Dissertation
submitted to the
Combined Faculty of Natural Sciences and Mathematics
of the Ruperto Carola University Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

presented by
M.Sc. Erika Tsingos
born in: Rio de Janeiro
Oral examination: 08.10.2019

Characterisation and computational modelling of retinal stem cells in medaka (*Oryzias latipes*)

Referees: Prof Dr Joachim Wittbrodt
Jun-Prof Dr Steffen Lemke

Abstract

The central functional unit of the vertebrate eye is the retina, composed of neural retina (NR), retinal pigmented epithelium (RPE), and non-visual retina (NVR). In amphibians and fish, the retina grows throughout life via different pools of stem cells (SCs). In this work, I combined experimental and computational approaches to elucidate SC dynamics in the three retinal tissues of the teleost fish medaka (*Oryzias latipes*).

I developed a cell centred agent based model to recapitulate post-embryonic growth of the NR and RPE. By accounting for 3D tissue geometry and continuous growth, the model reconciled conflicting hypotheses, demonstrating that competition between SCs is not mutually exclusive with lifelong coexistence of multiple SC lineages.

To understand how NR and RPE regulate their proliferative output to coordinate growth rates, I developed quantitative methods to compare experiment and simulation. I tested the experimental data against simulations implementing two modes of feedback between cell proliferation and organ growth. Thus, I identified that the NR acts upstream to set the growth pace by sending an inductive growth signal, while the RPE responds downstream to this signal.

Leveraging the model, I showed that NR SCs compete for niche space, but tissue geometry biases cells at certain positions to win this competition. Further, NR SCs modulate division axes and proliferation rate to change organ shape and retinal topology.

Motivated by model predictions, I experimentally characterised the large SC population of the RPE, which consisted of both cycling and non-cycling quiescent cells. Putative sister cells exhibited similar temporal dynamics in local clusters, indicating that quiescence was the major mechanism for regulating proliferative output in the RPE.

Finally, I experimentally showed that the NVR grows post-embryonically from a primordium, and shared all known markers for NR SCs in the same spatial distribution. Unlike NR and RPE, the NVR lacked a dedicated niche, instead proliferative cells were distributed throughout the tissue. Lineage tracing revealed a continuous relationship between RPE, NVR, and NR. Thus, the SCs of NR and RPE, and all cells of the NVR displayed plastic multipotency capable of generating all retinal tissues.

By taking advantage of the positive feedback loop between experiment and simulation, this work shines a new light into a fundamental problem – growth coordination of different SC populations in a complex vertebrate organ.

Zusammenfassung

Die zentrale funktionelle Einheit des Wirbeltierauges ist die Netzhaut oder Retina, bestehend aus neuronaler Retina (NR), retinalem Pigmentepithel (RPE), und nicht-visueller Retina (NVR). In Amphibien und Fischen wächst die Netzhaut lebenslang dank verschiedener Populationen von Stammzellen (SZ). In dieses Werk habe ich experimentelle und computergestützte Methoden kombiniert, um die Dynamik von SZ in den drei Netzhautgeweben des Knochenfisches Medaka (*Oryzias latipes*) zu charakterisieren.

Ich habe ein zellzentrum-agentenbasiertes Modell entwickelt, um das postembryonale Wachstum der NR und des RPE zu erfassen. Durch das Abbilden der 3D Gewebestruktur und des kontinuierlichen Wachstums konnte das Modell zwei in Konflikt stehende Hypothesen vereinheitlichen, und somit zeigen, dass Wettbewerb zwischen SZ lebenslange Koexistenz von SZ-Abstammungslinien nicht ausschließt.

Um zu verstehen, wie NR und RPE ihre Zellteilungsrate regulieren, um koordiniertes Wachsen zu bewerkstelligen, habe ich quantitative Methoden zum Vergleich von Experiment und Simulation entwickelt. Ich habe experimentelle Daten mit Simulationen zweier grundlegender Rückkopplungsmechanismen zwischen Zellteilung und Organwachstum verglichen. Dadurch habe ich herausgefunden, dass die NR übergeordnet das Wachstumstempo vorgibt und ein Wachstum-induzierendes Signal schickt, worauf das untergeordnete RPE antwortet.

Mit dem Modell habe ich gezeigt, dass SZ der NR zwar um Platz in der SZ-Nische konkurrieren, dabei aber die Geometrie des Gewebes Zellen an bestimmten Positionen einen Vorteil zum Gewinnen dieser Konkurrenz verschafft. Zusätzlich passen SZ der NR ihre Teilungsachse und Zellteilungsrate an, um Organform und Topologie der Netzhaut zu ändern.

Motiviert durch Modellvorhersagen habe ich die große SZ-Population des RPE experimentell charakterisiert; diese bestand sowohl aus sich teilenden Zellen als auch aus nicht-teilenden ruhenden Zellen. Mutmaßliche Schwesterzellen zeigten ähnliche Dynamik über die Zeit in räumlichen beschränkten Domänen, was darauf hindeutet, dass ruhende Zellen einen großen Beitrag zur Kontrolle der Zellteilungsrate im RPE leisten.

Schließlich habe ich experimentell gezeigt, dass die NVR postembryonal von einer Anlage wächst, und dass dieses Gewebe alle bekannten molekularen Marker in derselben räumlichen Verteilung aufweist wie SZ der NR. Anders als NR und RPE, hat die NVR keine festgelegte SZ-Nische, stattdessen sind teilende Zellen durch das ganze Gewebe zerstreut. Abstammungsverfolgungen zeigten eine kontinuierliche Beziehung zwischen NR, RPE, und NVR. Somit wiesen SZ von NR und RPE sowie alle Zellen der NVR plastische Multipotenz auf, mit der Fähigkeit alle Netzhautgewebe zu bilden.

Durch die positive Rückkopplung zwischen Experiment und Simulation gibt dieses Werk neue Einblicke in eine grundlegende Fragestellung – die Koordinierung des Wachstums verschiedener SZ-Populationen in einem hochkomplexen Wirbeltierorgan.

Contents

Abbreviations

3D	3-dimensional
ABM	agent based modelling
ArCoS	arched continuous stripe
BMP	bone morphogenetic protein
BMP4	bone morphogenetic protein 4
bp	base pairs
BrdU	5-bromo-2'-deoxyuridine
Cas9	CRISPR-associated system 9
ccl25b	chemokine (C-C motif) ligand 25 b gene
cDNA	complementary DNA
CFP	cyan fluorescent protein
CMZ	ciliary marginal zone
cre	causes recombination gene
cre-ert2	causes recombination and estrogen receptor tamoxifen-dependent 2 fusion gene
CRISPR	clustered regularly interspaced short palindromic repeats
csv	comma-separated value
DAPI	4',6-diamidino-2-phenyindole
DNA	deoxyribonucleic acid
dnmt	DNA methyl transferase gene
Dpp	decapentaplegic
dsRed	*Discoscoma sp.* red fluorescent protein
dph	days post hatch
ECM	extra-cellular matrix
EdU	5-Ethynyl-2'-deoxyuridine
eGFP	enhanced green fluorescent protein
GCL	ganglion cell layer
GFP	green fluorescent protein
HDR	homology-directed repair
HF	homology flank
H_2B	histone H2B
hrGFPII-NLS	humanised recombinant *R. Reniformis* GFP II fused to NLS
hpf	hours post fertilisation
hsp70	heat shock protein 70 gene
IdU	5-Iodo-2'-deoxyuridine
INL	inner nuclear layer
IPL	inner plexiform layer
lrp2a	low-density lipoprotein receptor-related protein 2 gene a
loxP	locus of X-over P1
LUT	Lookup Table
mCerulean	monomeric Cerulean
mCherry	monomeric Cherry

mRNA	messenger RNA
NA	numerical aperture
NHEJ	non-homologous end joining
NCC	neural crest cell
NLS	nuclear localisation signal
NR	neural retina
NVR	non-visual retina
oca2	oculo-cutaneous albinism 2 gene
ODE	ordinary differential equation
ONL	outer nuclear layer
OPL	outer plexiform layer
PAM	protospacer adjacent motif
PAI	plasminogen activator inhibitor-type 1 gene
PC	progenitor cell
PCNA	proliferating cell nuclear antigen
PCR	polymerase chain reaction
PDE	partial differential equation
pnp4	purine nucleoside phosphorylase 4 a gene
POM	peri-ocular mesenchyme
P-Smad-1/5/9	Phosphorylated SMAD Family Member 1/5/9
RNA	ribonucleic acid
RFP	red fluorescent protein
ROI	region of interest
RPE	retinal pigmented epithelium
rx1	retinal homeobox transcription factor 1 gene
rx2	retinal homeobox transcription factor 2 gene
rx2	rx2 protein
SC	stem cell
Shh	Sonic hedgehog protein
sgRNA	single guide RNA
smoc1	SPARC Related Modular Calcium Binding 1 gene
st.	developmental stage
tdimer2(12)	tandem dimer 2 (12)
TGFβ	transforming growth factor beta
TL	transmitted light
tlx	tailless gene
TMX	tamoxifen
ubi	ubiquitin gene
UV	ultraviolet
WD	working distance
wt	wildtype
x	-fold concentrate

Contributions

In the following, people that contributed to the work described in this thesis are listed:

Prof Dr Lázaro Centanin performed transplantations of the *Gaudí*$^{LoxP\text{-}OUT}$ line into *cab* medaka embryos; and induction, dissection, staining, and image acquisition of *rx2::*^{ERT2}*cre,Gaudí*$^{2.1}$ samples.

Dr Burkhard Höckendorf performed image processing on *rx2::*^{ERT2}*cre, Gaudí*$^{2.1}$ samples, established retinal map projections as a useful analysis tool, and developed the ImageJ plugin "ArCoS Analysis".

Dr Thomas Sütterlin wrote the majority of the code for EPISIM Simulator.

Mai Thu Nguyen performed cryosections, immunohistochemistry and image acquisition of almost all cryosectioned samples, and also measured the growth of the NVR on cryosectioned *cab* eyes.

Dr Stephan Kirchmaier did injections of medaka *cab* embryos with sgRNA against *oca2*, and performed dissections and image acquisition of mosaic *oca2* samples.

Dr Alicia Perez-Saturnino and **Cristina Muñoz** contributed samples of cryoprotected *ccl25b::GFP*, *tlx::*H_2*B-RFP* fish.

Omar Hammouda did injections of medaka *cab* embryos with sgRNA against *oca2*.

Alex Cornean did injections of medaka *CR(GFP-rx2)* embryos with sgRNA against *oca2*.

Dr Thomas Thumberger helped with design of sgRNAs.

1 Introduction

Light ultimately nourishes all life on Earth, but importantly light also carries information. Over 500 million years ago, during the Cambrian explosion, the information-carrying aspect of light was exploited to find prey or avoid being preyed on using an image-forming sensory system: the eye [Fernald, 2000; Lamb et al., 2007; Nilsson, 2013].

The basic requirement to form an eye is a sensory cell equipped with light-excitable molecules [Fernald, 2000; Nilsson, 2013]. In its simplest form, an image-forming eye requires at least two such photosensitive cells [Nilsson, 2013]. To detect light directionality, a dark pigment shield evolved to cast a shadow on the sensory tissue from certain directions [Nilsson, 2013]. In more complex eyes, this basic unit was extended to include accessory tissues that enhanced the optical properties and metabolic homeostasis of the core sensory-pigment tissue system [Fernald, 2000; Nilsson, 2013]. Further, molecules with sensitivities to different parts of the atmospheric light spectrum evolved to extract more nuanced color information from the environment [Fernald, 2000].

The sensory physiology of eyes demanded high maintenance costs, yet many animal lineages evolved eyes in parallel, evidence for a strong selective pressure [Fernald, 2000; Lamb et al., 2007; Nilsson, 2013]. Eyes might have been a major force behind vertebrate evolution: Enhanced vision in air might have driven our fish ancestors on land [MacIver et al., 2017]. The vertebrate lineage evolved a camera-type eye where a lens – a spherical tissue made up of cells enriched by a crystalline protein – focuses light rays onto the sensory cells to create a crisp image [Fernald, 2000]. This advanced optical system allowed vertebrate eyes to attain among the highest acuities in the animal kingdom [Caves et al., 2018].

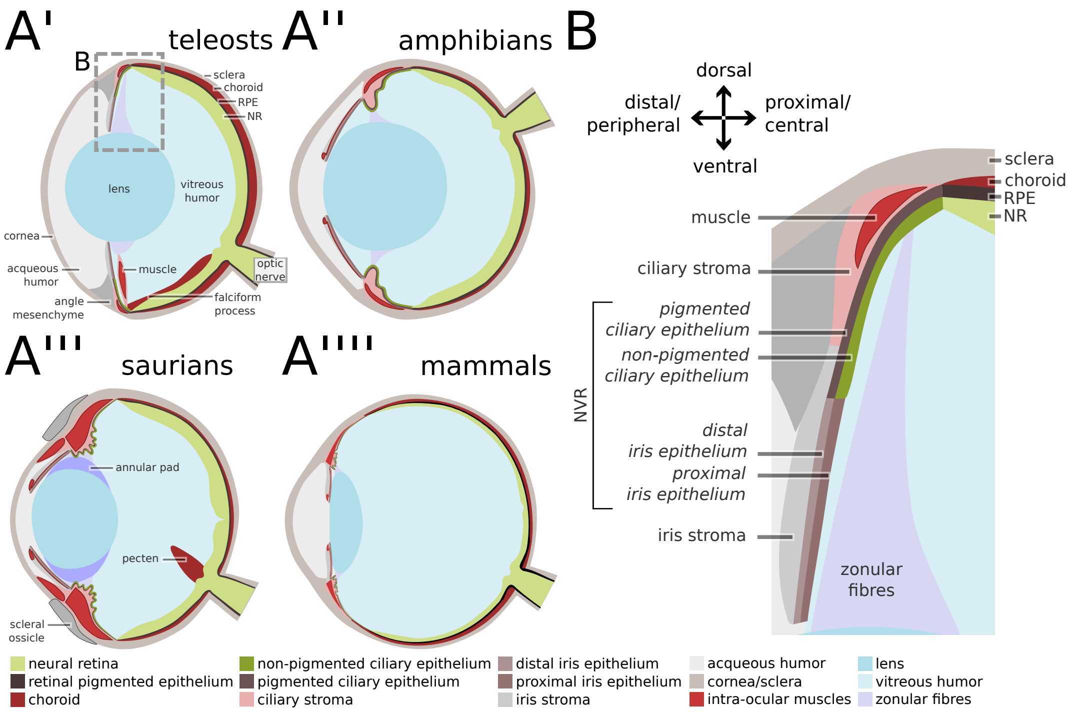


Figure 1.1: Schematic anatomy and embryonic morphogenesis of the vertebrate eye. **A′–A′′′′** Schematic drawings of the generalised structure of eyes in various vertebrate clades. Homologous structures have same colors. Schemes were drawn based on figures in Locket [1977]; Walls [1942] (A′, teleosts), Ott [2006]; Reyer [1977]; Walls [1942] (A″, amphibians), Meyer [1977]; Ott [2006]; Walls [1942] (A‴, birds and reptiles), and Gray [1918] (A′′′′, mammals). **A′** Typical eye of a bony fish. Presence or absence of muscles and falciform process depend on species. **A″** Typical amphibian eye. **A‴** Typical eye of birds and squamates. Pecten is a vascular structure that nourishes the retina, similar in function to the teleost falciform process. Annular pad and scleral ossicles are evolutionary innovations that allow to fine-tune focus by reshaping the lens [Walls, 1942]. **A′′′′** Typical mammalian eye exemplified by the human eye. A common nocturnal or fossorial ancestor lead to a reduction of features [Ott, 2006; Walls, 1942]. **B** Magnification of area boxed in A′, highlighting various tissues of the distal eye. Ciliary and iris epithelia (in italics) can be subsumed to the NVR.

1.1 The vertebrate eye is highly conserved

1.1.1 Anatomical features of the vertebrate eye

The structure of the eye evolved early in the vertebrate lineage, and its major constituent tissues are conserved from jawless fish to mammals [Lamb et al., 2007; Walls, 1942]. The eye is a complex organ, with many concentrically arranged tissues (Figure 1.1 A′–A′′′′).

The basic sensory-pigment tissue unit is created by the neural retina (NR) and the retinal pigmented epithelium (RPE), respectively. Distally, NR and RPE transition continuously to the ciliary and iris epithelia [Walls, 1942] (Figure 1.1 B). The iris epithelia are connected by a loop, creating an anatomically continuous structure [Walls, 1942]. The ciliary and iris epithelia perform diverse functions, including controlling pressure of the intra-ocular fluid and light entry into the eye [Bishop et al., 2002; Freddo, 2013; Walls, 1942]. As they are not directly involved in vision, they have been termed *pars caeca retinae, i.e.* the blind part of the retina [Walls, 1942], or non-visual retina (NVR). Together, NR, RPE, and NVR form the retina [Walls, 1942].

The NVR abuts the lens, which focuses light to a sharp image on the NR. The RPE is surrounded by the choroid, a densely vascularised tissue that also contains darkly pigmented cells that contribute to absorbing stray light [Walls, 1942]. Ensheathing the choroid is the sclera, a fibrous tissue that forms a hard shell. Distally, the sclera is fused to the corneal epithelium [Walls, 1942]. Interstitial tissue composed of blood vessels, pigment cells, and connective tissue forms the stroma distal to the NVR [Walls, 1942].

Although the basic *Bauplan* is conserved, evolutionary innovations independently modified it in a species- and clade-specific way. For example, some fishes evolved the falciform process as an extension of the choroid vasculature jutting into and nourishing the retina (Figure 1.1 A′) [Walls, 1942], while land-dwelling animals – in particular birds and some reptiles – evolved additional musculature in the NVR and its surrounding tissues to reshape the lens as an adaptation to the light refraction at the air-tissue interface [Walls, 1942] (Figure 1.1 A‴).

Vertebrate retinal anatomy

The NR has a highly conserved structure of layered neuronal arrays referred to as retinal lamination [Amini et al., 2018] (Figure 1.2 A′–A″). Nuclei of neuronal subtypes form three concentric nuclear layers normal to the retinal surface: The ganglion cell layer (GCL), inner nuclear layer (INL), and outer nuclear layer (ONL). The cells' elongated projections cluster in two plexiform layers: The inner plexiform layer (IPL) which separates GCL and INL, and the outer plexiform layer (OPL) which separates INL and ONL.

The RPE is a single-cell thick layer that surrounds the NR. Beyond its pigment shield function, it also supports the metabolism of the NR, and acts as a gatekeeper to prevent free diffusion of metabolites from the bloodstream to the NR [Fuhrmann et al., 2014]. At the periphery, the layered structure of the NR merges into an unlaminated single file of cells (Figure 1.2 A″), which progresses into the NVR; directly overlayed, the distal end of the RPE also transitions into the NVR [Johns and Easter Jr, 1977; Walls, 1942]. Both NVR epithelia are cuboidal with a single cell layer each [Walls, 1942].

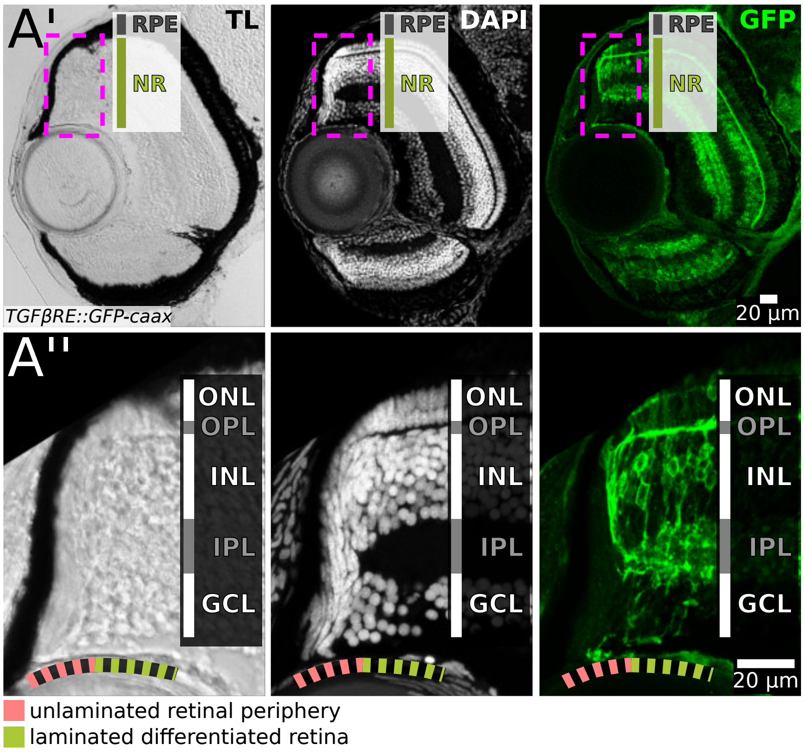

A'
RPE
NR
TL
DAPI
GFP
TGFβRE::GFP-caax
20 µm
A''
ONL
OPL
INL
IPL
GCL
20 µm
🟥 unlaminated retinal periphery
🟩 laminated differentiated retina

Figure 1.2: Vertebrate retina lamination exemplified by the eye of the teleost fish medaka. **A′** Cross-sections of a medaka (*Oryzias latipes*) hatchling eye. DAPI stains the cell nuclei, highlighting the organisation of retinal nuclear layers. The *TGFβ RE::GFP-caax* line drives expression of a membrane-bound GFP in a subset of retinal neurons [Stemmer and Wittbrodt, unpublished]; this line is used here to illustrate the elongated projections of retinal neurons that form the plexiform layers. Data provided by Mai Thu Nguyen. **A″** Enlarged detail of magenta bounded area in A′. Nuclear and plexiform layers of the retina are labelled. Lamination ceases at the distal periphery of the retina.

1.1.2 Embryonic origin of the vertebrate eye

The major steps in eye morphogenesis are conserved in vertebrates [Lamb et al., 2007; Walls, 1942]. Nearly all cellular dynamics of eye morphogenesis have been gleaned from fish embryos such as medaka (*Oryzias latipes*) and zebrafish (*Danio rerio*), which are amenable to acquiring time-lapse movies due to their small size, external development, and transparency [Cavodeassi, 2018].

The eye is a composite organ that develops from various embryonic tissues: the neuroepithelium, neural crest cells (NCCs), mesodermal mesenchyme, and the surface ectoderm [Walls, 1942]. At the anterior neuroepithelium, individual eye field cells migrate and coalesce into two optic vesicles [Cavodeassi, 2018; Ivanovitch et al., 2013; Rembold et al., 2006] (Figure 1.3 A′-A″). An epithelial tube that will form the optic nerve sheath connects developing retina and brain throughout development [Cavodeassi, 2018].

The dorsal optic vesicle contacts the surface ectoderm, inducing formation of the lens placode, and together these structures form the optic cup [Bazin-Lopez

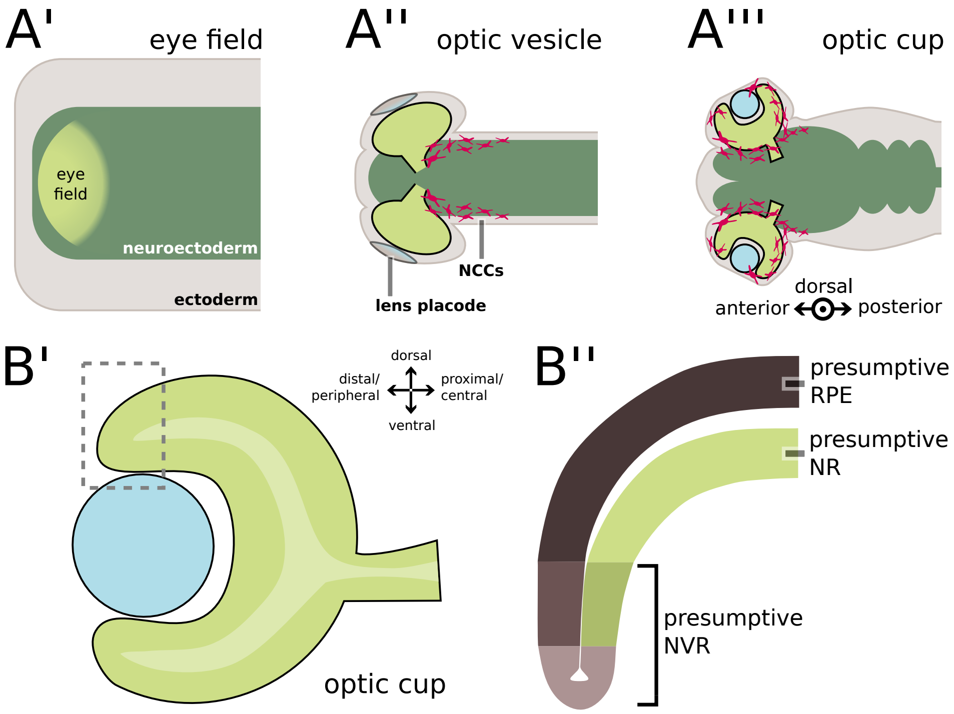


Figure 1.3: Schematic embryonic morphogenesis of the vertebrate eye.
A′–A‴ Major steps of embryonic development of the vertebrate eye. Schemes show dorsal view based on teleost development as depicted in Cavodeassi [2018]. **B′** Schematic optic cup. Note how the optic cup is made of a single continuous epithelial layer that wraps onto itself, enclosing a fluid-filled ventricle. **B″** More detailed schematic of boxed area in B′ illustrating the prospective fates of the distal optic cup.

et al., 2015; Cavodeassi, 2018; Kwan et al., 2012] (Figure 1.3 A″-A‴). The vesicle-to-cup transition entails dynamic neuroepithelial tissue flow akin to gastrulation [Heermann et al., 2015; Kwan et al., 2012; Sokolova, 2019]. The dorsal optic cup forms first, resulting in a ventral fissure [Bazin-Lopez et al., 2015; Cavodeassi, 2018]. Retinal cells lining this fissure have intermediate morphology between prospective NR and RPE [Gestri et al., 2018] and also appear fluid with respect to their fate specification [Eckert et al., 2019]. The fissure does not fuse in many teleosts, where the falciform process occupies the gap to serve as a blood vessel nexus and anchoring point for lens muscles [Locket, 1977; Walls, 1942].

The optic cup gives rise to the retinal tissues: NR, RPE, and NVR (Figure 1.3 B′–B″). Timing of NVR development varies; in anamniotes and some mammals it remains as a primordium that develops only post-embryonically [Gould et al., 2004; Hu et al., 2013; Soules and Link, 2005]. The lens and parts of the cornea are formed by the surface ectoderm, while NCCs give rise to almost all other eye tissues: Corneal stroma and endothelium, iris and ciliary stroma, choroid, sclera, and most eye-internal muscles [Walls, 1942; Williams and Bohnsack, 2015]. The mesodermal mesenchyme forms blood vessels [Gestri et al., 2018].

1.2 Post-embryonic eye morphogenesis

1.2.1 Post-embryonic eye morphogenesis adapts focal length

In all vertebrates, the eye continues maturing post-embryonically to adjust the optical properties of the organ, a process called "emmetropisation" [Wallman and Winawer, 2004]. Emmetropization is mediated by visual inputs integrated by the NR, and communicated to all other tissues of the organ to adjust eye dimensions such that images are always perfectly focused [Wallman and Winawer, 2004]. Failure of this feedback mechanism is thought to underlie eye disorders such as myopia, where the shape of the eye slightly deviates from the optimum [Wallman and Winawer, 2004].

The intra-organ coordination necessary to precisely adjust eye shape may be mediated by signals transmitted in turn by each eye tissue in a cascading relay model [Wallman and Winawer, 2004]. Supporting this model, the RPE appears to signal upstream to influence remodelling of the choroid in zebrafish [Collery and Link, 2018].

1.2.2 Lifelong post-embryonic morphogenesis in anamniotes

Among vertebrates, anamniotes are particularly remarkable for their indeterminate, lifelong growth strategy [Conlon and Raff, 1999; Karkach, 2006]. In these animals, the eye scales to the continuously increasing body size, while fully functioning and maintaining emmetropy [Easter Jr et al., 1977; Fernald, 1991; Johns and Easter Jr, 1977; Lyall, 1957].

Both tissue stretch and increase in cell numbers contribute to anamniote retinal growth [Easter Jr et al., 1977; Johns, 1977; Johns and Easter Jr, 1977]. Though tissue stretch tends to reduce cell density, this is compensated by cell number increase [Johns and Easter Jr, 1977]. Larval fishes have generally lower acuity than adults [Caves et al., 2017; Fernald, 1991], and more generally visual acuity correlates to eye size in fishes [Caves et al., 2018, 2017; Fernald, 1991]. Thus, constant growth has a selective advantage for vision.

A ring-shaped stem cell niche produces new cells in the anamniote retina

The source of new cells in the anamniote NR and RPE is located in the CMZ, a ring-shaped niche that harbours lifelong retinal stem cells (SCs) [Amato et al., 2004; Centanin et al., 2014, 2011; Easter Jr et al., 1977; Fischer et al., 2014; Harris and Perron, 1998; Johns, 1977, 1981; Johns and Easter Jr, 1977] (Figure 1.4

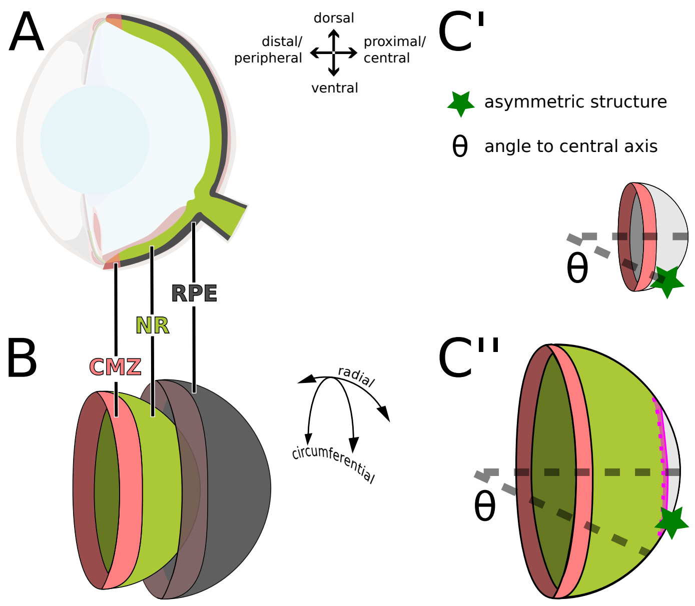


Figure 1.4: Schematic representation of the CMZ and retinal growth in fish.
A Schematic cross-section of the fish eye, with NR and RPE highlighted. The ciliary marginal zone (CMZ) appears as a peripheral band. **B** Schematic 3D blowup model of NR and RPE highlighting the circumferential extent of the CMZ. **C′–C″** Cell addition from the CMZ causes a shift in retinal anatomy; oldest cells are located centrally and newer cells peripherally. However, growth in isotropic concentric annuli would displace asymmetric structures (green star), changing the relative angle θ of these structures with respect to the central axis of the eye [Easter, 1992; Johns, 1977]. Magenta marking in C″ shows former location of the CMZ. Panels B–C″ have been adapted from Tsingos et al. [2019].

A and B). The CMZ adds new differentiated cells appositionally; differentiated retinal cells maintain their relative size over time, and lack cell death and cell mixing [Centanin et al., 2011; Conway et al., 1980; Hunt et al., 1987a,b; Johns, 1977; Johns and Easter Jr, 1977; Stenkamp, 2007].

The anamniote CMZ can be subdivided along the proximo–distal axis according to molecular marker expression and cell division behaviour [Harris and Perron, 1998; Raymond et al., 2006; Shi et al., 2017]. More distal positions bordering the NVR house multipotent lifelong retinal SCs, at intermediate positions there are multipotent progenitor cells (PCs) with limited proliferative capacity, while the most proximal positions next to the laminated retina house PCs with limited proliferative capacity and limited potency [Harris and Perron, 1998; Raymond et al., 2006; Shi et al., 2017]. SCs divide more slowly than PCs [Harris and Perron, 1998; Raymond et al., 2006; Shi et al., 2017].

Functional adaptation leads to asymmetric retinal growth

Fishes are highly diverse and inhabit a range of habitats with different visual information [Caves et al., 2017; Fernald, 1991]. Species-specific adaptations along the retinal circumference evolved to maximize the information gained from the animal's natural environment [Easter, 1992; Walls, 1942]. Cell subtypes and synaptic circuits in the NR vary in a highly correlated manner to chromatic information and instinctive behaviour such as the angle at which prey is detected [Zimmermann et al., 2018]. Similarly, NVR morphology can be asymmetric along the antero-posterior axis to accommodate for lens movement and forward-facing vision [Cameron, 1995; Walls, 1942].

Isotropic addition of concentric annuli of cells from the CMZ would result in displacement of specialised asymmetric structures [Easter, 1992] (Figure 1.4 C′–C″). Thus, various fish species have evolved asymmetric growth patterns correlated to structural asymmetries in the NR [Easter, 1992; Johns, 1977]. Growth asymmetries along the dorso-ventral axis were also observed in the RPE and NVR of frogs, but were not related to asymmetric structures in the eye [Conway et al., 1980; Hunt et al., 1987a,b].

1.3 Elucidating SC dynamics in the anamniote CMZ

1.3.1 Clonal analysis of the anamniote retina

Clonal analysis is a powerful tool to unravel the homeostatic behaviour of cell populations within the anamniote CMZ. This technique allows to investigate the endogenous potency of individual cells without extracting them from their context or entering regeneration regimes [Colom and Jones, 2016; Donati and Watt, 2015]. Individual cells are labelled such that all descendant cells inherit the label; the ensemble of labelled progeny is called a clone [Kretzschmar and Watt, 2012]. Early techniques to clonally label cells include microinjection of vital dyes and transplantation of optically distinguishable cells at embryonic stages (*e.g.* pigmented donor cells into albino hosts) [Conway et al., 1980; Hunt et al., 1987a]. More recently, this experimental repertoire was expanded by the introduction of genetic recombination such as the *cre*-loxP system derived from bacteriophage P1 [Kretzschmar and Watt, 2012].

In African clawed frog (*Xenopus laevis*) optic vesicle cells injected with vital dyes generate all NR cell types [Holt et al., 1988; Wetts and Fraser, 1988]. Prospective CMZ cells of the optic cup make clones exclusively in the NR, exclusively in the RPE, and small few-cell clones with cells in both NR and RPE [Wetts et al., 1989] (Figure 1.5 A′). Due to dye dilution, the NVR was not observed. Grafting optic vesicle or cup pieces of pigmented donors into albino frog hosts creates a permanent label, but restricts analysis to amalgamates of clones, *i.e.* "polyclones" [Conway et al., 1980; Hunt et al., 1987a,b]. Distal NVR and RPE of-

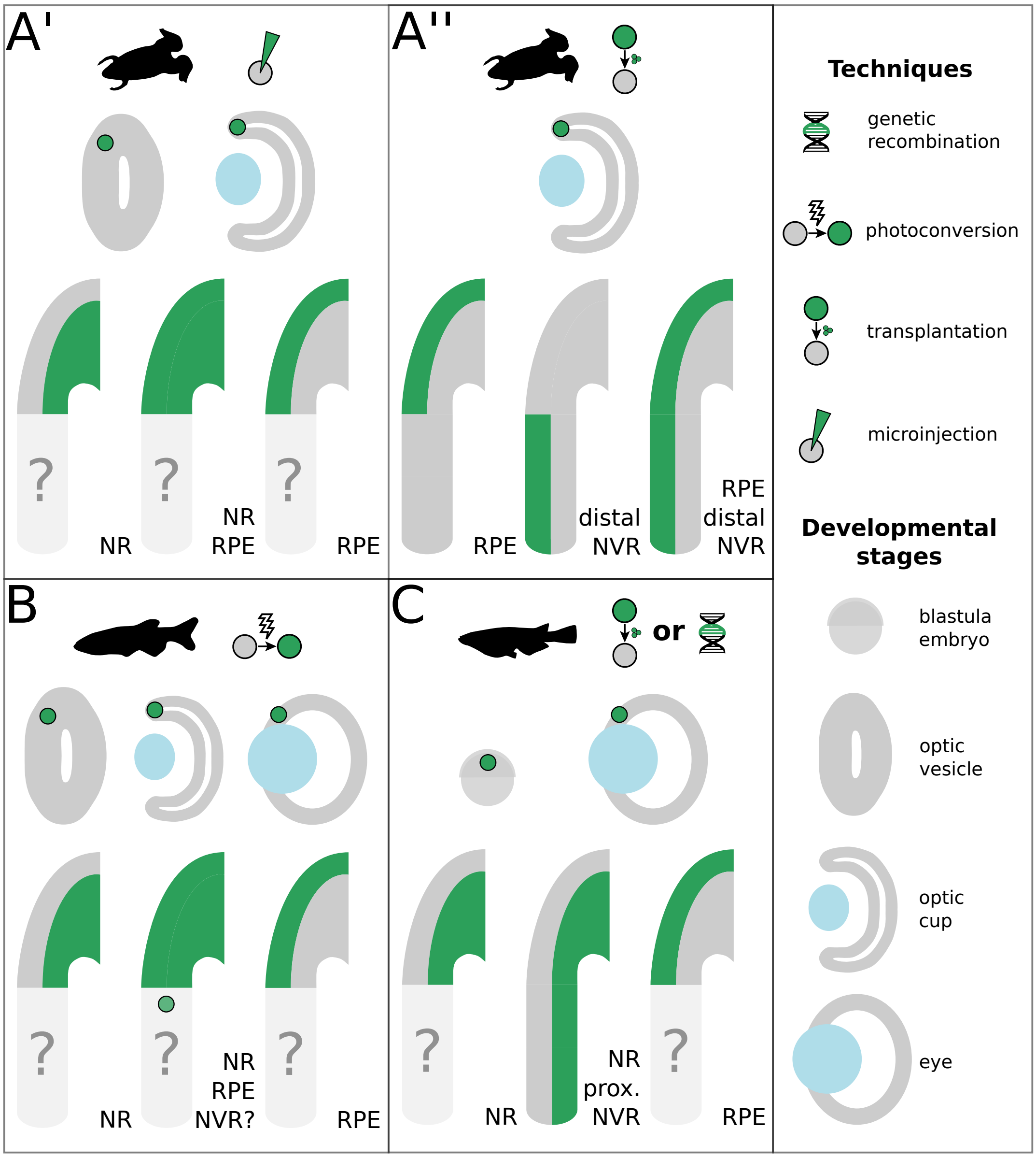


Figure 1.5: Summary of known clonal relations in the anamniote retina.
A′–A″ Data from *Xenopus* [Conway et al., 1980; Hunt et al., 1987a,b; Wetts et al., 1989]. Experiments done by microinjection could not look at post-embryonic timepoints when the NVR developed. **B** Data from zebrafish [He et al., 2012; Tang et al., 2017; Wan et al., 2016]. As for the frog data, the short-term nature of the label precluded longer analyses into the developmental period of the NVR. However, a dormant tip cell with morphology consistent with the NVR was labelled [Tang et al., 2017]. **C** Data from medaka [Centanin et al., 2014, 2011]. Though clones were followed long term, there was no systematic analysis of the NVR. Clones congruent between NR and proximal NVR were observed [Centanin and Wittbrodt, unpublished].

ten form congruent polyclones, but polyclones purely in one or the other tissue were also observed [Conway et al., 1980; Hunt et al., 1987a] (Figure 1.5 A″).

In zebrafish, optic vesicle cells give rise to clones that collectively contain all cell types in the NR [He et al., 2012]. In terms of tissue potency, clones form exclusively in the NR, exclusively in the RPE, or in both tissues [Tang et al., 2017]. The very distal end of the CMZ contains an uncharacterised "dormant tip cell" that is clonally related to both NR and RPE [Tang et al., 2017] (Figure 1.5 B). Interestingly, only cells in the two distal-most rows have a chance to retain daughter cells in the CMZ over time, *i.e.* behave as self-renewing SCs [Wan et al., 2016].

Similar to the experiments in Conway et al. [1980], using a permanent genetic clonal marker in medaka allowed following clones for months, revealing a characteristic shape of lifelong clones as arched continuous stripes (ArCoS) [Centanin et al., 2014, 2011]. Both transplantation in blastulae and *cre*-mediated recombination at hatchling stage result in clones exclusively in the NR or RPE, suggesting a very early lineage specification [Centanin et al., 2014, 2011]. Alternatively, bipotent clones produced too few cells to be seen in analysis of the adult samples. Clones in the NR are congruent to the proximal NVR [Centanin and Wittbrodt, unpublished] (Figure 1.5 C).

Consistently, all labelling experiments show variation in clone size and that clones spanning multiple neuronal layers form columns with relatively low spread tangential to the retina. At early optic vesicle stages, this variability was attributed to stochastic proliferation in an otherwise equipotent pool of cells [He et al., 2012; Holt et al., 1988; Wetts and Fraser, 1988]. At later stages, CMZ clones display a bimodal size distribution: Small clones confined to the proximal retina were attributed to PCs (with limited proliferative potential), while large ArCoS were attributed to SCs (with infinite proliferative potential) [Centanin et al., 2014, 2011; Wan et al., 2016; Wetts et al., 1989].

1.3.2 Competing models of retinal SC dynamics

Historically, SCs were regarded as undergoing deterministic asymmetric self-renewing divisions with one daughter cell fated to differentiate ("invariant asymmetry") [Clevers and Watt, 2018; Watt and Hogan, 2000]. In this model, "stemness" is a hard-wired cell property [Clevers and Watt, 2018; Watt and Hogan, 2000]. Mathematical modelling of clonal analysis bolstered an alternative stochastic model where a population of equipotent cells divides symmetrically, and the number of SCs is regulated by neutral competition for stemness factors ("neutral drift") [Clevers and Watt, 2018; Colom and Jones, 2016]. Thus, stemness may be collectively achieved by a population [Clevers and Watt, 2018].

The stochastic model assumes that stemness is controlled non-cell-autonomously – consistent with the current model of SC regulation through the local microenvironment, or niche [Morrison and Spradling, 2008; Scadden, 2014]. In *Xenopus*, the RPE CMZ appears to be set by such local cues, as cells transplanted from the prospective CMZ to the differentiated optic cup fail to proliferate, but cells transplanted from the differentiated optic cup to the prospective CMZ form lifelong clones [Hunt et al., 1987b]. Though not verified by equivalent experiments, a similar non-cell-autonomous regulation has been hypothesized for fish [Tang et al., 2017]. Indeed, the variability in anamniote CMZ clones supported a stochastic model [He et al., 2012; Wan et al., 2016].

Stochastic neutral competition predicts that the number of clones in the niche decreases over time, eventually reducing to a single clone [Clevers and Watt, 2018; Colom and Jones, 2016]. Contrary to this prediction, the number of clones in the medaka NR and RPE stabilised over time, reminiscent of a deterministic model with invariant asymmetry [Centanin et al., 2014, 2011]. On the other hand, the geometry and temporal evolution of the retinal niche may invalidate some of the underlying assumptions of the stochastic model. Thus, it remains unclear if SCs in the CMZ have invariant asymmetry, neutral drift, or a combination of both – a question that could be addressed via modelling.

1.4 Towards modelling clones in the growing retina

Modelling and experimental work can complement each other in a positive feedback loop of discovery [Bellaïche, 2016; Fletcher et al., 2014]. Models can test the feasibility of competing hypotheses that are difficult to address experimentally [Fletcher et al., 2014], such as invariant asymmetry versus neutral drift. One of the most important steps in creating a model is choosing the right level of abstraction – a model should be neither too simple nor too complex [Bellaïche, 2016; Kitano, 2002; Merks and Glazier, 2005].

Continuum models are suited to phenomena where cell and tissue properties can be averaged [Byrne and Drasdo, 2009]. For example, the temporal development of a population of cells can be described deterministically with an ordinary differential equation (ODE) or with a stochastic master equation. Partial differential equations (PDEs) can model the spatiotemporal distribution of cell density, but cannot explicitly represent discrete individual cells, cell-cell interactions, or the non-trivial geometry and orientation of cells [Alber et al., 2003;

Van Liedekerke et al., 2015]. To address spatially heterogeneous clonal experiments, discrete techniques such as spatial agent based models are the method of choice [Alber et al., 2003; Bartocci and Lió, 2016; Osborne et al., 2017].

1.4.1 Agent based modelling of individual cell heterogeneity

Agent based modelling (ABM) is a bottom-up approach to address complex systems with many interacting components [Abar et al., 2017]. ABM has found widespread application ranging from financial markets to traffic simulation [Abar et al., 2017]. The complexity of ABM precludes analytical solutions, thus models are solved computationally by simulation [Jones and Chapman, 2012].

An agent is defined as a discrete abstract entity that has an internal state, can process information from its local environment, and perform actions according to a predefined set of rules that describe how its internal state evolves in time [Abar et al., 2017; Alber et al., 2003; Bartocci and Lió, 2016; Gorochowski, 2016]. Rules can be formulated in various ways, *e.g.* continuous deterministic rules in the form of ODEs, discrete Boolean expressions, or stochastic rules [Alber et al., 2003]. For example, in a model of traffic, agents would be individual cars and a rule could prescribe that cars stop at red lights.

As the minimal functional unit of life, cells are naturally suited to be modelled as individual agents [Alber et al., 2003; Fletcher et al., 2014; Merks and Glazier, 2005]. In cell based models each cell is represented by a state vector containing information about a cell's position in space, its current internal biochemical state (*e.g.* phase of the cell cycle), and its mechanical properties [Fletcher et al., 2014; Jones and Chapman, 2012]. Spatial ABM approaches can be subdivided into lattice-based and lattice-free models [Byrne and Drasdo, 2009; Fletcher et al., 2014; Jones and Chapman, 2012].

In lattice-based models space is discretized as a lattice, and cells can occupy a given number of lattice spaces [Alber et al., 2003; Drasdo, 2005; Jones and Chapman, 2012; Metzcar et al., 2019]. Widely used formulations include cellular automata (one lattice site per cell) and the cellular Potts model (several lattice sites per cell) [Alber et al., 2003; Metzcar et al., 2019; Van Liedekerke et al., 2015]. Changes in lattice occupancy may be defined by abstract Boolean rules, or equations that account for bonding energies at cell-cell interfaces [Alber et al., 2003; Graner and Glazier, 1992; Jones and Chapman, 2012; Merks and Glazier, 2005].

Lattice-free models represent space as a continuum; the most widely used formulations include vertex models and centre based models. Vertex models represent cells as polygons defined by point particles at their vertices, and edges between vertices represent cellular interfaces [Buchmann et al., 2014; Fletcher et al., 2014]. Centre based models define one or several point particles in the cells' interior, cell shape results from an interaction volume surrounding each particle [Jones and Chapman, 2012; Metzcar et al., 2019; Van Liedekerke et al., 2015]. Rules for mechanical interaction between particles can range from abstract "displacement" models [Bodenstein, 1986] to realistic representations using equations of motion that account for the biophysics of membrane adhesion [Byrne and Drasdo, 2009; Fletcher et al., 2014; Jones and Chapman, 2012].

1.4.2 Previous models of the anamniote retina

ABM has been used to address cell patterning in the anamniote retina. The NR of fishes has a strikingly regular photoreceptor cell arrangement [Fernald, 1991]. Cellular automata were used to investigate how this cell "mosaic" may arise in the zebrafish and medaka retinae by local neighbour exchange biased by differential adhesion [Mochizuki, 2002; Ogawa et al., 2017; Takesue et al., 1998; Tohya et al., 1999]. An alternative vertex based model proposed that planar cell polarity and anisotropic tissue forces generate the mosaic during post-embryonic growth [Salbreux et al., 2012]. In all of these studies, the modelled domain consisted of a 2D flat rectangle of cells of fixed size.

Other retinal neurons of the same subtype (homotypic neurons) also form non-overlapping tilings in many vertebrates, believed to stem from homotypic repulsion mediated by their dendritic arbors [Amini et al., 2018]. This mechanism was investigated in zebrafish and goldfish retinae with a hybrid discrete-continuous approach on a 2D polar coordinate system [Cameron and Carney, 2004; Tyler et al., 2005]. Consistent with the absence of cell mixing and cell death in the post-embryonic NR, repulsion by a short-range signal is sufficient to propagate the pattern [Cameron and Carney, 2004; Tyler et al., 2005].

Clonal experiments in the zebrafish NR were previously modelled with a continuum stochastic approach to reconcile heterogeneous clone size distribution with identical proliferative cell potency [He et al., 2012; Wan et al., 2016]. Using a lattice-free ABM approach that was ahead of its time, Bodenstein [1986] and Hunt et al. [1988] modelled RPE polyclone formation in the growing *Xenopus* retina taking the 3D geometry into account. Their work showed that neutral cell competition and differential proliferation along the dorso-ventral eye axis can explain differences in polyclone morphology [Bodenstein, 1986; Hunt et al., 1988].

1.5 Open questions addressed in this work

1.5.1 Can neutral drift be reconciled with stable clone number?

Long-term retention of a stable number of clones in the medaka NR and RPE supported invariant asymmetry of retinal SCs [Centanin et al., 2014, 2011]. In contrast, a continuum stochastic approach suggested that neutral drift dynamics underlie CMZ proliferation [Wan et al., 2016].

Most experimental setups preclude constant monitoring, and usually only the final timepoint is observable. Mathematical and computational models are crucial to reconstruct the temporal sequence of possible events that generate a particular clone configuration [Klein et al., 2007]. Thus, accounting for 3D retinal geometry with an ABM approach that explicitly represents individual cells similar to Bodenstein [1986] and Hunt et al. [1988] could reconcile the conflicting findings and recapitulate clonal dynamics in the medaka NR and RPE.

1.5.2 How do eye tissues coordinate growth rates at all times?

The visual system is particularly intolerant to deviation in its geometry due to optical constraints [Wallman and Winawer, 2004]. During post-embryonic growth, the eyes of highly visual teleosts such as medaka maintain a precise near-hemispherical shape [Beck et al., 2004; Fernald, 1991; Nishiwaki et al., 1997]. To maintain this precise shape while continuously growing, the growth rates of all tissues must be tightly coordinated.

Throughout the animal kingdom, systemic signals triggered by nutrition coordinate whole-body growth by stimulating all organ systems [Droujinine and Perrimon, 2016; Hietakangas and Cohen, 2009; Lui and Baron, 2011]. However, a systemic signal interpreted individually by each organ (or each tissue within an organ) is susceptible to random fluctuations in growth kinetics, resulting in disproportionate growth [Garelli et al., 2012]. Thus, additional mechanisms must act in parallel to coordinate growth rates.

Hypotheses on why organs grow to a given size include the "chalone hypothesis", which postulates an auto-inhibitory feedback loop mediated by a diffusible chemical (the chalone), and the "mechanical feedback hypothesis", which states that as organ size increases, mechanical compression by surrounding tissue stops proliferation [Buchmann et al., 2014; Lui and Baron, 2011; Wallman and Winawer, 2004]. In the fruit fly (*Drosophila melanogaster*), inter- and intra-organ growth coordination of imaginal discs is mediated by a systemic developmental checkpoint [Colombani et al., 2015; Gokhale et al., 2016]. Damaged discs secrete a hormone that delays all other discs until the affected disc "catches up" [Boone et al., 2016; Colombani et al., 2015; Garelli et al., 2012]. "Catch-up growth" also occurs between long bones of left and right legs of mammals [Roselló-Díez and Joyner, 2015; Roselló-Díez et al., 2017].

Continuous growth of the anamniote visual system is at odds with the chalone hypothesis. Inter-species transplantations demonstrated that the eye grows autonomously at a genetically encoded rate: An eye of a large species of salamander transplanted into a smaller species grows at a rate typical of the donor; the same effect occurs in the reciprocal transplantation [Twitty and Schwind, 1931]. Thus, the mechanical feedback hypothesis cannot hold for the eye at the organ level, as the skull socket of the smaller species should have deterred excessive growth. During continuous growth, the growth rates of all tissues must be tightly coordinated at all times, and cannot rely on a single "catch-up" checkpoint. In summary, current models fall short of explaining intra-organ growth coordination in the anamniote eye.

1.5.3 How do CMZ cells modulate proliferation parameters?

External visual input that regulates emmetropy must be balanced with continuous growth of the organ and cellular growth in the CMZ. Unlike mammals, fish show lifelong emmetropisation potential [Shen and Sivak, 2007; Shen et al., 2005]. Nevertheless, emmetropisation plasticity decreases with age [Shen and Sivak, 2007; Shen et al., 2005], concomitant with decreased cellular proliferation in the CMZ [Johns, 1981]. In chicken (*Gallus gallus*), which has a CMZ during juvenile phases, CMZ proliferation is modulated by specialised neurons in the NR that sense defocus [Fischer et al., 2008]. Manipulation of this system results in eyes that are exclusively radially or circumferentially enlarged, suggesting that these are the two principal axes of eye shape modulation [Fischer et al., 2008]. Together, these data suggest a link between the capacity for CMZ proliferation and emmetropisation.

Maintenance of asymmetric retinal features also occurs at the level of CMZ cells [Cameron, 1995]. As shown in green sunfish (*Lepomis cyanellus*), asymmetric growth is due to differential addition of cells, and the area that grew slowest correlated with higher visual acuity [Cameron, 1995]. Interestingly, after surgical eye rotation in green sunfish, NR and NVR asymmetry continued in their pre-rotation configuration, suggesting that an eye-intrinsic signal independent of visual input regulated asymmetric growth [Cameron, 1996].

Thus, both external visual input and eye-internal signals converge on the CMZ, where these inputs are integrated at the cellular level. However, how CMZ cells modulate their behaviour in response to these stimuli remains unknown.

1.5.4 What is the homeostatic behaviour of SCs in the RPE?

In anamniotes, NR and RPE both grow from the CMZ by addition of concentric annuli of cells [Centanin et al., 2014, 2011; Conway et al., 1980; Hunt et al., 1987a,b]. The SCs that give rise to either tissue appear to derive from distinct immiscible populations [Centanin et al., 2014, 2011]. Despite extensive research on the CMZ little is known about the cells that give rise to the RPE, as most studies have focused on the NR [Amato et al., 2004; Fischer et al., 2014; Shi et al., 2017]. Therefore, it is generally unclear whether information gleaned from the NR can be extrapolated to the RPE. Indeed, even the precise position of RPE SCs in relation to NR SCs was never explored.

1.5.5 Is CMZ function conserved in the NVR of other vertebrates?

Transient CMZ-like function in the NR periphery has been found in juvenile marsupials and during embryonic development in placentals [Bélanger et al., 2017; Kubota et al., 2002; Marcucci et al., 2016]. Birds and some squamates also possess a proliferatively active CMZ during juvenile phases [Fischer et al., 2014, 2008; Todd et al., 2016]. The CMZ-like NR periphery of embryonic mouse (*Mus musculus*) was clonally related to the NVR [Bélanger et al., 2017]. Further, similarities at the level of marker expression between CMZ and NVR have been noted in the chicken [Fischer et al., 2014]. The mammalian proximal NVR has been implicated as the location of adult NR SCs – and was proposed to be functionally homologous to the anamniote CMZ [Ahmad et al., 2000; Tropepe et al., 2000].

To ultimately address functional homology between the anamniote CMZ and the NVR of other vertebrates, a characterisation of the homeostatic cell behaviour of these structures in anamniotes is indispensable. Though it has been speculated that the anamniote NVR grows from the CMZ [Conway et al., 1980], the cellular origin and homeostatic potency of the anamniote NVR has never been investigated. Thus, it remains unclear how the NVR relates to the CMZ in amphibians and fish and whether the relationship between these structures is conserved in vertebrate evolution.

Aims and Approaches

The aim of this thesis was to characterise homeostatic SC dynamics during post-embryonic growth of medaka in the three retinal tissues – NR, RPE, and NVR – and ultimately contextualise these data with findings in other vertebrates. The following key points were addressed with approaches as indicated:

1. **Design of a 3D agent based model of clonal growth in the NR and RPE.**
   A centre based model was implemented in a pre-existing software platform. I developed and tested different model versions, which formed the basis to design experiments and address the balance of stochastic neutral drift and deterministic cell-internal cues in clonal growth.

2. **Investigation of control of intra-organ growth coordination.**
   I used medaka NR and RPE as a model of intra-organ growth coordination. I extracted quantitative clone properties from spatial image data of experimental retinae and simulated clonal experiments to test two fundamental growth-coordinating regulatory feedback loops.

3. **In-depth analysis of SC proliferative parameters of the NR CMZ.**
   Using clonal experiments and simulations I characterised the cellular dynamics in the NR with regards to neutral cell competition, division axis orientation, and regional biases along the dorso-ventral axis. These data were related to regulation of eye growth and shape.

4. **Characterisation of homeostatic dynamics of SCs in the RPE.**
   Motivated by model predictions, I did immunohistochemical stainings and confocal microscopy of marker incorporation in the RPE to characterise previously unknown quiescence and proliferative activity of its SCs.

5. **Uncovering the clonal origin of the NVR and its relation to the CMZ.**
   I investigated growth kinetics, cellular composition, and marker expression of the NVR using immunohistochemical stainings and confocal microscopy. Clonal experiments allowed me to unravel the relationship between NVR and the rest of the retina at different timepoints during post-embryonic development of medaka.

2

Results

2.1 Cell-centre agent based model of the post-embryonic medaka NR and RPE

2.1.1 Scope of the model

The primary objective of the model was to reproduce spatial patterns of clonal experiments in the wildtype medaka retina to aid in interpretation of experiments designed to investigate SC homeostasis. To this end, the model had to incorporate the following properties of the system:

- CMZ cells appositionally add new cells in concentric rings [Centanin et al., 2011; Johns, 1977].
- Individual SCs form clonal progeny in radially oriented stripes [Centanin et al., 2014, 2011].

Based on prior observations, the following assumptions could be made:

- Retinal cell size is constant [Johns and Easter Jr, 1977].
- Retinal cells do not actively move or rearrange, and thus retain their relative position over time [Centanin et al., 2011; Johns, 1977].
- There is no mixing of NR and RPE cells [Centanin et al., 2014, 2011].
- Cell death rate is negligible [Johns and Easter Jr, 1977; Stenkamp, 2007].
- Retinal cells have a tight pseudo-crystalline packing [Johns, 1981; Nishiwaki et al., 1997; Saturnino et al., 2018].
- Proliferation occurs exclusively in the CMZ [Lust and Wittbrodt, 2018].
- There are circa 5 CMZ SC rows [Reinhardt et al., 2015; Wan et al., 2016].
- The CMZ has a fixed extent determined by yet unknown external factors [Tang et al., 2017; Wan et al., 2016].
- The retina is always hemispherical [Easter Jr et al., 1977; Fernald, 1990; Nishiwaki et al., 1997].
- Growth of the retinal radius tends to move cells apart akin to an expanding balloon [Johns, 1977; Lyall, 1957; Ohki and Aoki, 1985].

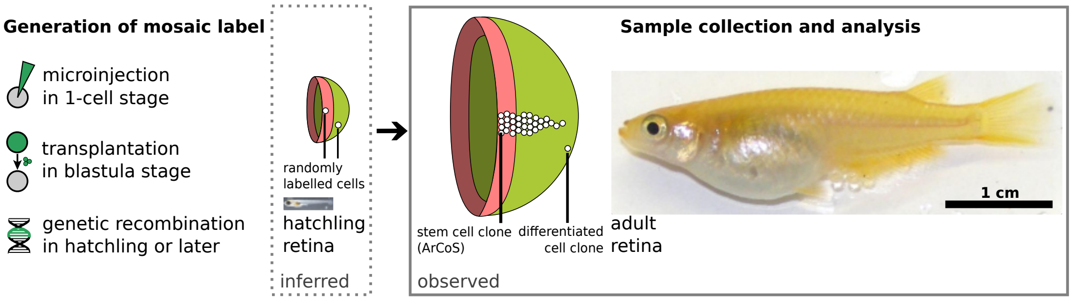

Generation of mosaic label
microinjection in 1-cell stage
transplantation in blastula stage
genetic recombination in hatchling or later
randomly labelled cells
hatchling retina
inferred
Sample collection and analysis
stem cell clone (ArCoS)
differentiated cell clone
adult retina
1 cm
observed

Figure 2.1: Scheme of experimental design for clonal analysis highlighting observed and inferred states.
Three different methods were used to generate mosaic labelling of cells for an ArCoS assay. At the final experimental timepoint, fish were sacrificed and the retina dissected for analysis. All previous states could not be observed had to be inferred. Photos of *cab* fish are to scale; photos and schematic retina drawings adapted from Tsingos et al. [2019].

2.1.2 Modelling choices

Selecting the type of modelling approach

Generating clonal data involves random labelling of individual cells at a given timepoint and subsequent analysis of samples after a certain amount of time has elapsed (Figure 2.1). Due to technical limitations, only the final timepoint is actually observed experimentally, and all previous states of the system must be inferred (Figure 2.1). Ideally, the chosen modelling framework should reproduce the experiment *in silico* and recapitulate basic system properties (Section 2.1.1). In addition, the model would allow to visualise the temporal evolution that cannot be experimentally observed. Spatial cell heterogeneity and following individual cell progeny over time precluded using PDE or continuum stochastic modelling, but agent based models could easily fulfill these conditions. Among agent based approaches, the most suitable method needed to balance computational efficiency (retinal cells number in the millions [Johns and Easter Jr, 1977]), and number of additional assumptions and parameters.

Lattice-based methods struggle with curved surfaces. Cellular Potts models could approximate retinal curvature by decreasing the mesh size, but at the expense of computational tractability. Further, a cellular Potts model would require suitable parametrisation of cell motility to prevent retinal cell mixing and rules describing displacement of lattice cells as a result of eye growth. Lattice-free vertex models in 3D would require simulating several vertices per cell, thus massively increasing the computational demand. Moreover, appropriate parameters for edge tension, intracellular pressure, and vertex transitions were required. In terms of computation, a centre based model where each cell was represented by its centre of mass was the most economic. A centre based model required formulation of appropriate interaction volumes and mechanical interaction rules to generate a tight cell packing. Rather than a detailed description of biophysical interactions at cell interfaces, a more abstract formulation that describes the quasi-steady state distribution of cells can be used when spatial and temporal scales of the modelled tissue are large [Jones and Chapman, 2012]. This simplification could be applied to the retina, as the eye has relatively slow growth dynamics with respect to subcellular time scales, and retinal cells are tightly packed and lack cell mixing throughout post-embryonic growth (Section 2.1.1). Thus, due to computational tractability and relatively few parameter requirements, a centre based model approach was selected.

Further simplifying assumptions

In addition to the prior assumptions described in Section 2.1.1, I made further simplifications based on the following considerations:

- Retinal cells were modelled as spheres.
  NR cells are typical neurons with elongated shapes that form a dense network of synaptic projections. The majority of cell shape variation occurs normal to the hemispherical surface, where cell bodies extend through the plexiform layers to form synaptic connections (Figure 2.2 A′–A″). Similarly, RPE cells are squamous epithelial cells flattened normal to the surface (Figure 2.2 B). The proximal view used for experimental evaluation of wholemount tissue preparations essentially removed the normal axis via projection. Further, retinal cells are tightly packed with little cell mixing, allowing to abstract the system as a collection of spherical particles on a hemispherical surface (Figure 2.2 B).
- The NR was modelled as a single layer of cells.
  NR cells arrange in multiple layers (Figure 2.2 A′–A″). Clonally related sister cells were observed to generally stay close together in normally oriented "columns" with little tangential spread [Centanin et al., 2014; Lust and Wittbrodt, 2018]. I confirmed that this proximity of clonal progeny held true along the entire retinal radius by careful dissection of retinal layers (Figure 2.2 C, pink and white arrowheads). Using a similar argument as above, multiple NR layers could be conceptually compressed into a single spherical particle representing one clonal column (Figure 2.2 D).
- Cells displaced one another due to forces acting on their centre of mass.
  This assumption was intrinsic to the choice of a centre based ABM framework and could be justified by the tight pseudocrystalline packing of the

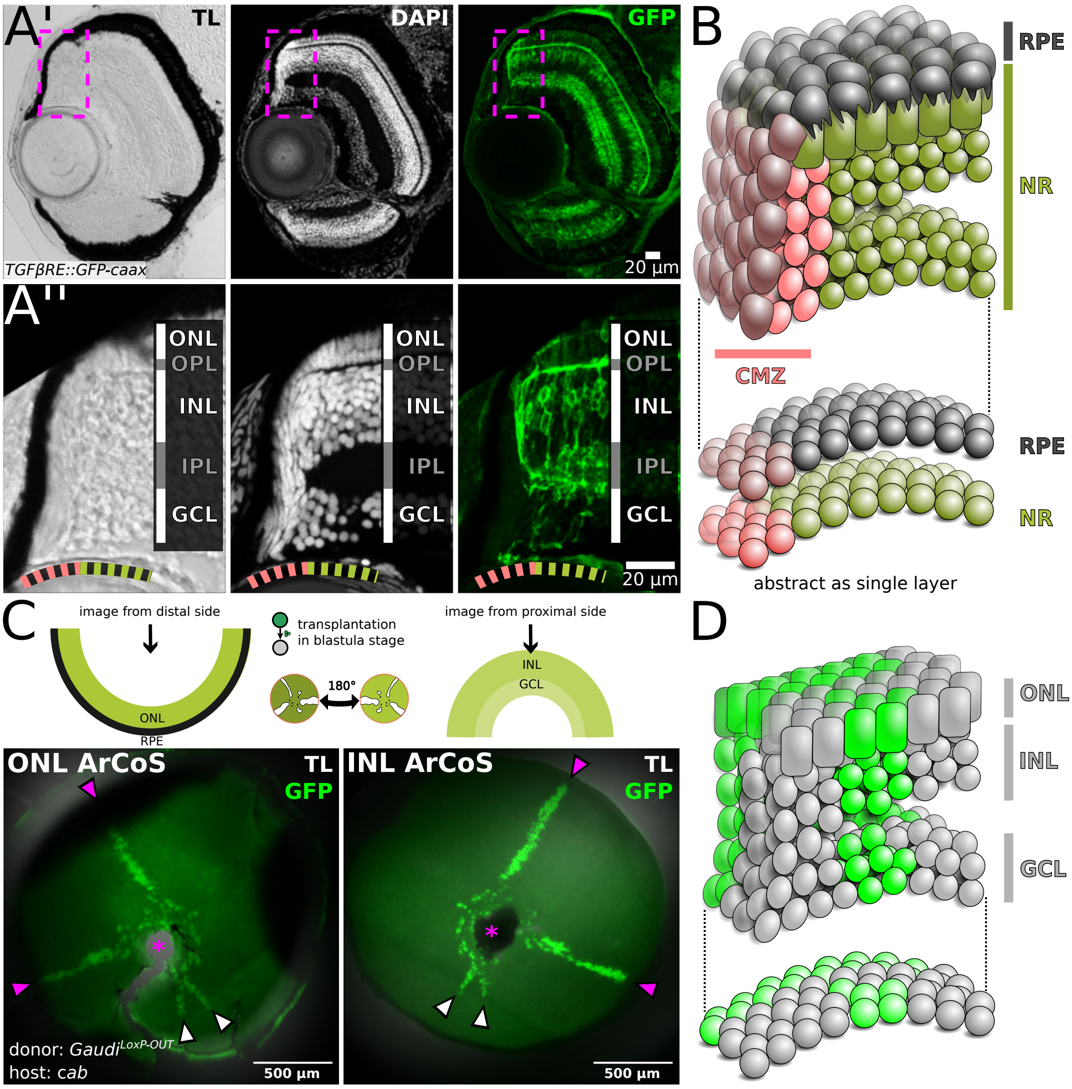


Figure 2.2: NR and RPE were abstracted as a layer of spherical cells on a hemispherical surface. **A′** Cross-sections of a medaka hatchling eye. DAPI stains the cell nuclei, highlighting the dense packing of retinal cells. The TGFβRE::GFP-caax line drives expression of a membrane-bound GFP in a subset of retinal cells [Stemmer and Wittbrodt, unpublished]; this line was used here to illustrate the elongated cell bodies of retinal neurons. Data provided by Mai Thu Nguyen. **A″** Enlarged detail of magenta bounded area in A′. Nuclear and plexiform layers of the retina are labelled. A′–A″ are the same as in Figure 1.2. **B** Simplified scheme of NR and RPE cell somata in the CMZ region in cross-section and in 3D perspective. Both tissues were abstracted as a single layer of spherical cells. **C** Two preparations of the same retina dissected at the level of the ONL. ArCoS spanned all retinal layers along the entire radius of the eye (pink arrowheads). Note how terminating clones also spanned all the layers (white arrowheads). **D** Scheme of 3D perspective of clonally related columns in the NR CMZ region and their projection to a spherical abstraction. Panel (C) has been adapted from Tsingos et al. [2019]; transplantation was performed by Prof Dr Lázaro Centanin.

retina [Johns, 1981; Nishiwaki et al., 1997; Saturnino et al., 2018], lack of active cell movement [Centanin et al., 2011; Johns and Easter Jr, 1977], and the abstraction of the system as a layer of spheres.

- Growth of the eye was quasi-static.
  The growth of the medaka eye is a slow process that occurs over several months (Figure 2.6 A). Thus, the system could be considered quasi-static, *i.e.* at any one timepoint there is no appreciable movement due to growth. This assumption allowed inertial forces to be ignored in the calculation of cell interactions.
- All cells in the CMZ were modelled as fate-equipotent SCs.
  Differences in molecular markers, as well as proliferative and fate potency are well-documented in the CMZ [Centanin et al., 2014; Raymond et al., 2006; Saturnino et al., 2018; Shi et al., 2017; Wan et al., 2016]. As every ArCoS contains all retinal cell types, they must derive from multipotent SCs [Centanin et al., 2014, 2011]. Differences in fate potency at the level of retinal PCs manifest only in the distribution of daughter cells within the layering [Saturnino et al., 2018]. Thus, abstracting the system as a single cell layer allows to ignore fate potency. The effect of introducing differential proliferative potency between SC and PC compartments of the CMZ will be considered in greater detail in section 2.1.5.
- All divisions were symmetric self-renewing.
  Based on previous clonal analyses, NR SCs were surmised to undergo predominantly asymmetric divisions wherein one daughter cell stays as a SC and one daughter cell differentiates [Centanin et al., 2014]. However, due to the absence of time-resolved data, it's unclear whether this asymmetry is predefined at cell division or emerges from the temporal dynamics of the system. For the initial model, I chose the implementation that required the least amount of new assumptions and parameters: All cell divisions were symmetrical divisions where both daughter cells were SCs. The fate of the daughter cells depended on whether they remained inside of the CMZ niche during the temporal evolution of the system.

2.1.3 Model implementation

The text in this section has been adapted from the appendix of Tsingos et al. [2019], the text of which was originally written in its entirety by myself. Where appropriate, the subject was changed from third to first person, and the verb tense was changed from present to past tense. Some paragraphs were expanded to explain the subject matter in greater detail.

The computational implementation of the model explained in the following was done in the platform EPISIM [Sütterlin et al., 2012]. The EPISIM Modeller implementation was designed entirely by myself (Appendix Section 5.8.2), while the Java code implementation in EPISIM Simulator was performed in collaboration with Dr Thomas Sütterlin from the group of Prof Dr Niels Grabe (Appendix Section 5.8.1).

Centre based biomechanical model

The biomechanical model governing physical interactions between cells and all associated parameter values were adapted in their entirety from previous work by Dr Thomas Sütterlin and will only be briefly summarised here [Sütterlin et al., 2017]. This model was designed for ellipsoidal cells; since the model of the retina developed in this work entailed only spherical cells, the notation in the following equations has been simplified.

The model used a lattice-free, centre based overlapping spheres framework to equilibrate the distance of each cell to each adjacent neighbour cell through adhesion or pressure forces (Figure 2.3 A). The attraction or adhesion force $\mathbf{F}_{\mathrm{adh}}$ between a cell c and a neighbouring cell n only occurred if cells were within the interaction distance d_{adh}. The magnitude of this force was calculated as

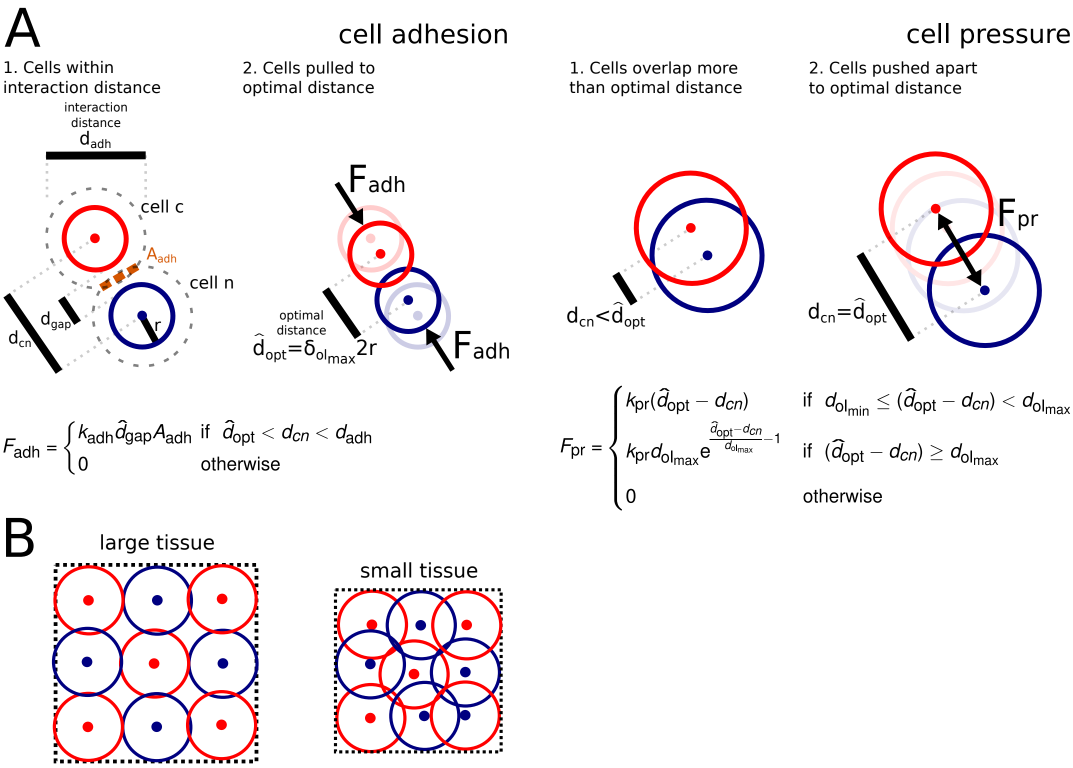
$$F_{\mathrm{adh}} = \begin{cases} k_{\mathrm{adh}} \hat{d}_{\mathrm{gap}} A_{\mathrm{adh}}, & \text{if } \hat{d}_{\mathrm{opt}} < d_{cn} < d_{\mathrm{adh}} \\ 0, & \text{otherwise} \end{cases}, \tag{2.1}$$

where k_{adh} is a spring constant, $\hat{d}_{\mathrm{gap}}$ is a normalised metric of the inter-cell gap, A_{adh} is the intersection area of the interaction spheres, $\hat{d}_{\mathrm{opt}}$ is the optimal target distance, and d_{cn} is the distance between cell centres.

Repulsive or pressure forces $\mathbf{F}_{\mathrm{pr}}$ occurred if cells overlapped more than a minimally tolerated threshold $d_{\mathrm{ol}_{\mathrm{min}}}$ chosen for numerical convenience. The magnitude of this force was given by

$$F_{\mathrm{pr}} = \begin{cases} k_{\mathrm{pr}} \left(\hat{d}_{\mathrm{opt}} - d_{cn}\right), & \text{if } d_{\mathrm{ol}_{\mathrm{min}}} \leq \left(\hat{d}_{\mathrm{opt}} - d_{cn}\right) < d_{\mathrm{ol}_{\mathrm{max}}} \\ k_{\mathrm{pr}} d_{\mathrm{ol}_{\mathrm{max}}} \mathrm{e}^{\frac{\left(\hat{d}_{\mathrm{opt}} - d_{cn}\right)}{d_{\mathrm{ol}_{\mathrm{max}}}} - 1}, & \text{if } \left(\hat{d}_{\mathrm{opt}} - d_{cn}\right) \geq d_{\mathrm{ol}_{\mathrm{max}}} \\ 0, & \text{otherwise} \end{cases}, \tag{2.2}$$

where k_{pr} is a spring constant, and $d_{\mathrm{ol}_{\mathrm{max}}}$ is a maximally tolerated overlap threshold. The pressure force equation had an exponential component triggered by



$$F_{adh} = \begin{cases} k_{adh}\hat{d}_{gap}A_{adh} & \text{if } \hat{d}_{opt} < d_{cn} < d_{adh} \\ 0 & \text{otherwise} \end{cases}$$

$$F_{pr} = \begin{cases} k_{pr}(\hat{d}_{opt} - d_{cn}) & \text{if } d_{ol_{min}} \leq (\hat{d}_{opt} - d_{cn}) < d_{ol_{max}} \\ k_{pr}d_{ol_{max}}e^{\frac{\hat{d}_{opt}-d_{cn}}{d_{ol_{max}}}-1} & \text{if } (\hat{d}_{opt} - d_{cn}) \geq d_{ol_{max}} \\ 0 & \text{otherwise} \end{cases}$$

Figure 2.3: Schematic representation of the biomechanical interactions used in the model. **A** Schematic summary of biomechanical adhesion and pressure rules. Equations were adapted from [Sütterlin et al., 2017], notation was simplified as the model developed in this work uses only equally-sized spheres. Cells exerted adhesive forces F_{adh} when the cell-cell distance d_{cn} lied between the interaction distance d_{adh} and the optimal distance $\hat{d}_{\text{opt}}$. Pressure forces F_{pr} were exerted when the cell-cell distance was shorter than the optimal distance by at least a minimum value $d_{\text{ol}_{\text{min}}}$; an exponential term was used to calculate the force magnitude if the cell-cell distance was shorter than the threshold value $d_{\text{ol}_{\text{max}}}$. $\hat{d}_{\text{gap}}$ was calculated by smoothing d_{gap} to prevent discontinuities in the force calculation [Sütterlin et al., 2017]. A_{adh} was the intersection area of the two spheres given by the interaction distance [Sütterlin et al., 2017]. Further, the following held: $0 < \delta_{\text{ol}_{\text{max}}} < 1$, $d_{\text{ol}_{\text{max}}} < \delta_{\text{ol}_{\text{max}}} 2r$, and $d_{\text{min}} > 0$ [Sütterlin et al., 2017]. **B** The size of the tissue affected the distance equilibrium by imposing a mechanical boundary to the tissue. The same number of cells could distribute less densely on a large tissue as opposed to a small tissue.

very tight cell packing to ensure minimal cell separation (*i.e.* "hard-core model" [Pathmanathan et al., 2009]). The magnitudes calculated by Equations 2.1 and 2.2 were multiplied with the direction vector from cell c to cell n to obtain the force vectors; the equations of motion of each cell centre were solved by numerical integration using the explicit Euler method [Sütterlin et al., 2017]. This model made the standard assumption that inertial terms were negligible [Jones and Chapman, 2012; Osborne et al., 2017]. The time step $\Delta t = 36$s was found to be the upper bound for stable numerical integration [Pathmanathan et al., 2009; Sütterlin et al., 2017]. Parameters were chosen based on previous publications or defined by parameter scan to create a densely packed cell ensemble [Pathmanathan et al., 2009; Sütterlin et al., 2017].

In essence, cells adjusted the distance to all neighbours until they reached an optimal target distance $\hat{d}_{\text{opt}} = \delta_{\text{ol}_{\text{max}}} 2r$, where r is the cells' radius, and $\delta_{\text{ol}_{\text{max}}}$ is the optimal cell-cell overlap. In the absence of proliferation, cell death, and movement, all cells reached a stable distance equilibrium. Additionally, the availability of space for the cells to move in (*e.g.* tissue boundaries) affected the distance equilibrium (Figure 2.2 B). In the model developed in this work, the distance equilibrium was continuously perturbed by proliferating cells in the CMZ and eye growth, and cells were allowed to move only on the hemispherical surface area of the eye globe.

Generating the model's initial condition

The initial condition consisted of a hemispherical surface covered by a single layer of identical spherical cells with radius r (Figure 2.4 A). The tissue surface area was defined as a sphere with centre $\mathbf{s}$ and radius R_{init} (Figure 2.4 A), and cells were constrained to remain on one hemisphere only. To generate this initial condition and achieve the initial distribution of cells on a hemisphere, I proceeded in four steps:

1. I approximated the ideal number of cells N_{init} that fit on the initial hemispherical area based on the overlapping spheres model:

$$N_{\text{init}} = \left\lceil \frac{R_{\text{init}}^2}{(r\delta_{\text{ol}_{\text{max}}})^2} \right\rceil, \tag{2.3}$$

   where $\delta_{\text{ol}_{\text{max}}}$ is the optimal overlap between cells [Sütterlin et al., 2017]. Equation 2.3 was derived from the equation for the curved surface area of a hemisphere and the assumption that each cell occupied a circular area proportional to its radius and the optimal overlap. Histological preparations of hatchling eyes were used to estimate initial retinal radius ($R_{\text{init}} \approx 100$ µm) and retinal cell radius ($r \approx 3.5$ µm). Using a value of $\delta_{\text{ol}_{\text{max}}} = 0.85$ (taken from [Sütterlin et al., 2017]) for the optimal overlap resulted in $N_{\text{init}} = 2261$ cells after rounding up.
2. I obtained a set of nodes by subdividing an icosahedral mesh on the sphere.
3. A cell c was placed on a mesh node located at $\mathbf{r}_c$ if it satisfied the condition

$$\mathbf{r}_{c_x} > \mathbf{s}_x, \tag{2.4}$$

   where the subscript denotes the x-component of the 3-dimensional vectors. This condition ensured that only one hemisphere was populated by

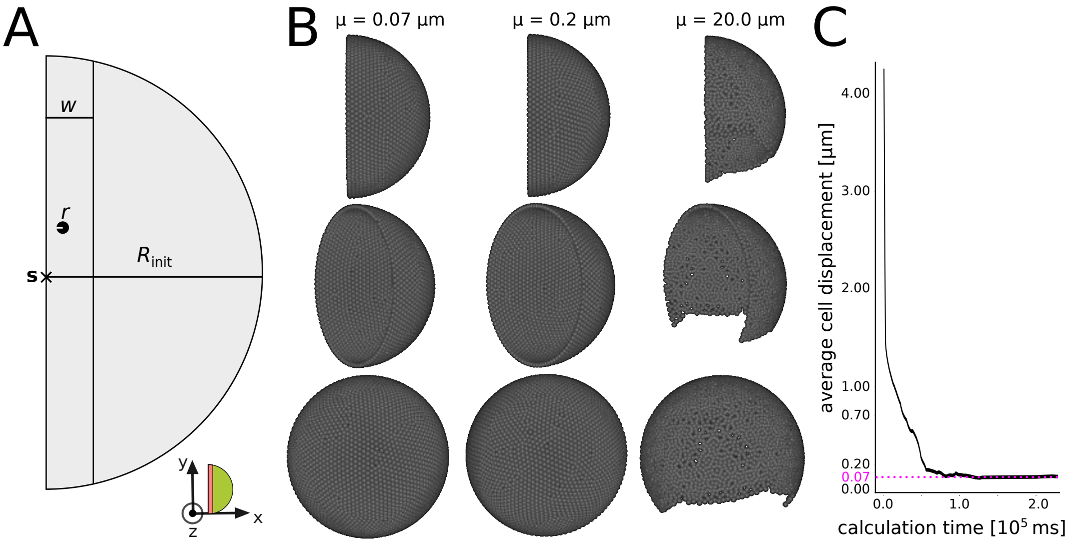


Figure 2.4: The initial distribution of cells on the hemisphere was achieved numerically. **A** Schematic representation of the hemispherical eye in the initial condition of the model. A single black cell with radius r is drawn to scale; w indicates the width of the CMZ. The central-peripheral axis was aligned with the x-axis in the simulation. **B** Initial condition of the simulation when the minimal average displacement threshold μ was varied. Other parameters were as listed in Table 2.1. When $\mu \leq 0.2\frac{\mu\text{m}}{\Delta t}$ cells distributed evenly on the hemisphere. **C** The average cell displacement during model initialisation asymptoted to a value of $0.07\frac{\mu\text{m}}{\Delta t}$. Numerical fluctuations and approximations due to the discrete number of cells versus the continuous area on the hemisphere accounted for the non-zero value. Panels (B-C) have been adapted from Tsingos et al. [2019].

cells. This step was repeated until all N_{init} cells had been placed.

4. To evenly distribute the generated cells on the hemisphere, biomechanical forces were simulated with the model developed in Sütterlin et al. [2017] until cells reached equilibrium, which was defined by the average displacement of all cells falling under a threshold μ during one step Δt of the biomechanical model calculation

$$\frac{1}{N_{\text{init}}} \sum_{i=1}^{N_{\text{init}}} \frac{\Delta \mathbf{r}_i}{\Delta t} < \mu, \tag{2.5}$$

where $\Delta \mathbf{r}_i$ is the displacement of the ith cell. The value of $\mu = 0.2\frac{\mu\text{m}}{\Delta t}$ was determined by parameter scan in preliminary work such that cells were well-distributed and the calculation time prior to simulation start was minimized (Figure 2.4 B-C).

Constraining cells to the hemispherical surface during simulation

Growth of the eye implied cell displacement along the direction normal to the hemispherical surface (Figure 2.5 A). For a cell c at $\mathbf{r}_c$, the new location $\tilde{\mathbf{r}}_c$ was obtained computationally by rescaling the unit vector from the hemisphere's

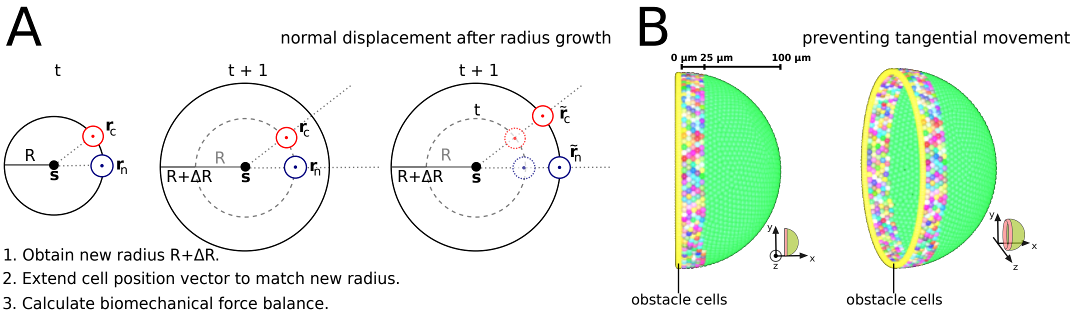

A
normal displacement after radius growth
t
t + 1
t + 1
R
s
$\mathbf{r}_c$
$\mathbf{r}_n$
R+ΔR
R
t
$\tilde{\mathbf{r}}_c$
$\tilde{\mathbf{r}}_n$
1. Obtain new radius R+ΔR.
2. Extend cell position vector to match new radius.
3. Calculate biomechanical force balance.
B
preventing tangential movement
0 µm 25 µm
100 µm
y
x
z
obstacle cells
obstacle cells

Figure 2.5: Computational means for constraining cells to the hemispherical surface.
A Any movement normal to the hemispherical surface was prevented by numerically repositioning all cells at every step of the biomechanical model calculation. This was done both during model initialisation and simulation of temporal development. **B** Obstacle cells (yellow) placed at the equator prevented cell movement in the direction tangential to the hemisphere's edge. Parameters were as listed in Table 2.1. Figure has been adapted from Tsingos et al. [2019].

centre **s** to $\mathbf{r}_c$ with the eye radius at a given simulation step $R(t)$

$$\tilde{\mathbf{r}}_c = \frac{\mathbf{s}-\mathbf{r}_c}{||\mathbf{s}-\mathbf{r}_c||} R(t), \tag{2.6}$$

where || || indicates the Euclidean norm.

Solving the equations of motion could result in cell displacement in any direction. To restrict movement along the direction normal to the hemispherical surface, Equation 2.6 was also used to reposition cells after each integration step Δt. The force balance was then iteratively recalculated with the new cell position. As each simulation step had a time scale of 1 h, one step consisted of 100 iterations of the biomechanical model, allowing the cell ensemble in the simulation to reach a distance equilibrium on the curved hemispherical surface.

To prevent cells from moving beyond the hemisphere's edge, I introduced a ring of tightly packed immobile "obstacle cells" on the sphere's equator that produced a biomechanical roadblock and did not otherwise participate in the simulation (Figure 2.5 B). Force balance between biological cells and obstacle cells was calculated without using the cell adhesion term, such that the obstacle cells acted purely as a repulsive barrier. To produce a tight packing of the obstacle cells, a different optimal overlap parameter $\delta_{\mathrm{ol}_{\mathrm{obstacleCells}}} = 0.5$ was used.

Growth of the hemispherical eye surface

A hatchling medaka grows to sexual maturity within 2–3 months at an approximately linear rate (Figure 2.6 A). For the initial model, I therefore formulated a linear growth equation for the radius R at every simulation step t with slope c_{R}:

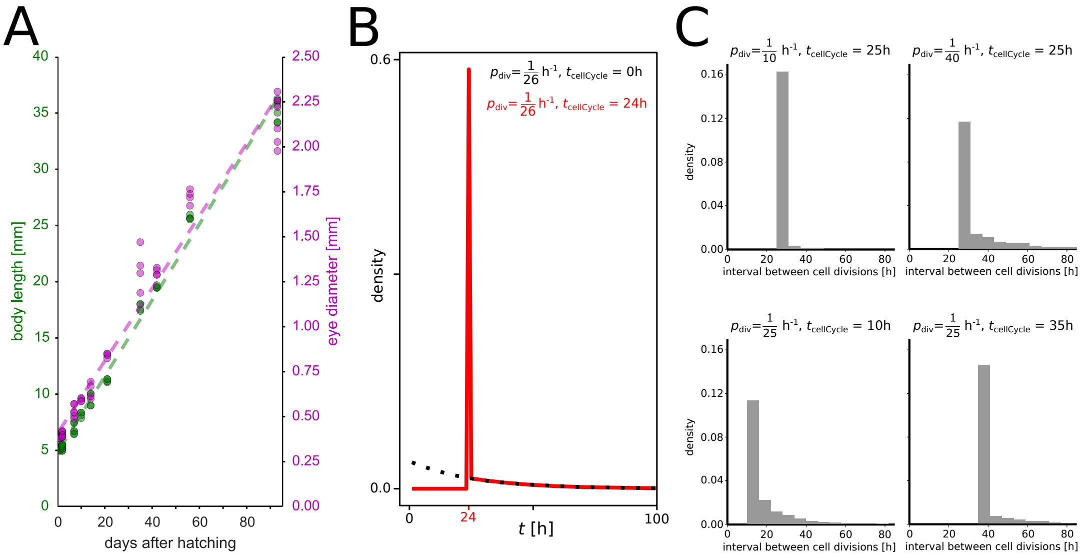


Figure 2.6: *In vivo* growth rates and *in silico* cell division interval distribution.
A Experimentally measured values and linear fit for body length and eye diameter of fish at different ages after hatching. Each timepoint consisted of a different cohort of fish: At 2 dph n = 5, at 7 dph and 93 dph n = 3, for all other timepoints n = 2. Data from both eyes were plotted. In this experiment, eye growth rate was ≈ 0.47 µmh^{-1}. **B** For $t_{\text{cellCycle}} = 0$, Equation 2.10 reduced to a geometric distribution (dashed black line). For $t_{\text{cellCycle}} > 0$ Equation 2.10 generated a truncated geometric distribution where all values $< t_{\text{cellCycle}}$ were summed to the function's value at $t = t_{\text{cellCycle}}$ (solid red line). **C** Histograms of cell division intervals that emerged in the simulation with different parametrisations of p_{div} and $t_{\text{cellCycle}}$. Other parameters were as listed in Table 2.1. Bin size is 6 h. Panels (A, C) have been adapted from Tsingos et al. [2019].

$$R(t) = R_{\text{init}} + c_{\text{R}}\, t\,. \tag{2.7}$$

Growth rates vary between individuals, and retinae recovered from young adult fish had radii in the range of 600-800 µm. Therefore, the growth rate c_{R} could be estimated to range from

$$\frac{600[\mu\text{m}] - 100[\mu\text{m}]}{90 \cdot 24[\text{h}]} \approx 0.23[\mu\text{mh}^{-1}] \tag{2.8}$$

to

$$\frac{800[\mu\text{m}] - 100[\mu\text{m}]}{60 \cdot 24[\text{h}]} \approx 0.49[\mu\text{mh}^{-1}]. \tag{2.9}$$

Further considerations on parameter values and model extensions coupling radial growth of the eye and retinal cell proliferation are described in section 2.2.

Cell proliferation model

The mechanisms underlying commitment to cell proliferation are not characterised in post-embryonic retinal SCs. In other proliferative SC systems, such as mouse tail skin, cell division intervals followed a right-skewed distribution, which could be modelled by combining a minimum division interval with a fixed probability for division [Klein et al., 2007]. In the absence of a minimum cell cycle, division intervals in this model followed an exponential distribution (or its discrete analogue, the geometric distribution, where cell division is a success that occurs with a given probability), and would thus allow arbitrarily short cell cycle times [Klein et al., 2007].

For the model of the medaka retina, I assumed that a proliferative cell may commit to cell division at any time with probability p_{div}. Once committed, if the time since the last cell division was less than the minimum cell cycle time $t_{\text{cellCycle}}$, the division was delayed until this time had elapsed, otherwise the cell divided immediately. These rules generated the following discrete probability distribution of cell cycle intervals:

$$\Pr(\mathrm{X}=k)=\begin{cases}0, & k<t_{\text{cellCycle}}\\ \sum_{k=1}^{t_{\text{cellCycle}}}\left[(1-p_{\text{div}})^{k-1}p_{\text{div}}\right], & k=t_{\text{cellCycle}}\\ (1-p_{\text{div}})^{k-1}p_{\text{div}}, & k>t_{\text{cellCycle}}\end{cases}, \tag{2.10}$$

where X is a random trial and k is the number of trials before a success (cell division). The distribution in Equation 2.10 differs from the geometric distribution by truncation of the values on the left of $t_{\text{cellCycle}}$ and summation of the truncated values to the peak at $t_{\text{cellCycle}}$ (Figure 2.6 B). The actual distribution in the simulation could diverge from Equation 2.10 due to factors that emerged at runtime, *e.g.* the non-random removal of cells from the proliferative pool by differentiation. Varying the value of the parameters showed that, as expected, the magnitude of the peak and the exponential decay increased with increasing p_{div}, while increasing $t_{\text{cellCycle}}$ shifted the distribution to the right, and also increased the peak height (Figure 2.6 C).

Further considerations on parameter values and model extensions coupling radial growth of the eye and retinal cell proliferation are described in section 2.2.

Placement of daughter cells after cell division

The introduction of new cells into the simulation followed the general procedure used in cell centred agent based models as previously described [Sütterlin et al., 2017]. Briefly, when a cell c located at $\mathbf{r}_c = \begin{pmatrix} x_c & y_c & z_c \end{pmatrix}^{\mathbf{T}}$ divided, a new cell n was introduced into the simulation at position $\mathbf{r}_n = \begin{pmatrix} x_n & y_n & z_n \end{pmatrix}^{\mathbf{T}}$. The initial distance between cells $\|\mathbf{r}_n - \mathbf{r}_c\|$ was chosen to be a small non-zero value. The coordinates $\mathbf{r}_c$ and $\mathbf{r}_n$ were fed as initial input to the biomechanical model, which then calculated how force balance repositioned the cells. This means that initially, the two daughter cells almost completely overlapped and then gradually separated, displacing any neighbouring cells in a "domino effect". Thus, the final position of the daughter cells at the beginning of the simulation step following division might not have fully corresponded to the initial position that was calculated upon division, but was biased by it.

By default, cells divided with random division axis orientation, where I calculated $\mathbf{r}_n$ as

$$\mathbf{r}_n = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \rho \begin{pmatrix} \mathrm{X}-0.5 \\ \mathrm{X}-0.5 \\ \mathrm{X}-0.5 \end{pmatrix} \tag{2.11}$$

where each X is a uniformly distributed random number in the interval $[0,1]$, and $\rho = 0.005$ µm is a scaling constant that defined the maximum initial distance between daughter cells.

Definition of proliferative and differentiated cell types

The model considered two cell types: Differentiated cells and proliferative (stem) cells. Only proliferative cells divided, and all divisions resulted in two identical proliferative daughter cells. The fate of cells depended on their position on the eye hemisphere. A cell c at $\mathbf{r}_c$ became a differentiated cell type if it moved beyond the width w of the CMZ:

$$\mathbf{r}_{c_x} > \mathbf{s}_x + w, \tag{2.12}$$

where $\mathbf{r}_{c_x}$ and $\mathbf{s}_x$ denote the x-component of $\mathbf{r}$ (cell position) and $\mathbf{s}$ (centre of eye globe), respectively. Differentiated cells could not revert to proliferative cells.

