

Aus dem Zentrum für Kinder -und Jugendmedizin der Universität Heidelberg
(Geschäftsführender Direktor: Univ. - Prof. Dr. med., Prof. h. c. (RCH) Georg F. Hoffmann)
Klinik für allgemeine Pädiatrie, Neuropädiatrie, Stoffwechsel, Gastroenterologie, Nephrologie
(Ärztlicher Direktor: Univ. - Prof. Dr. med., Prof. h. c. (RCH) Georg F. Hoffmann)

Modelling of Cell Movements and Aggregations During Early Nephrogenesis

Inauguraldissertation
zur Erlangung des Doctor scientiarum humanarum (Dr. sc. hum.)
an der
Medizinischen Fakultät Heidelberg
der
Ruprecht-Karls-Universität

vorgelegt von
Pauli Tikka

aus
Helsinki, Finnland

2019

Dekan: Prof. Dr. med. Hans-Georg Kräusslich

Doktorvater: Prof. Dr. med. Franz Schaefer

CONTENTS

ABBREVIATIONS	1
1. INTRODUCTION.....	3
1.1. Discovering Early Phases of Kidney Development	3
1.1.1. Mammalian Kidney Development.....	3
1.1.2. Formation of Pre-tubular Aggregate and Renal Vesicle.....	4
1.2. Early Nephrogenesis	4
1.2.1. Mechanisms Effecting Cell Patterns in Early Nephrogenesis	4
1.2.2. Chemotaxis and Cell-cell Adhesion	5
1.3. Modelling.....	6
1.3.1. Computational Models	6
1.3.2. Models in Kidney Development.....	6
1.3.3. Initial Computational Models for Directed Cell Movements and Aggregations	8
1.4. Study Objectives	9
2. MATERIALS AND METHODS	10
2.1. Model Building	10
2.1.1. Cellular Potts Model.....	10
2.1.2. Model Construction.....	12
2.2. Model Simulation Studies	14
2.2.1. Model Variants	14
2.2.2. Selection of Chemoattractant Saturation Type	18
2.2.3. Model Simulations	19
2.3. Computational Estimation of Model Parameters	21
2.3.1. Initial Estimation of Parameters	21
2.3.2. Particle Swarm Optimization of Parameters.....	23
2.4. Particle Swarm Optimization	24
2.4.1. Algorithm of Particle Swarm Optimization.....	24
2.4.2. Constraints and Ranges of Parameters in Particle Swarm Optimization	26
2.5. Experimental Design.....	27
2.5.1. Experimental Data Sources	27
2.5.2. Kidney Organoids.....	28
2.6. Technical Workflow of Analysis	29
2.7. Analysing Data with Self-Organizing Maps	32
2.7.1. Introduction	32
2.7.2. Self-Organizing Maps	32
3. RESULTS.....	35
3.1. Initial Model Parameter Estimation	35
3.2. Formation of Patterns.....	36
3.3. Experimental Studies	39
3.4. Comparing Cell Behaviours Between Models and Experiments	41
3.4.1. Estimation of Final Parameters with Particle Swarm Optimization	41
3.4.2. Speeds of Cells at Tip and Corner Regions	42
3.4.3. Self-Organizing Maps Applied to Speeds of Cells.....	45

3.4.4.	Quantities of Cells at Tip and Corner Regions	46
3.4.5.	Distances of Cells to Tip and Corner.....	49
3.4.6.	Chemoattractants in Cells at Tip and Corner Regions.....	50
4.	DISCUSSION	54
5.	SUMMARY	58
6.	ZUSAMMENFASSUNG	59
7.	REFERENCES	60
8.	SUPPLEMENT	70
8.1.	Description of Previous Model	70
8.2.	Example Ranges of Parameters and Simulation Lengths During Optimization.....	71
8.3.	Codes.....	72
8.3.1.	Example of Dynamic CompuCell3D Coding File	72
8.3.2.	Shortened Python Code of Particle Swarm Optimization Method	73
8.3.3.	Shortened Python Code of 2D and 3D Data Analysis	74
8.3.4.	Shortened Python Code of Self-Organized Maps.....	78
8.4.	Tables.....	80
9.	PUBLICATIONS AND PRESENTATIONS	83
10.	CURRICULUM VITAE.....	84
11.	ACKNOWLEDGEMENTS.....	85
12.	EIDESSTATTLICHE VERSICHERUNG	86

ABBREVIATIONS

$\alpha(s_i)$	learning coefficient
γ	global decay constant
δ	boolean switch designating cell types
$\theta(n, v, s_i)$	neighbourhood function
λ_A	surface lambda
λ_{CL}	chemotaxis lambda
λ_V	lambda volume
$\sigma_{\bar{i}}$	virtual cell by type (\bar{i})
$\tau(\sigma_{\bar{i}}), \tau(\sigma_{\bar{j}})$	cell pixel interface between two cell types ($\sigma_{\bar{i}}, \sigma_{\bar{j}}$)
χ	quality value for speed differences
ω	inertia weight
A_t	target surface area
AED	adhesion differences
BMU	best matching unit
c_n	positive constant parameter by index n
$c(\vec{x})$	concentrations of chemoattractant in coms of cells
CA	cellular automaton model
CAM	cell adhesion molecule
CAMA	computer assisted motion analysis
CI	confidence interval
COM	centre of mass
CC3D	compuCell3d program
CM	cap mesenchyme
CPM	cellular potts model
d	distance
D	global diffusion constant
D(t)	data vector with index
E	energy term
f	fitness function
FGF	fibroblast growth factor
gbest	particle's direction to best solution regarding any particle
GFP	green fluorescent protein
H	effective energy
i	particle's index
J	contact energy coefficient
k	time index k
k(d)	velocity with distance in specific model
l	lattice's voxel scale
MATLAB	mathematical laboratory program
MCS	monte carlo step
MM	metanephric mesenchyme
MMTV	mouse mammary tumor virus
N	total number of particles
n	previous iteration node
NP	nephron progenitor
pbest	particle's direction from current to best solution
P	position

PDGF-AA	platelet derived growth factor subunit aa
PSO	particle swarm optimization
PTA	pre-tubular aggregate
r	uniformly distributed random variable
RV	renal vesicle
s	saturation constant
si	iteration step index
S	chemoattractant secretion
SBML	systems biology markup language
SOM	self-organizing map
t	index number of training sample
t _c	current iteration number
T	cell surface motility or temperature at cellular potts model
TGF-β	transforming growth factor β
u	index number of bmu
UB	ureteric bud
v	node
v(d)	instantaneous velocity
V	velocity
V _t	target volume
WNT	wingless-type mmtv integration site
W _v	weight vector of node
y(d,t)	distances of cells to tip in certain time
1_REF	reference model (1)
2_REF_ADH	reference model with adhesion (2)
3_UB	ureteric bud secreting model (3)
4_NP	nephron progenitor cell secreting model (4)
5_UB_ADH	ureteric bud secreting model with adhesion (5)
6_NP_ADH	nephron progenitor cell secreting model with adhesion (6)
7_UB_NP_ADH	ureteric bud and nephron progenitor cell secreting model with adhesion (7)
8_ADH_ADH	multi-adhesion model (8)

1. INTRODUCTION

1.1. Discovering Early Phases of Kidney Development

1.1.1. Mammalian Kidney Development

The mammalian kidney is the product of a highly complex, orchestrated developmental process, which involves not only proliferation and differentiation processes, but also directed movement and aggregation of progenitor cells (Krause et al., 2015). Early nephrogenesis is characterized by the interplay of the branching and expanding ureteric bud (UB), the epithelial precursor structure destined to become the urinary tract, and the ‘cap’ metanephric mesenchyme (CM) surrounding the tips of the UB branches (Fig. 1) (BioPortal, 2019; Blake and Rosenblum, 2014; Bohnenpoll and Kispert, 2014; Costantini and Kopan, 2010; Desgrange and Cereghini, 2015; Obara-Ishihara T, 1999).

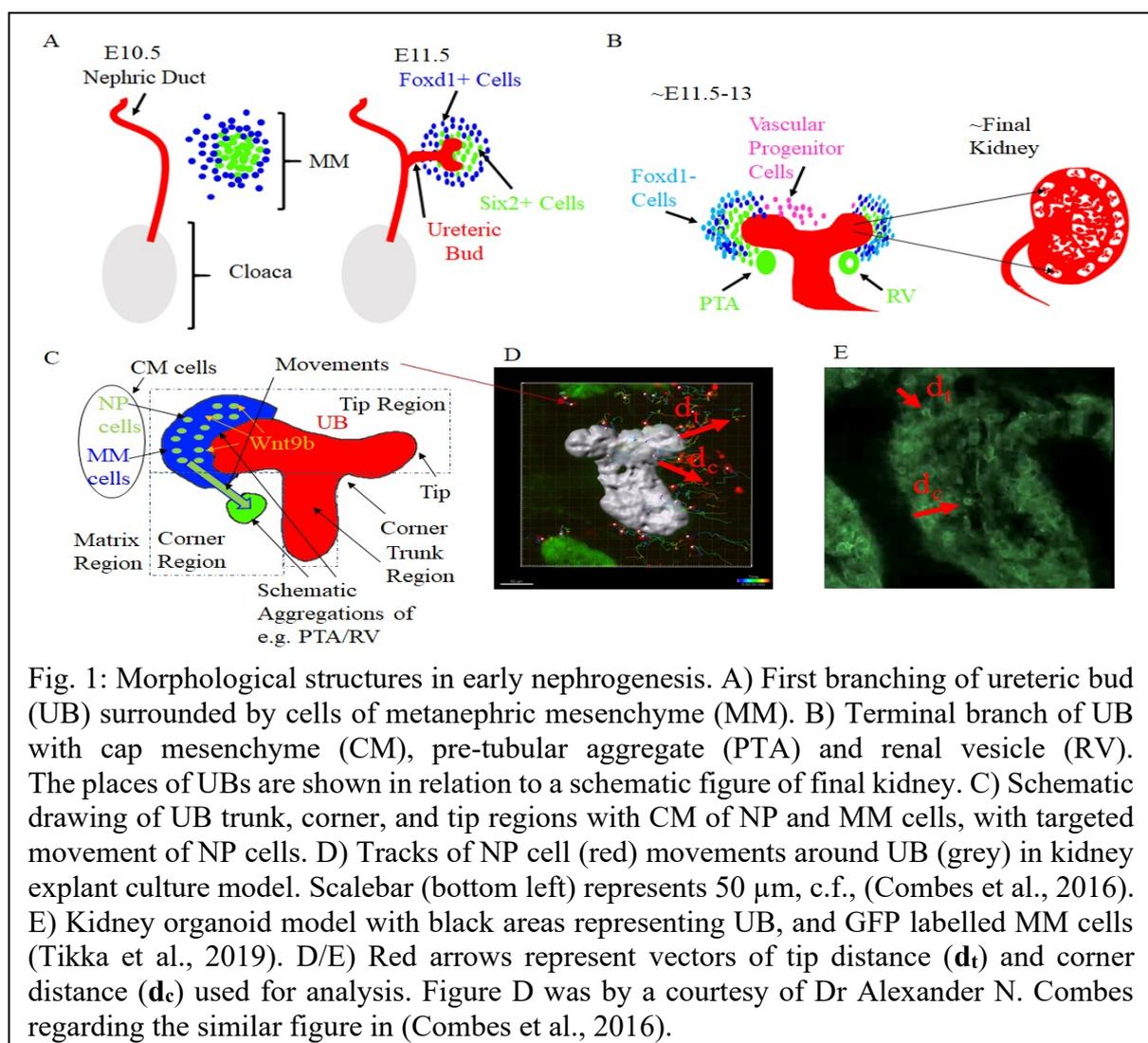


Fig. 1: Morphological structures in early nephrogenesis. A) First branching of ureteric bud (UB) surrounded by cells of metanephric mesenchyme (MM). B) Terminal branch of UB with cap mesenchyme (CM), pre-tubular aggregate (PTA) and renal vesicle (RV). The places of UBs are shown in relation to a schematic figure of final kidney. C) Schematic drawing of UB trunk, corner, and tip regions with CM of NP and MM cells, with targeted movement of NP cells. D) Tracks of NP cell (red) movements around UB (grey) in kidney explant culture model. Scalebar (bottom left) represents 50 μm , c.f., (Combes et al., 2016). E) Kidney organoid model with black areas representing UB, and GFP labelled MM cells (Tikka et al., 2019). D/E) Red arrows represent vectors of tip distance (d_t) and corner distance (d_c) used for analysis. Figure D was by a courtesy of Dr Alexander N. Combes regarding the similar figure in (Combes et al., 2016).

1.1.2 Formation of Pre-tubular Aggregate and Renal Vesicle

A fraction of CM cells differentiates into the nephron progenitor (NP) cells (Fig. 1). These cells migrate towards the corner of the UB branches, where they condensate to form circular pre-tubular aggregates (PTA), the earliest recognizable structure of the kidney (Little, 2012). PTA transforms into the renal vesicle (RV), from which the final structures of the nephron are derived in a process of elongation and invagination (Fig. 1) (BioPortal, 2019; Combes et al., 2016). The nephron in essence consists of the tubules and glomeruli. From the developmental biology point of view, nephron patterning reflects a time-dependent process of recruitment of mesenchymal progenitors into an epithelial nephron precursor (Lindström et al., 2018). The self-renewal and differentiation of NP cells is regulated by spatially restricted cues within a tissue microenvironment (Lawlor et al., 2019). The portion of MM cells that become NP cells can change their commitment status dependent on their location (Lawlor et al., 2019).

1.2. Early Nephrogenesis

1.2.1. Mechanisms Effecting Cell Patterns in Early Nephrogenesis

While early nephrogenesis research presumed that the movement of NPs towards the PTA is mostly linear (Little, 2012; McMahon, 2016), recent studies indicate that NP movements may be semi-stochastic or swarming-like (Combes et al., 2016). It was also described that these movements were mostly driven by adhesion differences and/or chemotaxis (Chen et al., 2015; Combes et al., 2016). Previous experimental work identified several mechanical and signalling mechanisms to be involved in this process (Fig. 1) (Chen et al., 2015; Chi et al., 2009; Combes et al., 2016; Lawlor et al., 2019; Little, 2015; Little, 2012; McMahon, 2016). The molecular mechanisms driving NP cell induction and PTA formation have been partially unravelled (Chen et al., 2015; Chi et al., 2009; Combes et al., 2016; Little, 2015; Little, 2012). UB epithelial cells secrete various diffusible signalling proteins that may trigger the differentiation of MM to NP cells, as well as their chemotactic movement towards the UB corner region (Combes et al., 2016; Little, 2015; Saarela et al., 2017). Furthermore, the cell aggregation appeared to be driven by the differences in the cell-cell adhesion properties (Lefevre et al., 2017), which could be driven by the autocrine and/or paracrine intercellular signalling (Dahl et al., 2002; Dudley et al., 1999; Oxburgh et al., 2011; Wallner et al., 1998). Paracrine cell signalling factors reach targets in a short range, whereas the autocrine ones reach farther distances. The former factors employ different receptors and pathways, such as Wingless-Type Mouse Mammary Tumor Virus (MMTV) Integration Sites (WNT), Fibroblast Growth Factors

(FGF), Platelet Derived Growth Factor Subunit AA (PDGF-AA), and Transforming Growth Factor β s (TGF- β) (Gilbert, 2000).

1.2.2. Chemotaxis and Cell-cell Adhesion

NP cells in my study move towards the higher concentration of the potential chemoattractant source (positive chemotaxis) (Rappel and Loomis, 2009). The chemotactic eukaryotic cells usually move in a crawling fashion that consists of three phases: the directional sensing (<10s), the formation of a stable leading edge with cell polarity (10-60s), and the cell motility (>60s) (Rappel and Loomis, 2009).

A number of essential components for the formation of polarity have been identified (Affolter and Weijer, 2005; Williams and Harwood, 2003). In reality, the polarization does not necessarily require a gradient stimulus (Postma et al., 2003). Spontaneous polarization has been observed in the absence of a gradient. In that case, the polarity module works even when it does not receive a direct input from the directional sensing module.

On the contrary, the motility arrives from the use of polymerization dynamics of actins, which comes from the adhesive contacts with the substrate (Pollard and Borisy, 2003; Rafelski and Theriot, 2004). Moreover, these contacts are not affected by chemotactic signalling (Andrew and Ewald, 2010). In this way, the movement is also influenced by the dissolution of extracellular membrane proteins synthesized by MM cells during early nephrogenesis (Chen et al., 2015; McMahon, 2016). This dissolution was described to arrive from spatial arrangements of the cells either in the tip or the corner region (Chen et al., 2015; Lawlor et al., 2019).

1.3. Modelling

1.3.1. Computational Models

Computational models try to describe the salient aspects of experimentally delineated phenomena by means of mathematics (Cartwright, 1983). These types of models consist of relationships that can be described as operators, such as functions, and variables that are abstractions of the parameters of interest, such as the initial cell densities (Moise and Friedman, 2019). The modelling formulas can be roughly divided into six categories according to the variable types and relationships: linear vs. nonlinear, static vs. dynamic (c.f., slow vs. fast), explicit vs. implicit, discrete vs. continuous, deterministic vs. probabilistic, deductive vs. inductive, or floating (Berg, 2011; Desroches et al., 2012). The respective computational model is the predictive explanation arising from the modelling. The most important tools for the biological modelling are the differential equations, probability and statistics, which roughly constitutes the field of bioinformatics. The model building work process consists approximately of five stages: 1) defining the problem, 2) creating the model, 3) developing the computational model for solving the problem, 4) implementation of the model, and 5) testing and assessing the solutions. The work process can consist of multiple trial-and-error model revisions if the data fed to the model is inconsistent, or if the assumptions behind the model are not adequately explored in the beginning (Tikka and Schaefer, 2019).

1.3.2. Models in Kidney Development

The computational efforts around the ureteric branching morphogenesis, interwoven to the early nephron formation in various levels, has advanced various models *in vivo*, *ex vivo*, *in vitro*, and *in silico* (Fig. 2) (Junttila et al., 2015; Lawson and Flegg, 2016; Short et al., 2014; Takasato et al., 2015; Zubkov et al., 2015). These models usually try to delineate with imaging methods the cell patterning arriving from signaling and evolving membrane interactions between ureteric, MM, extracellular matrix, or NP cells (Adivarahan et al., 2013; Blake and Rosenblum, 2014; Clendenon et al., 2013; Clevers, 2016; Krause et al., 2015). For instance, Lawson and Flegg (2016) created a mathematical model for the induction of the mammalian UB (Lawson and Flegg, 2016). Zubkov et al. (2015) established a spatially-averaged mathematical model of kidney branching morphogenesis (Zubkov et al., 2015). Nevertheless, these models did not account for the spatiotemporal evolution of NP cells and nephron development (Tikka and Schaefer, 2019).

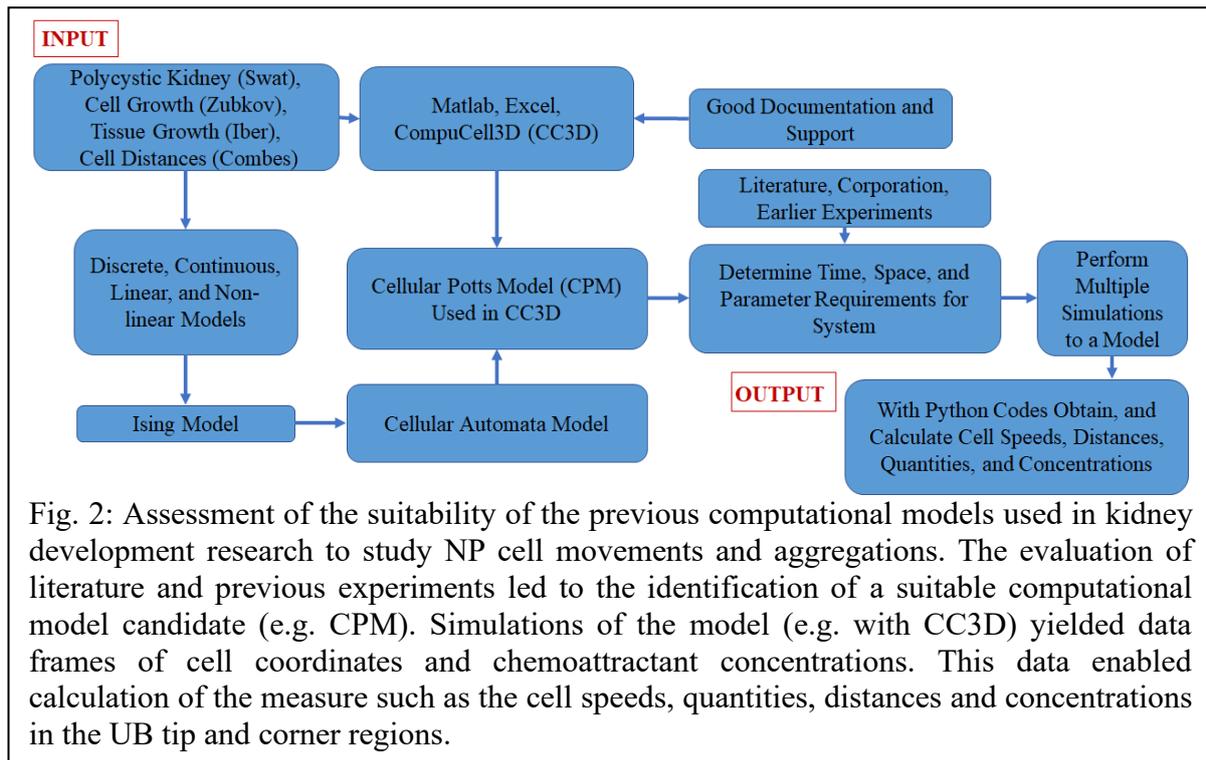


Fig. 2: Assessment of the suitability of the previous computational models used in kidney development research to study NP cell movements and aggregations. The evaluation of literature and previous experiments led to the identification of a suitable computational model candidate (e.g. CPM). Simulations of the model (e.g. with CC3D) yielded data frames of cell coordinates and chemoattractant concentrations. This data enabled calculation of the measure such as the cell speeds, quantities, distances and concentrations in the UB tip and corner regions.

Furthermore, computational methods have recently been introduced to mechanistically analyse early nephrogenesis (Combes et al., 2016; Lawlor et al., 2019). In fact, Combes et al. (2016) applied a convection-diffusion model, and showed that attractive and repulsive cues between CM cells and the UB, together with cell adhesion processes, can lead to the commitment and maintenance of CM in proximity to the tip (c.f., Fig. 1) (Combes et al., 2016). However, the underlying cellular processes leading to the observed attraction and repulsion have not been analysed in detail, and the study was focused on the formation and maintenance of CM rather than the formation of the PTA (see ‘7.1 Description of Previous Model’ for more details about the previous model).

Correspondingly, Cellular Potts Model (CPM) comprises a modelling structure, which can be used when the particularities of the intercellular interplay are mostly constrained to factors such as cell size, shape, and the common contact area to other cells (Marée et al., 2007). The reason for this reduction was that the collective behaviour of cells cannot be deduced from their individual attributes due the complexities of the developing cellular systems. Accordingly, CPM was developed as a spatial lattice-based formalism, which can be used to study the spatio-temporal behaviour of biological cell populations (Swat et al., 2009). The model handles the virtual cells as a deformable object that take their shapes from a synthesis of external and

internal forces. The energy minimisation-based CPM formalism enables to integrate these forces acting at different scales to the cells. In the first place, CPM has been especially elaborated to understand the sorting of cells. This sorting usually constitutes segregation of the heterotypic cell aggregates into homotypic cell clusters. The model has been used to demonstrate that this sorting is the consequence of the differences in the cell-cell adhesion molecules (CAM) between the different cell types in question. The formal demonstration is also called as the differential adhesion hypothesis (Glazier and Graner, 1993). This approach has been further improved and utilized for a large scale of phenomena in the developmental biology, including polycystic cystic kidney diseases (Belmonte et al., 2016). CompuCell3D (CC3D), a readily available and user-friendly model simulation software, is based on CPM (Swat et al., 2012). In the same way, it offers model example codes for various cell mechanisms. Partly due these reasons, CPM was redeemed a suitable model candidate in this work (Belmonte et al., 2016; Swat et al., 2009). In addition, new python codes and functions can be assessed and developed for calculating the appropriate measures from CC3D model simulation outcomes (Contributors, 2017). These measures could be for example the cell quantities, speeds, distances and the chemoattractant concentrations.

1.3.3. Initial Computational Models for Directed Cell Movements and Aggregations

In reality, a descriptive analysis of these processes, chemotaxis and adhesion, is possible only to a limited extent (Magno et al., 2015) since even simple cell-cell interactions can lead to complex and unexpected large-scale spatiotemporal patterns.

Notably, a full chemotaxis model with adhesion would undoubtedly involve modelling the two-way coupling between the ligand receptor and its downstream effectors and the actin-myosin cytoskeletal dynamics. In contrast, the previous work has focused on phenomenological models that directly relate rules of motion to external chemical gradients (Andrew and Ewald, 2010). The early models of motion focused on simple 1D models with the cell running linearly from the start to the finish (Andrew and Ewald, 2010; DiMilla et al., 1991). While these models have some predictive power, they obviously cannot properly address the dynamics of cell shape. Consequently, a relatively recent work has attempted to deal with 2D representation of the cell shape together with the actin-myosin dynamics (Mogilner and Rubinstein, 2005).

Much insight has been obtained regarding motility using computer assisted motion analysis (CAMA) (Soll et al., 2002; Wessels et al., 1998). CAMA has been mostly carried out in 2D, Regardless, it is also possible to perform in 3D. It traces the outskirts of the cell with a boundary

detection software. This outline is obtained from the video frames of the experiment. Henceforth, it was used to define various quantities related to cell movements, such as the cell speed or velocity, and directional change or persistence.

1.4 Study Objectives

In this study, I used a computational modelling approach to explore in detail the biophysical mechanisms underlying the directed movement and aggregation of NP cells, the critical first step of nephron development (BioPortal, 2019; Little, 2012; Tikka et al., 2019; Tikka and Schaefer, 2017). I extended the pioneering work of Combes et al. (2016) by a simulation study of the processes in question (Combes et al., 2016). This was achieved by imitating the chemical and mechanical cellular processes systematically with the simulations that potentially explain the pattern formation during early nephrogenesis. I used 3D tissue simulation approaches to analyse how different types of chemotaxis and adhesion differences between different cell types may explain the formation of both CM and the PTA. Parameter calibration and model validation was attained by comparison of the simulation results with both published and original experimental data (Combes et al., 2016; Tikka et al., 2019).

2. MATERIALS AND METHODS

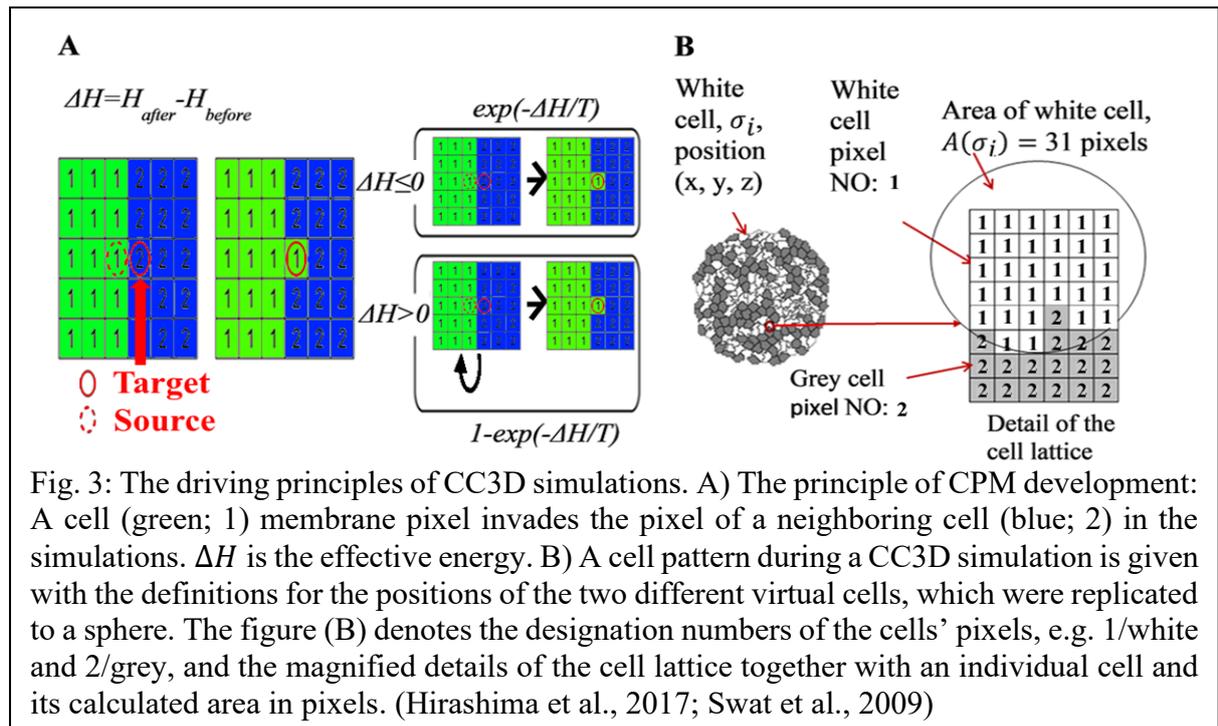
2.1. Model Building

2.1.1. Cellular Potts Model

The CC3D models used here were based on the framework of CPM (Hirashima et al., 2017; Swat et al., 2012). During each simulated time step of the model, here named as Monte Carlo Step (MCS), all cell border pixels were attempting to replace the neighbouring ones (Swat et al., 2012). This happened with a probability (P) according to a Boltzmann distribution, which depended on the change in an ‘effective energy’ H

$$P = \begin{cases} 1 & \text{if } \Delta H \leq 0 \\ e^{(-\Delta H/T)} & \text{if } \Delta H > 0 \end{cases} \quad \text{Eq. (1)}$$

where T comes from the ‘temperature’ of CPM, which means the virtual cell surface motility. This means also the relative amount of cell surface random fluctuations. The main idea of the model is that the displacements of individual cell pixels are accepted only if the overall effective energy is reduced (in lower part of Eq. 2). The power in the probability function depended also of the inverse of the ‘temperature’, which represents the effective motility of the cell's membrane. If the net energy change after MCS was negative, that is $\Delta H \leq 0$, the index changes were adopted. In cases where $\Delta H > 0$, the likelihood P, for a successful ‘index-copy attempt’, followed the Boltzmann distribution in (Eq. 1; Fig. 3).



The formulation of the effective energy contains quantities reflecting chemical and physical properties of the cells, such as area and volume constraints, cell-cell surface adhesion and processes related to chemotaxis (Andasari et al., 2012; Swat et al., 2012). As an illustration, the effective energy (H) can represent a mix of true physical energies, such as the cell-cell adhesion, and terms that mimicked energies. The latter can be in particular as a response to a chemotactic gradient experienced by the cell. Meanwhile, it can also have terms that reflect the basic principles of the evolution of the model (Hirashima et al., 2017). All things considered, the effective energy used in this study was formulated as

$$H = \sum_{\substack{\bar{i}, \bar{j} \\ \text{neighbors}}} J \left(\tau(\sigma_{\bar{i}}), \tau(\sigma_{\bar{j}}) \right) + \sum_{\substack{\bar{i}, \bar{j} \\ \text{neighbors}}} \lambda_{CL} \left(c(\vec{x}_{neighbor}) - c(\vec{x}) \right) + \lambda_V \sum (V - V_t) + \lambda_A \sum (A - A_t), \quad \text{Eq. (2)}$$

where the first sum denotes the adhesion energies, $\Delta E_{adhesion}$, and the second sum the chemotaxis energies, $\Delta E_{chemotaxis}$. Especially, J was defined as boundary energy per unit area between two different cells ($\sigma_{\bar{i}}$ and $\sigma_{\bar{j}}$) in the pixel interface of the cells ($(\tau(\sigma_{\bar{i}}), \tau(\sigma_{\bar{j}}))$). It is commonly called as contact energy coefficient. The second chemotaxis energy term was calculated for the cells experiencing chemotaxis and their neighbours. However, $c(\vec{x})$ and $c(\vec{x}_{neighbor})$ indicated the chemoattractant concentrations at the pixel source and its neighbouring pixel, respectively. In contrast, λ_{CL} reflected the strength of chemotaxis (Chen et al., 2015; Kopan, 2014; Swat et al., 2012). Finally, λ_V and λ_A penalized deviations of cell volume respectively area from the preferred values A_t respectively V_t .

