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. Colin Lischik
born in: Mainz, Germany
Oral examination: 24.05.2019

Combining *in vivo* imaging and mechanistic approaches to investigate Wnt regulation of retinal stem cells

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

for my Wife
for without her I would be nothing
for without her I would have nothing

Abstract

Adult postembryonic stem cells reside in tissues throughout the body of most vertebrates. Little is known, however, about the growth mode and regulation of single stem and progenitor cells *in vivo*. The continuous life-long growth and the accompanying presence of stem cells in all adult organs renders medaka a perfect model organism to address these unknowns. In particular, medaka's retinal stem cells are an ideal model for stem cell biology. Their position in surface proximity, their exclusive contribution to one of both retinal layers (neural retina or retinal pigmented epithelium) and their multipotency render retinal stem cells a great experimental system. Furthermore, medaka retinal stem cells can be investigated by *in vivo* assays in the context of the whole organism. In combination with the Cre/loxP system it is possible to mark and/or alter the signaling state of single cells. Subsequently, these cells and their progeny can be followed and clonally examined. Taken together, single cell spatial resolution and long-term observation of medaka retinal stem cells is possible.

This thesis focused on the *in vivo* behavior and Wnt signaling regulation of retinal stem and progenitor cells by *in vivo* imaging and clonal analysis.

To address this aim, three experimental lines were followed. First, *in vivo* imaging of medaka was enhanced to perform *in vivo* investigation of retinal stem cells. I optimized the choice of fluorescent proteins, anesthesia and presence of interfering pigmentation. Second, long-term *in vivo* microscopy of retinal stem and progenitor cells was performed, followed by tracking and track analysis. Finally, the Wnt signaling state of single retinal stem and progenitor cells was altered and the change in proliferative capacity and differentiation potential was investigated.

In conclusion, using the established *in vivo* imaging toolset, I unraveled fundamental mechanisms of the regulation of *in vivo* stem cells by Wnt, while being embedded in their organismal context. I showed that high Wnt stimulation in all cell types of the retina led to a high incidence of apoptosis. In contrast, low Wnt stimulation in retinal stem and progenitor cells restricts their proliferative capacity without altering their differentiation potential.

Zusammenfassung

Die meisten Vertebraten tragen adulte, postembryonale Stammzellen in sich. Allerdings ist über den Wachstumsmodus und die Regulation einzelner Stamm- und Vorläuferzellen *in vivo* wenig bekannt. Medaka ist ein ausgezeichneter Modellorganismus, um diese Unbekannten zu adressieren. Durch sein ununterbrochenes und lebenslanges Wachstum befinden sich Stammzellen in allen adulten Organen. Dies macht Medaka zum perfekten Modellorganismus, um den Wachstumsmodus und die Regulation zu adressieren. Insbesondere sind die retinalen Stammzellen zu diesem Forschungszweck ein ideales Modell. Ihre Position in Oberflächennähe, ihr exklusiver Beitrag zu einer der beiden retinalen Schichten (neuronale Retina oder retinales pigmentiertes Epithel) und ihre Multipotenz machen retinale Stammzellen zu einem ausgezeichneten System. Weiterhin können retinale Stammzellen in Medaka *in vivo* im ganzorganismischen Kontext untersucht werden. In Kombination mit dem Cre/loxP-System ist es möglich einzelne Zellen und ihre Nachkommen zu markieren und/oder ihren Signalstatus zu ändern. Anschließend können diese Zellen und ihre Nachkommen verfolgt und klonal untersucht werden. Zusammengefasst ist hiermit die räumliche Einzelzellauflösung und Langzeitbeobachtung von retinalen Stammzellen in Medaka möglich.

Schwerpunkt dieser Arbeit ist das *in vivo* Verhalten und die Wnt Signalwegregulation der retinalen Stamm- und Vorläuferzellen. Dies wird untersucht mithilfe von *in vivo* Mikroskopie und klonaler Analyse.

Um dieses Ziel zu erreichen wurden drei experimentelle Linien verfolgt. Erstens verbesserte ich die *in vivo* Mikroskopie von Medaka, um eine *in vivo* Untersuchung von retinalen Stammzellen durchführen zu können. Hierfür optimierte ich die Wahl des Fluoreszenzproteins, die Anästhesie und die vorhandene interferierende Pigmentierung. Zweitens führte ich *in vivo* Langzeitmikroskopie von retinalen Stamm- und Vorläuferzellen durch, gefolgt von Zellverfolgung und Verfolgungsanalyse. Schlussendlich veränderte ich den Wnt Signalstatus einzelner retinaler Stamm- und Vorläuferzellen. Darrauffolgend untersuchte ich die resultierenden Änderungen der Proliferationskapazität und des Differenzierungspotenzials.

Abschließend erforschte ich fundamentale Mechanismen der Regulation von *in vivo* Stammzellen durch den Wnt Signalweg. Durch die Analyse wurde gezeigt,

dass niedrige Wnt Stimulation in retinalen Stamm- und Vorläuferzellen ihre Proliferationskapazität einschränkt. Weiterhin konnte gezeigt werden, dass hohe Wnt Stimulation in allen Zelltypen der Retina zu einer hohen Inzidenz von Apoptose führt.

Publications

The following publications resulted from the presented and additional work:

Lischik CQ*, Lempp EK*, Heilig AK, Inoue D, Wittbrodt J. 2019. Modulation of Wnt signalling at single-cell level uncovers diverging functional domains in the ciliary marginal zone of medaka. *in preparation.*

Lischik CQ, Adelmann L, Wittbrodt J (2019) Enhanced *in vivo*-imaging in medaka by optimized anaesthesia, fluorescent protein selection and removal of pigmentation. PLoS ONE 14(3): e0212956. https://doi.org/10.1371/journal.pone.0212956

Seleit A*, Krämer I*, Ambrosio E, Stolper JS, Dross N, **Lischik CQ**, Centanin L. 2017. Neural stem cells induce the formation of their physical niche during organogenesis. Elife 6, e29173. doi:http://dx.doi.org/10.1101/149955

Aghaallaei N*, Gruhl F*, **Schaefer CQ**, Wernet T, Weinhardt V, Centanin L, Loosli F, Baumbach T, Wittbrodt J. 2016. Identification, visualization and clonal analysis of intestinal stem cells in fish. Development dev.134098. doi:10.1242/dev.134098

Contents

Abstract I

Zusammenfassung III

Publications V

Abbreviations XIII

Contributions XVII

1 Introduction 1

Medaka as a vertebrate model organism for developmental stem cell biology . . . 1

In vivo imaging of medaka . . . 2

The retina of medaka is an excellent stem cell model . . . 5

Wnt regulation of retinal stem and progenitor cells in medaka . . . 8

β-catenin dependent Wnt pathway . . . 8

Dominant-negative GSK3 was utilized to stimulate the Wnt pathway . . . 9

The spatiotemporal properties of the retina enable investigation of differentiation potential and proliferative capacity . . . 12

Maintained clones of single cells stimulated by Wnt are restricted in differentiation potential . . . 12

The major effects of high Wnt stimulation was clone loss and a change in differentiation potential . . . 13

2 Aims and Approaches 17

3 Results 19

Establishment of *in vivo* imaging in medaka . . . 19

A high-throughput assay allowed the *in vivo* investigation of fluorescent proteins . . . 20

α-Bungarotoxin anesthetized medaka embryos reliably . . . 28

Medaka pigmentation was optimized for *in vivo* imaging by CRISPR/Cas9 32
In vivo imaging of medaka was greatly enhanced by optimal fluorescent proteins, anesthesia with α-Bungarotoxin and the *spooky* pigment knockout 35
In vivo imaging of retinal stem and progenitor cells 37
α-Bungarotoxin and *spooky* were utilized to perform *in vivo* imaging of retinal stem and progenitor cells 37
Single retinal cells were tracked manually 38
Wnt regulation of retinal stem and progenitor cells 44
Independent insertions in two separate medaka lines were leveraged to investigate dosage effects of DN-GSK3 44
Apoptosis of cells exposed to high Wnt stimulation caused polyclone loss 47
The proliferative capacity of RSCs and eRPCs was decreased by low Wnt stimulation 50

4 Discussion **55**
In vivo imaging of medaka was enhanced 55
Fluorescent proteins were assayed *in vivo* 55
α-Bungarotoxin is the best available anesthetic for long-term imaging in medaka 58
In vivo imaging was enhanced by pigmentation mutants 58
A retinal stem cell and two modes of daughter cell behavior were observed *in vivo* 59
A presumable retinal stem cell was tracked 59
Two modes of daughter cell behavior were observed in the retina 60
Global movements were corrected subsequent to data collection 60
The present data offered a new resolution for tracking of stem cells 60
The effect of Wnt stimulation was dependent on dosage and cell type 62
High Wnt stimulation mainly led to apoptosis 64
Low Wnt stimulation decreased proliferative capacity of retinal stem and progenitor cells 65
High Wnt stimulation leads to apoptosis or immortalization, low Wnt stimulation leads to a decrease of proliferative capacity 66
Dominant-negative GSK3 has multiple targets 67

5 Conclusions **69**

6 Materials & Methods **71**
Materials . . . 71
Fish lines . . . 71
Plasmids . . . 72
Primers . . . 73
RNAs . . . 76
Antibodies . . . 78
Antibiotics . . . 78
Kits . . . 78
Enzymes and corresponding buffers . . . 79
Chemicals and reagents . . . 80
Consumables . . . 83
Media and buffers . . . 84
Equipment and Instruments . . . 86
Computers used . . . 89
Software and packages . . . 89
Methods . . . 90
Fish husbandry and microinjections . . . 90
Crossing . . . 91
Dechorionation with hatching enzyme . . . 91
Recombination of loxP constructs . . . 91
Fixation of fish . . . 92
Extraction of genomic DNA for PCR . . . 93
Total RNA extraction . . . 93
Reverse transcription . . . 94
Extraction of RNA and genomic DNA . . . 94
Whole mount immunostaining . . . 95
Oligonucleotide design and ordering . . . 97
PCR . . . 97
Q5 site-directed mutagenesis . . . 98
Oligonucleotide annealing . . . 99
Gel electrophoresis . . . 99
Molecular cloning . . . 100
Codon adaptation . . . 104
CRISPR/Cas9 . . . 104
mRNA transcription . . . 105
Microscopy . . . 106
Image and data analysis . . . 107

Workflow for SPIM data . . . 107
Startle response assay . . . 108
Comparison of medaka and *E. coli in vivo* fluorescence intensity 108
Semi-automated analysis of anesthesia movement profiles 108
Semi-automated analysis of fluorescent intensities of fluorescent proteins . . . 108

References **109**

Declaration **127**

List of Figures **130**

List of Tables **131**

7 Appendix **133**
Machine Learning analysis of fluorophore data only green data 145
data import . . . 145
Clustering analysis . . . 146
Classification . . . 149
Dimensionality Reduction . . . 154
Deep Learning . . . 155
Classification by ANN - Absolute data . . . 156
Classification by ANN - Relative data . . . 158
Deep Learning ANN for predicting time series . . . 160
Timecourse prediction by ANN - Absolute data . . . 161
Machine Learning analysis of fluorophore data only red data 163
data import . . . 163
Clustering analysis . . . 164
Classification . . . 166
Dimensionality Reduction . . . 170
Deep Learning . . . 171
Classification by ANN - Absolute data . . . 172
Classification by ANN - Relative data . . . 174
Deep Learning ANN for predicting time series . . . 176
Timecourse prediction by ANN - Absolute data . . . 177
Loading MaMuT xml and resaving tracks as csv . . . 178
Reading Spots . . . 180
Creating Pandas dataframe . . . 180

Reading Tracks and adding to the dataframe 181
Saving CSV 182
Global correction of affine transformation between timesteps 182
Loading csv 182
Building point cloud vectors 184
Iterating through all timepoints to find the corresponding points on the previous frame –> the transformation is done on cell data 186
Saving as csv 194
Calculating properties of points 194
calculating velocity 196
Saving as csv 197
3D Visualization of points using matplotlib 197
Loading the dataframe 197
Visualization 199
This notebook creates xyz.files for chimera 204
Loading csv affine corrected 204
Building point cloud vectors 204
Creating txt with XYZ coordinates of every timepoint to load into chimera 205
3D Visualization of single track points using matplotlib 206
Loading the dataframe from the xml file 206
Visualization 209

Abbreviations

3D	three-dimensional
3x pA	three times poly adenylation sequence
4D	four-dimensional
AC	amacrine cell
Actb	Actin B
ANN	artificial neural network
ArCoS	arched continuous stripe
atoh7	atonal BHLH transcription factor 7
ATP	adenosine triphosphate
BC	bipolar cell
C. elegans	*Caenorhabditis elegans*
C. intestinalis	*Ciona intestinalis*
CAI	codon adaptation index
Cas	CRISPR-associated system
ccl25b	chemokine (C-C motif) ligand 25b
cDNA	complementary DNA
CDS	coding sequence
CFP	cyan fluorescent protein
CMZ	ciliary marginal zone
conc.	concentration
CPU	central processing unit
CreERT2	Cre recombinase coupled to estrogen receptor 2
CRISPR	clustered regularly interspaced short palindromic repeats
cryaa	crystallin alpha a
Dkk	Dickkopf
dKO	double knockout
DMSO	dimethyl sulfoxide
DNA	deoxyribonucleic acid
DN-GSK3	dominant-negative GSK3
dpf	days post fertilization
dph	days post hatch

dpi	days post induction
DSB	double-strand break
E. coli	*Escherichia coli*
eGFP	enhanced green fluorescent protein
eGFP-DN-GSK3	eGFP coupled to DN-GSK3
ERM	embryo rearing medium
eRPC	early retinal progenitor cell
EtBr	ethidium bromide
FGF	fibroblast growth factor
FI	fluorescence intensity
FP	fluorescent protein
fps	frames per second
Fz	Frizzled
GaudíRSG	Gaudí red-switch-green
GB	giga byte
GCL	ganglion cell layer
gDNA	genomic DNA
GPU	graphical processing unit
GSK3	glycogen synthase kinase 3
H2A-mCherry	mCherry coupled to histone2a
H2B-eGFP	enhanced green fluorescent protein coupled to histone2b
HC	horizontal cell
HDR	homology-directed repair
H. sapiens	*Homo sapiens*
hpf	hours post fertilization
HR	heart rate
HSC	hematopoietic stem cell
hsp70	70 kilodalton heat shock protein
InDel	insertion and deletion
INL	inner nuclear layer
KO	knockout
LEF	Lymphoid Enhancer Binding Factor
LRP6	LDL Receptor Related Protein 6
lRPC	late retinal progenitor cell
LSFM	light-sheet fluorescence microscopy
mCherry	monomeric cherry fluorescent protein a

medaka	*Oryzias latipes*
MG	Müller glia
mGFPmut2	monomeric GFP carrying mutation number 2
ML	machine learning
mRNA	messenger RNA
MuVi-SPIM	multi-view single plane illumination microscope
NICD	notch intracellular domain
NMJ	neuromuscular junction
NR	neural retina
NSC	neural stem cell
oca2	oculocutaneous albinism II
OleGFP	*Oryzias latipes* codon-optimized eGFP
ONL	outer nuclear layer
OS	operating system
pax7a	paired box 7a
PC	principal component
PCR	Polymerase chain reaction
PNK	polynucleotide kinase
pnp4a	purine nucleoside phosphorylase
PR	photoreceptor
PTU	phenylthiourea
RAM	Random-Access Memory
RGC	retinal ganglion cell
RNA	ribonucleic acid
ROI	region of interest
RPC	retinal progenitor cell
RPE	retinal pigmented epithelium
RSDNGSK3	red-switch-eGFP-DN-GSK3
RSC	retinal stem cell
RT	room temperature
rx2	retinal homeobox transcription factor 2
SC	stem cell
SceGFP	*Saccharomyces cerevisiae* codon-optimized eGFP
sgRNA	single guide RNA
Shh	sonic hedgehog
slc2a15b	solute carrier family 2 (facilitated glucose transporter), member 15b

SNR	signal-to-noise ratio
sox2	sex determining region Y-box 2
SPIM	single-plane illumination microscopy
spookiest	oca2, pnp4a, slc2a15b triple KO
spooky	oca2 and pnp4a double KO
SVZ	subventricular zone
TB	tera byte
TCF3	transcription factor 3
TGF-β	transforming growth factor β
tlx	tailless
TMX	tamoxifen
tp	time point
TUNEL	terminal deoxynucleotidyl transferase dUTP nick end labeling
tyr	tyrosinase
UTR	untranslated region
wt	wild-type
xml file	extensible markup language file
zebrafish	*Danio rerio*

Contributions

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

Leonie Adelmann contributed cloning, transcription and microinjection of fluorescent proteins for the fluorescent protein comparison assay. She also conducted imaging of 96-well plates supervised by me. Design of experiments and constructs as well as analysis was conducted by me.

Philipp Stachel-Braum contributed cell tracks of the RSG data and changes to the analysis python script under my supervision.

Eva K. Lempp contributed the initial experiments on Wnt in the retina of medaka. Furthermore, she established the RSDGNSK3_high and RSDNGSK3_low lines and conducted experiments on the RSDGNSK3_high line. Exact results are obatainable from her PhD thesis [Möller, 2017].

"A man provided with
paper, pencil, and rubber,
and subject to strict discipline
is in effect a universal machine."

Alan Turing

1

Introduction

Medaka as a vertebrate model organism for developmental stem cell biology

The Japanese ricefish medaka (*Oryzias latipes*) is a teleost fish and an established model organism for developmental genetics and stem cell (SC) biology. Medaka was the first vertebrate in which mendelian segregation of alleles has been demonstrated [Toyama, 1916] and the first vertebrate to reproduce in space [Ijiri, 2003]. More importantly, its high tolerance to inbreeding offers the possibility to perform experiments in a characterized genetic background [Wittbrodt et al., 2002]. Furthermore, medaka offers a large experimental toolset, including the Cre/LoxP-system [Centanin et al., 2014], PhiC-system [Kirchmaier et al., 2013a], meganuclease transgenesis [Grabher et al., 2003], the CRISPR/Cas9-system [Ansai and Kinoshita, 2014, Stemmer et al., 2015] and the newly established inbred lines as a genomics resource [Spivakov et al., 2014]. Additionally, its life-long continuous growth is mediated by SCs contained in all adult tissues, rendering it an excellent model to investigate adult, homeostatic SCs *in vivo* [Seleit et al., 2017, Aghaallaei et al., 2016].

In vivo imaging of medaka

Medaka is easily accessible for *in vivo* imaging during early development. This is due to the transparent chorion and embryo. In the later stages of development, however, the embryo gets heavily pigmented. Previous studies circumvented these challenges by focusing on superficial features such as the lateral line [Seleit et al., 2017] or the optic vesicles [Rembold et al., 2006b]. In order to investigate deeper, more complex tissues harboring SCs the posed limitations for *in vivo* imaging were non-optimal fluorescent proteins and anesthesia. Furthermore, the pigmentation was impeding with light microscopy due to the dense mesh of reflective, absorptive and autofluorescent pigment cells across the body. These cells are especially prevalent at the head and even more so surrounding the eyes. In order to perform more extended imaging of deeper and more dense tissues under continuous anesthesia the *in vivo* approach needed to be enhanced. Challenges to overcome encompassed (1) choosing the right fluorescent protein, (2) efficacy of anesthesia and (3) pigmentation, in particular the heavy pigmentation of the eyes.

So far, the choice of fluorescent proteins (FPs) has not been systematically reviewed in any vertebrate. Rather, decisions have been based on the design of previous constructs or coding sequences (CDSs) present in the stock of the laboratory. To improve this situation, I performed a systematic assay, aimed at the identification of FPs with the optimal properties for *in vivo* imaging in medaka.

The standard anesthetic for teleost fish is tricaine (or: MS-222), which is approved in aquaculture for food production and research. Although prior studies have shown insufficient long-term anesthesia by tricaine in teleosts and adverse cardiac developmental effects [Culver and Dickinson, 2010], it is still widely used. Insufficient long-term anesthesia includes reactions to touch and light, which is especially problematic in light microscopy. In order to overcome the aforementioned disadvantages, I tested two alternative anesthetics in comparison to tricaine: etomidate and α-Bungarotoxin, the latter having been shown to be superior to tricaine in *Danio rerio* (zebrafish) [Swinburne et al., 2015].

While pigmentation was an issue for fluorescene microscopy in general,

pigmentation was especially an issue for light-sheet fluorescence microscopy (LSFM). This was due to the perpendicular arrangement of the illumination axis to the detection axis [Huisken et al., 2004, Keller et al., 2010] which increases the probability of light being influenced by pigments. The light absorbing, light reflecting and autofluorescent pigments [Fujii, 2000] are very likely to interfere with LSFM. A previously established method to abolish pigmentation was to grow teleost embryos in the toxic and teratogenic drug phenylthiourea (PTU) [Karlsson et al., 2001]. Moreover, PTU is also only effective in preventing the formation of melanin, the light absorbing pigment. The formation of autofluorescent and reflective pigment in the leucophores and iridophores, respectively, is not affected by PTU treatment. Also the carotenoid deposition in xanthophores is not affected by PTU treatment. Furthermore, already generated melanin is not removed by PTU, thus raising the need for an early developmental treatment, whenever the experiment relies on non-pigmented fish.

Overcoming the issue of pigmentation in medaka has so far been addressed either by mapping mutants with loss of a single type of pigment cell [Fukamachi et al., 2004, Kimura et al., 2014, Kimura et al., 2017] or by random mutation lines. These mutant lines are very delicate to maintain and also rely on mutations in unkown loci with a large need for rescreening each generation [Ohshima et al., 2013, Wakamatsu et al., 2001]. To resolve these pigmentation issues, our current understanding of the genetics of pigmentation was used in union with the CRISPR/Cas-system.

Light-sheet fluorescence microscopy

Fluorescence microscopy with point-scanning microscopes has long been the gold standard for microscopy due to its high spatial resolution. However, its acquisition speed and phototoxicity limited extended *in vivo* imaging in animals. *In vivo* imaging was dramatically improved with the advent of LSFM, which only illuminates each given point of a sample once during a single acquisition [Huisken et al., 2004, Keller et al., 2008].

The concept has been continuously improved with the addition of multiple illumination and detection paths and confocal detection [Krzic et al., 2012, de Medeiros et al., 2015, Chhetri et al., 2015]. The combination of several of these innovations resulted in the multi-view single plane illumination microscope (MuVi-SPIM). This microscope was utilized to perform *in vivo* imaging with confocal detection, relatively high spatial resolution (here: 0.263 to 0.406 $^{\mu m}/_{px}$, EMBL prototype as described in [Caroti et al., 2018]) and high temporal resolution (here: $\approx$15 to 20 min over 3 to 4 d). Furthermore, LSFM allows to adapt the geometry of the microscope to the properties of the sample, if needed [Höckendorf et al., 2012, Kromm et al., 2016]. One disadvantage of LSFM, however, is that the areal fluorescence detection is only adapted by filters. This renders the detection of the LSFM less flexible than the detection of a point-scanning microscope.

Data visualization and single cell tracking

Due to the high spatiotemporal resolution, resulting in a large amount of data, the visualization and screening of LSFM data is challenging. Data set sizes range from a few hundred giga bytes (GBs) to tens of tera bytes (TBs). The current state of Random-Access Memory (RAM) size is insufficient to load the data completely in order to interact with it in real-time. As a consequence, the BigDataViewer plugin for Fiji has been developed, which is able to visualize LSFM data on-the-fly by accessing only the presently required data, instead of loading the complete data set into the RAM [Schindelin et al., 2012, Pietzsch et al., 2015]. With this tool, the data is initially screened and data quality for subsequent analysis is checked.

This data was used for various analyses, one of them being single cell tracking. Single cell tracking is performed either automatically through automatic tracking algorithms [Amat et al., 2014], or manually by the aid of MaMuT, a plugin for manual and semi-automated cell tracking, based on the BigDataViewer visualization [Wolff et al., 2018]. The latter has been employed within this thesis.

The retina of medaka is an excellent stem cell model

Development of the retina

Retinal development in medaka begins with the establishment of the eye field, determined by patterning of the neural plate in the presumptive anterior neuroectoderm [Chow and Lang, 2001]. In the eye anlage Wnt is suppressed and subsequently the anlage is split by expression of transforming growth factor β (TGF-β), fibroblast growth factor (FGF) and sonic hedgehog (Shh) [Sinn and Wittbrodt, 2013]. The optic vesicles evaginate [Rembold et al., 2006b] and form the optic cups in a gastrulation-like movement [Heermann et al., 2015].

The retinal cells then differentiate and create the stereotypical structure of the vertebrate retina consisting of neural retina (NR) and retinal pigmented epithelium (RPE) (Fig. 1.1A). The NR is comprised of seven cell types: rods, cones (the two types of photoreceptors (PRs)), amacrine cells (ACs), horizontal cells (HCs), bipolar cells (BCs), retinal ganglion cells (RGCs) and Müller glia (MG) [Livesey and Cepko, 2001] (Fig. 1.1B). The NR is responsible for light detection, intermediate computation and relay of stimuli to the optic tectum in the brain. The RPE surrounds the NR and is in close contact with the PRs, providing stability and nutrients [Martinez-Morales et al., 2004]. Furthermore, the RPE is heavily pigmented in order to prevent light incidence from any other angle than the lens, establishing the directionality of visual stimuli. The described cell type composition, layering and spatiotemporal organization of the retina is conserved among vertebrates [Fischer et al., 2014, Livesey and Cepko, 2001, Perron and Harris, 2000].

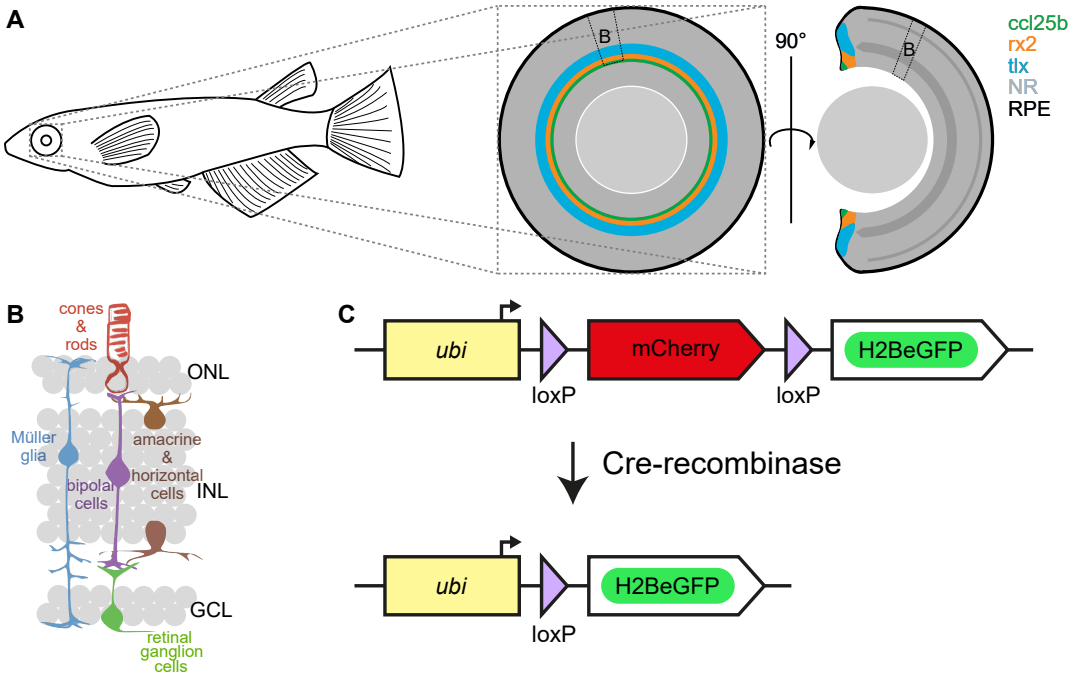

Figure 1.1: The retina of medaka is an ideal model to study stem cells *in vivo* using the GaudíRSG system.
A The retina of medaka consists of neural retina (NR) and retinal pigment epithelium (RPE). Expression domains of *ccl25b* (retinal stem cells), *rx2* (retinal stem cells) and *tlx* (retinal stem and progenitor cells) are located within the ciliary marginal zone (CMZ). Dashed box labeled B indicates a possible position for the scheme in panel B. **B** The NR of all vertebrates consists of seven cell types, that are arranged stereotypically. The retinal stem cells of medaka are multipotent and each stem cell gives rise to all seven cell types of the retina. Modified from [Centanin et al., 2014], reprinted with permission. **C** The GaudíRSG line is an established Cre/loxP-based tool to investigate stem cells and their progeny. It consists of a ubiquitously expressed mCherry, which is flanked by loxP sites. Upon Cre-mediated recombination of the loxP sites the mCherry is excised and H2B-eGFP is expressed. This is an irreversible genetically stable switch and positively marks the recombined cell and all its progeny. This mark allows the investigation of clones during development or in post hoc analyses. Modified from [Möller, 2017]. ONL: outer nuclear layer, INL: inner nuclear layers, GCL: ganglion cell layer.

Retinal stem and progenitor cells reside within the ciliary marginal zone in the postembryonic retina

Even though the structure of the vertebrate eye is conserved through evolution, there is a striking difference when comparing the retina of either amphibians or teleosts to the mammalian retina. While the mammalian retina does not grow in size in the adult animal, the amphibian and teleost retinae grow life-long along with the entire organism. This growth is mediated by retinal stem cells (RSCs), which are residing in a ring-shaped domain surrounding the lens, the ciliary marginal zone (CMZ). These RSCs are defined by the expression of chemokine (C-C motif) ligand 25b (*ccl25b*) [Lust, Becker and Wittbrodt, unpublished], retinal homeobox transcription factor 2 (*rx2*) [Sinn and Wittbrodt, 2013], tailless (*tlx*) and sex determining region Y-box 2 (*sox2*) [Reinhardt et al., 2015] (Fig. 1.1A).

RSCs contribute to the growth of NR and RPE by the addition of new cells from the periphery of the retina, located at the lateral side of the fish [Centanin et al., 2011, Centanin et al., 2014]. These RSCs are multipotent and exclusively contribute to NR or RPE. This suggests that each RSC either ultimately forms all seven cell types of the NR or contributes to the single cell type of the RPE [Centanin et al., 2011].

Furthermore, the RSCs are mainly dividing in an asymmetric growth mode, meaning RSCs self-renew and give rise to a retinal progenitor cell (RPC) [Centanin et al., 2014]. The NR-specific RSCs in the periphery of the retina (lateral side of the fish) give rise to early retinal progenitor cells (eRPCs) positive for *tlx*, which in turn give rise to late retinal progenitor cells (lRPCs), a subset of which is positive for atonal BHLH transcription factor 7 (*atoh7*) [Lust et al., 2016], and finally give rise to the differentiated cells in the center of the retina (medial) [Amato et al., 2004, Johns, 1977, Reh and Levine, 1998, Centanin et al., 2014]. RSCs in *Xenopus laevis* have been shown to be influenced by β-catenin dependent Wnt signaling [Borday et al., 2012]. Additionally, RPE cells directly adjacent to the RSCs express Wnt ligands in medaka and several components of the β-catenin dependent Wnt pathway are also active in RSCs and eRPCs [Möller, 2017]. This is in line with the observation that in mammals Wnt signaling plays a large role in SC maintenance of tissues containing adult SCs [Logan and Nusse, 2004], such as the intestine, hairs, blood and the brain [Voog and Jones, 2010].

GaudíRSG is a Cre/loxP-based system to investigate cell lineages

To date, our obtained understanding on RSCs originated from experiments using a Cre/loxP-based approach named Gaudí red-switch-green (GaudíRSG). The GaudíRSG construct consists of a ubiquitous promoter, followed by an monomeric cherry fluorescent protein (mCherry), flanked by loxP sites, which in turn is followed by an enhanced green fluorescent protein coupled to histone2b (H2B-eGFP) (Fig. 1.1C). This construct has been used to establish a transgenic line by random mutagenesis, which is ubiquitously expressing mCherry prior to recombination. Upon spatiotemporally controlled Cre-recombinase expression, the loxP sites are detected and recombined by the Cre-recombinase. This recombination eliminates the mCherry and leads to a stable genetic switch within this cell, which will be propagated to all its descendants [Centanin et al., 2014] (Fig. 1.1C). The recombined cell and all its progeny will thereafter only express H2B-eGFP. Together with a Cre driver line, this toolset allows to investigate SCs and their progeny *in vivo*, either post hoc by fixation and staining or by directly observing the recombined cells *in vivo*. So far, the investigation of SC properties has been limited to post hoc analysis.

Wnt regulation of retinal stem and progenitor cells in medaka

β-catenin dependent Wnt pathway

The β-catenin dependent Wnt pathway is intensively studied due to its involvement in development and homeostasis of organisms across evolution. A general summary is explained in the following and depicted in Fig. 1.2A. In general, the absence of Wnt triggers Axin1 and glycogen synthase kinase 3 (GSK3) among other proteins to form a complex that is sequestering β-catenin. Within this so-called destruction complex GSK3 is phosphorylating β-catenin, marking it for degradation through the proteasome and therefore depleting the cell of β-catenin. Dickkopf (Dkk) is an extracellular, competitive repressor of the Wnt ligand and therefore repressor of β-catenin dependent Wnt signaling. In

particular it inhibits the complex formation of the Wnt coreceptors Frizzled (Fz) and LDL Receptor Related Protein 6 (LRP6) [MacDonald et al., 2009], interfering with the interaction of Wnt and its receptors.

If Wnt is present, it binds to its receptors Fz and LRP6 and the destruction complex is destabilized via GSK3 and LRP6 interaction. This leads to free β-catenin, which translocates to the nucleus and together with transcription factor 3 (TCF3) and Lymphoid Enhancer Binding Factor (LEF) acts as a transcriptional regulator on its target genes. A subset of these target genes includes β-catenin dependent Wnt signaling components themselves, such as Axin2 and TCF3. These components are also acting autoinhibitory, since Axin2 is part of the destruction complex and TCF3 is mainly involved in β-catenin dependent negative transcriptional regulation [MacDonald et al., 2009] (Fig. 1.2A).

Due to the variable function of components, there are multiple angles to alter β-catenin dependent Wnt signaling by exterior stimuli. A commonly altered component is GSK3, whereof altered versions such as dominant-negative GSK3 (DN-GSK3) are available as tools. Due to the role of GSK3 in the stability of the destruction complex, the overexpression of DN-GSK3, which is competing for binding with wild-type (wt) GSK3 but catalytically inactive, leads to β-catenin dependent Wnt pathway stimulation [Yost et al., 1996].

Dominant-negative GSK3 was utilized to stimulate the Wnt pathway

To take advantage of DN-GSK3 as an established tool for Wnt stimulation, it has been introduced into the Cre/loxP-based GaudíRSG system. Therefore, the following Cre lines have been used with both the GaudíRSG and the established red-switch-eGFP-DN-GSK3 (RSDNGSK3) lines [Möller, 2017]: 70 kilodalton heat shock protein (*hsp70*):*Cre* (Fig. 1.2C), *ccl25b*:Cre recombinase coupled to estrogen receptor 2 (*CreERT2*) [Lust, Becker and Wittbrodt, unpublished] and *tlx*:*CreERT2* [Reinhardt and Tavhelidse et al., unpublished] (Fig. 1.2B-D).

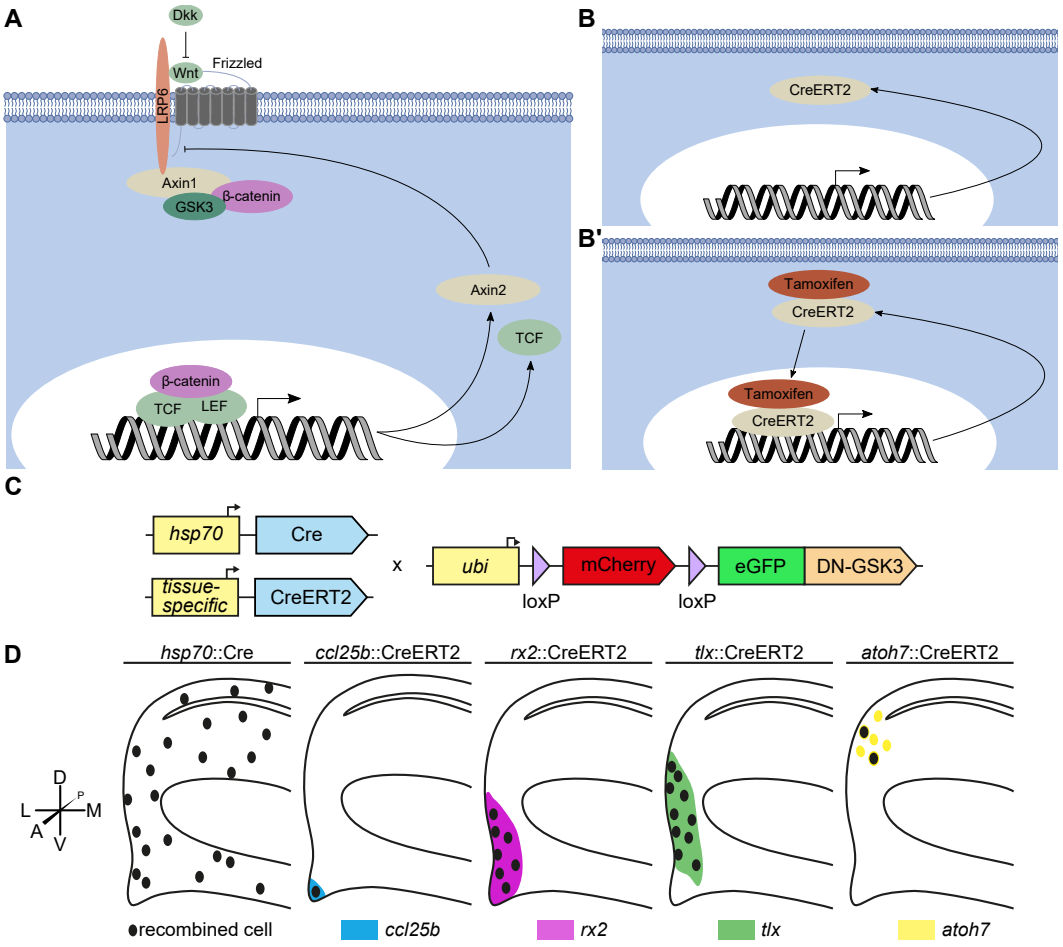

Figure 1.2: Stimulation of the β-catenin dependent Wnt pathway
A An overview of the β-catenin dependent Wnt pathway. Wnt binds to its receptors Frizzled and LRP6, which leads to the dissolution of the destruction complex formed by Axin1, GSK3 and β-catenin. β-catenin then translocates to the nucleus, where it exerts its function of transcriptional regulation in complex with TCF and LEF on its target genes, e.g. TCF and Axin2. Axin2 serves as a negative regulator of the β-catenin dependent Wnt pathway and increases the formation of the destruction complex. Extracellular Dkk inhibits Wnt binding and leads to inhibition of β-catenin dependent Wnt signaling. **B-B′** The *CreERT2*-recombinase is utilized to perform a spatiotemporally controlled irreversible genetically stable switch. Spatial expression of *CreERT2* is achieved by coupling to a promoter, whereas temporal control is achieved by the addition of tamoxifen, which leads to translocation to the nucleus, where it recombines same lox-site couples. **C** Wnt is stimulated by DN-GSK3 inserted into the GaudíRSG-construct. With this, Wnt is stimulated in the recombined cell and all its progeny, allowing to investigate the effect of β-catenin dependent Wnt signaling on single retinal stem and/or progenitor cells. Modified from [Möller, 2017]. **D** The expression domains of the promoters used for the Cre-recombinase. Recombination with *hsp70*:*Cre* will recombine cells stochastically and non-localized, whereas recombination with *ccl25b*:*CreERT2* and *rx2*:*CreERT2* will only recombine stem cells. Recombination with *tlx*:*CreERT2* will recombine stem and early progenitor cells. Recombination with *atoh7*:*CreERT2* will recombine a subset of late progenitor cells. Anatomical rosettes indicate the orientation of the schemes in D. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina). Modified from [Möller, 2017].

Recombination with the Cre-recombinase constructs was spatiotemporally confined depending on the used construct. While induction of *hsp70*:*Cre* leads to a temporally but not spatial confined recombination, the *CreERT2* constructs are confined spatiotemporally. The spatial control is achieved through coupling the CDS of *CreERT2* to a tissue-specific promoter, while the temporal control is achieved by tamoxifen (TMX) dependent activation of translocation (Fig. 1.2B-B′). Therefore, *ccl25b*:*CreERT2* recombines only the most peripheral RSCs [Lust, Becker and Wittbrodt, unpublished] (Fig. 1.2D), whereas *tlx*:*CreERT2* recombines more central RSCs and eRPCs, based on the expression domains of the promoters [Reinhardt and Tavhelidse et al., unpublished] (Fig. 1.2D). Both rely on stochastic activation of Cre in order to label only a subset of cells positive for the chosen markers. This allows for marker expression domain-specific biological deconvolution enabling to follow single cells of a defined origin and their progeny. In contrast, a global label will not allow stem or progenitor cell-specific lineage reconstruction.

The RSDNGSK3 construct was established by substituting H2B-eGFP in the GaudíRSG construct with eGFP coupled to DN-GSK3 (eGFP-DN-GSK3) (Fig. 1.2C) [Möller, 2017]. Using this construct, lines have been established and characterized for successful recombination and following Wnt stimulation [Möller, 2017]. With this construct, recombined cells are distinguished from negative cells by enhanced green fluorescent protein (eGFP) fluorescence in contrast to mCherry expression of non-recombined cells. Furthermore, recombination does not only alter the signaling state of a single cell, but also the state of its entire progeny.

Taken together, this construct allows to investigate the effect of Wnt stimulation on single cells and their possible progeny in their organismic context. This circumvents systemic effects, as a result of drug treatment or the ubiquitous overexpression of Wnt effectors. A similar construct has also been used recently using the notch intracellular domain (NICD) and has been shown to be functional and to induce a change in differentiation potential of RPCs [Perez-Saturnino et al., 2018].

The spatiotemporal properties of the retina enable investigation of differentiation potential and proliferative capacity

The retina of medaka grows stereotypically and the position of a cell is coupled to the time of its creation. These properties render it an excellent model to investigate the differentiation potential and proliferative capacity of RSCs and RPCs. Due to continuous growth, cells are continuously added from the CMZ to the retina. Once the cells are differentiated, the spatial arrangement of these cells is fixed. This establishes a direct correlation between the spatial coordinate of a cell and its temporal coordinate of differentiation. This simplifies post hoc analysis of clones. Cells are thereby linked to a specific clone and its relative time of birth is determined. Finally, the cell type also becomes apparent depending on the position in the retina. With this in mind, the GaudíRSG system was used to follow RSCs and RPCs over time, to analyze clones post hoc and to determine relative parameters of these cells by clone morphology. One property of clone morphology is the connection to the CMZ indicating a clone maintained by a RSC or RPC. If the clone is disconnected from the CMZ the founding cell has terminally differentiated and the clone is therefore not maintained. Additional parameters extractable from clones are e.g. the cell type composition of the clones, the width and length. These properties allow to draw conclusions about the differentiation potential (cell type composition) and the proliferative capacity (CMZ connection, width, length) of the clone founding cell.

Ultimately, changes introduced through the RSDNGSK3 system were monitored using the aforementioned parameters in comparison with the GaudíRSG system. Hence, the stereotypical growth mode of the retina was used as a direct readout in comparison studies.

Maintained clones of single cells stimulated by Wnt are restricted in differentiation potential

Stochastic recombination of single cells in GaudíRSG and RSDNGSK3_high by *hsp70*:*Cre* resulted in a variety of clones [Möller, 2017] (Fig. 1.3A-A′). These clones include maintained, terminated and late starting clones. The maintained clones in GaudíRSG were completely multipotent (Fig. 1.3B). The maintained clones of RSDNGSK3_high, however, were fate-restricted with a high probability and therefore in general had a decreased differentiation

potential. While 11 % of clones contributed to cells in all layers, 51 % of clones contributed only to the outer nuclear layer (ONL) and inner nuclear layer (INL). 38 % of clones even only contributed to the ONL [Möller, 2017] (Fig. 1.3B′-B‴). This decreased differentiation potential indicated, that a majority of maintained clones did not stem from RSCs or eRPCs, which both have been shown to be multipotent and give rise to all cell types of the retina [Möller, 2017]. The clone maintaining cells were most likely already committed RPCs, which were immortalized by Wnt stimulation.

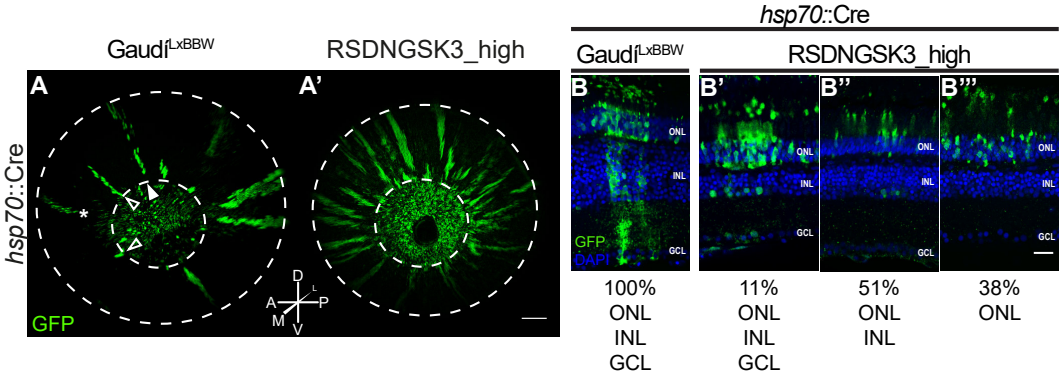

Figure 1.3: Stochastic recombination of single cells results in maintained clones. **A-A′** Gaudí^{LxBBW} and RSDNGSK3_ were recombined by *hsp70*:*Cre* and chased for 1.5 - 2.5 months. Both were stained for GFP, therefore the readout of GaudíLxBBW is very similar to the one of GaudíRSG. Shown is the central retina, with the CMZ facing away from the reader. Strikingly, clones are maintained long-term in both retinae. Examples for maintained clones (filled arrowhead), terminated clones (hollow arrowhead) and late starting clones (asterisk) are indicated. Scale bar 200 µm. **B-B‴** Maintained Gaudí^{LxBBW} clones (n = 7) are multipotent, where as maintained RSDNGSK3 clones (n = 37) have a limited differentiation potential. Only 11 % of clones consist of cells in all layers, whereas 51 % of clones consist of cells in the ONL and INL. 38 % of clones even only consist of cells in the ONL. Scale bar 20 µm. All panels are modified from [Möller, 2017]. Anatomical rosettes indicate the orientation of the retinae. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina). ONL: outer nuclear layer, INL: inner nuclear layer, GCL: ganglion cell layer.