Model parameters

Unless otherwise stated, the following parameters were used for simulations:

Table 2.1: Parameters used for simulations; adapted from Tsingos et al. [2019]. Parameters of the biomechanical model are identical to Sütterlin et al. [2017] and are not listed.

Description	Parameter	Value	Reference/Explanation
		Biomechanical model parameters	
Biomechanical calculation step	Δt	36 s	Sütterlin et al. [2017]
Seconds per simulation step	t_{simStep}	$3600\ \mathrm{s} \cdot \mathrm{simStep}^{-1}$	Sütterlin et al. [2017]
Optimal overlap (obstacle cells)	$\delta_{\mathrm{ol}_{\mathrm{obstacleCells}}}$	0.5	Determined by parameter scan to create a tight barrier to cell movement.
Optimal overlap (retinal cells)	$\delta_{\mathrm{ol}_{\max}}$	0.85	Sütterlin et al. [2017]

Initial distance between daughter cells	ρ	0.005 µm	Sütterlin et al. [2017]
Initial condition parameters			
Initial radius of eye globe	R_{init}	100 µm	Estimated from preparations of hatchling eyes.
Minimal displacement threshold	μ	$0.2\ \frac{\mu m}{\Delta t}$	Determined by parameter scan to generate an even initial cell distribution.
Simulation parameters			
Retinal cell radius	r	3.5 µm	Estimated from histological sections.
Width of the SC domain	w	25 µm	Estimated from histological sections.
Minimal cell cycle length	$t_{\text{cellCycle}}$	24 h	Chosen to produce a biologically plausible growth rate. See Section 2.2.
Probability of commitment to cell division	p_{div}	$\frac{1}{26}\text{h}^{-1}$	
Growth rate of the eye radius	c_R	$0.25\ \mu\text{mh}^{-1}$	

2.1.4 The model recapitulates basic features of the medaka retina

The model recapitulated the two basic properties of the system that were necessary to compare it to experimental datasets: Addition of cells in concentric rings, and clonal progeny in radially oriented stripes (Figure 2.7 A, B). Analogous to experimental data [Centanin et al., 2014, 2011; Höckendorf, 2013], the simulated data could be subdivided into three zones (Figure 2.7 D):

1. The **pre-induction retina** with cells that were differentiated at the timepoint of labelling,
2. the **induction ring** with small clusters derived from central SCs,
3. the **post-induction retina** with stable clonal sectors or ArCoS.

All proliferative cells in the model were SCs, yet the model generated an induction ring (Figure 2.7 C-D), which had previously been interpreted as originating from PCs [Centanin et al., 2014, 2011]. The virtual induction ring formed by cell competition: Divisions of more peripheral cells pushed their central neighbours out of the virtual CMZ, causing the latter to differentiate (forming "terminated clones"). This mechanism lead to full differentiation of more central clones; indeed, only a subset of cells from the two most peripheral rows contributed long term to the simulated retina by forming a stable sectoring pattern in ArCoS (Figure 2.7 C). Thus, a model accounting for 3D geometry and growth of the niche reconciled neutral drift dynamics with growth in a stable number of ArCoS.

A quantitative comparison of these data to experimental NR data is presented in greater depth in section 2.3.1.

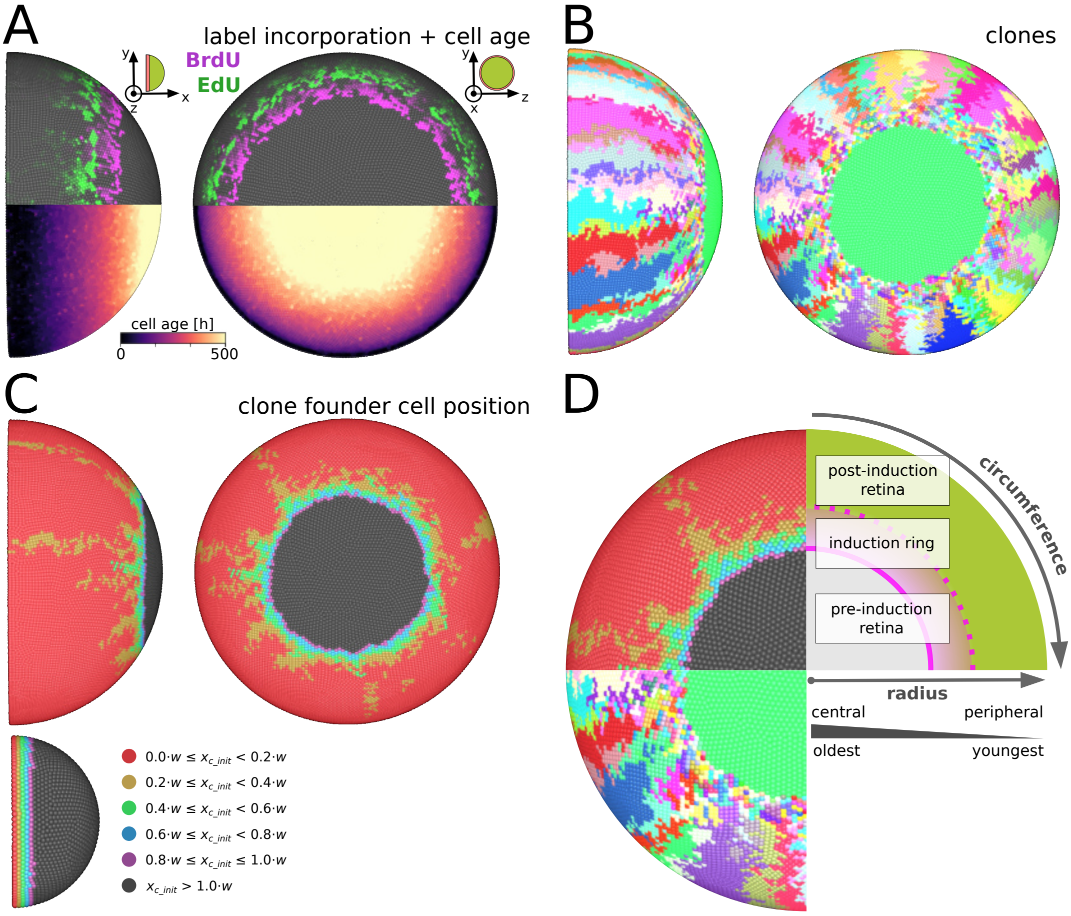

A
label incorporation + cell age
BrdU
EdU
y
x
z
y
z
x
cell age [h]
0
500
B
clones
C
clone founder cell position
0.0·w ≤ x_{c_init} < 0.2·w
0.2·w ≤ x_{c_init} < 0.4·w
0.4·w ≤ x_{c_init} < 0.6·w
0.6·w ≤ x_{c_init} < 0.8·w
0.8·w ≤ x_{c_init} ≤ 1.0·w
x_{c_init} > 1.0·w
D
post-induction retina
induction ring
pre-induction retina
circumference
radius
central
peripheral
oldest
youngest

Figure 2.7: The model reconciles neutral drift with a retinal clonal dynamics.

A Composite image of simulation from two different angles; top: screenshots of 3D render showing virtual thymidine analogue incorporation; bottom: 3D scatter plot of cell age (time elapsed since the last cell division). Virtual BrdU pulse was from 150 h to 175 h, virtual EdU pulse from 250 h to 275 h. Cells in the model incorporated BrdU/EdU only in the simulation step when they divided. Daughter cells inherited half of the mother cell's BrdU/EdU (Appendix Figures 5.16 and 5.17). **B** Same simulation as in A showing clonal growth in ArCoS. The central light green area consists of initially differentiated cells of the pre-induction retina. **C** Top: Same simulation as in A–B with clones coloured according to the initial position along the x-coordinate x_{c_init} of the clone founder cell. Bottom: Magnified example of the initial model condition with the same colouring. Each cell's colour was assigned based on its position relative to the CMZ width w. Note how central CMZ cells created the induction ring. **D** Composite image from B-C with a schematic representation of retinal domains adapted from Tsingos et al. [2019]. The retina could be subdivided into three concentric domains from the centre to the periphery. Simulations in A–C used $p_\text{div} = 30^{-1}\text{h}^{-1}$. Other parameters as in Table 2.1.

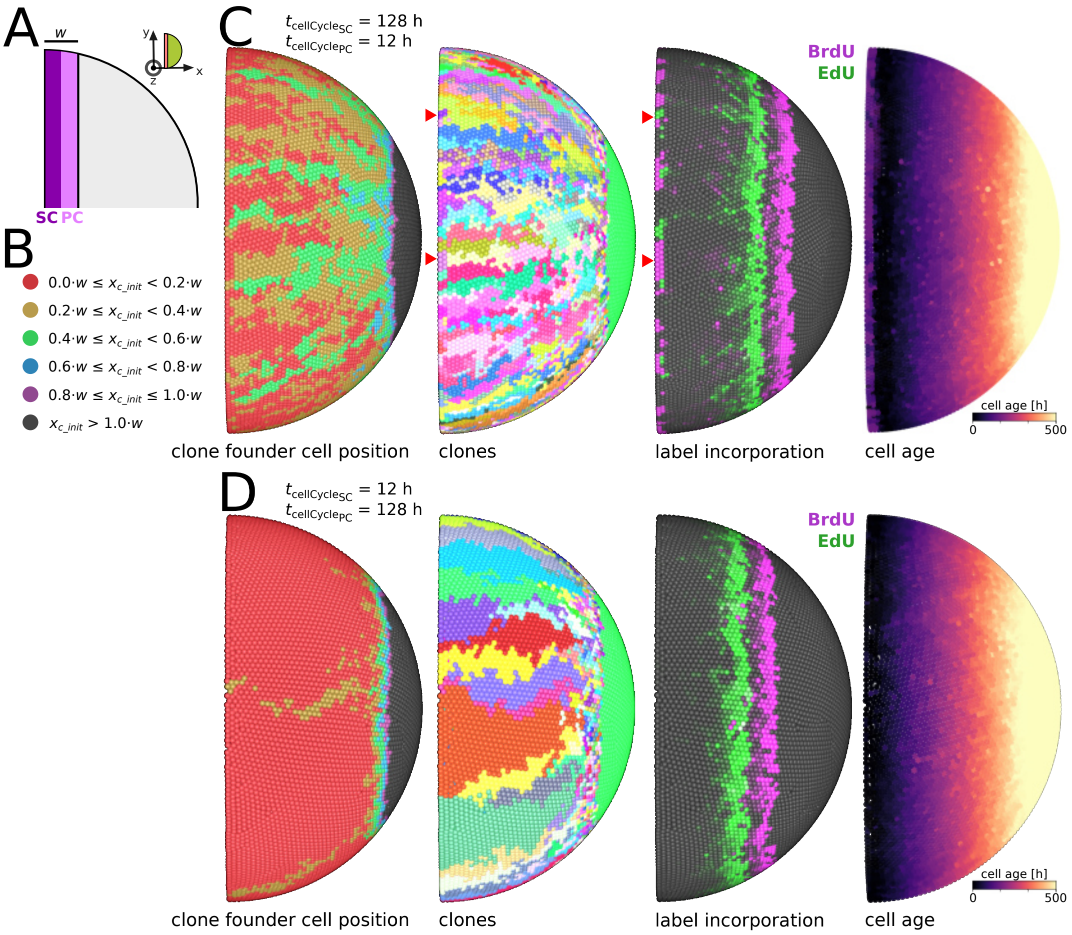


Figure 2.8: Introducing fast-cycling PCs disrupts ArCoS formation.
A Scheme of CMZ subdivision into SC and PC domains. **B** Legend for colour-code in left-most panel in C and D. Each cell's colour was assigned based on its clone founder cell position relative to the CMZ width w. **C** Screenshots and cell age plot of simulation with fast-cycling PCs and slow-cycling SCs. Many central cells invaded peripheral positions. The gradient in cell cycle times lead to clone fragmentation and label retention in peripheral CMZ cells, which were also older than more central CMZ cells. Red arrowheads mark examples of label-retaining clone fragments confined to the peripheral CMZ. **D** Screenshots and cell age plot of simulation with slow-cycling PCs and fast-cycling SCs. This simulation formed clearer ArCoS sectors, resembling simulations without a gradient of cell cycle times. Simulations in (C-D) used the following parameters: $t_{\text{cellCycle}}$ as indicated in figure. BrdU pulse was from 150 h to 180 h, EdU pulse from 250 h to 280 h. Other parameters were as listed in Table 2.1. Schematic drawing of coordinate system in (A) adapted from Tsingos et al. [2019].

2.1.5 Modelling differential cell cycle duration of PCs and SCs

One of the initial model abstractions was that NR CMZ cells were fate-equipotent SCs. To better justify this simplification, I investigated the impact of a more sophisticated CMZ model on the model's output.

In the NR, the CMZ is subdivided into stem and progenitor cell domains that differ in cell cycle dynamics; specifically, SCs divide less often than PCs [Shi et al., 2017]. SCs are estimated to divide once per week, while PCs have been recently shown to have an average cell cycle length of 12 h [Becker and Wittbrodt, unpublished], but otherwise the exact parameters for cell cycle length and probability of division are not known. I investigated the effect of introducing differential cell cycle times along the central-peripheral axis of the model by varying $t_{\text{cellCycle}}$ in each half of the CMZ. For a cell c located at $\mathbf{r}_c$, $t_{\text{cellCycle}}$ was set according to

$$t_{\text{cellCycle}} = \begin{cases} t_{\text{cellCycle}_{\text{SC}}}, & \mathbf{r}_{c_x} < \mathbf{s}_x + \frac{w}{2} \\ t_{\text{cellCycle}_{\text{PC}}}, & \mathbf{s}_x + \frac{w}{2} \leq \mathbf{r}_{c_x} < \mathbf{s}_x + w \end{cases}. \tag{2.13}$$

The change from a SC to a progenitor cell cycle length happened upon crossing the threshold $\frac{w}{2}$, *i.e.* half of the virtual CMZ's width w (Figure 2.8 A).

On a qualitative level, introducing faster proliferation in the proximal CMZ lead to an increase of the width of the induction ring and greater fragmentation of ArCoS (Figure 2.8 C). In contrast to simulations where all CMZ cells divided at the same rate, the peripheral CMZ contained many small clone fragments that retained thymidine analogues (Figure 2.8 C red arrowheads). These differences could be intuitively explained by considering that central clones terminated by being pushed out of the niche by more peripheral divisions. In this model where peripheral cells proliferated less often, central clone termination occurred at a reduced rate. Importantly, this "competition deficit" was continuous, affecting also the progeny of SCs, leading to isolated clusters of peripheral clonal SCs (Figure 2.8 C red arrowheads). Reversing the relationship, such that $t_{\text{cellCycle}_{\text{SC}}} < t_{\text{cellCycle}_{\text{PC}}}$ reduced the width of the induction ring and lead to clear sectoring into ArCoS (Figure 2.8 D).

The experimental NR data showed a relatively narrow induction ring zone and clear ArCoS sectoring [Centanin et al., 2014, 2011], which qualitatively was closest to the model where $t_{\text{cellCycle}_{\text{SC}}} < t_{\text{cellCycle}_{\text{PC}}}$. This result contrasted to experimental observations that SCs cycled more slowly than PCs. This paradox could be resolved by considering that the model only implemented one layer of differentiated cells, and that *in vivo* PCs may cycle faster to act as a "transit-amplifying" population, multiplying the proliferative output of the SCs to fill all the layers of the NR [Saturnino et al., 2018]. By abstracting the NR as a single-layered tissue, cell cycle differences along the CMZ width could be factored out in the model, reducing the number of parameters and allowing to focus the analysis purely on the SCs.

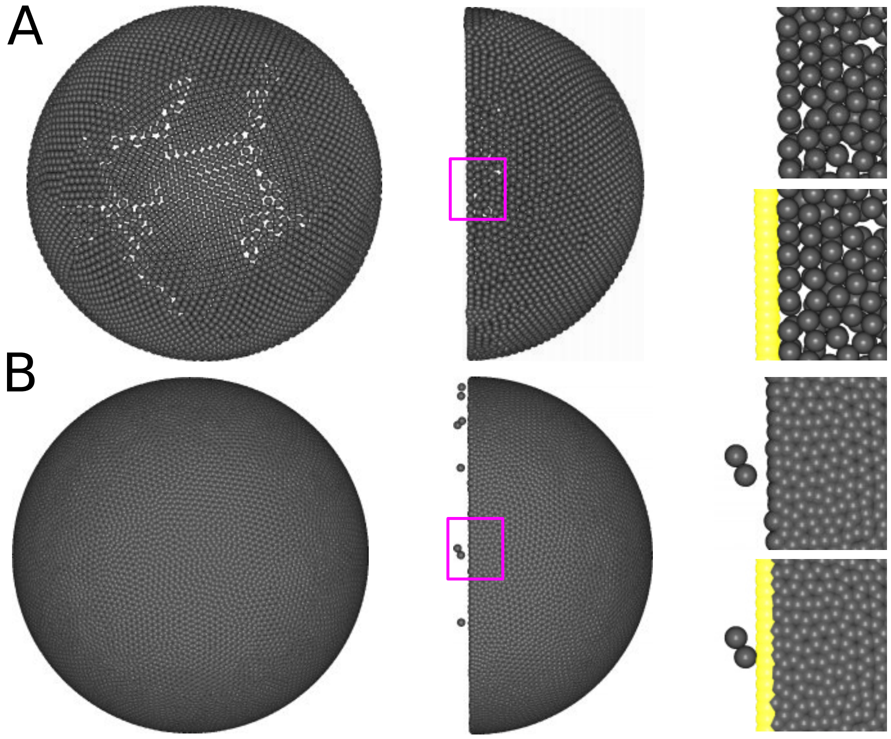


Figure 2.9: Mismatch of cell production rate and eye area growth rate impacted on cell density. **A** Screenshots showing two different views of example simulation where cell production rate was lower than eye area growth rate; $t_{\text{cellCycle}} = 100$ h, other parameters were as listed in Table 2.1. Area boxed in magenta was magnified on the right, illustrating loose cell packing. Lower right panel depicts same magnification with obstacle cells visualised. **B** Screenshots showing two different views of example simulation where cell production rate was higher than eye area growth rate; $t_{\text{cellCycle}} = 5$ h, other parameters were as listed in Table 2.1. Area boxed in magenta was magnified on the right, illustrating tight cell packing and two cells that squeezed through the obstacle cells. Lower right panel depicts same magnification with obstacle cells visualised.

2.1.6 Cell and tissue growth rates must match in homeostasis

One important limitation of the initial model was its low robustness with respect to different parametrisations for the probability of division p_{div}, the minimum cell cycle $t_{\text{cellCycle}}$, and the eye growth rate c_{R}. The parametrisation of p_{div} and $t_{\text{cellCycle}}$ affected cell proliferation, and thus the rate of change in the area taken up by cells A_{cells}, while different values of c_{R} affected the area of the tissue A_{eye}.

Since these model parameters were not coupled by any feedback loops, the growth rate of the eye could exceed cell production rate, resulting in few cells dispersed over a large surface as $A_{\text{cells}} < A_{\text{eye}}$ (Figure 2.9 A). In the opposite case where cell production exceeded organ growth rate, cells became packed to a physically implausible degree since $A_{\text{cells}} > A_{\text{eye}}$ (Figure 2.9 B). As cell density and thus inter-cell forces increased, some cells escaped through the layer of obstacle cells and proliferated on the unused half of the eye globe (Figure 2.9 B). During homeostatic growth of the medaka eye, only $A_{\text{cells}} \approx A_{\text{eye}}$ was a realistic outcome.

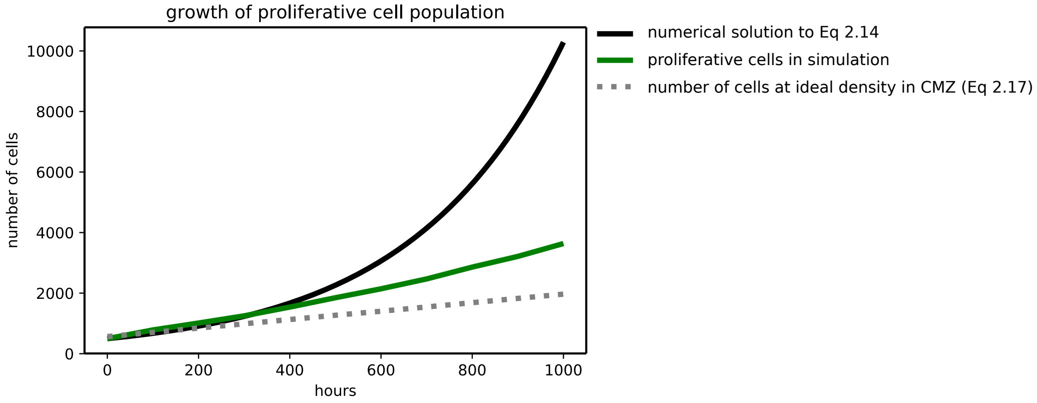


Figure 2.10: The proliferative cell population grows linearly in the simulation.
Plot of numerical solution to Eq 2.14 (solid black line), and simulation data of number of proliferative cells (solid green line) and number of cells that would fit into the simulated CMZ if they were ideally packed (according to Eq 2.17; gray dotted line). Note how the given simulation parameters result in excessively dense packing of proliferative cells. The simulation used the following parameters: $p_{\text{div}} = 30^{-1}$ h^{-1} , $t_{\text{cellCycle}} = 24$ h, $c_R = 0.25$ µmh^{-1}. Other parameters were as listed in Table 2.1. The following parameters were used for the numerical solution to Eq 2.14: $p = 30^{-1}$ h^{-1} and $d = 33^{-1}$ h^{-1}. Eq 2.17 was calculated using the eye radius in the simulation.

Total area occupied by retinal cells and organ tissue area grow quadratically

To better understand the relationship between cell proliferation and eye surface area growth in the model, I considered the theoretical growth rates that could be attained. As the radius R grew linearly (Equation 2.7), the area of the tissue grew quadratically (area of a hemisphere: $A_{\text{eye}} = 2\pi R^2$). The rate of change in proliferating cells P over time can be written in ODE form as

$$\frac{\mathrm{d}P}{\mathrm{d}t} = pP - dP, \tag{2.14}$$

where p is the rate of proliferation that resulted from the combination of p_{div} and $t_{\text{cellCycle}}$, and d is the rate of differentiation that emerged from the simulation. This ODE can be solved by separation of variables

$$\int_0^{\inf} \frac{1}{P}\,\mathrm{d}P = \int_0^{\inf} (p-d)\,\mathrm{d}t$$

$$\ln(P) = (p-d)t + k$$

$$P = K\mathrm{e}^{(p-d)t},$$

where k and K are constants. Using the fact that $P(0) = P_0$

$$P(t) = P_0\mathrm{e}^{(p-d)t}. \tag{2.15}$$

Thus, given $d < p$, the number of proliferative cells should grow exponentially with time and should eventually overtake the quadratic growth rate of the hemispherical surface, no matter which value is chosen for the parameters p_{div} and $t_{\mathrm{cellCycle}}$. However, the simulation showed otherwise (Figure 2.10; compare solid black and solid green lines). This discrepancy arose because the rate of differentiation d was not constant, but actually depended on the total number of cells and the radius of the hemisphere. More specifically, the area of the CMZ placed an upper bound to the number of proliferative cells that could exist in the system. Geometrically, the CMZ corresponded to a spherical zone with area

$$A_{\mathrm{CMZ}} = 2\pi R\,w. \tag{2.16}$$

Interestingly, the area of the CMZ – and therefore the maximum number of proliferative cells – grew linearly as the radius of the hemisphere increased. The actual value of the upper bound of proliferative cells depended on the cells' radius as well as the tolerated cell overlap

$$N_{\mathrm{CMZ}} = \frac{2\pi R\,w}{\pi(r\delta_{\mathrm{ol}_{\mathrm{max}}})^2} = \frac{2R\,w}{(r\delta_{\mathrm{ol}_{\mathrm{max}}})^2}. \tag{2.17}$$

Since cell overlap was dynamic in the simulation, Equation 2.17 is an approximation; depending on parameters, proliferative cells may be more or less densely packed (Figure 2.10, compare dotted grey line with solid green line). Nevertheless, growth of the proliferative population was approximately linear.

Thus, the proliferating cell population tended to grow exponentially, but was limited to grow linearly by the extent of the CMZ. Any "excess" proliferative cells became differentiated cells. As the cell population grew linearly, A_{cells} grew quadratically since every cell occupied an area $\propto r^2$. Similarly, the radius R of the eye grew linearly *in vivo*, which resulted in quadratic growth as $A_{\mathrm{eye}} \propto R^2$.

Summary

The model developed in this section showed that stochastic neutral drift was not necessarily mutually exclusive with clonal growth in ArCoS, laying the foundation for quantitative comparisons with experimental data. Modelling the tissue as a single layer of cells reduced the number of free parameters, enabling to focus analysis on SC proliferation. The simulation highlighted that the parametrisation of tissue area growth rate and cell proliferation rate played a major role in maintaining adequate cell density.

2.2 Uncovering modes of intra-organ growth coordination of NR and RPE

In this section, I address the sensitivity of the model to parametrisation of cell proliferation and eye growth rates by implementing feedback loops between these processes (Section 2.2.1). After establishing appropriate parameter values and evaluating parameter sensitivity for this model extension, I quantitatively compare simulation results with clonal data from the NR and RPE to answer how these tissues coordinate their growth rates (Sections 2.2.2 and 2.2.3).

The text in Sections 2.2.2 and 2.2.3 has been adapted from Tsingos et al. [2019], the text of which was originally written in its entirety by myself, and was edited and corrected by all the authors of Tsingos et al. [2019]. Where appropriate, subject was changed from third to first person, and verb from present to past tense. Some paragraphs were expanded to explain the content in greater detail.

2.2.1 Implementing different growth modes

Since the eye radius grew linearly and the extent of the CMZ imposed a linear growth constraint on the cell population (Section 2.1.6), the parameters affecting tissue area and cell-occupied area could be carefully selected to result in growth at equal quadratic rates. The eye consists of several concertedly growing tissues whose cells derive from different sources. Therefore growth parameters in all tissues must be matched – including in tissues where the proliferative cell population is not restricted to a linearly growing domain (*e.g.* the choroid pigment cells, which proliferate ubiquitously [Appendix Section 5.6.3]). Though matching parameter values for all growth and cellular proliferation rates could in principle have been selected by evolution, this situation is unlikely in the face of variability in growth rates in different individuals (Equations 2.8 and 2.9), and in the same individual over time [own unpublished observations].

As described in Tsingos et al. [2019] (emphasis added; figure references adapted): "Conceptually, [feedback regulating growth rates] between tissues in an organ can be wired in two fundamental ways: Either the tissue of interests acts upstream to induce growth of other tissues (Figure 2.11 A; **"inducer growth mode"**), or, vice versa, the tissue of interest lies downstream of growth cues from another tissue in the organ (Figure 2.11 B; **"responder growth mode"**). Possible biological mechanisms for these growth modes could be mechanical, biochemical, or a combination of both. For example, in the inducer growth mode

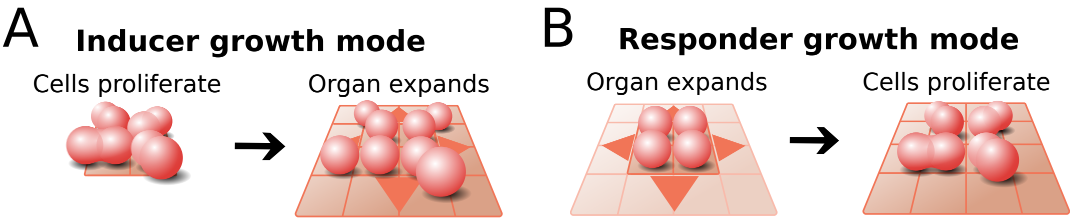


Figure 2.11: Conceptual modes of feedback coupling between a tissue and the rest of the organ. **A** In the inducer growth mode, the tissue of interest acts upstream to induce growth of other tissues. **B** In the responder growth mode, the tissue of interest responds downstream to an external growth-inducing stimulus. Figure has been adapted from Tsingos et al. [2019].

cells could instruct organ growth by modifying the extracellular matrix or by paracrine signalling. These stimuli instruct tissues with the responder growth mode to grow, *e.g.* by alleviating contact inhibition or by providing permissive proliferation signals [Buchmann et al., 2014; Droujinine and Perrimon, 2016]."

In an organ composed of multiple tissues, one tissue may act as a "coordinator" and induce growth in nearby tissues following the responder growth mode. Another possibility is that signals external to the organ concurrently modulate growth of all its constituent tissues, which therefore all grow akin to the responder growth mode. Transplantation experiments between different species demonstrated that the eye grows autonomously [Twitty and Schwind, 1931], indicating that at least one tissue in the eye must follow an inducer growth mode. Since the mode of growth for the NR and RPE were not known, I implemented both growth modes into the model and compared the results of simulations of both growth modes to experimental data (described in section 2.2.3).

Based on experimentally observed contact inhibition, cell centred models often include density-dependent arrest of cell proliferation to prevent implausibly high cell densities [Osborne et al., 2017; Pathmanathan et al., 2009; Sütterlin et al., 2017]. I chose a similar mechanism to achieve feedback between tissue growth and cell proliferation rate. Thus, I introduced the parameter $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$, *i.e.* a threshold level of overlap that induces cell cycle arrest when exceeded, normalised to the interval $[0,1]$. Cell division was inhibited in cells whose average overlap with all neighbours n exceeded a fraction of the cell's diameter

$$\frac{1}{n}\sum_{i=1}^{n} \mathrm{d}(\mathbf{r}_c, \mathbf{r}_{n_i}) > \delta_{\mathrm{ol}_{\mathrm{threshold}}} 2r, \tag{2.18}$$

where $\mathrm{d}(\mathbf{r}_c, \mathbf{r}_{n_i})$ is the overlap between cell c and its ith neighbour cell n_i.

Given Equation 2.18, implementing the responder growth mode was straightforward:

1. The growth equation for the eye radius stayed as Equation 2.7:

$$R(t) = R_{\text{init}} + c_{\text{R}}\, t.$$

2. The cell proliferation parameters p_{div} and $t_{\text{cellCycle}}$ had to be chosen such that the proliferation rate was equal to or greater than the maximal eye surface growth given by the radius growth rate observed *in vivo* (Equation 2.9; $c_{\text{R}} \approx 0.49\,\mu\text{mh}^{-1}$). This was necessary to ensure that cell density didn't decrease over time (as in Figure 2.9 A).
3. The overlap threshold $\delta_{\text{ol}_{\text{threshold}}}$ had to be chosen such that cells experiencing density higher than the homeostatic level were inhibited. This mechanism resulted in a self-regulation to a constant cell density, and therefore total cell-occupied area corresponded to the tissue surface area:

$$A_{\text{cells}} \approx A_{\text{eye}}$$

$$N_{\text{cells}}(t)\pi\left(r\,\delta_{\text{ol}_{\text{max}}}\right)^2 \approx 2\pi R^2, \tag{2.19}$$

where $N_{\text{cells}}(t)$ is the total number of cells at simulation step t that emerges from the simulation.

For implementing the inducer growth mode, the following must hold:

1. Instead of Equation 2.7, a growth equation proportional to the number of cells is formulated for the eye radius:

$$R(t) = \sqrt{\frac{N_{\text{cells}}(t)\left(r\,\delta_{\text{ol}_{\text{max}}}\right)^2}{2}}, \tag{2.20}$$

Equation 2.20 was derived from Equation 2.19. The cell population cannot decrease in the model, so the eye radius cannot shrink.
2. The cell proliferation parameters p_{div} and $t_{\text{cellCycle}}$ had to be chosen such that the proliferation rate resulted in eye radius growth rates within the measured range (Equations 2.8 and 2.9; $\frac{\mathrm{d}R}{\mathrm{d}t} \approx 0.23\text{–}0.49\,\mu\text{m}\,\text{h}^{-1}$).
3. "The overlap threshold $\delta_{\text{ol}_{\text{threshold}}}$ had to be chosen such that cell division was minimally inhibited at homeostatic cell density (otherwise cells would never proliferate and the tissue would never grow)" [Tsingos et al., 2019].

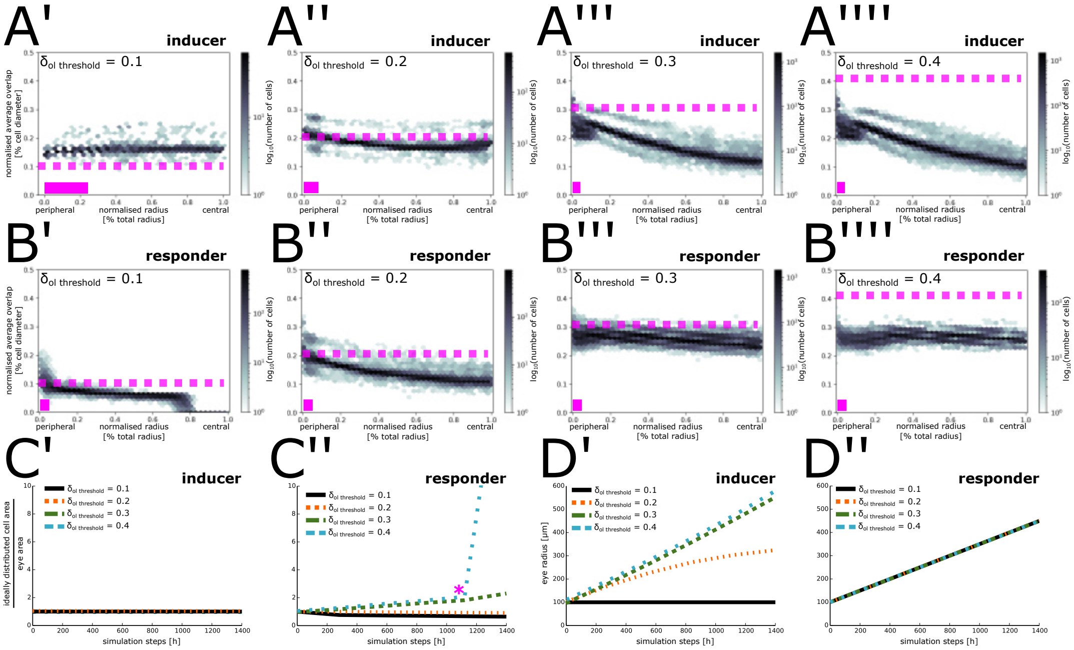


Figure 2.12: Parameter scan to determine optimal value for $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$.
A′–B′′′′ Parameter scan of $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$. Heatmaps of normalised average overlap against normalised position along the retinal radius for each cell in the simulation. Dotted magenta line: $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$. Solid magenta line: extent of the simulated CMZ along the normalised radius; the relative CMZ extent differs as the final radius of the simulated eye differed between conditions. A′-A′′′′ Inducer growth mode. B′-B′′′′ Responder growth mode. **C′–C′′** Area occupied by the sum of all cells assuming optimal density divided by the total eye area plotted against simulation time at different values of $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$. C′ Inducer growth mode. C′′ responder growth mode. Magenta asterisk denotes timepoint when cells escape through obstacle cell layer. **D′–D′′** Eye radius plotted against simulation time at different values of $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$. D′ Inducer growth mode. D′′ responder growth mode. The figure has been adapted from Tsingos et al. [2019]. Parameters as in Table 2.1.

Parametrisation of responder and inducer growth modes

To determine appropriate values for $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$ I performed a parameter scan (Figure 2.12). As described in Tsingos et al. [2019] (figure and equation references adapted; verbs were changed from present to past tense): "In the responder growth mode, the radius of the hemisphere steadily grew regardless of the number of cells in the simulation (Figure 2.12 D′′). A value of $\delta_{\mathrm{ol}_{\mathrm{threshold}}} = 0.1$ strongly inhibited cell proliferation, but as the radius grew cells became dispersed and eventually went under the threshold allowing some proliferation (Figure 2.12 B′). However, the ratio between A_{cells} and A_{eye} (area ratio) steadily decreased indicating the formation of inter-cell gaps (Figure 2.12 C′′, solid black line). At $\delta_{\mathrm{ol}_{\mathrm{threshold}}} = 0.2$, many, but not all, cells were inhibited (Figure 2.12 B′′), and the area ratio was near 1 throughout the simulation (Figure 2.12 C′′, dotted orange line). At $\delta_{\mathrm{ol}_{\mathrm{threshold}}} = 0.3$ and $\delta_{\mathrm{ol}_{\mathrm{threshold}}} = 0.4$, cell proliferation overtook area growth, resulting in high cell packing all over the hemisphere (Figure 2.12 B′′–B′′′′). As a result, the area ratio increased over time, until cell packing became so severe that cells escaped through the obstacle cell layer and proliferated exponentially on the unused half of the eye globe (Figure 2.12 C″, dashed green and cyan lines). Given these data, a value of $\delta_{\text{ol}_{\text{threshold}}} = 0.2$ best generated an even distribution of cells on the hemisphere for the full duration of the simulation.

In the inducer growth mode, $\delta_{\text{ol}_{\text{threshold}}}$ depended on the total number of cells in the simulation (Equation 2.20). Thus, growth rate increased as the overlap threshold was increased (Figure 2.12 D′). $\delta_{\text{ol}_{\text{threshold}}} = 0.1$ completely inhibited cell proliferation, as the equilibrium average overlap normalised to the cell diameter (normalised average overlap) exceeded the threshold value (Figure 2.12 A′). As a result, the organ did not grow at all (Figure 2.12 D′, solid black line). At $\delta_{\text{ol}_{\text{threshold}}} = 0.2$, a large population of cells in the proliferative domain exceeded the threshold (Figure 2.12 A″). Generation of new cells through division increased the local cell density, resulting in inhibition of proliferation due to the low overlap threshold and a gradual reduction in growth rate Figure 2.12 D′ dotted orange line). At $\delta_{\text{ol}_{\text{threshold}}} = 0.3$, only few cells exceeded the threshold (Figure 2.12 A‴), and growth was almost unconstrained (Figure 2.12 D′ dashed green line). At $\delta_{\text{ol}_{\text{threshold}}} = 0.4$, no cells exceeded the threshold and growth was completely unconstrained (Figure 2.12 A⁗, D′ dashed cyan line). At all values of $\delta_{\text{ol}_{\text{threshold}}}$, the area ratio was equal to 1 throughout the simulation, meaning that – on average – cells were evenly distributed and ideally packed (Figure 2.12 C′), as intended by Equation 2.20." Thus, a value of $\delta_{\text{ol}_{\text{threshold}}} = 0.4$ was used for all simulations of the inducer growth mode.

Table 2.2: Parameters for inducer and responder growth modes in addition to Table 2.1.

Description	Parameter	Value	Reference/Explanation
		Simulation parameters	
Overlap threshold beyond which cell cycle is arrested	$\delta_{\text{ol}_{\text{threshold}}}$	0.4	**Value for inducer growth mode.** Estimated from parameter scan to minimize density-dependent cell cycle arrest.
		0.2	**Value for responder growth mode.** Estimated from parameter scan to maximize density-dependent cell cycle arrest without completely suppressing cell division.

Sensitivity of responder and inducer growth modes to parameter values

Experimental data indicated a range of plausible radius growth rates (Equations 2.8 and 2.9). To determine which combinations of $t_{\text{cellCycle}}$ and p_{div} in the inducer growth mode lied within this range, I performed a parameter scan (Figure 2.13). Since in the responder growth mode cell proliferation rate must exceed tissue area growth rate, the parameter space in Figure 2.13 could also be used to infer

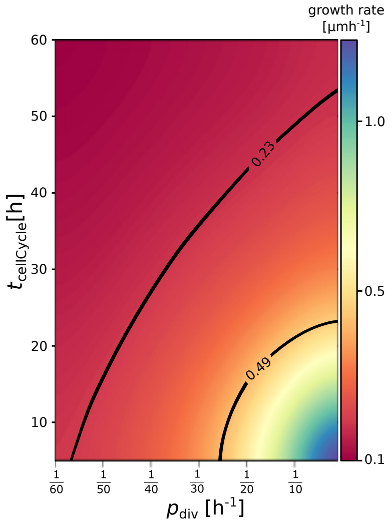


Figure 2.13: Parameter space of simulated growth rate in the inducer growth mode. p_{div} and $t_{\mathrm{cellCycle}}$ were varied in the inducer growth mode and the eye radius growth rate calculated from the simulation. The results were interpolated and smoothed to generate the plot. The range enclosed by Equations 2.8 and 2.9 is marked with solid black outline. Parameters not listed in the figure were as in Tables 2.1 and 2.2. Figure adapted from Tsingos et al. [2019].

values for $t_{\mathrm{cellCycle}}$ and p_{div} in the responder growth mode.

I qualitatively evaluated how clones varied in the responder growth mode by changing $t_{\mathrm{cellCycle}}$, p_{div}, and c_{R}; and in the inducer mode by changing $t_{\mathrm{cellCycle}}$ and p_{div}. As expected, in the responder growth mode, any combination of parameters where $A_{\mathrm{cells}} < A_{\mathrm{eye}}$ resulted in cell dispersal and could therefore be regarded as irrelevant for characterising the *in vivo* system (Figure 2.14 A V, B IV–V, C I). When $A_{\mathrm{cells}} \approx A_{\mathrm{eye}}$, clones formed distinct sectors (Figure 2.14 A IV, C II). When $A_{\mathrm{cells}} > A_{\mathrm{eye}}$, sector boundaries became less clear as clones fragmented and intermingled (Figure 2.14 A I–III, B I–III, C III–V), and this effect became more pronounced as $A_{\mathrm{cells}} \gg A_{\mathrm{eye}}$ (Figure 2.14 A I, B I, C IV).

The inducer growth mode showed more robust behaviour, as all combinations of parameters resulted in clear clone sectoring (Figure 2.15 A I–B V). Variations of $t_{\mathrm{cellCycle}}$ and p_{div} weakly impacted on clone width: As p_{div} decreased, the average cell cycle length increased, and this lead to greater differences in clonal sector width (Figure 2.15 B I). These differences could be explained by chance effects during early clone expansion, whereby one cell by chance proliferated sooner, giving its clone an early expansion advantage. The parameter range where these effects became more prominent was outside of the plausible range established by experimental observations (compare Figure 2.13).

With the exception of the responder growth mode when $A_{\mathrm{cells}} \approx A_{\mathrm{eye}}$, clones in both growth modes displayed qualitative differences in appearance even with equal parametrisation. In the responder growth mode, the virtual clones fre-

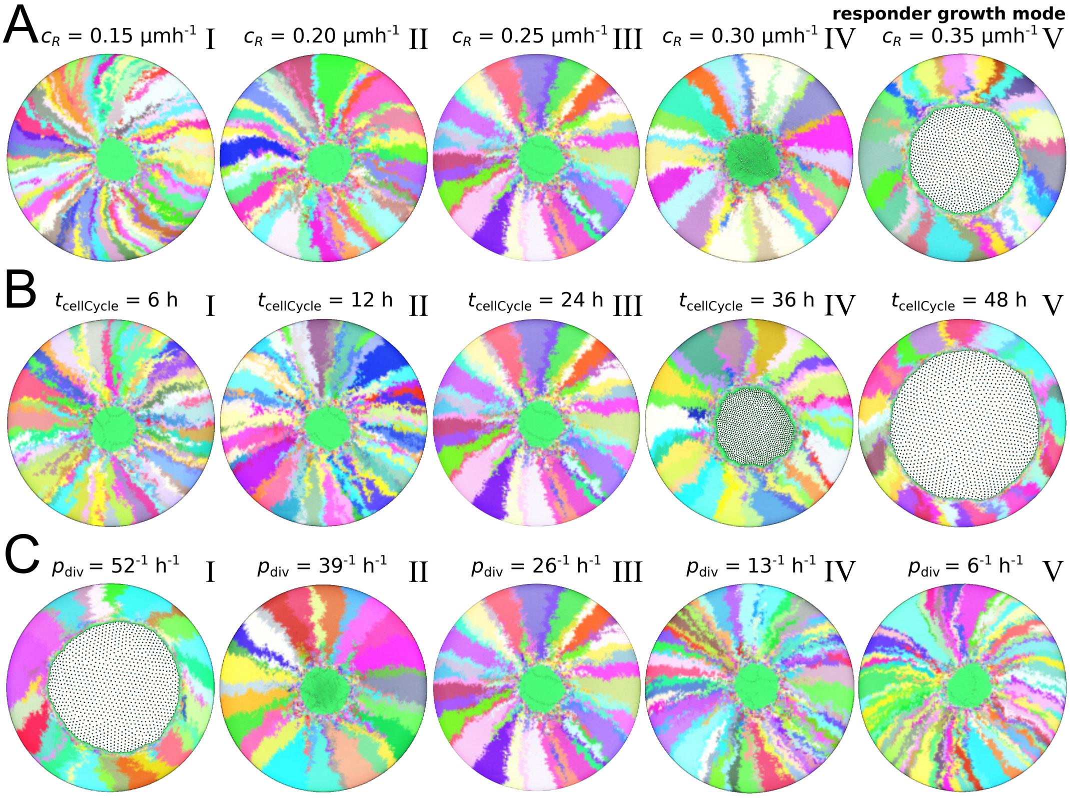

responder growth mode
A
c_R = 0.15 µmh^{-1} I
c_R = 0.20 µmh^{-1} II
c_R = 0.25 µmh^{-1} III
c_R = 0.30 µmh^{-1} IV
c_R = 0.35 µmh^{-1} V
B
$t_{\text{cellCycle}}$ = 6 h I
$t_{\text{cellCycle}}$ = 12 h II
$t_{\text{cellCycle}}$ = 24 h III
$t_{\text{cellCycle}}$ = 36 h IV
$t_{\text{cellCycle}}$ = 48 h V
C
p_{div} = 52^{-1} h^{-1} I
p_{div} = 39^{-1} h^{-1} II
p_{div} = 26^{-1} h^{-1} III
p_{div} = 13^{-1} h^{-1} IV
p_{div} = 6^{-1} h^{-1} V

Figure 2.14: Parameter sensitivity of the responder growth mode.

A Representative simulation screenshots of clonal growth in the responder growth mode with constant p_{div} and $t_{\text{cellCycle}}$ with varying c_R. **B** Representative simulation screenshots of clonal growth in the responder growth mode with constant p_{div} and c_R with varying $t_{\text{cellCycle}}$. **C** Representative simulation screenshots of clonal growth in the responder growth mode with constant $t_{\text{cellCycle}}$ and c_R with varying p_{div}. Screenshots in A III, B III, C III are from the same simulation. Other parameters as in Tables 2.1 and 2.2.

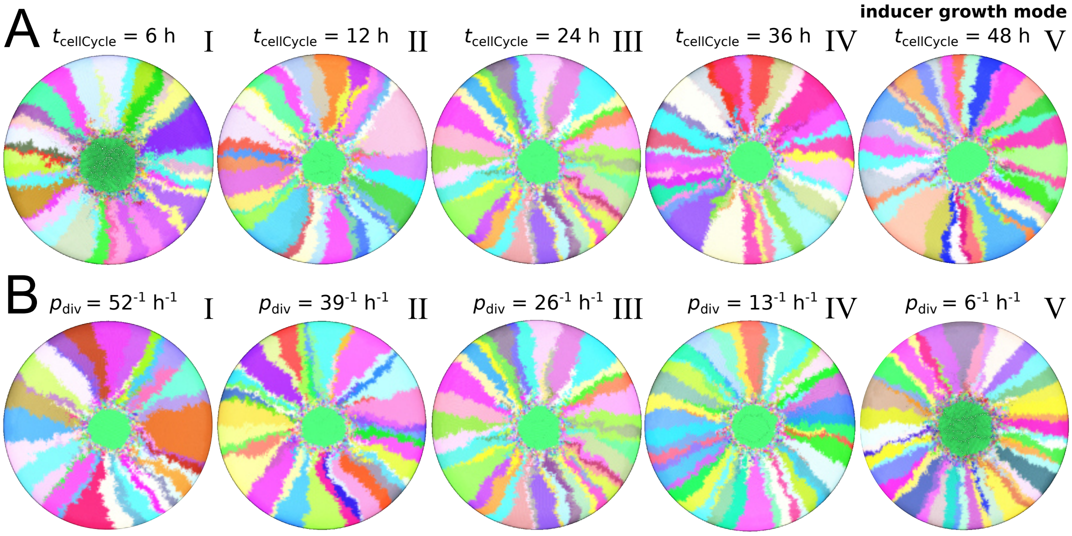

inducer growth mode
A
$t_{\text{cellCycle}}$ = 6 h I
$t_{\text{cellCycle}}$ = 12 h II
$t_{\text{cellCycle}}$ = 24 h III
$t_{\text{cellCycle}}$ = 36 h IV
$t_{\text{cellCycle}}$ = 48 h V
B
p_{div} = 52^{-1} h^{-1} I
p_{div} = 39^{-1} h^{-1} II
p_{div} = 26^{-1} h^{-1} III
p_{div} = 13^{-1} h^{-1} IV
p_{div} = 6^{-1} h^{-1} V

Figure 2.15: Parameter sensitivity of the inducer growth mode.

A Representative simulation screenshots of clonal growth in the inducer growth mode with constant p_{div} and varying $t_{\text{cellCycle}}$. **B** Representative simulation screenshots of clonal growth in the inducer growth mode with constant $t_{\text{cellCycle}}$ and varying p_{div}. Screenshots in A III, B III are from the same simulation. Other parameters as in Tables 2.1 and 2.2.

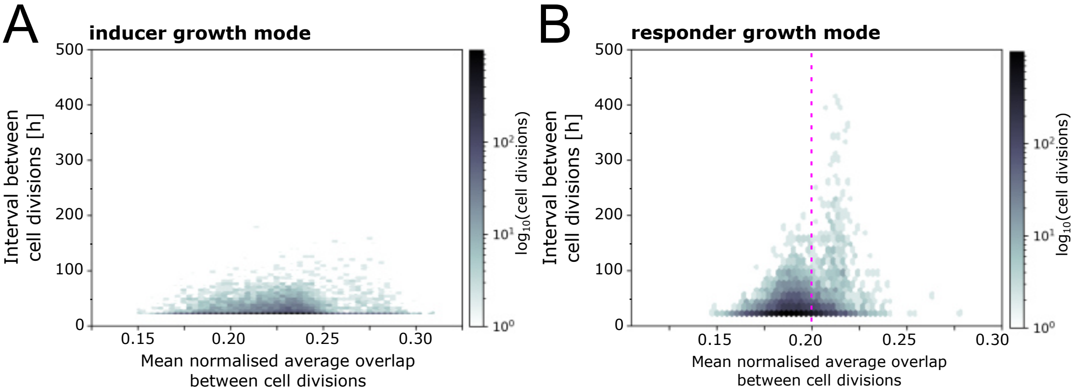


Figure 2.16: 2D histogram of cell division intervals and normalised average overlap
A Data from one representative simulation of the inducer growth mode run for 1000 simulation steps; n = 16940 cell division events. **B** Data from one representative simulation of the responder growth mode run for 1000 simulation steps; n = 10430 cell division events. Pink dashed line indicates overlap threshold value. The figure has been adapted from Tsingos et al. [2019]. All parameters as in Tables 2.1 and 2.2.

quently intermingled and broke up into smaller clusters, while the sectoring pattern was more apparent in the inducer growth mode. This difference in the growth modes was due to variation in cell division timing. Using the parametrisation in Tables 2.1 and 2.2, a representative simulation in the inducer growth mode resulted in cell divisions on average every 28.5 h with a standard deviation of 10.9 h; the maximum interval between cell divisions was 178.0 h (Figure 2.16 A). In contrast, a representative responder growth mode simulation resulted in average cell division interval of 33.9 h with standard deviation of 25.0 h, and a maximum interval of 601.0 h (Figure 2.16 B). Local competition for space increased cell division intervals, particularly among cells exceeding the tolerated overlap threshold (Figure 2.16 B, pink dashed line). Thus, the model predicted different variation in cell division timing in retinal tissues following the inducer or responder growth modes.

2.2.2 Generation of clonal data in experiment and simulation

Clonal data in the NR were generated by Prof Dr Lázaro Centanin and Dr Burkhard Höckendorf by randomly labelling individual NR SCs in hatchling medaka of the *rx2*::^{ERT2}*cre, Gaudí*$^{2.1}$ line, and analysing the eyes in adult fish as previously described [Centanin et al., 2014; Reinhardt et al., 2015]. The retinal homeobox transcription factor 2 gene (*rx2*) promoter drives *cre-ert2* in SCs at the very periphery of the CMZ [Reinhardt et al., 2015]. A recombined SC generated a stripe of green fluorescent protein (GFP)-positive progeny in an otherwise GFP-negative retina [Centanin et al., 2014]. In proximal view, NR ArCoS emanated as

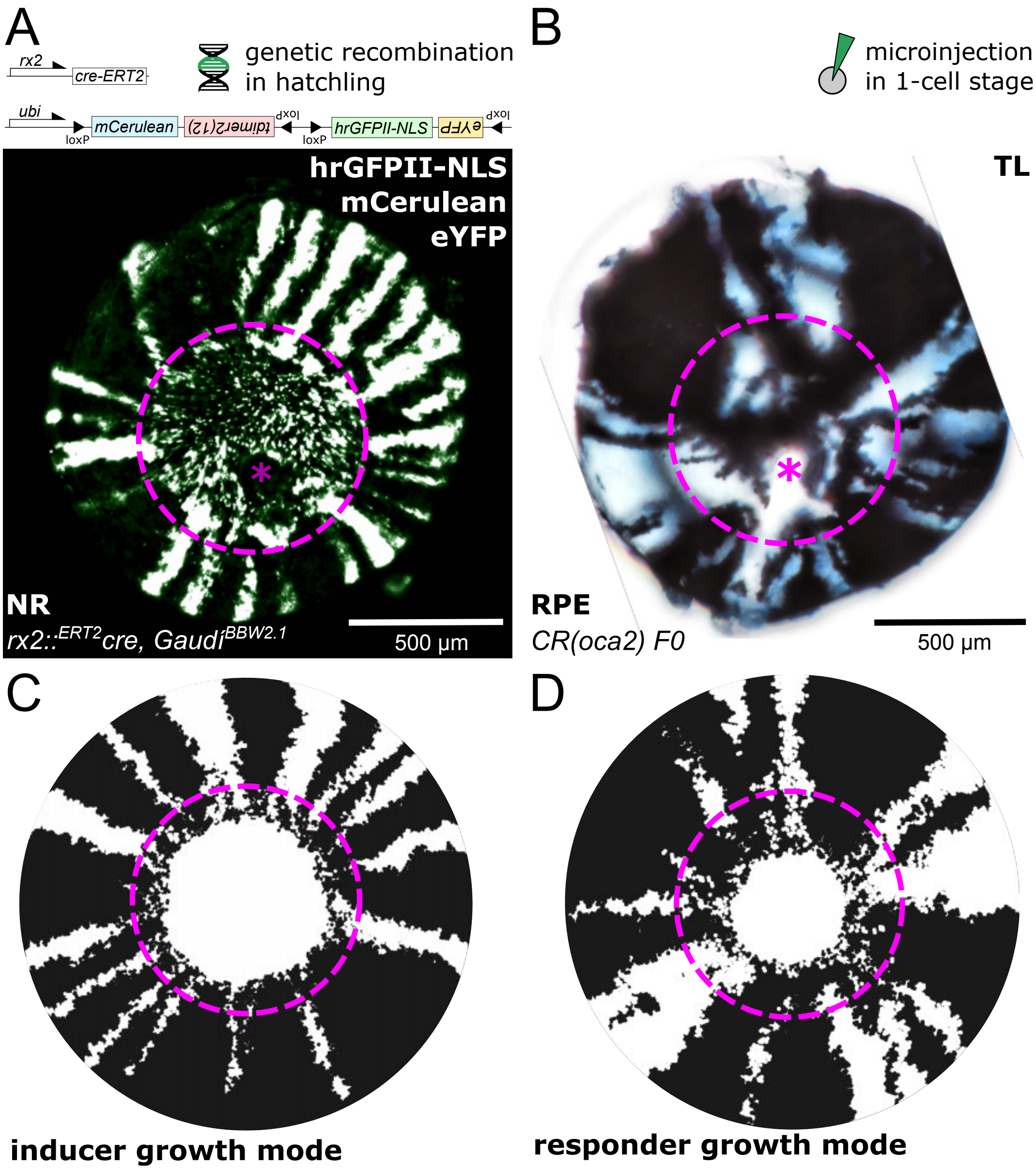


Figure 2.17: Representative examples of experimental and simulated ArCoS.
A Example of an antibody-stained NR with ArCoS; maximum projection of a confocal stack acquired from proximal view. Data provided by Prof Dr Lázaro Centanin and Dr Burkhard Höckendorf. The Gaudí$^{2.1}$ line permits visualisation of four different colours as indicated in the scheme, but the fluorescent proteins cannot be distinguished by antibodies resulting in identical staining. Magenta dashed line encloses pre-induction retina and induction ring. **B** Example of RPE with ArCoS; focused stack of transmitted light images visualising pigmented and unpigmented clones acquired from proximal view. Data obtained in collaboration with Dr Stephan Kirchmaier. Magenta dashed line chosen as the same size in (A); due to earlier induction it also encloses part of the post-induction retina. **C** Example of proximal view on inducer growth mode clones induced at $R = 150$ µm. Magenta dashed line chosen as the same size in (A). **D** Example of proximal view on responder growth mode clones induced at $R = 100$ µm. Magenta dashed line chosen as the same size in (A). Magenta asterisks mark optic nerve exit. Panels A and B were adapted from Tsingos et al. [2019].

rays from the central pre-induction retina (Figure 2.17 A).

Clonal data in the RPE were generated in collaboration with Dr Stephan Kirchmaier by mosaic knockout of pigmentation using clustered regularly interspaced short palindromic repeats (CRISPR)/CRISPR-associated system 9 (Cas9) targeted to the oculo-cutaneous albinism 2 gene (*oca2*), which is required for melanosome maturation [Fukamachi et al., 2004; Lischik et al., 2019]. RPE SCs with a bi-allelic mutation in *oca2* generated unpigmented stripes, akin to RPE ArCoS obtained by transplantation [Centanin et al., 2011]. RPE ArCoS frequently branched, forming irregular stripes variable in size and shape (Figure 2.17 B). These qualitative observations suggested that the RPE followed a responder growth mode where $A_{\text{cells}} > A_{\text{eye}}$, while the low variability in clone width and clearer sectoring pattern of the NR resembled the inducer growth mode.

For generating simulated clonal data, all proliferating cells in the model received a unique identifier when the eye reached a given radius, and this identifier was stored in all future daughter cells. To mimic the experiment, simulated clones were randomly sampled from the full population and plotted as a 2D projection akin to the proximal view in experimental data. For simulating NR clones, the virtual clonal induction was done at a retinal radius of 150 µm (Figure 2.17 C). This value was chosen based on the estimated radius of the NR when genetic recombination was induced *in vivo*. To replicate RPE clones, the virtual labelling experiment began at a radius of 100 µm, since the experimental mosaic knockout happened at an earlier timepoint in development (Figure 2.17 D).

For comparability, identical parameters were used for responder and inducer growth modes (Tables 2.1 and 2.2). The parameters for cell proliferation were in the middle of the plausible parameter space and resulted in a radius growth rate of $\approx 0.35\,\mu\text{mh}^{-1}$ (Figure 2.13). As discussed in section 2.2.1, the cell proliferation parameters of the responder growth mode must be set such that $A_{\text{cells}} \geq A_{\text{eye}}$. To ensure this condition was fulfilled despite using identical parameters to the inducer growth mode, the growth rate of the eye radius was set to $c_{\text{R}} = 0.25\,\mu\text{m}\,\text{h}^{-1}$.

2.2.3 NR and RPE match inducer and responder growth modes

I hypothesized that the RPE followed a responder growth mode and the NR an inducer growth mode. To test this hypothesis, I developed several quantitative metrics to compare clone topology, which could be used to infer differential division behaviour as predicted by the model. Since the position of cells in the retina reflects their birth order [Centanin et al., 2014, 2011], in the extreme case of no variation in cell division timing, each clone would form a continuous, un-

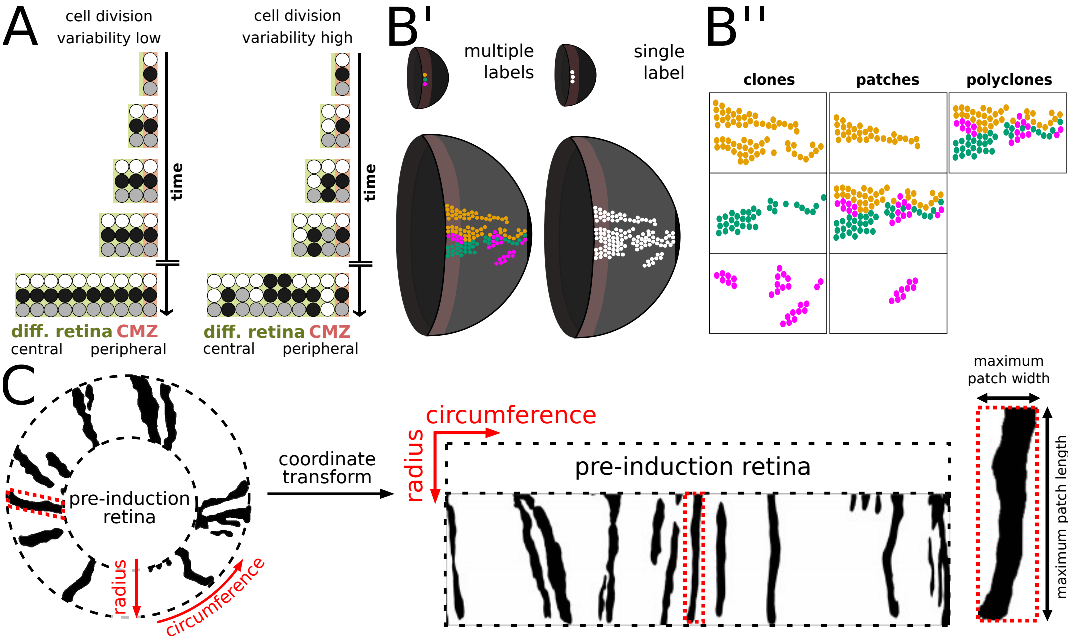
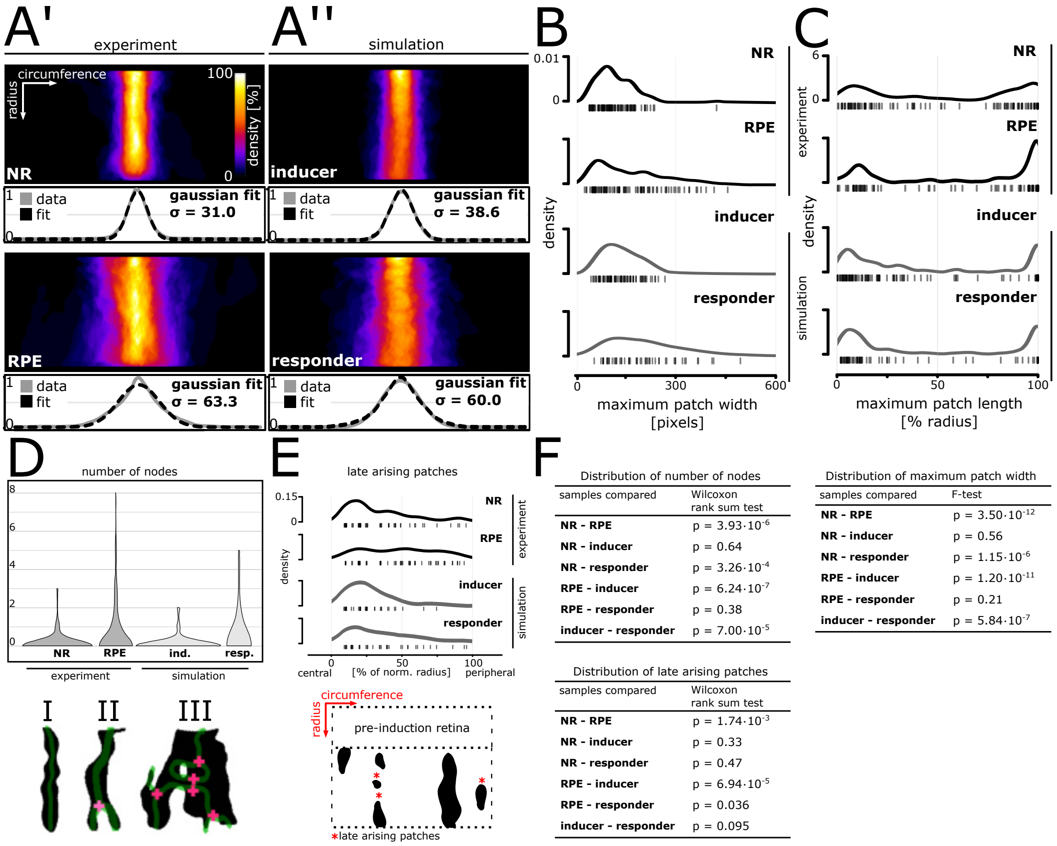


Figure 2.18: Concepts used in the quantitative analysis of clonal data.
A Schematic representation depicting how variability in the length of cell division intervals impacts on the clonal shape. **B′** Scheme comparing multiple unique labels to a single label in a thought experiment where three adjacent SCs were clonally labelled. Individual clones cannot be distinguished in the case of single label. **B″** Individual clones from panel (B') are highlighted and contrasted to patches (contiguous domains of labelled cells), and polyclones (conglomerates of clones). **C** Illustration of the unrolling transform used to project ArCoS onto a rectangular plane. Figure adapted from Tsingos et al. [2019].

branching stripe (Figure 2.18 A, left scenario). In the opposite highly variable case, clones would frequently branch or merge into polyclones, as well as fragment into several small patches (Figure 2.18 A, right scenario). Thus, with increasing variation in cell division timing, I expected an increasing variation in clone width, and an increasing incidence of clone branching and fragmentation.

I compared simulated clones of the inducer and responder growth modes to clones in the NR and RPE. To circumvent biases associated with fusion and fragmentation of clones, I analysed "patches", *i.e.* contiguous domains of segmented pixels. A patch might entail a (sub-)clone, or multiple clones (*i.e.* a polyclone) (Figure 2.18 B′–B″). To assay the experimental and simulated data, I unrolled the retina with a coordinate transform (Figure 2.18 C) and quantified three different metrics: patch width variance, branching, and fragmentation.

To assay patch width variance, I aligned and superimposed all patches (Figure 2.19 A′–A″), and quantified the distribution of maximum patch width (Figure 2.19 B). NR patches formed a narrow stripe, while the width of RPE patches showed much greater variation (Figure 2.19 A′). The variance of NR and RPE patches was significantly different at the 0.05 level ($p = 3.50 \cdot 10^{-12}$, F-test of equality of variance). In striking agreement to the experimental data, simulated



Distribution of number of nodes

samples compared	Wilcoxon rank sum test
NR - RPE	$p = 3.93 \cdot 10^{-6}$
NR - inducer	$p = 0.64$
NR - responder	$p = 3.26 \cdot 10^{-4}$
RPE - inducer	$p = 6.24 \cdot 10^{-7}$
RPE - responder	$p = 0.38$
inducer - responder	$p = 7.00 \cdot 10^{-5}$

Distribution of maximum patch width

samples compared	F-test
NR - RPE	$p = 3.50 \cdot 10^{-12}$
NR - inducer	$p = 0.56$
NR - responder	$p = 1.15 \cdot 10^{-6}$
RPE - inducer	$p = 1.20 \cdot 10^{-11}$
RPE - responder	$p = 0.21$
inducer - responder	$p = 5.84 \cdot 10^{-7}$

Distribution of late arising patches

samples compared	Wilcoxon rank sum test
NR - RPE	$p = 1.74 \cdot 10^{-3}$
NR - inducer	$p = 0.33$
NR - responder	$p = 0.47$
RPE - inducer	$p = 6.94 \cdot 10^{-5}$
RPE - responder	$p = 0.036$
inducer - responder	$p = 0.095$

Figure 2.19: Quantitative comparison of experimental and simulated patch properties. **A′–A″** Clone shape superposition. **B** Maximum patch width distribution. **C** Maximum patch length distribution. **D** Patch skeletonisation. Bottom: Examples of experimental patches (black) overlayed by skeletons (dark green) and nodes (pink crosses). **E** Late arising patches. Bottom: Cartoon of transformed retina with late arising patches (red asterisks). Patches very close to the pre-induction retina (within the first 10% of radial extent) were not considered as arising late. **F** Statistics for comparisons in A–E. Figure adapted from Tsingos et al. [2019].

patches in the inducer growth mode had low variation in width, while patches in the responder growth mode spread widely (Figure 2.19 A″). The variances in the simulated conditions were significantly different at the 0.05 level ($p = 5.84 \cdot 10^{-7}$, F-test of equality of variance), but highly similar between NR and inducer ($p = 0.56$, F-test of equality of variance); and RPE and responder ($p = 0.21$, F-test of equality of variance).

To measure branching I skeletonised the patches, and quantified the distribution of nodes per patch and condition (Figure 2.19 D). Patches in the NR and in the inducer growth mode were overwhelmingly stripe-like with no branch points (Figure 2.19 D I), with similar node distribution ($p = 0.64$, Wilcoxon rank sum test). In contrast, both NR and inducer differed significantly at the 0.05 level from the distribution in the RPE and responder growth mode (NR-RPE: $p = 3.93 \cdot 10^{-6}$; NR–responder: $p = 3.26 \cdot 10^{-4}$; inducer–RPE: $p = 6.24 \cdot 10^{-7}$; inducer–responder: $p = 7.00 \cdot 10^{-5}$, Wilcoxon rank sum test). Patches in the RPE and in the responder growth mode frequently bifurcated or merged, creating branching shapes with inclusions indicative of clone intermingling (Figure 2.19 D III). RPE and responder growth mode were highly similar in this metric ($p = 0.38$, Wilcoxon rank sum test).

Not all patches were contiguous with the pre-induction retina. Such "late arising patches" resulted if a cell divided intermittently with periods of quiescence, leaving clone fragments behind (Figure 2.19 E lower panel). I quantified fragmentation by plotting the occurrence of late arising patches along the normalised post-embryonic retinal radius (Figure 2.19 E). In the NR late patches clustered in the central post-embryonic retina and waned thereafter. Thus clone fragments were not equally distributed, consistent with lower levels of cell division variability and a majority of continuous stripe-like clones. In contrast, the RPE displayed an even distribution indicative of frequent fragmentation throughout the life of the animal as predicted for the highly variable scenario (NR–RPE: $p = 1.74 \cdot 10^{-3}$, Wilcoxon rank sum test). The simulated data showed the same tendency, to a lesser degree, as the central peak in late patches was higher in the inducer growth mode and peripheral late patches occurred more frequently in the responder growth mode (Figure 2.19 E; inducer–responder: $p = 0.10$, Wilcoxon rank sum test). In this metric, the RPE stood out from the NR and both simulated conditions (RPE–inducer: $p = 6.94 \cdot 10^{-5}$; RPE–responder: $p = 0.04$, Wilcoxon rank sum test), indicating a high degree of fragmentation and thus cell division variability. Together, these data showed that NR and RPE had different degrees of variability in cell division timing.

Summary

The simulation predicted that different growth modes impacted on the distribution of intervals between subsequent cell divisions, which in turn affected the shape of sub-clonal patches. The NR displayed lower variability in patch shape consistent with the simulated inducer growth mode, while the RPE showed higher levels of variability that even exceeded what I modelled with the responder growth mode. Thus, the data supported a model where NR and RPE concertedly expanded relying on different growth modes, which manifested in differently shaped ArCoS.

2.3 Analysis of proliferative properties of NR SCs

Having established that the NR followed an inducer growth mode, I leveraged the model to explore properties of the NR SC that were not directly accessible for experimentation but could be inferred from clonal topology. In the following, I first examine population dynamics of NR SCs with respect to neutral drift (Sections 2.3.1 and 2.3.2), then I analyse division axis and differential proliferation biases and relate them to potential functions in integrating external and internal stimuli to shape the organ (Sections 2.3.3, 2.3.4, and 2.3.5).

The text in Sections 2.3.1, 2.3.3, and 2.3.4 has been adapted from the main manuscript text of Tsingos et al. [2019], the text of which was originally written in its entirety by myself, and was edited and corrected by all the authors of Tsingos et al. [2019]. Where appropriate, the subject was changed from third to first person, and the verb tense was changed from present to past tense. Some paragraphs were expanded to explain the subject matter in greater detail.

2.3.1 Stem- and progenitor cell domains are an emergent property

As delineated in section 2.1.4, the clonally labelled retina could be subdivided into the pre-induction retina, the induction ring, and the post-induction retina which contained ArCoS. The virtual induction ring contained many few-cell clones unrelated to any ArCoS (Figure 2.20 A′, dashed magenta line). In these clones, all SCs left the niche and thus differentiated ("terminated clones"). Nested inductions showed that sister SCs within one clone segregated into subclones (Figure 2.20 A″). However, only some of these subclones generated virtual ArCoS. Again, terminated clones clustered in the virtual induction ring (Figure 2.20 A″, dashed black line), demonstrating that the pattern repeated itself regardless of the timepoint of virtual induction.

In the model, all proliferative cells were equipotent SCs. Nevertheless, a subset of these virtual SCs proliferated only a few times before terminally differentiating, resulting in a bimodal distribution of patch lengths – which was also observed in the experimental data (Figure 2.19 C). Notably, the overwhelming majority of virtual ArCoS emerged from the periphery of the induction ring (Figure 2.20 B), as confirmed by tracing back the position of the founder SCs at simulation step 0, while centrally located cells formed exclusively terminated clones (Figure 2.20 B). This behaviour was highly reminiscent of retinal PCs *in vivo*, which are believed to reside in the central CMZ [Raymond et al., 2006; Shi et al., 2017]. Importantly, this subdivision was not imposed onto the simulation, but

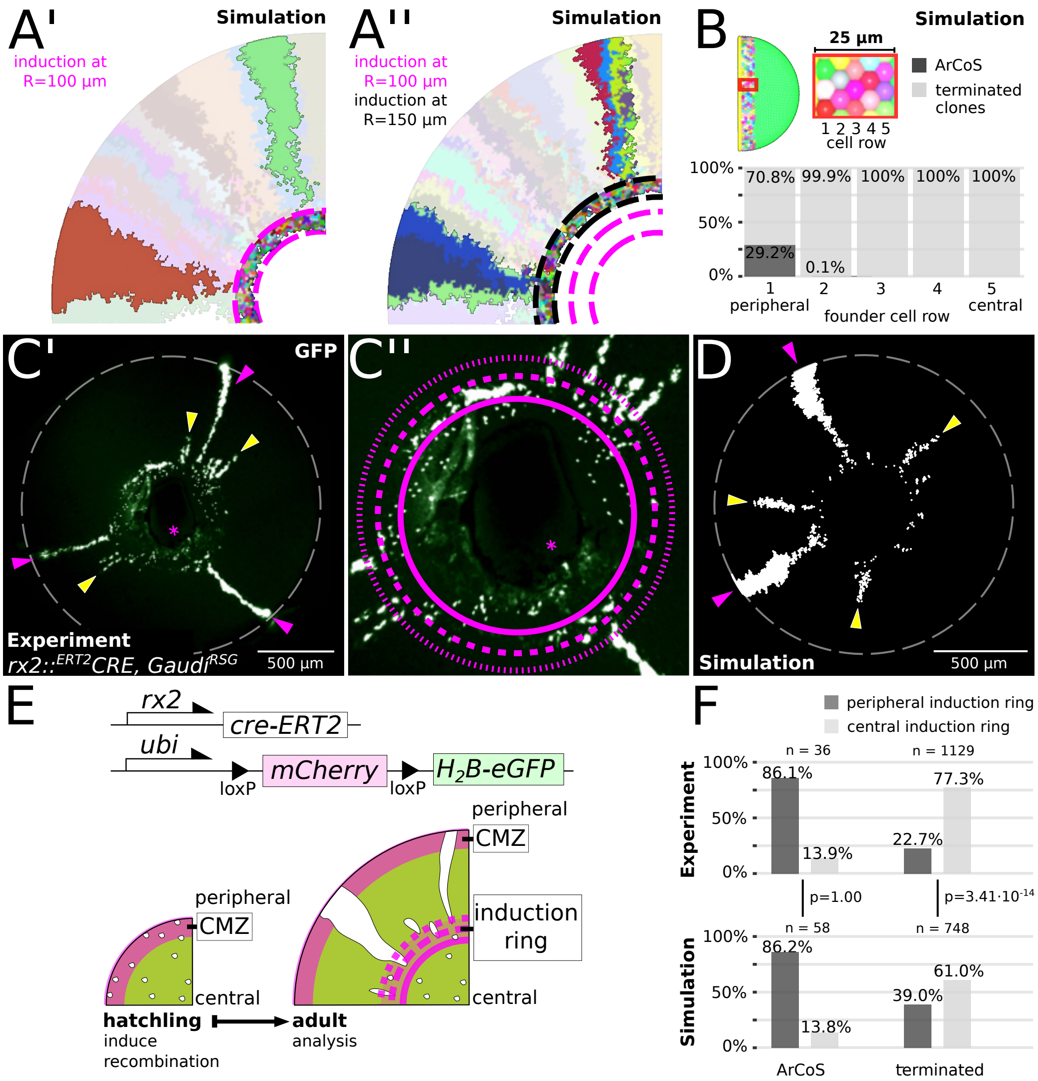


Figure 2.20: Quantitative comparison of the induction ring in the NR with simulations. **A′** Clones induced at $R = 100\ \mu m$ in the simulation. The induction ring is enclosed by dashed magenta lines. Two clones are highlighted. **A″** Clones induced at $R = 150\ \mu m$ of the same simulation as in A′. The induction ring is enclosed by dashed black lines; position of the induction ring at $R = 100\ \mu m$ is highlighted by dashed magenta lines. Two polyclones are highlighted which correspond to highlighted clones in A′. **B** Proportion of terminating and ArCoS-forming SCs in each row of the virtual CMZ. **C′** Maximum projection of proximal view on a sparsely labelled NR. ONL and RPE removed by dissection. Magenta arrowheads: ArCoS. Yellow arrowheads: terminating clones. Magenta asterisk: Optic nerve exit. **C″** Magnification of the central part of C′. Solid magenta line encloses pre-induction retina, dashed magenta line encloses central induction ring, dotted magenta line marks peripheral the induction ring. **D** Example of a sparsely labelled simulation. Magenta arrowheads: ArCoS. Yellow arrowheads: terminating clones. **E** Scheme of the experiment. **F** Proportions of peripheral and central ArCoS and terminating clones in the experiment and the simulation. Figure adapted from Tsingos et al. [2019].

emerged dynamically. Strikingly, only a minority of virtual SCs formed ArCoS, while the vast majority formed terminated clones (Figure 2.20 B). The central-most cells were poised to differentiate by being pushed out of the niche by divisions of their more peripheral neighbours. This neutral competition occurred continuously, as demonstrated by nested virtual inductions (Figure 2.20 A′–A″).

Thus, the simulation made the following two predictions:

1. A large proportion of SCs is lost by neutral competition and forms terminated clones. ArCoS should be a minority among labelled clones.
2. There is a spatial bias in this drift: The majority of ArCoS will derive from peripheral cells but some will derive from more central positions. Similarly, the majority of terminated clones will derive from central positions, but some will derive from peripheral positions.

To address these predictions experimentally, I performed clonal labelling of NR SCs in hatchlings using the *rx2::^{ERT2}cre, Gaudí^{RSG}* line [Centanin et al., 2014; Reinhardt et al., 2015], which when recombined resulted in a nuclear GFP signal, and analysed the eyes at adult stage. Few-cell clusters in the induction ring vastly outnumbered ArCoS, showing that terminated clones were the most common type of clone (n=1129 terminated clones in 20 retinae; Figure 2.20 F). About 3% of terminated clones extended into the post-embryonic retina (n = 32 out of 1129 terminating clones; Figure 2.20 C′, yellow arrowheads). ArCoS, which by definition always reach the retinal margin, were less frequent (Figure 2.20 F; n = 36 ArCoS in 20 retinae). Thus, *rx2*-expressing cells in the CMZ included cells that proliferated indefinitely as well as cells that proliferated only a few times before differentiating. The preponderance of terminated clones showed that ArCoS-forming cells were a minority, in line with the first prediction.

To address the spatially biased stochastic drift, I examined at which position in the induction ring clones contained their central-most pixels in experiment and simulation (Figure 2.20 E–F). Because cells could rearrange slightly during growth and therefore slide into more central positions, this retrospective quantification yielded a higher fraction of centrally starting ArCoS compared to Figure 2.20 B. Among terminated clones, the majority started in central positions (experiment: 77.3%; simulation: 61.0%), while a minority were exclusively located in the peripheral induction ring or in the post-embryonic retina (experiment: 22.7%; simulation: 39.0%). The simulation underestimated the number of terminated clones, probably because *in vivo rx2* was expressed in a larger domain than the reported 5 cell rows [Reinhardt et al., 2015; Wan et al., 2016]. Nevertheless, the experimental data clearly showed that a sizeable subset of terminated clones derived from the periphery of the SC domain of the CMZ, indicating that some SCs drifted into a progenitor-like state.

Among experimental ArCoS, the vast majority (86.1%) started in the periphery, but 13.9% derived from a central position, showing that some cells located in the central progenitor domain of the CMZ drifted into a lifelong SC fate. Strikingly, the ratios for peripheral and central ArCoS in the simulation were nearly identical ($p = 1.00$, 2-sample test for equality of proportions), showing that the simulation captured ArCoS dynamics extremely well. Together, these data supported a model of stochastic drift with a peripheral-stem and central-progenitor bias that was conditioned by the physical topology of the niche.

2.3.2 Terminating clones also occur in other *cre* drivers

My previous analyses showed that only a subset of the peripheral-most *rx2*-positive cells in the CMZ contributed to ArCoS formation. Although *rx2* is a *bona fide* SC marker in the medaka retina, it is not exclusively expressed in SCs, and also occurs in some differentiated retinal cells and potentially in PCs [Reinhardt et al., 2015]. Therefore, the possibility remained that terminated clones did not derive from SCs. Additionally, the uncertain position of the induction ring could have skewed the analysis. To address these issues and complement my previous experiment, I made use of other promoters expressed in the CMZ.

The chemokine (C-C motif) ligand 25 b gene (*ccl25b*) was recently identified as a marker for SCs in the very periphery of the CMZ (Figure 2.21 A′–A″) [Lust and Wittbrodt, unpublished]. Compared to *rx2*, *ccl25b* had a more restricted peripheral range [Lust and Wittbrodt, unpublished]. Another marker in the CMZ is the tailless gene (*tlx*), which is expressed both in differentiated cells and in the CMZ (Figure 2.21 A′–A″)[Reinhardt and Wittbrodt, unpublished]. The domains of *ccl25b* and *tlx* partially overlapped; the overlap was greater in the dorsal than in the ventral CMZ. In the 16 µm-section in Figure 2.21 A′–A″, I counted 20 *ccl25b*$^+$ and 12 *tlx*$^+$ cells in the dorsal CMZ (60% double positive cells), and 25 *ccl25b*$^+$ and 6 *tlx*$^+$ cells in the ventral CMZ (24% double positive cells). Interestingly, double positive cells in the dorsal CMZ were in the very periphery (Figure 2.21 A″, yellow asterisks), while all double positive ventral cells were exclusively centrally located, suggesting they were not SCs. These data were consistent with the dorsal CMZ having cells co-expressing *ccl25b*, *tlx*, and *rx2*, while the ventral CMZ had no triple positive cells [Lust and Wittbrodt, unpublished].

I induced *cre-ert2* expressed under the control of *rx2*, *tlx*, or *ccl25b* promoters in combination with the *Gaudí*^{RSG} transgene (Figure 2.21 B). After a tamoxifen (TMX) pulse lasting at least 3 hours, fish were incubated at least 14 h in 5-

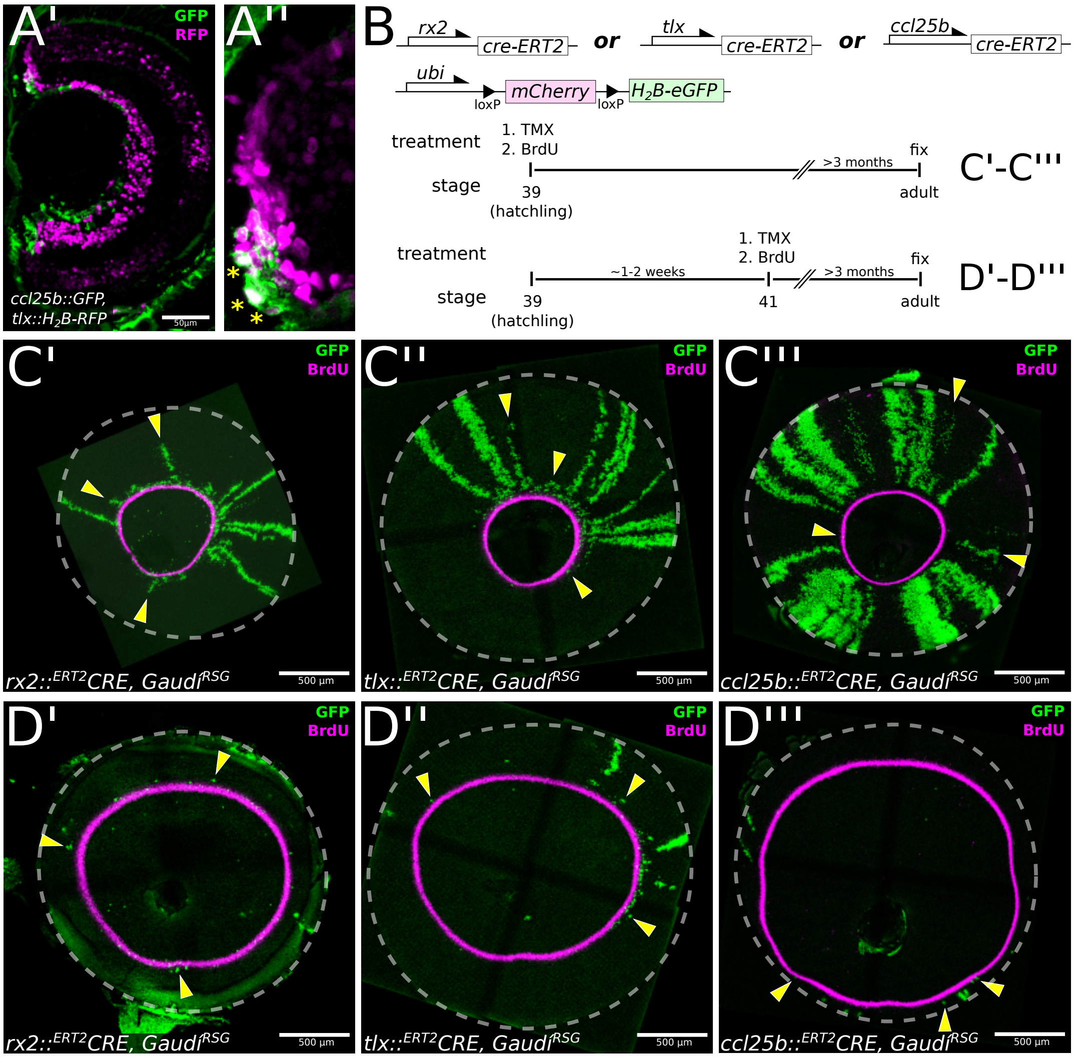


Figure 2.21: All available SC *cre* drivers generate terminating clones.

A′ Maximum projection of a cryosection of double reporter line for *ccl25b* and *tlx* showcasing expression in the hatchling neural retina. Experiment performed in collaboration with Dr Alicia Perez-Saturnino, Cristina Muñoz, and Mai Thu Nguyen. **A″** Magnification of dorsal CMZ in A′, highlighting overlap between *ccl25b* and *tlx* domains. **B** Scheme of transgenic lines and experiments in C′–D‴. Recombination was induced with TMX in hatchling fish at 0 dph, or in stage 41 larvae that were 1 to 2 weeks old. **C′–D‴** Maximum projection of confocal stacks of proximal view on samples immunostained for BrdU; GFP is endogenous fluorescence. ONL and RPE removed by dissection. Images were rotated and flipped horizontally to orient anterior to the left and ventral down. Three terminating clones are highlighted by yellow arrowheads in each sample. Dashed white line: retinal margin. **C′–C‴** NR clones obtained by induction in hatchlings. **D′–D‴** NR clones obtained by induction in stage 41 larvae of different ages.

bromo-2’-deoxyuridine (BrdU) to unambiguously mark the position of the pre-induction retina (Figure 2.21 B).

When inducing with an *rx2* promoter, I observed ArCoS and terminated clones in similar proportions as in previous experiments (Figure 2.21 C′). Notably, most terminated clones emerged peripherally from the BrdU ring (Figure 2.21 C′, yellow arrowheads), underscoring the accuracy of my previous estimate of the induction ring position and the SCs origin of terminated clones. A *cre-ert2* driven by a *tlx* promoter generated both ArCoS and terminated clones in similar proportions to the *rx2* promoter (Figure 2.21 C″). As previously described, ArCoS were exclusively located in the dorsal and posterior retina [Tavhelidse and Wittbrodt, unpublished]. The anterior and ventral retina, where *tlx* was expressed exclusively in central CMZ cells, had only small few-cell clones. In contrast, the more peripherally expressed *ccl25b::*^{ERT2}*cre* lead to a high proportion of ArCoS and very few terminated clones (Figure 2.21 C‴).

As the retina grows larger the number of competing SCs in the CMZ increases. Thus, the chance of labelling an ArCoS-forming SC should decrease if similar numbers of cells are labelled. Indeed, inductions at later timepoints resulted in a qualitatively greater proportion of terminated clones with all *cre* drivers (Figure 2.21 D′–D‴). These complementary experiments strengthen my previous analysis and support the notion that SCs in the neural retina undergo a spatially biased neutral drift throughout the life of the fish.

2.3.3 NR SCs modulate their division axis in a non-random way

NR ArCoS formed stripes that appeared slightly narrower than in the simulation (compare Figure 2.20 C′ and D). In simulations, the division axis was not oriented ("random division axis"; Equation 2.11). The thin clonal stripes suggested that NR SCs had a preferential axis of division along the radial (central-peripheral) coordinate, while circumferential divisions occurred with lower frequency than expected for a random division axis orientation. I therefore tested the impact of introducing a preferred direction for division axes into the model. Thus, I defined two principal division axis orientations along the radial and circumferential coordinate (Figure 2.22 A). When a cell c divided radially, the position $\mathbf{r}_n$ of the daughter cell n was obtained by

$$\mathbf{r}_n = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \mathrm{X}\frac{\rho}{2} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{2.21}$$

where $\begin{pmatrix} x_c & y_c & z_c \end{pmatrix}^{\mathbf{T}}$ are the coordinates of the mother cell c, ρ is the minimal distance between daughter cells, and X is a number chosen uniformly at random from the set (-1,1). When cell c divided circumferentially, $\mathbf{r}_n$ was given by

$$\mathbf{r}_n = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \mathrm{X}\frac{\rho}{2} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}. \tag{2.22}$$

Additionally, I introduced two model parameters: The probability to divide radially according to Equation 2.21 p_{rad}, and the probability to divide circumferentially according to Equation 2.22, which was calculated as $p_{\text{circ}} = 1 - p_{\text{rad}}$.

To quantify circumferential SC divisions in experimental and simulated data, I took advantage of the exquisite temporal order of NR growth to measure ArCoS width – a proxy for circumferential SC divisions. To this end, I modified a quantification pipeline originally developed by Dr Burkhard Höckendorf that unrolled the retina and measured the number of pixels along each radial position normalised by the total circumference – effectively the angle enclosed by two rays traversing the centre of the pre-induction retina and the clone boundaries at every radial position (Figure 2.22 B′–B″). Since patches in the induction ring were more numerous and lead to formation of clone fusions, I focused my analysis on the ArCoS of the post-embryonic retina and excluded the central portion including the induction ring.

As expected, with increasing probability to divide along the circumferential axis, average clone width increased in the simulation (Figure 2.22 C′–C″). Since in the simulation the hemispherical shape constrained the movement of cells, values of $p_{\text{circ}} > 50\%$ did not further increase clone width, and similarly low values of p_{circ} converged to similar mean angle θ.

An inducer growth mode does not necessarily imply regulation of organ shape. To use an analogy, a mass of dough grows from within (similar to the inducer growth mode), but its shape can be imposed externally by a mould (*i.e.* the dough does not affect shape regulation). In the NR, the shape could plausibly be imposed externally by any of the surrounding tissues, and in this case, it would have no role in organ shape regulation. As the space available for cells is imposed externally, any orientation of division axes is theoretically possible; after division cells will locally shift to optimally fill space, as demonstrated by the simulations. In an alternative scenario, organ shape could be regulated by oriented cell divisions of CMZ SCs (Figure 2.22 D). In this scenario, a precise orientation of division axes is necessary.

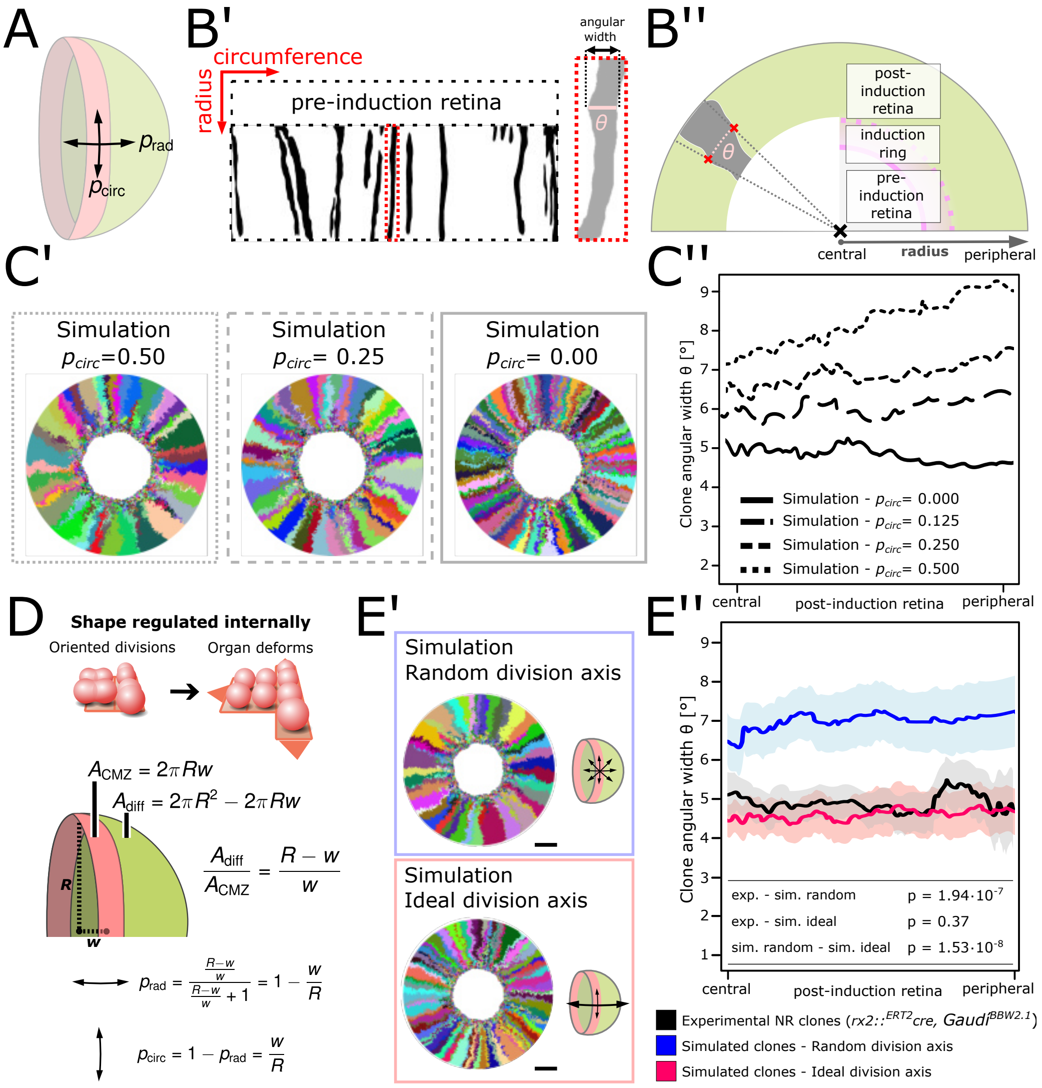


Figure 2.22: Quantitative comparison of ArCoS angular width in the NR with simulations.
A Schematic representation of two principal axes of cell division used in the model. **B′** Representation of unrolled retina and measurement of clone angular width θ. θ was measured at every y-coordinate position. **B″** θ corresponds to the angle enclosed by two rays traversing the centre of the pre-induction retina and the clone edges at a given radius. **C′** Example simulations with different values of p_{circ}. **C″** Plot of clone angular width θ against the normalised post-induction radius for one simulation each using different values for p_{circ}. Data are shown as mean for all simulated clones. **D** Given the assumption that cell division axes impact on organ shape, "ideal" p_{rad} and p_{circ} can be calculated from the area ratio of differentiated retina and CMZ. **E′** Example simulations with completely random division axis orientation, and division axis orientation according to the relationship obtained in (D). **E″** Plot of clone angular width θ against the normalised post-induction radius of experimental data (n = 7 retinae with a sample of 99 ArCoS; black), simulations with random division axis (n = 5 simulations with a sample of 102 ArCoS), and simulations with ideal division axis (n = 5 simulations with a sample of 133 ArCoS). Data are shown as mean and 95 % confidence interval. Figure adapted from Tsingos et al. [2019].