Equally important, a certain level of minimal adhesion (J) was necessary for the cell movements for two reasons. Firstly, the cells in a model according to (Eq. 2) without adhesion difference, together with only small quantities of chemoattractant, would become nearly immobile soon after the beginning of the simulation (Swat et al., 2012). Secondly, the small volume expansion at the beginning cannot sustain the movements either. Consequently, it was not possible to get the simulation cell speeds and patterns to a similar magnitude as observed experimentally merely by arbitrarily selecting the constants in (Eq. 2) such as (J). ΔH depended also on a separate field of the chemoattractant concentrations, and the other variables (Eqs 1, 2) (Swat et al., 2012). Correspondingly, the diffusion equation that governed the evolution of the chemoattractant concentration, $c(\vec{x})$, was adjusted as

$$\frac{\partial c(\vec{x})}{\partial t} = (D\delta)\nabla^2 c(\vec{x}) - (\gamma\delta)c(\vec{x}) + S\delta, \quad \text{Eq. (3)}$$

where D , γ , and S were the diffusion, degradation and secretion rates for the cell in question, such as NP, respectively. The chemoattractant saturation was assumed negligible (see '2.2.2 Selection of Chemoattractant Saturation Type'). δ represent Boolean switches that were either 1 or 0 depending on whether the concentration was located on the cell or not. If it was in the cell in the question, then the switch value was one. The chemoattractant diffusion field ($\frac{\partial c(\vec{x})}{\partial t}$) evolved in a separate domain independently of the cell domain. For this reason, this domain was similar in size as the cell domain.

The current modelling handled three cell types (UB, NP, MM) as the sources and receivers of the values of the field in the equation (3). Therefore, this partial differential equation in question can be expanded into three different equations; namely $\frac{\partial c(\vec{x})_{UB}}{\partial t}$, $\frac{\partial c(\vec{x})_{NP}}{\partial t}$, and $\frac{\partial c(\vec{x})_{MM}}{\partial t}$. Similarly, all the constants and variables in each of the equations can be subscripted with the respective name of the cell, for example ' D_{NP} ' (Swat et al., 2012). The ordinary differential equations for the signal transduction, metabolic or genetic pathways could have been also used in respect to subcellular processes, markedly with integrating CC3D codes to Systems Biology Markup Language (SBML) (Hucka et al., 2003). Notwithstanding, SBML was not considered so as to minimize the model assumptions, complexities and interdependencies. One example of the complexities was that the CPM algorithm tried to modify all of the parameters together with the concentration field simultaneously when minimizing the sum in (Eq. 2).

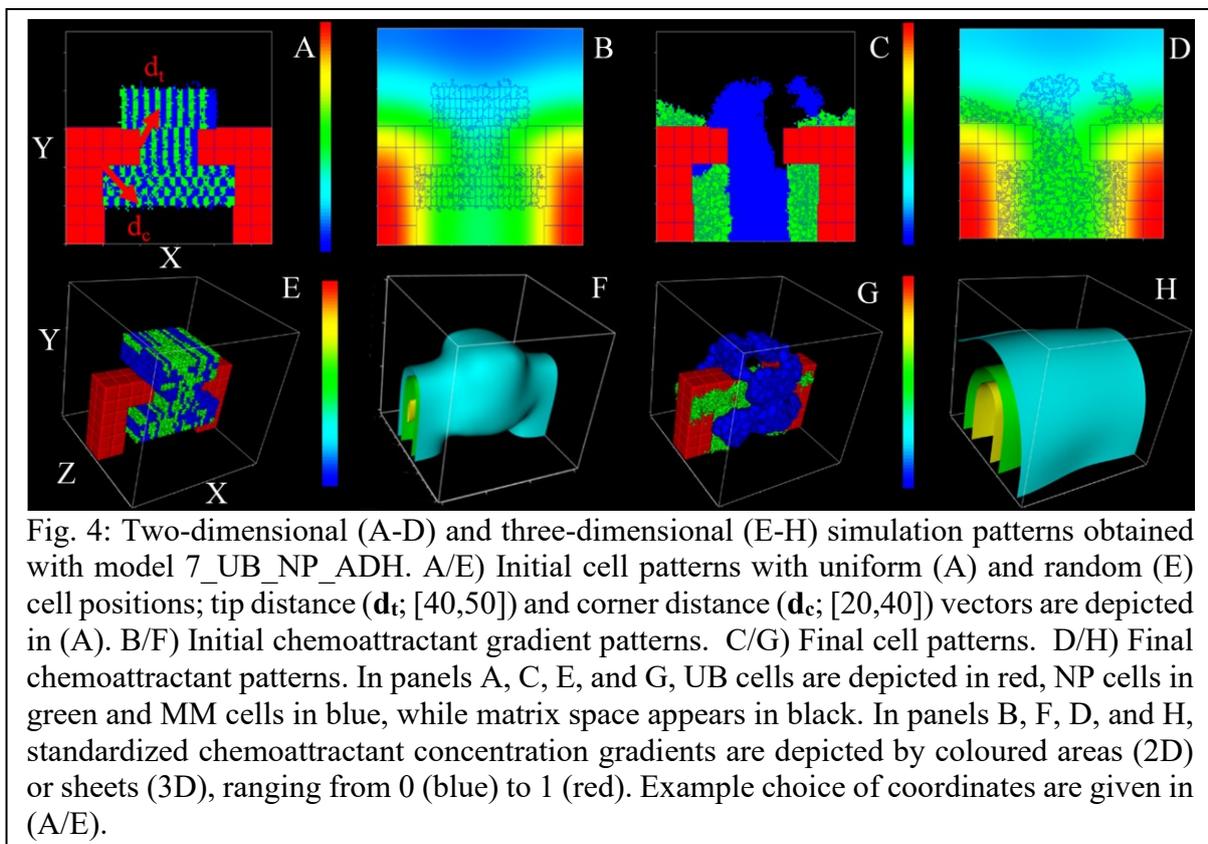
2.1.2. Model Construction

I applied and adjusted the previously described CPM to describe the behaviour of movements and aggregations of NP and MM cells during early murine nephrogenesis, postconceptional day 12 (Combes et al., 2016; Lefevre et al., 2017; Saarela et al., 2017; Tikka et al., 2019). For this purpose, I utilized the CC3D software. Explicitly, the following general assumptions were made to adjust CPM together with CC3D simulations (Combes et al., 2016; Saarela et al., 2017):

1. NP, MM, and matrix cells are able to migrate by cell sorting as described in the next chapter regarding CPM (Combes et al., 2016; Lawlor et al., 2019).
2. Matrix cells are chemically and physically neutral in comparison to other cells (Combes et al., 2016).
3. Only NP cells are able to perform chemotaxis, while UB, NP, or both cell types can secrete a chemoattractant (Junttila et al., 2015; Ricono, 2008).

4. UB cells are static (see ‘drift corrected UBs’ in (Combes et al., 2016)).
5. All cell types undergo neither proliferation nor apoptosis during the period of the observation (Combes et al., 2014; Lindström et al., 2018; Little, 2012).

The initial (3D) CC3D model setting comprised of two L-shaped structures composed of 64 UB cells each separated by a space filled with MM and NP cells ($n=196$ each), and a matrix compartment comprising all remaining empty (pixel) space (Swat et al., 2009). The setting mimicked the spatial structure of two adjacent UB branches with surrounding metanephric mesenchyme (c.f., Figs 1, 4) (Combes et al., 2016; Krause et al., 2015). CM cells initially surrounding the UB tips consisted of MM cells, which have similar cell volumes, masses, and general regulation mechanisms (Fig. 1C). NP cells were induced from CM cells at the outset of the modelled period (Lawlor et al., 2019). All cells were set to cubical shape, with initial cell surface areas of $375 \mu\text{m}^2$ for MM and NP cells, and $1000 \mu\text{m}^2$ for UB cells. The distal end of the UB branch structure was denoted as ‘tip region’ and the origin of the inner angle of UB as ‘corner region’ (Figs 1C, 4A). My primary interest was to analyse the model cell outcomes in the corner region, where the PTA was formed (Figs 1C-E).



2.2. Model Simulation Studies

2.2.1. Model Variants

Eight model variants were constructed to simulate the impact of (a) chemotaxis of NP cells related to chemoattractant secretion from NP and/or UB cells separately or together with (b) adhesion-based cell sorting related to adhesion differences between different cell types. The model characteristics are listed in (Table 1). These variants represented the mechanisms potentially driving the cells in the studied experimental biological systems (*ex vivo* kidney and renal organoid), which corresponded the evaluated *in vivo* early nephrogenesis (Combes et al., 2016).

Table 1: Model parameters with pre-set values applied in 3D simulations.

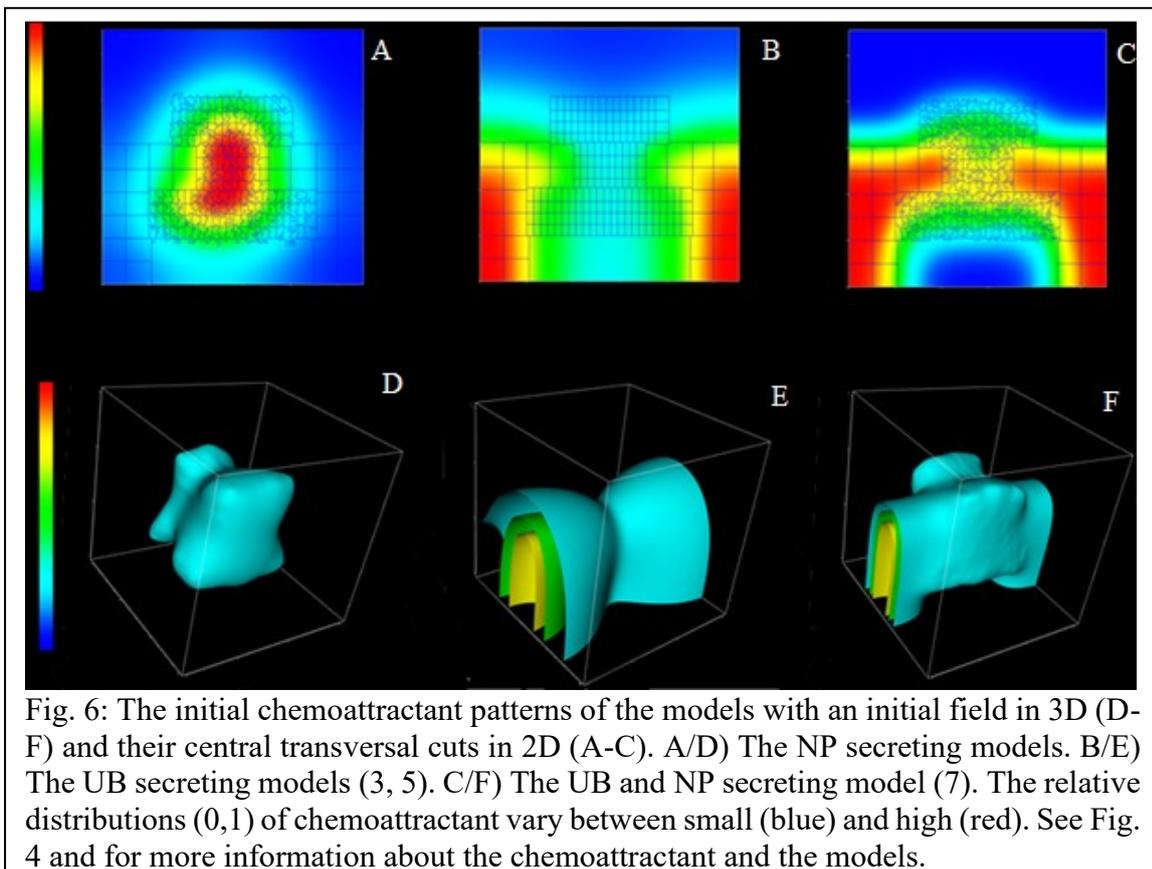
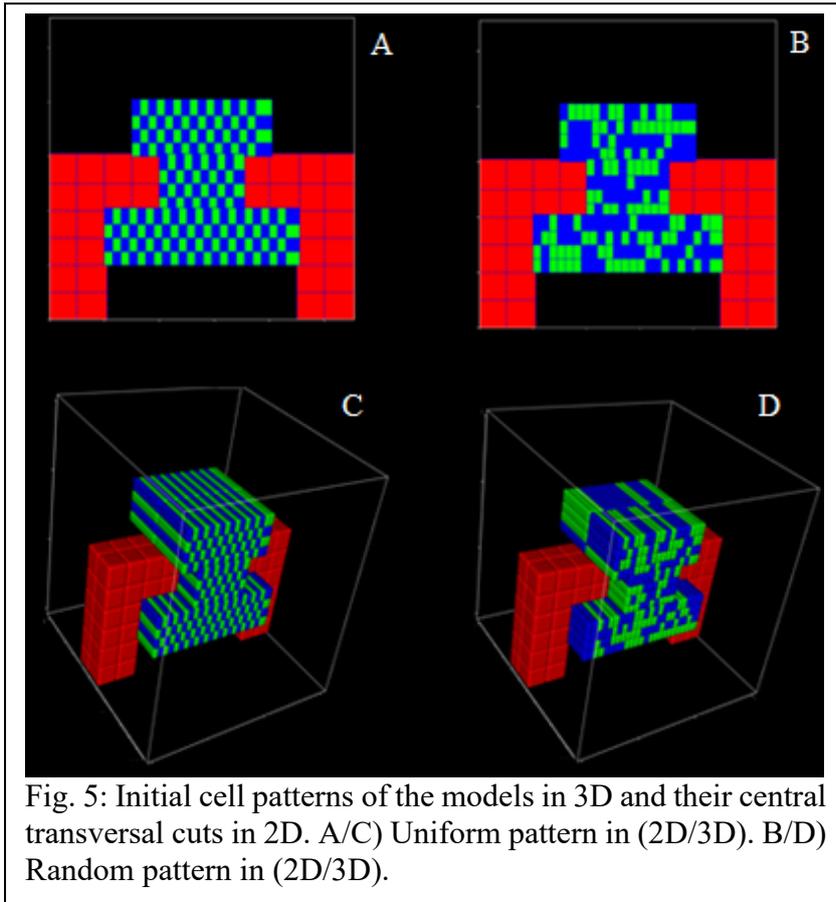
Parameter	Pre-set value	Models	Ref.
Target volume (V_t)	$375 \times 10^{-18} \text{ m}^3$	1-8	this work, (Combes et al., 2016)
Lambda volume (λ_V)	$20 \times 10^9 \text{ kgm}^{-4} \text{ s}^{-2}$	1-8	(Osborne et al., 2017; Swat et al., 2009)
Target surface area (A_t)	$312 \times 10^{-12} \text{ m}^2$	1-8	this work, (Combes et al., 2016)
Surface lambda (λ_A)	$0.1 \times 10^{-3} \text{ kgm}^{-2} \text{ s}^{-2}$	1-8	(Osborne et al., 2017; Swat et al., 2009)
Time step (MCS)	0.017 h	1-8	(Osborne et al., 2017; Swat et al., 2009)
Surface motility or CPM temperature (T)	20	1-8	(Osborne et al., 2017; Swat et al., 2009)
Contact energy coefficient (J)	(all cells): $5 \times 10^{-15} \text{ kgs}^{-2}$ (NP&MM): $13 \times 10^{-15} \text{ kgs}^{-2}$	1,3-4; 2, 5-7, 8*	(Osborne et al., 2017; Swat et al., 2009)
Chemoattractant secretion (S)	3 DU/s	3-7	(Osborne et al., 2017; Swat et al., 2009)
Chemotaxis lambda (λ_{CL})	$100 \times 10^{-27} \text{ kgm}^2 \text{ s}^{-2}$	3-7	(Osborne et al., 2017; Swat et al., 2009)
Global diffusion constant (D)	$1.0 \times 10^{-12} \text{ m}^2 \text{ s}^{-1}$	3-7	(Brown, 2011; Osborne et al., 2017; Swat et al., 2009)
Global decay constant (γ)	$1.0 \times 10^{-7} \text{ s}$	3-7	(Osborne et al., 2017; Swat et al., 2009)

*see Table 2 for description of multiple adhesion interfaces in model 8.

The first model variant (1_REF) did not include the mechanical or chemical differences between different cell types, but NP and MM cells showed patterns of ‘random walking’. This

model served as a reference, or minimum model. The second model (2_REF_ADH) assumed differences in adhesion energies between NP and MM cells (Lefevre et al., 2017), but no chemotaxis processes. Models 3 and 4 introduced chemotaxis of NP cells, assuming chemoattractant secretion either from UB (3_UB) or NP cells themselves (4_NP), while adhesion properties of NP and MM cells were set equal. In the following models (5 and 6), the models 3 and 4 were augmented by adding adhesion energy differences between NP and MM cells (5_UB_ADH, 6_NP_ADH). The seventh model included all features. i.e. the chemotaxis of NP cells, adhesion differences between NP and MM cells, and chemoattractant secretion by both NP and UB cells (7_UB_NP_ADH). Finally, a multi-adhesion model (8_ADH_ADH) was tested with the adhesion differences between all cells in the model (except UB-UB). For analysis and interpretation, the models (3, 5, and 7) were categorized as ‘UB secreting models’, and the models (4 and 6) as ‘NP secreting models’. Finally, the models (1, 2, and 8) were categorized as ‘non-secreting models’ and the models (2, 5, 6, 7, and 8) additionally as ‘adhesion-based models’.

The following model characteristics were permuted in the models to test their effect on the cell movements and aggregations in the different model variants: (i) initial (NP and MM) cell positions; (ii) initial spread of the chemoattractant from UB or NP or both (Combes et al., 2016); (iii) cell-cell adhesion properties, especially of the moving NP and MM cells (Lefevre et al., 2017). For this purpose, I test simulated NP and MM cells with random (R) or uniform (U) initial cell distribution together with the chemoattractant with or without the initial presence of a chemoattractant field (Figs 4-6). The initial uniform and random cell patterns of two-dimensional (2D) and three-dimensional (3D) models have been outlined in (Fig. 5). By the same fashion, the initial chemoattractant patterns have been depicted in (Fig. 6).



Each model variant was simulated 30 times with a predefined set of parameters and initial conditions. The repetition was done with the aim to extract an average behaviour from the inherently stochastic simulations. During each of these basic simulations ($8 \times 30 \times 4 = 960$), I recorded both the coordinates in the centre of masses (COMs) of the moving cell types (NP and MM) and the relative amount of chemoattractant in these COMs over 1000 MCS. With these COM values, I calculated the average cell speeds, quantities, distances, and the local concentration of the chemoattractant separately in the tip and corner regions (Fig. 1C). Nevertheless, the cells in the basic model simulations accelerated for a relatively long time, 300MCS, due to the lack of the initial concentration field. Consequently, it was added as one of the initial conditions (Fig. 6). As a result, the cells then almost immediately jumped to the constant chemotaxis speed at the start of the simulation. Hence, I did not calculate the first 10 MCS steps, and obtained reasonable cell speeds with only comparatively small effects to the final cell patterns.

The cell speeds in the corner and tip regions were denoted as ‘corner speeds’ and ‘tip speeds’, respectively. Similarly, the distances of these cells from the outermost point of the UB tip were denoted as ‘tip distances’ and the distances from the UB corner as ‘corner distances’ (Figs 1D/E, 4A/E). For direct comparison to experiments, I calculated the ratio of the tip to the corner values of the cell quantities, speeds, and distances in the simulations and experiments. The ratios were also needed because the corner distances of NP cells were within a certain range. Primarily, this range roughly extended to the diameter of perceived PTA in relation to simulation space, that is $\sim 50 \mu\text{m}$ (Fig. 1). Albeit, the moving cells were in between the two UB trunks. Then again, the tip distances of NP cells were in an even smaller range in between the two tips. Hence, the ratio between tip and corner distances required to be below one.

In the experimental data of Combes et al. the tip per corner NP cell speeds fluctuated relatively highly compared to the values in the test model simulations. This was one of the reasons for giving corner speeds separately. The other reason was that NP cells between or in the tips moved in swarming fashion. In addition, I calculated the chemoattractant ratios from the model simulation data, but not from the experiments. It was not technically feasible to measure the constant change of chemoattractant concentrations during the *ex vivo* and organoid experiments (see ‘2.5.1 Experimental Data Sources’).

2.2.2. Selection of Chemoattractant Saturation Type

There are three types of chemoattractant saturation in CC3D: (a) ‘regular’, (b) ‘saturating’, and (c) ‘linearly saturating’. The names indicate how the chemoattractant is spread to the virtual grid in CC3D. The ‘regular’ type of saturation (a) occurs when the chemoattractant spreads with a constant speed. By contrast, the ‘saturating’ type of saturation (b) indicates a propagation with a logarithmic curve across the simulation time. On the contrary to (a), the dissemination of the chemoattractant during the ‘linearly saturating’ type of a saturation (c) happens according to a straight line with a slope more than one. Consequently, the effective energy relating to the chemotaxis is different depending on the saturation method (Swat et al., 2017). The corresponding formulas as in (Swat et al., 2017) for the free energy as ‘saturating’ (sat) or ‘linearly saturating’ (linsat) are below

$$\Delta E_{sat} = \lambda \left(\frac{c(\vec{x}_{neighbor})}{s+c(\vec{x}_{neighbor})} - \frac{c(\vec{x})}{s+c(\vec{x})} \right), \quad \text{Eq. (4)}$$

$$\Delta E_{linsat} = \lambda \left(\frac{c(\vec{x}_{neighbor})}{s \cdot c(\vec{x}_{neighbor}) + 1} - \frac{c(\vec{x})}{s \cdot c(\vec{x}) + 1} \right), \quad \text{Eq. (5)}$$

This energy is constituted of three factors: 1) the saturation constant (s), 2) the chemotaxis lambda (λ), and 3) the concentration of the chemoattractant ($c(\vec{x})$; c.f., Eq. 3).

There were three reasons to assume the regular saturation (a) in all secreting models (3-7). First of all, the previously mentioned polarity module of the chemotaxis. It works even when it does not receive a direct input from the directional sensing module (Pollard and Borisy, 2003). This can happen when the levels of concentration or saturation are very low. Then one could insert ($s = 0$) in the previous equation (4, 5), and easily see that the free energies arriving from the alternative saturation types (b, c) disappear. Secondly, the regulation and feedback mechanisms of the secretion type (a, b, c) and magnitude ($s, c(\vec{x})$), or how cell respond to it (λ), were not directly available for the cells in the early nephrogenesis (E11-13) (Barasch et al., 1997; Desgrange and Cereghini, 2015; Ricono, 2008; Stuart et al., 2003). In this case, one could try to insert a large saturation ($s = \infty$) to the previous equations (4, 5), since it was known that the chemotaxis affected cell movements (Ricono, 2008). As a result, the linear saturation corresponded the regular saturation. In reality, the concentration of the chemoattractant cannot be enforced to infinity at will. It should be continuous and sufficiently large (c.f., Eq. 2). Thirdly, it was inconceivable to consider different types of saturation separately for UB and/or NP secreting models. For instance, it was unlikely that all secreting cells would turn their secretion on *pro rata* simultaneously with the same small magnitudes.

2.2.3. Model Simulations

The correspondence of the simulation time unit (MCS) as an experimental one depended on the voxel (3D) scale (l), and the maximum speed of the relevant experiments (v_{max}) (Andasari et al., 2012; Swat et al., 2012). Henceforth, it can be calculated together with the physically maximum allowed speed ($0.2 \frac{voxels}{MCS}$) of the simulations

$$t_{MCS}(s) = 0.2 \frac{voxels}{MCS} \cdot \frac{l}{v_{max}} \quad \text{Eq. (6)}$$

In this work, I supposed that cells crawled on a substrate of an experiment with a speed of approximately $0.2 \frac{\mu m}{min}$. For the simulations, I also assumed that $1 \mu m$ corresponded to 1 pixel in length. In other words, the lattice scale (l) was one $\frac{\mu m}{pixel/voxel}$. Then, the maximum time per MCS was $0.2 \cdot 1 / 0.2/60$ [seconds/minute] seconds = 60 seconds. Thus, the maximum real experimental time that can be assigned to 1000 MCS was $\sim 16.7h$ (Combes et al., 2016). (Eq. 6) also allowed the conversion of the model speeds to same scope as the experimental speeds. By contrast, a single CC3D simulation in 2D models typically took 30 seconds, and for 3D models 30 minutes. Be that as it may, there were no differences in the chemoattractant spread mechanisms between the different initial conditions of the simulations, such as U and R models with or without an initial chemoattractant field, as also indicated in ('2.2.2 Selection of Chemoattractant Saturation Type'). Exemplary tests of 2D model simulation in CC3D with these conditions are shown in Figures 7 and 8. The CC3D simulations (Swat et al., 2012) consisted of 3-6 python code files of dynamic/static kind (see '7.3.1. Example of Dynamic CompuCell3D Coding File') (Tikka, 2019b):

1. (Static file) for simulation configurations, which has for instance the Potts variables and parameters. Obtains values from the next (dynamic) file.
2. (Dynamic file) for simulation constraints and recording measurements, saving cell coordinates of each MCS to a file.
3. (Static files) for initial cell and signaling configuration, which can be modulated/randomized with the codes of the previous (dynamic) file.
4. (Static files) for all parameters and parameter scan values, such as for the adhesion differences and chemotaxis strength. (see '2.3.1 Initial Estimation of Parameters').

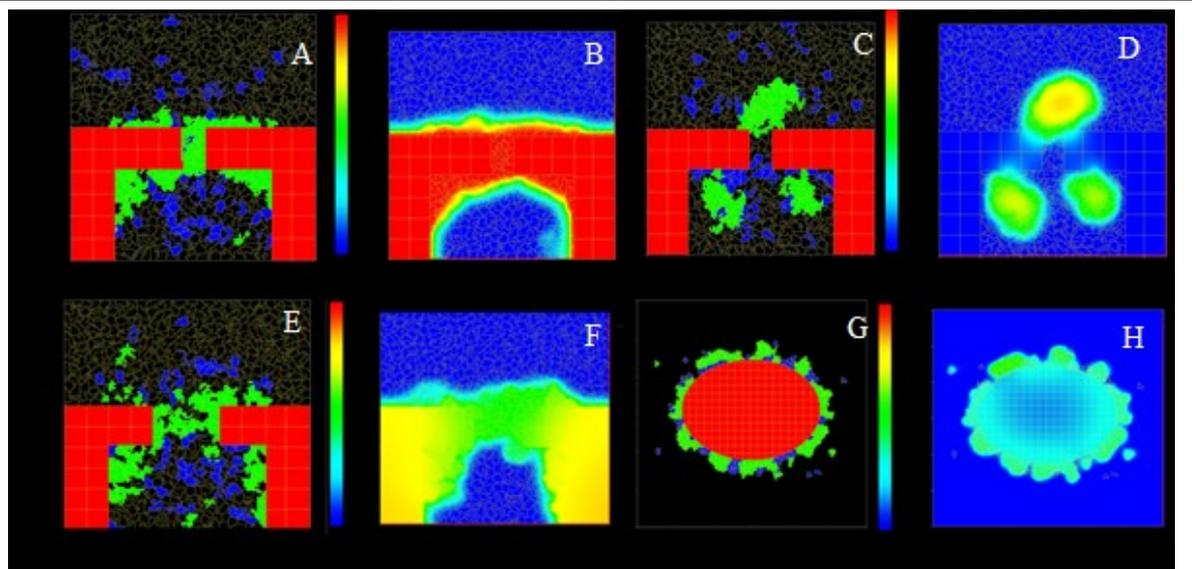


Fig. 8: CC3D (2D) simulation tests for finding cell and chemoattractant patterns that are within the range of the model assumptions (Tikka et al., 2019). A/B) UB secreting model with high secretion rate, and small gap between UBs. C/D) The NP secreting model with low secretion rate, and the small gap. E/F) UB model with low secretion rate, and big gap between UBs (closer to final version). G/H) The NP secreting model with low secretion rate, and one big UB. G/H) A test model for dissociation-reaggregation of NP and MM cells next to oval UB. In panels A, C, E, and G UB cells are depicted in red, NP cells in green and MM cells in blue, while matrix space appears in black. In panels B, F, D, and H, standardized chemoattractant concentration gradients are depicted by coloured areas (2D), ranging from 0 (blue) to 1 (red).

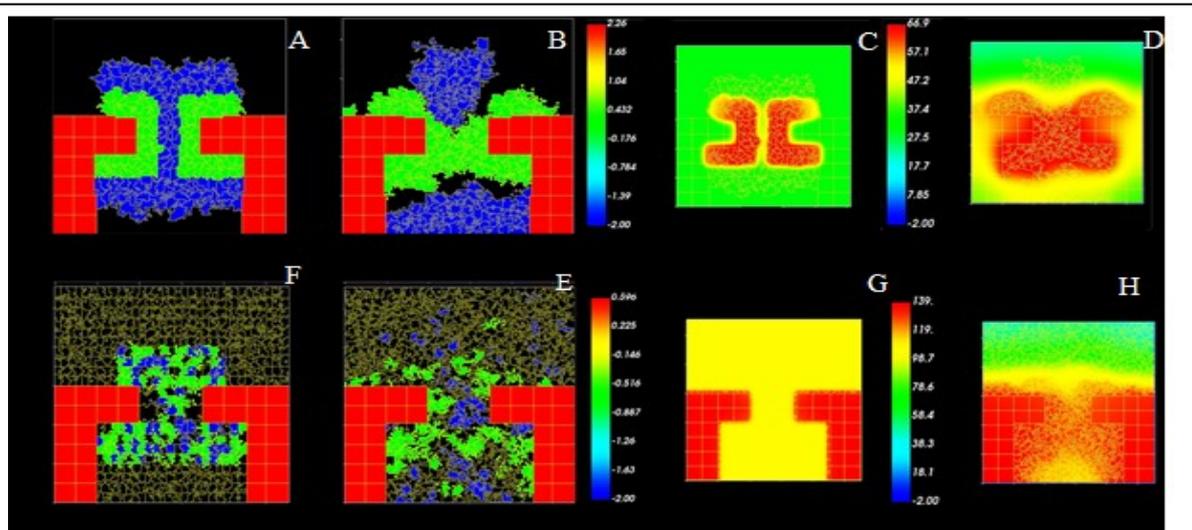


Fig. 7: CC3D (2D) simulation tests for finding appropriate cell and chemoattractant patterns for PTA formation at the end of simulations. A-D) Start (A/C) and end (B/D) cell/chemo (A-B/C-D) patterns of the uniform model (4). E-H) All similarly (as in A-D), but with the random model (3). In panels A, C, E, and G UB cells are depicted in red, NP cells in green and MM cells in blue, while matrix space appears in black. Non-standardized chemoattractant concentration gradients are depicted by coloured areas (2D), ranging from blue (low value) to red (high value).

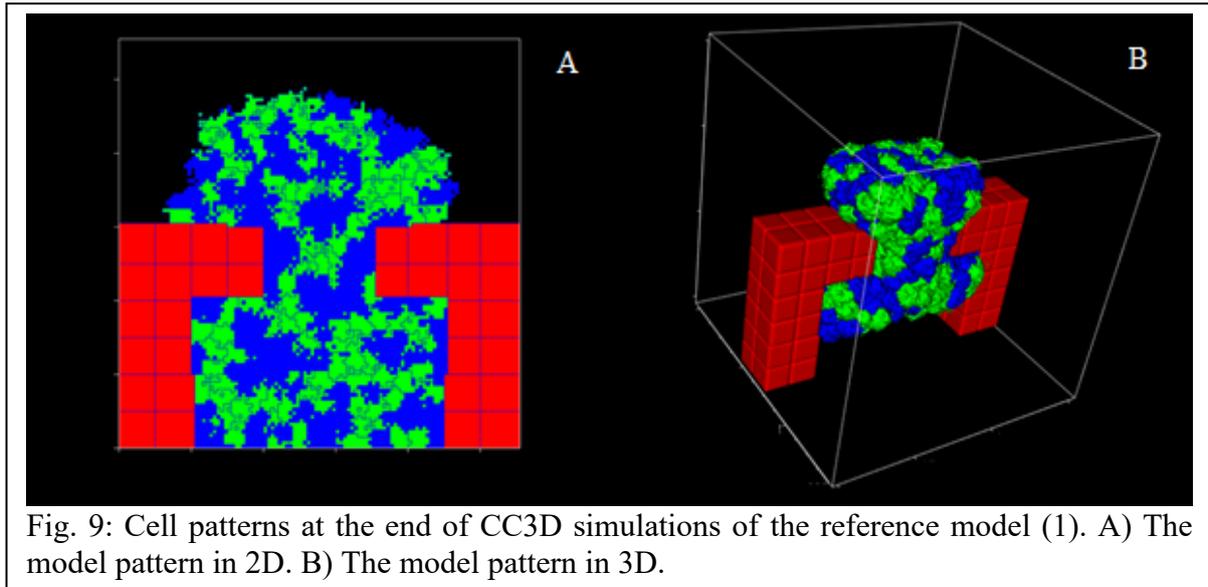
2.3. Computational Estimation of Model Parameters

2.3.1. Initial Estimation of Parameters

The process of the model parameter estimation was initiated by varying the values of a cell-sorting (2D) simulation in the CC3D software (Swat et al., 2012). The parameter modulation strategy was implemented as a back-inducing one, as will be described below (Andasari et al., 2012; Combes et al., 2016; Osborne et al., 2017; Swat et al., 2009; Swat et al., 2012). The initial parameters of spatial relationships (such as λ_A , λ_V , V_t , A_t in Eq. 2) and the cell numbers were chosen to be in comparable ranges as observed in the explant culture and kidney organoid experiments (see ‘2.5.1. Experimental Data Sources’, and Figs 1-4). These initial settings constituted the reference model (1; Fig. 9). The other initial parameter ranges were taken from the respective literature (Chen et al., 2015; Osborne et al., 2017; Swat et al., 2009). This especially applied to the contact energy coefficient (J), which refers to the cell-cell adhesion differences (Chen et al., 2015; Lefevre et al., 2017), and chemotaxis strengths (λ_{CL}) (Chen et al., 2015; Combes et al., 2016; Little, 2015; Little, 2012) in the other 2D models (2-8). The parameter values for all 2D models were subsequently scaled to 3D settings by multiplying with two as explained in (Magno et al., 2015), given the neighbourhood order of three. The final parameter values of the 3D models have been given in (Table 2).

