contrast, when we take a standard approach and neglect the disturbance phase our estimates of λ vary by more than a factor of three amongst the six estimates.
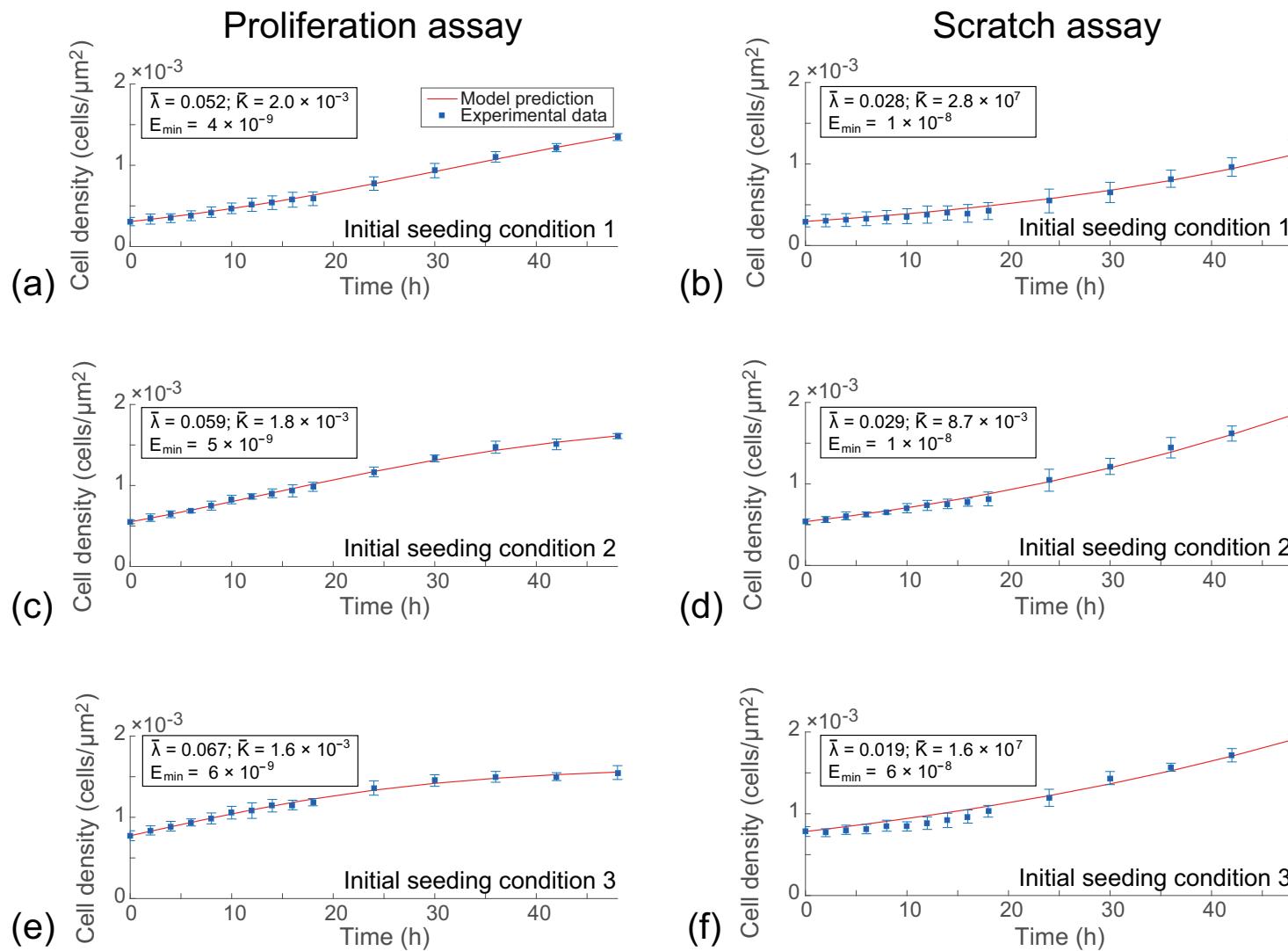
Table 3.3: Estimates of $\bar{\lambda}$ and $\bar{K}$ for the scratch assay using data from $0 \leq t \leq 48$ hours. All parameter estimates are given to two significant figures. Results are reported as the sample mean and the uncertainty is quantified in terms of the sample standard deviation.

Initial seeding condition	$\bar{\lambda}$ (/h)	$\bar{K}$ (cells/μm^2)
1	0.028 ± 0.001	$2.8 \times 10^7 \pm 1 \times 10^7$
2	0.029 ± 0.005	$8.7 \times 10^{-3} \pm 3 \times 10^6$
3	0.019 ± 0.0002	$1.6 \times 10^7 \pm 6 \times 10^6$
Average	0.025 ± 0.006	$1.5 \times 10^7 \pm 1 \times 10^7$

Figure 3.8: Calibrated solutions of the logistic growth equation using data from the growth phase. Results in (a)–(f) correspond to proliferation and scratch assays initiated with 12,000 (initial seeding condition 1); 16,000 (initial seeding condition 2); and, 20,000 (initial seeding condition 3) cells per well, as indicated. For each type of experiment the calibrated solution of the logistic growth equation (solid line) is compared to the experimental data in the growth phase (18–48 hours for scratch assays and 0–48 hours for proliferation assays). The least–squares estimates of $\bar{\lambda}$ and $\bar{K}$ are shown.

Figure 3.9: Calibrated solutions of the logistic growth equation using the entire data set. Results in (a)–(f) correspond to proliferation and scratch assays initiated with 12,000 (initial seeding condition 1); 16,000 (initial seeding condition 2); and, 20,000 (initial seeding condition 3) cells per well, as indicated. For each type of experiment the calibrated solution of the logistic growth equation (solid line) is compared to the entire experimental data set. The least–squares estimates of $\bar{\lambda}$ and $\bar{K}$ are shown.

3.4 Discussion

In this work we investigate the suitability of the logistic growth model to describe the proliferation of cells in scratch assays. Scratch assays are routinely used to study the ability of a population of cells to re–colonise an initially vacant region on a two–dimensional substrate [65, 71, 116, 118]. Most experimental interpretations of scratch assays are made using relatively straightforward measurements [71]. However, to provide additional insights into the mechanisms involved in the re–colonisation process, some previous studies have calibrated the solution of a reaction–diffusion equation to data from a scratch assay [14, 49, 54, 76, 77, 98, 100, 102, 103, 106]. In these reaction–diffusion equations, it is commonly assumed that carrying capacity–limited proliferation of cells can be described by a logistic growth model. However, the suitability of this assumption is rarely examined beyond the process of simply calibrating the solution of the relevant model to match the experimental data.

To examine the suitability of the logistic growth model, we perform a series of scratch assays and proliferation assays for three different initial cell densities. Cell proliferation assays are prepared in exactly the same way as a scratch assay, except that the monolayer of cells is not scratched. This allows us to treat the cell proliferation assays as a control experiment so that we can examine whether the process of artificially scratching the monolayer of cells affects the way that cells proliferate, even when those cells are located far away from the scratch. Instead of examining the dynamics of the cell density near the scratched region where there will be a net flux of cells into the vacant region [49], we quantify the cell density in two subregions that are located far behind the location of the scratch, where the cell density is approximately spatially uniform (Supplementary material). This means that the temporal dynamics of the cell density in these subregions is due to cell proliferation only [54].

We plot the time evolution of cell density, far away from the initially scratched region, in both the scratch and proliferation assays. To examine whether our results are sensitive to the initial density of cells, we repeat each experiment using three different initial cell densities. Plots of the evolution of the cell density are given over a total duration of 48 hours, and these plots appear to correspond to a series of sigmoid curves. At this point it would be possible to simply calibrate the solution of the logistic growth model to these data to provide an estimate of the proliferation rate, λ, and the carrying capacity density, K. This is a standard approach that has been used by us [54] and many others [14, 100, 118]. However, while this standard calibration procedure can be used to provide estimates of the parameters, this model calibration procedure does not provide any validation that logistic growth is relevant [112].

Rather than calibrating the logistic growth model to our experimental data, we attempt to assess the suitability of the logistic growth model by converting the cell density evolution profiles into plots of the per capita growth rate as a function of density. We find that the

plots of the per capita growth rate as a function of density reveal several key differences between the scratch and proliferation assays. If the logistic growth model is valid, then we expect to see a decreasing linear relationship between the per capita growth rate and the cell density for the entire duration of the experiment. While the plots of the per capita growth rate as a function of density for the proliferation assays appear to be consistent with the logistic model, the per capita growth rate data for the scratch assays are very different. For the scratch assay data, the per capita growth rate increases with cell density at low density during the early part of the experiment. This behaviour, which is observed for all three initial densities of cells in the scratch assays, is the opposite of what we would expect if the logistic growth model were valid. However, at higher cell densities during the latter part of the experiment, we observe that the per capita growth rate in the scratch assays appears to decrease, approximately linearly, with the cell density. This motivates us to propose that cell proliferation in a scratch assay involves two phases: (i) a *disturbance phase* in which proliferation does not follow the logistic growth model during the early part of the experiment; and, (ii) a *growth phase* where proliferation is approximately logistic during the latter part of the experiment. Guided by our per capita growth rate data, it appears that the disturbance phase in the scratch assays lasts for approximately 18 hours before the growth phase commences.

To estimate the parameters in the logistic growth model, we calibrate the solution of the model to our cell proliferation data for the entire duration of the experiment. This calibration procedure gives estimates of λ and K that are approximately consistent across the three initial conditions. We then calibrate the solution of the logistic growth model to the data from the growth phase in the scratch assay. This procedure also gives estimates of λ and K that are consistent across the three initial conditions, as well as being consistent with the estimates obtained from the cell proliferation assays. In contrast, if we take a naive approach and simply calibrate the solution of the logistic growth equation to the scratch assay data for the entire duration of the experiment, our estimates of λ and K vary wildly, despite the fact that the match between the experimental data and the calibrated solution of the logistic growth equation looks very good.

The results of our study strongly suggest that care ought to be taken when applying a logistic growth model, or a reaction–diffusion equation with a logistic source term, to describe scratch assays. Simply calibrating a mathematical model to experimental data might appear to produce an excellent match between the solution of the model and the experimental data, but this commonly–used procedure does not guarantee that the model is at all relevant [112]. Our results suggest that cell proliferation is impacted by the scratching procedure in a scratch assay, and that we require some time to pass before the disturbance phase ends. This is important because previous applications of logistic growth models and reaction–diffusion equations with logistic source terms have been calibrated to data from scratch assays without any regard for the disturbance phase [14, 49, 54, 118].

It is also relevant to note that for the particular cell line we use, the disturbance phase that we identify lasts for approximately 18 hours. This is important because many scratch

assays are performed for relatively short periods of time [71] and it is possible that standard experimental protocols do not allow for a sufficient amount of time to pass for the disturbance phase to end. Therefore, we suggest that scratch assays should be maintained for as long as possible so that sufficient time is allowed for the disturbance phase to pass.

It is worthwhile to note, and discuss, the fact that some of the features of our proposed two–phase growth model appear to be similar to the Allee effect [5, 55, 69, 94, 101, 114]. Typically, Allee growth kinetics are normally invoked to describe some kind of low-density reduction in proliferation, relative to the logistic model [69, 114]. The Allee growth model is given by

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \lambda C(t) \left(1 - \frac{C(t)}{K} \right) \left(\frac{C(t)}{A} - 1 \right),\tag{3.4}$$

where the parameter A is called the Allee threshold. The key difference between the Allee growth model (Equation (3.4)) and the standard logistic model (Equation (3.1)), is the inclusion of the third factor on the right hand side of Equation (3.4). The incorporation of this factor has several consequences: (i) the growth rate is negative for $C(t) < A$ (assuming $C(t) < K$); (ii) the growth rate is positive for $C(t) > A$ (assuming $C(t) < K$); and, (iii) the relationship between the per capita growth rate and the density is quadratic. In many previous implementations of Allee growth models, an argument is made that the growth rate at small densities is reduced, relative to the logistic model, because of some kind of biological competition [55, 69, 114], corresponding to $A \ll K$. Therefore, the Allee model is often used to represent reduced growth at small densities, $C(t) \ll K$. The experimental data we present in Figures 3.4 and 3.7 are inconsistent with the Allee model for two reasons. First, the per capita growth data in Figure 3.4 corresponds to a reduced growth rate at early times during the experiment. This reduction in growth rate is observed across a range of initial densities, including seeing conditions 2 and 3 which do not involve small densities, as discussed in Section 3.3.1. Second, the per capita growth data in Figure 3.7 varies approximately linearly with density during the growth phase, whereas the Allee model implies that the relationship is quadratic.

One of the limitations of our study is that we have not identified the precise mechanism that causes the disturbance phase or the mathematical form of the disturbance phase. However it seems clear that the process of scratching a monolayer of cells has some impact on the proliferative behaviour of the cells away from the scratch, suggesting that either chemical or mechanical disturbance is transported throughout the experimental well as consequence of the scratching action. For example, the washing procedure performed in scratch assays could lead to shear stress in fluid. Regardless of the mechanism at play, our procedure of converting the cell density profiles into plots of the per capita growth rate allows us to identify the result of this disturbance. Another limitation of our work is that we deal only with one particular cell line, and it is not obvious how our estimate of the duration of the disturbance phase will translate to other cell lines. In this work we study the proliferation mechanism by converting the density data into per capita growth data and exploring whether the relationship between the per capita growth

and density is approximately described by a linear function. We acknowledge that more sophisticated statistical techniques could be employed to provide further information [24]. However, since the main aim of this study is to explore the suitability of the logistic growth model for describing cell proliferation in a scratch assay, we do not pursue these more advanced statistical techniques here. Instead, we suggest that this could be the topic of a future study. Moreover, the experimental data in Chapter 2 can be re-visited to take the disturbance phase into account. Then the Fisher–Kolmogorov model and the Porous–Fisher model can be calibrated to data in the growth phase only, to investigate if a better match can be achieved.

Acknowledgments

3.5   Supplementary material

3.5.1   Location of the subregions

A key assumption in this work is that the two subregions we choose to quantify temporal changes in cell density are located sufficiently far away from the scratch that the cell density is approximately spatially uniform in these locations. To examine this assumption, following Jin et al. [49], we use the Fisher–Kolmogorov equation, which is given by

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} + \lambda C\left[1 - \left(\frac{C}{K}\right)\right],\tag{3.5}$$
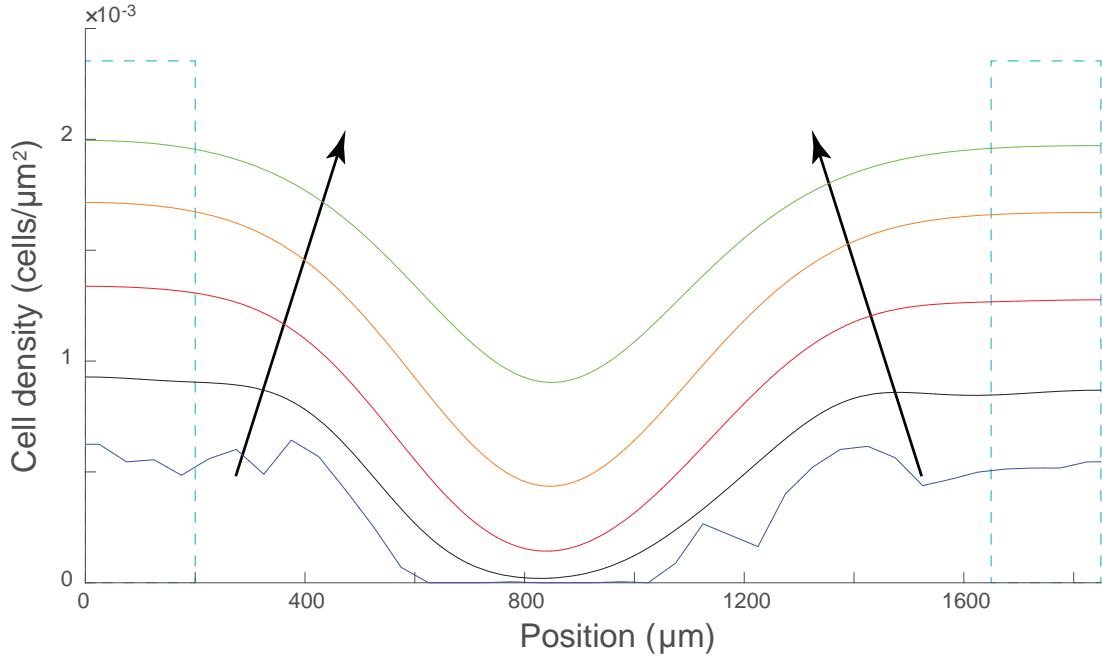
where $C(x,t)$ is the cell density as a function of position, x, and time, t. Here, D is the cell diffusivity, λ is the proliferation rate and K is the carrying capacity density. To match our experiments we have $0 \leq x \leq 1850$ μm [49], and to mimic the geometry of our experiments we impose symmetry boundary conditions ($\partial C/\partial x = 0$) at both $x = 0$ μm and $x = 1850$ μm [49]. We specify the initial cell density, $C(x,0)$, by using manual counting in a series of columns, of width 50 μm, across the image at $t = 0$ hours. Dividing the initial number of cells per column by the area of the column gives us an estimate of $C(x,0)$. Repeating this procedure across a number of identically prepared experiments allows us to average our estimates of $C(x,0)$ to reduce the fluctuations in the initial density profile.

To solve Equation (3.5) we uniformly discretise the spatial domain with grid spacing δx [81]. The spatial derivatives are approximated using a central–difference approximation, which leads to a system of coupled nonlinear ordinary differential equations that are integrated through time using a backward–Euler approximation with constant time steps of duration δt [81]. The resulting systems of coupled nonlinear algebraic equations are linearised using Picard iteration, with absolute convergence tolerance ϵ, and solved

using the Thomas algorithm [81]. We always choose δx, δt and ϵ so that our numerical algorithm produces grid–independent results.

We present results here for a scratch assay with one particular choice of initial seeding condition, 16,000 cells per well. Note that the results shown here for this initial seeding condition are representative of the results for the other two initial seeding conditions that we consider in the main document. Our estimate for the initial cell density profile from the experimental data at $t = 0$ hours is linearly interpolated to match the numerical discretisation. Using this information, we solve Equation (3.5) with $D = 570$ μm^2/h, $\lambda = 0.059$ /h and $K = 2.4 \times 10^{-3}$ cells/μm^2, which are representative estimates of λ and K from the main manuscript, and an estimate of D taken from our previous work [49].

Numerical solutions of Equation (3.5) are shown in Figure 3.10. Here we see that the initial fluctuations in $C(x,0)$ smooth out with time relatively quickly. Then, at later times, we see that there are negligible spatial gradients, $\partial C/\partial x \approx 0$, at the locations of the two subregions used to construct the cell density profile. This indicates that the net flux of cells into these subregions is approximately zero, and hence Equation (3.5) for $C(x,t)$ simplifies to the logistic growth model for $C(t)$ locally in these subregions that are located sufficiently far from the scratch. Ideally the two subregions used to count the cells could be further away from the scratch area. However, the field of view in the IncuCyte ZOOM$^{\text{TM}}$ assays cannot be moved or extended.

Figure 3.10: The numerical solution of the Fisher-Kolmorogov model for the scratch assay. Numerical solutions of Equation (3.5) are shown at $t = 0, 12, 24, 36, 48$ hours, with the arrow showing the direction of increasing t. The location of the two subregions are shown using the two cyan rectangles, as indicated. The numerical solution of Equation (3.5) is obtained with $D = 570~\mu m^2/h$, $\lambda = 0.059$ /h, $K = 2.4 \times 10^{-3}$ cells/μm^2, $\delta x = 0.5~\mu m$, $\delta t = 0.2$ h and $\epsilon = 10^{-5}$.

3.5.2 Experimental data

All raw experimental data used to construct the growth curves in Figure 3.3 are summarised as follows: (i) data for the cell proliferation assay are shown in Tables 3.4–3.6, corresponding to initial seeding conditions 1-3, respectively; and, (ii) data for the scratch assay are shown in Tables 3.7–3.9, corresponding to initial seeding condition 1-3, respectively. In Tables 3.4–3.9 we show data from three individual experimental replicates as well as averaged density data by averaging the three individual replicates from each experimental condition.

Table 3.4: Proliferation assay; Initial seeding condition 1

Time (h)			0	2	4	6	8	10	12	14	16	18	24	30	36	42	48
Cell density ($\times 10^{-3}$ cells/μm^2)	Replicate 1	Subregion 1	0.30	0.32	0.37	0.40	0.47	0.47	0.54	0.60	0.60	0.61	0.74	0.94	1.15	1.26	1.38
		Subregion 2	0.21	0.24	0.26	0.28	0.31	0.38	0.37	0.40	0.41	0.44	0.66	0.76	1.01	1.15	1.27
		Average	0.25	0.28	0.31	0.34	0.39	0.42	0.45	0.50	0.50	0.52	0.70	0.85	1.08	1.20	1.33
	Replicate 2	Subregion 1	0.35	0.38	0.37	0.38	0.44	0.53	0.59	0.60	0.63	0.66	0.85	1.00	1.17	1.28	1.36
		Subregion 2	0.32	0.31	0.32	0.35	0.39	0.41	0.48	0.50	0.54	0.54	0.78	0.96	1.08	1.17	1.32
		Average	0.33	0.34	0.35	0.37	0.41	0.47	0.54	0.55	0.58	0.60	0.81	0.98	1.13	1.23	1.34
	Replicate 3	Subregion 1	0.34	0.41	0.41	0.46	0.49	0.54	0.57	0.62	0.67	0.68	0.89	1.00	1.05	1.24	1.36
		Subregion 2	0.33	0.38	0.37	0.41	0.44	0.49	0.54	0.52	0.61	0.61	0.74	0.95	1.15	1.21	1.38
		Average	0.34	0.39	0.39	0.43	0.47	0.52	0.55	0.57	0.64	0.64	0.82	0.98	1.10	1.22	1.37
	Average		0.31	0.34	0.35	0.38	0.42	0.47	0.52	0.54	0.58	0.59	0.78	0.94	1.10	1.22	1.35

Table 3.5: Proliferation assay; Initial seeding condition 2

Time (h)			0	2	4	6	8	10	12	14	16	18	24	30	36	42	48
Cell density ($\times 10^{-3}$ cells/μm^2)	Replicate 1	Subregion 1	0.55	0.66	0.71	0.69	0.79	0.87	0.87	0.95	1.01	1.06	1.26	1.41	1.49	1.57	1.59
		Subregion 2	0.56	0.64	0.66	0.69	0.77	0.81	0.90	0.94	0.94	1.01	1.15	1.35	1.54	1.52	1.61
		Average	0.55	0.65	0.69	0.69	0.78	0.84	0.89	0.95	0.97	1.03	1.20	1.38	1.51	1.55	1.60
	Replicate 2	Subregion 1	0.57	0.62	0.67	0.72	0.82	0.87	0.88	0.92	0.97	0.98	1.18	1.34	1.37	1.47	1.62
		Subregion 2	0.54	0.58	0.61	0.68	0.66	0.73	0.82	0.82	0.79	0.89	1.10	1.28	1.47	1.40	1.57
		Average	0.56	0.60	0.64	0.70	0.74	0.80	0.85	0.87	0.88	0.93	1.14	1.31	1.42	1.44	1.59
	Replicate 3	Subregion 1	0.51	0.54	0.61	0.66	0.74	0.84	0.89	0.94	0.95	0.99	1.21	1.32	1.56	1.57	1.66
		Subregion 2	0.57	0.58	0.62	0.67	0.73	0.84	0.83	0.85	0.95	0.98	1.12	1.31	1.40	1.51	1.60
		Average	0.54	0.56	0.61	0.66	0.73	0.84	0.86	0.90	0.95	0.98	1.16	1.32	1.48	1.54	1.63
	Average		0.55	0.60	0.65	0.68	0.75	0.83	0.87	0.90	0.93	0.98	1.17	1.33	1.47	1.51	1.61

Table 3.6: Proliferation assay; Initial seeding condition 3

Time (h)			0	2	4	6	8	10	12	14	16	18	24	30	36	42	48
Cell density ($\times 10^{-3}$ cells/μm^2)	Replicate 1	Subregion 1	0.68	0.75	0.78	0.89	0.87	0.98	1.01	1.04	1.08	1.12	1.32	1.41	1.49	1.47	1.51
		Subregion 2	0.84	0.82	0.93	0.93	1.00	0.97	1.01	1.16	1.14	1.17	1.27	1.36	1.41	1.43	1.49
		Average	0.76	0.78	0.86	0.91	0.94	0.97	1.01	1.10	1.11	1.15	1.29	1.38	1.45	1.45	1.50
	Replicate 2	Subregion 1	0.80	0.92	0.88	0.92	0.96	1.03	1.00	1.08	1.10	1.19	1.34	1.50	1.55	1.53	1.51
		Subregion 2	0.75	0.85	0.90	0.91	1.05	1.09	1.08	1.13	1.17	1.18	1.31	1.45	1.44	1.48	1.51
		Average	0.77	0.89	0.89	0.91	1.00	1.06	1.04	1.10	1.14	1.18	1.33	1.48	1.49	1.51	1.51
	Replicate 3	Subregion 1	0.75	0.83	0.91	0.99	0.98	1.10	1.16	1.22	1.17	1.26	1.51	1.56	1.59	1.53	1.72
		Subregion 2	0.83	0.86	0.95	0.99	1.06	1.17	1.23	1.23	1.24	1.20	1.42	1.43	1.51	1.56	1.55
		Average	0.79	0.85	0.93	0.99	1.02	1.14	1.20	1.23	1.21	1.23	1.46	1.50	1.55	1.55	1.64
	Average		0.77	0.84	0.89	0.94	0.99	1.06	1.08	1.14	1.15	1.19	1.36	1.45	1.50	1.50	1.55

Table 3.7: Scratch assay; Initial seeding condition 1

Time (h)			0	2	4	6	8	10	12	14	16	18	24	30	36	42	48
Cell density $(\times 10^{-3}$ cells/μm^2)	Replicate 1	Subregion 1	0.25	0.26	0.28	0.26	0.29	0.31	0.31	0.35	0.32	0.35	0.43	0.58	0.82	1.00	1.33
		Subregion 2	0.30	0.31	0.30	0.30	0.33	0.35	0.38	0.41	0.41	0.43	0.53	0.58	0.79	0.91	0.99
		Average	0.27	0.29	0.29	0.28	0.31	0.33	0.35	0.38	0.36	0.39	0.48	0.58	0.80	0.96	1.16
	Replicate 2	Subregion 1	0.35	0.34	0.37	0.40	0.43	0.44	0.47	0.47	0.48	0.53	0.65	0.76	0.93	1.09	1.18
		Subregion 2	0.25	0.24	0.23	0.25	0.26	0.26	0.26	0.30	0.28	0.33	0.40	0.51	0.71	0.80	0.99
		Average	0.30	0.29	0.30	0.32	0.35	0.35	0.37	0.39	0.38	0.43	0.52	0.63	0.82	0.95	1.08
	Replicate 3	Subregion 1	0.40	0.44	0.44	0.47	0.46	0.50	0.54	0.53	0.56	0.56	0.78	0.84	0.96	1.08	1.36
		Subregion 2	0.23	0.25	0.25	0.30	0.31	0.30	0.32	0.35	0.32	0.33	0.49	0.65	0.72	0.90	1.02
		Average	0.32	0.34	0.34	0.39	0.39	0.40	0.43	0.44	0.44	0.45	0.64	0.75	0.84	0.99	1.19
	Average		0.30	0.31	0.31	0.33	0.35	0.36	0.38	0.40	0.40	0.42	0.55	0.65	0.82	0.96	1.13

Table 3.8: Scratch assay; Initial seeding condition 2

Time (h)			0	2	4	6	8	10	12	14	16	18	24	30	36	42	48
Cell density ($\times 10^{-3}$ cells/μm^2)	Replicate 1	Subregion 1	0.53	0.57	0.58	0.65	0.64	0.66	0.69	0.74	0.76	0.73	0.92	1.13	1.31	1.53	1.64
		Subregion 2	0.57	0.56	0.63	0.62	0.67	0.75	0.79	0.77	0.79	0.86	0.88	1.13	1.33	1.62	1.87
		Average	0.55	0.57	0.60	0.64	0.66	0.71	0.74	0.75	0.78	0.80	0.90	1.13	1.32	1.58	1.75
	Replicate 2	Subregion 1	0.56	0.61	0.67	0.66	0.66	0.77	0.82	0.85	0.82	0.84	1.24	1.38	1.62	1.78	1.98
		Subregion 2	0.51	0.53	0.55	0.59	0.63	0.71	0.73	0.72	0.72	0.74	1.12	1.16	1.36	1.52	1.80
		Average	0.54	0.57	0.61	0.62	0.65	0.74	0.77	0.78	0.77	0.79	1.18	1.27	1.49	1.65	1.89
	Replicate 3	Subregion 1	0.57	0.59	0.65	0.64	0.65	0.70	0.74	0.78	0.84	0.96	1.11	1.28	1.51	1.65	1.84
		Subregion 2	0.49	0.52	0.58	0.60	0.63	0.62	0.66	0.68	0.72	0.77	1.01	1.22	1.54	1.60	1.80
		Average	0.53	0.55	0.61	0.62	0.64	0.66	0.70	0.73	0.78	0.86	1.06	1.25	1.52	1.63	1.82
	Average		0.54	0.56	0.61	0.63	0.65	0.70	0.74	0.76	0.78	0.82	1.05	1.22	1.44	1.62	1.82

Table 3.9: Scratch assay; Initial seeding condition 3

Time (h)			0	2	4	6	8	10	12	14	16	18	24	30	36	42	48
Cell density $(\times 10^{-3}$ cells/μm^2)	Replicate 1	Subregion 1	0.85	0.84	0.87	0.91	0.98	0.94	1.03	1.07	1.09	1.14	1.30	1.52	1.60	1.66	1.93
		Subregion 2	0.73	0.75	0.72	0.77	0.82	0.82	0.85	0.89	0.90	1.01	1.14	1.35	1.56	1.68	1.88
		Average	0.79	0.79	0.80	0.84	0.90	0.88	0.94	0.98	1.00	1.08	1.22	1.44	1.58	1.67	1.90
	Replicate 2	Subregion 1	0.72	0.71	0.75	0.75	0.81	0.83	0.85	0.85	0.93	0.98	1.07	1.37	1.50	1.63	1.82
		Subregion 2	0.79	0.75	0.85	0.82	0.79	0.79	0.82	0.87	0.93	0.96	1.13	1.40	1.57	1.68	1.95
		Average	0.76	0.73	0.80	0.78	0.80	0.81	0.83	0.86	0.93	0.97	1.10	1.39	1.54	1.66	1.88
	Replicate 3	Subregion 1	0.75	0.75	0.80	0.79	0.84	0.81	0.87	0.85	0.90	1.00	1.18	1.55	1.64	1.81	1.97
		Subregion 2	0.86	0.82	0.83	0.85	0.87	0.88	0.89	0.98	1.05	1.09	1.34	1.42	1.53	1.83	1.94
		Average	0.80	0.78	0.82	0.82	0.86	0.84	0.88	0.92	0.97	1.05	1.26	1.48	1.59	1.82	1.95
	Average		0.78	0.77	0.80	0.81	0.85	0.85	0.89	0.92	0.97	1.03	1.20	1.44	1.57	1.72	1.91

4 Stochastic simulation tools and continuum models for describing two-dimensional collective cell spreading with universal growth functions

Abstract

Two–dimensional collective cell migration assays are used to study cancer and tissue repair. These assays involve combined cell migration and cell proliferation processes, both of which are modulated by cell–to–cell crowding. Previous discrete models of collective cell migration assays involve a nearest–neighbour proliferation mechanism where crowding effects are incorporated by aborting potential proliferation events if the randomly chosen target site is occupied. There are two limitations of this traditional approach: (i) it seems unreasonable to abort a potential proliferation event based on the occupancy of a single, randomly chosen target site; and, (ii) the continuum limit description of this mechanism leads to the standard logistic growth function, but some experimental evidence suggests that cells do not always proliferate logistically. Motivated by these observations, we introduce a generalised proliferation mechanism which allows non–nearest neighbour proliferation events to take place over a template of $r \geq 1$ concentric rings of lattice sites. Further, the decision to abort potential proliferation events is made using a *crowding function*, $f(C)$, which accounts for the density of agents within a group of sites rather than dealing with the occupancy of a single randomly chosen site. Analysing the continuum limit description of the stochastic model shows that the standard logistic source term, $\lambda C(1-C)$, where λ is the proliferation rate, is generalised to a universal growth function, $\lambda C f(C)$. Comparing the solution of the continuum description with averaged simulation data indicates that the continuum model performs well for many choices of $f(C)$ and r. For nonlinear $f(C)$, the quality of the continuum–discrete match increases with r.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

90

4.1   Background

Two–dimensional collective cell migration assays are routinely used to study combined cell migration and cell proliferation processes [91]. These assays provide insight into cancer [58] and tissue repair [76]. There are two different kinds of cell migration assays: (i) *Cell proliferation assays*, as shown in Figure 4.1(a)–(d), are initiated by uniformly distributing cells on a two–dimensional substrate. Over time, individual cells undergo migration and proliferation events, leading to the formation of a confluent monolayer [111, 118]; and, (ii) *Scratch assays*, as shown in Figure 4.1(e)–(h), are initiated in the same way as a cell proliferation assay, except that a wound, or scratch is made in the monolayer [49]. In a scratch assay, individual cells undergo motility and proliferation events with the net result being the spreading of cells into the vacant region [49]. A critical feature of collective cell migration assays is the role of crowding. At low cell density, individual cells are relatively free to move and proliferate because of the abundance of free space [14, 118]. In contrast, at high cell density, individual cells are strongly influenced by cell–to–cell crowding, which reduces their ability to move and proliferate [14, 118].

There are two different approaches to modelling collective cell migration assays. Firstly, a continuum reaction–diffusion equation can be applied to mimic certain features of the experiment [76]. Most previous continuum models represent cell migration with a diffusion-type mechanism, and a logistic source term to represent carrying capacity–limited proliferation [14, 22, 76, 98, 100, 102]. Secondly, a discrete random walk model can be used to mimic certain features of the experiment [19, 46]. Here, many previous studies represent cell migration using an unbiased exclusion process [72], which incorporates hard core exclusion to model cell–to–cell crowding [12, 15, 18, 59, 61, 109]. Cell proliferation is incorporated by allowing agents to place daughter agents on the lattice, with crowding effects incorporated by ensuring that potential proliferation events that would place a daughter agent on an occupied site are aborted [15, 109]. Discrete random walk models have an advantage over continuum models when it comes to comparing model predictions with experimental observations. Experimental data involving individual cells can be directly compared with the predictions of discrete models, whereas continuum models do not provide direct information about individual cells [34, 44].