I calculated the ideal proportion of circumferential and radial divisions required to maintain hemispherical geometry. Again, I assumed two principal axes of division, and that each new cell contributed either to the area of the CMZ or to the rest of the eye (Figure 2.22 D). Circumferential divisions (two daughter cells stay in the CMZ) must be balanced by radial divisions (one daughter cell is poised to leave the niche and differentiate). A hemispherical eye of radius R has the area

$$A_{\text{eye}} = 2\pi R^2, \tag{2.23}$$

while the CMZ forms a band of width w at the base of the eye with area

$$A_{\text{CMZ}} = 2\pi R w. \tag{2.24}$$

Thus, the area ratio between the eye without the CMZ and the CMZ is

$$\frac{A_{\text{eye}} - A_{\text{CMZ}}}{A_{\text{CMZ}}} = \frac{R-w}{w} = \frac{R}{w} - 1. \tag{2.25}$$

When division axes perfectly match the ratio in Equation 2.25, the simulation becomes the limiting case of shape regulation where the hemispherical shape is always maintained. Thus, I modelled how this "ideal division axis" ratio given by Equation 2.25 affected simulated ArCoS in the inducer growth mode and compared this to experimental data as well as simulations with random division axis (Figure 2.22 E′–E″). To obtain $\frac{R}{w} - 1$ radial divisions for every circumferential division, I formulated the probability of a radial division as

$$p_{\text{rad}} = \frac{\frac{R}{w} - 1}{\frac{R}{w} - 1 + 1} = 1 - \frac{w}{R}, \tag{2.26}$$

and the probability for circumferential divisions as

$$p_{\text{circ}} = 1 - p_{\text{rad}} = \frac{w}{R}. \tag{2.27}$$

Experimental ArCoS width averaged to 4.87° (Figure 2.22 E″; n = 99 ArCoS across 7 retinae from the *rx2::^{ERT2}cre, Gaudí$^{BBW2.1}$* line). In contrast to experimental data, ArCoS width in simulations with random division axis averaged to 7.28° (Figure 2.22 E″; n = 102 clones from 5 simulation runs; compared to experimental data: $p = 1.94 \cdot 10^{-7}$, Welch two-sample t-test). In the model with ideal division axes, radial divisions always exceeded circumferential divisions as $R \gg w$; moreover, the frequency of radial divisions increased as the retinal radius R grew. Here, ArCoS width closely matched experimental data, averaging at 4.54° (Figure 2.22 E″; n = 133 clones from 5 simulation runs; compared to experimental data: p = 0.37, Welch two-sample t-test).

These data show that NR SC divisions were not randomly oriented, but instead were preferentially oriented along the central–peripheral axis. Moreover NR SCs underwent radial and circumferential divisions at a rate consistent with a role in organ shape regulation.

2.3.4 Differential parameters locally influence retinal topology

I observed that the pre-induction retina was always shifted towards the ventral side (Figure 2.23 A′). Indeed, the distance between retinal margin and concentric rings of BrdU pulses was consistently longer dorsally than ventrally (Figure 2.23 A‴). Additionally, there appeared to be a slight anterior-posterior asymmetry, with posterior slightly longer but not to a significant degree (Figure 2.23 A″–A‴). The pre-induction retina covered the entire retinal surface at induction (Figure 2.23 B). Equal growth around the circumference should maintain the pre-induction retina in the centre. Therefore, I hypothesized that the ventralward shift indicated different division parameters in ventral CMZ cells.

I probed the feasibility of different scenarios in generating a ventral shift in an *in silico* screen. To this end, I defined a 90° sector of the hemisphere as the simulated ventral retina (Figure 2.24 A′–A″). "To determine if a given cell c located at $\mathbf{r}_c$ lies in this sector, I first calculated the radius of a small circle on a hemisphere of radius R enlarged by the radius r of the cell c:

$$R_{\text{small}}(t) = \sqrt{(R(t)+r)^2 - \left(\mathbf{r}_{c_1} - \mathbf{s}_1\right)^2}, \tag{2.28}$$

where $\mathbf{r}_{c_1}$ and $\mathbf{s}_1$ denote the x-component of cell position $\mathbf{r}_c$ and the centre of the hemisphere $\mathbf{s}$, respectively. A cell was in the ventral sector if

$$\mathbf{r}_{c_2} < \mathbf{s}_2 - \sin(45°) R_{\text{small}}(t), \tag{2.29}$$

where $\mathbf{r}_{c_2}$ and $\mathbf{s}_2$ denote the y-component of $\mathbf{r}_c$ and $\mathbf{s}$, respectively." [Tsingos et al., 2019]. Further, I discerned two mechanisms for SCs in this ventral domain to select a different division behaviour:

1. A lineage-bound intrinsic signal (*e.g.* epigenetic imprinting).
   In this scenario, any daughter cell from a lineage that fulfilled Equation 2.29 in the first simulation step will select the "ventral behaviour" regardless of its current position.
2. A lineage-independent extrinsic signal (*e.g.* a local diffusible molecule).

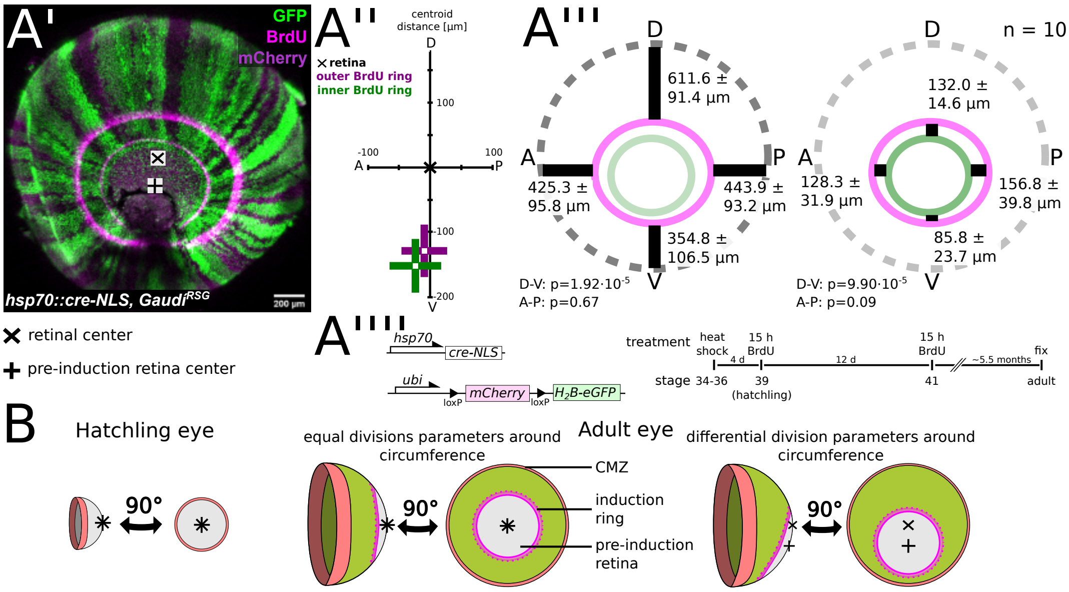


Figure 2.23: The medaka retina has a dorso-ventral asymmetry.
A′ NR of a fish treated with two 15 h BrdU pulses at 0 days post hatch (dph) and 13 dph and fixed 5.5 months later. Sample with endogenous GFP (green) and monomeric Cherry (mCherry) (dark magenta), and immunostained for BrdU (bright magenta). Rotated and flipped to orient anterior to the left and ventral down. Maximum projection of a confocal stack of proximal view; ONL and RPE removed by dissection. **A″** Mean and standard deviation of the relative position of centroids of ellipses fit to the concentric BrdU rings in relation to the retinal centroid. **A‴** Mean retina and BrdU ring proportions. Numbers denote length of solid black lines (mean ± standard deviation). Data in A″ and A‴ measured in 10 retinae from 5 fish. Abbreviations: A – anterior, P – posterior, D – dorsal, V – ventral. p-values: Welch two-sample t-test. **A⁗** Scheme of the transgenic construct and experiment. **B** Since the pre-induction retina covers almost the entirety of the hatchling eye's surface, equal growth around the CMZ circumference should maintain this area in the central position. A ventralward shift indicates differential division parameters around the CMZ circumference. Panel B adapted from Tsingos et al. [2019].

In this scenario, every cell constantly re-evaluates its current position according to Equation 2.29 prior to selecting its division behaviour.

For each mechanism, the following parameters were altered:

1. The probability of division in the ventral sector $p_{\text{div}_{\text{ventral}}}$.
   This value was either kept equal to the value in the non-ventral sector $p_{\text{div}_{\text{non-ventral}}} = \frac{1}{26}\ \text{h}^{-1}$, or halved $p_{\text{div}_{\text{ventral}}} = \frac{1}{2} p_{\text{div}_{\text{non-ventral}}} = \frac{1}{52}\ \text{h}^{-1}$ such that ventral cells had a lower chance to commit to cell division.
2. The probability of circumferential divisions in the ventral sector $p_{\text{circ}_{\text{ventral}}}$.
   This parameter was varied between circumferentially-biased $p_{\text{circ}_{\text{ventral}}} = 1$ and radially-biased $p_{\text{circ}_{\text{ventral}}} = 0$. The probability of circumferential divisions in the non-ventral sector was always radially biased, thus: $p_{\text{circ}_{\text{non-ventral}}} = 0$. Division axis orientation was given by Equations 2.21 and 2.22.

In short, I tested 8 different conditions (Figure 2.24 B′–C⁗). In control simulations where all cells behaved equally, the pre-induction retina stayed cen-

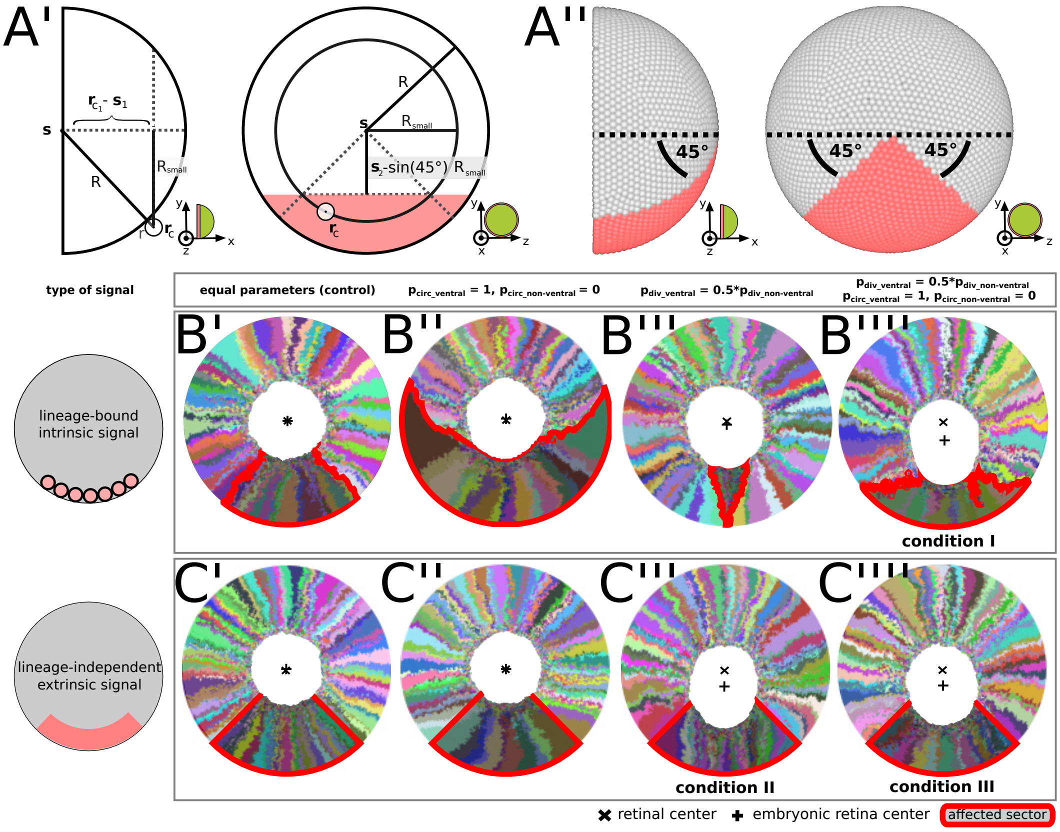

A'
$r_{c_1} - s_1$
s
R
R_{small}
r_c
r
y
x
z
R
R_{small}
s
s_2-sin(45°) R_{small}
r_c
A''
45°
45°
45°
type of signal
equal parameters (control)
$p_{circ\_ventral}$ = 1, $p_{circ\_non\text{-}ventral}$ = 0
$p_{div\_ventral}$ = 0.5*$p_{div\_non\text{-}ventral}$
$p_{div\_ventral}$ = 0.5*$p_{div\_non\text{-}ventral}$
$p_{circ\_ventral}$ = 1, $p_{circ\_non\text{-}ventral}$ = 0
lineage-bound intrinsic signal
B'
B''
B'''
B''''
condition I
lineage-independent extrinsic signal
C'
C''
C'''
C''''
condition II
condition III
× retinal center
+ embryonic retina center
affected sector

Figure 2.24: *In silico* screen for conditions that lead to a ventral shift.
A′ Scheme of the eye highlighting dimensions used in Equations 2.28 and 2.29. Shaded in red is the region that fulfills both equations for the example cell $\mathbf{r}_c$. **A″** Simulation screenshots corresponding to views in A′ showing in red cells that fulfill Equations 2.28 and 2.29. **B′–B⁗** Simulations where a lineage-bound intrinsic signal drives differential proliferative behaviour. **C′–C⁗** Simulations where a lineage-independent extrinsic signal drives differential proliferative behaviour. Figure adapted from Tsingos et al. [2019].

tered (Figure 2.24 B′, C′). For a lineage-bound intrinsic signal, a circumferential bias lead to massive enlargement of ventral lineages at the expense of adjacent clones without affecting the pre-induction retina (Figure 2.24 B″). Reducing proliferation probability resulted in termination of ventral lineages, as adjacent clones displaced them from the virtual niche (Figure 2.24 B‴). An intrinsic signal resulted in a ventral shift only if circumferential bias was combined with lower proliferation probability (Figure 2.24 B⁗ – condition I). In these simulations, circumferential divisions allowed ventral lineages to physically occupy niche positions (preventing their displacement) while lower proliferation reduced pressure on cells of the pre-induction retina, allowing a ventral shift. In the scenario of a lineage-independent extrinsic signal, two conditions resulted in a ventral shift of the pre-induction retina: Both lower division probability (Figure 2.24 C‴ – condition II) and the combination of lower division probability with circumferential division axis bias (Figure 2.24 C⁗ – condition III).

To identify which scenario was most plausible, I compared patches in the ven-

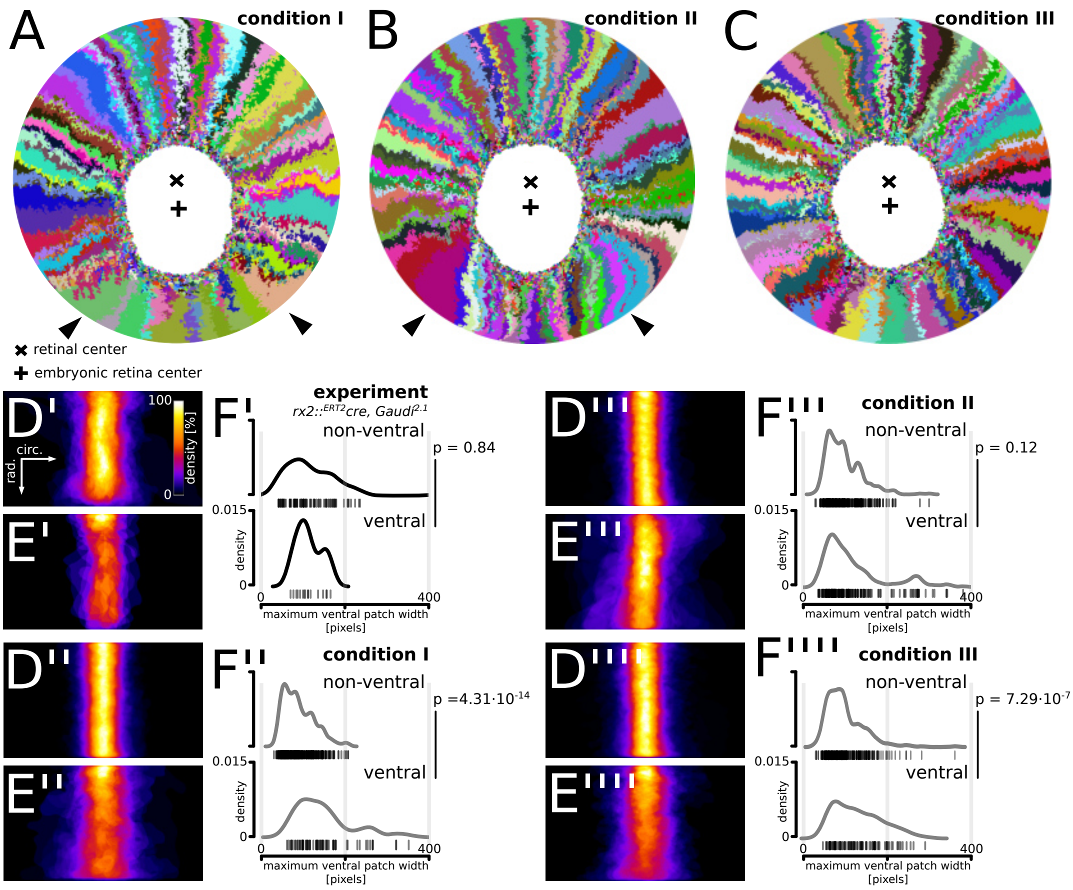


Figure 2.25: Quantitative comparison of non-ventral and ventral patch width.
A Magnification of 2.24 B′′′′. **B** Magnification of 2.24 C′′′. A–B: Black arrowheads highlight unusual clones. **C** Magnification of 2.24 C′′′′. **D′–D′′′′** Superposition of non-ventral patches. **E′–E′′′′** Superposition of ventral patches. **F′–F′′′′** Distribution of maximum patch width for non-ventral (top) and ventral patches (bottom). p-values: Wilcoxon rank sum test. Experimental NR data are the same as in 2.19. Figure adapted from Tsingos et al. [2019].

tral and nonventral sectors. Both in experiments and all three simulated conditions, patch shape in the non-ventral sector had low variability (Figure 2.25 D′–F′′′′). At a glance, ventral clones in the experiment didn't differ markedly from non-ventral clones, but appeared to have a tendency to terminate more often (Figure 2.25 D′, E′). In terms of the distribution of maximum patch width, experimental NR patches did not differ substantially between non-ventral and ventral sectors (Figure 2.25 F′).

In simulated condition I, ventral ArCoS started narrow but then broadened (Figure 2.25 D″, E″) and interdigitated circumferentially (Figure 2.25 A), unlike the very uniform stripes in the experimental data. The broader ventral ArCoS lead to a more dispersed distribution compared to the non-ventral sector (Figure 2.25 F”). In condition II, the majority of ventral ArCoS formed very narrow stripes, but at the border to the non-ventral sector ArCoS were broad and curved (Figure 6 2.25 B). Again, this resulted in more shape variation (Figure 2.25 D′′′, E$'''$). Nevertheless, these outliers were outweighed by a high density of narrow clones, such that the overall distribution was similar between ventral and non-ventral sectors (Figure 2.25 F$'''$). Clones in the ventral and non-ventral sectors were qualitatively similar in condition III (Figure 2.25 C, D$''''$, E$''''$). Ventral clones however tended to be broader, resulting in a more dispersed distribution compared to the non-ventral sector (Figure 2.25 F$''''$).

In conclusion, ventral NR SCs have a different behaviour than elsewhere along the circumference, leading to a ventral-ward shift of the embryonic retina. The simulations suggest that this different behaviour consists of modulation of proliferation parameters by an extrinsic signal in the ventral CMZ. Both proliferation rate and division axes are probably altered in the ventral sector, but the current experimental data lacks the resolution to confidently pinpoint the exact changes in parameter values.

2.3.5 The NR is anatomically subdivided into quadrants

Interestingly, some additional patterns emerged in the data. ArCoS generated by inducing *tlx::*^{ERT2}*cre* occurred exclusively in the dorso-posterior half of the NR (Figure 2.26 A$'$). Further, the *ccl25b::*^{ERT2}*cre* formed ArCoS less frequently at the four retinal poles, as visualised by overlaying all retinae induced at the same timepoint (Figure 2.26 B$'$). To investigate the origin of these patterns, I visualised expression domains of *tlx* and *ccl25b* in wholemount preparations of lines carrying GFP reporters for these genes. Indeed, *tlx::GFP* expression in the peripheral-most CMZ appeared confined to the dorso-posterior retinal quadrant (Figure 2.26 A$''$), while *ccl25b::GFP* expression was weaker in the retinal poles (Figure 2.26 B$''$), particularly anterior and ventral. Due to sparse labelling and a lack of BrdU pulse to aid in orientation of retinae, similar patterns were not as evident in ArCoS generated by the *rx2::*^{ERT2}*cre* and *ubi::*^{ERT2}*cre* (Figure 2.26 C$'$, D). Antibody staining against rx2 protein (rx2) showed no expression in the ventral pole, and slightly reduced levels in the anterior and posterior poles. Upon further inspection, the retinal poles appeared anatomically distinct, containing putative attachment sites for zonular fibres; the ventral pole also contained a ventral blood vessel and the ciliary canal (Figure 2.26 E), an anatomical feature of teleost eyes where the aqueous humor is drained [Soules and Link, 2005].

These data suggested that the NR CMZ was subdivided into four quadrants: Dorso–anterior, dorso–posterior, ventro–anterior, and ventro–posterior. These quadrants may result from anatomical specialisation at the retinal poles, as well as subfunctionalisation of retinal domains as suggested by *tlx* expression.

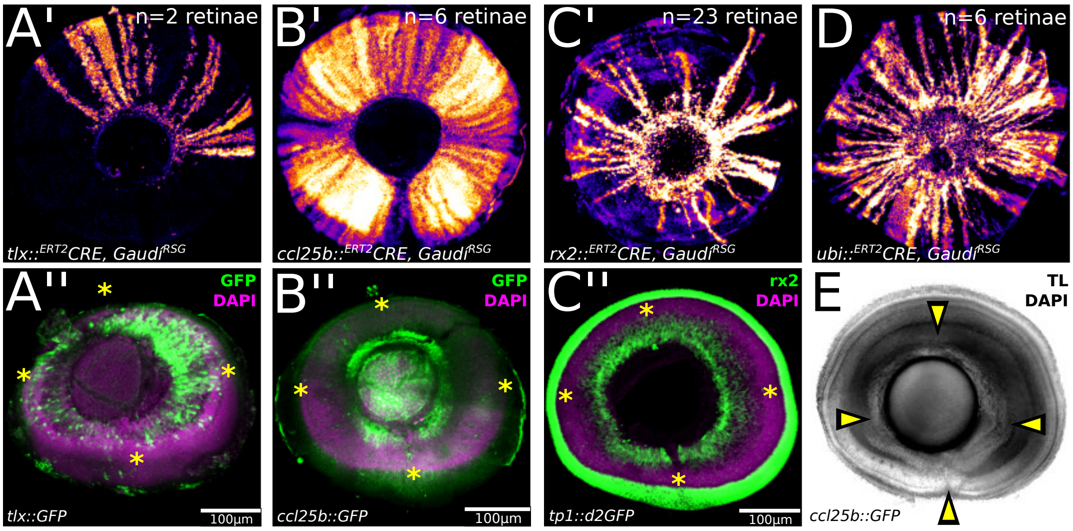


Figure 2.26: CMZ marker expression shows a pattern in quadrants.
A′ Overlay of ArCoS from 2 retinae of *tlx::^{ERT2}cre* induced at 0 dph. Tlx ArCoS form exclusively in the dorso–posterior quadrant. Retinae were oriented and aligned based on a BrdU ring labelling the pre-induction retina. **A″** Immunostained *tlx::GFP* reporter retina at 0 dph. *tlx* expression in the peripheral CMZ is confined to the dorso-posterior quadrant. **B′** Overlay of ArCoS from 6 retinae of *ccl25b::^{ERT2}cre* induced at 0 dph. *ccl25b* retinae were oriented and aligned based on a BrdU ring labelling the pre-induction retina. **B″** Immunostained *ccl25b::GFP* reporter retina at 0 dph. *ccl25b* reporter expression is reduced at the retinal poles, in particular anterior and ventral. **C′** Overlay of ArCoS from 23 retinae of *rx2::^{ERT2}cre* induced at 0 dph. Except for 2 samples, none of the retinae had a BrdU ring marking the pre-induction retina to aid in orientation of samples. **C″** Wildtype retina immunostained for rx2 at 0 dph. Note absent expression at the ventral pole and weaker expression at anterior and posterior poles. **D** Overlay of ArCoS from 6 retinae of *ubi::^{ERT2}cre* induced at 0 dph. None of the retinae had a BrdU ring marking the pre-induction retina to aid in orientation of samples. **E** Retinal anatomy differs near the retinal poles (yellow arrowheads). These anatomical structures are stereotypical, and can be detected in hatchlings. They become more pronounced as larvae grow. Image shows a single slice of an optical stack through a retina of a 13 dph larva. Panels A′, B′, C′, and D are proximal views. Panels A″, B″, C″ are maximum projections of confocal stacks of wholemount distal views of hatchling eyes. Retinae in all panels were rotated and horizontally flipped to orient anterior to the left and ventral down. Yellow asterisks in panels A″, B″, and C″ label retinal poles.

Summary

In the CMZ, position along the central–peripheral coordinate, orientation of division axes, and cell proliferation rates can locally enhance or reduce the impact of stochastic neutral drift on clonal dynamics. The quantitative comparison of NR clonal properties to different realisations of the model allowed to narrow down how NR SCs may modulate their proliferative parameters in different retinal quadrants to regulate organ shape and retinal topology.

2.4 Characterisation of homeostatic RPE SC dynamics

After characterising cell proliferation properties in the NR as a tissue growing with the inducer mode, I next focused on an in-depth analysis of the RPE. Patch shape analysis of clonal data in the RPE was compatible with the responder growth mode. Intriguingly, the degree of observed clonal variability in the experimental data appeared higher than in the model (section 2.2.3). Thus the simulation predicted that cell division intervals *in vivo* were even more variable than initially modelled. In the following, I address this prediction experimentally with thymidine incorporation assays (Section 2.4.1), re-evaluate and extend the model to better fit the data (Sections 2.4.2 and 2.4.3), and finally I experimentally characterise the extent and proliferative dynamics of the RPE stem cell niche (Sections 2.4.4, 2.4.5, and 2.4.5).

2.4.1 RPE SCs exhibit long quiescent periods

The responder growth mode predicted that the high heterogeneity of RPE clonal patches was due to greater variability in SC division intervals. In the model, chance events could lead to some SCs delaying cell division for long periods of time. This delay impacts not only on clonal shape, but also on BrdU incorporation and retention, which is more heterogeneous in the responder growth mode. Thus, I performed BrdU pulse-chase experiments consisting of an overnight BrdU pulse in hatchlings, followed by several months of chase before analysing the retinae in proximal view. For simulated data, the BrdU pulse was started at simulated day 2 and programmed to last 16 simulated hours, while the chase period was 2-3 simulated months; virtual cells incorporated BrdU only in the simulation step when they divided, and at every subsequent division daughter cells halved their BrdU content (Appendix Figures 5.16 and 5.17).

As expected of the spatiotemporal growth order of the retina, in all conditions the strongest label was retained in a central ring of cells demarcating the position of the CMZ at the time of the pulse. In the inducer growth mode, label dilution was homogeneous around the circumference (Figure 2.27 A′). In the responder growth mode, greater variability in cell division intervals in the simulation resulted in a greater incidence of label-retaining cells at a distance from the central ring of BrdU incorporation (Figure 2.27 A″ white asterisks). Strikingly, the BrdU incorporation profile of the RPE was highly heterogeneous, with many clusters of cells retaining strong signal far exceeding the simulation (Figure 2.27 B″ white asterisks). BrdU label-retaining cells formed "trails" emanat-

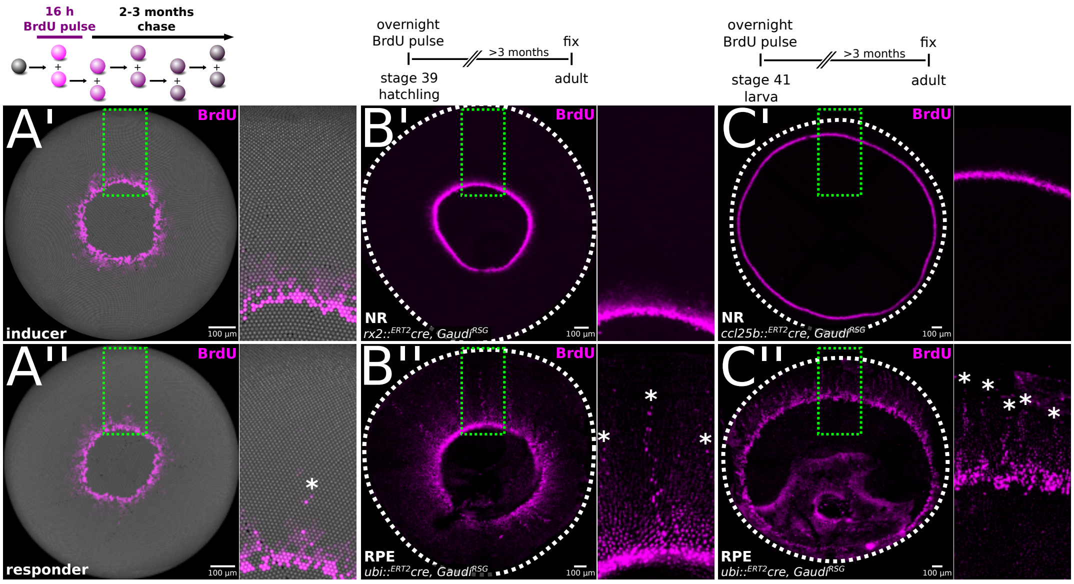


Figure 2.27: BrdU pulse-chase in experimental and simulated data.
A′ Left: Inducer growth mode simulation pulsed for 16 h with BrdU at simulation start. Right: Magnification of boxed area in A′. **A″** Left: Responder growth mode simulation pulsed for 16 h with BrdU at simulation start. Right: Magnification of boxed area in A″. There are a few label-retaining cells (asterisk). Simulation parameters as in Tables 2.1 and 2.2. **B′** Left: NR of a fish pulsed with BrdU once overnight at hatchling stage. Right: Magnification of green boxed area in B′. **B″** Left: RPE of a fish pulsed with BrdU once overnight at hatchling stage. Signal from underlying NR was removed by digital masking. Right: Magnification of boxed area in B″. Asterisks: label-retaining cells. **C′** Left: NR of a fish pulsed with BrdU once overnight at larval stage 41. Right: Magnification of boxed area in C′. **C″** Left: RPE of a fish pulsed with BrdU once overnight at larval stage 41. Right: Magnification of boxed area in C″. Note several strings of label-retaining cells (asterisks). Panels B′–C″ show maximum projections of a confocal stack. Samples were oriented to put ventral down.

ing from the central ring that appeared to form a sectoring pattern (Figure 2.27 B″ white asterisks).

To further investigate RPE trail sectoring, I repeated the experiment in 2 week-old larvae, which revealed an increase in the number of trails (Figure 2.27 C″ white asterisks). The distance between trails appeared to remain constant (Figure 2.27 C″ white asterisks). In contrast, the NR had a very homogenous label dilution profile regardless of the timepoint of the pulse similar to the inducer growth mode (Figure 2.27 B′, C′). Here, label strength decreased in a circular gradient from centre to periphery.

These data led me to the following hypotheses:

1. BrdU trails arose because RPE cells entered a stable but reversible quiescent state.
2. The sectoring pattern of BrdU trails resulted from neighbouring domains of quiescent and active RPE cells.

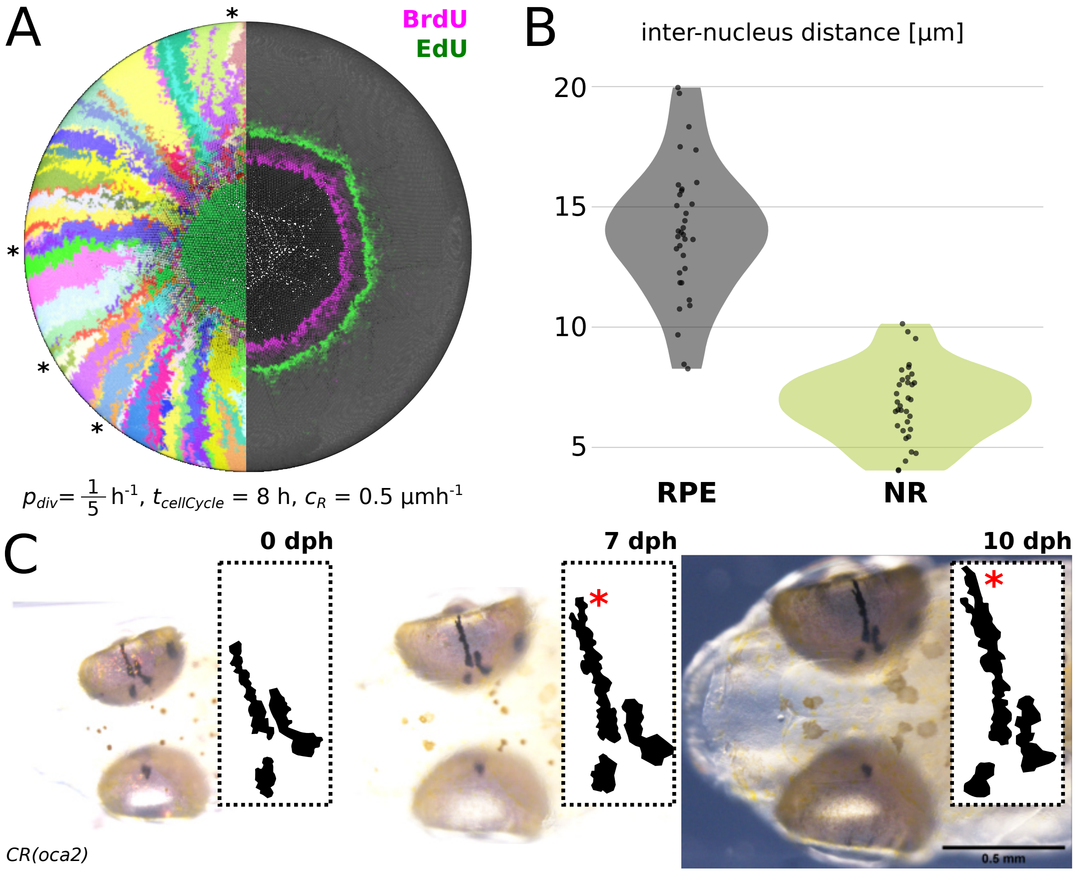


Figure 2.28: Parameter variation cannot capture thymidine analogue incorporation trails. **A** Composite image of simulated clonal growth and thymidine analogue incorporation. Responder growth mode with parameters maximizing late clone occurrence (asterisks) and realistic radius growth rate. BrdU and 5-Ethynyl-2'-deoxyuridine (EdU) incorporation patterns failed to display trails of label-retaining cells regardless of the time of pulse. BrdU and EdU pulses lasted 16 h each and were at 50 h and 150 h, respectively. Other parameters as in Tables 2.1 and 2.2. **B** Inter-nucleus distance of differentiated RPE and NR cells measured on a wholemount adult retina. Data points correspond to individual nuclei pairs. **C** Time-lapse of a mosaic albino fish. Fish were injected by Omar Hammouda. The pigmented clones on the right eye have been traced and magnified in the inset. Red asterisk: bifurcation emerging in the clone at the retinal periphery.

2.4.2 Extended model of RPE proliferative cell homeostasis

First, I verified that parametrisation could not explain the difference between simulated and experimental data. Clone variability and heterogeneous label retention suggested that the (potential) rate of cell proliferation far exceeded the rate of surface area growth ($A_{\text{cells}} \gg A_{\text{eye}}$). To more faithfully model the RPE, I set the growth rate of the eye radius to $c_{\text{R}} = 0.5\ \mu\text{mh}^{-1}$, which required adjustment of p_{div} and $t_{\text{cellCycle}}$ (see Sections 2.1.3 and 2.2.1). Indeed, more frequent late arising clones were captured by optimizing parameter values (Figure 2.28 A, asterisks). However, parameter variation could not replicate the striking difference in BrdU incorporation pattern, indicating that a more fundamental aspect of RPE biology was missing in the model.

Most studies on the CMZ have focused on the NR. Therefore, the assumptions for the model were primarily based on observations made in the NR extrapolated to the RPE (Sections 2.1.1 and 2.1.2). To evaluate the validity of these assumptions for the medaka RPE, I first confirmed that RPE cells did not rearrange in the central retina by observing that clones retained their relative position over time in mosaic albino fish (Figure 2.28 C). This experiment showed that changes in clone shape such as bifurcations occurred only in peripheral positions (Figure 2.28 C, red asterisk). Second, I measured the nuclear neighbourhood distance in wholemount preparations of the NR and RPE, which confirmed that average inter-nucleus distance (a proxy for cell diameter) in the NR was ≈ 6.8 µm, but in the RPE it was twice as large: ≈ 14.0 µm (Figure 2.28 B). Therefore, I increased the simulated cell radius to $r = 7.0$ µm.

In the responder growth mode model, cells became temporarily quiescent if local cell density exceeded a threshold (Equation 2.18). This transient quiescence was memoryless, meaning that previously experienced quiescence or lack thereof did not influence a cell's future decision to re-enter or exit quiescence. Building upon this model, I tested whether including memory of previous states could better explain the experimental data. To this end, I introduced two new parameters:

1. $t_{\text{quiescence}}$

   This parameter defines the duration of the quiescent state in hours.

2. t_{trigger}

   This parameter defines how much time in hours a cell must experience local cell density exceeding the threshold set by $\delta_{\text{ol}_{\text{threshold}}}$ (Equation 2.18) to trigger entry into quiescence.

Both parameters model different types of "memory"; $t_{\text{quiescence}}$ represents an internal timing mechanism that ensures persistence of quiescence after it has been triggered, while t_{trigger} represents memory of previous quiescence-inducing states. This latter parameter implies that a cell integrates information over a span of time to ascertain whether the threshold time t_{trigger} has been exceeded. In the model, I implemented this measurement as follows:

$$t_{\text{measured}}(t) = \begin{cases} t_{\text{measured}}(t-1)+1, & \frac{1}{n}\sum_{i=1}^{n} \mathrm{d}(\mathbf{r}_c, \mathbf{r}_{n_i}) > \delta_{\text{ol}_{\text{threshold}}} 2r \\ t_{\text{measured}}(t-1)-1, & \frac{1}{n}\sum_{i=1}^{n} \mathrm{d}(\mathbf{r}_c, \mathbf{r}_{n_i}) \leq \delta_{\text{ol}_{\text{threshold}}} 2r \end{cases}. \tag{2.30}$$

Thus, at every simulation step t, cells incremented t_{measured} if the average overlap to all neighbours given by Equation 2.18 exceeded the threshold given by the fraction $\delta_{\mathrm{ol}_{\mathrm{threshold}}}$ of their diameter $2r$, otherwise t_{measured} was decreased. The entry into a quiescent state was triggered if

$$t_{\mathrm{measured}}(t) > t_{\mathrm{trigger}}. \tag{2.31}$$

The formulation using Equations 2.30 and 2.31 ensured that chance fluctuations in cell density were averaged out. To initialise an appropriate distribution of t_{measured} at the beginning of the simulation, during the first t_{trigger} simulation steps only Equation 2.30 was calculated and no other actions occurred (*e.g.* no cell division, no eye radius growth).

2.4.3 Both memoryless and memoried quiescence can generate trailing BrdU patterns

To test whether the extended model could replicate BrdU trails, I performed an exploratory parameter scan on the following parameters:

- p_{div}
- $t_{\mathrm{cellCycle}}$
- t_{trigger}
- $t_{\mathrm{quiescence}}$

Moreover, the larger cell dimensions in the extended model motivated me to also vary the width of the CMZ w.

Interestingly, the larger cell size itself led to an increase in label-retaining cells (Figure 2.29). With increasing w, the model approached – but did not fully recapitulate – the BrdU heterogeneity of the experimental data (Figure 2.29 A′–A‴). Smaller w (*i.e.* fewer proliferative cell rows) reduced cell competition and lead to wide clones with more homogenously diluted BrdU and EdU signal (Figure 2.29 A′). Larger w lead to narrow ArCoS and more heterogeneous BrdU and EdU dilution, indicating a correlation between cell competition and the emergence of long quiescent periods in a memoryless fashion (Figure 2.29 A‴). Varying p_{div} and $t_{\mathrm{cellCycle}}$ had no qualitative impact (Figure 2.29 B′–B‴).

A memoried quiescent state also lead to BrdU trail formation for some parameter values (Figure 2.29 C″, D′, D‴). In this exploratory parameter scan, it became apparent that $t_{\mathrm{trigger}} \leq t_{\mathrm{cellCycle}}$, as otherwise feedback inhibition of excessive cell division would be too slow to prevent unphysical cell crowding. Notably, the best qualitative fit to the experimental data in the model with memoried quiescence also required a larger proliferative population in the CMZ.

In summary, both memoried and memoryless variants of the model could

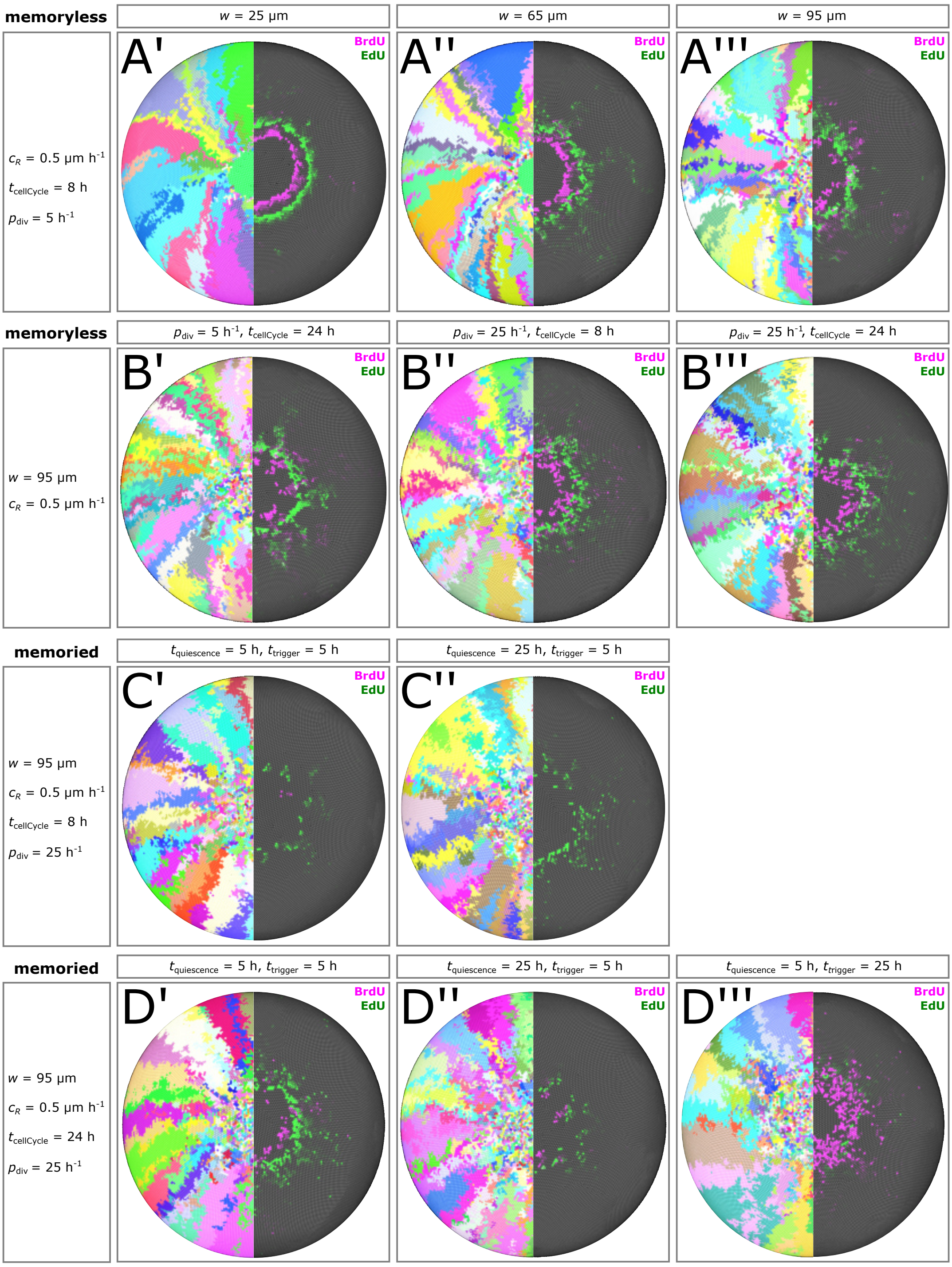


Figure 2.29: Parameter scan of the memoryless and memoried model with larger cell size. **A′–A‴** Determination of the optimal CMZ width w to maximize label-retaining cells. **B′–B‴** Variation of the minimum cell cycle time $t_{\mathrm{cellCycle}}$ and the probability of division p_{division} with the optimal CMZ width w. **C′–C″** Variation in $t_{\mathrm{quiescence}}$ with small values of $t_{\mathrm{cellCycle}}$. Simulations where $t_{\mathrm{trigger}} > t_{\mathrm{cellCycle}}$ produce implausibly high cell densities and are not shown. **D′–D″** Variation in $t_{\mathrm{quiescence}}$ and t_{trigger} with large values of $t_{\mathrm{cellCycle}}$. Images are composite from two screenshots. BrdU and EdU pulses lasted 16 h each and were done at 50 h and 150 h, respectively. $r = 7\,\mu\mathrm{m}$. Parameters not indicated were as listed in Tables 2.1 and 2.2.

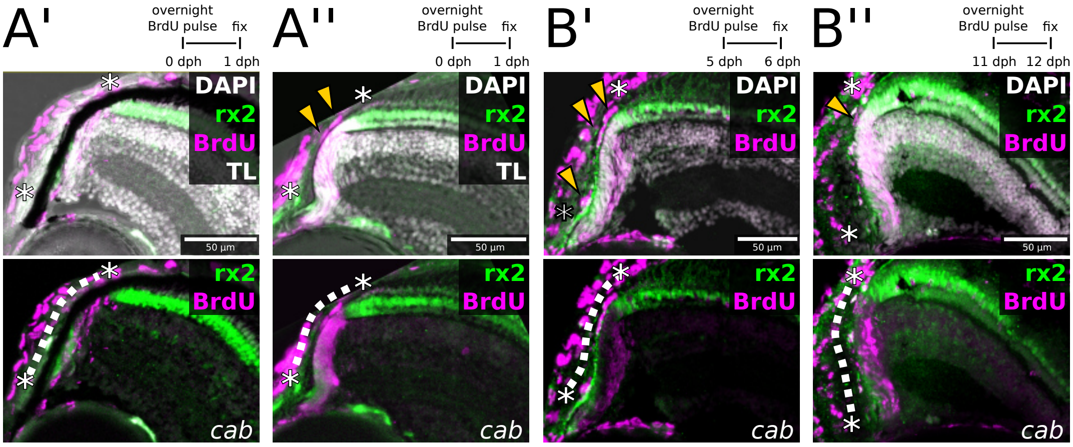


Figure 2.30: The RPE CMZ contains several flattened nuclei.
A′ Melanin pigment in the RPE prevents imaging of stained markers. **A″** In bleached sections RPE nuclei become visible. **B′** Bleached section of 6 dph fish. **B″** Bleached section of 12 dph fish. A′, A″, and B′ are sum projections, B″ is a max projection of confocal images of a cryosection; images show detail of dorsal retina. White asterisks mark extent of RPE CMZ based on BrdU incorporation and morphology. White dotted line contours RPE CMZ. Yellow arrowheads highlight BrdU-positive RPE cells. Experiments and data acquisition were performed in collaboration with Mai Thu Nguyen.

qualitatively recapitulate the experimental clonal ArCoS pattern while obtaining a more heterogeneous BrdU and EdU incorporation profile within certain parameter ranges. Label retention heterogeneity correlated with cell competition and number of persistent clones. However, neither model was able to replicate the sectoring pattern of BrdU label retention, indicating either a flaw in the model assumptions or missing information about the system. Both models predicted that *in vivo* the number of proliferative cell rows in the RPE was larger than initially assumed. Next, I set to validate this prediction experimentally and further characterise the RPE SC niche to constrain parameters in the model.

2.4.4 The RPE niche contains up to 12 rows of proliferative cells

Although generally acknowledged that RPE SCs reside within the CMZ [Shi et al., 2017], their precise location, number, and cell cycle dynamics have not been characterised. To identify RPE SCs, Mai Thu Nguyen and I performed BrdU pulse experiments without a chase period and analysed the samples in 16 µm cryosections. Since melanin pigments mask fluorescence (Figure 2.30 A′), Mai Thu Nguyen established a bleaching protocol on cryosections to visualise RPE cells (Figure 2.30 A″). This protocol improved clarity at the cost of reduced staining quality and tissue damage.

All RPE cells adjacent to the NR CMZ incorporated BrdU in some, but not all, sections (Figure 2.30 A″, yellow arrowheads). This held true for older larvae as

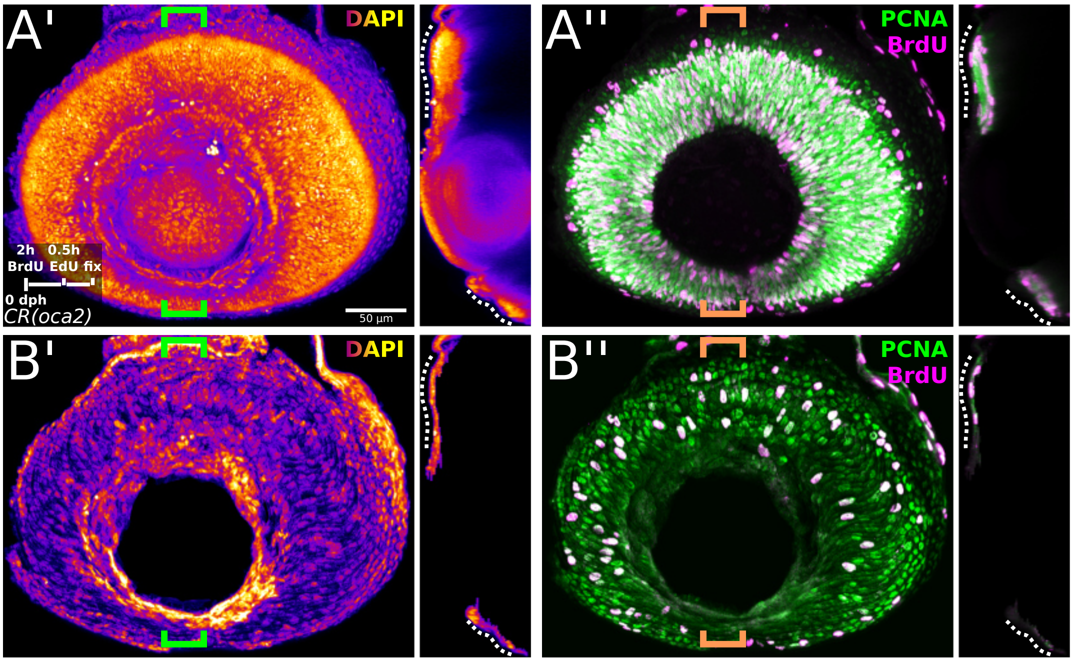
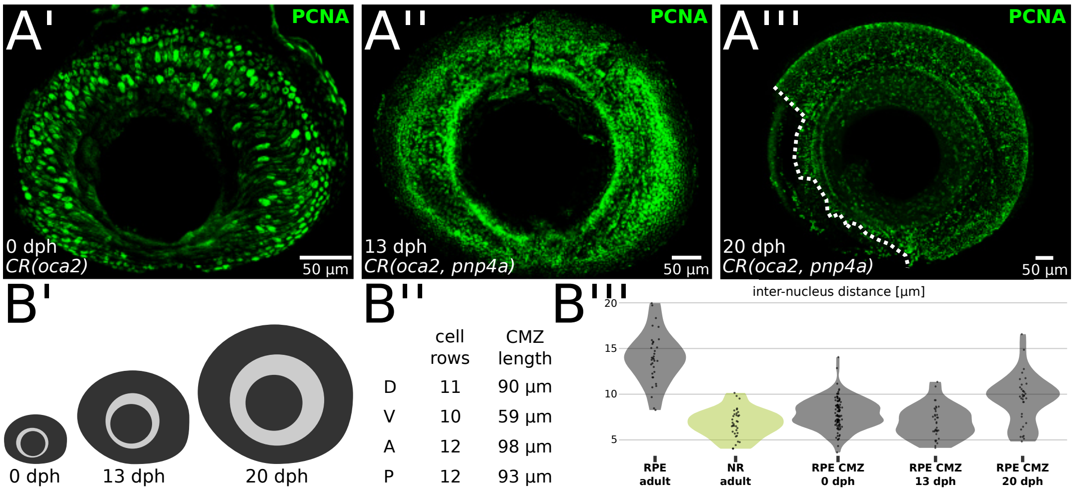


Figure 2.31: Wholemount preparations reveal a complete picture of the RPE CMZ.
A′ Left: Maximum projection of confocal stack of distal view on a wholemount hatchling retina. Right: Maximum projection of orthogonal view of green bracketed area in the left panel. **A″** BrdU and PCNA staining corresponding to A′. **B′–B″** Same sample as in A′–A″, where the area occupied by the NR and the lens was removed by manual masking of the 3D confocal stack. White dotted line: RPE CMZ. Images were rotated and horizontally flipped to place ventral down and anterior to the left. DAPI intensity shown with ImageJ "Fire" LUT.

well (Figure 2.30 B′–B″, yellow arrowheads). This sporadic BrdU incorporation suggested that these cells were competent to proliferate, but the timing of proliferation varied, consistent with my previous clonal and BrdU incorporation data. Henceforth, I use the term "RPE CMZ" for this proliferation-competent RPE region. Interestingly, some distal RPE CMZ cells also stained for rx2 (Figure 2.30 B′). The dorsal RPE CMZ consisted of roughly 8–11 cell rows stretching over a distance of on average 98 µm.

To corroborate these findings and investigate the full circumference of the RPE CMZ, I analysed wholemount preparations. Due to the extremely bright NR cell nuclei, observing the RPE in wholemount samples is technically difficult (Figure 2.31 A′–A″). To overcome this limitation, I manually masked confocal stacks to remove all extraneous tissues, enabling an unprecedented clear view on the entirety of the RPE CMZ (Figure 2.31 B′–B″).

Interestingly, proliferating cell nuclear antigen (PCNA) levels were heterogenous in the RPE CMZ; some cells had low or no staining at all (Figure 2.31 B″). PCNA levels vary throughout the cell cycle; minimal PCNA is expressed during G1 phase, while complete absence of PCNA indicates quiescence [Zerjatke et al., 2017]. Thus, heterogeneity in PCNA staining could result from asynchrony in



	cell rows	CMZ length
D	11	90 µm
V	10	59 µm
A	12	98 µm
P	12	93 µm

Figure 2.32: The RPE CMZ remains roughly equal in size over time.
A′ Distal view on PCNA staining of RPE in 0 dph hatchling. The sample is the same as in Figure 2.31. **A″** Distal view on PCNA staining of the RPE of a 13 day old larva. The image was processed to enhance RPE-specific signal. **A‴** Distal view on PCNA staining of the RPE of a 20 day old larva. White dotted line indicates a region where the RPE was torn off. The image was processed to enhance RPE-specific signal. **B′** Schematic scale drawings of retinae in A′–A‴. RPE and pupil in dark gray. **B″** Average number of cell rows and projected length of RPE CMZ in dorsal (D), ventral (V), anterior (A), and posterior (P) poles of the eye. CMZ length was measured on projected images, and thus the measurements did not account for curvature. Samples analysed: 0 dph: n = 3; 13 dph: n = 1; 20 dph: n = 1. **B‴** Inter-nucleus distance in adult differentiated RPE and NR, and in larval RPE CMZ. First two graphs in the plot are identical to Figure 2.28 B. Data points correspond to individual nuclei pairs.

cell cycle phases, quiescence, or a combination of both.

The average dimensions of the RPE CMZ remained approximately constant over time, with roughly 10–12 rows of cells marked by PCNA (Figure 2.32 A′–A‴, B″). Despite an apparently shorter ventral RPE CMZ, the number of PCNA-positive cell rows was similar (Figure 2.32 B″). This discrepancy suggested a tighter cell packing in the ventral RPE, but may also be due to greater curvature of the retina at the ventral pole, which was not accounted for. In line with the model's prediction, the RPE proliferative population by far exceeded the previous estimate of 5 cell rows.

While differentiated adult RPE cells averaged to 14 µm, cells were smaller in the RPE CMZ, averaging to 8 µm across all investigated stages (Figure 2.32 B‴). The spread of the distribution partly resulted from the ellipsoid shape of RPE CMZ nuclei. Across all investigated stages, RPE CMZ nuclei had on average a major axis of 9.3 µm and a minor axis of 5.5 µm, with the longer axis usually oriented along the peripheral–central coordinate except for the peripheral-most cells. In contrast, nuclei of differentiated RPE cells were circular. Thus, cells in the RPE CMZ and in the differentiated RPE had different size and shape.

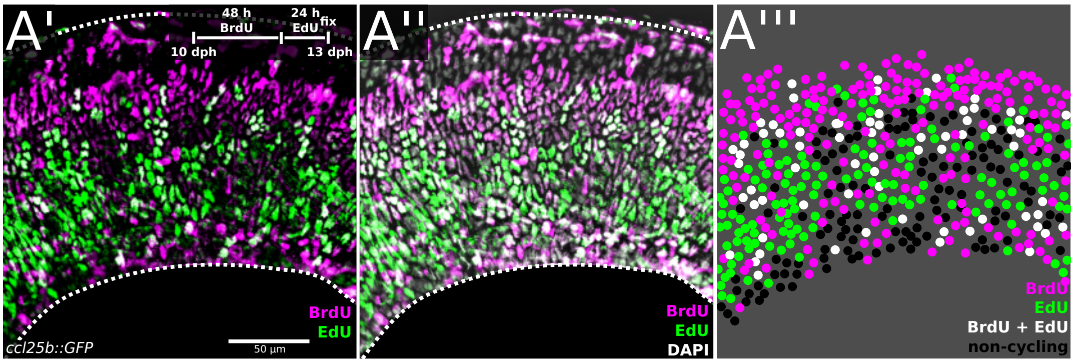


Figure 2.33: Long thymidine analogue pulses reveal quiescent cells in the RPE CMZ. **A′–A″** Distal view on RPE CMZ. Sum projection of confocal stack that was manually masked to remove as much extraneous tissue as possible. Dotted lines delineate border of RPE. The sample was not oriented. **A‴** Schematic distribution of BrdU-positive, EdU-positive, double-positive, and double-negative cells.

2.4.5 Characterising proliferation parameters in the RPE CMZ

Cell cycle dynamics are globally heterogeneous in the RPE CMZ

Parametrisation of $t_{\text{cellCycle}}$ and p_{div} could help discriminate memoryless and memoried quiescence (Figure 2.29). I experimentally determined cell cycle length (which should be $\approx t_{\text{cellCycle}}$) with the method of Das et al. [2009]. After subsequent BrdU and EdU pulses I counted PCNA-positive cells incorporating these markers in the RPE CMZ. Data from a single hatchling retina (see Figure 2.31 B″) gave an average cell cycle of 69 h, with an S phase of 13 h. Restricting p_{div} without time-resolved data is more difficult, but can be tackled by thymidine analogue pulse-chase experiments. Assuming independence of intrinsic variability in cell division timing (which is inversely proportional to p_{div}) and quiescence, there are four possibilities: First, no quiescence and low variability; second, no quiescence and high variability; third, quiescence and low variability; fourth, quiescence and high variability.

The first scenario is easiest to exclude: A population cycling with low variability (p_{div} large or equal to 1) and no quiescence should incorporate thymidine analogues near-uniformly with a pulse on the order of $t_{\text{cellCycle}}$. To test this scenario, I performed subsequent pulses of BrdU and EdU for a total of 72 h. Consistent with the spatiotemporal growth order of the retina, BrdU signal was strong centrally in the first few presumably differentiated cells (Figure 2.33). Peripherally to this band of strongly BrdU-positive cells is the RPE CMZ proper, where large clusters of EdU-positive cells and smaller BrdU-positive cell clusters were distributed (Figure 2.33). Notably, several cells lacked BrdU and EdU, but could be clearly identified by DAPI, excluding artifacts due to tissue damage

(Figure 2.33 A″). Interestingly, non-cycling cells also clustered (Figure 2.33 A‴). Beyond a tendency for peripheral location, there was no clear pattern for non-cycling clusters. Interestingly, double-labelled cells – likely sister cells – tended to orient along the peripheral–central axis (Figure 2.33 A‴).

These data show that the population of RPE SCs cycles heterogeneously and thus definitely exclude the first scenario of no quiescence and low cell division variability. Curiously, cell subpopulations segregate into clusters in the RPE CMZ, which may indicate that local cell groups have similar dynamics.

Cell cycle dynamics are locally homogeneous in the RPE CMZ

To further restrict p_{div} and extract temporal information indirectly, I exploited differential label retention dynamics. Specifically, 0 dph fish were treated overnight in BrdU, grown for 10 days, incubated overnight in EdU, and then chased for a further 8 or 14 days. These experiments revealed heterogeneous label retention within the RPE CMZ (Figure 2.34 A′–D″). Strong BrdU label persisted in some cells (Figure 2.34 A′–D″ pink asterisks), while other cells were double-labelled by BrdU and EdU (Figure 2.34 A′–D″ white arrowheads). BrdU label was almost completely diluted after 18–24 days of chase, on average 6–8 divisions (Figure 2.34 A′–D″). It follows that cells retaining the label must have completed less cycles. A similar argument holds for EdU, indicating that this heterogeneity persisted over time. Thus, cells in the RPE CMZ cycled at different rates, which could be due to high intrinsic cell cycle length variability alone (small p_{div}), quiescence alone, or both.

Many label-retaining cells were at the very periphery of the RPE CMZ, even after the longest chase (on average 8 divisions for BrdU, 4.7 divisions for EdU; Figure 2.34 D′–D″, pink asterisk and white arrowhead). Beyond the very periphery, label retaining cells were not randomly distributed but rather formed string-like trails in locally restricted domains along the peripheral–central axis, akin to the preferential orientation of sister cells. Thus, many trails might be cell families.

Intriguingly, trails were indicative of locally correlated cell cycle timing. For example, cell trails double-labelled by BrdU and EdU (Figure 2.34 A′–B″ white arrowheads) had a history of correlated division during both BrdU and EdU pulses, but overall less cycles than the population average to retain both labels. As cell trails might relate to clones, the low within-group difference suggested a low intrinsic cell cycle length variability, *i.e.* a large value of p_{div}. As large p_{div} with no quiescence was excluded before, these data supported a scenario where quiescence played the major role in driving heterogeneity in the RPE CMZ.

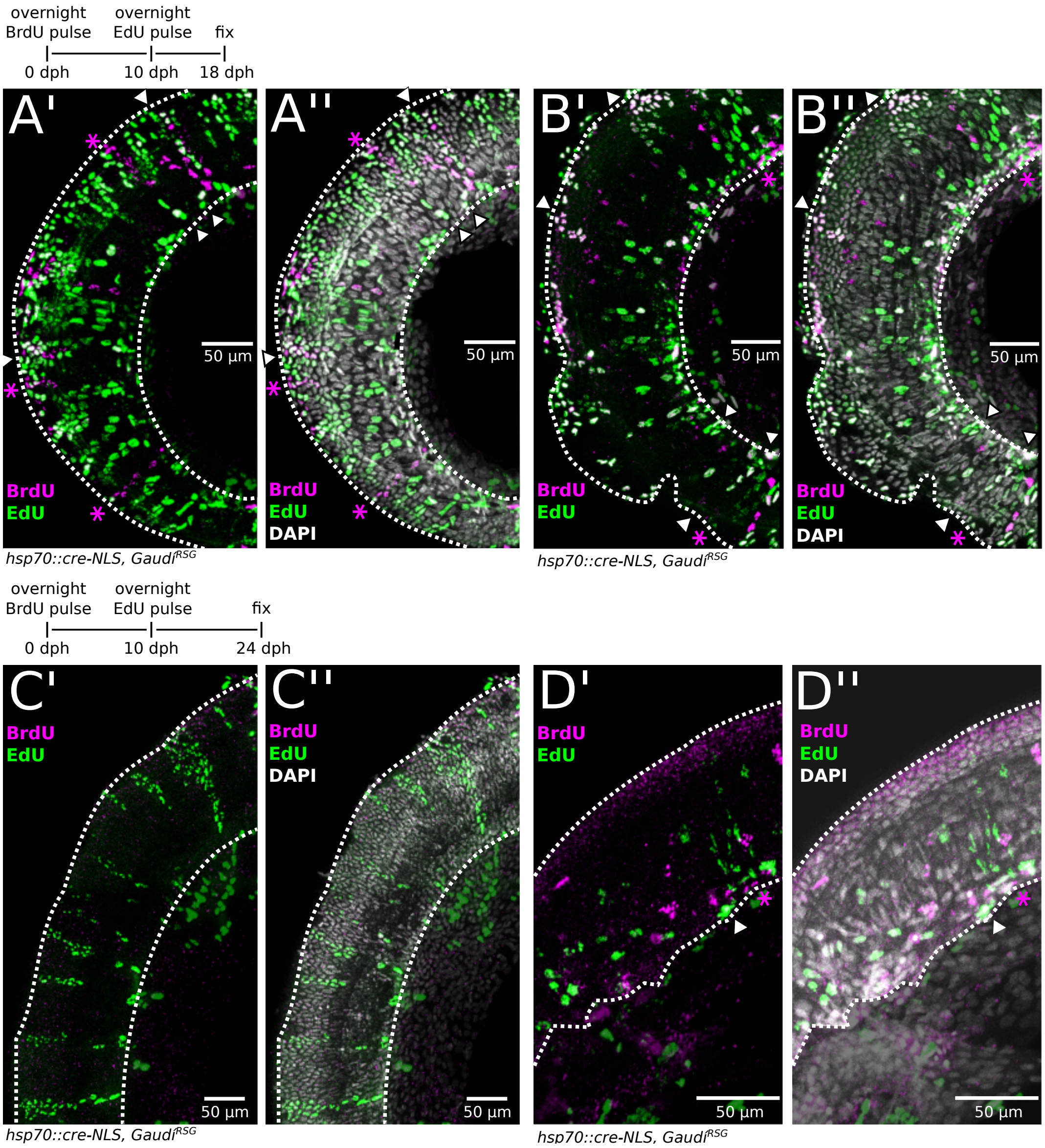


Figure 2.34: Label-retaining sectors form in the RPE CMZ within a week of chase. **A′–B″** Retina from a fish treated with BrdU at 0 dph, EdU at 10 dph, and fixed at 18 dph. A′–A″ and B′–B″ show different details of the same sample. **C′–D″** Retina from a fish treated with BrdU at 0 dph, EdU at 10 dph, and fixed at 24 dph. C′–C″ and D′–D″ show different details of the same sample. All samples show distal views of maximum or sum projections of manually masked confocal stacks. Dotted lines: border of RPE; pink asterisks: selected BrdU-only cells; white arrowheads: selected double-positive cells. Samples were not oriented.

Nevertheless, alternative scenarios could not completely be excluded. Further, the mechanism underlying quiescence remained unclear. To gain further insight, I simulated thymidine analogue incorporation using experimentally restricted parameters for cell cycle time ($t_{\mathrm{cellCycle}} = 70$ h), CMZ width ($w = 90$ µm), and CMZ cell size ($r = 3.5$ µm). I performed two subsequent thymidine analogue pulses for 35 h each (total pulse 70 h), and varied p_{div} (Figure 2.35). As expected, when p_{div} was large, cells divided near-synchronously and nearly all cells were labelled (Figure 2.35 A′, yellow arrowhead: synchronously dividing

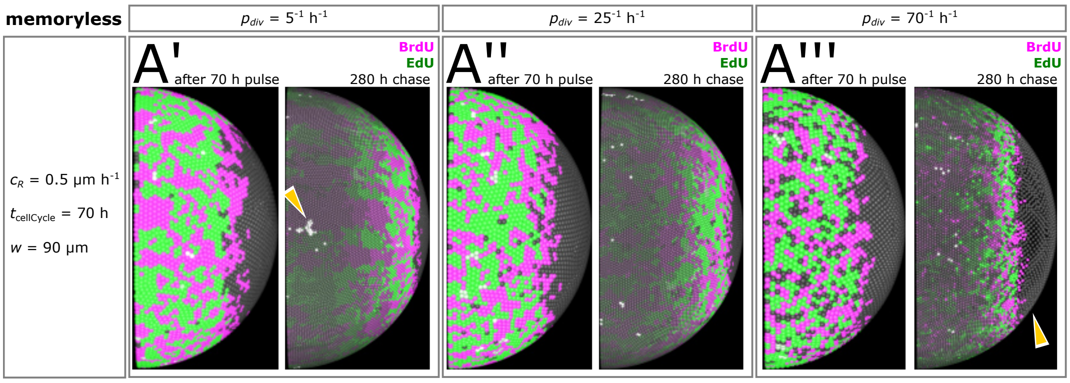

memoryless
p_{div} = 5^{-1} h^{-1}
p_{div} = 25^{-1} h^{-1}
p_{div} = 70^{-1} h^{-1}
c_R = 0.5 µm h^{-1}
$t_{\mathrm{cellCycle}}$ = 70 h
w = 90 µm
A'
A''
A'''
after 70 h pulse
280 h chase
BrdU
EdU

Figure 2.35: Simulations of the memoryless model with a long thymidine analogue pulse. **A′–A‴** Left: Simulation after 70 h consecutive BrdU and EdU pulses. Right: Simulation after 280 h chase. Left and right not to scale. **A′** Simulation with low variability in cell division timing ($p_{\mathrm{div}} = 5\,\mathrm{h}^{-1}$). Yellow arrowhead: Cluster of near-synchronously dividing cell family. **A″** Simulation with intermediate levels of variability in cell division timing ($p_{\mathrm{div}} = 25\,\mathrm{h}^{-1}$). **A‴** Simulation with high levels of variability in cell division timing ($p_{\mathrm{div}} = 70\,\mathrm{h}^{-1}$). Yellow arrowhead: Inter-cell gaps due to cell density drop. Unless noted, parameters were as in Tables 2.1 and 2.2.

cell family). As p_{div} decreased and division timing variability increased, the number of unlabelled CMZ cells also increased (Figure 2.35 A‴). However, with $t_{\mathrm{cellCycle}} = 70$ h cells failed to proliferate at an adequate rate to cover the hemisphere (Figure 2.35 A‴, yellow arrowhead), an effect that was exacerbated in the memoried quiescence model as quiescence further decreased proliferation rate. As low cell density affected quiescence, this parameter range was outside the scope of the present model. In conclusion, a more detailed model building upon the experimental characterisation of the RPE presented in this work will be needed to discern between alternative mechanisms of quiescence.

Summary

The RPE CMZ had a large population of proliferative cells that was globally heterogeneous, but local – potentially clonal – cell clusters displayed temporally homogeneous dynamics. Peripheral-most cell rows appear to have a higher rate of quiescence, which might be predetermined or an emergent property. Fully elucidating this complex dynamic system will require quantitative comparisons of experimental and simulated data. The computational model was built primarily based on findings in the NR – developments towards a comprehensive model of RPE SC dynamics will build upon the foundation laid by the experimental characterisation of the RPE CMZ in this work.

2.5 Elucidating the post-embryonic clonal origin of the medaka NVR

In my modelling of NR and RPE, I placed a border of "obstacle cells" as a biomechanical roadblock at the very periphery of the CMZ. *In vivo*, the CMZ forms a continuous epithelium with the poorly characterised bilayered NVR epithelium (Figure 2.36 A′). The NVR has been described to grow post-embryonically in fish, being essentially absent in the eye of hatchlings [Soules and Link, 2005]. This post-embryonic emergence raised several tantalising questions:

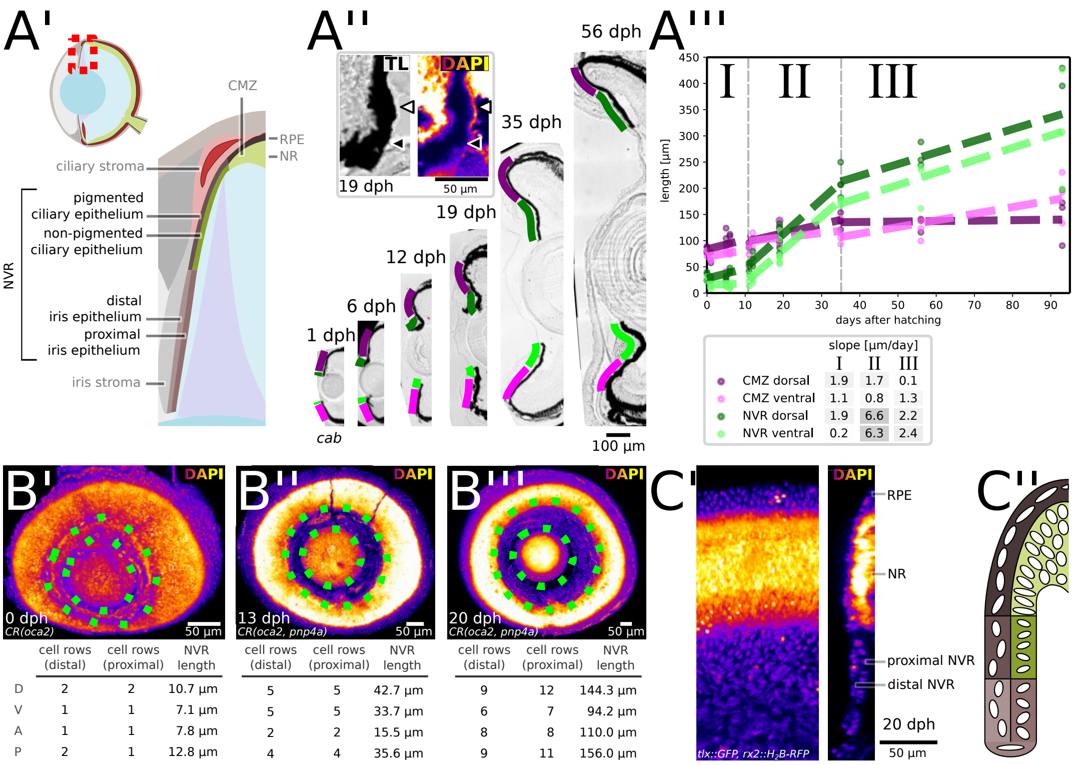
- Does this tissue arise from a fate-restricted NVR primordium or do the very same CMZ cells that give rise to NR or RPE also contribute to parts of the NVR?
- If CMZ cells contribute to the NVR, does this clonal relationship persist or does the NVR establish its own niche after it emerges?
- Can NVR cells contribute to the NR or RPE or are they fate-restricted?

In the following, I address these questions experimentally.

2.5.1 Growth of the medaka NVR

In collaboration with Mai Thu Nguyen, I characterised the growth kinetics of the medaka NVR in cryosections. In early stages, the NVR could not be clearly identified; a morphologically thicker pigmented epithelium distal to the CMZ might represent the NVR primordium (Figure 2.36 A″). At 19 dph, the NVR first became visible as a bilayered epithelium clearly distinct from the CMZ (Figure 2.36 A″). Consistent with descriptions of teleost NVR anatomy [Walls, 1942], cells in the proximal layer adjacent to the NR were unpigmented, while the rest was pigmented (Figure 2.36 A″, inset). The NVR grew rapidly until roughly 35 dph (Figure 2.36 A″–A‴). Thereafter, growth continued at a reduced rate (Figure 2.36 A‴). In contrast, the CMZ (measured as the distance from the NR SCs to the beginning of the laminated NR) increased only little in size over time (Figure 2.36 A‴).

In cryosections, the NVR was often bent or otherwise damaged and individual nuclei were difficult to spot even in bleached samples. Thus, to better investigate the morphology of the NVR, I examined wholemount retinae at different ages (Figure 2.36 B′–B‴). The wholemount data confirmed a morphologically distinct 1–2 cell wide bilayered NVR primordium between the CMZ and lens at hatchling stage (Figure 2.36 B′). The NVR primordium grew to a clearly distinct structure with up to 5 cell rows by 13 dph, and 12 cell rows by 20 dph (Figure



	slope [µm/day] I	II	III
● CMZ dorsal	1.9	1.7	0.1
● CMZ ventral	1.1	0.8	1.3
● NVR dorsal	1.9	6.6	2.2
● NVR ventral	0.2	6.3	2.4

B' (0 dph)

	cell rows (distal)	cell rows (proximal)	NVR length
D	2	2	10.7 µm
V	1	1	7.1 µm
A	1	1	7.8 µm
P	2	1	12.8 µm

B'' (13 dph)

	cell rows (distal)	cell rows (proximal)	NVR length
D	5	5	42.7 µm
V	5	5	33.7 µm
A	2	2	15.5 µm
P	4	4	35.6 µm

B''' (20 dph)

	cell rows (distal)	cell rows (proximal)	NVR length
D	9	12	144.3 µm
V	6	7	94.2 µm
A	8	8	110.0 µm
P	9	11	156.0 µm

Figure 2.36: Growth and anatomy of the medaka NVR.
A′ Scheme of the NVR anatomy. **A″** Representative images of cryosections of eyes of fish of different age. The extent of the dorsal and ventral NVR and CMZ have been labelled with colours corresponding to data in A‴. Inset: Detail of NVR in a 19 dph retina. White arrowhead: unpigmented proximal layer, black arrowhead: pigmented proximal layer. Experiment performed by Mai Thu Nguyen. **A‴** Length of dorsal and ventral NVR and CMZ at different timepoints after hatching measured in central sections of fish eyes. Data provided by Mai Thu Nguyen. The data were manually split into three groups for fitting of piecewise linear function; fitted slopes of data series are given in the legend. **B′–B‴** Maximum projections of confocal stacks showing distal view on wholemount medaka retinae at different ages. Green dotted line marks boundary of visual retina and NVR. Number of cell rows and length of the NVR are given for the respective stages; measurements done on one retina each. The samples are the same as shown in Figure 2.32. **C′** Left: Maximum projection of confocal stack showing detail of distal view on the dorsal part of a 20 dph retina. Right: Orthogonal projection of confocal stack, showcasing distal and proximal layers of the NVR. **C″** Scheme of relative size and shape of nuclei in the CMZ and NVR. DAPI intensity shown with ImageJ "Fire" LUT.

2.36 B″). Mirroring the asymmetry of the rest of the retina, the ventral NVR was shorter and contained less cells at all stages that were examined. Additionally, the posterior NVR tended to be longer than the anterior (Figure 2.36 B″).

Nuclei in the NVR had different sizes; the putative proximal ciliary epithelium (adjacent to the NR) had smaller rounder nuclei, which progressively flattened towards the lens (Figure 2.36 C′). The loop at the tip of the NVR frequently had a cell with an elongated nucleus spanning both layers. The distal NVR had flattened nuclei larger than the proximal NVR but smaller than RPE cells (Figure 2.36 C″).

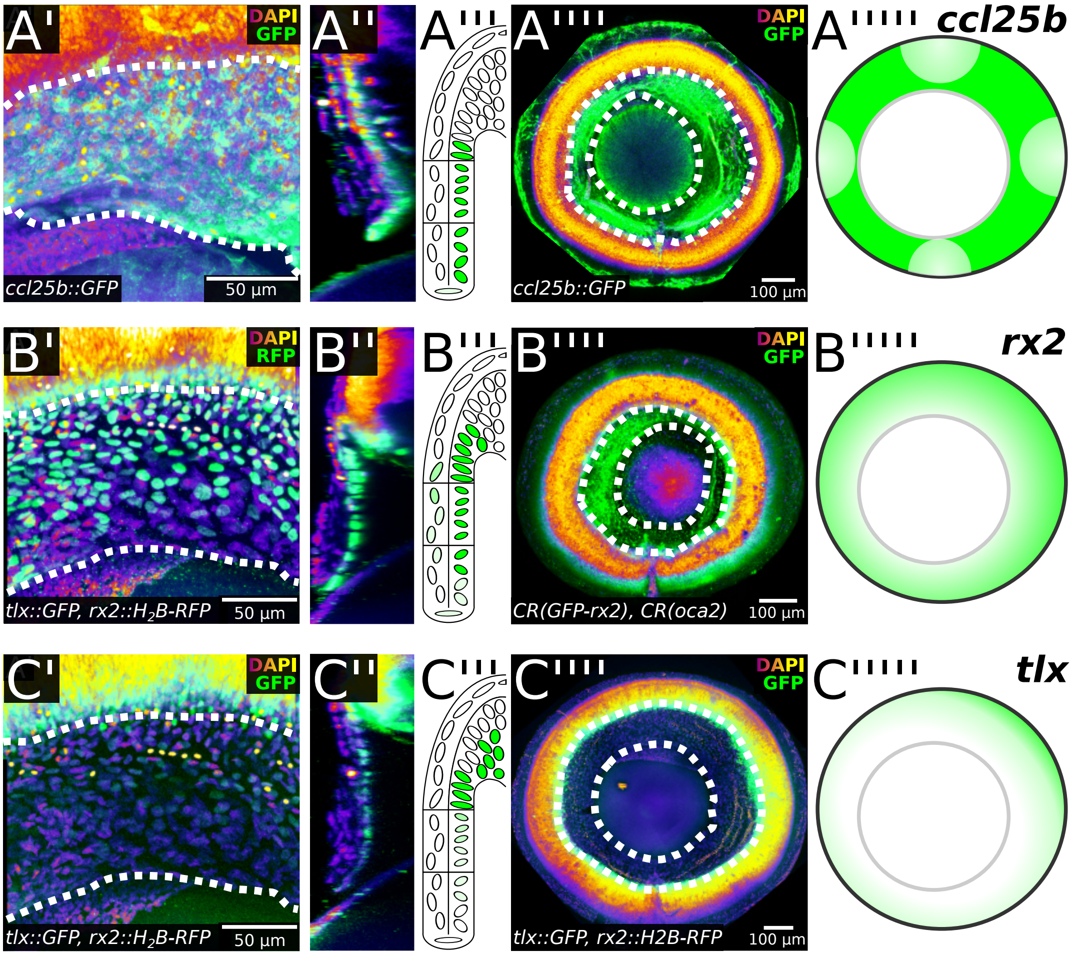


Figure 2.37: The proximal NVR shares markers with NR SCs.

A′ Maximum projection on detail of distal view of medaka retina at 20 dph. Sample was not oriented. **A″** Orthogonal projection of a subset of A′. **A‴** Schematic cross-section of the NVR and CMZ depicting *ccl25b* expression domain. **A⁗** Distal view of wholemount retina at 20 dph. Sample oriented to place ventral down and anterior to the left. **A′′′′′** Schematic circumferential expression pattern of *ccl25b* in the NVR. **B′** Maximum projection on detail of distal view of medaka retina at 20 dph. Sample was not oriented. **B″** Orthogonal projection of a subset of B′. **B‴** Schematic cross-section of the NVR and CMZ depicting *rx2* expression domain; expression is faintly visible in some nuclei of the distal NVR and RPE. **B⁗** Distal view of wholemount retina at 20 dph. Sample oriented to place ventral down and anterior to the left. **B′′′′′** Schematic circumferential expression pattern of *rx2* in the NVR. **C′** Maximum projection on detail of distal view of medaka retina at 20 dph. Same sample as B'. **C″** Orthogonal projection of a subset of C′. **C‴** Schematic cross-section of the NVR and CMZ depicting *tlx* expression domain. **C⁗** Distal view of wholemount retina at 20 dph. Sample oriented to place ventral down and anterior to the left. **C′′′′′** Schematic circumferential expression pattern of *tlx* in the NVR. DAPI intensity shown with ImageJ "Fire" LUT.

2.5.2 The proximal NVR shares marker expression with the NR SC

Interestingly, the proximal layer of the NVR expressed markers found in NR SCs such as *ccl25b* (Figure 2.37 A′–A′′′′′), *rx2* (Figure 2.37 B’-B′′′′′), and to a lesser extent *tlx* (Figure 2.37 C′–C′′′′′).

ccl25b was expressed in all proximal NVR cells at comparable levels to NR SCs, and more weakly in the "loop" cell (Figure 2.37 A′′′). Similar to the expression pattern observed in the CMZ, *ccl25b* expression was weaker (but not absent) at the poles (Figure 2.37 A′′′′–A′′′′′).

rx2 was strongly expressed in the putative proximal ciliary epithelium; expression levels decreased towards the lens (Figure 2.37 B′). Careful examination of orthogonal projections revealed that some distal NVR cells and the putative first few cells of the RPE also expressed low levels of *rx2* (Figure 2.37 B′′-B′′′). There was no apparent difference around the retinal circumference (Figure 2.37 B′′′′–B′′′′′).

tlx was faintly expressed in the proximal NVR, visible only by greatly increasing the contrast (Figure 2.37 C′–C′′′). At the dorso-posterior retina, however, there was clear *tlx* expression in the first few proximal NVR cells (Figure 2.37 C′′′′–C′′′′′), which mirrors the asymmetric distribution of *tlx* in the NR CMZ.

2.5.3 The NVR is a proliferative epithelium that is clonally related to the NR and the RPE

Already at hatchling stage the NVR appeared as a morphologically distinct primordium abutting the lens and connecting NR and RPE. During post-embryonic development, this primordium continuously grew in size by cell addition. To investigate the origin of these new cells and elucidate the relationship of the NVR with NR and RPE, I used clonal analysis.

I re-analysed data previously obtained in the lab by Dr Colin Lischik using the lines *tlx::*^{ERT2}*cre, Gaudí*^{RSG} and *ccl25b::*^{ERT2}*cre, Gaudí*^{RSG}, which were induced at 0 dph and chased for 14 or 28 days. In both lines, almost every NR clone was spatially contiguous to proximal NVR clones (Figure 2.38 A′, B′–B′′, C′–C′′, D′). Interestingly, terminated NR clones that were no longer connected to the CMZ lacked a corresponding NVR clone (Figure 2.38 A′, asterisks). Additionally, there were NVR-only clones not contiguous to any NR clones (Figure 2.38 A′–A′′, B′, D′–D′′). These data suggested that the NR and proximal NVR were at least partially clonally related.

Due to high labelling frequency, the possibility remained that spatially corre-

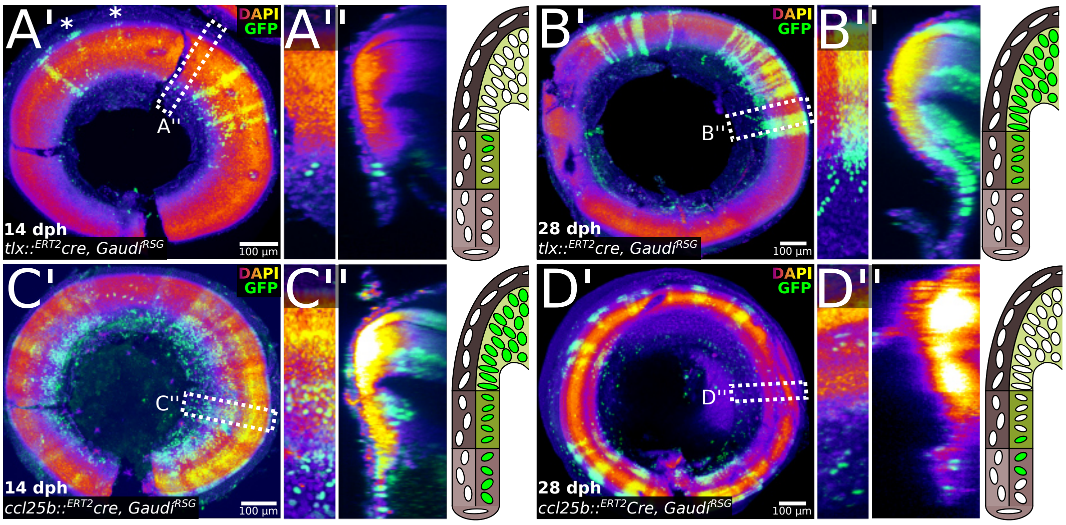


Figure 2.38: Both tlx and ccl25b create proximal NVR clones contiguous with NR clones. **A′** Maximum projection on distal view of medaka retina induced at 0 dph and chased for 14 days. Asterisks: NR clones not connected to the CMZ. **A″** Left: Magnification of boxed area in A′. Middle: Orthogonal projection of magnified image. Right: Schematic interpretation. **B′** Maximum projection on distal view of medaka retina induced at 0 dph and chased for 28 days. **B″** Left: Magnification of boxed area in B′. Middle: Orthogonal projection of magnified image. Right: Schematic interpretation. **C′** Maximum projection on distal view of medaka retina induced at 0 dph and chased for 14 days. **C″** Left: Magnification of boxed area in C′. Middle: Orthogonal projection of magnified image. Right: Schematic interpretation. **D′** Maximum projection on distal view of medaka retina induced at 0 dph and chased for 28 days. **D″** Left: Magnification of boxed area in D′. Middle: Orthogonal projection of magnified image. Right: Schematic interpretation. Experiments and image acquisition performed by Dr Colin Lischik. DAPI intensity shown with ImageJ "Fire" LUT.

lated NVR–NR clones were polyclones. To exclude this scenario, I took advantage of the *hsp70::cre-NLS, Gaudí^{RSG}* line, which recombines very poorly after fish hatch, to generate sparsely labelled samples. Additionally, the heat shock promoter allowed to unbiasedly target every single cell type. I induced the animals at the day of hatching, and chased for 18 or 24 days (Figure 2.39).

Clones restricted to the NR were always disconnected from the CMZ periphery (Figure 2.39 A). Similarly, RPE-only clones also emerged more distally in the niche (Figure 2.39 B). Clones purely confined to one NVR layer were also present (Figure 2.39 C). Further, I identified clones spanning proximal NVR and NR (Figure 2.39 D), as well as distal NVR and RPE (Figure 2.39 E). In one case, a clone appeared to connect distal and proximal NVR (Figure 2.39 F). In summary, this set of experiments revealed a multitude of clone types, which were highly likely to have been generated by a single recombined cell. These data support a model of continuous clonality between NR, proximal NVR, distal NVR, and RPE.

To investigate whether the spatial correlation between NR and proximal NVR clones persisted after NVR maturation, Mai Thu Nguyen and I performed *cre* inductions at various timepoints after hatching. Again, both *ccl25b* and *tlx*

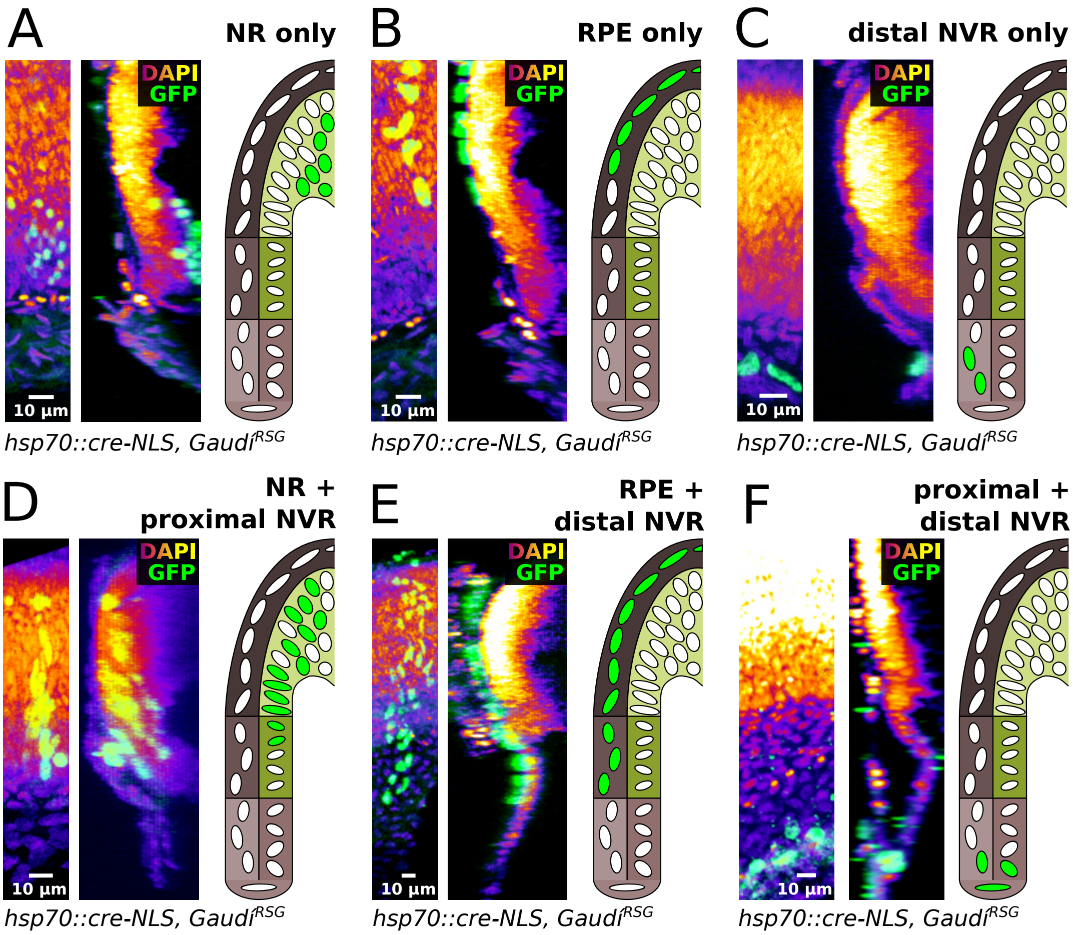

A
NR only
DAPI
GFP
10 µm
hsp70::cre-NLS, Gaudi^{RSG}
B
RPE only
DAPI
GFP
10 µm
hsp70::cre-NLS, Gaudi^{RSG}
C
distal NVR only
DAPI
GFP
10 µm
hsp70::cre-NLS, Gaudi^{RSG}
D
NR + proximal NVR
DAPI
GFP
10 µm
hsp70::cre-NLS, Gaudi^{RSG}
E
RPE + distal NVR
DAPI
GFP
10 µm
hsp70::cre-NLS, Gaudi^{RSG}
F
proximal + distal NVR
DAPI
GFP
10 µm
hsp70::cre-NLS, Gaudi^{RSG}

Figure 2.39: Various combinations of uni- and bipotent clones in sparsely labelled retinae.
A NR-only clone. Left: Detail of maximum projection on distal view of medaka retina induced at 0 dph and chased for 18 days. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. **B** RPE-only clone. Left: Detail of maximum projection on distal view of medaka retina induced at 0 dph and chased for 18 days. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. **C** Distal NVR-only clone. Left: Detail of maximum projection on distal view of medaka retina induced at 0 dph and chased for 18 days. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. **D** Hybrid NR-proximal NVR clone. Left: Detail of maximum projection on distal view of medaka retina induced at 0 dph and chased for 18 days. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. **E** Hybrid RPE-distal NVR clone. Left: Detail of maximum projection on distal view of medaka retina induced at 0 dph and chased for 24 days. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. **F** Hybrid proximal-distal NVR clone. Left: Detail of maximum projection on distal view of medaka retina induced at 0 dph and chased for 24 days. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. DAPI intensity shown with ImageJ "Fire" LUT.