Table 2: PSO optimized parameter values are bolded in each model (1-8). Values are presented as: Random (Uniform), for example in 6_NP_ADH: 7.9 (13). [*] 8_ADH_ADH assumed nine cell-cell adhesion interfaces: [1-3] ‘Wall (UB) and NP/MM/Medium’, [4] ‘NP and NP’, [5] NP and MM’, [6] ‘MM and MM’, and [7-9] ‘Medium and NP/MM/Medium’. The parameters of spatial relationships (V_t, A_t, λ_V, NO), except λ_S , and the ones not mentioned here were constant (see Table 1).

	REF (1)	ADH (2)	UB (3)	UB_ ADH (4)	NP (5)	NP_ ADH (6)	UB_NP_ ADH (7)	ADH_ADH (8)
Chemoattractant secretion rate (DU/s)	0	0	3	3	3	3	6.91	0
Chemotaxis lambda (10^{-27} ; $\text{kgm}^2\text{s}^{-2}$)	0	0	4.22 (4.19)	43.90 (43.76)	67.94 (71.28)	74.54 (74.27)	1.41	0
Global diffusion constant (10^{-12} ; m^2s^{-1})	0	0	1	1	1	1	1.83	0
Global decay constant (s)	0	0	1×10^{-7}	1×10^{-7}	0.002	0.002	4.69 x 10^{-7}	0
Surface motility or CPM temperature (DU)	10	25	25	25	25	25	49.8	25
Surface lambda (10^{-3} ; $\text{kgm}^{-2}\text{s}^{-2}$)	0.01	0.01	0.01	0.01	0.01	0.01	8.11	0.01
Contact energy coefficient between cell types (10^{-15} ; kgs^{-2})	5.0 [5]	7.9 [5]	5.0 [5]	6.68 [5] (6.98 [5])	5.0 [5]	7.9 [5] (13 [5])	7.86 [5]	21.07 [1], 25.17 [2], 1.75 [3], 0.50 [4], 33.33 [5], 7.68 [6], 35.83 [7], 29.74 [8], 1.57 [9] (26.59 [1], 47.81 [2], 29.53 [3], 0.50 [4], 20.47 [5], 0.52 [6], 24.09 [7], 1.54 [8], 38.09 [9])
Best Quality Value of PSO	-61, 269	-56, 436	-201, 902	-206, 525	-202, 545	-205, 261	-213, 787	-340,361



The simulation strategy for the models (2-8) was to produce NP cell aggregates, preferably close to spheroidal shape, by permutating the previously mentioned initial parameters (Andasari et al., 2012; Lefevre et al., 2017; Saarela et al., 2017; Swat et al., 2009). Particularly, I aimed that the aggregation was to be directed to towards the corner region at the end of the simulations (Figs 1-3). Accordingly, the initial literature estimates were modified one at the time, including in the following order:

1. λ_{CL} of NP cells.
2. D , γ , and S of the chemoattractant coming from UB and/or NP cells.
3. J between NP, MM, and/or UB cells.
4. λ_V and λ_A of NP and MM cells.
5. V_t and A_t of NP and MM cells.

2.3.2. Particle Swarm Optimization of Parameters

The Particle Swarm Optimization (PSO) technique was used to optimize the model parameters (Anum et al., 2016; Sluka et al., 2018). It improves the candidate solutions for parameters with respect to a given measure of quality (Bonyadi and Michalewicz, 2017; Zhang et al., 2015). The PSO was set to maximise the number of NP cells at the surface of UB in models (1, 2, 3, 5 and 8), mimicking the patterns in the kidney organoid experiments, while simultaneously minimizing the difference between simulated and experimental (UB) tip cell speeds (Anum et al., 2016; Lefevre et al., 2017). In the NP secreting models (4, 6 and 7), the common surface area between NP cells for forming PTA was maximized with the similar handling for the corner

cell speeds. Hence, the speed differences and NP cell coverages could be used to calculate the measures of quality.

The optimization of the simulated cell speeds was achieved by the use of the experimental data of Combes et al. (2016) (see ‘2.5.1 Experimental Data Sources’) (Anum et al., 2016; Combes et al., 2016). It was assumed that CM cells, described in that study, corresponded to NP cells in my model. Simulated cell speeds were also compared to the movement of MM cells observed in a kidney organoid model (see ‘2.5.1 Experimental Data Sources’). It was presumed that these cells corresponded to both NP and MM cells in my simulations. The cells attached or close to the UB tip in the organoid model were considered MM cells in the simulations, while the remaining MM cells were assumed to be NP cells (Figs 1C-E). CC3D development program ‘Twedit++’ incorporated various model coding components for the model PSO codes, and are hence presented in (Tikka, 2019b). The details of the PSO algorithm are discussed both in the next chapter and in the supplement (see ‘7.3.2 Shortened Python Code of Particle Swarm Optimization Method’). Simulation settings, parameters and constants that were optimized together using the PSO algorithm were referred to as ‘PSOed’.

2.4. Particle Swarm Optimization

2.4.1. Algorithm of Particle Swarm Optimization

The PSO algorithm solves the parameters by using a population of the candidate solutions (Bonyadi and Michalewicz, 2017; Zhang et al., 2015). These solutions are called swarms. The algorithm handles the individual candidate solutions, also known as particles. These items are moved around the search-space according to their position and speed. The movement of the swarms is influenced by their own best-known positions. The movements are also guided towards the best-known position across all particles in the swarm. These best solution places are then updated as better positions are found. This process moves the swarm towards the best solutions (Fig. 10).

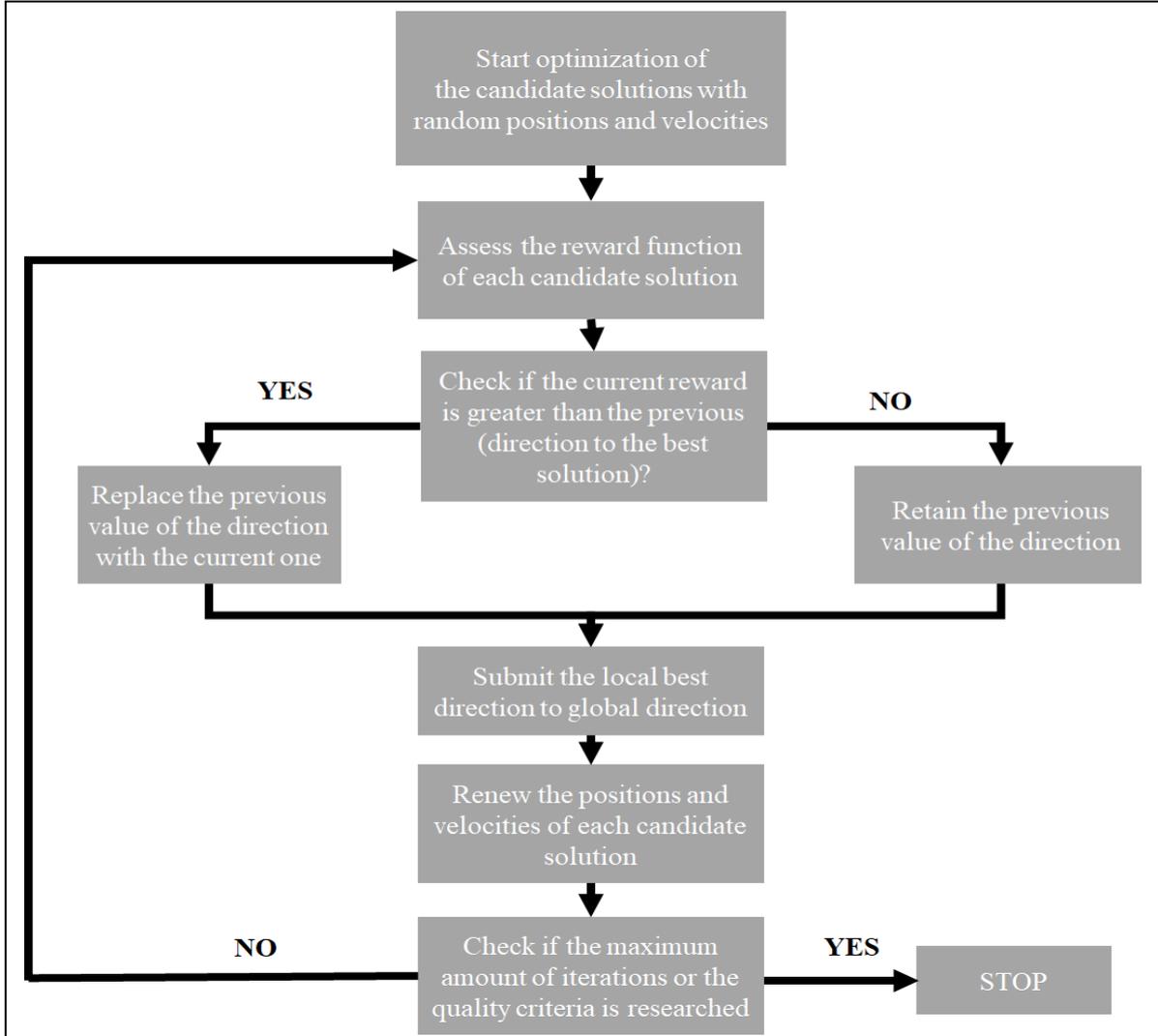


Fig. 10: The principle of the particle swarm optimization for the parameter estimation. (Mukhopadhyay and Hazra, 2015)

PSO makes only few assumptions about the problem being optimized. It can therefore search very large parameter spaces efficiently. In (Zhang et al., 2015) it was formulated as

$$pbest(i, t) = \underset{k=1, \dots, tc}{argmin} [f(Pi(k))], \quad \text{Eq. (7)}$$

$$gbest(t) = \underset{\substack{i=1, \dots, N_p \\ k=1, \dots, tc}}{argmin} [f(Pi(k))], \quad \text{Eq. (8)}$$

where $pbest$ in (Eq. 7) is the direction from the current position to the best solution the particular particle has found so far is. Similarly, $gbest$ in (Eq. 8) is the direction to the best solution regarding any particle in the swarm has found so far (Zhang et al., 2014). Furthermore, i denotes the particle index together with N_p , which is the total number of particles in a range

that is to say $i:[1, 2, \dots, N_p]$, t_c the current iteration number, f the fitness function as defined below, and $P_i(k)$ the position for time k . The fitness function f maps the values of the particles to a real value, and rewards those particles that are closest to the optimization criterion. The ‘velocity’ V and position P of the particles are updated by the following equations

$$V_i(t + 1) = \omega V_i(t) + c_1 r_1 (pbest(i, t) - P_i(t)) + c_2 r_2 (gbest(t) - P_i(t)),$$

Eq. (9)

$$P_i(t + 1) = P_i(t) + V_i(t + 1),$$

Eq. (10)

where V in (Eqs 9, 10) denotes the ‘velocity’, composed of a vector of the same dimension as the parameter search space, ω is the inertia weight used to balance the particles momentum with the global and local exploitations, r_1 and r_2 are uniformly distributed random variables within a range $[0, 1]$, and c_1 and c_2 are positive constant parameters used to weight the local and global affects, respectively.

In my case, the momentum and scaling factors in (Eq. 9) were $\omega = 0.73$ and $c_1 = c_2 = 1.5$. The maximum speed for a parameter was limited by the parameter's allowed maximum value, that is to say V_{max} was set to one. It is common to set an upper bound for the speed parameter to keep particles from flying out the allowed search space (Shahzad et al., 2014). If a particle's next movement would carry it out of the allowed parameter space, the particle is ‘reflected’ by the parameter space wall and the particle ‘rebounds’ back into the allowed search space. Important to realize was that the search space was already limited by the initial ranges of the parameters in each model. With this in mind, I studied maximum of seven parameters in PSOs. Example ranges are given in the next chapter (‘2.4.2 Constraints and Ranges of Parameters in Particle Swarm Optimization’).

2.4.2. Constraints and Ranges of Parameters in Particle Swarm Optimization

In order to minimize the time used for an appropriate convergence of the CC3D algorithm, I applied and restricted the following static constraints during the PSO iterations: chemotaxis lambda (λ_{CL}), secretion rate (S), the diffusion constant (D) as well as the decay constant (γ) by the upper bounds of 100, 30, 0.5 and 0.001, respectively. The initial contact energy coefficients (J) between NP and MM cell values in the adhesion increment models (2, 5, 6, 7, 8) were 13, and in the other models (1, 3, 4) J was 5. Typical swarm parameter ranges were: $D:[0.1, 2.0]$,

γ : $[10^{-8}, 10^{-6}]$, J : $[2.0, 8.0]$, λ_{CL} : $[10.0, 150.0]$, S : $[0.3, 30.0]$, T : $[5, 50.0]$, and λ_A : $[0.001, 10]$. PSOs yielded a variable number of PSOed parameters depending on the model variant (Table 2).

Consequently, 7_UB_NP_ADH_R was considered the most technologically advanced model in comparison to others, since it contained all the parameters that other models had and more. Hence, its parameter and quality values were used as a maximum appraisal that the other models should not exceed. Moreover, the PSO algorithm used in this work penalized the overall quality value with specific χ^2 values that were derived from the previously mentioned speed differences in each time point (Anum et al., 2016). Despite, these speed quality values were smaller than the quality value of the surface coverage (Combes et al., 2016). Accordingly, the speed quality values in each time point for every model were also multiplied by a scaling factor (of 100) that corresponded 20% of the maximum quality value from NP coverage at UB in 3D 7_UB_NP_ADH_R. It was deemed unlikely that the speed quality values were higher in the other models. For example, 4_UB_R had a nominal quality value of 199475 in one PSO, where the contribution of the surface area was 186264, and the speed value was -13211. By comparison, a quality value in the previous model (7) reached a plateau value of 220750 with example contributions of 198675 and -22015 respectively for the area and speed.

As an illustration, PSOs were run with 60 iterations using 4 independent swarms and 8 particles per swarm. Each simulation was run three times, and used the average result, since the CC3D simulations were stochastic. As a result, the replicate runs with the same parameter set did not give identical results. Example ranges of parameters and simulation lengths during the model (7) optimization are given in the supplement ('7.2 Example Ranges of Parameters and Simulation Lengths During Optimization').

2.5. Experimental Design

2.5.1. Experimental Data Sources

In this study, two sources of *ex vivo* experimental data were used to calibrate and validate the simulation models (Combes et al., 2016; Tikka, 2019c). These comprised the results obtained with an explant culture model reported by Combes et al. (2016), and original results reported in this work obtained with a dissociation-reaggregation kidney organoid model (Combes et al., 2016; Costantini and Kopan, 2010; Saarela et al., 2017; Tikka, 2019c). The recorded data frames consisted of the number of samples per number of time points. The sizes of these frames were 15 x 50 for the explant culture in 15h, and 1x15 for the kidney

organoid study in 2.5h (see ‘2.6. Technical Workflow of Analysis’ for more details) (Tikka, 2019c). The explant culture experiments of Combes et al. (2016) comprised of 500 NP cells, whereas the kidney organoid studies presented here comprised of 500-2000 MM cells (Combes et al., 2016). Outliers were replaced with a moving average of three previous data values, c.f., python code ‘rolling mean’ (Tikka, 2019d).

2.5.2. Kidney Organoids

Embryonic kidneys were dissected from embryonic day 11.5 mouse embryos from crossing of Wnt4Cre (Shan et al., 2010) and tomato floxed Rosa26 Green fluorescent protein (GFP; mT/mG) reporter mice (Muzumdar et al., 2007) as described in (Junttila et al., 2015). Intact UBs were treated with GDNF and dissociated MM with BMP7 and FGF2 as in (Junttila et al., 2015). The intact UB was reaggregated with MM cells and incubated overnight to form a kidney organoid. The organoids were set to grow in a FiZD culture (Saarela et al., 2017) for time-lapse imaging into a temperature and gas controlled on-stage incubator (Okolab, Italy) on a Zeiss LSM780 confocal microscope.

2.5.3. Microscopy, Image Processing and Data Segmentation

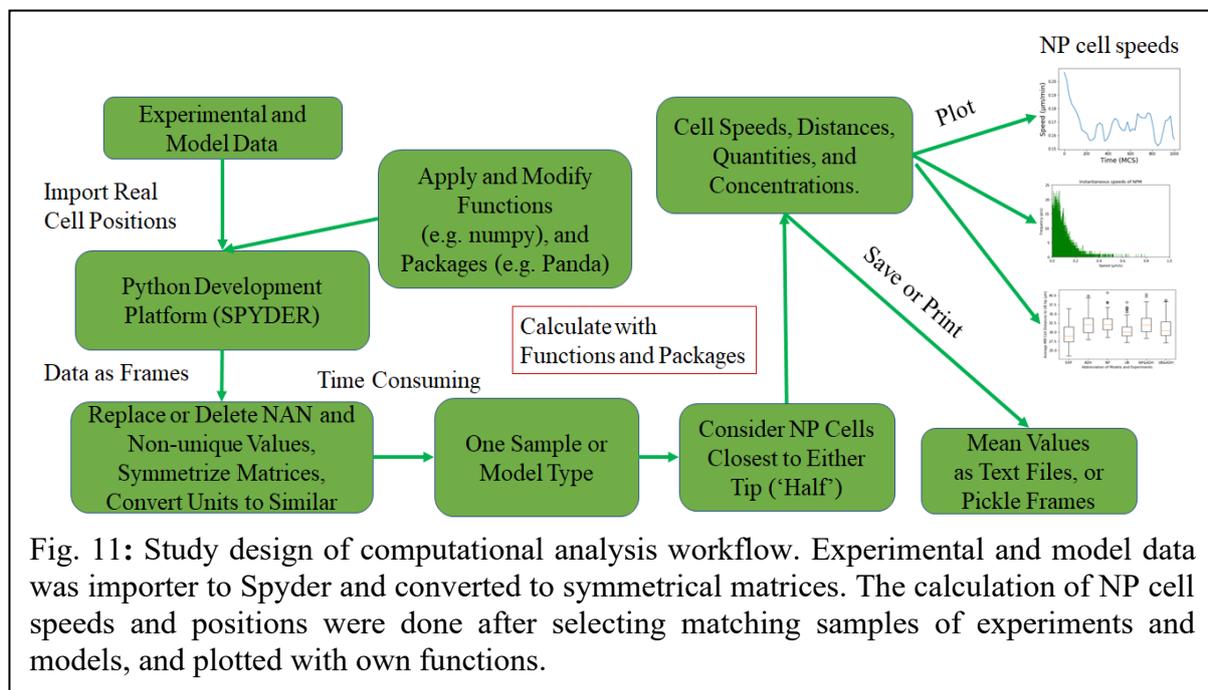
For the purpose of tracking and distinguishing MM cells from UB cells, MM cells were set to express GFP. The fluorescence microscopy of this work was performed with a Zeiss LSM 780 confocal microscope (Carl Zeiss, Germany) using 25x/0.8 Zeiss LCI PlanNeofluar water immersion objective with a 488nm wavelength for excitation, and a range of emission between 490-601nm. Out of original images of 250 μm x 300 μm x 29 μm (XYZ), an area of two close and relatively static UBs, approx. 150 μm x 100 μm , was selected with the Zen Blue (Carl Zeiss, Germany) imaging program. There were 16 Z-layers in the z axis stack, from which all were selected. Here, a XYZ pixel size was 0.24 μm x 0.24 μm x 1.46 μm , and the temporal resolution was given by 16-time frames with 10 min between the frames. The video of the moving MM cells in the kidney organoid experiment has been given stack-by-stack in an online repository (Tikka and Skovorodkin, 2019). Deconvolution was used to improve image signal to noise ratio, contrast and resolution. This process was done with Huygens Professional program (Scientific Volume Imaging, The Netherlands) using distilled PSF, with background value of 3.5, S/N value of 10, autobleach correction off, and saving the resulting deconvolved image as 16-bit tiff file format (Saarela et al., 2017; Tikka, 2019c). The segmentation of the tiff file of the deconvolved image was done with a previously compiled MATLAB code (Saarela et al., 2017). I specified the cell diameter range between

0.5-20 μm . The code produced a raw data file ‘.csv’ of the positions, sizes and speeds of all of the cells by identifying objects that it redeemed as cells from the images.

2.6. Technical Workflow of Analysis

The purpose of the current workflow was to develop appropriate ways to calculate measures from the recorded model data, and the data obtained from the experimental studies of kidney organoid and explant culture models (Combes et al., 2016; Tikka, 2019c). The primary goal was to calculate the regional cell quantities, speeds, distances and concentrations of the chemoattractant from the data. This enabled the comparison and analysis between the different data. Hence, the analysis required five coding phases, see below.

Before the calculations, the data frames were imported to a coding platform, such as Spyder with python version 3 (Contributors, 2017; Tikka, 2019d). The file extensions of these frames were ‘.xlsx’ for experiments and ‘.csv’ for models. With this information, these frames were then made similar in size with python’s PANDA package. This was needed for the remaining primary operations and calculations (Fig. 11).



Accordingly, I constructed up to five functions for calculating each measure from the model data (see ‘7.3.3. Shortened Python Code of 2D and 3D Data Analysis’) (Tikka, 2019c; Tikka, 2019d). The basic flow of routine was the same for each measure: 1) selection of the previously

preselected NP or MM cells in either tip or corner regions, 2) checking zero or NAN values out of the data, 3) calculating the measure for one/all conditions, 4) calculating the averages and confidence intervals (CI) of the measure across all of the previously selected data (in routine 1), 5) exporting the resulting data to '.csv' files. The experimental tip and corner distances and quantities of cells were normalized to [0, 1], and the transformed cell distances or quantities were multiplied by the model averages. The speeds and concentrations were not scaled between the simulations and experiments.

In the first place, the calculations were done with the NP and MM cells near to the left-hand side of the two simulated UBs for various reasons (c.f., Figs 1D/E, 4A). In their original analysis Combes et al. (2016) considered NP cell coordinates near one UB, whereas I simulated the behaviour of both NP and MM cells between two adjacent UB tips (Combes et al., 2016; Tikka et al., 2019). The latter was similar to the kidney organoid experiments of this work. In my spatio-temporal model setting, the cells surrounding either of the UB tips reacted presumably first to the chemoattractant coming from that UB (Short et al., 2014), as was illuminated in the previous similar experimental studies (Lefevre et al., 2017). In the other case, the proximity of the chemoattractant source to force cells go closer to that source, applied also to the chemoattractant secreting NP cells themselves (Figs 4, 7, 8). Either way, to have comparable results, I calculated the average measures and CIs for those NP and MM cells that were closer to left-hand side of the previous structure *in silico*. The experimental vectors have been outlined previously in (Figs 1D/E). Finally, the results were visualized in Excel together with the scaled experimental measures.

The reliability of the model simulation outcome was assessed by monitoring in tandem the dimensionless concentration levels or cell patterns during CC3D simulations (Figs 6, 9, 12-15). In comparison to the reference model (1; Fig. 9), the cell patterns (c.f., Figs 9, and 13, 14) in the secreting models (3-7) should be more coherent, and the levels of the concentration (c.f., Figs 6, and 15) higher. On the contrary, the outcome of the other non-secreting models (1, 2, 8), which did not allow for any chemoattractant secretion, should be different than the previous model outcomes (c.f., Figs 12, and 13, 14).

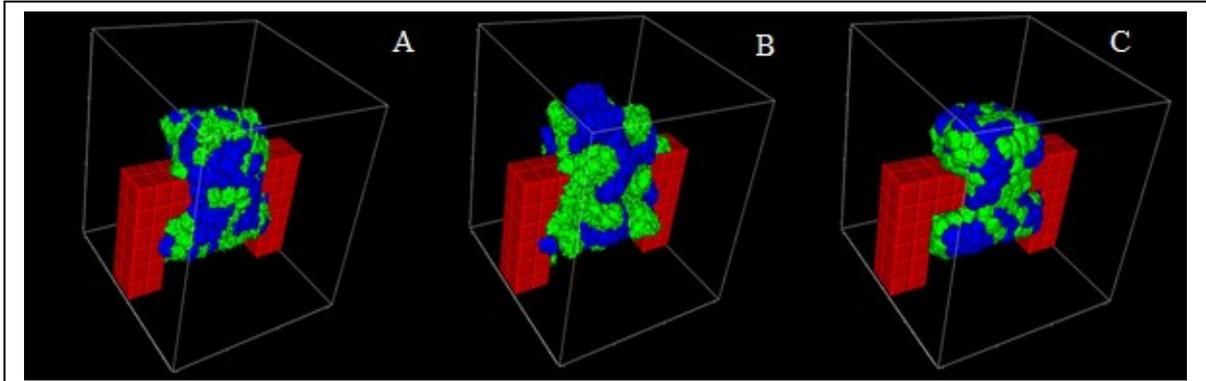


Fig. 13: Cell patterns at the end of CC3D simulations of the non-secreting models in 3D: A) 1_REF_R, B) 2_REF_ADH_R, C) 8_ADH_ADH_R.

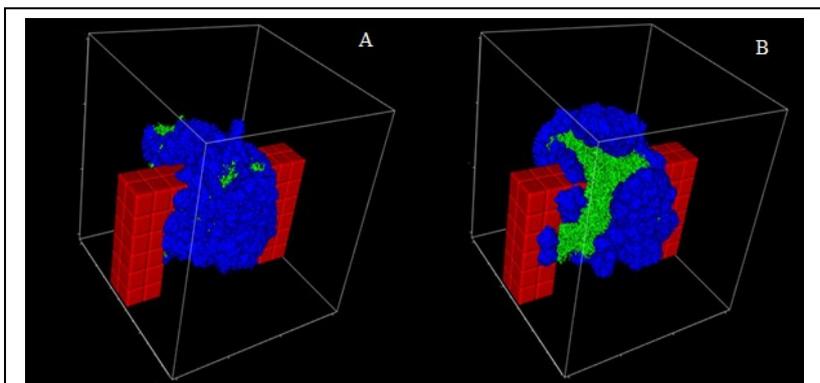


Fig. 12: Cell patterns at the end of CC3D simulations of the NP secreting models in 3D: A) 4_NP_R, B) 6_NP_ADH_R.

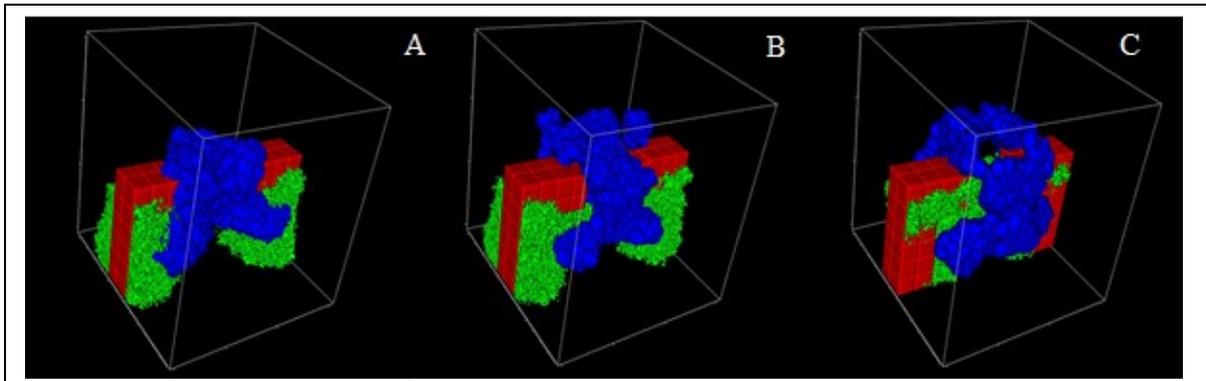


Fig. 14: Cell patterns at the end of CC3D simulations of the UB secreting models in 3D: A) 3_UB_R, B) 5_UB_ADH_R, C) 7_UP_NP_ADH_R.

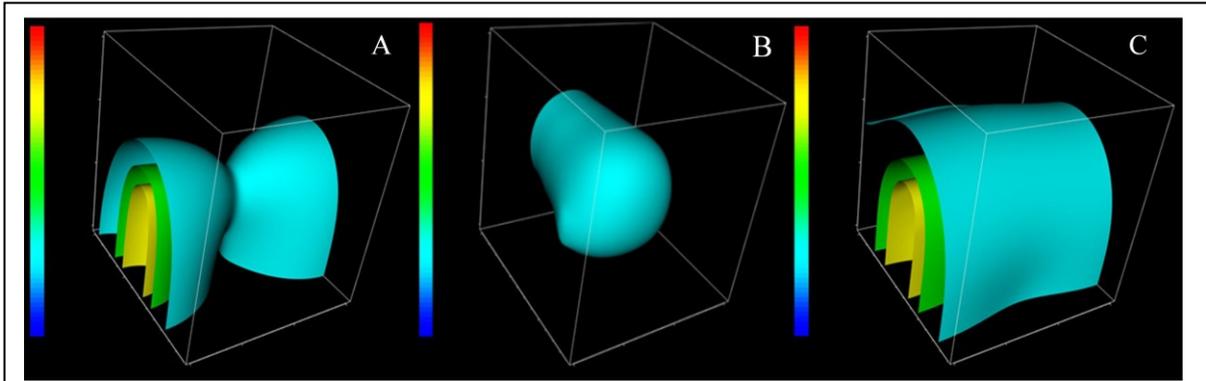


Fig. 15: Chemoattractant patterns at the end of CC3D simulations. The primary patterns for the UB and NP secreting models. A) 3_UB and 5_UB_ADH, B) 4_NP and 6_NP_ADH, C) 7_UP_NP_ADH. The relative distributions (0,1) of chemoattractant vary between small (blue) and high (red). See Fig. 4 and for more information about the chemoattractant and the models.

2.7. Analysing Data with Self-Organizing Maps

2.7.1. Introduction

The Self-Organizing Map (SOM) approach, an iterative machine learning method (Kohonen, 1982; Kohonen, 2007), was applied to identify and compare the stable cell speed regions in the simulations and experiments. Specifically, I was interested in the different behaviour of the simulated and real NP cells at the tip and corner regions. For this purpose, the python function ‘MiniSom’ (Vettigli, 2018) was applied to perform the SOM speed analysis, to my simulations and the experimental data as described above (Combes et al., 2016; Tikka et al., 2019).

2.7.2. Self-Organizing Maps

SOM is a machine learning method involving an artificial network to represent the original experimental or simulation data. The method resembles mathematical iteration by conducting an unsupervised learning to compress the data to 2D map of nodes. The primary relationships of the data elements, such as the cell names and their different coordinates, can be conserved with a neighbourhood function. (Kohonen, 1982)

The iteration routine consisted of giving weights to the real distances between the network nodes and the data. The node, whose weight vector was most similar to the data, was called the best matching unit (BMU). The weights of the BMU, and nodes close to it in the SOM grid, were adjusted towards the input vector. The magnitude of the change decreased with time and with the grid-distance from the BMU. The iteration equation for a node v with weight vector W_v was

$$W_s(s_i + 1) = W_v(s_i) + \theta(u, v, s_i)\alpha(s_i)D(t) - W_v(s_i), \quad \text{Eq. (11)}$$

where $D(t)$ in (Eq. 11) was the data vector, s_i was the iteration step index number, t was a randomly or otherwise selected index number of the training sample, u was the index number of the BMU for $D(t)$, and $\alpha(s_i)$ was a monotonically decreasing learning coefficient. Finally, $\theta(n, v, s_i)$ was the neighbourhood function. It gave a distance between the previous node (n) and the node (v) in the iteration step (s_i) (Kohonen, 1982). A use of computer program techniques was needed for performing this kind of an iterative calculation.