Lattice–based random walk models of collective cell migration assays typically involve a particular mechanism to assess how crowding influences potential proliferation events [12, 15, 59, 109]. Mean field analysis of this traditional proliferation mechanism leads to the logistic source term in the partial differential equation (PDE) description of the model [15, 109, 111]. In terms of continuum models, carrying–capacity limited proliferation is often represented using the logistic equation, $dC \,/\, dt \;=\; \lambda \, C(1 - C)$ [14, 22, 76, 98, 100, 102], where λ is the proliferation rate, and the density has been scaled relative to the carrying capacity. However, there is some awareness that the logistic model does not always match experimental data. For example, West and colleagues [124] examine data describing *in vivo* growth, showing that their data is best described by a generalised

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

91

logistic model, $\mathrm{d}C/\mathrm{d}t = \lambda\,C^{3/4}(1 - C^{1/4})$. Similarly, Laird [66] examines *in vivo* tumour growth data, showing that the dynamics is better described by a Gompertz growth law than the classic logistic growth model. More recently, our previous analysis of a suite of scratch assays suggests that when a logistic–type reaction–diffusion equation is calibrated to match experimental data with a range of initial cell densities, there is no unique choice of λ for which the logistic model matches the entire data set [49]. One way of interpreting this result is that cells do not proliferate logistically. While several previous theoretical studies have analysed generalised logistic growth models, such as $\mathrm{d}C/\mathrm{d}t = \lambda C^{\alpha}(1 - C^{\beta})^{\gamma}$ for arbitrary positive constants α, β and γ [119], it is presently unknown how to implement this generalised proliferation mechanism in an exclusion process.

The aim of this work is to analyse a discrete model of two-dimensional collective cell migration assays. In all cases we consider the cell migration to be modelled as an unbiased nearest neighbour exclusion process where potential migration events occur with probability P_m per time step of duration τ. This motility mechanism is able to capture certain features of previous *in vitro* experimental data [53, 111]. The focus of our work is on the details of the proliferation mechanism. Potential proliferation events occur with probability P_p per time step of duration τ. In the traditional model, the location of the daughter agent is chosen by randomly selecting a nearest neighbour site. If the randomly selected target site is vacant, the proliferation event is successful, whereas if the randomly selected target site is occupied, the proliferation event is aborted [15, 109, 111]. Two extensions of the standard proliferation model are analysed: (i) we consider non–nearest neighbour proliferation mechanisms, whereby the crowdedness of any individual agent is influenced by a larger template on the lattice, and it is possible for the daughter agent to be placed on a non–nearest neighbour site [27, 67]; and, (ii) we adjust the way that we measure the local density, $\hat{C} \in [0, 1]$, and implement a new way of deciding whether to abort potential proliferation events due to crowding by using a more general *crowding function*, $f(C)$. We derive the continuum limit PDE description of the generalised discrete model, and apply both the discrete and continuum models to mimic a suite of cell proliferation and scratch assays. Our results illustrate several interesting features about the relationship between the discrete model and the continuum limit description.

Figure 4.1: Experimental motivation. (a)–(c) Images from a *cell proliferation assay*, shown at $t = 0, 24$ and 48 h, respectively [111]. The cell proliferation assay is initiated by uniformly distributing 25,000 3T3 fibroblast cells into the wells of a 24–well tissue culture plate [111]. The dimension of the field of view is 640 μm × 480 μm, and the spatial extent of the growing population extends well beyond the field of view [111]. Results in (d) shows the increase in cell density as a function of time obtained by counting the number of cells in the images in (a)–(c). (e)–(g) Images from a *scratch assay*, shown at $t = 0, 24$ and 48 h, respectively [49]. Experiments are initiated by uniformly distributing 16,000 PC–3 cells into the well of a 96-well tissue culture plate [49]. A scratch (dashed green in (e)) is made at $t = 0$ [49], and the subsequent healing of the wound is observed with time. The dimension of the field of view is 1900 μm × 1400 μm, and the spatial extent of the population extends well beyond the field of view [49]. The plot in (h) shows the spatial distribution of cell density obtained by discretising the images into strips of width 50 μm and counting the number of cells per strip. Dividing the number of cells in each strip by the area of the strip gives an estimate of the cell density. Three plots are given in (h) showing the cell density profile at $t = 0, 24$ and 48 hours, with the arrows indicating the direction of increasing time. All scale bars correspond to 300 μm.

Chapter 4. Stochastic simulation tools and continuum models for describing two–dimensional collective cell spreading with universal growth functions

93

4.2   Discrete mathematical models

We adopt the convention that dimensional variables are primed and non–dimensional variables are unprimed. A lattice–based random walk model will be used to describe the collective motion of a population of cells with an average cell diameter of Δ'. The lattice spacing is taken to be equal to the average cell diameter so that there are, at most, one agent per site. All simulations are non–dimensional in the sense that they are performed on a hexagonal lattice with unit lattice spacing, $\Delta = 1$. These non–dimensional simulations can be used to model any particular cell population by re–scaling with the dimensional cell diameter, Δ'. Each lattice site, indexed (i, j) where $i, j \in \mathbb{Z}^+$, has position

$$(x, y) = \begin{cases} \left(i\Delta, j\Delta\sqrt{3}/2\right) & \text{if } j \text{ is even,} \\ \left((i + 1/2)\Delta, j\Delta\sqrt{3}/2\right) & \text{if } j \text{ is odd,} \end{cases}$$

such that $1 \leq i \leq I$ and $1 \leq j \leq J$. In any single realisation of the model, the occupancy of site $\mathbf{s}$ is denoted $C_{\mathbf{s}}$, with $C_{\mathbf{s}} = 1$ if the site is occupied, and $C_{\mathbf{s}} = 0$ if vacant. Since site $\mathbf{s}$ is associated with a unique index (i, j), we will use $C_{\mathbf{s}}$ and $C_{i,j}$ interchangeably.

Traditional discrete model: If there are $N(t)$ agents at time t, then during the next time step of duration τ, $N(t)$ agents are selected independently at random, one at a time with replacement, and given the opportunity to move [109, 111]. The randomly selected agent attempts to move, with probability P_m, to one of the six nearest neighbour sites (Figure 4.2(a)), with the target site chosen randomly. Motility events are aborted if an agent attempts to move to an occupied site.

Once $N(t)$ potential motility events are attempted, another $N(t)$ agents are selected independently, at random, one at a time with replacement, and given the opportunity to proliferate with probability P_p. The location of the daughter agent is chosen, at random, from one of the six nearest neighbour sites [15, 109, 111]. If the selected site is occupied, the potential proliferation event is aborted. In contrast, if the selected site is vacant, a new daughter agent is placed on that site. After the $N(t)$ potential proliferation events have been attempted, $N(t + \tau)$ is updated [15, 109, 111]. One of the limitations of the traditional discrete model is that the continuum limit description leads to the traditional logistic source term [15, 109], however certain experimental observations indicate that logistic growth is not always appropriate [49, 124]. As such, we consider two extensions of the discrete proliferation mechanism.

Extension 1: We first generalise the traditional discrete proliferation mechanism so that crowding effects are felt over a larger spatial template, and daughter agents can be placed on non–nearest neighbour sites. The placement of daughter agents on non-nearest neighbour sites is consistent with previous *in vitro* [67] and *in vivo* [27] experimental observations. For example, previous experimental observations show that neural crest cells proliferate and place daughter cells approximately six cell diameters away from the original location of the mother cell [109]. To achieve this, we consider a proliferative agent

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

94

at site $\mathbf{s}$, and we use $\mathcal{N}_r\{\mathbf{s}\}$ to denote the set of neighbouring sites, where $r \geq 1$ is the number of concentric rings of sites surrounding $\mathbf{s}$. For example, when $r = 1$, $\mathcal{N}_1\{\mathbf{s}\}$ denotes the set of six 4.2(a). In contrast, when $r = 2$, $\mathcal{N}_2\{\mathbf{s}\}$ also includes the set of the next nearest–neighbouring sites, as demonstrated in Figure 4.2(e). More generally, the number of sites in $\mathcal{N}_r\{\mathbf{s}\}$ is $Z_{\mathcal{N}_r} = 3r(r + 1)$. To implement this extension we first choose a value of r. For any potential proliferation event, the target site for the placement of the daughter agent is chosen from $\mathcal{N}_r\{\mathbf{s}\}$. If a randomly chosen target site is vacant, a daughter agent is placed at that site. If a randomly chosen target site is occupied the potential proliferation event is aborted.

Extension 2: Instead of deciding to abort a potential proliferation event depending on the occupancy of a single randomly chosen site, we consider a more general approach by assuming that a proliferative agent at site $\mathbf{s}$ senses the occupancy of all sites within $\mathcal{N}_r\{\mathbf{s}\}$, and detects a measure of the average occupancy of those sites, $\hat{C}_{\mathbf{s}} = (1/Z_{\mathcal{N}_r}) \sum_{\mathbf{s}' \in \mathcal{N}_r\{\mathbf{s}\}} C_{\mathbf{s}'}$.
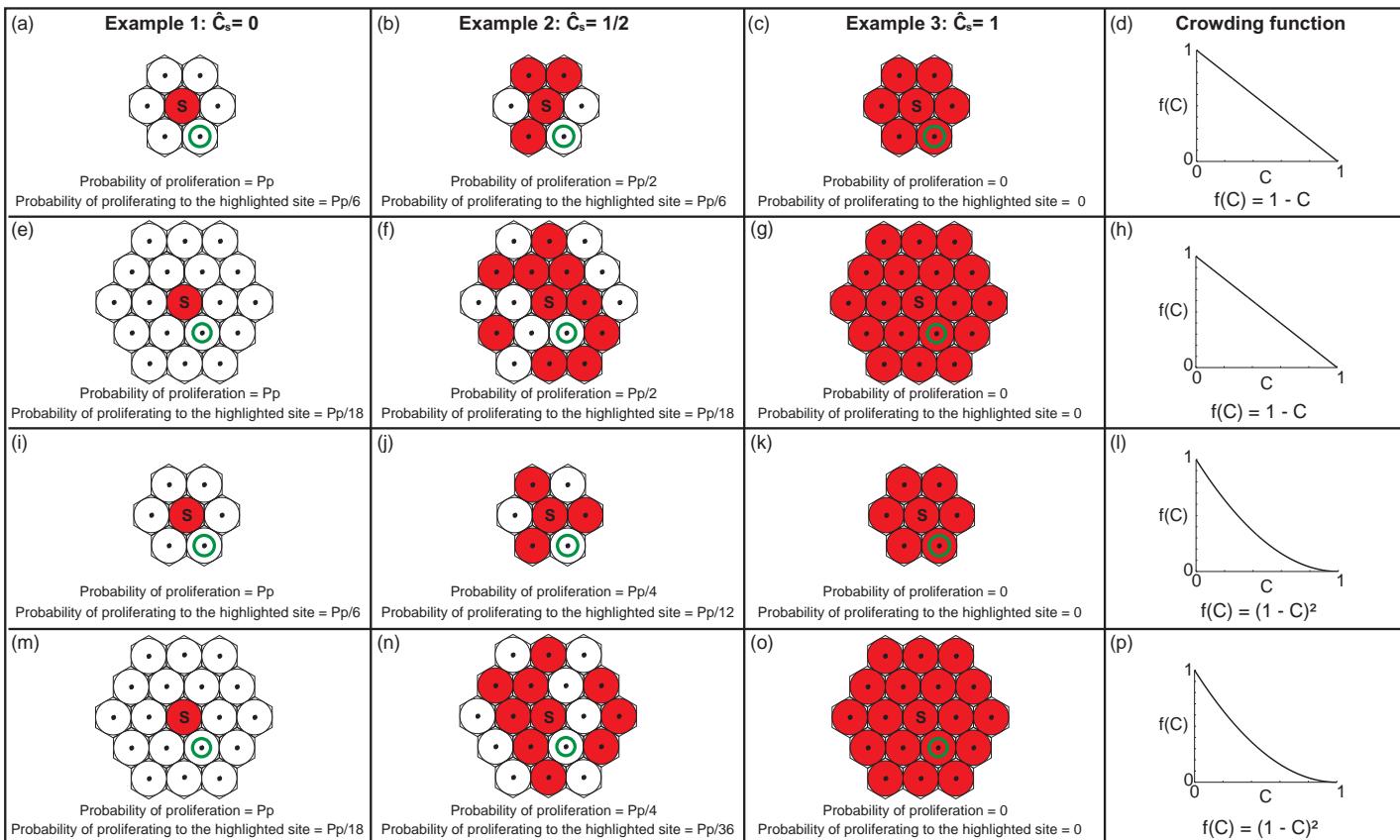
This means that $\hat{C}_{\mathbf{s}} \in [0, 1]$ is a measure of the crowdedness of the region surrounding $\mathbf{s}$. We anticipate that using $\hat{C}_{\mathbf{s}}$ to determine whether potential proliferation events are aborted is more realistic than the traditional model where the decision depends solely on the occupancy of a single, randomly chosen site. To use $\hat{C}_{\mathbf{s}}$ to determine whether a potential proliferation event succeeds, we introduce a *crowding function*, $f(C) \in [0, 1]$ with $f(0) = 1$ and $f(1) = 0$. The crowding function accounts for the density of agents within a group of sites to mimic the effect of cell-to-cell interactions on proliferation [128]. To incorporate crowding effects we sample a random number, $R \sim U(0, 1)$. If $R < f(\hat{C}_{\mathbf{s}})$, a daughter agent is placed at a randomly chosen vacant site in $\mathcal{N}_r\{\mathbf{s}\}$, whereas if $R > f(\hat{C}_{\mathbf{s}})$, the event is aborted. This extension can be applied to different sized templates by varying r.

Generalised discrete model: We now implement an algorithm that can be used to simulate both extensions 1 and 2. In this generalised model, a proliferative agent can place a daughter agent at any vacant target site in $\mathcal{N}_r\{\mathbf{s}\}$, and crowding effects are modeled by $f(C)$. Therefore, during a potential proliferation event, a randomly selected agent at site $\mathbf{s}$ attempts to proliferate with probability P_p per time step of duration τ. If the agent is to attempt to proliferate, crowding effects are incorporated by calculating $f(\hat{C}_{\mathbf{s}})$. If the potential proliferation event is to succeed, a daughter agent is placed at a randomly selected vacant site in $\mathcal{N}_r\{\mathbf{s}\}$.

Figure 4.2 shows four different schematic illustrations of the generalised proliferation mechanism on a hexagonal lattice. In each illustration, three examples are shown in which sites in $\mathcal{N}_r\{\mathbf{s}\}$ are either: (i) all vacant with $\hat{C}_{\mathbf{s}} = 0$ (Figure 4.2(a), (e), (i) and (m)); (ii) half occupied with $\hat{C}_{\mathbf{s}} = 0.5$ (Figure 4.2(b), (f), (j) and (n)); or, (iii) fully occupied with $\hat{C}_{\mathbf{s}} = 1$ (Figure 4.2(c), (g), (k) and (o)). The first row (Figure 4.2(a)–(d)) corresponds to the traditional model with $r = 1$ and $f(C) = 1 - C$. In this case, with $\hat{C}_{\mathbf{s}} = 0.5$, the probability that a potential proliferation event takes place is $P_p/2$, and the probability of a particular proliferation event producing a daughter agent at a particular

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

95

site is $P_p/6$ (Figure 4.2(b)). The second row (Figure 4.2(e)-(h)) corresponds to $r = 2$ and
$f(C) = 1 - C$. In this case, with $\hat{C}_\mathbf{s} = 0.5$, the probability that a potential proliferation
event takes place is $P_p/2$, and the probability of a particular proliferation event producing
a daughter agent at a particular site is $P_p/18$ (Figure 4.2(f)). Comparing the outcomes
in the first and second row of Figure 4.2 illustrates the first generalisation of the discrete
proliferation mechanism as we are simply applying the same proliferation mechanism,
with the same crowding function, over a larger template of lattice sites.

The schematic illustrations in the third (Figure 4.2(i)–(l)) and fourth (Figure 4.2(m)–(p))
rows of Figure 4.2 show how the outcomes in the first and second rows can be generalised
by choosing different $f(C)$. For example, with $f(C) = (1 - C)^2$, agents are less likely to
proliferate than when $f(C) = 1 - C$. With $r = 1$, $\hat{C}_\mathbf{s} = 0.5$ and $f(C) = (1 - C)^2$ (Figure
4.2(j)), the probability that a potential proliferation event takes place is $P_p/4$, and the
probability of a particular proliferation event producing a daughter agent at a particular
site is $P_p/12$, and this is very different to the traditional model (Figure 4.2(b)).

Figure 4.2: Schematic representation of the proliferation mechanisms considered in this work. Three examples are illustrated, in the first three columns, for $\hat{C} = 0, 0.5$ and 1, respectively. In each lattice fragment, the central site **s** is occupied (red), while some of the neighbouring sites are occupied (red) and others are vacant (white). In all cases we always consider the outcomes for a potential proliferation event of the agent at the central lattice site. The properties associated with the traditional proliferation mechanism are illustrated in (a)–(d). The properties associated with the first generalisation, where we consider the same crowding mechanism as the traditional model, but over a larger template of neighbouring lattice sites, is shown in (e)–(h). The properties associated with the second generalisation, where we consider the same nearest neighbour template as the traditional model, but we consider a different method of aborting potential proliferation events using $f(C)$, is shown in (i)–(l). The final row, (m)–(p), illustrates how the two generalisations can be combined. To highlight differences between the generalisations we report: $\hat{C}_{\mathbf{s}}$; the probability of a successful proliferation event taking place; and, the probability of a proliferation event successfully depositing a daughter at a particular site, highlighted with a green annulus, in (a)–(c), (e)–(g), (i)–(k) and (m)–(o).

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

97

4.3 Continuum description

While the individual–level details of the generalised discrete proliferation mechanism, highlighted in Figure 4.2, are very different to the traditional proliferation mechanism, it is not obvious how these differences affect the collective behaviour of a population of agents. To investigate this issue, we will derive the mean field continuum limit description of the discrete models, and then compare the performance of the continuum limit descriptions with averaged data from repeated discrete simulations.

Traditional model: We first derive the continuum limit description of the traditional model before considering the more general case. We average the occupancy of site $\mathbf{s}$ over many identically prepared realisations to obtain $\langle C_{\mathbf{s}} \rangle \in [0, 1]$ [109] and then develop an approximate discrete conservation statement describing the change in average occupancy of site $\mathbf{s}$ from time t to time $t + \tau$,

$$\delta\langle C_{\mathbf{s}} \rangle = \frac{P_m}{6} \left(1 - \langle C_{\mathbf{s}} \rangle\right) \sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} \langle C_{\mathbf{s}'} \rangle - \frac{P_m}{6} \langle C_{\mathbf{s}} \rangle \left(6 - \sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} \langle C_{\mathbf{s}'} \rangle\right)$$
$$+ \frac{P_p}{6} \left(1 - \langle C_{\mathbf{s}} \rangle\right) \sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} \langle C_{\mathbf{s}'} \rangle, \qquad (4.1)$$

where $\displaystyle\sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} \langle C_{\mathbf{s}'} \rangle$ is the sum of the average occupancy of sites in $\mathcal{N}_1\{\mathbf{s}\}$. The first and second terms on the right of Equation (4.1) represent the effects of migration into, and out of, site $\mathbf{s}$, respectively. The third term on the right of Equation (4.1) represents the effect of proliferation. To arrive at Equation (4.1), we make the usual mean field assumption that the occupancy of lattice sites is independent as we interpret the products of terms like $\langle C_{\mathbf{s}} \rangle$ and $\displaystyle\sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} \langle C_{\mathbf{s}'} \rangle$ as a net transition probability [15, 25, 109].

We expand each term in Equation (4.1) as a Taylor series about site $\mathbf{s}$, neglect terms of $\mathcal{O}(\Delta^3)$, and divide both sides of the resulting expression by τ. Identifying $\langle C_{\mathbf{s}} \rangle$ with a smooth function, $C(x, y, t)$, we consider the limit as $\Delta \to 0$ and $\tau \to 0$ jointly, with the ratio Δ^2/τ held constant, giving [109]

$$\frac{\partial C}{\partial t} = D\nabla^2 C + \lambda C \left(1 - C\right), \qquad (4.2)$$

where the diffusivity is $D = (P_m/4) \lim\limits_{\Delta \to 0, \tau \to 0} \left(\Delta^2/\tau\right)$, and the proliferation rate is $\lambda = \lim\limits_{\tau \to 0} \left(P_p/\tau\right)$. To obtain a well–defined continuum limit we require $P_p = \mathcal{O}(\tau)$ [19, 109]. Equation (4.2) confirms that the traditional proliferation mechanism is associated with the standard logistic source term. Furthermore, in one dimension, Equation (4.2) simplifies to the Fisher–Kolmogorov model [30].

Generalised model: The migration mechanism in the traditional and generalised models are equivalent, whereas the proliferation mechanism is different. The corresponding

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

98

approximate discrete conservation statement is

$$\delta\langle C_{\mathbf{s}}\rangle = \frac{P_m}{6}\left(1 - \langle C_{\mathbf{s}}\rangle\right)\sum_{\mathbf{s}'\in\mathcal{N}_1\{\mathbf{s}\}}\langle C_{\mathbf{s}'}\rangle - \frac{P_m}{6}\langle C_{\mathbf{s}}\rangle\left(6 - \sum_{\mathbf{s}'\in\mathcal{N}_1\{\mathbf{s}\}}\langle C_{\mathbf{s}'}\rangle\right)$$
$$+ \frac{P_p}{Z_{\mathcal{N}_r}}\left(1 - \langle C_{\mathbf{s}}\rangle\right)\sum_{\mathbf{s}'\in\mathcal{N}_1\{\mathbf{s}\}}\langle C_{\mathbf{s}'}\rangle\frac{f\left(\langle\hat{C}_{\mathbf{s}'}\rangle\right)}{1 - \langle\hat{C}_{\mathbf{s}'}\rangle}, \qquad (4.3)$$

where $\langle\hat{C}_{\mathbf{s}}\rangle = (1/Z_{\mathcal{N}_r})\sum_{\mathbf{s}'\in\mathcal{N}_r\{\mathbf{s}\}}\langle C_{\mathbf{s}'}\rangle$. The first two terms on the right of Equation (4.3) are identical to the corresponding terms in Equation (4.1). The third term on the right of Equation (4.3) represents the change in occupancy of site $\mathbf{s}$ due to proliferation. The factor $f(\langle\hat{C}_{\mathbf{s}'}\rangle)/(1 - \langle\hat{C}_{\mathbf{s}'}\rangle)$ is a measure of the crowding at site $\mathbf{s}'$, in terms of $f(C)$, relative to the probability that sites in the neighbourhood are vacant. Following the same procedure used previously for the traditional model we obtain,

$$\frac{\partial C}{\partial t} = D\nabla^2 C + \lambda C f\left(C\right). \qquad (4.4)$$

Details of the Taylor series expansions used to derive Equation (4.4) are given in the Supplementary material. Comparing Equations (4.2) and (4.4), the population–level impact of the change in the proliferation mechanism is to alter the per capita growth rate from the linearly decreasing function of density, $\lambda(1 - C)$, to the more general $\lambda f(C)$. For the remainder of this work we set $f(C) = C^{\alpha-1}(1 - C^{\beta})^{\gamma}$, where α, β and γ are positive constants [119], but many other choices of $f(C)$ are possible.

4.4   Results

Our main result, so far, is to describe how to incorporate a generalised proliferation mechanism into a discrete two-dimensional model of cell migration and cell proliferation with crowding effects, and to derive the mean field continuum limit description. However, at this stage, it is unclear how well the continuum model will predict averaged data from repeated stochastic simulations of the discrete model. To explore this issue we now apply the discrete and continuum models to mimic both a cell proliferation assay and a scratch assay (Figure 4.1). We systematically vary r and $f(C)$ to explore how these choices affect the performance of the continuum description.

A key parameter in the discrete model is P_p/P_m, which is the relative frequency of proliferation events to motility events for an isolated agent [109]. This ratio can be estimated from experimentally observable quantities including the doubling time $t'_d = (\tau' \log_e 2)/P_p$, the cell diffusivity D', and the average cell diameter, Δ'. For typical values: $t'_d = 24$ h [109]; $D' = 1000\ \mu\text{m}^2/\text{h}$ [111]; and, $\Delta' = 24\ \mu\text{m}$ [49], we have $P_p/P_m \approx 0.001$. Therefore, all simulations and analysis in the main document correspond to $\Delta = \tau = P_m = 1$ and $P_p = 0.001$. These non–dimensional simulations can be used to model a population of cells with an arbitrary dimensional cell diameter and an arbitrary doubling time by

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

99

re–scaling Δ and τ with appropriate choices of Δ' and τ', respectively [109]. To ensure
the conclusions drawn from these simulations are applicable to a wide range of cell lines,
we repeat all simulations and analysis for a higher proliferation rate, $\Delta = \tau = P_m = 1$
and $P_p = 0.05$ (Supplementary material).

Cell proliferation assay. We consider a suite of simulations of a cell proliferation assay
based on the geometry of the images in Figure 4.1(a)–(c). We use a lattice of size $I \times J$
to accommodate a typical population of cells ($\Delta' = 24$ μm [49]). Since the images in
Figure 4.1(a)–(c) show a fixed field of view that is much smaller than the spatial extent of
the real experiment, and the cells in the experiment are distributed uniformly, we apply
zero net flux boundary conditions along all boundaries [54]. Simulations are initiated by
randomly populating each lattice site with a constant probability of 5%, so that there
are, on average, no spatial gradients in agent density across the lattice. Snapshots from
the model, across a range of choices of $f(C)$, are given in Figure 4.3.

To mimic the way that cell proliferation assays are reported [118] (Figure 4.1(d)), we
calculate the time evolution of the total number of agents on the lattice, which, when
divided by the total number of lattice sites, gives the agent density per unit area [109]

$$\langle C \rangle = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} C_{i,j}.$$

We further average these results over many identically prepared simulations so that we
report relatively smooth data where stochastic fluctuations in the agent density are negli-
gible. To compare averaged simulation data with the solution of the continuum model, we
note that the absence of spatial gradients means that, on average, $\nabla^2 C = 0$. Therefore,
instead of dealing with a PDE for $C(x, y, t)$, Equation (4.4) simplifies to the ordinary
differential equation (ODE) for $C(t)$ [118]
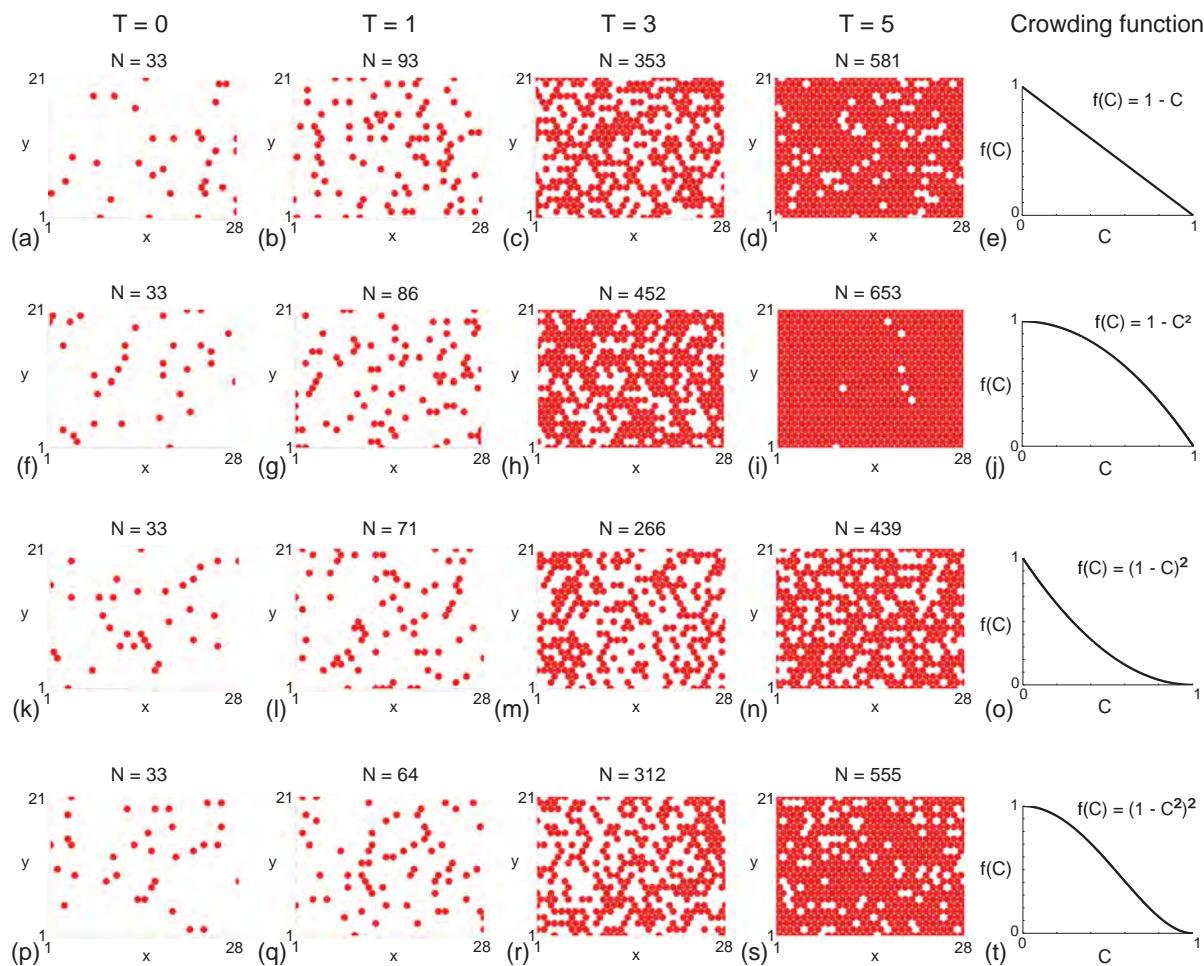
$$\frac{\mathrm{d}C}{\mathrm{d}T} = Cf(C), \tag{4.5}$$

where we have written $T = t\lambda$. This re–scaling of the time variable allows us to more easily
compare results in the main paper for a standard proliferation rate ($P_p/P_m = 0.001$) with
additional results for faster proliferation ($P_p/P_m = 0.05$) (Supplementary material). We
solve Equation (4.5) numerically using a backward Euler approximation with a constant
time step, δt, and Picard linearisation with convergence tolerance, ϵ.

Results in Figure 4.4(a)–(b) show that when $f(C)$ is linear, the discrete and continuum
density profiles are indistinguishable at this scale for $r = 1, 2, 3$ and 4. Results in Figure
4.4(c) quantify the discrepancy between the solution of the continuum model and averaged
discrete density data using $E = \langle C \rangle - C$, where $\langle C \rangle$ is the average density per unit area
from the discrete simulations and C is the solution of Equation (4.5). In summary, the
evolution of E (Figure 4.4(c)) shows that the error is extremely small, with no discernible
trends for the different choices of r. Additional results in Figures 4.4(d)–(l) show similar

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

100

comparisons for a range of nonlinear $f(C)$ and several choices of r. The averaged discrete data and the solution of the corresponding continuum model show that, broadly speaking, the continuum model provides a good prediction of the averaged discrete results but there are some differences for different choices of r when $f(C)$ is nonlinear. When we compare the evolution of the density data between Figures 4.4(b), (e), (h) and (k), we see that the choice of $f(C)$ impacts the evolution of the density profile. Similar comparisons to those in Figure 4.4 are made in the Supplementary material for a higher initial condition. In addition to these two different choices of initial condition, we also performed comparisons for a range of other initial conditions (not shown) and these results confirm that the trends we observe are relevant regardless of the initial condition.

Although the match between the average discrete data and the solution of the corresponding continuum model in Figure 4.4 is very good, there are some trends that are not obvious without making these comparisons explicit. For example, results in Figure 4.4(e)–(f), (h)–(i) and (k)–(l) indicate that the performance of the continuum model is slightly poorer when $f(C)$ is nonlinear compared to the results in Figure 4.4(b)–(c) where $f(C)$ is linear. However, for all choices of $f(C)$, the performance of the continuum model improves as r increases. For example, all results with $r = 4$ lead to an excellent match regardless of $f(C)$. Therefore, these results indicate that estimating r from experimental time lapse images, such as those reported by Druckenbrod and Epstein [27], will be important if we need to decide whether the continuum approximation is sufficient and r is sufficiently large, or whether we need to use more computationally demanding repeated simulations of the discrete model, when r is sufficiently small.

Figure 4.3: Snapshots of simulations for a suite of cell proliferation assays. In each row the distributions of agents at time $\lambda t = T = 0, 1, 3, 5$, as indicated, are shown along with the corresponding $f(C)$. Each simulation is initiated by randomly populating a lattice of size $I = 28$ and $J = 24$, so that each site is occupied with probability 5%. All simulations correspond to $\Delta = \tau = P_m = 1$, $P_p = 0.001$ and $r = 4$.

Scratch assay. We also consider a suite of simulations of a scratch assay based on the
geometry of the images in Figure 4.1(e)–(g). Since these images show a fixed field of view
that is much smaller than the spatial extent of the real experiment [49], we apply zero net
flux boundary conditions along all boundaries of the lattice [54]. We model the scratch
assays on a lattice of size $I \times J$ that is chosen to accommodate a typical population
of cells ($\Delta' = 24$ μm [49]). To model the initial condition, we randomly populate all
lattice sites with an equal probability of 30%. This initial density is smaller than the
maximum carrying capacity density, and this is consistent with previous experimental
procedures [49]. While we present results for an initial density of 30%, our stochastic
simulation tools and continuum limit description are general, and can deal with any
initial density. To simulate the scratch, we remove all agents from a vertical region, with
a width of 23 cell diameters (Figure 4.1(d)). Snapshots from the discrete model, for a
range of $f(C)$ are shown in Figure 4.5.