The major effects of high Wnt stimulation was clone loss and a change in differentiation potential

Previous experiments have been conducted with an RSDNGSK3 line, hereafter termed RSDNGSK3_high [Möller, 2017]. Three *CreERT2*-constructs were used for tissue-specific recombination. RSCs were recombined by *rx2*:*CreERT2*, RSCs and eRPCs were recombined by *tlx*:*CreERT2* [Reinhardt and Tavhelidse et al., unpublished] and a subset of lRPCs were recombined by *atoh7*:*CreERT2* (Fig. 1.2D). The previous results are illustrated in Fig. 1.4. In the following the

results of the preceding work [Möller, 2017] will be interpreted more specifically in order to compare the experiments conducted within this work with the preceding work and elaborate on open questions.

Wnt stimulation in retinal stem cells leads to loss of clones

Recombination of wt RSCs by *rx2*:*CreERT2* in the GaudíRSG line leads to the formation of clones. The clones are continuous from the CMZ to the spatial coordinate relative to the original position of the CMZ at the time point (tp) of recombination. These clones consist of all cell types of the retina (Fig. 1.4A,B) and are termed arched continuous stripe (ArCoS) [Centanin et al., 2014]. The position of the clone-founding cells and their potency confirms that the initially recombined cells are indeed SCs.

Upon recombination of RSDNGSK3_high retinae by *rx2*:*CreERT2*, however, no clones were detected in 38 recombined retinae (Fig. 1.4A′,B′) [Möller, 2017]. The reason behind this clone loss has so far not been addressed and will be elucidated within this work. It has been shown in the preceding work, however, that the construct is expressed in all retinal cell types and the construct recombines as expected [Möller, 2017].

Wnt stimulation in retinal stem and early progenitor cells leads to partial clone loss and multipotent maintained clones

Recombination of RSCs and eRPCs in GaudíRSG fish by *tlx*:*CreERT2* leads to the presence of a mixture of clones consisting of ArCoS and footprints, stemming from the RSCs and eRPCs, respectively [Reinhardt and Tavhelidse et al., unpublished] (Fig. 1.4C). These footprints are terminating clones, which are therefore connecting the spatial coordinate of the CMZ at the tp of recombination and the spatial coordinate of the CMZ at the tp of clone termination. Both, the ArCoS and the footprints are created by multipotent cells, i.e. each clone consists of all cell types of the NR (Fig. 1.4D). This indicates that eRPCs are not fate-restricted and therefore still multipotent.

However, when recombining RSDNGSK3_high retinae by *tlx*:*CreERT2*, two observations were made. First, similar to the recombination with *rx2*:*CreERT2*, a major clone loss was observed (8 clones in 6 retinae as opposed to 38 clones in 5 retinae in GaudíRSG). Second, the remaining clones were still multipotent, although less wide than the ones observed in wt (Fig. 1.4C′,D′).

Concluding, the most prominent effect of Wnt stimulation in RSCs and eRPCs is the loss of clones. This does not allow to investigate the properties of the effect of Wnt stimulation on single cells and will be elaborated on within this thesis.

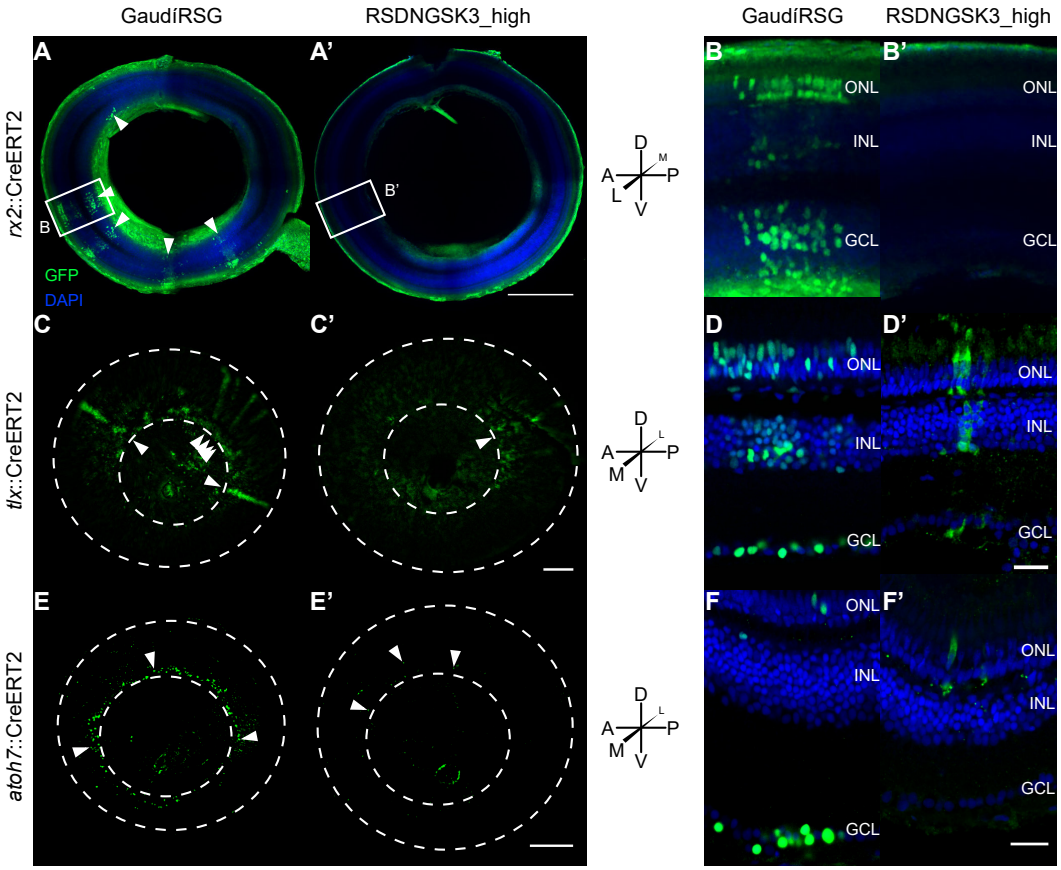

Figure 1.4: Previous experiments with DN-GSK3 reveal clone loss in RSCs and eRPCs upon Wnt stimulation and a change of differentiation potential in lRPCs.
Retinae were recombined by indicated CreERT2 and chased for 1.5 - 2 months. **A-A′** Recombination by *rx2*:*CreERT2* (RSCs) led to clone formation in GaudíRSG, whereas no clones were formed in RSDNGSK3_high. Scale bar 200 µm. **B-B′** Close-up of A-A′ showing that all cell types are present in the GaudíRSG clones, whereas no clones are present subsequently to recombination of RSDNGSK3_high. Scale bar 20 µm. **C-C′** Recombination by *tlx*:*CreERT2* (RSCs and eRPCs) led to clone formation in GaudíRSG, whereas fewer and less wide clones were formed in RSDNGSK3_high. Scale bar 200 µm. **D-D′** Close-up of recombined retinae, with cut-off CMZ showed all cell types are present in the GaudíRSG and RSDNGSK3 clones. Scale bar 20 µm. **E-E′** Recombination by *atoh7*:*CreERT2* (lRPCs) led to clone formation in GaudíRSG and RSDNGSK3_high. The number of clones also decreased upon stimulation of β-catenin dependent Wnt signaling. Scale bar 200 µm. White arrowhead marks exemplary positive cells. **F-F′** Close-up of recombined retinae, with cut-off CMZ showed a change in clone composition upon the stimulation of β-catenin dependent Wnt signaling, indicating a change in differentiation potential of recombined cells. Scale bar 20 µm. All panels are modified from [Möller, 2017]. Maintained clones are marked by arrowheads for *rx2*:*CreERT2* and *tlx*:*CreERT2*. Anatomical rosettes indicate the orientation of the retinae. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina).

Wnt stimulation in late retinal progenitor cells leads to a loss of clones and a change in differentiation potential

Recombination of a subset of lRPCs by *atoh7:CreERT2* in GaudíRSG retinae resulted in small clones, mainly comprised of RGCs and few PRs [Möller, 2017] (Fig. 1.4E,F). This hinted at an early termination of *atoh7*-positive lRPCs and inherent fate restriction. Therefore, wt lRPCs are already restricted in proliferative capacity and differentiation potential in comparison to RSCs and eRPCs.

When RSDNGSK3_high fish were recombined by *atoh7:CreERT2*, again fewer clones were formed in comparison to wt clones (GaudíRSG: 6 out of 6 retinae show clones, RSDNGSK3_high: 3 out of 6 retinae show clones) [Möller, 2017]. Furthermore, the differentiation potential of lRPCs was changed upon Wnt stimulation. The remaining clones showed a dramatically changed distribution of cell types, with no RGCs, but more PRs and presumable ACs [Möller, 2017] (Fig. 1.4E′,F′). A previous study showed that the activation of Notch in *atoh7*-positive cells also leads to a change in differentiation potential [Perez-Saturnino et al., 2018]. The resulting cell types, however, were MG, BCs and ACs located in the INL. Therefore it is very likely, that the effect of Wnt stimulation in *atoh7*-positive cells is not mediated through the Notch pathway.

2

Aims and Approaches

The aim of this thesis was to investigate the regulation of retinal stem and progenitor cells by β-catenin dependent Wnt signaling *in vivo*. For that I addressed the following goals with the aid of the listed approaches:

1. Enhancing *in vivo* imaging by light-sheet fluorescence microscopy in medaka.
   - Optimizing fluorescent protein selection
   - Improving efficacy of anesthesia
   - Abolishing imaging-interfering pigmentation

2. *In vivo* imaging of retinal stem and progenitor cells and extraction of developmental parameters.
   - Utilizing the established *in vivo* imaging toolset to image GaudíRSG retinae *in vivo*
   - Tracking single cells in the CMZ over extended time periods
   - Analyzing the behavior and division mode of cells within the CMZ

3. Unraveling the effect of Wnt stimulation on retinal stem and progenitor cells.
   - Characterizing expression level differences between two independent insertion lines
   - Leveraging expression level differences to investigate dosage effects
   - Extracting and analyzing parameters from lineage tracings
   - *In vivo* imaging and TUNEL assay to unravel clone loss subsequent to recombination

3
Results

Establishment of *in vivo* imaging in medaka

In principle *in vivo* imaging in medaka was already possible, but it was not fully established yet, and raised three main challenges. These challenges were restricting possible investigation and were therefore addressed. I determined the fluorescent proteins (FPs) with the highest fluorescence intensity (FI) in medaka and demonstrated why and how a transient *in vivo* assay is necessary and sufficient to do so. I established that α-Bungarotoxin is the best available anesthetic for medaka. Finally, I created pigment mutants, that render medaka more accessible to *in vivo* imaging, in particular the pigment-reduced lines, *spooky* and *spookiest*.

A high-throughput assay allowed the *in vivo* investigation of fluorescent proteins

mGFPmut2 and mCherry were the fluorescent proteins with the highest fluorescence intensity

A transient *in vivo* assay has been established to assess FI of commonly used FPs. This assay utilized the capability of the ACQUIFER Imaging Machine to image all wells of a 96-well plate in single well acquisitions over time. Medaka couples were synchronously mated and eggs were collected. The zygotes were microinjected with a green test FP and *mCherry* messenger RNA (mRNA) or with a red test FP and *eGFP* mRNA. *mCherry* and *eGFP* served as injection control and the measurements were normalized to the FI of the controls at 10 hours post fertilization (hpf). The injection mixes were assembled such that all mRNAs were present in equimolar amounts controlling for the different lengths and compositions of CDSs and therefore molecular weights of the mRNAs. Furthermore, all these CDSs were cloned into the same plasmid (pGGEV3), linearized with the same restriction enzyme (SpeI-HF) and transcribed with the same kit (mMessage mMachine® Sp6 Transcription Kit). This is to ensure maximal comparability of the microinjected mRNAs, which have the same 5' untranslated regions (UTRs) and three times poly adenylation sequences (3x pAs). The microinjected embryos were loaded in a volume of 150 µl by a pre-defined randomized loading scheme into a 96-well plate and imaged for at least 42 h. The resulting images were analyzed and visualized semi-automatically in Fiji and R by masking, cropping, measuring, normalizing and plotting (Graphical summary of the protocol is presented in Fig. 3.1A).

At 10 hpf mVenNB is the FP with the highest FI in the green channel, directly followed by monomeric GFP carrying mutation number 2 (mGFPmut2), whereas mCherry is the FP with the highest FI in the red channel (Fig. 3.1B). However, following the green FPs over time in Fig. 3.1C indicates that mGFPmut2 has an overall higher FI, while mVenNB's FI is only higher in a limited timeframe. For red fluorescent proteins, illustrated in Fig. 3.1D, no time-dependent difference in ranking was observed.

This high-throughput assay already hinted towards possible candidates for the FPs with the highest FI in medaka. To validate suitable FPs for *in vivo* imaging under experimental conditions via single-plane illumination microscopy (SPIM), a type of LSFM, sample FPs were imaged via SPIM. As presented in Fig. 3.1E these differences also hold true for data acquired with a SPIM, indicating that the established assay is sufficient for scoring of FPs in medaka.

The presented assay was performed with unhatched embryos, immediately after fertilization. Therefore I checked next, whether the chorion has an impact on FI in the two tested channels. To exclude this possibility medaka embryos were microinjected as previously described and half of each sample type (each injection of a FP) were dechorionated at 2 days post fertilization (dpf). Dechorionated and untreated embryos were simultaneously imaged via the AQUIFER Imaging Machine at 2, 3 and 4 dpf (exemplary in Fig. 3.1F and full in Fig. 7.1). No significant difference in FI was detected between dechorionated and untreated embryos, indicating the validity of the presented assay and the absence of the influence of the chorion on fluorescence microscopy.

In summary, mGFPmut2 and mCherry are the FPs with the highest FI in medaka.

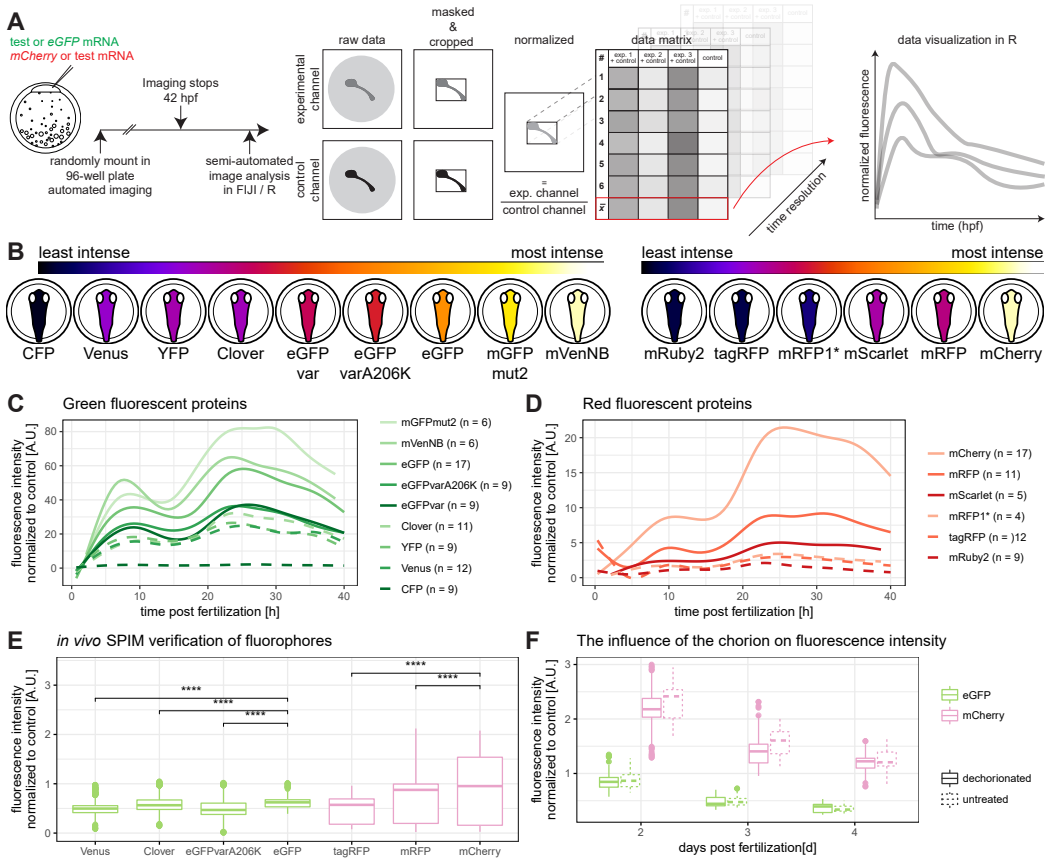

Figure 3.1: An imaging-based assay revealed that *mCherry* and *mGFPmut2* are the fluorescent proteins with the highest fluorescence intensities in medaka. **A** Illustration of semi-automated analysis. Medaka zygotes were microinjected with a green or red fluorescent test mRNA, along with mCherry or eGFP as injection control, respectively. The embryos were loaded according to a randomized loading scheme into a 96-well plate and imaged automatically for at least 42 h. The resulting images were masked, cropped and normalized to the injection control. The resulting values were plotted in R (1-cell stage modified from [Iwamatsu, 2004]). **B** At 10 hpf the FPs with the highest FI were mVenusNB and mCherry. Green fluorescent proteins are depicted at the left hand side, whereas red fluorescent proteins are depicted on the right hand side (embryos modified from [Iwamatsu, 2004]). **C** mGFPmut2 was the FP with the highest overall FI. FIs of green FPs normalized to the control over time. n-values indicate number of analyzed embryos per fluorescent protein. **D** mCherry was the FP with the highest FI. FIs of red FPs normalized to the control over time. n-values indicate number of analyzed embryos per fluorescent protein. **E** SPIM confirmed the trend of *in vivo* FIs acquired from single, hatched embryos. This indicated that the assay was sufficient to test FPs for subsequent imaging via SPIM. Venus $N = 3$ $n = 1483$, Clover $N = 3$ $n = 1294$, eGFPvarA206K $N = 3$ $n = 1395$, eGFP $N = 2$ $n = 794$, tagRFP $N = 3$ $n = 1580$, mRFP $N = 3$ $n = 2158$, mCherry $N = 2$ $n = 980$ (N indicates number of fish, n indicates number of z-slices analyzed). **F** No apparent influence of the chorion on FP FI was detected. Measured was the normalized FI of eGFP and mCherry untreated or dechorionated at 2,3 and 4 dpf. Asterisks indicate P-values: **** $P <= 0.0001$, *** $P <= 0.001$, ** $P <= 0.01$, * $P <= 0.05$, ns $P > 0.05$. The full experiment is depicted in Fig. 7.1. Figure from [Lischik et al., 2019].

In vitro properties of fluorescent proteins are no direct predictors for *in vivo* fluorescence intensity in medaka

A previous publication linked *in vivo* FIs to *in vitro* acquired properties of FPs in *Escherichia coli* [Balleza et al., 2017]. It was therefore tested, whether the correlation observed in *E. coli* also holds true in medaka. The relative FI values of medaka were plotted against the relative FI values published previously and normalized to the FPs common in both analyses. The relative FIs were diverging largely, indicating the necessity of an *in vivo* assay in vertebrate systems (Fig. 3.2A). This is in contrast to the previous publication, which demonstrated a dependence of the FIs on *in vitro* parameters, such as maturation time, expression and *in vitro* FI [Balleza et al., 2017]. These experiments were conducted, however, in a non-vertebrate, moreover a non-eukaryote. This difference in physiology seemingly had an impact on FI of FPs.

Taken together, FP *in vitro* parameters can not predict *in vivo* FI of FPs in medaka.

Codon usage table-driven codon averaging decreased fluorescence intensity of eGFP in medaka

Another question arising from the previous results is whether FI of FPs depended on codon usage and was species-specific. Therefore, all codon adaptation indices (CAIs) of the used FPs were calculated for medaka and *Homo sapiens* [Athey et al., 2017]. The CAI of a CDS is calculated based on the sequence and an averaged codon usage table for the species of interest [Puigbò et al., 2008]. The values for all tested fluorescent proteins were plotted in Fig. 3.2B with solely eGFP, mCherry and mRuby2 labeled (full labels in Fig. 7.2). Strikingly, all CAIs of FPs except mRuby2 clustered. Additionally, all CAIs indicated a marginally higher codon adaptation for *H. sapiens* than for medaka. This is not surprising, since the commonly available FPs are usually codon adapted for mammalian codon usage tables.

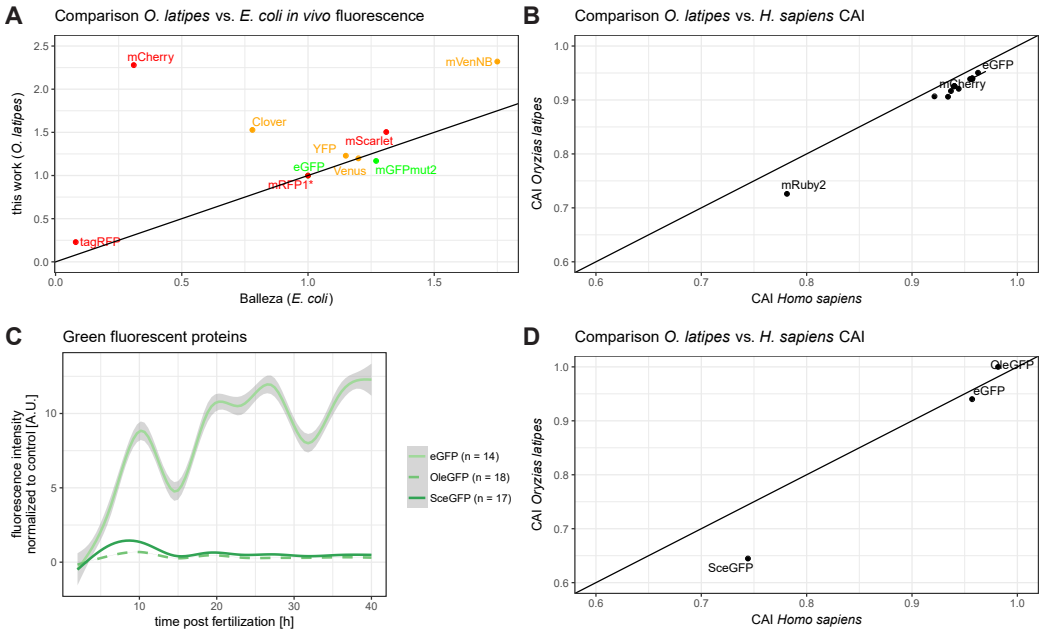

Figure 3.2: Codon usage table-driven codon adaptation decreased *in vivo* fluorescence intensity of eGFP in medaka.
A Comparison of relative *in vivo* FIs of FPs in medaka and *E. coli*. While FPs such as mGFPmut2 had a similar relative FI in both organisms, other FPs such as mCherry or Clover deviated strongly. *E. coli* FIs have been extracted from a previous publication [Balleza et al., 2017]. Medaka FIs were normalized to eGFP for green FPs, Venus for yellow FPs and mRFP1* for red FPs. **B** Comparison of codon adaptation indices (CAIs) of all investigated FPs showed that most used CDSs were similarly codon adapted. The CAI was calculated with the amino acid sequence of the FP and the species-specific codon usage table. Labeled are the controls and the outlier mRuby2. All data points are labeled in the full version in Fig. 7.2. **C** Codon adaptation of eGFP for medaka decreased its FI 25-30 fold. The experiment was conducted as outlined in Fig. 3.1A, but only with eGFP, eGFP adapted for codon usage of medaka (OleGFP) and eGFP adapted for codon usage of yeast (SceGFP) as negative control. **D** The CAIs for the sequences used in C. OleGFP was theoretically more codon adapted to medaka than wild-typic eGFP. The diagonal line in A, B and D is solely for orientation purposes and not part of the data. Figure from [Lischik et al., 2019].

In order to investigate the effect of codon adaptation, eGFP was adapted to the codon usage table of medaka [Puigbò et al., 2007], which resulted in *Oryzias latipes* codon-optimized eGFP (OleGFP). Together with *Saccharomyces cerevisiae* codon-optimized eGFP (SceGFP) [Xu et al., 2013], an experiment was performed similar to the one depicted in Fig. 3.1A with eGFP, OleGFP and SceGFP. CAIs of all used FPs were plotted indicating that OleGFP is theoretically more adapted to the average codon occurrence in medaka than eGFP and SceGFP, as expected (Fig. 3.2D). The FPs were subjected to the *in vivo* assay (Fig. 3.2C). Surprisingly, the OleGFP *in vivo* FI was not improved by pure codon usage table-driven codon averaging. On the contrary, its *in vivo* FI decreased 25 to 30-fold in comparison to the original eGFP (Fig. 3.2C). Interestingly, its FI was even lower than that of SceGFP, which was a control for low codon adaptation.

Recapping, pure codon usage table-driven codon adaptation decreased the *in vivo* FI of eGFP in medaka.

The established *in vivo* assay revealed different influences on fluorescence intensities of fluorescent proteins in zebrafish

Zebrafish (*Danio rerio*) represents another established model teleost. This raises the question whether the established *in vivo* assay can be also used to score for optimal FPs in this species. The experiments were performed according to Fig. 3.1A, with the exception, that due to the fast development of zebrafish imaging was conducted in a restricted timeframe of 12 h. Solely the most promising candidates identified in medaka were considered. The mean of FIs indicated that mVenNB and mCherry were the most suitable green and red FPs, respectively. However, if the plots themselves (Fig. 3.3B-C) were investigated the FIs indicate a strong time-dependency with striking fluctuations. Moreover, the overall FI decayed faster compared to medaka. In contrast to medaka, the fluorescent proteins are also not following the same relative pattern, complicating the prediction of useful FPs in juvenile and/or adult zebrafish.

Abbreviating, FIs of FPs are not comparable between zebrafish and medaka.

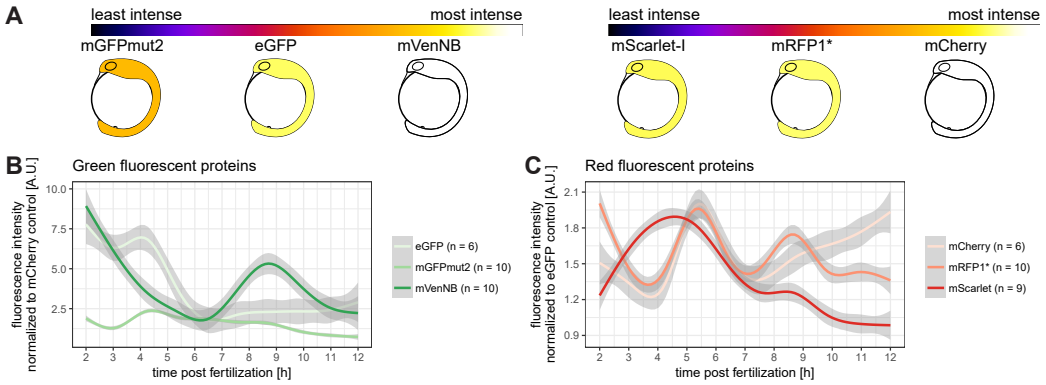

Figure 3.3: In zebrafish fluorescence intensity of fluorescent proteins varied over time.
FI measurements were conducted as outlined in Fig. 3.1A with the exception of a maximum imaging time of 12 h. **A** On average, mVenusNB and mCherry are the FPs with the highest FI in zebrafish. Green FPs are illustrated on the left hand side, while red FPs are illustrated on the right hand side (embryos modified from [Kimmel et al., 1995]). **B** Normalized FI of green FPs in zebrafish over time. Strikingly, no clear trend was detected. n-values indicate number of embryos analyzed. **C** Normalized FI of red FPs in zebrafish over time. Again, no clear trend was detected. n-values indicate number of embryos analyzed. Figure from [Lischik et al., 2019].

Time courses of fluorescent proteins were classified and predicted by artificial neural networks

To further investigate predictive power of measured time courses, machine learning (ML) and artificial neural networks (ANNs) were deployed. The first challenge was to classify a given time course to the name of the imaged FP. This enables the classification of novel tested FPs to similar, already tested FPs, indicating similar *in vivo* properties. However, this classification was challenging, even when employing several ML algorithms. For investigation, I fit the models to the same training set resulting from a standard random 80 % to 20 % training and test set split. The prediction accuracy is a measure defined as correctly predicted true positives. This ranged from 11 % to 16 % for all tested ML models, except for logistic regression, which was able to perform at an accuracy of 33 %. ML algorithms are statistical tools, which are based on fixed assumptions depending on the implemented algorithms. In contrast, ANNs offer a larger flexibility. This is due to the simulation of a neural network by using artificial neurons and training them for a specific task. Therefore, I implemented an ANN for classification in order to classify the FPs to the matching names. Biological data and time courses were thought to be too complex for standard ML algorithms. Due to their greater flexibility, ANNs are better at filtering out inherent and/or underlying noise when compared to ML algorithms.

Indeed, the established ANN was performing at an accuracy of 55 % to 65 % depending on the run. This variability is explained by the varying randomized training and test set split per run, which is used to ensure the ANN's robustness.

In addition to classification (supervised learning, addition of a dependent variable, here: name), I also applied clustering (unsupervised learning, no dependent variable) to the corresponding time courses. However, since the deployed algorithms were able to filter larger differences between different time courses, but not the subtle differences between fluorescent proteins with a similar FI, the challenge persisted. In simple terms, outgroups were detected, but more minute differences were not (Fig. 7.3).

The final goal was to predict the second fraction of the time course depending on the initial fraction of the present data. Once established, experiments could be shortened, following the demonstration of the predictive power of the first fraction of the experiment for the following time course. An ANN was deployed and the same training and test set split as previously described was used. Following training of the ANN the test set FI time course was predicted depending on the initial fraction of the time course in the test set. The full results are presented in Fig. 7.4, whereas Fig. 3.4 depicts exemplary graphs. Fig. 3.4A presents an example of the prediction of a green fluorescent protein time course, which was classified as acceptable. Currently no statistical measure is implemented for this classification, the classification as of now relies completely on similarity of the graphs. Using the similarity of 18 predicted test samples 15 of 18 predicted test samples were classified as acceptable. Three of 18 were classified as unacceptable. An example for which is depicted in Fig. 3.4A′. For the red fluorescent proteins Fig. 3.4B illustrates an acceptable time course, which was classified as such in 10 out of 12 samples, whereas 2 were classified as unacceptable. An example of which is depicted in Fig. 3.4B′.

Summarizing, ANNs predicted the continuation of *in vivo* FI time courses of FPs.

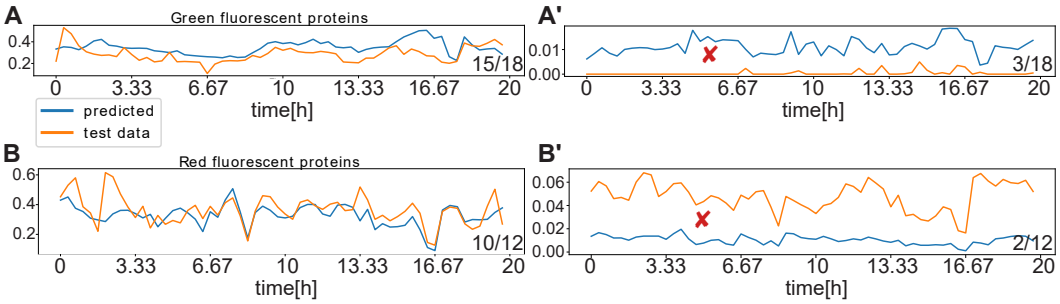

Figure 3.4: Prediction of the second fraction of experimental data based on the first fraction of the time course. Exemplary excerpt for acceptable and unacceptable predictions.
Data was split into a training set and a test set by a random 80 % to 20 % split. Illustrated are the results for the test set. An artificial neural network (ANN) was deployed to predict the continuation of the time course experiment, given the first fraction. Plotted is the normalized fluorescence intensity to the time in hours. **A-A′** Exemplary results of the green fluorescent protein test set predictions. The occurrences based on total occurrence are indicated in the bottom right corner. While 15 of 18 predictions were defined as acceptable (A), 3 of 18 predictions were defined as non-acceptable (A′). **B-B′** Exemplary results of the red fluorescent protein test set predictions. The occurrences based on total occurrence are indicated in the bottom right corner. While 10 of 12 predictions were defined as acceptable (B), 2 of 12 predictions were defined as non-acceptable (B′). All predictions in Fig. 7.4.

α-Bungarotoxin anesthetized medaka embryos reliably

Similar to the assay described previously the efficacy of three different anesthetics was tested in order to perform *in vivo* imaging of medaka embryos. Therefore, embryos were microinjected with α-Bungarotoxin and *eGFP* mRNA, mock injected with *eGFP* mRNA or collected without injection and treated later. All embryos were dechorionated at developmental stage 28. Dechorionated embryos were either untreated or treated, depending on whether or not they were previously injected, respectively. The treatments were performed with tricaine, a standard anesthetic for teleostei, etomidate, a commonly used human anesthetic, dimethyl sulfoxide (DMSO), as solvent control for etomidate and embryo rearing medium (ERM), as negative control. All embryos were transferred to a 96-well plate in 150 µl medium and imaged for at least 60 h.

Due to varying starting stages of embryos at the start of imaging all imaged plates were adjusted to the latest starting stage in order to enable fusion of the datasets. Subsequently, semi-automated image analysis has been performed in Fiji and R. An overview of the performed analysis is depicted in Fig. 3.5A. A normalized movement index supplying a relative readout of movement between tps, was the resulting parameter following analysis. The normalized movement index was obtained by squaring the difference between

tp n and tp $n+1$, very similar to the Euclidean distance (visual depiction in Fig. 3.5A).

Time course analysis of the normalized movement index indicated that embryo movement remained nearly unaffected by etomidate and tricaine treatment in 20 min intervals in comparison to the controls (Fig. 3.5B). This does not mean, however, that these embryos are continuously moving over time, since the images were acquired in 20 min intervals. In contrast, injection of α-*Bungarotoxin* mRNA leads to a strong reduction in the normalized movement index (Fig. 3.5B). This was also observed qualitatively in the corresponding wells (data not shown, online at https://doi.org/10.1371/journal.pone.0212956.s005). Under this treatment paradigm, however, only voluntary muscle movements are suppressed since α-Bungarotoxin acts on neuromuscular junctions (NMJs), which means that early, Ca^{2+}-induced yolk contractions were not suppressed.

Taken together, α-Bungarotoxin was reliably anesthetizing medaka embryos.

Anesthesia with α-Bungarotoxin was partially reversible

To assess whether embryos anesthetized by α-Bungarotoxin are surviving, imaged embryos were demounted from 96-well plates and assayed by a startle response regime at later tps. In brief, embryos were startled 10 consecutive times, each with a pipette tip and the startle responses were recorded. This assay was performed at 6, 8, 12 and 13 dpf. Strikingly, the surviving embryos showed a significant difference in response to both wt and mock injected controls at the beginning of the experiment. Over time, however, most effects of the anesthetic wore off in the surviving α-Bungarotoxin mRNA injected embryos (Fig. 3.5C). Notably, a fraction of α-*Bungarotoxin* mRNA injected embryos died, possibly due to starvation caused by complete anesthesia.

Summarizing, anesthesia with α-Bungarotoxin was partially reversible, but it must be taken into account that a fraction of fish also died from starvation.

The optimal concentration for α-*Bungarotoxin* mRNA injection in medaka was between 12 and 25 ng/µl

To assess the concentration-dependency of anesthesia with α-*Bungarotoxin* mRNA, serial dilutions of the original concentration (25 ng/µl) were microinjected. Strikingly, the number of hatched embryos was lowest at a concentration of 12 ng/µl α-Bungarotoxin mRNA and not 25 ng/µl α-Bungarotoxin mRNA (Fig. 3.5D). Furthermore, fewer embryos were actively swimming when injected with 12 ng/µl as opposed to 25 ng/µl α-*Bungarotoxin* mRNA (Fig. 7.5C).

In conclusion, the supposedly optimal injection concentration of α-*Bungarotoxin* mRNA lay between 12 and 25 ng/µl. This estimate takes the lethality rate and the degree of anesthesia into account.

Cardiac development and heart rate remained unaffected by anesthesia via α-Bungarotoxin

The major disadvantage of long-term tricaine treatment of teleostei was the ineffectiveness of anesthesia. An additional disadvantage was its adverse effect on cardiac development. To exclude that α-*Bungarotoxin* mRNA injections also impact on cardiac development, the previously imaged and long-term treated fish (Fig. 3.5B) were demounted for examination of gross cardiac morphology. While mild and strong cardiac defects were detected in tricaine- and etomidate-treated embryos, respectively, no defects were observed in α-*Bungarotoxin* mRNA injected embryos (Fig. 7.5A). Furthermore, heart rate (HR) recordings of these fish were taken through short videos (10 s, 25 frames per second (fps)) at 25 °C. No difference in HR was detected between α-*Bungarotoxin* mRNA and mock injected embryos (Fig. 7.5B).

Summarizing, no cardiac defects were observed when anesthetizing medaka embryos with α-Bungarotoxin.

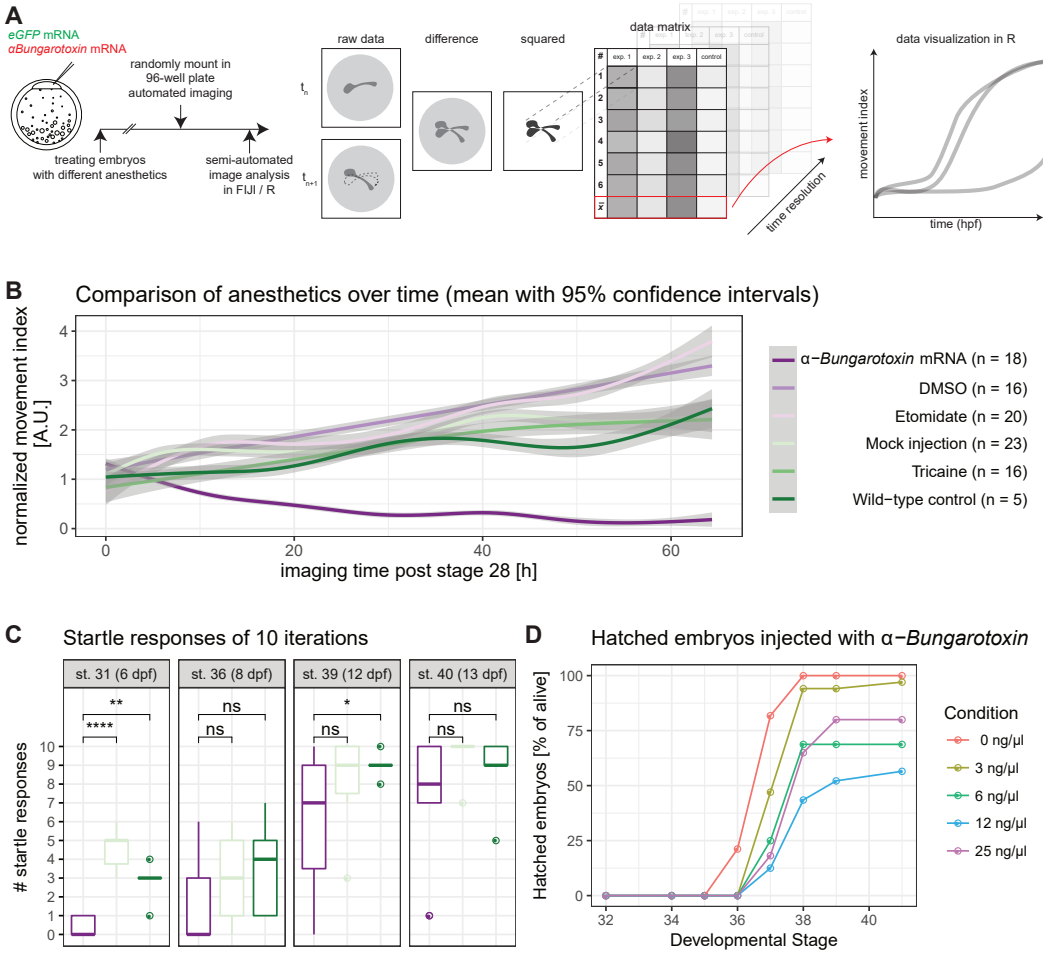

Figure 3.5: *α-Bungarotoxin* mRNA microinjection anesthetized medaka embryos long-term and partially reversible.
A Scheme of the conducted analysis. A subset of medaka zygotes were microinjected with *eGFP* or *α-Bungarotoxin* mRNA. The uninjected embryos were treated and imaged together with the injected embryos. One embryo each was loaded into a single well of a 96-well plate by a pre-defined randomized loading scheme and imaged for at least 60 h. The difference over time of the acquired images was obtained and the resulting images were squared, yielding a normalized movement index. This index was plotted in R (1-cell stage modified from [Iwamatsu, 2004]). **B** Treatment with α-Bungarotoxin was anesthetizing all embryos robustly, while tricaine and etomidate treatment was not distinguishable from the controls. Multiple plates were imaged and normalized to stage 28 [Iwamatsu, 2004]. The normalized movement index of embryos was plotted over time. Time resolution 20 min. n-values indicate fish analyzed per treatment condition. **C** A startle response assay reveals that anesthesia with α-Bungarotoxin is partially reversible. Fish were startled 10 consecutive times with the aid of a pipette tip. Response times were recorded. While α-Bungarotoxin-treated fish did not respond initially, the responses increased over time. The same color legend as in B applies. α-Bungarotoxin n = 12 fish, mock injected: n = 5 fish, wild-type control: n = 8 fish. **D** Hatching of fish is suppressed by α-Bungarotoxin. While control-injected embryos hatched completely, embryos injected with 12 ng/µl *α-Bungarotoxin* mRNA hatched less often. (0 ng/µl: n = 33 fish, 3 ng/µl: n = 34 fish, 6 ng/µl: n = 16 fish, 12 ng/µl: n = 24 fish, 25 ng/µl: n = 22 fish). Asterisks indicate P-values: **** P <= 0.0001, *** P <= 0.001, ** P <= 0.01, * P <= 0.05, ns P > 0.05. Figure from [Lischik et al., 2019].

Medaka pigmentation was optimized for *in vivo* imaging by CRISPR/Cas9

Finally, to eliminate imaging-interfering pigmentation of the embryo, pigment knockouts were established. The CRISPR/Cas9-system was used to introduce mutations in the genome by inducing double-strand breaks (DSBs), which in turn induce insertions and deletions (InDels). The specificity of the Cas9 protein and therefore the location of DSBs is mediated by a variable single guide RNA (sgRNA), which is easily synthesized in the laboratory. Several sgRNAs targeting previously published genes involved in pigmentation pathways were deployed in several combinations as listed in Table 3.1.

Table 3.1: sgRNA combinations deployed in CRISPR/Cas9 experiments.

	oca2			pnp4a			tyr		pax7a			slc2a15b		resulting
mix	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**1**	**2**	**3**	**1**	**2**	**line**
op_1	x	x		x										*spooky*
op_2	x	x	x	x	x	x								*spooky*
o	x	x	x											*oca2*$^{-/-}$
t							x	x						*tyr*$^{-/-}$
p									x	x	x			*pax7a*$^{-/-}$
s												x	x	*slc2a15b*$^{-/-}$
ops	x	x	x	x	x	x						x	x	*spookiest*
uninj.														iCab

Combining *oca2* and *pnp4a* mutations created a pigment-less *in vivo* imaging line (*spooky*)

In order to facilitate *in vivo* imaging, sgRNAs targeting oculocutaneous albinism II (*oca2*) and purine nucleoside phosphorylase (*pnp4a*) (mixes op_1 and op_2) were microinjected. For both genes mutants stemming from mutation screens were published previously. The combination of sgRNAs targeting both genes lead to nearly pigment free medaka with remaining leucophore autofluorescence. In the mosaic, injected generation 12 % (op_1) or 77 % (op_2) were classified suitable for immediate imaging (Fig. 3.6B).

Fish were incrossed to the filial generation (F1), resulting in compound heterozygotes. In this filial generation (F1) a drastic difference was detected qualitatively in comparison to wt by assessing the pigmentation of the eyes in embryonic and adult stages and pigmentation of the operculum and the peritoneum in adult stages (Fig. 3.6D). The added value of the oca2 and pnp4a double KO (*spooky*) in comparison to the knockout (KO) of *oca2* is also very easily visible by assessing the pigmentation of F1 adults (Fig. 7.6). *oca2* KO embryos retain a complete cover with iridescent pigment, hence the operculum, the peritoneum and the retina remain opaque. This constitutes the most striking differences to *spooky*.

Taken together, the *spooky* mutant enhanced imaging of previously obstructed tissues.

The probability of KO positively correlated with the amount of injected sgRNAs per gene

To investigate whether an increased number of sgRNAs targeting the same locus increases the percentage of resulting KOs various combinations of sgRNAs were used in the injection mixes. Injection mixes as indicated in Table 3.1 were microinjected together with Cas9 mRNA. Dead and malformed embryos were removed and embryos were scored for imaging suitability in F0 at stage 30. Embryos injected with sgRNAs targeting *oca2*, *pnp4a* or tyrosinase (*tyr*) were scored in a bright field setup. On the other hand, embryos injected with sgRNAs targeting paired box 7a (*pax7a*) or solute carrier family 2 (facilitated glucose transporter), member 15b (*slc2a15b*) [Kimura et al., 2014] were scored in the green fluorescence channel (Fig. 3.6A).

The percentage of non-developmentally impaired embryos suitable for imaging was visualized (Fig. 3.6B). Strikingly, increasing the number of sgRNAs targeting the same locus from mix op_1 to op_2 led to a higher prevalence of fish suitable for imaging, but also to a higher mortality rate. When looking at mosaic KO embryos with impaired melanin synthesis, *oca2* and *tyr* (injection mixes o and t), the rate of embryos suitable for imaging was higher in *oca2* knockout embryos in F0. Comparing mosaic KOs of genes responsible for the formation of autofluorescent leucophore pigment, *pax7a* and *slc2a15b* (injection mixes p and s), a higher prevalence for imaging suitability was observed in *slc2a15b* mosaic mutant embryos. Therefore, I performed an

injection combining injection mixes op_2 and s to induce mutations resulting in pigmentation-free embryos. These oca2, pnp4a, slc2a15b triple KO (*spookiest*) embryos were devoid of most of the present pigmentation, including melanin in the melanophores, iridophore pigment, leucophore autofluorescence and less carotenoid deposition in xanthophores.