Several options were available to perform the iteration (DZone, 2019; NG and Soo, 2017; Vettigli, 2018; Wittek et al., 2017). The most convenient for elementary purposes was the ‘MiniSom’ python function developed earlier (Vettigli, 2018). Shortly, it required four instructions: 1. Importing and normalizing the initial experimental or simulation data, which consisted here of all the cell speeds together with their respective COMs, 2. Specifications for the node map; the size of the node map matrix, learning coefficient, initial spread of neighborhood function as gaussian, and the neighborhood function, which was here the Mexican hat’ as long as to gather all 2D nodes inside an oval, 3. principle component analysis style weighted BMUs done in a random-like fashion, and finally 4. the primary iteration, which was by default a random training of 5000 times, since more than 300 did not enhance the accuracy significantly (Vettigli, 2018).

The other parameter values for SOM were: $\sigma=4$, learning rate=0.5, and the size of the random seed was 10. The routine took about 15-40 minutes depending on the original size of the initial data, in other words the memory space used by the computer. All SOM nodes were plotted with gradually increasing colors in respect to their closeness to the original data (Fig. 16). To find the cell speed regions with SOM, I identified all COMs and deduced the changes in the speed trends from these COMs. This was especially needed if the data was noisy, such as the segmented data of the kidney organoid experiments (Tikka, 2019c). In that case, the noise-cell coordinates were back-deduced to correspond their respective real cells by using the mean areas and speeds of the cells ($50 \mu\text{m}^2$, $0.15 \mu\text{m}/\text{min}$) (see ‘7.3.4. Shortened Python Code of Self-Organized Maps’). In other words, SOM improved the original segmentation accuracy and its interpretation (see ‘2.5.3 Microscopy, Image Processing and Data Segmentation’) (Stegmaier et al., 2016).

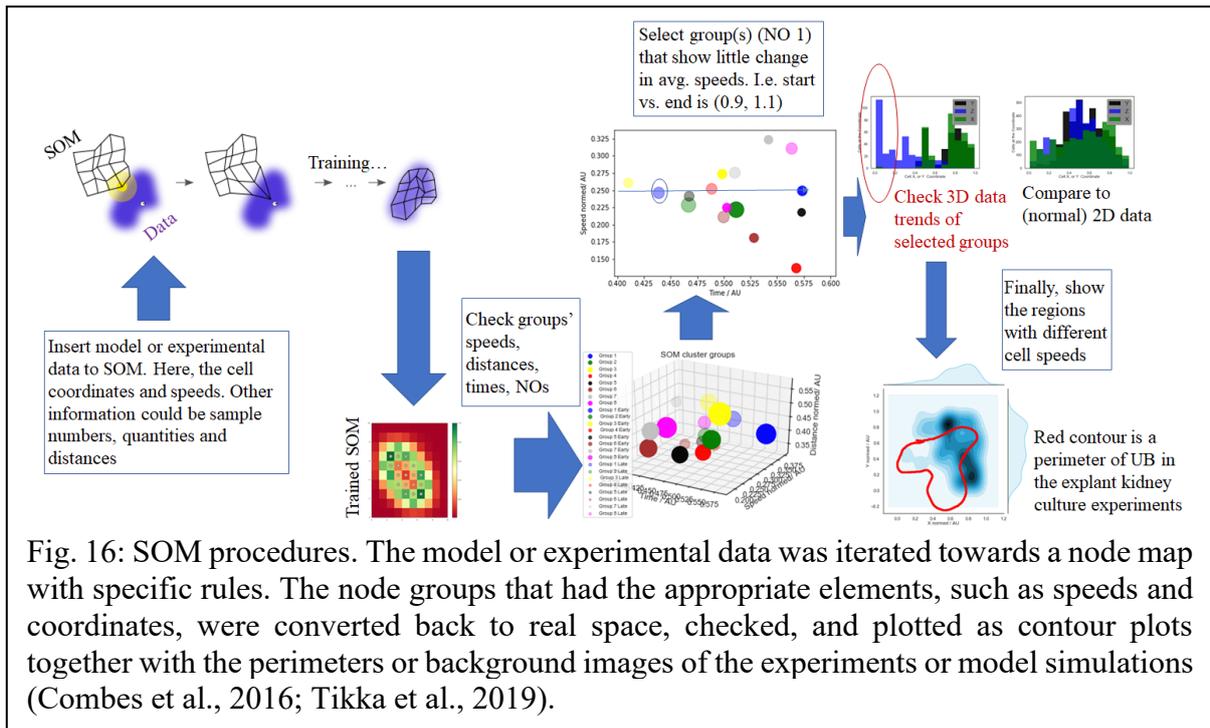


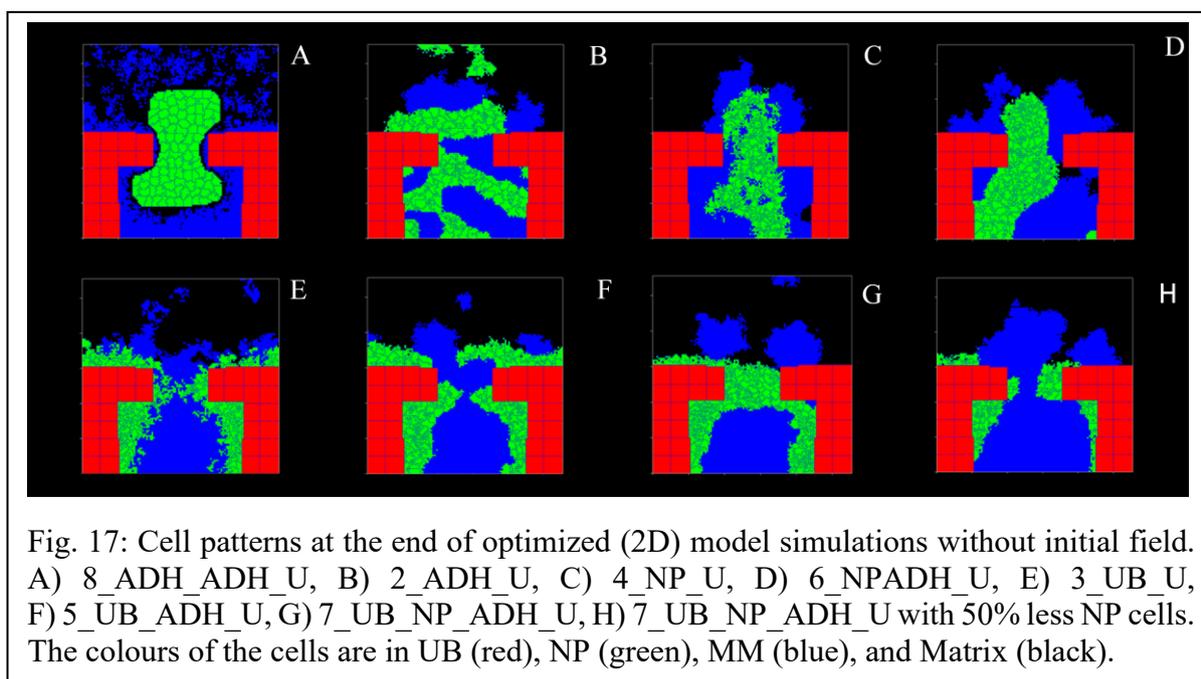
Fig. 16: SOM procedures. The model or experimental data was iterated towards a node map with specific rules. The node groups that had the appropriate elements, such as speeds and coordinates, were converted back to real space, checked, and plotted as contour plots together with the perimeters or background images of the experiments or model simulations (Combes et al., 2016; Tikka et al., 2019).

SOM node matrix was devised as size as 10 times 10 due computational efficiency (Vettigli, 2018). The method typically found after the iterative training 10-30 (from 100) nodes that were closest to the original data. Subsequently, only eight closest groups to the data were chosen (see '7.3.4. Shortened Python Code of Self-Organized Maps'). I further selected the groups, where the cell speeds were 95 % similar in the beginning and in the end (Fig. 16). This was because the regions of interests were the ones where the cell speeds remained approximately similar across the entire experimental or simulated time. There were usually three groups from the previous eight. Finally, I remapped these groups back to normalized space by recasting them to their original COMs, and plotted the speed contours with the representative images of the corresponding experiments or simulations in the background.

3. RESULTS

3.1. Initial Model Parameter Estimation

The initial parameter settings for λ_A , λ_V , V_t , A_t , and J (see Eq. 2; Table 1) defined the reference model (1_REF; c.f., Fig. 9) together with the predefined spatial relationships (Tikka et al., 2019). Starting from these settings, the contact energy coefficients (J) between NP, MM, UB cells and the matrix were modified, yielding the adhesion-based models (2, 8; Fig. 17A/B).



The purpose of this modification was to identify the range of contact energy coefficients required to induce cell aggregations, and to explore the aggregation behaviour of NP and MM cells in the CC3D model space (Andasari et al., 2012; Combes et al., 2016; Osborne et al., 2017; Swat et al., 2009). Examples of the initial aggregation tests have been given earlier in the methods (Figs 7, 8).

Next, I investigated the impact of chemotaxis and its source on cell patterning. Accordingly, I assumed the secretion of chemoattractants from either UB cells and/or NP cells themselves (Fig. 15). To that end, I modified the parameters related to chemoattractant secretion (S), diffusion (D), degradation (γ), and chemotaxis strength (λ_{CL} ; see Eqs 2, 3). First, NP cells (model 4; Fig. 17C) were assumed as the alternative source of the chemoattractant secretion with the intention to derive plausible value ranges for these parameters and explore the effect

of chemotaxis on cell clustering. Secondly, the combined effects of the contact energies and chemotaxis were considered in the following model (6; Fig. 17D). Finally, the same procedure was done with UB cells in the ensuing models (3, 5; Figs 17 E/F) and the secretion sources combined in the final model (7; Figs 15C, 17 G/H).

3.2. Formation of Patterns

The eight model variants yielded distinctly different final cell and chemoattractant patterns (Fig. 15, 17). The initial conditions and parameters had only a limited effect to the final patterns in 3D (Figs 18, 19). For this reason, the following description refers to the model patterns obtained with optimized simulations using random initial cell distribution and no pre-formed chemoattractant gradient, unless otherwise specified. The results of the other initial conditions and 2D patterns are given in (Tikka, 2019a).

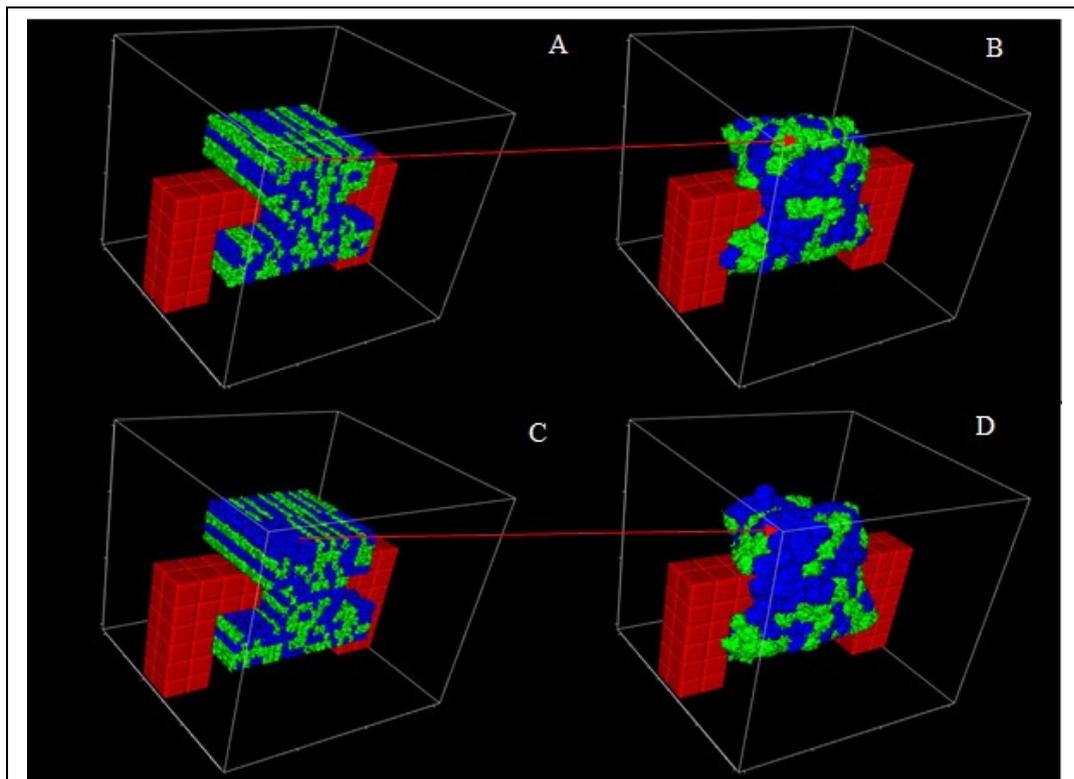
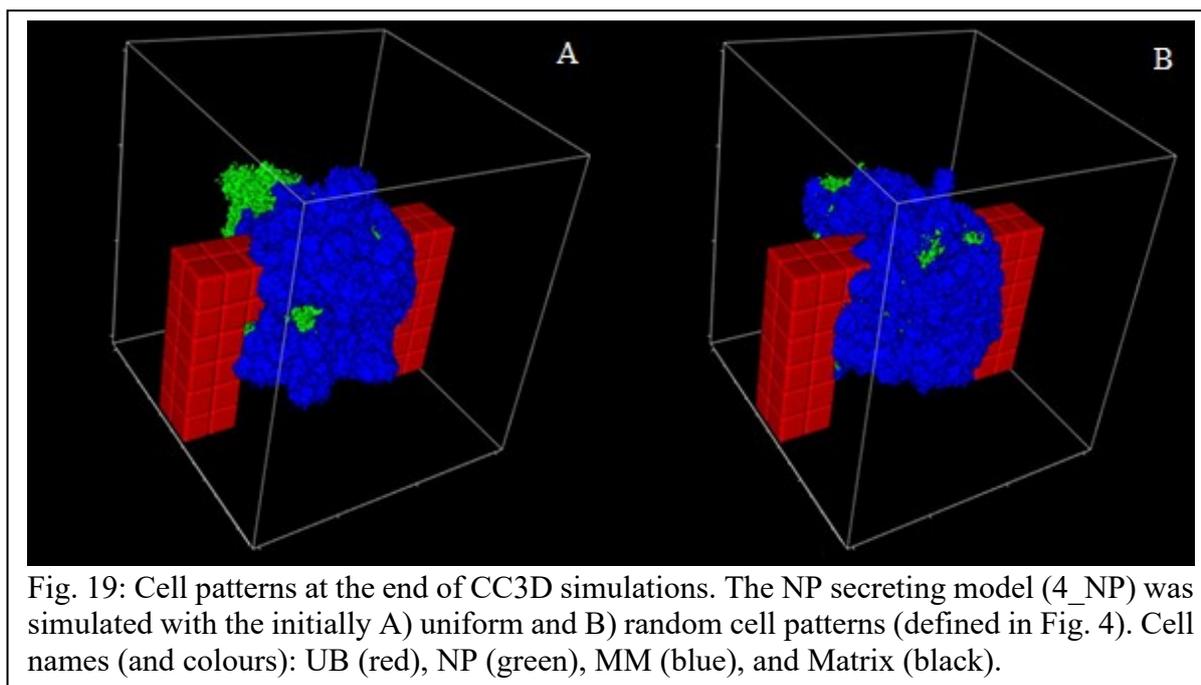
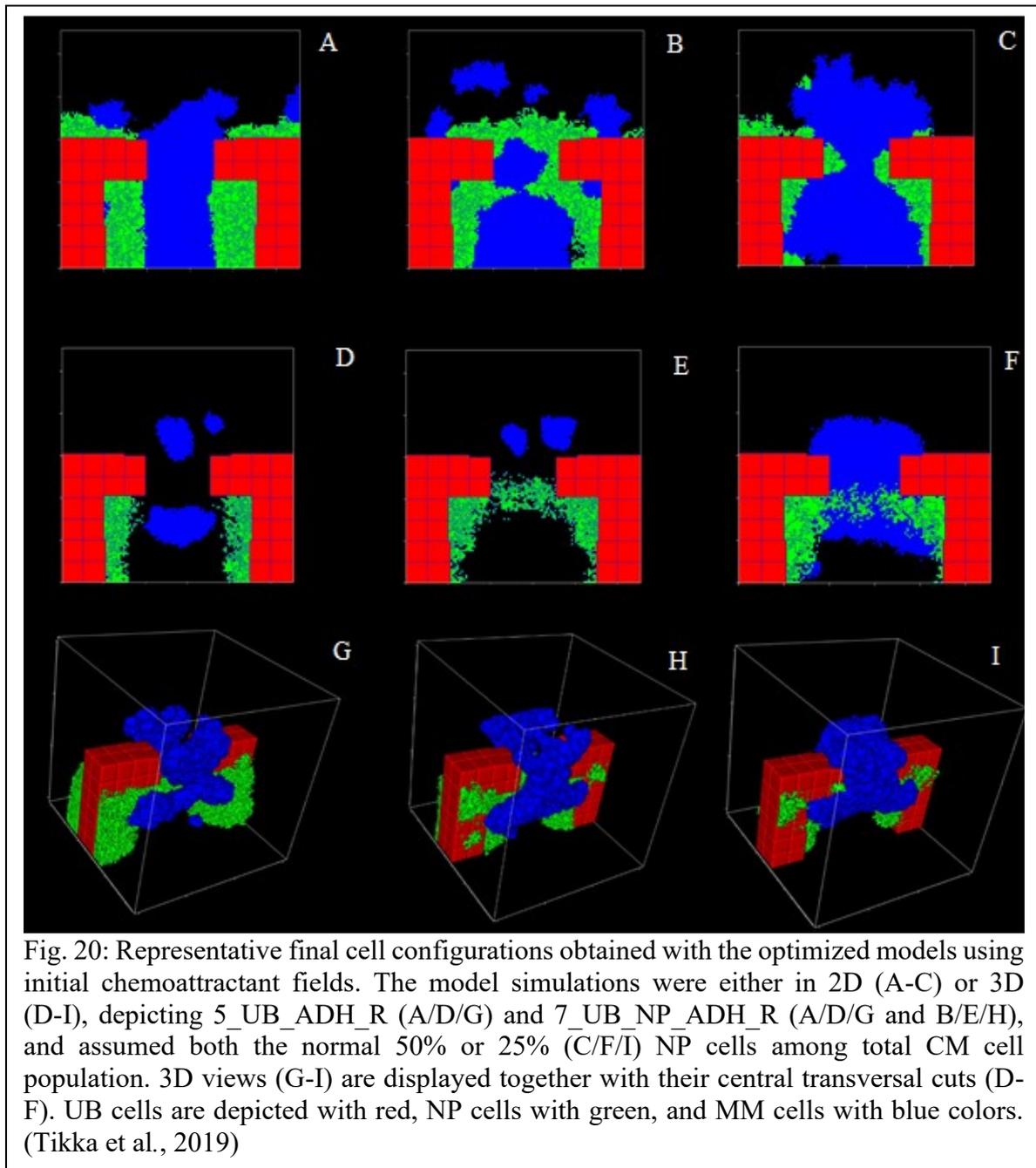


Fig. 18: The start (A/C) and end (B/D) cell patterns of CC3D simulations of the reference model (1) with two random initial positions (A/C). The contribution of the initial to the final aggregation patterns (B/D) are indicated with red arrows. Cell names (and colours): UB (red), NP (green), MM (blue), and Matrix (black).



The reference model yielded a cell pattern without coherent clustering over time (Figs 9, 18). In the models assuming adhesion differences between MM, NP, or other cells, but no chemotaxis, streak- or ball-like clusters emerged throughout the inter-UB area (Figs 12, 17A/B). The models involving chemoattractant secretion by NP cells with and without cell adhesion differences resulted in cell aggregates between the UB tips without adherence to the UB surface (Figs 13A/B, 17D/F). In the models involving UB cell chemoattractant secretion, NP cells aggregated along the UB surface, with a preference to the corner regions (Figs 14, 17E-H.). Directed migration and preferential aggregation of NP cells in the UB corner was observed in the previously mentioned models (3, 5, and 7; Figs 14, 17E-H), resembling the formation of PTA in the corner region during nephrogenesis (Fig. 1C). The most consistent formation of NP cell clusters resembling PTAs was observed with the most advanced model (7; Fig. 17G/H). It involved the chemoattractant secretion by both UB and NP cells and a stronger adhesion between NP and MM cells relative to other models (Figs 17, 20B/E/H, Table 2). The video of the model (7) behaviour can be viewed online (Tikka, 2019e).



3.3. Experimental Studies

The cell movement analysis performed on the data of the explant culture model experiments of Combes et al. (2016) and the kidney organoid model experiments of this work (Tikka et al., 2019) are provided in (Fig. 21) (Combes et al., 2016; Lawlor et al., 2019; Lefevre et al., 2017; Tikka et al., 2019).

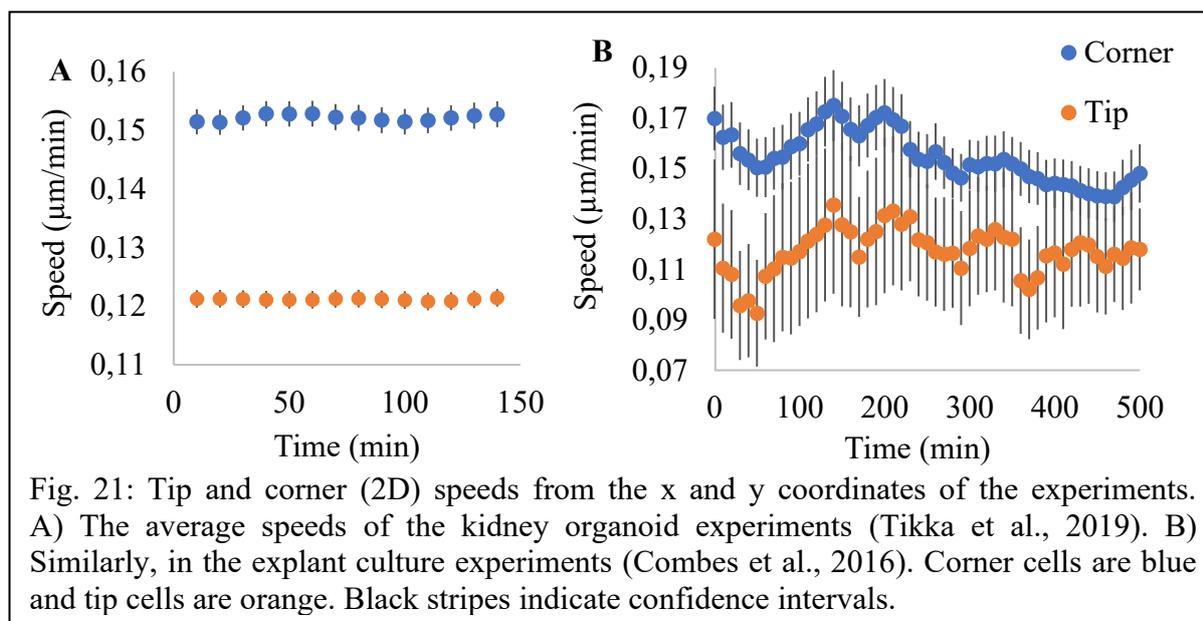
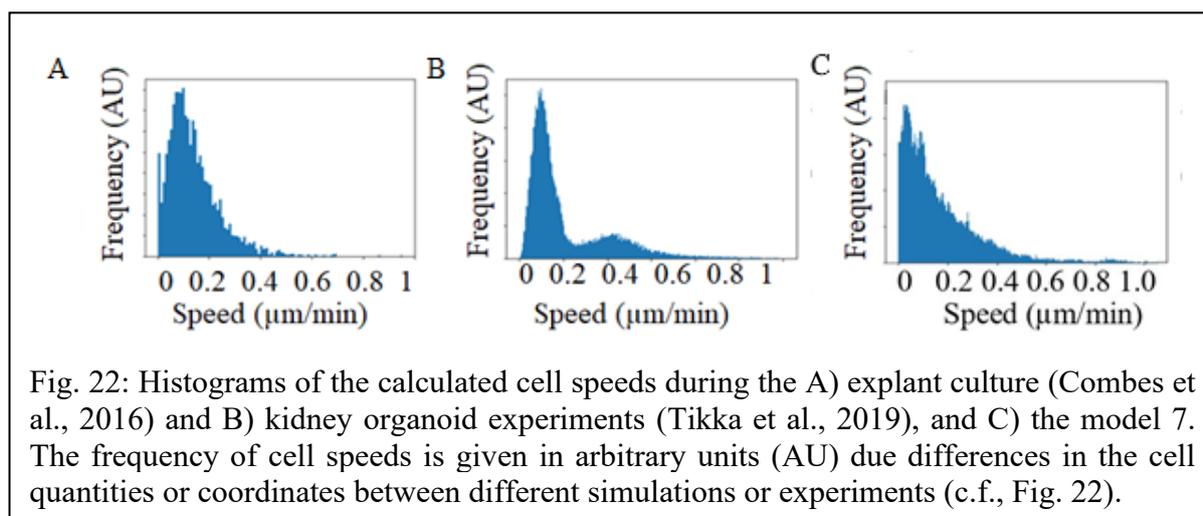
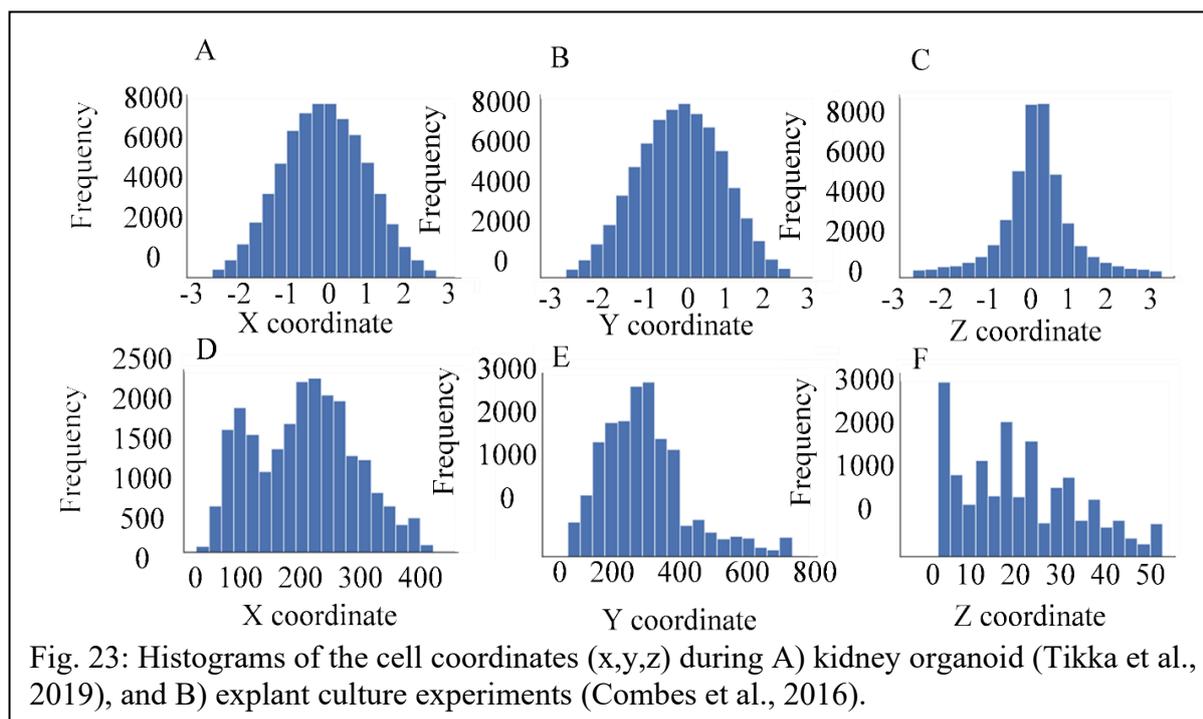


Fig. 21: Tip and corner (2D) speeds from the x and y coordinates of the experiments. A) The average speeds of the kidney organoid experiments (Tikka et al., 2019). B) Similarly, in the explant culture experiments (Combes et al., 2016). Corner cells are blue and tip cells are orange. Black stripes indicate confidence intervals.

The observed overall NP cell speed averages were $0.15 \pm 0.02 \mu\text{m}/\text{min}$ in the explant cultures, and similarly with MM cells' $0.13 \pm 0.01 \mu\text{m}/\text{min}$ in the kidney organoid. The histograms of the experimental cell coordinates and their speeds are shown in (Figs 22, 23). The distribution of the respective cell positions was largely normal (Fig. 22). However, as can be seen in Fig. 22F, the z axis data of the explant culture experiments was skewed. Nevertheless, the histograms of the experimental speeds were left-bound (Figs 23A/B). The data of model 7 is given as a comparison to the experiments (Fig. 23C).



While cell speeds in the explant culture fluctuated considerably more than in the kidney organoid experiments, in both experimental settings the two cell types moved at different rates depending on their location relative to the UB tip (Fig. 21). In the corner region, the average cell speeds were $0.16 \pm 0.02 \mu\text{m}/\text{min}$ in the explant culture experiments and $0.15 \pm 0.01 \mu\text{m}/\text{min}$ in the kidney organoid (Combes et al., 2016). Average speeds in the tip region were $0.12 \pm 0.01 \mu\text{m}/\text{min}$ in both experimental settings. The calculated averages of corner and tip distances at the end of the experiments are given at (Table 3).

Table 3: The final cell speed and distance values averaged from the last 10% of the values in the experiments (Combes et al., 2016; Tikka et al., 2019). The 3D values of the explant culture experiment have been estimated due to the abnormal z axis values (c.f., Fig. 22F).

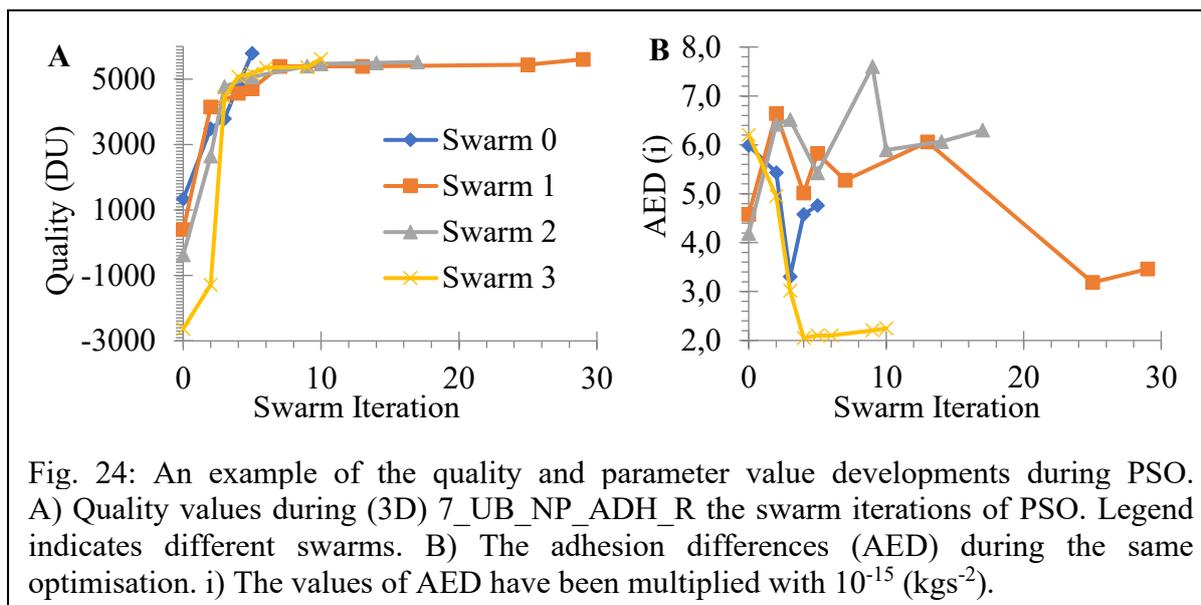
Experiment	Corner speed ($\mu\text{m}/\text{min}$)	Tip speed ($\mu\text{m}/\text{min}$)	Corner distance (μm)	Tip distance (μm)
Explant Culture (2D)	0.125 \pm 0.033	0.122 \pm 0.032	28.45 \pm 0.19	19.8 \pm 0.14
Kidney Organoid (2D)	0.104 \pm 0.005	0.092 \pm 0.004	25.41 \pm 0.55	21.34 \pm 0.46
Explant Culture (3D) ¹	0.154 \pm 0.033	0.150 \pm 0.032	32.07 \pm 0.22	22.77 \pm 0.16
Kidney Organoid (3D)	0.129 \pm 0.005	0.113 \pm 0.004	28.14 \pm 0.64	24.39 \pm 0.54

3.4. Comparing Cell Behaviours Between Models and Experiments

3.4.1. Estimation of Final Parameters with Particle Swarm Optimization

The parameter ranges for PSO were chosen according to the initial model parameter estimates and the additional PSO algorithms (see ‘2.4.2 Constraints and Ranges of Parameters in Particle Swarm Optimization’ and ‘7.3.2 Shortened Python Code of Particle Swarm Optimization Method’) (Tikka, 2019d). All models were subjected to PSO, where the optimization procedure aimed at maximising the amount of NP cells at the UB surface, while simultaneously aligning NP cell speeds in the model to the cell speeds observed experimentally in the explant culture setting, as mentioned in the methods. The final model parameter values obtained by the PSO technique were given in (Table 2).