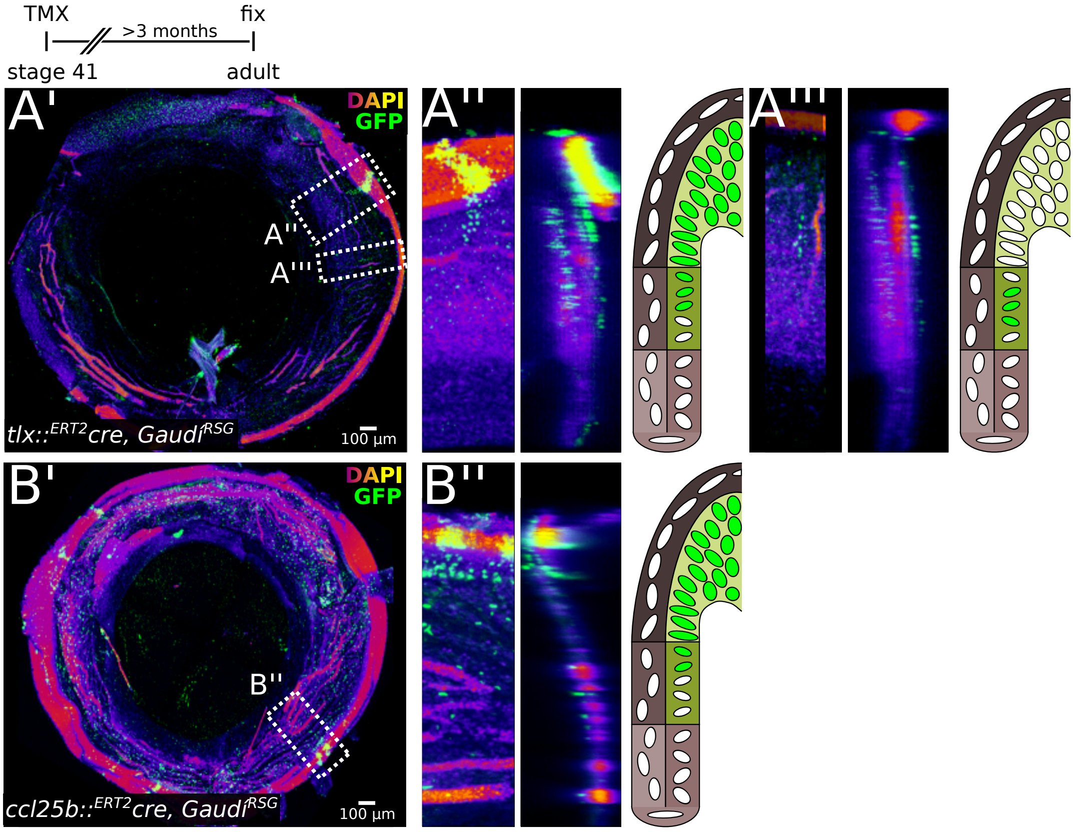


Figure 2.40: Hybrid NR-proximal NVR clones emerge even at later induction timepoints. **A′** Maximum projection of proximal view on dissected medaka NVR and CMZ. Induction was at 12 dph. Sample was oriented to place ventral down and anterior to the left. **A″** Left: Detail of A′. Middle: Orthogonal projection of image on left. Image was horizontally flipped to place distal to the left. Right: Schematic interpretation. **A‴** Left: Detail of A′. Middle: Orthogonal projection of image on left. Image was horizontally flipped to place distal to the left. Right: Schematic interpretation. **B′** Maximum projection of distal view on medaka retina. Induction was at 22 dph. Sample was oriented to place ventral down and anterior to the left. **B″** Left: Detail of B′. Middle: Orthogonal projection of image on left. Right: Schematic interpretation. DAPI intensity shown with ImageJ "Fire" LUT.

promoters driving *cre* expression generated spatially correlated NR-NVR clones (Figure 2.40 A′–B″). As expected from marker expression, the proximal NVR also contained small isolated clones (Figure 2.40 A‴, B′). Thus, the data strongly suggested that proximal NVR and NR, as well as distal NVR and RPE were clonally related. For the NR and proximal NVR, the clonal relationship held true throughout the animal's life. The observation of a distal-proximal NVR clone suggested that all retinal tissues were clonally related in principle, but effectively unlikely to form a large contiguous clone.

Importantly, these experiments could not resolve the cell of origin of hybrid clones. If NVR cells were the origin of clones, they must have the capacity to proliferate. Indeed, NVR cells at 20 dph readily incorporated BrdU and EdU, and stained positive for PCNA in a dispersed pattern in both layers (Figure 2.41). No

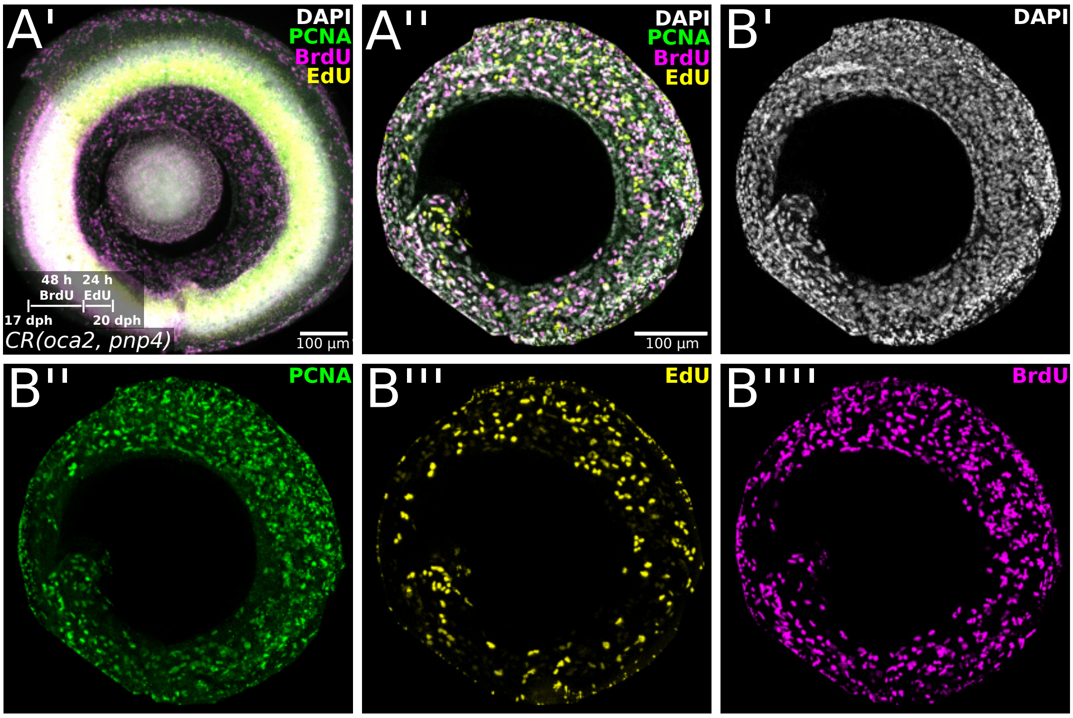


Figure 2.41: The NVR is a proliferative epithelium.
A′ Maximum projection of distal view of a medaka retina. Same sample as in Figure 2.32 A‴. **A″** Detail of NVR from A′ after manually masking all extraneous tissue. **B′** 4',6-diamidino-2-phenyindole (DAPI) staining of A″. **B″** PCNA staining of A″. **B‴** EdU staining of A″. **B⁗** BrdU staining of A″.

particular "hotspot" for proliferation emerged. Using the method described in Das et al. [2009], the cell cycle length at 20 dph was estimated to be 73.3 h, with an S-phase duration of 23.3 h.

Summary

The data show that a bilayered NVR primordium initially present in hatchlings grew over time. All three retinal tissues – NR, RPE, and NVR – were clonally related. This clonal relation was retained at later larval stages, as shown for NR and NVR clones. As cells distributed throughout the NVR proliferated, the origin of clones could not be pinpointed precisely in the retrospective clonal analysis. Time-resolved data or induction using markers exclusively expressed in the NVR or NR and RPE will be key to address this point.

3 Discussion

In this work, I combined computational and experimental approaches to address how SCs of the three constituent tissues of the retina – NR, RPE, and NVR – concertedly grow during the post-embryonic life of medaka:

- I designed a 3D cell centred agent based model that recapitulated basic features of clonal growth in the NR and RPE, reconciling the apparent contradiction of neutral drift with a stable clone number (Chapter 2 Section 2.1).

- Quantitative comparisons between simulated and experimental data demonstrated that NR and RPE followed distinct modes of growth to coordinate their growth rates, wherein the NR acted upstream to induce growth of the RPE (Chapter 2 Section 2.2).

- I further leveraged the simulation to narrow down how NR SCs in the CMZ might modulate their proliferative parameters to impinge on organ shape and retinal topology (Chapter 2 Section 2.3).

- Further, model predictions instigated an in-depth characterisation of homeostasis in the RPE niche, where I uncovered heterogeneous SC behaviour between potential clonal lineages driven by quiescence (Chapter 2 Section 2.4).

- Finally, I characterised the growth kinetics and clonality of the NVR, and showed that this tissue shared common markers and clonal relationship with the NR and RPE (Chapter 2 Section 2.5).

In the following, I discuss selected findings of this thesis.

3.1 System properties are highlighted by modelling

3.1.1 Neutral drift is compatible with a stable clone number

It had been proposed that SCs in the NR and RPE display deterministic invariant asymmetry based on the long-term stability of clonal number in these tissues [Centanin et al., 2014, 2011]. This experimental observation stood in contrast with the stochastic neutral drift model, which proposed that SC niches experience a progressive loss in clonality ultimately culminating in monoclonality [Clevers and Watt, 2018; Colom and Jones, 2016]. The model developed in this work reconciles these two views by highlighting the importance of temporal dynamics and 3D geometry of the tissue under consideration.

The model gives a two-fold explanation: First, growth increases the area of the niche, thus reducing the impact of competition on clonal loss, but without completely abrogating it. Second, radial growth in particular has the property that it preserves clonality. Indeed, mathematical modelling of radial expansion of populations in the context of evolutionary theory has proven that under neutral genetic drift a finite number of subpopulations or clones perpetually coexist [Hallatschek and Nelson, 2010; Korolev et al., 2012]. Thus, the 3D geometry and continuous growth of the retina intrinsically entail clonal coexistence.

3.1.2 Abstraction to one cell layer reduces the parameter space.

Several properties of the NR and RPE permit to make simplifying assumptions. One such assumption was the abstraction of the multiple NR cell layers to a model with a single layer of spherical cells. Including the fact that PCs cycle faster than SCs in this model clearly disrupted clonal growth, as PCs outcompeted SCs and produced small disconnected clonal patches unlike those observed experimentally (Figure 2.8). Conceptually, addition of differentiated cell layers creates more choices for the placement of a daughter cell of a dividing PC, thus effectively reducing the competition between SCs and PCs (Figure 3.1 A, B). According to this model, the probability to displace (and thus outcompete) a SC diminishes with increasing number of neuronal layers. PCs may have evolved to cycle faster to populate the neuronal cell layers [Saturnino et al., 2018]. As the number of layers decreases, so does the need for transit-amplification and faster PC cycling. In the limit of a single-layered tissue, no transit-amplification is required, and thus SC and PC populations become one and the same. Thus, abstracting the NR as a single-layered tissue has the advantage to forego dis-

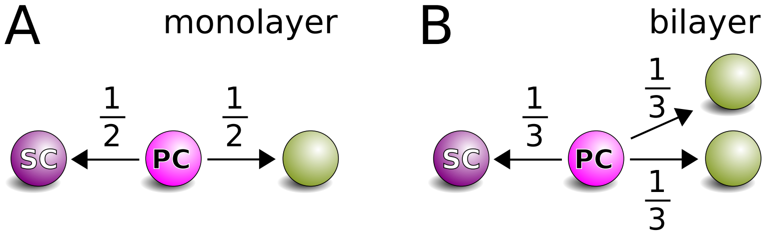


Figure 3.1: Conceptual model of competition reduction with increasing retinal layers. **A** If differentiated cells are a monolayer, then a dividing PC has equal chances of displacing its SC neighbour or differentiated cell neighbour. **B** If differentiated cells arrange as a bilayer, then there is a higher cumulative chance for a dividing PC to place a daughter cell in the differentiated domain, effectively reducing competition with the stem cell compartment.

tinct SC and PC populations with individually parametrised proliferation rates, reducing the parameter space.

3.1.3 The CMZ extent places a self-limiting bound to cell number.

The fixed width of the CMZ, which I confirmed experimentally (Figures 2.32 and 2.36), had the interesting consequence that it prevented exponential growth of the proliferative population in the simulation by limiting the number of SCs in the system (Section 2.1.6). Excessive cells were physically extruded from the domain of proliferative competence and thus ceased proliferating. This self-limiting property enforced linear growth of the cell population. Similarly, the radius of the eye also grew linearly *in vivo* (Figure 2.6 A).

It is widely assumed that an extrinsic determinant defines the extent of the anamniote CMZ, *e.g.* a local gradient of a diffusible molecule or nutritional cues from nearby blood vessels [Fischer et al., 2014; Tang et al., 2017]. Another unexplored possibility is that a self-regulatory feedback loop among CMZ cells locally controls the extent of the proliferative domain, as in plant stem cell niches [Aichinger et al., 2012]. Indeed, PC fate specification in the CMZ appears to be controlled by such a local feedback mechanism between cells [Saturnino et al., 2018].

3.2 Continuous feedback coordinates retinal growth

3.2.1 Patch properties enable unbiased clonal quantification

There are no standardized methods to quantify spatiotemporal data and compare it to agent based models [Bellaïche, 2016]. Previous work on clonal analysis used absolute size as a measure to evaluate statistical properties of clones [He et al., 2012; Snippert et al., 2010; Wan et al., 2016]. This approach has two deficits: First, spatio-temporal information is lost, particularly in the context of the radial growth order of the retina. Second, clone fragmentation and fusion events pose a challenge, and clonal identity has to be defined manually by the experimenter. I circumvented these issues by focusing the analysis on patches, defined objectively as a connected component of segmented pixels. Further, radial growth of the retina can be exploited to perform a coordinate transform [Höckendorf, 2013] and normalise patch position along the radius to infer temporal information, such as how intermittent cell divisions result in "late arising patches" (Figure 2.18).

For generating NR clones I used *cre*-mediated recombination at hatchling stage, while for RPE clones I used mosaic retinae generated by CRISPR/Cas9, which "induces" clone formation at an earlier timepoint. Because the formation of the CMZ takes place prior to hatch, RPE patches were potentially larger and more prone to fusion events. However, variability in RPE cell division behaviour overshadowed this effect, as very narrow patches still contributed to the distribution of patch width (Figure 2.19 B). Further, the distribution of late arising patches is a robust metric that should not be affected by timepoint of induction. Here, too, the RPE clearly diverged from the other datasets (Figure 2.19 E). Further experimental validation identified an extreme heterogeneity in the proliferative RPE (discussed in Section 3.4), consistent with the data from patch analysis. Thus, patch properties provided a robust and unbiased metric for quantitatively comparing spatial patterns in experimental and simulated clonal data.

3.2.2 Growth modes as a framework for tissue coordination

How different tissues within an organ coordinate their growth rates has seldom been addressed conceptually or experimentally. One reason may be that growth control has been investigated primarily in terms of dimensions of the entire organ as a functional unit in animals that grow to a determinate size [Conlon and Raff, 1999; Droujinine and Perrimon, 2016; Lui and Baron, 2011; Roselló-Díez and Joyner, 2015]. In "catch-up growth" a discrete growth control checkpoint temporarily pauses growth in some organs and tissues to enable synchronous developmental transitions [Conlon and Raff, 1999; Droujinine and Perrimon, 2016; Lui and Baron, 2011; Roselló-Díez and Joyner, 2015]. Conceptually, the responder growth mode resembles a continuous version of catch-up-growth – instead of a single discrete checkpoint, there is constant checkpointing of tissue status feeding back to cell proliferation.

To reproduce the tight packing of the retina, the responder growth mode had to be parametrised such that the area generated by cell proliferation was at least equal to or exceeded the area resulting from radial growth rate ($A_{\text{cells}} \geq A_{\text{eye}}$). In the inducer growth mode, by definition growth rate of the organ and proliferation rate of the cells were coupled such that ($A_{\text{cells}} \approx A_{\text{eye}}$) regardless of parametrisation. Thus, there was a range of parameters where responder and inducer growth modes produced identical results (Figures 2.14 and 2.15). The RPE displayed high cell division and patch shape variability indicative of a responder growth mode, while the NR with its relatively low variability and clearer patch sectoring pattern resembled the inducer growth mode (Figure 2.19).

Though unlikely due to variability in growth dynamics of individual fish [Johns, 1981], clonal variability alone could not completely exclude the possibility that the NR conformed to a "perfect" responder growth mode where cell proliferation parameters were tailored to organ growth rate. Given the self-limiting property of the CMZ extent (Section 3.1.3), an inducer growth mode of the NR would imply that the NR CMZ width would affect eye growth rate (Figure 3.2 A). In support of this hypothesis, medaka double mutants for SC genes *rx2* and its close paralog *rx1* exhibit small-eyed phenotypes [Tavhelidse, 2019]. Though interpretations of these mutants was complicated by the embryonic impact of lack of *rx* genes, conditional alleles have been developed that could be used to test this hypothesis by reducing the number of post-embryonic SCs in a targeted manner [Tavhelidse, 2019]. Similarly, increasing the CMZ's proliferative capacity by experimental modulation of growth factor signalling in the *rx2*-expressing cells lead to overgrowth of the entire eye in medaka [Becker and Wittbrodt, unpublished]. Further, the cascade relay model of emmetropy implies that the defocus signal originating from the NR is relayed from tissue to tissue within the eye [Wallman and Winawer, 2004]. This model was supported by recent findings that the defocus signal was relayed from the RPE to the choroid in zebrafish [Collery and Link, 2018]. Thus, the simplest explanation given the current data is that the NR acts as an inducer tissue upstream of the RPE in the signalling hierarchy coordinating growth rates of different eye tissues (Figure 3.2 B).

More generally, inducer and responder growth modes are a useful abstraction of feedback control between tissues, with potentially universal applicability in multicellular organisms. Evidence for functional partitioning of tissues in inducer and responder roles has been found in the coordinated growth of hair follicles and the underlying fat tissue, where coordination is mediated by hedgehog signalling [Zhang et al., 2016]. Interestingly, hedgehog signalling has been implied to regulate the CMZ in medaka [Reinhardt et al., 2015]. Similarly, intra-organ growth in *Drosophila* appears to be regulated hormonally [Gokhale et al., 2016]. These examples show that in addition to mechanical stimuli, biochemical stimuli can also mediate tissue coordination.

3.3 NR SCs impact growth and form of the medaka eye

3.3.1 Functional CMZ subdivisions are an emergent property

The subdivision of the anamniote CMZ into three domains with peripheral SCs, intermediate multipotent PCs, and central oligopotent PCs is a dogma of the field [Harris and Perron, 1998; Raymond et al., 2006; Shi et al., 2017]. The model prediction and experimental corroboration that SCs produce terminated clones even when using the peripheral–most marker *ccl25b* contradicts the current dogma, and posits that multipotent PCs are likely not functionally distinct from SCs (Figure 3.2 C). Rather, these cells are disadvantaged by their physical position in the niche, resulting in a spatially biased neutral drift (Figures 2.20 and 2.21). Similar dynamics were also observed in ABM of the intestinal crypt [Buske et al., 2011; Dunn et al., 2013; Osborne et al., 2017], and experimentally in time-lapse movies of mouse intestine [Ritsma et al., 2014]. Interestingly, with the popularisation of single cell transcriptomics, it has become increasingly clear that cell identities exist in a continuum of trajectories, and cannot be neatly subdivided into categories [Andrews and Hemberg, 2018]. Thus, both cell-intrinsic factors and the cellular environment are crucial determinants of cell fate.

3.3.2 Predominant radial orientation of cells may shape the eye

The topology of the retina generated a preferential alignment of clones along the central–peripheral axis. However, the stripes observed by induction of *rx2*-driven *cre* were considerably thinner (Figure 2.22). Other *cre* drivers, while not quantitatively evaluated, show good qualitative agreement by inspection (Figure 2.21). Any model parametrisation with a sufficiently large bias in division axis orientation along the radial direction fits the experimental data relatively well, though a pure radial bias generates even thinner ArCoS than experimentally observed (Figure 2.22 panel C″).

Following this observation, I speculated that orientation of clonal sisters (by cell division or rearrangement) in the NR may relate to shaping the hemispherical organ (Figure 3.2 D). Assuming that each new cell incremented the area either in the CMZ or in the differentiated retina, the geometry of the hemisphere

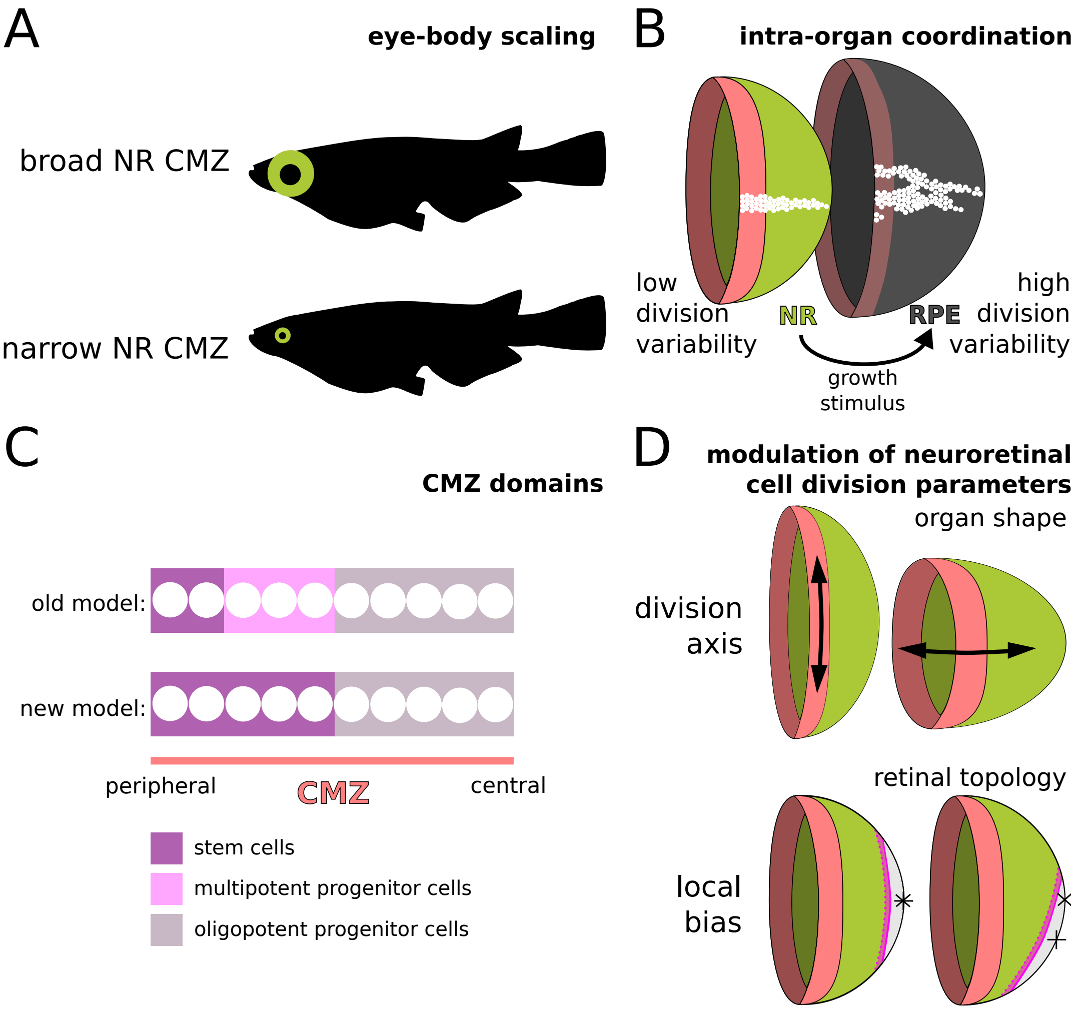


Figure 3.2: Modulation of NR parameters impinges on growth and form of the organ. **A** Assuming all other parameters are equal, a simple increase or decrease in NR CMZ width will accelerate or decelerate eye growth rate, effectively scaling eye size with respect to body size. **B** Model for intra-organ coordination of growth and its impact on cellular and clonal dynamics. **C** Subdivision of domains in the CMZ according to the previous understanding in the field [Shi et al., 2017] and the model proposed in this work [Tsingos et al., 2019]. **D** Modulation of NR CMZ division axis orientation may impact on organ shape. Further, local biases in division parameters along the CMZ circumference affect retinal topology. Panels B and D were adapted from Tsingos et al. [2019].

dictated that radial divisions must predominate and increase over time (Section 2.3.3). Consistent with this prediction, *in vivo* imaging in zebrafish detected an increasing incidence of radial divisions of NR SCs over time [Wan et al., 2016]. Shaping of tissues by cell division orientation has been reported in *Drosophila* [Baena-López et al., 2005]. Further, the NR clearly plays a conserved and central role in regulating eye shape in vertebrates [Wallman and Winawer, 2004]. In chicken, proliferation of the CMZ was indirectly linked to changes in radial and circumferential eye axes in response to visual stimuli [Fischer et al., 2008].

Nevertheless, rather than cause, radial cell orientation could be a downstream consequence of extra-cellular matrix (ECM) remodelling associated with the hemispherical growth of the organ. The changing shape of the eye during emmetropisation is thought to be mediated by remodelling of the sclera, the fibrous tissue ensheathing the eye [Wallman and Winawer, 2004]. There is ample evidence that external forces affect cell division axis orientation [Nestor-Bergmann et al., 2014], and anisotropic forces along radial and circumferential axes might exist in the fish retina [Salbreux et al., 2012].

In anamniotes the situation is complicated by continuous growth, which would require continuous ECM remodelling in all tissues, not just the sclera. Additionally, growth would require regulation of the intra-ocular fluid to create a counter-pressure. Interestingly, ECM components were found to be expressed in the CMZ [Cerveny et al., 2012]. Thus, a mechanistic explanation could be that the NR modulates organ shape by local ECM modification – a requirement for growth – which together with intra-ocular fluid build-up results in anisotropic tissue forces that ultimately bias cell divisions or arrangement of sister cells. Testing this hypothesis would require characterising the ECM in the wildtype CMZ and in fish with defective emmetropy, *e.g.* by lens-induced myopia.

3.3.3 A localised signal instructs asymmetry in the medaka NR

Asymmetric retinal growth patterns have long been known in fishes, and are highly correlated to visual function [Easter, 1992; Zimmermann et al., 2018]. In green sunfish, the retinal area that grew slowest had the highest visual acuity [Cameron, 1995]. Medaka gaze upwards in their natural habitat, and their ventral retina has a slightly higher photoreceptor cell density [Nishiwaki et al., 1997]. Thus, slower ventral growth in the medaka retina may play a functional role.

The mechanism for differential growth rate in green sunfish was found to be less proliferation in the CMZ [Cameron, 1995], consistent with my observations in medaka. The simulation enabled to pinpoint proliferation parameters that differ in the ventral NR, and indicated that lower proliferation rate was concomitant to regulation of division axis orientation (or re-orientation of daughter cells). Interestingly, in frogs the relationship is inverted and the ventral retina proliferated more than the dorsal retina [Conway et al., 1980; Hunt et al., 1988]. Differential expansion in frog also lead to differential clone termination, with many more clones terminating dorsally [Conway et al., 1980; Hunt et al., 1988]. Similarly, ventral retinal clones in medaka also displayed a greater propensity to terminate (Figure 2.25 E′). Though modelled as a discrete 90° sector, there is more likely a graded change in proliferative parameters as one approaches the ventral pole of the retina. In terms of retinal shape, the ventral side also appeared to deviate more from hemisphericity (see, *e.g.* Figure 2.26 E), further linking retinal proliferation parameters to eye shape.

Intriguingly, when grafting a piece of ventral frog retina into the anterior or posterior location of a host retina, the donor piece remembered its pre-graft position and expanded accordingly [Conway et al., 1980], suggesting a local determinant to the graft imparted its proliferation capacity. In green sunfish, experimental rotation of the eye in its orbit demonstrated that an eye-intrinsic signal imparted this asymmetry [Cameron, 1996]. The simulation suggested this signal was not intrinsic to cell lineages, but a locally confined extrinsic signal (Figure 2.25). These data are consistent with a short-range signal, perhaps a diffusible molecule or an ECM-linked component, that is exclusively locally propagated. Transcriptomic analysis of the ventral and dorsal retina may help elucidate the unknown nature of this signal.

Anatomical structures compartmentalise the NR

The embryonic origin of the ventral retinal pole is the fissure of the optic cup. In medaka and zebrafish, this fissure completely fuses, but nevertheless the ventral pole contains anatomical features such as a large ventral blood vessel and the ciliary canal, which drains acqueous humor [Soules and Link, 2005]. This structure was macroscopically visible in distal wholemount acquisitions of the medaka retina (*e.g.* Figure 2.26 E). The NR CMZ had less cells at this position, resulting in less clonal labelling (Figure 2.26). Similarly, in frog the ventral pole appeared to act as an anatomical barrier to clonal expansion [Hunt et al., 1988].

Other anatomical features of the medaka retina included potential attachment sites of the lens to the basal ECM of the NR (Figure 2.26 E). These anatomical features correlated with differential marker expression in the CMZ. Further, some markers such as *tlx* were asymmetrically distributed, hinting at functional differences in the CMZ which may relate to the creation of specialised neuronal subtypes as observed in zebrafish [Zimmermann et al., 2018].

3.4 Complex heterogeneity in the RPE CMZ

3.4.1 RPE lineages may display cell cycle correlations

The responder growth mode of the RPE predicted that cell division timing variability would be elevated in this tissue (Figure 2.19). Validation experiments with BrdU incorporation revealed a much higher heterogeneity than expected (Figure 2.27). Further experiments showed that at any one time, there was a subpopulation of non-cycling RPE cells, which were likely quiescent (*i.e.* in the G0 phase of the cell cycle) due to the lack of PCNA staining [Zerjatke et al., 2017], and label retention dynamics incompatible with heterogeneity purely driven by cell cycle length variability (Section 2.4.5). In general, quiescent cells seemed more common in the very periphery (Figures 2.33 and 2.34), reminiscent of the model with differential cell cycle time in SC and PC (Figure 2.7). Currently it's unclear if RPE SCs had different cell cycle timing along the central–peripheral axis or if peripheral quiescence bias was an emergent property. Estimation of cell cycle times along the extent of the RPE CMZ in additional experimental samples could clarify this point.

Quiescent and actively cycling cells clustered (Figure 2.33), and over time cells with similar cycling properties formed trails along the central–peripheral axis (Figure 2.34). Intriguingly, these trails bore a strong resemblance to RPE ArCoS (*e.g.* Figure 2.39 B and E), suggesting that clonally related cells were displaying similar cell cycle dynamics. Clonal correlations in cell division occurred in simulations with large values of p_{div}, *i.e.* low variability in cell division timing (Figure 2.35). Such effects have also been observed in cell culture, where cells within a lineage tree displayed correlations in cell cycle timing and quiescence; these correlations were attributed to the inheritance of factors that stimulated or inhibited cell cycle progression [Arora et al., 2017; Chao et al., 2019; Kuchen et al., 2018]. Thus, the clone-like clustering of cells with similar division dynamics in the niche of the RPE may represent lineage tree correlations. These data imply that the RPE responds to the inductive NR signal via a large proliferative population where individual cell lineages dynamically respond to the growth needs of the tissue by entering or exiting quiescence.

3.4.2 Improving model fit by accounting for RPE geometry

Varying the parameter values in the base model and an extended version that included a quiescent state could only partially replicate the qualitative behaviour observed experimentally (Figure 2.35). As nearly nothing is known about the SCs of the RPE, I experimentally characterised the RPE CMZ from hatchling until late larval stages. The RPE niche contained more proliferative cells, had different sizes of CMZ cells and differentiated cells, and the tissue curved around the underlying NR. The curving of the tissue lead to a shape that was better approximated by a hemisphere with a flat annulus at its base (Figure 3.3 A′′).

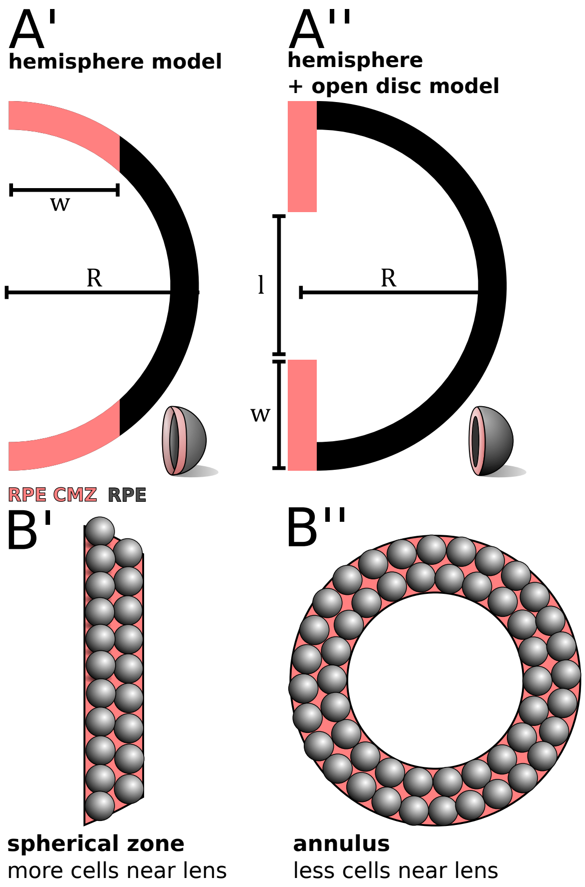


Figure 3.3: RPE geometry is better approximated by a hemisphere with a flat annulus. **A′** Schematic of the current hemispherical model's geometry. **A″** Schematic of an improved model version with an annulus-shaped CMZ at the base of the hemisphere. **B′** Side view of a 2-row wide spherical zone CMZ. The row closest to the lens (left) has the most cells. **B″** Distal view of a 2-row annulus CMZ. The row closest to the lens (inner circle) has the least cells.

As I showed in this work for the NR, geometry can impact clonal dynamics. Importantly, the number of cells that can be accommodated on the "annulus-CMZ" differs from the number that could fit on the "zone-CMZ". The area difference can be clearly demonstrated as follows: The area of the CMZ as a spherical zone with width w on a hemisphere with radius R is

$$A_{\text{zone}-\text{CMZ}} = 2\pi R w, \tag{3.1}$$

while the area of an annulus with inner diameter $l = 2(R - w)$ at the base of the hemisphere is

$$A_{\text{annulus}-\text{CMZ}} = \pi\left(R^2 - \frac{l^2}{4}\right), \tag{3.2}$$

which, using the above relation for l, simplifies to

$$A_{\text{annulus}-\text{CMZ}} = \pi\left(w^2 - 2Rw\right). \tag{3.3}$$

Thus, the area ratio is

$$\frac{A_{\mathrm{zone-CMZ}}}{A_{\mathrm{annulus-CMZ}}} = \frac{2\pi R w}{\pi(w^2 - 2Rw)} = \frac{1}{1 - \frac{w}{2R}}. \tag{3.4}$$

Equation 3.4 shows that $A_{\mathrm{annulus-CMZ}} \leq A_{\mathrm{zone-CMZ}}$. Not only is the absolute area different, but using a similar argument, it can be shown that the number of cells in the peripheral-most row is highest in the "zone-CMZ", but lowest in the "annulus-CMZ" (Figure 3.3 B′-B″).

3.4.3 Disentangling mechanisms of RPE quiescence

The RPE revealed unexpected spatiotemporal dynamics in the cycling behaviour of cells. At the population level SCs were highly heterogeneous, but paradoxically local clusters of presumably clonal cells had homogeneous cycling dynamics. One potential explanation to this paradox is that a population cycling with low cell cycle timing variability became desynchronised by bouts of quiescence (Figure 3.4). Incorporating a more detailed description of RPE geometry into future iterations of the agent based model developed in this work will prove invaluable to address this hypothesis and identify the mechanism of RPE SC quiescence.

Another key improvement will be a more realistic cell cycle model. The current model coupled a minimum cell cycle progression time $t_{\mathrm{cellCycle}}$ with a probability of division p_{div}, which essentially only affected the right skew of the distribution (Figure 2.6). Therefore, any variability introduced by p_{div} necessarily reduced population growth rate. Though useful as a first approximation, this cell cycle model cannot quantitatively reproduce thymidine analogue incorporation dynamics as it lacks a defined S-phase. Time-lapse data acquired in various cell culture systems indicated that individual phases of the cell cycle were independent and memoryless [Chao et al., 2019]. All phases followed a distribution characterised by a minimum duration time and right skew, which could be modelled by an Erlang distribution [Chao et al., 2019]. Despite the memoryless property of individual cell cycle phases, correlations between related cells still occurred due to inheritance of cell cycle progression factors [Chao et al., 2019].

Quiescence-inducing stimuli include mechanical contact inhibition, nutrient or mitogen starvation, and (endogenous) DNA damage [Arora et al., 2017; Rumman et al., 2015]. Though I chose an implementation with contact inhibition, in principle all of these mechanisms are plausible and occur in human immortalised RPE cell lines [Arora et al., 2017; Cadart et al., 2018; Chao et al.,

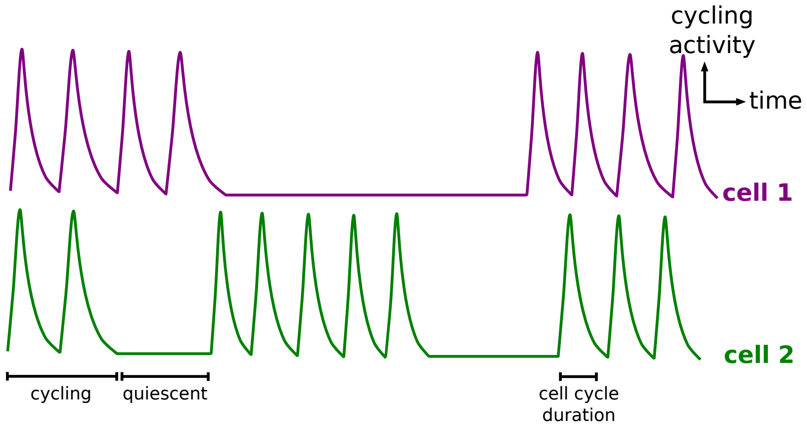


Figure 3.4: Hypothetical model of cell desynchronisation by quiescent bouts.
Cell 1 and cell 2 are assumed to have an average cell cycle duration with low variation. Though initially both cells were cycling near-synchronously, quiescent periods lead to desynchronisation.

2019]. An improved model coupled with immunohistochemical stainings for markers of different types of quiescence will allow to disentangle these possibilities. Further, the simulation will test memoryless versus memoried quiescence, and whether local sister cell synchronisation contributes to clustering of label-retaining cells. A preliminary quantitative analysis pipeline designed to extract features of the thymidine analogue incorporation pattern was developed by Anne Newrly [Newrly, Tsingos, and Wittbrodt, unpublished], and will form the basis for future quantitative comparisons.

3.5 Post-embryonic growth of a new tissue - the NVR

3.5.1 Growth kinetics of the medaka NVR

Despite its regenerative potential in amphibians [Reyer, 1977; Tsonis, 2000] and evolutionary conservation with the site of potential mammalian NR SC [Ahmad et al., 2000; Tropepe et al., 2000], there is little research on the anamniote NVR. In frog, it has been hypothesized that the NVR grew from the CMZ [Conway et al., 1980]. In zebrafish, the NVR first became morphologically distinct at 14 dph [Soules and Link, 2005], but its cellular origin was not investigated.

In medaka, combined analysis of tissue sections and wholemounts revealed a clearly discernible structure already at hatchling stage (Figure 2.36). This NVR primordium grew at a very fast rate between 10 and 35 dph, a period that coincided with the larval growth spurt and eventual metamorphosis into a juvenile medaka at stage 42 [Iwamatsu, 2004]. Afterwards, growth continued at a reduced rate. In contrast, the CMZ retained a near-constant size. All markers that were tested displayed shared expression between the NR CMZ and the proximal NVR, including the circumferential asymmetry in expression patterns (Figure 2.37). This shared marker expression implied a continuous molecular transition from CMZ cells into proximal NVR cells.

3.5.2 The NVR forms a clonal continuum with NR and RPE

Consistently, all tested TMX-inducible *cre* drivers generated clones both in the proximal NVR and NR, regardless of the timepoint of induction (Figure 2.38 and 2.40). Without fail, NR clones that reached into the very peripheral CMZ always had an NVR correlate, suggesting a single clonal origin. Clones that were solely in the NR were not directly contiguous to the CMZ (*i.e.* presumably terminated clones; Figure 2.39 A). These data suggested that there was no restriction in potency – NR SCs were capable of generating the proximal NVR both at early and late stages. More extensive clonal analysis with a heat-shock inducible *cre* revealed shared clones between all adjacent tissues, indicating a clonal continuum extending from the RPE through the NVR into the NR (Figure 3.5).

If the retina forms a clonal continuum, then why was a clone ranging from the NR into the RPE never observed? The answer may be a simple matter of cellular crowding (Figure 3.5). Assuming similar properties in the NVR as in the rest of the retina, *i.e.* no cell mixing and no cell death, the only way for a "full-loop clone" to arise is by displacement of all neighbours due to proliferation. However, all NVR cells had the potential to proliferate (Figure 2.41), thus there was a constant "competitive pressure" lowering the chance of RPE or NR of invading the NVR (and vice versa, invasion of NVR into the other tissues was also competitively inhibited). Consistent with this hypothesis, the length of the chase period correlated with how far NR clones reached into the proximal NVR (Figure 2.38).

In zebrafish, the prospective CMZ gives rise to clones in the NR, RPE, and a "dormant tip cell" at the very distal margin of the retina [Tang et al., 2017]. Though not further characterised, the position and morphology of this "tip cell" are consistent with the NVR primordium. According to the clonal continuum hypothesis, the small NVR primordium and late NVR development in zebrafish may explain why in this species hybrid clones between NR and RPE could be detected [Tang et al., 2017], but not in medaka [Centanin et al., 2014, 2011].

Clonal analysis only informed on the end result, not the position of the founder cell that gave rise to clones. The clonal continuum hypothesis predicts that cells originating in NR, RPE, and NVR can contribute to all retinal tissues. In the absence of specific markers for each tissue, this prediction can only be ad-

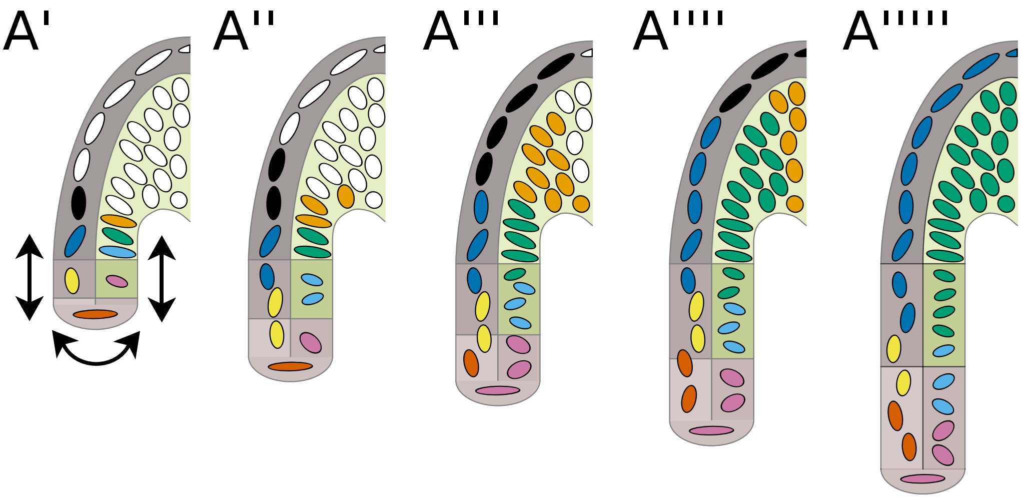


Figure 3.5: Clonal continuum model of retinal growth at the NVR.
A′-A′′′′′ Schematic drawing showing the temporal evolution of eight clones in a cross-section of the growing CMZ and NVR from hatchling (A′) to adult (A′′′′′). Note how one clone becomes dominant in NR and RPE, but neither clone can completely colonise the NVR because the cells have nowhere to be displaced to. In reality, displacement can also occur into and out of the plane of the drawing. Arrows in A′ indicate growth direction.

dressed by time-lapse acquisition of albino mutants generated by CRISPR/Cas9 in a background amenable to clonal induction [Lischik et al., 2019].

3.6 Perspective

3.6.1 To niche its own - growth strategies of retinal tissues

Despite their common embryonic origin, NVR, NR and RPE followed different growth strategies in medaka. Cells in the NR divided with low variability, while RPE cells displayed high variability at the population level with homogeneous local clusters. NR and RPE shared a topological organisation with distinct differentiated and proliferative domains and a dedicated population of SCs. In contrast, the NVR lacked a dedicated niche, and instead the entire bilayered epithelium was dotted with proliferative cells. Thus, retinal tissues have found a diversity of solutions to the problem of continuous growth.

Why do some tissues restrict stemness to a dedicated niche, while others distribute this function to all cells? To answer this question, it may be useful to take a step back. Multicellular organisms are characterised by specialisation and compartmentalisation that allowed division of labour; control mechanisms co-evolved to coordinate different organs, tissues, and cells [Droujinine and Perrimon, 2016]. The dimensions of a given (cellular) tissue ultimately depend on the influx and efflux rates of its constituent cells. Presumably, niches evolved at the onset of multicellularity to control the number and activity of SCs, and thus the influx part of the equation. This tacit assumption underlies the dogma of SC biology that defines the niche as a microenvironment that is an absolute prerequisite for SCs [Morrison and Spradling, 2008; Scadden, 2014].

The advantage of having a dedicated niche is exemplified by the concerted growth of the medaka retina: The fixed niche extent of the NR leads to linear growth of the eye radius (discussed in Section 3.1.3). Evolutionarily speaking, modulating a single parameter – the width of the NR CMZ – automatically scales the entire organ as the NR acts as an organising centre (discussed in Section 3.2.2). Beyond centralising control of SCs, a peripheral niche topology may be the most parsimonious way to ensure that retinal architecture is not disrupted by continuous proliferation. The restriction of the RPE CMZ may be due to similar constraints acting on photoreceptor–RPE cell interactions [Fuhrmann et al., 2014] or due to a common "niche factor" pool shared with the NR CMZ. Maintenance of a precise tissue architecture is arguably not as crucial in the NVR, allowing this tissue to have decentralised stemness, similar to the mammalian liver which lacks a "professional stem cell" [Clevers and Watt, 2018]. Thus, all cells in the NVR are capable of acting as SCs.

3.6.2 The RPE as a model system for homeostatic quiescence

Previous studies identified proliferative cells at the peripheral edge of the anamniote RPE [Conway et al., 1980; Johns and Easter Jr, 1977] and characterised its regenerative capacities [Hanovice et al., 2019; Tsonis, 2000], but the homeostatic growth of the anamniote RPE received little attention. The medaka RPE CMZ was populated by cycling and quiescent cells. Quiescence is a poorly understood cell state that may underlie long-term dormancy of cancer SCs causing relapse after years free of disease [Li and Bhatia, 2011]. In mammalian models, quiescence maintains a reserve pool of SCs in "standby" for rapid response after injury; quiescent and actively cycling stem cells reside in spatially compartmentalised niches [Li and Clevers, 2010]. Coexistence of quiescent and cycling cells within a single niche in the RPE contrasts these standard models, offering a different perspective into this fundamental cell state.

The spatio-temporal clustering of cells with similar cycling properties in the RPE CMZ may result from lineage-tree correlations. Correlations in cell cycle progression are known in cell culture and unicellular organisms such as yeast (*Saccharomyces cerevisiae*) [Arora et al., 2017; Chao et al., 2019; Sandler et al., 2015], but to date there has been no report of this phenomenon in multicellular organisms. Thus, the anamniote RPE may represent a new model system to understand the impact of lineage-tree correlations and SC quiescence in a vertebrate system.

Furthermore, RPE cell homeostasis plays a critical function in human retinal health. While non-proliferative in homeostasis, the RPE aberrantly proliferates in several retinal degenerative diseases [Fuhrmann et al., 2014]. Thus, characterising RPE CMZ homeostasis in a tractable anamniote model may help elucidate human pathologies.

3.6.3 Retinal tissue properties - conserved throughout vertebrates

The mammalian NVR came into research focus due to the discovery of potential retinal SCs based on marker expression and self-renewing potency in tissue culture [Ahmad et al., 2000; Haruta et al., 2001; Tropepe et al., 2000]. These findings were bolstered by common marker expression of mammalian and avian NVR with embryonic retinal PCs [Ballios et al., 2012; Das et al., 2005, 2006; Del Debbio et al., 2013; Fischer et al., 2014]. The mammalian NVR SCs were considered an evolutionary counterpart to the CMZ [Ahmad et al., 2000; Das et al., 2006; Del Debbio et al., 2013; Tropepe et al., 2000] – which was long touted as a distinctive feature of anamniotes absent in higher vertebrates [Amato et al., 2004; Harris and Perron, 1998].

Several lines of evidence however highlight that the CMZ is an ancestral feature that was not completely lost, but merely reduced in mammals. During late embryonic development in chicken [Venters et al., 2011, 2013, 2015] and mouse [Bélanger et al., 2017], a region anatomically similar to the NR CMZ proliferated, giving rise to hybrid clones that contributed to both NR and proximal NVR. Moreover, the peripheral RPE of mouse and rat retained proliferative capacity long after birth [Al-Hussaini et al., 2016; Kokkinopoulos et al., 2011]. Indeed, clones in the mouse RPE displayed a striped growth pattern consistent with CMZ-like proliferation [Bodenstein and Sidman, 1987a,b].

The full elucidation of retinal SC dynamics and coordination in an anamniote system allows to integrate these data: As in mammals, the anamniote NVR expresses retinal SC markers and can give rise to all other retinal tissues. Thus, throughout vertebrate evolution, retinal tissues conserved a close molecular and clonal relationship. Uncovering the molecular mechanisms underpinning this conserved tissue triad will enable unique insight into homeostatic SC dynamics in a physiological context.

4
Conclusions

In this work, I have combined experimental and computational approaches to characterise proliferation dynamics of different populations of SCs during post-embryonic growth of the medaka retina. I designed a 3D cell centred agent based model that encompassed previous knowledge of cellular behaviour in the NR and RPE. This model facilitated interpretation of experimental data and enabled answering several questions *in silico*, such as how NR and RPE coordinate their growth rates, and how stochastic neutral drift and deterministic behaviour are balanced in the CMZ. By leveraging the model, I uncovered how SCs in the NR modulate their proliferative parameters to regulate retinal shape and topology. Prompted by model predictions, I characterised the extent, cellular morphology, and proliferation dynamics of the RPE CMZ, uncovering a dynamic SCs population that balances quiescence and active cycling within one niche. Finally, I characterised the growth kinetics, molecular markers, and clonal relationship of the NVR, revealing a clonal continuum of retinal tissues.

The work presented in this thesis shows that cells in the CMZ have remarkable plasticity: Within one and the same tissue ostensibly equipotent cells can act as short-term PCs or lifelong SCs. Furthermore, retinal SCs have the potency to contribute to the NR, the RPE, or the NVR depending on their local microenvironment. As a testament to this remarkable plasticity, each of these tissues uses different strategies for growth and resource allocation to a dedicated niche. Thus, lifelong growth of anamniotes provides unique opportunities to study SC dynamics in their physiological context.

5

Materials & Methods

5.1 Materials

5.1.1 Organisms

Medaka fish lines

Table 5.1: References for single transgenic medaka fish lines.

Line name	References
cab	Loosli et al. [2000] Inbred isogenic lab strain derived from southern medaka population.
ccl25b::GFP	Line gifted by Dr Baubak Bajoghli. [Bajoghli, unpublished]
ccl25b::^{ERT2}cre	Lust and Wittbrodt, unpublished
CR(GFP-rx2)	Gutierrez-Triana et al. [2018]
CR(lrp2a::eGFPwCR13)	this work
CR(oca2)	Lischik et al. [2019]; Tsingos et al. [2019]
CR(oca2, pnp4a)	Lischik et al. [2019]
Gaudí$^{BBW2.1}$	Centanin et al. [2014]
Gaudí$^{LoxP\text{-}OUT}$	Centanin et al. [2014]
Gaudí^{LxBBW}	Centanin et al. [2014]
Gaudí^{RSG}	Centanin et al. [2014]
heino	Wittbrodt, unpublished
hsp70::cre-NLS	Centanin et al. [2014]
rx2::^{ERT2}cre	Reinhardt et al. [2015]
rx2::H_2B-RFP	Reinhardt et al. [2015]
tgfβRE::GFP-caax	Stemmer and Wittbrodt, unpublished The TGFβ responsive element from human PAI promoter drives a membrane-tethered GFP.

tlx::^{ERT2}*cre*	Tavhelidse and Wittbrodt, unpublished
tlx::GFP	Reinhardt and Wittbrodt, unpublished
tlx::H_2B*-RFP*	Tavhelidse and Wittbrodt, unpublished
tp1::d2GFP	Saturnino et al. [2018]
ubi::^{ERT2}*cre*	Centanin et al. [2014]

Table 5.2: Stock numbers of medaka fish lines used in this work.

Line name	Internal stock numbers
cab	6097, 6480, 6857, 7239, 7524
ccl25b::GFP	8483
ccl25b::^{ERT2}*cre, Gaudí*^{RSG}	8127, 8235, 8256
CR(GFP-rx2), CR(oca2)	8482
CR(lrp2a::eGFPwCR13)	6312, 6495, 6613, 6876, 7049, 7367, 7803, 8039, 8361, 8692
CR(lrp2a::eGFPwCR13), hsp70::cre-NLS, Gaudí^{RSG}	7309, 7451, 7512, 7614, 7967, 8369
CR(oca2)	6384
CR(oca2, pnp4a)	8463
Gaudí$^{BBW2.1}$	7229
Gaudí$^{LoxP\text{-}OUT}$	7169, 7670, 7527, 7818
Gaudí^{RSG}	6801
heino	5616
hsp70::cre-NLS, Gaudí^{LxBBW}	7192, 7560
hsp70::cre-NLS, Gaudí^{RSG}	6668, 7802, 7798, 8417
rx2::^{ERT2}*cre, Gaudí*^{RSG}	7174, 7185, 8090, 8245
tgfβRE::GFP-caax	6698, 8067, 8068
tlx::^{ERT2}*cre, Gaudí*^{RSG}	8119, 8244
tlx::H_2B*-RFP,ccl25b::GFP*	8268
tlx::GFP,rx2::H_2B*-RFP*	8267, 8461
ubi::^{ERT2}*cre, Gaudí*^{RSG}	7511, 7826
tp1::d2GFP	8509

Zebrafish fish lines

Table 5.3: Zebrafish fish lines used in this work. All lines were gifted by Dr Rita Mateus [Mateus, and González-Gaitán, unpublished].

Line designation	Internal stock numbers
CR(smoc1_1bp_del), Tg(BRE-AAVmlp:eGFP)_Collery, Tg(Xla.Eef1a1:H2B-mCherry)	8107, 8259
CR(smoc1_1bp_del), Tg(bre:eGFP)_Laux, Tg(Xla.Eef1a1:H2B-mCherry)	8130, 8260
Tg(BRE-AAVmlp:eGFP)_Collery, Tg(Xla.Eef1a1:H2B-mCherry)	8108
Tg(bre:eFP)_Laux, Tg(Xla.Eef1a1:H2B-mCherry)	8142

Bacteria

Table 5.4: Bacteria used in this work.

Designation	Source
MachT1^{TM} T1^{R} phage resistant chemically competent *Escherichia coli*	Thermo Fisher Scientific

5.1.2 Plasmids

Table 5.5: Plasmids used in this work.

Internal stock number	Name	Source
2877	pGEM-T Easy	Promega
3190	pGGEV-5stop_Linker	lab stock
3237	pGGW_EV2 LacZ	lab stock
3632	DR274 sgRNA backbone	lab stock
3776	pGGEV-1_-(CR13)-_+1/-1_OA	lab stock
3900	pGGDestSC-ATG	lab stock
4044	pGGEV-4_-(3xpolyA)-_+1_BK	lab stock
4170	pGGEV-3_+(eGFPwCR13)+_+_BK	lab stock
4279	pGGEV-2_-(Lrp2a_5'HF)-_+1_BK+	this work
4280	GGD(CR13_Lrp2a5'HF_eGFPwCR13_3xPolyA)	this work
4379	Lrp2a in situ probe	this work
4381	Lrp2a in situ probe #2	this work
4383	sgRNA Lrp2a #1	this work

4384	sgRNA Lrp2a #2	this work
4385	sgRNA Lrp2a #3	this work
5357	DR274(sgRNA 57 Oca2_ex9_T1)	lab stock
5358	DR274(sgRNA 58 Oca2_ex9_T3)	lab stock
5197	pCS2+ Inv X_Cas9	lab stock

5.1.3 Primers

All primers designed in this work were ordered at Eurofins Genomics via a custom FileMaker script from the lab.

Table 5.6: Primers used in this work. Sequences given from 5' to 3'.

Internal stock number	Alias	Sequence	Source
JW 1458	GFP_noStart	GTGAGCAAGGGCGAGGAGCT	lab stock
JW 1745	GFP-STOP-XbaI_R	TTACTTGTACAGCTCGTCCATG	lab stock
JW 2253	GFP_nested_R2	CTGAACTTGTGGCCGTTTACG	lab stock
JW 3451	lrp2 _ex2_T1_F	TAGgATTGGGACAGTTCCGCTG	this work
JW 3452	lrp2 _ex2_T1_R	AAACCAGCGGAACTGTCCCAAT	this work
JW 3453	lrp2 _ex2_T2_F	TAGGACAGGTCGATGTATACCA	this work
JW 3454	lrp2 _ex2_T2_R	AAACTGGTATACATCGACCTGT	this work
JW 3455	lrp2 _ex2_T4_F	TAggATCACAGTGCCAGTCCCC	this work
JW 3456	lrp2 _ex2_T4_R	AAACGGGGACTGGCACTGTGAT	this work
JW 3457	lrp2_5'HF_BamHI_F	GCCGGATCCCCTAGTACCACTATT TATTCCAATTACACA	this work
JW 3458	lrp2_seq_F	CGAGGAATTTGGGTTGAAATGACT	this work
JW 3459	lrp2_seq_R	ACAACTTTAAAACAGAAACTGGGGT	this work
JW 3460	lrp2_5'HF_KpnI_R	GCCGGTACCGCGGAACTGTCCCAA TTCACA	this work
JW 3501	lrp2a_S2_F	GCCGGTACCGCGGAACTGTCCCAA TTCACA	this work
JW 3503	lrp2a_S3_F	GCTCCAGTTCTGGTGCTCAGGTG	this work
JW 3504	lrp2a_S2_R	GTGCACCTGGATTAGTGTAATAGA GGTTCTCTG	this work
JW 3505	lrp2a_S3_R	GTGCTGACTATGACCTCACGGTTG GATCCG	this work

5.1.4 RNAs

Transcription of single guide RNAs (sgRNAs) was performed according to protocol in Section 5.2.12. Cas9 messenger RNA (mRNA) was transcribed from NotI-HF-linearised plasmids using the mMESSAGE mMACHINE Sp6 Transcription Kit according to the manufacturer's instructions [Life Technologies, 2012].

Table 5.7: RNAs used in this work.

Name	Source	Comments
sgRNA Lrp2a #1	this work	Transcribed from plasmid #4383.
sgRNA Lrp2a #2	this work	Transcribed from plasmid #4384.
sgRNA Lrp2a #3	this work	Transcribed from plasmid #4385.
sgRNA 57 Oca2_ex9_T1	lab stock	Transcribed from plasmid #5357.
sgRNA 58 Oca2_ex9_T3	lab stock	Transcribed from plasmid #5358.
Cas9	lab stock	Transcribed from plasmid #5197.

5.1.5 Antibodies

Table 5.8: Primary antibodies used in this work. Abbreviations: Cat. – Catalog; nr. – number; W – wholemounts; S – sections.

Lab nr.	Target	Host	Dilution	Type	Source	Cat. nr.
p7	BrdU	rat	1:100 (W) 1:200 (S)	monoclonal	abcam	ab6326
p25	dsRed	rabbit	1:100 (W) 1:200 (S)	polyclonal	Clontech	632496
p38	GFP	chicken	1:200 (S; W ArCoS assay) 1:50 (W; GFP reporters)	polyclonal	Life Technologies	A10262
p99	PCNA	mouse	1:25 (W)	monoclonal	Millipore	CBL407
p172	smoc1	mouse	1:200 (W)	monoclonal	Biozol / Abnova	H0-0064-093-M03
p173	P-Smad-1/5/9	rabbit	1:100 (W)	monoclonal	Cell Signaling	13820

–	rx2	rabbit	1:50 (W) 1:200 (S)	polyclonal	Charles River	custom

Table 5.9: Secondary antibodies used in this work. Abbreviations: Cat. – Catalog; nr. – number; conj. – conjugate.

Target	Host	Dilution	Type	Source	Cat. nr.
chicken	donkey	1:200	polyclonal, Alexa Fluor 488-conj.	Jackson/ Dianova	703-545-155
chicken	donkey	1:100	polyclonal, Alexa Fluor 647-conj.	Jackson	703-496-155
mouse	goat	1:200	polyclonal, Alexa Fluor 488-conj.	Life Technologies	A-11029
mouse	donkey	1:200	polyclonal, Alexa Fluor 633-conj.	Jackson/ Dianova	715-605-151
rabbit	goat	1:100	polyclonal, DyLight 549-conj.	Jackson	112-505-144
rabbit	goat	1:100	polyclonal, Alexa Fluor 647-conj.	Thermo LifeTech	A21245
rat	goat	1:200	polyclonal, DyLight 549-conj	Jackson Immunoresearch	112-505-143
rat	goat	1:100	polyclonal, Alexa Fluor 633-conj	Invitrogen	A-21094

Table 5.10: Other antibodies used in this work. Abbreviations: Cat. – Catalog; nr. – number; conj. – conjugate.

Target	Host	Dilution	Type	Source	Cat. nr.
Digoxigenin	Sheep	1:2000	Alkaline phosphatase-conj.	Roche	11093274910

5.1.6 Antibiotics

Table 5.11: Antibiotics used in this work for bacterial selection.

Antibiotic	Stock concentration	Working concentration	Supplier
Ampicillin	$100\,\mathrm{mg\,ml^{-1}}$	$100\,\mathrm{\mu g\,ml^{-1}}$	Roth
Kanamycin	$50\,\mathrm{mg\,ml^{-1}}$	$50\,\mathrm{\mu g\,ml^{-1}}$	Roth

5.1.7 Kits

Table 5.12: Kits and kit reagents used in this work.

Kit name and reagents used	Supplier
Click-iT EdU Alexa Fluor 647 Flow Cytometry Assay Kit • 5-ethynyl-2'-deoxyuridine (EdU; 10 mg) • Alexa Fluor 647 azide • $CuSO_4$ ($100\,\mathrm{mmol\,l^{-1}}$) • Click-iT EdU buffer additive (400 mg) • 10x Click-iT EdU reaction buffer	Invitrogen
innuPREP DOUBLEpure Kit • Gel solubiliser • Binding Optimizer • Binding Buffer • Washing Solution LS	Analytik Jena
MEGAshortscript T7 Transcription Kit • T7 Enzyme Mix • T7 10x Reaction Buffer • T7 ATP Solution ($75\,\mathrm{mmol\,l^{-1}}$) • T7 CTP Solution ($75\,\mathrm{mmol\,l^{-1}}$) • T7 GTP Solution ($75\,\mathrm{mmol\,l^{-1}}$) • T7 UTP Solution ($75\,\mathrm{mmol\,l^{-1}}$) • TURBO DNase $\left(2\,\mathrm{U\,\mu l^{-1}}\right)$ • Ammonium Acetate Stop Solution	Invitrogen
mMESSAGE mMACHINE Sp6 Transcription Kit • SP6 RNA Polymerase Enzyme Mix • 10x SP6 Reaction Buffer • 2x SP6 NTP/CAP solution	Life Technologies
Plasmid Midi Kit • Buffer P1 • Buffer P2 • Buffer P3 • Buffer QBT • Buffer QC • Buffer QF • RNase A	Qiagen

PCR DIG Probe Synthesis Kit • Enzyme mix $(3.5\mathrm{U\,\mu l^{-1}})$ • 10x PCR DIG Probe Synthesis Mix • 10x PCR buffer • 10x dNTP stock solution	Roche
pGEM-T Easy Vector System • pGEM-T Vector ($50\ \mathrm{ng\,\mu l^{-1}}$) • T4 DNA Ligase $(3\ \mathrm{U\,\mu l^{-1}})$ • 2x Rapid Ligation Buffer	Promega
RevertAid First Strand cDNA Synthesis Kit • RevertAidTM M-MuLV Reverse Transcriptase $(200\ \mathrm{U\,\mu l^{-1}})$ • RiboLock RNase Inhibitor $(20\ \mathrm{U\,\mu l^{-1}})$ • 5x RT Reaction Buffer • dNTP mix ($10\ \mathrm{mmol\,l^{-1}}$) • Oligo$(\mathrm{dT})_{18}$ Primer ($100\ \mathrm{\mu mol\,l^{-1}}$)	Thermo Fisher Scientific

5.1.8 Enzymes and buffers

Table 5.13: Enzymes and corresponding buffers used in this work.

Enzyme	Buffer (concentration)	Supplier
AatII $(20\ \mathrm{U\,\mu l^{-1}})$	CutSmart buffer (10x)	New England Biolabs
BamHI-HF $(20\ \mathrm{U\,\mu l^{-1}})$	CutSmart buffer (10x)	New England Biolabs
BsaI-HF $(20\ \mathrm{U\,\mu l^{-1}})$	CutSmart buffer (10x)	New England Biolabs
DraI FD	FastDigest buffer (10x)	Thermo Fisher Scientific
KpnI-HF $(20\ \mathrm{U\,\mu l^{-1}})$	CutSmart buffer (10x)	New England Biolabs
NotI-HF $(20\ \mathrm{U\,\mu l^{-1}})$	CutSmart buffer (10x)	New England Biolabs
Proteinase K ($20\ \mathrm{mg\,ml^{-1}}$)	–	Roche
PstI FD	FastDigest buffer (10x)	Thermo Fisher Scientific
RNase A, DNase- and protease-free ($10\ \mathrm{mg\,ml^{-1}}$)	–	Thermo Fisher Scientific
SP6 RNA polymerase $(20\ \mathrm{U\,\mu l^{-1}})$	SP6 Transcription Buffer (10x)	Roche
Taq Roboklon $(5\ \mathrm{U\,\mu l^{-1}})$	PolBuffer B (10x)	Roboklon
Taq Polymerase	Taq Polymerase Buffer (10x)	lab stock

TURBO DNase $\left(2\,\mathrm{U\,\mu l^{-1}}\right)$	DNase I buffer (10x)	Invitrogen
Q5 Polymerase	Q5 Reaction Buffer (5x)	New England Biolabs
XhoI $\left(20\,\mathrm{U\,\mu l^{-1}}\right)$	CutSmart buffer (10x)	New England Biolabs

5.1.9 Reagents

Table 5.14: Reagents used in this work.

Name	Alias	Supplier
2-Propanol	Isopropanol	Sigma-Aldrich
2x RNA loading dye		Thermo Fisher Scientific
4-(2-Hydroxyethyl)piperazine-1-ethanesulfonic acid	HEPES	Roth/ZBT
4',6-diamidino-2-phenylindole	DAPI	Sigma-Aldrich
4-nitro blue tetrazolium chloride	NBT	Roche
5-bromo-2'-deoxyuridine	BrdU	Sigma-Aldrich
5-bromo-4-chloro-3-indoyl phosphate	BCIP	Roche
Acetone		Sigma-Aldrich
Adenosine triphosphate	ATP	Thermo Fisher Scientific
Agar		Roth
Agarose		Sigma-Aldrich
Agarose (Low Melt)		Roth
Anti-Digoxigenin-AP Fab fragments		Roche
Bacto-Tryptone		Gibco
Blocking reagent		Roche
Bovine serum albumine	BSA	Sigma-Aldrich
Calcium chloride dihydrate	$\mathrm{CaCl_2 \cdot 2H_2O}$	AppliChem
CDP-Star		Roche
Chloroform		Sigma-Aldrich
Deoxyadenosine triphosphate	dATP	Promega
Deoxynucleotide triphosphates	dNTPs	Sigma-Aldrich
Digoxigenin-11-uridine-5'-triphosphate	DIG-UTP	Roche
Dimethyl sulfoxide	DMSO	Roth

Dithiothreitol	DTT	Thermo Fisher Scientific
Disodium phosphate	Na_2HPO_4	Sigma-Aldrich
Ethanol 99%	EtOH	Sigma-Aldrich
Ethydium bromide	EtBr	Sigma-Aldrich
Ethylenediaminetetraacetic acid	EDTA	Roth
Formamide		Sigma-Aldrich
GeneRuler DNA Ladder Mix		Thermo Fisher Scientific
Glacial acetic acid		Merck
Glycerol		Applichem
Glycine		Applichem
Glucose		Sigma-Aldrich
Heparin		Gibco
Hydrogen chloride	HCl	Merck
Hydrogen peroxide	H_2O_2	Sigma-Aldrich
Lithium chloride	LiCl	Sigma-Aldrich
Low-melt agarose		Roth
Magnesium sulphate heptahydrate	$MgSO_4 \cdot 7H_2O$	Merck
Maleic acid 99%		Roth
Methanol	MeOH	Roth
Methylene blue trihydrate	Methylene blue	Sigma-Aldrich
MS-222	tricaine	Sigma-Aldrich
Normal goat serum	NGS	Sigma-Aldrich
N-Phenylthiourea	PTU	Sigma-Aldrich
Nucleotide triphosphates	NTPs	Roche
Orange G		Sigma-Aldrich
Paraformaldehyde	PFA	Sigma-Aldrich
PCR DIG Probe Synthesis mix		Roche
Phenol/Chloroform/Isoamylalcohol pH 4.5	PCI pH 4.5	Roth
Phenol/Chloroform/Isoamylalcohol pH 8	PCI pH 8	Roth
Phosphoric acid	H_3PO_4	Grüssing
Potassium acetate	KAc	AppliChem
Potassium chloride	KCl	AppliChem
Potassium dihydrogen phosphate	KH_2PO_4	Merck

Potassium hydrogen phosphate	K_2HPO_4	Merck
Potassium hydroxide	KOH	Merck
Ribonucleic acid from torula yeast Type VI		Sigma-Aldrich
Ribonucleotide triphosphates	rNTPs	Roche
RiboLock RNase Inhibitor $\left(20\,\mathrm{U\,\mu l^{-1}}\right)$	RiboLock	Thermo Fisher Scientific
RNase-free water		Sigma-Aldrich
Sheep serum		Sigma-Aldrich
Sodium acetate	NaAc	Grüssing
Sodium citrate	$C_6H_5Na_3O_7 \cdot 2H_2O$	Sigma-Aldrich
Sodium chloride	NaCl	Sigma-Aldrich
Sodium dodecyl sulphate	SDS	Serva
Sodium hydrogen phosphate	Na_2HPO_4	AppliChem
Sodium hydrogen phosphate dihydrate	$Na_2HPO_4 \cdot H_2O$	Sigma-Aldrich
Sodium hydroxide	NaOH	Sigma-Aldrich
Sodium tetraborate	borax	Fluka
Sucrose		Roth
Tissue freezing medium	TFM	Leica
Trans-tamoxifen	tamoxifen (TMX)	Sigma-Aldrich
Tris base		Roth
Tris hydrochloride	Tris-HCl	Sigma-Aldrich
Tri-sodium citrate dihydrate	$C_6H_5Na_3O_7 \cdot 2H_2O$	Sigma-Aldrich
Triton-X 100		Sigma
Trizol		Invitrogen
Tween 20		Sigma-Aldrich
X-Gal		Thermo Fisher Scientific
Yeast extract		Roth

5.1.10 Consumables

Table 5.15: Consumables used in this work.

Name	Supplier
Aluminium foil	Paclan
Borosilicate glass capillaries GC100F-10	Harvard apparatus
Filter tips 10 µl, 20 µl, 200 µl, 1.25 ml	STARLAB
Folded filter paper	Sartorius
Glass beads	Roth
Glass bottom dishes	MatTek
Hybond-N$^+$ membrane	Amersham
Latex gloves	semperguard
Microloader tips 10 µl	Eppendorf
Micro pestles 1.5/2.0 ml	Eppendorf
Molding cup trays 6 mm x 12 mm x 5 mm	Polysciences
Nail polish	Rival De Loop
Nitrile gloves	STARLAB
Paper napkins	Kammerer
Pasteur pipettes	Sarstedt
Petri dishes	Greiner
Pipette tips 10 µl, 200 µl, 1 ml	Steinbrenner
Tubes for bacterial cultures, 13 ml	Sarstedt
Reaction tubes 1.5 ml, 2 ml	Sarstedt
Reaction tubes 1.5 ml, 2 ml (safe-lock)	Eppendorf
Reaction tubes for polymerase chain reaction (PCR)	Sarstedt
Scalpel blade	Schreiber Instrumente
Six-well plates	Böttger
Superfrost plus microscope slides	Thermo Scientific
Tubes 15 ml, 50 ml	Sarstedt
Whatman paper	Whatman

5.1.11 Solutions

Solutions for fish work

Table 5.16: Solutions for fish rearing. Ingredients were dissolved in H_2O unless indicated otherwise.

Name	Ingredients	Concentration
1x ERM	NaCl	17 mmol l^{-1}
	KCl	0.4 mmol l^{-1}
	$CaCl \cdot 2H_2O$	0.27 mmol l^{-1}
	$MgSO_4 \cdot 7H_2O$	0.66 mmol l^{-1}
	HEPES pH 7.3	17 mmol l^{-1}
	adjust to pH 7	
1x zebrafish medium	NaCl	1.72 g l^{-1}
	KCl	0.076 g l^{-1}
	$CaCl_2 \cdot 2H_2O$	0.29 g l^{-1}
	$MgSO_4 \cdot 7H_2O$	0.49 g l^{-1}
Medaka hatching solution	Methylene blue	2 mg l^{-1}
	in 1x ERM	
20 x tricaine	tricaine	4 g l^{-1}
	$Na_2HPO_4 \cdot H_2O$	10 g l^{-1}
	in 1x ERM	
	adjust pH to 7-7.5 with 1 N HCl	
1.5% agarose in water	agarose	1.5% w/v
	bring to a boil	
0.8% agarose (low-melt) in 1x zebrafish medium	agarose (low-melt)	0.8% w/v
	bring to a boil in 1x zebrafish medium	
1% agarose (low-melt) in 1x ERM	agarose (low-melt)	1% w/v
	bring to a boil in 1x ERM	
Tamoxifen stock (50 mmol l^{-1})	Trans-tamoxifen	18.5 mg ml^{-1}
	dissolve in DMSO	
50x PTU	PTU	1.65 g l^{-1}

	stir under the hood at 40°C	
1x PTU in zebrafish medium	50x PTU	2% v/v
	dilute in 1x zebrafish medium	

Solutions of general use

Table 5.17: Solutions of general use. Ingredients were dissolved in H_2O unless indicated otherwise.

Name	Ingredients	Concentration
16% PFA	PFA	160 gl^{-1}
	adjust pH to 7.0	
4% PFA/PTW	16% PFA	25 % v/v
	dilute in 1x PTW	
	adjust pH to 7-7.5	
1% PFA/PTW	4% PFA	25 % v/v
	dilute in 1x PTW	
10x PBS	NaCl	1370 mmoll^{-1}
	KCl	27 mmoll^{-1}
	KH_2PO_4	2.4 gl^{-1}
	Na_2HPO_4	14.4 gl^{-1}
1x PTW	10x PBS	10 % v/v
	Tween 20	0.1 % v/v

Solutions for molecular work

Table 5.18: Solutions for molecular work. Ingredients were dissolved in H_2O unless indicated otherwise.

Name	Ingredients	Concentration
1x TAE	Tris base	242 gl^{-1}
	Glacial acetic acid	5.71 % v/v
	EDTA	50 mmoll^{-1}
	adjust to pH 8.5	
EtBr bath	EtBr (10 mgml^{-1}	0.02% v/v
	in 1x TAE	
1% agarose in TAE	agarose	1% w/v
	boil in TAE	
1.5% agarose in TAE	agarose	1.5% w/v

	boil in TAE	
6x Orange G Loading dye	Orange G	$1.2\ \mathrm{mg\,ml^{-1}}$
	Glycerol	20% v/v
10 x oligo annealing buffer	Tris-HCl pH 8	$100\ \mathrm{mmol\,l^{-1}}$
	EDTA pH 8	$10\ \mathrm{mmol\,l^{-1}}$
	NaCl	$1\ \mathrm{mol\,l^{-1}}$
$0.1\ \mathrm{mol\,l^{-1}}$ NaOH	NaOH	$4\ \mathrm{g\,l^{-1}}$
98% EtOH, RNase-free	99% EtOH	98 % v/v
	in RNase-free water	
75% EtOH, RNase-free	99% EtOH	75 % v/v
	in RNase-free water	
70% EtOH, RNase-free	99% EtOH	70 % v/v
	in RNase-free water	
70% EtOH	99% EtOH	70 % v/v
3 M NaAc	NaAc	$246.1\ \mathrm{g\,l^{-1}}$
	in RNase-free water	
4 M LiCl	LiCl	$169.6\ \mathrm{g\,l^{-1}}$
	in RNase-free water	
Fin-clip buffer	Tris-HCl pH 8	$400\ \mathrm{mmol\,l^{-1}}$
	EDTA pH 8	$5\ \mathrm{mmol\,l^{-1}}$
	NaCl	$150\ \mathrm{mmol\,l^{-1}}$
	Tween 20	0.1% v/v
LB medium	Bacto-Tryptone	$10\ \mathrm{g\,l^{-1}}$
	Yeast Extract	$5\ \mathrm{g\,l^{-1}}$
	NaCl	$10\ \mathrm{g\,l^{-1}}$
LB plates	Agar	$15\ \mathrm{g\,l^{-1}}$
	boil in LB medium	
P1	Glucose	$50\ \mathrm{mmol\,l^{-1}}$
	Tris-HCl	$25\ \mathrm{mmol\,l^{-1}}$
	EDTA	$10\ \mathrm{mmol\,l^{-1}}$
	pH 8, store at 4°C	
P2	NaOH	$0.2\ \mathrm{mol\,l^{-1}}$
	SDS	1% w/v

P3	KAc	5 mol l^{-1}
	store at 4°C	
TB medium	Bacto-Tryptone	12 g l^{-1}
	Yeast Extract	24 g l^{-1}
	Glycerol	0.4 % v/v
	KH_2PO_4	2.13 g l^{-1}
	K_2HPO_4	12.54 g l^{-1}
TE buffer	Tris-HCl pH 8.0	10 mmol l^{-1}
	EDTA	1 mmol l^{-1}
TEN-9 buffer	Tris-HCl pH 8.5	100 mmol l^{-1}
	EDTA	10 mmol l^{-1}
	NaCl	200 mmol l^{-1}
	SDS	1% v/v
X-Gal	X-Gal	20 mg ml^{-1}
	in DMSO	

Solutions for immunohistochemistry

Table 5.19: Solutions for immunohistochemistry. Ingredients were dissolved in H_2O unless indicated otherwise.

Name	Ingredients	Concentration
Bleaching solution	H_2O_2	0.3 % v/v
	KOH	0.5 % v/v
	dilute in 1x PTW	
30% sucrose	Sucrose	30% w/v
	dissolve in 1x PTW	
60% glycerol	glycerol	60% v/v
	dilute in 1x PTW	
1% NGS	NGS	10% v/v
	dilute in 1x PTW	
Blocking solution 1	sheep serum	4 % v/v
	BSA	1 % v/v
	DMSO	0.1 % v/v
	dilute in 1x PTW	
Blocking solution 2	sheep serum	4 % v/v
	BSA	0.1 % v/v
	dilute in 1x PTW	

10% NGS	NGS dilute in 1x PTW	10% v/v
DAPI stock solution	DAPI dissolve in DMSO	2 mg ml^{-1}
1.7 N HCl solution	2 N HCl 10 x PBS Triton-X 100	85 % v/v 10 % v/v 5 % v/v
Saturated borax solution	borax	at least 35 g l^{-1}
40% borax/PTW	Saturated borax solution dilute in 1x PTW	40% v/v
ZF pre-blocking solution	PTW BSA DMSO	50 % v/v 1 % w/v 1 % v/v
ZF blocking solution	sheep serum dilute in ZF pre-blocking solution	40 µl ml^{-1}

Solutions for *in situ* hybridisation

Table 5.20: Solutions for *in situ* hybridisation. Ingredients were dissolved in RNase-free H_2O unless indicated otherwise.

Name	Ingredients	Concentration
2x PTW	10x PBS Tween 20	20 % v/v 0.2 % v/v
4% PFA/2x PTW	16% PFA dilute in 2x PTW adjust pH to 7-7.5	25 % v/v
75% MeOH/PTW	100% MeOH dilute in 1x PTW	75 % v/v
50% MeOH/PTW	100% MeOH dilute in 1x PTW	50 % v/v
25% MeOH/PTW	100% MeOH dilute in 1x PTW	25 % v/v
10 µg ml^{-1} Proteinase K/PTW solution	20 mg ml^{-1} Proteinase K dilute in 1x PTW	0.2 % v/v

Glycine/PTW solution	Glycine dilute in 1x PTW	2 mg ml^{-1}
20x SSC (saline sodium citrate buffer)	NaCl	175.3 gl^{-1}
	$C_6H_5Na_3O_7 \cdot 2H_2O$ adjust to pH 7 with 1 N HCl and autoclave	77.42 gl^{-1}
4x SSCT	20 x SSC Tween 20	20% w/v 0.1% v/v
50% Formamide/2xSCCT	4x SSCT	50% v/v
	Formamide	50% v/v
2x SSCT	20 x SSC Tween 20	10% w/v 0.1% v/v
0.2x SSCT	20 x SSC Tween 20	1% w/v 0.1% v/v
Hybridisation mix (Hyb-Mix)	Formamide	50 % v/v
	20x SSC Heparin Ribonucleic acid from torula yeast Type VI Tween 20	25 % v/v 150 µg ml^{-1} 5 mg ml^{-1} 0.1% v/v
Blocking buffer	Sheep serum in 1x PTW prepare fresh and store at 4°C until use	5 % v/v
Anti-Digoxigenin-AP antibody solution	Anti-Digoxigenin-AP Fab fragments dilute in Blocking buffer prepare fresh and store at 4°C until use	0.5% v/v
Pre-staining buffer	Tris-HCl pH 7.5 NaCl Tween 20	100 mmol l^{-1} 100 mmol l^{-1} 0.1% v/v
Staining buffer	Tris-HCl pH 9.5	100 mmol l^{-1}

	NaCl	100 $\mathrm{mmol\,l^{-1}}$
	$MgCl_2$	50 $\mathrm{mmol\,l^{-1}}$
	Tween 20	0.1% v/v
Staining solution	NBT	337.5 $\mathrm{\mu g\,ml^{-1}}$
	BCIP	175 $\mathrm{\mu g\,ml^{-1}}$
	dissolve in staining buffer	
	Do not shake or vortex solution!	
87% glycerol	glycerol	87% v/v

Solutions for Southern Blot

Table 5.21: Solutions for Southern blotting. Ingredients were dissolved in H_2O unless indicated otherwise.

Name	Ingredients	Concentration
0.8% agarose in TAE	agarose	0.8% w/v
	boil in TAE	
Depurination solution	HCl	0.25 $\mathrm{mol\,l^{-1}}$
Denaturation solution	NaOH	0.5 $\mathrm{mol\,l^{-1}}$
	NaCl	1.5 $\mathrm{mol\,l^{-1}}$
Neutralisation buffer	Tris	0.5 $\mathrm{mol\,l^{-1}}$
	NaCl	1.5 $\mathrm{mol\,l^{-1}}$
	adjust pH to 7.2-7.4 with 32% HCl	
20x SSC	NaCl	175.3 $\mathrm{g\,l^{-1}}$
	Sodium citrate	88.22 $\mathrm{g\,l^{-1}}$
	adjust pH to 7.0 with HCl	
	autoclave	
10x SSC	20xSSC	10% v/v
1 $\mathrm{mol\,l^{-1}}$ NaP_i (Church stock buffer)	Na_2HPO_4	0.5 $\mathrm{mol\,l^{-1}}$
	adjust pH to 7.2 with H_3PO_4	
	autoclave	
50 $\mathrm{mmol\,l^{-1}}$ NaP_i	1 $\mathrm{mol\,l^{-1}}$ NaP_i	20% w/v
10x DIG1	Maleic acid	1 $\mathrm{mol\,l^{-1}}$
	NaCl	1.5 $\mathrm{mol\,l^{-1}}$

	adjust pH to 7.5 with NaOH pellets	
	autoclave	
1x DIG1	10x DIG1	10% w/v
1x DIG1 + 0.3% Tween	10x DIG1	10% v/v
	Tween 20	1.5% v/v
	prepare fresh before using	
10x Blocking reagent	Blocking reagent	10% w/v
	dissolve in 1xDIG1	
	autoclave	
1x DIG2	10x Blocking reagent	10% v/v
	dilute in 1xDIG1	
	prepare fresh before using	
1x DIG3	Tris pH 9.5	$0.1\,\mathrm{mol\,l^{-1}}$
	NaCl	$0.1\,\mathrm{mol\,l^{-1}}$
	prepare fresh before using	
Church hybridisation buffer	$1\,\mathrm{mol\,l^{-1}}$ NaP$_\mathrm{i}$	50% v/v
	20% SDS	35% v/v
	EDTA pH 8	0.2% v/v
Church washing buffer	$1\,\mathrm{mol\,l^{-1}}$ NaP$_\mathrm{i}$	4% v/v
	20% SDS	5% v/v
1x DIG3 + CDP Star	CDP Star	$6\,\mathrm{\mu l\,ml^{-1}}$
	dilute in 1xDIG1	
	prepare immediately before using	

5.1.12 Equipment and instruments

Table 5.22: Equipment and instruments used in this work.

Name	Supplier
2.5 µl pipette	Eppendorf
Axio Imager M1 Microscope	Zeiss
Bacterial Shaker INNOVA 44	New Brunswick Scientific
Blot Documentation System	Intas
Cat S20 shaker	neoLab
Centrifuge 5417 C	Eppendorf

Centrifuge 5425	Eppendorf
Centrifuge 5430	Eppendorf
Centrifuge 5430 R	Eppendorf
Centrifuge MC 6	Sarstedt
Centrifuge for PCR tubes	Steinbrenner Laborsysteme
Cold light source for stereomicroscope KL 1500 LCD	Schott
Cryostat CM3050 S	Leica
ErgoOne 10 µl pipette	STARLAB
Fish incubator	Heraeus instruments
Fish incubator	RuMed
Forceps 5, 55 Inox stainless steel	Dumont
Freezer -20°C	Liebherr
Freezer -80°C	Thermo Scientific
Fridge 4°C	Liebherr
Gel chamber	peqLab and custom-made
Hybridisation oven Hybaid Micro-4	MWK Biotech
Incubator 37°C, 60°C	BINDER
Leica TCS SP5	Leica
Leica TCS SP8	Leica
Leica TCS SPE	Leica
Microinjector 5242	Eppendorf
Microwave R-939	Sharp
Needle puller P-30	Sutter Instrument Co USA
Nikon AZ100	Nikon
Nikon DS-Ri1 camera	Nikon
Parafilm	Pechiney Plastic Packaging
pH-meter	Sartorius
Pipetboy acu	Integra biosciences
Pipetman 20 µl pipette	GILSON
Pipetman 200 µl pipette	GILSON
Pipetman 1000 µl pipette	GILSON
Power supply PowerPac Basic	Bio RAD
PowerPac Basic	Bio RAD
Rocking shaker DRS-12	neoLab
Scale EW 2200-2NM	KERN

Spectrophotometer DS-11+	DeNovix
Staining container for immunohistochemistry on cryosections	custom-made
Stratalinker UV Crosslinker	Stratagene
Stereomicroscope Nikon SMZ18	Nikon
Stereomicroscope Zeiss Stemi 2000	Zeiss
Stereomicroscope Zeiss Stemi SV11	Zeiss
Thermocycler	Bio-Rad
Thermomixer Compact	Eppendorf
Thermomixer F1.5	Eppendorf
Tube revolver	Thermo Fisher Scientific
UV-Gel Documentation System	Intas
UV table	Vilber Lourmat
Vortex	Scientific Industries
Water bath	GFL

5.1.13 Software

Table 5.23: Software used in this work.

Name	Reference/Vendor
CCTop	Stemmer et al. [2015]
Eclipse IDE 4.3.1	Eclipse IDE Kepler 4.3.1 [2013]
EPISIM Modeller	Sütterlin et al. [2012]
EPISIM Simulator	Sütterlin et al. [2012]
EZ-C1	Nikon
Fiji distribution of ImageJ	Schindelin et al. [2012]
FileMaker Pro	FileMaker, Inc.
Geneious	Biomatters Limited [Kearse et al., 2012]
ggplot2	Wickham [2016]
Inkscape 0.92	Inkscape 0.92 [2017]
Java	Oracle Corporation
LasX	Leica
matplotlib	Hunter [2007]
MatrixStats	Bengtsson et al. [2014]
Microsoft Office	Microsoft

NIS-Elements AR	Nikon
numpy	Oliphant [2006]
Python 3.6	WinPython distribution of Python [2016]
reshape2	Wickham [2012]
RStudio	R Core Team [2015]; RStudio distribution of R [2015]
SciPy	Jones et al. [2001]

5.1.14 Computational resources

Simulations in this work were in part performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and Arts and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC. Simulations were also performed on the following machines:

Table 5.24: Computers used in this work.

OS	CPU	RAM	GPU	Comments
Windows 7 Pro 64-bit	Intel Core i7-4702MQ	16 GB	NVIDIA GeForce GT 740M 2 GB	Lenovo ThinkPad Edge E540 laptop PC with upgraded RAM
Windows 10 Pro 64-bit	Intel Core i7-8700K 3.70 GHz	16 GB	NVIDIA GeForce GTX 1060 6 GB	custom-built desktop PC

5.2 Methods

5.2.1 Fish husbandry

Medaka (*Oryzias latipes*) and zebrafish (*Danio rerio*) were kept as previously described [Loosli et al., 2000]. Fish were kept in closed stocks with a 14 h light and 10 h dark cycle at Heidelberg University's Centre for Organismal Studies. All experimental procedures were performed according to German animal welfare laws (Tierschutzgesetz §11, Abs. 1, Nr. 1) in accordance with the following permits:

1. husbandry permit number AZ 35-9185.64/BH Wittbrodt
2. line generation permit numbers AZ 35-9185.81/G-145/15 and AZ 35-9185.81/034-G
3. killing permit numbers AZ T-90/14 and AZ T-71-17

5.2.2 Genomic DNA extraction from medaka tissue

Genomic DNA extraction for PCR

To quickly obtain genomic samples, a short protocol was used. Tissue was homogenised in 100 µl fin-clip buffer. Eggs were ground with a pestle, while fin clippings required no pre-treatment. The homogenate was supplemented with 5 µl of 20 $\text{mg}\,\text{ml}^{-1}$ Proteinase K and incubated overnight at 60°C. The next day, 200 µl water were added to the sample before heating to 95°C for 10 min to inactivate Proteinase K. This solution was directly used as a template for PCR.

Genomic DNA extraction for Southern Blot

To obtain high-molecular undegraded deoxyribonucleic acid (DNA), a more thorough protocol was used. Tissue samples were placed in a 2 ml reaction tube and washed twice with water. The liquid was completely removed, then the sample was covered with at least 400 µl TEN-9 buffer with an added 20 µl of Proteinase K, scaling up as necessary. The sample was homogenised with a pestle, then incubated overnight at 60°C.

The following day, tubes were cooled down to room temperature for 5 min before adding 20 µl of RNase A. The ribonucleic acid (RNA) was digested for 15 min at room temperature. Samples were centrifuged at 10600 *g* for 5 min to remove remaining tissue debris, and a defined volume of supernatant was transferred to a 2 ml safe lock reaction tube. PCI pH 8 was added to the sample at a 1.5-fold volume; the sample was mixed by inversion. After incubating 10 min at room temperature, the sample was centrifuged at 10600 *g* for 20 min. The upper aqueous phase was transferred to a new 2 ml safe lock reaction tube and supplemented with 1.5-fold volume of chloroform. The sample was mixed by inverting, and then centrifuged again at 10600 *g* for 20 min. The upper aqueous phase was transferred to a new 2 ml safe lock reaction tube and supplemented with 1-fold volume of isopropanol. After thoroughly mixing, the sample was precipitated at -20°C for at least 30 min. The sample was pelleted by centrifugation 2700 *g* for 5 min. The supernatant was removed and 800 µl of 70% EtOH were used to wash the sample. After another centrifugation at 2700 *g* for 2 min, the supernatant was completely removed by careful pipetting, and the sample was left to dry for 10 min at 60°C. The pellet was dissolved either in 50 µl TE buffer or in 50 µl 1x enzyme buffer for subsequent digestion; pellet dissolution was done for at least 2 h at 60°C. DNA was measured spectrophotometrically and a test gel with 500 ng of DNA was run to assay the quality of the sample. The sample was stored at 4°C until use.

5.2.3 Total RNA extraction from medaka embryos

Wildtype *cab* embryos were collected and reared at 28°C in 1xERM until the desired stage; staging was done according to Iwamatsu [2004]. Eggs were transferred to 2 ml tubes and flash-frozen in liquid nitrogen. The sample was ground with a sterile pestle in 700 µl Trizol, then centrifuged 1 minute at 10600 *g*. Additional 300 µl Trizol were added and stirred with the pestle. 200 µl chloroform were added to the sample and mixed vigorously for 15 seconds by shaking the tube. The sample was left to rest for 10 minutes, and was then centrifuged for 5 minutes at 4°C at 10600 *g*. Roughly 500 µl of the upper clear aqueous phase were transferred to a clean 1.5 ml tube, and mixed with an equal volume of isopropanol. The mixture was left to rest for 10 minutes on ice before centrifuging for 10 minutes at 4°C at 10600 *g* to precipitate the RNA. The supernatant was discarded and the pellet was washed twice with 1 ml 75% RNase-free EtOH; in-between washes, the sample was centrifuged for 10 minutes at 4°C at 10600 *g* and the supernatant was removed by pipetting. The pellet was left to dry at room temperature for a maximum of 5 minutes, and was finally eluted in 20 µl RNase-free water. Elution was aided by gently flicking the tube.

The concentration of RNA was measured spectrophotometrically. If absorption ratios were $\frac{260}{280} \approx 1.8$ and $\frac{260}{230} \approx 2.0$, the RNA quality was assessed electrophoretically: An aliquot of the sample (≈ 400 ng) was mixed with 2x RNA loading dye and diluted with RNase-free water to a minimum volume of 6 µl. The sample was denatured for 10 minutes at 80°C to remove secondary structures, and was loaded on a freshly-made 1% agarose in TAE gel at 100 V for at least 30 min. The gel chamber had previously been cleaned with 0.1 $\mathrm{mol\,l^{-1}}$ NaOH. If the RNA produced strong ribosomal bands and little smearing, it was stored at -80°C.

5.2.4 Generation of cDNA

Total RNA extracted as described in Section 5.2.3 was digested with DNase I (Table 5.25) for 30 min at 37°C. The reaction was stopped by adding 1.5 µl 50 mM EDTA and incubating at 65°C for 10 min.

Table 5.25: DNase I digestion mix. Since DNase I is sensitive to physical denaturing, the mixture should not be vortexed!

Ingredient	Quantity
Sample	5 µg
TURBO DNase I	2 µl
10x DNase I buffer	3 µl
RNase-free water	*ad* 30µl

After DNA digestion, the sample was reverse transcribed using the RevertAid First Strand cDNA Synthesis Kit (Thermo Scientific). The reverse transcription reaction was set up in two steps: First, mix 1 was pipetted (Table 5.26), incubated 5 min at 65°C, and chilled on ice for 1 min. Then, mix 2 was set up using mix 1 (Table 5.27). The reaction was mixed gently, spun down, and incubated for 60 min at 42°C. The reaction was terminated by heating to 70°C for 5 min. The sample containing the complementary DNA (cDNA) was chilled on ice for 1 min and spun down. Remaining RNA in the sample was digested by supplementing with 1 µl RNase H and 19 µl water to a final volume of 40 µl. Finally, the sample was stored at -20°C.