Summing up, an increase of injected sgRNAs targeting the same locus resulted in a higher likelihood of KO.

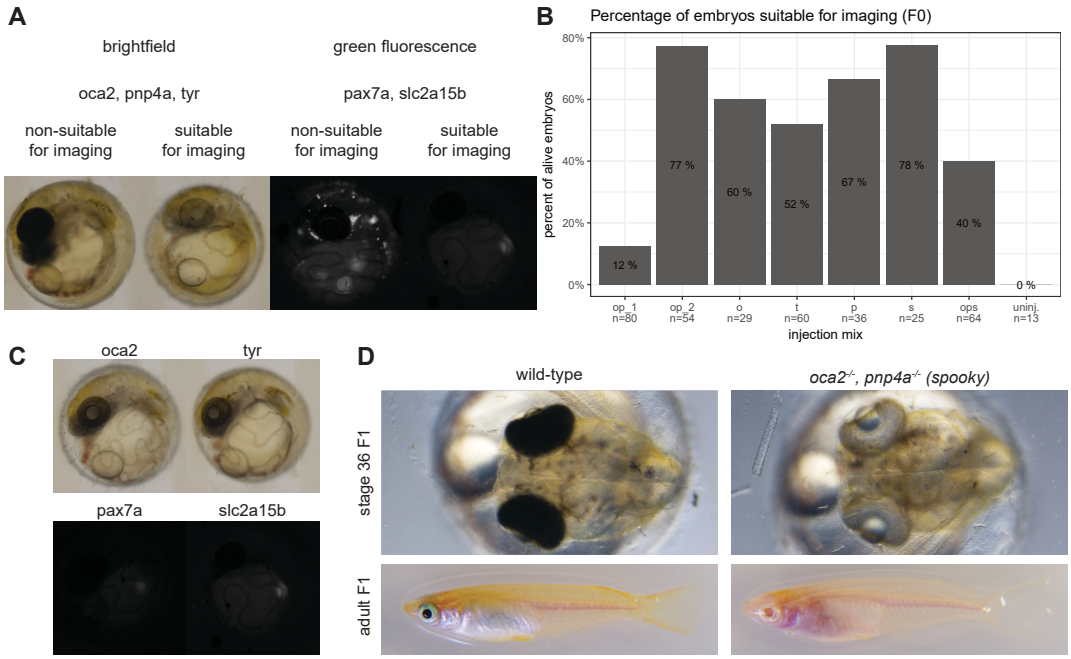

Figure 3.6: Medaka pigmentation mutants created by CRISPR/Cas9 facilitate *in vivo* imaging.
A Medaka zygotes were microinjected with sgRNA mixes as indicated in Table 3.1. Depending on the injected sgRNAs embryos were imaged in a brightfield (*oca2*, *pnp4a* or *tyr*) or green fluorescence (*pax7a*, *slc2a15b*) setup. Embryos were classified as non-suitable or suitable for imaging by loss of pigmentation. **B** Embryos were injected with the indicated injection mixes and classified according to A in the injected generation (F0). An increase of sgRNAs targeting the same locus was shown to be positively correlated with the percentage of knockout embryos (op_1 compared to op_2). n-values indicate the number of injected embryos per condition. **C** Visually no difference of *oca2* and *tyr* or *pax7a* and *slc2a15b* mutants was observed. **D** The *oca2*, *pnp4a* double knockout pigmentation mutant (*spooky*) was created using the CRISPR/Cas9-system. In comparison to wt fish absence of pigmentation in the eyes, the operculum and the peritoneum was observed. Figure from [Lischik et al., 2019].

oca2 KO was superior to *tyr* KO, while *slc2a15b* KO was superior to *pax7a* KO

Subsequent to comparing the prevalence of the KOs depending on the number of injected sgRNAs the question remains, whether there was a difference in pigmentation while targeting different genes responsible for the same pigmentation. Qualitatively comparing the outcome of injection in F0 in presumably fully mutant embryos results in no detectable difference between *oca2* and *tyr* or *pax7a* and *slc2a15b* mutant embryos (Fig. 3.6C). Therefore, other factors were considered, such as the additional effect of *oca2* mutation, which reduced the carotenoid deposition in xanthophores and the less deaths in the injection of sgRNAs targeting *slc2a15b* in comparison to injection of sgRNAs targeting *pax7a*.

In conclusion, melanin pigmentation was best eliminated by *oca2* mutation, while leucophore pigmentation was best eliminated by *slc2a15b* mutation.

In vivo imaging of medaka was greatly enhanced by optimal fluorescent proteins, anesthesia with α-Bungarotoxin and the *spooky* pigment knockout

Utilizing the established toolkit, *in vivo* imaging of medaka was greatly enhanced. For a proof of concept experiment *spooky* and wt embryos were microinjected with α-*Bungarotoxin*, *eGFP* and *mCherry coupled to histone2a (H2A-mCherry)* mRNA. Both mutant and wt embryos, were imaged via SPIM in order to assess the additional value of the *spooky* mutants. Several tissues, that were not accessible for investigation in the wt, e.g. the brain, the eyes, the gut were now accessible in the mutant (Fig. 3.7A-A′). One double mutant *spooky* fish was imaged for 48 h and maximum z projections of this fish were obtained to visualize the greatly enhanced imaging (Fig. 3.7B).

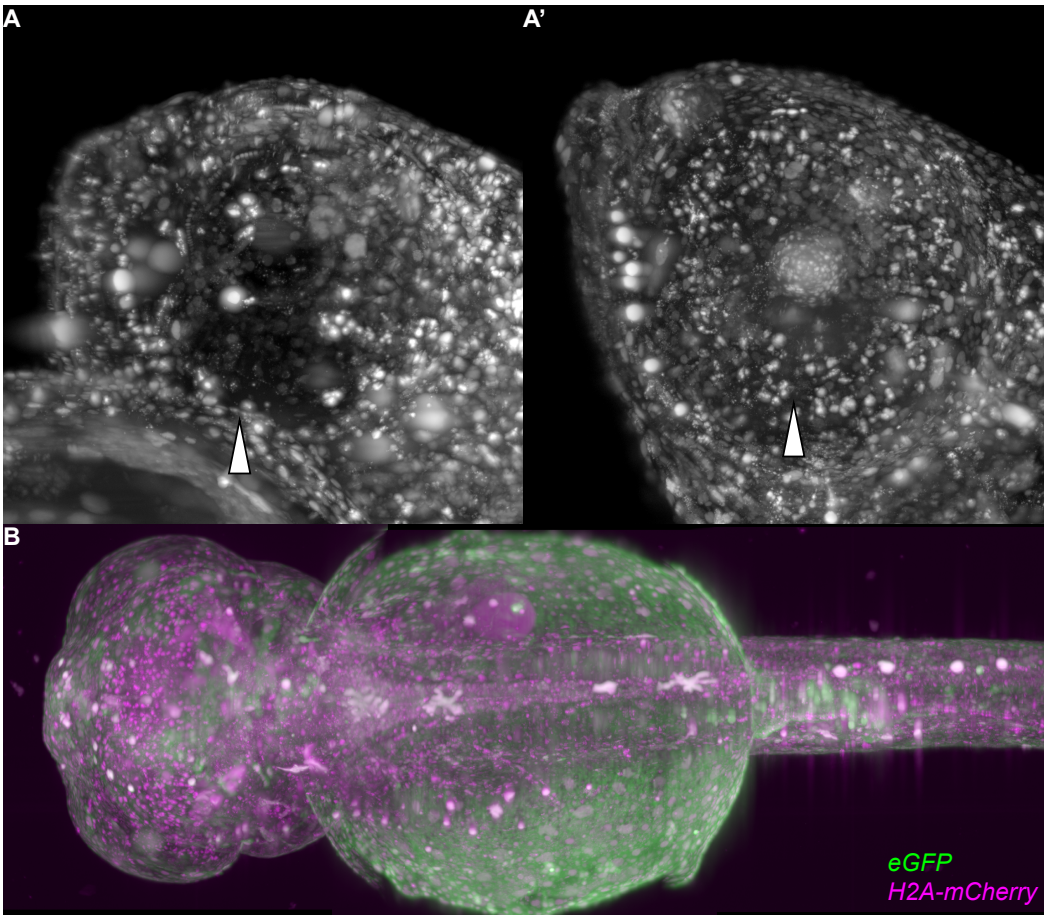

Figure 3.7: The combination of the established tools facilitated *in vivo* imaging of previously opaque structures.
A wt and a *spooky* mutant embryo were microinjected with α-*Bungarotoxin*, *eGFP* and *H2A-mCherry* mRNA and imaged via SPIM. **A** Lateral view of the head of the injected wt embryo. The arrowhead indicates the opaque retina, no nuclei are visible within. **A′** Lateral view of the head of an injected *spooky* embryo. The arrowhead indicates the retina, where nuclei were detectable as a consequence of the pigmentation KO. **B** The embryo from A′ was imaged for 48 h in 1 h intervals. Depicted is a stitched maximum z-projection of the whole body from dorsal, illustrating the increased penetrance, in particular in the head of the embryo. Figure from [Lischik et al., 2019].

In vivo imaging of retinal stem and progenitor cells

α-Bungarotoxin and *spooky* were utilized to perform *in vivo* imaging of retinal stem and progenitor cells

In order to perform *in vivo* microscopy of RSCs, I combined the established tools α-Bungarotoxin and *spooky* with the available GaudíRSG line. The original GaudíRSG construct contains a cyan fluorescent protein (CFP) driven by the crystallin alpha a (*cryaa*) promoter as insertional control. This strongly interferes with retinal *in vivo* imaging due to the strong expression of CFP and resulting high FI directly adjacent to the region of interest (ROI). A sgRNA that specifically targets CFP, but not eGFP, was designed (sgRNA 252 CFP_notGFP). This ensured that H2B-eGFP remained intact for lineage tracing.

GaudíRSG fish were crossed to the *hsp70*:*Cre* driver line and zygotes were microinjected with sgRNAs for a targeted *spooky* and CFP KO. The fish were raised and screened for both pigment and CFP loss. The integrity of H2B-eGFP was confirmed by recombination of individuals with a particularly low CFP expression. No impairment was detected (data not shown). Positively screened fish were incrossed and the progeny was microinjected with α-*Bungarotoxin* mRNA. These embryos were raised to stage 30, recombined and imaged at varying starting tps. This variability was introduced in order to investigate clonal properties of clones of different ages. An overview of the acquired data is provided in Table 3.2. After initial assessment the subsequent analyses were only conducted on data with a magnification of 250 x.

Taken together, the established toolset enabled *in vivo* imaging of RSCs and RPCs.

Table 3.2: Overview of GaudíRSG retinae imaged *in vivo*.

ID	heat shock to imaging [d]	duration [d]	heat shock to final tp [d]	magnification [x]
0	1	2.47	3.47	250
1	1	2.47	3.47	250
2	0	3.82	3.82	160
3	0	3.82	3.82	160
4	0	3.82	3.82	250
5	0	3.82	3.82	250
6	3	2.53	5.53	250
7	3	2.53	5.53	250
8	1	2.71	3.71	160
9	1	2.71	3.71	250
10	1	2.71	3.71	250

Single retinal cells were tracked manually

The present data with a 250 x magnification was used to perform manual single cell tracking (an example is depicted in Fig. 3.8A,B,C). MaMuT was employed to track cells on the raw data in a position of the retina coinciding with RSCs or RPCs. The tracked cells were either determined by proximity to the lens and the surface or, for older retinae, by being the most peripheral cell of a clone (Fig. 3.8A″,B″,C″, black arrowhead). The extensible markup language file (xml file) generated by MaMuT was used to extract the tracked data points. The data points were loaded into custom python scripts and corrected for the z-resolution of the acquired image stack (Scripts see appendix). Subsequently, a three-dimensional (3D) plotting package present in the matplotlib library was deployed to interactively visualize the data points. The 3D plot was supplemented with a slider for selection of the tp, resulting in a four-dimensional (4D) plot. This 4D plot was used for initial data visualization. Additionally, an export function was added to export single 3D plots for visualization (such as in Fig. 3.8).

Abbreviating, single cells were tracked within the acquired retinae and tracks were visualized for initial data screening.

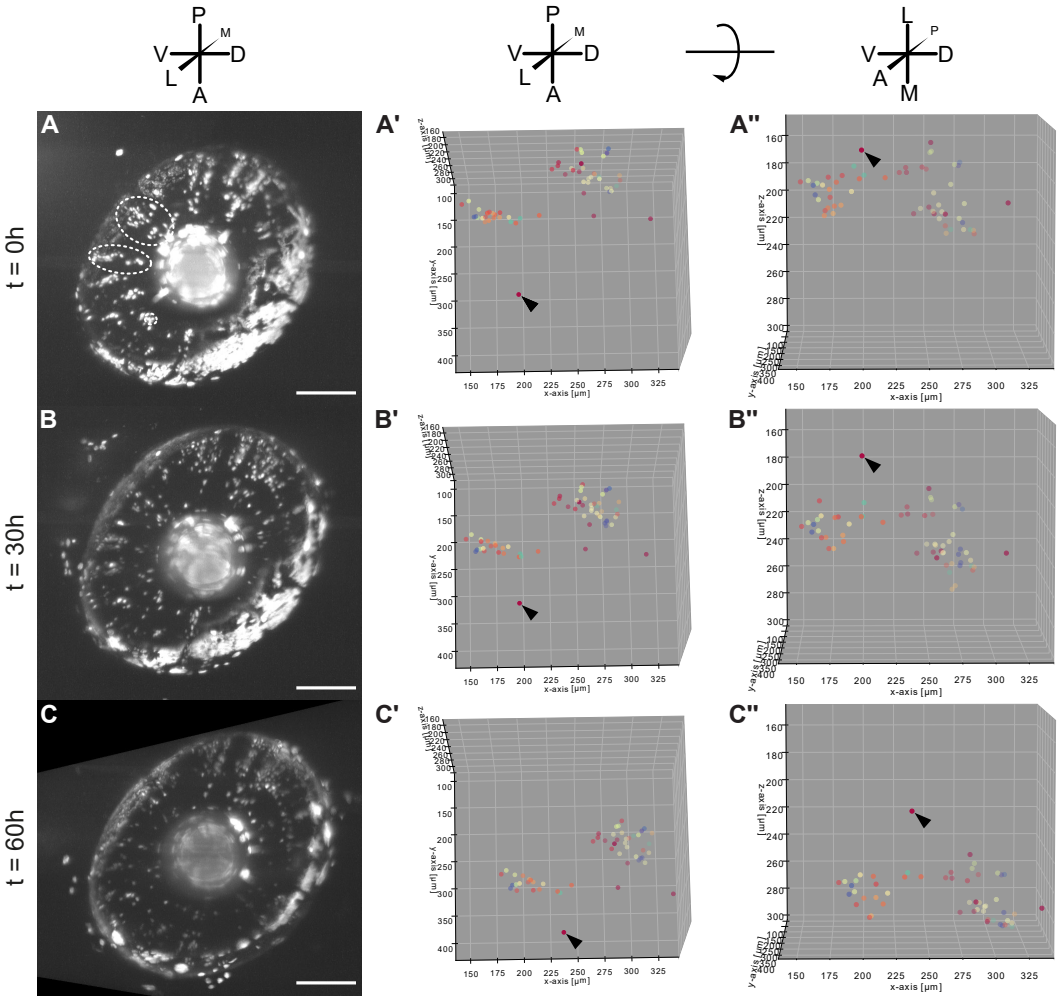

Figure 3.8: ***In vivo*** **single cell tracking of recombined retinae revealed different cell types.**
A fish positive for GaudíRSG, *hsp70*:*Cre* and *spooky* was recombined and imaged at 3 dpi (ID7). The color code indicates the Track_ID, the same color indicates cells descending from the same original cell. **A** Maximum projection of the analyzed retina at tp 0 h. Due to autofluorescence of remaining pigment cells the visualization was logarithmized to level FI differences. Striped ellipses indicate tracked cells. Scale bar 100 µm. **A′-A″** A single, non-dividing cell is remaining at the periphery of the retina (Black arrowhead). Uncorrected 3D visualizations of tracked cells from retina shown in A. A′ is in the same orientation as A. A″ is rotated 90 degrees as indicated at the top. **B-B″** Same as A-A″, but at tp 30 h. The marked cell has not divided. It also has not shifted to the central retina as much as the other tracked cells. **C-C″** Again, same as A-A″, but at tp 60 h. Strikingly, the marked cell did not divide and was located more periphal than the other tracked cells. Anatomical rosettes indicate the orientation of the retinae or 3D plots. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina).

Tracked data was corrected for minor movements

The xml file mentioned previously supplied by MaMuT contained cell positions and cell connections. Additionally, the xml file contained parameters such as velocity and direction of cell movement between tps. These parameters were, however, not accurate since the retinae were not registered to each other. This registration was needed due to minor movements within the imaged retinae caused by growth and space restriction. Hence, it was necessary to post-experimentally correct for these minor movements by software post acquisition. For correction of these minor movements, a rigid or affine correction were applied and the results were compared to each other. While, the rigid correction algorithm introduced movement artifacts, the correction with an affine algorithm corrected and stabilized the global movements satisfactorily. The affine correction, however, also included correction of shearing and scaling. In order to be employed the algorithm needs to be adapted to exclude shearing and scaling of the data, which would result in data inconsistency.

Comprising, two algorithms were tested for global movement correction, but the corrections were not robust and therefore not deployed.

A presumable retinal stem cell was tracked

The cell marked in Fig. 3.8 by a black arrowhead is a presumable RSC or eRPC. This presumption is based on the position of the cell at the periphery of a clone throughout the tracking experiment (Fig. 3.9). Furthermore, the cell does not divide within a time frame of more than 60 h. The overall FI and signal-to-noise ratio (SNR) decreased over time interfering with further tracking after 60 h.

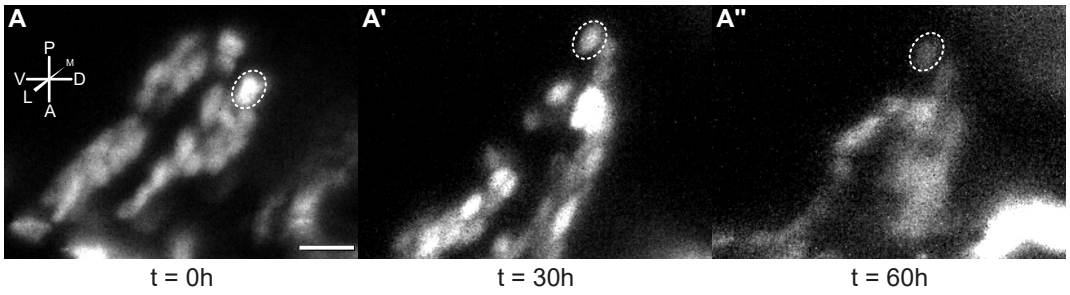

Figure 3.9: A presumable RSC was tracked but did not divide within 60 h. **A-A″** Tracking of a single, presumable RSC residing at the periphery of a clone over 60 h marked by a white striped ellipse (Data ID7). The cell is marked by a black arrowhead in Fig. 3.8. No division was observed within 60 h, while the cell remained at the periphery of the clone. Overall FI and SNR decreased over time. Scale bar 10 µm. Anatomical rosette indicates the orientation of the retina. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina).

Single cell tracking revealed two daughter cell behaviors

Tracking the position of single cells over time revealed at least two distinct daughter cell behaviors present in the retina. In total, six cells dividing during imaging were tracked. Visualized are three of these cells with distinct daughter cell behaviors (stemming from Data ID6). Some cells divide and remain in the vicinity of each other following division (Fig. 3.10A-B′) while others divide and strive away from one another (Fig. 3.10C-C′). On average, however, the three analyzed daughter cells traveled similar distances. In contrast, the distance to the lens and the CMZ tip was smaller in the cell lineage, whose daughters strove away from each other after division. Here, the data was not corrected for global movements in order to not interfere with the distance measurements.

Taken together, by tracking single cells, two distinct daughter cell behaviors were revealed in the retina.

Table 3.3: Overview of GaudíRSG retinae cell tracks visualized in Fig. 3.10 from *in vivo* imaging data (Data ID6). Distances were approximated by testing three different distances to the point of interest and choosing the smallest distance. All values, except TrackID and panel, are represented in µm. dis.: distance.

		origin cell			daughter cells at endpoint		
TrackID	**panel in Fig. 3.10**	**position (x,y,z)**	**dis. to the lens**	**dis. to the CMZ tip**	**positions (x1,y1,z1); (x2,y2,z2)**	**dis.**	**traveled dis.**
1	A	(288,365,224)	31	37	(281,435,146); (271,420,123)	29	105,116 ⌀: 110.5
3	B	(373,344,230)	35	40	(365,415,135); (367,422,129)	9	118,127 ⌀: 122.5
9	C	(390,352,134)	26	25	(385,427,44); (406,383,22)	53	117,117 ⌀: 117

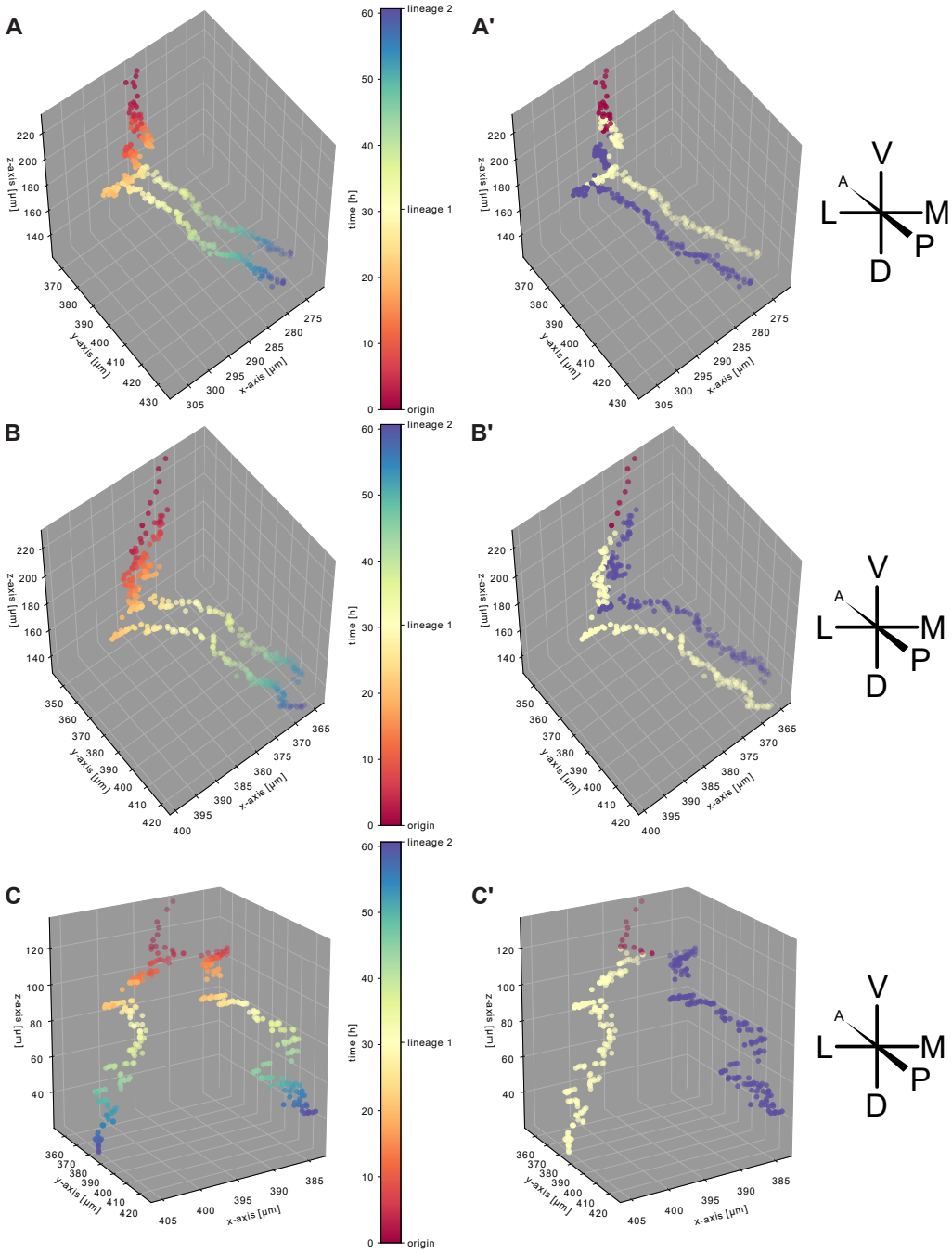

Figure 3.10: Two daughter cell behaviors were observed in the retina.
A fish positive for GaudíRSG, *hsp70*:*Cre* and *spooky* was recombined and imaged at 3 dpi (ID6). Single tracks including all tps were visualized. The first panel of each track shows all cell positions for all time points color-coded for time, whereas the second panel shows the same data, but color-coded by lineages as indicated with the color bar in the center of the image. **A-A′** Daughter cells in TrackID 1 stayed adjacent to one another subsequent to division. **B-B′** Similar to A-A′ daughter cells in TrackID 3 stayed adjacent to each other subsequent to division. **C-C′** Daughter cells in TrackID 9 strove away from each other subsequent to division. Further quantifications are summarized in Table 3.3. Anatomical rosettes indicate the orientation of the cell tracks. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina).

Wnt regulation of retinal stem and progenitor cells

Independent insertions in two separate medaka lines were leveraged to investigate dosage effects of DN-GSK3

Expression levels were approximately 10 times higher in the RSDNGSK3_high line compared to the RSDNGSK3_low line

Preceding experiments were conducted with a transgenic RSDNGSK3 line, which in the following will be referred to as RSDNGSK3_high [Möller, 2017]. Another insertion line was created prior, in the following referred to as RSDNGSK3_low. Most of the experimental within this thesis work has been conducted with the latter line. A qualitative distinction between both lines is already macroscopically visible. While fish of the RSDNGSK3_high line exhibit a distinct red body color, fish of the RSDNGSK3_low line appear wild-typic (Fig. 3.11A-A′). To further substantiate this, semiquantitative PCR was performed on complementary DNA (cDNA) of embryos of both lines and wt in quadruplets. The Polymerase chain reaction (PCR) was loaded onto a gel and a digital image was acquired without oversaturation. The bands were quantified and the background bands of the wt embryos were subtracted. Previous calibration of the PCR by 10-fold dilution enabled the estimation of the ratio of expression levels. The results indicated a significantly lower expression of mCherry in RSDNGSK3_low embryos compared to RSDNGSK3_high embryos (Fig. 3.11B). The expression difference was estimated to $\approx$ 9.7-fold comparing expression of RSDNGSK3_high to expression of RSDNGSK3_low insertions. The expression of *mCherry* was thereby quantified before recombination in order to exclude effects of Wnt stimulation on expression levels.

In summary, the expression difference of available lines was leveraged to investigate the dosage effect of Wnt stimulation.

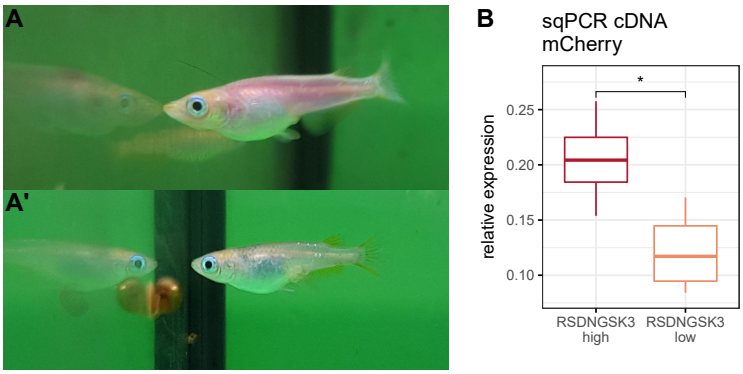

Figure 3.11: Two independent insertions of RSDNGSK3 were leveraged to investigate dosage effects of Wnt stimulation.
A-A′ Macroscopic view of fish of both transgenic lines. Fish of the RSDNGSK3_high line (A) appeared to have a red body color, indicating higher expression levels. Fish of the RSDNGSK3_low line (A′) appeared to have a wild-typic body color, indicating lower expression levels. Depicted fish are not age-matched, no objective size difference was observed between fish of both insertion lines. **B** The difference in mean expression of mCherry in both lines estimated to ≈ 9.7-fold. Semiquantitative PCR on cDNA of both lines with primers for mCherry was performed. n = 4 fish for each condition. Asterisks indicate P-values: * P <= 0.05.

ccl25b:*CreERT2* recombination led to results comparable with *rx2*:*CreERT2* recombination

The preceding work utilized the *rx2*:*CreERT2* transgenic line for stem cell-specific recombination. Due to experimental constraints of this line, however, the experiments presented within this thesis needed to be conducted with another Cre driver line. The *ccl25b*:*CreERT2* line was used for recombination. To ensure comparability, fish with an insertion of GaudíRSG or RSDNGSK3_high were crossed to *ccl25b*:*CreERT2* fish. These fish were recombined by tamoxifen addition at hatch and chased for 2 days to 4 weeks. Strikingly, the results were very similar to the results obtained with *rx2*:*CreERT2* [Möller, 2017]. Post hoc analysis of GaudíRSG retinae revealed a large amount of maintained clones (Fig. 3.12A). Analysis of RSDNGSK3_high retinae revealed no detectable clones (Fig. 3.12A′). Both of these results are comparable to the results of the precedence work [Möller, 2017] (Fig. 1.4A,A′).

Comprising, the newly introduced *ccl25b*:*CreERT2* line led to results comparable to the *rx2*:*CreERT2*.

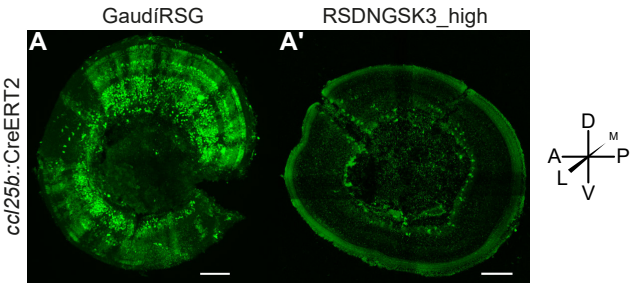

Figure 3.12: Recombination of RSDNGSK3_high with *ccl25b:CreERT2* led to results comparable to recombination with *rx2:CreERT2*.
A GaudíRSG fish were crossed with fish positive for *ccl25b:CreERT2*. Embryos were recombined with tamoxifen at hatch and chased for 2 weeks at 26 °C. Maintained clones were present throughout the retina. Scale bar 100 µm. **A′** RSDNGSK3_high fish were crossed with fish positive for *ccl25b:CreERT2*. Embryos were recombined with tamoxifen at hatch and chased for 2 weeks at 26 °C. No clones were detectable in the retina. Scale bar 100 µm.

High Wnt stimulation led to loss of clones

The difference in clone loss between the RSDNGSK3_low and RSDNGSK3_high lines was assessed. Fish of the GaudíRSG, RSDNGSK3_low and RSDNGSK3_high lines were crossed with fish of the *ccl25b:CreERT2* and *tlx:CreERT2* lines. Fish were recombined at hatch via tamoxifen, raised at 24 °C and fixed at various timepoints (2, 3, 7 or 14 days post induction (dpi)). Retinae were stained for *rx2* as well as for eGFP and screened initially with a Leica Sp8 microscope for polyclone presence by using the live view. Polyclones were defined as patches of cells, continuously connected from the most lateral to the most medial end. These polyclones consist of single or multiple clones, which is why they will be in the following referred to as polyclones. Confocal stacks were acquired from retinae positive for polyclones for further analysis. The percentage of retinae positive for any recombined cells was visualized (Fig. 3.13). The presence of recombined cells increased over time up to 100 % in GaudíRSG and RSDNGSK3_low retinae. In RSDNGSK3_high retinae, however, recombined cells were not detectable at all subsequent to recombination by *ccl25b:CreERT2* or after two weeks subsequent to recombination by *tlx:CreERT2*. A more detailed characterization of polyclones in GaudíRSG and RSDNGSK3_low retinae follows.

Taken together, while wt retinae and retinae exposed to low Wnt stimulation formed clones, clones were absent in retinae exposed to high Wnt stimulation.

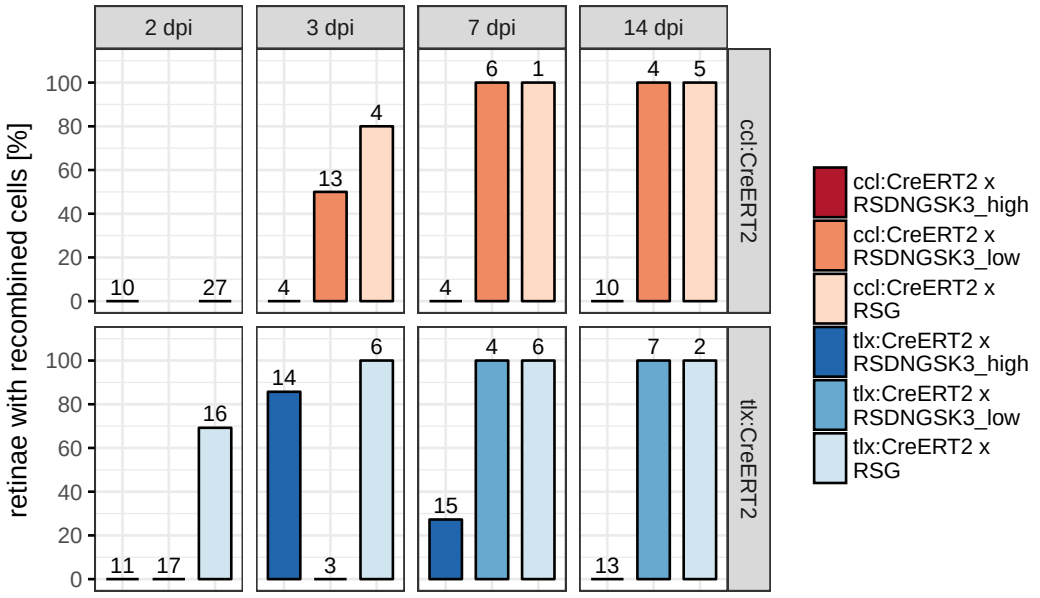

Figure 3.13: High Wnt stimulation led to loss of polyclones.
Fish of the GaudíRSG, RSDNGSK3_low and RSDNGSK3_high lines were crossed with fish of the *ccl25b*:*CreERT2* and *tlx*:*CreERT2* lines. Fish were recombined with tamoxifen at hatch, grown at 24 °C and fixed at the indicated tp (dpi). Retinae were imaged and screened for any cells positive for fluorescence. They were categorized into positive and negative retinae. Illustrated is the percentage of retinae with detectable recombined cells per condition. Indicated on top of the bar graphs is the number of retinae analyzed per condition. Notably, all retinae of various experimental conditions contain increasingly more polyclones over time, except for retinae of the RSDNGSK3_high line.

Apoptosis of cells exposed to high Wnt stimulation caused polyclone loss

Next, experiments were performed to identify the cause of the observed polyclone loss in the RSDNGSK3_high line. Among others, apoptosis and a change of division mode of cells exposed to high Wnt stimulation were reasonable possibilities. In order to address this question the established *in vivo* imaging toolset was utilized. The progeny of a RSDNGSK3_high and the *hsp70*:*Cre* cross were microinjected with sgRNAs for a targeted *spooky* KO. Embryos were screened for pigment loss and raised to fertility (Fig. 3.14A). The adult fish were crossed, and zygotes were microinjected with α-*Bungarotoxin* mRNA. These embryos (F1) were raised to stage 34 and recombined by heat shock in a thermal cycler. The thermal cyler protocol is very robust and adaptable to the needs of the experiment. The number of cycles as well as the severity of heat shock can be adapted easily and reproducibly. The heat shocked fish were dechorionated and subsequently two embryos (fish A and

B) were mounted for SPIM and one embryo was imaged via SPIM (fish A, Fig. 3.14A).

Long-term imaging was performed via SPIM. The resulting data were visualized with the same viewing range over all tps and indicated a dramatic loss of FI in recombined cells over time (Fig. 3.14B-B‴). At the final timepoint of imaging only few cells positive for eGFP were detected (Fig. 3.14C, white arrowheads). Even more so, these residual cells are only detectable after performing a maximum z projection and decreasing the viewing range in comparison to Fig. 3.14B-B‴.

In order to exclude the possibility that general FI was lost as a consequence of photobleaching the first embryo (fish A) was demounted, stained for eGFP and re-imaged via SPIM. Again, only few cells positive for eGFP remained at the last tp of observation (Fig. 3.14D, white arrowheads). Furthermore, a control embryo (fish B) was handled comparably and subjected to heat shock, mounting and pre-screening in the microscope. This embryo (fish B), however, was not imaged long-term and only mounted at the finalization of imaging of the first embryo (fish A). In the retinae of this embryo also only residual cells remained positive for eGFP fluorescence (Fig. 3.14D′, white arrowheads).

Embryos from the same line as described above were recombined by a thermal cycler at stage 35 and fixed at stage 39 (3 dpi). These embryos were stained for eGFP and Rx2 as well as subjected to the terminal deoxynucleotidyl transferase dUTP nick end labeling (TUNEL) assay to reveal apoptotic cells. Imaging was conducted by a Leica Sp8. Confocal microscopy of these retinae, did not detect eGFP immunoreactivity, whereas signal for the Rx2 antibody indicated the success of the staining procedure. Furthermore, TUNEL signal was prevalent in these retinae, indicating an increased level of apoptosis.

Summarizing, the clone loss observed previously in retinae exposed to high Wnt stimulation was caused by apoptosis.

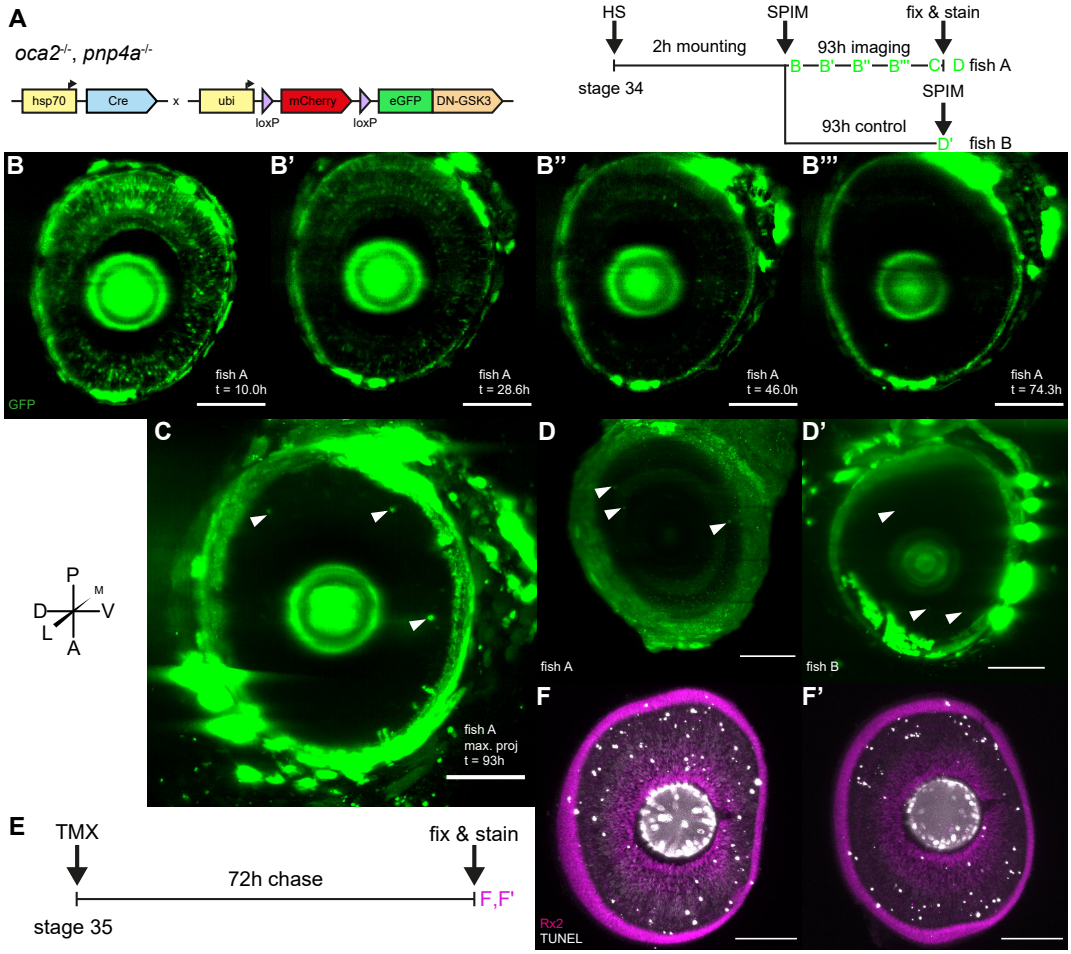

Figure 3.14: Apoptosis caused loss of cells exposed to high Wnt stimulation. **A** Experimental fish were raised as *spooky* mutants with an insertion of *hsp70*:*Cre* and RSDNGSK3_high. Embryos were microinjected with α-Bungarotoxin mRNA. Embryos were heatshocked at stage 34 and either imaged for 93h hours and subsequently fixed and stained (fish A) or treated the same way, but only imaged at the finalization of fish A, fixed and stained (fish B). Colored letters indicated the matching subpanels. Subpanels B-D′ were acquired via SPIM. **B-B‴** Long-term light-sheet microscopy of an RSDNGSK3_high embryo, recombined at stage 34. A severe loss of fluorescent cells was observed over time, presumably as a result of photobleaching or apoptosis of single recombined cells. All 4 subpanels were created with the same pixel viewing range. Scale bar 100 µm. **C** The final tp of fish A indicates few surviving cells, which were only detectable by a maximum z projection and a very low viewing range. The autofluorescent pigments appear therefore more intense compared to the prior visualization. Scale bar 100 µm. Filled arrowheads mark residual positive cells. **D-D′** Loss of fluorescence is not caused by photobleaching. **D** The same embryo as in B and C has been demounted and stained for eGFP in order to assess the presence of bleached eGFP. As previously only residual cells were detected. Scale bar 100 µm. Filled arrowheads mark residual positive cells. **D′** A control embryo treated similar to fish A showed also only residual cells positive for *in vivo* fluorescence. Scale bar 100 µm. Filled arrowheads mark residual positive cells. **E** Fish with the same genetic background as shown in A were recombined at stage 35 and chased for three days up to stage 39 (3 dpi (equivalent to chasing until B‴ during imaging). These fish were fixed and stained for eGFP, Rx2 and apoptotic cells. Colored letters indicate the matching subpanels. Images were acquired via Leica Sp8. **F-F′** Two RSDNGSK3_low retinae positive for TUNEL, indicating apoptosis of recombined cells. Scale bar 100 µm. Anatomical rosettes indicate the orientation of retinae. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina).

The proliferative capacity of RSCs and eRPCs was decreased by low Wnt stimulation

As shown previously the recombination in RSCs and eRPCs in the RSDNGSK3_high line resulted in a high loss of polyclones. In order to assess the effect of Wnt stimulation on single RSCs or eRPCs and their progeny the RSDNGSK3_low line was employed to overcome this challenge at a lower level of Wnt stimulation. The RSDNGSK3_low line was recombined in RSCs (*ccl25b*-positive cells) or a combination of RSCs and eRPCs (*tlx*-positive cells, all analyzed retinae are listed in Table 7.3). The resulting polyclones showed an apparent qualitative difference between control and experiment. When comparing recombined control GaudíRSG and experimental RSDNGSK3_low retinae the clones appeared narrower in the experiment (Fig. 3.15A-B′). Furthermore, subsequent to recombination with *ccl25b*:*CreERT2* and two weeks of chase only polyclones connected with the CMZ were present in GaudíRSG retinae (Fig. 3.15C). The connection with the CMZ indicates a polyclone maintained by a RSC or RPC. In RSDNGSK3_low retinae, however, a mixture of polyclones connected with and disconnected from the CMZ was observed (Fig. 3.15C′-C″). Furthermore, polyclone morphology seemed to change qualitatively after Wnt stimulation of RSCs or eRPCs (Fig. 3.15D-E′).

For further investigation 15 parameters were quantified for each positive retina in Table 7.3 (full quantifications in Fig. 7.7 and Fig. 7.8). It has to be noted that no change in cell type composition of the polyclones has been observed, i.e. all clones were founded by a multipotent cell and all cell types were included in all polyclones. In the following I will elaborate on the minor differences between polyclones. The distance of terminating clones to the CMZ did only change slightly in the retinae recombined with *tlx*:*CreERT2*. This was only the case in the maximum distance of clones to the CMZ (indicating the earliest terminating clone in the retina Fig. 7.7D), while the minimum distance of clones was unaltered (indicating the latest terminating clone in the retina Fig. 7.7C). Due to the spatiotemporal properties of the retina it can be deduced that therefore the time of clone termination remains unaltered. The total number of polyclones decreased upon Wnt stimulation when recombining with *ccl25b*:*CreERT2* (Fig. 7.7E, Fig. 7.8E). In contrast, no change of total number of polyclones was observed when recombining with *tlx*:*CreERT2*. Whether this was also an apoptosis effect that is more present in RSCs or an artifact of polyclone detection remains to be elucidated.

A robust trend for retinal diameter was only observed at later timepoints, where the retinae recombined with *ccl25b*:*CreERT2* increased in size upon Wnt stimulation of single cells. While there was no significant difference at 14 dpi, there was a significant difference at both 21 and 28 dpi (Fig. 7.8I). The cause for this increase in size remains unclear and needs to be elucidated in future work. Nothing could be concluded from the quantification of the amount and the starting position of late starting clones. Late starting clones were defined as clones connected with the CMZ but not being connected to the induction area, indicated by the accompanying longer clones in the retinae (indicating a partially quiescent founding cell, Fig. 7.7J-L,O, Fig. 7.8J-L,O).