Since the initial condition is uniform in the vertical direction, we average the agent pop-
ulation density along each vertical column of lattice sites to obtain
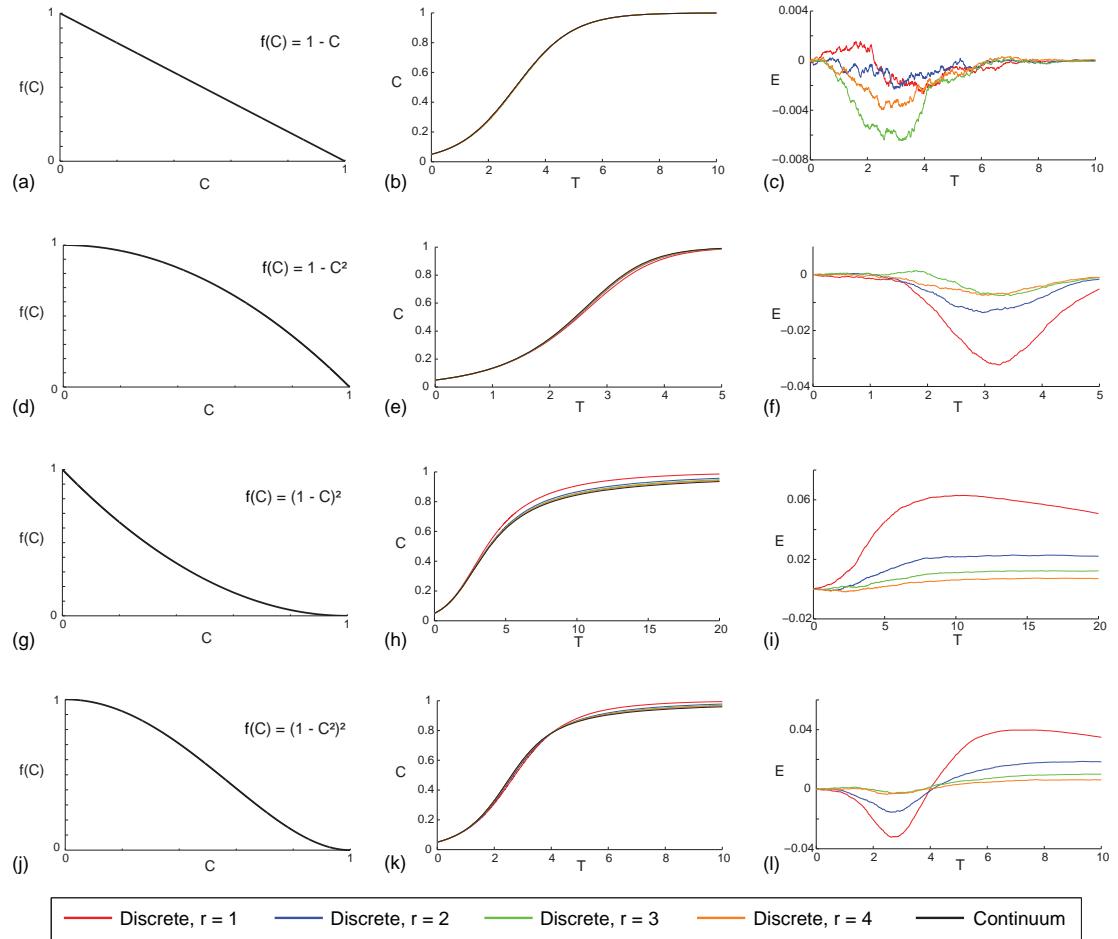
$$\langle C_i \rangle = \frac{1}{J} \sum_{j=1}^{J} C_{i,j},$$

which is then further averaged over many identically prepared discrete simulations to
reduce fluctuations. This procedure allows us to plot the time evolution of the average
agent density as a function of the horizontal coordinate, as shown in Figure 4.1(h) [49,54].
Typical results from the discrete model are shown in Figure 4.6(a)–(c), with the standard
linear $f(C)$, for $r = 1$ and $r = 4$, respectively. As time increases, we see the effects of
combined agent motility and agent proliferation as the agent density profiles spreads into
the initially-vacant region. The effects of proliferation can also be observed as the density
profile increases with time towards confluence.

To explore how well the continuum description matches this vertically–averaged discrete
density data, we note that since the initial condition is independent of the vertical location
(Figure 4.1(d)), we can average Equation (4.4) in the vertical direction [49,54] to give,
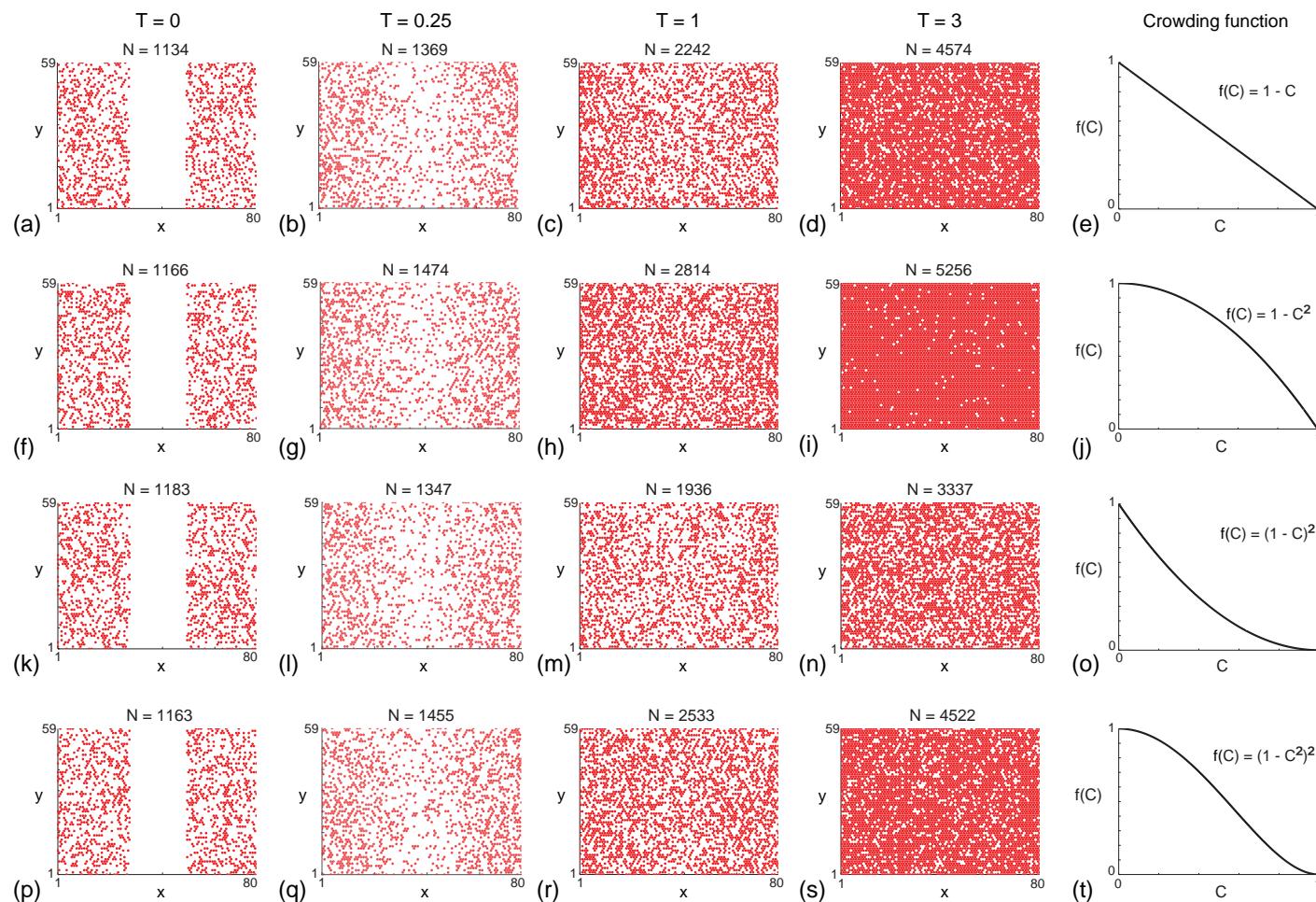
$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} + \lambda C f(C). \tag{4.6}$$

To solve Equation (4.6), we apply no flux boundary condition at both boundaries, and the
initial condition is given by $C(x,0) \equiv 0$ within the initially–vacant region, and $C(x,0) \equiv
0.30$ outside of this region. We solve Equation (4.6) numerically using a central difference
approximation with uniform spacing δx, and the temporal derivative is approximated
using a backward Euler method with a uniform time step, δt. The resulting system of
nonlinear algebraic equations is solved using Picard iteration with convergence tolerance,
ϵ. The numerical solution of Equation (4.6) is superimposed on the averaged discrete
density profiles in Figure 4.6(b)-(c) for $f(C) = 1 - C$, with $r = 1$ and $r = 4$, respectively.

Chapter 4. Stochastic simulation tools and continuum models for describing two–dimensional collective cell spreading with universal growth functions

103

Figure 4.4: Comparison of averaged simulation data and the solution of the corresponding continuum model for a cell proliferation assay with: $f(C) = 1 - C$, as shown in (a)–(c); $f(C) = 1 - C^2$, as shown in (d)–(f); $f(C) = (1 - C)^2$, as shown in (g)–(i); and $f(C) = (1 - C^2)^2$, as shown in (j)–(l). Results in (b), (e), (h) and (k) compare averaged simulation data and the solution of the corresponding continuum model for a range of $1 \leq r \leq 4$ where the initial condition corresponds to 5% of sites being randomly occupied. All simulations are performed on a lattice with $I = 28$ and $J = 24$, and results are averaged across 300 identically prepared realisations of the discrete model. Profiles in (c), (f), (i) and (l) quantify the discrepancy between the solution of the continuum model and the average simulation data. All simulation results correspond to $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, and the numerical solution of the continuum model is obtained with $\delta t = 1 \times 10^{-3}$ and $\epsilon = 1 \times 10^{-5}$. In all cases $T = \lambda t$.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

104

As with the cell proliferation assay results in Figure 4.4, the quality of the continuum–
discrete match for the scratch assay with $f(C) = 1 - C$ is excellent for all r considered.
Data in Figure 4.6(d)–(l) show similar comparisons for simulations of scratch assays for
a range of nonlinear $f(C)$. Comparing the evolution of $C(x, T)$ in Figure 4.6 shows how
the choice of $f(C)$ influences the evolution of the scratch assay. Again, as for the cell
proliferation assays, overall we observe a good quality of match between the solution of the
continuum model and the averaged agent density profiles across all choices of $f(C)$ and r
for the scratch assays. However, for nonlinear $f(C)$, we observe some small discrepancies,
and these discrepancies are most pronounced when $r = 1$. In contrast, the quality of
the continuum–discrete match is excellent for larger r across all choices of $f(C)$ that we
consider. Further comparisons of the performance of the continuum PDE description for
intermediate values of $r = 2$ and $r = 3$ are given in the Supplementary material.
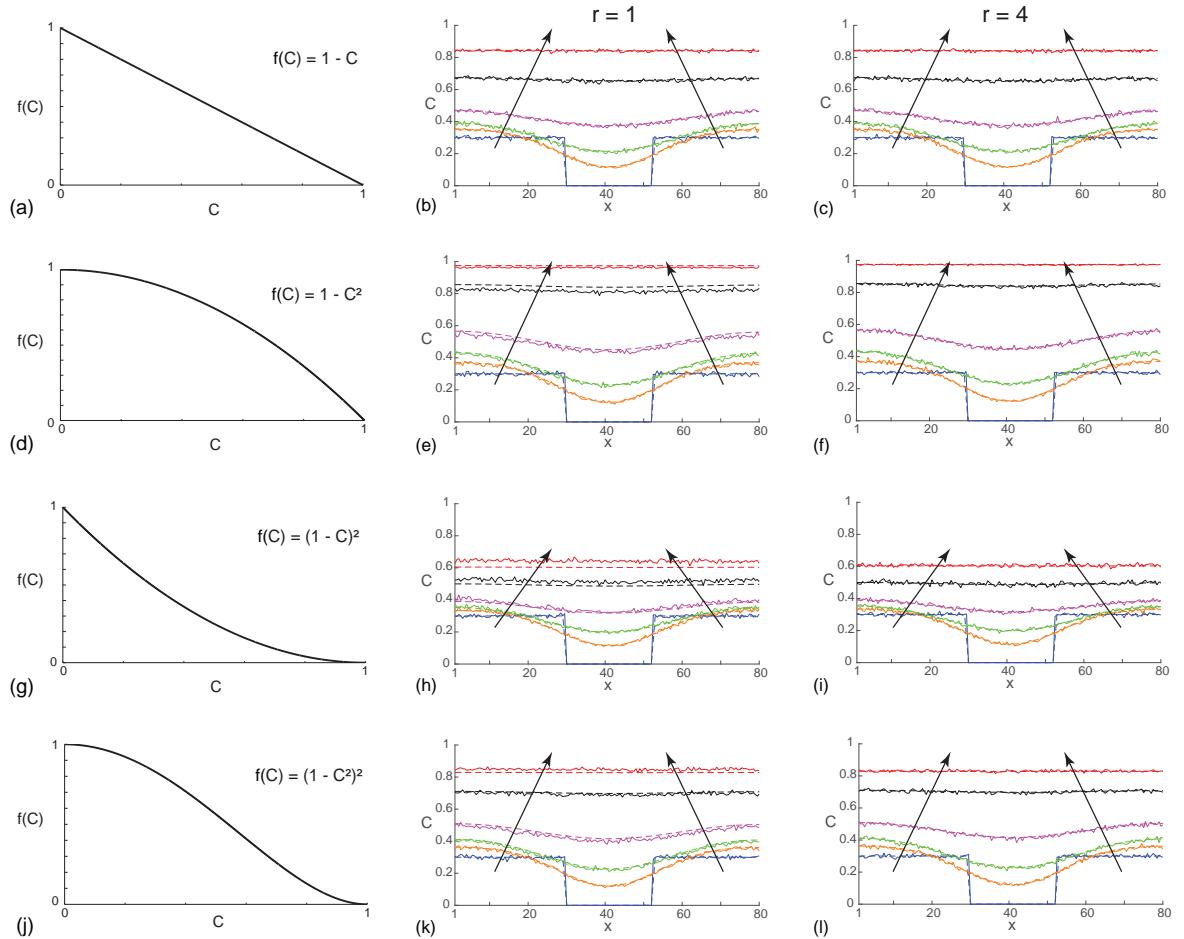
Figure 4.5: Snapshots of simulations for a suite of scratch assays. In each row the distributions of agents at time $\lambda t = T = 0, 0.25, 1, 3$ are shown along with the corresponding $f(C)$. Each simulation is initiated by randomly populating a lattice, corresponding to lattice of size $I = 80$ and $J = 68$, so that each site is occupied with probability 30%. A scratch of 23 lattice sites wide is made at $T = 0$. All simulations correspond to $\Delta = \tau = P_m = 1$, $P_p = 0.001$ and $r = 4$.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

106

4.5   Discussion

Most continuum [14, 22, 76, 98, 100] and discrete [15, 109, 111] models of collective cell spreading are associated with a logistic growth source term to model cell proliferation. However, certain *in vivo* [66, 124] and *in vitro* [49] evidence suggests that cells do not always proliferate logistically. Motivated by these observations, we extend the traditional exclusion process based discrete model of cell proliferation since the continuum limit description of this traditional model always leads to logistic growth. Our new, generalised discrete model encompasses two extensions. The first extension involves choosing a variably sized proliferation template so that the target site is chosen from a set of sites contained within $r \geq 1$ concentric rings about the mother agent. The second extension involves a measure of the crowdedness of the proliferative agent, $\hat{C}_{\mathbf{s}} \in [0, 1]$. A *crowding function*, $f(C)$, which incorporates information from a group of neighbouring sites is used to quantify the influence of crowding. Analysing the mean field continuum limit of the generalised model shows that the usual logistic source term, $\lambda C(1 - C)$, is generalised to $\lambda C f(C)$. There are several interesting consequences of this generalised mean field PDE, namely: (i) the traditional logistic source term corresponds to linear $f(C)$; (ii) the size of the template, r, does not appear in the continuum limit description; and, (iii) without making explicit comparisons, it is unclear how different choices of $f(C)$ and r affect the accuracy of the continuum limit description.

To provide insight into how different choices of $f(C)$ and r affect the accuracy of the continuum limit description, we generate averaged discrete data from the generalised random walk model for both a cell proliferation and a scratch assay for a typical cell line with $P_p/P_m = 0.001$ [111]. Averaged simulation data are generated for a range of choices of $f(C)$ and r and we find that, overall, the continuum description provides a good prediction of the average behaviour of the stochastic simulations. While there is a modest discrepancy in the continuum-discrete match for nonlinear $f(C)$, we find that the quality of the match improves as r increases. Therefore, to make a distinction between the need for using repeated stochastic simulations or simply working with the continuum description, we suggest that experimental time lapse images [27] ought to be used to estimate the size of the group of cells that a proliferating cell interacts with. This approach would provide an estimate of r. To ensure that our conclusions are broadly applicable across a wide range of cell lines, all continuum-discrete comparisons are repeated for a cell line with a particularly fast proliferation rate $P_p/P_m = 0.05$ (Supplementary material), and we find that the same conclusions apply.

There are two features of this study that could warrant further investigation. First, since the focus of the work is to investigate the role of the proliferation mechanism, all simulations and analysis invoke the most fundamental unbiased nearest neighbour exclusion motility mechanism. This mechanism provides a good approximation of the collective motility of mesenchymal cell lines that are largely unaffected by cell-to-cell

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

107

Figure 4.6: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with: $f(C) = 1 - C$, as shown in (a)–(c); $f(C) = 1 - C^2$, as shown in (d)–(f); $f(C) = (1-C)^2$, as shown in (g)–(i); and $f(C) = (1-C^2)^2$, as shown in (j)–(l). Results in (b), (e), (h) and (k) compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines) for $r = 1$. In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1, 2, 3$, and the direction of increasing t is shown with the arrows. Results in (c), (f), (i) and (l) show an equivalent comparison except here we have $r = 4$. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, on a lattice of size $I = 80$ and $J = 68$. The numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

108

adhesion [14, 111]. However, if dealing with an epithelial cell line, it would be more reasonable to invoke a motility mechanism that incorporates cell–to–cell adhesion [25, 115]. Under these conditions it would be interesting to extend the present analysis to investigate the performance of the continuum limit description of the generalised proliferation mechanism with an adhesive motility mechanism [25, 115]. Second, all discrete simulations in this work are lattice–based. While many previous studies have used lattice–based models to successfully mimic and predict a range of two–dimensional *in vitro* assays [14, 18, 54, 115], all lattice–based models make certain implicit assumptions. For example, all lattice–based exclusion process models effectively assume that agents are a fixed size [12, 15, 18, 59, 61, 109], which is clearly an approximation because it is well known that cell proliferation involves gradual changes in cell volume. An alternative way to simulate collective cell migration experiments would be to use a lattice–free framework [52, 89]. One of the advantages of using a lattice–free framework is that the model can be adapted to allow for dynamic cell size changes during a proliferation event. However, the main limitation of using a lattice–free method to mimic a real experiment is that the computation time is proportional to N^2, where N is the number of agents in the simulation. This can be prohibitive if we wish to mimic a real *in vitro* experiment with many tens of thousands, or hundreds of thousands cells [115]. In contrast, lattice–based methods are more convenient since the computation time is proportional to N.

Acknowledgments

4.6   Supplementary material

4.6.1   Taylor series expansions

In this section we demonstrate how to derive Equation (refPDE2) with $r = 1$ and arbitrary $f(C)$. Extending the derivation to deal with $r > 1$ is straightforward. To begin, we note that site $\mathbf{s}$, with position (x, y), has six nearest neighbouring sites: site $\mathbf{s}'_1$ with position $(x - \Delta, y)$; site $\mathbf{s}'_2$ with position $(x + \Delta, y)$; site $\mathbf{s}'_3$ with position $(x - \Delta/2, y + \Delta\sqrt{3}/2)$; site $\mathbf{s}'_4$ with position $(x + \Delta/2, y + \Delta\sqrt{3}/2)$; site $\mathbf{s}'_5$ with position $(x - \Delta/2, y - \Delta\sqrt{3}/2)$; and, site $\mathbf{s}'_6$ with position $(x + \Delta/2, y - \Delta\sqrt{3}/2)$. It is useful to first write down expressions for the average density of these six nearest neighbouring sites expanded in a Taylor series

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

109

about (x, y),

$$\langle C_{\mathbf{s}'_1} \rangle = \langle C_{\mathbf{s}} \rangle - \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial x} \Delta + \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x^2} \frac{\Delta^2}{2} + \mathcal{O}(\Delta^3), \tag{4.7}$$

$$\langle C_{\mathbf{s}'_2} \rangle = \langle C_{\mathbf{s}} \rangle + \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial x} \Delta + \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x^2} \frac{\Delta^2}{2} + \mathcal{O}(\Delta^3), \tag{4.8}$$

$$\langle C_{\mathbf{s}'_3} \rangle = \langle C_{\mathbf{s}} \rangle - \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial x} \frac{\Delta}{2} + \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial y} \frac{\sqrt{3}\Delta}{2} + \left[\frac{1}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x^2} + \frac{3}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial y^2} - \frac{\sqrt{3}}{2} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x \partial y} \right] \frac{\Delta^2}{2} + \mathcal{O}(\Delta^3), \tag{4.9}$$

$$\langle C_{\mathbf{s}'_4} \rangle = \langle C_{\mathbf{s}} \rangle + \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial x} \frac{\Delta}{2} + \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial y} \frac{\sqrt{3}\Delta}{2} + \left[\frac{1}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x^2} + \frac{3}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial y^2} + \frac{\sqrt{3}}{2} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x \partial y} \right] \frac{\Delta^2}{2} + \mathcal{O}(\Delta^3), \tag{4.10}$$

$$\langle C_{\mathbf{s}'_5} \rangle = \langle C_{\mathbf{s}} \rangle - \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial x} \frac{\Delta}{2} - \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial y} \frac{\sqrt{3}\Delta}{2} + \left[\frac{1}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x^2} + \frac{3}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial y^2} + \frac{\sqrt{3}}{2} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x \partial y} \right] \frac{\Delta^2}{2} + \mathcal{O}(\Delta^3), \tag{4.11}$$

$$\langle C_{\mathbf{s}'_6} \rangle = \langle C_{\mathbf{s}} \rangle + \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial x} \frac{\Delta}{2} - \frac{\partial \langle C_{\mathbf{s}} \rangle}{\partial y} \frac{\sqrt{3}\Delta}{2} + \left[\frac{1}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x^2} + \frac{3}{4} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial y^2} - \frac{\sqrt{3}}{2} \frac{\partial^2 \langle C_{\mathbf{s}} \rangle}{\partial x \partial y} \right] \frac{\Delta^2}{2} + \mathcal{O}(\Delta^3), \tag{4.12}$$

where we have truncated terms of $\mathcal{O}(\Delta^3)$. From this point on we drop the angle bracket
notation. There are two terms on the right of Equation (4.3) that are associated with
agent motility, and one term on the right of Equation (4.3) that is associated with agent
proliferation. We will deal with these two different types of terms separately.

For agent migration, we need to deal with terms like $\displaystyle\sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} C_{\mathbf{s}'}$, which can be obtained
by summing Equations (4.7)-(4.12) to give

$$\sum_{\mathbf{s}' \in \mathcal{N}_1\{\mathbf{s}\}} C_{\mathbf{s}'} = 6C_{\mathbf{s}} + \left(\frac{\partial^2 C_{\mathbf{s}}}{\partial x^2} + \frac{\partial^2 C_{\mathbf{s}}}{\partial y^2} \right) \frac{3\Delta^2}{2} + \mathcal{O}(\Delta^3).$$

For agent proliferation, we need to deal with the terms: $f(\hat{C}_{\mathbf{s}'_1}), f(\hat{C}_{\mathbf{s}'_2}), f(\hat{C}_{\mathbf{s}'_3}), f(\hat{C}_{\mathbf{s}'_4}), f(\hat{C}_{\mathbf{s}'_5})$
and $f(\hat{C}_{\mathbf{s}'_6})$. Here, we will show how to deal with $f(\hat{C}_{\mathbf{s}'_1})$ and we note that the other terms
are a straightforward extension of these results. First we expand the local density in a
Taylor series about site $\mathbf{s}$

$$\begin{aligned}
\hat{C}_{\mathbf{s}'_1} &= \frac{1}{6} \sum_{\mathbf{s}'' \in \mathcal{N}_1\{\mathbf{s}'\}} C_{\mathbf{s}''} \\
&= C_{\mathbf{s}'_1} + \left(\frac{\partial^2 C_{\mathbf{s}'_1}}{\partial x^2} + \frac{\partial^2 C_{\mathbf{s}'_1}}{\partial y^2} \right) \frac{\Delta^2}{4} + \mathcal{O}(\Delta^3), \\
&= C_{\mathbf{s}} - \frac{\partial C_{\mathbf{s}}}{\partial x} \Delta + \left(3\frac{\partial^2 C_{\mathbf{s}}}{\partial x^2} + \frac{\partial^2 C_{\mathbf{s}}}{\partial y^2} \right) \frac{\Delta^2}{4} + \mathcal{O}(\Delta^3). \tag{4.13}
\end{aligned}$$

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

110

We re-write Equation (4.13) as $\hat{C}_{\mathbf{s}'_1} = C_{\mathbf{s}} + \bar{C}$, where $\bar{C} = \mathcal{O}(\Delta)$. Therefore, the crowding function at site $\mathbf{s}'_1$ can be expanded in the following way,

$$
\begin{aligned}
f(\hat{C}_{\mathbf{s}'_1}) &= f(C_{\mathbf{s}} + \bar{C}), \\
&= f(C_{\mathbf{s}}) + \frac{\mathrm{d}f(C_{\mathbf{s}})}{\mathrm{d}C}\bar{C} + \frac{\mathrm{d}^2 f(C_{\mathbf{s}})}{\mathrm{d}C^2}\frac{\bar{C}^2}{2} + \mathcal{O}(\Delta^3).
\end{aligned}
\tag{4.14}
$$

Using Equations (4.13)-(4.14) we obtain a series expansion for $f(\hat{C}_{\mathbf{s}'_1})$. Repeating this procedure for $f(\hat{C}_{\mathbf{s}'_1}), f(\hat{C}_{\mathbf{s}'_2}), f(\hat{C}_{\mathbf{s}'_3}), f(\hat{C}_{\mathbf{s}'_4}), f(\hat{C}_{\mathbf{s}'_5})$ and $f(\hat{C}_{\mathbf{s}'_6})$, we expand all terms in Equation (4.3) that are associated with proliferation in a similar way. With all terms in Equation (4.3) expanded about (x, y), we obtain

$$
\delta C_{\mathbf{s}} = P_m \left(\frac{\partial^2 C_{\mathbf{s}}}{\partial x^2} + \frac{\partial^2 C_{\mathbf{s}}}{\partial y^2} \right) \frac{\Delta^2}{4} + P_p C_{\mathbf{s}} f(C_{\mathbf{s}}).
\tag{4.15}
$$

Following the same procedure outlined in the main manuscript, Equation (4.15) leads to Equation (4.4).

4.6.2 Additional results for the cell proliferation assays with the standard proliferation rate

All discrete-continuum comparisons in Figure 4.4 correspond to a cell proliferation assay with 5% of lattice sites initially occupied. Here, in Figure S1, we present an otherwise identical set of continuum-discrete comparisons for a different initial condition where all sites are initially occupied with probability 20%. In comparing the continuum-discrete comparison in Figure 4.4 with the results here in Figure 4.7 we see that the same trends regardless of the initial condition.
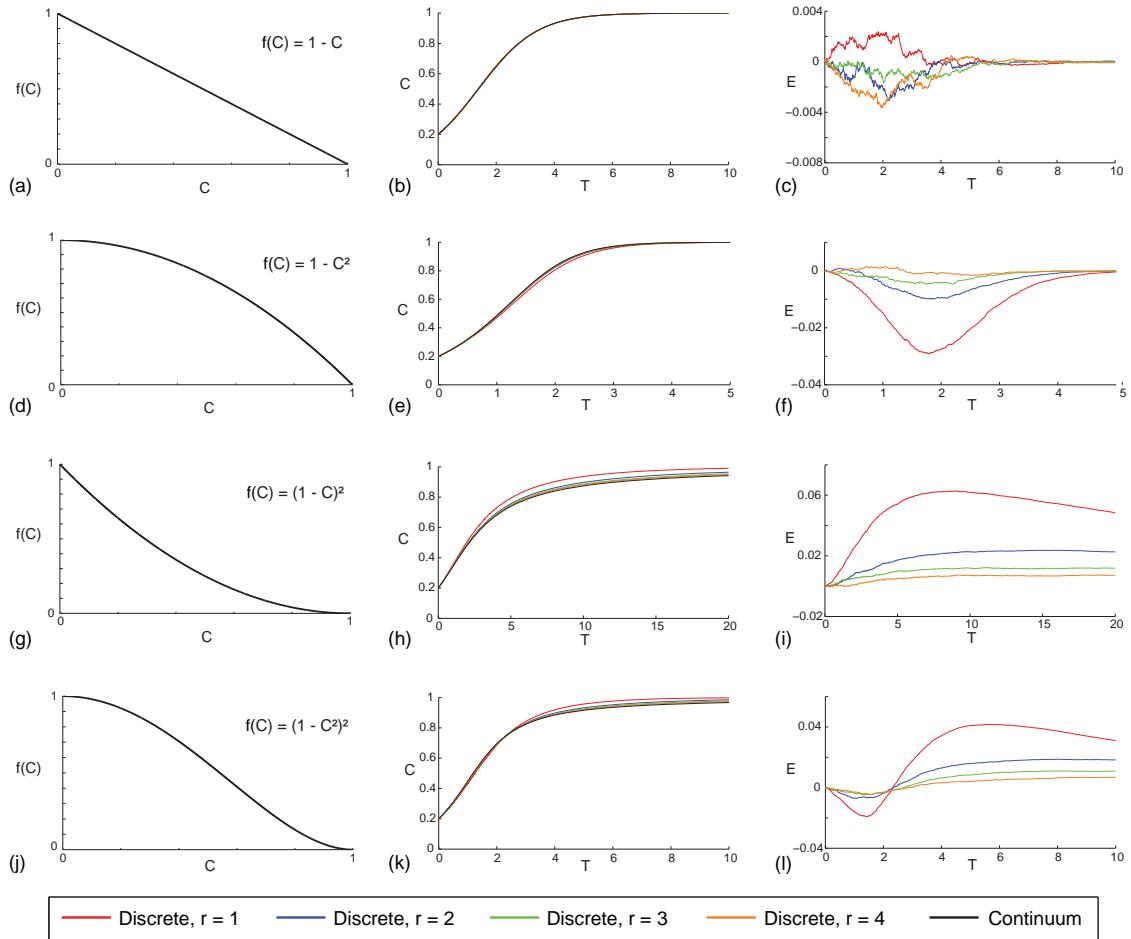
4.6.3 Additional results for the scratch assays with the standard proliferation rate

All discrete-continuum comparisons in Figure 4.6 compare the performance of the continuum limit description for the scratch assays for four choices of $f(C)$, and two choices of r ($r = 1$ and $r = 4$) with $P_p/P_m = 0.001$. Additional comparisons, given here in Figures 4.8–4.11, show further results for intermediate values of r ($r = 2, 3$) together with $r = 1$ and $r = 4$. All other parameters are the same as in Figure 4.6.
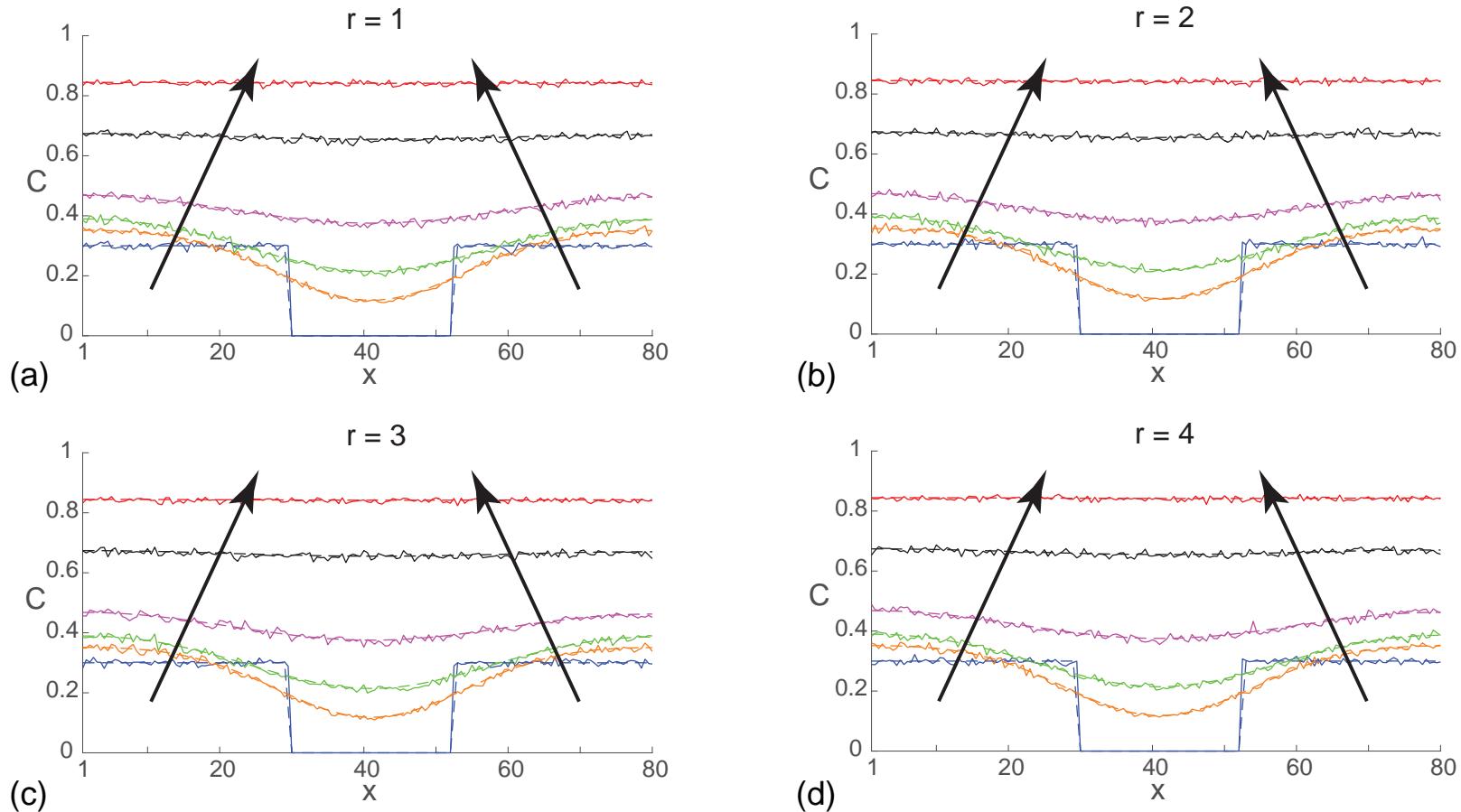
4.6.4 Additional results for higher proliferation rate