Table 5.26: Reverse transcription mix 1.

Ingredient	Quantity
RNase-free water	*ad* 12 µl
Sample	0.1 - 5 µg
Oligo(dT)$_{18}$ primer	1 µl

Table 5.27: Reverse transcription mix 2. Components were added to mix 1 (Table 5.26).

Ingredient	Quantity
Reverse transcription mix 1 (Table 5.26)	12 μl
5x RT buffer	4 μl
RiboLock (20 $\frac{\text{U}}{\text{ml}}$)	1 μl
dNTP-mix (10 mM)	2 μl
RevertAidTM M-MuLV Reverse Transcriptase (200 U μl^{-1})	2 μl

5.2.5 PCR

PCR was performed either with *Thermus aquaticus* Taq polymerase made in-house (Tables 5.28 and 5.29) or the commercial Q5 polymerase from New England Biolabs (Tables 5.30 and 5.30).

Table 5.28: Recipe for PCR with Taq polymerase.

Ingredient	Quantity	End concentration
RNase-free H_2O	*ad* 20 μl	-
10x Taq polymerase buffer	2 μl	1x
2.5 mmol l^{-1} MgCl$_2$	1.6 μl	0.2 mmol l^{-1}
2.5 mmol l^{-1} dNTP	1.6 μl	0.2 mmol l^{-1}
10 μmol l^{-1} Forward primer	1 μl	0.2 μmol l^{-1}
10 μmol l^{-1} Reverse primer	1 μl	0.2 μmol l^{-1}
DNA template	$\approx 10^{-1} - 10^{2}$ ng	0.005 - 5 ng μl^{-1}
Taq polymerase	0.3 μl	-

Table 5.29: General PCR cycler program for Taq polymerase.

Step	Temperature	Time	Cycles
Initial denaturation	94°C	3 min	1
Denaturation	94°C	25 s	35
Annealing	primer-dependent	25 s	
Extension	72°C	$\approx 60 \frac{\text{s}}{\text{kbp}}$	
Final extension	72°C	10 min	1
Cooling	10°C	10 min	1

Table 5.30: Recipe for PCR with Q5 polymerase.

Ingredient	Quantity	End concentration
RNase-free H_2O	*ad* 50 µl	-
5x Q5 polymerase buffer	10 µl	1x
2.5 $\mathrm{mmol\,l^{-1}}$ dNTP	4 µl	0.2 $\mathrm{mmol\,l^{-1}}$
10 $\mathrm{\mu mol\,l^{-1}}$ Forward primer	2.5 µl	0.5 $\mathrm{\mu mol\,l^{-1}}$
10 $\mathrm{\mu mol\,l^{-1}}$ Reverse primer	2.5 µl	0.5 $\mathrm{\mu mol\,l^{-1}}$
DNA template	$\approx 10^{-1} - 10^{2}$ ng	0.005 - 5 $\mathrm{ng\,\mu l^{-1}}$
Q5 polymerase 5 $\mathrm{U\,\mu l^{-1}}$	0.3 µl	0.03 $\mathrm{U\,\mu l^{-1}}$

Table 5.31: General PCR cycler program for Q5 polymerase. The annealing temperature was calculated using the manufacturer's web-based temperature calculator (`tmcalculator.neb.com`).

Step	Temperature	Time	Cycles
Initial denaturation	98°C	1 min	1
Denaturation	98°C	20 s	35
Annealing	primer-dependent	20 s	
Extension	72°C	$\approx 30\,\frac{\mathrm{s}}{\mathrm{kbp}}$	
Final extension	72°C	10 min	1
Cooling	10°C	10 min	1

5.2.6 Genotyping of adult fish

Individual adult fish were placed between two paper towels soaked in fish water, and a sterile scalpel was used to cut a small piece of the tail fin (fin-clip). The fish was immediately transferred to an individual tank with a 1:2 mixture of medaka hatch medium and fish water. Care was taken to minimize stress to the animals. Genomic DNA of the fin clipping was extracted for use in PCR according to Section 5.2.2. If only a band pattern was assayed by gel electrophoresis, PCR was done on the samples using Taq polymerase (Tables 5.28 and 5.29); if subsequent sequencing was done, the PCR was done using Q5 polymerase (Tables 5.30 and 5.31). Fish were later sorted according to genotype.

5.2.7 Gel extraction and clean-up

To isolate samples through gel electrophoresis, a 1% agarose in TAE gel with large pockets was prepared. Samples were loaded with 6x orange loading dye and 15 µl GeneRuler DNA Ladder Mix, and the gel was run for at least 30 min at 100 V. The gel was stained in an EtBr bath, and a picture was taken while minimizing ultraviolet (UV) exposure. The band was cut out with a scalpel and transferred to a pre-weighed 2 ml tube. The tube carrying the gel block was weighed again to obtain the weight of the gel block. The gel was solubilised and the sample was purified via clean-up on a column of the innuPREP DOUBLEpure Kit (Analytik Jena) following the manufacturer's instructions [Analytik Jena, 2012].

5.2.8 A-tailing and ligation into pGEM-T Easy

Overhangs of 3' adenosine were introduced to a linear sample via A-tailing (Table 5.32):

Table 5.32: A-tailing mix.

Ingredient	Quantity for 15 µl
Linear template	11.75 µl
10x PolBuffer B	1.5 µl
dATP (2 mM)	1.5 µl
Taq Roboklon $\left(5\,\mathrm{U\,\mu l^{-1}}\right)$	0.25 µl

The A-tailing mix was incubated at 72°C for 25 minutes, and then mixed with the commercially available vector backbone pGEM-T Easy (Promega; Table 5.33):

Table 5.33: Ligation mix for pGEM-T Easy. The 2x Rapid Ligation Buffer must be vigorously vortexed before use.

Ingredient	Quantity
2x Rapid Ligation Buffer	5 µl
pGEM-T Easy	0.5 µl
A-tailed product	3.5 µl
T4 DNA Ligase $\left(3\,\mathrm{U\,\mu l^{-1}}\right)$	1 µl

The ligation reaction was incubated for at least 10 min at room temperature.

5.2.9 Transformation of bacteria

If a blue-white selection was performed, LB plates with the appropriate antibiotic resistance were coated with 40 µl of X-Gal solution and pre-heated at 37°C for at least 5 min. 50 µl MachT1 cells were allowed to slowly defrost on ice with 5 µl of the sample to be transformed. The cells were heat shocked at 42°C for 30 s, followed by immediately chilling on ice for 2 min. The cells were allowed to grow in 400 µl TB medium for 1 h at 37°C while shaking. Aliquots of 100-300 µl transformed bacteria were spread onto the plates with glass beads before incubating overnight at 37°C.

5.2.10 Bacterial mini- and midi-preparation

Mini preparation

For mini-preparation (miniprep), individual bacterial colonies were hand-picked with a sterile toothpick and inoculated in 3 ml of LB medium supplemented with an appropriate antibiotic for selection. The inoculate was incubated at least 8 h in a shaking incubator at 180 rpm at 37°C. 2 ml of bacterial culture were pelleted by centrifugation at 20800 *g* for 2 min; the remaining bacterial culture was stored at 4°C. After discarding the supernatant, the pellet was resuspended in 200 µl P1 solution by vigorous mixing. Cells were lysed by adding 200 µl P2 solution, mixing by inverting the tube several times, and incubation for up to 5 min at room temperature. The lysis was stopped by adding 200 µl P3 and inverting several times. The solution was chilled for 5 min on ice, and then centrifuged at 20800 *g* for 20 min. The supernatant was transferred to a new reaction tube where it was mixed with 500 µl isopropanol to precipitate the DNA. After centrifugation for 15 min at 15300 *g*, the sample was washed with 500 µl 70% EtOH. The sample was centrifuged again for 5 min at 15300 *g*, and the supernatant was carefully decanted. Any remaining EtOH was pipetted out and left to air-dry. Finally, the sample was eluted in 50 µl RNase-free water. DNA concentration was measured spectrophotometrically. Minipreps were stored at -20°C.

Midi preparation

For midi-preparation (midiprep), 50 µl of miniprep culture that had been stored at 4°C were inoculated in a 250 ml Erlenmeyer flask with 50 ml LB medium supplemented with antibiotics. After at least 8 h of growth at 37°C in a shaking incubator at 180 rpm, the bacteria were pelleted at 4°C 3100 *g* for 30 min. The midi-preparation was performed according to the manufacturer's instruction of the QIAGEN Plasmid Midi Kit (QIAGEN) with the following modifications: After neutralisation by P3, instead of the centrifugation step the solution was dripped into a pre-equilibrated QIAGEN column through a funnel equipped with a folded paper filter; further, the final elution step was performed with 80 µl TE. Sample concentration was measured spectrophotometrically. If necessary, the sample was diluted until a final concentration of roughly 1 $\mu g \mu l^{-1}$. Midipreps were stored at -20°C.

5.2.11 Enzymatic digest

Digestion for quality control

A quality control digest (test digest) was performed on an aliquot of the sample of interest to verify that the correct cassette inserted into the plasmid backbone. The digestion mix (Table 5.34) was incubated at least 1 hour at 37°C (shorter times were possible with FastDigest enzymes from Thermo Fisher Scientific), and was then loaded on a gel to visualise the band pattern. Up to two samples per condition that generated the correct band pattern were sent for sequencing to confirm the insertion.

Table 5.34: Generalised enzymatic digestion for quality control.

Ingredient	Quantity
RNase-free water	*ad* 20 µl
10x buffer	2 µl
DNA template	up to 1 µg
Restriction enzymes	0.3 µl each

Digestion for cloning

When digesting greater quantities of material for cloning, a larger amount of volume and starting material was used (Table 5.35). The digest was incubated up to once overnight at 37°C, and was then loaded on a gel to extract the desired band via the the innuPREP DOUBLEpure Kit (Analytik Jena) following the manufacturer's instructions (Section 5.2.7) [Analytik Jena, 2012].

Table 5.35: Generalised digestion mix for cloning.

Ingredient	Quantity
RNase-free water	*ad* 50 µl
10x buffer	5 µl
DNA template	5-10 µg
Restriction enzymes	1 µl each

5.2.12 Generation of sgRNA

Design and cloning

Primers used for generating sgRNAs were designed with help from Dr Thomas Thumberger using CCTop [Stemmer et al., 2015]. The general procedure for generating sgRNAs consisted of: Annealing two primers to a double-stranded oligonucleotide (Table 5.36), performing a serial dilution (Tables 5.37 and 5.38), and finally cloning the oligonucleotide into a standard plasmid backbone (Table 5.39) for subsequent transcription (Table 5.40) and RNA extraction.

For generating sgRNA for targeting the low-density lipoprotein receptor-related protein 2 gene a (*lrp2a*) locus, the following primer pairs were used: JW 3451 + JW 3452; JW 3453+ JW 3454; JW 3455 + JW 3456.

Table 5.36: 50 µmol l^{-1} oligonucleotide mixture.

Ingredient	Quantity
H_2O	*ad* 20 µl
Forward primer (200 mM)	5 µl
Reverse primer (200 mM)	5 µl
10 x oligo annealing buffer	2 µl

The primers were annealed by heating the mixture in Table 5.36 to 95°C for 4 min in a PCR machine, and then were cooled down slowly to room temperature on the bench. After spinning down and mixing gently, the sample was diluted twice in series:

Table 5.37: First dilution: 500 nmol l^{-1} oligonucleotide mixture.

Ingredient	Quantity
H_2O	99 µl
Sample from Table 5.36	1 µl

Table 5.38: Second dilution: 10 $\text{nmol}\,\text{l}^{-1}$ oligonucleotide mixture.

Ingredient	Quantity
H_2O	44 µl
10 x oligo annealing buffer	5 µl
Sample from Table 5.37	1 µl

The diluted sample was used for ligation into the sgRNA backbone DR274 (internal stock #3632), which was previously digested with BsaI (Section 5.2.11 Table 5.35) and gel-extracted (Section 5.2.7):

Table 5.39: Reaction mix for sticky-end ligation of oligonucleotides.

Ingredient	Quantity
10x T4 DNA ligase buffer	1 µl
BsaI pre-digested DR274 (#3632)	1 µl
Sample from Table 5.38	7 µl
T4 DNA Ligase $\left(5\text{U}\,\mu\text{l}^{-1}\right)$	1 µl

The ligation reaction was incubated at room temperature for at least 10 min and then transformed into bacteria (Section 5.2.9). The plasmid was isolated via miniprep (Section 5.2.10), and a test digest was performed with the enzyme BsaI (Section 5.2.11 Table 5.34); incorporation of the correct insert was confirmed by sequencing. Bacterial cultures carrying the correct plasmid were amplified and the plasmid extracted via midiprep (Section 5.2.10).

Transcription of sgRNA

10 µg of the template plasmid were enzymatically digested with Dra I FD (Section 5.2.11 Table 5.35) and the 300 bp band corresponding to the insert (transcription template) was gel extracted (Table 5.2.7). The transcription template was transcribed using the MEGAshortscript T7 Transcription Kit according to the manufacturer's protocol [Invitrogen, 2010]:

Table 5.40: T7 transcription mixture. Components were pipetted in the order shown.

Ingredient	Quantity
RNase-free H_2O	*ad* 20 µl
10x T7 reaction buffer	2 µl
T7 ATP solution (75 mM)	2 µl
T7 CTP solution (75 mM)	2 µl

T7 GTP solution (75 mM)	2 µl
T7 UTP solution (75 mM)	2 µl
Template DNA	300 ng
T7 enzyme	2 µl

The reaction was pipetted with filter tips. At all steps care was taken not to vortex the sample. The frozen kit reagents were thawed on ice; the 10x T7 reaction buffer was kept at room temperature after thawing. The solution was thoroughly mixed without vortexing. Transcription was performed at 37°C for 3 h. DNA digestion was done by adding 1 µl of TurboDNase $\left(2\,\mathrm{U}\,\mu\mathrm{l}^{-1}\right)$ from the kit to the reaction, and incubating at 37°C for 15 min. The sample was placed on ice and the clean-up mix was set up:

Table 5.41: Clean-up mix.

Ingredient	Quantity
DNase-digested sample	21 µl
RNase-free H_2O	250 µl
Ammonium Acetate Stop Solution	30 µl

After thoroughly mixing, the sample was immediately processed for RNA extraction.

RNA extraction via phenol-chloroform-isopropanol

The sample in Table 5.41 was supplemented with 300 µl PCI pH 4.5 and mixed vigorously. After letting the sample rest 5 min at room temperature, phases were separated by centrifuging at 17900 *g* 4°C for 15 min. Roughly 260 µl of the upper aqueous phase containing the RNA were transferred to a new reaction tube and vigorously mixed with an equal volume of chloroform. After letting the sample rest 5 min at room temperature, phases were separated again by centrifuging at 17900 *g* 4°C for 15 min. Roughly 210 µl of the upper aqueous phase were transferred to a new tube and mixed with twice the volume of 99% EtOH to precipitate the RNA. Precipitation was performed overnight at -20°C. After pelleting by centrifugation at 20800 *g* 4°C for 25 min, the sample was washed twice with 75% ice-cold RNase-free EtOH. In-between washes, the sample was centrifuged again at 20800 *g* 4°C for 8 min. Finally, the pellet was resuspended in 30 µl RNase-free H_2O. After measuring concentration and quality spectrophotometrically, a small aliquot of the RNA was mixed with RNA loading dye and denatured for 10 min at 80°C before loading on a freshly-made 1.5% agarose in TAE gel for quality control. The gel chamber had previously been cleaned with 0.1 $\mathrm{mol\,l^{-1}}$ NaOH. Samples that passed the quality control were diluted to 150 nanog/µl and were stored at -80°C until use.

5.2.13 Cloning of donor cassette for knock-in into *Irp2a* locus

Q5 polymerase was used to amplify a 500 bp stretch upstream of the sgRNA target site using primers JW 3457 and JW 3460 on medaka genomic DNA (Tables 5.30 and 5.31). To generate sticky ends, the primers included overhangs with a BamHI and a KpnI site, respectively. The PCR product was gel extracted (Section 5.2.7) and digested with BamHI HF and KpnI HF (Section 5.2.11 Table 5.35). The digest was cleaned up with the innuPREP DOUBLEpure Kit (Analytik Jena) following the manufacturer's instructions for PCR samples [Analytik Jena, 2012]. In parallel, the entry vector #3237 was digested with BamHI HF and KpnI HF (Section 5.2.11 Table 5.35). The band at roughly 3000 bp representing the vector backbone was gel extracted (Section 5.2.7). The digested insert and vector backbone were ligated (Table 5.50):

Table 5.42: Ligation mix for generating plasmid #4279.

Ingredient	Quantity
RNase-free H_2O	*ad* 10 µl
10x T4 DNA ligase buffer	1 µl
JW 3457-3460 PCR product; BamHI-KpnI digested	3 µl
#3237 backbone; BamHI-KpnI digested	1 µl
T4 DNA Ligase $\left(5\ \mathrm{U\,\mu l^{-1}}\right)$	1 µl

The ligation was incubated at least 10 min at room temperature, and was subsequently transformed into bacteria (Section 5.2.9). After miniprep (Section 5.2.10), the sample was assayed by testdigest (Section 5.2.11 Table 5.34) with the enzymes BamHI-HF and KpnI-HF. Samples that were confirmed to have the correct insert by sequencing were amplified by midiprep (Section 5.2.10), and used for golden gateway assembly [Kirchmaier et al., 2013]:

Table 5.43: Golden gateway reaction mix for generating plasmid #4280

Ingredient	Quantity
RNase-free H_2O	*ad* 10 µl
10x FastDigest buffer	1 µl
Entry vector 1 - plasmid #3776	50 ng
Entry vector 2 - plasmid #4279	50 ng

Entry vector 3 - plasmid #4170	50 ng
Entry vector 4 - plasmid #4044	50 ng
Entry vector 5 - plasmid #3190	50 ng
Destination vector - plasmid #3900	50 ng
BsaI FD	0.3 µl
ATP (10 mM)	2 µl
T4 DNA Ligase $\left(30\,\mathrm{U\,\mu l^{-1}}\right)$	1 µl

The golden gateway reaction was run in a PCR cycler with the following settings:

Table 5.44: Golden gateway cycler program.

Step	Temperature	Time	Cycles
BsaI activity	37°C	33 min	10
T4 Ligase activity	15°C	20 min	
Cooling	25°C	1 s	1

The ligation product was transformed into bacteria (Section 5.2.9), miniprepped (Section 5.2.10), and assayed via testdigest with the enzyme XhoI (Section 5.2.11 Table 5.34). Samples that were confirmed to have the correct insert by sequencing were amplified via midiprep (Section 5.2.10), and stored at -20°C until use.

5.2.14 Wholemount *in situ* hybridisation

Probe generation

Probes for *in situ* were generated via PCR using Q5 polymerase (5.30 and 5.31) using cDNA of stage 39 embryos as a template. The following primer pairs were used: JW 3501 + JW 3504; JW 3503 + JW 3505. The PCR product was electrophoretically run on a gel and the band of interest was gel extracted (Section 5.2.7), A-tailed and ligated into pGEM-T Easy (Section 5.2.8), and transformed into bacteria (5.2.9). After miniprep of the plasmid (5.2.7), a testdigest was performed to verify if the probe had inserted in forward or reverse orientation (Table 5.34 using enzymes listed in Table 5.45).

Table 5.45: *In situ* hybridisation probes created in this work and enzymes used for testdigest.

Internal stock number	Enzyme	Expected fragment size (in bp)
#4379	PstI	forward: 3201 + 463
		reverse: 3426 + 238
#4381	PstI	forward: 3427 + 253
		reverse: 3216 + 509

After sequencing confirmed the orientation, plasmids were linearised at the appropriate position (protocol in Table 5.35 with enzymes from Table 5.46) using 10 µg of template DNA to obtain sufficient substrate for the transcription reaction. To ensure complete linearisation, the digestion was done overnight at 37°C.

Table 5.46: Enzyme used for linearisation and for transcription.

Internal stock number	Enzyme	RNA polymerase for transcription
#4379	AatII	SP6
#4381	AatII	SP6

Before proceeding, the quality of the linearisation was checked on a gel. Samples that passed this quality check were purified via ethanol/sodium acetate precipitation (Table 5.47):

Table 5.47: Ethanol/sodium acetate precipitation mix.

Ingredient	Quantity
Sample	48 µl
3 M NaAc pH 5.2	4.8 µl
98% EtOH	132 µl

The mixture was left to precipitate at -20°C for 20 min, and was then centrifuged at 20800 *g* for 15 min at 4°C. EtOH was removed carefully by pipetting and the sample was left to air dry 10 min at room temperature. Finally, the sample was eluted in 15 µl RNase-free water, and the concentration was measured spectrophotometrically.

Samples with at least 40 ngμl^{-1} were used for transcription (Table 5.48):

Table 5.48: Transcription mix.

Ingredient	Quantity
10x Transcription buffer	2 µl
DTT (100 mM)	2 µl

Dig-UTP (10 mM)	0.7 µl
rNTPs	1.3 µl
RiboLock	1 µl
Linearised template	11 µl
RNA polymerase	2 µl

For samples to be transcribed with SP6 polymerase, the transcription mix was incubated four 4 h at 37°C. Afterwards, 1 µl TURBO DNase I $\left(2\mathrm{U}\,\mu\mathrm{l}^{-1}\right)$ was added to digest the DNA and the sample was incubated for a further 15 min at 37°C.

The samples were then purified by LiCl precipitation (Table 5.49):

Table 5.49: LiCl precipitation mix.

Ingredient	Quantity
Sample	21 µl
4 M LiCl	2.1 µl
98% EtOH	66 µl

After pipetting everything together, the sample was mixed well and left to precipitate for 20 min at -80°C. Subsequently, the sample was centrifuged for 15 min at 20800 *g* at 4°C. The EtOH supernatant was carefully removed by pipetting. The pellet was washed by gently pipetting 400 µl of RNase-free 70% EtOH, taking care not to mix the sample. Another centrifugation step was done for 15 min at 20800 *g* at 4°C. The sample was then let to air dry for 10 min. Finally, the pellet was redissolved in 20 µl RNase-free water. Samples were either stored directly at -80°C, or first further diluted with 150 µl Hybridisation mix (HybMix) and then stored in safe lock tubes at -80°C.

Sample preparation

Eggs were collected from the albino *heino* line and reared to the desired stage; staging was done according to Iwamatsu [2004]. Eggs were fixed in 4% PFA/2x PTW for 4 h at room temperature while gently shaking. Fixed embryos were stored at 4°C until use.

Fixed eggs were washed twice in PTW to remove traces of PFA; the eggshells were removed mechanically by dissection with forceps at a stereoscopic microscope. The dechorionated embryos were transferred to a 2 ml safe-lock tube. After washing 4 times for 5 min each with PTW, embryos were bathed in 100% MeOH. The tube with the samples was left to rest at room temperature until embryos sunk to the bottom, indicating sample saturation with 100% MeOH. Once this point was reached, the old MeOH solution was exchanged by fresh one. Tubes were stored at least once overnight at -20°C until use.

Embryos were rehydrated in steps by transferring from 100% MeOH to 75% MeOH/PTW, then from 75% MeOH/PTW to 50%MeOH/PTW, then from 50%MeOH/PTW to 25%MeOH/PTW, and finally from 25%MeOH/PTW to PTW. Each rehydration step was performed for 5 minutes at room temperature while gently shaking. Afterwards, embryos were washed once again in PTW.

To permeabilise the tissue, embryos were digested in 10 $\mu g\,ml^{-1}$ Proteinase K/PTW solution. The duration of the digest depended on the stage of the embryos; stage 26 embryos were incubated for 15 min, stage 33/34 embryos were incubated for 1 h. The digestion reaction was stopped by saturating the enzyme with Glycine/PTW solution. Samples were then post-fixed in 4% PFA/2xPTW for 20 min, and washed 5 times for 5 min each in PTW while gently shaking.

Sample hybridisation and staining

All steps entailing HybMix were performed in safe lock tubes.

Transcribed RNA samples were diluted with 150 µl HybMix; if samples were previously diluted with HybMix, this step was skipped. A 6 µl aliquot of this mixture was further diluted in 300 µl HybMix. This final dilution was denatured for 10 min at 80°C before use.

HybMix was slowly thawed at room temperature. Embryos were transferred to 2 ml safe lock tubes, the PTW was removed and replaced by 1 ml HybMix. The sample was left to rest at room temperature until embryos sunk to the bottom of the tube, indicating saturation with HybMix. The HybMix solution was exchanged by a fresh 1 ml aliquot; samples were pre-hybridised for up to 2 h at 65°C in a water bath. The HybMix was carefully removed leaving a wet film covering the samples. Next, the denatured hybridisation probe was immediately added to the sample. Hybridisation was performed overnight at 65°C in a water bath.

Reagents for washing (50% formamide/2xSCCT, 2xSCCT, 0.2xSCCT) were preheated at 65°C. Samples were washed several times at 65°C in 2 ml safe lock tubes a water bath: The first two washing steps were in 50% formamide/2xSCCT for 30 min each, then followed two 15 min washing steps in 2xSCCT, and finally two 30 min washing steps in 0.2xSCCT.

After washing, samples were blocked for at least 1 h with 2 ml of blocking buffer at room temperature while gently shaking. The medium was changed with 400 µl anti-Digoxigenin-AP antibody solution; antibody incubation was performed overnight at 4°C on a turning wheel.

Unbound antibodies were washed out by 6 consecutive 10 min washes in PTW. Following these wash steps, embryos were first equilibrated in prestaining buffer by washing for twice 5 min each, and were then equilibrated in staining buffer again by washing twice 5 min each. Embryos were transferred to staining solution and incubated in the dark while gently shaking. Every 15 min colour development was assessed. The colour reaction was stopped by washing 3 times in PTW for 5 min each while gently shaking.

Further washes were done to remove unspecific staining: Both a MeOH- and a EtOH-based protocol were tested. As both protocols generated identical results, the EtOH version was preferred due to its lower toxicity. Embryos were washed 3 times for 10 min each in 100% EtOH, followed by 3 washes for 5 min each in PTW. Embryos were post-fixed in 4% PFA/2xPTW for 20 min at room temperature, then washed again for 3 times 5 min each in PTW. Finally, the yolk was excised mechanically using forceps. Embryos were transferred to 87% glycerol for storage at 4°C. Photos of stained embryos were taken at an upright Zeiss Axio imager M1 stereoscopic microscope.

5.2.15 Southern blot

Probe synthesis

A probe for detecting GFP was synthetised using the PCR DIG Probe Synthesis Kit (Roche):

Table 5.50: Southern blot probe synthesis reaction. All components except for primers, template, and water are from the kit.

Ingredient	Labelled probe	Unlabelled control
RNase-free H_2O	*ad* 50 µl	
10x PCR DIG Probe Synthesis buffer	5 µl	
PCR DIG Probe Synthesis Mix	5 µl	–
dNTP stock solution	–	5 µl
Forward primer (10 $\mu mol\,l^{-1}$) - JW 1458	2.5 µl	
Reverse primer (10 $\mu mol\,l^{-1}$) - JW 1745	2.5 µl	
Enzyme mix	0.75 µl	
Plasmid template - #4280	10-100 pg	

Table 5.51: Cycler program for GFP probe synthesis.

Step	Temperature	Time	Cycles
Initial denaturation	95°C	2 min	1
Denaturation	95°C	30 s	35
Annealing	60°C	30 s	
Extension	72°C	40 s	
Final extension	72°C	7 min	1
Cooling	10°C	10 min	1

Quality of the reaction was checked electrophoretically; due to incorporation of DIG-labelled nucleotides, the labelled probe ran at 1000 bp, while the unlabelled control ran at 700 bp. The probe was purified using the innuPREP DOUBLEpure Kit (Analytik Jena) following the manufacturer's instructions for PCR samples [Analytik Jena, 2012]. The probe was stored at -20°C until use.

Southern blotting

Genomic DNA was extracted as described in Section 5.2.2, "Genomic DNA extraction for Southern Blot". DNA was digested for at least 3 h at 37°C:

Table 5.52: Genomic DNA digestion for Southern Blot.

Ingredient	Quantity
Genomic DNA	at least 10 µg
10x buffer	2.5 µl
Enzyme	1 $U\mu g^{-1}$ genomic DNA; maximum 10% of reaction volume
Total reaction volume	25 µl

The digested DNA was loaded on a 0.8% agarose in TAE gel of a predefined volume, such that each well pocket fit exactly 30 µl of solution (equivalent to the amount in Table 5.52 plus 5 µl of loading dye). The gel was run at 90 V for 2 h. The gel was stained in a freshly prepared EtBr bath, the gel was documented both under UV and visible light next to a ruler to create a reference image. Afterwards, the gel was bathed in depurination solution for 30 min while shaking in the dark. After rinsing the gel with water, the gel was bathed in denaturation solution for 30 min while shaking in the dark. The pH was recovered by bathing the gel in neutralisation buffer for 30 min while shaking in the dark. This step was repeated as many times as necessary until the pH was between 7.0 and 7.5.

Finally the blotting pyramid was mounted. The gel was placed upside down on a platform lined with Whatman paper hanging on top of a bath of 10x SSC; around the gel parafilm strips were placed to prevent sideways capillary fluid movement from the underlying Whatman paper. A hybond-N$^+$ membrane of exactly the same dimensions as the gel was placed on top of the gel. At least 5 leaves of Whatman paper cut to the exact dimensions of the gel were placed on top of the hybond-N$^+$ membrane. A roughly 10 cm-tall stack of paper napkins were placed on top and held down with a small weight. The napkins were exchanged after 20 min to promote capillary flow. Blotting was done overnight.

Probe hybridisation and detection

The hybond-N$^+$ membrane was marked on one side then washed with 50 mmol l^{-1} NaP$_i$ for 5 min while shaking. After placing the membrane on a clean dry piece of Whatman paper, it was crosslinked with UV light at the UV Stratalinker equipment using 120 mJ and 120 s exposure time. The membrane was rolled into a hybridisation bottle with the DNA facing the center. The membrane was incubated with 15 ml of Church hybridisation buffer at 65°C for at least 30 min while rotating gently. Hybridisation probe was diluted in Church hybridisation buffer at 2 µl ml^{-1} and was heated to 95°C for 10 min to melt the double-stranded probe. The Church hybridisation buffer on the membrane was decanted and replaced with the probe dilution. Hybridisation was done at 65°C overnight while rotating.

After hybridisation, the membrane was washed with pre-heated 65°C Church hybridisation buffer for 10 min in a tray while shaking. A second washing step was performed with room temperature Church hybridisation buffer. Then, the membrane was washed with 1xDIG1 + 0.3% Tween for 5 min. To minimize volume of solutions, the membrane was sealed in plastic foil, and the solutions were poured into a small cut made in one of the corners. Blocking was performed with 1x DIG2 for at least 30 min. Anti-Digoxigenin-AP Fab fragments (Roche) were diluted 1 to 10000 in blocking reagent immediately prior to incubating with the membrane for 30 min while gently shaking. Afterwards, the membrane was washed twice with 1xDIG1 + 0.3% Tween for 20 min. After one more wash in 1x DIG3 for 5 min, the membrane was incubated with 1x DIG3 + CDP Star for up to 5 min. Immediately after removing the liquid the chemiluminescence was imaged. Settings on the Intas machine were: 40 repeats of 45 s exposures in sequential integrate mode. A reference picture was taken of the membrane next to a ruler.

5.2.16 Medaka microinjections for CRISPR/Cas9 experiments

Microinjections were performed as previously described [Rembold et al., 2006]. Briefly, *cab* medaka males were separated from the females on the day before the injection, then re-united the following morning. Freshly fertilised eggs were collected, separated, and immobilised in an 1.5% agarose in water gel mold. Borosilicate needles were pulled in a vertical needle puller. Microinjections were performed with a semi-automated microinjection setup into the cytosol of 1-cell stage eggs. Staging was performed according to Iwamatsu [2004].

For performing CRISPR/Cas9 experiments, the following injection mixture was prepared on ice:

Table 5.53: Injection mix for CRISPR/Cas9 experiments.

Ingredient	Quantity
RNase-free water	ad 10 µl
Cas9 mRNA	150 ngμl^{-1}
sgRNA	15-30 ngμl^{-1} each

If a donor cassette was to be integrated in the genome, the plasmid was added to the mixture at an end concentration of 10 ngμl^{-1}.

5.2.17 Photos of entire fish

An agarose mold was prepared by pouring liquid 1.5% agarose in water into a petri dish and placing pipette tips of the appropriate size into the liquid. Fish were anesthetized with tricaine and placed in the agarose-coated petri dish bathed in 1x ERM or 1xZFM. Photos were taken through a Nikon DIGITAL SIGHT DS-Ri1 camera mounted on a Nikon SMZ18 stereomicroscope. An analog scale bar was photographed at the same magnification. If needed, the pictures were stitched together using the ImageJ plugin "pairwise stitching" [Preibisch et al., 2009]. Image scale was defined using the ImageJ function "set scale" (menu path: [Analyse > Set Scale...]).

5.2.18 Transplantation of blastula stage cells

Transplantations were performed by Prof Dr Lázaro Centanin as described in Centanin et al. [2011].

5.2.19 Tamoxifen induction of cre-ERT2

Tamoxifen was diluted in 1x ERM to a final concentration of 5-10 μmol l^{-1}. Fish were transferred to tamoxifen-containing medium and incubated in an opaque box at 28°C for at least 3 hours. To wash out tamoxifen, the fish medium was exchanged at least three times with 5 minute intervals between each medium exchange.

5.2.20 Heat shock induction of cre-NLS

Fish were heat shocked by one of two methods:

1. Medium-based method: Fish were cold-shocked with 4°C 1x ERM for 15 min on the lab bench followed by bathing in 42°C 1x ERM for 2 hours in 37°C incubator.
2. PCR-based method: Single fish were transferred to PCR tubes with 50 µl 1x ERM. Tubes were subjected to at least 4 cycles of at least 5 min each of alternating 8°C and 39°C in a PCR machine before transferring fish back into petri dishes at room temperature.

5.2.21 Thymidine analogue incorporation assay

Thymidine analogues were diluted to the desired final concentration in 1x ERM. Fish were transferred to thymidine analogue-containing medium and incubated in an opaque box at 28°C for the desired amount of time. To wash out the thymidine analogues, the fish medium was exchanged at least three times with 5 min waiting intervals between each medium exchange.

BrdU was used at a concentration of 2.5 mmol l^{-1}. Different concentrations of EdU were tested; staining was weak when using 100 μmol l^{-1}, and was best with 500 μmol l^{-1}. A concentration of 1 mmol l^{-1} EdU appeared to cause cellular damage and stained too strongly, bleeding into other acquisition channels.

5.2.22 Fixation of fish

Fish were euthanised with an overdose of tricaine in 1x ERM. When gill movements ceased and fish no longer responded to pressure on the caudal fin, they were transferred to a solution of 4% PFA/PTW. Fixation was performed overnight at 4°C on a rocking shaker.

5.2.23 Dissection of the retina for wholemount preparation

Dissection of whole retinae was performed at a stereomicroscope with the fixed fish resting on an agarose mold bathed in 1x PTW. The agarose mold was created by placing pipette tips of the appropriate size on unhardened 1.5% agarose in water that had been poured into a petri dish.

First, the conjunctiva (the connective tissue between cornea and skin) was peeled off by pinching and pulling with fine forceps. Then, the cornea was punctured and carefully shredded. For samples of the *Gaudí*^{RSG} line, the lens was removed by first cutting the lens filaments, and then by pulling on the lens with a forceps. The forceps was then inserted between the eye and its socket to cut the optic nerve and ocular muscles, allowing the eye to be enucleated and immobilised in the agarose. Finally, the sclera and choroid were carefully punctured and peeled off until the entire retina became visible.

If necessary, the retina was further dissected by creating a small puncture at the distal end near the CMZ and separating the ONL and RPE from the INL and GCL.

5.2.24 Immunostaining of wholemount samples

Immunostaining of medaka retinae

If necessary, melanin pigment was bleached by bathing samples in bleaching solution in a six-well plate on a shaker at room temperature until the samples became completely clear.

To increase accessibility of the tissue to antibodies, the samples were permeabilised in an ice-cold solution of acetone for 10-15 min at -20°C inside 2 ml reaction tubes. Afterwards, samples were transferred to a six-well plate, rinsed in 1x PTW, and washed for three times in 1x PTW for 5 min each at room temperature on a shaker. Samples were returned to a 2 ml reaction tube for blocking with blocking solution. Both blocking solution 1 and blocking solution 2 were equally effective. This blocking step was performed for at least 1 hour at room temperature on a tube revolver. Samples were transferred to a PCR tube and incubated in primary antibodies diluted in blocking solution at least once overnight at 4°C on a tube revolver. Samples were returned to the six-well plate for rinsing and washing with PTW, then transferred into a new PCR tube for incubation with secondary antibodies and DAPI (1:500 dilution of stock) in blocking solution. Incubation was done at least once overnight at 4°C on a tube revolver. From this point on, all steps were performed in the dark by covering tubes or plates with aluminium foil.

BrdU staining was performed after other antibodies had been stained, and required an antigen retrieval step. Samples were rinsed in 1x PTW, washed at least once for 5 min in 1x PTW, then post-fixed in 4% PFA/PTW for 1 hour at room temperature. Antigen retrieval consisted of treatment for 45 min in 1.7 N HCl solution in a six-well plate on a shaker. The solution was washed out by rinsing with 1x PTW and washing three times 5 min with 1x PTW in a six-well plate on a shaker. The pH was recovered by incubating samples in 40% borax solution for 10 min in a six-well plate on a shaker. After rinsing and washing at least once for 5 min in 1x PTW, samples were transferred to a 1.5 ml reaction tube for blocking with blocking solution for at least 1 hour at room temperature in a tube revolver. Samples were then transferred to PCR tubes for incubation with primary antibody in blocking solution. Incubation was done at least overnight at 4°C. Samples were rinsed and washed at least three times for 5 min in 1x PTW before incubation with secondary antibody in blocking solution. This incubation step was also done at least once overnight at 4°C. Samples were rinsed and washed at least twice for 5 min in 1x PTW.

EdU staining was performed after immunostaining and BrdU staining according to the kit manufacturer's protocol [ThermoFischer Scientific, 2011].

Immunostained samples were stored in 1% PFA/PTW at 4°C or in 100% EtOH at -20°C.

Immunostaining of zebrafish embryos

Fixed zebrafish embryos were cleared in 100% MeOH at -20°C for at least a week prior to immunostaining. After step-wise rehydration with PTW and washing with PTW for 5 min while gently shaking, embryos were mechanically dechorionated. Permeabilisation was done in ice-cold acetone at -20°C for 7 min. Embryos were washed 5 min in ZF pre-blocking solution before blocking in ZF blocking solution for at least 1 h. Samples were transferred to PCR tubes for incubation with primary antibodies diluted in ZF blocking solution. Primary antibody incubation was done for 18 h at 4°C. Samples were washed 4 times in ZF blocking solutions and transferred to new PCR tubes. Embryos were incubated in secondary antibody solution diluted in ZF blocking solution for 18 h at 4°C. After washing twice for 5 min in ZF blocking solution, samples were rinsed in PTW. Samples were stored in 1% PFA/PTW at 4°C.

5.2.25 Cryosectioning

Fixed samples were cryoprotected in 30% sucrose for 3 to 5 days. Samples were transferred into a 50% w/v mixture of 30% sucrose and TFM and incubated for 5 days in this medium before proceeding with sectioning. Fish were oriented in TFM using a molding cup tray, snap-frozen in liquid nitrogen, and then sectioned in 16 µm-thick sections at a Leica CMZ3050 S cryostat with temperature set to -25°C. Sections were gently applied to Superfrost plus microscope slides by touching them with the slide. Slides were left to dry at 4°C for at least one night.

5.2.26 Immunostaining of cryosections

Sections were rehydrated by bathing the slides in PTW for at least 30 min. To prevent evaporation and ensure equal solution distribution on the slides, a piece of parafilm was gently applied to the slides after applying solution at every step of the protocol.

If required, melanin pigment was bleached: First sections were post-fixed in 4% PFA/PTW for 20 min. Second, sections were bleached with bleaching solution for 90 min at room temperature. After bleaching was complete, the slides were rinsed several times in PTW.

To improve tissue accessibility, slides were pre-treated with 100% ice-cold acetone at -20°C for 10 min. Acetone was washed off by rinsing with PTW before blocking for at least 2 h with 10% NGS. Primary antibodies were diluted in 1% NGS and applied in the smallest possible volume on the slides. Slides were incubated at least overnight at 4°C. After washing slides at least 6 times for 5 min each in PTW, secondary antibodies and DAPI diluted in 1% NGS were applied onto the slides. Slides were incubated for at least 2 h at 37°C and then washed at least 3 times for 5 min each with PTW.

Afterwards, BrdU staining was performed: First, tissue was post-fixed in 4% PFA/PTW for 30 min at room temperature. After washing slides 3 times for 5 min each with PTW, an antigen retrieval step was done with 1.7 N HCl solution for 60 min at 37°C. After washing again 3 times for 5 min each with PTW, sample pH was recovered by applying 40% borax/PTW solution to the slides and incubating for 15 min at room temperature. After washing for 3 times for 5 min each with PTW, slides were blocked with 10% NGS for at least 2 h at room temperature.

Slides were again washed 2 times for 5 min each with PTW before applying primary antibody diluted in 1% NGS. Primary antibody-covered slides were incubated at least once overnight at 4°C. After washing slides 6 times for 5 min each with PTW, secondary antibody diluted in 1% NGS was applied onto the slides. Slides were incubated with secondary antibody at least once overnight at 4°C. The secondary antibody solution was washed off by rinsing slides several times with PTW.

Finally, 60 µl of 60% glycerol was applied to the slides before covering with a coverslip. To prevent evaporation, the edges of the coverslip were sealed with nail polish. Slides were stored at 4°C.

5.2.27 Microscopy

Fixed samples were mounted in 1% low-melt agarose in 1x ERM in a glass bottom dish and covered in PTW to prevent dehydration.

Imaging was performed at the following instruments with the listed objectives:

- Nikon AZ100 multizoom upright confocal laser scanning microscope; objectives:
  - "AZ Plan Fluor" 2x magnification; 0.2 numerical aperture (NA); 45 mm working distance (WD); dry objective (2x dry),
  - "AZ Plan Fluor" 5x magnification; 0.5 NA; 15 mm WD; dry objective (5x dry);
- Zeiss 710 inverted confocal laser scanning microscope; objectives:
  - "Plan-Apochromat" 20x magnification; 1.0 NA; 70 mm WD; DIC M27 water-immersion objective (20x water);
- Leica TCS SPE inverted confocal laser scanning microscope; objectives:
  - "ACS APO" 10x magnification; 0.30 NA; 11 mm WD; dry objective (10x dry),
  - "HC PL APO" 20x magnification; 0.70 NA; 0.62 mm WD; water-immersion objective (20x water),
  - "ACS APO" 40x magnification; 1.15 NA; 0.65 mm WD; oil-immersion objective (40x oil);
- Leica TCS SP5 inverted confocal laser scanning microscope; objectives:
  - HCX PL APO CS 40x magnification; 1.10NA; 0.65 mm WD; water-immersion objective (40x water);
- Leica TCS SP8 inverted confocal laser scanning microscope; objectives:
  - "HC PL FLUOTAR" 10x magnification; 0.30 NA; 11 mm WD; dry objective (10x dry),
  - "HC PL APO CS2" 20x magnification; 0.75 NA; 0.68 mm WD; glycerol-immersion objective (20x glycerol-immersion),
  - "HC PL APO CS2" 63x magnification; 1.30 NA; 0.30 mm WD; glycerol-

immersion objective (63x glycerol-immersion).

Acquisition parameters for images shown in figures of this work are listed in Appendix Table 5.54. Multi-tile images were stitched together using the microscope vendor's provided software functionality; if necessary, the ImageJ plugin "pairwise stitching" was used to correct errors in merging the tiles [Preibisch et al., 2009].

5.2.28 Time-lapse of zebrafish development

Male and female adult zebrafish were kept separate for one night in a small breeding tank and reunited in the morning. Eggs were collected and reared at 28°C until the desired stage, which was determined according to standard developmental tables [Kimmel et al., 1995]. Embryos were selected for fluorescent reporters, mechanically dechorionated, and mounted in 0.8% agarose in zebrafish medium which was previously cooled to 37°C. Embryos were first positioned with the head touching the coverslip dorsally, and were then carefully tilted to 45° such that on one side the nascent optic vesicle would abut the coverslip. The agarose was left to harden at room temperature before adding zebrafish medium supplemented with tricaine and PTU. Imaging was done at a Leica SP5 equipped with a heating chamber set to 28.5°C, ensuring normal developmental timing during imaging.

5.3 Image processing

All image processing was performed in ImageJ [Schindelin et al., 2012]. In the following, all ImageJ menu paths to specific commands are indicated in square brackets. Different strategies were used for background subtraction and feature enhancement depending on the image quality and acquisition parameters. The processing steps needed to reproduce all figures in this work are as listed in Appendix Table 5.55. More complex processing pipelines are explained in detail in the following sections. After processing, the brightness and contrast of all channels was adjusted individually using the built-in method in ImageJ [Image>Adjust>Brightness/Contrast...].

5.3.1 Focused stack projection

When focusing through a hemispherical object, the parts that are in focus are annuli of increasing size (and a circle or point in the very first focal plane). Exploiting this property, in-focus parts of confocal slices of wholemount RPE samples were extracted and collated to a focused projection by a custom-written ImageJ macro (Appendix Section 5.7.1). Based on the user-provided value for z-stepping, the macro calculated the radii of the hemispherical annuli in focus at each plane. This was done using the formula for the radius of the base of a spherical cap

$$r = \sqrt{hi(2R - hi)}, \tag{5.1}$$

where R is the radius of the hemispherical sample, h is the z-step used for image acquisition, and i is the ith step in the interval [0, nSteps] where nSteps is the total number of focal planes. After generation of the focused projection, features were enhanced by local contrast adjustment [Process > Enhance Local Contrast (CLAHE)].

5.3.2 Experimental clone segmentation

Clones in experimental data were segmented using the Phansalkar method as it is implemented in ImageJ [Phansalkar et al., 2011]. The segmentation was manually curated using the "Paintbrush Tool" in ImageJ; this was aided by overlaying the segmentation onto the original image using the ImageJ plugin "GDSC SMLM" at 50% opacity [Plugins>GDSC SMLM>Tools>Overlay Image] [Herbert, 2019].

5.3.3 Creation of retinal overlays

Maximum projection of confocal stacks were aligned manually with the following procedure:

Each image was projected using the option "Max Intensity" [Image>Stacks>Z Project...]. Images of left eyes acquired at Leica SPE and Leica SP8 microscopes were horizontally flipped to align anterior to the left [Image>Transform>Flip Horizontally]. One image was chosen as a reference and rotated with bicubic interpolation [Image>Transform>Rotate...] and if needed horizontally flipped to orient the retina with with ventral down. The reference image was overlayed to all other images using the ImageJ plugin "GDSC SMLM" at 50% opacity [Plugins>GDSC SMLM>Tools>Overlay Image] [Herbert, 2019]. The image underneath the overlay was manually translated [Image > Transform > Translate...], then rotated with bicubic interpolation [Image>Transform>Rotate...] until the overlap of anatomical features (optic nerve, ventral pole, post-induction BrdU ring) was optimal.

After alignment, the images were collated to a stack [Image > Stacks > Images to Stack], and maximum projected using the option "Max Intensity" [Image>Stacks>Z Project...]. The Lookup Table (LUT) was changed to "Fire" [Image > Lookup Tables > Fire].

5.3.4 Relative reduction of NR signal intensity in wholemount RPE samples

Due to its smaller nuclei and multiple layers of tightly packed cells, signal intensity for all tested markers was several-fold higher in the NR than in the RPE. This difference in intensity lead to out-of-focus bleedthrough of NR signal into the RPE during the confocal acquisition. I reasoned that convolution of a Gaussian kernel with such an image would result in a blurred image weighted towards the signal coming from the NR. In a subsequent step, this blurred image could be subtracted from the original image to specifically reduce NR signal (or, viceversa, to specifically enhance RPE signal). Thus, I devised the following image processing pipeline:

1. Create a duplicate of the original confocal stack.
   [Image>Duplicate...]
2. Convert the duplicate to 32-bit.
   [Image>Type>32-bit]
3. Blur the duplicate confocal stack using a 3-dimensional (3D) Gaussian blur filter; radii in x, y, z in pixels: $(10b, 10b, b)$.
   [Process>Filters>Gaussian Blur 3D...]
4. Divide the resulting blurred image by d.
   [Process>Math>Divide...]
5. Subtract the blurred image from the original image; check option "32-bit (float) result".
   [Process>Image Calculator...]

Parameters b and d were manually chosen depending on the image. The z-radius of the Gaussian kernel had to be set smaller due to the lower z-resolution of images acquired at confocal setups; a 10-fold difference with respect to the x- and y-radii produced best results. Although 32-bit images can store negative floating point numbers, some ImageJ operations functioned better with strictly positive values (e.g. background subtraction with rolling ball algorithm, thresholding algorithm). To maintain compatibility with further downstream image processing steps, I divided the blurred image by d. This step ensured that values in the blurred duplicate image were smaller than values at the corresponding position of the original image, such that subtraction resulted in values ≥ 0.

5.3.5 Removal of background staining for combined BrdU and EdU detection

The monoclonal BrdU antibody has a low degree of cross-reactivity with EdU [Liboska et al., 2012]. To correct for this cross-reactivity, I used a similar strategy as in section 5.3.4:

1. Create a duplicate of the EdU channel of the source image.
   [Image>Duplicate...]
2. Convert the duplicate to 32-bit.
   [Image>Type>32-bit]
3. Apply a 3D Gaussian blur filter; radii in x, y, z in pixels: (5, 5, 1).
   [Process>Filters>Gaussian Blur 3D...]
4. Divide the resulting blurred image by 2.
   [Process>Math>Divide...]
5. Subtract the blurred image from the BrdU channel of the original image; check option "32-bit (float) result".
   [Process>Image Calculator...]
6. Z project; projection type "Min Intensity".
   [Image>Stacks>Z Project...]
7. Obtain the minimum value of the projection from the histogram method.
   [Analyse>Histogram]
8. Add the minimum value to the unprojected image obtained in step 5.
   [Process>Math>Add...]
9. Subtract background with rolling ball algorithm; radius = 50 pixels.
   [Process>Subtract Background...]

5.3.6 Manual masking of areas of interest

To create a mask for a tissue of interest in a confocal stack, I proceeded with the following steps:

1. Create a new blank channel with the same dimensions as the original image [File>New>Image...]. This channel will be the mask.
2. Merge the mask channel with the original image to a multi-channel image [Image>colour>Merge Channels...].
3. Set the foreground colour to white in the "colour picker" tool (R=255, G=255, B=255). Use the "Paintbrush" tool to manually label all areas belonging to the tissue of interest at every focal plane in the mask channel.
4. Isolate the mask channel by splitting channels [Image>colour>Split Channels].

5. Divide the mask channel by 255 to turn it into a binary mask with values of 0 an 1 [Process>Math>Divide...].

5.3.7 Enhancement of PCNA signal

To enhance nuclear-localised PCNA signal the following steps were done:

1. Create a duplicate of the DAPI channel of the source image.
   [Image>Duplicate...]
2. Convert the duplicate to 32-bit.
   [Image>Type>32-bit]
3. Apply a 3D Gaussian blur filter; radii in x, y, z in pixels: (2, 2, 1).
   [Process>Filters>Gaussian Blur 3D...]
4. Z project; projection type "Max Intensity".
   [Image>Stacks>Z Project...]
5. Obtain the maximum value of the projection from the histogram method.
   [Analyse>Histogram]
6. Divide the image obtained in step 3 by the maximum value obtained in step 5 to normalise it to [0, 1].
   [Process>Math>Divide...]
7. Multiply the source image's PCNA channel with the image obtained in step 6; check option "32-bit (float) result".
   [Process>Image Calculator...]

5.3.8 Removal of high-intensity stromal cells

Despite careful dissection, remnants of overlying stromal tissue remained attached to wholemount samples. Due to its position at the outer surface of the sample, stromal tissue trapped antibodies and stained very intensively for some markers, *e.g.* BrdU. To remove unwanted stromal signal, I used a strategy similar to section 5.3.4 with an added thresholding step:

1. Create a duplicate of the source image.
   [Image>Duplicate...]
2. Threshold high-intensity cells by inspection of the stack histogram.
   [Image>Adjust>Threshold...]
3. Expand the thresholded image with a maximum filter; radii in x, y, z in pixels: (10, 10, 5).
   [Process>Filters>Maximum 3D...]
4. Convert the thresholded image to 32-bit.
   [Image>Type>32-bit]
5. Blur the image using a 3D Gaussian blur filter; radii in x, y, z in pixels: (3, 3, 1).
   [Process>Filters>Gaussian Blur 3D...]

6. Invert the pixel values of the image.
   [Edit>Invert]
7. Divide the resulting image by 255 to normalise it to [0, 1].
   [Process>Math>Divide...]
8. Multiply the resulting image with the source image; check option "32-bit (float) result".
   [Process>Image Calculator...]

5.4 Data analysis

5.4.1 Simulation renders

Simulation renders were automatically generated by EPISIM Simulator's visualisation screen; screenshots were taken using the built-in screenshot tool [Sütterlin et al., 2012].

5.4.2 Plotting simulated clonal data

Coordinates and clonal identity of all cells in a given simulation were exported as comma-separated value (csv) files using EPISIM Simulator's built-in data export functionality [Sütterlin, 2015]. The output file was parsed by a Python script to extract cell-individual properties such as coordinates and clonal colour (Appendix Section 5.7.2). The extracted data was used as input to the methods described in the following.

Plots of full simulated clonal complement

Extracted simulated data were plotted as a 3D scatterplot in Python (Appendix Source Code 5.26). Marker size was chosen to match the size of cells in the simulation, and marker colour was extracted from the clonal colour used in the simulation. The orientation of the plot was chosen to simulate the proximal view of a retina.

Selection of clones emerging from a cell originally in the ventral sector

Simulated data were parsed for the initial simulation step and the simulation step of interest. The cell coordinates for the initial simulation step were plugged into Equations 2.28 and 2.29 to obtain a list of clones whose originating cell was in the ventral sector. This list was used to filter out and plot clones in the simulation step of interest (Appendix Source Code 5.27).

Generation of simulated clone segmentation data

Extracted simulated data were used to generate a simulated segmentation in Python (Appendix Source Code 5.25). The method grouped cells by clonal families and plotted a 3D scatterplot oriented as a simulated proximal retinal view; scatterplot marker size was chosen such that the plot maintained correct cell proportions. A randomly sampled subset of clones was plotted using black markers, and the rest were plotted with white markers. The proportion of sampled clones was between 8-13%, which generated a comparable number of clones as in the experimental data. For subsequent data analysis, each simulation was sampled twice.

5.4.3 Extraction of patch properties from segmented data

Post-processing of segmented retinal images

Due to different image acquisition and pre-processing steps leading to the segmented images, the fine details of patch edges were not the same. Therefore, post-processing was done with different combinations of smoothing filters meant to reduce noise but preserve patch shape depending on the type of data: Data from the NR required no smoothing (Appendix Source Code 5.2), data from RPE were run through median and shape smoothing filters (Appendix Source Code 5.3), and simulated data were run through consequent binary erosion/dilation to reduce 1-cell clones, median and shape smoothing filters (Appendix Source Code 5.4). The latter two macros used the Shape Smoothing Plugin in ImageJ [Erdenetsogt and Wagner, 2019]. The post-processed images were automatically saved in a user-defined folder.

Subdivision of the retina into several ROIs

Starting from two manually created region of interests (ROIs) demarcating the retinal edge and the pre-induction retina (in this order), a custom ImageJ macro created a number of annular and angular subdivisions (Appendix 5.7.1). These compartments allowed to assess patch properties along each retinal quadrant as well as along the radius.

Radially normalised retinal projections

Dr Burkhard Höckendorf developed an analysis strategy that involved treating the projected retinal wholemount as a polar plot that could be converted to Cartesian coordinates [Höckendorf, 2013]. However, due to deformations caused by handling the tissue and endogenous retinal asymmetries along the dorso-ventral axis, the retinae were never perfectly circular and the projection deviated from the ideal rectangular form. To circumvent these issues I implemented a simple normalisation of the radial axis by effectively stretching the Cartesian projection such that at every x-coordinate the retinal tissue had an equal number of pixels along y (Appendix Section 5.7.1). In a subsequent step, all images were normalised to the same height by stretching [Image>Adjust>Size...]. Though this naive approach ignored differences in radial extent due to curvature, it worked fairly well as a first approximation to compare patch superposition.

Patch superposition

To extract individual patches from the radially normalised retinal projections, the image was first smoothed with the Shape Smoothing Plugin [Erdenetsogt and Wagner, 2019], then connected components were identified with the Find Connected Regions Plugin [Longair, 2019]. This plugin assigned a unique integer value to each contiguous patch of pixels. This property was exploited to loop through each patch via thresholding pixel values, which allowed to automatically extract and save a ROI outline of each patch. The ROIs were aligned along the x-coordinate using translation operations. Once aligned, a new image was created where the underlying pixel values were increased by 1 for each ROI that contained that pixel. Finally, the pixel intensity was normalised to the interval $[0,1]$, generating a patch superposition. This procedure was used to generate the images in Figure 2.19 panels A′-A″, and Figure 2.25 panels D′-D″″ and E′-E″″. The macro implementing these commands is in the Appendix (Section 5.7.1).

Patch size analysis

The properties of patches in radially normalised retinal projection were automatically extracted (Appendix Section 5.7.1) and saved to a csv file. This file was parsed in Python for extracting maximum patch width and length (Appendix Section 5.7.2). Additionally, the Python method removed patches shorter than 20% of the normalised radius from the width distribution (spot-like patches that would skew the data towards narrower patches), and patches whose uppermost y-coordinate was after 20% of the normalised radial extent for the height distribution (late arising patches that would skew the distribution towards shorter patches). The processed data were pasted into R and plotted as rugplots (Appendix Section 5.7.3). To compare the variance of widths, the maximum patch width distributions were tested using an F-test for equal variances with the R function `var.test`.

Extracting patch statistics

An ImageJ macro (Appendix Section 5.7.1) was used to extract patch statistics, including number of nodes and position of central-most pixel for plotting late arising patches. This macro proceeded as follows:

1. Subdivided the retina into multiple ROIs, analogously to Section 5.4.3, thus generating radial bins.
2. Projected the retina to Cartesian coordinates using the plugin "Polar Transformer" [Donnelly and Mothe, 2013].
3. Corrected for any patches that were split during the transform.
4. Found connected regions with the plugin "Find Connected Regions" [Longair, 2019].
5. Isolated each patch at a time by thresholding.
6. Generated a skeleton of each patch by breaking it up into radial bins, finding the average x-coordinate, and then linking all the skeleton sections together.
7. Transformed each skeletonised patch back into polar coordinates.
8. Used the plugin "Skeleton Analyzer" to obtain nodes in the skeletons [Arganda-Carreras et al., 2010].
9. Collated all the skeletons into one image for display.
10. Measured other patch statistics such as position in the retina with respect to the bins generated in step 1.

The macro also used the Shape Smoothing Plugin during some processing steps [Erdenetsogt and Wagner, 2019].

The resulting data were imported into R via RStudio's manual "Import Dataset" functionality to generate violin plots of node counts (Appendix Section 5.7.3) and rugplots of late arising patches (Appendix Section 5.7.3) . Late arising patches were defined as all patches that did not contain pixels in the first radial bin. Statistical tests were performed using the Wilcoxon rank sum test with the R function `wilcox.test`.

5.4.4 Data plotting

Plot of average cell displacement against calculation time

Average cell displacement and calculation time were printed out to console in EPISIM Simulator. The values were plotted using a Python script (Appendix Section 5.7.2) to generate the plot in Figure 2.4 panel C.

Plot of fish body and eye size measurements

Photos of fish were acquired as described in Section 5.2.17 and analysed in ImageJ. Body length was measured with a straight line from the anterior-most point on the lower jaw to the posterior-most point of the central tail fin; eye diameter was measured by manually fitting an ellipse to the eye and taking the average of major and minor axes. These data were copied into a Python script (Appendix Section 5.7.2) to generate the plot in Figure 2.6 panel A.

Plot of cell cycle distribution times

The theoretical distributions in Figure 2.6 panel B were plotted using R (Appendix Section 5.7.3). For the histograms in Figure 2.6 panel C, simulation data were output by EPISIM Simulator's built-in data export functionality, preprocessed, and then plotted using a Python script (Appendix Section 5.7.2).

3D plot of cell age

Data extracted from the simulation were plotted as a 3D scatterplot using a custom Python script (Appendix Section 5.7.2) and used in Figures 2.7 panel C, and 2.8 B″ and C rightmost image.

Heatmap of average cell overlap against radial position on hemisphere

Data in Figure 2.12 panels A′-B″″ were extracted from the simulation and plotted as a heatmap using a custom Python script (Appendix Section 5.7.2).

Diagnostic plots of area density and eye radius over time

Diagnostic plots of relative cell density and eye radius growth in Figure 2.12 panels C′-D″ were obtained by plotting extracted data from the simulation using custom Python scripts (Appendix Section 5.7.2).

Parameter scan of p_{div} and $t_{\text{cellCycle}}$

Several simulation runs were done with varying parameter values for 200 simulation steps each. The average eye radius growth rate was calculated by assuming a linear growth rate and calculating the slope between simulation step

200 and simulation step 100. These values were manually entered into a custom Python script that interpolated, smoothed, and plotted a contour plot of the data (Appendix Section 5.7.2).

2D histogram of cell division intervals against normalised average overlap

Data in Figure 2.16 were extracted from the simulation and plotted as a 2D histogram using a custom Python script (Appendix Section 5.7.2).

Proportion of terminated clones and ArCoS

Data extracted from the simulation were used to sort all clones at a given simulation step according to the following criteria: Clones that retained at least one cell in the CMZ at the simulation step analysed were considered "persistent" or ArCoS and all clones that failed to retain all of their cells within the CMZ were considered "terminated". Persistent and terminated clones were sorted into 5 bins along the extent of the virtual CMZ. These data were obtained from a Python script (Appendix Section 5.7.2), and were manually input into R for plotting as a stacked bar plot (Appendix Section 5.7.3). These data are shown in Figure 2.20 panel B.

To compare experimental and simulated data directly, a different strategy was used. The inner and outer limits of the induction ring in maximum-projected experimental data (not segmented) and simulated segmentation (Section 5.4.2) were manually fitted by two ellipses using the "Oval Tool" in ImageJ. The criteria for choosing the ellipse limits were: To encircle as many single-cell clones as possible by the inner ellipse, to cross as many ArCoS as possible with the outer ellipse. A third ellipse bisected these two circles using an ImageJ macro (Appendix Section 5.7.1), generating a "central" and a "peripheral" induction ring. Contiguous patches that contained at least one pixel in the central induction ring were considered as emerging centrally. Patches were then classified based on whether they reached the outer retinal margin as persistent/ArCoS or if they failed to do so as terminated. Data were input manually in R and plotted as a stacked bar plot (Appendix Section 5.7.3). Statistical test of equality of proportions was performed in R using the function `prop.test`. These data are shown in Figure 2.20 panel F.

Clone angular width

To obtain clone angular width, experimental and simulated samples were projected to Cartesian coordinates using the Polar Transformer plugin in

ImageJ [Donnelly and Mothe, 2013]. Experimental patches were assigned unique integer values using the Find Connected Regions Plugin [Longair, 2019]. Simulated patches were generated with a unique colour using the option `unique_colour=True` in the script used to generate simulated segmentation (Appendix Source Code 5.25). The ArCoS Analysis plugin was then used to measure the circumferential extent of each patch at every radial position [Höckendorf, 2013]. For simulated data, the analysis was automated by a macro (Appendix Section 5.7.1). The resulting file was saved to a csv.

The corresponding files of different experimental or simulated datasets were imported into R (Appendix Section 5.7.3), where they were merged together, and pre-processed. The pre-processing consisted of removing very small patches (less than 10% of retinal radius spanned). Since not all patches were present at every radial position, the data formed a large sparse matrix. Therefore, custom functions for calculating mean, standard deviation, and standard error were implemented. To exclude the induction ring from the final plot, the extent of the zone of stable lineages was estimated by inspection of the original images. Welch modified two-sample t-test was calculated in R using the mean, standard deviation, and the number of patches with the function `tsum.test` from the package `BSDA`. The built-in function `t.test` could not be used due to the unconventional structure of the data.

Evaluation of retinal asymmetries

Retinal dimensions in samples treated with multiple BrdU pulses were measured by manually drawing lines in ImageJ. The lines were drawn from the dorsal, ventral, anterior, and posterior poles of each BrdU incorporation ring to the respective poles on the retinal margin. Mean and standard deviation were calculated with built-in function in R. Plots in Figure 2.23 A″ and A‴ were made in Inkscape by scaling vector graphic elements to the proportions obtained by measurement. Welch two-sample t-test was done in R using the built-in function `t.test`.

Measurement of inter-nucleus distance

Inter-nucleus distance was measured in ImageJ by drawing a line from the centroid of a nucleus to the centroid of a neighboring cell's nucleus. The area for measurement was chosen to be a flattened piece of tissue to avoid biases introduced by curvature. The data were copied into R to plot the violin plots in Figure 2.28 panel B and Figure 2.32 B‴ (Appendix Source Code 5.7.3).

Counting of cell rows and measurement of CMZ length

Counts of cell rows in the RPE and NVR were done manually by inspection of the original confocal stacks. Orthogonal projections were used to verify the counts. Measurement of RPE CMZ dimensions was done in ImageJ on projected images. The curvature of the tissue was not taken into account. Numbers shown in Figure 2.32 panel B″ and Figure 2.36 panels B′-B‴ are averages of 1 to 3 samples per condition.

NVR growth kinetics

The data used for the plot in Figure 2.36 panel A‴ were acquired by Mai Thu Nguyen [Nguyen, 2018]. The data were parsed in a Python script, plotted as a scatterplot, and a piecewise linear function was fit to three intervals. The slope of the fit was noted.

Measurement of relative eye dimensions

Eye anterior-posterior and dorso-ventral diameters, and fish body length were measured in ImageJ by drawing a line. The data were copied into R to plot the violin plots in Figure 5.9 panel D using the script described in the Appendix Source Code 5.36.

5.5 Figure composition

All figures presented in this work were composed in Inkscape [Inkscape 0.92, 2017]. Images and plots obtained from external programs were embedded into the Inkscape canvas, and were cropped and resized if necessary. Graphical elements of the images (axis legends, line colours) were adapted to improve aesthetics. Images were not modified in such a way that the content would be falsified or in any way misrepresented.

References

Abar, S., Theodoropoulos, G. K., Lemarinier, P., and O'Hare, G. M. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33.

Ahmad, I., Tang, L., and Pham, H. (2000). Identification of neural progenitors in the adult mammalian eye. *Biochemical and biophysical research communications*, 270(2):517–521.

Aichinger, E., Kornet, N., Friedrich, T., and Laux, T. (2012). Plant stem cell niches. *Annual review of plant biology*, 63:615–636.

Al-Hussaini, H., Kilarkaje, N., Shahabi, G., and Al-Mulla, F. (2016). Proliferation and migration of peripheral retinal pigment epithelial cells are associated with the upregulation of wingless-related integration and bone morphogenetic protein signaling in dark agouti rats. *Medical Principles and Practice*, 25(5):408–416.

Alber, M. S., Kiskowski, M. A., Glazier, J. A., and Jiang, Y. (2003). On cellular automaton approaches to modeling biological cells. In *Mathematical systems theory in biology, communications, computation, and finance*, pages 1–39. Springer.

Amato, M. A., Arnault, E., and Perron, M. (2004). Retinal stem cells in vertebrates: parallels and divergences. *International Journal of Developmental Biology*, 48(8-9):993–1001.

Amini, R., Rocha-Martins, M., and Norden, C. (2018). Neuronal migration and lamination in the vertebrate retina. *Frontiers in neuroscience*, 11:742.

Analytik Jena (2012). https://assets.thermofisher.com/TFS-Assets/LSG/manuals/cms_055516.pdf.

Andrews, T. S. and Hemberg, M. (2018). Identifying cell populations with scrnaseq. *Molecular aspects of medicine*, 59:114–122.

Arganda-Carreras, I., Fernández-González, R., Muñoz-Barrutia, A., and Ortiz-De-Solorzano, C. (2010). 3d reconstruction of histological sections: Application to mammary gland tissue. *Microscopy research and technique*, 73(11):1019–1029.

Arora, M., Moser, J., Phadke, H., Basha, A. A., and Spencer, S. L. (2017). Endogenous replication stress in mother cells leads to quiescence of daughter cells. *Cell reports*, 19(7):1351–1364.

Baena-López, L. A., Baonza, A., and García-Bellido, A. (2005). The orientation of cell divisions determines the shape of drosophila organs. *Current Biology*, 15(18):1640–1644.

Ballios, B. G., Clarke, L., Coles, B. L., Shoichet, M. S., and Van Der Kooy, D. (2012). The adult retinal stem cell is a rare cell in the ciliary epithelium whose progeny can differentiate into photoreceptors. *Biology open*, 1(3):237–246.

Bartocci, E. and Lió, P. (2016). Computational modeling, formal analysis, and tools for systems biology. *PLoS computational biology*, 12(1):e1004591.

Bazin-Lopez, N., Valdivia, L. E., Wilson, S. W., and Gestri, G. (2015). Watching eyes take shape. *Current opinion in genetics & development*, 32:73–79.

Beck, J. C., Gilland, E., Tank, D. W., and Baker, R. (2004). Quantifying the ontogeny of optokinetic and vestibuloocular behaviors in zebrafish, medaka, and goldfish. *Journal of neurophysiology*, 92(6):3546–3561.

Bélanger, M.-C., Robert, B., and Cayouette, M. (2017). Msx1-positive progenitors in the retinal ciliary margin give rise to both neural and non-neural progenies in mammals. *Developmental cell*, 40(2):137–150.

Bellaïche, Y. (2016). Cell division in the light of modeling. *Developmental cell*, 38(6):584–586.

Bengtsson, H., Corrada-Bravo, H., Gentleman, R., and Jaffee, H. (2014). matrixstats: Methods that apply to rows and columns of a matrix. *R package version 0.8*, 14.

Bishop, P., Takanosu, M., Le Goff, M., and Mayne, R. (2002). The role of the posterior ciliary body in the biosynthesis of vitreous humour. *Eye*, 16(4):454.

Bodenstein, L. (1986). A dynamic simulation model of tissue growth and cell patterning. *Cell Differentiation*, 19(1):19–33.

Bodenstein, L. and Sidman, R. L. (1987a). Growth and development of the mouse retinal pigment epithelium: I. cell and tissue morphometrics and topography of mitotic activity. *Developmental biology*, 121(1):192–204.

Bodenstein, L. and Sidman, R. L. (1987b). Growth and development of the mouse retinal pigment epithelium: Ii. cell patterning in experimental chimaeras and mosaics. *Developmental biology*, 121(1):205–219.

Boone, E., Colombani, J., Andersen, D. S., and Léopold, P. (2016). The hippo signalling pathway coordinates organ growth and limits developmental variability by controlling dilp8 expression. *Nature communications*, 7:13505.

Buchmann, A., Alber, M., and Zartman, J. J. (2014). Sizing it up: the mechanical feedback hypothesis of organ growth regulation. In *Seminars in cell & developmental biology*, volume 35, pages 73–81. Elsevier.

Buske, P., Galle, J., Barker, N., Aust, G., Clevers, H., and Loeffler, M. (2011). A comprehensive model of the spatio-temporal stem cell and tissue organisation in the intestinal crypt. *PLoS computational biology*, 7(1):e1001045.

Byrne, H. and Drasdo, D. (2009). Individual-based and continuum models of growing cell populations: a comparison. *Journal of mathematical biology*, 58(4-5):657.

Cadart, C., Monnier, S., Grilli, J., Sáez, P. J., Srivastava, N., Attia, R., Terriac, E., Baum, B., Cosentino-Lagomarsino, M., and Piel, M. (2018). Size control in mammalian cells involves modulation of both growth rate and cell cycle duration. *Nature communications*, 9(1):3275.

Cameron, D. A. (1995). Asymmetric retinal growth in the adult teleost green sunfish (lepomis cyanellus). *Visual neuroscience*, 12(1):95–102.

Cameron, D. A. (1996). Asymmetric retinal growth: evidence for regulation by a retinotopic mechanism. *Visual neuroscience*, 13(3):493–500.

Cameron, D. A. and Carney, L. H. (2004). Cellular patterns in the inner retina of adult zebrafish: quantitative analyses and a computational model of their formation. *Journal of Comparative Neurology*, 471(1):11–25.

Caves, E. M., Brandley, N. C., and Johnsen, S. (2018). Visual acuity and the evolution of signals. *Trends in ecology & evolution*, 33(5):358–372.

Caves, E. M., Sutton, T. T., and Johnsen, S. (2017). Visual acuity in ray-finned fishes correlates with eye size and habitat. *Journal of Experimental Biology*, 220(9):1586–1596.

Cavodeassi, F. (2018). Dynamic tissue rearrangements during vertebrate eye morphogenesis: insights from fish models. *Journal of developmental biology*, 6(1):4.

Centanin, L., Ander, J.-J., Hoeckendorf, B., Lust, K., Kellner, T., Kraemer, I., Urbany, C., Hasel, E., Harris, W. A., Simons, B. D., et al. (2014). Exclusive multipotency and preferential asymmetric divisions in post-embryonic neural stem cells of the fish retina. *Development*, 141(18):3472–3482.

Centanin, L., Hoeckendorf, B., and Wittbrodt, J. (2011). Fate restriction and multipotency in retinal stem cells. *Cell stem cell*, 9(6):553–562.

Cerveny, K. L., Varga, M., and Wilson, S. W. (2012). Continued growth and circuit building in the anamniote visual system. *Developmental neurobiology*, 72(3):328–345.

Chao, H. X., Fakhreddin, R. I., Shimerov, H. K., Kedziora, K. M., Kumar, R. J., Perez, J., Limas, J. C., Grant, G. D., Cook, J. G., Gupta, G. P., et al. (2019). Evidence that the human cell cycle is a series of uncoupled, memoryless phases. *Molecular systems biology*, 15(3):e8604.

Christ, A., Christa, A., Kur, E., Lioubinski, O., Bachmann, S., Willnow, T. E., and Hammes, A. (2012). Lrp2 is an auxiliary shh receptor required to condition the forebrain ventral midline for inductive signals. *Developmental cell*, 22(2):268–278.

Clevers, H. and Watt, F. M. (2018). Defining adult stem cells by function, not by phenotype. *Annual review of biochemistry*, 87:1015–1027.

Collery, R. F. and Link, B. A. (2011). Dynamic smad-mediated bmp signaling revealed through transgenic zebrafish. *Developmental Dynamics*, 240(3):712–722.

Collery, R. F. and Link, B. A. (2018). Proteolytic processing of lrp2 on rpe cells regulates bmp activity to control eye size and refractive error. *bioRxiv*, page 365817.

Colom, B. and Jones, P. H. (2016). Clonal analysis of stem cells in differentiation and disease. *Current opinion in cell biology*, 43:14–21.

Colombani, J., Andersen, D. S., Boulan, L., Boone, E., Romero, N., Virolle, V., Texada, M., and Léopold, P. (2015). Drosophila lgr3 couples organ growth with maturation and ensures developmental stability. *Current biology*, 25(20):2723–2729.

Conlon, I. and Raff, M. (1999). Size control in animal development. *Cell*, 96(2):235–244.

Conway, K., Feiock, K., and Hunt, R. K. (1980). Polyclones and patterns in growing xenopus eye. In *Current topics in developmental biology*, volume 15, pages 217–317. Elsevier.

Das, A. V., James, J., Rahnenführer, J., Thoreson, W. B., Bhattacharya, S., Zhao, X., and Ahmad, I. (2005). Retinal properties and potential of the adult mammalian ciliary epithelium stem cells. *Vision research*, 45(13):1653–1666.

Das, A. V., Zhao, X., James, J., Kim, M., Cowan, K. H., and Ahmad, I. (2006). Neural stem cells in the adult ciliary epithelium express gfap and are regulated by wnt signaling. *Biochemical and biophysical research communications*, 339(2):708–716.

Das, G., Choi, Y., Sicinski, P., and Levine, E. M. (2009). Cyclin d1 fine-tunes the neurogenic output of embryonic retinal progenitor cells. *Neural development*, 4(1):15.

Del Debbio, C. B., Peng, X., Xiong, H., and Ahmad, I. (2013). Adult ciliary epithelial stem cells generate functional neurons and differentiate into both early and late born retinal neurons under non-cell autonomous influences. *BMC neuroscience*, 14(1):130.

Donati, G. and Watt, F. M. (2015). Stem cell heterogeneity and plasticity in epithelia. *Cell stem cell*, 16(5):465–476.

Donnelly, E. and Mothe, F. (2013). Polar transformer plugin for imagej. https://imagej.nih.gov/ij/plugins/polar-transformer.html.

Drasdo, D. (2005). Coarse graining in simulated cell populations. *Advances in Complex Systems*, 8(02n03):319–363.

Droujinine, I. A. and Perrimon, N. (2016). Interorgan communication pathways in physiology: focus on drosophila. *Annual review of genetics*, 50:539–570.

Dunn, S.-J., Näthke, I. S., and Osborne, J. M. (2013). Computational models reveal a passive mechanism for cell migration in the crypt. *PLoS One*, 8(11):e80516.

Easter, S. (1992). Retinal growth in foveated teleosts: nasotemporal asymmetry keeps the fovea in temporal retina. *Journal of Neuroscience*, 12(6):2381–2392.

Easter Jr, S. S., Johns, P. R., and Baumann, L. (1977). Growth of the adult goldfish eye—i: Optics. *Vision research*, 17(3):469–477.

Eckert, P., Knickmeyer, M. D., Schütz, L., Wittbrodt, J., and Heermann, S. (2019). Morphogenesis and axis specification occur in parallel during optic cup and optic fissure formation, differentially modulated by bmp and wnt. *Open biology*, 9(2):180179.

Eclipse IDE Kepler 4.3.1 (2013). `https://www.eclipse.org/`.

Erdenetsogt, U. and Wagner, T. (2019). Shape smoothing plugin. `https://github.com/thorstenwagner/ij-shape-smoothing`.

Fernald, R. D. (1990). *The Visual System of Fish*, chapter The optical system of fishes, pages 41–61. Springer Netherlands.

Fernald, R. D. (1991). Teleost vision: seeing while growing. *Journal of Experimental Zoology*, 256(S5):167–180.

Fernald, R. D. (2000). Evolution of eyes. *Current opinion in neurobiology*, 10(4):444–450.

Fischer, A. J., Bosse, J. L., and El-Hodiri, H. M. (2014). Reprint of: The ciliary marginal zone (cmz) in development and regeneration of the vertebrate eye. *Experimental eye research*, 123:115–120.

Fischer, A. J., Ritchey, E. R., Scott, M. A., and Wynne, A. (2008). Bullwhip neurons in the retina regulate the size and shape of the eye. *Developmental biology*, 317(1):196–212.

Fletcher, A. G., Osterfield, M., Baker, R. E., and Shvartsman, S. Y. (2014). Vertex models of epithelial morphogenesis. *Biophysical journal*, 106(11):2291–2304.

Freddo, T. F. (2013). A contemporary concept of the blood–aqueous barrier. *Progress in retinal and eye research*, 32:181–195.

Fuhrmann, S., Zou, C., and Levine, E. M. (2014). Retinal pigment epithelium development, plasticity, and tissue homeostasis. *Experimental eye research*, 123:141–150.

Fukamachi, S., Asakawa, S., Wakamatsu, Y., Shimizu, N., Mitani, H., and Shima, A. (2004). Conserved function of medaka pink-eyed dilution in melanin synthesis and its divergent transcriptional regulation in gonads among vertebrates. *Genetics*, 168(3):1519–1527.

Garelli, A., Gontijo, A. M., Miguela, V., Caparros, E., and Dominguez, M. (2012). Imaginal discs secrete insulin-like peptide 8 to mediate plasticity of growth and maturation. *Science*, 336(6081):579–582.

Garrick, D., Fiering, S., Martin, D. I., and Whitelaw, E. (1998). Repeat-induced gene silencing in mammals. *Nature genetics*, 18(1):56.

Gestri, G., Bazin-Lopez, N., Scholes, C., and Wilson, S. W. (2018). Cell behaviors during closure of the choroid fissure in the developing eye. *Frontiers in cellular neuroscience*, 12:42.

Gokhale, R. H., Hayashi, T., Mirque, C. D., and Shingleton, A. W. (2016). Intra-organ growth coordination in drosophila is mediated by systemic ecdysone signaling. *Developmental biology*, 418(1):135–145.

Gorochowski, T. E. (2016). Agent-based modelling in synthetic biology. *Essays in biochemistry*, 60(4):325–336.

Gould, D. B., Smith, R. S., and John, S. W. (2004). Anterior segment development relevant to glaucoma. *International Journal of Developmental Biology*, 48(8-9):1015–1029.

Grabher, C., Joly, J.-S., and Wittbrodt, J. (2004). Highly efficient zebrafish transgenesis mediated by the meganuclease i-scei. In *Methods in cell biology*, volume 77, pages 381–401. Elsevier.

Graner, F. and Glazier, J. A. (1992). Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical review letters*, 69(13):2013.

Gray, H. (1918). *Anatomy of the human body.*

Gutierrez-Triana, J. A., Tavhelidse, T., Thumberger, T., Thomas, I., Wittbrodt, B., Kellner, T., Anlas, K., Tsingos, E., and Wittbrodt, J. (2018). Efficient single-copy hdr by 5'modified long dsdna donors. *eLife*, 7:e39468.

Hallatschek, O. and Nelson, D. R. (2010). Life at the front of an expanding population. *Evolution: International Journal of Organic Evolution*, 64(1):193–206.

Hanovice, N. J., Leach, L. L., Slater, K., Gabriel, A. E., Romanovicz, D., Shao, E., Collery, R., Burton, E. A., Lathrop, K. L., Link, B. A., et al. (2019). Regeneration of the zebrafish retinal pigment epithelium after widespread genetic ablation. *PLoS genetics*, 15(1):e1007939.

Harris, W. A. and Perron, M. (1998). Molecular recapitulation: the growth of the vertebrate retina. *International Journal of Developmental Biology*, 42(3):299–304.

Haruta, M., Kosaka, M., Kanegae, Y., Saito, I., Inoue, T., Kageyama, R., Nishida, A., Honda, Y., and Takahashi, M. (2001). Induction of photoreceptor-specific phenotypes in adult mammalian iris tissue. *Nature neuroscience*, 4(12):1163.

He, J., Zhang, G., Almeida, A. D., Cayouette, M., Simons, B. D., and Harris, W. A. (2012). How variable clones build an invariant retina. *Neuron*, 75(5):786–798.

Heermann, S., Schütz, L., Lemke, S., Krieglstein, K., and Wittbrodt, J. (2015). Eye morphogenesis driven by epithelial flow into the optic cup facilitated by modulation of bone morphogenetic protein. *Elife*, 4:e05216.

Herbert, A. (2019). Gdsc single molecule light microscopy (smlm) imagej plugins. https://github.com/aherbert/GDSC-SMLM.

Hietakangas, V. and Cohen, S. M. (2009). Regulation of tissue growth through nutrient sensing. *Annual review of genetics*, 43:389–410.

Hirose, E. and Matsumoto, J. (1994). Differentiation of pigmented tissues in the eyes of medaka (oryzias latipes) embryos: An electron microscope study. *The fish biology journal Medaka*, 6:25–31.

Höckendorf, B. (2013). *A Quantitative Clonal Analysis of Adult Neural Stemcells During the Post-embryonic Growth of the Medaka Retina*. PhD thesis, Ruprecht-Karls Universität Heidelberg.

Holt, C. E., Bertsch, T. W., Ellis, H. M., and Harris, W. A. (1988). Cellular determination in the xenopus retina is independent of lineage and birth date. *Neuron*, 1(1):15–26.

Hu, W., Haamedi, N., Lee, J., Kinoshita, T., and Ohnuma, S.-i. (2013). The structure and development of xenopus laevis cornea. *Experimental eye research*, 116:109–128.

Hunt, R. K., Bodenstein, L., Cohen, J. S., and Sidman, R. L. (1988). Positional variations in germinal cell growth in pigment-chimeric eyes of xenopus: posterior half of the developing eye studied in genetic chimerae and in computer simulations. *Proceedings of the National Academy of Sciences*, 85(10):3459–3463.

Hunt, R. K., Cohen, J. S., and Mason, B. J. (1987a). Cell patterning in pigment-chimeric eyes in xenopus: germinal transplants and their contributions to growth of the pigmented retinal epithelium. *Proceedings of the National Academy of Sciences*, 84(10):3302–3306.

Hunt, R. K., Cohen, J. S., and Mason, B. J. (1987b). Cell patterning in pigment-chimeric eyes of xenopus: local cues control the decision to become germinal cells. *Proceedings of the National Academy of Sciences*, 84(15):5292–5296.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90.

Iida, A., Shimada, A., Shima, A., Takamatsu, N., Hori, H., Takeuchi, K., and Koga, A. (2006). Targeted reduction of the dna methylation level with 5-azacytidine promotes excision of the medaka fish tol2 transposable element. *Genetics Research*, 87(3):187–193.

Inkscape 0.92 (2017). https://inkscape.org/.

Invitrogen (2010). https://assets.thermofisher.com/TFS-Assets/LSG/manuals/fm_1354.pdf.

Ivanovitch, K., Cavodeassi, F., and Wilson, S. W. (2013). Precocious acquisition of neuroepithelial character in the eye field underlies the onset of eye morphogenesis. *Developmental cell*, 27(3):293–305.

Iwamatsu, T. (2004). Stages of normal development in the medaka oryzias latipes. *Mechanisms of development*, 121(7-8):605–618.

Johns, P. R. (1977). Growth of the adult goldfish eye. iii. source of the new retinal cells. *Journal of Comparative Neurology*, 176(3):343–357.

Johns, P. R. (1981). Growth of fish retinas. *American Zoologist*, 21(2):447–458.

Johns, P. R. and Easter Jr, S. S. (1977). Growth of the adult goldfish eye. ii. increase in retinal cell number. *Journal of Comparative Neurology*, 176(3):331–341.

Jones, E., Oliphant, T., and Peterson, P. (2001). Scipy: Open source scientific tools for python. `http://www.scipy.org`.

Jones, G. W. and Chapman, S. J. (2012). Modeling growth in biological materials. *Siam review*, 54(1):52–118.

Karkach, A. S. (2006). Trajectories and models of individual growth. *Demographic Research*, 15:347–400.

Kearse, M., Moir, R., Wilson, A., Stones-Havas, S., Cheung, M., Sturrock, S., Buxton, S., Cooper, A., Markowitz, S., Duran, C., et al. (2012). Geneious basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics*, 28(12):1647–1649.

Kimmel, C. B., Ballard, W. W., Kimmel, S. R., Ullmann, B., and Schilling, T. F. (1995). Stages of embryonic development of the zebrafish. *Developmental dynamics*, 203(3):253–310.

Kirchmaier, S., Lust, K., and Wittbrodt, J. (2013). Golden gateway cloning–a combinatorial approach to generate fusion and recombination constructs. *PloS one*, 8(10):e76117.

Kitano, H. (2002). Systems biology: a brief overview. *science*, 295(5560):1662–1664.

Klein, A. M., Doupé, D. P., Jones, P. H., and Simons, B. D. (2007). Kinetics of cell division in epidermal maintenance. *Physical Review E*, 76(2):021910.

Kokkinopoulos, I., Shahabi, G., Colman, A., and Jeffery, G. (2011). Mature peripheral rpe cells have an intrinsic capacity to proliferate; a potential regulatory mechanism for age-related cell loss. *PloS one*, 6(4):e18921.

Korolev, K. S., Müller, M. J., Karahan, N., Murray, A. W., Hallatschek, O., and Nelson, D. R. (2012). Selective sweeps in growing microbial colonies. *Physical biology*, 9(2):026008.

Kretzschmar, K. and Watt, F. M. (2012). Lineage tracing. *Cell*, 148(1-2):33–45.

Kubota, R., Hokoc, J., Moshiri, A., McGuire, C., and Reh, T. (2002). A comparative study of neurogenesis in the retinal ciliary marginal zone of homeothermic vertebrates. *Developmental brain research*, 134(1-2):31–41.

Kuchen, E. E., Becker, N., Claudino, N., and Hofer, T. (2018). Long-range memory of growth and cycle progression correlates cell cycles in lineage trees. *bioRxiv*, page 373258.

Kwan, K. M., Otsuna, H., Kidokoro, H., Carney, K. R., Saijoh, Y., and Chien, C.-B. (2012). A complex choreography of cell movements shapes the vertebrate eye. *Development*, 139(2):359–372.

Lamb, T. D., Collin, S. P., and Pugh Jr, E. N. (2007). Evolution of the vertebrate eye: opsins, photoreceptors, retina and eye cup. *Nature Reviews Neuroscience*, 8(12):960.

Li, L. and Bhatia, R. (2011). Stem cell quiescence. *Clinical cancer research*, 17(15):4936–4941.

Li, L. and Clevers, H. (2010). Coexistence of quiescent and active adult stem cells in mammals. *science*, 327(5965):542–545.

Liboska, R., Ligasová, A., Strunin, D., Rosenberg, I., and Koberna, K. (2012). Most anti-brdu antibodies react with 2'-deoxy-5-ethynyluridine—the method for the effective suppression of this cross-reactivity. *PloS one*, 7(12):e51679.

Life Technologies (2012). https://assets.thermofisher.com/TFS-Assets/LSG/manuals/cms_055516.pdf.

Lischik, C. (2019). *Combining in vivo imaging and mechanistic approaches to investigate Wnt regulation of retinal stem cells*. PhD thesis, Ruprecht-Karls Universität Heidelberg.

Lischik, C. Q., Adelmann, L., and Wittbrodt, J. (2019). Enhanced in vivo-imaging in medaka by optimized anaesthesia, fluorescent protein selection and removal of pigmentation. *PloS one*, 14(3):e0212956.

Locket, N. (1977). Adaptations to the deep-sea environment. In *The visual system in vertebrates*, pages 67–192. Springer.

Longair, M. (2019). Find connected regions plugin. https://www.longair.net/edinburgh/imagej/find-connected-regions/.

Loosli, F., Köster, R. W., Carl, M., Kühnlein, R., Henrich, T., Mücke, M., Krone, A., and Wittbrodt, J. (2000). A genetic screen for mutations affecting embryonic development in medaka fish (oryzias latipes). *Mechanisms of development*, 97(1-2):133–139.

Lui, J. C. and Baron, J. (2011). Mechanisms limiting body growth in mammals. *Endocrine reviews*, 32(3):422–440.

Lust, K. and Wittbrodt, J. (2018). Activating the regenerative potential of müller glia cells in a regeneration-deficient retina. *Elife*, 7:e32319.

Lyall, A. (1957). The growth of the trout reina. *Journal of Cell Science*, 3(41):101–110.

MacIver, M. A., Schmitz, L., Mugan, U., Murphey, T. D., and Mobley, C. D. (2017). Massive increase in visual range preceded the origin of terrestrial vertebrates. *Proceedings of the National Academy of Sciences*, 114(12):E2375–E2384.

Maier, S. (2006). *Analyse der Expression und Funktion von SMOC-1 und SMOC-2*. PhD thesis, Universität zu Köln.

Marcucci, F., Murcia-Belmonte, V., Wang, Q., Coca, Y., Ferreiro-Galve, S., Kuwajima, T., Khalid, S., Ross, M. E., Mason, C., and Herrera, E. (2016). The ciliary margin zone of the mammalian retina generates retinal ganglion cells. *Cell reports*, 17(12):3153–3164.

Merks, R. M. and Glazier, J. A. (2005). A cell-centered approach to developmental biology. *Physica A: Statistical Mechanics and its Applications*, 352(1):113–130.

Metzcar, J., Wang, Y., Heiland, R., and Macklin, P. (2019). A review of cell-based computational modeling in cancer biology. *JCO clinical cancer informatics*, 2:1–13.

Meyer, D. B. (1977). The avian eye and its adaptations. In *The visual system in vertebrates*, pages 549–611. Springer.

Mochizuki, A. (2002). Pattern formation of the cone mosaic in the zebrafish retina: A cell rearrangement model. *Journal of theoretical biology*, 215(3):345–361.

Morrison, S. J. and Spradling, A. C. (2008). Stem cells and niches: mechanisms that promote stem cell maintenance throughout life. *Cell*, 132(4):598–611.

Nestor-Bergmann, A., Goddard, G., and Woolner, S. (2014). Force and the spindle: mechanical cues in mitotic spindle orientation. In *Seminars in cell & developmental biology*, volume 34, pages 133–139. Elsevier.

Nguyen, M. T. (2018). Characterization of growth and morphology of the non-visual retina in medaka (oryzias latipes).

Nilsson, D.-E. (2013). Eye evolution and its functional basis. *Visual neuroscience*, 30(1-2):5–20.

Nishiwaki, Y., Oishi, T., Tokunaga, F., and Morita, T. (1997). Three-dimensional reconstitution of cone arrangement on the spherical surface of the retina in the medaka eyes. *Zoological science*, 14(5):795–802.

Ogawa, N., Hatsuda, T., Mochizuki, A., and Tachikawa, M. (2017). Dynamical pattern selection of growing cellular mosaic in fish retina. *Physical Review E*, 96(3):032416.

Ohki, H. and Aoki, K. (1985). Development of visual acuity in the larval medaka, oryzias latipes (communication)(physiology). *Zoological science*, 2(1):123–126.

Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.

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.

Ott, M. (2006). Visual accommodation in vertebrates: mechanisms, physiological response and stimuli. *Journal of Comparative Physiology A*, 192(2):97.

Pathmanathan, P., Cooper, J., Fletcher, A., Mirams, G., Murray, P., Osborne, J., Pitt-Francis, J., Walter, A., and Chapman, S. (2009). A computational study of discrete mechanical tissue models. *Physical biology*, 6(3):036001.

Phansalkar, N., More, S., Sabale, A., and Joshi, M. (2011). Adaptive local thresholding for detection of nuclei in diversity stained cytology images. In *2011 International Conference on Communications and Signal Processing*, pages 218–220. IEEE.

Preibisch, S., Saalfeld, S., and Tomancak, P. (2009). Globally optimal stitching of tiled 3d microscopic image acquisitions. *Bioinformatics*, 25(11):1463–1465.

R Core Team (2015). R: A language and environment for statistical computing.

Raymond, P. A., Barthel, L. K., Bernardos, R. L., and Perkowski, J. J. (2006). Molecular characterization of retinal stem cells and their niches in adult zebrafish. *BMC developmental biology*, 6(1):36.

Reinhardt, R., Centanin, L., Tavhelidse, T., Inoue, D., Wittbrodt, B., Concordet, J.-P., Martinez-Morales, J. R., and Wittbrodt, J. (2015). Sox2, tlx, gli3, and her9 converge on rx2 to define retinal stem cells in vivo. *The EMBO journal*, 34(11):1572–1588.

Rembold, M., Loosli, F., Adams, R. J., and Wittbrodt, J. (2006). Individual cell migration serves as the driving force for optic vesicle evagination. *Science*, 313(5790):1130–1134.

Reyer, R. (1977). The amphibian eye: Development and regeneration. In *The visual system in vertebrates*, pages 309–390. Springer.

Ritsma, L., Ellenbroek, S. I., Zomer, A., Snippert, H. J., de Sauvage, F. J., Simons, B. D., Clevers, H., and van Rheenen, J. (2014). Intestinal crypt homeostasis revealed at single-stem-cell level by in vivo live imaging. *Nature*, 507(7492):362.

Roselló-Díez, A. and Joyner, A. L. (2015). Regulation of long bone growth in vertebrates; it is time to catch up. *Endocrine reviews*, 36(6):646–680.

Roselló-Díez, A., Stephen, D., and Joyner, A. L. (2017). Altered paracrine signaling from the injured knee joint impairs postnatal long bone growth. *Elife*, 6:e27210.

RStudio distribution of R (2015). `https://www.rstudio.com/`.

Rumman, M., Dhawan, J., and Kassem, M. (2015). Concise review: quiescence in adult stem cells: biological significance and relevance to tissue regeneration. *Stem cells*, 33(10):2903–2912.

Salbreux, G., Barthel, L. K., Raymond, P. A., and Lubensky, D. K. (2012). Coupling mechanical deformations and planar cell polarity to create regular patterns in the zebrafish retina. *PLoS computational biology*, 8(8):e1002618.