The most prominent differences were an increase of terminating clones and a decrease of clone width upon low Wnt stimulation. This held true for both RSCs and eRPCs. The detailed quantification method for both parameters is illustrated in Fig. 3.15F,G. Strikingly, the low stimulation of Wnt in single RSCs or RPCs led to fewer polyclones connected with the CMZ in comparison to wt (Fig. 3.15I) and narrower clonal stripes (Fig. 3.15J-J′).

Taken together, the proliferative capacity of RSCs and eRPCs is decreased by low Wnt stimulation. The differentiation potential however is unaltered.

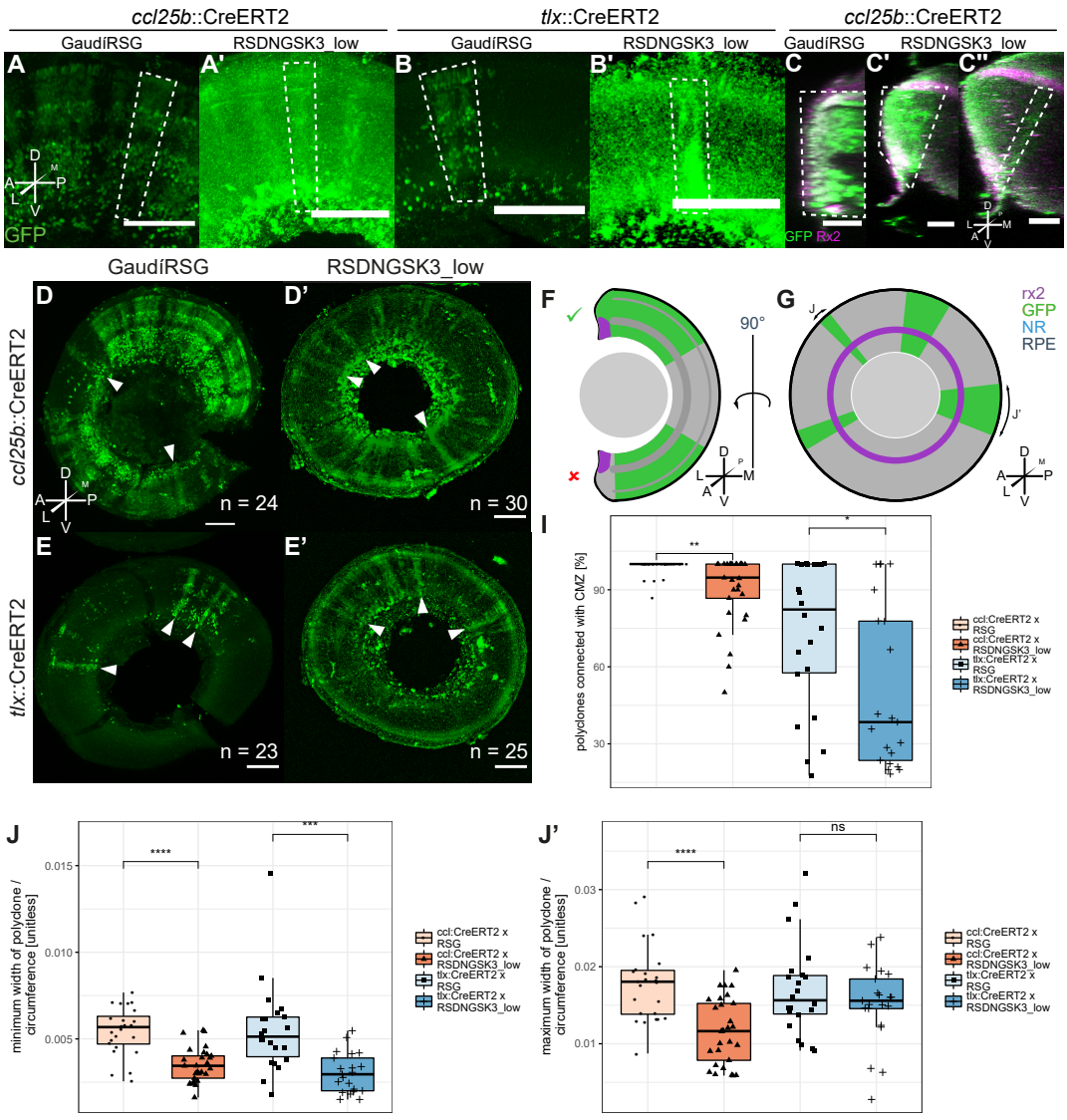
ccl25b::CreERT2
tlx::CreERT2
ccl25b::CreERT2
GaudíRSG
RSDNGSK3_low
GaudíRSG
RSDNGSK3_low
GaudíRSG
RSDNGSK3_low
A
A'
B
B'
C
C'
C''
GFP
GFP Rx2
GaudíRSG
RSDNGSK3_low
ccl25b::CreERT2
tlx::CreERT2
D
D'
E
E'
n = 24
n = 30
n = 23
n = 25
F
90°
G
rx2
GFP
NR
RPE
I
polyclones connected with CMZ [%]
ccl:CreERT2 x RSG
ccl:CreERT2 x RSDNGSK3_low
tlx:CreERT2 x RSG
tlx:CreERT2 x RSDNGSK3_low
J
minimum width of polydone / circumference [unitless]
J'
maximum width of polydone / circumference [unitless]
ns

Figure 3.15: (Opposite page) Low Wnt stimulation decreased the proliferative capacity of RSCs and eRPCs.
Fish of the GaudíRSG and RSDNGSK3_low line were crossed with *ccl25b*:*CreERT2* or *tlx*:*CreERT2*. At hatch fish were treated with tamoxifen to induce recombination. The fish were raised for 1 to 4 weeks. Subsequently, fish were fixed, stained and imaged in a Leica Sp8. **A-A′** Polyclones were narrower in RSDNGSK3_low compared to GaudíRSG (white quadrilaterals). Maximum z projections of GaudíRSG (A) or RSDNGSK3_low (A′) retinae imaged from lateral. The retinae were recombined by *ccl25b*:*CreERT2* and chased for 2 weeks. Scale bar 100 µm. Anatomical rosette in A valid for A-B′. **B-B′** Again, polyclones were narrower in RSDNGSK3_low compared to GaudíRSG (white quadrilaterals). Maximum z projection of GaudíRSG (B) or RSDNGSK3_low (B′) retinae imaged from lateral. Retinae were recombined by *ccl25b*:*CreERT2* and chased for 2 weeks. Scale bar 100 µm. **C-C″** No GaudíRSG polyclones did terminate and therefore all polyclones form polyclonal patches connected with the CMZ (C). RSDNGSK3_low polyclones, however, consisted of a mixture of polyclones connected with the CMZ which were not terminated (C′) and polyclones disconnected from the CMZ which were terminated (C″). Representative polyclones of GaudíRSG (C) and RSDNGSK3_low (C′-C″) retinae recombined with *ccl25b*:*CreERT2*, which have been chased for 2 weeks. Depicted are orthogonal sections. The white quadrilaterals mark single polyclones. Scale bar 50 µm. Anatomical rosette in C″ valid for C-C″. **D-D′** Lateral view of GaudíRSG (D) and RSDNGSK3_low (D′) retinae recombined by *ccl25b*:*CreERT2* (maximum z projections). Retinae have been chased for 2 weeks. Polyclones were detected in both, GaudíRSG and RSDNGSK3_low retinae. Solid arrowheads mark exemplary polyclones. Scale bar 100 µm. Anatomical rosette in D valid for D-E′. n-values indicate the total number of analyzed retinae per condition. **E-E′** Lateral view of GaudíRSG (E) and RSDNGSK3_low (E′) retinae recombined by *ccl25b*:*CreERT2* (maximum z projections). Retinae have been chased for 2 weeks. Polyclones were detected in both, GaudíRSG and RSDNGSK3_low, retinae. Solid arrowheads mark exemplary polyclones. Scale bar 100 µm. n-values indicate the total number of analyzed retinae per condition. **F** Present polyclones have been categorized into being connected with or disconnected from the CMZ. Categorization was achieved by assessing overlap of the polyclone with the *rx2* expression domain. The results are depicted in panel I. **G** Polyclone widths were quantified for all polyclones. The minimum and maximum width of polyclones per retina was normalized to the circumference and visualized in J and J′. **I-J** Plots depicting major diverging parameters comparing controls and experiments. **I** The percentage of polyclones connected with the CMZ decreased drastically upon stimulation of the β-catenin dependent Wnt pathway within the polyclone. This indicates a decreased proliferation potential of recombined cells. **J-J′** The minimum (J) and maximum (J′) width of polyclones normalized to the circumference decreased significantly upon the stimulation of the β-catenin dependent Wnt pathway, again indicating a change in proliferation potential. Asterisks indicate P-values: **** $P <= 0.0001$, *** $P <= 0.001$, ** $P <= 0.01$, * $P <= 0.05$, ns $P > 0.05$. Anatomical rosettes indicate the orientation of microscopy images and schemes. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (NR center), L: lateral (NR periphery).

4
Discussion

In vivo imaging of medaka was enhanced

Fluorescent proteins were assayed *in vivo*

mGFPmut2/eGFP and mCherry are the fluorescent proteins of choice in medaka

While mGFPmut2 was the FP with the highest overall FI of green FPs, eGFP, one of the most popular FPs, was still ranking third in FI in medaka (Fig. 3.1C). Depending on the experiment both, mGFPmut2 or eGFP are suitable for the experimental procedure. For exclusive *in vivo* imaging, mGFPmut2 is superior to eGFP. It also harbors the A206K mutation, abolishing multimerization [Zacharias et al., 2002]. This renders it suitable for endogenous tagging approaches such as CRISPR/Cas-based homology-directed repair (HDR)-mediated genomic insertions [Gutierrez-Triana et al., 2018]. In contrast, a large toolset has already been established for eGFP, including antibodies, nanobodies, split-GFP, etc. [Kubala et al., 2010, Caussinus et al., 2011, Kamiyama et al., 2016]. In conclusion, mGFPmut2 can be utilized for all experiments that rely on high *in vivo* FI and eGFP can be utilized for experiments which require another downstream interaction factor or staining.

For red FPs mCherry is clearly outstanding in terms of FI and applicability (Fig. 3.1D). It also offers a large variety of tools such as nanobodies, antibodies, etc. [Katoh et al., 2016]. As a monomeric FP it is also suitable for HDR-mediated endogenous tagging methods [Gutierrez-Triana et al., 2018].

The *in vivo* assay was necessary and sufficient to investigate fluorescent protein properties for light-sheet microscopy

A previous study in *E. coli* has shown a direct correlation of a set of parameters acquired *in vitro* and *in vivo* FI [Balleza et al., 2017]. As shown here this was not the case in medaka and therefore the established *in vivo* assay was necessary to test novel FPs (Fig. 3.1E). This is in concordance with a previous study which showed that there is no direct correlation of *in vitro* and *in vivo* FIs of FPs in *Caenorhabditis elegans* [Heppert et al., 2016]. The cause of the divergence of *in vitro* and *in vivo* FIs has so far not been investigated.

Testing of a subset of investigated FPs via SPIM furthermore revealed that the established assay is sufficient to predict *in vivo* FIs for LSFM.

Codon adaptation decreased eGFP *in vivo* fluorescence intensity

Codon usage table-driven codon adaptation of eGFP for medaka was decreasing FI by 25 to 30-fold (Fig. 3.2C). The exact reasons for this decrease are unknown, but are more extensively discussed in my recent publication [Lischik et al., 2019]. This is in contrast to prior studies indicating that codon usage table-driven codon adaptation was beneficial for FI of eGFP in *Ciona intestinalis* [Zeller et al., 2006].

The results of the *in vivo* assay are not directly transferable from medaka to zebrafish

The results of the *in vivo* assay cannot directly be transferred from medaka to zebrafish (Fig. 3.3B-C). The causes for this difference are yet unknown, but differential time of MBT (cell cycle 10, ≈1024 cells in zebrafish and 64-cell stage in medaka) [Kane and Kimmel, 1993, Kraeussling et al., 2011], different metabolism or differential mechanisms of mRNA stability are likely causes for the apparent differences.

An ANN performed Classification and prediction of time course continuation

The classification of time courses to a FP name by ML algorithms was limited due to inherent noise. However, classification was more accurate with an ANN. This is most likely due to the ability of ANNs to distinguish predictive components from inherent biological noise during the learning process. By implication this also means that classification by ML algorithms might be improved by first extracting the principal components (PCs) and subsequently fitting the model.

Additionally, time course continuation prediction by ANN was successful. The ANN was able to predict the continuation of a time course with a high accuracy (Fig. 3.4). This implicates that all the needed information was contained in the initial fraction of the dataset. Therefore, the imaging time of future experiments could be shortened at least by half the time to facilitate faster result acquisition. There is no precedence of a similar study predicting time lapse FP data by an ANN. A previous study, however, predicted the excitation and emission wavelengths of eGFP-derived FPs accurately, based on structural information of their chromophore core [Nantasenamat et al., 2007]. This study used an ANN with less hidden layers (one) than in the present thesis. However, while testing the ANN during this study, it became obvious, that a classification or prediction of time courses was not possible with one hidden layer.

The established high-throughput assay is applicable to future investigations

The established high-throughput assay is applicable to future investigations, in particular for fluorescence-based approaches. It is already deployed in another project utilizing a fluorescent reporter and comparing its FI between mutant and wt fish. This combination is particularly useful if the number of mutants is limited. Therefore, imaging of up to 96 presumably mutant fish can be performed with subsequent genotyping [Hammouda et al., 2019]. This allows a high-throughput readout of affected pathways in mutants via fluorescent reporters.

α-Bungarotoxin is the best available anesthetic for long-term imaging in medaka

The standard anesthetic for teleosts, tricaine, has shown incomplete anesthesia and impairment of cardiac development during extended *in vivo* imaging [Culver and Dickinson, 2010]. Several groups were therefore aiming at improving this treatment [Dray et al., 2015, Readman et al., 2017, Barbosa et al., 2015]. The most effective anesthetic without induction of cardiac defects in medaka, however, is α-Bungarotoxin, as has been shown before in zebrafish [Swinburne et al., 2015] (Fig. 3.5).

Microinjection of α-Bungarotoxin mRNA does not only lead to effective anesthesia, it also supplies a photostable anesthetic. This is of particular importance during fluorescence microscopy of whole organs *in vivo* as conducted here. A more specific discussion on the effects and applicability of α-Bungarotoxin is supplied in my recent publication [Lischik et al., 2019].

In vivo imaging was enhanced by pigmentation mutants

Imaging of *spooky* pigment mutant embryos was possible in the injected generation

The pigment double knockout *spooky* is readily established in any existing transgenic or mutant background (Fig. 3.6 and [Lischik et al., 2019]). This is helpful in establishing transparent medaka, similar to the see-through medaka [Wakamatsu et al., 2001], but without the need for extensive breeding and screening. The mutant facilitated deep imaging as demonstrated in this thesis for multiple lines in particular for the retina. The injection was additionally enhanced by increasing the number of sgRNAs targeting the same locus, as has been implicated in a previous investigation [Wu et al., 2018] (Fig. 3.6B).

spookiest pigmentation mutants will eliminate the residual autofluorescent pigment

The *spookiest* mutant consists of KOs of *pnp4a* [Kimura et al., 2017], *oca2* [Fukamachi et al., 2004] and *slc2a15b* [Kimura et al., 2014]. These mutants lack the pigments of iridophores, melanophores/xanthopores and leucophores, respectively. The loss of the latter is a valuable addition to *spooky* due to their high FI.

In terms of pigmentation state this mutant is similar to the zebrafish *crystal* mutant [Antinucci and Hindges, 2016] and see-through medaka [Wakamatsu et al., 2001, Ohshima et al., 2013]. In contrast to the latter, it lacks complicated maintenance due to known mutated loci. The *spookiest* KO is also employable to any known medaka inbred line with available genomic data [Spivakov et al., 2014].

The established *spooky* pigmentation mutant together with microinjection of α-Bungarotoxin mRNA was employed in the following to perform *in vivo* microscopy of RSCs and RPCs.

A retinal stem cell and two modes of daughter cell behavior were observed *in vivo*

A presumable retinal stem cell was tracked

The established tool set allowed to track single cells in the retina over long time periods (Fig. 3.8). These tracks were subsequently analyzed. One of the cells was not dividing in a time frame of 60 h and stayed in the periphery of a patch of cells, which is due to the tight connection presumably the clone stemming from this cell (Fig. 3.9). These two properties render this cell a presumable RSC, showing that RSCs are feasible to track. However, the length of division time might complicate the analysis of RSCs via *in vivo* imaging data. In order to continue the analysis, more presumptive RSCs need to be tracked in order to investigate their *in vivo* behavior.

Two modes of daughter cell behavior were observed in the retina

Several cell divisions were observed within the tracked data. Investigation revealed two distinct behaviors of daughter cells subsequent to division: (1) daughter cells stayed in close proximity to each other (2) daughter cells strove away from each other (Fig. 3.10). Whether these movements are active or passive, imposed by local tissue architecture, remains to be elucidated. Strikingly, the original cell which was located closer to the SC domain at the beginning of tracking, gave rise to the daughter cells that strove away from each other. Again, more tracking data will be needed in order to investigate the causative parameters for the different daughter cell behavior modes.

Global movements were corrected subsequent to data collection

All acquired data showed global drift or movements in need of correction. I implemented two different algorithms to do so, but the results were not convincing. This is most likely due to the low number of anchoring tracked cells. Therefore, I propose that for future efforts either more cells or hallmarks, such as the total retina, need to be tracked and taken into account. This would not only allow the registration of tps to each other, but also enable the derivation of positions relative to the retina. This is especially helpful, since the general domains of RSC and RPC presence are known from previous studies [Centanin et al., 2014].

The present data offered a new resolution for tracking of stem cells

Post-embryonic RSCs and RPCs have been extensively studied post hoc in medaka [Centanin et al., 2011, Centanin et al., 2014]. The non-invasive *in vivo* imaging, tracking and subsequent analysis of RSCs and RPCs will lead to novel insights into the growth mode and behavior of the analyzed cells *in vivo*. In comparison with the previous studies the growth mode, cell cycle intervals and number of divisions could be directly deduced from the data without estimation and therefore averaging the distribution. In order to investigate these parameters and the stereotypy of cells the number of analyzed cells needs to be improved drastically. Subsequent to the addition

of cell tracks the stereotypy of tracks could be analyzed. This analysis could then be directly compared to a previous study that showed the variance in division angle and rate depending on the position of dividing cells in the embryonic retina in zebrafish [Wan et al., 2016]. The present work focuses solely on the post-embryonic retina, which is differentiated and harbors homeostatic RSCs. The investigation was guided temporally by choice of the recombination tp. Stage 30 was chosen, as it was shown that the retina is exhibiting a post-embryonic growth mode at this stage [Sinn and Wittbrodt, 2013].

Compared with previous studies this data offers a new level of detail for non-invasive, long-term investigation of SCs *in vivo* with a high spatiotemporal resolution. Previous studies were limited in non-invasiveness by labeling cells *in vitro* and transplanting them [Sabapathy et al., 2015] or surgically adding an imaging window [Ritsma et al., 2014]. *In vivo* investigation was often performed in a post hoc analysis [Tolar et al., 2005, Suh et al., 2007]. Long-term acquisition was often limited to a maximum of 12 h [Park et al., 2017, Rompolas et al., 2012, Tata et al., 2013]. Other studies sacrificed spatial resolution for being able to image *in vivo* [Kraitchman and Bulte, 2009]. The present work aimed at a systematic investigation of wild-typic and uninfluenced SCs at a high resolution in order to investigate SC behavior *in vivo*.

Previous studies of SC behavior have shown that SCs depend on the circadian rhythm of the organism. This has been shown for hematopoietic stem cell (HSC) mobilization [Lucas et al., 2008, Méndez-Ferrer et al., 2008], skin SCs [Bjarnason et al., 2001] and neural stem cells (NSCs) in mouse and zebrafish [Moore and Whitmore, 2014, Kochman et al., 2006]. The high temporal resolution of the present data set will allow the comparison of the behavior of RSCs between day and night. Previous studies have shown that the β-catenin dependent Wnt pathway links the circadian clock with the cell cycle in adult stem cells [Matsu-ura et al., 2018]. This link is mainly mediated through GSK3 [Hirota et al., 2008]. Therefore, combining the here discussed single cell analysis of RSCs and RPCs with the following analysis of the effect of Wnt stimulation through DN-GSK3 could lead to a more fundamental understanding of the regulation of the circadian clock in SCs.

The effect of Wnt stimulation was dependent on dosage and cell type

The preceding work and the resulting conclusions were limited by analyzing polyclone formation only in the RSDNGSK3_high transgenic line. Polyclones are patches of cells that are connected from its most lateral to its most medial point (relative to the retina: connected from its most peripheral to its most central point). A polyclone can consist of a single or multiple clones, which are directly adjacent to each other, not allowing to draw reproducible borders. Even though it was possible to deduce that polyclone formation was affected by Wnt stimulation in RSDNGSK3_high retinae, it was not clear whether proliferative capacity or differentiation potential were affected. This was masked by the polyclone loss which was the predominant effect of recombination (Fig. 1.4A′, Fig. 3.13). To assess these two inherent properties the recombined cells need to form polyclones. Therefore, both DN-GSK3 lines (RSDNGSK3_high and RSDNGSK3_low, Fig. 3.11) were leveraged to analyze SCs in a β-catenin dosage-dependent manner reflecting different Wnt activity states in entire clones.

Combining both lines it was shown that high Wnt stimulation led to apoptosis of most cells in the retina. In lRPCs, however, it changed the differentiation potential. High Wnt stimulation also immortalized fate-restricted RPCs. Low Wnt stimulation decreased the proliferative capacity of RSCs and eRPCs (summarized in Fig. 4.1).

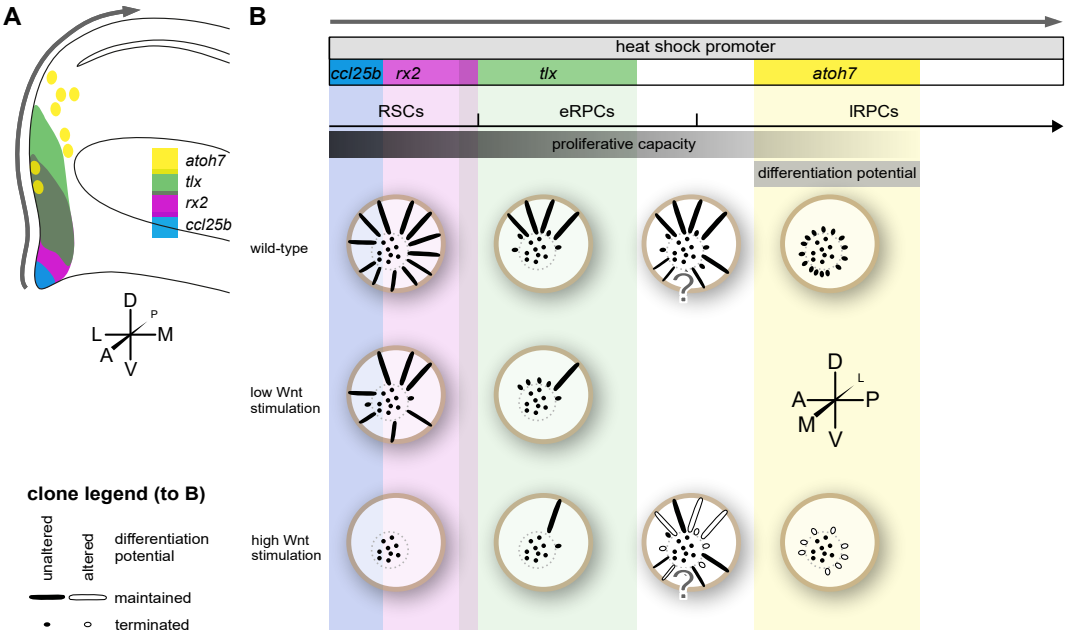

Figure 4.1: Wnt stimulation resulted in apoptosis, a decrease in proliferative capacity in RSCs and eRPCs and alteration of differentiation potential in lRPCs. **A** Expression domains of the promoters used for *CreERT2* expression. The gray arrow indicates the progression of cells, linearized in B. **B** The top scheme indicates the expression domains together with the matching cell types. The gray arrow indicates the same progression of cells as in A. Summarized are the clone properties of GaudíRSG, RSDNGSK3_low and RSDNGSK3_high retinae. Recombination in RSCs led to maintained clones in GaudíRSG retinae. In contrast, fewer clones were present in RSDNGSK3_low retinae. The residual clones were also terminating with a higher probability in comparison to wt. This indicated a decrease in proliferative capacity. No clones were present in RSDNGSK3_high retinae due to increased apoptosis upon Wnt stimulation. Recombination in eRPCs led to a mixture of maintained and terminating clones in GaudíRSG retinae. In contrast, fewer clones were present in RSDNGSK3_low retinae. The residual clones were again terminating with a higher probability in comparison to wt, again indicating a decrease in proliferative capacity. Very few clones were present in RSDNGSK3_high retinae due to increased apoptosis upon Wnt stimulation. Recombination of all cell types by *hsp70:Cre* showed a mixture of all clone types in GaudíRSG retinae. In RSDNGSK3_high retinae maintained clones were observed. This is in contrast to all experiments utilizing an available *CreERT2* line, indicating that the maintained clones were most likely formed by cells not included in the promoters of the *CreERT2* lines. Together with the fate restriction of maintained clones, the clone-founding cells were most likely located between or in eRPCs and/or lRPCs. Most likely, these cells were immortalized by Wnt stimulation, but their fate restriction was fixed. Recombination in lRPCs in GaudíRSG retinae led to formation of terminated clones consisting of few cells. Wnt stimulation in lRPCs, however, led to fewer clones due to apoptosis and a change of differentiation potential. Anatomical rosettes indicate the orientation of the schemes. A: anterior, P: posterior, D: dorsal, V: ventral, M: medial (central in respect to the retina), L: lateral (peripheral in respect to the retina). Illustrations in panel B are modified from [Centanin et al., 2014], with permission.

High Wnt stimulation mainly led to apoptosis

As pointed out previously, the preceding work was limited by the loss of polyclones, the reason for which remained to be elucidated. The shown combination of *in vivo* imaging of recombined RSDNGSK3_high retinae and the TUNEL assay revealed that most likely a large fraction of recombined cells underwent apoptosis (Fig. 3.14). The acquisition of these results was again supported by the established toolset.

This demarcates this work from others, which have shown that Wnt is inducing an alteration of division symmetry [Habib et al., 2013], delay of division, induction of quiescence [Chavali et al., 2018, Fleming et al., 2008] or apoptosis through inhibition of β-catenin dependent Wnt signaling [Chen et al., 2001]. The discrepancy of Wnt stimulation and Wnt inhibition both leading to apoptosis might be explained by autoinhibition of the β-catenin dependent Wnt pathway due to high stimulation [Jho et al., 2002].

The apoptosis of recombined cells is, however, contradicting previous studies which show that Wnt stimulation typically leads to proliferation and survival of cells [Reya and Clevers, 2005, Crowder and Freeman, 2000] whereas Wnt inhibition leads to apoptosis [Grotewold and Rüther, 2002, Ellies et al., 2000]. A possible reason for this discrepancy is that the previous studies were either performed *in vitro* or via drug treatment, while this work focuses on Wnt stimulation of single cells in their organismal context. Another possible reason is the presumable pleiotropic effect of DN-GSK3 on other pathways, which I will elaborate on later.

Residual positive cells were observed in RSDNGSK3_high retinae

Even though it has been shown that most cells underwent apoptosis in response to high Wnt stimulation, residual positive cells were observed (Fig. 3.14C). These residual cells are most likely the founding cells for the maintained fate-restricted clones observed in the preceding work subsequent to recombination of RSDNGSK3_high with *hsp70:Cre* (Fig. 1.3A′).

The position of the residual cells was overlapping with RPCs. This is fitting the observation that the resulting maintained clones were fate-restricted in 89 % of the cases. This implicated immortalization of these residual cells and therefore formation of exogenous clones. This is in line with a previous study showing that Wnt stimulation increases long-term maintenance of zebrafish RPCs [Meyers et al., 2012].

Low Wnt stimulation decreased proliferative capacity of retinal stem and progenitor cells

The results indicate that low Wnt stimulation decreased the proliferative capacity of RSCs and eRPCs while not altering their differentiation potential (Fig. 3.15). The decrease in proliferative capacity was shown by an increase in terminating clones upon Wnt stimulation (Fig. 3.15I). Additionally, the clones resulting from Wnt stimulated cells were narrower than wt clones (Fig. 3.15J-J′).

This effect might be mediated directly through GSK3 acting on the cell cycle [Hirota et al., 2008]. Even though most previous studies show a positive effect of Wnt stimulation on proliferative capacity [Reya and Clevers, 2005, Crowder and Freeman, 2000], there are also supporting studies which show a negative effect of Wnt stimulation which will be discussed in the following. These studies have shown *in vitro* that Wnt3a and Wnt5a increase the differentiation of extracted cells from the telencephalon, the ventral midbrain and the striatum into neurons [Muroyama et al., 2004, Schulte et al., 2005, Kasai et al., 2005]. Moreover, it has been shown *in vitro* and *in vivo* that Wnt7a promotes differentiation of cortex neural precursors [Hirabayashi et al., 2004] and that the differentiation of progenitors in the subventricular zone (SVZ) is promoted by Wnts *in vivo* [Munji et al., 2011].

In order to elucidate how these differential effects of Wnt stimulation came about a previous study identified FGF2 as the main switch for the effect of Wnt stimulation [Israsena et al., 2004]. This previous study was able to show that presence or absence of FGF2 determines whether Wnt stimulation leads to SC maintenance or differentiation, respectively. It is conceivable, that such a factor is also present in the medaka NR.

High Wnt stimulation leads to apoptosis or immortalization, low Wnt stimulation leads to a decrease of proliferative capacity

In conclusion, the effect of Wnt stimulation is dependent on the level of stimulation and the initially recombined cell type.

Along the axis of dosage effect it appears that cells are more prone to undergo apoptosis when experiencing a high level of Wnt stimulation. This was the case for all investigated cell types: *ccl25b*-positive RSCs (Fig. 3.15), *rx2*-positive RSCs (Fig. 1.4A-A′ and [Möller, 2017]), *tlx*-positive RSCs and eRPCs (Fig. 1.4C-C′., Fig. 3.15 and [Möller, 2017]) and *atoh7*-positive lRPCs (Fig. 1.4E-E′ and [Möller, 2017]). Furthermore, a change of differentiation potential was observed in the remaining polyclones of *atoh7*-positive cells (Fig. 1.4F-F′ and [Möller, 2017]). The loss of RGCs matches a previous study, which showed that Wnt stimulation in the retina leads to a loss of ipsilaterally projecting RGCs [Iwai-Takekoshi et al., 2018].

Along the axis of the initially recombined cell type it was concluded that there is a similar effect of low Wnt stimulation on RSCs and eRPCs (Fig. 3.15). The main effect in these cell types is a decrease of proliferative capacity, indicated by the more frequently terminating and narrower polyclones. However, there does not seem to be an impact on differentiation potential, since in all analyzed retinae the full set of cell types was present in all polyclones. This is in contrast to previous studies, which indicated that the effect of Wnt stimulation differs depending on the cell type [Kubo et al., 2005, Kubo and Nakagawa, 2009, Kubo, 2003, Denayer et al., 2008].

Combining both axes, it was concluded, that high Wnt stimulation is inducing apoptosis in varying degrees descending from RSCs via eRPCs to lRPCs. It is also inducing a change in differentiation potential in the surviving lRPCs. Presumably, high Wnt stimulation is also able to immortalize a subset of RPCs, which give rise to fate-restricted lineages. Low Wnt stimulation however, decreases proliferative capacity of RSCs and eRPCs. All experimental interpretations are summarized in Fig. 4.1.

Dominant-negative GSK3 has multiple targets

An important point to mention is the presumed pleiotropic effect of DN-GSK3. However, DN-GSK3 has been widely used in prior studies to investigate Wnt signaling [Taelman et al., 2010, Pachenari et al., 2017, Abdul et al., 2018, Pierce and Kimelman, 1995, Yost et al., 1996] and is also targeted by chemical compounds in many clinical trials. In contrast, there are also studies that link GSK3 to various pathways such as the TGF-β pathway [Yang et al., 2018,Beurel et al., 2015], the PI3K/Akt/mTOR pathway [Vallée and Vallée, 2018] and the insulin pathway [Chami et al., 2016]. Therefore, it cannot be excluded that additional pathways are involved and/or affected by overexpression of DN-GSK3.

Δ90-β-catenin is an alternative agent to dominant-negative GSK3

A solution for this presumed pleiotropic effect of DN-GSK3 is the usage of Δ90-β-catenin, a stabilized form of β-catenin. This protein is not binding to the cytoskeleton or the destruction complex but remains transcriptionally active targeting its original target genes [DasGupta et al., 2002]. This tool, together with the presented results, might help address the question of how specific DN-GSK3 is to β-catenin dependent Wnt signaling. Experiments will be conducted by establishing another GaudíRSG-based construct, substituting H2B-eGFP with Δ90-β-catenin.

5
Conclusions

Within this work I have approached three major goals with the aim to unravel the regulation of RSCs and RPCs by β-catenin dependent Wnt signaling.

First, *in vivo* imaging of medaka was enhanced by improving fluorescent protein selection, anesthesia and pigmentation. The established assay is also already employed for different fluorescence-based investigations on wt and mutant embryos. Furthermore, with the advent of inbred lines the pigment knockout is now universally adaptable to any line, if investigation via microscopy is necessary.

Second, the established toolset offered the unique opportunity to perform *in vivo* microscopy of neural stem cells. In contrast to previous approaches it a high spatiotemporal resolution (xyzt: 0.26 µm, 0.26 µm, 1 µm, 20 min), long imaging times (up to 4 d), and non-invasive imaging (no surgery needed) were achieved.

Finally, this work elucidated the effect of Wnt stimulation on single retinal stem cells. High Wnt stimulation was shown to induce apoptosis in RSCs and eRPCs, with a few residual cells remaining. These residual cells were immortalized while keeping their differentiation potential. High Wnt stimulation in lRPCs decreased proliferative capacity and altered their differentiation potential. Low Wnt stimulation in RSCs and eRPCs decreased the proliferative capacity of the cells while keeping their differentiation potential.

In the near future the continued *in vivo* investigation of retinal stem cells will lead to novel insights into their wild-typic behavior. Furthermore, the combination of the shown and established tools will enable *in vivo* investigation of retinal stem cells exposed to Wnt stimulation.

6
Materials & Methods

Materials

Fish lines

The *Oryzias latipes* inbred and isogenic iCab line was used as the wt and reference strain for the experiments [Wittbrodt et al., 2002]. All fish lines were generated from iCab and used lines are indicated in Table 6.1.

Table 6.1: Fish lines used in this thesis

Fish line name	Internal stock numbers	Source
CR(Oca2 sgRNA57, sgRNA58)	7487, 7733, 7883, 8230	this thesis
CR(Oca2 sgRNA57, sgRNA58, Pnp4a sgRNA250)	7732, 7886, 8038, 8229, 8232	this thesis
GaudíRSG, CR(Oca2, Pnp4a, CFP)	7975	crossing and this thesis
HsCre, GaudíRSG, CR(Oca2, Pnp4a, CFP)	7470, 7730, 7731, 7974, 8379, 8391, 8392, 8564	crossing and this thesis
HsCre, RSDNGSK3_high, CR(Oca2 sgRNA57, sgRNA58, Pnp4a sgRNA250)	7457, 7734, 8231, 8474	crossing and this thesis
ccl:*CreERT2*, GaudíRSG	8042	crossing
ccl:*CreERT2*, RSDNGSK3_high	8043, 8189	crossing
ccl:*CreERT2*, RSDNGSK3_low	8044	crossing
tlx:*CreERT2*, GaudíRSG	8045	crossing

Fish line name	Internal stock numbers	Source
tlx:*CreERT2*, RSDNGSK3_high	7976, 8046, 8190	crossing
tlx:*CreERT2*, RSDNGSK3_low	8047	crossing
RSDNGSK3_high	8186	crossing
RSDNGSK3_low	8258	crossing

Plasmids

The plasmids used in this thesis can be found in Table 6.2.

Table 6.2: Plasmids used in this thesis.

Internal plasmid stock number	Name (additional description)	Source
3632	DR274 sgRNA backbone (T7)	lab stock
5357	DR274(sgRNA 57 Oca2_ex9_T1)	this work
5358	DR274(sgRNA 58 Oca2_ex9_T3)	this work
5359	DR274(sgRNA 250 Pnp4a_T2)	lab stock
5433	DR274(sgRNA 251 Pnp4a_T41)	lab stock
5000	DR274(sgRNA 252 CFP_notGFP)	this work
5197	Cas9	lab stock
5432	DR274(sgRNA 280 Oca2_intron_T1)	lab stock
5419	DR274(sgRNA 295 Pnp4a_T2_3)	this work
5420	DR274(sgRNA 296 Pax7a_T3)	this work
5421	DR274(sgRNA 298 Pax7a_T21)	this work
5422	DR274(sgRNA 299 Pax7a_T34)	this work
5423	DR274(sgRNA 300 slc2a15b_T2)	this work
5426	DR274(sgRNA 303 slc2a15b_T8)	this work
5427	DR274(sgRNA 304 tyr_T1)	this work
5428	DR274(sgRNA 305 tyr_T3)	this work
5180	pGGEV3_+(CloverwCR13wKpnI)+_+1_PCR	this work

Internal plasmid stock number	Name (additional description)	Source
5181	pGGEV3_+(eGFP)+_+1_PCR	this work
5182	pGGEV3_+(eGFPwCR13_A206K)+_+1_PCR	this work
5183	pGGEV3_+(mCherry)+_+1_PCR	this work
5184	pGGEV3_+(mRFP)+_+1_PCR	this work
5185	pGGEV3_+(mRuby2)+_+1_PCR	this work
5186	pGGEV3_+(tagRFP)+_+1_PCR	this work
5187	pGGEV3_+(Venus)+_+1_PCR	this work
5188	pGGEV3_+(YFP)+_+1_PCR	this work
5202	pGGEV3_+(CFPwCR13)+_+1_PCR	this work
5340	pGGEV3_+(mGFPmut2)+_+1_PCR	this work
5341	pGGEV3_+(mRFP1asterisk)+_+1_PCR	this work
5342	pGGEV3_+(mScarlet-I)+_+1_PCR	this work
5343	pGGEV3_+(mVenusNB)+_+1_PCR	this work
5350	pGGEV3_+(H2A-mCherry)+_+1_PCR	this work
5351	pGGEV3_+(lifeact-eGFP)+_+1_PCR	this work
5352	pGGEV3_+(OleGFP)+_+1_PCR	this work
5353	pGGEV3_+(SceGFP)+_+1_PCR	this work
5173	pmtb-T7-alpha-bungarotoxin	this work

Primers

All primers were ordered from Eurofins MWG Operon, Table 6.3 contains all primers in 5' to 3' orientation, which were already present in the lab and could be used as present resources, whereas Table 6.4 contains the primers in 5' to 3' orientation which were designed and ordered while conducting this work.

Table 6.3: Present primers used in this thesis.

Number	Alias	Sequence (5' to 3' orientation)
JW1452	ACTB_seq_R	CAGGGGCAATTCTCAGCTCA

Number	Alias	Sequence (5' to 3' orientation)
JW2035	newGFPBamHI	GCCGGATCCATGGTGAGCAAGGGCGA
JW2915	ActB_SeqF2	CCTTGAAACGAAAAGCCCCC
JW3188	newTagRFP_F_BamHI	GCCGGATCCATGGTGTC TAAGGGCGAAGAG
JW3189	newTagRFP_R_KpnI	GCCGGTACCTTAATTAAGT TTGTGCCCCAGTTTGC
JW3566	mRFP_STOP_KpnI_R	GCCGGTACCTTAGGCGCC GGTGGAGTGGCGGCC
JW3667	3377 rev	TGTAGATGAACTCGCCGTCC
JW5653	mCherry seq_downstream	CTCAGTTCATGTACGGCTCCAAG
JW6070	mRuby2_KpnI_R	GCCGGTACCTTACTTGTACAGCTCGTCCA
JW6237	mRFP fwd_BamHI	GCCGGATCCATGGCCTCCTCCGAGGACG

Table 6.4: Primers designed and used in this thesis.

Number	Alias	Sequence (5' to 3' orientation)
JW6523	sgRNA252_F	TAGgTATAGACGTTGTCGCTGA
JW6524	sgRNA252_R	AAACTCAGCGACAACGTCTATA
JW7506	ctnnb2_cDNA_R_KpnI	GCCGGTACCTTACAGGTCGGT ATCAAACC
JW7510	ctnnb2_delta90_cDNA_ATG	GCCAGATCTATGCGTGCTCAGAG GGTGCGTGCAGCCATG
JW7654	eGFPmutV68L_S72A	CTGCAGTGCTTCGCCCGCTACC CCGACCACATGAA
JW7655	eGFPmutF64F_S65A	GCCGTAGGCGAAGGTGGTCACG AGGGTGGGCC
JW7658	mVenus to mVenus NB_F	GTGCAGTGCTTCGCCCGCTAC
JW7659	mVenus to mVenus NB_R	GCCGTAGCCCAGGGTGGTCA
JW7660	mRFP to mRF-Paster_F_BamHI	GCCGGATCCATGAGTAAAGGAG AAGAAAACAACTTAGCTGTCA TCAAGGAGTTCATGCGC

Number	Alias	Sequence (5' to 3' orientation)
JW7661	mRFP to mRF-Paster_R_KpnI	GCCGGTACCTTATTTGTATAGT TCATCCATGCCGCCGGTGGAGT GGCGGCCCTC
JW8200	OlGFP_F_ATG_BsaI_EV3	GCCGGTCTCAACCTCTATGGTG AGCAAGGGAGAGGA
JW8201	OlGFP_R_TGA_BsaI_EV3	GCCGGTCTCATAGTTCACTTGT ACAGCTCGTCCATTC
JW8206	eGFP_F_BamHI_lifeact	GCCGGATCCATGGGCGTGGCC GACCTGATCAAGAAGTTCGAGA GCATCAGCAAGGAAGAGGGCGA CCCACCGGTCGCCACCATGGTG AGCAAGGGCGAGGA
JW8207	H2A_F_ATG	GCCGGATCCATGGCAGGTGGAA AAGCAGG
JW8208	ScGFP_F_BamHI_ATG	GCCGGATCCATGGTTAGTAAAG GAGAAGAACTTTT
JW8209	ScGFP_R_KpnI_STOP	GCCGGTACCTTATTTGTATAGT TCATCCATGC
JW8311	delta90_ctnnb2_cDNA_T2A	GCCTCTAGAGAGGGCAGAGGAA GTCTTCTAACATGCGGTGACGTG GAGGAGAATCCCGGCCCTATGCGT GCTCAGAGGGTGCG
JW8579	pnp4a_T2_3_F	TAGGAGGGCGTCTACGCCATGG
JW8580	pnp4a_T2_3_R	AAACCCATGGCGTAGACGCCCT
JW8581	pax7a_T3_F	TAGGTAATTCTGGCCTGGCGCA
JW8582	pax7a_T3_R	AAACTGCGCCAGGCCAGAATTA
JW8585	pax7a_T21_F	TAggGGGCTCGGTGGCGTAAGC
JW8586	pax7a_T21_R	AAACGCTTACGCCACCGAGCCC
JW8587	pax7a_T34_F	TAGgGGAGTGTTCATCAACGGG
JW8588	pax7a_T34_R	AAACCCCGTTGATGAACACTCC
JW8589	slc2a15b_T2_F	TAggCTCCGGTCATCCCGCCGA
JW8590	slc2a15b_T2_R	AAACTCGGCGGGATGACCGGAG
JW8595	slc2a15b_T8_F	TAggGGTAACAATAAGGACCCG
JW8596	slc2a15b_T8_R	AAACCGGGTCCTTATTGTTACC
JW8597	tyr_T1_F	TAggTCCAGACAAATAGGTCGT

Number	Alias	Sequence (5' to 3' orientation)
JW8598	tyr_T1_R	AAACACGACCTATTTGTCTGGA
JW8599	tyr_T3_F	TAGgACGTGGGTAGATGGACCG
JW8600	tyr_T3_R	AAACCGGTCCATCTACCCACGT

RNAs

The following tables list all RNAs used during this work. Table 6.5 lists all sgRNAs, whereas Table 6.6 lists all mRNAs used in this thesis. Transcription of sgRNAs was performed according to the protocol in the subsection sgRNA transcription, whereas transcription of mRNAs was performed according to the protocol in the subsection mRNA transcription. Fluorescent protein CDSs were cloned from present plasmids except for the following. cytoplasmic EKAR (Cerulean-Venus) was a gift from Karel Svoboda (Addgene plasmid # 18679) [Harvey et al., 2008], mRuby2-C1 was a gift from Michael Davidson (Addgene plasmid # 54768) [Lam et al., 2012], pcDNA3-Clover was a gift from Michael Lin (Addgene plasmid # 40259) [Lam et al., 2012], pmScarlet-i_C1 was a gift from Dorus Gadella (Addgene plasmid # 85044) [Bindels et al., 2016] and SceGFP was a gift from Sabine Strahl [Xu et al., 2013]. The exact amino acid sequences and comparison with the publication of Balleza and colleagues are listed in Table 7.1.

Table 6.5: sgRNAs used in this thesis.

Name (additional description)	Source
sgRNA 57 Oca2_ex9_T1	this work
sgRNA 58 Oca2_ex9_T3	this work
sgRNA 250 Pnp4a_T2	lab stock
sgRNA 251 Pnp4a_T41	lab stock
sgRNA 252 CFP_notGFP	this work
sgRNA 280 Oca2_intron_T1	lab stock
sgRNA 295 Pnp4a_T2_3	this work
sgRNA 296 Pax7a_T3	this work
sgRNA 298 Pax7a_T21	this work
sgRNA 299 Pax7a_T34	this work

Name (additional description)	Source
sgRNA 300 slc2a15b_T2	this work
sgRNA 303 slc2a15b_T8	this work
sgRNA 304 tyr_T1	this work
sgRNA 305 tyr_T3	this work

Table 6.6: mRNAs used in this thesis.