All discrete-continuum comparisons presented in the main paper are for a standard cell doubling time, and correspond to non-dimensional discrete simulations with $P_p/P_m = 0.001$. To be sure that the conclusions drawn from this work are consistent across a wide range of cell lines, we also present additional results for both the proliferation assay (Figure 4.12–4.14) and the scratch assay (Figure 4.15–4.19) for a cell line with a much faster

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

111
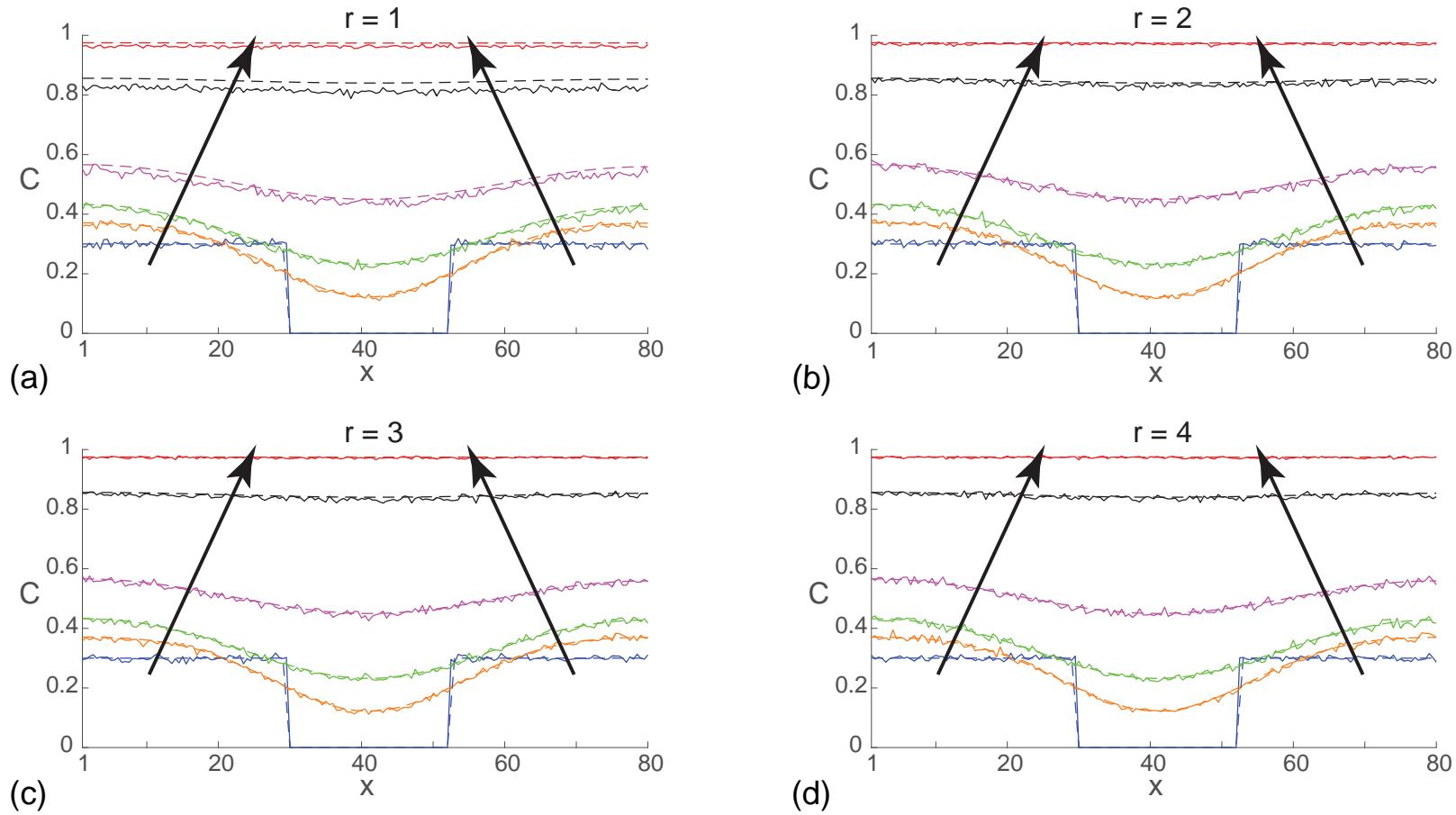
proliferation rate, $P_p/P_m = 0.05$. In summary, comparing the quality of the continuum-discrete match in Figure 4.4 with the additional results in Figures 4.12–4.14 confirms that the same trends and results discussed in the main document are also relevant for the faster proliferation rate in the cell proliferation assay. Similarly, comparing the quality of the continuum-discrete match in 4.6 with the additional results in Figures 4.15–4.19 confirms that the similar trends and results discussed in the main document are also relevant for the faster proliferation rate in the scratch assay.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

112
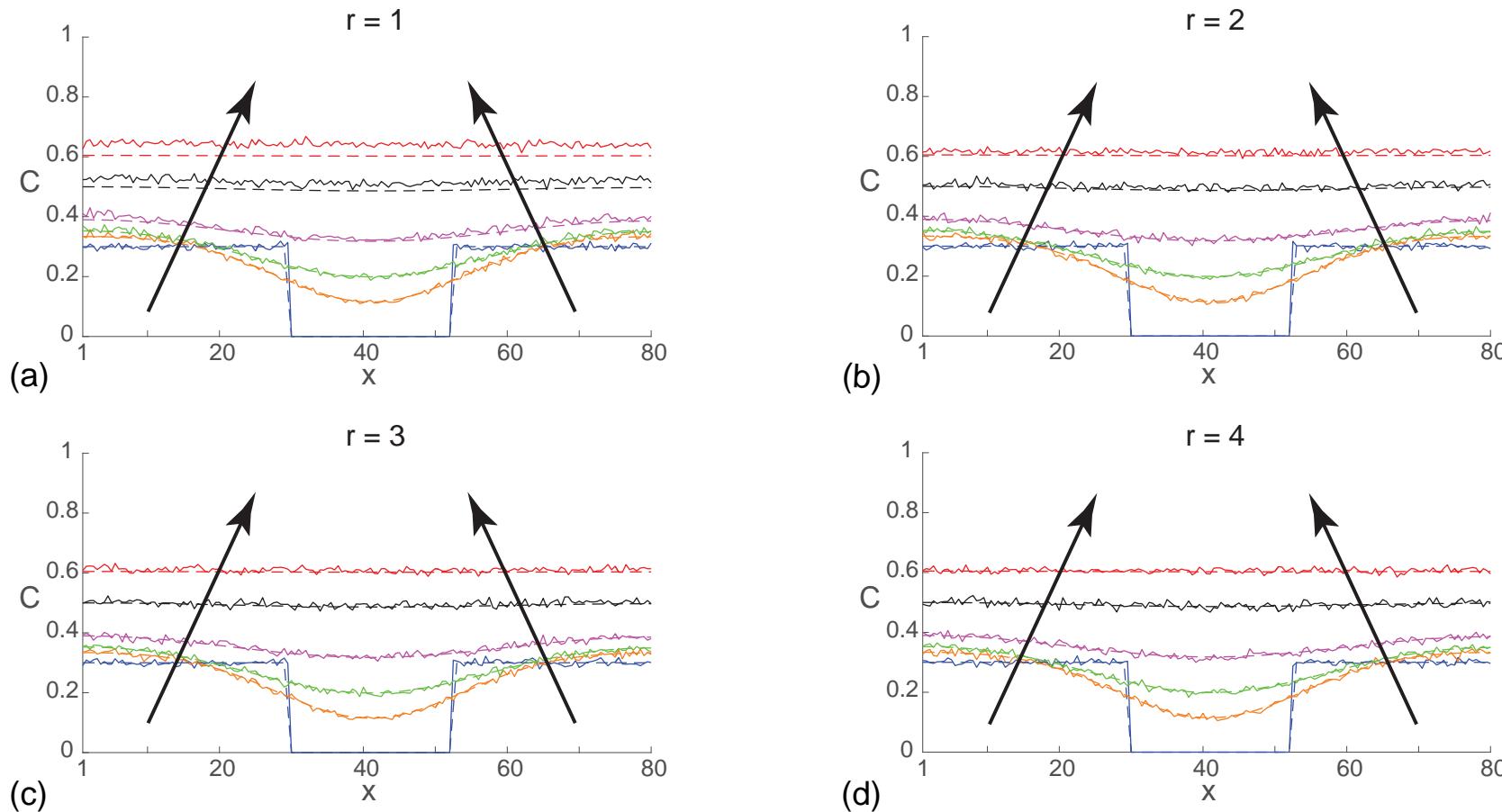
Figure 4.7: Comparison of averaged simulation data and the solution of the corresponding continuum model for a cell proliferation assay with: $f(C) = 1 - C$, as shown in (a)-(c); $f(C) = 1 - C^2$, as shown in (d)-(f); $f(C) = (1 - C)^2$, as shown in (g)-(i); and $f(C) = (1 - C^2)^2$, as shown in (j)-(l). Results in (b), (e), (h) and (k) compare averaged simulation data and the solution of the corresponding continuum model for a range of $1 \leq r \leq 4$ where the initial condition corresponds to 20% of sites being randomly occupied. All simulations are performed on a lattice with $I = 28$ and $J = 24$, and results are averaged across 300 identically prepared realisations of the discrete model. Profiles in (c), (f), (i) and (l) quantify the discrepancy between the solution of the continuum model and the average simulation data. All simulation results correspond to $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, and the numerical solution of the continuum model is obtained with $\delta t = 1 \times 10^{-3}$ and $\epsilon = 1 \times 10^{-5}$. In all cases $T = \lambda t$.

Figure 4.8: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = 1 - C$. Results in (a)-(d) correspond to $r = 1, 2, 3$ and 4, respectively. Results compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines). In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1, 2, 3$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, on a lattice of size $I = 80$ and $J = 68$. The numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Figure 4.9: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = 1 - C^2$. Results in (a)-(d) correspond to $r = 1, 2, 3$ and 4, respectively. Results compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines). In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1, 2, 3$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, on a lattice of size $I = 80$ and $J = 68$. The numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Figure 4.10: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = (1-C)^2$. Results in (a)-(d) correspond to $r = 1, 2, 3$ and 4, respectively. Results compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines). In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1, 2, 3$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, on a lattice of size $I = 80$ and $J = 68$. The numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Figure 4.11: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = (1 - C^2)^2$. Results in (a)-(d) correspond to $r = 1, 2, 3$ and 4, respectively. Results compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines). In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1, 2, 3$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.001$, on a lattice of size $I = 80$ and $J = 68$. The numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Figure 4.12: Snapshots of simulations for a suite of cell proliferation assays. In each row the distributions of agents at time $\lambda t = T = 0, 1, 3, 5$, as indicated, are shown along with the corresponding $f(C)$. Each simulation is initiated by randomly populating a lattice of size $I = 28$ and $J = 24$, so that each site is occupied with probability 5%. All simulations correspond to $\Delta = \tau = P_m = 1$, $P_p = 0.05$ and $r = 4$.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

118
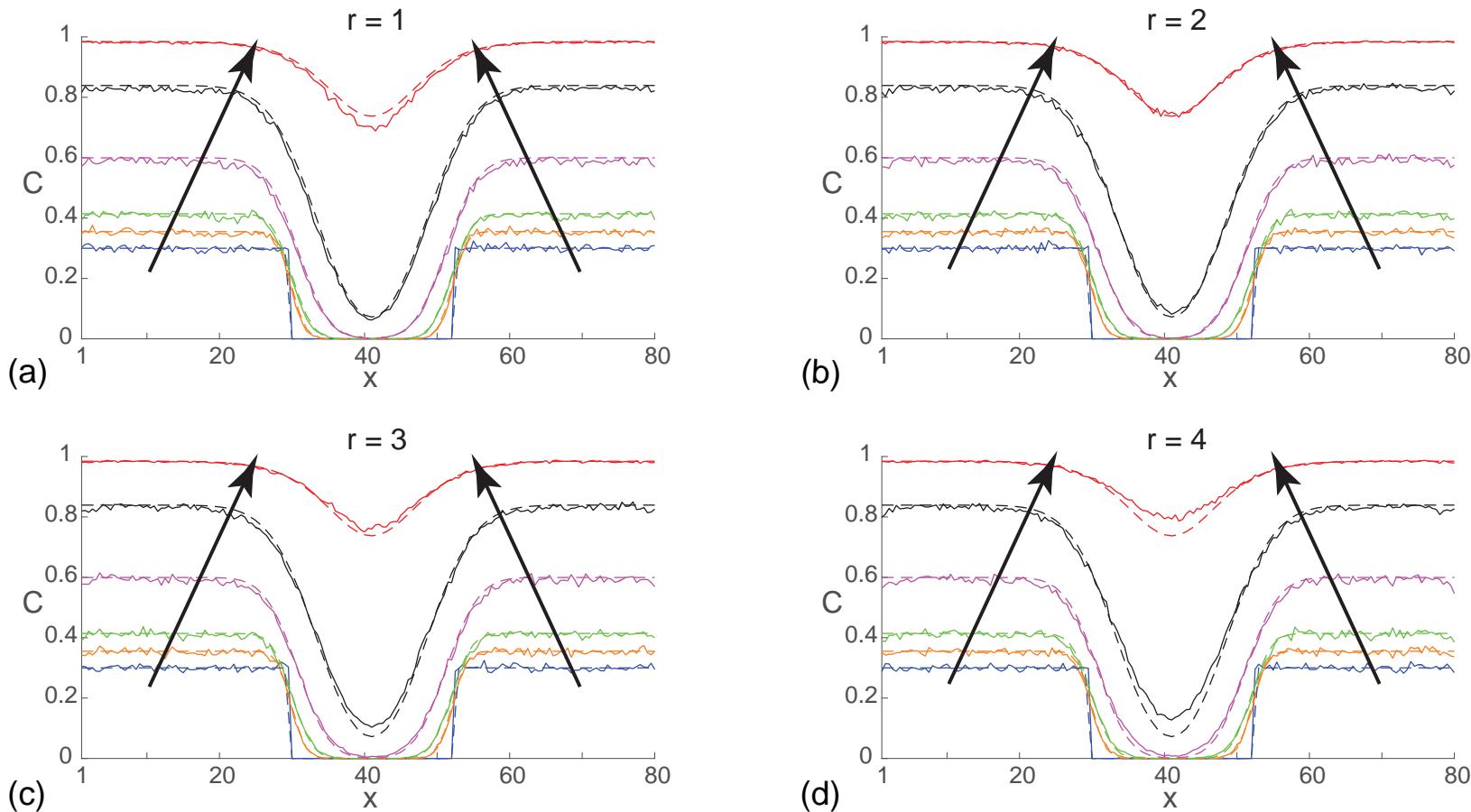
Figure 4.13: Comparison of averaged simulation data and the solution of the corresponding continuum model for a cell proliferation assay with: $f(C) = 1 - C$, as shown in (a)-(c); $f(C) = 1 - C^2$, as shown in (d)-(f); $f(C) = (1 - C)^2$, as shown in (g)-(i); and $f(C) = (1 - C^2)^2$, as shown in (j)-(l). Results in (b), (e), (h) and (k) compare averaged simulation data and the solution of the corresponding continuum model for a range of $1 \leq r \leq 4$ where the initial condition corresponds to 5% of sites being randomly occupied. All simulations are performed on a lattice with $I = 28$ and $J = 24$, and results are averaged across 300 identically prepared realisations of the discrete model. Profiles in (c), (f), (i) and (l) quantify the discrepancy between the solution of the continuum model and the average simulation data. All simulation results correspond to $\Delta = \tau = P_m = 1$ and $P_p = 0.05$, and the numerical solution of the continuum model is obtained with $\delta t = 1 \times 10^{-3}$ and $\epsilon = 1 \times 10^{-5}$. In all cases $T = \lambda t$.

Chapter 4. Stochastic simulation tools and continuum models for describing
two–dimensional collective cell spreading with universal growth functions

119

Figure 4.14: Comparison of averaged simulation data and the solution of the corresponding continuum model for a cell proliferation assay with: $f(C) = 1 - C$, as shown in (a)-(c); $f(C) = 1 - C^2$, as shown in (d)-(f); $f(C) = (1 - C)^2$, as shown in (g)-(i); and $f(C) = (1 - C^2)^2$, as shown in (j)-(l). Results in (b), (e), (h) and (k) compare averaged simulation data and the solution of the corresponding continuum model for a range of $1 \leq r \leq 4$ where the initial condition corresponds to 20% of sites being randomly occupied. All simulations are performed on a lattice with $I = 28$ and $J = 24$, and results are averaged across 300 identically prepared realisations of the discrete model. Profiles in (c), (f), (i) and (l) quantifies the discrepancy between the solution of the continuum model and the average simulation data. All simulation results correspond to $\Delta = \tau = P_m = 1$ and $P_p = 0.05$, and the numerical solution of the continuum model is obtained with $\delta t = 1 \times 10^{-3}$ and $\epsilon = 1 \times 10^{-5}$. In all cases $T = \lambda t$.
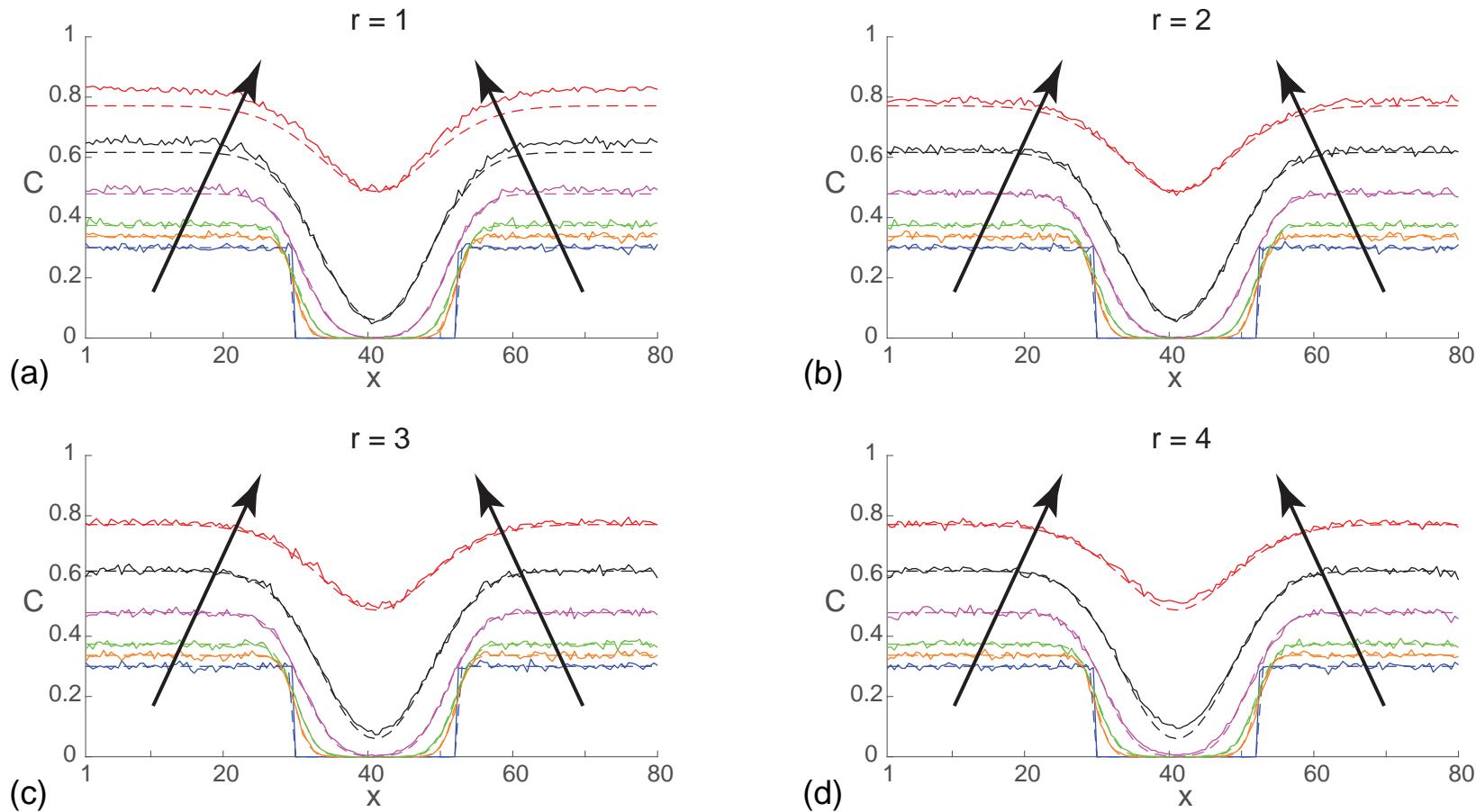
Figure 4.15: Snapshots of simulations for a suite of scratch assays. In each row the distributions of agents at time $\lambda t = T = 0, 0.25, 1, 3$ are shown along with the corresponding $f(C)$. Each simulation is initiated by randomly populating a lattice, corresponding to lattice of size $I = 80$ and $J = 68$, so that each site is occupied with probability 30%. A scratch of 23 lattice sites wide is made at $T = 0$. All simulations correspond to $\Delta = \tau = P_m = 1$, $P_p = 0.05$ and $r = 4$.
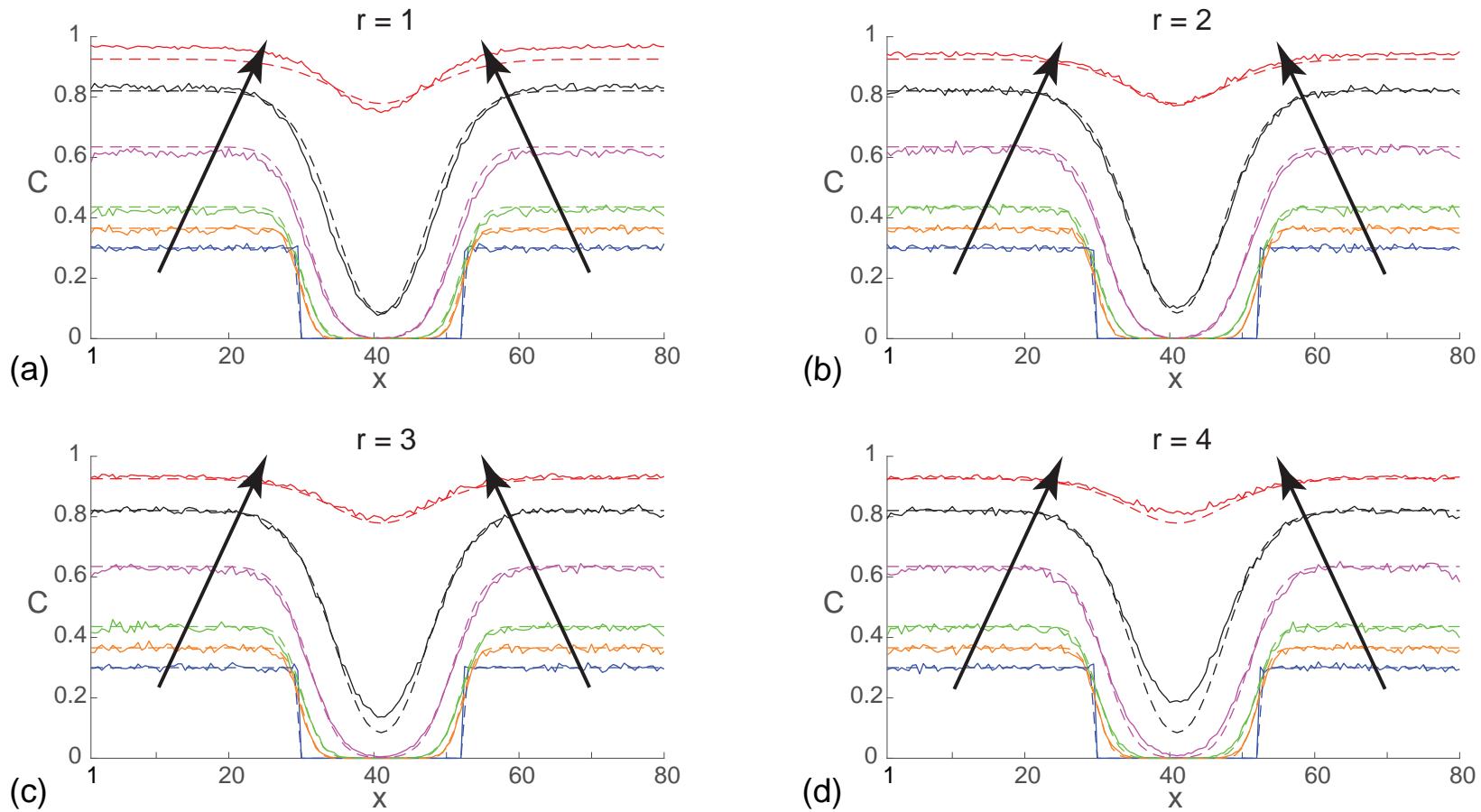
Figure 4.16: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = 1 - C$. Results in (a)-(d) compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines) for $r = 1, 2, 3$ and 4, respectively. In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1.25, 2.5, 5$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.05$, and the numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Figure 4.17: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = 1 - C^2$. Results in (a)-(d) compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines) for $r = 1, 2, 3$ and 4, respectively. In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1.25, 2.5, 5$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.05$, and the numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

**Figure 4.18:** Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = (1-C)^2$. Results in (a)-(d) compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines) for $r = 1, 2, 3$ and 4, respectively. In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1.25, 2.5, 5$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.05$, and the numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

Figure 4.19: Comparison of averaged simulation data and the solution of the corresponding continuum model for a scratch assay with $f(C) = (1 - C^2)^2$. Results in (a)-(d) compare averaged simulation data (solid lines) and the solution of the corresponding continuum model (dashed lines) for $r = 1, 2, 3$ and 4, respectively. In each subfigure, agent density profiles are given at $\lambda t = T = 0, 0.25, 0.5, 1.25, 2.5, 5$, and the arrows show the direction of increasing time. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = \tau = P_m = 1$ and $P_p = 0.05$, and the numerical solution of the continuum model is with $\delta x = 0.25$, $\delta t = 0.1$ and $\epsilon = 1 \times 10^{-5}$.

5 A computational modelling framework to quantify the effects of passaging cell lines

Abstract

In vitro cell culture is routinely used to grow and supply a sufficiently large number of cells for various types of cell biology experiments. Previous experimental studies report that cell characteristics evolve as the passage number increases, and various cell lines can behave differently at high passage numbers. To provide insight into the putative mechanisms that might give rise to these differences, we perform *in silico* experiments using a random walk model to mimic the *in vitro* cell culture process. Our results show that it is possible for the average proliferation rate to either increase or decrease as the passaging process takes place, and this is due to a competition between the initial heterogeneity and the degree to which passaging damages the cells. We also simulate a suite of scratch assays with cells from near–homogeneous and heterogeneous cell lines, at both high and low passage numbers. Although it is common in the literature to report experimental results without disclosing the passage number, our results show that we obtain significantly different closure rates when performing *in silico* scratch assays using cells with different passage numbers. Therefore, we suggest that the passage number should always be reported to ensure that the experiment is as reproducible as possible. Furthermore, our modelling also suggests some avenues for further experimental examination that could be used to validate or refine our simulation results.

5.1 Background

In vitro cell culture is routinely used to grow and supply cells for various types of cell biology experiments [2]. These experiments are used to study a wide range of biological phenomena including drug design, cancer spreading and tissue repair [32, 49, 63, 121]. According to the American Type Culture Collection (ATCC) protocols, to grow cells

in traditional two–dimensional (2D) *in vitro* cell culture, cells propagated in a growth medium are initially seeded as a monolayer in a cell culture flask [8], as shown in Figure 5.1(a). Cells are seeded in a monolayer with a density typically varying from 10–20% of confluence [8]. Cells are then cultured in an incubator, in an appropriate temperature and CO_2 concentration, and grown until they reach a density of 80%–90% of confluence [8]. To continue growing the population, cells are lifted, often using trypsin, and split into smaller proportions. The smaller subpopulations are transferred into new cell culture flasks to re-grow [8]. This process is referred to as *passaging*, with passage number indicating the number of splits [8, 9]. Although passaging is a standard process in 2D cell culture, the passage number of cells used in experiments is not always reported in experimental protocols [6, 42, 115–117, 120].

It is known that passaging can affect cells in a number of ways, and therefore has the potential to impact the reproducibility of *in vitro* experiments [120]. There are many ways in which passaging can affect cells. For example, primary cells, which are directly isolated from living tissues [41], undergo morphological changes and cumulative damage as the passage number increases [21, 31, 45, 62, 68, 88, 95, 99]. As a result, the cell morphology, migration rate and proliferation rate can become increasingly varied, which is thought to increase the heterogeneity in cell lines [31, 45, 68, 95, 99]. Because a range of cell behaviours could depend on passage number, the passaging process can be a source of variability that affects the reproducibility of various *in vitro* experiments, such as 2D scratch assays [6, 9, 120].

Seemingly contradictory observations have been reported about the effects of passaging cell lines [31, 45, 73, 95, 99]. For example, Hayflick reports that for human diploid cell lines that are immortalised, cells at high passage numbers demonstrate increased generation time, gradual cessation of mitotic activities, and accumulation of cellular debris [45]. This observation of decreased cell proliferation rate is also supported by studies of other cell lines [31, 95, 99]. However, Lin and coworkers show that the population of LNCaP cells at passage number 70 is over two times larger than that at passage number 38 after five days [73]. It has also been stated that for some cell lines, changes due to the passaging process occur at relatively low passage numbers, whereas for other cell lines the changes occur at relatively high passage numbers [9]. Therefore, we are motivated to undertake a mechanistic study to quantify how different variables relevant to the passaging process might give rise to such seemingly contradictory observations and to explore how these effects might impact the reproducibility of *in vitro* experiments.

Although problems associated with high passage numbers are widely acknowledged, the mechanism of passage–induced changes is not well understood [9, 31, 36, 45, 73, 90, 95, 99, 129]. For example, standard experimental protocols suggest avoiding cells at high passage numbers, whereas the definition of a 'high passage number' is rather vague [9, 90]. On the other hand, the mechanism that causes the seemingly contradictory observations at high passage numbers still remains unknown [31, 45, 73, 95, 99]. Computational models can be useful for exploring mechanisms and trade-offs between various factors. Therefore,

the problems with high passage numbers invoke us to apply a computational model to investigate putative mechanisms that could lead to the seemingly contradictory changes. As far as we are aware, this is the first time that problems with passaging of cell lines are investigated using a computational approach of this kind.

In this work, we describe a mathematical model that can be used to study the passaging process in 2D *in vitro* cell culture [50, 122], particularly for primary cell lines. A key feature of our model is that we allow individual cells within the population to take on a range of characteristics, such as variable proliferation rates, and therefore it is natural to focus on using a discrete model for this purpose [28, 50]. In particular, we are interested in examining whether the apparently contradictory effects of passaging reported in the literature can be recapitulated using a fairly straightforward discrete model. After examining the trade-off between cell heterogeneity and passage–induced damage, we then use the *in silico* model to examine how the passaging process might affect the reproducibility of scratch assays [49, 71]. In this work we focus on the impact of passaging on the cell proliferation rate, and apply a discrete model to explain how passaging can lead to either increasing or decreasing proliferation rates, depending on the competing effects of natural inheritance versus passaging–induced damage. In our model we impose three key assumptions: (i) the passaging process does not affect the cells' ability to migrate; (ii) initially the proliferation rate of each cell is assigned randomly from a normal distribution; and (iii) when proliferating, daughter cells inherit the same proliferation rate as the mother cell. Our approach is to focus on two prototype cell populations. The first is near–homogeneous in the sense that the proliferation rate of the cells is close to constant throughout the population initially. The second has a distinctively heterogeneous distribution of proliferation rates. For each prototype population, we systematically vary the amount of damage caused by passaging to investigate the impact of the damage.

5.2   Discrete model

5.2.1   Model framework

We use a discrete random walk model to simulate the passaging process and we refer to individual random walkers in the model as cells. All simulations are performed on a hexagonal lattice, with the lattice spacing Δ taken to be equal to the average cell diameter [50]. The model includes crowding effects by ensuring that there is, at most, one cell per lattice site [109]. Each lattice site, indexed (i, j) where $i, j \in \mathbb{Z}^+$, has position

$$(x, y) = \begin{cases} \left((i-1)\Delta, \sqrt{3}(j-1)\Delta/2\right) & \text{if } j \text{ is even,} \\ \left((i-1/2)\Delta, \sqrt{3}(j-1)j\Delta/2\right) & \text{if } j \text{ is odd,} \end{cases}$$

such that $1 \leq i \leq I$ and $1 \leq j \leq J$ [50]. In any single realisation of the model, the occupancy of site (i, j) is denoted $C_{i,j}$, with $C_{i,j} = 1$ if the site is occupied, and $C_{i,j} = 0$ if vacant.