Sandler, O., Mizrahi, S. P., Weiss, N., Agam, O., Simon, I., and Balaban, N. Q. (2015). Lineage correlations of single cell division time as a probe of cell-cycle dynamics. *Nature*, 519(7544):468.

Saturnino, A. P., Lust, K., and Wittbrodt, J. (2018). Notch signalling patterns retinal composition by regulating atoh7 during post-embryonic growth. *Development*, 145(21):dev169698.

Scadden, D. T. (2014). Nice neighborhood: emerging concepts of the stem cell niche. *Cell*, 157(1):41–50.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676.

Shen, W. and Sivak, J. G. (2007). Eyes of a lower vertebrate are susceptible to the visual environment. *Investigative ophthalmology & visual science*, 48(10):4829–4837.

Shen, W., Vijayan, M., and Sivak, J. G. (2005). Inducing form-deprivation myopia in fish. *Investigative ophthalmology & visual science*, 46(5):1797–1803.

Shi, D., Tavhelidse, T., Thumberger, T., Wittbrodt, J., and Greb, T. (2017). Bifacial stem cell niches in fish and plants. *Current opinion in genetics & development*, 45:28–33.

Snippert, H. J., Van Der Flier, L. G., Sato, T., Van Es, J. H., Van Den Born, M., Kroon-Veenboer, C., Barker, N., Klein, A. M., Van Rheenen, J., Simons, B. D., et al. (2010). Intestinal crypt homeostasis results from neutral competition between symmetrically dividing lgr5 stem cells. *Cell*, 143(1):134–144.

Sokolova, N. (2019). Neuroepithelial flow and bmp signaling shape the optic cup in medaka. Master's thesis, Ruprecht-Karls Universität Heidelberg.

Soules, K. A. and Link, B. A. (2005). Morphogenesis of the anterior segment in the zebrafish eye. *BMC Developmental Biology*, 5(1):12.

Stemmer, M., Thumberger, T., del Sol Keyer, M., Wittbrodt, J., and Mateo, J. L. (2015). Cctop: an intuitive, flexible and reliable crispr/cas9 target prediction tool. *PloS one*, 10(4):e0124633.

Stenkamp, D. L. (2007). Neurogenesis in the fish retina. *International review of cytology*, 259:173–224.

Sütterlin, T. (2019). Episim simulator. https://gitlab.com/EPISIM/EPISIM-Simulator, commit 4e0a37e7b0c23155aeee347bbdad98334f81406b.

Sütterlin, T. (2015). *Entwurf und Realisierung eines computergestützten Systems zur in silico Modellierung und Simulation von Epithelgeweben*. PhD thesis, Ruprecht-Karls-Universität Heidelberg.

Sütterlin, T., Kolb, C., Dickhaus, H., Jäger, D., and Grabe, N. (2012). Bridging the scales: semantic integration of quantitative sbml in graphical multi-cellular models and simulations with episim and copasi. *Bioinformatics*, 29(2):223–229.

Sütterlin, T., Tsingos, E., Bensaci, J., Stamatas, G. N., and Grabe, N. (2017). A 3d self-organizing multicellular epidermis model of barrier formation and hydration with realistic cell morphology based on episim. *Scientific reports*, 7:43472.

Takesue, A., Mochizuki, A., and Iwasa, Y. (1998). Cell-differentiation rules that generate regular mosaic patterns: modelling motivated by cone mosaic formation in fish retina. *Journal of Theoretical Biology*, 194(4):575–586.

Tang, X., Gao, J., Jia, X., Zhao, W., Zhang, Y., Pan, W., and He, J. (2017). Bipotent progenitors as embryonic origin of retinal stem cells. *J Cell Biol*, 216(6):1833–1847.

Tavhelidse, T. (2019). *The role of rx genes in establishment and maintenance of the medaka ciliary marginal zone*. PhD thesis, Ruprecht-Karls Universität Heidelberg.

Thermes, V., Grabher, C., Ristoratore, F., Bourrat, F., Choulika, A., Wittbrodt, J., and Joly, J.-S. (2002). I-scei meganuclease mediates highly efficient transgenesis in fish. *Mechanisms of development*, 118(1-2):91–98.

ThermoFischer Scientific (2011). Click-it® edu imaging kits. https://www.thermofisher.com/order/catalog/product/C10340.

Todd, L., Suarez, L., Squires, N., Zelinka, C. P., Gribbins, K., and Fischer, A. J. (2016). Comparative analysis of glucagonergic cells, glia, and the circumferential marginal zone in the reptilian retina. *Journal of Comparative Neurology*, 524(1):74–89.

Tohya, S., Mochizuki, A., and Iwasa, Y. (1999). Formation of cone mosaic of zebrafish retina. *Journal of theoretical biology*, 200(2):231–244.

Tropepe, V., Coles, B. L., Chiasson, B. J., Horsford, D. J., Elia, A. J., McInnes, R. R., and van der Kooy, D. (2000). Retinal stem cells in the adult mammalian eye. *Science*, 287(5460):2032–2036.

Tsingos, E., Hoeckendorf, B., Suetterlin, T., Kirchmaier, S., Grabe, N., Centanin, L., and Wittbrodt, J. (2019). Retinal stem cells modulate proliferative parameters to coordinate post-embryonic morphogenesis in the eye of fish. *eLife*, 8:e42646.

Tsonis, P. A. (2000). Regeneration in vertebrates. *Developmental biology*, 221(2):273–284.

Twitty, V. C. and Schwind, J. L. (1931). The growth of eyes and limbs transplanted heteroplastically between two species of amblystoma. *Journal of Experimental Zoology*, 59(1):61–86.

Tyler, M. J., Carney, L. H., and Cameron, D. A. (2005). Control of cellular pattern formation in the vertebrate inner retina by homotypic regulation of cell-fate decisions. *Journal of Neuroscience*, 25(18):4565–4576.

Van Liedekerke, P., Palm, M., Jagiella, N., and Drasdo, D. (2015). Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. *Computational particle mechanics*, 2(4):401–444.

Venters, S. J., Cuenca, P. D., and Hyer, J. (2011). Retinal and anterior eye compartments derive from a common progenitor pool in the avian optic cup. *Molecular vision*, 17:3347.

Venters, S. J., Mikawa, T., and Hyer, J. (2013). Central and peripheral retina arise through distinct developmental paths. *PloS one*, 8(4):e61422.

Venters, S. J., Mikawa, T., and Hyer, J. (2015). Early divergence of central and peripheral neural retina precursors during vertebrate eye development. *Developmental Dynamics*, 244(3):266–276.

Veth, K. N., Willer, J. R., Collery, R. F., Gray, M. P., Willer, G. B., Wagner, D. S., Mullins, M. C., Udvadia, A. J., Smith, R. S., John, S. W., et al. (2011). Mutations in zebrafish lrp2 result in adult-onset ocular pathogenesis that models myopia and other risk factors for glaucoma. *PLoS genetics*, 7(2):e1001310.

Vuilleumier, R., Affolter, M., and Pyrowolakis, G. (2011). Pentagone: patrolling bmp morphogen signaling. *Fly*, 5(3):210–214.

Wallman, J. and Winawer, J. (2004). Homeostasis of eye growth and the question of myopia. *Neuron*, 43(4):447–468.

Walls, G. L. (1942). *The vertebrate eye and its adaptive radiation. Cranbrook Institute of Science.* Cranbrook Press.

Wan, Y., Almeida, A. D., Rulands, S., Chalour, N., Muresan, L., Wu, Y., Simons, B. D., He, J., and Harris, W. A. (2016). The ciliary marginal zone of the zebrafish retina: clonal and time-lapse analysis of a continuously growing tissue. *Development*, 143(7):1099–1107.

Wartlick, O., Mumcu, P., Kicheva, A., Bittig, T., Seum, C., Jülicher, F., and Gonzalez-Gaitan, M. (2011). Dynamics of dpp signaling and proliferation control. *science*, 331(6021):1154–1159.

Watt, F. M. and Hogan, B. L. (2000). Out of eden: stem cells and their niches. *science*, 287(5457):1427–1430.

Weekes, J. M. (2015). Investigating the role of bmp antagonists in vertebrate eye development. Master's thesis, University of Alberta.

Wetts, R. and Fraser, S. E. (1988). Multipotent precursors can give rise to all major cell types of the frog retina. *Science*, 239(4844):1142–1145.

Wetts, R., Serbedzija, G. N., and Fraser, S. E. (1989). Cell lineage analysis reveals multipotent precursors in the ciliary margin of the frog retina. *Developmental biology*, 136(1):254–263.

Wickham, H. (2012). reshape2: Flexibly reshape data: a reboot of the reshape package. *R package version*, 1(2).

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Williams, A. L. and Bohnsack, B. L. (2015). Neural crest derivatives in ocular development: Discerning the eye of the storm. *Birth Defects Research Part C: Embryo Today: Reviews*, 105(2):87–95.

WinPython distribution of Python (2016). http://winpython.github.io/.

Zerjatke, T., Gak, I. A., Kirova, D., Fuhrmann, M., Daniel, K., Gonciarz, M., Müller, D., Glauche, I., and Mansfeld, J. (2017). Quantitative cell cycle analysis based on an endogenous all-in-one reporter for cell tracking and classification. *Cell reports*, 19(9):1953–1966.

Zhang, B., Tsai, P.-C., Gonzalez-Celeiro, M., Chung, O., Boumard, B., Perdigoto, C. N., Ezhkova, E., and Hsu, Y.-C. (2016). Hair follicles' transit-amplifying cells govern concurrent dermal adipocyte production through sonic hedgehog. *Genes & development*, 30(20):2325–2338.

Zimmermann, M. J., Nevala, N. E., Yoshimatsu, T., Osorio, D., Nilsson, D.-E., Berens, P., and Baden, T. (2018). Zebrafish differentially process color across visual space to match natural scenes. *Current Biology*, 28(13).

Appendix

5.6 Additional work

In this section, I present other work that was not directly related to the stated aims of the thesis.

5.6.1 Generation of a medaka *lrp2a* mutant

Zebrafish *lrp2a* mutants exhibited increased intra-ocular pressure that lead to pathological eye enlargement [Veth et al., 2011]. As this mutant provided a unique opportunity to decouple eye growth coordination, I decided to recapitulate it in medaka using CRISPR/Cas9 to ultimately do an ArCoS analysis in the mutant background.

In situ hybridization

First I verified the expression domain of *lrp2a* in medaka by *in situ* hybridization. As described for zebrafish [Veth et al., 2011], medaka *lrp2a* localised to the inner ear, the epithelium lining the brain ventricle, and to peripheral retinal structures (Figure 5.2). Unlike in zebrafish, an expression in the pronephros was not detected.

Generation of medaka *lrp2a* mutant line

With the help of Dr Thomas Thumberger, I used the CCTop tool to design sgRNA directed towards the second annotated exon of *lrp2a* (Ensembl gene ID: ENSORLG00000017126) [Stemmer et al., 2015] as well as homology arms 5' to the protospacer adjacent motif (PAM) to integrate a GFP probe via homology-directed repair (HDR) and knock out the gene in the process (Figure 5.1 A). The first annotated exon lacked a valid reading frame with a start codon; sequence comparison to zebrafish *lrp2a* mRNA from Veth et al. [2011] suggested there was an upstream putative first exon. Nevertheless, I proceeded with the initially devised strategy. The transgenesis strategy consisted of co-injection of CRISPR/Cas9 machinery with plasmid #4280, which contained a sgRNA target site absent from the medaka genome 5' to the homology arm flanking the GFP cassette (see Methods Section 5.2.13 for cloning strategy).

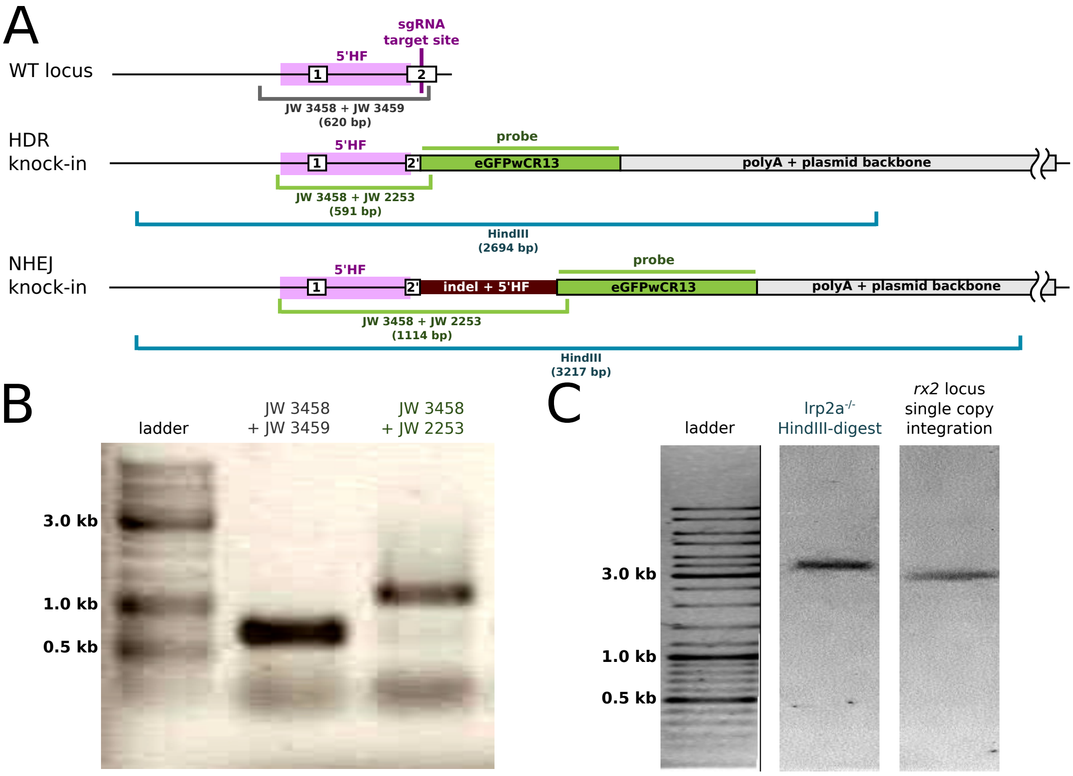


Figure 5.1: Design and validation of CRISPR knock-in.

A Top: The sgRNA target site was chosen in the second annotated exon. Flanking primers JW 3458 and JW 3459 were designed to amplify the locus for genotyping. A 5' HF was cloned to instigate HDR. Middle: Appearance of the locus in case of HDR-mediated knock-in. Primers JW 3458 and JW 2253 were designed to amplify the locus containing the GFP insertion. Bottom: Experimentally observed locus. The 5' HF was duplicated by NHEJ. Primer pair JW 3458 and JW 2253, and HindIII digest result in longer fragments than for HDR-mediated donor integration. **B** Agarose gel of a genotyping PCR of a heterozygous fish. The band amplified by pair JW 3458 and JW 2253 was sent for sequencing, showing an indel and duplication of the 5' HF. **C** Composite image of DNA ladder on agarose gel (left) and Southern blot lanes (middle and right). Middle lane: HindIII-digest of a homozygous *lrp2a* mutant fish detected with a probe directed against GFP. Right lane: Another lane from the same blot and using the same probe showing a single-copy integration in the *rx2* locus generated by HDR [Gutierrez-Triana et al., 2018]. This sample is shown here as a comparison of band intensity.

Among the injected generation, several GFP-positive embryos displayed various degrees of head and eye malformations (Figure 5.2 B), similar to the holoprosenchephalic phenotypes described in mouse *lrp2* mutants [Christ et al., 2012]. Embryos that appeared morphologically normal and had weak GFP expression partially corresponding to the pattern seen in *in situ* hybridizations were raised and screened for germline transmission. The line was named *CR(lrp2a::eGFPwCR13)*. Germline transmission was very high: Out of 5 initially screened injected adults, 2 produced GFP-positive progeny when outcrossed to *cab*; these fish were called L2 and L4. For both founders, roughly 20% of the progeny were carriers for the transgene. GFP expression recapitulated the *in situ* pattern; strongest expression was in the otoliths (Figure 5.2 C). Expression was

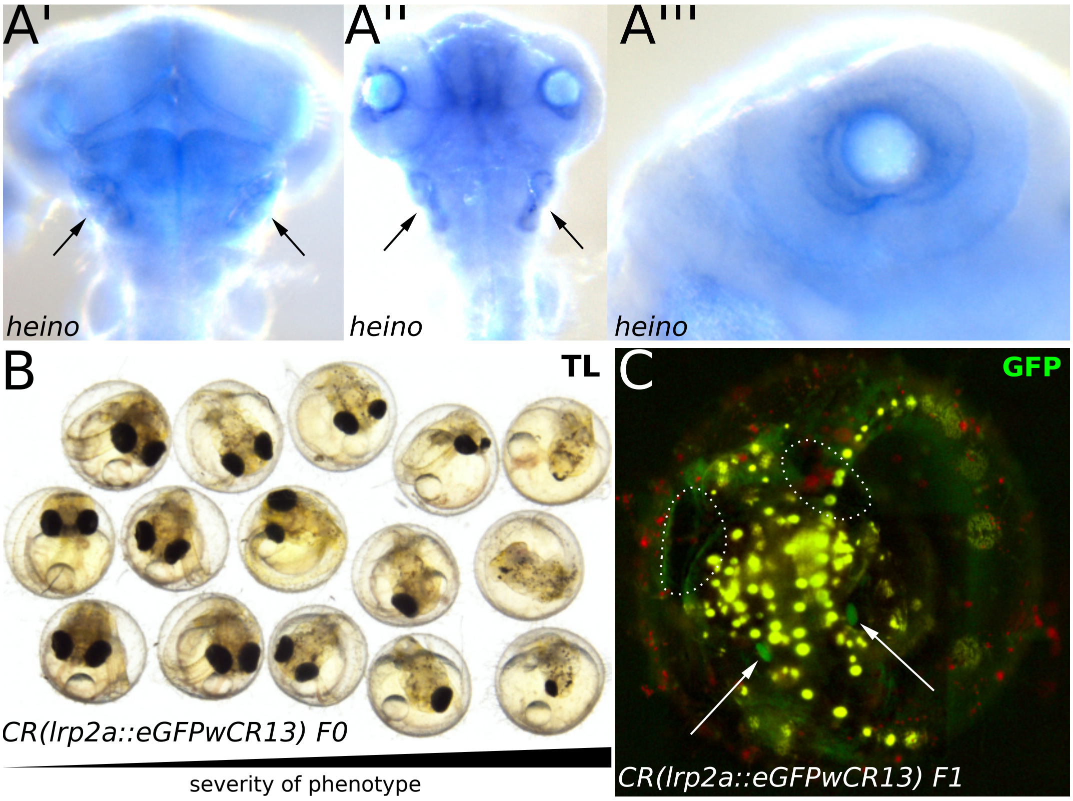


Figure 5.2: *In situ* hybridization and CRISPR knock-in of GFP into the *lrp2a* locus in medaka. **A′** Dorsal view on the head (anterior up) of a stage 34 embryo. The brain ventricles are lined by two fine layers of expression. The expression domains at the ventricle extend through the entire neural tube. The inner ear (arrows) shows strongest label. **A″** Ventral view on the head (anterior up) of a stage 34 embryo. The region around the retina and around the lens also stains heavily. Arrows: Inner ear. **A‴** Lateral view on the eye (anterior left) of a stage 34 embryo. **B** Fish injected with sgRNA against annotated exon 2 of *lrp2a* show various degree of head and eye defects. **C** Typical F1 embryo with visible GFP knock-in. GFP expression was faintly visible lining the brain and spinal cord ventricles; expression surrounding the lens was very faint. The strongest expression was visible in the otoliths (white arrows), which were used for screening. Dotted lines highlight position of eyes.

stronger in progeny derived from founder L4. Embryos were screened based on GFP expression. GFP-positive progeny of both lines were raised and outcrossed to *cab*; individual F2 generation couples were incrossed.

Consistent with the data from zebrafish, homozygous F3 fish revealed an adult-onset enlargening of the eyeballs. Upon closer inspection, the phenotype was already visible by 14 dph (Figure 5.3), aligning with the post-embryonic development of the NVR which regulates intra-ocular fluid influx. Penetrance was 100%, but expressivity varied. Individual fish showed differences in left and right eyes (Figure 5.4 A″). Besides enlarged eyes, no other anatomical or behavioral abnormality was noted. Fish derived from founder L2 had generally a stronger expressivity, and thus I chose to proceed with this line.

The F3 generation was genotyped using primers JW 3458 and JW 3459 to de-

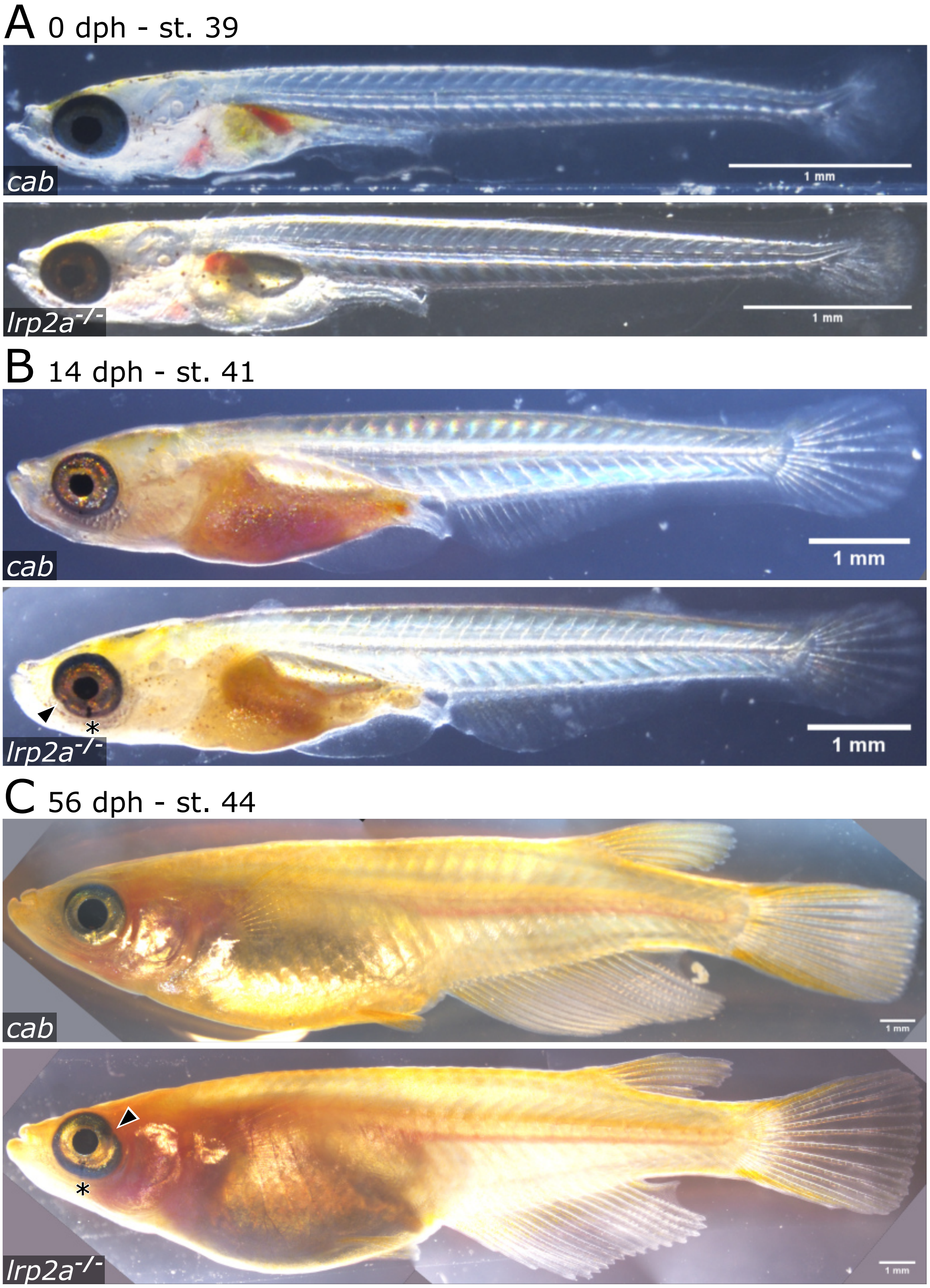


Figure 5.3: Development of the phenotype in *lrp2a* mutants.

A *Cab* and *lrp2a* mutant hatchlings show no obvious morphological difference. **B** Upon close inspection, slight bulging (arrowhead) can be observed in the eyes of *lrp2a* mutant larvae at 14 dph. The ventral blood vessel is enlarged (asterisk). **C** Adult *lrp2a* mutants display eye bulging and deformations to different degrees (arrowhead). The position of indentations correlates with the position of extra-ocular muscles. The ventral blood vessel is enlarged (asterisk). Images are stitches of three pictures each.

tect the wildtype locus, and JW 3458 and JW 2253 to detect the GFP insertion in the locus. In the line derived from founder L2 a band shift occurred relative to the expected locus size (Figure 5.1 B). Sequencing confirmed that the transgenic cassette had entered via non-homologous end joining (NHEJ), and the 5' homology arm was duplicated. Southern blot analysis confirmed that no other locus was affected and that the integration was most likely via NHEJ. Although no further bands were observed, band intensity was stronger in this line compared to another characterised line with a single-copy integration [Gutierrez-Triana et al., 2018], suggesting more than one insertion in tandem had occurred (Figure 5.1 C). However, this was not verified.

ArCoS analysis in medaka *lrp2a* mutants

The line derived from founder L2 was crossed into a *Gaudí*^{RSG} background. After backcross to *CR(lrp2a::eGFPwCR13)* double mutants I induced individuals, then separated heterozygotes from homozygotes by genotyping at adult stage. A few adults were sacrificed to analyze NR ArCoS. Depending on the severity of the phenotype, ArCoS ranged from apparently unaffected to thinly spread along the stretched-out retina (Figure 5.4). These data were consistent with a failure of CMZ cells to compensate for extreme tissue stretch. Unfortunately, no RPE samples were recovered as the RPE was extremely thin and labile in these fish.

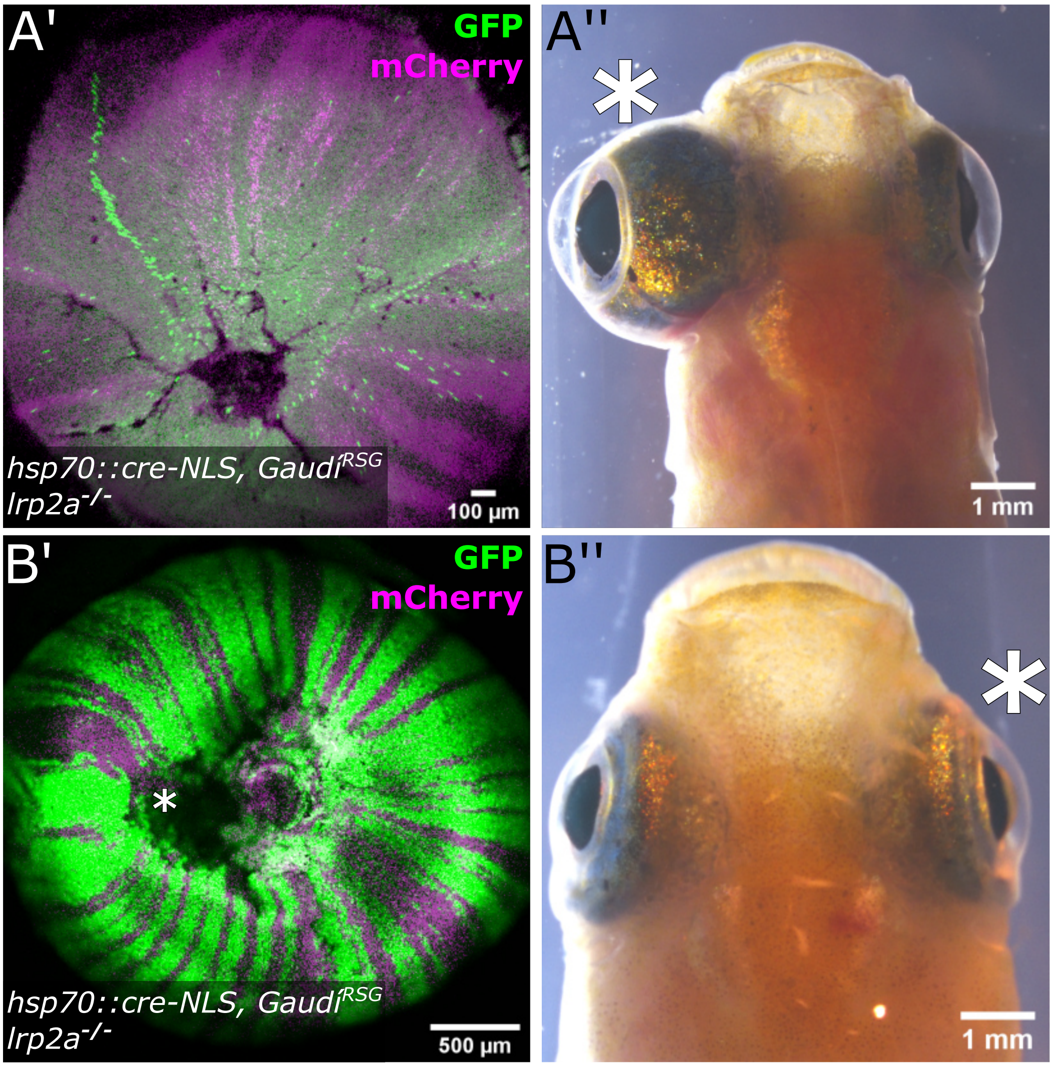


Figure 5.4: NR ArCoS in the *lrp2a* mutant background become stretched only in very large eyes. **A′** Mutant retina showing one string-like ArCoS and several dispersed cells that could have been stretched away from each other. **A″** Dorsal photo of adult mutant fish with strong phenotype in one eye. Asterisk: Eye corresponding to A′. **B′** Almost all ArCoS look typical. Asterisk: Clones that appear to become curved correspond to the position of a macroscopic indentation visible in the eye of the fish before dissection. **B″** Dorsal photo of adult mutant fish with weak phenotype in both eyes. Asterisk: Eye corresponding to B′.

5.6.2 Epigenetic imprinting at the transition from embryonic to post-embryonic growth in the medaka retina

Unbleached preparations of medaka retinae where I removed the RPE mechanically (Methods Section 5.2.23) allowed to visualise endogenous fluorescent protein expression. Unexpectedly, regardless of *cre* driver, retinae of the *Gaudí*^{RSG} line revealed a striped pattern of mCherry expression levels highly reminiscent of ArCoS in the absence of GFP expression (Figure 5.5 A′–A‴). Upon closer inspection, a similar pattern was visible in the recombined GFP ArCoS (Figure 5.5 B′–B‴).

As the fish used for the experiment were heterozygotes for the *Gaudí*^{RSG} transgene, the differences could not be ascribed to differential recombination in two alleles. Insertions of the transgene into multiple loci of the genome was also unlikely as the line had been outcrossed several times and the transgene segregated in a Mendelian fashion. Southern blot analysis revealed a massive band and two weaker bands detected by GFP probe indicating that the transgene inserted into the same locus numerous times in tandem (Figure 5.6 A). Thus, differential levels of recombination could be explained by a multimerised *Gaudí*^{RSG} cassette in the same genomic region.

An epigenetic pattern of expression is imprinted in founder SCs

If the pattern of expression levels resulted from differential degree of recombination of a multimerised cassette, then fish of the *Gaudí*^{RSG} line lacking a *cre* in the genome should display uniform mCherry expression. Surprisingly, an ArCoS banding pattern also occurred in these fish (Figure 5.6 A). Spontaneous recombination was never observed. This pattern could be explained by epigenetic silencing of mCherry expression in a variable number of the inserted cassettes; silencing events imprinted once in founder SCs were propagated lifelong. Notably, the central embryonic part of the retina did not display any pattern, suggesting that the imprinting event must have occurred at the transition from embryonic to post-embryonic growth modes.

To verify whether epigenetic activity was specific to the locus where the *Gaudí*^{RSG} transgene inserted or specific to the construct, I examined three additional *Gaudí* lines (Figure 5.6 D–F); for each line, I analyzed four retinae from two individuals. Epigenetic imprinting patterns with ArCoS darker and brighter than usual occurred in the *Gaudí*^{LxBBW} line (Figure 5.6 D) and in the *Gaudí*$^{LoxP\text{-}OUT}$ line (Figure 5.6 E), but not in the *Gaudí*$^{BBW2.1}$ line (Figure 5.6 F).

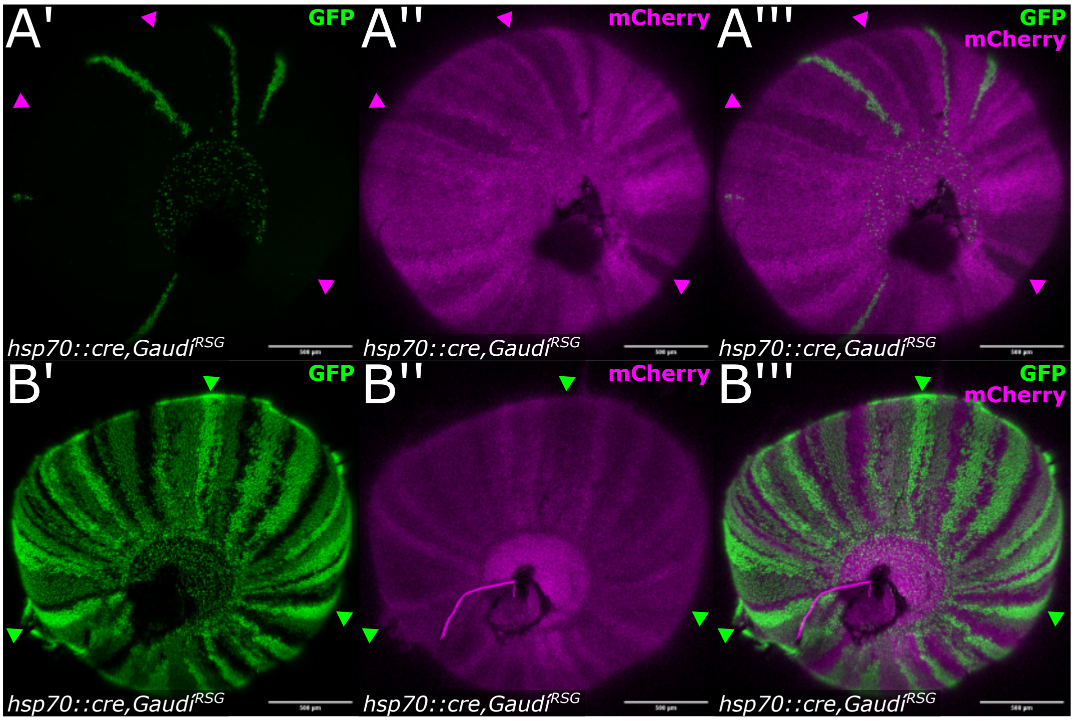


Figure 5.5: Striped pattern of intensity differences of endogenous mCherry and GFP. **A′–A‴** Radial stripes of differential endogenous mCherry expression in the absence of GFP expression. Magenta arrowheads: Three examples of particularly low expression domains. **B′–B‴** Radial stripes of differential endogenous GFP expression; higher GFP expression does not necessarily correlate with lower mCherry expression. Green arrowheads: Three examples of particularly high expression domains.

The data in the *Gaudí*^{LxBBW} line showed that imprinting could occur with other constructs. Moreover, in addition to silencing, activatory imprinting could also occur (Figure 5.6 E), asterisk. Both *Gaudí*^{LxBBW} and *Gaudí*$^{BBW2.1}$ lines were generated by independent injections, indicating that epigenetic imprinting could occur in a locus-dependent fashion, but was not restricted to a single locus.

Epigenetic imprinting correlates with plasmid backbone integration

The transgenic cassettes used to generate these lines were integrated into the genome with meganuclease, a restriction enzyme that cuts an 18 bp-long sequence of DNA [Grabher et al., 2004]. Co-injection of meganuclease improves transgenesis in fish by cutting out the transgenic cassette from the vector backbone. Importantly, meganuclease protects the longer sticky ends, which prevents concatemerisation thus improving the chances of integration of few copy numbers into the genome [Grabher et al., 2004; Thermes et al., 2002]. Therefore, to incorporate a transgenic cassette into the genome, the meganuclease sites have to be oriented facing each other [Grabher et al., 2004; Thermes et al., 2002].

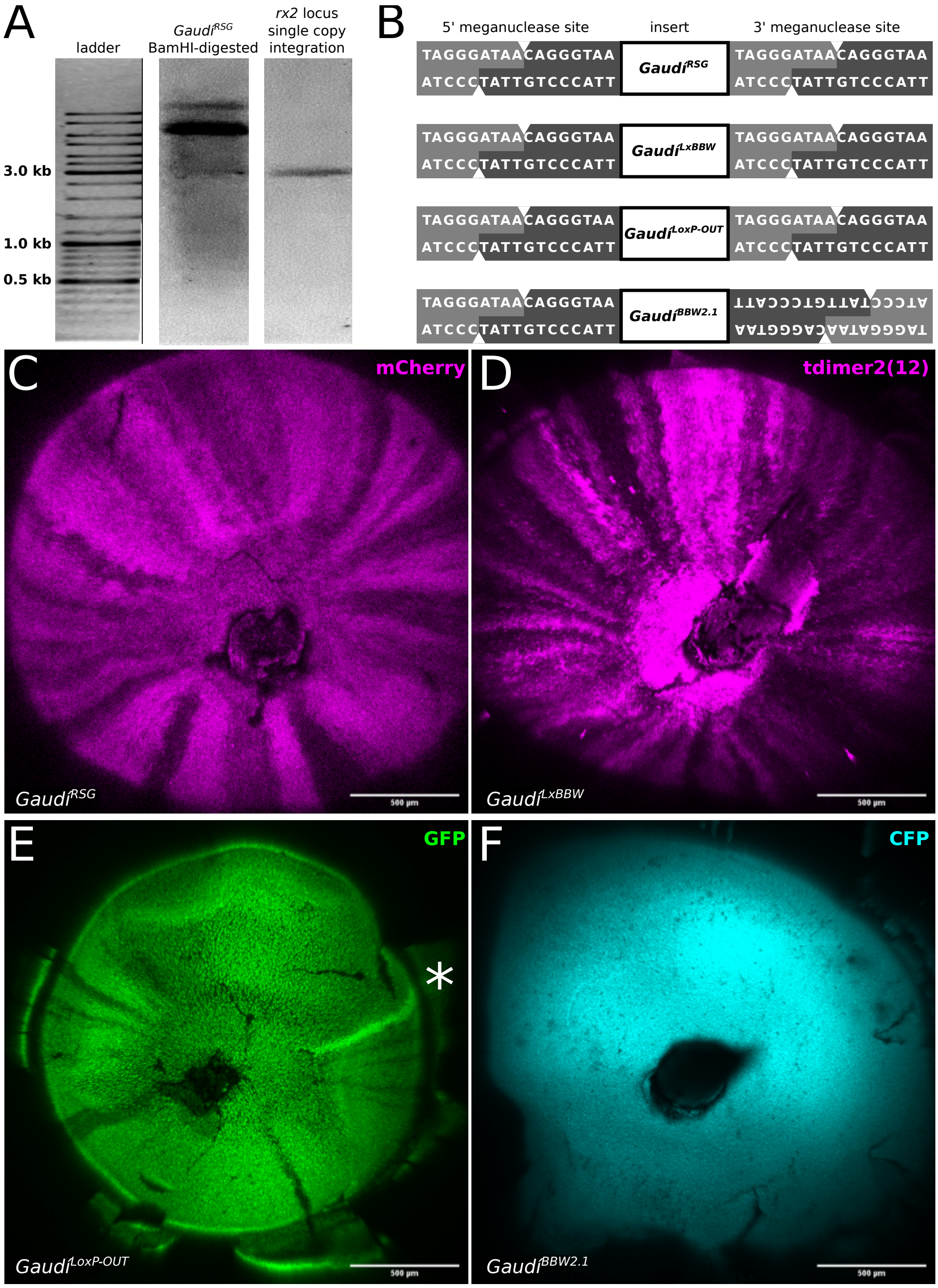


Figure 5.6: Stripes of expression intensity correlate with misoriented meganuclease sites. **A** Composite image of DNA ladder on agarose gel (left) and Southern blot lanes (middle and right). Middle lane: BamHI-digest of a fish from the *Gaudí*^{RSG} line detected with a probe directed against GFP. Right lane: Another lane from the same blot and using the same probe showing a single-copy integration in the *rx2* locus generated by HDR [Gutierrez-Triana et al., 2018]. This sample is shown here as a comparison of band intensity. The data are from the same experiment shown in Figure 5.1 C. **B** Orientation of 5' and 3' meganuclease sites for the lines shown in panels C–F. **C** *Gaudí*^{RSG} retina with endogenous mCherry fluorescence. **D** *Gaudí*^{LxBBW} retina with endogenous tdimer2(12) fluorescence. **E** *Gaudí*$^{LoxP\text{-}OUT}$ retina with endogenous GFP fluorescence. Asterisk: Radial stripe with unusually strong GFP expression. **F** *Gaudí*$^{BBW2.1}$ retina with endogenous CFP fluorescence.

However, the lines that showed epigenetic imprinting had meganuclease sites facing the same direction (Figure 5.6 B). Thus, only one end of both insert and plasmid backbone were protected from concatamerisation. Consequently, these constructs were prone to multimerise, leading to local enrichment of bacterial and viral sequences used in the cloning vector. Tandem insertions, and repetitive foreign DNA such as transposable elements are often silenced by the epigenetic machinery in eukaryotes [Garrick et al., 1998; Iida et al., 2006]. Thus, the multimerised loci likely functioned as an attractor for the epigenetic machinery.

SCs downregulate the epigenetic machinery at the transition from embryonic to post-embryonic growth mode

Recently, Dr Lucie Zilova showed that DNA methyl transferase genes (*dnmt*s) are expressed in the embryonic optic cup margin, but are downregulated in nascent retinal SCs during the transition from an embryonic to a post-embryonic mode of growth in medaka [Zilova, Gutierrez-Triana, and Wittbrodt, unpublished]. These data suggested an explanation for the pattern of epigenetic inheritance of fluorophore expression: The multimerised loci of the *Gaudí* lines were targeted by the epigenetic machinery during early embryonic stages, resulting in random degrees of silencing or activation. Then, at the transition to post-embryonic growth, nascent SCs stopped to modify their epigenetic landscape, thus fixing the last imprinted pattern. Surprisingly, even though CMZ PCs retained *dnmt* expression, they did not alter the pattern passed down by SCs, suggesting that they only maintained existing methylation patterns, without any *de novo* alterations.

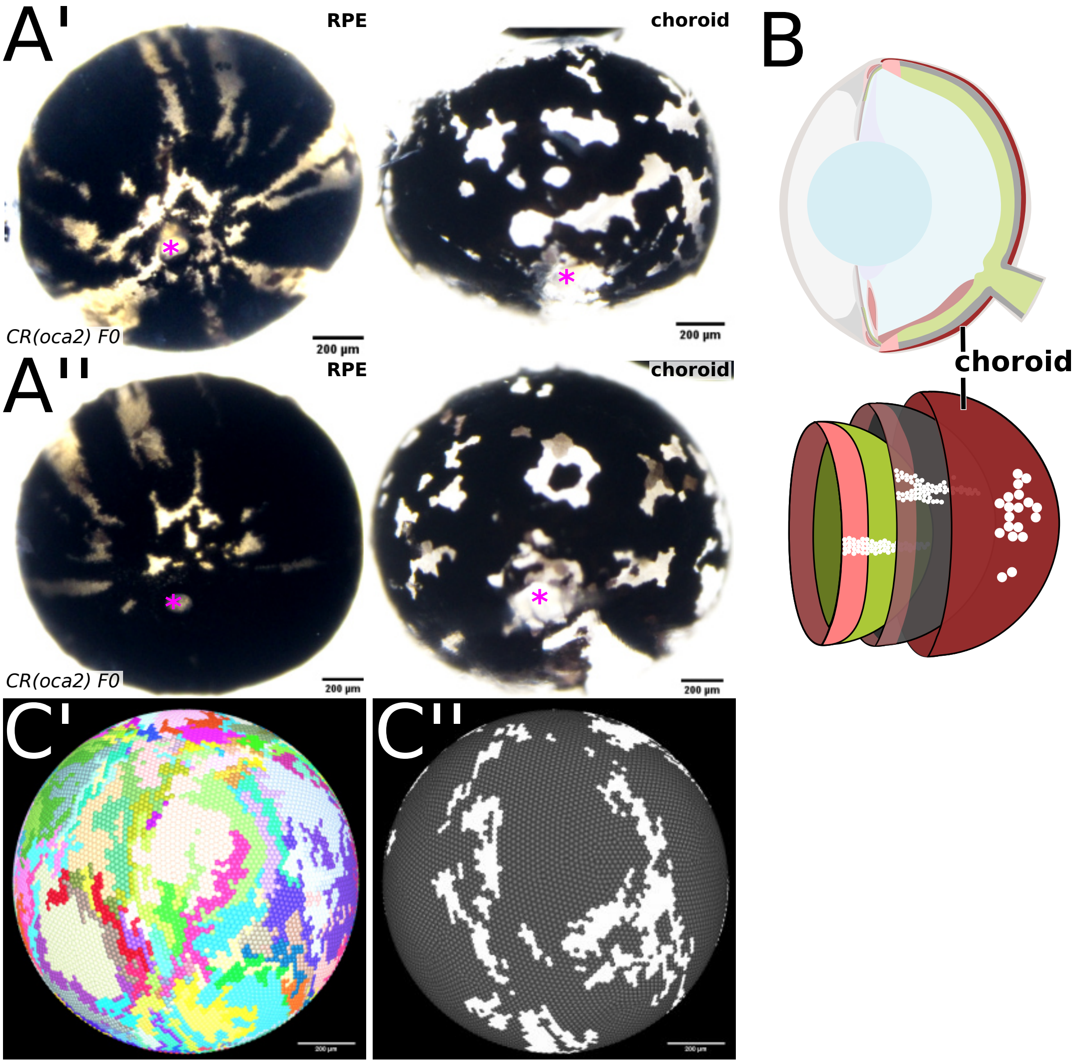


Figure 5.7: Choroid melanocytes clones are consistent with distributed proliferation
A′–A″ Two corresponding pairs of RPE and choroid melanocyte layer with mosaic albino clones. Asterisk: Optic nerve exit. **B** Top: Schematic cross-section of the eye highlighting the choroid. Bottom: 3D scheme of clonal growth in the NR, RPE, and choroid. **C′–C″** Screenshots of a choroid growth simulation using the responder growth mode. The CMZ width was increased to $w = 2000\ \mu\text{m}$, and cell radius $r = 12.5\ \mu\text{m}$. Other parameters were as listed in Tables 2.1 and 2.2. C′ shows all lineages in unique colours, C″ a random selection of lineages in black or white.

5.6.3 Clonal growth of choroid melanocytes

The tissue surrounding the RPE, the choroid, contains a continuous layer of melanocytes in medaka [Hirose and Matsumoto, 1994]. The choroid of mosaic F0 *oca2* crispants could be recovered during dissection in some cases, revealing a distributed pattern of clonal patches (Figure 5.7 A′–A″). This pattern was consistent with proliferation all over the tissue, as tested in the model (Figure 5.7 C′–C″). The capacity of cells to proliferate all over the tissue was verified by lineage tracing with the *Gaudí*^{RSG} line and thymidine analogue incorpora-

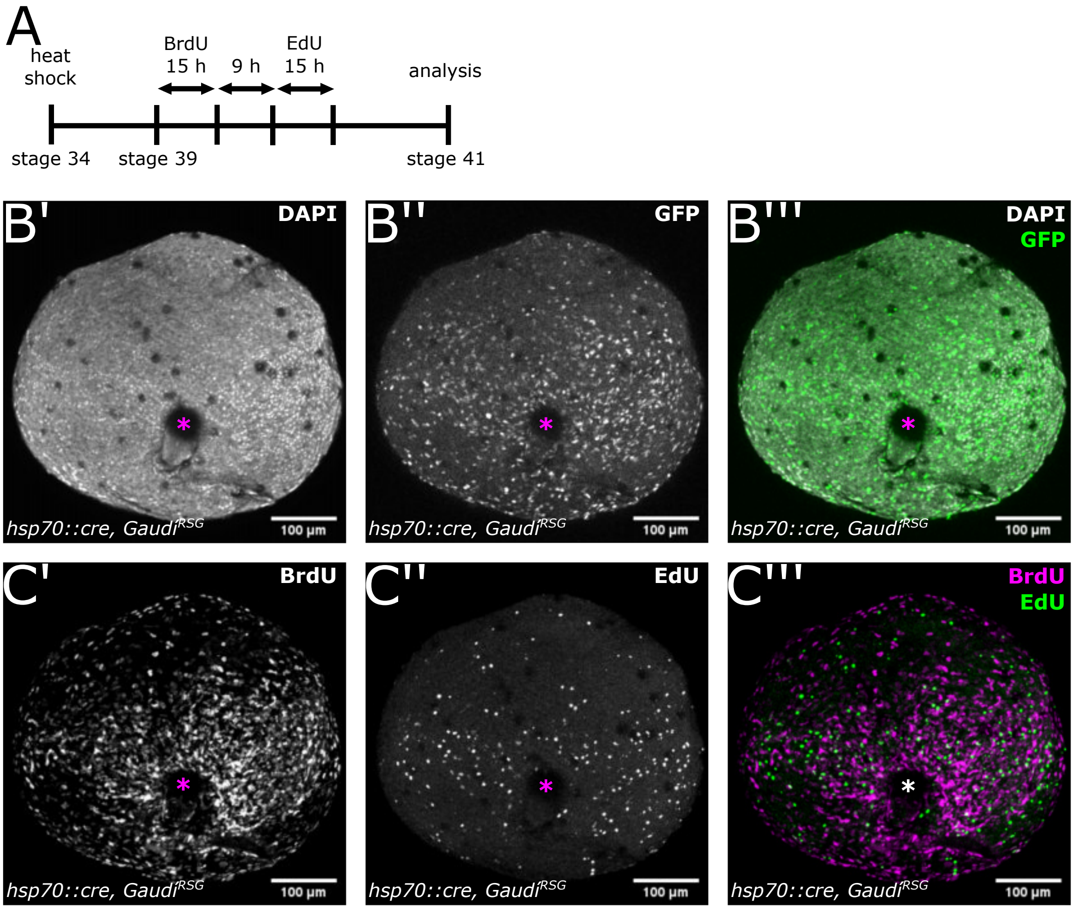


Figure 5.8: Distributed proliferative potential in the choroid.
A′ Scheme of the experiment. Fish were fixed at 10 dph. **B′** DAPI. **B″** GFP. **B‴** Overlay of DAPI and GFP. **C′** BrdU. **C″** EdU. **C‴** Overlay of BrdU and EdU. Panels B′–C‴ show the same sample; the image was rotated to place the optic nerve (asterisk) ventrally.

tion (Figure 5.8 A–C‴). Thus, like the NVR, the choroid melanocyte layer lacks a dedicated stem cell niche.

5.6.4 Characterisation of the eye phenotype in zebrafish smoc1 mutant

In the *Drosophila* imaginal wing disc, tissue growth depends on the signalling molecule decapentaplegic (Dpp). Dpp forms a gradient that scales with the size of the organ due to changing ligand degradation rate over time [Wartlick et al., 2011]. The dynamics of Dpp turnover and thus gradient scaling are influenced by Pentagone, an extracellular protein that competes with Dpp for binding to membrane proteins that promote cell-to-cell ligand movement [Vuilleumier et al., 2011]. As a result, in *Drosophila* wing discs mutant for Pentagone, the Dpp gradient has a shorter range [Vuilleumier et al., 2011].

To assess whether a similar gradient scaling system is conserved in vertebrates, Dr Rita Mateus, a researcher in the lab of Marcos González-Gaitán at University of Geneva, generated a zebrafish mutant for SPARC Related Modular Calcium Binding 1 gene (*smoc1*), the vertebrate orthologue of *Drosophila* Pentagone. Dr Rita Mateus generated a CRISPR mutant with a 1 bp deletion in exon 4 that lead to a frameshift and premature stop codon [Mateus, and González-Gaitán, unpublished]. Mutant embryos had a shorter body axis, smaller eyes and fins (Figure 5.9 A–B; personal communication with Dr Rita Mateus). The growing embryonic fin bud depends on bone morphogenetic protein 4 (BMP4) signalling, the vertebrate orthologue of *Drosophila* Dpp. The *smoc1* phenotype in the fin appeared to relate to a defect in scaling the BMP4 gradient, consistent with the mechanism in *Drosophila*. Mutants were viable and macroscopically normal (Figure 5.9 C). Interestingly, the mutant fish retained subtle differences in eye scaling throughout life (Figure 5.9 D). These findings were confirmed by independent measurements from Dr Rita Mateus [personal communication].

Morphogenesis of the embryonic fish eye also depends on BMP4 [Heermann et al., 2015]. I have begun characterising the eye phenotype of the *smoc1* mutant in collaboration with Dr Rita Mateus. In wildtype fish, antibody stainings revealed that Smoc1 protein was absent from the eye up to 16 hpf, and by 18 hpf it appeared in the surface ectoderm (Figure 5.10 A). As the lens invaginated, Smoc1 protein moved into the interior of the eye at the interface of lens and neuroepithelium (Figure 5.10 A). At later stages, Smoc1 protein became undetectable except for small remnants surrounding the lens. The period of maximum Smoc1 protein intensity correlated with the timing of "eye gastrulation", *i.e.* the coordinate movement of neuroepithelial cells to transform the optic vesicle into the optic cup [Heermann et al., 2015; Kwan et al., 2012]. These stainings were consistent with reported expression patterns for *smoc1* mRNA [Weekes,

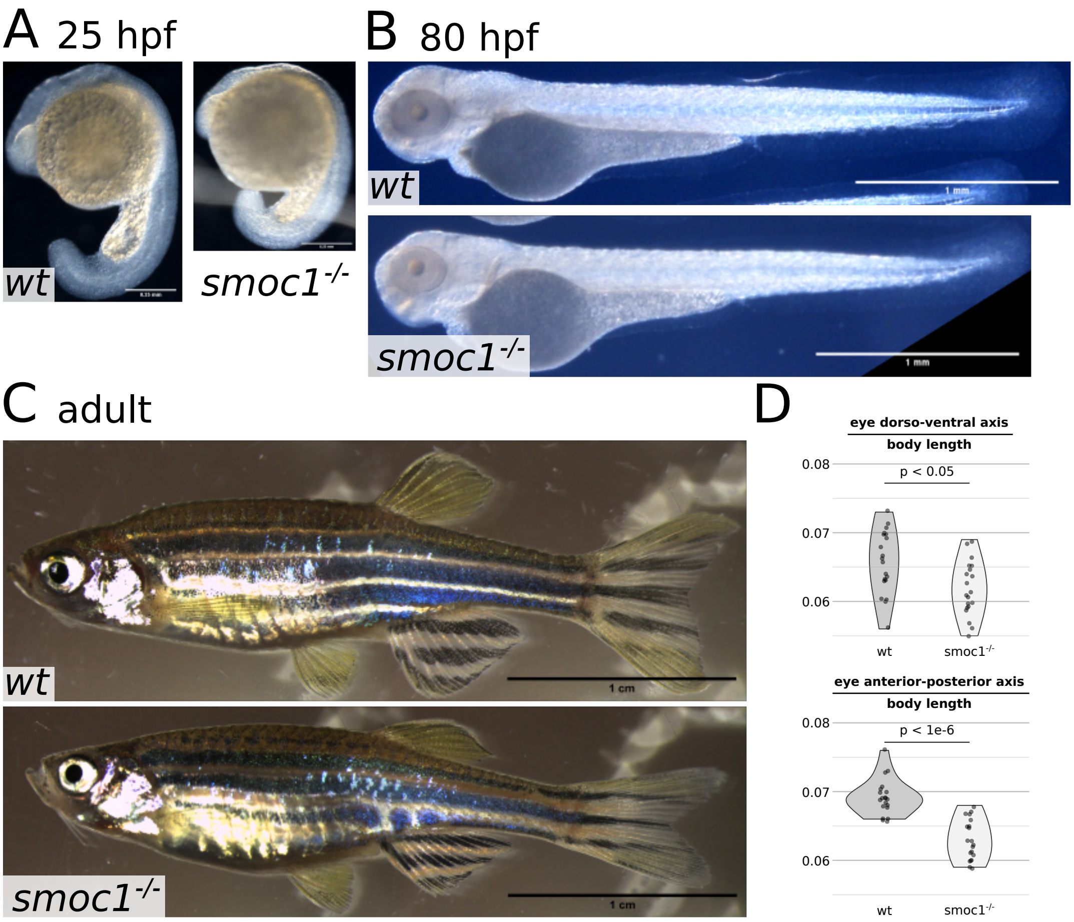


Figure 5.9: *Smoc1* mutants have slightly shorter body axis and smaller eyes.
A Development appeared slightly delayed in *smoc1* mutants. Image depicts fixed embryos. **B** By hatching, the developmental delay was overcome but the mutants remained slightly smaller. Image depicts fixed larvae that had been reared in PTU. **C** Adult mutants were viable and macroscopically normal. **D** Eye dimensions relative to body length were slightly affected in *smoc1* mutants. p-values calculated by Wilcoxon rank sum test.

2015]. Interestingly, *smoc1* mutants retained antibody staining at lower levels with no sign of mislocalisation, indicating that the mutant had a hypomorphic allele rather than a null allele (Figure 5.10 B). Together, these data supported an exclusive role of *smoc1* in early eye morphogenesis; in the mutant, the eye failed to form its proper initial geometry leading to a scaling defect that was not corrected afterwards.

To investigate the cellular origin of the *smoc1* phenotype, I performed parallel time-lapse movies of age-matched wildtype and *smoc1* mutant embryos in a bone morphogenetic protein (BMP) reporter background [Collery and Link, 2011]. The BMP reporter background was chosen based on a potential interaction of *smoc1* and BMP in homology to *Drosophila* Dpp and Pentagone.

Parallel time-lapse acquisition started when the embryos were roughly 12 hpf, *i.e.* when the optic vesicle first becomes macroscopically visible [Kimmel et al.,

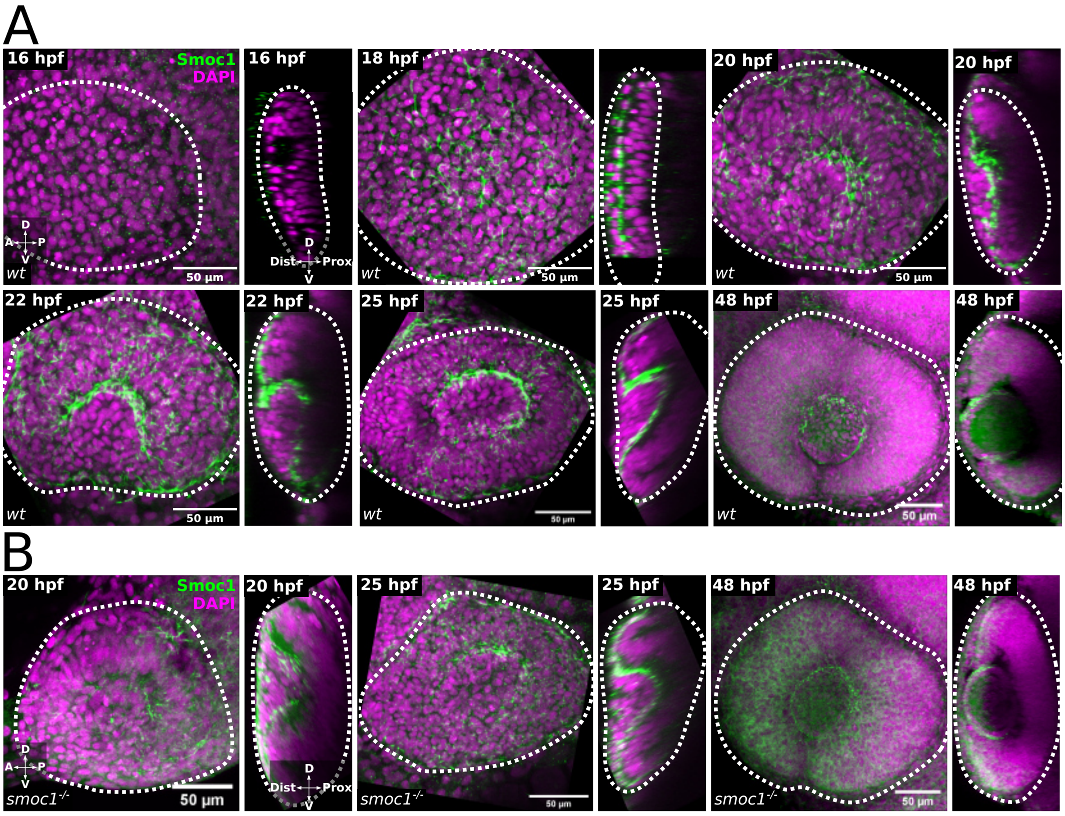


Figure 5.10: *Smoc1* protein localises to the interface of lens and neuroectoderm during optic cup formation.

A Immunohistochemical staining for Smoc1 protein and cell nuclei in wildtype embryos at different developmental stages. **B** Immunohistochemical staining for Smoc1 protein and cell nuclei in *smoc1* mutant embryos. Smoc1 protein is still detectable, albeit at lower levels. Left panels show maximum projection of distal view, right panels show orthogonal view of optical sections going through lens primordium. White dotted line: Outline of embryonic eye. Abbreviations: D –Dorsal, V –ventral, A –Anterior, P –posterior, Dist –distal, Prox –proximal.

1995]. In *smoc1* mutants, the optic vesicle was difficult to discern –indeed confocal images revealed it was located more proximally at this stage, indicative of a slight developmental delay (Figure 5.11 A–B, 12 hpf). Nevertheless, other eye developmental milestones such as onset of BMP expression (≈18 hpf), onset of lens invagination (≈20 hpf), and completion of optic cup formation (≈26 hpf) occurred at the same time as in wildtype embryos (Figure 5.11 A–B). Expression levels of the BMP reporter tended to be lower in *smoc1* mutant embryos, but this was not consistent; the pattern of expression domains was comparable to wildtype.

These data showed that the timing of morphogenetic events after optic vesicle formation was not impaired. The *smoc1* mutant phenotype may be caused by more subtle changes in cell motility, as *smoc1* promoted cell motility in *in vitro* scratch wound assays [Maier, 2006]. Lower cell motility in the hypomorphic context may have reduced the initial number of cells contributing to the op-

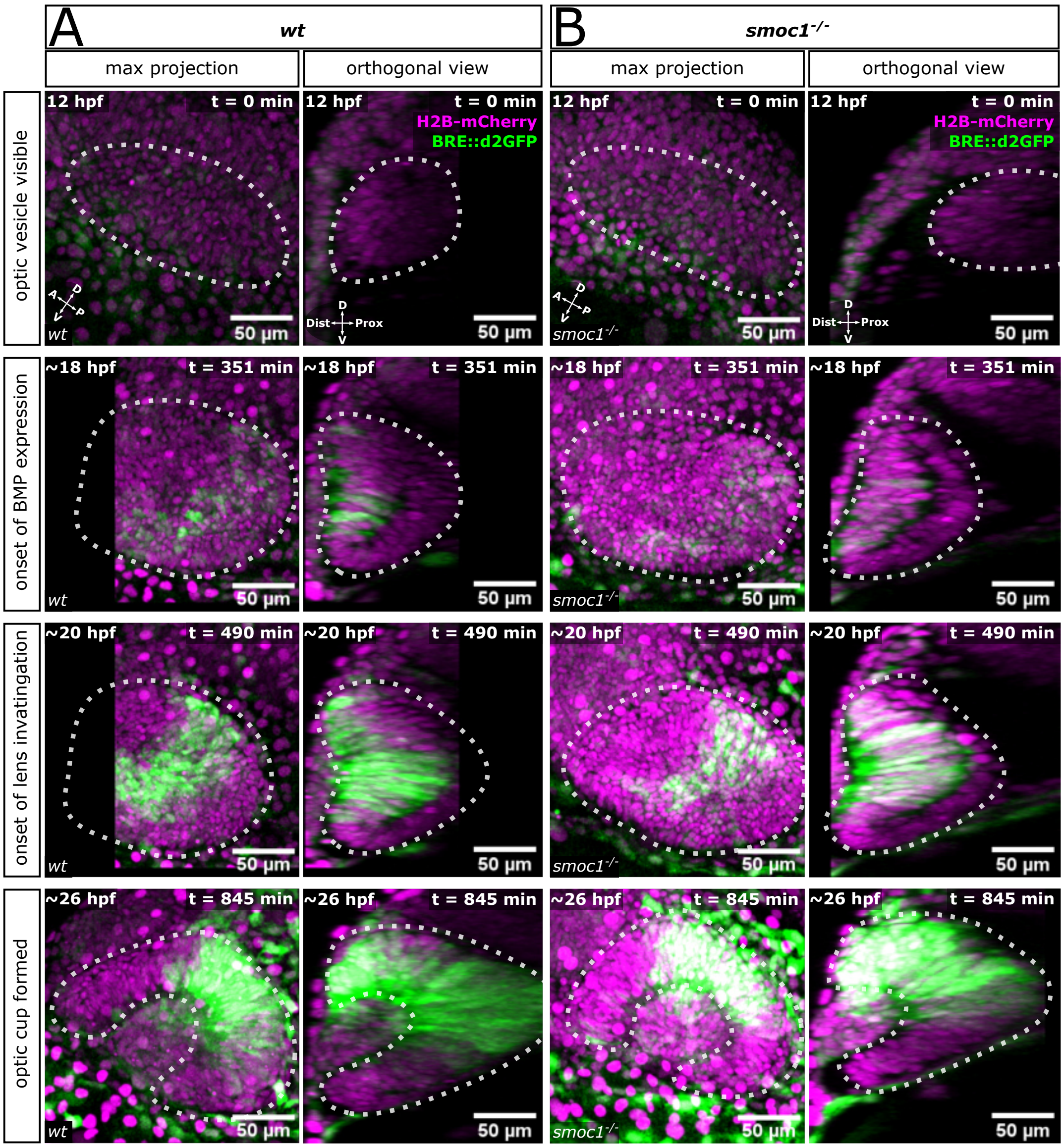


Figure 5.11: Frames from a parallel time-lapse movie of developing wildtype and *smoc1* mutant embryos.

A Frames from a time-lapse movie of a wildtype embryo. **B** Frames from a time-lapse movie of a *smoc1* mutant embryo. Wildtype and mutant embryos were crossed into reporter lines expressing a ubiquitous nuclear mCherry and destabilised GFP under the control of a BMP-responsive element [Collery and Link, 2011]. Orthogonal views are maximum projections of the central area of the optic vesicle or cup, roughly corresponding to the position of the lens primordium. White dotted line surrounds the neuroepithelium.

tic vesicle, resulting in a slight delay in evagination (Figure 5.11 B 12 hpf), which ultimately impacted on the shape of the optic cup. This altered "initial condition" in retinal geometry could explain why subtle scaling differences persisted throughout the life of the fish (Figure 5.9 D). Additional stainings for Smoc1 protein at early stages of optic vesicle evagination and cell motility analysis in the time-lapse movies could elucidate these points.

5.7 Code

5.7.1 ImageJ macro scripts

Create a focused stack

The macro assumes the stack was ordered such that in the first slice (image frame) the base of the hemisphere was in focus, and in the final slice the curved side of the hemisphere was in focus. The user needs to provide the value for h, and a circular selection (manually made in ImageJ using the "Oval Tool") encompassing the entire hemispherical sample in the first slice of the confocal stack. The confocal stack should be trimmed to remove out-of-focus slices at the beginning and end of the stack. The macro outputs a collated focused image.

Source Code 5.1: ImageJ macro for creating a focused stack.

```
// Required: First ROI in manager is retinal outline.
if (roiManager("Count")<1) { exit("ROI in ROI Manager required"); }
// z-step used for image acquisition in micrometers
h = 12.5;

// Measure size of initial circular selection
run("Clear Results");
run("Set Measurements...", "bounding fit redirect=None decimal=3");
run("Measure");
// Create a duplicate of the entire image to work on
W = getWidth(); H = getHeight(); makeRectangle(0,0, W, H);
run("Duplicate...", "duplicate");
// Get dimensions of circular selection from the results table
// Remove scaling as it affects the calculation
X  = getResult("X"); Y = getResult("Y"); toUnscaled(X, Y);
BX = getResult("BX"); BY = getResult("BY"); toUnscaled(BX, BY);
BW = getResult("Width"); BH = getResult("Height"); toUnscaled(BW); toUnscaled(BH);
D = getResult("Major"); toUnscaled(D); R = D/2;
// Get image properties and split channels
slices = nSlices;
title = getTitle(); run("Split Channels");
string1 = "C1-" + title; string2 = "C2-" + title; string3 = "C3-" + title;
array  = newArray(string1, string2, string3);

// Loop through each channel
for (channel = 0; channel <=2; channel++) {
    image = array[channel];     selectWindow(image);
    i = 1;
    // Loop through slices, starting with last
    for (slice = nSlices; slice > 0; slice--) {
        setSlice(slice);
        // Remove inner cap, starting from second-to-last slice
        if (slice < nSlices) {
            makeOval(BXr, BYr, 2*r , 2*r); run("Cut");
        }
        r = sqrt(h*i*(2*R-h*i)); // Get inner radius. Formula: radius of base of hemispherical cap
        BXr = BX + (R - r);        BYr = BY + (R - r);
        makeOval(BXr, BYr, 2*r , 2*r); // Create an oval around the bigger annulus
        run("Make Inverse"); run("Cut"); run("Make Inverse");        // Cut out everything outside the oval
        i++; // Proceed with next slice in stack
    }
    // Collate all focused parts together using a sum projection
    makeRectangle(0,0, W, H);
    run("Invert", "stack"); temp = getImageID();
    run("Z Project...", "projection=[Sum Slices]");
    array[channel] = getTitle();
    run("Invert");
    selectImage(temp); close();
}
string = "c1=" + array[0] + " c2=" + array[1] + " c3=" + array[2] + " create";
run("Merge Channels...", string);
selectWindow("Composite"); rename(title);
```

Post-processing segmented retinal images

The user needs to define absolute folder path for root folder (`inputfolder`), the subfolders containing source images (`imagefolder`) and input ROIs (`roifolder`), and the absolute path to the output folder where images will be saved (`outputfolder`). As input, the macros need a set of two ROIs files demarcating the retinal edge and the pre-induction retina for each image in this order. A set of two ROIs files demarcating the retinal edge and the pre-induction retina for each image were drawn manually using the "Oval Tool" in ImageJ, added to the "ROI manager", and saved as a separate file with identical filename to the corresponding image.

Source Code 5.2: ImageJ macro for smoothing segmented patches of the NR.

```
// Variables requiring user modification
inputfolder  = "C:\\inputfolder\\";
imagefolder  = "imagefolder\\";
roifolder    = "input_roi\\";
outputfolder = "C:\\outputfolder\\"

// Required: First ROI in manager is retinal outline, second ROI is pre-induction retina.
if (roiManager("Count")<2) { exit("Two ROIs in ROI Manager required"); }

function smoothNR() {
    run("Select None");
    roiManager("Open", roifile);
    roiManager("Select", 0);
    roiManager("Select", 1);
    setBackgroundColor(0,0,0);
    setForegroundColor(255,255,255);
    run("Cut");
    run("Select None");
}

```

```
list = getFileList(inputfolder + imagefolder);
numprocess = list.length;
setBatchMode(true);

for (I = 0; I < numprocess; I++) {
    open(inputfolder + imagefolder + list[I]);
    filename = getInfo("image.filename");
        filepath = getInfo("image.directory") + filename;
        roifile  = inputfolder + roifolder + filename + "_RoiSet.zip";
        smoothNR();
        string = outputfolder + filename;
        saveAs("Tiff", string);
        run("Close All");
        roiManager("Reset")
        run("Collect Garbage");
}
```

Source Code 5.3: ImageJ macro for smoothing segmented patches of the RPE.

```
// Variables requiring user modification
inputfolder  = "C:\\inputfolder\\";
imagefolder  = "imagefolder\\";
roifolder    = "input_roi\\";
outputfolder = "C:\\outputfolder\\"

// Required: First ROI in manager is retinal outline, second ROI is pre-induction retina.
if (roiManager("Count")<2) { exit("Two ROIs in ROI Manager required"); }

function smoothRPE() {
        run("Select None");
        run("Median...", "radius=2");
        run("Shape Smoothing", "relative_proportion_fds=6 absolute_number_fds=2 keep=[Relative_proportion of FDs] black");
        run("Select None");
        roiManager("Open", roifile);
        roiManager("Select", 0);
        roiManager("Select", 1);
        run("Set...", "value=0");
        run("Select None");
        run("Median...", "radius=2");
}

list = getFileList(inputfolder + imagefolder);
numprocess = list.length;
setBatchMode(true);

for (I = 0; I < numprocess; I++) {
    open(inputfolder + imagefolder + list[I]);
    filename = getInfo("image.filename");
        filepath = getInfo("image.directory") + filename;
        roifile  = inputfolder + roifolder + filename + "_RoiSet.zip";
        smoothRPE();
        string = outputfolder + filename;
        saveAs("Tiff", string);
        run("Close All");
        roiManager("Reset")
        run("Collect Garbage");
}
```

Source Code 5.4: ImageJ macro for smoothing simulated segmentation.

```
// Variables requiring user modification
inputfolder  = "C:\\inputfolder\\";
imagefolder  = "imagefolder\\";
roifolder    = "input_roi\\";
outputfolder = "C:\\outputfolder\\"

// Required: First ROI in manager is retinal outline, second ROI is pre-induction retina.
if (roiManager("Count")<2) { exit("Two ROIs in ROI Manager required"); }

function smoothsim() {
	run("Invert LUT");
	run("Median...", "radius=15");
	run("Erode");run("Erode");run("Erode");run("Erode");run("Erode");
	run("Dilate");run("Dilate");run("Dilate");run("Dilate");run("Dilate");
	run("Shape Smoothing", "relative_proportion_fds=6 absolute_number_fds=2 keep=[Relative_proportion of FDs] black");
	roiManager("Open", roifile);
	run("Select None");
	run("Median...", "radius=2"); run("Median...", "radius=2");
	roiManager("Select", 1);
	run("Cut");
	run("Select None");
}

list = getFileList(inputfolder + imagefolder);
numprocess = list.length;
setBatchMode(true);

for (l = 0; l < numprocess; l++) {
  open(inputfolder + imagefolder + list[l]);
  filename = getInfo("image.filename");
	filepath = getInfo("image.directory") + filename;
	roifile  = inputfolder + roifolder + filename + "_RoiSet.zip";
	smoothsim();
	string = outputfolder + filename;
	saveAs("Tiff", string);
	run("Close All");
	roiManager("Reset")
	run("Collect Garbage");
}
```

Subdivision of the retina into several ROIs

Source Code 5.5: ImageJ macro for creating several ROI compartmentalisations on the retina. This macro requires the same user input as 5.7.1.

```
// Variables requiring user modification
inputfolder  = "C:\\inputfolder\\";
imagefolder  = "imagefolder\\";
roifolder    = "input_roi\\";
outputfolder = "C:\\outputfolder\\"

// ROI subdivisions
numrings = 19;
```

```
numsecs  = 10;

// Required: First ROI in manager is retinal outline, second ROI is pre-induction retina.
if (roiManager("Count")<2) { exit("Two ROIs in ROI Manager required"); }

// Measures the post-induction length of the retina by constructing four line ROIs
function makeDVRL() {
    totalROIs = roiManager("count")-1;
    roiManager("Select", 0); //  retinal outline
    Roi.getBounds(x, y, width, height);
    // Get poles of retinal outline
    D_x = x + width/2;      D_y = y;                        // Dorsal
    V_x = D_x;      V_y = y + height;                       // Ventral
    R_x = x + width;        R_y = y + height/2;     // Right
    L_x = x;        L_y = R_y;                                      // Left
    // Make lines from the pre-induction retina to the poles
    roiManager("Select", 1); //pre-induction retina
    final_y = 0;
    for (i = D_y; i <= D_y + height; i++) {
        if (selectionContains(D_x, i)) {
            final_y = i;
            i = D_y + height + 1;
        }
    }
    // Dorsal line
    makeLine(D_x, D_y, D_x, final_y);
    roiManager("Add");
    roiManager("Select", totalROIs+1);
    roiManager("Rename", "d");
    run("Clear Results");
    run("Measure");
    D = getResult("Length");
    roiManager("Select", 1); //pre-induction retina
    final_y = 0;
    for (i = V_y; i >= V_y - height; i--) {
        if (selectionContains(V_x, i)) {
            final_y = i;
            i = V_y - height - 1;
        }
    }
    // Ventral line
    makeLine(V_x, V_y, V_x, final_y);
    roiManager("Add");
    roiManager("Select", totalROIs+2);
    roiManager("Rename", "v");
    run("Clear Results");
    run("Measure");
    V = getResult("Length");
    roiManager("Select", 1); //pre-induction retina
    final_x = 0;
    for (i = R_x; i >= R_x - width; i--) {
        if (selectionContains(i, R_y)) {
            final_x = i;
            i = R_x - width - 1;
        }
    }
    // Right line
    makeLine(R_x, R_y, final_x, R_y);
    roiManager("Add");
    roiManager("Select", totalROIs+3);
    roiManager("Rename", "r");
    run("Clear Results");
    run("Measure");
    R = getResult("Length");
    // Left line
    roiManager("Select", 1); //pre-induction retina
    final_x = 0;
    for (i = L_x; i <= L_x + width; i++) {
        if (selectionContains(i, L_y)) {
            final_x = i;
            i = L_x + width + 1;
        }
    }
    makeLine(L_x, L_y, final_x, L_y);
    roiManager("Add");
    roiManager("Select", totalROIs+4);
    roiManager("Rename", "l");
    run("Clear Results");
    run("Measure");
    L = getResult("Length");
    // Return array with length of lines
    ary = newArray(D, V, R, L);
    return ary;
}

// Constructs concentric annular ROIs based on retinal outline and pre-induction retina ROIs
function makeConcentric(ringnum) {
    divs  = ringnum + 1;
    // retina ROI
    run("Clear Results");
    roiManager("Select", 0);
    run("Measure");
    rX  = getResult("XM");     rY = getResult("YM");          toUnscaled(rX, rY);
    rBX = getResult("BX");     rBY = getResult("BY");         toUnscaled(rBX, rBY);
    rBW = getResult("Width"); rBH = getResult("Height"); toUnscaled(rBW); toUnscaled(rBH);
    // pre-induction retina ROI
    run("Clear Results");
    roiManager("Select", 1);
    run("Measure");
    pirX  = getResult("XM");     pirY = getResult("YM");          toUnscaled(pirX, pirY);
    pirBX = getResult("BX");     pirBY = getResult("BY");         toUnscaled(pirBX, pirBY);
    pirBW = getResult("Width"); pirBH = getResult("Height"); toUnscaled(pirBW); toUnscaled(pirBH);
    // lower left corner
    rBllY   =   rBY +   rBH;
    pirBllY = pirBY + pirBH;
    // upper right corner
    rBurX   =   rBX +   rBW;
    pirBurX = pirBX + pirBW;
    // Repeat ringnum times
    for (i = 1; i <= ringnum; i++) {
        newBX = pirBX - i*(pirBX - rBX)/divs;
        newBY = pirBY - i*(pirBY - rBY)/divs;
        newBllY = pirBllY - i*(pirBllY - rBllY)/divs;
        newBurX = pirBurX - i*(pirBurX - rBurX)/divs;
        newW  = newBllY - newBY;
        newH  = newBurX - newBX;
         makeOval(newBX, newBY, newH , newW);
        Roi.setStrokeColor("ff993333");
         roiManager("Add");
     }
    roiManager("Show All");
    toScaled(rBH); toScaled(rBW); toScaled(rX, rY); toScaled(pirBH); toScaled(pirBW); toScaled(pirX, pirY);
    ary = newArray(rBH, rBW, rX, rY, pirBH, pirBW, pirX, pirY);
    return ary;
}

// Constructs ROIs that subdivide the retina into angular sectors ("pie slices")
function makeSectors(retina, secnum) {
    totalROIs = roiManager("count")-1;
    roiManager("Select", 0); // retinal outline
    Roi.getBounds(x, y, width, height);
    centerX = x + width/2; centerY = y + height/2;
    if (retina == false) { // use
        roiManager("Select", 1); //pre-induction retina
        Roi.getBounds(x, y, w, h);
        centerX = x + w/2; centerY = y + h/2;
    }
    r = 0; if (height >= width) { r = 1.1*height; } else { r = 1.1*width; }; r /= 2;
    sectors = secnum*4; delta = (360*PI)/(sectors*180);
    start = -0.75*PI+delta;
    j = 0;
    k = 1;
    for (i = start; i < 2*PI+start; i+=delta) {
        if (j < sectors ) {
            oldX = r*cos(i-delta);
            oldY = r*sin(i-delta);
            newX = r*cos(i);
            newY = r*sin(i);
            makePolygon(centerX+oldX, centerY+oldY, centerX, centerY, centerX+newX, centerY+newY);
            roiManager("Add");
            roiManager("Select", roiManager("Count")-1);
            roiManager("Rename", "sector " + toString(j+1));
            if (j%(secnum/4) == 0) { \\ use modulo operator to figure out retinal quadrant
                k += 1;
            }
            Roi.setStrokeColor("ff003300");
            Roi.setFillColor(toString(11*k) + toString(11*k) + "cc" + toString(11*k));
        }
        j += 1;
    }
}
// utility function to XOR concentric circles into an annulus
function makeRing(idx1, idx2, ringnum) {
    roiManager("Select", newArray(idx1, idx2));
    roiManager("XOR");
    roiManager("Add");
    roiManager("Select", roiManager("Count")-1);
    roiManager("Rename", "ring " + toString(ringnum));
}
// utility function to loop makeRing
function makeAllRings(first, last) {
    j = 1;
    makeRing(1, first, j); // pir - first concentric ring
    for (i = first; i < last; i++) {
        j += 1;
        makeRing(i, i+1, j);
    }
    makeRing(last, 0, j+1); // last concentric ring - retina
}
// utility function to group sectors to quadrants
function makeQuads(numsecs) {
    first = getFirstROIbyName("sector");
    firstquadidx = roiManager("Count");
    for (i = 1; i < 5; i++) {
        start = first + (i-1);
        a = Array.getSequence(numsecs);
        for (j = 0; j < numsecs; j++) {
            a[j] = a[j] + start + (numsecs-1)*(i-1);
        }
        roiManager("Select", a);
        roiManager("Combine");
        roiManager("Add");
        roiManager("Select", roiManager("Count")-1);
        roiManager("Rename", "quadrant " + toString(i));
    }
}
// utility function to find a ROI's index in the ROI manager based on the ROI name
function getFirstROIbyName(ROIname) {
    for (i = 2; i <= (roiManager("Count")); i++) {
        name = call("ij.plugin.frame.RoiManager.getName", i);
        if (name == ROIname + " 1") {
            firstROIidx = i;
            i = roiManager("Count") + 999;
        }
    }
    return(firstROIidx);
}

// Function execution
list = getFileList(inputfolder + imagefolder);
numprocess = list.length;
setBatchMode(true);
for (l = 0; l < numprocess; l++) {
  open(inputfolder + imagefolder + list[l]);
  filename = getInfo("image.filename");
    filepath = getInfo("image.directory") + filename;
    roifile  = inputfolder + roifolder + filename + "_RoiSet.zip";
    roiManager("Open", roifile);
    run("Select None");
    // Make ROIs and get various measurements
    gen_DVRL  = makeDVRL();
    gen_sizes = makeConcentric(numrings);
    roinum    = roiManager("Count");
    makeAllRings(roinum - numrings, roinum - 1);
    roinum    = roiManager("Count");
    makeSectors(true, numsecs);
    makeQuads(numsecs);
    roiManager("Save", inputfolder + roifolder + filename + "_RoiSet.zip");
    run("Close All");
    roiManager("Reset")
    run("Collect Garbage");
}
```

Radially normalised retinal projections

Source Code 5.6: ImageJ macro for creating a radially normalised Cartesian coordinate projection of the retina. It requires the same user input as 5.7.1.

```
// Variables requiring user modification
inputfolder  = "C:\\inputfolder\\";
imagefolder  = "imagefolder\\";
roifolder    = "input_roi\\";
outputfolder = "C:\\outputfolder\\"

normRetSavePath = inputfolder + roifolder;

function closeROImanager() { // utility function that speeds up things
    winnames = getList("window.titles");
  for ( win = 0 ; win < winnames.length ; win++ ){
      winame = winnames[win];
      if ( winame == "ROI Manager" ) {
          selectWindow(winame);
          run("Close");
      }
  }
}
// Creates a Cartesian projection of the post-induction retina
function normRetinaOutline(savepath) {
    ori = getImageID();
    roiManager("Open", roifile);
    // Create duplicate to subtract pre-induction retina from full retina
    run("Select None");
    run("Duplicate...", " ");
    dup = getImageID();
    selectImage(dup);
    // Use the first pre-defined retina ROI to fill in white [=255]
    roiManager("Select", 0);
    setBackgroundColor(255,255,255);
    setForegroundColor(0,0,0);
    run("Cut");
    run("Select None");
    // Use the second pre-defined retina ROI to subtract pre-induction retina --> black circle [=0]
    roiManager("Select", 1);
    setBackgroundColor(0,0,0);
    setForegroundColor(255,255,255);
    run("Cut");
    run("Select None");
    // Use pre-induction retina to get center coordinates for transform
    run("Clear Results"); roiManager("Select", 1);
    run("Measure"); run("Select None");
    xcenter = getResult("X"); ycenter = getResult("Y");
    h = getHeight(); w = getWidth();
    toUnscaled(xcenter, ycenter);
    // Transform using Polar Transformer plugin
    run("Polar Transformer", "method=Polar degrees=360 number=3600 center_x="+xcenter+" center_y="+ycenter+"
    ↪  interpolation=Bilinear");
    run("Rotate 90 Degrees Right");
    pol = getImageID();
    // Get ROI of selection pre- and post-trim
    closeROImanager();
    setThreshold(1, 255);
    run("Create Selection");
    roiManager("Add"); // Selection with index 0 --> uncropped
    run("Crop");
    roiManager("Add"); // Selection with index 1 --> cropped
    // Save ROIs
    roiManager("Save", savepath + filename + "_NormRetinaRois.zip");
    // Close unneeded windows
    selectImage(dup); run("Close");
    selectImage(pol); run("Close");
    selectImage(ori); // make active image
}

function normPatchRadius(savepath) {
    ori = getImageID();
    closeROImanager();
    roiManager("Open", roifile);
    // Subtract pre-induction retina --> black circle [=0]
    selectImage(ori);
    roiManager("Select", 1);
    setBackgroundColor(0,0,0);
    setForegroundColor(255,255,255);
    run("Cut");
    run("Select None");
    // Use pre-induction retina to get center coordinates for transform
    run("Clear Results"); roiManager("Select", 1);
    run("Measure"); run("Select None");
    xcenter = getResult("X"); ycenter = getResult("Y");
    h = getHeight(); w = getWidth();
    toUnscaled(xcenter, ycenter);
    // Transform using polar transformer plugin
    run("Polar Transformer", "method=Polar degrees=360 number=3600 center_x="+xcenter+" center_y="+ycenter+"
    ↪  interpolation=Bilinear");
    run("Rotate 90 Degrees Right");
    pol = getImageID();
    // Get normalized ROIs
    closeROImanager();
    roiManager("Open", savepath + filename + "_NormRetinaRois.zip");
    // Crop
    selectImage(pol);
    roiManager("Select", 0);
    run("Crop");
    // Close original image
    selectImage(ori); run("Close");
    selectImage(pol); // make active image
    w = getWidth();
    h = getHeight();
    // Create retina outline reference
    newImage("ret", "8-bit black", w, h, 1);
    ret = getImageID();
    roiManager("Select", 1); // unwrapped retina roi after cropping
    setForegroundColor(0,0,0);
    setBackgroundColor(255,255,255);
    run("Cut");
    // Create result image
    newImage("result", "8-bit black", w, h, 1);
    res = getImageID();
    // Loop through X coordinate
    setBatchMode(true);
    for ( x = 0; x < w; x++) {
        yend = -99;
        ysta = h+1;
        // Find lowest pixel in unwrapped retina ROI at given X coordinate (inverse y loop)
        selectImage(ret);
        for ( y = h; y > 0; y-- ) {
            v = getPixel(x, y);
            if ( v == 255 ) {
                yend = y;
                y = -99; // break loop
            }
        }
        // Find highest pixel in unwrapped retina ROI at given X coordinate (y loop)
        selectImage(ret);
        for ( y = 0; y < h; y++ ) {
            v = getPixel(x, y);
            if ( v == 255 ) {
                ysta = y;
                y = h+1; // break loop
            }
        }
        // Now back to original image --> interpolate 1-px wide strips
        selectImage(pol);
        makeRectangle(x, ysta, 1, yend-ysta); // make a 1-px wide strip at x position
        run("Duplicate...", " "); // isolate the strip
        run("Size...", "width=1 height="+h+" interpolation=None"); // stretch it
        run("Select All");
        run("Copy");
        run("Close");
        selectImage(res);
        makeRectangle(x, 0, 1, h);
        run("Paste"); // paste the stretched bit into the result image
    }
    selectImage(ret); run("Close");
    selectImage(res);
    run("Median...", "radius=2");
    setBatchMode("exit and display");
    // Save and close all
    selectImage(res);
    saveAs("Tiff", outputfolder + filename + "_normalized.tif");
    run("Close All");
    closeROImanager();
}

// Execute functions
list = getFileList(inputfolder + imagefolder);
print(inputfolder);
numprocess = list.length;
```

```
for (l = 0; l < numprocess; l++) {
        setBatchMode(false);
    open(inputfolder + imagefolder  + list[l]);
    filename = getInfo("image.filename");
        roifile = inputfolder + roifolder + filename + "_RoiSet.zip";
        normRetinaOutline(normRetSavePath); // Create an outline of the entire post-induction retina to know how much
        ↪   stretching is needed at every x position
        normPatchRadius(normRetSavePath); // Apply the stretch to the projected image
        run("Close All");
        closeROImanager();
}
```

Patch superposition

The user needs to define absolute folder path for input folder, which is the same folder where images will be saved (`inputfolder`). The input image should be the output of 5.6.

Source Code 5.7: ImageJ macro for creating normalised patch superposition.

```
// Variable requiring user modification
inputfolder  = "C:\\inputfolder\\";

function closeROImanager() { // utility function
        winnames = getList("window.titles");
    for ( win = 0 ; win < winnames.length ; win++ ){
            winame = winnames[win];
            if ( winame == "ROI Manager" ) {
                    selectWindow(winame);
                    run("Close");
            }
    }
}
function getPatchROI(loopiter) { // Wrapper function
        pol = getImageID();
        setThreshold(1, 255); run("Convert to Mask");
        wwrap = getWidth();
        correction = correctSplitBlobs(wwrap); setBatchMode("show"); // correct for patches that wrap around the edge
        run("Shape Smoothing", "relative_proportion_fds=5 absolute_number_fds=2 keep=[Relative_proportion of FDs] black");
        run("Find Connected Regions", "allow_diagonal display_one_image regions_for_values_over=100
        ↪   minimum_number_of_points=1 stop_after=-1");
        run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");
        closeROImanager();
        if (loopiter > 0) {
                roiManager("Open", inputfolder + "_PatchRois.zip");
        }
        polarcc = getImageID();
        wmax = getCCROIs(); // get ROIs of connected regions
        roiManager("Save", inputfolder + "_PatchRois.zip");
        ary = newArray(wmax, getHeight(), getWidth());
        return ary;
}
// get the patch superposition
function patchDensity(wmax) {
        dens = getImageID(); denstit = getTitle(); setBatchMode("show"); // create result image
    numcc = roiManager("Count");
    for (roi = 0; roi < numcc; roi++) { // loop through patch ROIs
        run("Select None");
        newImage("temp", "32-bit black", wmax, getHeight(), 1);
        temp = getImageID(); temptit = getTitle();
        roiManager("Select", roi); run("Set...", "value=1"); run("Select None");
        imageCalculator("Add 32-bit", denstit, temptit);
        selectImage(temp); run("Close");
    }
    selectImage(dens);
    run("Select None");
    getStatistics(area, mean, min, max);
    run("Divide...", "value=" + max);
    setMinAndMax(0, 1);
    run("Fire");
    wait(1000); // race condition!
}
// loop through connected components by thresholding, save their ROIs to the ROI manager
function getCCROIs() {
    polori = getImageID();
    getStatistics(area, mean, min, max);
    wmax = -1;
    for (i = 1; i <= max; i++) {
        // get connected component selection and save
        selectImage(polori);
        setThreshold(i, i); run("Create Selection");
        roiManager("Add");
        run("Clear Results");
        run("Set Measurements...", "area mean min centroid bounding fit feret's redirect=None decimal=3");
        roiManager("Select", roiManager("Count")-1);
        getSelectionBounds(curx, cury, roiw, roih);
        run("Measure");
        if (wmax < getResult("Width")) {
            wmax = getResult("Width");
        }
    }
    toUnscaled(wmax);
    return wmax;
}
// align all ROIs by translation such that ther central x-coordinate is the same; don't align by y!
function alignCCROIs(wmax) {
    numcc = roiManager("Count");
    for (roi = 0; roi < numcc; roi++) {
        roiManager("Select", roi);
        getSelectionBounds(curx, cury, roiw, roih);
        xnew = wmax/2 - roiw/2;
        setSelectionLocation(xnew, cury);
        roiManager("Update");
    }
    roiManager("Save", inputfolder + "RoiSetAligned.zip");
}
// correct for patches that wrap around the edge by expanding canvas and cut/pasting them over
function correctSplitBlobs(wwrap) {
    correction = false;
    blobFirst  = -1;
    blobLast   = -1;
    for (y = 0; y <= getHeight(); y++) { // loop through y coordinate at x = 0
```

```
        v = getPixel(0, y);
        if (v > 0) { // potentially split patch
            correction = true;
            if (blobFirst == -1) {
                doWand(0, y); roiManager("Add");
                blobFirst = roiManager("Count")-1;
            }
            else { // already found a patch previously
                doWand(0, y); roiManager("Add");
                blobLast  = roiManager("Count")-1;
                roiManager("Select", newArray(blobFirst, blobLast));
                roiManager("Combine"); roiManager("Add");
                blobFirst = roiManager("Count")-1;
                roiManager("Select", blobFirst-1); roiManager("Delete"); run("Select None");
                blobFirst = roiManager("Count")-1;
            }
        }
        if (blobFirst > -1) { // if split patches were found
            blobFirst = (roiManager("Count")-1);
            run("Clear Results"); roiManager("Select", blobFirst);
            run("Measure");
            setForegroundColor(255,255,255); setBackgroundColor(0,0,0);
            roiManager("Select", 0); run("Set...", "value=0");
            run("Select None");
            blobw = getResult("Width"); toUnscaled(blobw);
            // enlarge canvas
            newWidth  = wwrap + blobw;
            newHeight = getHeight();
            run("Canvas Size...", "width="+ newWidth +" height="+newHeight+" position=Center-Left");
            // translate selection
            roiManager("Select", blobFirst); roiManager("translate", wwrap, 0);
            run("Select None"); roiManager("Select", blobFirst);
            run("Set...", "value="+255);
            roiManager("Select", blobFirst); roiManager("Delete"); run("Select None");
            blobFirst = -1;
        }
    }
    return correction;
}

// Initialize variables
polwmax = -1;
wmaxmax = -1;
hmax    = -1;
polwmax = -1;
wmaxmax = -1;
hmax    = -1;
list = getFileList(inputfolder);
numprocess = list.length;
// Execute functions
for (l = 0; l < numprocess; l++) {
    setBatchMode(false);
    open(inputfolder  + list[l]);
    filename = getInfo("image.filename");
    filepath = getInfo("image.directory") + filename;
    run("Select None");
    ary = getPatchROI(l);
    if (wmaxmax < ary[0]) { wmaxmax = ary[0]; }
    if (hmax    < ary[1])    { hmax = ary[1]; }
    if (polwmax < ary[2]) { polwmax = ary[2]; }
    run("Close All");
    closeROImanager();
}
roiManager("Open", inputfolder + "_PatchRois.zip");
setBatchMode(true);
newImage("PatchDensity", "32-bit black", polwmax, hmax, 1); setBatchMode("show");
alignCCROIs(wmaxmax); // align ROIs
makeRectangle(0, 0, wmaxmax, getHeight());
run("Crop");
patchDensity(wmaxmax); // get the patch density by superposition
saveAs("Tiff", inputfolder +"PatchDensity_norm.tif");
closeROImanager();
wait(500); // race condition!
run("Close All");
```

Output properties of a list of ROIs

This macro takes as input an image of patch superposition and the associated list of ROIs in the ROI Manager (generated by Appendix Scripts 5.7). It loops through all ROIs currently in the ROI Manager to print their dimensions and coordinates to the ImageJ Log. The Log file has to be manually saved.