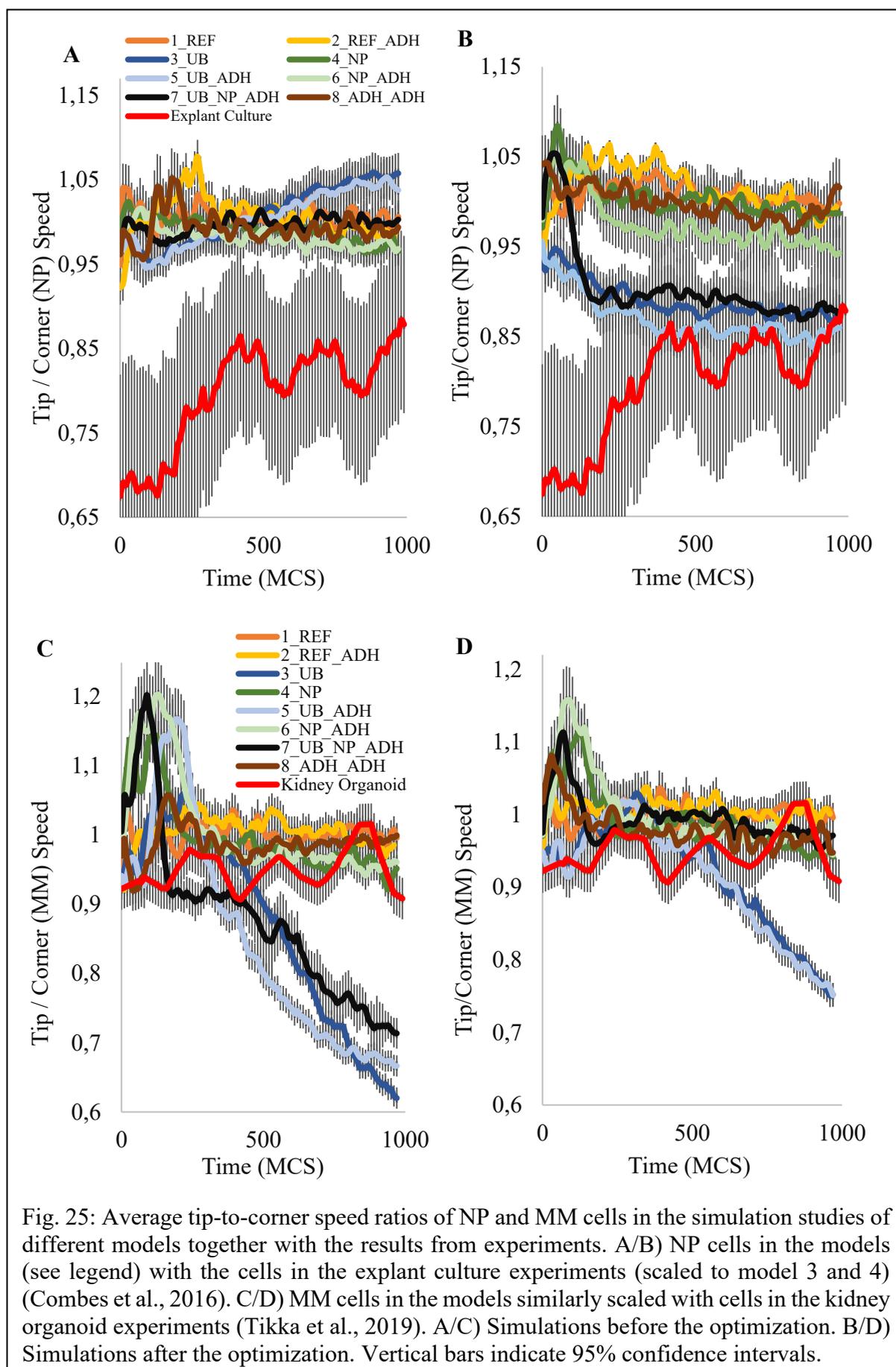
The improvement of the models achieved by the application of PSO was illustrated by the Best Quality Values (Table 2; lower numbers indicating better quality). The best model quality was obtained with the model 7, while the other models showed either substantially lower quality or a spuriously high quality without matching the experimental situation. This applied in particular for the NP secreting models (4 and 6) and the adhesion model (8). In the latter model, NP cells did not aggregate towards the UB surface in the first place. Examples of running PSO quality values and parameters in the model 7 are given in the following graph (Fig. 24), and in the supplemental table (Table S1).



Nevertheless, the optimized adhesion difference (J) values for each model, except for models 1, 3, 4, and 8, were close to non-optimized ones. Markedly, 5_UB_ADH had (J of) 6.68, which is in the similar magnitude as 3_UB's 5. Even so, 8_ADH_ADH had the greatest variety of adhesion differences as expected; for example in the uniform model with: 0.50279 ('NP and NP') to 47.808 ('Medium and NP'), and random model with: 0.50203 ('NP and NP') to 37.680 ('MM and MM'; Table 2). Finally, the chemotaxis lambda values (λ_{CL}) were higher in the NP secreting models than in the UB secreting ones.

3.4.2. Speeds of Cells at Tip and Corner Regions

The slower relative cell movements of both MM and NP cells around the UB tip region was also apparent, when expressed as the tip-to-corner speed ratio, which was below one (1) for most of the observation time (Fig. 25). Average cell speeds in their average coordinates are given in (Fig. 26). In addition, averages of the final corner and tip speeds of the simulated NP and MM cells are given in the supplementary tables (Table S2, S3).



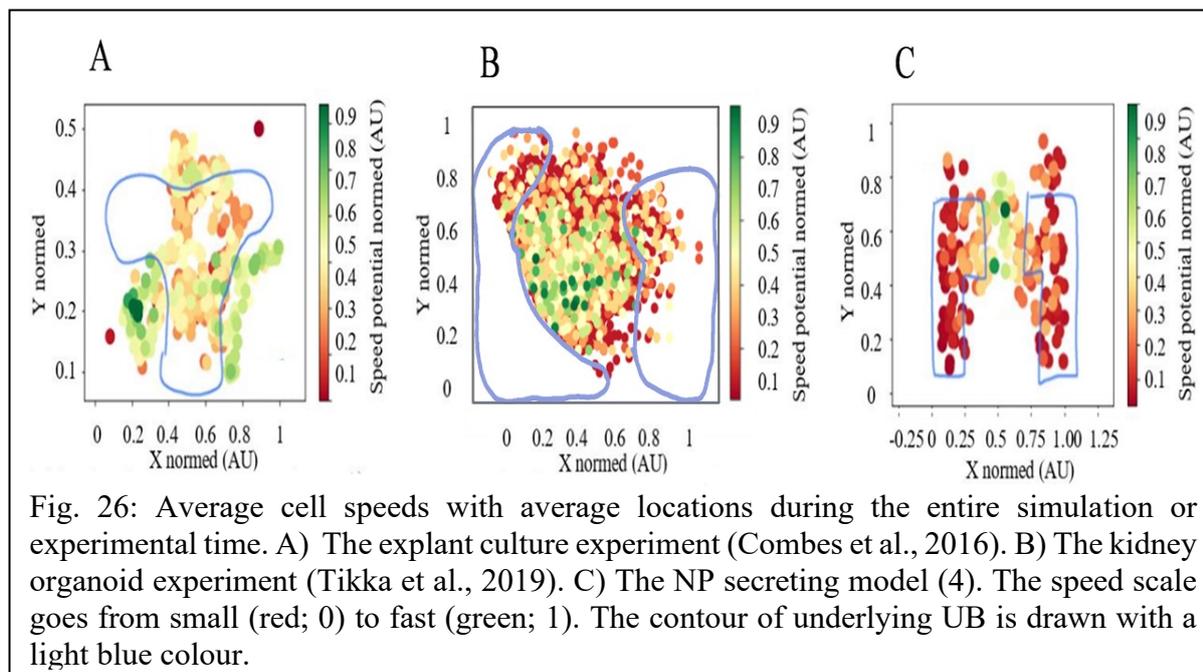


Fig. 26: Average cell speeds with average locations during the entire simulation or experimental time. A) The explant culture experiment (Combes et al., 2016). B) The kidney organoid experiment (Tikka et al., 2019). C) The NP secreting model (4). The speed scale goes from small (red; 0) to fast (green; 1). The contour of underlying UB is drawn with a light blue colour.

The optimization procedure resulted in a more accentuated difference in the corner and tip cell speeds of the models (3, 5, 7) involving chemoattractant secretion by the UB as seen in (Fig. 27). At the same time, the tip-to-corner speed ratio of NP cells in these models decreased with time, aligning with NP cell speed ratio observed towards the end of the explant culture experiments (c.f., Figs 25B, 26A). By contrast, MM cell speeds in the kidney organoid experiments were better matched by the optimized NP secreting models (4 and 6; c.f., Figs 26B/C, 27D). In the secreting models (3-7), the overall speeds of NP cells both in the tip and corner regions, consistently exceeded those of MM cells (0.13 ± 0.03 vs. 0.03 ± 0.02 $\mu\text{m}/\text{min}$; Fig. 27).

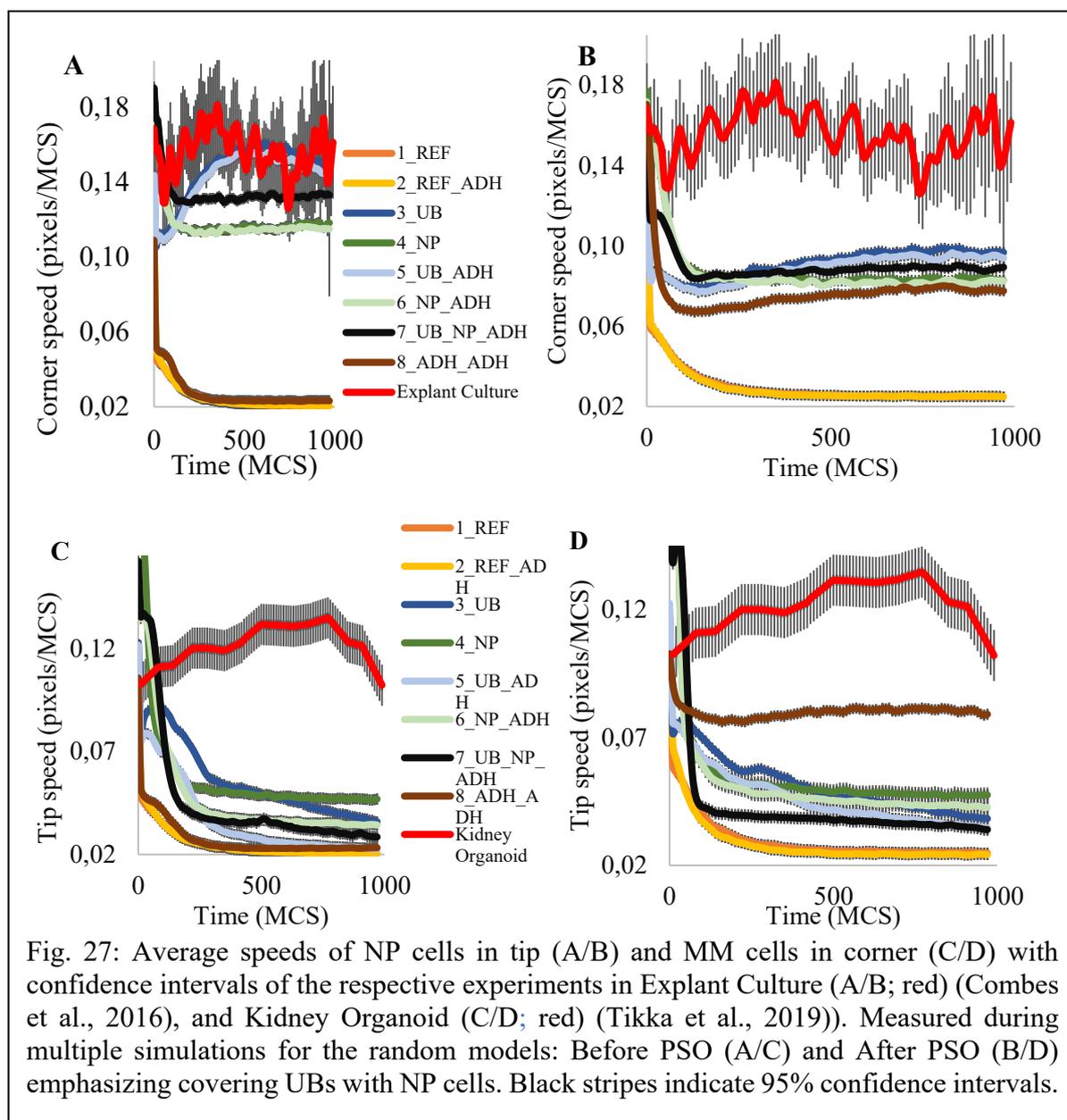
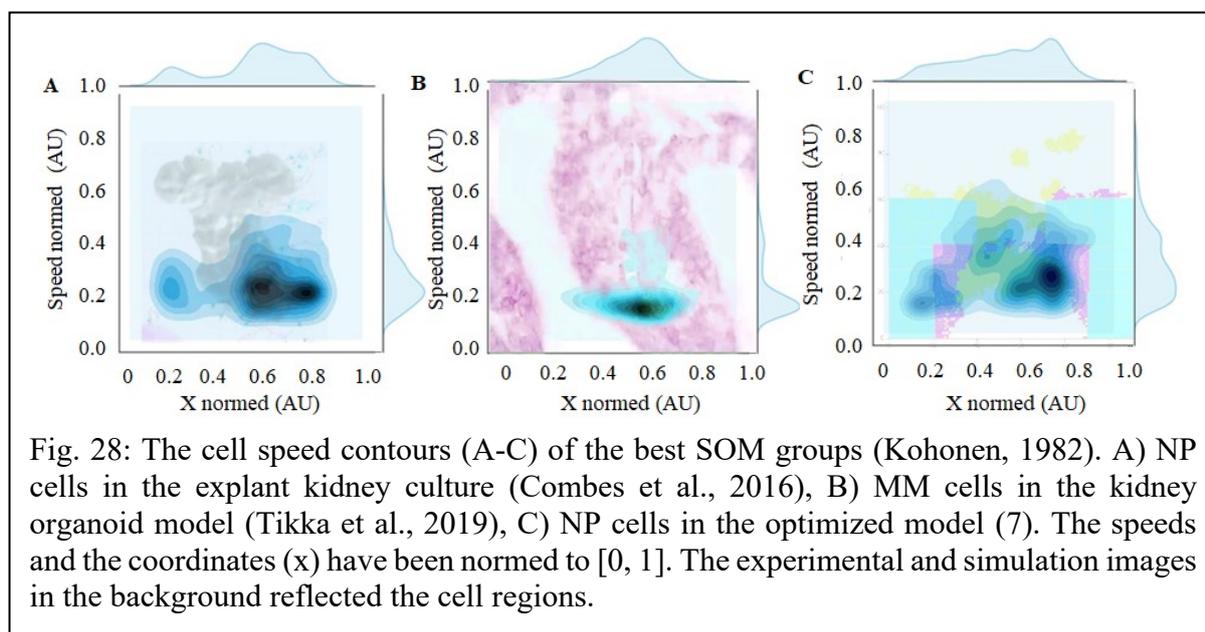


Fig. 27: Average speeds of NP cells in tip (A/B) and MM cells in corner (C/D) with confidence intervals of the respective experiments in Explant Culture (A/B; red) (Combes et al., 2016), and Kidney Organoid (C/D; red) (Tikka et al., 2019)). Measured during multiple simulations for the random models: Before PSO (A/C) and After PSO (B/D) emphasizing covering UBs with NP cells. Black stripes indicate 95% confidence intervals.

3.4.3. Self-Organizing Maps Applied to Speeds of Cells

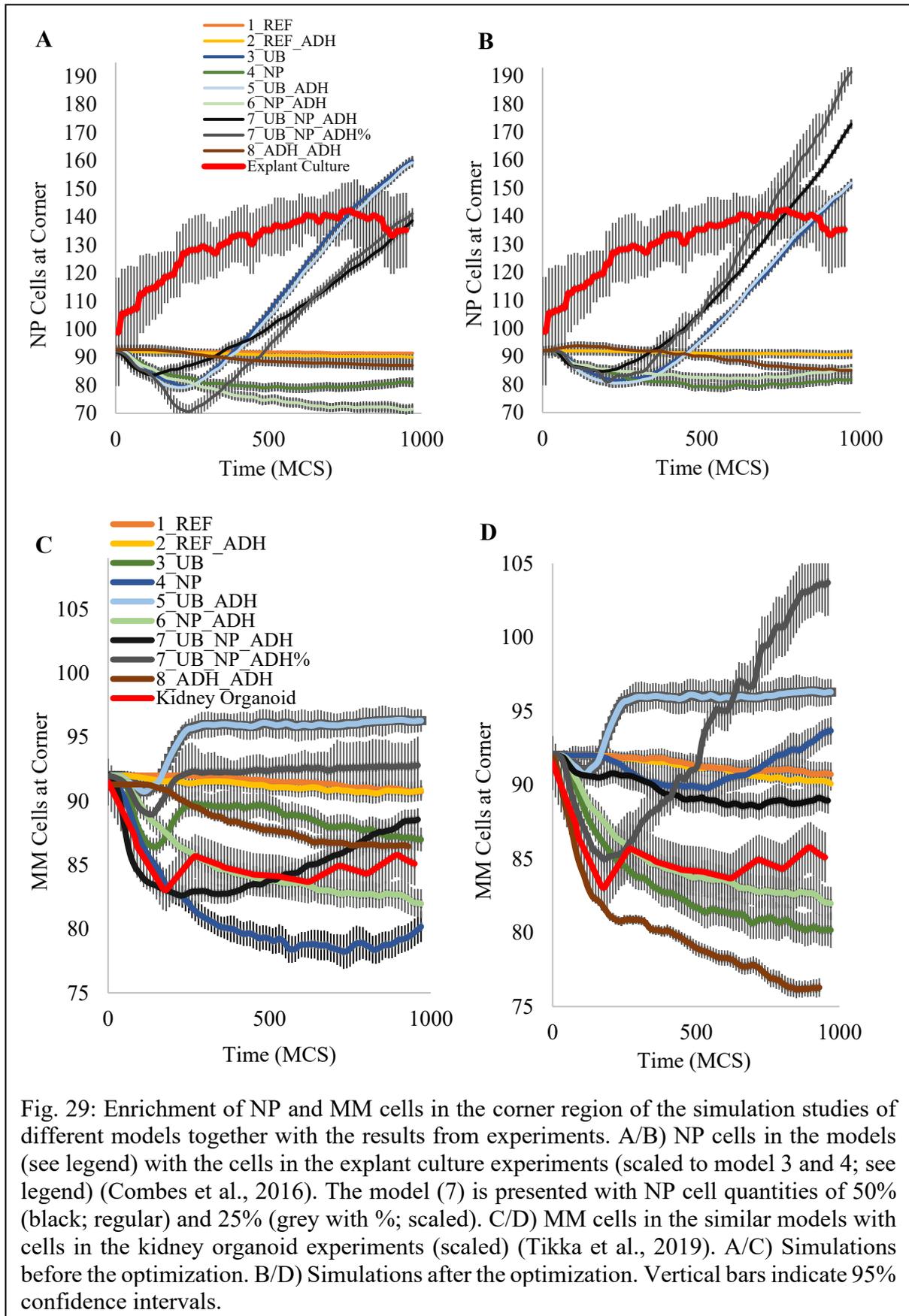
According to 2D SOM analysis (Fig. 28), stable speeds in the explant culture data were $0.19 \pm 0.02 \mu\text{m}/\text{min}$ in the corner, and $0.14 \pm 0.01 \mu\text{m}/\text{min}$ in the tip region (Combes et al., 2016; Kohonen, 1982). 2D analysis was performed, because 3D data, or more specifically z axis data, of the explant culture experiments was found regularly skewed (Fig. 22F). Stable cell speeds of the kidney organoid data calculated by 3D SOM analysis were $0.25 \pm 0.02 \mu\text{m}/\text{min}$ in the corner and $0.18 \pm 0.02 \mu\text{m}/\text{min}$ in the tip region, respectively. The different speed of motion of

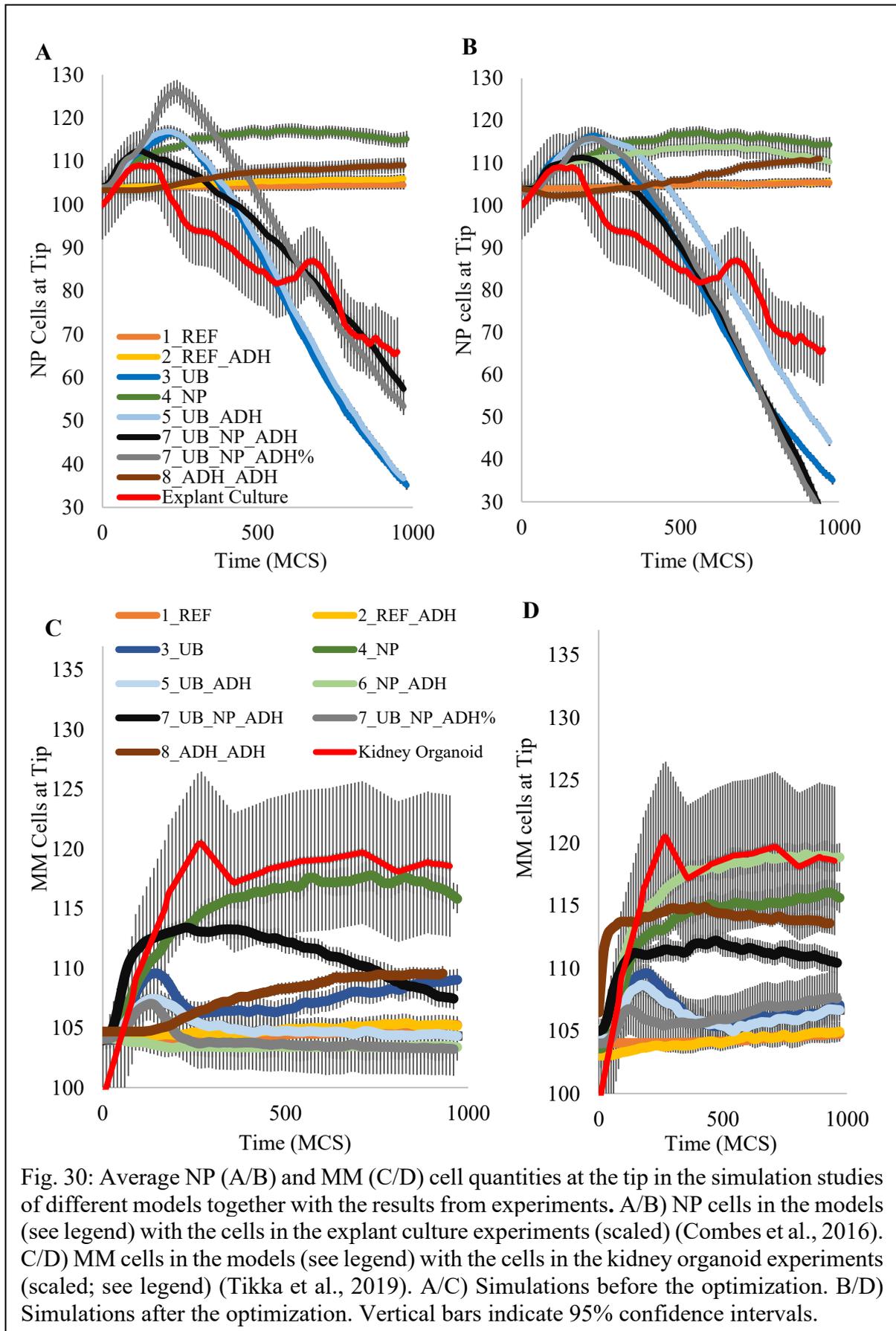
cells in the tip and corner regions is illustrated in the coloured speed contours of the transformed SOM plots (Fig. 28).



3.4.4. Quantities of Cells at Tip and Corner Regions

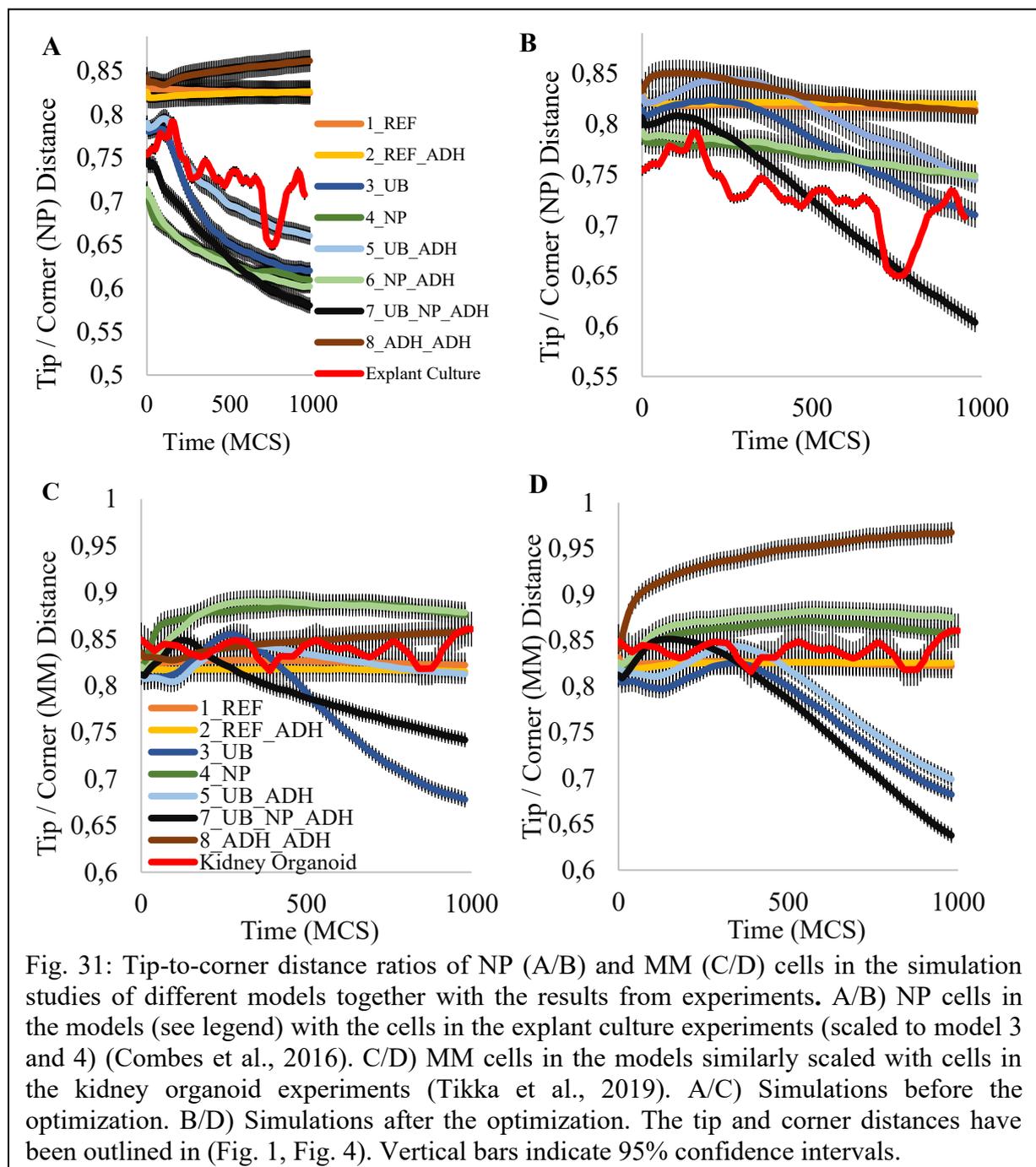
The number of NP or MM cells accumulating in the corner region was increased by the optimisation procedure with the UB secreting models (Fig. 29). Similarly, the development of cell quantities in the different models at the tip region have been given in (Fig. 30).





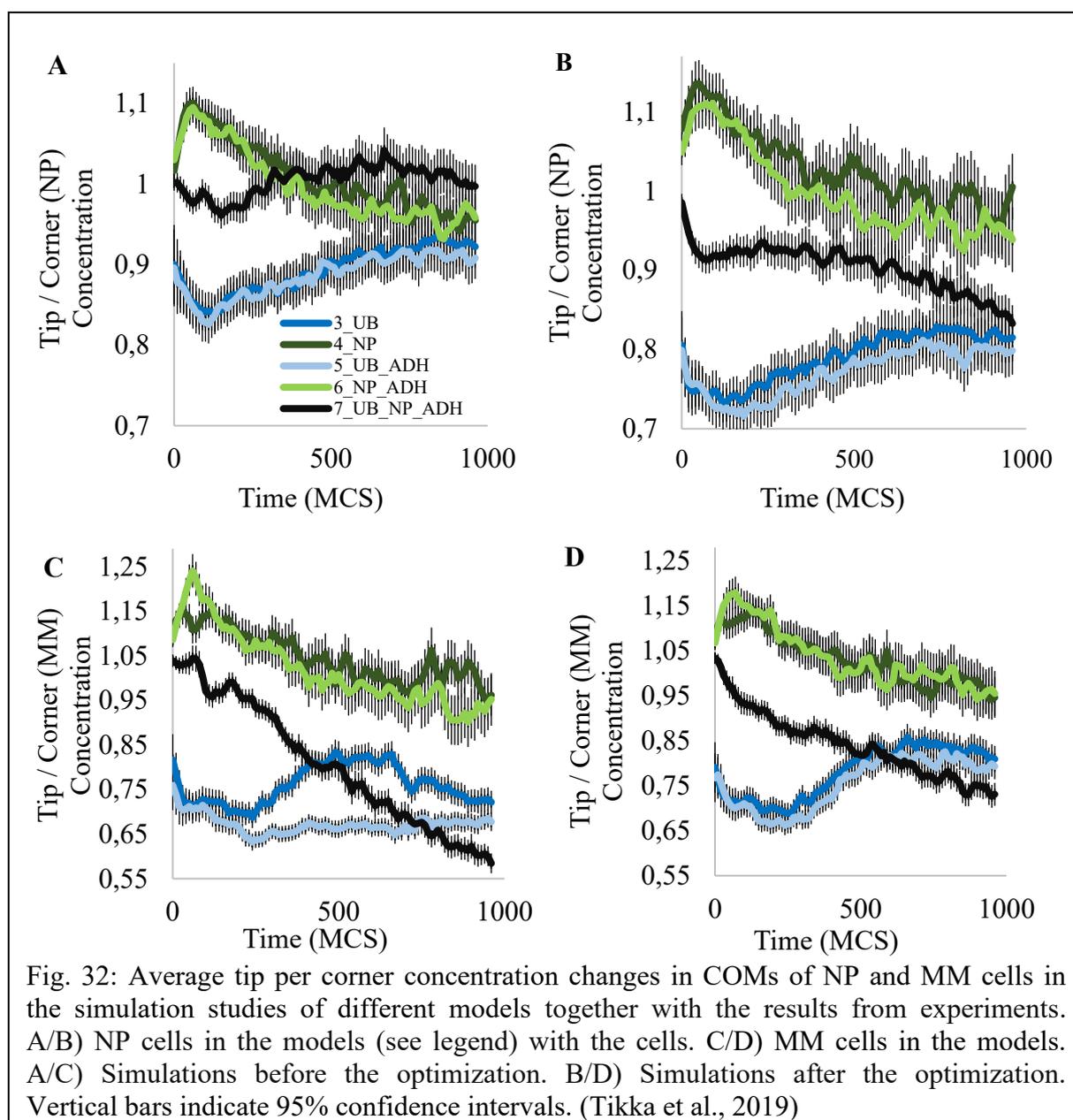
3.4.5. Distances of Cells to Tip and Corner

In the optimized models, the calculated total distance travelled by NP cells during the simulation period was between 90-160 μm . Regardless, the relative tip-per-corner cell distances of around 0.66 with the explant culture experiments were closest to the UB secreting models (3, 5, 7; c.f., Figs 7, 8, 20, 31A/B) (Combes et al., 2016). Even though, the relative distances in the kidney organoids were best recapitulated by the optimized adhesion-based or NP secreting models (1, 2, 4, 6; c.f., Figs 7-13, 31C/D).