If there are $N(t)$ cells at time t, then during the next time step of duration τ, $N(t)$ cells are selected independently at random, one at a time with replacement, and given the opportunity to move [50, 109]. The randomly selected cell attempts to move, with probability P_m, to one of the six nearest neighbour sites, with the target site chosen randomly. Motility events are aborted if a cell attempts to move to an occupied site. Once $N(t)$ potential motility events are attempted, another $N(t)$ cells are selected independently, at random, one at a time with replacement, and given the opportunity to proliferate with probability P_p. The location of the daughter cell is chosen, at random, from one of the six nearest neighbour lattice sites [50, 109]. If the selected lattice site is occupied, the potential proliferation event is aborted. In contrast, if the selected site is vacant, a new daughter cell is placed on that site. After the $N(t)$ potential proliferation events have been attempted, $N(t + \tau)$ is updated [50, 109].
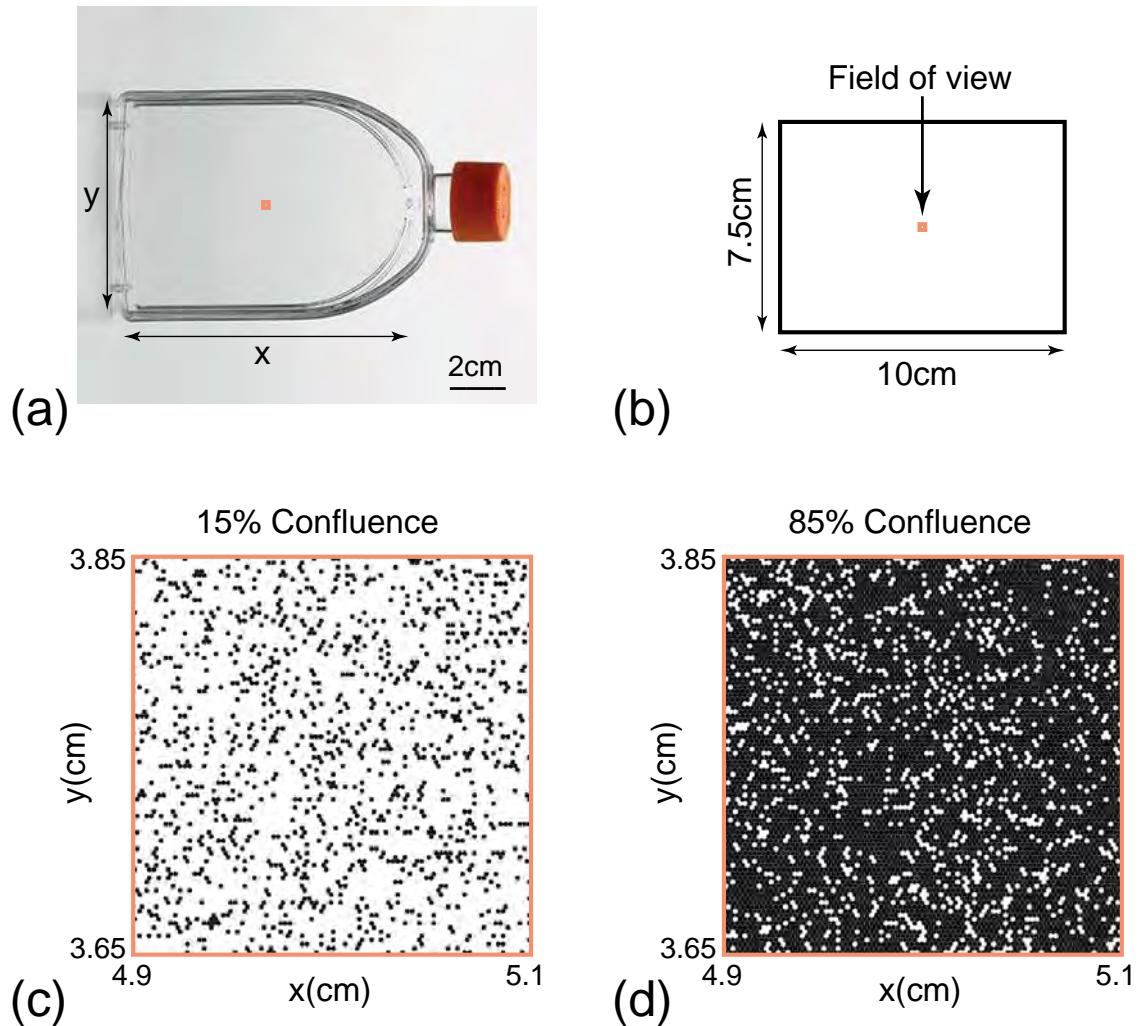
The discrete models in this study are coded in C++. And the C++ simulation code is supplied in the Supplementary Material (Section 5.5).

5.2.2 Simulation domain

The domain is a rectangle of dimensions 10 cm by 7.5 cm, which we use to represent the 75 cm^2 cell culture flask in Figure 5.1(a). This corresponds to a simulation domain in which $I = 4168$ and $J = 3610$, with $\Delta = 24$ μm [49]. Therefore, the maximum number of cells in a 100% confluent monolayer is approximately 15 million. To simplify our visualisation of the model output, although we always perform simulations on the entire 10 cm by 7.5 cm simulation domain, we visualise a smaller, 2 mm by 2 mm, subregion in the centre of the simulation domain, as shown in Figure 5.1(b). No flux boundary conditions along the boundaries of the simulation domain are applied in all cases. For the remainder of this work, we visualise snapshots of the distribution of cells in the smaller field of view, such as the results in Figure 5.1(c)–(d).

5.2.3 Initial condition

Simulations are initiated by randomly populating 15% of lattice sites [8]. At each passage number, the growth of cells in the culture is terminated when 85% confluence is reached. The migration probability P_m of each cell is held constant. Motivated by experimental data of the duration of the mitotic phase for individual cells [39], each cell is initially assigned a random value of P_p, drawn from a normal distribution $\mathcal{N}(\mu_p, \sigma)$ to mimic the stochasticity in proliferation rate among the initial population. When a proliferation event takes place, we invoke the simplest mechanism by assuming that both daughter cells inherit the proliferation rate of the mother cell. For all simulations we set $P_m = 0.35$, $\mu_p = 0.004$, $\Delta = 24$ μm and $\tau = 1/12$ h so that we are considering cell populations with a typical cell diameter, cell diffusivity ($D \approx 600$ μm^2/h) and average proliferation rate ($\lambda \approx 0.05$ /h) [50, 109]. We consider two prototype cell populations: (i) a near–homogeneous cell population with a relatively small variance, $\sigma = 1 \times 10^{-4}$; and (ii)

Figure 5.1: Schematic illustration of the simulation domain. (a) Photograph of a 75 cm^2 cell culture flask. (b) Schematic of the 10 cm $\times$ 7.5 cm simulation domain that represents the 75 cm^2 flask. The orange squares in (a) and (b) indicate the 2 mm $\times$ 2 mm field of view. (c) Snapshot of the field of view at 15% confluence. (d) Snapshot of the field of view at 85% confluence.

a heterogeneous cell population with a larger variance, $\sigma = 1 \times 10^{-3}$. We choose the values of the standard deviation σ, so that the proliferation rate distribution is within a biologically reasonable range, and the degree of heterogeneity in the near–homogeneous and heterogeneous cell lines are distinguishable.

5.2.4 Passaging

In our simulations passaging takes place immediately after the population grows to 85% confluence [8]. To split the populations we randomly select a number of cells that is equivalent to cover 15% of lattice sites. These cells are randomly placed on an empty simulation domain to mimic the splitting of cells in the passaging process. Note that P_m is constant for all cells whereas we allow P_p to vary amongst the population and we also assume that the process of passaging the cells involves some damage [95]. Considering that the passaging process involves a combination of chemical (e.g. the usage of trypsin)

and mechanical disturbances known to disrupt normal cell behavior, it is reasonable to incorporate some kind of damage mechanism into the passage simulations [8,21,31,45,88, 95,99]. However, the exact cause and the form of the passage–induced damage have not been established. Therefore any form of the passage–induced damage which illustrates certain degrees of stochasticity could be a reasonable choice. In this study, we consider two different degrees of passage–induced damage:

- Small amount of damage: P_p of each cell is decreased by ϵ, where $\epsilon \sim \mathcal{N}(2\times10^{-5}, 2\times 10^{-5})$; and
- Large amount of damage: P_p of each cell is decreased by ϵ, where $\epsilon \sim \mathcal{N}(1\times10^{-4}, 1\times 10^{-4})$.

Each time the population of cells is split, the passage number increases by one. As previous studies indicate that cell proliferation increases at high passage numbers [73], it is possible to assume that the passage–induced damage could lead to the increase in proliferation rate. However, since the aim of this study is to examine the trade-offs between the initial heterogeneity in cell proliferation and the passage–induced damage, in both scenarios we only consider non–negative passage–induced damage by changing any negative damage to zero. This assumption allows us to limit the factors that can increase cell proliferation.
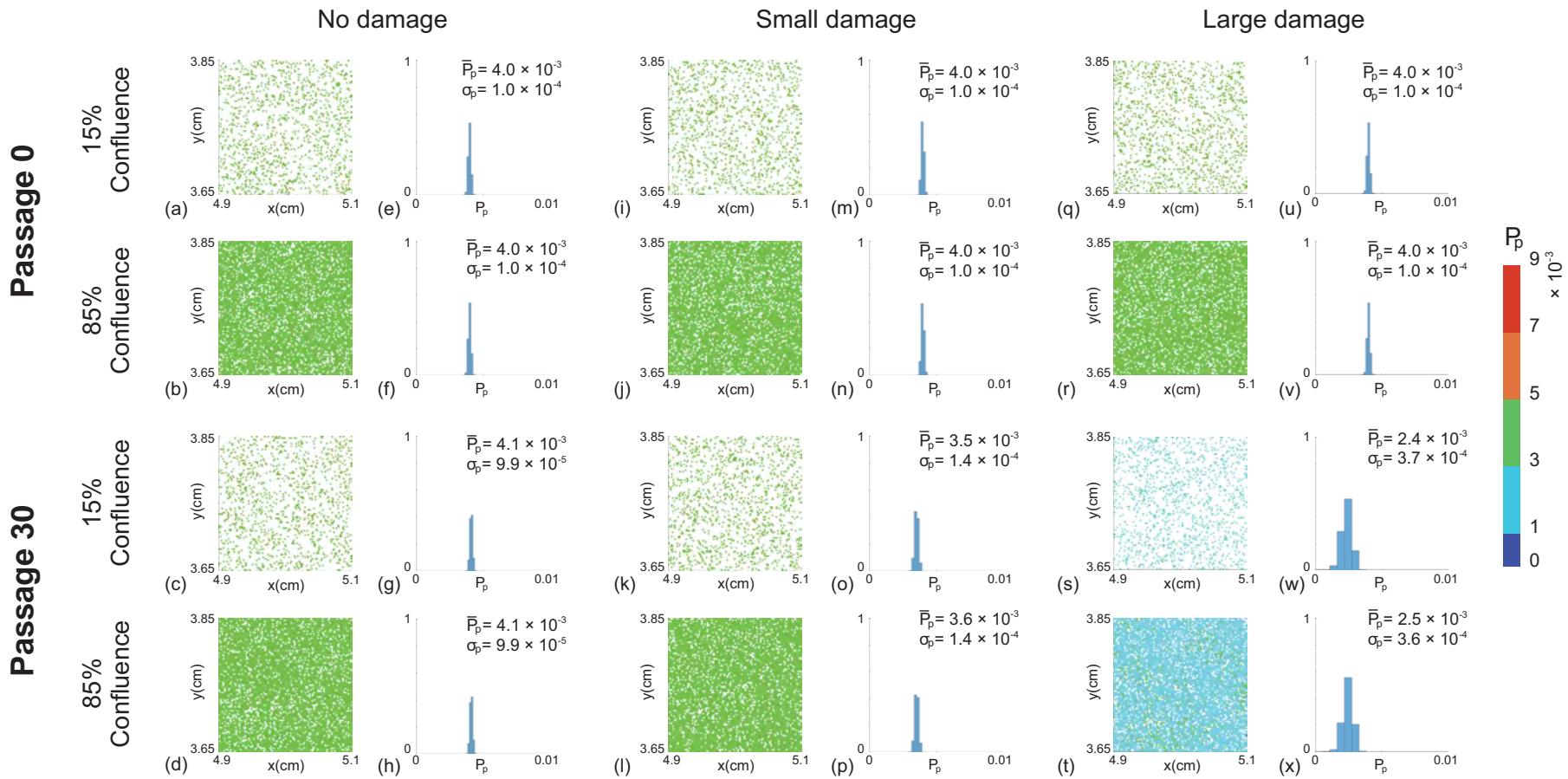
5.3 Results

5.3.1 Passaging cell lines without passage–induced damage
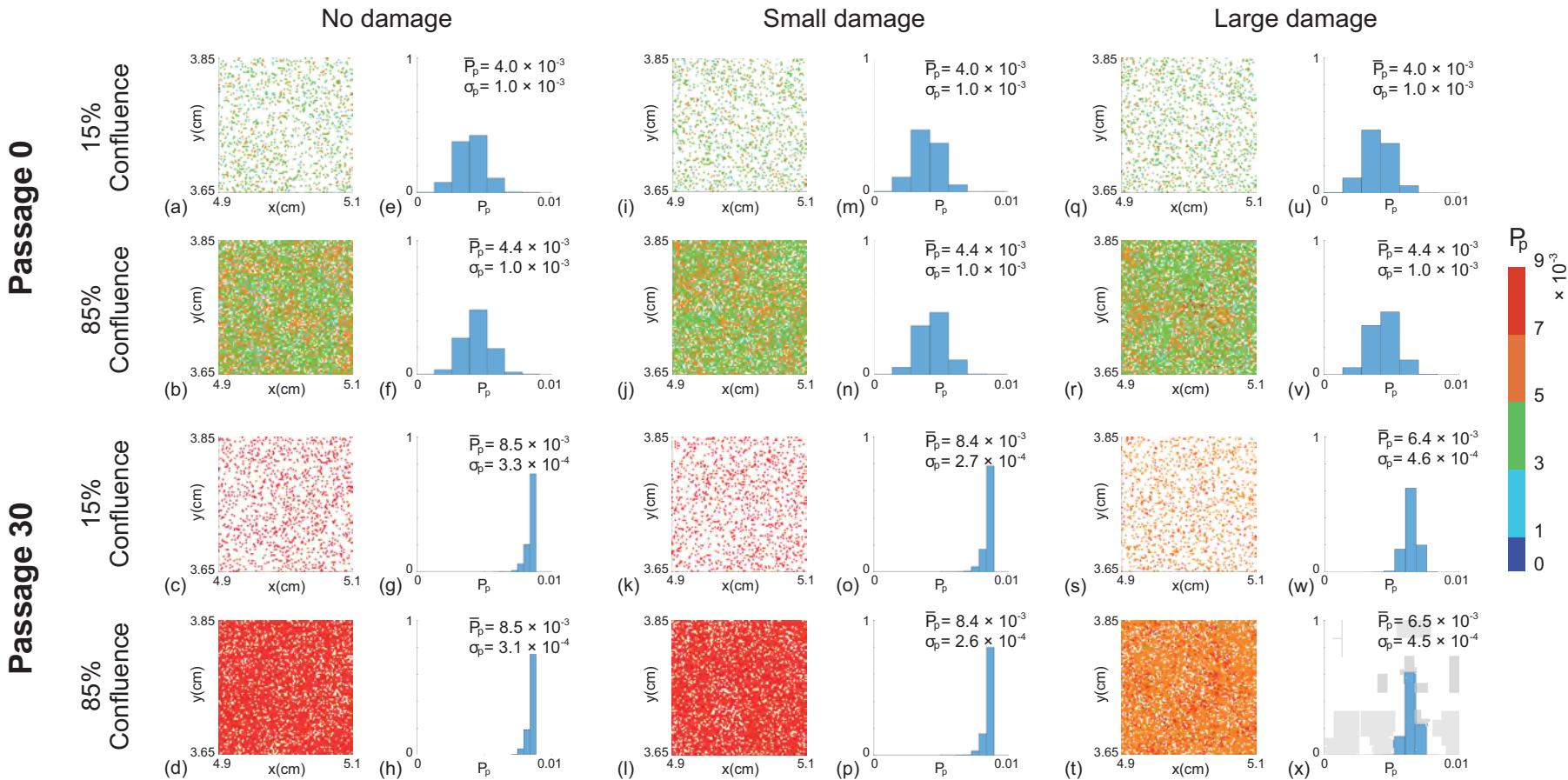
We first investigate how the initial degree of heterogeneity in proliferation rate changes as the passage number increases. In this first set of results we do not consider any form of passage–induced damage. We consider a suite of simulations from passage number 0 to 30 and present results for both the near–homogeneous cell line and the heterogeneous cell line. Snapshots of the field of view at passage number 0 and passage number 30, for both prototype cell populations, are shown in Figure 5.2(a)–(d) and Figure 5.3(a)–(d), respectively. In each snapshot, different colours of cells represent different ranges of the proliferation rate, with red indicating the fastest–proliferating cells and blue showing the slowest–proliferating cells. At the end of passage number 0 we observe a larger variation in cell proliferation rate in the heterogeneous cell line than the near–homogeneous cell line, as we might expect. At the end of passage number 30 we see that there is a dramatic change in the average proliferation rate of cells in the heterogeneous cell line. This change is caused by the fact that cells with higher proliferation rates are more likely to produce daughter cells that directly inherit the higher proliferation rate of the mother cell. Therefore, we observe a greater proportion of faster–proliferating cells in the heterogeneous cell line at high passage number. This leads to a larger average value of P_p and a greater variation in P_p across the whole population of cells in the heterogeneous cells line, as shown in Figure 5.2(e)–(h) and Figure 5.3(e)–(h), respectively.

To summarise how the cell proliferation rate changes with passage number, we plot the evolution of the proliferation rate data from the entire populations as boxplots [79] in Figure 5.4. The boxplots show the median and quartiles of the distribution of P_p from the entire population as a function of the passage number. Comparing results in Figure 5.4(a) and Figure 5.4(d) shows that the median P_p increases much faster in the heterogeneous cell line than the near–homogeneous cell line. For the near–homogeneous cell line the distribution of P_p appears to be approximately independent of the passage number in this case. In contrast, the distribution of P_p for the heterogeneous cell line is strongly dependent on the passage number. In particular, the median P_p increases, and the distribution of P_p becomes increasingly negatively skewed as the passage number increases. Overall, these results suggest that starting with the same average proliferate rate, the degree of heterogeneity of the cell line alone is enough to lead to very different outcomes when the two cell lines are sufficiently passaged. Therefore, the initial heterogeneity of the cell line appears to be important in terms of understanding how passaging affects properties of cell lines.
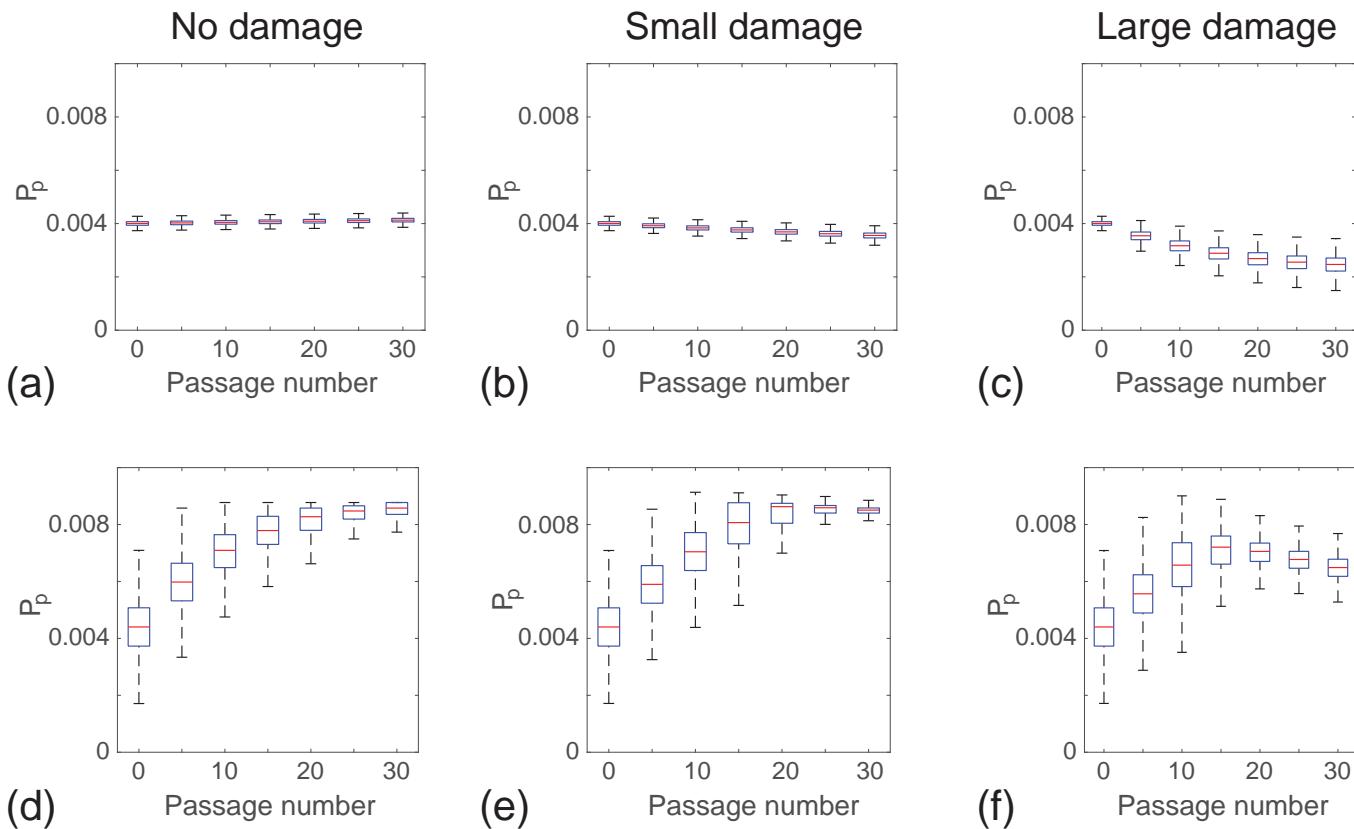
In this first set of results, we find that differences in the cell proliferation rate among the cell population can lead to changes in the overall population behaviour at sufficiently high passage numbers. We note that in both prototype cell populations, the average proliferation rate increases with the passage number and this is consistent with some previous experimental studies [73]. However, most experimental studies report a decrease in average proliferation rate with increasing passage number [31, 45, 95, 99]. This observation motivates us to include a second mechanism in our discrete model, namely passage–induced damage.

Figure 5.2: Snapshots of simulations for a near–homogeneous cell line. For each passage number, snapshots at the beginning (15% confluence) and end (85% confluence) of the experiments are shown. Results in (a)–(d),(i)–(l) and (q)–(t) show snapshots of the field of view at passage number 0 and 30, with $\epsilon = 0$, $\epsilon \sim \mathcal{N}(2 \times 10^{-5}, 2 \times 10^{-5})$ and $\epsilon \sim \mathcal{N}(1 \times 10^{-4}, 1 \times 10^{-4})$, respectively. Results in (e)–(h),(m)–(p) and (u)–(x) show distributions of P_p for the entire domain at passage number 0 and 30, with $\epsilon = 0$, $\epsilon \sim \mathcal{N}(2 \times 10^{-5}, 2 \times 10^{-5})$ and $\epsilon \sim \mathcal{N}(1 \times 10^{-4}, 1 \times 10^{-4})$, respectively. The distribution of P_p in each subfigure is obtained from one single realisation. The colour bar indicates P_p for individual cells. $\bar{P}_p$ and σ_p represent the mean and standard deviation of P_p. Each simulation is initiated by randomly populating 15% of lattice sites, on a lattice of size $I = 4168$ and $J = 3610$, with $P_p \sim \mathcal{N}(4 \times 10^{-3}, 1 \times 10^{-4})$ for each cell. All simulations correspond to $\Delta = 24~\mu$m, $\tau = 1/12$ h, and $P_m = 0.35$.

Figure 5.3: Snapshots of simulations for a heterogeneous cell line. For each passage number, snapshots at the beginning (15% confluence) and end (85% confluence) of the experiments are shown. Results in (a)–(d),(i)–(l) and (q)–(t) show snapshots of the field of view at passage number 0 and 30, with $\epsilon = 0$, $\epsilon \sim \mathcal{N}(2 \times 10^{-5}, 2 \times 10^{-5})$ and $\epsilon \sim \mathcal{N}(1 \times 10^{-4}, 1 \times 10^{-4})$, respectively. Results in (e)–(h),(m)–(p) and (u)–(x) show distributions of P_p for the entire domain at passage number 0 and 30, with $\epsilon = 0$, $\epsilon \sim \mathcal{N}(2 \times 10^{-5}, 2 \times 10^{-5})$ and $\epsilon \sim \mathcal{N}(1 \times 10^{-4}, 1 \times 10^{-4})$, respectively. The distribution of P_p in each subfigure is obtained from one single realisation. The colour bar indicates P_p for individual cells. $\bar{P}_p$ and σ_p represent the mean and standard deviation of P_p. Each simulation is initiated by randomly populating 15% of lattice sites, on a lattice of size $I = 4168$ and $J = 3610$, with $P_p \sim \mathcal{N}(4 \times 10^{-3}, 1 \times 10^{-3})$ for each cell. All simulations correspond to $\Delta = 24$ μm, $\tau = 1/12$ h, and $P_m = 0.35$.

Figure 5.4: Distribution of P_p as a function of passage number. Results in (a)–(c) show boxplots of P_p for a near–homogeneous cell line at 85% confluence for: (a) no damage $\epsilon = 0$; (b) small amount of damage, $\epsilon \sim \mathcal{N}(2 \times 10^{-5}, 2 \times 10^{-5})$; and (c) large amount of damage, $\epsilon \sim \mathcal{N}(1 \times 10^{-4}, 1 \times 10^{-4})$. Results in (d)–(f) show boxplots of P_p for a heterogeneous cell line at 85% confluence for: (d) no damage $\epsilon = 0$; (e) small amount of damage, $\epsilon \sim \mathcal{N}(2 \times 10^{-5}, 2 \times 10^{-5})$; and (f) large amount of damage, $\epsilon \sim \mathcal{N}(1 \times 10^{-4}, 1 \times 10^{-4})$. In each subfigure the distribution of P_p at individual passage numbers is obtained from one single realisation. Each simulation is initiated by randomly populating 15% of lattice sites on a lattice of size $I = 4168$ and $J = 3610$, with $P_p \sim \mathcal{N}(4 \times 10^{-3}, 1 \times 10^{-4})$ for the near–homogeneous cell line and $P_p \sim \mathcal{N}(4 \times 10^{-3}, 1 \times 10^{-3})$ for the heterogeneous cell line. All simulations correspond to $\Delta = 24$ μm, $\tau = 1/12$ h, and $P_m = 0.35$.

5.3.2   Passaging cell lines with passage–induced damage

We now investigate the impact of including passage–induced damage, and we consider
both small and large amounts of damage scenarios. All other features of our simulations
are maintained as described in the section without passage–induced damage. Snapshots
of simulations including small and large amounts of damage, and boxplots showing the
distribution of P_p data are shown in Figure 5.2–5.4. Comparing results in Figure 5.2(a)–
(h) and Figure 5.2(i)–(p) suggests that we observe very similar outcomes when we include
a small amount of damage in the simulations of the near–homogeneous cell line. Similarly,
results in Figure 5.3(a)–(h) and Figure 5.3(i)–(p) suggest that the small amount of damage
has a negligible impact on the passaging process for the heterogenous cell line. In contrast,
with the large amount of damage we see that the proliferation rate decreases by passage
number 30 in the near–homogeneous cell line, as shown in Figure 5.2(q)–(x), whereas
results in Figure 5.3(q)–(x) show that the proliferation rate increases by passage number
30, but the increase in proliferation rate is not as pronounced as in the case where there
is no damage in the heterogenous cell line.

Results in Figure 5.2–5.3 focus on snapshots of the population at passage numbers 0
and 30. Additional results in Figure 5.4(b)–(c) and Figure 5.4(e)–(f) to show how the
distribution of P_p evolves as a function of the passage number. For the near–homogeneous
cell line, the median P_p decreases monotonically with the passage number for both small
and large amounts of damage. In contrast, the median P_p for the heterogeneous cell line
behaves very differently as it increases until approximately passage number 20, and then
decreases with further passaging. These results, combined, provide a simple explanation
for why some previous studies have reported that the proliferation rate can increase with
passage number, as in the case of Figure 5.4(d)–(e), whereas other studies suggest that
the proliferation rate can decrease with passage number, as in the case of Figure 5.4(c).
In fact, our results suggest that it is possible to have a situation where the proliferation
rate both increases and decreases with passage number, as in the case of Figure 5.4(f),
and we observe different trends depending on the passage number. These differences arise
in our model due to a trade-off between the initial heterogeneity of the cell line and the
amount of damage sustained in the passaging process.

5.3.3   Scratch assay with passaged cells

Having demonstrated that the interplay between cell heterogeneity and passage–induced
damage can lead to complicated trends in the relationship between the proliferation rate
and passage number, it is still unclear how these kinds of differences can affect how
we interpret *in vitro* experiments. To explore this issue we use cells from a range of
passage conditions to mimic a scratch assay [71]. For this purpose we focus on the
geometry associated with experimental images obtained from an IncuCyte ZOOM$^{\text{TM}}$
scratch assay [33, 37, 49, 96, 127], as shown in Figure 5.5. The images, of dimension 1400
μm $\times$ 1900 μm, show a fixed field of view that is much smaller than the spatial extent
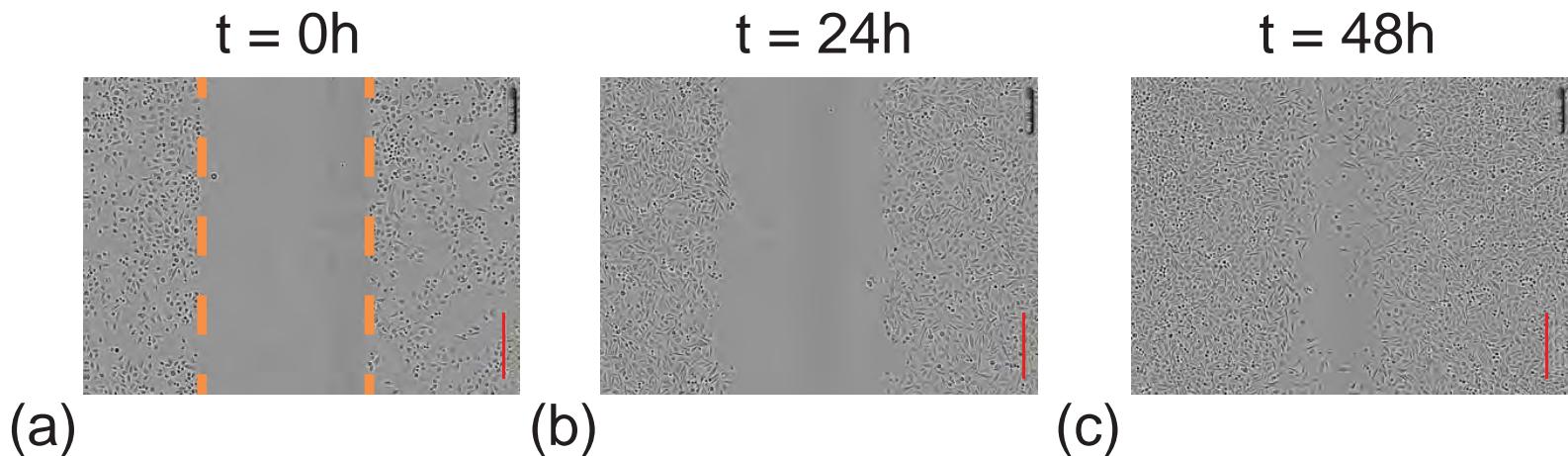
of the cells in the scratch assay [50, 51]. To model this situation we apply zero net flux
boundary conditions along all boundaries of the lattice. We use a lattice of size 80×68
to accommodate a typical population of cells with $\Delta = 24$ μm. To initialise the scratch
assay, we randomly populate all lattice sites with an equal probability of 30% [49]. To
simulate the scratch, we remove all cells from a vertical strip of width approximately
550 μm, and we then observe the rate at which the populations spread into the vacant
area. All cells have the same constant value of $P_m = 0.35$, and we assign values of P_p
by sampling from the various histograms in Figure 5.2 and 5.3. This means that we are
effectively simulating a scratch assay using cells from different cell lines, with different
amounts of passage-induced damage, and from different passage numbers according to
our *in silico* results of cell culture growth in the previous section.