Name (additional description)	Source
CloverwCR13wKpnI	this work
eGFP	this work
eGFPwCR13_A206K	this work
mCherry	this work
mRFP	this work
mRuby2	this work
tagRFP	this work
Venus	this work
YFP	this work
CFPwCR13	this work
mGFPmut2	this work
mRFP1asterisk	this work
mScarlet-I	this work
mVenusNB	this work
H2A-mCherry	this work
lifeact-eGFP	this work
OleGFP	this work
SceGFP	this work
α*-bungarotoxin*	this work

Antibodies

Antibodies used are listed in Table 6.7.

Table 6.7: Antibodies used in this thesis.

Target	Host	Conjunction	Used Dilution	Supplier	Cat. no.
Rx2	rabbit	none	1:250	lab made	NA
eGFP	chicken	none	1:200	Life technologies	A10262
chicken	donkey	DyLight488	1:250	Jackson	703-485-155
rabbit	goat	AlexaFluor647	1:125	Life Technologies	A-21245

Antibiotics

Antibiotics used for bacterial selection are listed in Table 6.8.

Table 6.8: Antibiotics used in this thesis.

Antibiotic	Stock conc.	Working conc.	Supplier
Ampicillin	30 $^{mg}/_{ml}$	50 $^{\mu g}/_{ml}$	Roth
Kanamycin	50 $^{mg}/_{ml}$	100 $^{\mu g}/_{ml}$	Roth

Kits

Kits used in this thesis are listed in Table 6.9

Table 6.9: Kits used in this thesis.

Name	Supplier
innuPREP DOUBLEpure Kit	Analytik Jena
MEGAShortScript T7 Kit	Ambion
MinElute Gel Extraction Kit	QIAGEN
mMessage mMachine® Sp6 Transcription Kit	Invitrogen

Name	Supplier
QIAPrep® Spin Miniprep Kit	QIAGEN
QIAquick® Gel Extraction Kit	QIAGEN
QIAquick® Nucleotide Removal Kit	QIAGEN
QIAquick® PCR Purification Kit	QIAGEN
QIAGEN Plasmid Midi Kit	QIAGEN
RNeasy Mini Kit	QIAGEN
RevertAid First Stand cDNA Synthesis Kit	Thermo Fisher Scientifc

Enzymes and corresponding buffers

Enzymes used in this thesis are listed in Table 6.10, whereas the corresponding buffers are listed in Table 6.11.

Table 6.10: Enzymes used in this thesis.

Type	Name	conc.	Supplier
Restriction Enzyme	BamHI-HF	20 U/µl	NEB
Restriction Enzyme	BsaI-HF	20 U/µl	NEB
Restriction Enzyme	KpnI-HF	20 U/µl	NEB
Restriction Enzyme	DpnI	20 U/µl	NEB
Restriction Enzyme	SpeI-HF	20 U/µl	NEB
Restriction Enzyme	EcoRV-FD	-	Thermo Fisher Scientifc
Restriction Enzyme	BglII-FD	-	Thermo Fisher Scientifc
Restriction Enzyme	DraI-FD	-	Thermo Fisher Scientifc
Restriction Enzyme	Eco31I-FD	-	Thermo Fisher Scientifc
DNA Ligase	T4 DNA Ligase	5 U/µl	Thermo Fisher Scientifc

Type	Name	conc.	Supplier
DNA Ligase	T4 DNA Ligase	30 U/µl	Thermo Fisher Scientifc
DNA Polymerase	Q5 High-Fidelity DNA Polymerase	2 U/µl	NEB
Kinase	T4 Polynucleotide Kinase	10 U/µl	NEB
Proteinase	Proteinase K	10 mg/ml	Roche
DNase	TurboDNase I	2 U/µl	Life Technologies
hatching enzyme	hatching enzyme	lab made	

Table 6.11: Enzyme buffers used in this thesis.

Buffer	Concentration	Supplier
CutSmart	10 x	NEB
FastDigest	10 x	Thermo Fisher Scientifc
FastDigest Green	10 x	Thermo Fisher Scientifc
T4 DNA Ligase Buffer	10 x	Thermo Fisher Scientifc
Q5 Reaction Buffer	5 x	NEB

Chemicals and reagents

The used chemicals and reagents are listed in Table 6.12.

Table 6.12: Chemicals and reagents used in this thesis.

Chemical/Reagent	**Abbreviation/ Synonym**	**Supplier**
2-Propanol	Isopropanol	Sigma-Aldrich
4-(2-hydroxyethyl)-1-piperazineethanesulfonic acid	HEPES	Roth
4',6-Diamidin-2-phenylindol	DAPI	Sigma-Aldrich
Acetone		Sigma-Aldrich
Adenosine triphosphate	ATP	Thermo Fisher Scientifc
Agar		Roth
Agarose		Sigma-Aldrich
Agarose Low Melt		Roth
Bacto-Tryptone		Gibco
Bovine Serum Albumin	BSA	Sigma-Aldrich
Calcium chloride dihydrate	$CaCl_2 \cdot 2\,H_2O$	AppliChem
Chloroform		Sigma-Aldrich
Deoxyadenosine triphosphate	dATP	Thermo Fisher Scientifc
Deoxynucleotide triphosphates	dNTPs	Sigma-Aldrich
Dimethyl sulfoxide	DMSO	Roth
DNA loading dye 10 x		NEB
Ethylenediaminetetraacetic acid	EDTA	Roth
Ethanol 70 % (denatured)	EtOH	Roth
Ethanol 96 % (denatured)	EtOH	Roth
Ethanol 99 %	EtOH	Sigma-Aldrich
Ethidium Bromide	EtBr	Sigma-Aldrich
Etomidate		Sigma-Aldrich
GeneRuler™DNA Ladder Mix		Thermo Fisher Scientifc
Glacial acetic acid		Merck
Glucose		Sigma-Aldrich
Glycerin	Glycerol	Merck

Chemical/Reagent	Abbreviation/ Synonym	Supplier
Hydrogen Chloride	HCl	Merck
Hydrogen peroxide	H_2O_2	Sigma-Aldrich
Mach1™T1^R phage resistant chemical competent *E. coli*		Life Technologies
Magnesium Sulphate Heptahydrate	$MgSO_4 \cdot 7\,H_2O$	Merck
Methylene blue trihydrate		Sigma-Aldrich
N-Phenylthiourea	PTU	Sigma-Aldrich
Normal Goat Serum	NGS	Gibco
Orange G		Sigma-Aldrich
Paraformaldehyde	PFA	Sigma-Aldrich
Phenol/Chloroform/Isoamylalcohol	PCI	Roth
Polyethylene glycol - 4000	PEG-4000	Thermo Fisher Scientifc
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
Red sea salt		Red Sea
RNA Loading Dye 2x rapid		Thermo Fisher Scientific
RNase-free water		Sigma-Aldrich
Roti®		Roth
Sheep Serum		Sigma-Aldrich
Sodium acetate	NaAc	Grüssing
Sodium chloride	NaCl	Sigma-Aldrich
Sodium citrate		Sigma-Aldrich
Sodium dodecyl sulphate sodium salt	SDS	Serva
Sodium hydrogen phosphate	Na_2HPO_4	Applichem
Sodium hydroxide	NaOH	Sigma-Aldrich
Sucrose		Roth

Chemical/Reagent	Abbreviation/ Synonym	Supplier
Trans-Tamoxifen	Tamoxifen	Sigma-Aldrich
Tricaine	MS-222	Sigma-Aldrich
Tris base		Roth
Tris-hydrochloride	Tris-HCl	Sigma-Aldrich
Trizol		invitrogen
Tween 20		Sigma-Aldrich
X-Gal		Thermo Fisher Scientifc
Yeast Extract		Roth

Consumables

The used consumables are listed in Table 6.13.

Table 6.13: Consumables used in this thesis.

Consumable	Supplier
BLAUBRAND® intraMARK	BRAND
Cell saver tips 200µl	Roth
D1000 ScreenTape	Agilent Technologies
FEP tubes ⌀ 1 mm	Karl Schupp AG
Filter paper	Whatman
Filter Tips 10µl, 20µl, 200µl, 1.25 ml	Starlab
Filter Tips TipOne® RPT (sterile) 10µl, 20µl, 200µl	Starlab
Folded Filters	Sartorius
Gas permeable moisture barrier seal (4ti-0516/96)	4titude
Glass beads	Roth
Glass Petri dishes STERIPLAN® 9cm	Roth
Glass vials	Roth
Injection needles GC100F-10	Harvard Apparatus

Consumable	Supplier
Latex Gloves	Semperguard
Microloader tips	Eppendorf
Micro pestles 0.5/1.5 ml	Laborversand Hartenstein
Micro pestles 1.5/2.0 ml	Eppendorf
Nitrile Gloves	Starlab
Pasteur pipettes	Sarstedt
Petri dishes	greiner
Pipette tips	Steinbrenner
Reaction tubes 1.5 ml, 2 ml	Sarstedt
Sandpaper 1000 grit	Bauhaus
Tubes 15 ml, 50 ml	Sarstedt
Well plates (6 well)	böttger
Well plates (96 well)	Greiner bio-one
Whatman® Cellulose Filter Paper	Whatman

Media and buffers

Table 6.14 lists all used media and buffers including instructions according to the standard protocols.

Table 6.14: Buffers and solutions prepared for this thesis. If not stated otherwise, reagents were dissolved in H_2O.

Name	Ingredient	Concentration
LB-Medium	Bacto-Tryptone	10 g/l
	Yeast Extract	5 g/l
	Sodium Chloride	10 g/l
LB-Plates	Bacto-Tryptone	10 g/l
	Yeast Extract	5 g/l
	Sodium Chloride	10 g/l

Name	Ingredient	Concentration
	Agar	15 g/l
TB-Medium	Bacto-Tryptone	12 g/l
	Yeast Extract	24 g/l
	Glycerin	0.4 % v/v
	KH_2PO_4	2.13 g/l
	K_2HPO_4	12.54 g/l
P1	Glucose	50 mmol/l
	Tris-HCl	25 mmol/l
	EDTA	10 mmol/l
	pH 8, stored at 4 °C	
P2	NaOH	0.2 mol/l
	SDS	1 % w/v
P3	KAc	5 mol/l
	stored at 4 °C	
TAE	Tris base	242 g/l
	Glacial acetic acid	5.71 % v/v
	EDTA	50 mmol/l
	pH 8.5	
1 x PBS	NaCl	137 mmol/l
	KCl	2.7 mmol/l
	KH_2PO_4	240 mg/l
	Na_2HPO_4	1.44 g/l
1 x PTW	NaCl	137 mmol/l
	KCl	2.7 mmol/l
	KH_2PO_4	240 mg/l
	Na_2HPO_4	1.44 g/l
	Tween 20	0.1 % v/v
1 x ERM	NaCl	17 mmol/l
	KCl	0.4 mmol/l
	$CaCl_2 \cdot 2\,H_2O$	0.27 mmol/l
	$MgSO_4 \cdot 7\,H_2O$	0.66 mmol/l

Name	Ingredient	Concentration
	HEPES pH 7.3	17 mmol/l
	pH 7	
Medaka hatching solution	Methylene blue	2 mg/l
	in 1 x ERM	
1 x Zebrafish medium	red sea salt	300 mg/l
Finclip buffer	TrisHCl pH 8	400 mmol/l
	EDTA pH 8	5 mmol/l
	NaCl	150 mmol/l
	Tween 20	0.1 % v/v
	Proteinase K	1 mg/ml
Oligo annealing buffer	Tris	10 mmol/l
	NaCl	30 mmol/l
10 x Orange G loading dye	Orange G	2 mg/ml
	Glycerol	33 % v/v
20 x Tricaine	Tricaine	4 g/l
	$Na_2HPO_4 \cdot 2\,H_2O$	10 g/l
	in Millipore water	
50 mmol/l Tamoxifen	Trans-tamoxifen	18.5 mg/ml
	in DMSO	
PBDT	BSA	1 mg/ml
	DMSO	1 % v/v
	in final 1 x PTW	
Blocking buffer	Sheep serum	4 % v/v
	in PBDT	

Equipment and Instruments

Table 6.15 lists all equipment, which was used during this thesis.

Table 6.15: Equipment used in this thesis.

Equipment	Supplier
1.5ml tube centrifuge	neoLab
Acquifer Imaging Machine	DITABIS, Pforzheim Germany
Bacterial Shaker INNOVA 44	New Brunswick scientific
Borosilicate glass capillaries GC100F-10	Harvard apparatus
CAT S20	neoLab
Centrifuge 5417 C	Eppendorf
Centrifuge 5425	Eppendorf
Centrifuge 5430	Eppendorf
Centrifuge 5430 R	Eppendorf
Centrifuge 5810 R	Eppendorf
Diamond Glass Writer	VWR (Bruchsal)
DNAEngine Dyad®	Bio RAD
DNAEngine®	Bio RAD
FemtoJet express	Eppendorf
Fish incubator	Heraeus instruments
Fish incubator	RuMed
Forceps 5, 55 Inox stainless steel	Dumont
Freezer -20 ℃	Bosch
Freezer -80 ℃	Thermo Fisher Scientific
Fridge 4 ℃	Liebherr
Gel chamber	peqLab and lab made
Glass Bottom Dishes	MatTek
Glass Capillary Breaking Tool	lab made
Incubator B 28	Binder
Incubator BD 115	Binder
InjectMan NI2	Eppendorf
KL 1500 electronic	Schott
Leica DFC 500	Leica
Leica TCS SP8	Leica
Leica TCS SPE	Leica

Equipment	Supplier
Microinjector 5242	Eppendorf
Microwave R-939	Sharp
Multipette plus	Eppendorf
Needle puller P-30	Sutter Instrument Co USA
Gel iX20	Intas
MuVi-SPIM	EMBL
Nikon Digital Sight DS-Ri1	Nikon
Parafilm	Pechiney Plastic Packaging
PCR tube centrifuge	neoLab
pH-meter	Sartorius
pipetboy acu	Integra biosciences
Pipettes 2µl, 10µl, 20µl, 200µl, 1 ml	Eppendorf
Power supply PowerPac Basic	Bio RAD
PowerPac 300	Bio RAD
Q-POD	Merck Millipore
Scale EW 2200-2NM	KERN
Shaker DRS-12	neoLab
Spectrophotometer DS-11+	DeNovix
Stereomicroscope Nikon SMZ18	Nikon
Stereomicroscope Olympus SZX7	Olympus
Stereomicroscope Zeiss Stemi 2000	Zeiss
Stereomicroscope Zeiss Stemi SV11	Zeiss
Synology RS4017xs+	Synology
Synology RX1217RP	Synology
Thermal Cycler C1000 Touch™	Bio RAD
Thermal Cycler PTC-200	MJ Research
Thermomixer 5436	Eppendorf
ThermoMixer F1.5	Eppendorf
Tube revolver	Thermo Fisher Scientifc
Vortexer VF2	Janke & Kunkel
X-T 20	Fujifilm

Equipment	Supplier

Computers used

An overview over the used computers can be found in Table 6.16.

Table 6.16: Computers used in this thesis.

Component	Laptop	Desktop 1	Desktop 2
Main purpose	general, R, development, testing	headless processing	visualization
OS	Windows 10	CentOS 7.5.1804	Windows 8.1
CPU	Intel i7-5700HQ	Intel Xeon E5-2650	Intel Xeon E5-2620 v3 (2x)
RAM	32 GB DDR3	64 GB DDR3	256 GB DDR3
GPU	NVIDIA GeForce GTX 970M	AMD Radeon HD 6450	NVIDIA GeForce GTX Titan X
Storage capacity	1.4 TB	2.3 TB	22 TB
Vendor	Schenker XMG	Custom	Custom

Software and packages

The used software and packages along with the reference or license are listed in Table 6.17.

Table 6.17: Software and software packages used in this thesis.

Software/Package	Reference/License
Geneious	Biomatters Limited [Kearse et al., 2012]
Microsoft Office	Microsoft
Adobe Illustrator	Adobe
Affinity Designer	Serif Europe Ltd.

Software/Package	Reference/License
CCTOP	[Stemmer et al., 2015]
FileMaker Pro	FileMaker, Inc.
Python	Open-Source
numpy	[Oliphant, 2006]
pandas	[McKinney, 2010]
Scikit-learn	[Pedregosa et al., 2011]
Matplotlib	[Hunter, 2007]
Tensorflow	[Martin Abadi et al., 2015]
Keras	[Chollet and Others, 2015]
Fiji	[Schindelin et al., 2012]
BigDataViewer	[Pietzsch et al., 2015]
MaMuT	[Wolff et al., 2018]
LAS X	Leica, Inc.
R	[R Core Team, 2018]
condformat	[Oller Moreno, 2017]
data.table	[Dowle and Srinivasan, 2017]
ggplot2	[Wickham, 2009]
ggrepel	[Slowikowski, 2017]
readr	[Wickham et al., 2017]
xlsx	[Dragulescu, 2014]
zoo	[Zeileis and Grothendieck, 2005]

Methods

Fish husbandry and microinjections

Medaka (*Oryzias latipes*) and zebrafish (*Danio rerio*) stocks were maintained as previously described [Koster et al., 1997]. All fish are maintained in the closed stocks of COS at Heidelberg University. Fish husbandry and experiments were performed according to local animal welfare standards (Tierschutzgesetz 111,

Abs. 1, Nr. 1, Haltungserlaubnis) and in accordance with European Union animal welfare guidelines. The fish facility is under the supervision of the local representative of the animal welfare agency. Fish were maintained with a cycle of 14 h of light and 10 h of darkness. Embryos were staged according to standard protocol [Iwamatsu, 2004]. Medaka microinjections were performed as previously described [Rembold et al., 2006a].

Crossing

Crossing of medaka was performed in two ways. Either the fish were split according to sex over night and reunited at the next day to allow synchronous mating and eggs were collected subsequently or they were left in the same tank over night and eggs were collected immediately at the onset of illumination. While the first procedure was the standard procedure for wildtypes, the second procedure was used for microinjections of RSDNGSK3 fish, which will not mate if split over night, or for collection for the next generation.

Dechorionation with hatching enzyme

Fish to be dechorionated, were pre-treated as described in the following. Either they were rolled on Whatman paper immediately subsequent to egg collection, which allows for following microinjection, or they were grown to the desired stage and then rolled on sandpaper to remove the chorion hairs and weaken the chorion. At the desired stage embryos were treated with hatching enzyme and incubated at 28 °C for 60 - 120 min. Upon hatching of embryos, embryos were washed with 1 x ERM and transferred into a glass Petri dish, without touching any water/air surface. Embryos that were injected with α-*Bungarotoxin* mRNA were dechorionated latest 3 dpf, due to space restriction in the chorion, resulting in developmental malformation in properly anesthetized embryos.

Recombination of loxP constructs

Heatshock-induced recombination

Fish with an insertion of *hsp70*:*Cre* and either GaudíRSG or RSDNGSK3 were recombined by heat shock. Therefore, embryos (unhatched, dechorionated or hatched) were individually transferred to PCR tubes with 50 µl 1x ERM. Possible air was removed at the bottom of the tubes and they were transferred to a thermal cycler, where the heat shock was performed as can be extracted

from Table 6.18. The protocol is variable in length, depending on the tissue of investigation and the desired number of recombined cells. Whereas for the eye 7 cycles are advisable, for the somites 5 cycles are sufficient for a full recombination of all cells.

Table 6.18: Heatshock of single embryos via thermal cycler.

Step	Time
18 ℃	10 min
39 ℃	10 min
goto step 1	4-6 times
12 ℃	5 min

Tamoxifen-induced recombination

For tamoxifen-induced recombination of embryos containing *ccl25b*:*CreERT2* or *tlx*:*CreERT2* and GaudíRSG or RSDNGSK3, embryos were collected in a Petri dish. Upon hatching the treatment was started. Therefore, a new Petri dish was filled with 5 µM tamoxifen in 1x ERM. Embryos were transferred to the new Petri dish individually with a pasteur pipette and food was added. Fish were incubated in the dark, over night and subsequently washed three times with 1x ERM, being transferred with a pasteur pipette to minimize contamination with tamoxifen. Fish were raised according to protocol between 2 days to 4 weeks until fixation and followed by staining and investigation via microscopy described in the following section.

Fixation of fish

Fish were euthanized by a 20 x Tricaine solution. Euthanization was ensured by checking vital signs, such as gill movement and stimulus reaction. Subsequently, fish were transferred to 4 % w/v PFA and fixed for a stage-dependent time at 4 °C. Hatchlings were fixed for 2-2.5 h, juveniles were fixed for 2.5-4 h, young adults were fixed for 4-7 h, large adults were fixed for 6 h to over night. Finally, fish were washed 3 x in 1 x PTW.

Extraction of genomic DNA for PCR

Genomic DNA of single embryos for subsequent PCR was extracted and used for standard PCR mixes as published [Hammouda et al., 2019].

Total RNA extraction

Medaka embryos were collected and incubated in 1 x ERM until they reached the desired developmental stage [Iwamatsu, 2004]. The embryos were euthanized as described prior, transferred to 2 ml Eppendorf tubes and 700 µl Trizol (invitrogen) was added. Afterwards the embryos were homogenized with a pestle, which were subsequently stored in 0.5 N HCl for cleaning. The ground tissue was spun down for 1 min at 10,000 g and the supernatant was transferred to a new 2 ml Eppendorf tube. After addition of 300 µl Trizol the reaction mix was incubated for 5 min at room temperature (RT), thereafter 200 µl of Chloroform were added, followed by 15 s of rigorous shaking. This was incubated for 3-10 min at RT and centrifuged at 10,000 g for 5 min at 4 °C. The upper, colorless phase was transferred to an 1.5 ml Eppendorf tube and 500 µl 2-Propanol were added, followed by rigorous shaking after which the sample was incubated on ice for 10 min. Subsequently the sample was centrifuged at 10,000 g for 10 min at 4 °C and the supernatant was removed while checking for pellet presence. The present pellet was washed with 1 ml of 75 % Ethanol and centrifuged twice at 8,000 g for 10 min at 4 °C. Thereafter the supernatant was discarded by extracting it with a pipette tip and the pellet was dried for 5 min at RT. To dissolve the pellet 10 - 20 µl H_2O were added, where the amount of added water was dependent on pellet size. The pellet was solved by gently flicking the tube and the isolated ribonucleic acid (RNA) was stored at -80 °C until further methods were applied.
For quality control the RNA concentration of the obtained extractions was determined with the NanoDrop, while comparing the $\frac{OD\,260}{OD\,280}$ and $\frac{OD\,260}{OD\,230}$ values to optimal values (1.8 and 2.0, respectively). As a second part of quality control the RNA was run with a RNA gel electrophoresis to determine the 28 S and 18 S ribosomal bands and general intensity.

Reverse transcription

DNase treatment

1 µg of extracted RNA was treated with 1 µl 10 x reaction buffer with $MgCl_2$ and 1 µl DNase I (1U) in 10 µl reaction volume and incubated for 30 min at 37 °C. Subsequently, 50 mM EDTA was added and the reaction was incubated for 10 min at 65 °C. The prepared RNA can be directly used for reverse transcription.

Reverse transcription

RNA was reverse transcribed with the RevertAid™Kit (Thermo Fisher Scientifc, all following components are part of the Kit). 1 µg of total RNA was mixed with 0.5 µl oligo dT_{18} primer and 0.5 µl Random hexamer primers in a PCR tube and filled up with RNase-free water to a volume of 12 µl. The reaction was incubated at 65 °C for 5 min and subsequently chilled on ice for at least 1 min. Afterwards 4 µl of reverse transcription buffer, 1 µl of RiboLock, 2 µl of 10 mM dNTPs and 1 µl of RevertAid™Reverse Transcriptase were added. The assembled reaction was gently mixed and spun down in a table centrifuge. Afterwards it was incubated at 25 °C for 5 min, followed by an incubation at 42 °C for 60 min. Accordingly, the reaction was terminated by incubation at 70 °C for 5 min and chilled on ice for at least 1 min. 1 µl RNase H was added and incubated for 20 min at 37 °C. Subsequently, 19 µl H_2O were added and the obtained cDNA was stored at -20 °C prior to usage.

Extraction of RNA and genomic DNA

Trizol treatment

Embryos were euthanized by a 20 x Tricaine solution and transferred individually to reaction tubes. Liquid was substituted by 700 µl Trizol and the embryos were homogenized by using a pestle. Samples were centrifuged for 10 min at 12,000 g at 4 °C. The clear supernatant was transferred to a new reaction tube, 300 µl Trizol were added and incubated for 5 min at RT. Subsequently 200 µl Chloroform were added, the content was mixed by shaking and incubated for 2-3 min at RT. Samples were centrifuged for 15 min at 12,000 g at 4 °C. The aqueous, colorless, upper phase was transferred to a new reaction tube (approximately 500 µl), which was further used in RNA extraction. The leftover lower, red phenol-chloroform phase and interphase were further used in genomic DNA (gDNA) extraction.

RNA extraction

500 µl of Isopropanol were added to the previously extracted aqueous phase, containing RNA. Samples were mixed and incubated for 10 min at RT. Samples were then centrifuged for 10 min at 12,000 g at 4 °C. The supernatant was discarded by pipetting and the pellet was washed with 1 ml 75 % ethanol. The supernatant was again discarded by pipetting and the pellet was air-dried for 5-10 min. The pellet was resuspended in 20 µl RNase-free water by pipetting. Samples were then incubated for 10 min at 60 °C. RNA was stored at -80 °C until reverse transcription.

Genomic DNA extraction

Any remaining aqueous phase was removed from the previously obtained samples. 300 µl of 100 % ethanol were added, mixed by inversion and incubated for 2-3 min. Samples were centrifuged for 5 min at 2,000 g at 4 °C. The supernatant was discarded with the pipette (could also be used for protein extraction, if needed) and the pellet was resuspended in 1 ml of 0.1 M sodium citrate in 10 % ethanol, incubated for 30 min and occasionally mixed by gentle inversion. Samples were centrifuged for 5 min at 2,000 g at 4 °C and the supernatant was discarded. Again, the pellet was resuspended in 1 ml of 0.1 M sodium citrate in 10 % ethanol, incubated for 30 min, occasionally mixed by gentle inversion and centrifuged for 5 min at 2,000 g at 4 °C. The supernatant was discarded, and the pellet was resuspended in 1.5 ml 75 % ethanol, incubated for 10-20 min while occasionally inverting. The resuspended solution was centrifuged for 5 min at 2,000 g at 4 °C and the supernatant was again discarded. The pellet was air-dried for 5-10 min and resuspended in 50 µl TE by pipetting. This DNA solution can be stored and used for subsequent PCR.

Whole mount immunostaining

Retinae extraction

Fixed embryos were prebleached in 3 % v/v H_2O_2 and 0.5 % w/v KOH in 1 x PTW. Retinae were extracted in 1 x PTW utilizing two forceps. If the fish was older than 7 dph, the lens was also enucleated from the retina.

Bleaching

Fixed embryos or extracted retinae were bleached in 3 % v/v H_2O_2 and 0.5 % w/v KOH in 1 x PTW for 2 h or until no residual pigment was detectable. Samples were incubated in open reaction tubes due to the strong gas development during the procedure.

Acetone treatment

Samples were washed 5 x in 1 x PTW for 5 min each at RT. These samples can be stored at 4 °C for 1-2 d. Subsequently samples were transferred to a glass tube, containing acetone and incubated for 20 min at -20 °C. Samples were rinsed by H_2O.

Blocking

Samples were washed 5 x in 1 x PTW for 5 min at RT and subsequently incubated in blocking solution in 2 ml reaction tubes for 2 h at RT or over night at 4 °C.

Primary antibody incubation

Blocked samples were incubated with primary antibody solution, containing rabbit-α-Rx2 (1:250, lab made) and chicken-α-GFP (1:200, Thermo Fisher Scientifc) in blocking buffer for 3 x over night at 4 °C.

Secondary antibody incubation

Samples were washed once in 1 x PTW and transferred to a new reaction tube. Samples were again washed 5 x in 1 x PTW. Samples were incubated with secondary antibody solution containing donkey-α-chicken DyLight488 (1:250, Jackson), goat-anti-rabbit AlexaFluor647 (1:125, Thermo Fisher) and DAPI (1:250) in blocking buffer for 2 x over night at 4 °C in the dark. Subsequently, samples were washed 5 x in 1 x PTW in the dark. If samples should be stored, they were transferred to 1 % w/v PFA in 1 x PTW and stored at 4 °C until further analysis via microscopy.

TUNEL

For TUNEL, already stained retinae were incubated in 10 $\mu g/ml$ Proteinase K for 30-45 min at RT. Samples were then postfixed in 4 % w/v PFA for 20 min at RT and subsequently washed 4 x in 1 x PTW for 5 min each. Samples were

incubated in 33 % acetone in ethanol for 20 min at -20 °C and washed 3 x in 1 x PTW for 5 min each. Staining was performed with 90 µl labeling solution and 10 µl enzyme solution from the In Situ Cell Death Detection Kit (Roche) for 2 h at 37 °C. Samples were washed 3 x in 1 x PTW for 5 min each and subsequently imaged and analyzed.

Oligonucleotide design and ordering

Oligonucleotides for PCR were designed via Geneious and, if needed, tested via *in silico* PCR in the UCSC browser on the whole medaka genome. Oligonucleotides for sgRNAs were designed using CCTOP [Stemmer et al., 2015]. Oligonucleotides for Q5 site-directed mutagenesis were designed using the NEBaseChanger® (http://nebasechanger.neb.com/). All designed primers were ordered from MWG via a custom FileMaker script and thereby also given a unique identifier.

PCR

The PCRs were set up according to the recommendations of the manufacturer, New England Biolabs (NEB), which can be obtained from Table 6.19.

Table 6.19: Standard PCR mix.

Component	50 µl reaction	Final conc.
5 x Q5 Reaction buffer (NEB)	10 µl	1 x
2.5 mM dNTPs (Sigma-Aldrich)	4 µl	200 µmol/l
10 µM Forward Primer	1 µl	0.2 µmol/l
10 µM Reverse Primer	1 µl	0.2 µmol/l
Template DNA	0.1-10 ng	0.002 - 0.2 ng/µl
Q5 DNA-Polymerase (NEB)	0.5 µl	0.02 U/µl
RNase-free H_2O	ad 50 µl	

Depending on the template and the primers the PCR conditions such as annealing temperature (calculated with http://tmcalculator.neb.com) and elongation time were adapted to optimize the reaction.

semiquantitative PCR

For semiquantitative PCR mRNA was extracted from wt, GaudíRSG, RSDNGSK3_high and RSDNGSK3_low fish as previously described. A standard 50 µl PCR reaction was assembled with primers for mCherry (JW5653 and JW3667) and primers for Actb (JW2915 and JW1452). Primer choice was based on previous tests of several primer pairs for both transcripts. The PCR was performed according to standard protocol, but an aliquot of 5 µl was taken from each reaction each 5 cycles between 20 and 35 cycles in order to get a sample, in which the amplification was still in an exponential phase and not in a satured phase. The samples were analyzed via gel electrophoresis. For this all the samples of one tp, including control and experiment (Actin B (Actb) and mCherry) were loaded onto the same gel to ensure comparability. Band intensity was documented by digital acquisition of the gel documentation and analyzed by Fiji. Mean values of bands were extracted and mCherry values were divided by Actb values for loading control. The mean intensity of the position of the not present mCherry band in wt was subtracted from all other values to control for background noise. The resulting values were plotted using ggpubr and analysis on the difference of expression of RSDNGSK3_high and RSDNGSK3_low was performed. Previous calibration with a 1:10 dilution could be used to determine the relative expression difference between both lines.

Q5 site-directed mutagenesis

Oligonucleotides were designed as previously described. These primers were used in a standard PCR reaction to amplify a mutated version of the vector. This reaction mix was treated with polynucleotide kinase (PNK), T4 DNA Ligase and DpnI (KLD treatment) for 10 min at RT. Therefore, a 15 µl reaction mix was set up containing 1 µl of the PCR product, 9.5 µl H_2O, 1.5 µl CutSmart Buffer (NEB), 1.5 µl 10 mM adenosine triphosphate (ATP) (Thermo Fisher Scientifc), 0.5 µl PNK (NEB), 0.5 µl T4 DNA Ligase (Thermo Fisher Scientifc) and 0.5 µl DpnI (NEB). The PNK added 5' phosphates to the PCR fragments to allow ligation, DpnI was digesting the bacterial amplified deoxyribonucleic acid (DNA) (PCR template) and the T4 DNA Ligase ligated the mutated vectors.
Finally, the mutated and ligated vector was transformed into bacteria according to protocol.

Oligonucleotide annealing

Oligonucleotide annealings were set up in a PCR reaction tube with 18 µl of dH_2O, 20 µl annealing buffer and 1 µl of a 100 µM dilution of each oligonucleotide. The oligonucleotides were annealed in the PCR cycler with the program outlined in Table 6.20.

Table 6.20: PCR cycler program for the annealing of oligonucleotides.

95 °C for 5 min
ramp down to 70 °C (0.1 °C/s)
hold for 10 min
ramp down to 65 °C (0.1 °C/s)
hold for 10 min
ramp down to 60 °C (0.1 °C/s)
hold for 10 min
ramp down to 10 °C (0.1 °C/s)

The annealed product was diluted to 0.075 pmol/µl (1 µl of the annealing reaction was diluted with 32 µl H_2O). Thereafter, 1 µl of this dilution was used for ligation with 0.025 pmol of the desired vector, with 1 µl PEG-4000 (Thermo Fisher Scientifc), 1 µl 10 x Ligation buffer (Thermo Fisher Scientifc) and 1 µl T4 Ligase (5 U/µl, Thermo Fisher Scientifc) filled up to 10 µl reaction volume with H_2O and incubated for at least 20 min at RT or over night at 4 °C.

Gel electrophoresis

DNA gel electrophoresis

Gel electrophoresis was performed with 1.0 % w/v or 1.5 % w/v Agarose in 1 x TAE gels, depending on DNA sample size, in chambers filled with 1 x TAE. The samples were mixed with loading dye and loaded into the wells of the gel. The gel was run at ≈ 10 V/cm. Subsequently the gel was stained in a 0.0002 % v/v ethidium bromide (EtBr) bath and illuminated in an UV transilluminator. GeneRuler™DNA Ladder Mix (Thermo Fisher Scientifc) was used as a reference for determining the size of DNA fragments in basepairs.

For extraction of DNA, the corresponding band was excised using a scalpel blade and transferred into a 2 ml Eppendorf tube. DNA gel purification was conducted with innuPREP DOUBLEpure Kit according to protocol (Analytik Jena, https://www.analytik-jena.de/fileadmin/content/pdf_life_science/Manual/Manual_innuPREP_DOUBLEpure_Kit.pdf) or QIAquick Gel Extraction

Kit according to protocol (QIAGEN, https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en).

RNA gel electrophoresis

RNA was run in a non-denaturing agarose gel. Therefore the comb, sledge and chamber were preincubated with 0.1 N NaOH for 30 min. The gel was freshly prepared with 1 x TAE prepared with Millipore water and pre-run for 10 min. The samples were prepared with 2 x RNA loading dye containing formamide and incubated at 80 °C for 10 min to ensure the unfolding of secondary structures prior to loading in the wells. The gel electrophoresis and documentation was conducted according to the DNA gel electrophoresis protocol.

Molecular cloning

Ligation

DNA ligation was performed according to Table 6.21 depending on the concentration of insert and vector.

Table 6.21: DNA ligation.

Component	10 µl reaction	Final conc.
Vector	n pmol	$\frac{n}{10}\frac{pmol}{\mu l}$
Insert	$3n$ pmol	$\frac{3n}{10}\frac{pmol}{\mu l}$
10x DNA Ligase buffer (Thermo Fisher Scientifc)	1 µl	1 x
T4 DNA Ligase (5 U/µl, Thermo Fisher Scientifc)	1 µl	0.5 U/µl
RNase-free H_2O (Sigma-Aldrich)	ad 10 µl	

After assembly the reaction was incubated for 10 - 120 min at RT or over night at 4 °C.

Transformation of chemically competent *E. coli*

A 50 µl aliquot of Mach1™T1^{R} competent cells (Life Technologies) was thawed on ice and transformed with up to 5 µl of DNA solution. The bacteria were incubated for 5 min on ice, followed by heat-shocking at 42 °C for 30 s. Subsequently the bacteria were chilled on ice for 1 min and 300 µl T-B buffer was added. The bacteria were incubated for 10 to 60 min at 37 °C depending on the antibiotic resistance present on the transformed plasmid. Subsequently, bacteria were plated onto LB plates with antibiotic for selection utilizing glass beads. The plated volume of transformed bacteria and used antibiotic was dependent on the transformed plasmid. The plates were incubated at 37 °C over night.

Plasmid mini preparation

For a small amount of plasmid 2 ml LB medium with the appropriate antibiotic depending on the plasmid were inoculated with a single colony and incubated over night at 37 °C and 200 rpm. On the next day the cultures were transferred to a 2 ml eppendorf tube and spun down at 21,000 g for 2 min. The supernatant was discarded and 200 µl of P1 buffer was added. The pellet was solved and 200 µl P2 buffer was added followed by inverting 5 - 6 times and incubation at RT for up to 5 min. After incubation 200 µl of P3 buffer was added and the sample was thoroughly mixed by inverting 5 - 6 times. The mix was spun down at 21,000 g for 15 min at 15 °C and the resulting supernatant was transferred to a new 1.5 ml eppendorf tube followed by addition of 500 µl of 2-Propanol. The tubes were shaken rigorously and afterwards centrifuged at 15 °C and 21,000 g for 15 min. The resulting supernatant was discarded and 500 µl of 70 % Ethanol were added followed by centrifugation at 15 °C and 21,000 g for 5 min. The supernatant was discarded, and the pellet was air-dried for 5 - 10 min. The dried pellet was solved in 40 µl H_2O for further usage.

Plasmid mini preparation with QIAGEN kit

For a larger and cleaner preparation of plasmid, which is sufficient for transcription of mRNAs and sgRNAs, a Mini preparation was performed using the QIAPrep Spin Miniprep Kit (QIAGEN). Therefore, 20 ml LB medium with the appropriate antibiotic depending on the plasmid were inoculated in a 50 ml Erlenmeyer flask with a single colony and incubated over night at 37 °C and 200 rpm. Per culture two 2 ml reaction tubes were filled with the incubated

LB medium and spun down at 8,000 g for 2 min at RT. The supernatant was discarded, incubated LB medium was again transferred to the same tubes and the reaction tubes were centrifuged at 8,000 g for 2 min at RT. The supernatant was discarded, and the pellet was completely resuspended in 250 µl P1 resuspension buffer. 250 µl of P2 lysis buffer were added and the reaction was mixed thoroughly by inverting 4-6 times, lysis was indicated by blue color switch. 350 µl N3 neutralization buffer were added and the reaction was mixed thoroughly by inverting 4-6 times, neutralization was indicated by white/colorless color switch. The reaction was centrifuged for 10 min at 16,000 g and the supernatant (≈850 µl) of the first reaction tube was transferred to a spin column, contained in the kit. The column was centrifuged for 1 min at 16,000 g and the flow-through was discarded. The same centrifugation steps were repeated with the supernatant of the second reaction tube. Subsequently, the column was washed by addition of 750 µl PE washing buffer and centrifuging for 1 min at 20,000 g, the flow-through was discarded and the column was centrifuged for 1 min at 20,000 g to remove residual washing buffer. The column was placed into an RNase-free reaction tube and 50 µl RNase-free water were added to the filter followed by an incubation for 4 min at RT. The tube was centrifuged for 1 min at 16,000 g, the flow-through was again applied to the column and the tube was centrifuged for 1 min at 16,000 g. The resulting solution can be used for transcription of mRNAs or sgRNAs.

Plasmid midi preparation

For a very large and clean preparation of plasmid the QIAGEN Plasmid Midi Kit was used. 50 ml LB-medium were set up in a 250 ml Erlenmeyer flask. The appropriate antibiotic was added, depending on the amplified plasmid. Inoculated was either a single colony from a plate or a leftover of an over night culture for Mini preparation. The inoculated culture was incubated at 37 °C and 200 rpm shaking over night. The resulting culture was transferred to a 50 ml falcon and spun down at 4 °C and 4,000 g for 30 min. The supernatant was discarded and 4 ml of P1 buffer was added followed by solving the pellet via vortexing. Afterwards 4 ml of P2 buffer was added and the falcon was inverted for 4 - 5 times. After 5 min of incubation at RT 4 ml of P3 buffer was added. The falcon was inverted 4 - 5 times and incubated on ice for 15 min. Meanwhile a QIAGEN-tip column was prepared by adding a funnel and a filter and equilibrating both by adding 4 ml QBT (QIAGEN). The mixture was applied to the filter and after flow-through the column was washed by

adding 10 ml QC buffer (QIAGEN) two times. Afterwards the column was placed on top of a 15 ml falcon and 5 ml QF buffer (QIAGEN) was added. After flow-through 3.5 ml of 2-Propanol were added and the mixture was shaken rigorously. Afterwards the falcon was centrifuged at 4 °C and 4,000 g for 60 min. The resulting supernatant was discarded and 2 ml of 70 % Ethanol were added, followed by centrifugation at 4 °C and 4,000 g for 30 min. The supernatant was discarded, and the pellet was air-dried. The dried DNA was solved in 50 - 100 µl TE depending on pellet size.

Restriction digest

DNA restriction digestion was conducted according to Table 6.22. The 15 µl reaction mix has been used for the determination of successful ligation of insert and vector after plasmid purification, whereas the 50 µl reaction mix has been used for digestion of already purified plasmids.

Table 6.22: DNA restriction digest.

Component	**15 µl reaction**	**50 µl reaction**
DNA template	1 µg	1-10 µg
Enzyme	0.2-0.3 µl	0.5-1 µl
optional: second Enzyme	0.3 µl	0.5 µl
corresponding 10x Buffer	1.5 µl	5 µl
RNase-free H_2O (Sigma-Aldrich)	ad 15 µl	ad 50 µl

After assembly the 15 µl reaction was incubated for 1 to 1.5 h at 37 °C and the 50 µl reaction overnight at 37 °C. The 15 µl test restriction digests were analyzed via gel electrophoresis to test for successful ligation of the insert into the vector. Whereas the 50 µl restriction digests were either directly purified via a purification kit (innuPREP DOUBLEpure kit according to protocol https://www.analytik-jena.de/fileadmin/content/pdf_life_science/Manual/Manual_innuPREP_DOUBLEpure_Kit.pdf or QIAquick PCR purification kit according to protocol https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en) or separated by gel electrophoresis, extracted from the gel and afterwards puri-

fied via a purification Kit (innuPREP DOUBLEpure kit according to protocol https://www.analytik-jena.de/fileadmin/content/pdf_life_science/Manual/Manual_innuPREP_DOUBLEpure_Kit.pdf or QIAquick Gel Extraction kit according to protocol https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en), depending on the digested template DNA.

DNA sequencing

Standard sequencing of DNA templates was performed by MWG according to manufacturer's protocol.

Codon adaptation

Codon adaptation was performed as described in [Lischik et al., 2019].

CRISPR/Cas9

Cas9 mRNA transcription

10 µg of the plasmid # 5197 were digested using NotI-HF (NEB) and completed linearization was checked by gel electrophoresis of 250 ng of the digestion. Subsequent to successful linearization the digested plasmid was purified by QIAquick PCR Purification kit (QIAGEN according to protocol https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en) and eluted twice in 40 µl RNase-free water. Transcription was performed with mMessage mMachine® Sp6 Transcription Kit (Invitrogen, according to protocol https://assets.thermofisher.com/TFS-Assets/LSG/manuals/cms_055516.pdf) with 1 µg template DNA, according to protocol. Finally, mRNA was purified by RNeasy Mini Kit (QIAGEN, according to protocol https://www.qiagen.com/us/resources/download.aspx?id=14e7cf6e-521a-4cf7-8cbc-bf9f6fa33e24&lang=en). Transcription was checked by gel electrophoresis and the resulting mRNA was aliquoted.

sgRNA transcription

The larger, clean mini plasmid preparation of the template plasmid was digested with DraI-FD (Thermo Fisher Scientifc) over night at 37 °C, the resulting bands were separated by gel electrophoresis and the 300 bp band purified via innuPREP DOUBLEpure kit (Analytik Jena, according to

protocol https://www.analytik-jena.de/fileadmin/content/pdf_life_science/Manual/Manual_innuPREP_DOUBLEpure_Kit.pdf). Transcription was performed with MEGAShortScript T7 Kit according to manufacturer's protocol (https://assets.thermofisher.com/TFS-Assets/LSG/manuals/fm_1354.pdf) and sgRNA was purified via RNeasy Mini Kit (QIAGEN, according to protocol https://www.qiagen.com/us/resources/download.aspx?id=14e7cf6e-521a-4cf7-8cbc-bf9f6fa33e24&lang=en). The concentration was measured by a spectrophotometer and RNA integrity was checked via gel electrophoresis.

mRNA transcription

Transcription of mRNA was performed by digesting the plasmid with the corresponding enzyme and purification of the template via kit (QIAquick Gel Extraction Kit according to protocol https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en, QIAquick PCR purification kit according to protocol https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en or innuPREP DOUBLEpure Kit according to protocol https://www.analytik-jena.de/fileadmin/content/pdf_life_science/Manual/Manual_innuPREP_DOUBLEpure_Kit.pdf). Transcription was performed with mMessage mMachine Sp6 Transcription Kit (Invitrogen, according to protocol https://assets.thermofisher.com/TFS-Assets/LSG/manuals/cms_055516.pdf) according to protocol and mRNAs were purified with RNeasy Mini Kit (QIAGEN, according to protocol https://www.qiagen.com/us/resources/download.aspx?id=14e7cf6e-521a-4cf7-8cbc-bf9f6fa33e24&lang=en) according to protocol. Integrity of transcribed mRNAs was ensured via gel electrophoresis and spectrophotometer.

Transcription of fluorescent proteins

Fluorescent protein coding sequences have been cloned into GoldenGate entry vector 3 [Kirchmaier et al., 2013b] with a start and a stop codon in order to ensure comparability of transcribed mRNAs [Lischik et al., 2019]. Vectors were digested with SpeI-HF (NEB) and mRNA was transcribed according to the previously described protocol.