Source Code 5.8: ImageJ macro for printing ROI properties.

```
w = getWidth();
h = getHeight();
print(w, h);
numcc = roiManager("Count");
for (roi = 0; roi < numcc; roi++) {
    roiManager("Select", roi);
    getSelectionBounds(bx, by, roiw, roih);
    getStatistics(area, mean, min, max, std, histogram);
    print("ROI " + roi + " " + bx + " " + by + " " + roiw + " " + roih + " " + area);
    getSelectionCoordinates(x, y);
    for (i=0; i<x.length; i++)
      print(i+" "+x[i]+" "+y[i]);
}
```

Extraction of patch statistics

The user needs to define folder paths for `inputfolder`, `imagefolder`, `roifolder`, and `outputfolder`. The macro requires a segmented retina image, and oval ROIs for retinal margin and pre-induction retina. The input image should be the result of Appendix Section 5.7.1. Due to a cryptic bug introduced by an ImageJ update, the macro only successfully counts patch nodes in ImageJ version 1.50 or earlier.

Source Code 5.9: ImageJ macro for counting patch nodes and other statistics.

```
// Variables requiring user modification
inputfolder  = "C:\\inputfolder\\";
imagefolder  = "imagefolder\\";
roifolder    = "input_roi\\";
outputfolder = "C:\\outputfolder\\"
numrings = 19;
numsecs  = 10;

// measures post-induction retina length
function makeDVRL() {
    if (roiManager("Count")<2) { exit("Two ROIs in ROI Manager required"); }
    totalROIs = roiManager("count")-1;
    roiManager("Select", 0); //retinal margin ROI
    Roi.getBounds(x, y, width, height);

    D_x = x + width/2;      D_y = y;                    // Dorsal node
    V_x = D_x;      V_y = y + height;                   // Ventral node
    R_x = x + width; R_y = y + height/2;        // Right node
    L_x = x; L_y = R_y;                                 // Left node

    // make dorsal length
    roiManager("Select", 1); //pre-induction retina ROI
    final_y = 0;
    for (i = D_y; i <= D_y + height; i++) {
        if (selectionContains(D_x, i)) {
            final_y = i;
            i = D_y + height + 1;
        }
    }
    makeLine(D_x, D_y, D_x, final_y);
    roiManager("Add");
    roiManager("Select", totalROIs+1);
    roiManager("Rename", "d");
    run("Clear Results");
    run("Measure");
    D = getResult("Length");

    // make ventral length
    roiManager("Select", 1); //pre-induction retina ROI
    final_y = 0;
    for (i = V_y; i >= V_y - height; i--) {
        if (selectionContains(V_x, i)) {
            final_y = i;
            i = V_y - height - 1;
        }
    }
    makeLine(V_x, V_y, V_x, final_y);
    roiManager("Add");
    roiManager("Select", totalROIs+2);
    roiManager("Rename", "v");
    run("Clear Results");
    run("Measure");
    V = getResult("Length");

    // make right length
    roiManager("Select", 1); //pre-induction retina ROI
    final_x = 0;
    for (i = R_x; i >= R_x - width; i--) {
        if (selectionContains(i, R_y)) {
            final_x = i;
            i = R_x - width - 1;
        }
    }
    makeLine(R_x, R_y, final_x, R_y);
    roiManager("Add");
    roiManager("Select", totalROIs+3);
    roiManager("Rename", "r");
    run("Clear Results");
    run("Measure");
    R = getResult("Length");

    // make left length
    roiManager("Select", 1); //pre-induction retina ROI
    final_x = 0;
    for (i = L_x; i <= L_x + width; i++) {
        if (selectionContains(i, L_y)) {
            final_x = i;
            i = L_x + width + 1;
        }
    }
    makeLine(L_x, L_y, final_x, L_y);
    roiManager("Add");
    roiManager("Select", totalROIs+4);
    roiManager("Rename", "l");
    run("Clear Results");
    run("Measure");
    L = getResult("Length");

    ary = newArray(D, V, R, L);
    return ary;
}
// measures retinal size and creates concentric rings
function makeConcentric(ringnum) {
    divs  = ringnum + 1;
    // r ROI
    run("Clear Results");
    roiManager("Select", 0);
    run("Measure");
    rX  = getResult("XM");     rY = getResult("YM");           toUnscaled(rX, rY);
    rBX = getResult("BX");    rBY = getResult("BY");          toUnscaled(rBX, rBY);
    rBW = getResult("Width"); rBH = getResult("Height"); toUnscaled(rBW); toUnscaled(rBH);
    // pir ROI
    run("Clear Results");
    roiManager("Select", 1);
    run("Measure");
    pirX  = getResult("XM");     pirY = getResult("YM");           toUnscaled(pirX, pirY);
    pirBX = getResult("BX");    pirBY = getResult("BY");          toUnscaled(pirBX, pirBY);
    pirBW = getResult("Width"); pirBH = getResult("Height"); toUnscaled(pirBW); toUnscaled(pirBH);
    // lower left corner
    rBllY   =   rBY +   rBH;
    pirBllY = pirBY + pirBH;
    // upper right corner
    rBurX   =   rBX +   rBW;
    pirBurX = pirBX + pirBW;
    // Repeat ringnum times
    for (i = 1; i <= ringnum; i++) {
        newBX = pirBX - i*(pirBX - rBX)/divs;
        newBY = pirBY - i*(pirBY - rBY)/divs;
        newBllY = pirBllY - i*(pirBllY - rBllY)/divs;
        newBurX = pirBurX - i*(pirBurX - rBurX)/divs;
        newW  = newBllY - newBY;
        newH  = newBurX - newBX;
    }
    roiManager("Show All");
    toScaled(rBH); toScaled(rBW); toScaled(rX, rY); toScaled(pirBH); toScaled(pirBW); toScaled(pirX, pirY);
    ary = newArray(rBH, rBW, rX, rY, pirBH, pirBW, pirX, pirY);
    return ary;
}
// utility function to check if two ROIs intersect
function doesIntersect(idx1, idx2) {
    setBatchMode(true);
    roiManager("Select", idx1); run("Create Mask"); mask1 = getImageID();
    roiManager("Select", idx2);       run("Create Mask");       mask2 = getImageID();
    imageCalculator("AND", mask2, mask1);
    run("Select All"); getStatistics(area, mean);
    close();
    if (mean>0) {return(true); }
    else              {return(false);}
}
// utility function to get intersection area of two ROIs
function getIntersectArea(idx1, idx2) {
    setBatchMode(true);
    roiManager("Select", idx1); run("Create Mask"); mask1 = getImageID();
    roiManager("Select", idx2);       run("Create Mask");       mask2 = getImageID();
    imageCalculator("AND create", mask2, mask1);
    setThreshold(1, 255);
    run("Create Selection");
    getStatistics(area, mean);
    return(area);
}
// utility function to get first index of ROI in ROI manager with given name
function getFirstROIbyName(ROIname) {
    for (i = 2; i <= (roiManager("Count")); i++) {
        name = call("ij.plugin.frame.RoiManager.getName", i);
        if (name == ROIname + " 1") {
            firstROIidx = i;
            i = roiManager("Count") + 999;
        }
    }
    return(firstROIidx);
}
// wrapper for getFirstROIbyName where name is "sector"
function getSecs(selection, numsecs) {
    firstsec = getFirstROIbyName("sector");
    first = 0;
    postgap = 0;
    gap = 0;
    last = 0;
    for (i = 0; i < numsecs*4; i++) {
        cursec = i + 1;
        intsct = doesIntersect(firstsec + i, selection);
        if (intsct == 1) {
            last = cursec; // last sector found
            if ( first == 0 ) { first   = cursec; } // first sector found
            else if ( gap > 0 && postgap == 0 ) { postgap = cursec; } // first sector found after gap
        }
        else if ( first > 0 ) {
            gap = gap + 1;
        }
    }
    // correct for wrapping around circle
    if (postgap == 0) {
        secs = newArray(first, last);
    }
    else {
        secs = newArray(postgap, postgap - gap - 1);
    }
    return secs;
}
// wrapper for getFirstROIbyName where name is "ring"
function getRings(selection, numrings) {
    firstring = getFirstROIbyName("ring");
    first = 0;
    last  = 0;
    for (i = 0; i < numrings+1; i++) {
        intsct  = doesIntersect(firstring + i, selection);
        if (intsct == 1) {
            last = i + 1;
            if (first == 0) { first = i + 1; }
        }
    }
    rings = newArray(first, last); // if no intersection returns [0,0]
    return rings;
}
// wrapper for getFirstROIbyName where name is "quadrant"
function getQuads(selection) {
    firstquad = getFirstROIbyName("quadrant");
    prevarea = 0;
    quadid   = 0;
    for (quad = 0; quad < 4; quad++) {
        intsct  = doesIntersect(selection, firstquad + quad);
        if (intsct == 1) {
            area = getIntersectArea(selection, firstquad + quad);
            if (prevarea < area) {
                prevarea = area;
                quadid   = quad + 1;
            }
        }
    }
    return quadid;
}
// skeletonize and get node map using Skeleton Analyser plugin
function getSkeletonNodeMap() {
    run("Skeleton Analyser", " min_length=0 show_node_map");
    ske = getImageID();
    return ske;
}
```

```
// count nodes in skeletonized image by finding green pixels
function countAndFindNodes(numrings) {
    branches  = 0;
    nodelist = "";
    // Count green pixels (= branch points)
    W = getWidth(); H = getHeight();
    for (x=0;x<W;x++){
        for (y=0;y<H;y++){
            v = getPixel(x,y);
        red   = (v>>16)&0xff;  // extract red byte (bits 23-17)
        green = (v>>8)&0xff;   // extract green byte (bits 15-8)
        blue  = v&0xff;        // extract blue byte (bits 7-0)
        if (green == 255) {
            branches += 1;
            makePoint(x, y);
            roiManager("Add");
            pID = roiManager("Count")-1;
            nodes = getRings(pID, numrings); // if node outside of rings nodes = [0,0]
            if (nodelist == "") { // first element
                nodelist = toString(nodes[0]);
            }
            else {
                nodelist = nodelist + "; " + toString(nodes[0]);
            }
            // delete CC selection
                roiManager("Select", pID);
                roiManager("Delete");
        }
        }
    }
    ary = newArray(branches, nodelist);
    return ary;
}
// wrapper function for custom skeletonization algorithm
function getCustomSkeleton(blocks, deltax) {
    // get measurements
    setBatchMode(true);
    ori2 = getImageID(); selectImage(ori2);
    setForegroundColor(255, 255, 255);
    setBackgroundColor(0, 0, 0);
    run("Clear Results"); roiManager("Select", 1);
    run("Measure"); run("Select None");
    xcenter = getResult("X"); ycenter = getResult("Y");
    h = getHeight(); w = getWidth();
    toUnscaled(xcenter, ycenter);
    unwrap(xcenter, ycenter);
    pol = getImageID();
    setThreshold(1, 255); run("Convert to Mask");
    wwrap = getWidth();
    correction = correctSplitBlobs(wwrap);
    run("Shape Smoothing", "relative_proportion_fds=5 absolute_number_fds=2 keep=[Relative_proportion of FDs] black");
    run("Find Connected Regions", "allow_diagonal display_one_image regions_for_values_over=100
    ↪  minimum_number_of_points=1 stop_after=-1");
    ccimage = getImageID();
    loopCC(blocks);
    if (correction == true) {
        reconstruct(wwrap);
    }
    skecart = getImageID();
    backTransf(xcenter, ycenter, w, h);
    skepol = getImageID();
    selectImage(skecart); run("Close");
    selectImage(skepol);
    // get rid of interpolation smoothing
    setThreshold(1, 255); run("Convert to Mask");
    // pre-process to skeletonize again
    run("Dilate"); run("Dilate"); run("Erode");
    run("Dilate"); run("Dilate"); run("Erode");
    run("Median...", "radius=1");
    run("Shape Smoothing", "relative_proportion_fds=5 absolute_number_fds=2 keep=[Relative_proportion of FDs] black");
    run("Median...", "radius=1");
    run("Skeletonize");
    selectImage(skepol);
    return getImageID();
}
// loop through connected components to get their skeletons
function loopCC(blocks) {
    polori = getImageID();
    getStatistics(area, mean, min, max);
    newImage("skeleton", "8-bit black", getWidth(), getHeight(), 1);
    skefull = getImageID(); skeftit = getTitle();
    run("Median...", "radius=0");
    for (i = 1; i <= max; i++) {
        setBatchMode(true);
        selectImage(polori);
        run("Duplicate...", " "); temp = getImageID();
        setThreshold(i, i); run("Convert to Mask");
        // get skeleton
        customSkeleton(blocks, deltax);
        skeres = getImageID(); sketit = getTitle();
        imageCalculator("Add", skeftit, sketit);
        selectImage(skeres); run("Close");
    }
    selectImage(polori); run("Close");
    selectImage(skefull);
}
// correct for patches that were split in transform
function correctSplitBlobs(wwrap) {
    correction = false;
    blobFirst  = -1;
    blobLast   = -1;
    for (y = 0; y <= getHeight(); y++) {
        v = getPixel(0, y);
        if (v > 0) { // potentially split blob
            correction = true;
            if (blobFirst == -1) {
                doWand(0, y); roiManager("Add");
                blobFirst = roiManager("Count")-1;
            }
            else { // OR found blobs
                doWand(0, y); roiManager("Add");
                blobLast  = roiManager("Count")-1;
                roiManager("Select", newArray(blobFirst, blobLast));
                roiManager("Combine"); roiManager("Add");
                blobFirst = roiManager("Count")-1;
                roiManager("Select", blobFirst-1); roiManager("Delete"); run("Select None");
                blobFirst = roiManager("Count")-1;
            }
        }
    }
    if (blobFirst > -1) {
        blobFirst = roiManager("Count")-1;
        run("Clear Results"); roiManager("Select", blobFirst);
        run("Measure"); run("Cut");
        run("Select None");
        blobw = getResult("Width"); toUnscaled(blobw);
        // enlarge canvas
        newWidth  = wwrap + blobw;
        newHeight = getHeight();
        run("Canvas Size...", "width="+ newWidth +" height="+newHeight+" position=Center-Left");
        // translate selection
        roiManager("Select", blobFirst); roiManager("translate", wwrap, 0);
        run("Set...", "value="+255);
        roiManager("Select", blobFirst); roiManager("Delete"); run("Select None");
    }
    return correction;
}
// reconstruct original image before transforming back
function reconstruct(wwrap) {
    makeRectangle(wwrap + 1, 0, getWidth()-wwrap, getHeight());
    roiManager("Add"); extra = roiManager("Count")-1;
    // threshold all white pixels, then AND to extra canvas area
    setThreshold(1, 255); run("Create Selection"); roiManager("Add");
    skelsel = roiManager("Count")-1;
    roiManager("Select", newArray(extra, skelsel));
    roiManager("AND"); andID = roiManager("Count")-1;
    // translate selection
    roiManager("Select", andID); roiManager("translate", -wwrap, 0);
    run("Set...", "value="+255);
    roiManager("Select", andID); roiManager("Delete"); run("Select None");
    // crop to original size
    run("Canvas Size...", "width="+wwrap+" height="+getHeight()+" position=Center-Left");
}
// transform from polar to Cartesian
function unwrap(xcenter, ycenter) {
    run("Polar Transformer", "method=Polar degrees=360 number=3600 center_x="+xcenter+" center_y="+ycenter+"
    ↪  interpolation=Bilinear");
    run("Rotate 90 Degrees Right");
}
// transform from Cartesian to polar
function backTransf(xcenter, ycenter, w, h) {
    run("Rotate 90 Degrees Left");
    run("Polar Transformer", "method=Cartesian degrees=360 center_x="+xcenter+" center_y="+ycenter+"");
    // resize to original size
    run("Canvas Size...", "width="+w+" height="+h+" position=Top-Left");
}
// custom skeletonization algorithm tailored to radial extent of retinal clones
function customSkeleton(ybins, deltax) {
    // create skeleton based on x-row average within y-bins
    ori = getImageID(); selectImage(ori); run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");
    newImage("x-Skeleton", "8-bit black", getWidth(), getHeight(), 1);
    ske = getImageID();
    maxblobs = floor(getWidth()/2)+getWidth%2;
    // create blank images and fill with resulting values
    newImage("blob_x1", "32-bit black", maxblobs, getHeight(), 1); bx1 = getImageID();
    newImage("blob_x2", "32-bit black", maxblobs, getHeight(), 1); bx2 = getImageID();
    newImage("blob_xm", "32-bit black", maxblobs, getHeight(), 1); bxm = getImageID();
    selectImage(ori);
    blobcoords = findBlobCoords();
    deltay = round(blobcoords[3]/(ybins-1));
    if (deltay < 5) { deltay = 5; } // deltay shouldn't be less than 5 pixels --> faster
    pblobs = -1;
    deltax = deltay;
    for (ybin = blobcoords[1]; ybin < blobcoords[1] + blobcoords[3]-deltay; ybin+=deltay) { // loop through y-bins
        // create rectangle spanning entire width and one y-bin of height deltay
        selectImage(ori);
        makeRectangle(0, ybin, getWidth(), deltay);
        // isolate rectangle and get all clone fragments as connected components
        run("Duplicate...", " "); binbox = getImageID(); selectImage(binbox); run("Select None");
        run("Find Connected Regions", "allow_diagonal display_one_image regions_for_values_over=100
        ↪   minimum_number_of_points=1 stop_after=-1");
        // threshold each component (=blob) and find the margins and mid-point
        bincc = getImageID(); selectImage(binbox); run("Close");
        selectImage(bincc); getStatistics(area, mean, min, max);
        if ( max != 0 ) { // only loop if there are blobs
            for (i = 1; i <= max; i++) { // loop through blobs by thresholding them
                selectImage(bincc);        run("Duplicate...", " ");
                cctemp = getImageID(); selectImage(cctemp);
                setThreshold(i, i); setOption("BlackBackground", true); run("Convert to Mask");
                selectImage(cctemp);
                // use custom function to find bounding box of blob in this bin
                cccords = findBlobCoords();
                x1 = cccords[0]; x2 = x1 + cccords[2];
                selectImage(cctemp); run("Close");
                // save bounding box x-values
                selectImage(bx1); setPixel(i, ybin, x1);
                selectImage(bx2); setPixel(i, ybin, x2);
                // calculate and save middle x value
                xm = x1 + round((x2 - x1)/2);
                selectImage(bxm); setPixel(i, ybin, xm);
                selectImage(ske);
                // draw a line through middle x value spanning the bin and up to clone extremes
                if (ybin + deltay > blobcoords[1] + blobcoords[3]) {
                    drawLine(xm, ybin, xm, blobcoords[1] + blobcoords[3]);
                }
                else {
                    drawLine(xm, ybin, xm, ybin+deltay);
                }
                // check connectivity
                if (pblobs > -1) {
                    for (ypyblob = 1; ypyblob <= pblobs+1; ypyblob++) { // loop through blobs in line y-1
                        pbin = ybin - deltay;
                        selectImage(bx1); x1prev = getPixel(ypyblob, pbin);
                        selectImage(bx2); x2prev = getPixel(ypyblob, pbin);
                        selectImage(bxm); xmprev = getPixel(ypyblob, pbin);

                        // as long as at least one pixel of the y-py blob is contained, count as a connection (8-connected)
                        cnx = false;
                        if (xm == xmprev || xm == xmprev - 1 || xm == xmprev + 1) {
                                //already connected
                                cnx = true;
                            }
                            else if ( ( x1 - deltax ) <= x1prev && x1prev <= ( x2 + deltax ) ) {
                                cnx = true;
                            }
                            else if ( ( x1 - deltax )<= x2prev && x2prev <= ( x2 + deltax ) ) {
                                cnx = true;
                            }
                            else if ( x1prev <= ( x1 - deltax ) && ( x1 - deltax ) <= x2prev )  {
                                cnx = true;
                            }
                            else if ( x1prev <= ( x2 + deltax ) && ( x2 + deltax ) <= x2prev ) {
                                cnx = true;
                        }

                        if (cnx == true && xmprev > -1) {
                            selectImage(ske);
                            drawLine(xmprev, ybin, xm, ybin);
                        }
                    }
                }
            pblobs = max;
            }
            selectImage(bincc); run("Close");
        }
    }
    selectImage(ske);
}
// utility function to find coordinates of a sub-patch element (blob) during skeletonization
function findBlobCoords() {
    foundx = -99;
    foundy = -99;

    getStatistics(area, mean, min, max);
    if (max == 0) { //no blob in picture
        ary = newArray(0, 0, 0, 0);
    }
    else {
        for (y = 0; y < getHeight(); y++) {
            for (x = 0; x < getWidth(); x++) {
                v = getPixel(x,y);
                if (v == 255) {
                    foundx = x;
                    foundy = y;
                    x = getWidth + 1;
                    y = getWidth + 1;
                }
            }
        }
        doWand(foundx, foundy);
        run("Select Bounding Box");
        run("Clear Results");
        run("Measure");
        run("Select None");
        foundx = getResult("BX");
        foundy = getResult("BY");
        foundh = getResult("Height");
        foundw = getResult("Width");
        toUnscaled(foundx, foundy);
        toUnscaled(foundh);
        toUnscaled(foundw);
        ary = newArray(foundx, foundy, foundw, foundh);
    }
    return ary;
}
// wrapper function that calls everything else and measures all properties
function quantifyCC(numrings, numsecs, aryDVRL, arySizes) {
    ori = getImageID();
    // get skeleton map
    skefull = getCustomSkeleton(45, -5);
    setBatchMode(true);
    selectImage(skefull); skefulltit = getTitle();
    // get skeleton node map
    run("Skeleton Analyser", "  min_length=0 show_node_map");
    nodemap = getImageID();
    // re-open rois and continue
    roiManager("Open", roifile);
    selectImage(ori);
    run("Clear Results"); roiManager("Select", 0);  run("Measure"); run("Select None");
    xcenter = getResult("XM"); ycenter = getResult("YM");
    toUnscaled(xcenter, ycenter);
    run("Find Connected Regions", "allow_diagonal display_one_image regions_for_values_over=100
    ↪   minimum_number_of_points=1 stop_after=-1");
    ccimage = getImageID(); getStatistics(area, mean, min, max);
    // Initialize arrays to number of connected components
    arr_id  = newArray(max); // pixel value
    arr_q   = newArray(max); // quadrant
    arr_s1  = newArray(max); // start sector
    arr_s2  = newArray(max); // end sector
    arr_r1  = newArray(max); // start ring
    arr_r2  = newArray(max); // stop ring
    arr_n   = newArray(max); // nodes
    arr_nr  = newArray(max); // node ring range
    dorsalCC  = 0;
    rightCC   = 0;
    ventralCC = 0;
    leftCC    = 0;
    firstring = getFirstROIbyName("ring");
    for (i = 1; i <= max; i++) {
        setBatchMode(true);
        j = i-1;
        // get a selection for the current cc
        selectImage(ccimage);
        arr_id[j] = i;
        setThreshold(i, i);      run("Create Selection"); roiManager("Add");      selectionID = roiManager("Count")-1;
        // find quadrant with greatest area occupation
        arr_q[j] = getQuads(selectionID);
        if      ( arr_q[j] == 1 ) { dorsalCC  += 1; }
        else if ( arr_q[j] == 2 ) { rightCC   += 1; }
        else if ( arr_q[j] == 3 ) { ventralCC += 1; }
        else                      { leftCC    += 1; }
        // find range of sectors
        secary = getSecs(selectionID, numsecs);
        arr_s1[j] = secary[0];
        arr_s2[j] = secary[1];
        // find range of rings
        ringary = getRings(selectionID, numrings);
        arr_r1[j] = ringary[0];
        arr_r2[j] = ringary[1];
        // find corresponding skeleton
        // isolate CC area in skeleton map
        selectImage(skefull); run("Duplicate...", " "); dps = getImageID();
        roiManager("Select", selectionID); run("Make Inverse");
        setBackgroundColor(0, 0, 0); run("Cut"); run("Select None");
        // find largest of all skeletal elements inside this area
        run("Find Connected Regions", "allow_diagonal display_one_image regions_for_values_over=100 minimum_number_of_points=1 stop_after=-1");
        skeccimage = getImageID(); getStatistics(skarea1, skmean, skmin, skmax);
        selectImage(dps); run("Close");
        selectImage(skeccimage); preskarea = -1; largest = -1;
        for (skecc = 1; skecc <= skmax; skecc++) { // loop through skeletal elements
            setThreshold(skecc, skecc);
            run("Create Selection"); run("Clear Results"); run("Measure");
            skarea = getResult("Area");
            if (preskarea < skarea) {
                preskarea = skarea;
                largest = skecc;
            }
        }
        if (preskarea > 0) { // skip small patches that leave no skeleton
            // find a point inside largest skeletal element and get the full selection
            selectImage(skefull); run("Duplicate...", " "); dps = getImageID();
            selectImage(nodemap); run("Duplicate...", " "); dpn = getImageID();
            selectImage(skeccimage); setThreshold(largest, largest); run("Create Selection"); run("Clear Results"); run("Measure");
            selectImage(skeccimage); skey = getResult("BY");
            for (skex = getResult("BX"); skex <= getResult("BX") + getResult("Width"); skex++) {
                skev = getPixel(skex, skey);
                if (skev > 0) {
                    // isolate corresponding part in the skeleton node map
                    selectImage(dps); doWand(skex, skey);
                    roiManager("Add"); skeselID = roiManager("Count")-1;
                    selectImage(dpn); roiManager("Select", skeselID); run("Make Inverse");
                    setBackgroundColor(0, 0, 0); run("Cut");
                    run("Select None");
                    selectImage(dps); run("Close");
                    // stop loop
                    skex = getResult("BX") + getResult("Width") + 1;
                }
            }
            // get nodessmooth
            selectImage(dpn); setBatchMode("show"); // turning off batch mode no longer fixes counting problem since update!!
            result = countAndFindNodes(numrings); // BROKEN SINCE UPDATE !!
            selectImage(dpn); run("Close");
            arr_n[j]  = result[0];
            arr_nr[j] = result[1];
        }
        else { // small patches that leave no skeleton
```

```
            arr_n[j]  = 0;
            arr_nr[j] = "";
        }
        print(f, filepath +"\t "+ arr_id[j] +"\t "+ arr_q[j] +"\t "+ arr_s1[j] +"\t "+ arr_s2[j] +"\t "+ arr_r1[j] +"\t "+ arr_r2[j] +"\t "+ arr_n[j] +"\t "+ arr_nr[j]);
    }
    selectImage(ori); setBatchMode("show");
    selectImage(skefull); setBatchMode("show");
    selectImage(nodemap); setBatchMode("show");
    getPixelSize(unit, pixelWidth, pixelHeight);
    scale = toString(pixelWidth) + " " + toString(unit) + "/pixel";
    print(s, filepath + "\t "+ scale +"\t "+ arySizes[0] +"\t "+ arySizes[1] +"\t "+ arySizes[2] +"\t "+ arySizes[3] +"\t "+ arySizes[4] +"\t "+ arySizes[5] +"\t "+ arySizes[6] +"\t "+ arySizes[7] +"\t "+ aryDVRL[0] +"\t "+ aryDVRL[1] +"\t "+ aryDVRL[2] +"\t "+ aryDVRL[3] +"\t "+ dorsalCC +"\t "+ ventralCC +"\t "+ rightCC +"\t "+ leftCC +"\t "+ max);
    setBatchMode(false);
    selectImage(skefull); run("Dilate"); run("Dilate"); run("Dilate"); run("Green");
    smoothT = getTitle(); selectImage(ori);
    run("Overlay Image", "image=[" + smoothT + "] x=0 y=0 opacity=50 transparent replace");
    rename("Overlay");
    string = outputfolder + filename + "_overlay";
    saveAs("Tiff", string);
    run("Close All");
}

// Create table with results - general descriptors
title1 = "General_Descriptors";
title2 = "["+title1+"]";
s      = title2;
if (isOpen("General_Descriptors")) {
    }
else {
    run("New... ", "name="+title2+" type=Table");
    print(s,"\\Headings:File\t Resolution\t Retina Height\t Retina Width\t Retina X\t Retina Y\t Pre-Induction Height\t Pre-Induction Width\t Pre-Induction X\t Pre-Induction Y\t Post-Induction Dorsal\t Post-Induction Ventral\t Post-Induction Right\t Post-Induction Left\t Dorsal CC\t Ventral CC\t Right CC\t Left CC\t Total CC");
}
// Create table with results - clone descriptors
title1 = "Clone_Descriptors";
title2 = "["+title1+"]";
f      = title2;
if (isOpen("Clone_Descriptors")) {
    }
else {
    run("New... ", "name="+title2+" type=Table");
    print(f,"\\Headings:File\t Clone CC ID\t Quadrant ID\t Start Sector\t End Sector\t Start Ring\t End Ring\t Nodes\t Node Rings");
}
// Execute functions
list = getFileList(inputfolder + imagefolder);
numprocess = list.length;
for (l = 0; l < numprocess; l++) {
  open(inputfolder + imagefolder + list[l]);
  filename = getInfo("image.filename");
    filepath = getInfo("image.directory") + filename;
    roifile  = inputfolder + roifolder + filename + "_RoiSet.zip";
    roiManager("Open", roifile);
    roiManager("Show None");
    run("Select None");
    gen_DVRL  = makeDVRL();
    gen_sizes = makeConcentric(numrings);
    quantifyCC(numrings, numsecs, gen_DVRL, gen_sizes);
    run("Close All");
    run("Collect Garbage");
}
// Save data windows
list = getList("window.titles");
for (i=0; i<list.length; i++) {
    if (list[i] == "Recorder" || list[i] == "Log" || list[i] == "Results" || list[i] == "Command Finder" || list[i] == "Exception" || list[i] ==
    ↪  "CP" || list[i] == "ROI Manager" || list[i] == "Debug") {
        //ignore these windows
    }
    else {
        selectWindow(list[i]);
        string = outputfolder + savename + "_" + list[i] + ".csv";
        saveAs("Text", string);
        //run("Close");
    }
}
```

Create concentric rings

This macro is a standalone version of one of the functions in Appendix Section 5.7.1. It requires two concentric elliptic ROIs in the ROI Manager; the outer ellipse should be first.

Source Code 5.10: ImageJ macro for making concentric rings.

```
// ROI subdivisions
numrings = 1;
// Required: First ROI in manager is larger ellipse, second ROI is smaller ellipse.
if (roiManager("Count")<2) { exit("Two ROIs in ROI Manager required"); }
makeConcentric(numrings);
// Constructs concentric annular ROIs based on two concentric ellipses
function makeConcentric(ringnum) {
    divs  = ringnum + 1;
    // outer ellipse ROI
    run("Clear Results");
    roiManager("Select", 0);
    run("Measure");
    rX  = getResult("XM");    rY = getResult("YM");        toUnscaled(rX, rY);
    rBX = getResult("BX");    rBY = getResult("BY");       toUnscaled(rBX, rBY);
    rBW = getResult("Width"); rBH = getResult("Height"); toUnscaled(rBW); toUnscaled(rBH);
    // inner ellipse ROI
    run("Clear Results");
    roiManager("Select", 1);
    run("Measure");
    pirX  = getResult("XM");    pirY = getResult("YM");        toUnscaled(pirX, pirY);
    pirBX = getResult("BX");    pirBY = getResult("BY");       toUnscaled(pirBX, pirBY);
    pirBW = getResult("Width"); pirBH = getResult("Height"); toUnscaled(pirBW); toUnscaled(pirBH);
    // lower left corner
    rBllY   =   rBY +   rBH;
    pirBllY = pirBY + pirBH;
    // upper right corner
    rBurX   =   rBX +   rBW;
    pirBurX = pirBX + pirBW;
    // Repeat ringnum times
    for (i = 1; i <= ringnum; i++) {
        newBX = pirBX - i*(pirBX - rBX)/divs;
        newBY = pirBY - i*(pirBY - rBY)/divs;
        newBllY = pirBllY - i*(pirBllY - rBllY)/divs;
        newBurX = pirBurX - i*(pirBurX - rBurX)/divs;
        newW  = newBllY - newBY;
        newH  = newBurX - newBX;
         makeOval(newBX, newBY, newH , newW);
         roiManager("Add");
     }
}
```

Process simulated segmentation

Input images are simulated segmentations generated by Appendix Script 5.25 using the option `unique_colour=True`. The Python matplotlib library only allows plotting RGB images, so the unique ID of each clone is encoded in the R and B values: ID = R + 255·B. This allows for a maximum of $255^2 = 65025$ IDs, which fits into 16-bit images and is more than enough to capture all clones.

The macro loops through files in folder "input" and runs the ArCoS Analysis plugin. The results are saved to the folder "output".

Source Code 5.11: ImageJ macro for automatically processing simulated segmentation files.

```
// Variables requiring user modification
input = "C:\\Inputfolder\\";
output ="C:\\Outputfolder\\";

// Utility function for trimming zero value rows from the bottom of the image
function trimBottom() {
    W = getWidth(); H = getHeight();
    for (y=H;y>0;y--){
        for (x=0;x<W;x++){
            v = getPixel(x,y);
            if (v != 0) {
                makeRectangle(0,0, W, y+1);
                run("Crop");
                y = -99;
            }
        }
    }
}

function process(output) {
    ori    = getImageID();       oritit = getTitle();
    run("Rotate 90 Degrees Left"); // dorsal goes up
    run("Auto Crop (guess background color)");
    // Now the spacer gray elements are not needed anymore
    run("Set Measurements...", "area centroid center perimeter bounding fit shape feret's redirect=None decimal=3");
    run("Clear Results");        run("Measure"); title = getTitle();
    X = round(getResult("X")); Y = round(getResult("Y")); W = getWidth(); H = getHeight();
    makeRectangle(0,0, W, H);
    // This for loop erases the gray ring
    for (x=0;x<W;x++){
        for (y=0;y<H;y++){
            v = getPixel(x,y);
        red   = (v>>16)&0xff;  // extract red byte (bits 23-17)
        green = (v>>8)&0xff;   // extract green byte (bits 15-8)
        blue  = v&0xff;        // extract blue byte (bits 7-0)
        if (red != 255 && red == green && red == blue) {
            setColor(255, 255, 255);
            drawLine(x, y, x, y);
        }
        }
    }
    run("Split Channels");
    selectWindow(oritit + " (green)"); close();
    selectWindow(oritit + " (blue)"); blue = getImageID(); bluetit = getTitle();
    run("16-bit"); run("Multiply...", "value=255");
    selectWindow(oritit + " (red)");  red = getImageID();  redtit = getTitle();
    run("16-bit");
    imageCalculator("Add create", redtit, bluetit); ori = getImageID();
    // Since the background is white, it gets a value of 256.
    // The Arcos plugin needs this value to be 0, so the value is corrected in this loop.
    run("Invert");       W = getWidth(); H = getHeight();
    for (x=0;x<W;x++){
        for (y=0;y<H;y++){
            v = getPixel(x,y);
        if (v == 256 || v == 257) {  // sometimes the value needs to be 257
            setColor(0);
            drawLine(x, y, x, y);
        }
        }
    }
    // Smooth the image a bit
    run("Median...", "radius=3");
    for (j = 0; j < 10; j++) {
        run("Median...", "radius=7");
        }
    run("Median...", "radius=3");
    // Save the processed image and reopen it (allows some speedup somehow)
    string = output + "Images\\" + oritit + "_pol.tif";
    saveAs("Tiff", string); run("Close All");
    open(string);
    // Remove scale bar
    W = getWidth(); H = getHeight();
    for (x=0;x<W;x++){
        for (y=0;y<H;y++){
            v = getPixel(x,y);
        if (v == 1785) {
            setColor(0);
            drawLine(x, y, x, y);
        }
        }
    }
    // Transform the polar plot into a rectangular projection for analysis
```

```
    string = "in=" + title + " direction=[to Polar] originx=" + X +" originy=" + Y + " standardazimuth=0 dimensions=x=Radius
    ↪   interpolation=Nearest msg1=[to Polar only] azimuthalsampling=1.0 radiusfactor=1.0  msg2=[from Polar only] sizex="
    ↪   + W +" sizey=" + H + " interactive=false";
    run("Polar Transform", string); map = getImageID(); run("Rotate 90 Degrees Right");
    pol = getImageID();
    // Trim the bottom of the image up to the first segmented pixel
    trimBottom();
    selectImage(pol);
    // Save the processed image and reopen it (somehow prevents arcos plugin from giving wrong values?)
    string = output + "Images\\" + oritit + "_rec.tif";
    saveAs("Tiff", string); run("Close All");
    open(string);
    // Run the Arcos plugin
    outtit = getTitle(); H = getHeight(); W = getWidth();
    string = "in=" + outtit +" axialdimension=Y startax=0 endax=" + H + " clonesize=false clonecontribution=true
    ↪   cloneexpansion=false clonemeanexpansion=false cloneregression=false sectormeanexpansionrate=false
    ↪   sectormeanexpansionratio=false setlut=false movingaverageradius=5";
    run("ArCoS Analysis Rectangular", string);
    run("Close All");
    // Save data windows
    list = getList("window.titles");
    for (i=0; i<list.length; i++) {
        if (list[i] == "Recorder" || list[i] == "Log" || list[i] == "Results" || list[i] == "Command Finder" || list[i] == "Exception" ||
        ↪   list[i] == "ROI Manager" || list[i] == "Debug") {
            //ignore these windows
        }
        else {
            selectWindow(list[i]);
            string = output + oritit + "_" + list[i] + ".csv";
            saveAs("Results", string);
            run("Close");
        }
    }
}
// Wrapper function
function fileloop (input, output, filename) {
    open(input + filename);
    process(output);
}

list = getFileList(input);
numprocess = list.length; //the macro may slow down when processing too many files
for (j = 0; j < numprocess; j++) {
  fileloop(input, output, list[j]);
}
```

5.7.2 Python scripts

Plot of average cell displacement against calculation time

EPISIM Simulator was used to output the average distance of migration and calculation time after every step during model initialisation. An excerpt of such a file is shown in 5.1. The data was parsed and plotted using the script in 5.12

Listing 5.1: Excerpt of output from EPISIM Simulator console

```
Global BM Sim Step: 1141 ms
Min Avg Migration um: 4.2412436033157315
Global BM Sim Step: 798 ms
Min Avg Migration um: 1.5811785834082985
Global BM Sim Step: 879 ms
Min Avg Migration um: 1.406825817353061
Global BM Sim Step: 1224 ms
Min Avg Migration um: 1.3312041047226355
Global BM Sim Step: 1144 ms
Min Avg Migration um: 1.2702085653995145
```

Source Code 5.12: Code used to generate the plot in Figure 2.4 panel C

```
# -*- coding: utf-8 -*-
import csv
import matplotlib as mpl
import pylab as pl

def plotMigrationMilliS(csvfile):

    # names of headers in csv file
    stringMigration = "Min Avg Migration um: "

    # initialize arrays for data
    avgMigration = []
    calcms       = []
    calcSumms    = []
    idx          = 0

    with open(csvfile, "r") as csvfile:
        reader = csv.reader(csvfile)
        for item in reader: # iterates over rows
            string = item[0]
            if stringMigration in string:
                avgMigration.append(float(string.split()[-1]))
            else:
                calcms.append(float(string.split()[-2]))
                if idx == 0:
                    calcSumms.append(calcms[idx])
                else:
                    calcSumms.append(calcSumms[idx-1]+calcms[idx])
                idx += 1

    fig = pl.figure()
    ax  = fig.add_subplot(111)

    finalIdx = 200

    ax.plot(calcSumms[0:finalIdx], avgMigration[0:finalIdx],
            'g', linewidth = 1)

    mpl.pyplot.yticks([0, 0.07, 0.2, 0.7, 1, 2, 3, 4])

inputdir  = "C:\\Users\\ET\\Seafile\\Ronbun\\Ronbun\\Scripts\\"
csvfile = inputdir + "example_EPISIM_console.txt"
plotMigrationMilliS(csvfile)
```

Plot of fish body and eye size measurements

Measurements made in ImageJ were copied directly into the script in 5.13 for plotting.

Source Code 5.13: Code used to generate the plot in Figure 2.6 panel A

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import numpy as np

# raw data copied manually from ImageJ
# measurements for 1 dph
eye_01 = [0.3795, 0.381, 0.3785, 0.3795]
bod_01 = [4.969, 5.005, 5.124, 4.965]
n_01 = 2
# measurements for 2 dph
eye_02 = [0.418, 0.413, 0.4, 0.386, 0.3875, 0.3975, 0.3965, 0.388, 0.3885, 0.4205]
bod_02 = [5.529, 5.416, 5.443, 4.953, 4.97, 5.18, 5.169, 5.354, 5.354, 5.52]
n_02 = 5
# measurements for 7 dph
eye_07 = [0.571, 0.5695, 0.527, 0.503, 0.5205, 0.4885]
bod_07 = [7.425, 7.515, 6.705, 6.475, 6.704, 6.433]
n_07 = 3
# measurements for 10 dph
eye_10 = [0.604, 0.599, 0.5875, 0.585]
bod_10 = [8.359, 8.345, 8.119, 7.861]
n_10 = 2
# measurements for 14 dph
eye_14 = [0.694, 0.599, 0.613, 0.669]
bod_14 = [10.06, 8.989, 8.998, 10.017]
n_14 = 2
# measurements for 21 dph
eye_21 = [0.8495, 0.8245, 0.841, 0.851]
bod_21 = [11.052, 11.146, 11.357, 11.33]
n_21 = 2
# measurements for 35 dph
eye_35 = [1.34, 1.188, 1.298, 1.47]
bod_35 = [18.03, 17.393, 17.98, 17.524]
n_35 = 2
# measurements for 42 dph
eye_42 = [1.29, 1.328, 1.314, 1.288]
bod_42 = [19.441, 19.524, 19.481, 19.695]
n_42 = 2
# measurements for 49 dph
eye_49 = [1.534, 1.5525, 1.562, 1.51]
bod_49 = [23.469, 23.36, 23.532, 23.442]
```

```
n_49 = 2
# measurements for 56 dph
eye_56 = [1.7645, 1.738, 1.7185, 1.6725]
bod_56 = [25.658, 25.968, 25.622, 25.554]
n_56 = 2
# measurements for 93 dph
eye_93 = [2.027, 1.977, 2.1015, 2.258, 2.258, 2.307, 2.2215]
bod_93 = [34.135, 34.209, 35.72, 36.033, 35.03, 36.279]
n_93 = 3

# group data into one list each
# list of eye diameters in mm
eye_growth = [eye_01, eye_02, eye_07, eye_10, eye_14, eye_21, eye_35, eye_42, eye_56, eye_93]
# list of body length in mm
bod_growth = [bod_01, bod_02, bod_07, bod_10, bod_14, bod_21, bod_35, bod_42, bod_56, bod_93]
# number of fish measured per condition
ns = [n_01, n_02, n_07, n_10, n_14, n_21, n_35, n_42, n_56, n_93]
# days post hatch
ds = [1, 2, 7, 10, 14, 21, 35, 42, 56, 93]

def adjustFigAspect(fig,aspect=1):
    '''
    Adjust the subplot parameters so that the figure has the correct aspect ratio.
    '''
    xsize,ysize = fig.get_size_inches()
    minsize = min(xsize,ysize)
    xlim = .4*minsize/xsize
    ylim = .4*minsize/ysize
    if aspect < 1:
        xlim *= aspect
    else:
        ylim /= aspect
    fig.subplots_adjust(left=.5-xlim, right=.5+xlim, bottom=.5-ylim, top=.5+ylim)

# initialize empty lists
y_ep = []
y_bp = []
x_ed = []
x_bd = []

# restructure data lists
for idx in range(len(eye_growth)):
    # eye data
    for item in eye_growth[idx]:
        x_ed.append(ds[idx])
        y_ep.append(item)
    # body data
    for item in bod_growth[idx]:
        x_bd.append(ds[idx])
        y_bp.append(item)

fig, ax1 = plt.subplots()
ax1.patch.set_facecolor((1.0,1.0,1.0))

# plot body length in green
ax1.scatter(x_bd, y_bp, c = 'g', s = 25, alpha = 0.5)
ax1.set_ylabel('body length [mm]', color='g')
```

```python
ax1.tick_params('y', colors='g')
plt.yticks(np.arange(0, 41, 5))
ax1.set_ylim([0, 40])

# fit a linear equation to the points
plt.plot(np.unique(x_bd), np.poly1d(np.polyfit(x_bd, y_bp, 1))(np.unique(x_bd)), 'g--', alpha = 0.5)

# duplicate the axis to plot eye diameter in magenta into the same plot
ax2 = ax1.twinx()
ax2.scatter(x_ed, y_ep, c = 'm', s = 25, alpha = 0.5)
ax2.set_ylabel('eye diameter [mm]', color='m')
ax2.tick_params('y', colors='m')
plt.yticks(np.arange(0, 2.6, 0.25))
ax2.set_ylim([0, 2.5])

# fit a linear equation to the points
plt.plot(np.unique(x_ed), np.poly1d(np.polyfit(x_ed, y_ep, 1))(np.unique(x_ed)), 'm--', alpha = 0.5)

# set x-axis properties
ax1.set_xlabel('days after hatching')
ax1.set_xlim([0, 95])
plt.xticks(np.arange(0, 95, 10))

adjustFigAspect(fig,aspect=.5)
plt.show()
```

Reading global simulation output

EPISIM Simulator's built-in data export functionality was used to export global properties such as total number of cells and eye radius at every 100 simulation steps in a csv file format (5.2). These data are parsed in Python for further analysis (5.14).

Listing 5.2: Example excerpt of output from EPISIM Simulator data export file containing global simulation data.

```
Episim Simulation Run on 4/23/19 4:19 PM;
Data-Export-Name:;
Global_Sim_Data;
Data-Export-Description:;
;
sim step no;Total Cell Number;Proliferative Cell Number;Differentiated Cell Number;Eye_Radius;
0;2268.0;527.0;1741.0;100.0;
100;3316.0;704.0;2612.0;124.75;
200;4649.0;825.0;3824.0;149.75;
300;6163.0;943.0;5220.0;174.75;
```

Source Code 5.14: Code used to read global simulation data as in 5.2.

```python
# -*- coding: utf-8 -*-
import csv
from itertools import islice

def getGlobalSimData(filepath):

    # names of headers in csv file
    hSimstep    = "sim step no"
```

```
hTotalCells = "Total Cell Number"
hProlCells  = "Proliferative Cell Number"
hDiffCells  = "Differentiated Cell Number"
hRadius1    = 'Eye_Radius'

# initialize arrays for data
simSteps        = []
totalCells      = []
prolCells       = []
diffCells       = []
eyeRadius       = []

# Open the file once to find all header rows
with open(filepath) as f:
    # Find out how many rows need to be skipped
    rowskip = 0
    for line in f.readlines():
        rowskip += 1
        if hSimstep in line: # line with field names was found
            header = line.split(';')
            break

# re-open file while skipping header rows
csvfile = islice(open(filepath, 'r'), rowskip, None) #Skip the first rowskip header rows
reader = csv.DictReader(csvfile, delimiter=';', fieldnames = header)
for item in reader: # iterates over rows
    # headers as indices for the columns
    simSteps.append(int(item[hSimstep]))
    totalCells.append(float(item[hTotalCells]))
    prolCells.append(float(item[hProlCells]))
    diffCells.append(float(item[hDiffCells]))
    eyeRadius.append(float(item[hRadius1]))

return [simSteps, eyeRadius, totalCells, prolCells, diffCells]
```

Reading cell-individual simulation output

EPISIM Simulator's built-in data export functionality was used to export cell-individual properties such as coordinates in 3D space, clonal identity, or age at every 100 simulation steps in a csv file format. A simulation typically ran for over 2000 simulation steps, and the total number of cells was in the order of 10^5 by the end of the simulation. Thus, the files frequently surpassed 10^6 lines. An excerpt of the first 8 lines of an example simulation is shown in 5.3.

Due to the file length, the files were first pre-processed with the functions in 5.15; these functions split the file into smaller chunks pertaining to the same simulation step. Then, the output csv files were processed by the function in 5.16 to extract meaningful data for plotting.

Listing 5.3: Example excerpt of output from EPISIM Simulator data export file containing coordinates and various properties of each cell for a defined number of simulation steps. Each row in the document shows values for one cell. The simulation step number is omitted in all but the first cell counted at each step. Before a new simulation step, there is a blank row.

```
Episim Simulation Run on 4/23/19 4:19 PM;
Data-Export-Name:;
Cell_Sim_Data;
Data-Export-Description:;
;
sim step no;X-Coordinate;Y-Coordinate;Z-Coordinate;ClonalID;Clone_R;Clone_G;Clone_B;InitialX;Clonal_ID2;Clone_R2;Clone_G2;Cl
    ↪ one_B2;xAtLateInd;Clonal_ID3;Clone_R3;Clone_G3;Clone_B3;xAtLateInd2;Eye_Radius;Time Since Division;Age;avg_overlap;Brd
    ↪ U;IdU;Division Interval;Division Events;cum_avg_overlap;
0;52.06402923841428;935.6969046625985;1076.9302664297222;1.250957409E9;243.0;81.0;83.0;52.06402923841428;-1.0;0.0;0.0;0.0;-9
    ↪ 9.0;-1.0;0.0;0.0;0.0;-99.0;100.0;100001.0;1.0;0.8266580088413829;0.0;0.0;-99.0;-99.0;0.0;
;51.81025754307236;1103.3935344059055;1092.284711160693;1.250957412E9;146.0;152.0;157.0;51.81025754307236;-1.0;0.0;0.0;0.0;-
    ↪ 99.0;-1.0;0.0;0.0;0.0;-99.0;100.0;100001.0;1.0;1.1088381625246584;0.0;0.0;-99.0;-99.0;0.0;
```

Source Code 5.15: Code used to pre-process simulation data like in 5.3.

```python
# -*- coding: utf-8 -*-
import csv
import os
from itertools import islice
import shutil

inputdir = os.path.dirname(__file__) # script folder path
csvtemp  = "csvtemp" # sub-directory in script folder path
csvfile = "CoordinatesClones.csv" # filename of EPISIM Simulator output file
hSimstep  = "sim step no" # first header in the csv file

def splitIntoSubfiles(filename, pattern, tempDir):
    # Open the file once to find all the rows pertaining to one simstep
    with open(filename) as f:
        # Skip the first few header rows
        rowskip = 0
        for line in f.readlines():
            rowskip += 1
            if hSimstep in line: # line with field names was found
                header = line.split(';')
                break
        f.close()
        # Open f at the header, skipping previous rows
        f          = islice(open(filename), rowskip, None)
        ph         = csv.DictReader(f, delimiter=';', fieldnames = header)
        idx        = 0
        simstep     = next(ph)[hSimstep]
        currentStep = simstep

    # Open the file again to start copying the rows into a new file
    with open(filename) as f:
        f      = islice(open(filename, 'r'), rowskip, None)
        dreader = csv.DictReader(f, delimiter=';', fieldnames = header)

        # Create an initial CSV writer
        foupath = os.path.join(tempDir, filename + pattern.format(idx))
        out     = open(foupath, 'w', newline='')
        a       = csv.DictWriter(out, fieldnames = header, delimiter=';')
        a.writeheader()

        for item in dreader:
            # Get current simstep
            if item[hSimstep] != '':
                idx     = int(float(item[hSimstep]))
                simstep = item[hSimstep]
            # Create a new CSV writer if the simstep changed
            if simstep != currentStep:
                currentStep = simstep
                foupath     = os.path.join(tempDir, filename + pattern.format(idx))
                out         = open(foupath, 'w', newline='')
                a           = csv.DictWriter(out, fieldnames = header, delimiter=';')
                a.writeheader()
                a.writerow(item)
                continue
            else:
                a.writerow(item)

def deleteCopyFile(filelist):
    for the_file in filelist:
        file_path = os.path.join(the_file)
        try:
            if os.path.isfile(file_path):
                os.unlink(file_path)
        except Exception:
            print("Error")

def getFilesFromSim(filename, scriptDir, tempDir):
    foldername = os.path.join(scriptDir)
    originalFile = os.path.join(foldername + os.sep + filename)
    copyFile = os.path.join(foldername + os.sep  + tempDir + os.sep + filename)
    shutil.copyfile(originalFile, copyFile)
    splitIntoSubfiles(copyFile, 'simstep_{0:04d}.csv', tempDir)
    deleteCopyFile([copyFile])

getFilesFromSim(csvfile, inputdir, csvtemp)
```

Source Code 5.16: Code used to read data output by 5.15.

```
# -*- coding: utf-8 -*-
import csv

def getCSVdata(absFilePath):

    # names of headers in csv file
    hSimstep = "sim step no"
    hX       = 'X-Coordinate'
    hY       = 'Y-Coordinate'
    hZ       = 'Z-Coordinate'

    hCloneID = ['ClonalID', 'Clonal_ID2', 'Clonal_ID3']
    hCloneR  = ['Clone_R', 'Clone_R2', 'Clone_R3']
    hCloneG  = ['Clone_G', 'Clone_G2', 'Clone_G3']
    hCloneB  = ['Clone_B', 'Clone_B2', 'Clone_B3']
    hInitX   = ['InitialX', 'xAtLateInd', 'xAtLateInd2']

    hRadius  = 'Eye_Radius'
    hAgeDiv  = 'Time Since Division'
```

```
hOverlap = 'avg_overlap'
hBrdU    = 'BrdU'
hIdU     = 'IdU'
hDivIntv = 'Division Interval'
hDivEven = 'Division Events'

#new data exports
hCumAvgOv = 'cum_avg_overlap'

# initialize dicts for data
# 0 - first induction, 1 - second induction, 2 - third induction
nInd = 3
initX    = [{}, {}, {}]
cellNum  = [{}, {}, {}]
IDtoTup  = [{}, {}, {}]
preIndID = [[], [], []]

cellXYZ     = [{}, {}, {}]
cellAgeDiv  = [{}, {}, {}]
cellAge     = [{}, {}, {}]
cellOverlap = [{}, {}, {}]
cellBrdU    = [{}, {}, {}]
cellIdU     = [{}, {}, {}]
divInterval = [{}, {}, {}]
divEvents   = [{}, {}, {}]

cumAvgOvl   = [{}, {}, {}]

with open(absFilePath) as csvfile:
    reader = csv.DictReader(csvfile, delimiter=";")
    for item in reader:
        if item[hSimstep] != '':
            simStep = int(item[hSimstep])

        eyeRadius = float(item[hRadius])

        # build induction-specific dicts
        for induction in range(0,nInd):
            indID = item[hCloneID[induction]]

            IndTup = (
                        float(item[hCloneR[induction]]),
                        float(item[hCloneG[induction]]),
                        float(item[hCloneB[induction]])
                     )

            try:
                cellXYZ[induction][indID].append(
                                                  (
                                                  float(item[hX]),
                                                  float(item[hY]),
                                                  float(item[hZ])
                                                  )
                                                 )

                cellNum[induction][indID] += 1
                initX[induction][indID].append(float(item[hInitX[induction]]))
```

```
                    cellAgeDiv[induction][indID].append(float(item[hAgeDiv]))
                    cellOverlap[induction][indID].append(float(item[hOverlap]))
                    cellBrdU[induction][indID].append(float(item[hBrdU]))
                    cellIdU[induction][indID].append(float(item[hIdU]))
                    divInterval[induction][indID].append(float(item[hDivIntv]))
                    cumAvgOvl[induction][indID].append(float(item[hCumAvgOv]))
                    divEvents[induction][indID].append(float(item[hDivEven]))

                except KeyError: # first time getting this clone

                    IDtoTup[induction][indID]     = (IndTup)
                    cellXYZ[induction][indID]     = [(float(item[hX]),float(item[hY]),float(item[hZ]))]
                    cellNum[induction][indID]     = 1
                    initX[induction][indID]       = [float(item[hInitX[induction]])]
                    cellAgeDiv[induction][indID]  = [float(item[hAgeDiv])]
                    cellOverlap[induction][indID] = [float(item[hOverlap])]
                    cellBrdU[induction][indID]    = [float(item[hBrdU])]
                    cellIdU[induction][indID]     = [float(item[hIdU])]
                    divInterval[induction][indID] = [float(item[hDivIntv])]
                    cumAvgOvl[induction][indID] = [float(item[hCumAvgOv])]
                    divEvents[induction][indID]   = [float(item[hDivEven])]

                    # Group cells that were differentiated prior to induction
                    if (float(item[hInitX[induction]])) > 75 and IndTup == (0.0, 168.0, 21.0):
                        preIndID[induction].append(indID)

            return_data =  [simStep, eyeRadius, IDtoTup, cellXYZ, initX, preIndID, cellNum, cellAgeDiv, cellAge, cellOverlap,
            ↪   cellBrdU, cellIdU, divInterval, divEvents, cumAvgOvl]
        return return_data
```

Histogram of cell cycle intervals

To plot histograms of cell division intervals, I first used the scripts in 5.15 and 5.16, and stored the result of the latter in the variable `cell_data`. This variable is given as input to the script 5.17. Since simulation parameters were not stored in the same data export output file, the value for $t_{\mathrm{cellCycle}}$ had to be input manually into the script variable `minimum_cellcycle`. This was needed to remove cells from the histogram that undergo their very first division in the simulation, which was allowed to be shorter than the minimum threshold.

Source Code 5.17: Code used to generate the plot in Figure 2.6 panel C.

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import numpy as np

def plotHistDivisionInterval(cellDataForSimStep, induction = 0, filterOverlap = 0, binwidth=6, mincc=24):

    divInterval = cellDataForSimStep[12][induction]
    cumAvgOvl   = cellDataForSimStep[14][induction]

    X = []

    for clone in divInterval:
        idx = 0
        for interval in divInterval[clone]:
            if float(interval) > mincc-1: # exclude first cell division in simulation
                mean_overlap = cumAvgOvl[clone][idx]/interval
                if mean_overlap > filterOverlap: # exclude cells that never divided
                    X.append(float(interval))
            idx += 1

    # define data range
    minInt = int(min(X))
    maxInt = int(max(X))

    # histogram of the data
    n, bins, patches = plt.hist(X, bins=range(minInt, maxInt + binwidth, binwidth),
                       density=True, facecolor='g', alpha=0.75)

    # print useful metrics
    print("mean: " + str(np.mean(X)))
    print("median: " + str(np.median(X)))
    print("standard deviation: " + str(np.std(X)))
    print("max value: " + str(maxInt))
    print(len(X))

    plt.xlabel('Interval between cell divisions [h]')
    plt.ylabel('Density')
    plt.axis([0, 510, 0, 1.1*n.max()])
    plt.grid(True)

    plt.show()

# user-defined values
minimum_cellcycle = 24 # parameter t_cellCycle that was used in the simulation

# cell_data is the result from the function "getCSVdata" used in a previous step
plotHistDivisionInterval(cell_data, mincc=minimum_cellcycle)
```

3D plot of cell age

The scripts in 5.15 and 5.16 were used to pre-process and extract data for a given simulation step, which was stored in the variable `cell_data`. This variable was passed as the first argument to the function in 5.18; modifying the values of the parameters `elev` and `azim` allowed to rotate the plot in 3D.

Source Code 5.18: Code used to generate the 3D age plots in Figures 2.7 panel C, and 2.8 C and D rightmost image.

```
# -*- coding: utf-8 -*-
from mpl_toolkits.mplot3d import Axes3D
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np

def plotAge3D(cell_data,induction = 0, elev = 0, azim = 0, cell_size = 16, center = (50 , 1025 , 1025), def_val = 100000):
    '''
    3D plot of all the cells' ages defined as simulation steps elapsed since the last cell division. The default value initialized at simulation start is 'def_val = 100000'; this value is subtracted if a cell exceeds this value. 'cell_size' is not the actual cell size in the simulation but a plotting parameter for the scatterplot function. 'center' is the center coordinates of the simulated retina
    '''
    fig = plt.figure()
    fig.set_size_inches(10,10)
    ax = fig.add_subplot(111, projection='3d')

    cellXYZ = cell_data[3][induction]
    age     = cell_data[7][induction]

    X = []
    C = []

    # loop first through divEvents to get max value
    maxvalue = 0
    for clone in age:
        ageList = age[clone]
        for cell in ageList:
            # Correct for default value
            if cell >= def_val:
                cell = cell - def_val
            maxvalue = max(maxvalue, cell)

    grain = 100
    cmap  = mpl.cm.magma(np.linspace(0, 1, grain))

    for clone in age:
        idx = 0
        for cellValue in age[clone]:
            # Correct for default value
            if cellValue >= def_val:
                cellValue = cellValue - def_val

            # Get index corresponding to position in colormap
            if cellValue > 0:
                dataColor = (cellValue/maxvalue) * grain
            else:
                dataColor = 0

            cidx = int(dataColor)
            # Correct index for largest value
            if cidx >= grain:
                cidx = grain - 1

            cc = cmap[cidx][:-1] # get the appropriate color for cell cc
            cellCoords = cellXYZ[clone][idx]

            C.append(cc)
            X.append(cellCoords)

            idx += 1

    # Plot as a 3D scatterplot
    X = np.array(X)
    C = np.array(C)
    ax.scatter(X[:,0], X[:,1], X[:,2], s = cell_size, c = C, lw=0)

    axcb = fig.add_axes([0.85, 0.3, 0.015, 0.15]) # Colorbar axes [left, bottom, width, height]
    # Set the colormap and norm to correspond to the data
    cmap = mpl.cm.magma
    norm = mpl.colors.Normalize(vmin=0, vmax=maxvalue)
    cb1 = mpl.colorbar.ColorbarBase(axcb, cmap=cmap, norm=norm, orientation="vertical")
    numticks = 3
    ticks = list(range(0, int(maxvalue), int(maxvalue/numticks)))
    ticks.append(maxvalue)
    cb1.set_ticks(ticks)
    ax.axis('off')
    ax.view_init(elev, azim)
```

Plot of cell population over time

Global simulation data were obtained from EPISIM Simulator's built-in data export functionality and parsed by 5.12, which stored the result in `global_data`. This result was used for plotting with the script in 5.19.

Source Code 5.19: Code used to generate the plot in Figure 2.10.

```
# -*- coding: utf-8 -*-
import pylab as pl
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

# parameters for exponential decay function
p = 30**(-1)
d = 33**(-1)

def exponential_decay(t, y): return (p-d) * y

sol = solve_ivp(exponential_decay,  # function
                [0, 1000],          # t_0 and t_end
                [497],              # y_0
                max_step = 1)       # stepsize; small=smoother

fig = pl.figure()
ax  = fig.add_subplot(111, title = 'growth of proliferative cell population')
ax.plot(sol.t, sol.y[0], 'k', label='numerical solution to Eq 12', linewidth = 2)

'''Area Development'''
simsteps  = [] # global_data index 0
radius    = [] # global_data index 1
cellNum   = [] # global_data index 2
prolCells = [] # global_data index 3
eyeArea   = [] # global_data calculated below

# parameters for calculation
cell_radius = 3.5
w_cmz = 25
d_ol = 0.85 # tolerated overlap between cells
simStepRange = 11 # total number of measurements extracted from simulation

```

```
for i in range(simStepRange):
    if global_data[0][i] <= 1400:
        simsteps.append(global_data[0][i])
        prolCells.append(global_data[3][i])
        cellNum.append( (2*global_data[1][i]*w_cmz)/((d_ol*cell_radius)**2) )

ax.plot(simsteps, prolCells, 'g', label='proliferative cells in simulation', linewidth = 2)
ax.plot(simsteps, cellNum, ':', color='grey', label='number of cells at ideal density in CMZ (Eq 15)', linewidth = 2)
plt.xlabel('hours')
plt.ylabel('number of cells')
ax.legend(loc='upper left',prop={'size':12})
```

Heatmap of average cell overlap against radial position on hemisphere

Global simulation data were obtained from EPISIM Simulator's built-in data export functionality and parsed by 5.12, which stored the result in `global_data`. This result was used for plotting with the script in 5.20.

Source Code 5.20: Code used to generate the plot in Figure 2.12 panels A′-B′′′′.

```
# -*- coding: utf-8 -*-
import pylab as pl
import random

def plotOverlapHeatMap(cell_data, induction = 0, cell_radius = 3.5):
    cellXYZ     = cell_data[3][induction] # cell coordinates
    cellOverlap = cell_data[9][induction] # average overlap
    fig = pl.figure()
    ax  = fig.add_subplot(111, title = 'Average Overlap')

    X = []
    Y = []
    counter = 0
    keys = list(cellOverlap.keys())
    random.shuffle(keys) # randomize the order, so only a sample is plotted
    for clone in keys:
        xyz = cellXYZ[clone]
        for tup in xyz:
            if counter < 10**5: # don't plot more than 10^5 cells (script stalls)
                idx = xyz.index(tup)
                # normalize overlap to cell diameter
                normOverlap = float(cellOverlap[clone][idx])/(2*cell_radius)
                # normalize radial position to hemisphere edge
                Xposition = float(tup[0]-50)/(cell_radius+cell_data[1])
                counter = counter + 1
                # Exclude plotting new cells with no initialized value!
                if not (normOverlap == 0 and float(tup[0]-50) < 25):
                    X.append(Xposition)
                    Y.append(normOverlap)
            else:
                break
    print("Reached " + str(counter) + " cells")

    fig.subplots_adjust(hspace=0.5, right=0.97)
    ax.set_ylabel("Normalized average overlap")
    ax.set_xlabel("Normalized Radius (Peripheral -> Central)")
    minx = cell_radius/(cell_radius+cell_data[1])
    ax.set_xlim([-minx, 1+minx])
    ax.set_ylim([0, 0.5])
    hb = ax.hexbin(X, Y,
            gridsize=100,
            bins='log',
            extent=[-minx, 1+minx, 0, 0.5],
            cmap = "bone_r")
    cb = fig.colorbar(hb, ax=ax)
    cb.set_label('log10(N)')
```

Plot of cell area density and eye radius over time

Global simulation data were obtained from EPISIM Simulator's built-in data export functionality and parsed by 5.12, which stored the result in `global_data`. This result was used for plotting with the script in 5.21 and 5.22.

Source Code 5.21: Code used to generate the plot in Figure 2.12 panels C′-C″.

```
# -*- coding: utf-8 -*-
import pylab as pl
import math

def plotCellVsEyeArea(global_data, cell_radius = 3.5):
    simsteps = [] # global_data index 0
    eyeArea  = []
    cellArea = []
    ratio    = []

    for i in range(len(global_data[0])):
        if global_data[0][i] <= 1400:
            simsteps.append(global_data[0][i])
            eyeArea.append( 2*math.pi*float(global_data[1][i])**2 )
            cellArea.append( (math.pi*(0.85*cell_radius)**2)*float(global_data[2][i]) )
            ratio.append(cellArea[i]/eyeArea[i])

    fig2 = pl.figure()
    ax2  = fig2.add_subplot(111)
    ax2.set_ylabel("Cell/Eye Area Ratio")
    ax2.set_xlabel("Simsteps")
    ax2.set_xlim([0, 1400])
    ax2.set_ylim([0, 10])
    ax2.plot(simsteps, ratio, 'g', linewidth = 3)
```

Source Code 5.22: Code used to generate the plot in Figure 2.12 panels D′-D″.

```
# -*- coding: utf-8 -*-
import pylab as pl

def plotEyeRadius(global_data, cell_radius = 3.5):
    fig = pl.figure()
    ax2 = fig.add_subplot(111)
    ax2.set_ylabel("Eye Radius")
    ax2.set_xlabel("Simsteps")
    ax2.set_xlim([0, 1400])
    ax2.set_ylim([0, 600])
    ax2.plot(global_data[0], global_data[1], 'g', linewidth = 3)
```

Parameter scan of p_{div} and $t_{\text{cellCycle}}$

Global simulation data were obtained from EPISIM Simulator's built-in data export functionality for several simulation runs with varying parameter values; simulations were run for 200 simulation steps. The average eye radius growth rate was calculated and entered into 5.23 for plotting. Data were interpolated and smoothed to reduce step-wise transitions in the plot.

Source Code 5.23: Code used to generate the plot in Figure 2.12 panels C′-C″.

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
import numpy as np
from scipy.interpolate import griddata
from scipy import ndimage

# growth rates obtained from simulation runs were added to this grid
z = [#1     5     10    15    20    25    30    35    40    45    50    mincc
    2.741, 1.811, 1.104, 0.844, 0.624, 0.529, 0.433, 0.375, 0.315, 0.284, 0.257, #  5
    1.285, 1.135, 0.851, 0.665, 0.540, 0.449, 0.387, 0.336, 0.299, 0.268, 0.244, # 10
    0.861, 0.724, 0.607, 0.515, 0.440, 0.385, 0.334, 0.307, 0.284, 0.245, 0.228, # 15
    0.59 , 0.514, 0.468, 0.416, 0.376, 0.334, 0.302, 0.269, 0.247, 0.233, 0.216, # 20
    0.486, 0.410, 0.379, 0.351, 0.321, 0.296, 0.270, 0.242, 0.231, 0.215, 0.194, # 25
    0.416, 0.352, 0.329, 0.302, 0.280, 0.260, 0.240, 0.218, 0.210, 0.191, 0.179, # 30
    0.354, 0.298, 0.258, 0.265, 0.249, 0.234, 0.224, 0.206, 0.193, 0.181, 0.171, # 35
    0.291, 0.247, 0.236, 0.227, 0.215, 0.210, 0.192, 0.185, 0.176, 0.166, 0.159, # 40
    0.293, 0.246, 0.230, 0.218, 0.208, 0.196, 0.184, 0.177, 0.164, 0.159, 0.148, # 45
    0.235, 0.193, 0.186, 0.182, 0.176, 0.169, 0.165, 0.157, 0.153, 0.146, 0.138, # 50
    ] #growth rate [um/h]
x = [1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]*10 # pdiv
y = [5]*11 + [10]*11 + [15]*11 +[20]*11 + [25]*11 + [30]*11 + [35]*11 + [40]*11 + [45]*11 + [50]*11 # mincc

# additional simulation runs
z = z + [0.208, 0.212, 0.193, 0.181, 0.178, 0.162, 0.149, 0.141, 0.133, 0.127, 0.123, 0.121, 0.126, 0.136, 0.150, 0.165, 0.184,
↪  0.220, 0.194, 0.166]
x = x + [60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 50, 40, 30, 20, 10, 1, 5, 25]
y = y + [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 60, 60, 60, 60, 60, 60, 55, 55]

x = np.asarray(x)
y = np.asarray(y)
z = np.asarray(z)

# interpolate data
xi = np.linspace(x.min(),x.max(),13)
yi = np.linspace(y.min(),y.max(),13)
zi = griddata((x, y), z, (xi[None,:],yi[:,None]), method='cubic') #, method='nearest'
# smooth with a gaussian kernel and small radius
for i in range(15):
    zi = ndimage.gaussian_filter(zi, 0.5)

```

```
# interpolate again on smoothed data
xi2 = np.linspace(x.min(),x.max(),100)
yi2 = np.linspace(y.min(),y.max(),100)
X, Y = np.meshgrid(xi, yi)
zi2 = griddata((X.flatten(), Y.flatten()), zi.flatten(), (xi2[None,:],yi2[:,None]), method='cubic') #, method='nearest'
# smooth again with a median and gaussian kernel, this time using larger radius
for i in range(25):
    z2 = ndimage.median_filter(zi2, 3)
    z2 = ndimage.gaussian_filter(zi2, 3)

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('p_divE-1')
ax.set_ylabel('mincc')
ax.set_xlim([x.min(),x.max()])
ax.set_ylim([y.min(),y.max()])
contours = plt.contour(np.flip(xi2, axis = 0), yi2, z2, levels = [0.23,0.49], colors='black')
plt.clabel(contours, inline=True, fontsize=8)
plt.contourf(np.flip(xi2, axis = 0), yi2, z2, 250, origin='lower', cmap='Spectral', alpha=1)
plt.colorbar();
```

2D histogram of cell division intervals and normalised average overlap

The scripts in 5.15 and 5.16 were used to pre-process and extract data for a given simulation step, which was stored in the variable `cell_data`. This variable was passed as the first argument to the function in the following script:

Source Code 5.24: Code used to generate the plot in Figure 2.16.

```
# -*- coding: utf-8 -*-
import pylab as pl

def plotHist2D(cell_data, induction = 0, cell_radius = 3.5):
    divIntvl   = cell_data[12][induction]
    cumAvgOvl  = cell_data[14][induction]
    X = []
    Y = []
    for clone in divIntvl:
        cells = divIntvl[clone]
        idx = 0
        for cell in cells:
            # Adjust for cells with no initialized value!
            if divIntvl[clone][idx] > 23 and cumAvgOvl[clone][idx] > -1:
                mean_avg_overlap = cumAvgOvl[clone][idx]/divIntvl[clone][idx]
                if mean_avg_overlap >=0 and mean_avg_overlap <= 1:
                    X.append(divIntvl[clone][idx])
                    Y.append(cumAvgOvl[clone][idx]/divIntvl[clone][idx])
            idx += 1

    fig = pl.figure()
    ax  = fig.add_subplot(111)
    fig.subplots_adjust(hspace=0.5, right=0.97)
    ax.set_ylabel("Last cell division interval")
    ax.set_xlabel("Mean normalized average overlap before last cell division")
    ax.set_ylim([0, 500])
    ax.set_xlim([0.1, 0.27])
    hb = ax.hexbin(Y, X, gridsize=50, bins='log', cmap = "bone_r")
    cb = fig.colorbar(hb, ax=ax)
    cb.set_label('log10(N)')
```

Generation of 3D clonal plots from simulated data

For a given simulation step, cell-individual data were parsed and extracted (Section 5.7.2) into the variable `cell_data`. These data were used to generate simulated segmentation plots (Script 5.25) and clonal plots (Script 5.26) . Clonal plots containing only lineages emerging from the original ventral sector were done by pre-filtering the input data (Script 5.27); pre-filtering required pre-processed and parsed cell-individual data of the very first simulation step (`cell_data_0`) as well as the simulation step to be plotted (`cell_data_t`).

Source Code 5.25: Code used to generate the plots in Figure 2.17 panels C-D.

```
# -*- coding: utf-8 -*-
import random
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

def plotSampleClones(cell_data, samplesize = 0.1, induction = 0, unique_color = False, elev = 0, azim = 0, center = (50 , 1025 , 1025)):
    cellXYZ = cell_data[3][induction] # cell coordinates grouped by clone
    preindID = cell_data[5][induction] # ids of pre-induction clones
    prunedXYZ = {}
    for key in cellXYZ.keys():
        if key not in preindID:
            prunedXYZ[key] = cellXYZ[key]
    try:
        sampleXYZ = random.sample(list(prunedXYZ),int(round(samplesize*len(prunedXYZ))))
    except ValueError: # samplesize > 1
        pass
    # Parameters of plot
    fig1 = plt.figure()
    fig1.set_size_inches(10,10)
    ax1   = fig1.add_subplot(111, projection='3d')
    xlim = [50, 1050]
    ylim = [0, 2050]
    zlim = [0, 2050]
    ax1.view_init(elev, azim)
    ax1.set_xlim(xlim)
    ax1.set_ylim(ylim)
    ax1.set_zlim(zlim)
    ax1.axis('off')
    # Auxiliary plot elements
    offset = 20
    r_off = 1000 + offset
    c1   = (center[0], center[1]+r_off, center[2]+r_off)
    c2   = (center[0], center[1]+r_off, center[2]-r_off)
    c3   = (center[0], center[1]-r_off, center[2]+r_off)
    c4    = (center[0], center[1]-r_off, center[2]-r_off)
    ax1.scatter(c1[0], c1[1], c1[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax1.scatter(c2[0], c2[1], c2[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax1.scatter(c3[0], c3[1], c3[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax1.scatter(c4[0], c4[1], c4[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    # Loop through each element == each clone
    clr = [0,0,0]
    t    = 0
    R    = []
    CR   = []
    for this_clone in prunedXYZ:
        if this_clone in sampleXYZ:
            if unique_color: # cycle through unique RGB values
                if t % 255 != 0:
                    clr = [clr[0] + 1/255.0, 0, clr[2]]
                else:
                    clr = [1/255.0, 0, clr[2] + 1/255.0]
                for coords in prunedXYZ[this_clone]:
                    R.append(coords)
                    CR.append(tuple(clr))
                t = t + 1
            else: # paint all sampled clones black
                for coords in prunedXYZ[this_clone]:
                    R.append(coords)
                    CR.append((0, 0, 0))
        else: # Plot white markers for clones out of sample size
            for coords in prunedXYZ[this_clone]:
                R.append(coords)
                CR.append((1, 1, 1))
    R   = np.array(R)
    CR  = np.array(CR)
    try:
        # s = 2.0 for correctly-sized cells
        ax1.scatter(R[:,0], R[:,1], R[:,2], s = 2.0, c = CR, lw=0, alpha = 1, antialiased = False)
    except IndexError: # empty array
        pass
```

Source Code 5.26: Code used to generate the plots in Figure 2.20 panels A′-A″, Figure 2.22 panels C′ and E′, Figure 2.24 panels B′-C′‴, and Figure 2.25 panels A-C.

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

def plot3DcloneVisualization(cell_data, induction = 0, preind = False, elev = 0, azim = 0, cell_size = 0.75, center = (50 , 1025 ,
↪   1025)):
    cellXYZ = cell_data[3][induction]
    colorFromID = cell_data[2][induction]
    X = []
    C = []
    fig = plt.figure()
    ax  = fig.add_subplot(111, projection='3d')
    fig.set_size_inches(10,10)
    if preind == True: # plot the pre-induction retina
        for clone in cellXYZ:
            xyz = cellXYZ[clone]
            col = colorFromID[clone]
            for tup in xyz:
                C.append((col[0]/255, col[1]/255, col[2]/255))
                X.append(tup)
    else: # do not plot the pre-induction retina
        preindID = cell_data[5][induction]
        for clone in cellXYZ:
            if not clone in preindID:
                xyz = cellXYZ[clone]
                col = colorFromID[clone]
                for tup in xyz:
                    C.append((col[0]/255, col[1]/255, col[2]/255))
                    X.append(tup)
    X = np.array(X)
    C = np.array(C)
    ax.scatter(X[:,0], X[:,1], X[:,2], s = cell_size, c = C, lw=0, alpha = 1)
    if not preind: # Plot scalebar of 100 um in figure center
        scale100 = np.asarray([(center[0],center[1]-50,center[2]), (center[0],center[1]+50,center[2])])
        cbar = (1.0/255,0,250.0/255)
        ax.plot(scale100[:,0], scale100[:,1], scale100[:,2], c = cbar, linewidth=1, alpha = 1, antialiased = False)
    # Delimiters at figure margins to normalize figure size and allow automatic cropping
    offset = 20
    r_off = 1000 + offset
    c1   = (center[0], center[1]+r_off, center[2]+r_off)
    c2   = (center[0], center[1]+r_off, center[2]-r_off)
    c3   = (center[0], center[1]-r_off, center[2]+r_off)
    c4   = (center[0], center[1]-r_off, center[2]-r_off)
    ax.scatter(c1[0], c1[1], c1[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.scatter(c2[0], c2[1], c2[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.scatter(c3[0], c3[1], c3[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.scatter(c4[0], c4[1], c4[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.axis('off')
    ax.view_init(elev, azim)
```

Source Code 5.27: Code used to generate the plots of ventral-originating clones in Figure 2.24 panels B′-B′′′′. As input, the code requires the extracted simulation data from simulation step 0 in addition to the current simulation step.

```
# -*- coding: utf-8 -*-
import math
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

def plot_ventral_clones(cell_data_0, cell_data_t, induction = 0, init_eye_radius = 100, cell_radius = 3.5, elev = 0, azim = 0,
↪   cell_size = 0.75, center = (50 , 1025 , 1025)):
    cellXYZ_0   = cell_data_0[3][0] # cell coordinates at simulation step 0
    colorFromID = cell_data_t[2][induction] # clonal ID at current simulation step
    cellXYZ_t   = cell_data_t[3][induction] # cell coordinates at current simulation step
    preindID    = cell_data_t[5][induction] # pre-induction retina at current simulation step
    ventrallist = []
    big_radius  = init_eye_radius + cell_radius
    for key in cellXYZ_0.keys():
        xcoord  = cellXYZ_0[key][0][0]
        ycoord  = cellXYZ_0[key][0][1]
        h       = (big_radius)-(xcoord-center[0])
        r_small = math.sqrt(h*(2*big_radius-h))
        sin45   = math.sqrt(2.0)/2.0
        if (ycoord < (center[1] - sin45*r_small-cell_radius)):
            ventrallist.append(key)
    prunedXYZ = {}
    for key in cellXYZ_t.keys():
        if key in ventrallist:
            prunedXYZ[key] = cellXYZ_t[key]
    X = []
    C = []
    fig = plt.figure()
    ax  = fig.add_subplot(111, projection='3d')
    fig.set_size_inches(10,10)
    for clone in prunedXYZ:
        if not clone in preindID:
            xyz = prunedXYZ[clone]
            col = colorFromID[clone]
            for tup in xyz:
                C.append((col[0]/255, col[1]/255, col[2]/255))
                X.append(tup)
    X = np.array(X)
    C = np.array(C)
    ax.scatter(X[:,0], X[:,1], X[:,2], s = cell_size, c = C, lw=0, alpha = 1)
    # Plot scalebar of 100 um in figure center
    scale100 = np.asarray([(center[0],center[1]-50,center[2]), (center[0],center[1]+50,center[2])])
    cbar = (1.0/255,0,250.0/255)
    ax.plot(scale100[:,0], scale100[:,1], scale100[:,2], c = cbar, linewidth=1, alpha = 1, antialiased = False)
    # Delimiters at figure margins to normalize figure size and allow automatic cropping
    offset = 20
    r_off = 1000 + offset
    c1   = (center[0], center[1]+r_off, center[2]+r_off)
    c2   = (center[0], center[1]+r_off, center[2]-r_off)
    c3   = (center[0], center[1]-r_off, center[2]+r_off)
    c4   = (center[0], center[1]-r_off, center[2]-r_off)
    ax.scatter(c1[0], c1[1], c1[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.scatter(c2[0], c2[1], c2[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.scatter(c3[0], c3[1], c3[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.scatter(c4[0], c4[1], c4[2], c = 'k', lw = 0, alpha = 0.5, antialiased = False)
    ax.axis('off')
    ax.view_init(elev, azim)
```

Parsing of ROI properties

This script parses and processes data output by Appendix script 5.8, and saves it to a csv file. The user must define an input directory, name and absolute path to the input csv file, and optionally an output directory (by default the input directory is used).

Source Code 5.28: ode used to parse data obtained by running ImageJ Macro 5.8.

```
# -*- coding: utf-8 -*-
import csv
import os

# User-defined variables
inputdir = os.path.dirname(__file__) # script folder path
```

```python
csvfile = "ROI_coords.csv" # filename of the file output by ImageJ
outputdir = inputdir
path = inputdir + "/" + csvfile

def readROIfile(absPath):
    with open(absPath, 'r') as f:
        dreader = csv.reader(f, delimiter=' ')
        imagedata = []
        outputList = []
        rowcount = 0
        curROI = 0
        for item in dreader:
            if rowcount == 0: # image dimensions
                stringlist = item
                imagedata.append( float(stringlist[0]) ) # image width
                imagedata.append( float(stringlist[1]) ) # image height
            elif "ROI" in item: # general ROI dimensions
                stringlist = item
                curROI = int(stringlist[1])
                outputList.append([ [float(stringlist[2]), # index 0 ROI boundary x
                                     float(stringlist[3]), # index 1 ROI boundary y
                                     float(stringlist[4]), # index 2 ROI width
                                     float(stringlist[5]), # index 3 ROI height
                                     float(stringlist[6])  # index 4 ROI area
                                    ],
                                    [] # empty list
                                    ])
            else: # individual ROI coordinates
                stringlist = item
                outputList[curROI][1].append(
                                              (
                                                float(stringlist[1]), # x coord
                                                float(stringlist[2])  # y coord
                                              )
                                            )
            rowcount += 1
    return([ imagedata, outputList ])

def filterLateROI(roiData):
    imageHeight = roiData[0][1]
    roiList = roiData[1]
    roiLengthList = []
    for subList in roiList:
        upperBoundY = subList[0][1]/imageHeight
        roiHeight = subList[0][3]/imageHeight
        if upperBoundY < 0.2: # exclude patches whose upper y-coord is after 20% of radius
            roiLengthList.append(roiHeight)
    return(roiLengthList)

def filterShortROI(roiData):
    imageHeight = roiData[0][1]
    roiList = roiData[1]
    roiWidthList = []
    for subList in roiList:
        roiHeight = subList[0][3]/imageHeight
        roiWidth = subList[0][2] # absolute ROI width (in pixels)
        if roiHeight >= 0.2: # exclude clones shorter than 20% of normalized radius
        roiWidthList.append(roiWidth)
    return(roiWidthList)

def writeResultToCSV(outputdata, outputdir, namestring):
    # Create an initial CSV writer
    out = open(outputdir + os.sep + namestring, 'w', newline='')
    a   = csv.writer(out, delimiter=',')
    for item in outputdata:
        a.writerow([item])

roiData = readROIfile(path)
width_dist = filterShortROI(roiData)
height_dist = filterLateROI(roiData)
writeResultToCSV(width_dist, outputdir, "width_dist.csv")
writeResultToCSV(height_dist, outputdir, "height_dist.csv")
```

Counting proportion of persistent and terminated clones

Scripts in 5.15 and 5.16 were used to pre-process and extract data to the variable `cell_data`. This variable was passed to the following script to obtain the proportion of terminated and persistent clones (defined as a clone that retains at least one cell in the CMZ at the timepoint of analysis) for each row of the virtual CMZ.

Source Code 5.29: Code used to count the number of persistent and terminated clones per CMZ row.

```
# -*- coding: utf-8 -*-

def getPersistentTerminatedClones(cell_data, induction=1, cmz = 25, center = (50, 1050, 1050)):
    cellXYZ  = cell_data[3][induction]
    initialX = cell_data[4][induction]
    preindID = cell_data[5][induction]
    # count 5 rows of cells in cmz in 5-um bins
    persistent = [0, 0, 0, 0, 0, 0]
    terminated = [0, 0, 0, 0, 0, 0]
    for key in cellXYZ.keys():
        if key in preindID: # ignore clones that were initially differentiated
            continue
        else: # find if ArCoS still contains cells in CMZ
            cmzcell = False
            icell   = 0
            ncells  = len(cellXYZ[key])
            while not cmzcell:
                # x-coordinate of ith cell of thisarcos <= center[0] + cmz
                if cellXYZ[key][icell][0] <= (center[0] + cmz):
                    cmzcell = True
                    break
                if icell == (ncells - 1):
                    break
                else:
                    icell = icell + 1
            initX = initialX[key][0] - center[0]
            binnum = int(initX/(cmz/5))
            if cmzcell:
                persistent[binnum] += 1
            else:
                terminated[binnum] += 1
    return [persistent, terminated]
```

Counting proportion of persistent and terminated clones

Input data are from the appendix of Nguyen [2018].

Source Code 5.30: Code used to plot the graphic in Figure 2.36 panel A‴.

```
# -*- coding: utf-8 -*-
import csv
import matplotlib.pylab as plt
import numpy as np

def readCSVfile(foldername, filename):
    # Open the file
    with open(foldername + filename, 'r') as f:
        # read out data as in "Dictionary" format
        dreader = csv.DictReader(f, delimiter=',')
        # generate empty lists where data will be filled
        timepoints  = []
        CMZ_dorsal  = []
        CMZ_ventral = []
        NVR_dorsal  = []
        NVR_ventral = []
        measurements = [ CMZ_dorsal, CMZ_ventral, NVR_dorsal, NVR_ventral ]
        headerlist   = [ "CMZ dorsal", "CMZ ventral", "NVR dorsal", "NVR ventral" ]
        # loop through each row in the file
        for row in dreader:
            if row['dph'] == "":
            # skip empty lines
                print('empty line!')
            # all other cases are measured points
            else:
                timepoints.append(int(row['dph']))
                for i in range(len(measurements)):
                    try:
                        measurements[i].append(float(row[headerlist[i]]))
                    except ValueError:
                        print("missing value: " + str(row[headerlist[i]]))
                        measurements[i].append(np.nan)
        # return a list where element 0 is the center, element 1 the fissure,
        # and element 2 the list of point coordinates
        return [ timepoints, measurements ]

def plotData(data):
    fig, ax1 = plt.subplots()
    ax1.patch.set_facecolor((1.0,1.0,1.0))
    ax1.set_xlabel('days after hatching')
    ax1.set_ylabel('[um]')
    colors = [[102/255,0,102/255], [1,102/255,1], [0,102/255,0], [102/255,1,102/255]]
    labels = ["CMZ dorsal", "CMZ ventral", "NVR dorsal", "NVR ventral"]
    idx = 0
```

```
for i in data[1]:
    ax1.scatter(data[0], i,
                c = np.array(colors[idx]),
                label=labels[idx],
                s = 25, alpha = 0.5)
    # split the original data along set points to make piecewise linear fit
    breakpoints = [0, 11, 35, 99] # pre-defined breakpoints
    for k in range(len(breakpoints)-1):
        # find indices of elements above breakpoint k and below breakpoint k+1
        i1 = np.where(np.asarray(data[0]) >= breakpoints[k])[0]
        i2 = np.where(np.asarray(data[0]) <= breakpoints[k+1])[0]
        # pad smaller array to same length
        if len(i1) < len(i2):
            i1 = np.append(i1, np.ones(len(i2)-len(i1))*-99)
        else:
            i2 = np.append(i2, np.ones(len(i1)-len(i2))*-99)
        # get the indices of elements that are in both arrays
        combined = np.intersect1d(i1, i2, assume_unique=False)
        combined = combined.astype(np.int64)
        # create sub-arrays that lie within the bounds given by breakpoints k and k+1
        sub_x = [ data[0][j] for j in combined ]
        sub_y = [ i[j] for j in combined ]
        # find and remove elements where one of the arrays is NaN
        iis = np.isfinite(sub_x) & np.isfinite(sub_y) # find NaN indices
        nanidx = np.where(iis == False) # get NaN indices
        sub_x = np.delete(sub_x, nanidx) # delete NaN entries
        sub_y = np.delete(sub_y, nanidx) # delete NaN entries
        # create a linear fit with polyfit
        fit = np.polyfit(sub_x, sub_y, 1) # [slope, y-intercept]
        print(labels[idx] + ":  slope = " + str(round(fit[0],1)) + " y-intercept: " + str(round(fit[1])))
        # create corresponding y values for unique values of sub_x and plot them
        plt.plot(np.unique(sub_x), np.poly1d(fit)(np.unique(sub_x)),
                 c = np.array(colors[idx]), linestyle="--", linewidth = 3, alpha = 0.8)
    idx += 1
    print("")

ax1.legend(loc='upper left',prop={'size':11})
ax1.set_xlim([0, 95])
ax1.set_ylim([0, 450])
plt.xticks(np.arange(0, 95, 10))
plt.yticks(np.arange(0, 451, 50))
```

5.7.3 R scripts

Distribution of cell cycle intervals

The built-in R function **dnbinom**(x, y, p) calculates the probability that a number of failures x occurs before y-th success in a sequence of Bernoulli trials, for which the probability of individual success is p. For the cell cycle model described in section 2.1.3, y is p_{div}, and a success is a cell division event.

Source Code 5.31: Code used to generate the plot in Figure 2.6 panel B.

```
ntrials = 10000
failures = 1:100
success = 1
# cell cycle parameters
prob = 1/26 # p_div
mincc = 24  # t_cellcycle
# Evaluate and plot the function in black
dist <- dnbinom(x=failures, size=success, prob=prob)
ylim <- c(0, 0.6)
xlim <- c(0, 100)
plot(dist, type='l', xlim = xlim, ylim = ylim, xlab = "t[h]", ylab = "density")
# Sum all values less than "mincc" to the function's value at "mincc". Plot the result in red.
d <- sum(dist[1:mincc])
e <- c(rep(0, mincc-1), sum(dist[1:mincc]), dist[mincc+1 : length(dist)])
lines(e, col="red")
```

Rugplots

Data from the output of the Python script in Appendix Section 5.7.2 or ImageJ macro in Appendix Section 5.7.1 were imported via RStudio's manual "Import Dataset" functionality and plotted with the following script.

Source Code 5.32: Code used to generate the rugplot in Figure 2.19 panels B-C, Figure 2.28 panel B, and Figure 2.32 panel B‴.

```
data <- c(external_input_data)
k <- unlist(data)
plot(density(k, bw = "SJ"), col="gray", lwd=2)
rug(jitter(k))
```

Violin plots

Data from the output of the ImageJ macro in Appendix Section 5.7.1 were imported via RStudio's manual "Import Dataset" functionality and plotted with the following script.

Source Code 5.33: Code used to generate the violin plots in Figure 2.19 panel D, and Figure 2.25 panels F′-F⁗.

```
library(ggplot2)
var <- imagej_output
```

```
cat <- c(rep("category", length(imagej_output)))
data <-  data.frame("Cat"=cat, "Var"=var)
p <- ggplot(data, aes(x=Cat, y=Var))+geom_violin(adjust=2)+scale_y_continuous(breaks=seq(0,9,1))
p + geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.3)
print(p)
```

Stacked bar plots

Data obtained via Python (Appendix Section 5.7.2) or by measurements in ImageJ were manually input as a vector in R and plotted with the following script.

Source Code 5.34: Code used to generate the stacked bar plots in Figure 2.20 panels B and F.

```
library(ggplot2)
library(reshape2)
persistent_data <- c(input_data_persistent)
terminated_data <- c(input_data_terminated)
bar_exp <- as.table(rbind(persistent_data, terminated_data), colnames=colnames)
datm <- melt(cbind(bar_exp))
p <- ggplot(datm, aes(x = Var2, y = value, fill = Var1)) + geom_bar(position = "fill", stat = "identity")
print(p)
```

Clone width over radius

The script prompts the user to select several files. Input are files obtained from the ArCoS Analysis plugin in ImageJ.