Snapshots from the discrete model, showing the progression of the scratch assays, are
shown in Figure 5.6–5.7. In general we see that, regardless of the initial cell population,
all of the scratch assays lead to successful closure by approximately 48-72 h, which is
consistent with standard experimental observations [37, 49]. However, close examination
of the results reveals some differences. In particular, visual inspection of the snapshots
suggests that those cell populations with higher initial proliferation rate lead to larger
numbers of cells at later times, and hence more rapid closure of the initially–vacant
space. These trends are subtle, but are most obvious in Figure 5.7 where the population
corresponds to cells taken from passage number 30, with no damage, leading to more
effective re-colonisation of the initially–vacant space than cells from passage number 0.
Since these differences are subtle it may be difficult to detect them when visually compar-
ing results from scratch assays. Therefore, we will now quantify the spatial and temporal
distribution of cells in Figure 5.6–5.7 to provide more information.
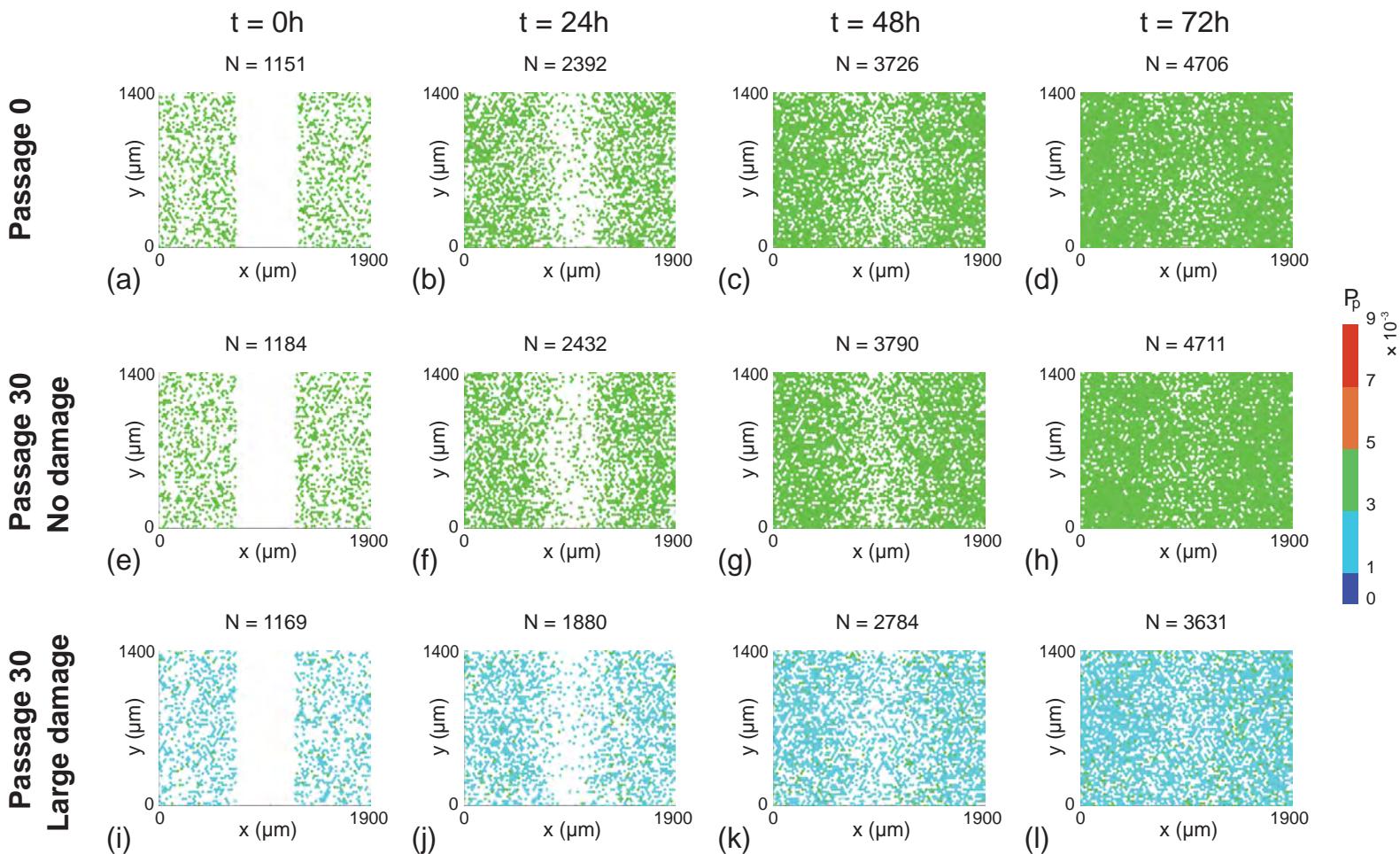
Since the initial condition is uniform in the vertical direction [50, 109], we average the
population density in Figure 5.6–5.7 along each vertical column of lattice sites to obtain

$$\langle C_i \rangle = \frac{1}{J} \sum_{j=1}^{J} C_{i,j}.$$
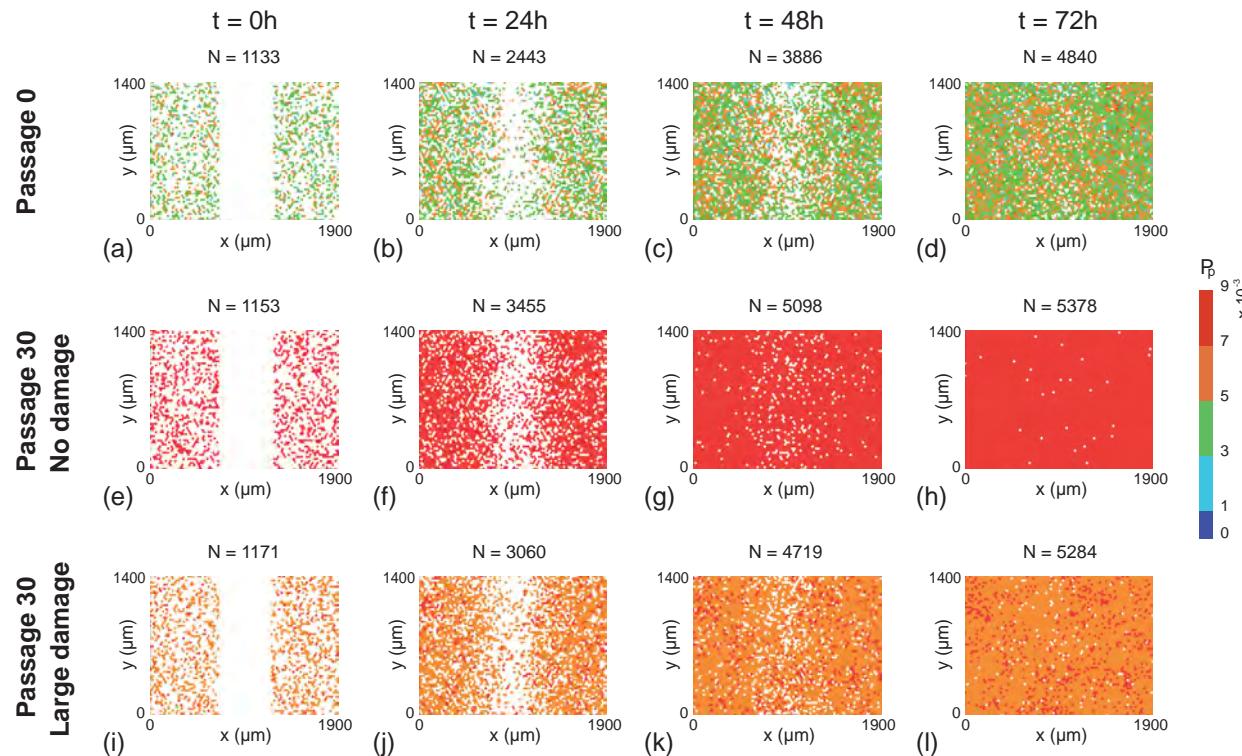
This quantity is further averaged by considering 100 identically prepared simulations of
the discrete model to reduce fluctuations [50]. This procedure allows us to plot the time
evolution of the average cell density as a function of the horizontal coordinate, as shown in
Figure 5.8 [50]. Results in Figure 5.8(a)–(b) suggest that the evolution of the cell density
profile is practically indistinguishable when we consider cells from the near–homogeneous
cell line that is passaged without damage, as we might expect from the results in Figure
5.4(a). In contrast, comparing results in Figure 5.8(a) with results in Figure 5.8(c) shows
that we observe very different results when damage is included in the passaging process
for the near–homogeneous cell line. When we consider the results in Figure 5.8(d)–(f),
for the heterogeneous cell line, we see that the evolution of the cell density profiles is very
different for all three cases considered.
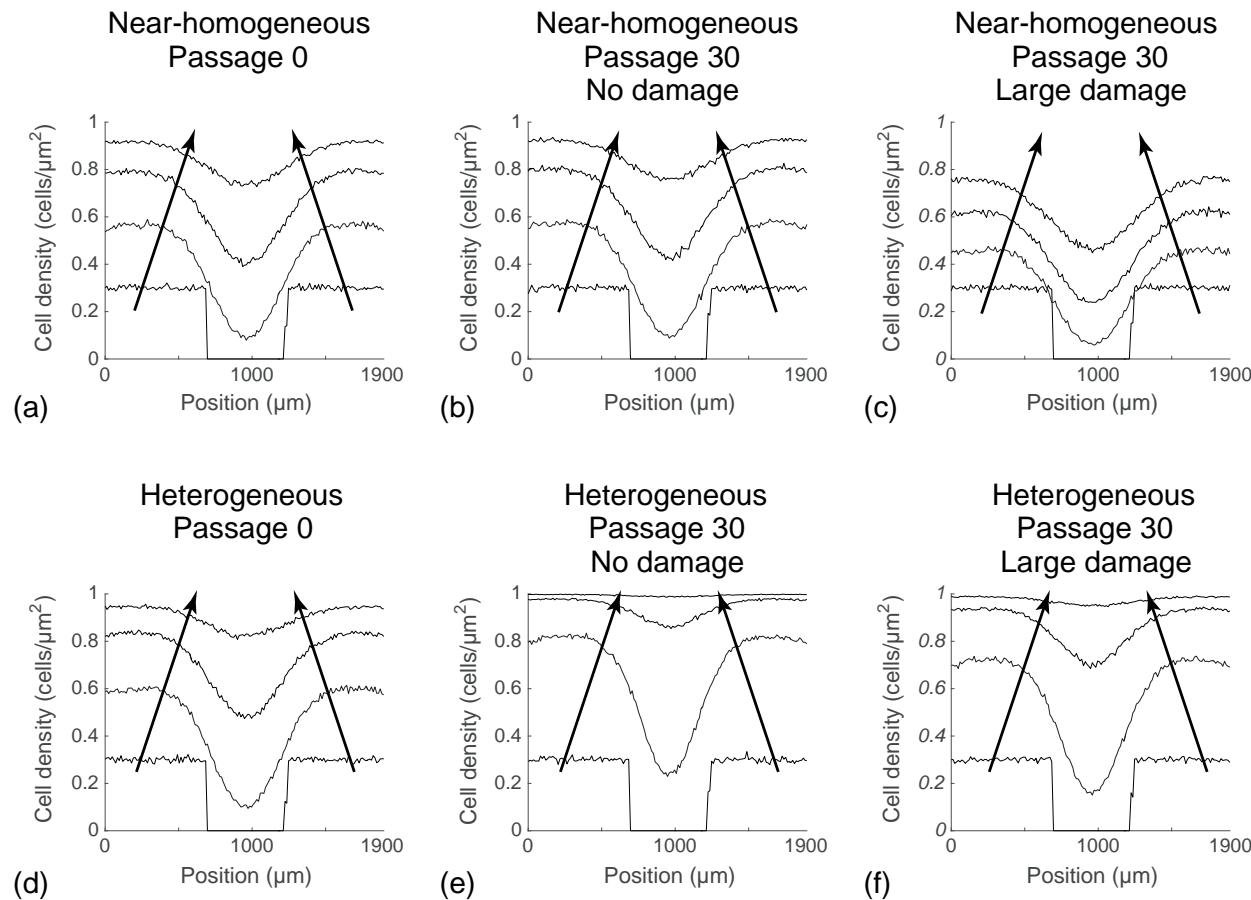
Figure 5.5: Experimental images of IncuCyte ZOOM$^{\text{TM}}$ scratch assay [49,51]. The images in (a)–(c) show the closure of the initially scratched region which is highlighted by the dashed orange lines at $t = 0$. The red scale bar corresponds to 300 μm.

Figure 5.6: Snapshots of a suite of scratch assays performed using a near–homogeneous cell line. In each column the distributions of cells at time $t = 0, 24, 48, 72$ h are shown. Each simulation is initiated by randomly populating a lattice of size 80×68, so that each site is occupied with probability 30%. A scratch of 23 lattice sites wide is made at $t = 0$ h. All simulations correspond to $\Delta = 24$ μm, $\tau = 1/12$ h, and $P_m = 0.35$. In each row the initial P_p of individual cells is assigned by randomly selecting from the *in silico* data in Figure 5.2(f), (h) and (x), respectively.

Figure 5.7: Snapshots of a suite of scratch assays performed using a heterogeneous cell line. In each column the distributions of cells at time $t = 0, 24, 48, 72$ h are shown. Each simulation is initiated by randomly populating a lattice of size 80×68, so that each site is occupied with probability 30%. A scratch of 23 lattice sites wide is made at $t = 0$ h. All simulations correspond to $\Delta = 24$ μm, $\tau = 1/12$ h, and $P_m = 0.35$. In each row the initial P_p of individual cells is assigned by randomly selecting from the *in silico* data in Figure 5.3(f), (h) and (x), respectively.

Figure 5.8: Averaged simulation data showing cell density profiles from the scratch assays. (a)–(c): Cell density profiles for a near–homogeneous cell line. (d)–(f): Cell density profiles for a heterogeneous cell line. In each subfigure cell density profiles are given at $t = 0, 24, 48, 72$ h, and the direction of increasing t is shown with the arrows. All simulation results are averaged across 100 identically prepared realisations of the discrete model, with $\Delta = 24$ μm, $\tau = 1/12$ h, and $P_m = 0.35$, on a lattice of size 80×68. In each subfigure the initial P_p of individual cells is assigned by randomly selecting from the corresponding *in silico* data in Figure 5.2(f), (h), (x), and Figure 5.3(f), (h), and (x), respectively.

5.4 Discussion

Passaging of cell lines is an essential processes of growing cells in cell culture [8, 9]. The passaging process involves both chemical and mechanical disturbances which accumulatively change cell characteristics. Problems associated with high passage numbers, such as the change of cell proliferation, are widely acknowledged. However, the mechanisms are not well understood [31, 36, 45, 73, 90, 95, 99, 129]. Therefore, the aim of this work is to use a computational approach to provide insight into the putative mechanisms that could possibly lead to the problems.

In this work, we apply a lattice–based discrete model to investigate and quantify the impact of passaging cell lines. Although there are many properties of cells that are affected by the passaging process [21, 31, 36, 45, 88, 95, 99, 129], we choose to focus on how passaging affects the cell proliferation rate. In our model, when a cell proliferates, the daughter cells directly inherit the same proliferation rate as the mother cell. Furthermore, we also assume that during the passaging process, the cell proliferation rate is decreased by some passage–induced damage. For all results presented, we investigate the role of cell heterogeneity by comparing results where we begin the passaging process with a hear-homogeneous population of cells where P_p is almost constant, with a heterogeneous population of cells where P_p varies significantly among the population.

In the literature, previous experimental studies have reported apparently contradictory results where some studies suggest that the average proliferation rate of cells can increase at large passage number [73], whereas other studies suggest that the average proliferation rate of cells can decrease with passage number [31, 45, 95, 99]. We find that by varying the competition between passage-induced damage and cell heterogeneity, our relatively straightforward simulation model can predict each of these outcomes.

To study how passage number can affect *in vitro* experiments, we simulate a suite of scratch assays using various populations of cells that are harvested from our *in silico* passaging process. Our simulation results show that the passage number can lead to subtle changes in the evolution of the scratch assay and these changes might be very difficult to detect visually. We provide additional information about how the distribution of cells in a scratch assay might be influenced by passage number by performing a large number of realisations and examining the average cell density profiles. These average cell density profiles make it obvious that the passage number could affect the rate of scratch closure. This observation, together with the fact that cell passage number is often unreported in the experimental literature [6, 120], could explain why scratch assays are notoriously difficult to reproduce [35]. In addition, the results of cell culture growth and scratch assays indicate that even at the same passage number, the initial heterogeneity in cell proliferation can give rise to very differently behaving cell populations. Therefore, separating cell population without reporting the proliferative capacity can also affect the reproducibility of *in vitro* experiments. However, the proliferative capacity of cell lines can

be difficult to measure experimentally, as most of the previous experiments only report the cell population evolution [45, 73], or the duration of the cell cycle [39].

There are several implications of this study that could be of interest to the experimental community. First, we suggest that the passage number of cell lines should always be reported. Second, there is a need for more experimental evidence about the impact of passaging on proliferation rates of various cell lines. For example, careful measurements of proliferation rates over a sequence of passage numbers would provide more insight into the variability of key cell properties in cell culture. This type of quantitative information would be invaluable for understanding reproducibility of standard *in vitro* experiments. Third, we acknowledge that our choices of the standard deviation, σ, to define the spread of the distribution of proliferation rates in the near–homogeneous and heterogeneous cell lines is rather theoretical. Recently, Haass et al. have devised new experimental methods that can be used to measure the durations of different phases in cell cycle for a range of melanoma cell lines [39]. This data could be used used to estimate the properties of the distribution of cell proliferation rates, such as the mean and standard deviation of the distribution of proliferation rates. Therefore, we suggest that similar experiments could be performed to generate proliferation rate distribution over various passage numbers for a range of different cell lines of interest. This data could then be directly integrated within our *in silico* models to examine the interplay between the degree of heterogeneity and passage–induced damage.

There are also several implications of this study that are of interest to the applied mathematics and mathematical biology communities. First, here we focus on the case where there is heterogeneity in the rate at which individual cells proliferate in the population but, we treat the motility rate as a constant. This is because most previous experimental studies have reported differences in the rate of proliferation as a function of passage number rather than differences in the rate of migration [45, 73, 99]. However, heterogeneity in cell migration rate can also affect the reproducibility of *in vitro* experiments [95], especially scratch assays in which cell migration plays a key role in wound closure [49]. An interesting extension of our present study would involve dealing with both variability in the motility rate and the proliferation rate [87]. Secondly, in our work we make the most straightforward assumption that daughter cells inherit P_p directly from the mother cell. It might be more plausible to introduce some stochasticity in the inheritance process and a reduction in proliferation rate according to the number of divisions of individual cells. It might also be plausible to incorporate some kind of ageing process where the proliferation depends on the age structure of the population [10, 95]. We have chosen not to include these additional details as we wish to present a simpler model that is capable of illustrating a proof–of–principle concept rather than capturing every possible feature of the underlying biology. Finally, another extension of this work would be to consider the derivation of an accurate mean–field approximation that could be used to describe the evolution of the cell density profiles in Figure 5.7. This is a challenging task because all previous derivations of such mean–field partial differential equations involve populations

of cells with constant rates [23, 50, 75, 100, 109, 119], whereas we are dealing with a more realistic heterogeneous population of cells.

5.5   Supplementary Material

C++ code for cell culture growth

```cpp
1   #include <cmath>
2   #include <iostream>
3   #include <vector>
4   #include <stdio.h>
5   #include <stdlib.h>
6   #include <fstream>
7   #include <ctime>
8   #include <string>
9   #include <sstream>
10  #include <algorithm>
11  #include <iterator>
12  #include <random>
13
14  using namespace std;
15
16  void migration1 (const int agent_row, const int agent_col,
17                   const int rowN, const int colN, int * migPosition) {
18      migPosition[0] = agent_row;
19      migPosition[1] = agent_col;
20      double P = ((double) rand() / (RAND_MAX)); //choose migration directions
21      if ((P < 1.0/6.0) && (agent_row > 0) && (agent_col > 0)) {//Position 1
22          migPosition[0] = migPosition[0] - 1;
23          migPosition[1] = migPosition[1] - 1;}
24      else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
              < colN-1)) {//Position 2
25          migPosition[0] = migPosition[0] - 1;}
26      else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-1)) {//
              Position 3
27          migPosition[1] = migPosition[1] + 1;}
28      else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
              agent_col < colN-1)) {//Position 4
29          migPosition[0] = migPosition[0] + 1;}
30      else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1) && (
              agent_col > 0)) {//Position 5
31          migPosition[0] = migPosition[0] + 1;
32          migPosition[1] = migPosition[1] - 1;}
33      else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
34          migPosition[1] = migPosition[1] - 1;}
35  }
36
37  void migration2 (const int agent_row, const int agent_col,
```

```c
38                      const int rowN, const int colN, int * migPosition) {
39      migPosition[0] = agent_row;
40      migPosition[1] = agent_col;
41      double P = ((double) rand() / (RAND_MAX)); //choose migration directions
42      if ((P < 1.0/6.0) && (agent_row > 0)) {//Position 1
43          migPosition[0] = migPosition[0] - 1;}
44      else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
            < colN-1)) {//Position 2
45          migPosition[0] = migPosition[0] - 1;
46          migPosition[1] = migPosition[1] + 1;}
47      else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-2)) {//
            Position 3
48          migPosition[1] = migPosition[1] + 1;}
49      else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
            agent_col < colN-1)) {//Position 4
50          migPosition[0] = migPosition[0] + 1;
51          migPosition[1] = migPosition[1] + 1;}
52      else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1)) {//
            Position 5
53          migPosition[0] = migPosition[0] + 1;}
54      else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
55          migPosition[1] = migPosition[1] - 1;}
56  }
57
58  void proliferation1 (const int agent_row, const int agent_col,
59                       const int rowN, const int colN, int * proPosition) {
60      proPosition[0] = agent_row;
61      proPosition[1] = agent_col;
62      double P = ((double) rand() / (RAND_MAX)); //choose proliferation
            directions
63      if ((P < 1.0/6.0) && (agent_row > 0) && (agent_col > 0)) {//Position 1
64          proPosition[0] = proPosition[0] - 1;
65          proPosition[1] = proPosition[1] - 1;}
66      else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
            < colN-1)) {//Position 2
67          proPosition[0] = proPosition[0] - 1;}
68      else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-1)) {//
            Position 3
69          proPosition[1] = proPosition[1] + 1;}
70      else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
            agent_col < colN-1)) {//Position 4
71          proPosition[0] = proPosition[0] + 1;}
72      else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1) && (
            agent_col > 0)) {//Position 5
73          proPosition[0] = proPosition[0] + 1;
74          proPosition[1] = proPosition[1] - 1;}
75      else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
76          proPosition[1] = proPosition[1] - 1;}
77  }
78
79  void proliferation2 (const int agent_row, const int agent_col,
80                       const int rowN, const int colN, int * proPosition) {
```

```
81        proPosition[0] = agent_row;
82        proPosition[1] = agent_col;
83        double P = ((double) rand() / (RAND_MAX)); //choose proliferation
                directions
84        if ((P < 1.0/6.0) && (agent_row > 0)) {//Position 1
85            proPosition[0] = proPosition[0] - 1;}
86        else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
                < colN-1)) {//Position 2
87            proPosition[0] = proPosition[0] - 1;
88            proPosition[1] = proPosition[1] + 1;}
89        else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-2)) {//
                Position 3
90            proPosition[1] = proPosition[1] + 1;}
91        else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
                agent_col < colN-1)) {//Position 4
92            proPosition[0] = proPosition[0] + 1;
93            proPosition[1] = proPosition[1] + 1;}
94        else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1)) {//
                Position 5
95            proPosition[0] = proPosition[0] + 1;}
96        else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
97            proPosition[1] = proPosition[1] - 1;}
98    }
99
100   int CalTotal (const int rowN, const int colN) {
101       int Ntotal;
102       if (rowN % 2 == 0) {
103           Ntotal = rowN * colN - rowN/2;
104       } else {
105           Ntotal = rowN * colN - (rowN+1)/2;
106       }
107       return Ntotal;
108   }
109
110   int main(int argc, char **argv) {
111       srand(time(NULL)); //random seeds
112       random_device rd;
113       mt19937 generator(rd());
114       //set clock
115       clock_t TIME;
116       TIME = clock();
117
118       //Initialisation
119       int passageI, passageN, t, tau;
120       double Pm, Pp;
121       passageI = 0; //passage index
122       passageN = 30; //max passage number
123       tau = 1; //time step: 1 ~ 1/12h
124       t = 0; //current time
125       const int d = 24; //spacing: 1 ~ 24mum
126       enum {rowN = 3610, colN = 4168}; //full scale
127       //enum {rowN = 362, colN = 418}; //scale ~ 1/10
```

```
128        const int Ntotal = CalTotal(rowN,colN);
129        const int Nstart = Ntotal * 0.15; //seeding condition
130        const int Nend = Ntotal * 0.85; //85% confluent
131        static vector<vector<int> > domain(rowN, std::vector<int>(colN)); //
               simulation domain
132        static vector<vector<double> > domain_Pp(rowN, std::vector<double>(colN)
               ); //Pp
133        //static vector<vector<int> > domain_passage(rowN, std::vector<int>(colN
               )); //passage
134        //vector<vector<int> > domain_age(rowN, std::vector<int>(colN)); //age
135        //static vector<vector<int> > domain_generation(rowN, std::vector<int>(
               colN)); //generation
136      Pm = 0.35; //migration probability
137      //default_random_engine generator;
138      normal_distribution<double> distribution(4e−3,1e−3); //Pp distribution
139      normal_distribution<double> eps1(2e−5,2e−5); //eps1 distribution
140      normal_distribution<double> eps2(1e−5,1e−6); //eps2 distribution
141      int passageEff = 0;
142      int ageEff = 0;
143      double Pd = 1.0;
144      double Pd2 = 0.0;
145      double avePp;
146
147      ofstream file1;
148      ofstream file2;
149      //ofstream file3;
150      //ofstream file4;
151      //ofstream file5;
152      ofstream file6;
153      ofstream file7;
154      //ofstream file8;
155      //ofstream file9;
156      //ofstream file10;
157      ofstream file11;
158      file1.open("domain.csv");
159      file2.open("domain_Pp.csv");
160      //file3.open("domain_passage.csv");
161      //file4.open("domain_age.csv");
162      //file5.open("domain_generation.csv");
163      file6.open("domain2.csv");
164      file7.open("domain2_Pp.csv");
165      //file8.open("domain2_passage.csv");
166      //file9.open("domain2_age.csv");
167      //file10.open("domain2_generation.csv");
168      file11.open("td.csv");
169
170      //Seeding
171      int agent_row, agent_col;
172      for (int i=Nstart; i>0; −−i) {
173          agent_row = rand() % rowN;
174          agent_col = rand() % colN;
175          if (((rowN−agent_row) % 2 != 0) && (agent_col == colN−1)) {
```

```
176              i = i + 1;
177          } else if (domain[agent_row][agent_col] == 0) {
178              domain[agent_row][agent_col] = 1;
179              domain_Pp[agent_row][agent_col] = distribution(generator);
180              if (domain_Pp[agent_row][agent_col] < 0) {
181                  domain_Pp[agent_row][agent_col] = 0.0;
182              }
183
184          } else {
185              i = i + 1;
186          }
187      }
188
189      //Count initial cells
190      int agent_N = 0;
191      for (int i=0; i<rowN; ++i) {
192          for (int j=0; j<colN; ++j) {
193              if (domain[i][j] == 1) {
194                  agent_N = agent_N + 1;
195              }
196          }
197      }
198      //cout << agent_N;
199
200      //Write to files
201      for (int i=0; i<rowN; ++i) {
202          for (int j=0; j<colN; ++j) {
203              file1 << domain[i][j];
204              file2 << domain_Pp[i][j];
205              //file3 << domain_passage[i][j];
206              //file4 << domain_age[i][j];
207              //file5 << domain_generation[i][j];
208              if (j < colN-1) {
209                  file1 << ", ";
210                  file2 << ", ";
211                  //file3 << ", ";
212                  //file4 << ", ";
213                  //file5 << ", ";
214              } else {
215                  file1 << "\n";
216                  file2 << "\n";
217                  //file3 << "\n";
218                  //file4 << "\n";
219                  //file5 << "\n";
220              }
221          }
222      }
223
224      t = t + tau;
225      int migPosition[2], proPosition[2];
226      double P;
227
```

```
228       while (passageI <= passageN) {//check passage number
229           cout << "passage_number_#" << passageI << ",_";
230           int N = agent_N;
231           cout << "N(0)_=_" << N << endl;
232
233           //check average Pp
234           avePp = 0;
235           for (int i=0; i<rowN; ++i) {
236               for (int j=0; j<colN; ++j) {
237                   if (domain[i][j] == 1) {
238                       avePp = avePp + domain_Pp[i][j];
239                   }
240               }
241           }
242           avePp = avePp/N;
243           cout << avePp << endl;
244           while (N < Nend) { //check cell population
245               //Migration
246               for (int i=0; i<N; ++i) {
247                   P = ((double) rand() / (RAND_MAX)); //generate a random
                          number
248                   if (P <= Pm) { //perform migration
249                       //select a random site
250                       int ifGhost = 0;
251                       agent_row = rand() % rowN;
252                       agent_col = rand() % colN;
253                       while ((domain[agent_row][agent_col] != 1) || (ifGhost
                              == 1)) { //select a random site, stop until finding
                              an agent
254                           agent_row = rand() % rowN;
255                           agent_col = rand() % colN;
256                       }
257                       //test ghost node
258                       if (((rowN-agent_row) % 2 != 0) && (agent_col == colN-1)
                              ) {
259                           ifGhost = 1;
260                       }
261                       if (ifGhost == 0) {
262                           if ((rowN-agent_row) % 2 == 0) {
263                               migration1(agent_row, agent_col, rowN, colN,
                                      migPosition);
264                               if (domain[migPosition[0]][migPosition[1]] == 0)
                                      {//if the new site is not taken
265                                   domain[agent_row][agent_col] = 0; //delete
                                          the previous site
266                                   domain[migPosition[0]][migPosition[1]] = 1;
                                          //move to the new site
267                                   /*if (domain_Pp[migPosition[0]][migPosition
                                          [1]]!= 0 && domain_passage[migPosition
                                          [0]][migPosition[1]]!= 0 && domain_age[
                                          migPosition[0]][migPosition[1]]!= 0 &&
```

```
                                     domain_generation[migPosition[0]][
                                     migPosition[1]]!= 0) {
268                                   cout << "error mig 1" << endl;
269                              }*/
270                              domain_Pp[migPosition[0]][migPosition[1]] =
                                     domain_Pp[agent_row][agent_col]; //move
                                     the corresponding Pp
271                              //domain_passage[migPosition[0]][migPosition
                                     [1]] = domain_passage[agent_row][
                                     agent_col]; //move the corresponding
                                     passage index
272                              //domain_age[migPosition[0]][migPosition[1]]
                                      = domain_age[agent_row][agent_col]; //
                                     move the corresponding age
273                              //domain_generation[migPosition[0]][
                                     migPosition[1]] = domain_generation[
                                     agent_row][agent_col]; //move the
                                     corresponding generation
274                              domain_Pp[agent_row][agent_col] = 0.0; //
                                     delete the previous Pp
275                              //domain_passage[agent_row][agent_col] = 0;
                                     //delete the previous passage index
276                              //domain_age[agent_row][agent_col] = 0; //
                                     delete the previous age
277                              //domain_generation[agent_row][agent_col] =
                                     0; //delete the previous generation
278                              /*if (domain_Pp[agent_row][agent_col]!=0 ||
                                     domain_passage[agent_row][agent_col]!=0
                                     || domain_age[agent_row][agent_col]!=0
                                     || domain_generation[agent_row][
                                     agent_col]!=0) {
279                                   cout << "error mig 1" << endl;
280                              }*/
281                         }
282                    } else {
283                         migration2(agent_row, agent_col, rowN, colN,
                                 migPosition);
284                         if (domain[migPosition[0]][migPosition[1]] == 0)
                                 {//if the new site is not taken
285                              domain[agent_row][agent_col] = 0; //delete
                                     the previous site
286                              domain[migPosition[0]][migPosition[1]] = 1;
                                     //move to the new site
287                              /*if (domain_Pp[migPosition[0]][migPosition
                                     [1]]!= 0 && domain_passage[migPosition
                                     [0]][migPosition[1]]!= 0 && domain_age[
                                     migPosition[0]][migPosition[1]]!= 0 &&
                                     domain_generation[migPosition[0]][
                                     migPosition[1]]!= 0) {
288                                   cout << "error mig 2" << endl;
289                              }*/
```

```
290                                         domain_Pp [ migPosition [ 0 ] ] [ migPosition [ 1 ] ] =
                                                domain_Pp [ agent_row ] [ agent_col ] ;  //move
                                                the  corresponding  Pp
291                                         //domain_passage [ migPosition [ 0 ] ] [ migPosition
                                                [ 1 ] ]  =  domain_passage [ agent_row ] [
                                                agent_col ] ;  ////move  the  corresponding
                                                passage  index
292                                         //domain_age [ migPosition [ 0 ] ] [ migPosition [ 1 ] ]
                                                 =  domain_age [ agent_row ] [ agent_col ] ;  //
                                                move  the  corresponding  age
293                                         //domain_generation [ migPosition [ 0 ] ] [
                                                migPosition [ 1 ] ]  =  domain_generation [
                                                agent_row ] [ agent_col ] ;  //move  the
                                                corresponding  generation
294                                         domain_Pp [ agent_row ] [ agent_col ]  =  0.0 ;  //
                                                delete  the  previous  Pp
295                                         //domain_passage [ agent_row ] [ agent_col ]  =  0;
                                                //delete  the  previous  passage  index
296                                         //domain_age [ agent_row ] [ agent_col ]  =  0;  //
                                                delete  the  previous  age
297                                         //domain_generation [ agent_row ] [ agent_col ]  =
                                                0;  //delete  the  previous  age
298                                         /*if  (domain_Pp [ agent_row ] [ agent_col ]!=0  ||
                                                domain_passage [ agent_row ] [ agent_col ]!=0
                                                ||  domain_age [ agent_row ] [ agent_col ]!=0
                                                ||  domain_generation [ agent_row ] [
                                                agent_col ]!=0)  {
299                                             cout  <<  "error  mig  2"  <<  endl ;
300                                         }*/
301                                     }
302                                 }
303                         } else  {
304                             i  =  i  −  1;
305                         }
306
307                     }
308                 }
309             //Proliferation
310             for  ( int  i=0;  i<N;  ++i )  {
311                 //select  a  random  site
312                 int  ifGhost  =  0;
313                 agent_row  =  rand ( )  %  rowN ;
314                 agent_col  =  rand ( )  %  colN ;
315                 while  (( domain [ agent_row ] [ agent_col ]  !=  1)  ||  ( ifGhost  ==  1)
                        )  {  //select  a  random  site ,  stop  until  finding  an  agent
316                     agent_row  =  rand ( )  %  rowN ;
317                     agent_col  =  rand ( )  %  colN ;
318                 }
319                 //test  ghost  node
320                 if  ((( rowN−agent_row )  %  2  !=  0)  &&  ( agent_col  ==  colN−1))  {
321                     ifGhost  =  1;
322                 }
```

```
323                     if (ifGhost == 0)
324                     {
325                         Pp = domain_Pp[agent_row][agent_col];
326                         P = ((double) rand() / (RAND_MAX)); //generate random
                                number
327                         if (P <= Pp) {
328                             if ((rowN-agent_row) % 2 == 0) {
329                                 proliferation1(agent_row, agent_col, rowN, colN,
                                        proPosition);
330                                 if (domain[proPosition[0]][proPosition[1]] == 0)
                                        {//if the new site is not taken
331                                     domain[proPosition[0]][proPosition[1]] = 1;
                                        //proliferate at the new site
332                                     P = ((double) rand() / (RAND_MAX)); //
                                        generate a random number
333                                     if (ageEff && P<=Pd2) {//damage
334                                         domain_Pp[proPosition[0]][proPosition
                                            [1]] = Pp - max(eps2(generator),
                                            0.0);
335                                         domain_Pp[proPosition[0]][proPosition
                                            [1]] = max(domain_Pp[proPosition
                                            [0]][proPosition[1]], 0.0);
336                                         domain_Pp[agent_row][agent_col] =
                                            domain_Pp[proPosition[0]][
                                            proPosition[1]];
337                                         if (domain_Pp[agent_row][agent_col]<0 ||
                                             domain_Pp[agent_row][agent_col]!=
                                            domain_Pp[proPosition[0]][
                                            proPosition[1]]) {
338                                             cout << "error" << endl;
339                                         }
340                                     } else {//no damage
341                                         domain_Pp[proPosition[0]][proPosition
                                            [1]] = Pp; //adopt Pp
342                                     }
343                                     //domain_passage[proPosition[0]][proPosition
                                        [1]] = passageI; //passage number when
                                        it was born
344                                     //domain_age[proPosition[0]][proPosition[1]]
                                         = -1; //initiate age = -1 so that when
                                        updating cell age it equals 0
345                                     //domain_generation[proPosition[0]][
                                        proPosition[1]] = domain_generation[
                                        agent_row][agent_col] + 1; //add
                                        genertion
346                                     //domain_generation[agent_row][agent_col] =
                                        domain_generation[agent_row][agent_col]
                                        + 1; //add generation
347                                     agent_N = agent_N + 1; //update agent
                                        population
348                                 }
349                             } else {
```

```
350                          proliferation2(agent_row, agent_col, rowN, colN,
                                 proPosition);
351                      if (domain[proPosition[0]][proPosition[1]] == 0)
                             {//if the new site is not taken
352                          domain[proPosition[0]][proPosition[1]] = 1;
                                 //proliferate at the new site
353                          P = ((double) rand() / (RAND_MAX)); //
                                 generate a random number
354                          if (ageEff && P<=Pd2) {//damage
355                              domain_Pp[proPosition[0]][proPosition
                                     [1]] = Pp - max(eps2(generator),
                                     0.0);
356                              domain_Pp[proPosition[0]][proPosition
                                     [1]] = max(domain_Pp[proPosition
                                     [0]][proPosition[1]], 0.0);
357                              domain_Pp[agent_row][agent_col] =
                                     domain_Pp[proPosition[0]][
                                     proPosition[1]];
358                              if (domain_Pp[agent_row][agent_col]<0 ||
                                      domain_Pp[agent_row][agent_col]!=
                                     domain_Pp[proPosition[0]][
                                     proPosition[1]]) {
359                                  cout << "error" << endl;
360                              }
361                          } else {//no damage
362                              domain_Pp[proPosition[0]][proPosition
                                     [1]] = Pp; //adopt Pp
363                          }
364                          //domain_passage[proPosition[0]][proPosition
                                 [1]] = passageI; //passage number when
                                 it was born
365                          //domain_age[proPosition[0]][proPosition[1]]
                                 = -1; //initiate age = -1 so that when
                                 updating cell age it equals 0
366                          //domain_generation[proPosition[0]][
                                 proPosition[1]] = domain_generation[
                                 agent_row][agent_col] + 1; //add
                                 generation
367                          //domain_generation[agent_row][agent_col] =
                                 domain_generation[agent_row][agent_col]
                                 + 1; //add generation
368                          /*if (passageI!=0)
369                          {
370                              cout << "Pp = " << Pp << endl;
371                          }*/
372                          agent_N = agent_N + 1; //update agent
                                 population
373                      }
374                  }
375              }
376          } else {
377              i = i - 1;
```

```
378                    }
379                }
380
381            //update cell age
382            /*for (int i=0; i<rowN; ++i) {
383                for (int j=0; j<colN; ++j) {
384                    if (domain[i][j]==1) { //check if there is a cell
385                        domain_age[i][j] = domain_age[i][j] + tau;
386                    }
387                }
388            }*/
389
390            N = agent_N; //update cell number
391            //cout << "N(" << t << ") = " << N << endl;
392
393            //Write to files
394            if (N >= Nend) {
395                for (int i=0; i<rowN; ++i) {
396                    for (int j=0; j<colN; ++j) {
397                        file6 << domain[i][j];
398                        file7 << domain_Pp[i][j];
399                        //file8 << domain_passage[i][j];
400                        //file9 << domain_age[i][j];
401                        //file10 << domain_generation[i][j];
402                        if (j < colN-1) {
403                            file6 << ", ";
404                            file7 << ", ";
405                            //file8 << ", ";
406                            //file9 << ", ";
407                            //file10 << ", ";
408                        } else {
409                            file6 << "\n";
410                            file7 << "\n";
411                            //file8 << "\n";
412                            //file9 << "\n";
413                            //file10 << "\n";
414                        }
415                    }
416                }
417            }
418
419            t = t + tau;
420
421        }
422        //cout << " DONE N = " << agent_N << endl;
423        cout << "DONE N(" << (t-tau) << ") = " << N << endl;
424        file11 << (t-tau);
425        if (passageI != passageN + 1)
426        {
427            file11 << ", ";
428        }
429
```

```cpp
430            //check average Pp
431            avePp = 0;
432            for (int i=0; i<rowN; ++i) {
433                    for (int j=0; j<colN; ++j) {
434                        if (domain[i][j] == 1) {
435                            avePp = avePp + domain_Pp[i][j];
436                        }
437                    }
438            }
439            avePp = avePp/N;
440            cout << avePp << endl;
441
442            //Subculturing
443            passageI = passageI + 1; //update passage number
444            t = 0; //reset time
445            if (passageI != passageN + 1) {//check if subculture
446                agent_N = 0; //delete current cell number
447                double * Pp_temp;
448                //int * passage_temp;
449                //int * generation_temp;
450                Pp_temp = new double [Nstart];
451                //passage_temp = new int [Nstart];
452                //generation_temp = new int[Nstart];
453                //Select cells
454                for (int i=Nstart; i>0; --i) {
455                    agent_row = rand() % rowN;
456                    agent_col = rand() % colN;
457                    if (((rowN-agent_row) % 2 != 0) && (agent_col == colN-1)) {
458                        i = i + 1;
459                    } else if (domain[agent_row][agent_col] == 1) {
460                                            Pp_temp[Nstart-i] = domain_Pp[
                                                agent_row][agent_col];
461                                            //passage_temp[Nstart-i] =
                                                domain_passage[agent_row][
                                                agent_col];
462                                            //age_temp[Nstart-i] = domain_age[
                                                agent_row][agent_col];
463                                            //generation_temp[Nstart-i] =
                                                domain_generation[agent_row][
                                                agent_col];
464                    } else {
465                        i = i + 1;
466                    }
467                }
468
469                //Delete cells
470                for (int i=0; i<rowN; ++i) {
471                    for (int j=0; j<colN; ++j) {
472                        if (domain[i][j] == 1) {
473                            domain[i][j] = 0;
474                            domain_Pp[i][j] = 0.0;
475                            //domain_passage[i][j] = 0;
```

```
476                        //domain_age[i][j] = 0;
477                        //domain_generation[i][j] = 0;
478                  }
479              }
480          }
481
482              //Shuffle & delete temp values
483              for (int i=Nstart; i>0; --i) {
484                      agent_row = rand() % rowN;
485                      agent_col = rand() % colN;
486                      if (((rowN-agent_row) % 2 != 0) && (
                           agent_col == colN-1)) {
487                              i = i + 1;
488                      } else if (domain[agent_row][agent_col] ==
                               0) {
489                              domain[agent_row][agent_col] = 1;
490          P = ((double) rand() / (RAND_MAX)); //generate a random
                number
491          if (passageEff && P<=Pd) {//damage
492              domain_Pp[agent_row][agent_col] = Pp_temp[Nstart-i]
                   - max(eps1(generator), 0.0);
493              //cout << domain_Pp[agent_row][agent_col] << endl;
494              domain_Pp[agent_row][agent_col] = max(domain_Pp[
                   agent_row][agent_col], 0.0);
495              if (domain_Pp[agent_row][agent_col] < 0) {
496                  cout << "error" << endl;
497              }
498          } else {//no damage
499              domain_Pp[agent_row][agent_col] = Pp_temp[Nstart-i];
500          }
501          //domain_passage[agent_row][agent_col] = passage_temp[
                Nstart-i];
502          //domain_age[agent_row][agent_col] = age_temp[Nstart-i];
503          //domain_generation[agent_row][agent_col] =
                generation_temp[Nstart-i];
504
505                              agent_N = agent_N + 1;
506                      } else {
507                              i = i + 1;
508                      }
509              }
510
511      //Delete arrays
512      delete [] Pp_temp;
513      //delete [] passage_temp;
514      //delete [] generation_temp;
515
516      //Write to files
517      for (int i=0; i<rowN; ++i) {
518          for (int j=0; j<colN; ++j) {
519              file1 << domain[i][j];
520              file2 << domain_Pp[i][j];
```

```
521                          //file3 << domain_passage[i][j];
522                          //file4 << domain_age[i][j];
523                          //file5 << domain_generation[i][j];
524                          if (j < colN-1) {
525                              file1 << ",_";
526                              file2 << ",_";
527                              //file3 << ", ";
528                              //file4 << ", ";
529                              //file5 << ", ";
530                          } else {
531                              file1 << "\n";
532                              file2 << "\n";
533                              //file3 << "\n";
534                              //file4 << "\n";
535                              //file5 << "\n";
536                          }
537                      }
538                  }
539              }
540          }
541
542
543      file1.close();
544      file2.close();
545      //file3.close();
546      //file4.close();
547      //file5.close();
548      file6.close();
549      file7.close();
550      //file8.close();
551      //file9.close();
552      //file10.close();
553      file11.close();
554
555      TIME = clock() - TIME;
556      cout << "It_takes_"<< (float)TIME/CLOCKS_PER_SEC <<"_second(s)."<< endl;
557      cin.get();
558      return 0;
559  }
```