3.4.6. Chemoattractants in Cells at Tip and Corner Regions

The average tip-to-corner concentration changes in the cells have been given in (Fig. 32). The NP secreting models exhibited relative higher amounts of chemoattractant than the UB secreting ones (c.f., Fig. 15, 32B/E). Comparatively, the tip-per-corner concentrations highlighted the differences in the UB and NP cell secretion magnitudes between the UB and NP secreting models (Fig. 32). The final chemoattractant patterns of the models have been depicted at (Fig. 33).



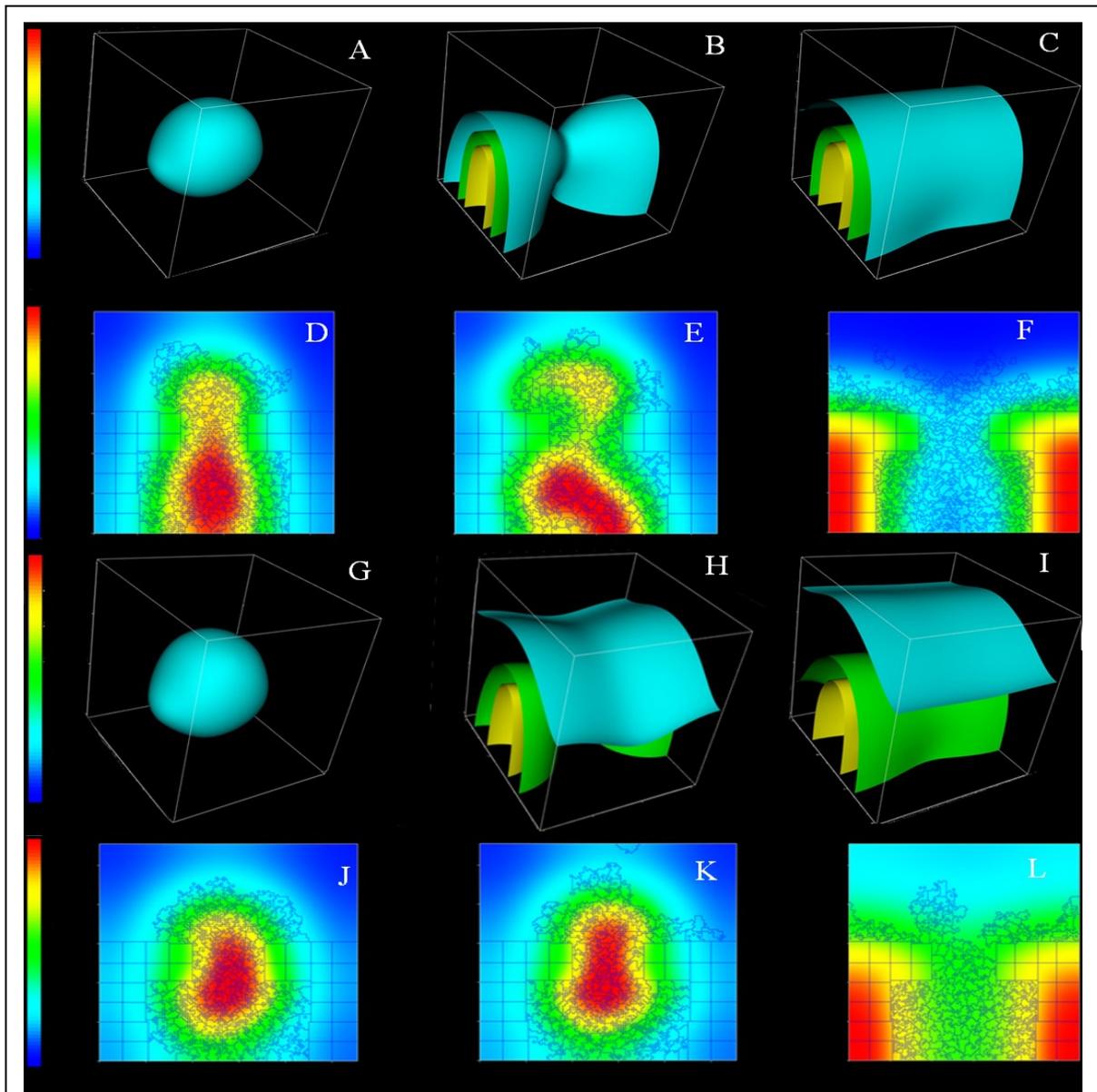
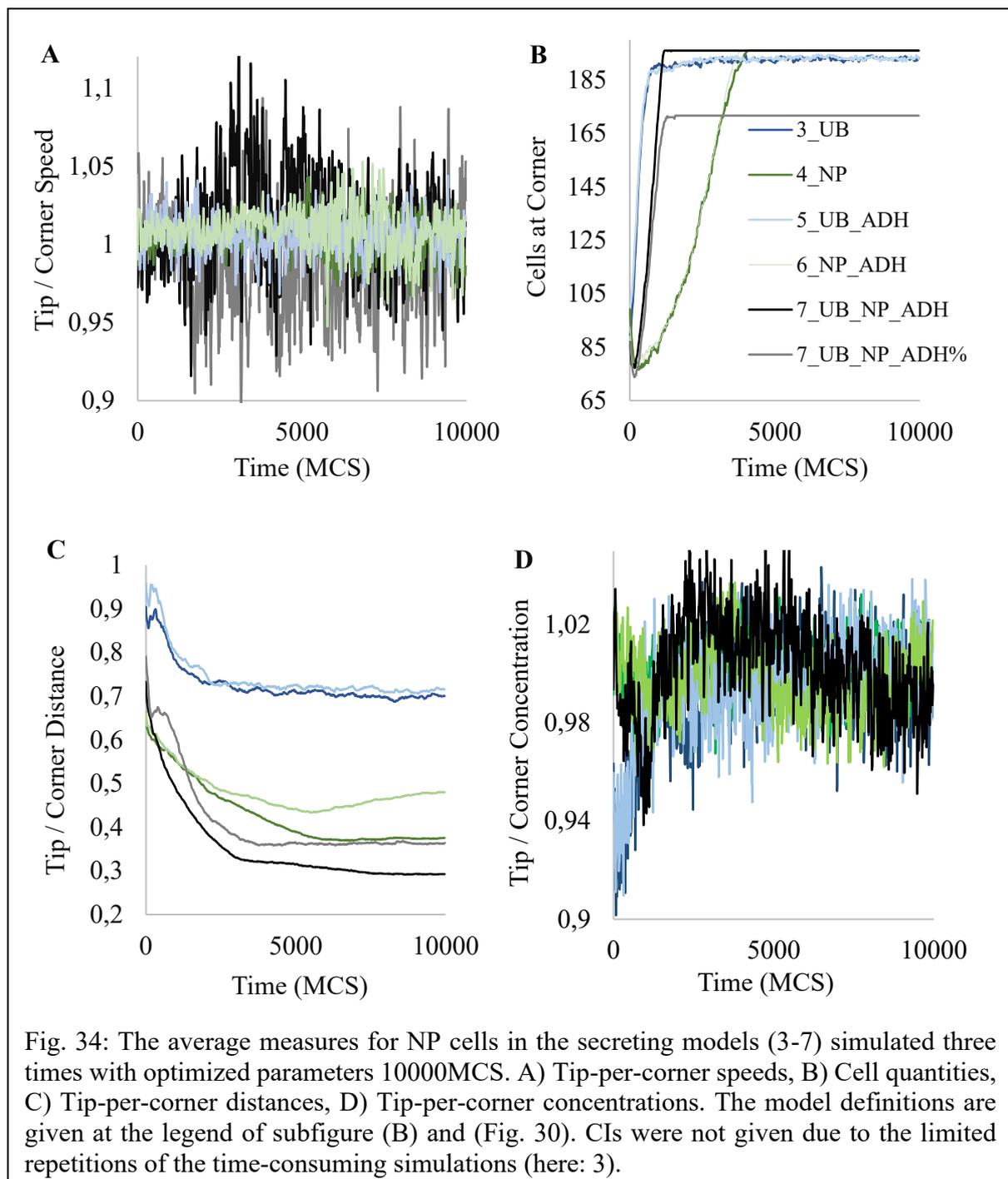
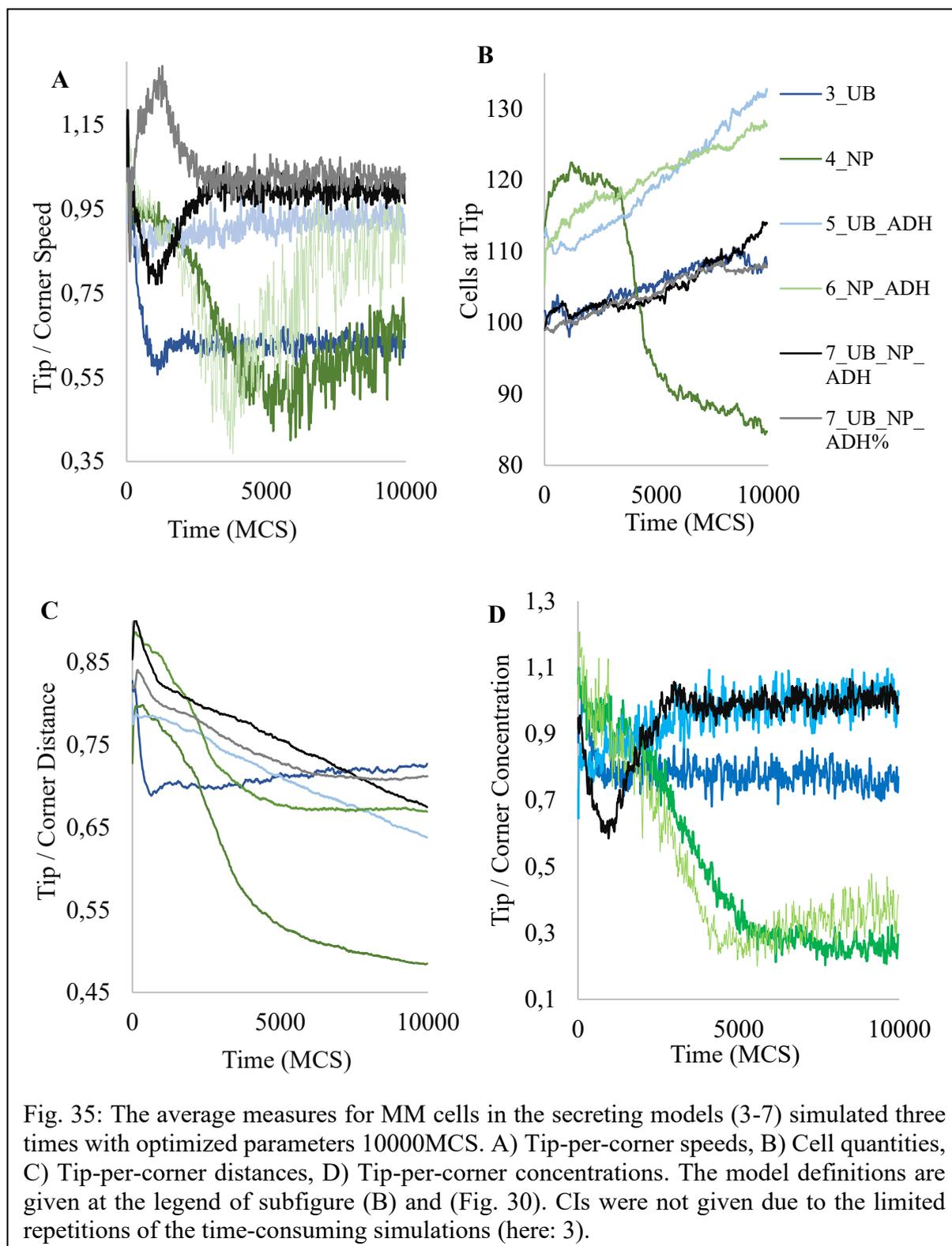


Fig. 33: The final chemoattractant patterns of all models in 2D (D-F, J-L), and 3D (A-C, G-H). The NP secreting models are at (A/D/G/J), The UB secreting models are at (B/E/H/K), and the UB and NP secreting models at (C/F/I/L). A)-F) Model outcomes without the initial field. G)-L) Model outcomes with the initial field. The relative distributions (0,1) of chemoattractant vary between small (blue) and high (red). See Fig. 4 and for more information about the chemoattractant.

Correspondingly, I also tested the model simulations with 10000MCS, in order to assess the impact of longer simulation times to the spread of the chemoattractant and other measurement values as seen in (Figs 34, 35) (c.f., ‘2.2.2 Selection of Chemoattractant Saturation Type’). The impact of the longer simulation time to the NP secreting model (4, 6) simulations was that the chemoattractant quantities in MM cells at the corner region increased significantly more than during the regular simulation time (1000MCS vs. 10000MCS; c.f., Figs 32D, 35D).

By contrast, the UB secreting models drove NP cells fastest to the corner region (Fig. 34B). Similarly, the tip-per-corner speeds of both cells types in all models, except in the UB secreting model (3), appeared to converge towards the value of one (1; Figs 34A, 35A). The same general trend of the secreting model (7) was seen in the tip-per-corner distances, where NP cells were closest to the corner region after the simulations (Fig. 34C). However, this was not necessarily the case for MM cells (Fig. 35C).





4. DISCUSSION

In this work, I used a computational modelling approach to explore the biophysical mechanisms driving the committed nephron progenitor (NP) cells to move towards and form the pre-tubular tubular aggregates (PTA), the early nephron precursor structure during kidney organogenesis (Little, 2012; Tikka et al., 2019). The simultaneous movements and aggregations of the cap mesenchyme (CM) cells were studied using a Cellular Potts model, which was modulated to simulate the relative impact of chemotaxis and cell-cell adhesion forces (Allena et al., 2016; Belmonte et al., 2016; Swat et al., 2012). The movements of NP cells towards the corner regions of the branching ureteric bud (UB) and the formation of the cell aggregates was best reproduced by assuming a combination of chemotactic and differential cell adhesion forces. The parameter estimates were validated and optimized by analysis of the cell behaviours in two experimental models of early nephrogenesis. I observed, both *ex vivo* and *in silico*, an accelerating speed of motion of committed NP cells as they migrate from the UB tip to the corner region.

CM cells have been described to move in a quasi-stochastic fashion between the corner and tip regions of the branching UBs following certain environmental cues (Combes et al., 2016; Lawlor et al., 2019). While the bulk of metanephric mesenchymal (MM) cells have been considered as static, an emerging subpopulation of MM cells, namely NP cells, was believed to move linearly from the UB tip to the corner region (Little, 2012). In order to reproduce these movement patterns *in silico*, several assumptions had to be made. These assumptions related to the initial spatial structure, cell quantities and properties. Equally importantly, they were founded on both experimental evidence and on the established insights of cellular biophysics.

General assumptions included the assertion that all, and only, NP cells were committed to the formation of PTA. The assumptions presumed also that the changes in the three main energies should stir the movements and aggregations of NP and MM cells. These energies were the contact surface energies, chemotaxis energy, and cell size changes. Yet, the impact of cell sizes was minimal. This was because the initial most expansive phases were discounted in the simulation. This left chemotaxis and cell-cell adhesion as the main energies to drive cell sorting between regions. Indeed, there has been substantial experimental evidence to support the notion that the sorting of NP cells between the tip and corner regions was established by cell-cell adhesion differences and both inductive and chemotactic molecular signalling from

UB epithelia (Albert and Schwarz, 2016; Brown et al., 2013; Karner et al., 2011; Lefevre et al., 2017; Ricono, 2008). Previous studies identified different CAMs such as cadherins, to drive the cell sorting (Junttila et al., 2015; Lefevre et al., 2017). Likewise, the extracellular signals inducing NP cell commitment involved the secretion of WNT11, BMP7, FGF9, and WNT9B, which upregulates WNT4 (Bohnenpoll and Kispert, 2014). NP cell induction has also been shown to activate of Notch and additional signalling pathways (Lindstrom et al., 2015; Perantoni et al., 2005). Concurrently, MM cells require SMAD1/5-mediated BMP signalling to transition towards a state in which they can receive the inductive cues (Brown et al., 2013). While WNT9B was the first secreted molecule attributed a role as a chemoattractant for NP cells, subsequent studies identified PDGF-AA, FGF8, BMP4 and CXCL12 as further potential effectors driving NP cell chemotaxis (Atsuta and Takahashi, 2015; Carroll et al., 2005; Grieshammer et al., 2005; Ricono, 2008). Recently, evidence has been provided that, NP cells within the UB tip region do not move in a linear fashion, but exhibit a nearly stochastic swarm-like behaviour (Combes et al., 2016). Moreover, NP cell commitment and migration towards the corner region may not be a unidirectional irreversible process (Lawlor et al., 2019). A subset of NP cells at the corner region were observed to migrate back to the tip region to re-enter the uncommitted MM cell pool, losing WNT4 expression. This behaviour was tentatively explained by a semi-stochastic cell movement with exposure of NP cells to different cues depending on their spatial position, with prolonged or additional signals to NP cells being required for persistent clustering in the corner region and PTA formation.

Subsequently, I performed various model simulations to test the impact of chemotaxis and adhesion driven cell sorting on NP cell trafficking and clustering towards the corner region, where PTA could form. The initial model setup and simulation studies were followed by a validation and optimization step, where the explant culture and organoid experiments were utilized to further align the model parameters to fit the cell movements and aggregations observed *ex vivo*.

As a result, my simulation studies support an important role of the chemoattractant gradients arising from the UB surface for a net directed NP cell movement. The model variants lacking UB chemoattractant secretion resulted in an ectopic NP cell aggregation. Moreover, differential cell-cell adhesion properties appear to be required for the formation of NP cell aggregates. Paracrine chemotactic signalling by NP cells may play a role in this process, as suggested by the optimal performance observed with a model combining adhesion differences with

chemoattractant secretion by both the UB and NP cells in the respective model (7) (Dahl et al., 2002; Dudley et al., 1999; Oxburgh et al., 2011; Wallner et al., 1998). My model also adequately recapitulated the semi-stochastic movement of NP cells around UBs. Both the average cell speed and their net travelled distance fitted the speeds and distances observed in the explant culture system. Notably with the model variant 7, one even observed the individual cells returning from the corner to the tip region, recapitulating the recent experimental findings (Lawlor et al., 2019).

Another remarkable finding was the faster NP cell speeds in the corner as compared to the tip region. This was obtained both in the analysis of the explant culture data and by computational modelling and confirmed by self-organized speed mapping. My model simulations indicated that the acceleration of NP cells approaching the UB corner was due to the high chemoattractant gradient present in this region. Although this may be true, I also tested the previous finding with extended simulation times (10000MCS). Indeed, under those circumstances all NP cells reached the corner region faster with the UB secreting models (3, 5, 7) than with the NP secreting models (4, 6).

I discovered during the initial construction of the models that the value of the chemotaxis secretion rate (S) multiplied by chemotaxis lambda (λ_{CL}) should be less than twenty, especially in the UB secreting models. If the product threshold was exceeded, one experienced a loss of virtual cells, which would arrive from the disruption of the virtual cell membranes. It is possible that the same may apply in early nephrogenesis. That is to say that the amount, source and location of the chemoattractant secretion, as well as the cells' responsivity to chemotaxis would require some degree of regulation.

The different magnitudes of UB and NP cell secretion highlighted the effect of the different propensities between the UB and NP secreting models. Particularly, the secretion gradient from UB cells swayed more prominently to chemotaxis of NP cells towards UB than the similar gradient from NP cells. Similarly, after optimization the concentration levels near NP cells became wider between the UB and NP secreting models. In contrast, the levels narrowed near MM cells. This primarily meant that, the amount of the chemoattractant near that NP cells increased and lowered near MM cells. In other words, the amount of the concentration near MM cells did not discharge chemoattractants from NP cells. In fact, the relative amounts of the chemoattractant near MM cells in both model types approached one. Thus, the concentration near MM cells did not play a significant role for the movement of NP cells.

MM cells did not move in the first place by chemotaxis but rather due the adhesion differences in the models.

Consequently, the behaviour of cells in the dissociation-reaggregation kidney organoid culture experiments was more challenging to simulate. This was also because there were no markers available for differentiating native MM from committed NP cells. As in the explant culture study, persistently faster cell movement was observed in the corner region and slower movement in the tip region. Correspondingly, my model simulations suggested UB as the predominant source of the chemoattractant in the explant culture experiments. On the contrary, the regional cell speeds and quantities in the organoid studies were best captured by another computational model (6). The respective simulations imitated the chemoattractant release from NP cells as well as the differences in the cell-cell adhesion energies. This difference might have been caused by the altered secretory functions of UB epithelia following the dissociation and reaggregation of the cells. Other reason for this difference could have been the shorter experimental time period utilized in the kidney organoid model. These organoids were primarily designed to study early MM cell movement patterns. Nonetheless, model 6 indicated that MM cell movements may be primarily driven by auto/paracrine chemotaxis of MM cells (Dahl et al., 2002; Dudley et al., 1999; Gilbert, 2000; Oxburgh et al., 2011; Wakabayashi et al., 1994; Wallner et al., 1998).

While my work demonstrates the suitability of a relatively simple computational model to reproduce the main cellular events in early nephrogenesis, several limitations should be emphasized. First, I did not allow for continuous recruitment of NP cells from MM cell pool, but assumed fixed cell quantities during the time window of the analysis. Moreover, the lack of *in vitro* models deficient in the individual components of the biological system, which prevented an external validation of the performance of my model in simulating impairments of nephrogenesis under abnormal conditions. Finally, the current model system did not allow to explore the roles of individual molecular signalling pathways or more than a single chemoattractant gradient. Such models would have required far more granular spatiotemporal biochemical information than was currently available. Given these limitations, it was even more remarkable that NP cell migration and PTA formation can be accurately modelled based on two biophysical mechanisms, namely chemotaxis and cell-cell adhesion difference.

5. SUMMARY

During early kidney organogenesis, nephron progenitor cells move from the tip to the branch of the ureteral bud to form the so-called pretubular aggregate, the precursor structure of the later nephron. It is assumed that cell pattern formation during this critical phase of organogenesis is primarily controlled by chemotactic mechanisms and differential cell-cell adhesion. The spatial-temporal organization of this process is not yet fully understood. In recent studies, a nonlinear swarm-like pattern of cell movement has been observed.

In order to gain a better understanding of these processes, I elaborated a three-dimensional mathematical Cellular Potts model, and carried out, validated and applied corresponding model simulations. The model parameters were estimated from experimental data obtained in *ex vivo* kidney explant and dissociation-reaggregation organoid culture studies.

The simulations showed that an optimal enrichment and aggregation of nephron progenitor cells in the corner niche of the ureteral bud branch depends on three factors: the secretion of chemoattractant molecules by a) the epithelial cells of the ureteral bud and b) the nephron progenitor cells themselves, and c) by different adhesion energies between the different cell types. Furthermore, it was observed both experimentally and in the model simulation that nephron progenitor cells move at a higher speed in the corner region of the ureteral bud branches than in their region of origin at the tip of the bud from which they originate. The existence of different cell velocity domains along the ureteral bud was also evaluated with the self-organizing map technique.

In summary, I was able to confirm in the present work the suitability of the Cellular Potts model approach for simulating cell movements and pattern formation during early nephrogenesis. A further refinement of the model should allow the effects of developmental changes the cell phenotypes and the molecular interactions during organ development.

6. ZUSAMMENFASSUNG

Während der frühen Nierenorganogenese bewegen sich Nephron-Vorläuferzellen (NP-Zellen) von der Spitze zum Abgangsbereich des Ureterknospens-Zweigs, um dort das sog. prätubuläre Aggregat (PTA) zu bilden, die Vorläuferstruktur des späteren Nephrons. Es wird angenommen, dass die Zellmusterbildung während dieser kritischen Phase der Organogenese vor allem durch chemotaktische Mechanismen und differentielle Zell-Zell-Adhäsion gesteuert wird.

Die räumlich-zeitliche Organisation dieses Prozesses ist bisher nicht vollständig verstanden. In kürzlichen Studien wurde ein nichtlineares, schwarmartiges Zellbewegungsmuster beobachtet.

Um ein besseres Verständnis dieser Vorgänge zu erreichen habe ich ein dreidimensionales Cellular Potts Modell erstellt und entsprechende Modell-Simulationen durchgeführt, validiert und angewendet. Die Modellparameter wurden anhand experimenteller Daten geschätzt, die in *ex vivo* Nierenexplantations- und Dissoziations-Reaggregations-Organoid-Kulturstudien erhalten wurden.

Die Simulationen zeigten, dass eine optimale Anreicherung und Aggregation von NP Zellen in der Ecknische des Ureterknospenszweigs von drei Faktoren abhängig ist: der Sekretion von chemoattraktanten Molekülen durch a) die Epithelzellen der Ureterknospe und b) die NP-Zellen selbst, und c) durch unterschiedliche Adhäsionsenergien zwischen den verschiedenen Zelltypen. Weiterhin wurde sowohl experimentell als auch in der Modellsimulation beobachtet, dass sich NP-Zellen mit höherer Geschwindigkeit in der Eckregion der Ureterknospenszweige bewegen als in ihrer Ursprungsregion an der Knospenspitze, aus der sie stammen. Die Existenz verschiedener Zellgeschwindigkeits-Domänen entlang der Ureterknospe wurde mit Hilfe der Self-Organizing Map (SOM)-Technik bestätigt.

Zusammenfassend konnte ich in der vorliegenden Arbeit die Eignung des Cellular Potts-Modell-Ansatzes zur Simulation von Zellbewegungen und -musterbildungen während der frühen Nephrogenese aufzeigen. Eine weitere Verfeinerung des Modells sollte es ermöglichen, auch die Auswirkungen von entwicklungsbedingten Veränderungen der Zellphänotypen und der molekularen Interaktionen während der Organentwicklung abzubilden.

7. REFERENCES

- Adivarahan, S., Menshykau, D., Michos, O. and Iber, D. (2013). **Dynamic image-based modelling of kidney branching morphogenesis**. Paper presented at: Computational Methods in Systems Biology (Berlin, Heidelberg: Springer).
- Affolter, M. and Weijer, C. J. (2005). **Signaling to cytoskeletal dynamics during chemotaxis**. *Dev, Cell* 9(1), 19-34, doi: 10.1016/j.devcel.2005.06.003.
- Albert, P. J. and Schwarz, U. S. (2016). **Dynamics of cell ensembles on adhesive micropatterns: bridging the gap between single cell spreading and collective cell migration**. *PLoS Comput Biol* 12(4), e1004863, doi: 10.1371/journal.pcbi.1004863.
- Allena, R., Scianna, M. and Preziosi, L. (2016). **A cellular potts model of single cell migration in presence of durotaxis**. *Math Biosci* 275, 57-70, doi: 10.1016/j.mbs.2016.02.011.
- Andasari, V., Roper, R. T., Swat, M. H. and Chaplain, M. A. (2012). **Integrating intracellular dynamics using compucell3d and Bionetsolver: applications to multiscale modelling of cancer cell growth and invasion**. *PLoS One* 7(3), e33726, doi: 10.1371/journal.pone.0033726.
- Andrew, D. J. and Ewald, A. J. (2010). **Morphogenesis of epithelial tubes: insights into tube formation, elongation, and elaboration**. *Dev Biol* 341(1), 34-55, doi: 10.1016/j.ydbio.2009.09.024.
- Anum, R., Imran, M., Hahsim, R., Mahmood, A. and Majeed, S. (2016). **A hybrid particle swarm optimization (PSO) with chi-square and stable mutation jump strategy**. *Int J Adv Appl Sci* 3(12), 5.
- Atsuta, Y. and Takahashi, Y. (2015). **FGF8 coordinates tissue elongation and cell epithelialization during early kidney tubulogenesis**. *Development (Cambridge, England)* 142(13), 2329-2337, doi: 10.1242/dev.122408.
- Barasch, J., Qiao, J., McWilliams, G., Chen, D., Oliver, J. A. and Herzlinger, D. (1997). **Ureteric bud cells secrete multiple factors, including bFGF, which rescue renal progenitors from apoptosis**. *Am J Physiol* 273(5), F757-767, doi: 10.1152/ajprenal.1997.273.5.F757.
- Belmonte, J. M., Clendenon, S. G., Oliveira, G. M., Swat, M. H., Greene, E. V., Jeyaraman, S., Glazier, J. A. and Bacallao, R. L. (2016). **Virtual-tissue computer simulations define the roles of cell adhesion and proliferation in the onset of kidney cystic disease**. *Mol Biol Cell* 27(22), 3673-3685, doi: 10.1091/mbc.E16-01-0059.

-
- Berg, V. (2011). **Mathematical models of biological systems**, 1. edn, Oxford University Press, U.S.A., p. 256.
- BioPortal (2019). **Biomedical ontologies**. National Center for Biomedical Ontology, URL: <https://bioportal.bioontology.org/> [as of 18.4.2019].
- Blake, J. and Rosenblum, N. D. (2014). **Renal branching morphogenesis: morphogenetic and signaling mechanisms**. *Semin Cell Dev Biol* 36, 2-12, doi: 10.1016/j.semcdb.2014.07.011.
- Bohnenpoll, T. and Kispert, A. (2014). **Ureter growth and differentiation**. *Semin Cell Dev Biol* 36, 21-30, doi: 10.1016/j.semcdb.2014.07.014.
- Bonyadi, M. R. and Michalewicz, Z. (2017). **Particle swarm optimization for single objective continuous space problems: a review**. *Evol Comput* 25(1), 1-54, doi: 10.1162/evco_r_00180.
- Brown, A. C., Adams, D., de Caestecker, M., Yang, X., and Friesel, R. (2011). **FGF/EGF signaling regulates the renewal of early nephron progenitors during embryonic development**. *Development* 138(23), 5099-5112, doi: 10.1242/dev.065995.
- Brown, A. C., Muthukrishnan, S. D., Guay, J. A., Adams, D. C., Schafer, D. A., Fetting, J. L. and Oxburgh, L. (2013). **Role for compartmentalization in nephron progenitor differentiation**. *Proc Natl Acad Sci U S A* 110(12), 4640-4645, doi: 10.1073/pnas.1213971110.
- Carroll, T. J., Park, J. S., Hayashi, S., Majumdar, A. and McMahon, A. P. (2005). **Wnt9b plays a central role in the regulation of mesenchymal to epithelial transitions underlying organogenesis of the mammalian urogenital system**. *Dev Cell* 9(2), 283-292, doi: 10.1016/j.devcel.2005.05.016.
- Cartwright, N. (1983). **How the laws of physics lie**, Vol 34, Oxford University Press, p. 513.
- Chen, S., Brunskill, E. W., Potter, S. S., Dexheimer, P. J., Salomonis, N., Aronow, B. J., Hong, C. I., Zhang, T. and Kopan, R. (2015). **Intrinsic age-dependent changes and cell-cell contacts regulate nephron progenitor lifespan**. *Dev Cell* 35(1), 49-62, doi: 10.1016/j.devcel.2015.09.009.
- Chi, P., Kwon, Y., Moses, D. N., Seong, C., Sehorn, M. G., Singh, A. K., Tsubouchi, H., Greene, E. C., Klein, H. L. and Sung, P. (2009). **Functional interactions of meiotic recombination factors Rdh54 and Dmc1**. *DNA Repair* 8(2), 279-284, doi: 10.1016/j.dnarep.2008.10.012.