Source Code 5.35: Code used to generate the plots in Figure 2.22 panels C″ and E″.

```
require("matrixStats")
# This function joins data sets by column and pads with NA
cbind.fill <- function(...){
  nm <- list(...)
  nm <- lapply(nm, as.matrix)
  n  <- max(sapply(nm, nrow) )
  do.call(cbind, lapply( nm,
                    function (x) {
                      rbind( x, matrix(, n-nrow(x), ncol(x) ) )
                    }))
}
# standard error function
stderr <- function(x) sqrt( var(x,na.rm=TRUE)/length( na.omit(x) ) )
# custom mean function that excludes rows with less than threshold datapoints
lowNxcludeMean <- function(x, threshold) {
  means <- rep(NA, dim(x)[1])
  for (rowIdx in 1:dim(x)[1]) {
    n <- dim(x)[2]-sum(is.na(x[rowIdx,]))
    if (is.na(n) == TRUE) { n <- 0 }
    if ((n > threshold-1) == TRUE) {
      means[rowIdx] = mean(x[rowIdx,], na.rm=T)
    }
  }
  return(means)
}
```

```r
# custom stdev function that excludes rows with less than threshold datapoints
lowNxcludeStDev <- function(x, threshold) {
  # initialize empty containers
  means  <- rep(NA, dim(x)[1])
  stdevs <- rep(NA, dim(x)[1])
  # loop through dataset and store value if threshold exceeded, otherwise store NA
  for (rowIdx in 1:dim(x)[1]) {
    ssqd <- 0
    n    <- dim(x)[2]-sum(is.na(x[rowIdx,]))
    if (is.na(n) == TRUE) { n <- 0 }
    if ((n > threshold-1) == TRUE) {
      means[rowIdx] = mean(x[rowIdx,], na.rm=T)
    }
    # loop through each row element to get sum of squared difference
    if ((n > threshold-1) == TRUE) {
      for (elem in 1:length(x[rowIdx,])){
        if (is.na(x[rowIdx,elem]) == FALSE) {
          ssqd <- ssqd + (x[rowIdx,elem]-means[rowIdx])**2
        }
      }
      stdevs[rowIdx] <- sqrt(ssqd/n)
    }
  }
  return(stdevs)
}
# function to get several csv files, merge them, and pre-process
getdata <- function(default.search, factor = c(1)){
  data  <- c(NA) # Initialize vector with placeholder NA
  files <- choose.files(default.search) # Ask for csv files
  # Loop through files
  for (file in 1:length(files)) {
    datatemp = read.table(files[file], header = TRUE, row.names = 1, sep = ",")
    data   <- cbind.fill(data, datatemp) # Join all files in succession
  }
  data <- data[,-1]       # remove placeholder NA
  data[data == 0.0] <- NA # replace 0 values by NA
  # Loop through rows in inverse order to trim bottom 0 rows
  for ( row in (dim(data)[1]:1) ) {
    rowtest <- unname( unlist(data[row,]) )
    rowcomp <- as.numeric( rep( NA, dim(data)[2] ) )
    if ( isTRUE( all.equal (rowtest, rowcomp) ) ) {
      data <- data[-row,]
    }
    else {
      break
    }
  }
  # remove columns with less than 10% non-zero entries (small patches)
  to.remove <- c()
  for (col in 1:dim(data)[2]) {
    sumnas <- sum(is.na(data[,col]))
    if ((dim(data)[1] - sumnas) < 0.1*dim(data)[1]) {
      to.remove <- c(to.remove, col)
    }
  }
  if (length(to.remove) > 0) {
    data <- data[,-to.remove]
  }
  return(data)
}
# function for building data frame from input data
process <- function(data) {
  means    <- lowNxcludeMean(data, 0) # exclude points with less than 0 ArCoS
  stdev    <- lowNxcludeStDev(data, 0)
  sterr    <- apply(as.matrix(data[,1:dim(data)[2]]), 1, stderr)
  df       <- as.data.frame(means)
  df$sds   <- stdev
  df$sterr <- sterr
  return(df)
}
# plotting function
plotdata <- function(xs, df, reset = FALSE) {
  if (reset == TRUE) {
    i <<- 1
  }
  op    <- par(mar=c(5, 6, 4, 2) + 0.1)
  cols1 <- c("black", "blue", "red")
  cols2 <- c("gray", "lightblue", "salmon")
  ylim <- c(0, 12/3.6)
  xlim <- c(0, 350)
  ylab <- "Mean Clone Contribution \n [% circumference]"
  xlab <- "Radial Coordinate [um]"
  # standard error: df$sterr; 95% CI: 1.96*df$sterr
  if (i == 1) {
    plot(xs, df$means, type = "l", lwd = 2,
         ylim = ylim, ylab = ylab, xlim = xlim, xlab = xlab, yaxp = c(0/3.6, 12/3.6, 12), cex.axis=0.5)
    par(op)
    lines(c(xs,-xs),c(df$means + 1.96*df$sterr, df$means - 1.96*df$sterr), col = cols2[i])
  }
  else {
    points(df$means, type = "l", lwd = 2, col=cols1[i])
    lines(df$means + 1.96*df$sterr, col = cols2[i])
    lines(df$means - 1.96*df$sterr, col = cols2[i])
  }
  i <<- i + 1
}

default.search <- paste(getwd())
i <- 1

#get the data
data1 <- getdata(default.search) # plotted in black
data3 <- getdata(default.search) # plotted in blue
data4 <- getdata(default.search) # plotted in red
df1 <- process(data1)
df3 <- process(data3)
df4 <- process(data4)

# plot mean value in region of stable lineages
start = 240
end   = 445
xs1 <- as.numeric(1:(end-start+1))*1.6678 # um/pixel of experimental data
plotdata(xs1, df1[start:end,], reset=T)
start = 400
```

```
end   = 750
xs3 <- as.numeric(1:(end-start+1))*0.8427 # um/pixel of simulated data
xs4 <- as.numeric(1:(end-start+1))*0.8427
plotdata(xs3, df3[start:end,])
plotdata(xs4, df4[start:end,])

# calculate p-values using tsum.test from BSDA package
library(BSDA)
mu1 = mean(data1[240:445,], na.rm=T)*360/100
sd1 = sd(data1[240:445,], na.rm=T)*360/100
n1  = dim(data1)[2]
mu2 = mean(data3[400:750,], na.rm=T)*360/100
sd2 = sd(data3[400:750,], na.rm=T)*360/100
n2  = dim(data3)[2]
mu3 = mean(data4[400:750,], na.rm=T)*360/100
sd3 = sd(data4[400:750,], na.rm=T)*360/100
n3  = dim(data4)[2]
print(tsum.test(mean.x = mu1, s.x = sd1, n.x = n1, mean.y = mu2, s.y = sd2, n.y = n2))
print(tsum.test(mean.x = mu1, s.x = sd1, n.x = n1, mean.y = mu3, s.y = sd3, n.y = n3))
print(tsum.test(mean.x = mu2, s.x = sd2, n.x = n2, mean.y = mu3, s.y = sd3, n.y = n3))
```

Eye dimensions relative to body length

The ratio of eye axes to body length measured in ImageJ was plotted using the following script.

Source Code 5.36: Code used to generate the plots in Figure 5.9 panel D.

```
eye_height_rel_body_length_smoc <- c(0.055,0.060,0.065,0.065,0.061,0.059,0.066,0.063,0.061,0.056,0.057,0.060,0.064,0.069,0.068,0.059,0.059,0.065,0.064,0.061)
eye_width_rel_body_length_smoc <- c(0.059,0.060,0.062,0.067,0.065,0.062,0.068,0.063,0.066,0.060,0.059,0.061,0.063,0.065,0.067,0.061,0.060,0.065,0.067,0.061)
eye_height_rel_body_length_wt <- c(0.068,0.064,0.060,0.066,0.069,0.063,0.060,0.071,0.071,0.056,0.070,0.066,0.060,0.073,0.070,0.063,0.063,0.067,0.070,0.064)
eye_width_rel_body_length_wt <- c(0.069,0.068,0.069,0.070,0.069,0.068,0.068,0.068,0.069,0.066,0.073,0.066,0.069,0.076,0.070,0.066,0.066,0.070,0.071,0.073)
library(ggplot2)
var <- c(eye_height_rel_body_length_wt,
      eye_height_rel_body_length_smoc,
      eye_width_rel_body_length_wt,
      eye_width_rel_body_length_smoc)
cat <- c(rep("DV axis wt", length(eye_height_rel_body_length_wt)),
      rep("DV axis smoc", length(eye_height_rel_body_length_smoc)),
      rep("AP axis wt", length(eye_width_rel_body_length_wt)),
      rep("AP axis smoc", length(eye_width_rel_body_length_smoc)))
data <-  data.frame("Cat"=cat, "Var"=var)
p <- ggplot(data, aes(x=Cat, y=Var))+geom_violin(adjust=2)+scale_y_continuous(breaks=seq(0,9,1))
print(p
    + geom_jitter(shape=19, position=position_jitter(0.05), alpha=0.3, size=3)
    + ylim(0.05, 0.08) + ylab("eye dimension relative to body length") )
print(wilcox.test(eye_height_rel_body_length_wt, eye_height_rel_body_length_smoc)$p.val)
print(wilcox.test(eye_width_rel_body_length_wt, eye_width_rel_body_length_smoc)$p.val)
```

5.8 EPISIM implementation of the model

5.8.1 Java implementation in EPISIM Simulator

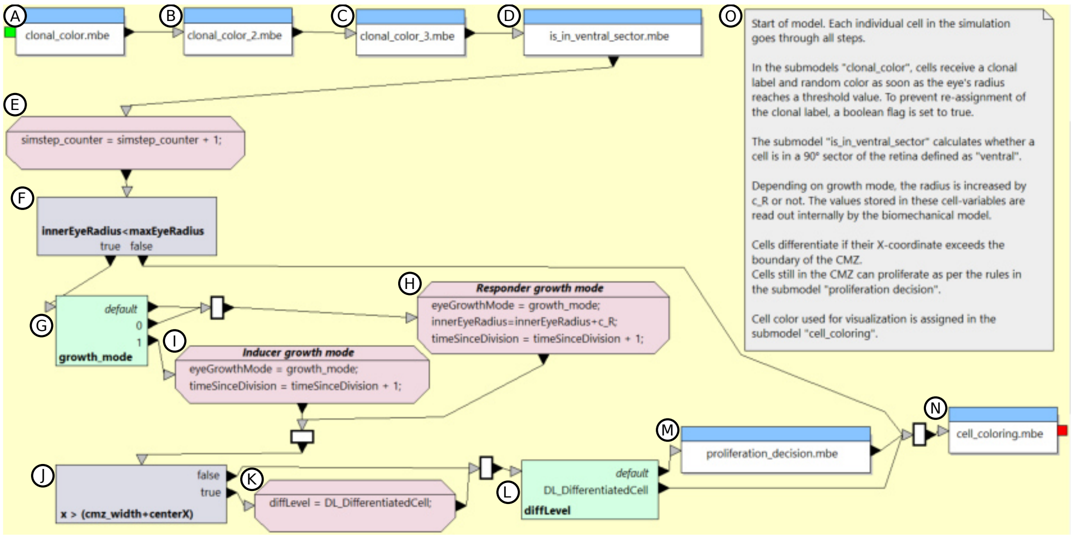
The Java implementation in EPISIM Simulator contains the biomechanical model implementation developed by Dr Thomas Sütterlin [Sütterlin et al., 2017], and extensions specific to this work such as the hemispherical geometry and growth rules that I developed in collaboration with Dr Thomas Sütterlin as described in the Chapter 2, Section 2.1. This code was embedded into the framework of EPISIM Simulator and was deposited in a GitLab repository [Sütterlin, 2019].

5.8.2 Graphical model implementation in EPISIM Modeller

EPISIM Modeller allows to code the cell-internal logic via a graphical modelling interface, which is subsequently compiled into a Java executable that can be read by EPISIM Simulator [Sütterlin et al., 2012]. This section presents screenshots of the EPISIM Modeller interface showing the implementation of the rules that were described in the Results (Chapter 2). For reasons of clarity, the model version published in Tsingos et al. [2019] (Chapter 2 Section 2.1.3, 2.2, and 2.3) is elucidated first, then differences in other model versions presented in this work are highlighted.

Model with feedback coupling of eye growth and cell proliferation

The model shown in this section corresponds to the published version in Tsingos et al. [2019], and was described in Chapter 2 Sections 2.1 and 2.3).



```
simstep_counter = simstep_counter + 1;
```

```
eyeGrowthMode = growth_mode;
innerEyeRadius=innerEyeRadius+c_R;
timeSinceDivision = timeSinceDivision + 1;
```

```
eyeGrowthMode = growth_mode;
timeSinceDivision = timeSinceDivision + 1;
```

```
diffLevel = DL_DifferentiatedCell;
```

Start of model. Each individual cell in the simulation goes through all steps.

In the submodels "clonal_color", cells receive a clonal label and random color as soon as the eye's radius reaches a threshold value. To prevent re-assignment of the clonal label, a boolean flag is set to true.

The submodel "is_in_ventral_sector" calculates whether a cell is in a 90° sector of the retina defined as "ventral".

Depending on growth mode, the radius is increased by c_R or not. The values stored in these cell-variables are read out internally by the biomechanical model.

Cells differentiate if their X-coordinate exceeds the boundary of the CMZ.
Cells still in the CMZ can proliferate as per the rules in the submodel "proliferation decision".

Cell color used for visualization is assigned in the submodel "cell_coloring".

Figure 5.12: Top-level model in the hierarchy.

A-C Link to submodel elements that initialise clonal identity (shown in Figures 5.13 and 5.14). **D** Link to submodel that determines the ventral sector of the simulated retina (shown in Figure 5.15). **E** Cell-internal counter for simulation steps; utility variable to trigger certain events in the simulation. **F** Condition that checks whether the simulation has attained a predefined maximum eye radius. When this point is reached, cells no longer divide and the eye radius no longer increases effectively "stopping" the simulation, but cell colouring (N) can still be altered in the EPISIM Simulator visualisation screen (this cannot be done if the simulation is stopped entirely). **G** Condition that checks the growth mode. By default and with variable value `growth_mode = 0`, the simulation runs with the responder growth mode (H). If `growth_mode = 1`, the simulation grows with the inducer growth mode (I). **H** Responder growth mode: The cell-internal variable for growth mode is set and the cell-internal value that tracks the eye radius is increased by c_R. The utility variable `timeSinceDivision` that is used to track cell age is increased. **I** Inducer growth mode: The cell-internal variable for growth mode is set and the utility variable `timeSinceDivision` that is used to track cell age is increased. **J** Condition that implements Equation 2.12. `centerX` is the x-coordinate of the eye globe. **K** Cell differentiation level is changed to a differentiated cell type `DL_DifferentiatedCell`. **L** Condition that checks cell differentiation level. **M** Non-differentiated cells (*i.e.* proliferative cells), proceed to the submodel that decides proliferation (shown in Figure 5.16). **N** Link to submodel that defines cell colour in the visualisation window of EPISIM Simulator (shown in Figure 5.22). **O** Explanatory note embedded into the modelling canvas.

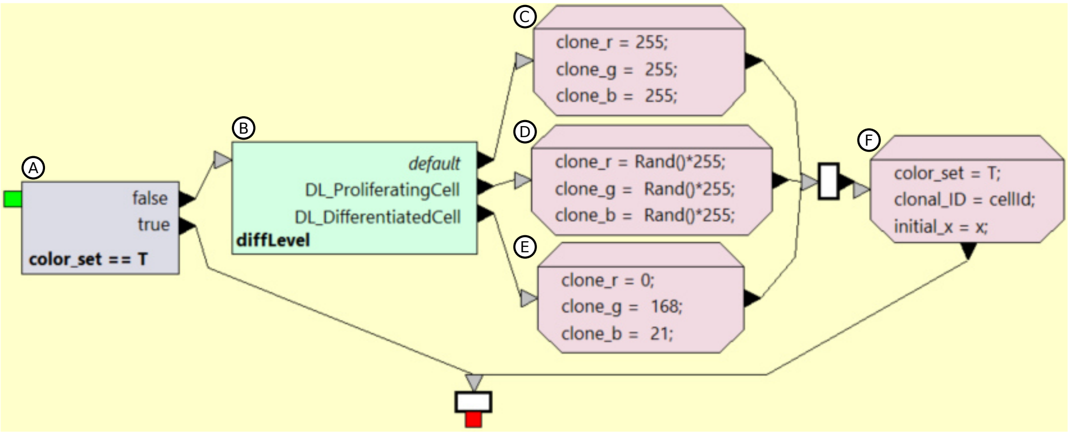


Figure 5.13: Submodel for defining clonal colour; corresponds to Figure 5.12 A.
A Check if the Boolean flag `colour_set` is set to True. By default it is set to False, meaning that branches B-G will be executed at the very first simulation step. **B** The differentiation level of cells is checked to decide which of the path branches will be taken. **C-E** The value of three variables is set that will be used to generate a cell colour profile in a later submodel (shown in Figure 5.22). C - Cells with undefined differentiation level; D - cells with proliferative differentiation level receive a random value distributed in the interval [0, 255] for each variable; E - cells with differentiated differentiation level. **F** The flag `colour_set` is set to True, such that the operations B-F are never executed again during the same simulation. A unique clonal ID is assigned to all cells. The current x-coordinate is stored in the variable `initial_x`.

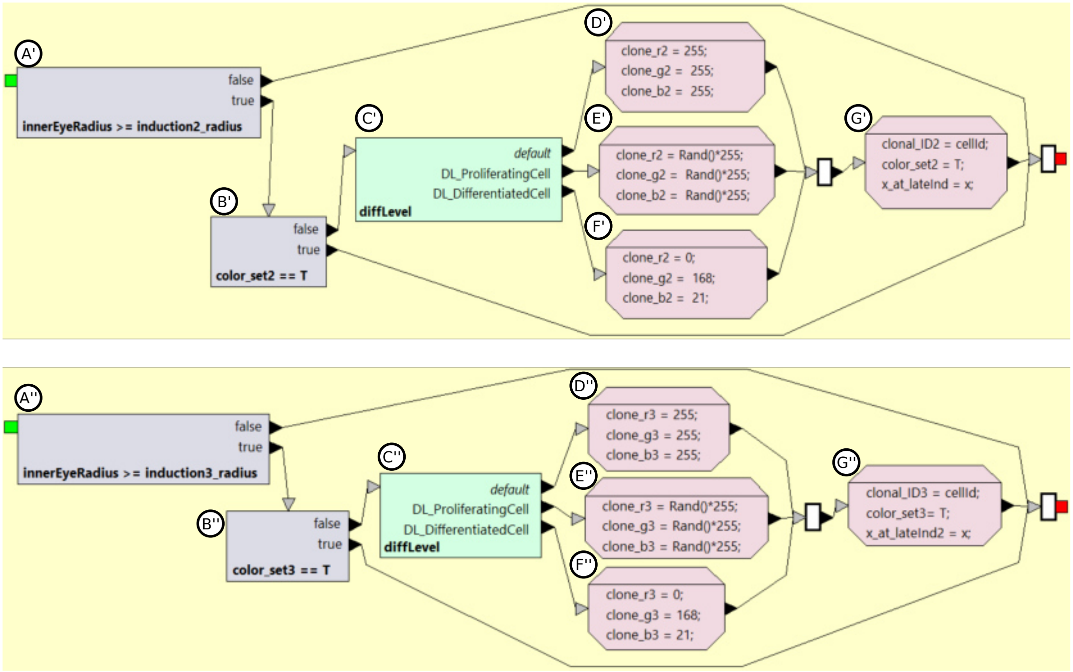


Figure 5.14: Submodels for defining clonal colour; correspond to Figure 5.12 B and C.
A'-A" The only difference to the submodel in Figure 5.13: The submodels are only executed if the eye radius exceeds a pre-defined threshold. **B'-G"** Constructed analogously to Figure 5.13 B-F.

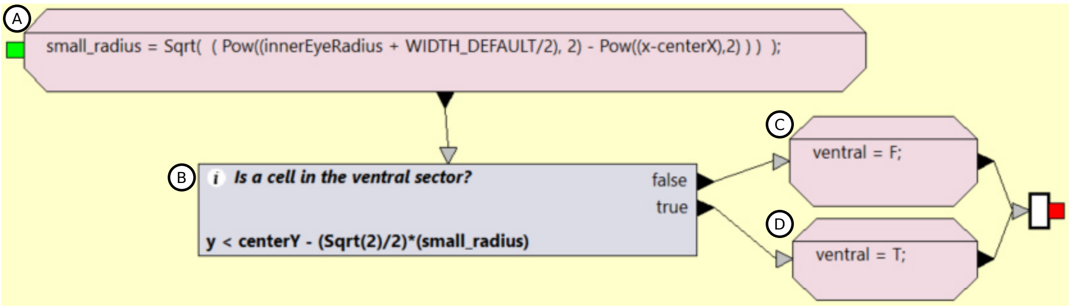
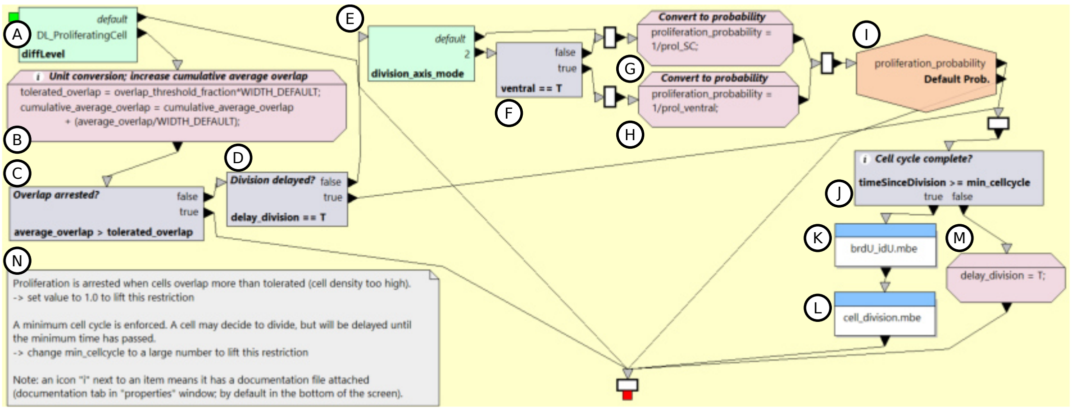

A
small_radius = Sqrt((Pow((innerEyeRadius + WIDTH_DEFAULT/2), 2) - Pow((x-centerX),2)));
B
i Is a cell in the ventral sector?
false
true
y < centerY - (Sqrt(2)/2)*(small_radius)
C
ventral = F;
D
ventral = T;

Figure 5.15: Submodel for defining the ventral sector; corresponds to Figure 5.12 D.

A Implementation of Equation 2.28. **B** Implementation of Equation 2.29. **C** Boolean flag `ventral` is set to False. **D** Boolean flag`ventral` is set to True.


A
default
DL_ProliferatingCell
diffLevel
B
i Unit conversion; increase cumulative average overlap
tolerated_overlap = overlap_threshold_fraction*WIDTH_DEFAULT;
cumulative_average_overlap = cumulative_average_overlap
+ (average_overlap/WIDTH_DEFAULT);
C
Overlap arrested?
false
true
average_overlap > tolerated_overlap
D
Division delayed?
false
true
delay_division == T
E
default
2
division_axis_mode
F
false
true
ventral == T
G
Convert to probability
proliferation_probability =
1/prol_SC;
H
Convert to probability
proliferation_probability =
1/prol_ventral;
I
proliferation_probability
Default Prob.
J
i Cell cycle complete?
timeSinceDivision >= min_cellcycle
true
false
K
brdU_idU.mbe
L
cell_division.mbe
M
delay_division = T;
N

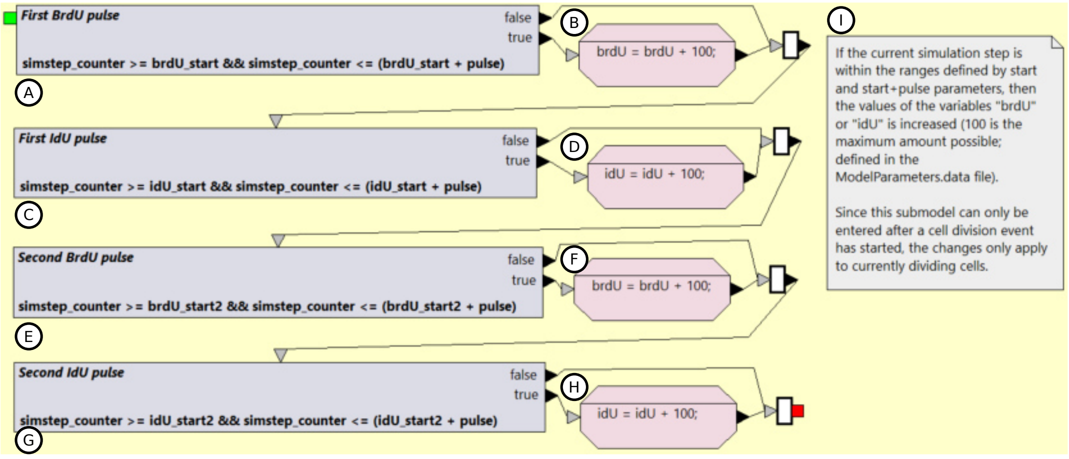
Proliferation is arrested when cells overlap more than tolerated (cell density too high).
-> set value to 1.0 to lift this restriction

A minimum cell cycle is enforced. A cell may decide to divide, but will be delayed until the minimum time has passed.
-> change min_cellcycle to a large number to lift this restriction

Note: an icon "i" next to an item means it has a documentation file attached (documentation tab in "properties" window; by default in the bottom of the screen).

Figure 5.16: Submodel for cell proliferation decision; corresponds to Figure 5.12 M.

A Check that only proliferative cells enter branch B-M. **B** Obtain absolute average overlap from normalised average overlap. Utility variable `cumulative_average_overlap` which is used for plotting is updated. **C** Implementation of Equation 2.18. **D** Check if boolean flag `delay_division` is True. **E** Select between no circumferential bias (default) or ventrally different behavior (`division_axis_mode = 2`). **F** Check if cell is in the ventral sector (flag set to True). **G** Default division probability p_{div} is calculated. **H** Ventral division probability $p_{\text{div\_ventral}}$ is calculated. **I** Stochastic decision node with probability p_{div} and $1 - p_{\text{div}}$. **J** Check if minimum cell cycle time $t_{\text{cellCycle}}$ has elapsed. **K** Submodel for thymidine analogue incorporation. **L** Submodel for cell division. **M** If $t_{\text{cellCycle}}$ steps have not elapsed, the flag `delay_division` is set to True. **N** Explanatory note embedded into the modelling canvas.

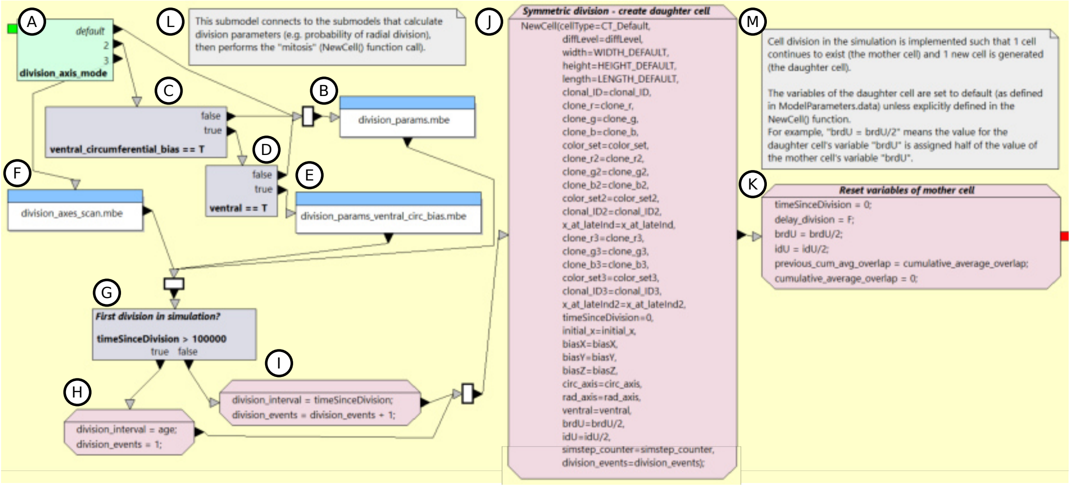

First BrdU pulse
false
true
simstep_counter >= brdU_start && simstep_counter <= (brdU_start + pulse)
A
B
brdU = brdU + 100;
I
First IdU pulse
false
true
simstep_counter >= idU_start && simstep_counter <= (idU_start + pulse)
C
D
idU = idU + 100;
Second BrdU pulse
false
true
simstep_counter >= brdU_start2 && simstep_counter <= (brdU_start2 + pulse)
E
F
brdU = brdU + 100;
Second IdU pulse
false
true
simstep_counter >= idU_start2 && simstep_counter <= (idU_start2 + pulse)
G
H
idU = idU + 100;

If the current simulation step is within the ranges defined by start and start+pulse parameters, then the values of the variables "brdU" or "idU" is increased (100 is the maximum amount possible; defined in the ModelParameters.data file).

Since this submodel can only be entered after a cell division event has started, the changes only apply to currently dividing cells.

Figure 5.17: Submodel for thymidine analogue incorporation; corresponds to Figure 5.16 K.

A The first BrdU pulse occurs within the range of simulation steps defined by the variables `brdU_start` and `brdU_end`. **B** The cell-internal variable for BrdU is increased. The variable saturates at 100 arbitrary units. **C-D** Constructed analogously to A-B, but incrementing a variable for 5-Iodo-2'-deoxyuridine (IdU) instead. **E-F** Constructed analogously to A-B, but different start and end variables. **G-H** Constructed analogously to C-D, but different start and end variables. **I** Explanatory note embedded into the modelling canvas.


A
default
2
3
division_axis_mode
L
J
C
false
true
ventral_circumferential_bias == T
B
division_params.mbe
D
false
true
ventral == T
E
division_params_ventral_circ_bias.mbe
F
division_axes_scan.mbe
G
First division in simulation?
timeSinceDivision > 100000
true
false
I
H
Symmetric division - create daughter cell
M
K
Reset variables of mother cell

This submodel connects to the submodels that calculate division parameters (e.g. probability of radial division), then performs the "mitosis" (NewCell() function call).

```
division_interval = timeSinceDivision;
division_events = division_events + 1;
```

```
division_interval = age;
division_events = 1;
```

```
NewCell(cellType=CT_Default,
        diffLevel=diffLevel,
        width=WIDTH_DEFAULT,
        height=HEIGHT_DEFAULT,
        length=LENGTH_DEFAULT,
        clonal_ID=clonal_ID,
        clone_r=clone_r,
        clone_g=clone_g,
        clone_b=clone_b,
        color_set=color_set,
        clone_r2=clone_r2,
        clone_g2=clone_g2,
        clone_b2=clone_b2,
        color_set2=color_set2,
        clonal_ID2=clonal_ID2,
        x_at_lateInd=x_at_lateInd,
        clone_r3=clone_r3,
        clone_g3=clone_g3,
        clone_b3=clone_b3,
        color_set3=color_set3,
        clonal_ID3=clonal_ID3,
        x_at_lateInd2=x_at_lateInd2,
        timeSinceDivision=0,
        initial_x=initial_x,
        biasX=biasX,
        biasY=biasY,
        biasZ=biasZ,
        circ_axis=circ_axis,
        rad_axis=rad_axis,
        ventral=ventral,
        brdU=brdU/2,
        idU=idU/2,
        simstep_counter=simstep_counter,
        division_events=division_events);
```

Cell division in the simulation is implemented such that 1 cell continues to exist (the mother cell) and 1 new cell is generated (the daughter cell).

The variables of the daughter cell are set to default (as defined in ModelParameters.data) unless explicitly defined in the NewCell() function.
For example, "brdU = brdU/2" means the value for the daughter cell's variable "brdU" is assigned half of the value of the mother cell's variable "brdU".

```
timeSinceDivision = 0;
delay_division = F;
brdU = brdU/2;
idU = idU/2;
previous_cum_avg_overlap = cumulative_average_overlap;
cumulative_average_overlap = 0;
```

Figure 5.18: Submodel for cell division; corresponds to Figure 5.16 L.

A Branch point for selecting division parameters. **B** Submodel setting default division parameters. **C-E** Branch for ventral circumferential bias in division orientation. C, D - Checking of Boolean flags. E - Setting of division parameters with circumferential bias. **F** Submodel allowing further variation of division parameters. **G-I** Setting of utility variables used for extracting data from simulation. **J** Mitosis. A new cell is created and its variables are set. **K** Variables of the currently simulated cell are altered. **L-M** Explanatory notes embedded into the modelling canvas.

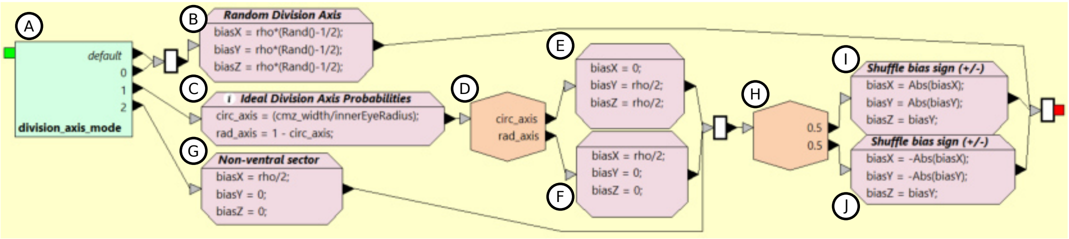


Figure 5.19: Submodel for default division parameters; corresponds to Figure 5.18 B.

A Branch point for selecting division parameters. **B** Randomly oriented division axis; implements Equation 2.11. **C** Ideally oriented division axis; implements Equations 2.26 and 2.27. **D** Uses values obtained in C for stochastic choice. **E** Implements Equation 2.22. **F-G** Implement Equation 2.21. **H-J** Stochastic choice with two equally likely outcomes to randomize the sign of division axis orientation bias.

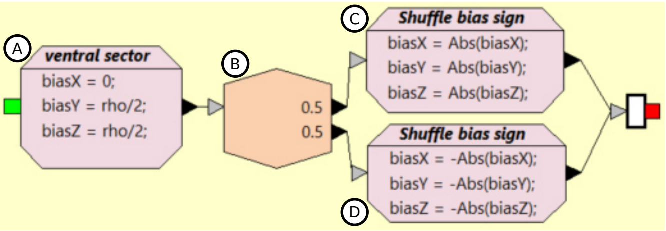


Figure 5.20: Submodel for ventral circumferential division parameters; corresponds to Figure 5.18 E.

A Implements Equation 2.22. **B-D** Stochastic choice with two equally likely outcomes to randomize the sign of division axis orientation bias.

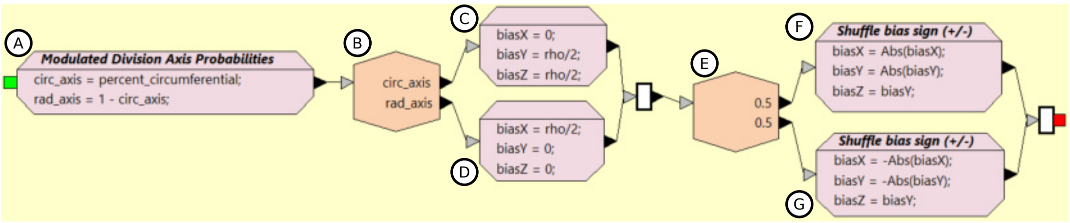


Figure 5.21: Submodel for modulating division parameters; corresponds to Figure 5.18 F.

A In this submodel, the probabilities for circumferential and radial division axis orientation are fixed according to the pre-defined parameter `percent_circumferential`. **D** Uses values obtained in A for stochastic choice. **E** Implements Equation 2.22. **D** Implements Equation 2.21. **E-G** Stochastic choice with two equally likely outcomes to randomize the sign of division axis orientation bias.

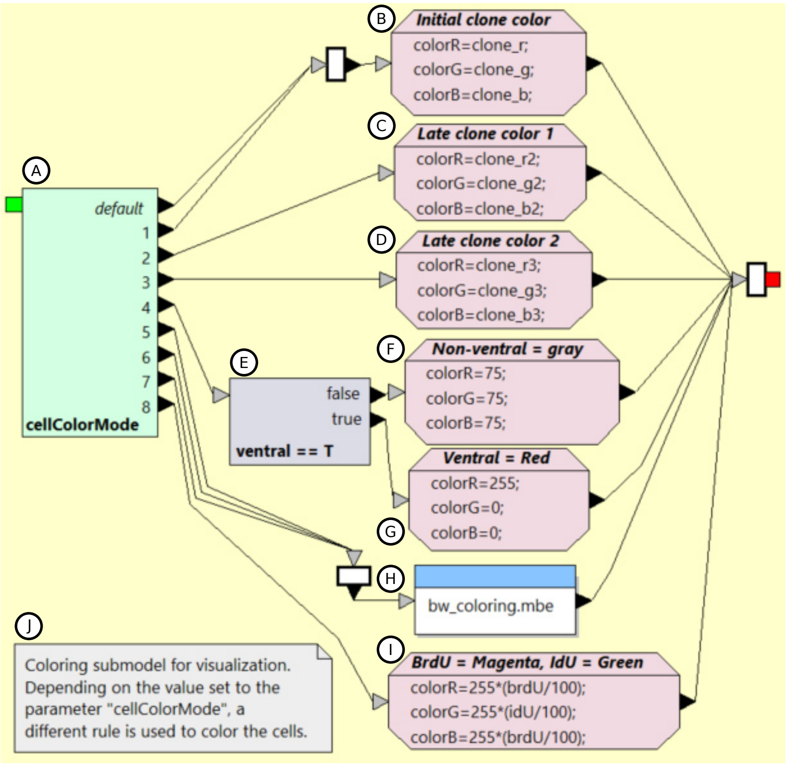


Figure 5.22: Submodel for choosing cell colour; corresponds to Figure 5.12 N.

A Branch decision point. The parameter `cellColorMode` can be changed at simulation runtime to change the colour of cells in the EPISIM Simulator visualisation window. **B-D** Clonal colour as defined in submodels in Figures 5.13 and 5.14. **E-G** Color according to ventral sector identity as defined in submodel in Figure 5.15. **H** Submodel to generate black and white clonal colouration. **I** BrdU and IdU colouration. **J** Explanatory notes embedded into the modelling canvas.

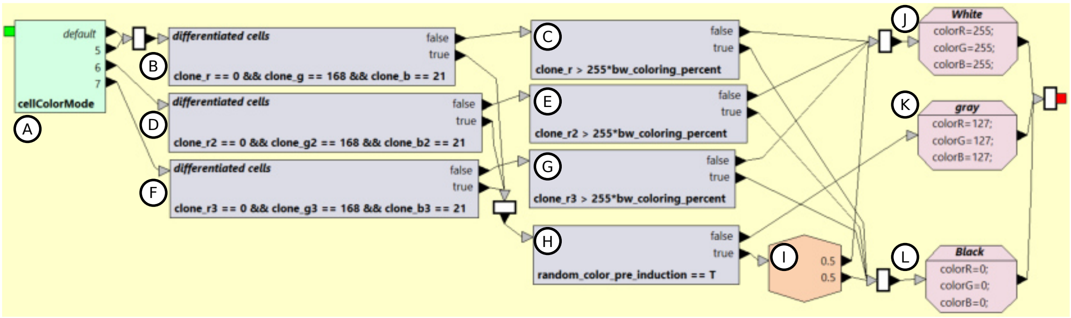

default
5
6
7
cellColorMode
A
B
differentiated cells
false
true
clone_r == 0 && clone_g == 168 && clone_b == 21
D
differentiated cells
false
true
clone_r2 == 0 && clone_g2 == 168 && clone_b2 == 21
F
differentiated cells
false
true
clone_r3 == 0 && clone_g3 == 168 && clone_b3 == 21
C
false
true
clone_r > 255*bw_coloring_percent
E
false
true
clone_r2 > 255*bw_coloring_percent
G
false
true
clone_r3 > 255*bw_coloring_percent
H
false
true
random_color_pre_induction == T
I
0.5
0.5
J
White
colorR=255;
colorG=255;
colorB=255;
K
gray
colorR=127;
colorG=127;
colorB=127;
L
Black
colorR=0;
colorG=0;
colorB=0;

Figure 5.23: Additional submodel for choosing cell colour; corresponds to Figure 5.22 H.

A Branch decision point analogous to Figure 5.22 A. **B** Clonal colour as defined in submodel in Figure 5.13 is used to discern pre-induction clones and induced clones. **C** Clonal colour as defined in submodel in Figure 5.13 is used to select a random sample of cells to colour in black or white. **D-G** Analogous to B-C, but using clonal colour as defined in submodel in Figure 5.14. **H-I** Randomize colour of pre-induction cells with a stochastic choice. **J** Definition of white cell colour. **K** Definition of gray cell colour that is used for pre-induction clones if H is False. **L** Definition of black cell colour.

Model without feedback coupling of eye growth and cell proliferation

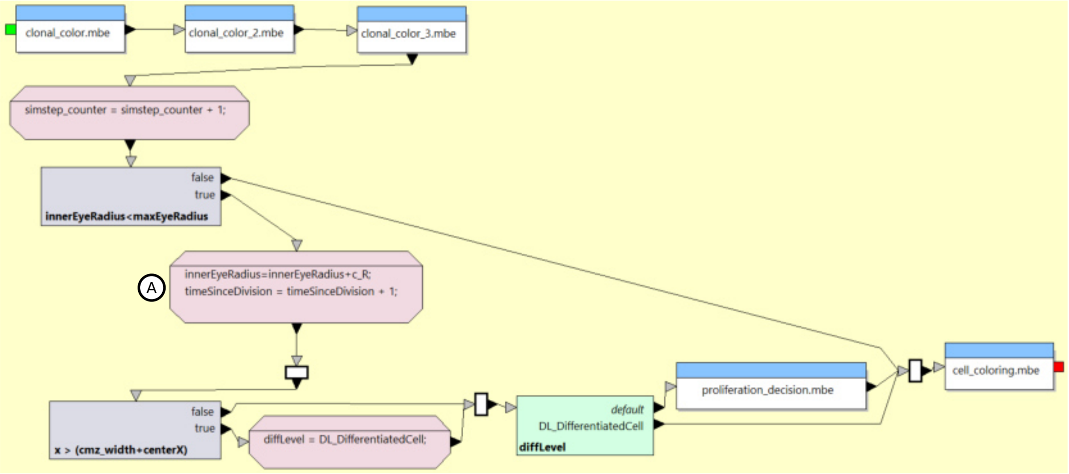

clonal_color.mbe
clonal_color_2.mbe
clonal_color_3.mbe
simstep_counter = simstep_counter + 1;
false
true
innerEyeRadius<maxEyeRadius
A
innerEyeRadius=innerEyeRadius+c_R;
timeSinceDivision = timeSinceDivision + 1;
proliferation_decision.mbe
cell_coloring.mbe
false
true
x > (cmz_width+centerX)
diffLevel = DL_DifferentiatedCell;
default
DL_DifferentiatedCell
diffLevel

Figure 5.24: Top-level model in the hierarchy in the model without feedback coupling.
A Only difference in model logic compared to Figure 5.12 is that there is only one growth mode that corresponds to the responder growth mode.

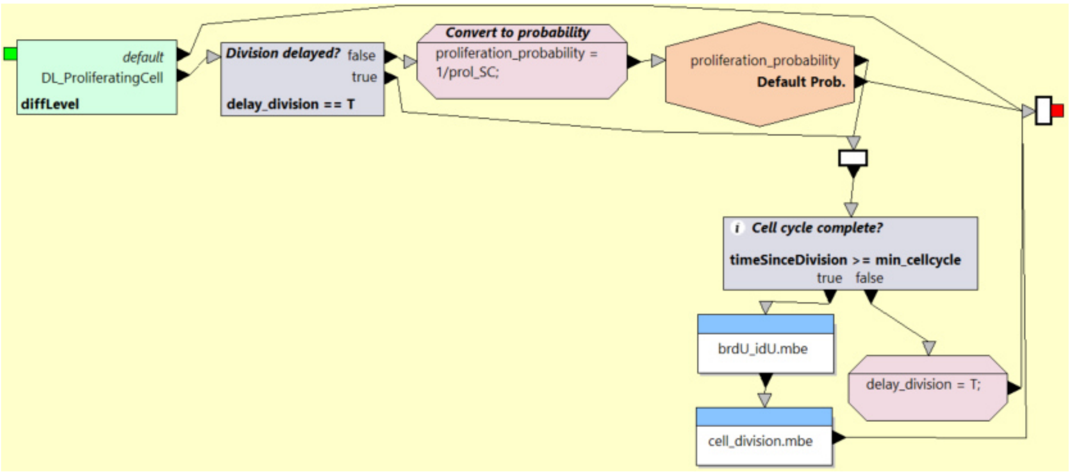


Figure 5.25: Cell division model in the model without feedback coupling.
The submodel is constructed analogously to Figure 5.16, but lacks the feedback coupling of average cell overlap to cell division as well as branch points relating to differential division parametrisation, which were introduced at a later model version.

Model with differential proliferation of PCs and SCs

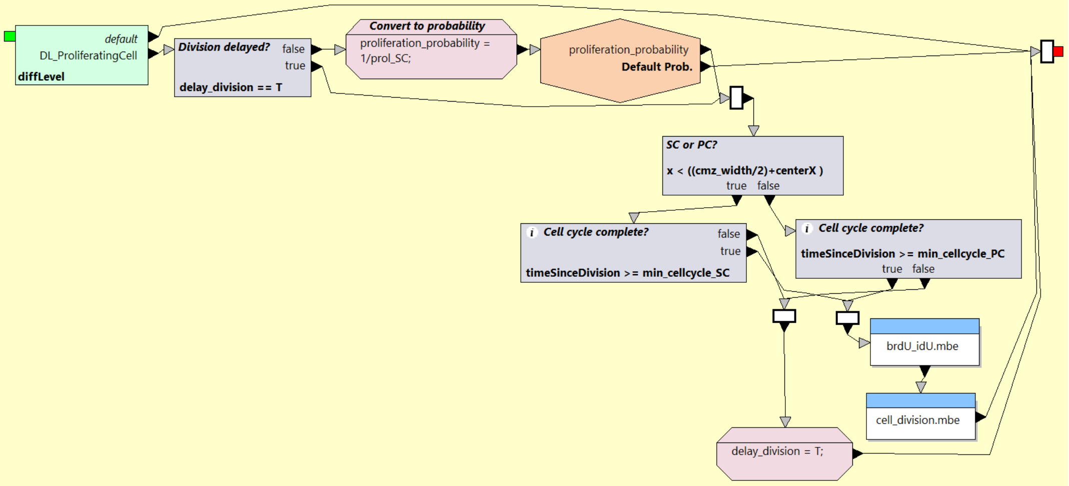


Figure 5.26: Cell division model in the model with differential SC and PC behavior.
The submodel is constructed analogously to Figure 5.16, but lacks the feedback coupling of average cell overlap to cell division as well as branch points relating to differential division parametrisation, which were introduced at a later model version. The submodel differentiates between PCs and SCs according to position along the CMZ.

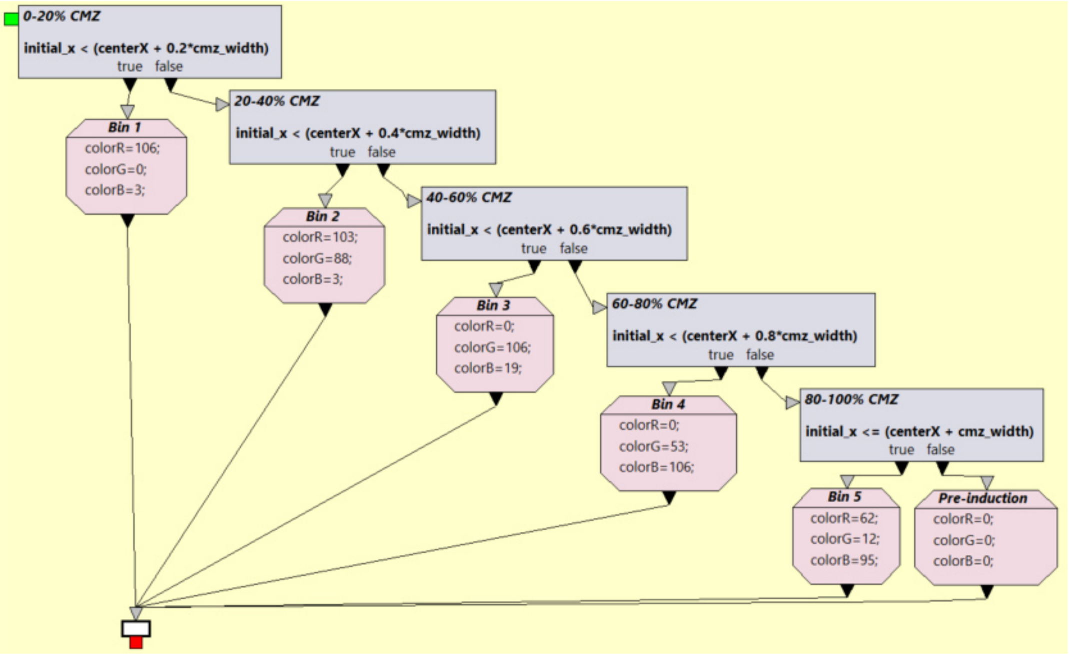
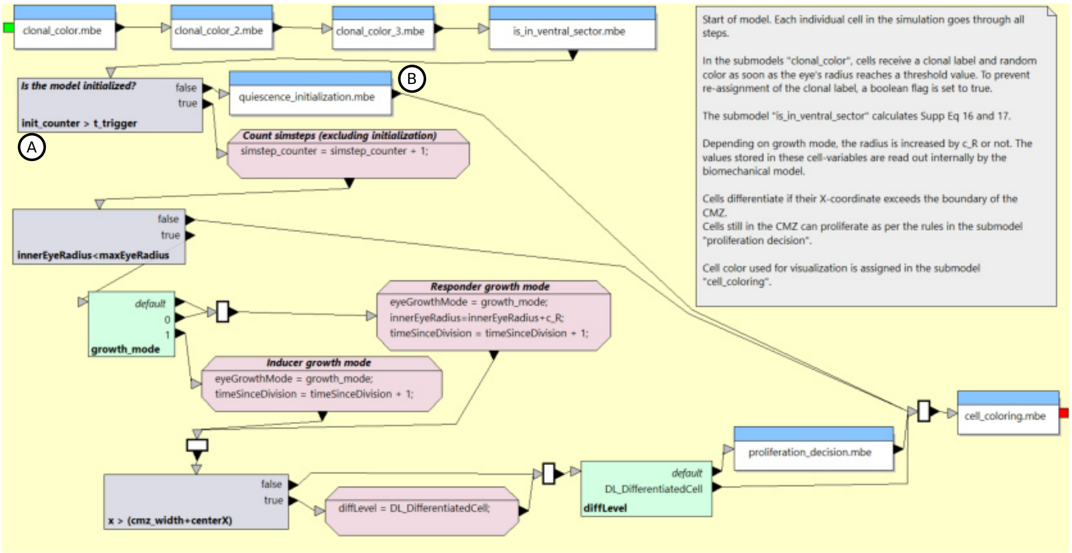


Figure 5.27: Additional colouring submodel used for Figures 2.7 panels C-D, and 2.8 leftmost part of panels C and D.
The submodel relates the x-coordinate of the clonal founder cell (`initial_x`) to the extent of the CMZ to determine colouring of each clone.

RPE model with quiescent state

As described in Results Section 2.4.2, the model from Tsingos et al. [2019] was extended to include a stable quiescent state with memory. To generate an initial distribution of quiescent and actively cycling cells, the first t_{trigger} simulation steps are performed without any further cell action (Figure 5.28 and 5.29).


clonal_color.mbe
clonal_color_2.mbe
clonal_color_3.mbe
is_in_ventral_sector.mbe
Is the model initialized?
false
true
init_counter > t_trigger
A
quiescence_initialization.mbe
B
Count simsteps (excluding initialization)
simstep_counter = simstep_counter + 1;
false
true
innerEyeRadius<maxEyeRadius
default
0
1
growth_mode
Responder growth mode
eyeGrowthMode = growth_mode;
innerEyeRadius=innerEyeRadius+c_R;
timeSinceDivision = timeSinceDivision + 1;
Inducer growth mode
eyeGrowthMode = growth_mode;
timeSinceDivision = timeSinceDivision + 1;
false
true
x > (cmz_width+centerX)
diffLevel = DL_DifferentiatedCell;
default
DL_DifferentiatedCell
diffLevel
proliferation_decision.mbe
cell_coloring.mbe

Start of model. Each individual cell in the simulation goes through all steps.

In the submodels "clonal_color", cells receive a clonal label and random color as soon as the eye's radius reaches a threshold value. To prevent re-assignment of the clonal label, a boolean flag is set to true.

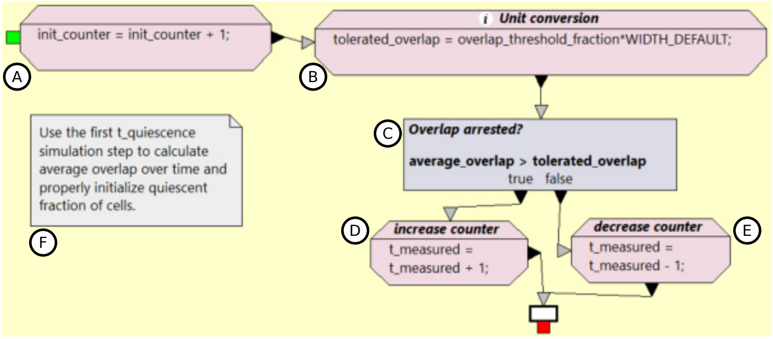
The submodel "is_in_ventral_sector" calculates Supp Eq 16 and 17.

Depending on growth mode, the radius is increased by c_R or not. The values stored in these cell-variables are read out internally by the biomechanical model.

Cells differentiate if their X-coordinate exceeds the boundary of the CMZ.
Cells still in the CMZ can proliferate as per the rules in the submodel "proliferation decision".

Cell color used for visualization is assigned in the submodel "cell_coloring".

Figure 5.28: Top-level model in the hierarchy of the model with quiescence.
A Check if t_{trigger} simulation steps have elapsed. **B** Submodel for initialisation of simulation state.



Use the first t_quiescence simulation step to calculate average overlap over time and properly initialize quiescent fraction of cells.

Figure 5.29: Initialisation of the quiescent state; corresponds to Figure 5.28 B.
A Counter increase. **B** Conversion from absolute to relative units. **C** Implementation of Equation 2.18. **D-E** Implementation of Equation 2.30. **F** Explanatory note embedded into the modelling canvas.

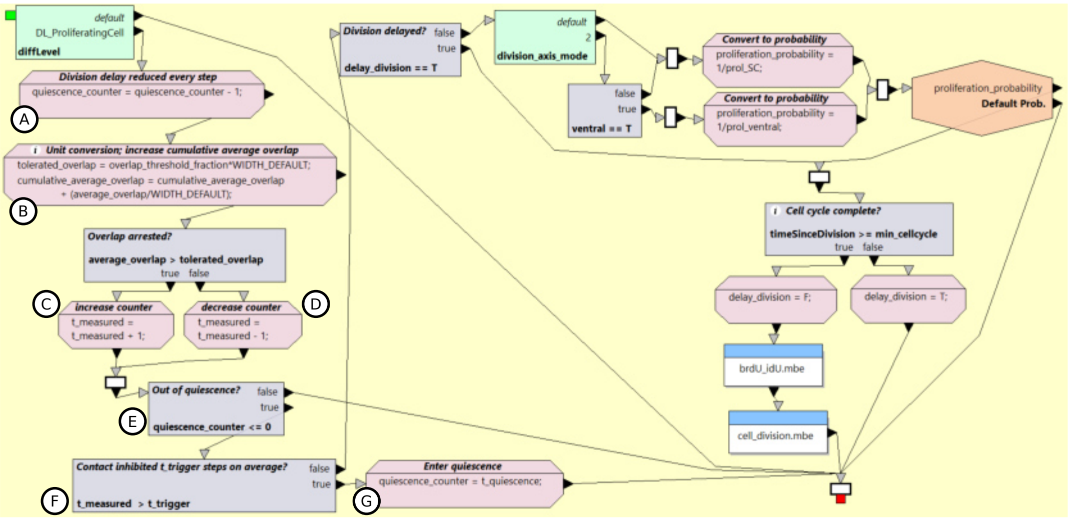

default
DL_ProliferatingCell
diffLevel
Division delay reduced every step
quiescence_counter = quiescence_counter - 1;
A
Unit conversion; increase cumulative average overlap
tolerated_overlap = overlap_threshold_fraction*WIDTH_DEFAULT;
cumulative_average_overlap = cumulative_average_overlap + (average_overlap/WIDTH_DEFAULT);
B
Overlap arrested?
average_overlap > tolerated_overlap
true false
C
increase counter
t_measured = t_measured + 1;
decrease counter
t_measured = t_measured - 1;
D
Out of quiescence?
false
true
E
quiescence_counter <= 0
Contact inhibited t_trigger steps on average?
false
true
F
t_measured > t_trigger
G
Enter quiescence
quiescence_counter = t_quiescence;
Division delayed?
false
true
delay_division == T
default
2
division_axis_mode
false
true
ventral == T
Convert to probability
proliferation_probability = 1/prol_SC;
Convert to probability
proliferation_probability = 1/prol_ventral;
proliferation_probability
Default Prob.
Cell cycle complete?
timeSinceDivision >= min_cellcycle
true false
delay_division = F;
delay_division = T;
brdU_idU.mbe
cell_division.mbe

Figure 5.30: Proliferation decision in the model with quiescence.

A Counter decrease of $t_{\text{quiescence}}$. **B** Conversion from absolute to relative units. Setting of utility variable for plotting. **C-D** Implementation of Equation 2.30. **E** Check if $t_{\text{quiescence}}$ steps elapsed since entry into quiescence. **F** Implementation of Equation 2.31. **G** Set the quiescence counter to $t_{\text{quiescence}}$.

5.9 Additional tables

5.9.1 Microscope acquisition parameters for all figures

Table 5.54: Acquisition parameters for images shown in figures of this work. Parameters listed are the ones that were changed in the vendor-supplied software used to control the microscope's settings. As files obtained from Leica microscopes come bundled, both the top-level filename and the corresponding image's subfile name are listed. Unless otherwise noted, each image consisted of one acquired tile.

Figures 1.2 A′ and 2.2 A′	
Acquisition date	12.06.18
Filename	"180612_408-DAPI_488-GFP_647-Rx2.lif"
Subfile name	"TGF_beta_1dph_s007_10x"
Microscope; objective	Leica SPE; 10x dry
Zoom	1.5
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Data acquired in collaboration with Mai Thu Nguyen.	
Figures 1.2A″ and 2.2 A″	
Acquisition date	12.06.18
Filename	"180612_408-DAPI_488-GFP_647-Rx2.lif"
Subfile name	"TGF_beta_1dph_s007_1_40x"
Microscope; objective	Leica SPE; 40x oil-immersion
Zoom	1
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Data acquired in collaboration with Mai Thu Nguyen.	
Figure 2.2 C	
Acquisition date	09.03.17
Filenames	"170309_7024_TPL_D-p53wt-loxPout_H-Cab_left_2_1"
	"170309_7024_TPL_D-p53wt-loxPout_H-Cab_left_2_2"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2 µm
Figure 2.17 A	
Data acquired by Dr Burkhard Höckendorf and Prof Dr Lázaro Centanin [Höckendorf, 2013].	
Figure 2.17 B	
Acquisition date	17.07.15
Filename	"oca2_cab_righteye_1011.nd2"
Microscope; objective	Nikon AZ100; 2x dry
Zoom	6
Exposure time	9 ms
Output tile size	1636x1088 pixels
z-step	12.5 µm
Data acquired in collaboration with Dr Stephan Kirchmaier at the Nikon Imaging Center.	
Figure 2.20 C′-C″	
Acquisition date	07.04.17
Filename	"170407_7174_rx2iCreRSG_right_1_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	2.16 µs
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Figure 2.21 A′-A″	
Acquisition date	06.08.18
Filename	"180806_Cristina_5dBrdU-fix_405-DAPI_488-GFP_552-RFP_638-BrdU.lif"
Subfile name	"Cristina_5dBrdU+fix_ccl25b-GFP_tlx-H2BRFP_s007_1_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Number of tiles	2

Data acquired in collaboration with Mai Thu Nguyen.

Figure 2.21 C′

Acquisition date	05.10.18
Filename	"181005_8090_rx2CreRSG_0dph_TMX+BrdU_endogenous-GFP_mCherry_BrdU-647.lif"
Subfile name	"LEFT-2_2"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 2.21 C″

Acquisition date	23.10.18
Filename	"181023_tlxCreRSG_0dph_tmx+BrdU.lif"
Subfile name	"8119_LEFT_0dph_t+b_405-DAPI_549-BrdU_647-GFP_TileScan_001_Merging_001"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.21 C‴

Acquisition date	30.11.18
Filename	"181130_8127_ccl25bCreRSG_0dph_tmx+BrdU_488-GFP_647-BrdU_born-180330_fix-181123.lif"
Subfile name	"retina_2_RIGHT-retina_2_RIGHT_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.21 D′

Acquisition date	21.09.18
Filename	"180921_rx2CreRSG_405-DAPI_488-GFP_552_mCherry_638_BrdU.lif"
Subfile name	"8090_rx2RSG_LEFT_12dph-tmx+BrdU_fix-180918/TileScan_001_Merging_001"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	400
Output tile size	512 x 512 pixels
z-step	2 µm
Number of tiles	4

Figure 2.21 D″

Acquisition date	12.11.18
Filename	"181112_8119_tlxCreRSG_13dph_tmx+BrdU_488-GFP_549-BrdU.lif"
Subfile name	"LEFT_1/TileScan_001_Merging_001"
Microscope; objective	Leica SP8; 10x dry
Zoom	1
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2 µm
Number of tiles	4

Figure 2.21 D‴

Acquisition date	12.11.18
Filename	"181112_ccl25bCre_RSG_22dph_tmx+BrdU_488-GFP_549-BrdU.lif"
Subfile name	"LEFT-TileScan_001_Merging_001"
Zoom	1
Scan speed	600
Output tile size	512x512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.23 A′

Acquisition date	23.12.16
Filename	"161223_6668_HsRSG_brdu_left_8.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels

z-step	2.94 µm

Figure 2.26 A′

Composite of 2 datasets acquired with the following parameters:

Acquisition date	12.11.18
Filename	"181023_tlxCreRSG_0dph_tmx+BrdU.lif"
Subfile name	"8119_RIGHT_0dph_t+b_405-DAPI_549-BrdU_647-GFP/TileScan_001_Merging_001" "8119_LEFT_0dph_t+b_405-DAPI_549-BrdU_647-GFP/TileScan_001_Merging_001"
Microscope; objective	Leica SP8; 10x dry
Zoom	1
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2 µm
Number of tiles	4

Figure 2.26 A″

Acquisition date	22.11.18
Filename	"181122_tlx-GFP_0dph_BrdU-EdU.lif"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	1.47
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 2.26 B′

Composite of 2 datasets acquired with the following parameters:

Acquisition date	23.10.18
Filename	"181023_ccl25bRSG_0dph_tmx+BrdU.lif"
Subfile name	"8127_RIGHT_odph-t+b_405-DAPI_549-BrdU_647-GFP/TileScan_001_Merging" "8127_LEFT_odph-t+b_405-DAPI_549-BrdU_647-GFP_2/TileScan_001_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	1
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Composite of 4 datasets acquired with the following parameters:

Acquisition date	30.11.18
Filename	"181130_8127_ccl25bCreRSG_0dph_tmx+BrdU_488-GFP_647-BrdU_born-180330_fix-181123.lif"
Subfile name	"retina_1_RIGHT-retina_1_RIGHT_Merging" "retina_2_RIGHT-retina_2_RIGHT_Merging" "retina_LEFT_1_Merging" "retina_LEFT_2_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	1
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.26 B″

Acquisition date	14.11.18
Filename	"181114_8268_tlx-H2B-RFP_ccl25b-GFP_0dph_B48_C0_405-DAPI_488-GFP_549-dsRed-BrdU.lif"
Subfile name	"retina_up_right/TileScan_001" (individual tiles)
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	1
Scan speed	600
Output tile size	512 x 512 pixels
z-step	1 µm
Number of tiles	6

Figure 2.26 C′

Composite of 23 datasets.

2 datasets acquired with the following parameters:

Acquisition date	05.10.18
Filename	"181005_8090_rx2CreRSG_0dph_TMX+BrdU_endogenous-GFP_mCherry_BrdU-647.lif"
Subfile names	"LEFT-2_2" "RIGHT-1_2"
Microscope; objective	Leica SP8; 10x dry

Zoom	0.75
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

4 datasets acquired with the following parameters:

Acquisition date	23.01.17
Filenames	"170123_7174_rx2iCre-RSG_left_4.ids"
	"170123_7174_rx2iCre-RSG_left_5.ids"
	"170123_7174_rx2iCre-RSG_right_1.ids"
	"170123_7174_rx2iCre-RSG_right_3.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2.09 µm

9 datasets acquired with the following parameters:

Acquisition dates	07.04.17 and 16.06.17
Filenames	"170407_7174_rx2iCreRSG_left_1_1.ids"
	"170407_7174_rx2iCreRSG_left_2_1.ids"
	"170407_7174_rx2iCreRSG_right_1_1.ids"
	"170407_7174_rx2iCreRSG_right_2_1.ids"
	"170616_7174_left_2.ids"
	"170616_7174_left_4.ids"
	"170616_7174_right_2.ids"
	"170616_7185_left_2.ids"
	"170616_7185_right_2.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	2.16 µs
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

4 datasets acquired with the following parameters:

Acquisition date	15.09.17
Filenames	"170915_7174_rx2RSG_left_1.ids"
	"170915_7174_rx2RSG_left_2.ids"
	"170915_7174_rx2RSG_left_3.ids"
	"170915_7174_rx2RSG_left_4.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2.94 µm

4 datasets acquired with the following parameters:

Acquisition date	15.09.17
Filenames	"170915_7174_rx2RSG_right_1.ids"
	"170915_7174_rx2RSG_right_2.ids"
	"170915_7174_rx2RSG_right_3.ids"
	"170915_7174_rx2RSG_right_4.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 2.26 C$''$

Acquisition date	21.12.18
Filename	"181221_8509_tp1-GFP_0dph_405-DAPI_488-GFP_549-rx2.lif"
Subfile name	"retina_3-1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	1.35
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 2.26 D

Composite of 6 datasets acquired with the following parameters:

Acquisition date	21.09.17
Filename	"170921_7511_ubiCreRSG_3h-5uM-TMX_left_1.ids"
	"170921_7511_ubiCreRSG_3h-5uM-TMX_left_3.ids"
	"170921_7511_ubiCreRSG_3h-5uM-TMX_right_1.ids"
	"170921_7511_ubiCreRSG_3h-5uM-TMX_right_2.ids"
	"170921_7511_ubiCreRSG_ON-5uM-TMX_left_1.ids"

	"170921_7511_ubiCreRSG_ON-5uM-TMX_right_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 2.26 E

Acquisition date	07.12.18
Filename	"181207_8483_ccl25b-GFP_10dph_BrdU_EdU_405-DAPI_488-GFP_549-BrdU_647-EdU.lif"
Subfile name	"retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	0.85
Scan speed	600
Output tile size	512 x 512 pixels
z-step	1.5 µm
Number of tiles	4

Figure 2.27 B′-B″

Acquisition date	05.10.18
Filename	"181005_8090_rx2CreRSG_0dph_TMX+BrdU_endogenous-GFP_mCherry_BrdU-647.lif"
Subfile name	"RIGHT-2_GFP_mCherry_BrdU"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 2.27 B‴-B⁗

Acquisition date	14.06.18
Filename	"180614_405-DAPI_488-GFP_552-BrdU.lif"
Subfile name	"8119_tlxCreRSG_0dph-tmx-BrdU_left_VR_10x_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.27 C′-C″

Acquisition date	12.11.18
Filename	"181112_ccl25bCre_RSG_22dph_tmx+BrdU_488-GFP_549-BrdU.lif"
Subfile name	"RIGHT-TileScan_001_Merging_001"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.27 C‴-C⁗

Acquisition date	30.08.18
Filename	"180830_7826_ubiCreRSG_17dTB_405-DAPI_488-GFP_552-BrdU_638-rx2.lif"
Subfile name	"RIGHT_2_RPE-side_001_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.75
Scan speed	400
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.30 A′-B″

Panel A′

Acquisition date	08.06.18
Filename	"180608_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0dph_fix_1dph_s013_63x_Merging"

Panel A″

Acquisition date	28.06.18
Filename	"180628_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0dph_fix_1dph_s002_63x"

Panel B′

Acquisition date	28.06.18
Filename	"180628_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_5dph_fix_6dph_s001_63x"

Panel B″

Acquisition date	28.06.18

Filename	"180628_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_11dph_fix_12dph_s004_2_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	2 (A′); 6 (B″)

Data acquired in collaboration with Mai Thu Nguyen.

Figure 2.31 A′-B″

Acquisition date	27.03.19
Filename	"190327_6384-oca2-F0_BrdU-EdU_double_pulse_405-DAPI_488-PCNA_549-BrdU_647-EdU.lif"
Subfile name	"retina_3_retina_3_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	1
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.7 µm
Number of tiles	4

Figure 2.32 A′

Acquisition date	27.03.19
Filename	"190327_6384-oca2-F0_BrdU-EdU_double_pulse_405-DAPI_488-PCNA_549-BrdU_647-EdU.lif"
Subfile name	"retina_3_retina_3_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	1
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.7 µm
Number of tiles	4

Figure 2.32 A″

Acquisition date	16.11.18
Filename	"181116_8463_spooky_10dph_BrdU-48h_EdU-24h_C0_405-DAPI_488-PCNA_549_BrdU_647-EdU.lif"
Subfile name	"retina_2-retina_2_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	512 x 512 pixels
z-step	1.0 µm
Number of tiles	9

Figure 2.32 A‴

Acquisition date	22.11.18
Filename	"181122_8463_spooky_17dph_BrdU-48h_EdU-24h_C0_405-DAPI_488-PCNA_549-BrdU_647-EdU.lif"
Subfile name	"retina_1-retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 2.33 A′-A″

Acquisition date	07.12.18
Filename	"181207_8483_ccl25b-GFP_10dph_BrdU_EdU_405-DAPI_488-GFP_549-BrdU_647-EdU.lif"
Subfile name	"retina_6_zoom"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.85
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 2.34 A′-A″

Acquisition date	20.12.18
Filename	"181220_8417_HsCreRSG_founder1_0d-Hs+BrdU_10d-EdU_18d-fix.lif"
Subfile name	"retina_5_zoom-TileScan_001_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	600

Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	2

Figure 2.34 B′-B″

Acquisition date	20.12.18
Filename	"181220_8417_HsCreRSG_founder1_0d-Hs+BrdU_10d-EdU_18d-fix.lif"
Subfile name	"retina_5_zoom_2-TileScan_001_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	2

Figure 2.34 C′-C″

Acquisition date	20.12.18
Filename	"181220_HsCreRSG_founder1_0d-HS+BrdU_10d-EdU_24d-fix.lif"
Subfile name	"retina_4_zoom-TileScan_001_Merging"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	1.45
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.75 µm
Number of tiles	3

Figure 2.34 D′-D″

Acquisition date	20.12.18
Filename	"181220_HsCreRSG_founder1_0d-HS+BrdU_10d-EdU_24d-fix.lif"
Subfile name	"retina_4_zoom_3"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	2.05
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.75 µm

Figure 2.36 A″

1 dph retina	
Acquisition date	08.06.18
Filename	"180608_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0dph_fix_1dph_s013_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	2
6 dph retina	
Acquisition date	08.06.18
Filename	"180608_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_5dph_fix_6dph_s003_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	2
12 dph retina	
Acquisition date	11.06.18
Filename	"180611_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0dph_fix_12dph_s002_1_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	6
19 dph retina	
Acquisition date	05.07.18
Filename	"180705_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0_18dph_fix_19dph_ub_s001_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	8

19 dph retina inset	
Acquisition date	05.07.18
Filename	"180705_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0_18dph_fix_19dph_ub_s003_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	12
35 dph retina inset	
Acquisition date	29.06.18
Filename	"180629_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0dph_fix_35dph_s006_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	15
56 dph retina inset	
Acquisition date	29.06.18
Filename	"180629_405-DAPI_488-rx2_549-BrdU_sections.lif"
Subfile name	"Cab_BrdU_0dph_fix_56dph_s002ub_1_63x_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	2.0 µm
Number of tiles	14

All datasets acquired in collaboration with Mai Thu Nguyen.

Figure 2.36 B′

Acquisition date	27.03.19
Filename	"190327_6384-oca2-F0_BrdU-EdU_double_pulse_405-DAPI_488-PCNA_549-BrdU_647-EdU.lif"
Subfile name	"retina_3_retina_3_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	1
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.7 µm
Number of tiles	4

Figure 2.36 B″

Acquisition date	16.11.18
Filename	"181116_8463_spooky_10dph_BrdU-48h_EdU-24h_C0_405-DAPI_488-PCNA_549_BrdU_647-EdU.lif"
Subfile name	"retina_2-retina_2_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	512 x 512 pixels
z-step	1.0 µm
Number of tiles	9

Figure 2.36 B‴

Acquisition date	22.11.18
Filename	"181122_8463_spooky_17dph_BrdU-48h_EdU-24h_C0_405-DAPI_488-PCNA_549-BrdU_647-EdU.lif"
Subfile name	"retina_1-retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 2.36 C′

Acquisition date	26.11.18
Filename	"181126_8461_tlx-GFP_rx2-H2B-RFP_17dph_BrdU-71h_C0_405-DAPI_488-GFP_549-RFP_647-BrdU.lif"
Subfile name	"retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	1
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Number of tiles	4

Figure 2.37 A′-A″

Acquisition date	17.12.18
Filename	"181217_8483_ccl25b-GFP_17dph_BrdU_EdU_405-DAPI_488-GFP_549-BrdU_647-EdU.lif"
Subfile name	"retina_2_zoom-retina_2_zoom"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	1.15
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 2.37 A′′′′

Acquisition date	17.12.18
Filename	"181217_8483_ccl25b-GFP_17dph_BrdU_EdU_405-DAPI_488-GFP_549-BrdU_647-EdU.lif"
Subfile name	"retina_6"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 2.37 B′-B″, C′-C″

Acquisition date	26.11.18
Filename	"181126_8461_tlx-GFP_rx2-H2B-RFP_17dph_BrdU-71h_C0_405-DAPI_488-GFP_549-RFP_647-BrdU.lif"
Subfile name	"retina_1_zoom"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	0.95
Scan speed	200
Output tile size	1024 x 1024 pixels
z-step	1.0 µm
Number of tiles	2

Figure 2.37 B′′′′

Acquisition date	17.12.18
Filename	"181217_8482_rx2-GFP_17dph_BrdU_EdU_405-DAPI_488-GFP_549-BrdU_647-EdU.lif"
Subfile name	"retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	0.8
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.5 µm

Figure 2.37 C′′′′

Acquisition date	26.11.18
Filename	"181126_8461_tlx-GFP_rx2-H2B-RFP_17dph_BrdU-71h_C0_405-DAPI_488-GFP_549-RFP_647-BrdU.lif"
Subfile name	"retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	1
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 2.38 A′-D″

Data acquired by Dr Colin Lischik. Refer to Lischik [2019].

Figure 2.39 A, B, C, D

Acquisition date	20.12.18
Filename	"181220_8417_HsCreRSG_founder1_0d-Hs+BrdU_10d-EdU_18d-fix.lif"
Subfile name	"retina_6_zoom-TileScan_001_Merging"
Microscope; objective	Leica SP8; 63x glycerol-immersion
Zoom	1.1
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.5 µm
Number of tiles	9

Figure 2.39 E

Acquisition date	20.12.18
Filename	"181220_HsCreRSG_founder1_0d-HS+BrdU_10d-EdU_24d-fix.lif"
Subfile name	"retina_4_zoom-TileScan_001_Merging"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	1.45

Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.75 µm
Number of tiles	3

Figure 2.39 F

Acquisition date	20.12.18
Filename	"181220_HsCreRSG_founder1_0d-HS+BrdU_10d-EdU_24d-fix.lif"
Subfile name	"retina_4_zoom_2"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	2.05
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.75 µm

Figure 2.40 A′-A‴

Acquisition date	03.12.18
Filename	"181203_8119_tlxCreRSG_12dph-tmx+BrdU+405-DAPI_488-GFP_549-BrdU.lif"
Subfile name	"retina_1_LEFT-retina_1_LEFT_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	0.9
Scan speed	600
Output tile size	512 x 512 pixels
z-step	1.5 µm
Number of tiles	8

Figure 2.40 B′

Acquisition date	03.12.18
Filename	"181203_8256_ccl25bCreRSG_22dph-tmx+BrdU_405-DAPI_488-GFP_549-BrdU_NVR.lif"
Subfile name	"retina_1_RIGHT-retina_1_RIGHT_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	1
Scan speed	600
Output tile size	512 x 512 pixels
z-step	2.0 µm
Number of tiles	4

Figure 2.40 B″

Acquisition date	03.12.18
Filename	"181203_8256_ccl25bCreRSG_22dph-tmx+BrdU_405-DAPI_488-GFP_549-BrdU_NVR.lif"
Subfile name	"retina_1_RIGHT_zoom-retina_1_RIGHT_zoom_Merging"
Microscope; objective	Leica SP8; 10x dry
Zoom	3
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm
Number of tiles	2

Figure 2.41 A′-B⁗

Acquisition date	22.11.18
Filename	"181122_8463_spooky_17dph_BrdU-48h_EdU-24h_C0_405-DAPI_488-PCNA_549-BrdU_647-EdU.lif"
Subfile name	"retina_1-retina_1"
Microscope; objective	Leica SP8; 20x glycerol-immersion
Zoom	0.75
Scan speed	600
Output tile size	1024 x 1024 pixels
z-step	1.0 µm

Figure 5.4 A′

Acquisition date	15.09.17
Filenames	"170915_7174_rx2RSG_left_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	5.04 µs
Output tile size	1024 x 1024 pixels
z-step	2.0 µm

Figure 5.4 B′

Acquisition date	06.04.18
Filenames	"180406_7451_Lrp2a–HsCreRSG_M_right.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2

Pixel dwell time	5.04 µs
Output tile size	1024 x 1024 pixels
z-step	2.94 µm

Figure 5.5

Acquisition date	01.12.16
Filenames	"161201_GaudiRSG_right_1.ids" "161201_GaudiRSG_right_2.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2.94 µm

Figure 5.6 C

Acquisition date	19.12.16
Filenames	"161219_6801_RSG_F2_NR_right_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2.94 µm

Figure 5.6 D

Acquisition date	14.06.17
Filenames	"170614_7192_right_3_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	1.92 µs
Output tile size	1024 x 1024 pixels
z-step	2 µm

Figure 5.6 E

Acquisition date	16.06.17
Filenames	"170616_7527_right_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2 µm

Figure 5.6 F

Acquisition date	16.06.17
Filenames	"170616_7229_right_1_1.ids"
Microscope; objective	Nikon AZ100; 5x dry
Zoom	2
Pixel dwell time	0.96 µs
Output tile size	1024 x 1024 pixels
z-step	2 µm

Figures 5.8 B′-C′′′

Acquisition date	15.08.16
Filename	"160815_DAPI_GFP_BrdU_EdU.lif"
Subfile name	"LEFT_04_back"
Microscope; objective	Leica SPE; 10x dry
Zoom	1.5
Scan speed	400
Output tile size	512 x 512 pixels
z-step	4.3 µm

Figure 5.10 A-B

Acquisition date	09.04.18
Filename	"180409_405-DAPI_488-bregfp_532-psmad_635-smoc1.lif"
Subfile names	"wt_14somites_r1" "wt_18somite_r2" "wt_22somites_r1" "wt_26somite_r3"
Microscope; objective	Leica SPE; 20x water
Zoom	3
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	1.19 µm

Acquisition date	01.02.18
Filename	"180201_405-DAPI_488-breGFP_532-Psmad_635-Smoc1.lif"
Subfile names	"wt_t2_r6.tif"
	"smoc_t2_l2.tif"
Microscope; objective	Leica SPE; 20x water
Zoom	3
Scan speed	400
Output tile size	1024 x 1024 pixels
z-step	1.19 µm
Acquisition date	25.05.17
Filenames	"170525_timepoint4_1_DAPI_488_660.lsm"
	"170525_timepoint-1_1.lsm"
	"170525_timepoint-3_1.lsm"
	"170525_timepoint4_2_DAPI_488_660.lsm"
Microscope; objective	Zeiss 710; 20x water
Output tile size	1024 x 1024 pixels
z-step	2 µm
Figure 5.11	
Acquisition dates	14.06.18-15.06.18
Filename	"180614_zf_BRE-Collery_mCherry.lif"
Subfile names	"2237_2317/8108_wt_2"
	"2317_0927/8108_wt_2"
	"0927_1300/8108_wt_2"
	"2237_2317/8107_smoc1_1"
	"2317_0927/8107_smoc1_1"
	"0927_1300/8107_smoc1_1"
Microscope; objective	Leica SP5; 40x water
Time interval	13 min
Scan speed	400
Output tile size	512 x 512 pixels
z-step	2 µm

5.9.2 Image processing steps for all figures

Table 5.55: Image processing steps for data in figures. Menu paths to built-in methods in ImageJ are listed in square brackets. More complex processing pipelines are described in the referenced subsections.

Figures 1.2 A′ and 2.2 A′ TL	
Sharpening filter	[Process>Sharpen]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 122°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Figures 1.2 A′ and 2.2 A′ DAPI, GFP	
Subtract background; rolling ball radius "50"	[Process>Subtract Background...]
Sharpening filter	[Process>Sharpen]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 122°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Figures 1.2 A″ and 2.2 A″	
Z project; projection type "Max Intensity"; slice range "4-5"	[Image>Stacks>Z Project...]
Rotate 122°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Figure 2.2 C	
Left panel	
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Median filter; radius "2.0" pixels	[Process>Filters>Median...]
Rotate −90°; interpolation "bicubic"	[Image>Transform>Rotate...]
Right panel	
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Median filter; radius "2.0" pixels	[Process>Filters>Median...]
Figure 2.17 A	
The image was pre-processed by Dr Burkhard Höckendorf [Höckendorf, 2013].	
Change LUT	[Image>Lookup Tables>Thallium]
Figure 2.17 B	
Rotate −110°; interpolation "bicubic"	[Image>Transform>Rotate...]
Create focused stack	Methods Section 5.3.1
Figure 2.20 C′-C″	
Subtract background; rolling ball radius "50"	[Process>Subtract Background...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 125°; interpolation "bicubic"	[Image>Transform>Rotate...]
Median filter; radius "2.0" pixels	[Process>Filters>Median...]
Change LUT	[Image>Lookup Tables>Thallium]
Figure 2.21 A′-A″	
Subtract background; rolling ball radius "50"	[Process>Subtract Background...]
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -19°; interpolation "bicubic"	[Image>Transform>Rotate...]
Figure 2.21 C′-C″	
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Panel C′	
Rotate 70°; interpolation "bicubic"	[Image>Transform>Rotate...]
Panel C″	
Rotate 82°; interpolation "bicubic"	[Image>Transform>Rotate...]
Panel C‴	
Rotate 70°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Figure 2.21 D′	
Z project; projection type "Max Intensity"; slice range "14-350"	[Image>Stacks>Z Project...]
Rotate -97°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Figure 2.21 D″	
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 200°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.21 D$'''$

Rotate 98°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Figure 2.23 A$'$

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 195°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]

Figure 2.26 A$'$

Manually overlay retinae	Methods Section 5.3.3

Figure 2.26 A$''$

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 23°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.26 B$'$

Manually overlay retinae	Methods Section 5.3.3

Figure 2.26 B$''$

Re-stitch tiles	[Plugins > Stitching > Pairwise stitching] [Preibisch et al., 2009]
Subtract background; rolling ball radius "50"	[Process>Subtract Background...]
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Median filter; radius "1.0" pixels	[Process>Filters>Median...]

Figure 2.26 C$'$

Manually overlay retinae	Methods Section 5.3.3

Figure 2.26 C"

Median filter; radius "1.0" pixels	[Process>Filters>Median...]
Minimum filter 3D; radius "x=2.0, y=2.0, z=1.0"	[Process>Filters>Minimum 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 145°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.26 D

Manually overlay retinae	Methods Section 5.3.3

Figure 2.26 E

Rotate 17°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.27 B$'$-B$''$

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 43°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.27 B$'''$-B$''''$

Create mask of RPE	Methods Section 5.3.6
Apply to mask: Maximum filter; radius "50.0" pixels	[Process>Filters>Maximum...]
Apply to mask: Convert to 32-bit	[Image>Type>32-bit...]
Apply to mask: Gaussian blur3D; radius [pixels] "x=20, y=20, z=1"	[Process>Filters>Gaussian Blur 3D...]
Multiply blurred mask with original stack	[Process>Image Calculator...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -88°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.27 C$'$-C$''$

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 43°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.27 C$'''$-C$''''$

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -93°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.30 A$'$

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Figure 2.30 A$''$

Z project; projection type "Sum slices"	[Image>Stacks>Z Project...]
Rotate -27°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.30 B$'$

Z project; projection type "Sum slices"	[Image>Stacks>Z Project...]
Rotate -90°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.30 B$''$

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Flip vertically	[Image>Transform>Flip Vertically]
Rotate 102°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.31 A′-A″

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 116°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]

Figure 2.31 A′-A″ orthogonal views

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 116°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Create rectangular selection on image	
Duplicate selection	[Image>Duplicate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]

Figure 2.31 B′-B″

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 2$, $d = 10$	Methods Section 5.3.4
Correct BrdU signal	Methods Section 5.3.5
Create mask of RPE and NVR	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 116°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]

Figure 2.31 B′-B″ orthogonal views

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 2$, $d = 10$	Methods Section 5.3.4
Correct BrdU signal	Methods Section 5.3.5
Create mask of RPE and NVR	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Rotate 116°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Create rectangular selection on image	
Duplicate selection	[Image>Duplicate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]

Figure 2.32 A′

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 2$, $d = 10$	Methods Section 5.3.4
Create mask of RPE and NVR	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 116°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]

Figure 2.32 A″

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 1$, $d = 2$	Methods Section 5.3.4
Create mask of RPE and NVR	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Enhance PCNA signal	Methods Section 5.3.7
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -74°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.32 A‴

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 1$, $d = 2$	Methods Section 5.3.4
Create mask of RPE and NVR	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Enhance PCNA signal	Methods Section 5.3.7
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -120°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.33 A′-A″

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 3$, $d = 3$	Methods Section 5.3.4
Correct BrdU signal	Methods Section 5.3.5
Create mask of RPE	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Remove high-intensity stromal cells in BrdU channel	Methods Section 5.3.8
Subtract background; rolling ball radius "25"	[Process>Subtract Background...]
Z project; projection type "Sum Slices"	[Image>Stacks>Z Project...]

Rotate -90°	[Image>Transform>Rotate...]

Figure 2.34 A′-D″

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 1$, $d = 2$	Methods Section 5.3.4
Correct BrdU signal	Methods Section 5.3.5
Create mask of RPE	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Panels A′-A″	
Rotate -90°	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Panels B′-B″	
Rotate 90°	[Image>Transform>Rotate...]
Flip vertically	[Image>Transform>Flip Vertically]
Panels C′-C″	
Rotate 90°	[Image>Transform>Rotate...]
Panels D′-D″	
Rotate -90°	[Image>Transform>Rotate...]
Flip vertically	[Image>Transform>Flip Vertically]

Figure 2.36 A″

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
1 dph retina	
Flip vertically	[Image>Transform>Flip Vertically]
Rotate -100°; interpolation "bicubic"	[Image>Transform>Rotate...]
6 dph retina	
Rotate -148°; interpolation "bicubic"	[Image>Transform>Rotate...]
12 dph retina	
Flip vertically	[Image>Transform>Flip Vertically]
Rotate 75°; interpolation "bicubic"	[Image>Transform>Rotate...]
19 dph retina	
Flip vertically	[Image>Transform>Flip Vertically]
Rotate 78°; interpolation "bicubic"	[Image>Transform>Rotate...]
19 dph retina inset	
Rotate 128°; interpolation "bicubic"	[Image>Transform>Rotate...]
35 dph retina	
Flip vertically	[Image>Transform>Flip Vertically]
Rotate -117°; interpolation "bicubic"	[Image>Transform>Rotate...]
56 dph retina	
Flip vertically	[Image>Transform>Flip Vertically]
Rotate -92°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.36 B′-B‴

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Panel B′	
Rotate 116°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Panel B"	
Rotate -74°; interpolation "bicubic"	[Image>Transform>Rotate...]
Panel B"’	
Rotate -120°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.36 C′

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 127°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]

Figure 2.36 C′ orthogonal projection

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Rotate 127°; interpolation "bicubic"	[Image>Transform>Rotate...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Create rectangular selection on image	
Duplicate selection	[Image>Duplicate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Figure 2.37 A′-A″

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 90°	[Image>Transform>Rotate...]
Change LUT of DAPI channel to "Fire"	[Image>Lookup Tables>Fire]

Figure 2.37 A′-A″ orthogonal projection

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Rotate 90°	[Image>Transform>Rotate...]

Orthogonal projection; options: "Start at right", "Rotate 90 degrees" [Image>Stacks>Reslice[\]...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 A′′′′
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Rotate 42°; interpolation "bicubic" [Image>Transform>Rotate...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 B′-B′′
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 B′-B′′ orthogonal projection
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees" [Image>Stacks>Reslice[\]...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Adjust brightness and contrast for each channel [Image>Adjust>Brightness/Contrast...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 B′′′′
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Rotate 55°; interpolation "bicubic" [Image>Transform>Rotate...]
Flip horizontally [Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 C′-C′′
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 B′-B′′ orthogonal projection
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees" [Image>Stacks>Reslice[\]...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.37 C′′′′
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Rotate 131°; interpolation "bicubic" [Image>Transform>Rotate...]
Flip horizontally [Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]

Figure 2.38 A′, B′, C′, D′
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]
Panel A′
Rotate 120°; interpolation "bicubic" [Image>Transform>Rotate...]
Panel B′
Rotate 32°; interpolation "bicubic" [Image>Transform>Rotate...]
Flip horizontally [Image>Transform>Flip Horizontally]
Z project only in slice range "14-350"
Panel C′
Rotate 70°; interpolation "bicubic" [Image>Transform>Rotate...]
Panel D′
Rotate 42°; interpolation "bicubic" [Image>Transform>Rotate...]

Figure 2.37 A′′, B′′, C′′, D′′ orthogonal projections
Median filter 3D; radius [pixels]: "x=1, y=1, z=1" [Process>Filters>Median 3D...]
Create rectangular selection on image
Duplicate selection [Image>Duplicate...]
Rotate (see below for angle); interpolation "bicubic" [Image>Transform>Rotate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees" [Image>Stacks>Reslice[\]...]
Z project; projection type "Max Intensity" [Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire" [Image>Lookup Tables>Fire]
Panel A′′ orthogonal projection
Rotate -30° prior to orthogonal projection
Panel B′′ orthogonal projection
Rotate -90° prior to orthogonal projection
Panel C′′ orthogonal projection
Rotate -110° prior to orthogonal projection
Panel D′′ orthogonal projection
Rotate -90° prior to orthogonal projection

Figure 2.39 A-F

Left subpanel

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire"	[Image>Lookup Tables>Fire]

Right subpanel

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire"	[Image>Lookup Tables>Fire]

Panel C

Rotate 170° prior to Z or orthogonal projection

Panel D

Rotate 140° prior to Z or orthogonal projection

Panel E

Rotate 90° prior to Z or orthogonal projection

Figure 2.40 A′-A‴

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Change LUT of DAPI channel to "Fire"	[Image>Lookup Tables>Fire]

Panel A′

Rotate 115°; interpolation "bicubic"	[Image>Transform>Rotate...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Panel A"

Rotate 50°; interpolation "bicubic"	[Image>Transform>Rotate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Panel A"'

Rotate 40°; interpolation "bicubic"	[Image>Transform>Rotate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Figure 2.40 B′

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Rotate 30°; interpolation "bicubic"	[Image>Transform>Rotate...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire"	[Image>Lookup Tables>Fire]
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]

Figure 2.40 B″

Rotate -90°; interpolation "bicubic"	[Image>Transform>Rotate...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]
Flip horizontally	[Image>Transform>Flip Horizontally]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Change LUT of DAPI channel to "Fire"	[Image>Lookup Tables>Fire]

Figure 2.41 A′

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -120°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 2.41 A″-B⁗

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Reduce NR signal; $b = 1$, $d = 2$	Methods Section 5.3.4
Create mask of RPE and NVR	Methods Section 5.3.6
Multiply mask with original stack	[Process>Image Calculator...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate -120°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 5.4 A′

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 90°	[Image>Transform>Rotate...]

Figure 5.4 B′

Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"; slices 15-17	[Image>Stacks>Z Project...]

Figure 5.5 A′-A‴

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 180°	[Image>Transform>Rotate...]

Figure 5.5 B′-B‴

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 180°	[Image>Transform>Rotate...]

Figure 5.6 C

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 180°	[Image>Transform>Rotate...]

Figure 5.6 D

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 180°	[Image>Transform>Rotate...]

Figure 5.6 E

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 90°	[Image>Transform>Rotate...]

Figure 5.6 F

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate 180°	[Image>Transform>Rotate...]

Figure 5.8 B′-C‴

Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Rotate −45°; interpolation "bicubic"	[Image>Transform>Rotate...]

Figure 5.10 A-B

Operations performed on all maximum projections of distal views:	
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]

Operations performed on all orthogonal views:	
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]

Specific operations on images:	
16 hpf wildtype eye maximum projection:	
Rotate -90°	[Image>Transform>Rotate...]
16 hpf wildtype eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 400-450	[Image>Stacks>Z Project...]

18 hpf wildtype eye maximum projection:	
Rotate 140°; interpolation "bicubic"	[Image>Transform>Rotate...]
18 hpf wildtype eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 365-400	[Image>Stacks>Z Project...]

20 hpf wildtype eye maximum projection:	
Rotate -145°; interpolation "bicubic"	[Image>Transform>Rotate...]
20 hpf wildtype eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 570-630	[Image>Stacks>Z Project...]

22 hpf wildtype eye maximum projection:	
Rotate -25°; interpolation "bicubic"	[Image>Transform>Rotate...]
22 hpf wildtype eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 648-700	[Image>Stacks>Z Project...]

25 hpf wildtype eye maximum projection:	
Rotate -60°; interpolation "bicubic"	[Image>Transform>Rotate...]
25 hpf wildtype eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 70-120	[Image>Stacks>Z Project...]

48 hpf wildtype eye maximum projection:	
Flip horizontally	[Image>Transform>Flip Horizontally]
48 hpf wildtype eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 400-500	[Image>Stacks>Z Project...]
Rotate -30°; interpolation "bicubic"	[Image>Transform>Rotate...]

20 hpf smoc1 mutant eye maximum projection:	
Flip horizontally	[Image>Transform>Flip Horizontally]
20 hpf smoc1 mutant eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 300-400	[Image>Stacks>Z Project...]
Rotate 60°; interpolation "bicubic"	[Image>Transform>Rotate...]

25 hpf smoc1 mutant eye maximum projection:	
Flip horizontally	[Image>Transform>Flip Horizontally]
Rotate 170°; interpolation "bicubic"	[Image>Transform>Rotate...]
25 hpf smoc1 mutant eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 400-500	[Image>Stacks>Z Project...]
Rotate -25°; interpolation "bicubic"	[Image>Transform>Rotate...]

48 hpf smoc1 mutant eye maximum projection:	
Flip horizontally	[Image>Transform>Flip Horizontally]
Rotate 30°; interpolation "bicubic"	[Image>Transform>Rotate...]
48 hpf smoc1 mutant eye orthogonal view:	
Z project; projection type "Max Intensity"; slices 350-450	[Image>Stacks>Z Project...]
Figure 5.11	
Operations performed on all maximum projections of distal views:	
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Z project; projection type "Max Intensity"	[Image>Stacks>Z Project...]
Operations performed on all orthogonal views:	
Median filter 3D; radius [pixels]: "x=1, y=1, z=1"	[Process>Filters>Median 3D...]
Orthogonal projection; options: "Start at right", "Rotate 90 degrees"	[Image>Stacks>Reslice[\]...]
12 hpf wildtype eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 35-65	[Image>Stacks>Z Project...]
18 hpf wildtype eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 25-45	[Image>Stacks>Z Project...]
20 hpf wildtype eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 40-60	[Image>Stacks>Z Project...]
26 hpf wildtype eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 35-50	[Image>Stacks>Z Project...]
12 hpf smoc1 mutant eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 35-55	[Image>Stacks>Z Project...]
18 hpf smoc1 mutant eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 30-50	[Image>Stacks>Z Project...]
20 hpf smoc1 mutant eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 40-60	[Image>Stacks>Z Project...]
26 hpf smoc1 mutant eye orthogonal projection:	
Z project; projection type "Max Intensity"; slices 35-50	[Image>Stacks>Z Project...]

Publications

Tsingos, E., Hoeckendorf, B., Suetterlin, T., Kirchmaier, S., Grabe, N., Centanin, L., and Wittbrodt, J. (2019). Retinal stem cells modulate proliferative parameters to coordinate post-embryonic morphogenesis in the eye of fish. *eLife*, 8, e42646.

Gutierrez-Triana, J.A.*, Tavhelidse, T.*, Thumberger, T.*, Thomas, I., Wittbrodt, B., Kellner, T., Anlas, K., **Tsingos, E.** and Wittbrodt, J.(2018). Efficient single-copy HDR by 5' modified long dsDNA donors. *eLife*, 7, e39468.

Roussigné, M., Wei, L., **Tsingos, E.**, Kuchling, F., Alkobtawi, M., Tsalavouta, M., Wittbrodt, J., Carl, M., Blader, P., and Wilson, S. W. (2018). Left/right asymmetric collective migration of parapineal cells is mediated by focal FGF signaling activity in leading cells. *Proceedings of the National Academy of Sciences*, 115(42), E9812-E9821.

Sütterlin, T.*, **Tsingos, E.***, Bensaci, J., Stamatas, G. N., and Grabe, N. (2017). A 3D self-organizing multicellular epidermis model of barrier formation and hydration with realistic cell morphology based on EPISIM. *Scientific Reports*, 7, 43472.

* joint first authorship

Acknowledgments

First and foremost, I want to thank Jochen for giving me the freedom to try out different things, for being always supportive, and encouraging me to take bold steps.

I want to thank all the members of the Wittbrodt lab for the friendly and open atmosphere that allowed me to develop both personally and professionally. Further, thanks to the entire 5th floor of COS for fostering such a great environment. I am very grateful to Venera for her helpful comments on an earlier draft of this thesis.

Big thanks to my parents and to my younger brother Alexandro for their boundless support.

Last but definitely not least, special thanks to my dearest husband Tiago for walking this path together with me.

Declaration

Herewith I declare that I prepared the PhD Thesis "Characterisation and computational modelling of retinal stem cells in medaka (*Oryzias latipes*)" on my own and with no other sources and aids than quoted.

Heidelberg, 2019

List of Figures

List of Tables