C++ code for scratch assays

```cpp
1  #include <cmath>
2  #include <iostream>
3  #include <vector>
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <fstream>
7  #include <ctime>
8  #include <string>
9  #include <sstream>
10 #include <algorithm>
11 #include <iterator>
12 #include <random>
13
14 using namespace std;
15
16 void migration1 (const int agent_row, const int agent_col,
17                  const int rowN, const int colN, int * migPosition) {
18     migPosition[0] = agent_row;
19     migPosition[1] = agent_col;
20     double P = ((double) rand() / (RAND_MAX)); //choose migration directions
21     if ((P < 1.0/6.0) && (agent_row > 0) && (agent_col > 0)) {//Position 1
22         migPosition[0] = migPosition[0] - 1;
23         migPosition[1] = migPosition[1] - 1;}
24     else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
             < colN-1)) {//Position 2
25         migPosition[0] = migPosition[0] - 1;}
26     else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-1)) {//
             Position 3
27         migPosition[1] = migPosition[1] + 1;}
28     else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
             agent_col < colN-1)) {//Position 4
29         migPosition[0] = migPosition[0] + 1;}
30     else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1) && (
             agent_col > 0)) {//Position 5
31         migPosition[0] = migPosition[0] + 1;
32         migPosition[1] = migPosition[1] - 1;}
33     else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
34         migPosition[1] = migPosition[1] - 1;}
35 }
36
37 void migration2 (const int agent_row, const int agent_col,
38                  const int rowN, const int colN, int * migPosition) {
39     migPosition[0] = agent_row;
40     migPosition[1] = agent_col;
41     double P = ((double) rand() / (RAND_MAX)); //choose migration directions
42     if ((P < 1.0/6.0) && (agent_row > 0)) {//Position 1
43         migPosition[0] = migPosition[0] - 1;}
44     else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
             < colN-1)) {//Position 2
45         migPosition[0] = migPosition[0] - 1;
46         migPosition[1] = migPosition[1] + 1;}
```

```
47        else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-2)) {//
              Position 3
48            migPosition[1] = migPosition[1] + 1;}
49        else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
              agent_col < colN-1)) {//Position 4
50            migPosition[0] = migPosition[0] + 1;
51            migPosition[1] = migPosition[1] + 1;}
52        else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1)) {//
              Position 5
53            migPosition[0] = migPosition[0] + 1;}
54        else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
55            migPosition[1] = migPosition[1] - 1;}
56 }
57
58 void proliferation1 (const int agent_row, const int agent_col,
59                       const int rowN, const int colN, int * proPosition) {
60        proPosition[0] = agent_row;
61        proPosition[1] = agent_col;
62        double P = ((double) rand() / (RAND_MAX)); //choose migration directions
63        if ((P < 1.0/6.0) && (agent_row > 0) && (agent_col > 0)) {//Position 1
64            proPosition[0] = proPosition[0] - 1;
65            proPosition[1] = proPosition[1] - 1;}
66        else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
              < colN-1)) {//Position 2
67            proPosition[0] = proPosition[0] - 1;}
68        else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-1)) {//
              Position 3
69            proPosition[1] = proPosition[1] + 1;}
70        else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
              agent_col < colN-1)) {//Position 4
71            proPosition[0] = proPosition[0] + 1;}
72        else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1) && (
              agent_col > 0)) {//Position 5
73            proPosition[0] = proPosition[0] + 1;
74            proPosition[1] = proPosition[1] - 1;}
75        else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
76            proPosition[1] = proPosition[1] - 1;}
77 }
78
79 void proliferation2 (const int agent_row, const int agent_col,
80                       const int rowN, const int colN, int * proPosition) {
81        proPosition[0] = agent_row;
82        proPosition[1] = agent_col;
83        double P = ((double) rand() / (RAND_MAX)); //choose migration directions
84        if ((P < 1.0/6.0) && (agent_row > 0)) {//Position 1
85            proPosition[0] = proPosition[0] - 1;}
86        else if ((P >= 1.0/6.0) && (P <1.0/3.0) && (agent_row > 0) && (agent_col
              < colN-1)) {//Position 2
87            proPosition[0] = proPosition[0] - 1;
88            proPosition[1] = proPosition[1] + 1;}
89        else if ((P >= 1.0/3.0) && (P < 1.0/2.0) && (agent_col < colN-2)) {//
              Position 3
```

```
90             proPosition[1] = proPosition[1] + 1;}
91         else if ((P >= 1.0/2.0) && (P < 2.0/3.0) && (agent_row < rowN-1) && (
               agent_col < colN-1)) {//Position 4
92             proPosition[0] = proPosition[0] + 1;
93             proPosition[1] = proPosition[1] + 1;}
94         else if ((P >= 2.0/3.0) && (P < 5.0/6.0) && (agent_row < rowN-1)) {//
               Position 5
95             proPosition[0] = proPosition[0] + 1;}
96         else if ((P >= 5.0/6.0) && (agent_col > 0)) {//Position 6
97             proPosition[1] = proPosition[1] - 1;}
98  }
99
100 int CalTotal (const int rowN, const int colN) {
101     int Ntotal;
102     if (rowN % 2 == 0) {
103         Ntotal = rowN * colN - rowN/2;
104     } else {
105         Ntotal = rowN * colN - (rowN+1)/2;
106     }
107     return Ntotal;
108 }
109
110 void importdata (const int N, double * dataPp) {
111         ifstream ifs ("Pp.csv");
112         char dummy;
113         for (int i = 0; i < N; ++i){
114                 ifs >> dataPp[i];
115                 if (i < (N - 1))
116             ifs >> dummy;
117         }
118 }
119
120 int main(int argc, char **argv) {
121     srand(time(NULL)); //random seeds
122     random_device rd;
123     mt19937 generator(rd());
124     //set clock
125     clock_t TIME;
126     TIME = clock();
127
128         //Initialisation
129     int simuNum, sampleNum, t, T, tau, agent_row, agent_col, migPosition[2],
             proPosition[2];
130     double Pm, Pp, P, avePp;
131         simuNum = 1; //simulation index
132     sampleNum = 2; //total simulations
133     tau = 1; //time step: 1 ~ 1/12h
134     t = 0; //current time
135     T = 864; //end time
136     Pm = 0.35;
137     const int d = 24; //spacing: 1 ~ 24mum
138     enum {rowN = 68, colN = 80}; //1400mum by 1900 mum
```

```
139        enum {sCol1 = 29, sCol2 = 51}; //scratch width
140        const int Ntotal = CalTotal(rowN,colN);
141        const int Nstart = Ntotal * 0.3; //seeding condition
142        cout << "Ntotal_=_" << Ntotal << endl;
143        cout << "Nseed_=_" << Nstart << endl;
144        static vector<vector<int> > domain(rowN, std::vector<int>(colN)); //
               simulation domain
145        static vector<vector<double> > domain_Pp(rowN, std::vector<double>(colN)
               ); //Pp
146
147        ofstream file1;
148        ofstream file2;
149        ofstream file3;
150        ofstream file4;
151        ofstream file5;
152        file1.open("domain0.csv");
153        file2.open("domain24.csv");
154        file3.open("domain48.csv");
155            file4.open("domain72.csv");
156            file5.open("domain96.csv");
157        //Import data
158        int size_data = 12790529; //size of import data
159        double * dataPp;
160        dataPp = new double [size_data];
161        importdata(size_data, dataPp);
162
163        //Seeding
164        for (int i = Nstart; i > 0; --i) {
165            agent_row = rand() % rowN;
166            agent_col = rand() % colN;
167            if (((rowN-agent_row) % 2 != 0) && (agent_col == colN-1)) {
168                i = i + 1;
169            } else if (domain[agent_row][agent_col] == 0) {
170                    int randPp = rand() % size_data;
171                domain[agent_row][agent_col] = 1;
172                domain_Pp[agent_row][agent_col] = dataPp[randPp];
173            } else {
174                i = i + 1;
175            }
176        }
177        delete [] dataPp;
178
179        //Scratch
180        for (int i=0; i<rowN; ++i) {
181            if ((rowN-agent_row) % 2 == 0) {
182                for (int j=sCol1; j<=sCol2; ++j) {
183                    domain[i][j] = 0;
184                    domain_Pp[i][j] = 0.0;
185                }
186            } else {
187                for (int j=sCol1; j<=sCol2-1; ++j) {
188                    domain[i][j] = 0;
```

```cpp
189                        domain_Pp[i][j] = 0.0;
190                }
191            }
192        }
193
194        //Count initial cells
195        int agent_N = 0;
196        avePp = 0.0;
197        for (int i=0; i<rowN; ++i) {
198            for (int j=0; j<colN; ++j) {
199                if (domain[i][j] == 1) {
200                    agent_N = agent_N + 1;
201                    avePp = avePp + domain_Pp[i][j];
202                }
203            }
204        }
205        avePp = avePp / (double)agent_N;
206
207        //Write to files
208        for (int i=0; i<rowN; ++i) {
209            for (int j=0; j<colN; ++j) {
210                file1 << domain[i][j];
211                //file2 << domain_Pp[i][j];
212                if (j < colN-1) {
213                    file1 << ", ";
214                    //file2 << ", ";
215                } else {
216                    file1 << "\n";
217                    //file2 << "\n";
218                }
219            }
220        }
221
222        t = t + tau;
223        int timestop = T / t;
224
225        while (simuNum <= sampleNum) {
226            cout << "SimuNum " << simuNum << ", ";
227            cout << "N(0) = " << agent_N << ", Average Pp = " << avePp << endl;
228            for (int tindex=0; tindex<timestop; ++tindex) { //check time
229                    int N = agent_N;
230                    //Migration
231                    for (int i=0; i<N; ++i) {
232                    P = ((double) rand() / (RAND_MAX)); //generate a random
                        number
233                    if (P <= Pm) { //perform migration
234                        //select a random site
235                        int ifGhost = 0;
236                        agent_row = rand() % rowN;
237                        agent_col = rand() % colN;
```

```
238                     while ((domain[agent_row][agent_col] != 1) || (ifGhost
                            == 1)) { //select a random site, stop until finding
                            an agent
239                         agent_row = rand() % rowN;
240                         agent_col = rand() % colN;
241                     }
242                     //test ghost node
243                     if (((rowN-agent_row) % 2 != 0) && (agent_col == colN-1)
                            ) {
244                         ifGhost = 1;
245                     }
246                     if (ifGhost == 0) {
247                         if ((rowN-agent_row) % 2 == 0) {
248                             migration1(agent_row, agent_col, rowN, colN,
                                    migPosition);
249                             if (domain[migPosition[0]][migPosition[1]] == 0)
                                    {//if the new site is not taken
250                                 domain[agent_row][agent_col] = 0; //delete
                                        the previous site
251                                 domain[migPosition[0]][migPosition[1]] = 1;
                                        //move to the new site
252                                 domain_Pp[migPosition[0]][migPosition[1]] =
                                        domain_Pp[agent_row][agent_col]; //move
                                        the corresponding Pp
253                                 domain_Pp[agent_row][agent_col] = 0.0; //
                                        delete the previous Pp
254                             }
255                         } else {
256                             migration2(agent_row, agent_col, rowN, colN,
                                    migPosition);
257                             if (domain[migPosition[0]][migPosition[1]] == 0)
                                    {//if the new site is not taken
258                                 domain[agent_row][agent_col] = 0; //delete
                                        the previous site
259                                 domain[migPosition[0]][migPosition[1]] = 1;
                                        //move to the new site
260                                 domain_Pp[migPosition[0]][migPosition[1]] =
                                        domain_Pp[agent_row][agent_col]; //move
                                        the corresponding Pp
261                                 domain_Pp[agent_row][agent_col] = 0.0; //
                                        delete the previous Pp
262                             }
263                         }
264                     } else {
265                         i = i - 1;
266                     }
267                 }
268             }
269             //Proliferation
270             for (int i=0; i<N; ++i) {
271             //select a random site
272             int ifGhost = 0;
```

```
273                      agent_row = rand() % rowN;
274                      agent_col = rand() % colN;
275                      while ((domain[agent_row][agent_col] != 1) || (ifGhost == 1)
                            ) { //select a random site, stop until finding an agent
276                        agent_row = rand() % rowN;
277                        agent_col = rand() % colN;
278                      }
279                      //test ghost node
280                      if (((rowN-agent_row) % 2 != 0) && (agent_col == colN-1)) {
281                        ifGhost = 1;
282                      }
283                      if (ifGhost == 0) {
284                        Pp = domain_Pp[agent_row][agent_col];
285                        P = ((double) rand() / (RAND_MAX)); //generate a random
                            number
286                        if (P <= Pp) {
287                          if ((rowN-agent_row) % 2 == 0) {
288                            proliferation1(agent_row, agent_col, rowN, colN,
                                proPosition);
289                            if (domain[proPosition[0]][proPosition[1]] == 0)
                                {//if the new site is not taken
290                              domain[proPosition[0]][proPosition[1]] = 1;
                                  //proliferate at the new site
291                              domain_Pp[proPosition[0]][proPosition[1]] =
                                  Pp; //adopt Pp
292                              agent_N = agent_N + 1; //update agent
                                  population
293                            }
294                          } else {
295                            proliferation2(agent_row, agent_col, rowN, colN,
                                proPosition);
296                            if (domain[proPosition[0]][proPosition[1]] == 0)
                                {//if the new site is not taken
297                              domain[proPosition[0]][proPosition[1]] = 1;
                                  //proliferate at the new site
298                              domain_Pp[proPosition[0]][proPosition[1]] =
                                  Pp; //adopt Pp
299                              agent_N = agent_N + 1; //update agent
                                  population
300                            }
301                          }
302                        }
303                      } else {
304                        i = i - 1;
305                      }
306                    }
307                    //Write to files
308                  if (t == 288) { //24h
309                    for (int i=0; i<rowN; ++i) {
310                      for (int j=0; j<colN; ++j) {
311                        file2 << domain[i][j];
312                        if (j < colN-1) {
```

```cpp
313                        file2 << ",_";
314                    } else {
315                        file2 << "\n";
316                    }
317                }
318            }
319        }
320        if (t == 576) { //48h
321            for (int i=0; i<rowN; ++i) {
322                for (int j=0; j<colN; ++j) {
323                    file3 << domain[i][j];
324                    if (j < colN-1) {
325                        file3 << ",_";
326                    } else {
327                        file3 << "\n";
328                    }
329                }
330            }
331        }
332        if (t == 864) { //72h
333            for (int i=0; i<rowN; ++i) {
334                for (int j=0; j<colN; ++j) {
335                    file4 << domain[i][j];
336                    if (j < colN-1) {
337                        file4 << ",_";
338                    } else {
339                        file4 << "\n";
340                    }
341                }
342            }
343        }
344        /*if (t == 1152) { //96h
345            for (int i=0; i<rowN; ++i) {
346                for (int j=0; j<colN; ++j) {
347                    file5 << domain[i][j];
348                    if (j < colN-1) {
349                        file5 << ", ";
350                    } else {
351                        file5 << "\n";
352                    }
353                }
354            }
355        }*/
356            t = t + tau; //forward time
357        }
358
359        cout << "DONE_N_=_" << agent_N << endl;
360        simuNum = simuNum + 1; //update the simulation index
361        t = 0; //reset time
362
363        if (simuNum != sampleNum+1) {
364            for (int i=0; i<rowN; ++i) {
```

```
365              for (int j=0; j<colN; ++j) {
366                  domain[i][j] = 0;
367                  domain_Pp[i][j] = 0.0;
368              }
369          }
370          //Import data
371          double * dataPp;
372              dataPp = new double [size_data];
373              importdata(size_data, dataPp);
374          //Re-seed cells
375              for (int i = Nstart; i > 0; --i) {
376                      agent_row = rand() % rowN;
377                      agent_col = rand() % colN;
378                      if (((rowN-agent_row) % 2 != 0) && (agent_col ==
                             colN-1)) {
379                          i = i + 1;
380                      } else if (domain[agent_row][agent_col] == 0) {
381                              int randPp = rand() % size_data;
382                          domain[agent_row][agent_col] = 1;
383                          domain_Pp[agent_row][agent_col] = dataPp[randPp];
384                      } else {
385                          i = i + 1;
386                      }
387              }
388              delete [] dataPp;
389          //Scratch
390          for (int i=0; i<rowN; ++i) {
391                      if ((rowN-agent_row) % 2 == 0) {
392                      for (int j=sCol1; j<=sCol2; ++j) {
393                              domain[i][j] = 0;
394                              domain_Pp[i][j] = 0.0;
395                      }
396                      } else {
397                      for (int j=sCol1; j<=sCol2-1; ++j) {
398                              domain[i][j] = 0;
399                              domain_Pp[i][j] = 0.0;
400                      }
401                      }
402              }
403          //Count cells
404          agent_N = 0;
405          avePp = 0.0;
406              for (int i=0; i<rowN; ++i) {
407                      for (int j=0; j<colN; ++j) {
408                      if (domain[i][j] == 1) {
409                              agent_N = agent_N + 1;
410                      avePp = avePp + domain_Pp[i][j];
411                      }
412                      }
413              }
414          avePp = avePp / (double)agent_N;
415          //Write to files
```

```cpp
416                for (int i=0; i<rowN; ++i) {
417                    for (int j=0; j<colN; ++j) {
418                        file1 << domain[i][j];
419                        if (j < colN-1) {
420                            file1 << ",";
421                        } else {
422                            file1 << "\n";
423                        }
424                    }
425                }
426            t = t + tau;
427        }
428    }
429
430    file1.close();
431    file2.close();
432    file3.close();
433    file4.close();
434    file5.close();
435    TIME = clock() - TIME;
436    cout << "It takes "<< (float)TIME/CLOCKS_PER_SEC <<" second(s)."<< endl;
437    cin.get();
438    return 0;
439 }
```

6 Conclusions

In this chapter we summarise the results and novel contributions of this work. We then discuss the broad themes of this work. In addition, we outline possible extensions for further investigation. Lastly, we remark the significance of our work.

6.1 Summary

In this thesis, we used a combined mathematical and experimental approach to investigate the sources of variability that affect the reproducibility of *in vitro* cell biology assays. From the modelling perspective, the overall objective of this work is to characterise the variability associated with reproducibility and test the suitability of commonly used mathematical models. In addition, our improved experimental approach allows us to minimise the variables in the experimental design.

The objectives of this thesis were to:

- Investigate how the initial degree of confluence affects the parameter estimates of cell diffusivity and proliferation rate in *in vitro* scratch assays.

- Investigate the cell proliferation mechanisms in *in vitro* cell biology assays.

- Develop a cell proliferation mechanism in lattice–based random walk models to include more biologically realistic crowding effects for *in vitro* cell biology assays.

- Use lattice–based random walk models to investigate the passaging process and how passaging of cell lines affects the reproducibility of *in vitro* cell biology assays.

In Chapter 2 we performed a suite of scratch assays with six different initial seeding conditions, to investigate how the initial degree of confluence affects the reproducibility of scratch assays. Using IncuCyte ZOOM$^{\mathrm{TM}}$ assays [29], we were able to obtain high-resolution experimental images with consistent scratch areas. The experimental images then were further processed to construct cell density profiles for each initial seeding condition. We calibrated the solution of the Fisher–Kolmogorov model to the data of cell

density profiles, to quantify the cell diffusivity D and proliferation rate λ. The results showed that cell diffusivity D varies significantly with the initial degree of confluence. Further calibration of the Porous–Fisher model to the experimental data showed a smaller variation in D over the six initial seeding conditions, suggesting that the Porous–Fisher model might better describe the cell line that we investigated. However, the difference over the six initial seeding conditions is still considerably large. Therefore we suggested that when performing scratch assays, the initial seeding condition should be reported so that the experiments can be better reproduced.

The results in Chapter 2 also showed that cell proliferation rate λ varies over the initial seeding conditions, which motivated us to think whether cells in scratch assays proliferate logistically. To investigate this question, we performed a suite of scratch assays with three different initial seeding conditions in Chapter 3. In addition, we performed a suite of proliferation assays with the same initial seeding conditions as the control experiment. We counted individual cells in selected regions that are far from the scratched area, and plotted the initial cell density for individual replicates. We found that for both scratch assays and proliferation assays, the variation in the initial cell density for each initial seeding condition is large. Our largest recorded initial cell density for the smallest initial seeding condition was larger than the smallest recorded initial cell density for the largest initial seeding condition. We suggested that it is important to measure the initial cell density for each replicate.

One feature of the logistic growth model is that the per capita growth rate is a decreasing straight line. To investigate whether cells proliferate logistically in the scratch assays and proliferation assays, we constructed the per capita growth rates for each initial seeding condition. The per capita growth rate profiles suggested that cells in the proliferation assays follow the logistic growth, while the cell proliferation in the scratch assays is more complicated. Based on the results, we proposed a two–phase growth profile for scratch assays: a disturbance phase at early time, in which cell proliferation is not logistic, and a growth phase at later time, in which cells proliferate logistically. Further data calibration to the logistic growth model showed that simply calibrating the solution of the logistic growth model to experimental data of all time from scratch assays could result in biologically unrealistic parameter estimates. We suggested that when performing scratch assays, the growth phase should be distinguished from the disturbance phase, and the logistic growth model should be applied only to the growth phase.

In Chapter 2 and 3 we investigated both cell migration and cell proliferation in cell biology assays using continuum models. One of our main findings is that cells do not always proliferate logistically. Apart from continuum models, discrete models also have been used to mimic certain features of cell biology assays. However, the traditional lattice–based discrete models involve a cell proliferation mechanism which is based on a simple unbiased exclusion process. This proliferation mechanism leads to the logistic source term in the continuum description. Therefore, using the traditional lattice–based discrete models could lead to misleading predictions and affect the reproducibility of *in silico* experiments.

To improve the proliferation mechanism for lattice–based random walk models, in Chapter 4 we first considered crowding effects with a larger spatial template, so that daughter agents can be placed in non–nearest neighbour sites. We then further generalised the proliferation process, by assuming that the proliferating agents sense the occupancy of non–nearest neighbour sites. Whether potential proliferation events succeed depends on a crowding function which accounts for the occupancy of neighbour sites. The continuum limit description of this discrete model leads to a reaction–diffusion equation. Comparing the solution of the continuum description with averaged simulation data illustrated that generally the continuum description matches the discrete model well. The quality of the match between the continuum description and the discrete model improves with larger sizes of template for neighbour sites. We suggest using the generalised discrete model when investigating non–logistic proliferation and when the size of template of neighbour cells is small, to better predict experimental observations.

Although lattice–based random walk models have been widely used to mimic certain features of *in vitro* cell biology assays, the passaging process has not been studied using any type of discrete model. Previous studies revealed that cell characteristics change at high passage numbers, while the passage number of cells used in experiments is seldom reported. We anticipated that passage number can be a variable that potentially affects the reproducibility of various *in vitro* experiments. In Chapter 5 we used a traditional lattice–based random walk model to mimic the passaging of cell lines as a part of cell culture growth. Our model included the initial heterogeneity in cell proliferation and various amounts of damage during the passaging process. The results showed that depending on the trade–off between the initial heterogeneity and the passage–induced damage, the average cell proliferate rate can either decrease or increase at high passage numbers, as observed in previous experiential studies. We then simulated a suite of scratch assays using cells at low and high passage numbers, and found that the closure rate of the scratched area significantly depends on the passage number.

Overall, using a combined mathematical and experimental approach, we identified three overlooked variables that affect the reproducibility of *in vitro* cell biology assays, namely the initial degree of confluence, the two–phase cell proliferation, and the passage number. Our work highlighted the importance of reporting these three variables in experimental methods in order to make the experiments more reproducible. In addition, we further generalised the proliferation mechanism in lattice–based random walk models to predict experimental observations better.

6.2   Broad themes

6.2.1   Appropriate mathematical models

In mathematical biology, there are certain standard models for cell migration and cell proliferation that are widely used. In particular, there is the continuum model:

$$\frac{\partial C}{\partial t} = -\nabla \cdot \mathbf{J} + f(C), \tag{6.1}$$

which is a combination of the flux of cells $\mathbf{J}$ and the source term $f(C)$. Models of cell biology assays normally use the Fick's law

$$\mathbf{J} = -D\nabla C, \tag{6.2}$$

to represent cell migration, and the logistic growth model

$$f(C) = \lambda C \left(1 - \frac{C}{K} \right), \tag{6.3}$$

to represent cell proliferation. Similarly, there are lattice–based random walk models that mimic cell migration and cell proliferation using a simple unbiased nearest–neighbour exclusion process. The continuum and discrete models are related, such that the continuum limit description of the discrete models lead to Equation (1.1).

In reality, these models are not always appropriate, and one of the themes in this thesis is to explore some simple case studies of these situations. For example, we show that the linear diffusion term in Equation (1.1) is not appropriate for cell migration in scratch assays with a certain cell line (Chapter 2). Also, we show that the logistic growth model is not always appropriate for cell proliferation in scratch assays (Chapter 3). In terms of mathematical modelling, we explore this theme further by showing that if we change the rules in a discrete model to better take into account crowding effects, then the resulting PDE will be different to Equation (1.1).

A simple conclusion is that care must be taken when choosing mathematical models, even for simple applications. Choosing an inappropriate model would lead to misleading results, such that experimental observations can not be reproduced by *in silico* simulations. Therefore, one should not only consider the quality of match, but also the mechanisms and model assumptions.

6.2.2   Reproducibility of experiments

The other core theme in this project is reproducibility of *in vitro* experiments. A startling result of our study in Chapter 2 is that the diffusion coefficient D of cells in the scratch assays was not a constant, but in fact depended on the initial seeding condition (or, in other words, diffusion was not linear). Without this understanding, experiments of

this type are not completely reproducible, since the initial seeding condition is not even typically recorded and experiments are not routinely conducted for different initial seeding conditions.

Our simulation results in Chapter 5 showed that cell proliferation can be significantly affected by the passaging process. Cell proliferation can either increase or decrease with the passage number. Even for cells from the same cell line, the proliferation rate can be very different at different passage numbers. Without reporting the passage number, experiments can be difficult to reproduce.

Our conclusion is that there are many sources of variability that could affect the reproducibility of *in vitro* cell biology assays. Some of variability is associated with experimental design, while others are associated with cell characteristics. It is important to identify and report those factors in experiments to make them reproducible.

6.3 Future Work

- **Investigating how initial degree of confluence affects various cell lines and their two–phase growth in *in vitro* scratch assays.**
  In this thesis we investigated the initial degree of confluence and the proliferation mechanisms using a combined mathematical and experimental approach. In all the experiments we used a prostate cancer cell line. However, previous studies show that different cell lines behave differently in cell biology experiments [88, 100]. Therefore it is still unclear how the initial degree of confluence would affect other cell lines, and whether the disturbance phase occurs in other cell lines. To investigate these questions, we could also use the combined mathematical and experimental approach. From the experimental point of view, we could perform *in vitro* scratch assays with various types of cell lines. Experiments for each cell line contain six different initial seeding conditions. The experimental images could be processed to construct cell density profiles, cell density information and per capita growth rate. For each type of cell line, we then compare the cell density profiles with different initial seeding conditions, to investigate the differences caused by the initial degree of confluence. We could also identify the disturbance phase from the per capita growth rate profiles and investigate when the growth phase starts. From the mathematical point of view, we could calibrate solutions of various reaction–diffusion equations to experimental data of cell density profiles and cell density information, to quantify the parameters and explore the suitability of continuum models for different types of cell lines.