- Clendenon, S. G., Ward, H. H., Dunn, K. W. and Bacallao, R. (2013). **High resolution 4-dimension imaging of metanephric embryonic kidney morphogenesis.** *Kidney Int* 83(4), 757-761, doi: 10.1038/ki.2012.464.
- Clevers, H. (2016). **Modeling development and disease with organoids.** *Cell* 165(7), 1586-1597, doi: 10.1016/j.cell.2016.05.082.
- Combes, A. N., Lefevre, J. G., Wilson, S., Hamilton, N. A. and Little, M. H. (2016). **Cap mesenchyme cell swarming during kidney development is influenced by attraction, repulsion, and adhesion to the ureteric tip.** *Dev Biol* 418(2), 297-306, doi: 10.1016/j.ydbio.2016.06.028.
- Combes, A. N., Short, K. M., Lefevre, J., Hamilton, N. A., Little, M. H. and Smyth, I. M. (2014). **An integrated pipeline for the multidimensional analysis of branching morphogenesis.** *Nat Protoc* 9(12), 2859-2879, doi: 10.1038/nprot.2014.193.
- Contributors, T. S. W. (2017). **Spyder program.** Spyder Doc Contributors MIT License, URL: <https://www.spyder-ide.org/> [as of 15.11.2017].
- Costantini, F. and Kopan, R. (2010). **Patterning a complex organ: branching morphogenesis and nephron segmentation in kidney development.** *Dev Cell* 18(5), 698-712, doi: 10.1016/j.devcel.2010.04.008.
- Dahl, U., Sjödin, A., Larue, L., Radice, G. L., Cajander, S., Takeichi, M., Kemler, R. and Semb, H. (2002). **Genetic dissection of cadherin function during nephrogenesis.** *Mol Cell Biol* 22(5), 1474-1487, doi: 10.1128/mcb.22.5.1474-1487.2002.
- Desgrange, A. and Cereghini, S. (2015). **Nephron patterning: lessons from xenopus, zebrafish, and mouse studies.** *Cells* 4(3), 483-499, doi: 10.3390/cells4030483.
- Desroches, M., Guckenheimer, J., Krauskopf, B., Kuehn, C., Osinga, H. M. and Wechselberger, M. (2012). **Mixed-mode oscillations with multiple time scales.** *SIAM Review* 54(2), 211-288, doi: 10.1137/100791233.
- DiMilla, P. A., Barbee K Fau - Lauffenburger, D. A. and Lauffenburger, D. A. (1991). **Mathematical model for the effects of adhesion and mechanics on cell migration speed.** *Biophys. J.* 60(4), 15-37, doi: 10.1016/S0006-3495(91)82027-6.
- Dudley, A. T., Godin, R. E. and Robertson, E. J. (1999). **Interaction between FGF and BMP signaling pathways regulates development of metanephric mesenchyme.** *Genes Dev* 13(12), 1601-1613, doi: 10.1101/gad.13.12.1601.
- DZone (2019). **Self-organizing maps.** URL: <https://dzone.com/articles/self-organizing-maps> [as of 31.5.2019].

- Fish, D. t. (2019). **How to convert strings to floats in pandas dataframe**. URL: <https://datatofish.com/convert-string-to-float-dataframe/> [as of 30.1.2019].
- Gilbert, S. (2000). **Developmental biology: paracrine factors** 6th edition. edn, Sunderland (MA): Sinauer Associates.
- Glazier, J. A. and Graner, F. (1993). **Simulation of the differential adhesion driven rearrangement of biological cells**. *Physical Review E* 47(3), 2128-2154, doi: 10.1103/PhysRevE.47.2128.
- Grieshammer, U., Cebrian, C., Ilagan, R., Meyers, E., Herzlinger, D. and Martin, G. R. (2005). **FGF8 is required for cell survival at distinct stages of nephrogenesis and for regulation of gene expression in nascent nephrons**. *Development* 132(17), 3847-3857, doi: 10.1242/dev.01944.
- Hirashima, T., Rens, E. G. and Merks, R. M. H. (2017). **Cellular potts modeling of complex multicellular behaviors in tissue morphogenesis**. *Dev Growth Differ* 59(5), 329-339, doi: 10.1111/dgd.12358.
- Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novère, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J. and Forum, a. t. r. o. t. S. (2003). **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models**. *Bioinformatics* 19(4), 524-531, doi: 10.1093/bioinformatics/btg015.
- Hunter, J., Dale, D., Firing, E., Droettboom, M. and team, M. d. (2019). **Algorithm of matplotlib.pyplot.xticks**. Matplotlib, URL: https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.xticks.html [as of 2.2.2019].
- Junttila, S., Saarela, U., Halt, K., Manninen, A., Parssinen, H., Lecca, M. R., Brandli, A. W., Sims-Lucas, S., Skovorodkin, I. and Vainio, S. J. (2015). **Functional genetic targeting of embryonic kidney progenitor cells ex vivo**. *J Am Soc Nephrol* 26(5), 1126-1137, doi: 10.1681/ASN.2013060584.
- Karner, C. M., Das, A., Ma, Z., Self, M., Chen, C., Lum, L., Oliver, G. and Carroll, T. J. (2011). **Canonical Wnt9b signaling balances progenitor cell expansion and differentiation during kidney development**. *Development* 138(7), 1247-1257, doi: 10.1242/dev.057646.

- Kohonen, T. (1982). **Self-organized formation of topologically correct feature maps.** *Biological Cybernetics* 43(1), 59-68, doi: doi.org/10.1007/BF00337288.
- Kohonen, T., Honkela, T (2007). **Kohonen network.** Scholarpedia, 2(1):1568., URL: http://www.scholarpedia.org/article/Kohonen_network [as of 19.3.2019].
- Kopan, R., Chen, S., and Little, M (2014). **Nephron progenitor cells: shifting the balance of self-renewal and differentiation.** *Curr. Top. Dev. Biol.* 107, 293-300, doi: 10.1016/B978-0-12-416022-4.00011-1.
- Krause, M., Rak-Raszewska, A., Pietila, I., Quaggin, S. E. and Vainio, S. (2015). **Signaling during kidney development.** *Cells* 4(2), 112-132, doi: 10.3390/cells4020112.
- Lawlor, K. T., Zappia, L., Lefevre, J., Park, J.-S., Hamilton, N. A., Oshlack, A., Little, M. H. and Combes, A. N. (2019). **Nephron progenitor commitment is a stochastic process influenced by cell migration.** *eLife* 8, e41156, doi: 10.7554/eLife.41156.
- Lawson, B. A. J. and Flegg, M. B. (2016). **A mathematical model for the induction of the mammalian ureteric bud.** *J Theor Biol* 394, 43-56, doi: 10.1016/j.jtbi.2015.12.025.
- Lefevre, J. G., Chiu, H. S., Combes, A. N., Vanslambrouck, J. M., Ju, A., Hamilton, N. A. and Little, M. H. (2017). **Self-organisation after embryonic kidney dissociation is driven via selective adhesion of ureteric epithelial cells.** *Development* 144(6), 1087-1096, doi: 10.1242/dev.140228.
- Lefevre, J. G., Combes, A. N., Little, M. H. and Hamilton, N. A. (2016). **Analysed cap mesenchyme track data from live imaging of mouse kidney development.** *Data Brief* 9, 149-154, doi: 10.1016/j.dib.2016.08.053.
- Lindström, N. O., De Sena Brandine, G., Tran, T., Ransick, A., Suh, G., Guo, J., Kim, A. D., Parvez, R. K., Ruffins, S. W., Rutledge, E. A., Thornton, M. E., Grubbs, B., McMahon, J. A., Smith, A. D. and McMahon, A. P. (2018). **Progressive recruitment of mesenchymal progenitors reveals a time-dependent process of cell fate acquisition in mouse and human nephrogenesis.** *Dev Cell* 45(5), 651-660.e654, doi: <https://doi.org/10.1016/j.devcel.2018.05.010>.
- Lindstrom, N. O., Lawrence, M. L., Burn, S. F., Johansson, J. A., Bakker, E. R., Ridgway, R. A., Chang, C. H., Karolak, M. J., Oxburgh, L., Headon, D. J., Sansom, O. J., Smits, R., Davies, J. A. and Hohenstein, P. (2015). **Integrated beta-catenin, BMP, PTEN, and Notch signalling patterns the nephron.** *Elife* 3, e04000, doi: 10.7554/eLife.04000.
- Little, M. H. (2015). **Kidney development, disease, repair and regeneration**, 1. edn, Academic Press, United Kingdom.

-
- Little, M. H., McMahon, A. P. (2012). **Mammalian kidney development: principles, progress, and projections**. Cold Spring Harb Perspect Biol 4(5), doi: 10.1101/cshperspect.a008300.
- Magno, R., Grieneisen, V. A. and Maree, A. F. (2015). **The biophysical nature of cells: potential cell behaviours revealed by analytical and computational studies of cell surface mechanics**. BMC Biophys 8, 8, doi: 10.1186/s13628-015-0022-x.
- Marée, A. F. M., Grieneisen, V. A. and Hogeweg, P. (2007). **The cellular potts model and biophysical properties of cells, tissues and morphogenesis**. In: Single-Cell-Based Models in Biology and Medicine, eds. Anderson, A. R. A., Chaplain, M. A. J. and Rejniak, K. A., Birkhäuser Basel, Basel, pp. 107-136.
- McMahon, A. P. (2016). **Development of the mammalian kidney**. Curr Top Dev Biol 117, 31-64, doi: 10.1016/bs.ctdb.2015.10.010.
- Midha, R. (2019). **Data visualization with matplotlib**. Github, URL: <https://rohitmidha23.github.io/Matplotlib-Explained/> [as of 1.2.2019].
- Mogilner, A. and Rubinstein, B. (2005). **The physics of filopodial protrusion**. Biophys J 89(2), 782-795, doi: 10.1529/biophysj.104.056515.
- Moise, N. and Friedman, A. (2019). **Rheumatoid arthritis - a mathematical model**. J Theor Biol 461, 17-33, doi: 10.1016/j.jtbi.2018.10.039.
- Mukhopadhyay, S. and Hazra, L. (2015). **Pareto optimality between width of central lobe and peak sidelobe intensity in the far-field pattern of lossless phase-only filters for enhancement of transverse resolution**. Appl Opt 54(31), 9205-9212, doi: 10.1364/AO.54.009205.
- Muzumdar, M. D., Tasic, B., Miyamichi, K., Li, L. and Luo, L. (2007). **A global double-fluorescent Cre reporter mouse**. Genesis 45(9), 593-605, doi: 10.1002/dvg.20335.
- NG, A. and Soo, K. (2017). **Self-organizing maps tutorial** Algobeans: layman tutorials in analytics, URL: <https://annalyzin.wordpress.com/2017/11/02/self-organizing-map/> [as of 10.5.2019].
- Obara-Ishihara T, K. J., Niswander L, and Herzlinger D (1999). **The surface ectoderm is essential for nephric duct formation in intermediate mesoderm**. Development 126(6), 1103-1108.

- Osborne, J. M., Fletcher, A. G., Pitt-Francis, J. M., Maini, P. K. and Gavaghan, D. J. (2017). **Comparing individual-based approaches to modelling the self-organization of multicellular tissues.** *PLOS Computational Biology* *13*(2), e1005387, doi: 10.1371/journal.pcbi.1005387.
- Oxburgh, L., Brown, A. C., Fetting, J. and Hill, B. (2011). **BMP signaling in the nephron progenitor niche.** *Pediatr Nephrol* *26*(9), 1491-1497, doi: 10.1007/s00467-011-1819-8.
- Perantoni, A. O., Timofeeva, O., Naillat, F., Richman, C., Pajni-Underwood, S., Wilson, C., Vainio, S., Dove, L. F. and Lewandoski, M. (2005). **Inactivation of FGF8 in early mesoderm reveals an essential role in kidney development.** *Development* *132*(3859-3871), doi: 10.1242/dev.01945.
- Pollard, T. D. and Borisy, G. G. (2003). **Cellular motility driven by assembly and disassembly of actin filaments.** *Cell* *112*(4), 453-465, doi: 10.1016/s0092-8674(03)00120-x.
- Postma, M., Roelofs, J., Goedhart, J., Gadella, T. W. J., Visser, A. J. W. G. and Van Haastert, P. J. M. (2003). **Uniform cAMP stimulation of dictyostelium cells induces localized patches of signal transduction and pseudopodia.** *Mol Biol Cell* *14*(12), 5019-5027, doi: 10.1091/mbc.e03-08-0566.
- Rafelski, S. M. and Theriot, J. A. (2004). **Crawling toward a unified model of cell mobility: spatial and temporal regulation of actin dynamics.** *Annu Rev, Biochem* *73*(1), 209-239, doi: 10.1146/annurev.biochem.73.011303.073844.
- Rappel, W.-J. and Loomis, W. F. (2009). **Eukaryotic chemotaxis.** *Wiley Interdiscip Rev Syst Biol Med* *1*(1), 141-149, doi: 10.1002/wsbm.28.
- Ricono, J. M., Wagner, B., Gorin, Y., Arar, M., Kazlauskas, A. (2008). **PDGF receptors modulates metanephric mesenchyme chemotaxis induced by PDGF-AA.** *Am J Physiol Renal Physiol.* *296*(2), F406–F417, doi: 10.1152/ajprenal.90368.2008.
- Saarela, U., Akram, S. U., Desgrange, A., Rak-Raszewska, A., Shan, J., Cereghini, S., Ronkainen, V. P., Heikkila, J., Skovorodkin, I. and Vainio, S. J. (2017). **Novel fixed z-direction (FiZD) kidney primordia and an organoid culture system for time-lapse confocal imaging.** *Development* *144*(6), 1113-1117, doi: 10.1242/dev.142950.
- Santarcangelo, J. (2019). **From python nested lists to multidimensional numpy arrays.** URL: <https://cognitiveclass.ai/blog/nested-lists-multidimensional-numpy-arrays> [as of 1.3.2019].

-
- Shahzad, F., Masood, S. and Khan, N. K. (2014). **Probabilistic opposition-based particle swarm optimization with velocity clamping**. Knowledge and Information Systems 39(3), 703-737, doi: 10.1007/s10115-013-0624-z.
- Shan, J., Jokela, T., Skovorodkin, I. and Vainio, S. (2010). **Mapping of the fate of cell lineages generated from cells that express the Wnt4 gene by time-lapse during kidney development**. Differentiation 79(1), 57-64, doi: 10.1016/j.diff.2009.08.006.
- Short, K. M., Combes, A. N., Lefevre, J., Ju, A. L., Georgas, K. M., Lamberton, T., Cairncross, O., Rumballe, B. A., McMahon, A. P., Hamilton, N. A., Smyth, I. M. and Little, M. H. (2014). **Global quantification of tissue dynamics in the developing mouse kidney**. Dev Cell 29(2), 188-202, doi: 10.1016/j.devcel.2014.02.017.
- SignalProcessing (2019). **Given position measurements, how to estimate velocity and acceleration**. Stackexchange, URL: <https://dsp.stackexchange.com/questions/9498/given-position-measurements-how-to-estimate-velocity-and-acceleration> [as of 11.11. 2018].
- Sluka, J. P., Shaikh, A. and Glazier, J. A. (2018). **Parameter fitting in a multiscale model: parameter scanning vs. particle swarm optimization**. IMAG/MSM Meeting 1(1), 1.
- Soll, D. R., Wessels D, Heid, P. and Zhang, H. (2002). **A contextual framework for characterizing motility and chemotaxis mutants in dictyostelium discoideum**. J Muscle Res Cell Motil 23(7-8), 659-672, doi: 10.1023/a:1024459124427.
- StackOverflow (2019a). **Accessing the index in 'for' loops?** StackOverflow, URL: <https://stackoverflow.com/questions/522563/accessing-the-index-in-for-loops> [as of 15.2.2019].
- StackOverflow (2019b). **Compute a confidence interval from sample data**. StackOverflow, URL: <https://stackoverflow.com/questions/15033511/compute-a-confidence-interval-from-sample-data> [as of 10.1.2019].
- Stegmaier, J., Amat, F., Lemon, W. C., McDole, K., Wan, Y., Teodoro, G., Mikut, R. and Keller, P. J. (2016). **Real-time three-dimensional cell segmentation in large-scale microscopy data of developing embryos**. Dev Cell 36(2), 225-240, doi: 10.1016/j.devcel.2015.12.028.
- Stuart, R. O., Bush, K. T. and Nigam, S. K. (2003). **Changes in gene expression patterns in the ureteric bud and metanephric mesenchyme in models of kidney development**. Kidney Int 64(6), 1997-2008, doi: 10.1046/j.1523-1755.2003.00383.x.

-
- Swat, M. H., Belmonte, J. M. and Glazier, J. A. (2017). **Chemotaxis**. URL: https://compucell3dreferencemanual.readthedocs.io/en/latest/chemotaxis_plugin.html#chemotaxis [as of 15.2.2019].
- Swat, M. H., Hester, S. D., Balter, A. I., Heiland, R. W., Zaitlen, B. L. and Glazier, J. A. (2009). **Multicell simulations of development and disease using the compucell3d simulation environment**. *Methods Mol Biol* 500, 361-428, doi: 10.1007/978-1-59745-525-1_13.
- Swat, M. H., Thomas, G. L., Belmonte, J. M., Shirinifard, A., Hmeljak, D. and Glazier, J. A. (2012). **Multi-scale modeling of tissues using compucell3d**. *Methods Cell Biol* 110, 325-366, doi: 10.1016/B978-0-12-388403-9.00013-8.
- Takasato, M., Er, P. X., Chiu, H. S., Maier, B., Baillie, G. J., Ferguson, C., Parton, R. G., Wolvetang, E. J., Roost, M. S., Chuva de Sousa Lopes, S. M. and Little, M. H. (2015). **Kidney organoids from human iPS cells contain multiple lineages and model human nephrogenesis**. *Nature* 526(7574), 564-568, doi: 10.1038/nature15695.
- Tikka, P. (2019a). **Additional supplementary figures of thesis**. FigShare, URL: <https://doi.org/10.6084/m9.figshare.10315910.v1> [as of 16.11.2019].
- Tikka, P. (2019b). **CC3D model example codes**. GitHub, URL: <https://github.com/paulitikka/CC3D-Model-Example-Codes> [as of 16.11.2019].
- Tikka, P. (2019c). **Data of non-optimized and optimized CC3D models and kidney organoid experiments**. FigShare, URL: https://figshare.com/projects/Research_Data/71801 [as of 16.11.2019].
- Tikka, P. (2019d). **Model data analysis**. GitHub, URL: <https://github.com/paulitikka/Model-Data-Analysis> [as of 16.11.2019].
- Tikka, P. (2019e). **Video of model 7**. URL: https://figshare.com/articles/Optimized_Model_7_s_video/10315904 [as of 25.11.2019].
- Tikka, P., Mercker, M., Skovorodkin, I., Saarela, U., Sluka, J., Glazier, J. A., Marciniak-Czochra, A. and Schaefer, F. (2019). **Computational modelling of nephron progenitor cell movement and aggregation during kidney organogenesis**. Publication in Preparation.
- Tikka, P. and Schaefer, F. (2017). **Computational modelling of early nephrogenesis**. *Nephron* 137(2), 155-167, Eurenomics joint annual meeting abstract, doi: 10.1159/000477523.

- Tikka, P. and Schaefer, F. (2019). **Methods of computational analysis in kidney development**. *Methods Mol Biol* 1926, 235-246, doi: 10.1007/978-1-4939-9021-4_19.
- Tikka, P. and Skovorodkin, I. (2019). **The video of kidney organoid experiments**. URL: https://figshare.com/articles/Kidney_Organoid_Experiments_video/10315895 [as of 25.11.2019].
- Vettigli, G. (2018). **Minisom self organizing maps**. GitHub, URL: <https://github.com/JustGlowing/minisom> [as of 15.1.2019].
- Wakabayashi, H., Cavanaugh Pg Fau - Nicolson, G. L. and Nicolson, G. L. (1994). **Responses to paracrine chemotactic and autocrine chemokinetic factors and lung metastatic capability of mouse RAW117 large-cell lymphoma cells**. *70(6)*, 1089-1094, doi: 10.1038/bjc.1994.453.
- Wallner, E., Yang, Q., Peterson, D., Wada, J. and Kanwar, Y. (1998). **Relevance of extracellular matrix, its receptors, and cell adhesion molecules in mammalian nephrogenesis**. *Am J Physiol* 275(4), F467-477, doi: 10.1152/ajprenal.1998.275.4.F467.
- Waskom, M. (2019). **Visualizing the distribution of a dataset**. Seaborn, URL: <https://seaborn.pydata.org/tutorial/distributions.html> [as of 14.1.2019].
- Wessels, D., Voss, E., Von Bergen, N., Burns, R., Stites, J. and Soll, D. R. (1998). **A computer-assisted system for reconstructing and interpreting the dynamic three-dimensional relationships of the outer surface, nucleus and pseudopods of crawling cells**. *Cell Motil, Cytoskeleton* 41(3), 225-246, doi: 10.1002/(SICI)1097-0169(1998)41:3<225::AID-CM4>3.0.CO;2-I.
- Williams, H. P. and Harwood, A. J. (2003). **Cell polarity and dictyostelium development**. *Curr Opin, Microbiol* 6(6), 621-627, doi: 10.1016/j.mib.2003.10.008.
- Wittek, P., Gao, S. C., Lim, I. S. and Zhao, L. (2017). **Somoclu: an efficient parallel library for self-organizing maps**. *J Stat Softw* 78(9), 1-21.
- Zhang, Y.-D., Balochian, S., Agarwal, P., Bhatnagar, V. and Houshia, O. (2014). **Artificial intelligence and its applications**, Vol 2014, p. 10.
- Zhang, Y., Wang, S. and Ji, G. (2015). **A comprehensive survey on particle swarm optimization algorithm and its applications**. *Math Probl Eng* 2015(1), 38, doi: 10.1155/2015/931256.
- Zubkov, V. S., Combes, A. N., Short, K. M., Lefevre, J., Hamilton, N. A., Smyth, I. M., Little, M. H. and Byrne, H. M. (2015). **A spatially-averaged mathematical model of kidney branching morphogenesis**. *J Theor Biol* 379, 24-37, doi: 10.1016/j.jtbi.2015.04.015.

8. SUPPLEMENT

8.1. Description of Previous Model

In short, Combes et al. (2016) solved a steady state convection-diffusion model for the observed distribution of tip distances $y(d)$ (Combes et al., 2016; Lefevre et al., 2016). The purpose was to test whether the pattern of attraction and repulsion was sufficient to maintain the cap cells in proximity to ureteric tip. Combes et al. (2016) presumed that the movement of NP cells towards or way from the tip surface was a combination of an undirected diffusion D , independent of tip distance, convection, and the instantaneous velocity $v(d)$ (Combes et al., 2016). This velocity was a function of the tip distance (d), where the positive values corresponded to repulsion and the negative ones to attraction to the tip. It was assumed that the distribution of cap cells with distance and time, $y(d,t)$ in (Lefevre et al., 2016), satisfied

$$D \frac{\partial^2 y}{\partial x^2} = v \frac{\partial y}{\partial x}. \quad \text{Eq. (12)}$$

Although individual cells moved significantly, they also assumed that the overall distribution of tip distances $y(d)$ was approximately stable and considered the well-known steady state solution

$$y(d) = y(0) e^{\int_0^d k(u) du}, \quad \text{Eq. (13)}$$

where

$$k(d) = \frac{v(d)}{D}, \quad \text{Eq. (14)}$$

and in principle, the function $k(d)$ was described as the velocity, especially for D being 1. Otherwise, $k(d)$ was presumed to be constant k_1 if $d < d_1$, or k_2 if $d > d_2$, and vary linearly between d_1 and d_2 . This four-parameter model was fitted to the observed tip distances using a maximum likelihood method. This statistical method estimated its parameters towards the observed values in a certain way. Particularly, it attempted to find the parameter values that maximized a likelihood function, given the observations. The resulting estimate was called the maximum likelihood estimate. As a result, the fitted model gave $d_1 = 0.005 \pm 0.381$, $d_2 = 14.95 \pm 0.18$, $k_1 = 0.293 \pm 0.010$, $k_2 = -0.098 \pm 0.001$. Since d_1 was not significantly different from 0 they simplified the model:

$$k(d) = \begin{cases} 0.293 - 0.026d, & 0 \leq d \leq 14.95 \\ -0.098, & d > 14.95 \end{cases}, \quad \text{Eqs (15a,b)}$$

It appeared, that after around 15 μm radius from the tip, NP cell movements started to slow down and eventually switch direction and move backwards towards tip (Eq. 15b). And after a cell had returned to the tip effecting domain of less than around 15 μm , the cell would start to accelerate towards tip until the cell would have attained the maximum velocity of 0.293 $\mu\text{m}/\text{min}$ precisely at the tip location (Eq. 15a).

First and foremost, it was questionable to make a velocity dependent only regards to the distance of one spatial domain, that is to say the tip. One can notice the problem of this model from the first fitted distance, d_I , which was described to be close to zero, even though the variance of it was not. That was to say that, the biggest value of it, for instance ‘precise’ tip location, could have been 0.386 μm away from the tip in respect to any random direction. The model should have accounted other domains around static UB, such as the corner below tip, the domain above T or L-shaped tip, and all the cells around UB swarming this or that direction, not necessarily bluntly away or towards tip. These different places most likely would have had a differential expression of the chemoattractant and the adhesion molecules.

8.2. Example Ranges of Parameters and Simulation Lengths During Optimization

An example set of optimized parameter values for the advanced model (7) were: $D=1.83 \times (10^{-12})\text{m}^2\text{s}^{-1}$ (with the respective integer values) between [1, 2] with (the averages during PSO run) $[1.42 \pm 0.39]$, and $\gamma = 4.7 \times 10^{-7} \times 10^{-7}\text{s}$ between $[1 \times 10^{-8}, 1 \times 10^{-6}]$ with $[7.2 \times 10^{-7} \pm 1.7 \times 10^{-7}]$. It also had $J=7.90 \times 10^{-15} \text{ kgs}^{-2}$ between [2, 8] with $[6.49 \pm 2.23]$, $\lambda_{CL} = 14.1 \times 10^{-27} \text{ kgm}^2\text{s}^{-2}$ between [10, 150] with $[15.2 \pm 5.1]$, $S=6.91 \text{ DU/s}$ between [0.3, 30] with $[5.46 \pm 2.05]$, $T=49.8 \text{ DU}$ between [5, 50] with $[48.14 \pm 1.15]$, and finally $\lambda_S=8.1 \times 10^{-3} \text{ kgm}^{-2} \text{ s}^{-2}$ between [0.001, 10] with $[8.30 \pm 0.83]$. With this in mind, the quality value reached a plateau value of 10809.

32 PSO simulations were run in parallel and the total time required for a particular model was consequently 60x4 times the time of a single run. This was because I had an access to a 32-node computer cluster. A complete PSO run for a particular model included 60x4x8x3=5760 simulations. Consequently, the elapsed time for a complete PSO run was approximately 2 hours for the 2D models and 5 days for the 3D models. Notwithstanding, the simulations took different period of time to perform depending on the amount and interdependence of the parameters. For example, with the previous cluster computer, 3D PSO simulations could take several weeks, whereas 2D PSO simulations required about one day. The 2D version of the model (7) took 3.3h for the 60 iterations, whereas the 3D model took 20.7 days using the same PSO setup.

8.3. Codes

8.3.1. Example of Dynamic CompuCell3D Coding File

The dynamic file named as 'steppables' (c.f., 1b at '2.2.3 Model Simulations') consists of 'ConstraintInitializer' and 'NewSimulation' classes. The first class constraints the variables, while the other initiates a new simulation.

```

class ConstraintInitializerSteppable(SteppableBasePy):
    def __init__(self, _simulator, _frequency=1):
        SteppableBasePy.__init__(self, _simulator, _frequency)
    def start(self):
        NPCell_count = 0
        ACell_count = 0
        for cell in self.cellList:
            cell.targetVolume=375
            ...(shortened)
            if cell.type == 1: #cell.type is important variable to insert for recording sim data(!)
                NPCell_count +=1 #similarly for MM cells, here 'A' cells
                ..(shortened)
            # randomly reassign the cell types (NP and A cells) to give a random starting
            # configuration, make sure though that the total number of NP cells stays the same
            controlled by a parameter in the parameters file
            if p.randomizeCells == 1: # 1=yes, 0=no
                print "\n\nRandomizing cell types.\nNumber of NPCell and ACell intially
                =",NPCell_count,ACell_count
                ..(shortened)

class NewSimulationSteppable(SteppableBasePy):
    def __init__(self, _simulator, _frequency=1):
        SteppableBasePy.__init__(self, _simulator, _frequency)
    def start(self):
        # actually, better to put directly to the analysis folder:
        FileName2="distances.csv"
        self.File2,self.fullFileName2=self.openFileInSimulationOutputDirectory(FileName2,"w")
        self.File2.write("cell_no,cell_type,time_mcs,cell_dist_to_corner,cell_dist_to_tip,x_position_
        (px),y_position_(px), z_position_(px),cell_field, field_mean\n") #recording cell types enables
        analysis of simulations with (initial) random cell positions
        # load a file with the starting field for Wint9b controlled by a parameter in the parameters file
        if p.loadWntFieldFile == 1:
            print "\n\nLoading Wnt file."
            ..(shortened)
            csv_reader = csv.reader(fieldDataFile, delimiter='\t')
            for aLine in csv_reader: # aLine is a python list of values on the line of the input file
                x=int(aLine[0]); y=int(aLine[1]); z=int(aLine[2]); conc=float(aLine[3])
                field[x,y,z] = conc
            fieldDataFile.close()
    def step(self,mcs):
        # only do this every 10 MCS
        if mcs % 10 == 0:

```

```

        field=self.getConcentrationField("Wnt9b")
    #    to get the global mean one needs extra coding; for individual mean: meanValue =
np.mean(field[55,55,0])
        averVol = 0.0, numberOfCells = 0, meanValue =[]
        for i in range(110)...(shortened for j and k): meanValue.append(field[i,j,k])
        meanValue_mcs=np.mean(meanValue)
        for cell in self.cellListByType(self.NPCELLS,self.ACELLS):
#distances to the static uretic bud is irrelevant
self.File2.write(str(cell.id)+","+str(cell.type)+","+str(mcs)+","+str(self.distance(_from=[20,
40, 50],_to=[cell.xCOM,..(same to y and z)])))+","+str(self.distance(_from=[40, 50, 50],\
_to=[cell.xCOM,..(short)])))+","+str(cell.xCOM)..(short)..str(field[int(cell.xCOM), ..(short)]))
+",""+str(meanValue_mcs)+"\n")
        if mcs == p.mcs-1:
# do this on the last mcs step (i.e. not beautiful fix for "finish" not working): self.File2.close()
        def finish(self):
            # this closing is needed for getting the total files, in order to 'flush' them from memory:
            self.File2.close()

```

8.3.2. Shortened Python Code of Particle Swarm Optimization Method

PSO code procedures summarized from James Sluka's original codes given personally or partly found at (Bonyadi and Michalewicz, 2017; Sluka et al., 2018).