Transcription of α-Bungarotoxin

For transcription of α-Bungarotoxin 10 µg of the vector have been digested by EcoRV-FD (Thermo Fisher Scientifc), the digested plasmid has been purified via gel electrophoresis and extraction via QIAquick Gel Extraction Kit (QIAGEN according to protocol https://www.qiagen.com/us/resources/download.aspx?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en). Transcription was performed according to standard protocol, except 2 µg of linearized plasmid were used and the reaction was incubated for 4 h at 37 °C. During clean up via RNeasy Mini Kit (QIAGEN, according to protocol https://www.qiagen.com/us/resources/download.aspx?id=14e7cf6e-521a-4cf7-8cbc-bf9f6fa33e24&lang=en) the mRNA was eluted twice with 25 µl water pre-warmed to 50 °C.

Microscopy

Light-sheet microscopy using SPIM

Glass capillaries were shortened to a specified length using a lab made tool for holding the capillaries at a certain length while using the Diamond Cutter. Afterwards FEP tubes were cut and fixed into one end of the capillaries and the compound capillaries were disinfected with 70 % Ethanol at least over night. The compound capillaries were cleared from ethanol and dried for at least one night. Medaka embryos were anesthetized with 1x Tricaine or by α-*Bungarotoxin* mRNA microinjection. Anesthetized embryos were transferred to a drop of 0.6 % w/v low-melt Agarose and sucked and adjusted into the FEP tube with the help of an Eppendorf pipette and a 200µl pipette tip. Importantly, the capillary should be filled with agarose to prevent movement of the sample. The agarose was allowed to try, subsequently the compound capillaries were mounted into the microscope and stacks were acquired with a MuVi-SPIM (multiview selective plane illumination microscope) [Krzic et al., 2012, Tomer et al., 2012, de Medeiros et al., 2015] configured as described before for the 25x detection setup [Caroti et al., 2018]. Added was a 525/50 nm bandpass filter and a 488 nm illumination.
The measurements of fluorescence intensity were performed as published previously [Lischik et al., 2019].

High-throughput time lapse imaging

High-throughput time lapse imaging was performed as published previously [Lischik et al., 2019].

Photography of adults

Adult fish were anesthetized by 1 x Tricaine and transferred into an agarose-coated Petri dish. Images of the fish were taken by an X-T 20 digital camera (Fujifilm) using a macro lens.

Leica Sp8

For imaging of whole-mount immunostained retinae, samples were mounted in glass bottom dishes (MatTek) and imaged using matching laser and PMT settings. Acquisition was adjusted such that no overexposure was detectable.

Image and data analysis

General image analysis was performed with Fiji [Schindelin et al., 2012]. All scripts used for the analysis, which are published in [Lischik et al., 2019] can be found at github via: https://git.io/fAPnh. The machine and deep learning scripts for classification and time course analysis of fluorescent protein data can be found in the appendix.

Extraction of polyclonal features

The polyclonal features of imaged stacks were extracted via Fiji. Polyclonal connection with the CMZ was defined as overlap/non-overlap with the rx2 expression domain. All polyclones were classified into either group and the number of clones for each group was noted. Distance to the CMZ was measured from the most lateral position of the CMZ to the the most lateral position of the clone. Clone width was measured in µm and cell diameters based on DAPI staining. The retina diameter was measured twice and averaged for more robust measurement.

Workflow for SPIM data

Acquired SPIM data was copied to the Synology storage solution for accessibility. A BigDataViewer compatible .xml file was created by using a custom LabView program supplied by the Hufnagel lab at EMBL Heidelberg, Germany. The

BigDataViewer [Pietzsch et al., 2015] was used for screening and visualization of time-series data. Single cells in GaudíRSG retinae were tracked by MaMuT [Wolff et al., 2018] and further analyzed by custom Python scripts, which can be found in the appendix.

Startle response assay

The startle response assay was performed as published previously [Lischik et al., 2019].

Comparison of medaka and *E. coli in vivo* fluorescence intensity

Comparison of medaka ac *E. coli in vivo* fluorescence intensity was performed as published previously [Lischik et al., 2019].

Semi-automated analysis of anesthesia movement profiles

Semi-automated analysis of anesthesia movement profiles was performed as published previously [Lischik et al., 2019]. Customized scripts are available through github (https://git.io/fAPnh).

Semi-automated analysis of fluorescent intensities of fluorescent proteins

Semi-automated analysis of fluorescent intensities of fluorescent proteins was performed as published previously [Lischik et al., 2019]. Customized scripts are available through github (https://git.io/fAPnh).

References

[Abdul et al., 2018] Abdul, A. u. R. M., De Silva, B. and Gary, R. K. (2018). The GSK3 kinase inhibitor lithium produces unexpected hyperphosphorylation of β-catenin, a GSK3 substrate, in human glioblastoma cells Ata. Biol. Open *7*, bio030874.

[Aghaallaei et al., 2016] Aghaallaei, N., Gruhl, F., Schaefer, C. Q., Wernet, T., Weinhardt, V., Centanin, L., Loosli, F., Baumbach, T. and Wittbrodt, J. (2016). Identification, visualization and clonal analysis of intestinal stem cells in fish. Development *dev.134098*, dev.134098.

[Amat et al., 2014] Amat, F., Lemon, W., Mossing, D. P., McDole, K., Wan, Y., Branson, K., Myers, E. W. and Keller, P. J. (2014). Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. Nat. Methods *11*, 951–958.

[Amato et al., 2004] Amato, M. A., Arnault, E. and Perron, M. (2004). Retinal stem cells in vertebrates: parallels and divergences. Int. J. Dev. Biol. *48*, 993–1001.

[Ansai and Kinoshita, 2014] Ansai, S. and Kinoshita, M. (2014). Targeted mutagenesis using CRISPR/Cas system in medaka. Biol. Open *3*, 362–371.

[Antinucci and Hindges, 2016] Antinucci, P. and Hindges, R. (2016). A crystal-clear zebrafish for in vivo imaging. Sci. Rep. *6*, 29490.

[Athey et al., 2017] Athey, J., Alexaki, A., Osipova, E., Rostovtsev, A., Santana-Quintero, L. V., Katneni, U., Simonyan, V. and Kimchi-Sarfaty, C. (2017). A new and updated resource for codon usage tables. BMC Bioinformatics *18*, 1–10.

[Balleza et al., 2017] Balleza, E., Kim, J. M. and Cluzel, P. (2017). A systematic characterization of maturation kinetics of fluorescent proteins in live cells. Nat. Methods *6*, 1–10.

[Barbosa et al., 2015] Barbosa, J. S., Sanchez-gonzalez, R., Di Giaimo, R., Baumgart, E. V., Theis, F. J., Götz, M., Ninkovic, J., Giaimo, R. D., Baumgart, E. V., Theis, F. J. and Ninkovic, J. (2015). Live imaging of adult neural stem cell behavior in the intact and injured zebrafish brain. Science *348*, 789–93.

[Beurel et al., 2015] Beurel, E., Grieco, S. F. and Jope, R. S. (2015). Glycogen synthase kinase-3 (GSK3): Regulation, actions, and diseases. Pharmacol. Ther. *148*, 114–131.

[Bindels et al., 2016] Bindels, D. S., Haarbosch, L., van Weeren, L., Postma, M., Wiese, K. E., Mastop, M., Aumonier, S., Gotthard, G., Royant, A., Hink, M. A. and Gadella Jr, T. W. J. (2016). mScarlet: a bright monomeric red fluorescent protein for cellular imaging. Nat. Methods *14*, 53.

[Bjarnason et al., 2001] Bjarnason, G. A., Jordan, R. C., Wood, P. A., Li, Q., Lincoln, D. W., Sothern, R. B., Hrushesky, W. J. and Ben-David, Y. (2001). Circadian expression of clock genes in human oral mucosa and skin: Association with specific cell-cycle phases. Am. J. Pathol. *158*, 1793–1801.

[Borday et al., 2012] Borday, C., Cabochette, P., Parain, K., Mazurier, N., Janssens, S., Tran, H. T., Sekkali, B., Bronchain, O., Vleminckx, K., Locker, M. and Perron, M. (2012). Antagonistic cross-regulation between Wnt and Hedgehog signalling pathways controls post-embryonic retinal proliferation. Development *139*, 3499–3509.

[Caroti et al., 2018] Caroti, F., González Avalos, E., Noeske, V., González Avalos, P., Kromm, D., Wosch, M., Schütz, L., Hufnagel, L. and Lemke, S. (2018). Decoupling from yolk sac is required for extraembryonic tissue spreading in the scuttle fly Megaselia abdita. Elife *7*, 1–19.

[Caussinus et al., 2011] Caussinus, E., Kanca, O. and Affolter, M. (2011). Fluorescent fusion protein knockout mediated by anti-GFP nanobody. Nat. Struct. Mol. Biol. *19*, 117–121.

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

[Centanin et al., 2011] Centanin, L., Hoeckendorf, B. and Wittbrodt, J. (2011). Fate Restriction and Multipotency in Retinal Stem Cells. Cell Stem Cell *9*, 553–562.

[Chami et al., 2016] Chami, B., Steel, A. J., De La Monte, S. M. and Sutherland, G. T. (2016). The rise and fall of insulin signaling in Alzheimer's disease. Metab. Brain Dis. *31*, 497–515.

[Chavali et al., 2018] Chavali, M., Klingener, M., Kokkosis, A. G., Garkun, Y., Felong, S., Maffei, A. and Aguirre, A. (2018). Non-canonical Wnt signaling regulates neural stem cell quiescence during homeostasis and after demyelination. Nat. Commun. *9*, 36.

[Chen et al., 2001] Chen, S., Guttridge, D. C., You, Z., Zhang, Z., Mayo, M. W., Kitajewski, J., Wang, C.-y., The, S., Biology, C., Jan, N., Fribley, A. and Mayoji, M. W. (2001). Wnt-1 Signaling Inhibits Apoptosis by Activating

beta-Catenin / T Cell Factor-mediated Transcription. J. Cell Biol. *152*, 87–96.

[Chhetri et al., 2015] Chhetri, R. K., Amat, F., Wan, Y., Höckendorf, B., Lemon, W. C. and Keller, P. J. (2015). Whole-animal functional and developmental imaging with isotropic spatial resolution. Nat. Methods *12*, 1171–1178.

[Chollet and Others, 2015] Chollet, F. and Others (2015). Keras. \url{https://keras.io}.

[Chow and Lang, 2001] Chow, R. L. and Lang, R. A. (2001). Early eye development in vertebrates. Annu. Rev. Cell Dev. Biol. *17*, 255–296.

[Crowder and Freeman, 2000] Crowder, R. J. and Freeman, R. S. (2000). Glycogen synthase kinase-3β activity is critical for neuronal death caused by inhibiting phosphatidylinositol 3-kinase or Akt but not for death caused by nerve growth factor withdrawal. J. Biol. Chem. *275*, 34266–34271.

[Culver and Dickinson, 2010] Culver, J. C. and Dickinson, M. E. (2010). The effects of hemodynamic force on embryonic development. Microcirculation *17*, 164–178.

[DasGupta et al., 2002] DasGupta, R., Rhee, H. and Fuchs, E. (2002). A developmental conundrum: A stabilized form of β-catenin lacking the transcriptional activation domain triggers features of hair cell fate in epidermal cells and epidermal cell fate in hair follicle cells. J. Cell Biol. *158*, 331–344.

[de Medeiros et al., 2015] de Medeiros, G., Norlin, N., Gunther, S., Albert, M., Panavaite, L., Fiuza, U.-M., Peri, F., Hiiragi, T., Krzic, U. and Hufnagel, L. (2015). Confocal multiview light-sheet microscopy. Nat. Commun. *6*, 8881.

[Denayer et al., 2008] Denayer, T., Locker, M., Borday, C., Deroo, T., Janssens, S., Hecht, A., van Roy, F., Perron, M. and Vleminckx, K. (2008). Canonical Wnt signaling controls proliferation of retinal stem/progenitor cells in postembryonic Xenopus eyes. Stem Cells *26*, 2063–74.

[Dowle and Srinivasan, 2017] Dowle, M. and Srinivasan, A. (2017). data.table: Extension of 'data.frame'.

[Dragulescu, 2014] Dragulescu, A. A. (2014). xlsx: Read, write, format Excel 2007 and Excel 97/2000/XP/2003 files.

[Dray et al., 2015] Dray, N., Bedu, S., Vuillemin, N., Alunni, A., Coolen, M., Krecsmarik, M., Supatto, W., Beaurepaire, E. and Bally-Cuif, L. (2015). Large-scale live imaging of adult neural stem cells in their endogenous niche. Development *142*, 3592–3600.

[Ellies et al., 2000] Ellies, D. L., Church, V., Francis-West, P. and Lumsden, A. (2000). The WNT antagonist cSFRP2 modulates programmed cell death in

the developing hindbrain. Development *127*, 5285–95.

[Fischer et al., 2014] 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. Exp. Eye Res. *123*, 115–120.

[Fleming et al., 2008] Fleming, H. E., Janzen, V., Lo Celso, C., Guo, J., Leahy, K. M., Kronenberg, H. M. and Scadden, D. T. (2008). Wnt Signaling in the Niche Enforces Hematopoietic Stem Cell Quiescence and Is Necessary to Preserve Self-Renewal In Vivo. Cell Stem Cell *2*, 274–283.

[Fujii, 2000] Fujii, R. (2000). The regulation of motile activity in fish chromatophores. Pigment Cell Res. *13*, 300–319.

[Fukamachi et al., 2004] 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*, 1519–1527.

[Grabher et al., 2003] Grabher, C., Henrich, T., Sasado, T., Arenz, A., Wittbrodt, J. and Furutani-Seiki, M. (2003). Transposon-mediated enhancer trapping in medaka. Gene *322*, 57–66.

[Grotewold and Rüther, 2002] Grotewold, L. and Rüther, U. (2002). The Wnt antagonist Dickkopf-1 is regulated by Bmp signaling and c-Jun and modulates programmed cell death. EMBO J. *21*, 966–975.

[Gutierrez-Triana et al., 2018] 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.

[Habib et al., 2013] Habib, S. J., Chen, B.-c., Tsai, F.-c., Anastassiadis, K., Meyer, T., Betzig, E. and Nusse, R. (2013). Asymmetric Stem Cell Division in Vitro. Science (80-.). *1424*, 1445–1448.

[Hammouda et al., 2019] Hammouda, O. T., Böttger, F., Wittbrodt, J. and Thumberger, T. (2019). Swift Large-scale Examination of Directed Genome Editing. PLoS One *14*, 1–11.

[Harvey et al., 2008] Harvey, C. D., Ehrhardt, A. G., Cellurale, C., Zhong, H., Yasuda, R., Davis, R. J. and Svoboda, K. (2008). A genetically encoded fluorescent sensor of ERK activity. Proc Natl Acad Sci U S A *105*, 19264–19269.

[Heermann et al., 2015] 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.

[Heppert et al., 2016] Heppert, J. K., Dickinson, D. J., Pani, A. M., Higgins, C. D., Steward, A., Ahringer, J., Kuhn, J. R. and Goldstein, B. (2016). Comparative assessment of fluorescent proteins for in vivo imaging in an animal model system. Mol. Biol. Cell *27*, 3385–3394.

[Hirabayashi et al., 2004] Hirabayashi, Y., Itoh, Y., Tabata, H., Nakajima, K., Akiyama, T., Masuyama, N. and Gotoh, Y. (2004). The Wnt/β-catenin pathway directs neuronal differentiation of cortical neural precursor cells. Development *131*, 2791–2801.

[Hirota et al., 2008] Hirota, T., Lewis, W. G., Liu, A. C., Lee, J. W., Schultz, P. G. and Kay, S. A. (2008). A chemical biology approach reveals period shortening of the mammalian circadian clock by specific inhibition of GSK-3 . Proc. Natl. Acad. Sci. *105*, 20746–20751.

[Höckendorf et al., 2012] Höckendorf, B., Thumberger, T. and Wittbrodt, J. (2012). Quantitative analysis of embryogenesis: a perspective for light sheet microscopy. Dev. Cell *23*, 1111–20.

[Huisken et al., 2004] Huisken, J., Swoger, J., Del Bene, F., Wittbrodt, J. and Stelzer, E. H. K. (2004). Optical sectioning deep inside live embryos by selective plane illumination microscopy. Science *305*, 1007–9.

[Hunter, 2007] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Comput. Sci. Eng. *9*, 90–95.

[Ijiri, 2003] Ijiri, K. (2003). Life-Cycle Experiments of Medaka Fish Aboard the International Space Station. In Dev. Biol. Res. Sp. vol. 9, of Advances in Space Biology and Medicine pp. 201–216. Elsevier.

[Israsena et al., 2004] Israsena, N., Hu, M., Fu, W., Kan, L. and Kessler, J. A. (2004). The presence of FGF2 signaling determines whether β-catenin exerts effects on proliferation or neuronal differentiation of neural stem cells. Dev. Biol. *268*, 220–231.

[Iwai-Takekoshi et al., 2018] Iwai-Takekoshi, L., Balasubramanian, R., Sitko, A., Khan, R., Weinreb, S., Robinson, K. and Mason, C. (2018). Activation of Wnt signaling reduces ipsilaterally-projecting retinal ganglion cells in pigmented retina. Development *145*, dev.163212.

[Iwamatsu, 2004] Iwamatsu, T. (2004). Stages of normal development in the medaka Oryzias latipes. Mech. Dev. *121*, 605–618.

[Jho et al., 2002] Jho, E.-h., Zhang, T., Domon, C., Joo, C.-k., Freund, J.-N. and Costantini, F. (2002). Wnt/β -Catenin/Tcf Signaling Induces the Transcription of Axin2 , a Negative Regulator of the Signaling Pathway. Mol. Cell. Biol. *22*, 1172–1183.

[Johns, 1977] Johns, P. R. (1977). Growth of the adult goldfish eye. III. Source

of the new retinal cells. J. Comp. Neurol. *176*, 343–57.

[Kamiyama et al., 2016] Kamiyama, D., Sekine, S., Barsi-Rhyne, B., Hu, J., Chen, B., Gilbert, L. A., Ishikawa, H., Leonetti, M. D., Marshall, W. F., Weissman, J. S. and Huang, B. (2016). Versatile protein tagging in cells with split fluorescent protein. Nat. Commun. *7*, 11046.

[Kane and Kimmel, 1993] Kane, D. A. and Kimmel, C. B. (1993). The zebrafish midblastula transition. Development *119*, 447–56.

[Karlsson et al., 2001] Karlsson, J., von Hofsten, J. and Olsson, P.-E. (2001). Generating Transparent Zebrafish: A Refined Method to Improve Detection of Gene Expression During Embryonic Development. Mar. Biotechnol. *3*, 0522–0527.

[Kasai et al., 2005] Kasai, M., Satoh, K. and Akiyama, T. (2005). Wnt signaling regulates the sequential onset of neurogenesis and gliogenesis via induction of BMPs. Genes to Cells *10*, 777–783.

[Katoh et al., 2016] Katoh, Y., Terada, M., Nishijima, Y., Takei, R., Nozaki, S., Hamada, H. and Nakayama, K. (2016). Overall Architecture of the Intraflagellar Transport (IFT)-B Complex Containing Cluap1/IFT38 as an Essential Component of the IFT-B Peripheral Subcomplex. J. Biol. Chem. *291*, 10962–10975.

[Kearse et al., 2012] Kearse, M., Moir, R., Wilson, A., Stones-Havas, S., Cheung, M., Sturrock, S., Buxton, S., Cooper, A., Markowitz, S., Duran, C., Thierer, T., Ashton, B., Meintjes, P. and Drummond, A. (2012). Geneious Basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. Bioinformatics *28*, 1647–9.

[Keller et al., 2010] Keller, P. J., Schmidt, A. D., Santella, A., Khairy, K., Bao, Z., Wittbrodt, J. and Stelzer, E. H. K. (2010). Fast, high-contrast imaging of animal development with scanned light sheet-based structured-illumination microscopy. Nat. Methods *7*, 637–642.

[Keller et al., 2008] Keller, P. J., Schmidt, A. D., Wittbrodt, J. and Stelzer, E. H. K. (2008). Reconstruction of Zebrafish Early Embryonic Development by Scanned Light Sheet Microscopy. Science (80-.). *322*, 1065–1069.

[Kimmel et al., 1995] Kimmel, C. B., Ballard, W. W., Kimmel, S. R., Ullmann, B. and Schilling, T. F. (1995). Stages of embryonic development of the zebrafish. Dev. Dyn. an Off. public *203*, 253–310.

[Kimura et al., 2014] Kimura, T., Nagao, Y., Hashimoto, H., Yamamoto-Shiraishi, Y.-i., Yamamoto, S., Yabe, T., Takada, S., Kinoshita, M., Kuroiwa, A. and Naruse, K. (2014). Leucophores are similar to xanthophores in their specification and differentiation processes in medaka. Proc. Natl. Acad. Sci.

U. S. A. *111*, 7343–8.

[Kimura et al., 2017] Kimura, T., Takehana, Y. and Naruse, K. (2017). pnp4a Is the Causal Gene of the Medaka Iridophore Mutant, guanineless. G3 *7*, g3.117.040675.

[Kirchmaier et al., 2013a] Kirchmaier, S., Höckendorf, B., Möller, E. K., Bornhorst, D., Spitz, F. and Wittbrodt, J. (2013a). Efficient site-specific transgenesis and enhancer activity tests in medaka using PhiC31 integrase. Development *140*, 4287–95.

[Kirchmaier et al., 2013b] Kirchmaier, S., Lust, K. and Wittbrodt, J. (2013b). Golden GATEway Cloning - A Combinatorial Approach to Generate Fusion and Recombination Constructs. PLoS One *8*, e76117.

[Kochman et al., 2006] Kochman, L. J., Weber, E. T., Fornal, C. A. and Jacobs, B. L. (2006). Circadian variation in mouse hippocampal cell proliferation. Neurosci. Lett. *406*, 256–259.

[Koster et al., 1997] Koster, R., Stick, R., Loosli, F. and Wittbrodt, J. (1997). Medaka spalt acts as a target gene of hedgehog signaling. Development *124*, 3147–3156.

[Kraeussling et al., 2011] Kraeussling, M., Wagner, T. U. and Schartl, M. (2011). Highly asynchronous and asymmetric cleavage divisions accompany early transcriptional activity in pre-blastula medaka embryos. PLoS One *6*, e21741.

[Kraitchman and Bulte, 2009] Kraitchman, D. L. and Bulte, J. W. M. (2009). In vivo imaging of stem cells and beta cells using direct cell labeling and reporter gene methods. Arterioscler. Thromb. Vasc. Biol. *29*, 1025–1030.

[Kromm et al., 2016] Kromm, D., Thumberger, T. and Wittbrodt, J. (2016). An eye on light-sheet microscopy, vol. 133,. Elsevier Ltd.

[Krzic et al., 2012] Krzic, U., Gunther, S., Saunders, T. E., Streichan, S. J. and Hufnagel, L. (2012). Multiview light-sheet microscope for rapid in toto imaging. Nat. Methods *9*, 730–733.

[Kubala et al., 2010] Kubala, M. H., Kovtun, O., Alexandrov, K. and Collins, B. M. (2010). Structural and thermodynamic analysis of the GFP:GFP-nanobody complex. Protein Sci. *19*, 2389–2401.

[Kubo, 2003] Kubo, F. (2003). Wnt2b controls retinal cell differentiation at the ciliary marginal zone. Development *130*, 587–598.

[Kubo and Nakagawa, 2009] Kubo, F. and Nakagawa, S. (2009). Hairy1 acts as a node downstream of Wnt signaling to maintain retinal stem cell-like progenitor cells in the chick ciliary marginal zone. Development *136*, 1823–33.

[Kubo et al., 2005] Kubo, F., Takeichi, M. and Nakagawa, S. (2005). Wnt2b

inhibits differentiation of retinal progenitor cells in the absence of Notch activity by downregulating the expression of proneural genes. Development *132*, 2759–70.

[Lam et al., 2012] Lam, A. J., St-Pierre, F., Gong, Y., Marshall, J. D., Cranfill, P. J., Baird, M. A., Mckeown, M. R., Wiedenmann, J., Davidson, M. W., Schnitzer, M. J., Tsien, R. Y. and Lin, M. Z. (2012). Improving FRET dynamic range with bright green and red fluorescent proteins. Nat. Methods *9*, 1005–1012.

[Lischik et al., 2019] 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*, 1–19.

[Livesey and Cepko, 2001] Livesey, F. J. and Cepko, C. L. (2001). Vertebrate neural cell-fate determination: lessons from the retina. Nat. Rev. Neurosci. *2*, 109–18.

[Logan and Nusse, 2004] Logan, C. Y. and Nusse, R. (2004). The Wnt signaling pathway in development and disease. Annu. Rev. Cell Dev. Biol. *20*, 781–810.

[Lucas et al., 2008] Lucas, D., Battista, M., Shi, P. A., Isola, L. and Frenette, P. S. (2008). Mobilized Hematopoietic Stem Cell Yield Depends on Species-Specific Circadian Timing. Cell Stem Cell *3*, 364–366.

[Lust et al., 2016] Lust, K., Sinn, R., Saturnino, A. P., Centanin, L. and Wittbrodt, J. (2016). De novo neurogenesis by targeted expression of atoh7 to Müller glia cells. Development *143*, 1874–1883.

[MacDonald et al., 2009] MacDonald, B. T., Tamai, K. and He, X. (2009). Wnt/beta-catenin signaling: components, mechanisms, and diseases. Dev. Cell *17*, 9–26.

[Martin Abadi et al., 2015] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Talwar, K., Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

[Martinez-Morales et al., 2004] Martinez-Morales, J. R., Rodrigo, I. and Bovolenta, P. (2004). Eye development: a view from the retina pigmented epithelium. BioEssays *26*, 766–777.

[Matsu-ura et al., 2018] Matsu-ura, T., Moore, S. R. and Hong, C. I. (2018). WNT Takes Two to Tango: Molecular Links between the Circadian Clock and the Cell Cycle in Adult Stem Cells. J. Biol. Rhythms *33*, 5–14.

[McKinney, 2010] McKinney, W. (2010). Data structures for statistical computing in python. In Proc. 9th Python Sci. Conf. vol. 445, pp. 51–56, Austin, TX.

[Méndez-Ferrer et al., 2008] Méndez-Ferrer, S., Lucas, D., Battista, M. and Frenette, P. S. (2008). Haematopoietic stem cell release is regulated by circadian oscillations. Nature *452*, 442–447.

[Meyers et al., 2012] Meyers, J. R., Hu, L., Moses, A., Kaboli, K., Papandrea, A. and Raymond, P. A. (2012). β-catenin/Wnt signaling controls progenitor fate in the developing and regenerating zebrafish retina. Neural Dev. *7*, 30.

[Möller, 2017] Möller, E. K. (2017). Modulation of the Wnt pathway at single-cell level uncovers diverging functional domains in the ciliary marginal zone of medaka. PhD thesis,.

[Moore and Whitmore, 2014] Moore, H. A. and Whitmore, D. (2014). Circadian rhythmicity and light sensitivity of the zebrafish brain. PLoS One *9*, e86176.

[Munji et al., 2011] Munji, R. N., Choe, Y., Li, G., Siegenthaler, J. A. and Pleasure, S. J. (2011). Wnt signaling regulates neuronal differentiation of cortical intermediate progenitors. J. Neurosci. *31*, 1676–1687.

[Muroyama et al., 2004] Muroyama, Y., Kondoh, H. and Takada, S. (2004). Wnt proteins promote neuronal differentiation in neural stem cell culture. Biochem. Biophys. Res. Commun. *313*, 915–921.

[Nantasenamat et al., 2007] Nantasenamat, C., Isarankura-Na-Ayudhya, C., Tansila, N., Naenna, T. and Prachayasittikul, V. (2007). Prediction of GFP Spectral Properties Using Artificial Neural Network. J Comput Chem *28*, 1275–1289.

[Ohshima et al., 2013] Ohshima, A., Morimura, N., Matsumoto, C., Hiraga, A., Komine, R., Kimura, T., Naruse, K. and Fukamachi, S. (2013). Effects of Body-Color Mutations on Vitality : An Attempt to Establish Easy-to-Breed See-Through Medaka Strains by Outcrossing. G3 *3*, 1577–1585.

[Oliphant, 2006] Oliphant, T. E. (2006). A guide to NumPy, vol. 1,. Trelgol Publishing USA.

[Oller Moreno, 2017] Oller Moreno, S. (2017). condformat: Conditional Formatting in Data Frames.

[Pachenari et al., 2017] Pachenari, N., Kiani, S. and Javan, M. (2017). Inhibition of glycogen synthase kinase 3 increased subventricular zone stem cells

proliferation. Biomed. Pharmacother. *93*, 1074–1082.

[Park et al., 2017] Park, S., Gonzalez, D. G., Guirao, B., Boucher, J. D., Cockburn, K., Marsh, E. D., Mesa, K. R., Brown, S., Rompolas, P., Haberman, A. M., Bellaïche, Y. and Greco, V. (2017). Tissue-scale coordination of cellular behaviour promotes epidermal wound repair in live mice. Nat. Cell Biol. *19*, 155–163.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. J. Mach. Learn. Res. *12*, 2825–2830.

[Perez-Saturnino et al., 2018] Perez-Saturnino, A., Lust, K. and Wittbrodt, J. (2018). Notch signalling patterns retinal composition by regulating atoh7 during post-embryonic growth. Development *145*, dev169698.

[Perron and Harris, 2000] Perron, M. and Harris, W. a. (2000). Retinal stem cells in vertebrates. Bioessays *22*, 685–8.

[Pierce and Kimelman, 1995] Pierce, S. E. and Kimelman, D. (1995). Regulation of Spemann organiser formation by the intracellular Xgsk-3. Development *121*, 755–765.

[Pietzsch et al., 2015] Pietzsch, T., Saalfeld, S., Preibisch, S. and Tomancak, P. (2015). BigDataViewer: visualization and processing for large image data sets. Nat. Methods *12*, 481–483.

[Puigbò et al., 2008] Puigbò, P., Bravo, I. G. and Garcia-Vallvé, S. (2008). E-CAI: A novel server to estimate an expected value of Codon Adaptation Index (eCAI). BMC Bioinformatics *9*, 65.

[Puigbò et al., 2007] Puigbò, P., Guzmán, E., Romeu, A. and Garcia-Vallvé, S. (2007). OPTIMIZER: A web server for optimizing the codon usage of DNA sequences. Nucleic Acids Res. *35*, 126–131.

[R Core Team, 2018] R Core Team (2018). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing Vienna, Austria.

[Readman et al., 2017] Readman, G. D., Owen, S. F., Knowles, T. G. and Murrell, J. C. (2017). Species specific anaesthetics for fish anaesthesia and euthanasia. Sci. Rep. *7*, 7102.

[Reh and Levine, 1998] Reh, T. A. and Levine, E. M. (1998). Multipotential stem cells and progenitors in the vertebrate retina. J. Neurobiol. *36*, 206–220.

[Reinhardt et al., 2015] 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. EMBO J. *34*, 1572–1588.

[Rembold et al., 2006a] Rembold, M., Lahiri, K., Foulkes, N. S. and Wittbrodt, J. (2006a). Transgenesis in fish: efficient selection of transgenic fish by co-injection with a fluorescent reporter construct. Nat. Protoc. *1*, 1133–9.

[Rembold et al., 2006b] Rembold, M., Loosli, F., Adams, R. J. and Wittbrodt, J. (2006b). Individual Cell Migration Serves as the Driving Force for Optic Vesicle Evagination. Science (80-.). *313*, 1130–1134.

[Reya and Clevers, 2005] Reya, T. and Clevers, H. (2005). Wnt signalling in stem cells and cancer. Nature *434*, 843–50.

[Ritsma et al., 2014] Ritsma, L., Ellenbroek, S. I. J., 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*, 362–365.

[Rompolas et al., 2012] Rompolas, P., Deschene, E. R., Zito, G., Gonzalez, D. G., Saotome, I., Haberman, A. M. and Greco, V. (2012). Live imaging of stem cell and progeny behaviour in physiological hair-follicle regeneration. Nature *487*, 496–499.

[Sabapathy et al., 2015] Sabapathy, V., Mentam, J., Jacob, P. M. and Kumar, S. (2015). Noninvasive Optical Imaging and in Vivo Cell Tracking of Indocyanine Green Labeled Human Stem Cells Transplanted at Superficial or In-Depth Tissue of SCID Mice. Stem Cells Int. *0*, 1–8.

[Schindelin et al., 2012] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J. Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P. and Cardona, A. (2012). Fiji: An open-source platform for biological-image analysis. Nat. Methods *9*, 676–682.

[Schulte et al., 2005] Schulte, G., Bryja, V., Rawal, N., Castelo-Branco, G., Sousa, K. M. and Arenas, E. (2005). Purified Wnt-5a increases differentiation of midbrain dopaminergic cells and dishevelled phosphorylation. J. Neurochem. *92*, 1550–1553.

[Seleit et al., 2017] Seleit, A., Krämer, I., Ambrosio, E., Stolper, J. S., Dross, N., Lischik, C. Q. and Centanin, L. (2017). Neural stem cells induce the formation of their physical niche during organogenesis. Elife *6*, e29173.

[Sinn and Wittbrodt, 2013] Sinn, R. and Wittbrodt, J. (2013). An eye on eye development. Mech. Dev. *130*, 347–358.

[Slowikowski, 2017] Slowikowski, K. (2017). ggrepel: Repulsive Text and Label Geoms for 'ggplot2'.

[Spivakov et al., 2014] Spivakov, M., Auer, T. O., Peravali, R., Dunham, I., Dolle, D., Fujiyama, A., Toyoda, A., Aizu, T., Minakuchi, Y., Loosli, F., Naruse, K., Birney, E. and Wittbrodt, J. (2014). Genomic and phenotypic characterization of a wild medaka population: towards the establishment of an isogenic population genetic resource in fish. G3 (Bethesda). *4*, 433–45.

[Stemmer et al., 2015] 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*, e0124633.

[Suh et al., 2007] Suh, H., Consiglio, A., Ray, J., Sawai, T., D'Amour, K. A. and Gage, F. (2007). In Vivo Fate Analysis Reveals the Multipotent and Self-Renewal Capacities of Sox2+ Neural Stem Cells in the Adult Hippocampus. Cell Stem Cell *1*, 515–528.

[Swinburne et al., 2015] Swinburne, I. A., Mosaliganti, K. R., Green, A. A. and Megason, S. G. (2015). Improved Long-Term Imaging of Embryos with Genetically Encoded α-Bungarotoxin. PLoS One *10*, e0134005.

[Taelman et al., 2010] Taelman, V. F., Dobrowolski, R., Plouhinec, J. L., Fuentealba, L. C., Vorwald, P. P., Gumper, I., Sabatini, D. D. and De Robertis, E. M. (2010). Wnt signaling requires sequestration of Glycogen Synthase Kinase 3 inside multivesicular endosomes. Cell *143*, 1136–1148.

[Tata et al., 2013] Tata, P. R., Mou, H., Pardo-Saganta, A., Zhao, R., Prabhu, M., Law, B. M., Vinarsky, V., Cho, J. L., Breton, S., Sahay, A., Medoff, B. D. and Rajagopal, J. (2013). Dedifferentiation of committed epithelial cells into stem cells in vivo. Nature *503*, 218–223.

[Tolar et al., 2005] Tolar, J., Osborn, M., Bell, S., McElmurry, R., Xia, L., Riddle, M., Panoskaltsis-Mortari, A., Jiang, Y., McIvor, R. S., Contag, C. H., Yant, S. R., Kay, M. A., Verfaillie, C. M. and Blazar, B. R. (2005). Real-time in vivo imaging of stem cells following transgenesis by transposition. Mol. Ther. *12*, 42–48.

[Tomer et al., 2012] Tomer, R., Khairy, K., Amat, F. and Keller, P. J. (2012). Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. Nat. Methods *9*, 755–763.

[Toyama, 1916] Toyama, K. (1916). Some examples of Mendelian characters. Nihon Ikushugaku Kaiho *1*, 1–9.

[Vallée and Vallée, 2018] Vallée, A. and Vallée, J.-N. (2018). Warburg effect hypothesis in autism Spectrum disorders. Mol. Brain *11*, 1.

[Voog and Jones, 2010] Voog, J. and Jones, D. L. (2010). Stem Cells and the Niche: A Dynamic Duo. Cell Stem Cell *6*, 103–115.

[Wakamatsu et al., 2001] Wakamatsu, Y., Pristyazhnyuk, S., Kinoshita, M.,

Tanaka, M. and Ozato, K. (2001). The see-through medaka : A fish model that is transparent throughout life. PNAS *98*, 10046–10050.

[Wan et al., 2016] Wan, Y., Almeida, A. D., Rulands, S., Chalour, N., Muresan, L., Wu, Y., Simons, B. D., He, J. and Harris, W. (2016). The ciliary marginal zone of the zebrafish retina: clonal and time-lapse analysis of a continuously growing tissue. Development *143*, 1099–1107.

[Wickham, 2009] Wickham, H. (2009). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

[Wickham et al., 2017] Wickham, H., Hester, J. and Francois, R. (2017). readr: Read Rectangular Text Data.

[Wittbrodt et al., 2002] Wittbrodt, J., Shima, A. and Schartl, M. (2002). Medaka — a Model Organism From the Far East. Nat. Rev. Genet. *3*, 53–64.

[Wolff et al., 2018] Wolff, C., Tinevez, J.-Y., Pietzsch, T., Stamataki, E., Harich, B., Guignard, L., Preibisch, S., Shorte, S., Keller, P. J., Tomancak, P. and Pavlopoulos, A. (2018). Multi-view light-sheet imaging and tracking with the MaMuT software reveals the cell lineage of a direct developing arthropod limb. Elife *7*, 1–31.

[Wu et al., 2018] Wu, R. S., Lam, I. I., Clay, H., Duong, D., Deo, R. C. and Coughlin, S. R. (2018). A Rapid Method for Directed Gene Knockout for Screening in G0 Zebrafish. Dev. Cell *46*, 112–125.e4.

[Xu et al., 2013] Xu, C., Wang, S., Thibault, G. and Ng, D. T. W. (2013). Futile Protein Folding Cycles in the ER Are Terminated by the Unfolded Protein O-Mannosylation Pathway. Science (80-.). *340*, 978–982.

[Yang et al., 2018] Yang, Y., Li, Z., Chen, G., Li, J., Li, H., Yu, M., Zhang, W., Guo, W. and Tian, W. (2018). GSK3β regulates ameloblast differentiation via Wnt and TGF-β pathways. J. Cell. Physiol. *233*, 5322–5333.

[Yost et al., 1996] Yost, C., Torres, M., Miller, J. R., Huang, E., Kimelman, D. and Moon, R. T. (1996). The axis-inducing activity, stability, and subcellular distribution of beta-catenin is regulated in Xenopus embryos by glycogen synthase kinase 3. Genes Dev. *10*, 1443–1454.

[Zacharias et al., 2002] Zacharias, D. A., Violin, J. D. and Newton, A. C. (2002). Partitioning of Lipid-Modified Monomeric CFPs into Membrane Microdomains of Live Cells. Science *296*, 913–6.

[Zeileis and Grothendieck, 2005] Zeileis, A. and Grothendieck, G. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. J. Stat. Softw. *14*, 1–27.

[Zeller et al., 2006] Zeller, R. W., Weldon, D. S., Pellatiro, M. A. and Cone,

A. C. (2006). Optimized green fluorescent protein variants provide improved single cell resolution of transgene expression in ascidian embryos. Dev. Dyn. *235*, 456–467.

Acknowledgments

The following acknowledgments are not necessarily in order of importance, but logically.
First of all, I want to thank **Jochen Wittbrodt**, who has been a great supervisor and helped through the good and challenging times of the PhD. I also greatly appreciate the human aspect he is still able to show in his daily work with his students. It has been an honor and pleasure to work in your lab. I also thank for all the opportunities which were presented to me in the time working with you.

I want to thank the **5th floor** for the great working environment and the fun we had together. I also thank all of you for the great input, personally and scientifically, you have given to me over the course of my time with you.

I thank **Jun.-Prof. Dr. Steffen Lemke**, who has been a great second supervisor and a very clever help at many times, when I got stuck with my approach.

I thank **Dr. Lars Hufnagel** for valuable advice during my TAC meetings and cooperation on SPIM matters.

I thank **Prof. Dr. Thomas Höfer** for valuable advice given during my TAC meetings and being part of my committee.

I thank **Dr. Alexander Aulehla** for being part of my examination committee.

I thank **Eva Lempp** for the initial start of the Wnt project and transferring all the data while being open for discussions and still harboring interest.

I thank **Bay 2** for keeping a good spirit, being running buddies and the excellent collaborative work. **Tanja Kellner**, I thank you for the help with methods and the input on all protocols. **Jakob Gierten**, I am thankful for working together with you, exchanging ideas, giving each other advice and being the voice of reason for experimental planning and writing things. It was

very enjoyable to have somebody so close, who is in similar stages of his PhD. **Alex Cornean**, thank you for working together with me to establish new methods such as reporters and all the fun we had besides work.

I thank **Omar Hammouda** for the fun times we had together and the scientific exchange we had.

I again thank **Omar Hammouda**, **Jakob Gierten** and **Alex Cornean** very much for the critical proof reading of my thesis draft and the very helpful comments. Your criticism pushed this to a whole new level.

I thank **Thomas Thumberger** for being the funny guy you are and for learning a lot from you. Especially I thank you for working together so long during the fish dissection course, it has been a lot of fun and I learned a great deal.

I thank **Tinatini Tavhelidse** for a lot of help during my initial time of being a PhD-student and all the help regarding stainings. Also thank you for sharing your knowledge for the final phase of writing the thesis.

I thank **Leonie Adelmann**, who has been a great help in the experiments to establish *in vivo*-imaging. I am very thankful for having such an excellent, hard-working and independent HiWi.

I thank my students **Sophie-Luise Landua**, **Yasmin Riesinger** and **Philipp Stachel-Braum** for working with me and giving me the opportunity to learn how to teach and hopefully giving me the opportunity to share some of my knowledge and some of my excitement for the topics we have worked on.

I thank **Lucas Schütz** for the relaxed cooperation on any hardware or software issues and problems we encountered. Thank you for being such a thoughtful organizer of our network infrastructure.

I thank my scientific ex-roommates **Linda Manhart** and **Tanja Mederer**, who have been part of my life for the full time of my studies and will be beyond. It was very efficient and insightful to learn with you. I also enjoyed our discussions on the way to our PhDs.

I thank **Dimitri Kromm** for exchange on SPIM matters and for being able to learn a lot about physics, optics and microscopy from him.

I thank **Ali Seleit** and **Isabel Krämer** for cooperation on their manuscript and giving me the chance to contribute.

I thank **Narges Aghaallaei** for cooperation on her manuscript and giving me the chance to contribute so early on in my PhD.

I thank **Frederike Seibold** and **Ute Volbehr** for help and support with administrative stuff. This is a big deal and helps focus on your own scientific projects.

Beate Wittbrodt, **Erik Leist**, **Antonino Saraceno**, **Marzena Majewski**, thank you for taking care of my fish and offering advice when needed.

I thank my dancing partner **Stella Wilhelm** for the time on the dancefloor besides my research, which was helping me to unwind and refill my mental reserves. I also thank **Philipp Oestreicher**, **Jochen Clormann** and **Daniel Bausch** for the fun times we had dancing.

I thank my guys **Robin Kobus**, **Jan Schweikardt**, **Nils Böhmer** and **Ken Schumacher** for being part of my life since a very long time. With you I can always be myself and relax from any troubles that might be going on somewhere else.

I thank my family **Theresia Schäfer**, **Manfred Schäfer** and **Bettina Lischik** for supporting me throughout my educational path from school to university and beyond. Thank you for promoting and encouraging my interests the way you did.

Last but for sure not least, **Miriam Lischik**, I thank you for everything and more. The inscription is true, without you I would be and have nothing. You make me who I am and helped me to develop freely but with your input. I thank you for supporting me, for understanding the troubles I have gone through, offering advice and criticism and being excited with me when things worked the way they should. I look forward to our future adventures my lovely *Weif*. I love you with all my heart.

Declaration

Herewith I declare that I prepared the PhD-thesis "Combining *in vivo* imaging and mechanistic approaches to investigate Wnt regulation of retinal stem cells" on my own and with no other sources and aids than quoted.