- **Investigating the impact of the shape of scratch on cell migration and cell proliferation**
  Wound healing has been studied for decades to understand how cells react after

being damaged [76, 77, 118]. One of the experimental approaches to mimic wound-
ing healing process is to perform *in vitro* starch assays [14, 26, 71, 85, 86, 91, 118].
Current experimental techniques can create different shapes of scratch in scratch
assays [91]. This allows us to investigate how different shapes of wound affect cells'
ability to close the scratched area. Using a combined mathematical and experi-
mental approach, we could perform scratch assays with different shapes of scratch,
constructing cell density profiles for each case, and then compare the experimental
results with solutions of mathematical models.

- **Investigating the cause of the disturbance phase in *in vitro* scratch as-
  says.**
  In Chapter 3 we identified the two–phase growth in *in vitro* scratch assays, namely
  a disturbance phase, in which cells do not proliferate logistically, and a logistic
  growth phase. We suggested that the disturbance phase needs to be identified and
  ignored, by applying the logistic growth model only to the growth phase. However,
  the cause of the disturbance phase remains unknown. At this stage it is still com-
  pletely unclear whether the disturbance takes place due to mechanical stress [48] or
  chemicals secreted by damaged cells [85, 86]. A combined mathematical and experi-
  mental approach might be applied to investigate the cause of the disturbance phase.

- **Exploring the role of cell–to–cell adhesion on the reproducibility of *in
  vitro* scratch assays.**
  Cell migration is considered one of the key characteristics of cells [109, 118]. In the
  discrete models we used in this thesis, cell migration is represented by the most
  fundamental unbiased nearest–neighbour exclusion mechanism. However, this kind
  of mechanism dose not include any information about cell–to–cell adhesion [109].
  Cell–to–cell adhesion is an important feature of cells that affects the cell motility
  and causes the formation of cell clusters [1, 25, 52, 115]. For some types of cells, such
  as epithelial cells, the effect of cell–to–cell adhesion is strong, so that one can not
  simply ignore it. Therefore, it is necessary to model cell–to–cell adhesion in the
  discrete framework and investigate its impact using a combined mathematical and
  experimental approach.

- **Extending the lattice–based random walk models for passaging of cell
  lines.**
  We developed a discrete model to mimic the passaging of cell lines in Chapter 5.
  In our model we assume that the passaging process only affects cell proliferation.
  However, previous studies reported that cells' ability to migrate also changes at high
  passage numbers [45, 95]. In addition, our discrete model assumes that when pro-
  liferating, daughter cells simply inherit proliferation rate P_p from their mother cell.
  This is biologically unrealistic due to the ageing effect plays an important role in
  cell proliferation [95]. To mimic these two features, we could apply passage–induced
  damage to individual cells' motility during the passaging process. The ageing effects

could be modelled by adding some stochasticity into the proliferation rate P_p when proliferating. With the extended model of passaging process, we could study the trade–off between the initial heterogeneity in cell migration and cell proliferation, the ageing effect, and the passage–induced damage on both cell migration and cell proliferation.

- **Exploring the continuum limit description of lattice–based random walk models that involve initial heterogeneity in cell proliferation.**
  Both traditional lattice–based random walk models and our generalised discrete models in Chapter 4 lead to reaction–diffusion equations at the continuum limit. It is unclear what kind of continuum model the discrete model for passaging of cell lines would lead to. In the discrete model, there are initial heterogeneity in cell proliferation and passage–induced damage during the passaging process. One way of modelling these factors at the continuum limit is to separate fast proliferating cells from slow proliferating cells. Consequently, each type of cells has distinctive average proliferation rate. We anticipate that there will be two continuum equations for the fast and slow proliferating cells.

- **Modelling *in vitro* cell biology assays using off–lattice models.**
  While lattice–based random walk models are routinely used to mimic certain features in *in vitro* cell biology assays [80, 109], other studies claim that off–lattice models are more biologically realistic [28, 52, 89]. Compared to lattice–based models, off–lattice models can mimc the change of cell volume during migration and proliferation. In addition, cells in off–lattice models can migrate or place a daughter cell in any direction, rather than having only a fixed number of directions in lattice–based models. In our experimental images we found that cell volume changes during proliferation. In our lattice–based discrete models we compromise this volume change effect by applying an average cell diameter to all cells. However, this assumption may not be valid when the variation in cell volume change is large. Therefore, off–lattice models can provide insight into some features that lattice–based models can not capture.

- **Providing a software package for simulating *in vitro* cell biology assays**
  In vitro cell biology assays are routinely used to study cancer spreading, drug design and tissue repair [2, 7, 58, 65, 76, 77]. Mathematical models that mimic features of cell biology assays have been applied to predict experimental outcomes [54, 80, 109]. However, there is a lack of software package that systematically models the whole process of cell biology assays. For example, traditional lattice–based models for cell biology assays do not offer the chance to choose cells at different passage numbers. When using those models, one can not choose different functionalities or modules to model different features of the experiment. In addition, current models require knowledge of programming to simulate, which makes them almost impossible for experimentalists to use. It would be useful to develop a user–friendly software package

that simulates cell biology assays. The package would allow users to choose different types of models, features of cells, mechanisms, and output options.

6.4   Final Remarks

In vitro cell biology assays are widely used to investigate various kinds of cell biology questions. However, there are issues associated with the reproducibility of cell biology assays. In this thesis, we identified some of the previously overlooked variability that affects the reproducibility, and investigated the appropriateness of commonly used mathematical models that mimic features in cell biology assays.

Bibliography

[1] Anguige, K., & Schmeiser, C. (2009). A one–dimensional model of cell diffusion and aggregation, incorporating volume filling and cell–to–cell adhesion. *J. Math. Biol.*, 58, 395–427.

[2] Abbott, A. (2003). Cell culture: biology's new dimension. *Nature*, 424, 870–872.

[3] Adobe Systems Incorporated. (2016). Count objects in an image. Retrieved from http://helpx.adobe.com/photoshop/using/counting-objects-image.html

[4] Alarcón, T., Byrne, H. M., & Maini, P. K. (2003). A cellular automaton model for tumour growth in inhomogeneous environment. *J. Theor. Biol.*, 225, 257–274.

[5] Allee, W. C., & Bowen, E. S. (1932). Studies in animal aggregations: mass protection against colloidal silver among goldfishes. *J. Exp. Zool. A Ecol. Genet. Physiol.*, 61, 185–207.

[6] Allegrucci, C., & Young, L. E. (2007). Differences between human embryonic stem cell lines. *Hum. Reprod. Update*, 13, 103–120.

[7] Ashby, W. J., & Zijlstra, A. (2012). Established and novel methods of interrogating two-dimensional cell migration. *Integr. Biol.*, 4, 1338–1350.

[8] ATCC. (2012). Thawing, propagating, and cryopreserving protocol NCI–PBCF–CRL1435 (PC–3) Prostate Adenocarcinoma (ATCC$^{®}$CRL-1435^{TM}). Retrieved from: http://physics.cancer.gov/docs/bioresource/prostate/NCI–PBCF–CRL1435_PC–3_SOP–508.pdf. (Accessed August 2017).

[9] ATCC. (2016). Passage number effects in cell lines. Retrieved from: https://www.atcc.org/~/media/PDFs/Technical%20Bulletins/tb07.ashx. (Accessed August 2017).

[10] Ayati, B. P. (2007). Modeling the role of the cell cycle in regulating *Proteus mirabilis* swarm-colony development. *Appl. Math. Lett.*, 20, 913–918.

[11] Binder, B. J., & Simpson, M. J. (2015). Spectral analysis of pair–correlation bandwidth: application to cell biology images. *R. Soc. Open Sci.*, 2, 140494.

[12] Bramson, M., Calderoni, P., De Masi, A., Ferrari, P., Lebowitz, J., & Schonmann, R. H. (1986). Microscopic selection principle for a diffusion–reaction equation. *J. Stat. Phys.*, 45, 905–920.

[13] Bryant, C. S., Kumar, S., Chamala, S., Shah, J., Pal, J., Haider, M., Seward, S., Qazi, A. M., Morris, R., Semaan, A., Shammas, M. A., Steffes, C., Potti, R. B., Prasad, M., Weaver, D. W., & Batchu, R. B. (2010). Sulforaphane induces cell cycle arrest by protecting RB–E2F–1 complex in epithelial ovarian cancer cells. *Mol. Cancer*, 9, 47.

[14] Cai, A. Q., Landman, K. A., & Hughes, B. D. (2007). Multi–scale modeling of a wound–healing cell migration assay. *J. Theor. Biol.*, 245, 576–594.

[15] Callaghan, T., Khain, E., Sander, L. M., & Ziff, R. M. (2006). A stochastic model for wound healing. *J. Stat. Phys.*, 122, 909–924.

[16] Chan, M. H., & Kim, P. S. (2013). Modelling a Wolbachia invasion using a slow–fast dispersal reaction–diffusion approach. *Bull. Math. Biol.*, 75, 1501–1523.

[17] Chapra, S. C., & Canale, R. P. (2010). *Numerical methods for engineers.* 6th ed. Boston: McGraw–Hill.

[18] Charteris, N., & Khain, E. (2014). Modeling chemotaxis of adhesive cells: stochastic lattice approach and continuum description. *New J. Phys.*, 16, 025002.

[19] Codling, E. A., Plank, M. J., & Benhamou, S. (2008). Random walk models in biology. *J. R. Soc. Interface*, 5, 813–834.

[20] Coleman, T. F., & Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optimiz.*, 6, 418–445.

[21] Cooley, C., Baird, K. M., Faure, V., Wenner, T., Stewart, J. L., Modino, S., Slijepcevic, P., Farr, C. J., & Morrison, C. G. (2009). Trf1 is not required for proliferation or functional telomere maintenance in chicken DT40 cells. *Mol. Biol. Cell*, 20, 2563–2571.

[22] Curtis, C. W., & Bortz, D. M. (2012). Propagation of fronts in the Fisher–Kolmogorov equation with spatially varying diffusion. *Phys. Rev. E*, 86, 066108.

[23] Dale, P. D., Maini, P. K., & Sherratt, J. A. (1994). Mathematical modelling of corneal epithelial wound healing. *Math Biosci.*, 124, 127–147.

[24] Dennis, B., & Taper, M. L. (1994). Density dependence in time series observations of natural populations: estimation and testing. *Ecol. Monogr.*, 64, 205–224.

[25] Deroulers, C., Aubert, M., Badoual, M., & Grammaticos, B. (2009). Modeling tumor cell migration: From microscopic to macroscopic models. *Phys. Rev. E*, 79, 031917.

[26] Doran, M. R., Mills, R. J., Parker, A. J., Landman, K. A., & Cooper–White, J. J. (2009). A cell migration device that maintains a defined surface with no cellular damage during wound edge generation. *Lab Chip*, 9, 2364–2369.

[27] Druckenbrod, N. R., & Epstein, M. L. (2005). The pattern of neural crest advance in the cecum and colon. *Dev. Biol.*, 287, 125–133.

[28] Dunn, S. J., Osborne, J. M., Appleton, P. L., & Näthke I. (2016). Combined changes in Wnt signaling response and contact inhibition induce altered proliferation in radiation–treated intestinal crypts. *Mol. Biol. Cell*, 27, 1863–1874.

[29] EssenBioScience. (2017). IncuCyte ZOOM. Retrieved from http://www.essenbioscience.com/essen-products/incucyte/ (Accessed August 2017)

[30] Fisher, R. A. (1937). The wave of advance of advantageous genes. *Ann. Eugen.*, 7, 353–369.

[31] Foster, S. A., & Galloway, D. A. (1996). Human papillomavirus type 16 E7 alleviates a proliferation block in early passage human mammary epithelial cells. *Oncogene* 12, 1773–1779.

[32] Frascoli, F., Hughes, B. D., Zaman, M. H., & Landman, K. A. (2013). A computational model for collective cellular motion in three dimensions: general framework and case study for cell pair dynamics. *PLOS ONE*, 8, e59249.

[33] Gao, Y., Theng, S. S., Zhuo, J., Teo, W. B., Ren, J., & Lee, C. G. (2014). FAT10, an ubiquitin–like protein, confers malignant properties in non–tumorigenic and tumorigenic cells. *Carcinogenesis*, 35, 923–934.

[34] Gerlee, P., & Anderson, A. R. (2015). The evolution of carrying capacity in constrained and expanding tumour cell populations. *Phys. Biol.*, 12, 056001.

[35] Gough, W., Hulkower, K. I., Lynch, R., Mcglynn, P., Uhlik, M., Yan, L., & Lee, J. A. (2011). A quantitative, facile, and high–throughput image–based cell migration methods is a robust alternative to the scratch assay. *J. Biomol. Screen*, 16, 155–163.

[36] Gracia, J. C., Van den Hoecke, S., Saelens, X., & Van Reeth, K. (2017). Effect of serial pig passages on the adaptation of an avian H9N2 influenza virus to swine. *PLOS ONE*, 12, e0175267.

[37] Gujral, T. S., Chan, M., Peshkin, L., Sorger, P. K., Kirschner, M. W., & MacBeath, G. (2014). A noncanonical frizzled2 pathway regulates epithelial–mesenchymal transition and metastasis. *Cell*, 159, 844–856.

[38] Habbal, A., Barelli, H., & Malandin, G. (2014). Assessing the ability of the 2D Fisher–KPP equation to model cell–sheet wound closure. *Math. Biosci*, 252, 45–59.

[39] Haass, N. K., Beaumont, K. A., Hill, D. S., Anfosso, A., Mrass, P., Munoz, M. A., Kinjyo, I., & Weninger, W. (2014). Real–time cell cycle imaging during melanoma growth, invasion, and drug response. *Pigm. Cell Melanoma R.*, 27, 764–776.

[40] Hammond, J. F., & Bortz, D. M. (2011). Analytical solutions to Fisher's equation with time–variable coefficients. *Appl. Math. Comput*, 218, 2497–2508.

[41] Haridas, P., McGovern, J. A., Kashyap, A. S., McElwain, D. L. S., & Simpson, M. J. (2016). Standard melanoma–associated markers fo not identify the MM127 metastatic melanoma cell line. *Sci. Rep.*, 6, 24569.

[42] Haridas, P., Penington, C. J., McGovern, J. A., McElwain, D. L. S., & Simpson, M. J. (2017) Quantifying rates of cell migration and cell proliferation in co–culture barrier assays reveals how skin and melanoma cells interact during melanoma spreading and invasion. *J. Theor. Biol.*, 423, 13–25.

[43] Harris, S. (2004). Fisher equation with density–dependent diffusion: special solutions. *J. Phys. A*, 37, 6267–6268.

[44] Harvey, D. G., Fletcher, A. G., Osborne, J. M., & Pitt–Francis J. (2015). A parallel implementation of an off–lattice individual–based model of multicellular populations. *Comput. Phys. Commn.*, 192, 130–137.

[45] Hayflick, L. (1965). The limited *in vitro* lifetime of human diploid cell strains. *Exp. Cell Res.*, 37, 614–636.

[46] Hughes, B. D. (1995). *Random Walks and Random Environments. Volume 1: Random Walks.* Oxford: Clarendon Press.

[47] Jackson, P. R., Juliano, J., Hawkins–Daarud, A., Rockne, R., & Swanson, K. R. (2015). Patient–specific mathematical neuro–oncology: Using a simple proliferation and invasion tumor model to inform clinical practice. *Bull. Math. Biol.*, 77, 846–856.

[48] Janet, M. T., Cheng, G., Tyrrell, J. A., Wilcox–Adelman, S. A., Boucher, Y., Jain, R. K., & Munn, L. L. (2012). Mechanical compression drives cancer cells toward invasive phenotype. *P. Natl. Acad. Sci. USA*, 109, 911–916.

[49] Jin, W., Shah, E. T., Penington, C. J., McCue, S. W., Chopin, L. K., & Simpson, M. J. (2016). Reproducibility of scratch assays is affected by the initial degree of confluence: Experiments, modelling and model selection. *J. Theor. Biol.*, 390, 136–145.

[50] Jin, W., Penington, C. J., McCue, S. W., & Simpson, M. J. (2016). Stochastic simulation tools and continuum models for describing two–dimensional collective cell spreading with universal growth functions. *Phys. Biol.*, 13, 056003.

[51] Jin, W., Shah, E. T., Penington, C. J., McCue, S. W., Maini, P. K., & Simpson, M. J. (2017). Logistic proliferation of cells in scratch assays is delayed. *Bull. Math. Biol.*, 79, 1028–1050.

[52] Johnston, S. T., Simpson, M. J., & Plank, M. J. (2013). Lattice–free descriptions of collective motion with crowding and adhesion. *Phys. Rev. E*, 88, 062720.

[53] Johnston, S. T., Simpson, M. J., & McElwain, D. L. S. (2014). How much information can be obtained from tracking the position of the leading edge in a scratch assay? *J. R. Soc. Interface*, 11, 20140325.

[54] Johnston, S. T., Shah, E. T., Chopin, L. K., McElwain, D. L. S., & Simpson, M. J. (2015). Estimating cell diffusivity and cell proliferation rate by interpreting IncuCyte ZOOM$^{\text{TM}}$ assay data using the Fisher–Kolmogorov model. *BMC Syst. Biol.*, 9, 38.

[55] Johnston, S. T., Baker, R. E., McElwain, D. L. S., & Simpson, M. J. (2017). Co–operation, competition and crowding: a discrete framework linking Allee kinetics, nonlinear diffusion, shocks and sharp-fronted travelling waves. *Sci. Rep.*, 7, 42134.

[56] Jones, L. J., Gray, M., Yue, S. T., Haugland, R. P., & Singer, V. L. (2001). Sensitive determination of cell number using the CyQUANT$^{®}$ cell proliferation assay. *J. Immunol. Methods*, 254, 85–98.

[57] Kaighn, M. E., Narayan, K. S., Ohnuki, Y., Lechner, J. F., & Jones, L. W. (1979). Establishment and characterization of a human prostatic carcinoma cell line (PC–3). *Invest. Urol.*, 17, 16–23.

[58] Kam, Y., Guess, C., Estrada, L., Weidow, B., & Quaranta, V. (2008). A novel circular invasion assay mimics *in vivo* invasive behavior of cancer cell lines and distinguishes single–cell motility *in vitro. BMC Cancer*, 8, 198–210.

[59] Kerstein, A. R. (1986). Computational study of propagating fronts in a lattice–gas model. *J. Stat. Phys.*, 45, 921–931.

[60] Khain, E., Katakowski, M., Hopkins, S., Szalad, A., Zheng, X., Jiang, F., & Chopp, M. (2011). Collective behavior of brain tumor cells: the role of hypoxia. *Phys. Rev. E*, 83, 031920.

[61] Khain, E., Katakowski, M., Charteris, N., Jiang, F., & Chopp, M. (2012). Migration of adhesive glioma cells: Front propagation and fingering. *Phys. Rev. E*, 86, 011904.

[62] Kippin, T. E., Martens, D. J., & van der Kooy, D. (2005). p21 loss compromises the relative quiescence of forebrain stem cell proliferation leading to exhaustion of their proliferation capacity. *Gene. Dev.*, 19, 756–767.

[63] Kisiday, J., Jin, M., Kurz, B., Hung, H., Semino, C., Zhang, S., & Grodzinsky, A. J. (2002). Self–assembling peptide hydrogel fosters chondrocyte extracellular matrix production and cell division: implications for cartilage tissue repair. *P. Natl. Acad. Sci. USA*, 99, 9996–10001.

[64] Kolmogorov, A., Petrovsky, I., & Piscounov, N. (1937). Étude de l'équation de la diffusion avec croissance de la quantité de matière et son application á un probléme biologique. *Moscow University Bulletin of Mathematics*, 1, 1–25.

[65] Kramer, N., Walzl, A., Unger, C., Rosner, M., Krupitza, G., Hengstschlager, M., & Dolznig, H. (2013). *In vitro* cell migration and invasion assays. *Mut. Res–Rev. Mutat.*, 752, 10–24.

[66] Laird, A. K. (1964). Dynamics of tumour growth. *Brit. J. Cancer*, 18, 490–502.

[67] Landman, K. A., Cai, A. Q., & Hughes, B. D. (2007). Travelling waves of attached and detached cells in a wound–healing cell migration assay. *Bull. Math. Biol.*, 69, 2119–2138.

[68] Laurent, L. C., Ulitsky, I., Slavin, I., Tran, H., Schork, A., Morey, R., Lynch, C., Harness, J. V., Lee, S., Barrero, M. J., & Ku, S. (2011). Dynamic changes in the copy

number of pluripotency and cell proliferation genes in human ESCs and iPSCs during reprogramming and time in culture. *Cell Stem Cell*, 8, 106–118.

[69] Lewis, M. A., & Kareiva, P. (1993). Allee dynamics and the spread of invading organisms. *Theor. Pop. Biol.*, 43, 141–158.

[70] Neal, M. L., Trister, A. D., Ahn, S., Baldock, A., Bridge, C. A., Guyman, L., Lange, J., Sodt, R., Cloke, T., Lai, A., Coughesy, T. F., Mrugala, M. M., Rockhill, J. K., Rockne, R. C., & Swanson, K. R. (2013). Response classification based on a minimal model of glioblastoma growth is prognostic for clinical outcomes and distinguishes progression from pseudoprogression. *Cancer Res.*, 73, 2976–2986.

[71] Liang, C. C., Park, A. Y., & Guan, J. L. (2007). *In vitro* scratch assay: a convenient and inexpensive method for analysis of cell migration *in vitro. Nat. Protoc.*, 2, 329–333.

[72] Liggett, T. M. (2005). *Interacting Particle Systems.* New York: Springer.

[73] Lin, H. K., Hu, Y. C., Yang, L., Altuwaijri, S., Chen, Y. T., Kang, H. Y., Chang, C. (2003). Suppression versus induction of androgen receptor functions by the phosphatidylinositol 3–kinase/Akt pathway in prostate cancer LNCaP cells with different passage numbers. *J. Biol. Chem.* 278, 50902–50907.

[74] Louis, K. S., & Siegel, A. C. (2011). Cell Viability Analysis Using Trypan Blue: Manual and Automated Methods. In: J. M. Stoddart (Ed.), *Mammalian Cell Viability: Methods and Protocols (pp. 7–12).* Totowa, NJ: Humana Press.

[75] Mahasa, K. J., Ouifki, R., Eladdadi, A., & de Pillis, L. (2016). Mathematical model of tumor–immune surveillance. *J. Theor. Biol.*, 404, 312–330.

[76] Maini, P. K., McElwain, D. L. S., & Leavesley, D. I. (2004a). Traveling wave model to interpret a wound-healing cell migration assay for human peritoneal mesothelial cells. *Tissue Eng.*, 10, 475–482.

[77] Maini, P. K., McElwain, D. L. S., & Leavesley, D. I. (2004b). Travelling waves in a wound healing assay. *Appl. Math. Lett.*, 17, 575–580.

[78] Mallet, D. G., & De Pillis, L. G. (2006). A cellular automata model of tumor–immune system interactions. *J. Theor. Biol.*, 239, 334–350.

[79] MathWorks. (2016). Solve nonlinear curve–fitting (data–fitting) problems in least–squares sense. Retrieved from http://au.mathworks.com/help/optim/ug/lsqcurvefit.html

[80] Mort, R. L., Ross, R. J., Hainey, K. J., Harrison, O. J., Keighren, M. A., Landini, G., Baker, R. E., Painter, K. J., Jackson, I. J., & Yates, C. A. (2016). Reconciling diverse mammalian pigmentation patterns with a fundamental mathematical model. *Nat. Commun.*, 7, 10288.

[81] Morton, K. W., & Mayers, D. F. (2005). *Numerical solution of partial differential equations.* Cambridge: Cambridge University Press.

[82] Murray, J. D. (2002). *Mathematical Biology I: An Introduction.* Heidelberg: Springer.

[83] Nardini, J. T., Chapnick, D. A., Liu, X., & Bortz, D. M. (2015). The effects of MAPK activity on cell–cell adhesion during wound healing. *arXiv*, 1506.07081.

[84] Nauman, J. V., Campbell, P. G., Lanni, F., & Anderson, J. L. (2007). Diffusion of insulin–like growth factor–I and ribonuclease through fibrin gels. *Biophys. J.*, 92, 4444–4450.

[85] Nikolić, D. L., Boettiger, A. N., Bar–Sagi, D., Carbeck, J. D., & Shvartsman, S. Y. (2006). Role of boundary conditions in an experimental model of epithelial wound healing. *Am. J. Physiol. Cell Physiol.*, 291, C68–C75.

[86] Nishio, T., Kawaguchi, S., Yamamoto, M., Iseda, T., Kawasaki, T., & Hase, T. (2005). Tenascin–C regulates proliferation and migration of cultured astrocytes in a scratch wound assay. *Neuroscience*, 132, 87–102.

[87] Petrovskii, S., Mashanova, A., & Jansen, V. A. A. (2011). Variation in individual walking behaviour creates the impression of a Levy flight. *P. Natl. Acad. Sci. USA*, 108, 8704–8707.

[88] Pfeiffer, M. J., & Schalken, J. A. (2010). Stem cell characteristics in prostate cancer cell lines. *Eur. Urol.*, 57, 246–255.

[89] Plank, M. J., & Simpson, M. J. (2012). Models of collective cell behaviour with crowding effects: comparing lattice-based and lattice–free approaches. *J. R. Soc. Interface*, 9, 2983–2996.

[90] Public Health England. (2017). Passage numbers explained. Public Health England. Retrieved from: https://www.phe-culturecollections.org.uk/media/114565/m219_passage-numbers-explained.pdf. (Accessed August 2017).

[91] Riahi, R., Yang, Y., Zhang, D. D., & Wong, P. K. (2012). Advances in wound–healing assays for probing collective cell migration. *J. Lab. Autom.*, 17, 59–65.

[92] Ribba, B., Colin, T., & Schnell, S. (2006). A multiscale mathematical model of cancer, and its use in analyzing irradiation therapies. *Theor. Biol. Med. Model*, 3, 7.

[93] Rockne, R. C., Trister, A. D., Jacobs, J., Hawkins–Daarud, A. J., Neal, M. L., Hendrickson, K., Mrugala, M. M., Rockhill, J. K., Kinahan, P., Krohn, K. A., & Swanson, K. R. (2014). A patient–specific computational model of hypoxia–modulated radiation resistance in glioblastoma using ^{18}F–FMISO–PET. *J. R. Soc. Interface*, 12, 20141174.

[94] Roques, L., Garnier, J., Hamel, F., & Klein, E. K. (2012). Allee effect promotes diversity in traveling waves of colonization. *Proc. Natl. Acad. Sci. USA.*, 109, 8828–8833.

[95] Rubin, H. (1997). Cell aging *in vivo* and *in vitro. Mech. Ageing. Dev.*, 98, 1–35.

[96] Ruttala, H. B., Ko, Y. T. (2015). Liposomal co–delivery of curcumin and albumin/paclitaxel nanoparticle for enhanced synergistic antitumor efficacy. *Colloid Surface B*, 128, 419–426.

[97] Sarapata, E. A., & de Pillis, L. G. (2014). A comparison and catalog of intrinsic tumor growth models. *Bull. Math. Biol.*, 76, 2010–2024.

[98] Savla, U., Olson, L. E., & Waters, C. M. (2004). Mathematical modeling of airway epithelial wound closure during cyclic mechanical strain. *J. Appl. Phys.*, 96, 566–574.

[99] Schneider, E. L., & Mitsui, Y. (1976). The relationship between in vitro cellular aging and *in vivo* human age. *P. Natl. Acad. Sci. USA*, 73, 3584–3588.

[100] Sengers, B. G., Please, C. P., & Oreffo, R. O. C. (2007). Experimental characterization and computational modelling of two–dimensional cell spreading for skeletal regeneration. *J. R. Soc. Interface*, 4, 1107–1117.

[101] Sewalt, L., Harley, K., van Heijster, P., & Balasuriya, S. (2016). Influences of Allee effects in the spreading of malignant tumours. *J. Theor. Biol.*, 394, 77–92.

[102] Shakeel, M., Matthews, P. C., Graham, R. S., & Waters, S. L. (2013). A continuum model of cell proliferation and nutrient transport in a perfusion bioreactor. *Math. Med. Biol.*, 30, 21–44.

[103] Sheardown, H., & Cheng, Y. L. (1996). Mechanisms of corneal epithelial wound healing. *Chem. Eng. Sci.*, 51, 4517–4529.

[104] Sherratt, J. A., & Murray, J. D. (1990). Models of epidermal wound healing. *Proc. R. Soc. B*, 241, 29–36.

[105] Sherratt, J. A., & Marchant, B. P. (1996). Nonsharp travelling wave fronts in the Fisher equation with degenerate nonlinear diffusion. *Appl. Math. Lett.*, 9, 33–38.

[106] Simpson, M. J., Landman, K. A., & Bhaganagarapu, K. (2007). Coalescence of interacting cell populations. *J. Theor. Biol.*, 247, 525–543.

[107] Simpson, M. J., Landman, K. A., & Hughes, B. D. (2009). Multi–species simple exclusion processes. *Physica A*, 388, 399–406.

[108] Simpson, M. J. (2009). Depth–averaging errors in reactive transport modeling. *Water Resour. Res.*, 45, W02505.

[109] Simpson, M. J., Landman, K. A., & Hughes, B. D. (2010). Cell invasion with proliferation mechanisms motivated by time–lapse data. Physica A, 389, 3779–3790.

[110] Simpson, M. J., Baker, R. E., & McCue, S. W. (2011). Models of collective cell spreading with variable cell aspect ratio: A motivation for degenerate diffusion models. *Phys. Rev. E*, 83, 021901.

[111] Simpson, M. J., Binder, B. J., Haridas, P., Wood, B. K., Treloar, K. K., McElwain, D. L. S., & Baker, R. E. (2013). Experimental and modelling investigation of monolayer development with clustering. *Bull. Math. Biol.*, 75, 871–889.

[112] Simpson, M. J., Sharp, J. A., & Baker, R. E. (2014). Distinguishing between mean–field, moment dynamics and stochastic descriptions of birth–death–movement processes. *Physica A*, 395, 236–246.

[113] Stoscheck, C. M., & Carpenter, G. (1984). Down regulation of epidermal growth factor receptors: Direct demonstration of receptor degradation in human fibroblasts. *J. Cell Biol.*, 98, 1048–1053.

[114] Taylor, C. M., & Hastings, A. (2005). Allee effects in biological invasions. *Ecol. Lett.*, 8, 895–908.

[115] Treloar, K. K., Simpson, M. J., Haridas, P., Manton, K. J., Leavesley, D. I., McElwain, D. L. S., & Baker, R. E. (2013). Multiple types of data are required to identify the mechanisms influencing the spatial expansion of melanoma cell colonies.

BMC Syst. Biol., 7, 137.

[116] Treloar, K. K., & Simpson, M. J. (2013). Sensitivity of edge detection methods for quantifying cell migration assays. *PLOS ONE*, 8, e67389.

[117] Treloar, K. K., Simpson, M. J., McElwain, D. L. S., & Baker, R. E. Are *in vitro* estimates of cell diffusivity and cell proliferation rate sensitive to assay geometry? *J. Theor. Biol.*, 356, 71–84.

[118] Tremel, A., Cai, A., Tirtaatmadja, N., Hughes, B. D., Stevens, G. W., Landman, K. A., & O?Connor, A. J. (2009). Cell migration and proliferation during monolayer formation and wound healing. *Chem. Eng. Sci.*, 64, 247–253.

[119] Tsoularis, A., & Wallace, J. (2002). Analysis of logistic growth models. *Math. Biosci.*, 179, 21–55.

[120] Vasilevsky, N. A., Brush, M. H., Paddock, H., Ponting, L., Tripathy, S. J., LaRocca, G. M., & Haendel, M. A. (2013). On the reproducibility of science: unique identification of research resources in the biomedical literature. *PeerJ*, 1, e148.

[121] Vichai, V., & Kirtikara, K. (2006). Sulforhodamine B colorimetric assay for cytotoxicity screening. *Nat. Protoc.*, 1, 1112–1116.

[122] Vo, B. N., Drovandi, C. C., Pettitt, A. N., & Simpson, M. J. (2015). Quantifying uncertainty in parameter estimates for stochastic models of collective cell spreading using approximate Bayesian computation. *Math. Biosci.*, 263, 133–142.

[123] Waters, E. K., Sidhu, H. S., Sidhu, L. A., & Mercer, G. N. (2015). Extended Lotka–Volterra equations incorporating population heterogeneity: Derivation and analysis of the predator–prey case. *Ecol. Model*, 297, 187–195.

[124] West, G. B., Brown, J. H., & Enquist, B. J. (2001). A general model for ontogenetic growth. *Nature*, 413, 628–631.

[125] Witelski, T. P. (1994). An asymptotic solution for traveling waves of a nonlinear–diffusion Fisher's equation. *J. Math. Biol.*, 33, 1–16.

[126] Witelski, T. P. (1995). Merging traveling waves for the Porous–Fisher's equation. *Appl. Math. Lett.*, 4, 57–62.

[127] Yang, Q., Jia, C., Wang, P., Xiong, M., Cui, J., Li, L., Wang, W., Wu, Q., Chen, Y., & Zhang, T. (2014). MicroRNA–505 identified from patients with essential hypertension impairs endothelial cell migration and tube formation. *Int. J. Cardiol.*,

177, 925–934.

[128] Zetterberg, A., & Auer, G. (1970). Proliferative activity and cytochemical properties of nuclear chromatin related to local cell density of epithelial cells. *Exp. Cell Res.*, 62, 262–270.

[129] Zhang, H., Zhu, N. X., Huang, K., Cai, B. Z., Zeng, Y., Xu, Y. M., Liu, Y., Yuan, Y. P., & Lin, C. M. (2016). iTRAQ–Based Quantitative Proteomic Comparison of Early–and Late–Passage Human Dermal Papilla Cell Secretome in Relation to Inducing Hair Follicle Regeneration. *PLOS ONE*, 11, e0167474.

[130] Zwietering, M. H., Jongenburger, I., Rombouts, F. M., & Van't Riet, K. (1990). Modeling of the bacterial growth curve. *Appl. Environ. Microbiol.*, 56, 1875–1881.