- 1) Import packages, such as Slurm (system management) and OS (operations) in python.
- 2) Set PSO parameters and options with a function and submit the set of simulations via Slurm.
- 3) Load the parameters with a function that returns a list of parameter names, and min and max values.
- 4) Create the working directories with a function that returns matrix indexed by swarm number, then particle number, with value of the directory's path. The working directories are all created in the same directory as the .cc3d file. There is a folder for each particle in each swarm. The folders are named: S00001_P00001 for particle 1 of swarm 1.
- 5) Create a function for each particle instance, which contains:
 - a) position, b) velocity, c) pBest[], d) pBestE, which mean:
 - a) vector of current x values of particles, b) vector of current x velocities of particles,
 - c) vector of x the values in the best solutions of particles (from .pos[]), and
 - d) value (e.g., energy, criteria) at pBest[].
- 6) The particle swarm optimization algorithm (class) does the particle instance algorithm (function) globally and contains:
 - a) solution, b) swarm, c) gBest, d) gBestE. The function updates the parameters in the *_parameters.py file for the CC3D jobs. The new values are added to the end of the existing file in each of the job directories, since the parameters file is a python script the last assignments is what is used during the simulation. In addition, this method retains a history of all the parameter sets and energy value, see below, for each particle. Also add the best energy for the particular particle so it can be used to terminate CC3D runs when

the residual error exceeds the previous best residual error. Finally, write the results (self.gBest, lastIterWchange, self.gBestE, iter) to file. The final optimization algorithm (in the class) is thus in short:

```
def optimize(self):
    for iter in range(iterations):
        self.updateParams(iter) / self.theResults=submitJobs(iter)
        for iS in range(numSwarms):
            if iter%5 == 0:
                print "iter %4i swrm %2i gBestE %9.3f " % ..(shortened))"
            #Update the personal (particle) best positions:
            for iP in range(swarmSize):
                E = self.theResults[iS][iP]
                if E < self.swarm[iS][iP].pBestE:
                    self.swarm[iS][iP].pBest = list(self.swarm[iS][iP].pos ..shortened)
                    self.swarm[iS][iP]
                    print " new particle gBest",[iS],[iP], 'E=%8.2f (' % E...(shortened))
            #Get the global (within the particular sub-swarm) best particle:
            for iP in range(swarmSize):
                E = self.theResults[iS][iP] #E = self.f(self.swarm[iS][iP].pBest)
                if E < self.gBestE[iS]:
                    self.gBest[iS] = list(self.swarm[iS][iP].pBest)self.gBestE[iS] = E
                    print "new SWARM gBestE,gBest",[iS],[iP], 'E=%8.2f (..shortened)
                    lastIterWchange = iter
            self.runningLog(iter)
            for iS in range(numSwarms):
                #Update position of each particle
                for iP in range(swarmSize):
                    self.swarm[iS][iP].updateVelocities(self.gBest[iS])/positions/constraints    (..shortened)
    return self.gBest, lastIterWchange, self.gBestE, iter
```

8.3.3. Shortened Python Code of 2D and 3D Data Analysis

The following shortened python coding routines are for the calculations of speeds, distances and concentrations of the models (2D/3D). The codes are compiled from various sources (Fish, 2019; Midha, 2019; Santarcangelo, 2019; SignalProcessing, 2019; StackOverflow, 2019a; StackOverflow, 2019b). The codes for cell quantities are separately at the end of this chapter.

```
1) Import the packages, e.g.:
import pandas as pd #for importing files
import glob #for importing files
import numpy as np #for calculations, array manipulations, and fun, i.e. functions
import matplotlib.pyplot as plt #for scientific plots

2) Import the data:
all_files2=(glob.glob(C://Model data//3d.// *distances.csv"))
#distances file contains also concentrations and coordinates
# Create the list for the three DataFrames you want to create:
for filename in all_files2:
```

```
dataframes.append(pd.read_csv(filename))
```

- 3) Create a function that select either NP or MM cells in the tip or corner regions (of U or R):

```
def crit_fun(dataframes,x0,x1,y0,y1,z0,z1):
    dataframes2 = []
    crit1=dataframes['x_position_(px)']>=x0
    ... (shortened)
    criteria_all=crit1 & crit2 & crit3 & crit4 & crit5 & crit6
    dataframes2=dataframes[criteria_all]
    return dataframes2
# Auxiliary function for intersecting values in list needed for the next function (speed3_prel):
def intersection(lst1, lst2):
    lst3 = [value for value in lst1 if value in lst2]
    return lst3

def speed3_prel(dataframes,cond,conda):
    df_dist_adh3d=dataframes
    if conda == 'rnd':
    ... (shortened)
        dfcell_names3d.ix[:,1]=list(np.where(dfcell_names3d.ix[:,1] > 1, 'ACells', 'NPCells'))
        dfcell_names3da=dfcell_names3d.ix[:,0:2]
        elif conda == 'norm':
            dfcell_names3d= pd.read_csv('all cells uniform3d_v2.csv', delimiter=";", header=None)
            dfcell_names3da=dfcell_names3d[[0,1]]
# selecting common indices in the list of lists of indices
    dfoki22=[], setti=[], settit=[], time=[]
    if cond == 'both': dfcell_names3da.columns=['a','b']
    ... (shortened)
    else:
        setti=dfcell_names3da[dfcell_names3da.ix[:,1]==cond].ix[:,0] #adequate quantity,
len(setti)=196
    dfoki22=df_dist_adh3d.set_index("cell_no"), dfoki22=dfoki22.loc[list(setti),:]
    settit=list(crit_fun(dfoki22,0,66,0,81,24,76).index.unique())
    # 'corner':
    setti2=intersection(setti, settit)
    ... (shortened)
    c1=(40 >= dataframesa['y_position_(px)'])
    ... (shortened)
# 'tip':
    c3=(40 <= dfoki22['y_position_(px)'])
    ... (shortened)
    return settit, setti22, setti3 #both, corner, tip (regions)
```

- 4) Symmetrize matrices and remove nan values:

```
def x_y_z_dis_fun(dataframes,setti,time):
    from random import randrange, uniform #just in case here, needed for random indexing
#The X values:
```

```

Append model values from file with your indeces (with 'for loop') to obtain a list of lists
dfJEE=dataframes.set_index("cell_no")
index = 0, mmf=[], t_je=time
for index in setti:
    mmf.append(np.array(dfJEE.ix[index,'x_position_(px)']))
# Removing 'nans'
for i in range(len(mmf)):
    mmf[i][np.where(np.isnan(mmf[i]))]=+0.1
    for j in range(len(mmf[i])):
        if mmf[i][j] == +0.1:
            lenin=int(round(len(mmf[i])/4)) # latest previous values
            mmf[i][j] == abs(mean(mmf[i][-lenin:-2])) # replacing the latest prev. with mean val.
# The symmetrical numpy matrix made from this list of lists by padding constant values to the
# end of those lists that are below the amount of cells
mtt=np.zeros((len(setti),t_je)), mmf=np.array(mmf)
if mtt.shape == mmf.shape:
    mtt=mmf
elif mtt.shape != mmf.shape:
    for i in range(len(mmf)):
        Ashape=mmf[i].shape[0], new_shape=t_je
        shape_diff = np.asarray(new_shape) - np.asarray(Ashape)
        mtt[i,:]=np.lib.pad(mmf[i], (0,shape_diff), 'constant', constant_values=(0.1))
(Similarly for y and z values)
return mtt, mtt2, mtt3 #correspond x, y, and z

```

- 5) Calculate/report 3D a) speeds, b) distances, or c) concentrations from the previous:
- for j in range(index_time-1): $r[:,j] = \text{np.sqrt}((x[:,(j+1)]-x[:,j])**2+(y[:,(j+1)]-y[:,j])**2+(z[:,(j+1)]-z[:,j])**2)$
 - 'cell_dist_to_corner' and 'cell_dist_to_tip' are recorded in the data frames
 - Concentration is calculated with COMs and a new function:

```

def concentration(dataframes, setti, time):
    COM= x_y_z_dis_fun(dataframes, setti, time)
    #Selecting the correct spaces
    index_cells=len(setti), index_time=time
    C=np.zeros((index_cells, index_time))
    for j in range(index_time-1):
        C[:,j] = abs((COM[:,(j+1)]- COM[:,j]))
    ctot=C[:,0:(time)]
return ctot

```

- 6) Calculate confidence intervals:

```

def mean_confidence_interval(data, confidence=0.95):
    a = 1.0 * np.array(data), n = len(a), m, se = np.mean(a), scipy.stats.sem(a)
    h = se * scipy.stats.t.ppf((1 + confidence) / 2., n-1)
return m, m-h, m+h

```

7) Gather all the calculated data as per regions:

```
for i in range(len(dataframes)):
    tot.append(speed3_preli(dataframes[i],cond,conda))
    tot2.append(speed_fun(dataframes[i],tot[i][0],tot[i][1],tot[i][2],time,dt))
for i in range(len(dataframes)):
    frames.append(tot2[i][0]), frames2.append(tot2[i][1]), frames3.append(tot2[i][2])
result = pd.concat(frames), result2 = pd.concat(frames2), result3 = pd.concat(frames3)
for i in range(time):
    yes.append(mean_confidence_interval(result[[i]], confidence=0.95))
    yes2.append(mean_confidence_interval(result2[[i]], confidence=0.95)), #yes3 similarly
```

8) Change the variable types and save the resulting file:

```
yes[1] = yes[1].astype(float) (and similarly for yes2 and yes3)
resulti = pd.concat([yes, yes2, yes3], axis=1, sort=False)
resulti.columns = ['Speed_tip_avg', 'Speed_tip_CI_min', 'Speed_tip_CI_max',
'Speed_corner_avg', 'Speed_corner_CI_min','Speed_corner_CI_max','Speed_overall_avg',
'Speed_overall_CI_min', 'Speed_overall_CI_max',
resulti.to_csv(name,index=False,header='infer')
```

9) The cell quantities of either NP or MM cells in the regions are calculated:

```
def crit_funt(dataframes,y0, cond1):
```

```
    if cond1 == 'tip': (shortened), elif cond1 == 'corner': (shortened)
    return dataframes2
```

```
def amount(dataframes, cond, cond1, conda):
```

```
    dataframes2=crit_funt(dataframes,40,cond1), df_dist_adh3d=dataframes2
    if conda == 'pers': #shortened
        dfcell_names3d= pd.read_csv('all cells uniform3d_v3a.csv', delimiter="\t", header=None)
    elif conda == 'no-pers': #shortened
        dfcell_names3d= pd.read_csv('all cells uniform3d_v2.csv', delimiter=";", header=None)
    elif conda == 'pers2': #shortened
        dfcell_names3d.ix[:,1]=list(np.where(dfcell_names3d.ix[:,1] > 1, 'ACells', 'NPCells'))
    if cond == 'NPCells': #shortened
        setti=dfcell_names3da[dfcell_names3da.ix[:,1]==cond].ix[:,0]
    elif cond == 'ACells': #shortened
        setti=dfcell_names3da[dfcell_names3da.ix[:,1]==cond].ix[:,0]
    dfoki22=df_dist_adh3d.set_index("cell_no"), dfoki22=dfoki22.loc[list(setti),:]
    dfoki22=dfoki22.dropna(), a=dfoki22['time_mcs']
    from itertools import groupby
    results = 'value: len(list(freq)) for value, freq in groupby(sorted(a))' #check the syntax
    return results
```

```
def Amount_3D_ci(dataframes,cond,cond1,conda,time,name):
```

```
    tot2 = [], frames2=[], yes=[]
    for i in range(len(dataframes)):
        tot2.append(amount(dataframes[i],cond,cond1,conda)) #ub half
    for i in range(len(dataframes)):
        frames2.append(pd.DataFrame([(v for v in tot2[i].values())]))
    result2 = pd.concat(frames2, axis=1)
```

```

for i in range(time):
    yes.append(mean_confidence_interval(result2.ix[i], confidence=0.95))
yes=pd.DataFrame(yes)
for i in range(time):
    yes.loc[i,1]=float(yes.loc[i,1]), yes.loc[i,2]=float(yes.loc[i,2])
yes[1] = yes[1].astype(float), yes[2] = yes[2].astype(float)
resulti = pd.concat([yes], axis=1, sort=False)
resulti.columns = ['Cells_region_avg', 'Cells_region_CI_min', 'Cells_region_CI_max']
resulti.to_csv(name, index=False, header='infer')
return resulti

```

8.3.4. Shortened Python Code of Self-Organized Maps

The original SOM code has been described elsewhere (Vettigli, 2018). Here is a shortened and explained version of the code with the original transformations for the resulting data and plots (Hunter et al., 2019; Midha, 2019; Waskom, 2019), where the syntax and content have been shortened.

1) Import the packages, e.g. python's MiniSom, PANDA and NUMPY packages.

2) Import and symmetrize the experimental and model data frames, e.g. for explant culture:
`dfc_experim2 = pd.read_excel('C:/python/' + 'mmc4.xlsx')`

3) Calculate speeds: (d_som is the data frame)
`d_som['speed']=(np.sqrt(d_som['dx']**2+d_som['dy']**2)/d_som['time_interval_sec'])`

4) Perform SOM to these speeds and coordinates with preliminary steps:

```

from minisom import MiniSom
data = np.genfromtxt(
'C:/python/SOM_REF_EXPS AND MODS/irist.csv', delimiter=',', usecols=(1,2,3,4))
#check the used columns (nos) per dataset, model or various experiments
#% Data normalization:
data = np.apply_along_axis(lambda x: x/np.linalg.norm(x), 1, data)

```

Initialization and training:

```

som = MiniSom(10, 10, 4, sigma=4, learning_rate=0.5,
    neighborhood_function='mexican_hat', random_seed=10)
#for mexican hat:seed=10, sigma=2, learning_rate=0.3 are good,
som.random_weights_init(data)
som.pca_weights_init(data)
print("Training...")
som.train_random(data, 5000) # random training
print("\n...ready!")

```

#%If you have already trained, your SOM is ready, and you may want to print the figure:

```

plt.figure(figsize=(10, 10))
# Plotting the response for each pattern in the iris dataset:
plt.pcolor(som.distance_map().T, cmap="RdYlGn")
# plotting the distance map as background, rainbow is good: plt.colorbar()
t=list(range(1,(len(d_som)+1)))
#check the size of markers and colors...

```

```

markers =(['s', 'D', 'o', ',', '!', 's', 'D']*5000+['*']*4592), colors =(['C1']*5000 +['C2']*5000+['C3']*
5000+['C4']*5000+['C5']*5000+['C6']*5000+['C7']*5000+['C8']*4592)
for cnt, xx in enumerate(data):
    w = som.winner(xx) # getting the winner, place a marker on the winning pos. for sample xx
    plt.plot(w[0]+.5, w[1]+.5, markers[t[cnt]-1], markerfacecolor='None',
            markeredgewidth=3, markeredgewidth=3)
plt.axis([0,som._weights.shape[0],0,som._weights.shape[1]]) #this works!
plt.savefig('C:/python/./SOM_REFERENCE_velocity_and_location_4.1.2019_mod.png')
plt.show() # show the figure, do not use pyplot

```

5) Transform the SOM plot to stable speed and coordinate groups:

```

def groups(data,d_som,amount=8):
    a=[], b=[], ll=[]
    for i in range(len(data)): a.append(som.winner(data[i]))
    %%This is how one obtains all groups: aa=pd.DataFrame(a)
    aa.columns = ['a', 'b'], aa.index=d_som.index, oon=[]
    for i in range(0,9):
        for j in range(0,9): oon.append(aa.loc[(aa['a'] == i) & (aa['b'] == j)])
    for i in range(len(oon)):
        ll.append(np.shape(oon[i])),
    lll=pd.DataFrame(ll), lll.columns = ['a', 'b']
    d=(-np.sort(-lll['a'],axis=0))[0:amount] #descending sort
    dd=[], d2=[]
    for i in range(len(oon)):
        if int(np.shape(oon[i])[0]) >= int(min(d)):
            dd.append(int(np.unique(oon[i]['a'])), d2.append(int(np.unique(oon[i]['b'])))
    dd=pd.DataFrame(dd), d2=pd.DataFrame(d2)
    dd['d2'] = d2
    dd.columns = ['dd', 'd2']
    for i in range(amount):
        oon.append(aa.loc[(aa['a'] == dd.ix[i,0]) & (aa['b'] == dd.ix[i,1])])
    return oon,d

```

6) Plot the normed coordinates and speeds with joint plot function (in python):

```

for i in range(len(x)):
    os.mkdir(path[i]), os.chdir(path[i]), show(d_som,on_index[i])
    semi_som(x[i],y[i],z[i],colors[i],condition=True, c2="RdYlGn")
    td=[]
    td.append(pd.DataFrame({"X normed / AU": x[i], (Y and Z likewise)
        "Speed normed / AU": colors[i], (time and distance likewise) }))
    for i in range(len(td)): sns.set(style="white", color_codes=True)
        sns.jointplot(x=td[i]["X normed / AU"], y=td[i]["Y normed / AU"], kind='kde',
            color="skyblue") (and similarly for other measures)

```

7) Finally save the resulting plot with:

```

result2 = test_df2.sort_values(by=["Speed potential normed / AU"],ascending=True)
result2.plot(kind="scatter", x="X normed / AU", y="Speed normed / AU", s=10,
            c="Speed normed / AU", cmap='RdYlGn', ax=ax)
pylab.savefig('plot_speed and x.png'), fig.show()

```

8.4. Tables

Table S1: An example of running results of PSO to 7_B_NP_ADH, optimizing AED (i.e. J), and CL (i.e. λ_{CL}) with Quality (Qual) values, as explained in methods. The (2100) rows of individual iteration values between ‘iter’ 1-58 are not shown.

iter	swrm	prctl	AED	CL	Qual	Time	pBestE	sBestE	ggBestE	Individual_Qual_for_ multiple_runs/ parameter_set
0	0	0	7.04E+00	6.65E+01	3.33E+98	1259.7	1	--	--	-199121.7662
0	0	1	7.22E+00	8.57E+00	-1.33E+05	1601.7	1	--	--	-133649.9206
0	0	2	7.65E+00	1.15E+02	-1.93E+05	1772.4	1	--	--	-194714.4855
0	0	3	7.23E+00	5.32E+01	-1.97E+05	1768.8	1	1	--	-196980.3054
0	0	4	7.26E+00	9.81E+01	-1.95E+05	1922.5	1	--	--	-193914.1285
0	0	5	6.87E+00	3.82E+01	-1.96E+05	1885.3	1	--	--	-195940.7652
0	0	6	6.34E+00	8.51E+01	-1.96E+05	1938	1	--	--	-197419.517
0	0	7	3.43E+00	1.02E+02	-1.94E+05	1954.8	1	--	--	-194569.58
0	1	0	4.32E+00	9.11E+01	-1.96E+05	1941.9	1	--	--	-195977.7241
0	1	1	7.11E+00	4.78E+01	-1.97E+05	1913.3	1	--	--	-197991.3051
0	1	2	6.16E+00	7.17E+01	-1.98E+05	1929.4	1	1	--	-196129.2983
0	1	3	4.93E+00	4.68E+01	-1.96E+05	1935.1	1	--	--	-199050.9773
0	1	4	2.35E+00	1.61E+01	-1.67E+05	1887.1	1	--	--	-164141.1992
0	1	5	5.04E+00	4.37E+01	-1.96E+05	1913.3	1	--	--	-195384.9057
0	1	6	4.53E+00	4.74E+01	-1.96E+05	1918.8	1	--	--	-196442.271
0	1	7	2.89E+00	7.34E+01	-1.95E+05	1940.4	1	--	--	-195150.7899
...
59	2	0	6.28E+00	7.45E+01	-2.01E+05	1780	--	--	--	-197362.9088
59	2	1	6.21E+00	7.47E+01	-2.00E+05	1784.5	--	--	--	-199803.6351
59	2	2	6.51E+00	7.45E+01	3.33E+98	1270.2	--	--	--	1.00E+99
59	2	3	6.64E+00	7.47E+01	-2.01E+05	1788.9	--	--	--	-199013.3661
59	2	4	6.60E+00	7.30E+01	-2.02E+05	1921.8	--	--	--	-200939.4041
59	2	5	6.49E+00	7.44E+01	-2.01E+05	1925.3	--	--	--	-202486.487
59	2	6	7.47E+00	7.22E+01	-2.02E+05	1929.7	--	--	--	-201525.9934
59	2	7	6.84E+00	7.35E+01	-2.01E+05	1918.6	--	--	--	-200825.7131
59	3	0	6.50E+00	6.92E+01	-2.02E+05	1912.5	--	--	--	-201165.8338
59	3	1	6.51E+00	6.89E+01	-1.99E+05	1929.2	--	--	--	-199057.8997
59	3	2	6.54E+00	6.04E+01	-2.02E+05	1914.5	--	--	--	-200621.8862
59	3	3	6.67E+00	6.79E+01	-2.04E+05	1915.8	1	--	--	-202217.8179
59	3	4	6.63E+00	7.34E+01	-2.01E+05	1812.9	--	--	--	-202304.0917
59	3	5	6.31E+00	6.98E+01	-2.02E+05	1814.3	--	--	--	-201656.1594
59	3	6	6.46E+00	7.14E+01	3.33E+98	1283.8	--	--	--	-201400.4207
59	3	7	6.32E+00	7.32E+01	-2.03E+05	1811.9	--	--	--	-203224.0722

Table S2: Average corner and tip values for NP cells at the end of 2D and 3D simulations (800-1000MCS). Distances (d), concentrations (c), and speeds are given for each model (M, 1-8) starting from 2D and then in 3D. The first value is for R, and second for U model type.

	Corner	Tip	Corner	Tip	Corner	Tip
M	d (μm) ^b	d (μm) ^b	c (DL/m) ^c	c (DL/m) ^c	v ($\mu\text{m}/\text{min}$)	v ($\mu\text{m}/\text{min}$)
	30.1±0.8,	24.7±0.8,	0	0	0.125±0.005,	0.128±0.004,
1	30.4±0.8	24.4±0.7	0	0	0.126±0.005	0.126±0.004
	30.0±0.8,	24.6±0.8,	0	0	0.114±0.005,	0.118±0.004,
2	30.6±0.8	24.7±0.8	0	0	0.113±0.005	0.116±0.004
	32.2±1,	24.1±0.8,	21.3±1.3,	19.2±1.1,	0.180±0.008,	0.174±0.006,
3	27.3±1.1	24.2±0.8	19.6±0.9	13.8±0.7	0.177±0.007	0.172±0.006
	31.9±0.8,	21.1±0.7,	9.4±0.5,	9.7±0.5,	0.154±0.007,	0.146±0.005,
4	34.0±0.6	21.1±0.7	8.8±0.5	9.7±0.5	0.162±0.007	0.148±0.005
	29.9±0.8,	23.2±0.7,	8.4±0.5,	7.2±0.4,	0.140±0.007,	0.132±0.005,
5	26.3±1	24.3±0.8	11.9±0.6	7.4±0.4	0.151±0.007	0.132±0.005
	32.3±0.8,	22.3±0.7,	8.2±0.4,	8.6±0.4,	0.139±0.007,	0.137±0.005,
6	34.9±0.6	22.6±0.8	8.8±0.5	8.6±0.4	0.155±0.007	0.139±0.005
	30.1±0.9,	23.7±0.8,	14.0±0.8,	11.8±0.7,	0.151±0.007,	0.140±0.005,
7	25.5±1	23.2±0.8	19.2±1.0	11.7±0.7	0.169±0.007	0.140±0.005
	29.5±0.8,	22.9±0.7,	0	0	0.074±0.003,	0.074±0.002,
8	29.7±0.8	23.0±0.7	0	0	0.074±0.003	0.074±0.002
	25.0±0.4,	20.5±0.3,	0	0	0.0153±0.0003,	0.0153±0.0003,
1	24.9±0.4	20.5±0.3	0	0	0.0154±0.0003	0.0153±0.0003
	25.3±0.4,	20.9±0.3,	0	0	0.0166±0.001,	0.0169±0.0009,
2	27.8±0.4	23.8±0.4	0	0	0.0181±0.0004	0.0180±0.0003
	29.6±0.7,	20.0±0.3,	8.4±0.2,	7.8±0.2,	0.1134±0.003,	0.0973±0.003,
3	31.6±0.7	22.1±0.3	6.8±0.2	5.5±0.2	0.1456±0.0032	0.1525±0.0037
	27.2±0.5,	20.1±0.3,	6.5±0.2,	6.3±0.2,	0.0818±0.002,	0.0792±0.002,
4	26.7±0.5	16.9±0.2	5.4±0.2	5.2±0.2	0.1105±0.0025	0.1087±0.0020
	29.8±0.6,	22.9±0.3,	8.3±0.2,	7.7±0.2,	0.1042±0.003,	0.0923±0.003,
5	30.6±0.6	22.3±0.4	6.4±0.2	5.1±0.2	0.1446±0.0031	0.1522±0.0036
	27.2±0.5,	20.6±0.3,	6.1±0.2,	6.0±0.2,	0.0749±0.002,	0.0724±0.002,
6	25.8±0.5	16.7±0.2	4.7±0.2	4.5±0.2	0.1079±0.0025	0.1068±0.0021
	28.6±0.5,	18.7±0.3,	8.1±0.2,	8.2±0.3,	0.0955±0.002,	0.0828±0.002,
7	29.2±0.6	16.7±0.3	7.8±0.3	6.9±0.3	0.1276±0.0027	0.1271±0.0027
	25.5±0.4,	21.4±0.3,	0	0	0.0152±0.000,	0.0155±0.0003,
8	25.5±0.4	21.3±0.3	0	0	0.0154±0.0003	0.0158±0.0003

Table S3: Average corner and tip values for MM cells at the end of simulations (800-1000MCS). Distances (d), concentrations (c), and speeds are given for each model (M, 1-8) starting from 2D and then in 3D. The first value is for R, and second for U model type.

	Corner	Tip	Corner	Tip	Corner	Tip
M	d (μm) ^b	d (μm) ^b	c (DL/m) ^c	c (DL/m) ^c	v ($\mu\text{m}/\text{min}$)	v ($\mu\text{m}/\text{min}$)
	30.5±0.8,	24.7±0.7,	0	0	0.127±0.005,	0.127±0.004,
1	30.6±0.8	25.0±0.7	0	0	0.126±0.005	0.126±0.004
	30.1±0.8,	25.4±0.8,	0	0	0.114±0.005,	0.127±0.004,
2	29.9±0.9	25.0±0.8	0	0	0.109±0.004	0.126±0.004
	32.0±1.0,	24.1±0.7,	21.7±1.3,	19.4±1.2,	0.180±0.008,	0.176±0.006,
3	37.1±0.7	23.6±0.7	10.7±0.6	13.5±0.7	0.183±0.008	0.174±0.006
	32.5±0.9,	22.3±0.7,	9.5±0.5,	9.7±0.5,	0.152±0.007,	0.146±0.005,
4	29.61±1.0	21.2±0.7	9.1±0.5	9.9±0.5	0.141±0.007	0.145±0.005
	30.3±0.9,	23.3±0.7,	8.4±0.5,	7±0.4,	0.138±0.006,	0.131±0.004,
5	34.5±0.7	23.4±0.8	5.3±0.3	7.1±0.4	0.127±0.006	0.131±0.005
	31.1±0.9,	22.2±0.7,	8.3±0.4,	8.3±0.4,	0.137±0.006,	0.135±0.005,
6	29.5±1.0	22.1±0.7	8.2±0.4	8.7±0.4	0.124±0.006	0.136±0.005
	29.9±0.9,	23.4±0.7,	14±0.8,	12.1±0.7,	0.151±0.007,	0.140±0.005,
7	34.9±0.8	23.3±0.7	8.1±0.5	11.5±0.7	0.134±0.007	0.139±0.005
	29.6±0.8,	23.2±0.7,	0	0	0.074±0.003,	0.074±0.002,
8	29.5±0.8	23.2±0.7	0	0	0.074±0.003	0.074±0.002
	24.7±0.4,	20.7±0.3,	0	0	0.0152±0.0003,	0.0154±0.0003,
1	24.8±0.5	20.7±0.3	0	0	0.0152±0.0003	0.0153±0.0003
	24.9±0.5,	21±0.3,	0	0	0.0167±0.0011,	0.0154±0.0003,
2	27.2±0.4	24.1±0.4	0	0	0.018±0.0004	0.0153±0.0003
	29.8±0.7,	20±0.3,	6.9±0.2,	5.6±0.2,	0.112±0.0034,	0.0969±0.0035,
3	25.2±0.3	17.5±0.2	4.1±0.2	3±0.1	0.0511±0.0012	0.0337±0.0008
	27.3±0.5,	20.4±0.3,	5.3±0.2,	5.2±0.2,	0.0801±0.0025,	0.0783±0.002,
4	27.8±0.4	24.5±0.2	3.8±0.2	3.7±0.2	0.0439±0.001	0.0414±0.0008
	30.1±0.6,	23.0±0.3,	6.5±0.2,	5.2±0.2,	0.1052±0.0036,	0.0922±0.0037,
5	28.8±0.3	23.7±0.2	2.7±0.1	1.8±0.05	0.0277±0.0007	0.0191±0.0004
	27.6±0.5,	20.9±0.3,	4.7±0.2,	4.4±0.2,	0.0752±0.0026,	0.0713±0.0021,
6	28.8±0.3	23.7±0.2	2.5±0.2	2.4±0.1	0.0305±0.0008	0.03±0.0006
	28.6±0.5,	18.3±0.3,	7.7±0.2,	6.8±0.2,	0.0948±0.0029,	0.0832±0.0026,
7	27.7±0.4	19.8±0.2	7.1±0.3	5.3±0.2	0.0409±0.001	0.0358±0.0007
	25.7±0.4,	21.6±0.3,	0	0	0.0152±0.0003,	0.0155±0.0003,
8	25.7±0.4	21.7±0.3	0	0	0.0154±0.0003	0.0158±0.0003

9. PUBLICATIONS AND PRESENTATIONS

Original publications

1. Tikka, P. and Schaefer, F. (2017). **Computational modelling of early nephrogenesis**. *Nephron* 137(2), 155-167, Eurenomics joint annual meeting abstract, doi: 10.1159/000477523.
2. Tikka, P. and Schaefer, F. (2019). **Methods of computational analysis in kidney development**. *Methods Mol Biol* 1926, 235-246, doi: 10.1007/978-1-4939-9021-4_19.
3. Tikka, P. (2019a). **Additional supplementary figures of thesis**. FigShare, URL: <https://doi.org/10.6084/m9.figshare.10315910.v1> [as of 16.11.2019].
4. Tikka, P. (2019b). **CC3D model example codes**. GitHub, URL: <https://github.com/paulitikka/CC3D-Model-Example-Codes> [as of 16.11.2019].
5. Tikka, P. (2019c). **Data of non-optimized and optimized cc3d models and kidney organoid experiments**. FigShare, URL: https://figshare.com/projects/Research_Data/71801 [as of 16.11.2019].
6. Tikka, P. (2019d). **Model data analysis**. GitHub, URL: <https://github.com/paulitikka/Model-Data-Analysis> [as of 16.11.2019].
7. Tikka, P., Mercker, M., Skovorodkin, I., Saarela, U., Sluka, J., Glazier, J. A., Marciniak-Czochra, A. and Schaefer, F. (2019). **Computational modelling of nephron progenitor cell movement and aggregation during kidney organogenesis**. Publication in Preparation.

Presentations

Posters and presentations during EU-funded project of Marie-Curie International Training Network RenalTract (2016-19). The main presentations were held at universities of Paris, Manchester, Glasgow, Bidart, and Indiana (U.S.). Above all, the posters were presented similarly in the university-held conferences at Manchester, Berlin, and Napoli.

10. CURRICULUM VITAE

Personal and Contact Details

Name: Pauli Tikka
Telephone: +49(0) 176 21789056
E-mail: pauli.tikka@med.uni-heidelberg.de

Education

M.Sc. (Tech.) Biotechnology with grade 3/5 University of Turku, 12.6.2014
M.Sc. (Tech.) Mechanical Engineering with grade 3/5 Aalto University, 18.12.2006
Matriculation examination: Normal Lyceum of Helsinki, 3.6.2000
Diploma of Conservatory: The Conservatory of Central Helsinki, 30.5.2000 (cello)

Employment History

- PhD student, University of Heidelberg, Medical Faculty, 1/2016-
Part of RenalTract Consortium studying ‘modelling of nephron progenitor cell movements in early nephrogenesis’
- Research and Master’s Thesis Worker, Hans Knöll Institute, Jena, 9/2013-5/2014
The effect of miRNAs to the regulation of triple negative breast cancer cells
- Summer trainee, Finnish Museum of Natural History, 05/2013-08/2013
Conducting R language testing with phylogenetic tree data and training Perl language
- Design Engineer, Outotec Inc./Lamprotek Inc., 02/2011-06/2011
Design engineering tasks for mineral filtering process equipment
- Data Saver, National Institute for Health and Welfare (HW), 10/2010-11/2010
- Statistical evaluation of the meeting procedures of the executive groups of sections and fields of HW for the use of strategic unit of HW
- Design Engineer, Neste Jacobs Inc., 5/2007-11/2009
- Responsible of the process plant equipment (e.g. filters, safety valves) tasks at plant design division: technical purchase requisition documents and drawings, strength calculations, quotation comparison and checking vendor documentation
- Project worker, Wärtsilä Inc., 1/2006-5/2006
- Student assignment done together with Wärtsilä Inc., Helsinki University of Technology and Kungliga Tekniska Högskolan (KTH, Stockholm, Sweden)
- Master’s Thesis Worker, Outotec Research Inc., 5/2006-12/2006
- Work was carried out in the Laboratory of Physical Chemistry at the Aalto University of Technology. It was a part of a project of Outotec Research Inc.
- Assistant of Mathematics, Helsinki University of Technology, 8/2003-9/2003
Task was to teach a mathematical program in a basic course

11. ACKNOWLEDGEMENTS

There are many people that I would like to give my wholehearted gratitude. Starting from my friend and friends, I would like to acknowledge Dr Sotiris Zarogiannis. With him, I had inspirational discussions amongst writing my works. Naturally, my indebtedness goes to persons such as my parents and brother, who sometimes read my writings. By the same fashion, my favourable connotations should reach the project workers of the RenalTract consortium and Zentrum für Kinder -und Jugendmedizin. The latter have been a constant support in the important as well as mundane matters.

Moreover, I got many insightful advices regarding my computing works from Drs Moritz Mercker, James P. Sluka, and James Glazier. Not to mention to the biology related observations to my modelling, which I received from Drs Ilya Skovorodkin, Tanja Wlodkowski, Maria Bartosova and Ulla Saarela. And of course, finally, my acknowledgements go to my principal instructors, Drs Franz Schaefer, Anna Marciniak-Czochra, and Seppo Vainio. Thank you all very much for your support during the years.

12. EIDESSTATTLICHE VERSICHERUNG

1. Bei der eingereichten Dissertation zu dem Thema

.....

handelt es sich um meine eigenständig erbrachte Leistung.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.

3. Die Arbeit oder Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.

5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt. Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erkläre und nichts verschwiegen habe.

Ort und Datum

Unterschrift