Heidelberg, 2019

List of Figures

1.1 The retina of medaka is an ideal model to study stem cells *in vivo* 6

1.2 Stimulation of the β-catenin dependent Wnt pathway 9

1.3 Stochastic recombination of single cells results in maintained clones. 13

1.4 Previous experiments with DN-GSK3 reveal clone loss upon Wnt stimulation and a change of differentiation potential in lRPCs . 15

3.1 Most intense fluorescent proteins in medaka 22

3.2 Codon usage table-driven codon adaptation decreased *in vivo* fluorescence intensity of eGFP in medaka 24

3.3 In zebrafish fluorescence intensity of fluorescent proteins varied over time 26

3.4 Prediction of the second fraction of experimental data based on the first fraction of the time course 28

3.5 α-*Bungarotoxin* mRNA microinjection anesthetized medaka embryos long-term and partially reversible 31

3.6 Medaka pigmentation mutants created by CRISPR/Cas9 facilitate *in vivo* imaging 34

3.7 The combination of the established tools facilitated *in vivo* imaging of previously opaque structures 36

3.8 *In vivo* single cell tracking of recombined retinae revealed different cell types 39

3.9 A presumable RSC was tracked but did not divide within 60 h . 41

3.10 Two daughter cell behaviors were observed in the retina 43

3.11 Two independent insertions of RSDNGSK3 were leveraged to investigate dosage effects of Wnt stimulation 45

3.12 Recombination of RSDNGSK3_high with *ccl25b*:*CreERT2* led to results comparable to recombination with *rx2*:*CreERT2* . . . 46

3.13 High Wnt stimulation led to loss of polyclones 47

3.14 Apoptosis caused loss of cells exposed to high Wnt stimulation . 49

3.15 Low Wnt stimulation decreased the proliferative capacity of RSCs and eRPCs 53

4.1 Wnt stimulation resulted in apoptosis, a decrease in proliferative capacity in RSCs and eRPCs and alteration of differentiation potential in lRPCs 63

7.1 The chorion does not interfere with fluorescence intensity of fluorescent proteins in medaka 137
7.2 SceGFP, mRuby2 and OleGFP codon adaptation indices deviate strongly from the main cluster of indices 138
7.3 Clustering of fluorescent protein time courses by machine learning is only detecting the outgroups 139
7.4 The second fraction of experimental data can be predicted based on the first fraction of the time course by ANN 140
7.5 In addition to anesthesia α-Bungarotoxin does not induce cardiac developmental defects 141
7.6 The *spooky* double knockout (dKO) is superior to the *oca2* KO . 141
7.7 All extracted features from GaudíRSG and RSDNGSK3_low in combination with *ccl25b*:*CreERT2* and *tlx*:*CreERT2* 143
7.8 All extracted features from GaudíRSG and RSDNGSK3_low in combination with *ccl25b*:*CreERT2* and *tlx*:*CreERT2*, faceted by time. 144

List of Tables

3.1 sgRNA combinations deployed in CRISPR/Cas9 experiments. . 32
3.2 Overview of GaudíRSG retinae imaged *in vivo*. 38
3.3 Overview of GaudíRSG retinae cell tracks visualized in Fig. 3.10 from *in vivo* imaging data (Data ID6). Distances were approximated by testing three different distances to the point of interest and choosing the smallest distance. All values, except TrackID and panel, are represented in µm. dis.: distance. 42

6.1 Fish lines used in this thesis 71
6.2 Plasmids used in this thesis. 72
6.3 Present primers used in this thesis. 73
6.4 Primers designed and used in this thesis. 74
6.5 sgRNAs used in this thesis. 76
6.6 mRNAs used in this thesis. 77
6.7 Antibodies used in this thesis. 78
6.8 Antibiotics used in this thesis. 78
6.9 Kits used in this thesis. 78
6.10 Enzymes used in this thesis. 79
6.11 Enzyme buffers used in this thesis. 80
6.12 Chemicals and reagents used in this thesis. 81
6.13 Consumables used in this thesis. 83
6.14 Buffers and solutions prepared for this thesis. If not stated otherwise, reagents were dissolved in H_2O. 84
6.15 Equipment used in this thesis. 87
6.16 Computers used in this thesis. 89
6.17 Software and software packages used in this thesis. 89
6.18 Heatshock of single embryos via thermal cycler. 92
6.19 Standard PCR mix. 97
6.20 PCR cycler program for the annealing of oligonucleotides. 99
6.21 DNA ligation. 100
6.22 DNA restriction digest. 103

7.1 Fluorescent proteins tested in the *in vivo* assay 133
7.2 Plates imaged in the Acquifer Imaging machine 135
7.3 Overview of analyzed retinae. 142

7 Appendix

Table 7.1: Fluorescent proteins tested in the *in vivo* assay

name	addgene #	relative to	mutations (amino acids)	equals Balleza et al.	with the exception of (amino acid)
CFP	52109	wtGFP	F64L, S65T, Y66W, S72A, Y145A, N146I, H148D, M153T, V163A, H231L	mCerulean	A206K, H231L
Clover	40259	wtGFP	S30R, Y39N, S65G, Q69A, F99S, N105T, Y145F, M153T, V163A, I171V, T203H	Clover	-
eGFP	45567	wtGFP	F64L, S65T	eGFP	-
eGFPvar	45567	wtGFP	F64L, S65T	eGFP	-
eGFPvar A206K	45567	wtGFP	F64L, S65T, A206K	meGFP	-
Venus	15753	wtGFP	F46L, F64L, S65G, V68L, S72A, M153T, V163A, S175G, T203Y, A206K, H231L	mVenus JBC	H231L
YFP	13016	wtGFP	S65G, V68L, S72A, T203Y, A206K, H231L	meYFP	A206K, H231L

name	addgene #	relative to	mutations (amino acids)	equals Balleza et al.	with the exception of (amino acid)
mCherry	23243	DsRed	MSKGEE DNMA IIKEF. . . V7I, R17H, T21S, H41T, N42Q, V44A, Q66M, V71A, K83L, C117E, F124L, I125R, V127T, T147S, L150M, R153E, V156A, H162K, K163Q, A164R, L174D, V175A, F177V, S179T, I180T, M182K, Y192A, Y194N, D196N, S197I, T217A, H222S, L223T, F224G . . . EGRHSTG GMDE-LYK	mCherry-L	-
mRFP	13032	DsRed	R2A, K5E, N6D, T21S, H41T, N42Q, V44A, V71A, K83L, C117E, F124L, I125R, V127T, L150M, R153E, V156A, H162K, K163M, A164R, L174D, V175A, F177V, S179T, I180T, Y192A, Y194K, V195T, S197I, T217A, H222S, L223T, F224G	mRFP1	-
mRuby2	40260	eqFP611	mRuby2	-	
tagRFP	57823	eqFP578	R32G, K42R, K67R, L79F, I93V, N112D, I115L, N122R, S131P, R155E, H157R, Q159D, Y169H, H171I, S173N, F192V, H193Y, F194Y, M216V, K220R, R231K	tagRFP	5 SKGE
mGFPmut2	103980	mGFPmut2	-		

name	addgene #	relative to	mutations (amino acids)	equals Balleza et al.	with the exception of (amino acid)
mRFP1*	104000	mRFP1*	-		
mScarlet-I	85044	mScarlet-I	-		
mVenNB	103986	mVenNB	-		
OleGFP	NA	wtGFP	F64L, S65T	eGFP	-
SceGFP	NA	wtGFP	S65T, Q80R	eGFP	F64L, Q80R

Table 7.2: Plates imaged in the Acquifer Imaging machine (obj: objective, ch: channel, fil: filter)

plate	total time [h]	time step [min]	figure	obj	ch1/fil1/ ch2/fil2	treatment/ fluorophore
Bungarotoxin plate 1	45.3	20	3.5	2x	BF/BF/-/-	Bungarotoxin mRNA, Etomidat, Tricaine, DMSO, mock injected, wildtype
Bungarotoxin plate 2	64.35	33	3.5	2x	BF/BF/-/-	Bungarotoxin mRNA, Etomidat, Tricaine, DMSO, mock injected, wildtype
Fluorophore plate 1	40.67	20	3.1, 3.2A, 3.4, 7.3, 7.4	4x	470/FITC/ 550/TRITC	tagRFP, mRFP, mCherry, mRuby2, eGFP
Fluorophore plate 2	48	20	3.1, 3.2A, 3.4, 7.3, 7.4	4x	470/FITC /550/TRITC	Clover, eGFPvar, eGFPvarA206K, Venus, YFP, CFP, eGFP, mCherry
Fluorophore plate 3	83.85	43	3.1, 3.2A, 3.4, 7.3, 7.4	4x	470/FITC/ 550/TRITC	eGFP, mCherry, mScarlet, mRFP1*, mGFPmut2, mVenNB
Fluorophore plate 4	72	1440	3.1F, 7.1	4x	470/FITC/ 550/TRITC	tagRFP, mRFP, mCherry, mRuby2, eGFP dechorionated

plate	total time [h]	time step [min]	figure	obj	ch1/fil1/ ch2/fil2	treatment/ fluorophore
Fluorophore plate 5	72	1440	3.1F, 7.1	4x	470/FITC/ 550/TRITC	Clover, eGFPvar, eGFPvarA206K, Venus, YFP, CFP, eGFP, mCherry dechorionated
Fluorophore plate 6	72	1440	3.1F, 7.1	4x	470/FITC/ 550/TRITC	tagRFP, mRFP, mCherry, mRuby2, eGFP
Fluorophore plate 7	72	1440	3.1F, 7.1	4x	470/FITC/ 550/TRITC	Clover, eGFPvar, eGFPvarA206K, Venus, YFP, CFP, eGFP, mCherry
codon adaptation	48.67	20	3.2, 7.2	4x	470/FITC/ 550/TRITC	eGFP, OleGFP, SceGFP

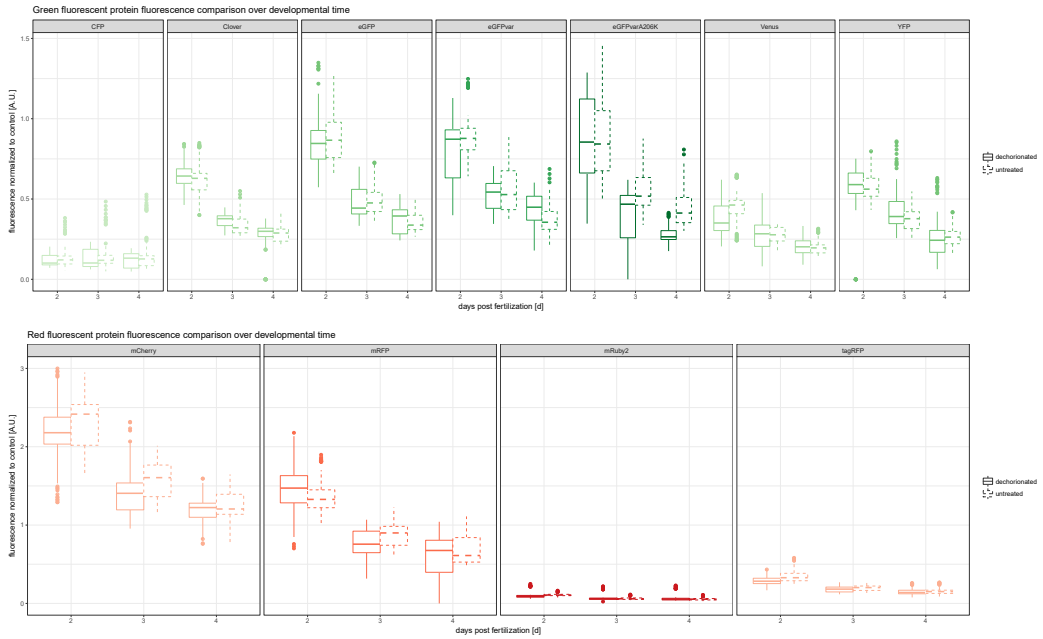

Figure 7.1: The chorion does not interfere with fluorescence intensity of fluorescent proteins in medaka.
Fluorescent protein mRNA was injected into medaka zygotes as described previously. Half of the embryos injected with a fluorescent protein were dechorionated. All embryos were loaded into 96-well plates and imaged at 2, 3 and 4 dpf. The resulting fluorescence intensities were normalized to the injection controls and compared. No effect of the chorion fluorescence intensity of fluorescent proteins in medaka was detected. Figure from [Lischik et al., 2019]

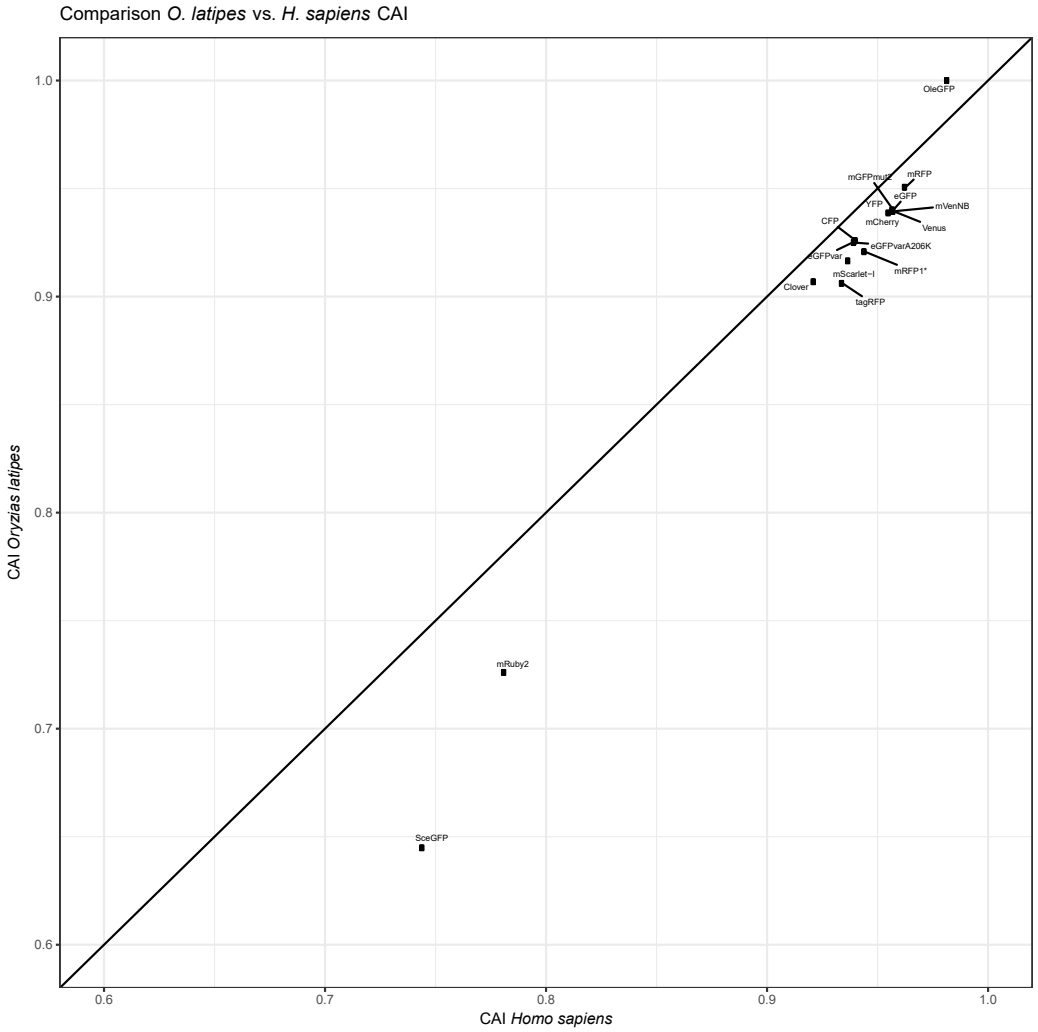

Figure 7.2: SceGFP, mRuby2 and OleGFP codon adaptation indices deviate strongly from the main cluster of indices.
The codon adaptation index for each sequence was determined for medaka and human. Strikingly, most codon adaptation indices cluster indicating a similar codon adaptation for both medaka and human. Deviating are SceGFP and OleGFP, as expected, and mRuby2. Figure from [Lischik et al., 2019]

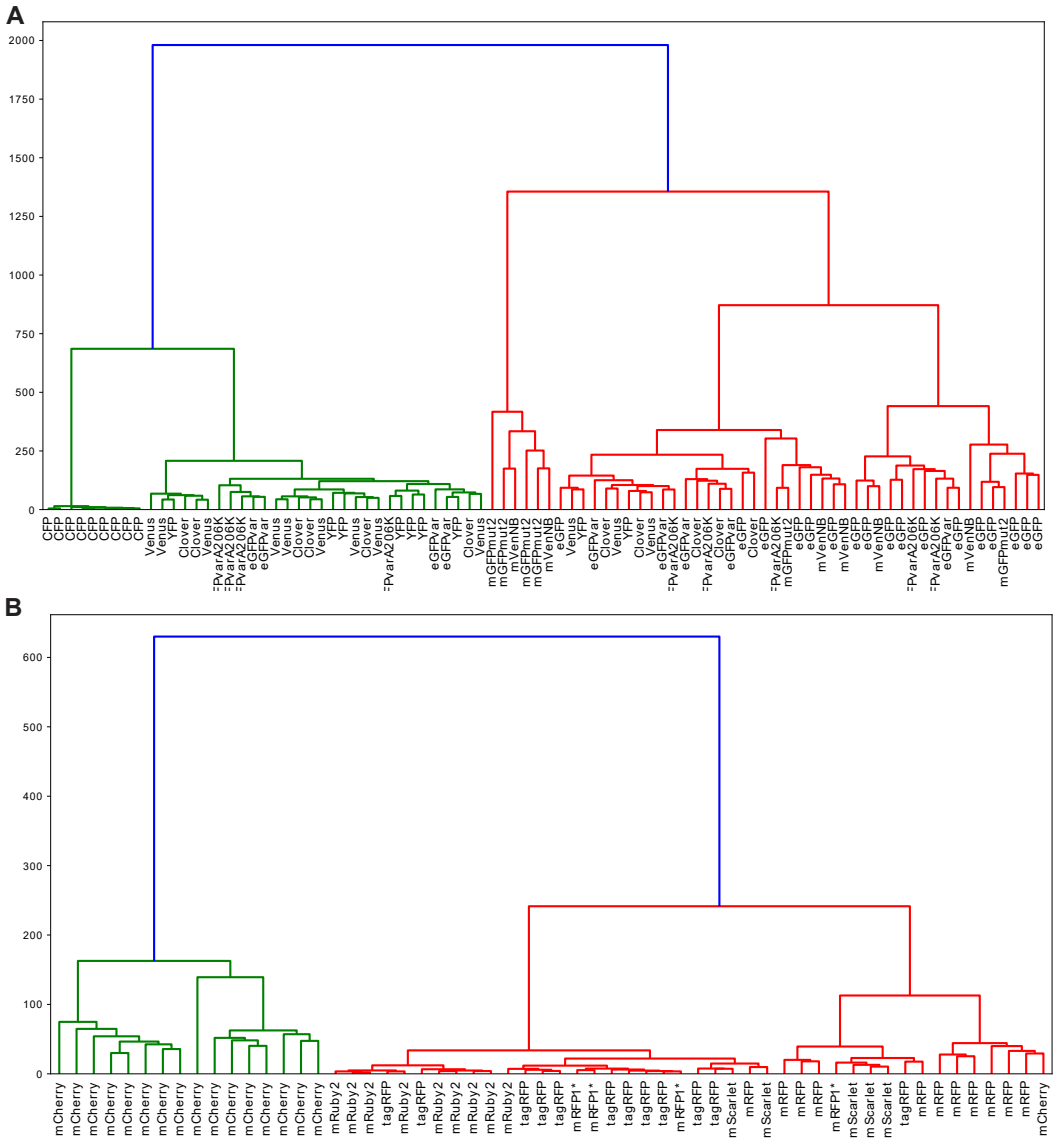

Figure 7.3: Clustering of fluorescent protein time courses by machine learning is only detecting the outgroups.
Full data of time course have been clustered by several machine learning classification algorithms. Shown is the dendrogram for hierarchical linkage clustering. Strikingly, large differences can be detected and clustered accordingly, but fine differences are not easily clusterable. **A** Clustering of data of green fluorescent proteins. **B** Clustering of data of red fluorescent proteins.

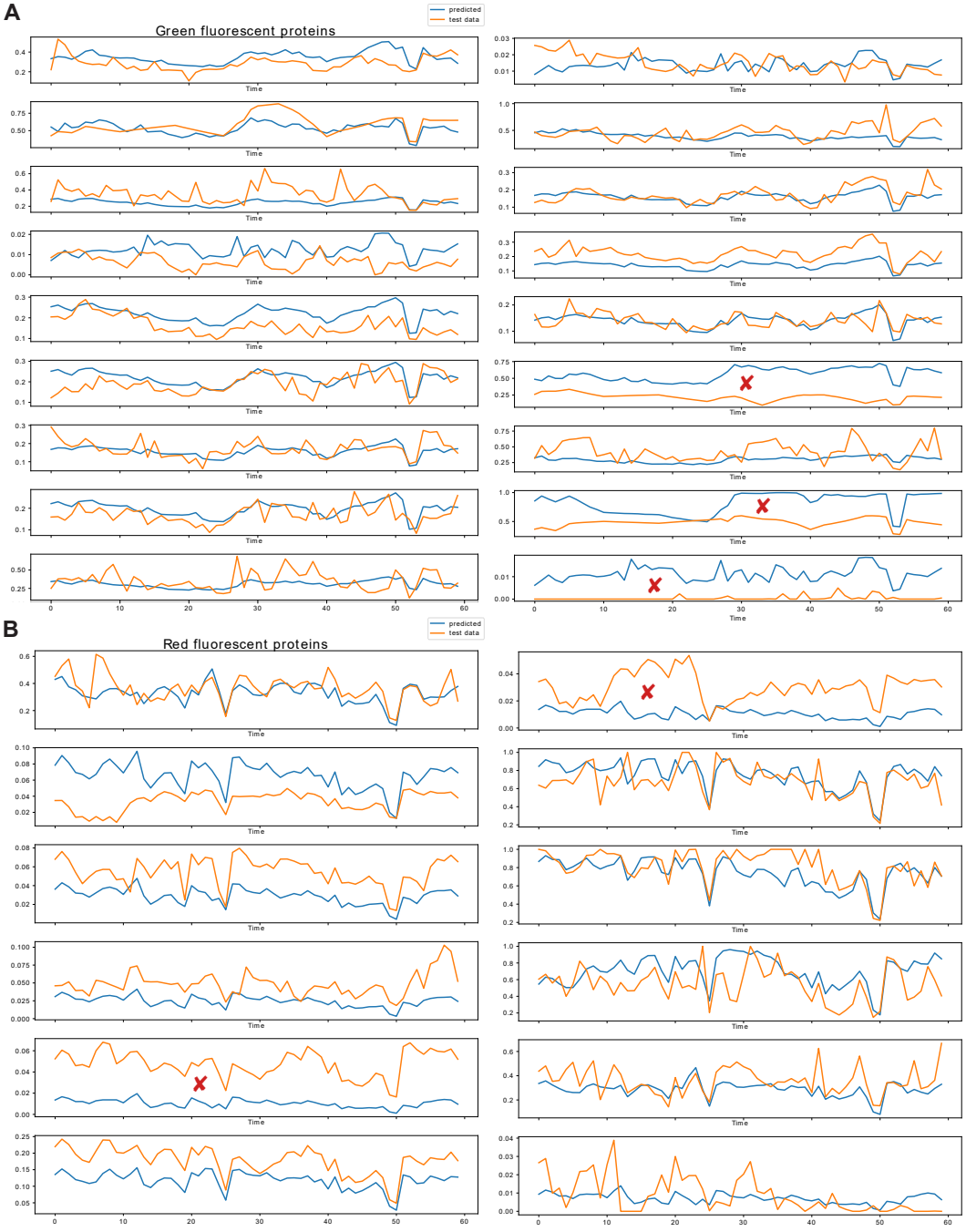

Figure 7.4: The second fraction of experimental data can be predicted based on the first fraction of the time course by ANN.
Data was split into a training set and test set by a standard 80 % to 20 % split. Shown are the results for the test set. An artificial neural network (ANN) was used to predict the second half of the time course experiment, given the first half. **A** shows the results of the green fluorescent protein test set predictions. **B** shows the results of the red fluorescent protein test set predictions. The general trend can be extracted from the predictions except for 3 or 2 test samples marked by red crosses. This indicates that with some tuning the results will be acceptable so the experiment can be shortened to the first half in order to determine the overall fluorescence intensity.

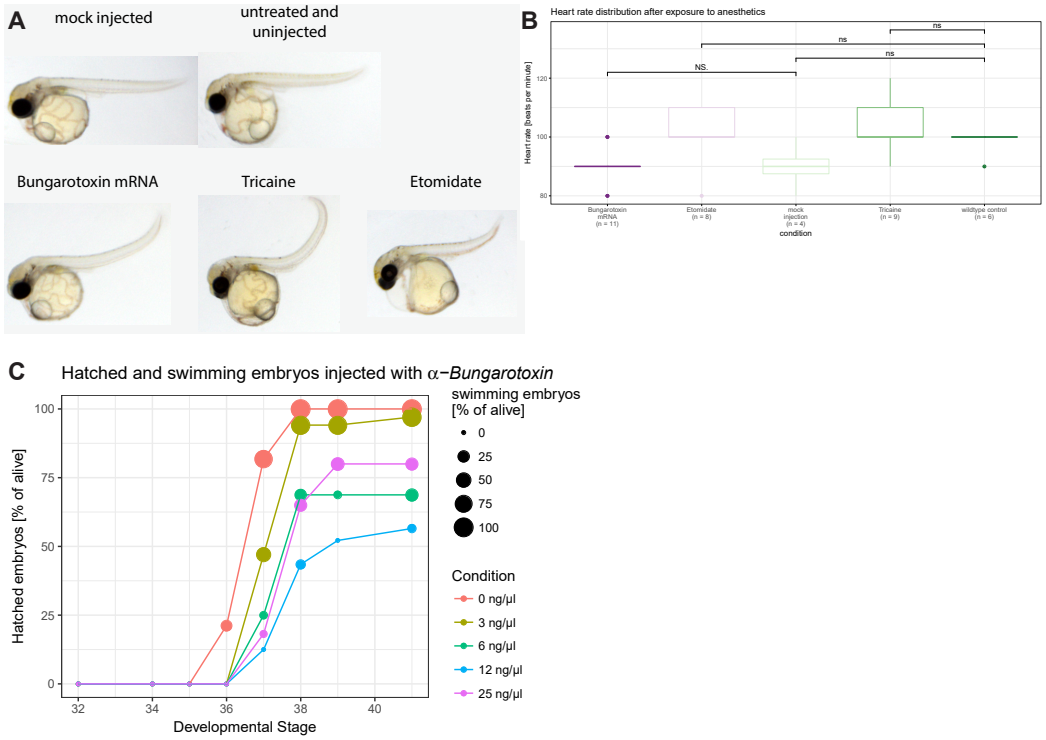

Figure 7.5: In addition to anesthesia α-Bungarotoxin does not induce cardiac developmental defects.
A Embryos treated in Fig. 3.5B were demounted and subjected to video acquisition of 10 s. Exemplary embryos are depicted here. **B** The heart rate was extracted from the previously acquired videos. No significant difference in heart rate was observed between the injection of α-*Bungarotoxin* mRNA and the mock injection. **C** Additional to Fig. 3.5D. In addition to the number of hatched embryos the relative number of swimming embryos was quantified (0 ng/µl: n = 33 fish, 3 ng/µl: n = 34 fish, 6 ng/µl: n = 16 fish, 12 ng/µl: n = 24 fish, 25 ng/µl: n = 22 fish). Both were positively correlated. Asterisks indicate P-values: **** P <= 0.0001, *** P <= 0.001, ** P <= 0.01, * P <= 0.05, ns P > 0.05. Figure from [Lischik et al., 2019].

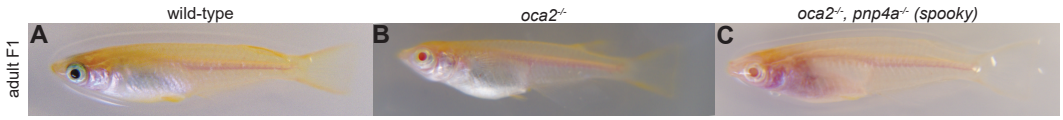

Figure 7.6: The *spooky* dKO is superior to the *oca2* KO.
A Wild-typic F1 fish are heavily pigmented at the peritoneum and the eyes. **B** Compound heterozygous F1 *oca2*$^{-/-}$ adults are devoid of melanin pigmentation. **C** Compound heterozygous F1 *oca2*$^{-/-}$ and *pnp4a*$^{-/-}$ adults (*spooky*) are devoid of melanin and iridophore pigmentation. The inner organs and the retina is more accessible than in *oca2*$^{-/-}$ fish.

Table 7.3: Overview of analyzed retinae.

age [d]	temperature [℃]	Cre-driver	tracing construct	n
7	24	*ccl25b*	GaudíRSG	1
7	24	*ccl25b*	RSDNGSK3_low	3
7	24	*tlx*	GaudíRSG	3
7	24	*tlx*	RSDNGSK3_low	2
14	24	*ccl25b*	GaudíRSG	3
14	24	*ccl25b*	RSDNGSK3_low	9
14	24	*tlx*	GaudíRSG	1
14	24	*tlx*	RSDNGSK3_low	3
14	26	*ccl25b*	GaudíRSG	4
14	26	*ccl25b*	RSDNGSK3_low	4
14	26	*tlx*	GaudíRSG	7
14	26	*tlx*	RSDNGSK3_low	14
21	25	*ccl25b*	GaudíRSG	10
21	25	*ccl25b*	RSDNGSK3_low	2
28	26	*ccl25b*	GaudíRSG	6
28	26	*ccl25b*	RSDNGSK3_low	12
28	26	*tlx*	GaudíRSG	12
28	26	*tlx*	RSDNGSK3_low	6
		Complete		
		ccl25b	GaudíRSG	24
		ccl25b	RSDNGSK3_low	30
		tlx	GaudíRSG	23
		tlx	RSDNGSK3_low	25

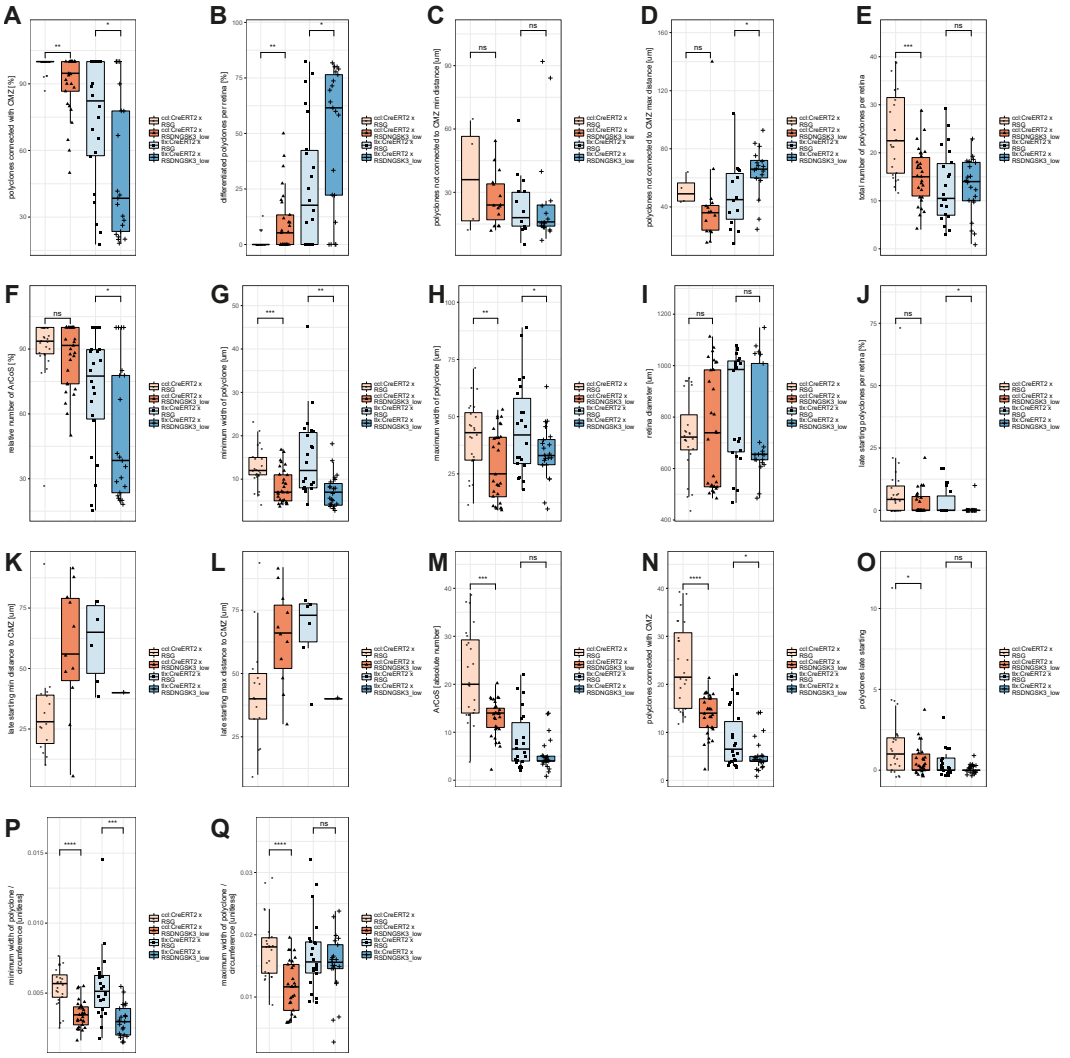

Figure 7.7: All extracted features from GaudíRSG and RSDNGSK3_low in combination with *ccl25b:CreERT2* and *tlx:CreERT2*. For discussion refer to subsection 3. Differences between experiment and control were observed in the percentage of polyclones connected with the CMZ (A), the maximum distance of terminating clones from the CMZ (D), the total number of polyclones (E), the minimum and maximum width of polyclones (G-H) and the minimum and maximum width of polyclones normalized to the circumference (P-Q).
A Percentage of polyclones connected with the CMZ **B** Percentage of polyclones, which are differentiated, per retina. **C** The minimum distance of polyclones, which are not connected to the CMZ, per retina. **D** The maximum distance of polyclones, which are not connected to the CMZ, per retina. **E** The absolute number of polyclones per retina. **F** Percentage of ArCoS per retina. **G** Minimum polyclone width of all clones in a single retina. **H** Maximum polyclone width of all clones in a single retina. **I** Retina diameter. **J** Percentage of late starting polyclones per retina. **K** Minimum distance of start of late starting clones. **L** Maximum distance of start of late starting polyclones. **M** Absolute number of ArCoS per retina. **N** Absolute number of polyclones connected with the CMZ. **O** Absolute number of late starting polyclones. **P** Minimum width of polyclones normalized to circumference. **Q** Maximum width of polyclones normalized to circumference.

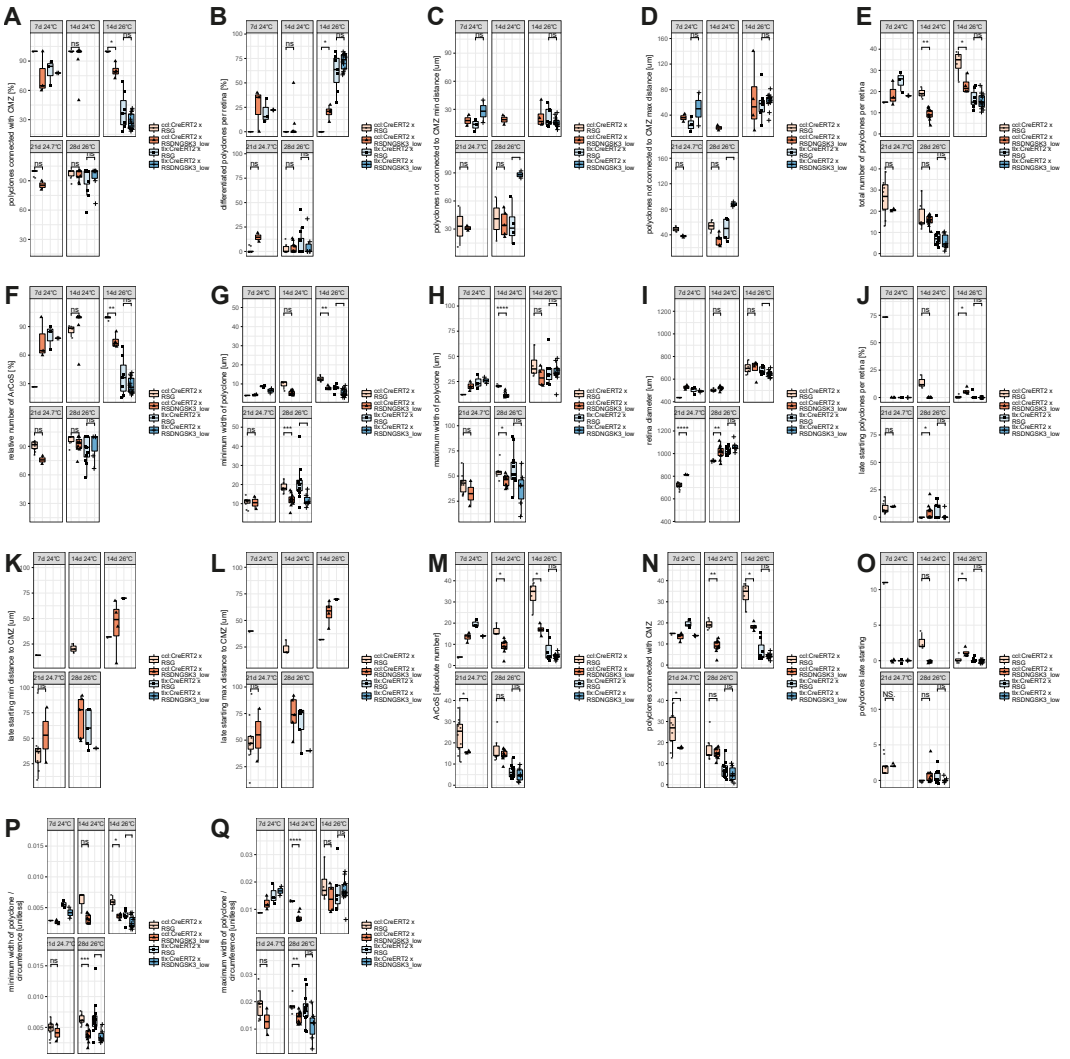

Figure 7.8: All extracted features from GaudíRSG and RSDNGSK3_low in combination with *ccl25b*:*CreERT2* and *tlx*:*CreERT2*, faceted by time. For discussion refer to subsection 3. Differences between experiment and control were observed in the total number of polyclones (E), retina diameter (I).
A Percentage of polyclones connected with the CMZ **B** Percentage of polyclones, which are differentiated, per retina. **C** The minimum distance of polyclones, which are not connected to the CMZ, per retina. **D** The maximum distance of polyclones, which are not connected to the CMZ, per retina. **E** The absolute number of polyclones per retina. **F** Percentage of ArCoS per retina. **G** Minimum polyclone width of all clones in a single retina. **H** Maximum polyclone width of all clones in a single retina. **I** Retina diameter. **J** Percentage of late starting polyclones per retina. **K** Minimum distance of start of late starting clones. **L** Maximum distance of start of late starting polyclones. **M** Absolute number of ArCoS per retina. **N** Absolute number of polyclones connected with the CMZ. **O** Absolute number of late starting polyclones. **P** Minimum width of polyclones normalized to circumference. **Q** Maximum width of polyclones normalized to circumference.

Machine Learning analysis of fluorophore data only green data

data import

```
import numpy as np
import pandas as pd
greendata = pd.read_csv('greenplotdata.csv')
```

Checking data integrity

```
greendata.head()
```

Concatenating dataframes

```
mydata = greendata.iloc[:, 1:]
mydata.head()
```

Fusing well and fluorophore as identifier

```
mydata['ID'] = mydata['well'] + '_' + mydata['fluorophore']
mydata = mydata.iloc[:, 1:]
mydata.head()
```

Creating time series data frame

```
mydata_T = mydata.pivot_table(index = ['ID', 'fluorophore'], columns = 'hpf', values =
    ↪ 'mean').reset_index()
mydata_T.head()
```

Interpolating time series data frame

```
mydata_T.loc[:,2:] = mydata_T.iloc[:,2:].interpolate(method = 'linear', axis = 1)
mydata_T = mydata_T.dropna(axis = 1)
mydata_T.head()
```

Normalizing time series to the maximum per row

```
mydata_T_norm = mydata_T.copy()
mydata_T_norm.iloc[:,2:] = mydata_T_norm.iloc[:,2:].divide(mydata_T_norm.iloc[:,2:].
    ↪ apply(max, axis = 1), axis = 0)
mydata_T_norm.head()
```

How many fluorophores have been tested?

```
total_fluorophores = mydata_T.iloc[:,1].unique()
fluorophore_number = len(total_fluorophores)
fluorophore_number
```

Clustering analysis

Defining X's

```
X = mydata_T.iloc[:,2:]
X_rel = mydata_T_norm.iloc[:,2:]
X.head()
```

Plot X's

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(X.T)
plt.show()
```

```
%matplotlib inline
plt.plot(X_rel.T)
plt.show()
```

Clustering absolute data by k-means clustering

Clustering

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt



kmeans = KMeans(n_clusters = len(total_fluorophores),
                init = 'k-means++', ## preventing the random initialization trap
                max_iter = 300, ## maximum number of iterations
                n_init = 10, ## number the algo will be run with different
                    ↪ initializations
                random_state = 0)
## this will not only fit, but also predict for each point the cluster it belongs to
y_kmeans = kmeans.fit_predict(X)
```

```
y_kmeans
```

“Confusion matrix”

```
cluster_test = pd.concat([mydata_T.iloc[:,1], pd.DataFrame(y_kmeans)], axis = 1)
cluster_test
```

Plotting

```
#%matplotlib inline
#plt.figure(figsize = (25,10))
#dendrogram(cluster_test)
#plt.show()
```

Clustering relative data by k-means clustering

Clustering

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

kmeans = KMeans(n_clusters = len(total_fluorophores),
                init = 'k-means++', ## preventing the random initialization trap
                max_iter = 300, ## maximum number of iterations
                n_init = 10, ## number the algo will be run with different
                    ↪ initializations
                random_state = 0)
## this will not only fit, but also predict for each point the cluster it belongs to
y_kmeans = kmeans.fit_predict(X_rel)
y_kmeans
```

“Confusion matrix”

```
cluster_test_rel = pd.concat([mydata_T.iloc[:,1], pd.DataFrame(y_kmeans)], axis = 1)
```

Plotting

```
#%matplotlib inline
#plt.figure(figsize = (25,10))
#dendrogram(cluster_test)
#plt.show()
```

Comparing absolute and relative k-means clustering

```
total_clusters = pd.concat([cluster_test, cluster_test_rel.iloc[:, 1]], axis = 1)
```

Clustering absolute data by linkage clustering

```
from scipy.cluster import hierarchy
from matplotlib.pyplot import show
from pylab import savefig
import matplotlib

matplotlib.rcParams['lines.linewidth'] = 2
plt.figure(1, figsize=(17, 8.5))
Z = hierarchy.linkage(X, 'ward')
hierarchy.dendrogram(Z, leaf_rotation=90, leaf_font_size=12, labels = mydata_T.iloc
    ↪ [:,1].tolist())
savefig('link_cluster_green.pdf')
savefig('link_cluster_green.png', dpi = 1000)
show()
```

Clustering relative data by linkage clustering

```
from scipy.cluster import hierarchy
from matplotlib.pyplot import show
from pylab import savefig
import matplotlib

matplotlib.rcParams['lines.linewidth'] = 2
plt.figure(1, figsize=(40, 5))
Z = hierarchy.linkage(X_rel, 'ward')
hierarchy.dendrogram(Z, leaf_rotation=90, leaf_font_size=14, labels = mydata_T.iloc
    ↪ [:,1].tolist())
show()
```

Clustering by spectral clustering

```
from sklearn.cluster import SpectralClustering
clustering = SpectralClustering(n_clusters=len(total_fluorophores),
          assign_labels="discretize",
          random_state=0).fit(X)
clustering.labels_
```

“Confusion matrix”

```
cluster_test = pd.concat([mydata_T.iloc[:,1], pd.DataFrame(clustering.labels_)], axis =
    ↪ 1)
cluster_test
```

Classification

Preparing dataframes

```
from sklearn.cross_validation import train_test_split
y = mydata_T.iloc[:,1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
    ↪ = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
X_train_rel, X_test_rel, y_train_rel, y_test_rel = train_test_split(X_rel, y, test_size
    ↪ = 0.2, random_state = 0)
sc_rel = StandardScaler()
X_train_rel = sc.fit_transform(X_train_rel)
X_test = sc.transform(X_test_rel)
```

Random Forest Classifier

Absolute data

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, ## default number, be aware of
    ↪ overfitting to the training set
                                    criterion = 'entropy', ## explanation in prior
                                        ↪ tutorial
                                    random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, ## default number, be aware of
    ↪ overfitting to the training set
                                    criterion = 'entropy', ## explanation in prior
                                        ↪ tutorial
```

```
                                        random_state = 0)
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred_rel = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

K-nearest neighbors

Absolute data

```
from sklearn.neighbors import KNeighborsClassifier
## n_neighbors = 5 is default
classifier = KNeighborsClassifier(n_neighbors = 5,
                                  metric = 'minkowski', p = 2 ## needed for using
                                      ↪ euklidian distance
                                  )
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
from sklearn.neighbors import KNeighborsClassifier
## n_neighbors = 5 is default
classifier = KNeighborsClassifier(n_neighbors = 5,
                                  metric = 'minkowski', p = 2 ## needed for using
                                      ↪ euklidian distance
                                  )
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred = classifier.predict(X_test_rel)

# Making the Confusion Matrix
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

Kernel SVM

Absolute data

```
from sklearn.svm import SVC
## penalty at the end of classification
classifier = SVC(kernel = 'rbf', ## round base function, gaussian
                 random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

```
from sklearn.svm import SVC
## penalty at the end of classification
classifier = SVC(kernel = 'linear', ## round base function, gaussian
                 random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

```
from sklearn.svm import SVC
## penalty at the end of classification
classifier = SVC(kernel = 'poly', ## round base function, gaussian
                 random_state = 0)
classifier.fit(X_train, y_train)
```

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
from sklearn.svm import SVC
## penalty at the end of classification
classifier = SVC(kernel = 'rbf', ## round base function, gaussian
                 random_state = 0)
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

Decision Tree Classification

Absolute data

```
# Fitting classifier to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', ## most basic and common, but
    ↪ not default
                                    random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
# Fitting classifier to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', ## most basic and common, but
    ↪ not default
                                    random_state = 0)
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

Naive Bayes classification

```
# Fitting classifier to the Training set
from sklearn.naive_bayes import GaussianNB
## no arguments, since it is a naive classifier
## naive means it is assumed the variables are independent from each other
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Logistic regression

```
# Fitting Logistic Regression to the Training Set
## import library
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

```
# Making the Confusion Matrix
## Evaluation of the model
## containing correct data and predictions
## class in capitals, funtions in small letters by import
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Dimensionality Reduction

PCA

Absolute data

```
# Applying PCA
from sklearn.decomposition import PCA
pca = PCA(n_components = 9
          )
Y_train_PCA = pca.fit_transform(X.T)
#Y_test_PCA = pca.transform(X_test)
```

```
## look at the accumulated explained variance of the PCA
explained_variance = pca.explained_variance_ratio_
explained_variance
```

```
sum(explained_variance)
```

extracting both PC's - which timepoints most important?

```
pca.components_
```

```
pca.components_.shape
```

k-means clustering on pca

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt



kmeans2 = KMeans(n_clusters = len(total_fluorophores),
```

```
                    init = 'k-means++', ## preventing the random initialization trap
                    max_iter = 300, ## maximum number of iterations
                    n_init = 10, ## number the algo will be run with different
                        ↪ initializations
                    random_state = 0)
## this will not only fit, but also predict for each point the cluster it belongs to
y_kmeans2 = kmeans2.fit_predict(pca.components_.T)
y_kmeans2
```

"Confusion matrix"

```
cluster_test = pd.concat([mydata_T.iloc[:,1], pd.DataFrame(y_kmeans2)], axis = 1)
cluster_test
```

Relative data

```
# Applying PCA
from sklearn.decomposition import PCA
pca = PCA(n_components = 2 ## number of extracted features, which explain the most
    ↪ of the variance. Here none, because this is explaining all the variance. None was
    ↪ substituted by 2 after checking by explained variance.
        )
X_train_rel_new = pca.fit_transform(X_train_rel)
X_test_rel_new = pca.transform(X_test_rel)
## look at the accumulated explained variance of the PCA
explained_variance = pca.explained_variance_ratio_
explained_variance
```

Deep Learning

Preparing dataframes

```
DLdata = mydata_T.iloc[:, 1:]
DLdata_rel = mydata_T_norm.iloc[:, 1:]
```

Getting dimensions of the data

```
datapoints = DLdata.shape[0]
columns = DLdata.shape[1]
```

```
DLdata.head()
```

```
DLdata_rel.head()
```

Encoding absolute data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
DLdata_enc = pd.concat((pd.get_dummies(labelencoder_X_1.fit_transform(DLdata.iloc
    ↪ [:, 0].values), prefix = 'enc'), DLdata.iloc[:, 1:]), axis = 1)
DLdata_enc.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
DLdata_enc = scaler.fit_transform(DLdata_enc)
DLdata_enc[0:5, :]
```

Encoding relative data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_2 = LabelEncoder()
DLdata_rel_enc = pd.concat((pd.get_dummies(labelencoder_X_2.fit_transform(
    ↪ DLdata_rel.iloc[:, 0].values), prefix = 'enc'), DLdata_rel.iloc[:, 1:]), axis = 1)
DLdata_rel_enc.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
DLdata_rel_enc = scaler.fit_transform(DLdata_rel_enc)
DLdata_rel_enc[0:5, :]
```

Classification by ANN - Absolute data

Splitting dataset in X and y and test and train

```
X_DL_class = DLdata_enc[:, fluorophore_number:columns]
y_DL_class = DLdata_enc[:, 0:fluorophore_number]
X_DL_class_train, X_DL_class_test, y_DL_class_train, y_DL_class_test =
    ↪ train_test_split(X_DL_class, y_DL_class, test_size = 0.2)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
optimizer = 'adam'
classifier = Sequential()
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu',
    ↪ input_dim = (columns − fluorophore_number)))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
#classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
#classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = fluorophore_number, kernel_initializer = 'uniform',
    ↪ activation = 'sigmoid'))
classifier.compile(optimizer = optimizer, loss = 'categorical_crossentropy', metrics = ['
    ↪ accuracy'])

classifier.fit(X_DL_class_train, y_DL_class_train, epochs=500, batch_size=25)

y_DL_class_pred = classifier.predict(X_DL_class_test)
```

Checking on test data

```
pd.DataFrame.from_records(y_DL_class_pred.round(decimals= 2))
```

Comparing y_pred to data

```
y_DL_class_pred_max = pd.DataFrame.from_records(y_DL_class_pred)
y_DL_class_pred_maxima = y_DL_class_pred_max.apply(lambda x: max(x), axis =
    ↪  1)
y_DL_class_pred_max = y_DL_class_pred_max.isin(y_DL_class_pred_maxima)
y_DL_class_pred_max = y_DL_class_pred_max.stack()
y_DL_class_pred_max = pd.Series(pd.Categorical(y_DL_class_pred_max[
    ↪ y_DL_class_pred_max!=0].index.get_level_values(1)))
y_DL_class_pred_max = labelencoder_X_1.inverse_transform(y_DL_class_pred_max
    ↪ )
y_DL_class_pred_max = pd.DataFrame(y_DL_class_pred_max)
y_DL_class_pred_max
```

```
y_DL_class_test_dec = pd.DataFrame(y_DL_class_test)
```

```
y_DL_class_test_dec = pd.DataFrame.from_records(y_DL_class_test_dec)
y_DL_class_test_dec_maxima = y_DL_class_test_dec.apply(lambda x: max(x), axis
    ↪  = 1)
y_DL_class_test_dec = y_DL_class_test_dec.isin(y_DL_class_test_dec_maxima)
y_DL_class_test_dec = y_DL_class_test_dec.stack()
y_DL_class_test_dec = pd.Series(pd.Categorical(y_DL_class_test_dec[
    ↪ y_DL_class_test_dec!=0].index.get_level_values(1)))
y_DL_class_test_dec = labelencoder_X_1.inverse_transform(y_DL_class_test_dec)
y_DL_class_test_dec = pd.DataFrame(y_DL_class_test_dec)
y_DL_class_test_dec
```

```
y_DL_class_test_dec.columns = ['test']
y_DL_class_pred_max.columns = ['pred']
y_DL_class_test_dec = y_DL_class_test_dec.reset_index()
y_DL_class_pred_max = y_DL_class_pred_max.reset_index()
y_DL_class_pred_compare = pd.concat((y_DL_class_test_dec,
    ↪ y_DL_class_pred_max), axis = 1)
y_DL_class_pred_compare
```

Prediction accuracy

```
row_ids = y_DL_class_pred_compare[y_DL_class_pred_compare.test ==
    ↪ y_DL_class_pred_compare.pred].index
pred_acc = (len(row_ids)/len(y_DL_class_pred_compare))
pred_acc
```

Classification by ANN - Relative data

Splitting dataset in X and y and test and train

```
X_DL_class_rel = DLdata_rel_enc[:, fluorophore_number:columns]
y_DL_class_rel = DLdata_rel_enc[:, 0:fluorophore_number]
X_DL_class_rel_train, X_DL_class_rel_test, y_DL_class_rel_train,
    ↪ y_DL_class_rel_test = train_test_split(X_DL_class_rel, y_DL_class_rel,
    ↪ test_size = 0.2)
```

```
optimizer = 'adam'
classifier = Sequential()
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu',
    ↪ input_dim = (columns − fluorophore_number)))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
```

```
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = fluorophore_number, kernel_initializer = 'uniform',
    ↪ activation = 'sigmoid'))
classifier.compile(optimizer = optimizer, loss = 'categorical_crossentropy', metrics = ['
    ↪ accuracy'])

classifier.fit(X_DL_class_rel_train, y_DL_class_rel_train, epochs=500, batch_size=25)

y_DL_class_rel_pred = classifier.predict(X_DL_class_rel_test)
```

Checking on test data

```
pd.DataFrame.from_records(y_DL_class_rel_pred)
```

Comparing y_pred to data

```
y_DL_class_rel_pred_max = pd.DataFrame.from_records(y_DL_class_rel_pred)
y_DL_class_rel_pred_maxima = y_DL_class_rel_pred_max.apply(lambda x: max(x
    ↪ ), axis = 1)
y_DL_class_rel_pred_max = y_DL_class_rel_pred_max.isin(
    ↪ y_DL_class_rel_pred_maxima)
y_DL_class_rel_pred_max = y_DL_class_rel_pred_max.stack()
#y_DL_class_rel_pred_max = pd.Series(pd.Categorical(y_DL_class_rel_pred_max[
    ↪ y_DL_class_rel_pred_max!=0].index.get_level_values(1)))
#y_DL_class_rel_pred_max = labelencoder_X_1.inverse_transform(
    ↪ y_DL_class_rel_pred_max)
#y_DL_class_rel_pred_max = pd.DataFrame(y_DL_class_rel_pred_max)
y_DL_class_rel_pred_maxima
```

```
y_DL_class_rel_test_dec = pd.DataFrame(y_DL_class_rel_test)
y_DL_class_rel_test_dec = pd.DataFrame.from_records(y_DL_class_rel_test_dec)
y_DL_class_rel_test_dec_maxima = y_DL_class_rel_test_dec.apply(lambda x:
    ↪ max(x), axis = 1)
y_DL_class_rel_test_dec = y_DL_class_rel_test_dec.isin(
    ↪ y_DL_class_rel_test_dec_maxima)
y_DL_class_rel_test_dec = y_DL_class_rel_test_dec.stack()
y_DL_class_rel_test_dec = pd.Series(pd.Categorical(y_DL_class_rel_test_dec[
    ↪ y_DL_class_rel_test_dec!=0].index.get_level_values(1)))
y_DL_class_rel_test_dec = labelencoder_X_1.inverse_transform(
    ↪ y_DL_class_rel_test_dec)
```

```
y_DL_class_rel_test_dec = pd.DataFrame(y_DL_class_rel_test_dec)
y_DL_class_rel_test_dec
```

```
y_DL_class_rel_test_dec.columns = ['test']
y_DL_class_rel_pred_max.columns = ['pred']
y_DL_class_rel_test_dec = y_DL_class_rel_test_dec.reset_index()
y_DL_class_rel_pred_max = y_DL_class_rel_pred_max.reset_index()
pd.concat((y_DL_class_rel_test_dec, y_DL_class_rel_pred_max), axis = 1)
```

Deep Learning ANN for predicting time series

Preparation of data frames - Absolute data

Main question: How long do we need to record to predict all the following timepoints? Feature scaling is already done. Setting variable parameters:

```
timepoints_to_predict = 60
```

```
DLdata_time = mydata_T.iloc[:, 1:]
DLdata_rel_time = mydata_T_norm.iloc[:, 1:]
```

```
DLdata_time.head()
```

```
DLdata_rel_time.head()
```

Encoding absolute data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_3 = LabelEncoder()
DLdata_enc_time = pd.concat((pd.get_dummies(labelencoder_X_3.fit_transform(
    ↪ DLdata_time.iloc[:, 0].values), prefix = 'enc'), DLdata_time.iloc[:, 1:]), axis = 1)
DLdata_enc_time.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler2 = MinMaxScaler()
DLdata_enc_time = scaler.fit_transform(DLdata_enc)
```

Encoding relative data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_4 = LabelEncoder()
DLdata_rel_enc_time = pd.concat((pd.get_dummies(labelencoder_X_4.fit_transform(
    ↪ DLdata_rel_time.iloc[:, 0].values), prefix = 'enc'), DLdata_rel_time.iloc[:, 1:]),
    ↪ axis = 1)
DLdata_rel_enc_time.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler3 = MinMaxScaler()
DLdata_rel_enc_time = scaler.fit_transform(DLdata_rel_enc_time)
```

Timecourse prediction by ANN - Absolute data

Splitting dataset in X and y and test and train

```
X_DL_class_time = DLdata_enc_time[:, 1:(columns − timepoints_to_predict)]
y_DL_class_time = DLdata_enc_time[:, (columns − timepoints_to_predict):columns]
X_DL_class_train_time, X_DL_class_test_time, y_DL_class_train_time,
    ↪ y_DL_class_test_time = train_test_split(X_DL_class_time,
    ↪ y_DL_class_time, test_size = 0.2)
```

```
(columns − timepoints_to_predict)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout

def create_model():
    optimizer = 'rmsprop'
    time_predictor = Sequential()
    time_predictor.add(Dense(units = 60, kernel_initializer = 'uniform', activation = '
        ↪ sigmoid', input_dim = (columns − timepoints_to_predict − 1)))
    time_predictor.add(Dense(units = 150, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dense(units = 140, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
```

```
    time_predictor.add(Dense(units = 120, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = 110, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dense(units = 70, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = 120, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = timepoints_to_predict, kernel_initializer = '
        ↪ uniform', activation = 'sigmoid'))
    time_predictor.compile(optimizer = optimizer, loss = 'mse', metrics = ['accuracy'])

    return time_predictor

model = create_model()
model.fit(X_DL_class_train_time, y_DL_class_train_time, epochs=2500, batch_size
    ↪ =25)

y_DL_class_pred_time = model.predict(X_DL_class_test_time)
```

Checking on test data

```
pd.DataFrame.from_records(y_DL_class_pred_time.round(decimals= 2))
```

```
len(y_DL_class_pred_time)
```

Comparing y_pred to data graphically

```
fig, axes = plt.subplots(len(y_DL_class_pred_time), 1, sharex=True, figsize=(10,25))
fig.suptitle('Green fluorescent proteins', fontsize = 20)
for i in range(0, len(y_DL_class_pred_time)):
    axes[i].plot(y_DL_class_pred_time[i, :])
    axes[i].plot(y_DL_class_test_time[i, :])
    axes[i].set_xlabel('Time', fontsize = 10)
    #axes[i].set_ylabel('Fluorescence intensity', fontsize = 10)
fig.legend(('predicted', 'test data'), fontsize = 10)
plt.tight_layout()
plt.subplots_adjust(top = 0.97)
savefig('time_pred_green.pdf')
```

```
savefig('time_pred_green.png', dpi = 100)
plt.show
```

Machine Learning analysis of fluorophore data only red data

data import

```
import numpy as np
import pandas as pd
reddata = pd.read_csv('redplotdata.csv')
```

Checking data integrity

```
reddata.head()
```

Concatenating dataframes

```
mydata = reddata.iloc[:, 1:]
mydata.head()
```

Fusing well and fluorophore as identifier

```
mydata['ID'] = mydata['well'] + '_' + mydata['fluorophore']
mydata = mydata.iloc[:, 1:]
mydata.head()
```

Creating time series data frame

```
mydata_T = mydata.pivot_table(index = ['ID', 'fluorophore'], columns = 'hpf', values =
    ↪ 'mean').reset_index()
mydata_T.head()
```

Interpolating time series data frame

```
mydata_T.loc[:,2:] = mydata_T.iloc[:,2:].interpolate(method = 'linear', axis = 1)
mydata_T = mydata_T.dropna(axis = 1)
mydata_T.head()
```

Normalizing time series to the maximum per row

```
mydata_T_norm = mydata_T.copy()
mydata_T_norm.iloc[:,2:] = mydata_T_norm.iloc[:,2:].divide(mydata_T_norm.iloc[:,2:].
    ↪ apply(max, axis = 1), axis = 0)
mydata_T_norm.head()
```

How many fluorophores have been tested?

```
total_fluorophores = mydata_T.iloc[:,1].unique()
fluorophore_number = len(total_fluorophores)
fluorophore_number
```

Clustering analysis

Defining X's

```
X = mydata_T.iloc[:,2:]
X_rel = mydata_T_norm.iloc[:,2:]
X.head()
```

Plot X's

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(X.T)
plt.show()
```

```
%matplotlib inline
plt.plot(X_rel.T)
plt.show()
```

Clustering absolute data by k-means clustering

Clustering

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt



kmeans = KMeans(n_clusters = len(total_fluorophores),
                        init = 'k−means++', ## preventing the random initialization trap
```

```
                    max_iter = 300, ## maximum number of iterations
                    n_init = 10, ## number the algo will be run with different
                        ↪ initializations
                    random_state = 0)
## this will not only fit, but also predict for each point the cluster it belongs to
y_kmeans = kmeans.fit_predict(X)
y_kmeans
```

“Confusion matrix”

```
cluster_test = pd.concat([mydata_T.iloc[:,1], pd.DataFrame(y_kmeans)], axis = 1)
cluster_test
```

Plotting

```
#%matplotlib inline
#plt.figure(figsize = (25,10))
#dendrogram(cluster_test)
#plt.show()
```

Clustering relative data by k-means clustering

Clustering

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

kmeans = KMeans(n_clusters = len(total_fluorophores),
                    init = 'k−means++', ## preventing the random initialization trap
                    max_iter = 300, ## maximum number of iterations
                    n_init = 10, ## number the algo will be run with different
                        ↪ initializations
                    random_state = 0)
## this will not only fit, but also predict for each point the cluster it belongs to
y_kmeans = kmeans.fit_predict(X_rel)
y_kmeans
```

“Confusion matrix”

```
cluster_test_rel = pd.concat([mydata_T.iloc[:,1], pd.DataFrame(y_kmeans)], axis = 1)
```

Plotting

```
#%matplotlib inline
```

```
#plt.figure(figsize = (25,10))
#dendrogram(cluster_test)
#plt.show()
```

Comparing absolute and relative k-means clustering

```
total_clusters = pd.concat([cluster_test, cluster_test_rel.iloc[:, 1]], axis = 1)
```

Clustering absolute data by linkage clustering

```
from scipy.cluster import hierarchy
from matplotlib.pyplot import show
from pylab import savefig
import matplotlib

matplotlib.rcParams['lines.linewidth'] = 2
plt.figure(1, figsize=(17, 8))
Z = hierarchy.linkage(X, 'ward')
hierarchy.dendrogram(Z, leaf_rotation=90, leaf_font_size=12, labels = mydata_T.iloc
    ↪ [:,1].tolist())
savefig('link_cluster_red.pdf')
savefig('link_cluster_red.png', dpi = 1000)
show()
```

Clustering relative data by linkage clustering

```
from scipy.cluster import hierarchy
from matplotlib.pyplot import show
from pylab import savefig
import matplotlib

matplotlib.rcParams['lines.linewidth'] = 2
plt.figure(1, figsize=(40, 5))
Z = hierarchy.linkage(X_rel, 'ward')
hierarchy.dendrogram(Z, leaf_rotation=90, leaf_font_size=14, labels = mydata_T.iloc
    ↪ [:,1].tolist())
show()
```

Classification

Preparing dataframes

```
from sklearn.cross_validation import train_test_split
y = mydata_T.iloc[:,1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
    ↪ = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
X_train_rel, X_test_rel, y_train_rel, y_test_rel = train_test_split(X_rel, y, test_size
    ↪ = 0.2, random_state = 0)
sc_rel = StandardScaler()
X_train_rel = sc.fit_transform(X_train_rel)
X_test = sc.transform(X_test_rel)
```

Random Forest Classifier

Absolute data

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, ## default number, be aware of
    ↪ overfitting to the training set
                                    criterion = 'entropy', ## explanation in prior
                                        ↪ tutorial
                                    random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, ## default number, be aware of
    ↪ overfitting to the training set
                                    criterion = 'entropy', ## explanation in prior
                                        ↪ tutorial
                                    random_state = 0)
classifier.fit(X_train_rel, y_train_rel)
```

```
# Predicting the Test set results
y_pred_rel = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

K-nearest neighbors

Absolute data

```
from sklearn.neighbors import KNeighborsClassifier
## n_neighbors = 5 is default
classifier = KNeighborsClassifier(n_neighbors = 5,
                                  metric = 'minkowski', p = 2 ## needed for using
                                      ↪ euklidian distance
                                  )
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
from sklearn.neighbors import KNeighborsClassifier
## n_neighbors = 5 is default
classifier = KNeighborsClassifier(n_neighbors = 5,
                                  metric = 'minkowski', p = 2 ## needed for using
                                      ↪ euklidian distance
                                  )
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

Kernel SVM

Absolute data

```
from sklearn.svm import SVC
## penalty at the end of classification
classifier = SVC(kernel = 'rbf', ## round base function, gaussian
                 random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
from sklearn.svm import SVC
## penalty at the end of classification
classifier = SVC(kernel = 'rbf', ## round base function, gaussian
                 random_state = 0)
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

Decision Tree Classification

Absolute data

```
# Fitting classifier to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', ## most basic and common, but
    ↪ not default
```

```
                                        random_state = 0)
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Relative data

```
# Fitting classifier to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', ## most basic and common, but
    ↪ not default
                                        random_state = 0)
classifier.fit(X_train_rel, y_train_rel)

# Predicting the Test set results
y_pred = classifier.predict(X_test_rel)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test_rel, y_pred_rel)
cm
```

Dimensionality Reduction

PCA

Absolute data

```
# Applying PCA
from sklearn.decomposition import PCA
pca = PCA(n_components = 2 ## number of extracted features, which explain the most
    ↪ of the variance. Here none, because this is explaining all the variance. None was
    ↪ substituted by 2 after checking by explained variance.
        )
Y_train_PCA = pca.fit_transform(X_train)
Y_test_PCA = pca.transform(X_test)
```

```
## look at the accumulated explained variance of the PCA
explained_variance = pca.explained_variance_ratio_
explained_variance
```

Relative data

```
# Applying PCA
from sklearn.decomposition import PCA
pca = PCA(n_components = 2 ## number of extracted features, which explain the most
    ↪ of the variance. Here none, because this is explaining all the variance. None was
    ↪ substituted by 2 after checking by explained variance.
        )
X_train_rel_new = pca.fit_transform(X_train_rel)
X_test_rel_new = pca.transform(X_test_rel)
## look at the accumulated explained variance of the PCA
explained_variance = pca.explained_variance_ratio_
explained_variance
```

Deep Learning

Preparing dataframes

```
DLdata = mydata_T.iloc[:, 1:]
DLdata_rel = mydata_T_norm.iloc[:, 1:]
```

Getting dimensions of the data

```
datapoints = DLdata.shape[0]
columns = DLdata.shape[1]
```

```
DLdata.head()
```

```
DLdata_rel.head()
```

Encoding absolute data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
DLdata_enc = pd.concat((pd.get_dummies(labelencoder_X_1.fit_transform(DLdata.iloc
    ↪ [:, 0].values), prefix = 'enc'), DLdata.iloc[:, 1:]), axis = 1)
DLdata_enc.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
DLdata_enc = scaler.fit_transform(DLdata_enc)
DLdata_enc[0:5, :]
```

Encoding relative data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_2 = LabelEncoder()
DLdata_rel_enc = pd.concat((pd.get_dummies(labelencoder_X_2.fit_transform(
    ↪ DLdata_rel.iloc[:, 0].values), prefix = 'enc'), DLdata_rel.iloc[:, 1:]), axis = 1)
DLdata_rel_enc.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
DLdata_rel_enc = scaler.fit_transform(DLdata_rel_enc)
DLdata_rel_enc[0:5, :]
```

Classification by ANN - Absolute data

Splitting dataset in X and y and test and train

```
X_DL_class = DLdata_enc[:, fluorophore_number:columns]
y_DL_class = DLdata_enc[:, 0:fluorophore_number]
X_DL_class_train, X_DL_class_test, y_DL_class_train, y_DL_class_test =
    ↪ train_test_split(X_DL_class, y_DL_class, test_size = 0.2)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
optimizer = 'adam'
classifier = Sequential()
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu',
    ↪ input_dim = (columns − fluorophore_number)))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
#classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
```

```
#classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = fluorophore_number, kernel_initializer = 'uniform',
    ↪ activation = 'sigmoid'))
classifier.compile(optimizer = optimizer, loss = 'categorical_crossentropy', metrics = ['
    ↪ accuracy'])

classifier.fit(X_DL_class_train, y_DL_class_train, epochs=300, batch_size=50)

y_DL_class_pred = classifier.predict(X_DL_class_test)
```

Checking on test data

```
pd.DataFrame.from_records(y_DL_class_pred.round(decimals= 2))
```

Comparing y_pred to data

```
y_DL_class_pred_max = pd.DataFrame.from_records(y_DL_class_pred)
y_DL_class_pred_maxima = y_DL_class_pred_max.apply(lambda x: max(x), axis =
    ↪  1)
y_DL_class_pred_max = y_DL_class_pred_max.isin(y_DL_class_pred_maxima)
y_DL_class_pred_max = y_DL_class_pred_max.stack()
y_DL_class_pred_max = pd.Series(pd.Categorical(y_DL_class_pred_max[
    ↪ y_DL_class_pred_max!=0].index.get_level_values(1)))
y_DL_class_pred_max = labelencoder_X_1.inverse_transform(y_DL_class_pred_max
    ↪ )
y_DL_class_pred_max = pd.DataFrame(y_DL_class_pred_max)
y_DL_class_pred_max
```

```
y_DL_class_test_dec = pd.DataFrame(y_DL_class_test)
y_DL_class_test_dec = pd.DataFrame.from_records(y_DL_class_test_dec)
y_DL_class_test_dec_maxima = y_DL_class_test_dec.apply(lambda x: max(x), axis
    ↪  = 1)
y_DL_class_test_dec = y_DL_class_test_dec.isin(y_DL_class_test_dec_maxima)
y_DL_class_test_dec = y_DL_class_test_dec.stack()
y_DL_class_test_dec = pd.Series(pd.Categorical(y_DL_class_test_dec[
    ↪ y_DL_class_test_dec!=0].index.get_level_values(1)))
y_DL_class_test_dec = labelencoder_X_1.inverse_transform(y_DL_class_test_dec)
y_DL_class_test_dec = pd.DataFrame(y_DL_class_test_dec)
y_DL_class_test_dec
```

```
y_DL_class_test_dec.columns = ['test']
y_DL_class_pred_max.columns = ['pred']
y_DL_class_test_dec = y_DL_class_test_dec.reset_index()
y_DL_class_pred_max = y_DL_class_pred_max.reset_index()
y_DL_class_pred_compare = pd.concat((y_DL_class_test_dec,
    ↪ y_DL_class_pred_max), axis = 1)
y_DL_class_pred_compare
```

Prediction accuracy

```
row_ids = y_DL_class_pred_compare[y_DL_class_pred_compare.test ==
    ↪ y_DL_class_pred_compare.pred].index
pred_acc = (len(row_ids)/len(y_DL_class_pred_compare))
pred_acc
```

Classification by ANN - Relative data

Splitting dataset in X and y and test and train

```
X_DL_class_rel = DLdata_rel_enc[:, fluorophore_number:columns]
y_DL_class_rel = DLdata_rel_enc[:, 0:fluorophore_number]
X_DL_class_rel_train, X_DL_class_rel_test, y_DL_class_rel_train,
    ↪ y_DL_class_rel_test = train_test_split(X_DL_class_rel, y_DL_class_rel,
    ↪ test_size = 0.2)
```

```
optimizer = 'adam'
classifier = Sequential()
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu',
    ↪ input_dim = (columns − fluorophore_number)))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 20, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 10, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = 60, kernel_initializer = 'uniform', activation = 'relu'))
classifier.add(Dense(units = fluorophore_number, kernel_initializer = 'uniform',
    ↪ activation = 'sigmoid'))
classifier.compile(optimizer = optimizer, loss = 'categorical_crossentropy', metrics = ['
    ↪ accuracy'])

classifier.fit(X_DL_class_rel_train, y_DL_class_rel_train, epochs=500, batch_size=25)

y_DL_class_rel_pred = classifier.predict(X_DL_class_rel_test)
```

Checking on test data

```
pd.DataFrame.from_records(y_DL_class_rel_pred)
```

Comparing y_pred to data

```
y_DL_class_rel_pred_max = pd.DataFrame.from_records(y_DL_class_rel_pred)
y_DL_class_rel_pred_maxima = y_DL_class_rel_pred_max.apply(lambda x: max(x
    ↪ ), axis = 1)
y_DL_class_rel_pred_max = y_DL_class_rel_pred_max.isin(
    ↪ y_DL_class_rel_pred_maxima)
y_DL_class_rel_pred_max = y_DL_class_rel_pred_max.stack()
#y_DL_class_rel_pred_max = pd.Series(pd.Categorical(y_DL_class_rel_pred_max[
    ↪ y_DL_class_rel_pred_max!=0].index.get_level_values(1)))
#y_DL_class_rel_pred_max = labelencoder_X_1.inverse_transform(
    ↪ y_DL_class_rel_pred_max)
#y_DL_class_rel_pred_max = pd.DataFrame(y_DL_class_rel_pred_max)
y_DL_class_rel_pred_maxima
```

```
y_DL_class_rel_test_dec = pd.DataFrame(y_DL_class_rel_test)
y_DL_class_rel_test_dec = pd.DataFrame.from_records(y_DL_class_rel_test_dec)
y_DL_class_rel_test_dec_maxima = y_DL_class_rel_test_dec.apply(lambda x:
    ↪ max(x), axis = 1)
y_DL_class_rel_test_dec = y_DL_class_rel_test_dec.isin(
    ↪ y_DL_class_rel_test_dec_maxima)
y_DL_class_rel_test_dec = y_DL_class_rel_test_dec.stack()
y_DL_class_rel_test_dec = pd.Series(pd.Categorical(y_DL_class_rel_test_dec[
    ↪ y_DL_class_rel_test_dec!=0].index.get_level_values(1)))
y_DL_class_rel_test_dec = labelencoder_X_1.inverse_transform(
    ↪ y_DL_class_rel_test_dec)
y_DL_class_rel_test_dec = pd.DataFrame(y_DL_class_rel_test_dec)
y_DL_class_rel_test_dec
```

```
y_DL_class_rel_test_dec.columns = ['test']
y_DL_class_rel_pred_max.columns = ['pred']
y_DL_class_rel_test_dec = y_DL_class_rel_test_dec.reset_index()
y_DL_class_rel_pred_max = y_DL_class_rel_pred_max.reset_index()
pd.concat((y_DL_class_rel_test_dec, y_DL_class_rel_pred_max), axis = 1)
```

Deep Learning ANN for predicting time series

Preparation of data frames - Absolute data

Main question: How long do we need to record to predict all the following timepoints?
Feature scaling is already done. Setting variable parameters:

```
timepoints_to_predict = 60
```

```
DLdata_time = mydata_T.iloc[:, 1:]
DLdata_rel_time = mydata_T_norm.iloc[:, 1:]
```

```
DLdata_time.head()
```

```
DLdata_rel_time.head()
```

Encoding absolute data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_3 = LabelEncoder()
DLdata_enc_time = pd.concat((pd.get_dummies(labelencoder_X_3.fit_transform(
    ↪ DLdata_time.iloc[:, 0].values), prefix = 'enc'), DLdata_time.iloc[:, 1:]), axis = 1)
DLdata_enc_time.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler2 = MinMaxScaler()
DLdata_enc_time = scaler.fit_transform(DLdata_enc)
```

Encoding relative data

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_4 = LabelEncoder()
DLdata_rel_enc_time = pd.concat((pd.get_dummies(labelencoder_X_4.fit_transform(
    ↪ DLdata_rel_time.iloc[:, 0].values), prefix = 'enc'), DLdata_rel_time.iloc[:, 1:]),
    ↪ axis = 1)
DLdata_rel_enc_time.head()
```

Encoding Min Max of absolute data

```
from sklearn.preprocessing import MinMaxScaler
scaler3 = MinMaxScaler()
DLdata_rel_enc_time = scaler.fit_transform(DLdata_rel_enc_time)
```

Timecourse prediction by ANN - Absolute data

Splitting dataset in X and y and test and train

```
X_DL_class_time = DLdata_enc_time[:, 1:(columns - timepoints_to_predict)]
y_DL_class_time = DLdata_enc_time[:, (columns - timepoints_to_predict):columns]
X_DL_class_train_time, X_DL_class_test_time, y_DL_class_train_time,
    ↪ y_DL_class_test_time = train_test_split(X_DL_class_time,
    ↪ y_DL_class_time, test_size = 0.2)
```

```
(columns - timepoints_to_predict)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout

def create_model():
    optimizer = 'rmsprop'
    time_predictor = Sequential()
    time_predictor.add(Dense(units = 60, kernel_initializer = 'uniform', activation = '
        ↪ sigmoid', input_dim = (columns - timepoints_to_predict - 1)))
    time_predictor.add(Dense(units = 150, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dense(units = 140, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = 120, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = 110, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dense(units = 70, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = 120, kernel_initializer = 'uniform', activation = '
        ↪ relu'))
```

```
    time_predictor.add(Dropout(0.2))
    time_predictor.add(Dense(units = timepoints_to_predict, kernel_initializer = '
        ↪ uniform', activation = 'sigmoid'))
    time_predictor.compile(optimizer = optimizer, loss = 'mse', metrics = ['accuracy'])

    return time_predictor

model = create_model()
model.fit(X_DL_class_train_time, y_DL_class_train_time, epochs=2500, batch_size
    ↪ =25)

y_DL_class_pred_time = model.predict(X_DL_class_test_time)
```

Checking on test data

```
pd.DataFrame.from_records(y_DL_class_pred_time.round(decimals= 2))
```

```
len(y_DL_class_pred_time)
```

Comparing y_pred to data graphically

```
fig, axes = plt.subplots(len(y_DL_class_pred_time), 1, sharex=True, figsize=(10,25))
fig.suptitle('Red fluorescent proteins', fontsize = 20)
for i in range(0, len(y_DL_class_pred_time)):
    axes[i].plot(y_DL_class_pred_time[i, :])
    axes[i].plot(y_DL_class_test_time[i, :])
    axes[i].set_xlabel('Time', fontsize = 10)
    #axes[i].set_ylabel('Fluorescence intensity', fontsize = 10)
fig.legend(('predicted', 'test data'), fontsize = 10)
plt.tight_layout()
plt.subplots_adjust(top = 0.97)
savefig('time_pred_red.pdf')
savefig('time_pred_red.png', dpi = 100)
plt.show
```

Loading MaMuT xml and resaving tracks as csv

This notebook is loading a MaMuT xml and resaving the included tracks as csv.
Setting the filenames

```
import os.path
newpath = r'C:\Users\..'
if not os.path.exists(newpath):
    os.makedirs(newpath)

#*********************************************************

## CHANGE TRUE to FALSE if there is not eye data avaliable!
eye_data_present = True

## Put in your ID of your DATA here! (look exel table)

here_your_eye_ID_data = 'ID7'

# *********************************************************


input_cell = 'Data/trackdata/data'+ str(here_your_eye_ID_data) + '_cell_track.xml'

output_cell = 'Data/trackdata/cell_track' + str (here_your_eye_ID_data) + '.csv'

if eye_data_present:

    input_eye = 'Data/trackdata/data'+ str(here_your_eye_ID_data) + '_eye_track.
        ↪ xml'
    output_eye = 'Data/trackdata/eye_track' + str (here_your_eye_ID_data) + '.csv'
```

Loading the xml

```
from xml.dom import minidom

if eye_data_present:
    mydoc_eye = minidom.parse(input_eye)
    spots_eye = mydoc_eye.getElementsByTagName('Spot')
    tracks_eye = mydoc_eye.getElementsByTagName('Track')

mydoc_cell = minidom.parse(input_cell)
spots_cell = mydoc_cell.getElementsByTagName('Spot')
tracks_cell = mydoc_cell.getElementsByTagName('Track')
```

Creating numpy arrays

```
import numpy as np
```

```
spotdatatable_cell = np.zeros((spots_cell.length, 6))
spotIDtable_cell = np.zeros((spots_cell.length, 1))

if eye_data_present:
    spotdatatable_eye = np.zeros((spots_eye.length, 6))
    spotIDtable_eye = np.zeros((spots_eye.length, 1))
```

Reading Spots

Populating arrays for cell tracks

```
i = 0
for elem in spots_cell:
    spotIDtable_cell[i] = elem.attributes['ID'].value
    spotdatatable_cell[i,0] = elem.attributes['POSITION_X'].value
    spotdatatable_cell[i,1] = elem.attributes['POSITION_Y'].value
    spotdatatable_cell[i,2] = elem.attributes['POSITION_Z'].value
    spotdatatable_cell[i,3] = elem.attributes['POSITION_T'].value
    i += 1
```

```
# for eye tracks
```

```
if eye_data_present:
    i = 0
    for elem in spots_eye:
        spotIDtable_eye[i] = elem.attributes['ID'].value
        spotdatatable_eye[i,0] = elem.attributes['POSITION_X'].value
        spotdatatable_eye[i,1] = elem.attributes['POSITION_Y'].value
        spotdatatable_eye[i,2] = elem.attributes['POSITION_Z'].value
        spotdatatable_eye[i,3] = elem.attributes['POSITION_T'].value
        i += 1
```

```
spotIDtable_cell[0:5,:]
```

```
spotdatatable_cell[0:5,:]
```

Creating Pandas dataframe

for cell tracks

```
import pandas as pd

spotIDdataframe_cell = pd.DataFrame(data = spotIDtable_cell, columns = ['ID'])
spotdataframe_cell = pd.DataFrame(data = spotdatatable_cell, columns = ['X', 'Y', 'Z', '
    ↪ time', 'TrackID', 'prevID'])
finalspotdata_cell = pd.concat((spotIDdataframe_cell, spotdataframe_cell), axis = 1)
finalspotdata_cell.head()
```

for eye tracks

```
if eye_data_present:
    spotIDdataframe_eye = pd.DataFrame(data = spotIDtable_eye, columns = ['ID'])
    spotdataframe_eye = pd.DataFrame(data = spotdatatable_eye, columns = ['X', 'Y', '
        ↪ Z', 'time', 'TrackID', 'prevID'])
    finalspotdata_eye = pd.concat((spotIDdataframe_eye, spotdataframe_eye), axis = 1)
    finalspotdata_eye.head()
```

Reading Tracks and adding to the dataframe

for cell tracks

```
i = 0

for elem in tracks_cell:
    trackid_cell = elem.attributes['TRACK_ID'].value
    edges_cell = elem.getElementsByTagName('Edge')
    for edge in edges_cell:
        finalspotdata_cell.loc[finalspotdata_cell.ID == int(edge.attributes['
            ↪ SPOT_SOURCE_ID'].value), 'TrackID'] = trackid_cell
        finalspotdata_cell.loc[finalspotdata_cell.ID == int(edge.attributes['
            ↪ SPOT_TARGET_ID'].value), 'TrackID'] = trackid_cell
        finalspotdata_cell.loc[finalspotdata_cell.ID == int(edge.attributes['
            ↪ SPOT_TARGET_ID'].value), 'prevID'] = int(edge.attributes['
            ↪ SPOT_SOURCE_ID'].value)
    i += 1
finalspotdata_cell.head()
```

for eye tracks

```
if eye_data_present:
    i = 0
```

```
    for elem in tracks_eye:
        trackid_eye = elem.attributes['TRACK_ID'].value
        edges_eye = elem.getElementsByTagName('Edge')
        for edge in edges_eye:
            finalspotdata_eye.loc[finalspotdata_eye.ID == int(edge.attributes['
                ↪ SPOT_SOURCE_ID'].value), 'TrackID'] = trackid_eye
            finalspotdata_eye.loc[finalspotdata_eye.ID == int(edge.attributes['
                ↪ SPOT_TARGET_ID'].value), 'TrackID'] = trackid_eye
            finalspotdata_eye.loc[finalspotdata_eye.ID == int(edge.attributes['
                ↪ SPOT_TARGET_ID'].value), 'prevID'] = int(edge.attributes['
                ↪ SPOT_SOURCE_ID'].value)
        i += 1
    finalspotdata_eye.head()
```

```
finalspotdata_cell.tail()
```

Saving CSV

```
finalspotdata_cell.to_csv(path_or_buf=output_cell)
if eye_data_present:
    finalspotdata_eye.to_csv(path_or_buf=output_eye)
```

Global correction of affine transformation between timesteps

This notebook corrects the drift and size increase between each timepoint by finding the most likely affine transformation to the previous point cloud using tracks as anchor points.

Loading csv

```
#************************************************************

## Put in your ID of your DATA here! (look at excel table)

here_your_eye_ID_data = 'ID7'


# ************************************************************************
####choose your transformation (0=affine,1=rigid,2=no transformation)####
```

```
correction_you_desire = [1]

#### is eye data present?####

eye_data_present = False

#*********************************************************************

# ***********************************************************

from pathlib import Path

eye_file = Path('Data/trackdata/eye_track' + str (here_your_eye_ID_data)+ '.csv')
if eye_file.is_file():
    # file exists

    input_eyetrack = 'Data/trackdata/eye_track' + str (here_your_eye_ID_data)+ '.
        ↪ csv'



input_celltrack = 'Data/trackdata/cell_track' + str (here_your_eye_ID_data) + '.csv'

import os.path
newpath_affine = r'C:\Users\...'
if not os.path.exists(newpath_affine):
    os.makedirs(newpath_affine)

output_eyetrack_affine_corr = 'Data/affine/eye_track_affine_corr' + str (
    ↪ here_your_eye_ID_data) + '.csv'
output_celltrack_affine_corr = 'Data/affine/cell_track_affine_corr' + str (
    ↪ here_your_eye_ID_data) + '.csv'

newpath_rigid = r'C:\Users\...'
if not os.path.exists(newpath_rigid):
    os.makedirs(newpath_rigid)

output_eyetrack_rigid_corr = 'Data/rigid/eye_track_rigid_corr' + str (
    ↪ here_your_eye_ID_data) + '.csv'
output_celltrack_rigid_corr = 'Data/rigid/cell_track_rigid_corr' + str (
    ↪ here_your_eye_ID_data) + '.csv'

newpath_no_corr = r'C:\Users\...'
if not os.path.exists(newpath_no_corr):
    os.makedirs(newpath_no_corr)
```

```
output_eyetrack_no_corr = 'Data/no_corr/eye_track_no_corr' + str (
    ↪ here_your_eye_ID_data) + '.csv'
output_celltrack_no_corr = 'Data/no_corr/cell_track_no_corr' + str (
    ↪ here_your_eye_ID_data) + '.csv'
```

```
import pandas as pd
import numpy as np

cell_tracks = pd.read_csv(input_celltrack).iloc[:, 1:]

eye_tracks = pd.read_csv(input_eyetrack).iloc[:, 1:]
eye_tracks.head()
```

```
eye_tracks.tail()
```

```
cell_tracks.head()
```

```
cell_tracks.tail()
```

Building point cloud vectors

```
timemin = min(cell_tracks.time)
timemin
```

```
timemax = max(cell_tracks.time)
timemax
```

Preparing new dataframe

```
temp_dataframe_cell = np.zeros((len(cell_tracks), 3)) #creating matrix with zeros #3
    ↪ collums #rows same as cell tracknumbers
temp_dataframe_cell = pd.DataFrame(temp_dataframe_cell, columns = ['Xcorr', 'Ycorr',
    ↪ 'Zcorr']) #converting into Dataframe in Pandas -–> renaming collums
```

```
temp_dataframe2_cell = cell_tracks.copy() #copies dataframe from IN 1 to add the
    ↪ newly forged dataframe to the original in the next step
target_dataframe_cell = pd.concat([temp_dataframe2_cell, temp_dataframe_cell], axis
    ↪ = 1, sort = False) #concatinates\adds the two together to get a full dataframe
    ↪ with XyZcorr collums

# Adding first points to X,Y,Zcorr for cell tracking points # dataframe.collumname calls
    ↪ collum
target_dataframe_cell.Xcorr[target_dataframe_cell.time == timemin] =
    ↪ target_dataframe_cell.X[target_dataframe_cell.time == timemin]
target_dataframe_cell.Ycorr[target_dataframe_cell.time == timemin] =
    ↪ target_dataframe_cell.Y[target_dataframe_cell.time == timemin]
target_dataframe_cell.Zcorr[target_dataframe_cell.time == timemin] =
    ↪ target_dataframe_cell.Z[target_dataframe_cell.time == timemin]

if eye_data_present:
    temp_dataframe_eye = np.zeros((len(eye_tracks), 3)) #same as above, but for eye
        ↪ tracks
    temp_dataframe_eye = pd.DataFrame(temp_dataframe_eye, columns = ['Xcorr', '
        ↪ Ycorr', 'Zcorr']) # same as aove, but for eyes
    temp_dataframe2_eye = eye_tracks.copy() #copies dataframe from IN 1 to add the
        ↪ newly forged dataframe to the original in the next step
    target_dataframe_eye = pd.concat([temp_dataframe2_eye, temp_dataframe_eye],
        ↪ axis = 1, sort = False) #concatinates\adds the two together to get a full
        ↪ dataframe with XyZcorr collums

    # Adding first points to X,Y,Zcorr for eye tracking points # dataframe.collumname
        ↪ calls collum

    target_dataframe_eye.Xcorr[target_dataframe_eye.time == timemin] =
        ↪ target_dataframe_eye.X[target_dataframe_eye.time == timemin]
    target_dataframe_eye.Ycorr[target_dataframe_eye.time == timemin] =
        ↪ target_dataframe_eye.Y[target_dataframe_eye.time == timemin]
    target_dataframe_eye.Zcorr[target_dataframe_eye.time == timemin] =
        ↪ target_dataframe_eye.Z[target_dataframe_eye.time == timemin]

#view table head
target_dataframe_cell.head()
```

```
temp_dataframe2_cell.tail()
```

```
target_dataframe_cell["TrackID"].value_counts()
```

```
target_dataframe_cell = target_dataframe_cell.loc[target_dataframe_cell.TrackID != 0]
```

```
if eye_data_present:
    target_dataframe_cell["TrackID"].value_counts()
    target_dataframe_eye = target_dataframe_eye.loc[target_dataframe_eye.TrackID
        ↪ != 0]
```

Iterating through all timepoints to find the corresponding points on the previous frame –> the transformation is done on cell data

```
####AFINE TRANSFORMATION

if correction_you_desire == [0]:

    for i in range(int(timemin) + 1, int(timemax)+1):

        tp1 = target_dataframe_cell.loc[target_dataframe_cell.time == i - 1]
        tp2 = target_dataframe_cell.loc[target_dataframe_cell.time == i]

        # eliminate all points in tp2, that are not present in the tp1
        tp2 = tp2.loc[tp2.TrackID.isin(tp1.TrackID)]

        # prepare arrays for following optimization
        primary = np.zeros((len(tp2), 3))
        secondary = np.zeros([len(tp2), 3])

        # iterate over all points and add to optimization arrays
        for j in range(0,len(tp2)):
            prevID = tp2.prevID.iloc[j]
            primary[j] = [tp2.X.iloc[j], tp2.Y.iloc[j], tp2.Z.iloc[j]]
            try:
                secondary[j] = tp1.loc[tp1.ID == prevID, ['Xcorr', 'Ycorr', 'Zcorr']]
            except ValueError:
                # if new lineage, that did not exist before
                secondary[j] = [0, 0, 0]

        # Check whether one of the matrices is empty
        if (primary.size == 0 | secondary.size == 0):
            continue

        # Pad the data with ones, so that our transformation can do translations too
```

```
        n = primary.shape[0]
        pad = lambda x: np.hstack([x, np.ones((x.shape[0], 1))])
        unpad = lambda x: x[:,:-1]
        X = pad(primary)
        Y = pad(secondary)

        # Solve the least squares problem X * A = Y
        # to find our transformation matrix A
        A, res, rank, s = np.linalg.lstsq(X, Y)

        # defininig the transformation function
        transform = lambda x: unpad(np.dot(pad(x), A))

        # Transformation of the actual datapoints and adding to the dataframe
        target_dataframe_cell.loc[target_dataframe_cell.time == i, ['Xcorr', 'Ycorr', '
            ↪ Zcorr']] = transform(np.array(target_dataframe_cell.loc[
            ↪ target_dataframe_cell.time == i, ['X', 'Y', 'Z']]))
        if eye_data_present:
            target_dataframe_eye.loc[target_dataframe_eye.time == i, ['Xcorr', 'Ycorr',
                ↪ 'Zcorr']] = transform(np.array(target_dataframe_eye.loc[
                ↪ target_dataframe_eye.time == i, ['X', 'Y', 'Z']]))
        # Troubleshooting

        #print("Target:")
        #print(secondary)
        #print("Result:")
        #print(transform(primary))
        #print("Max error:", np.abs(secondary - transform(primary)).max())
        #A[np.abs(A) < 1e-10] = 0 # set really small values to zero
        #print(A)
        #print(np.array(target_dataframe.loc[target_dataframe.time == i, ['X', 'Y', 'Z']])
            ↪ )
        #print(transform(np.array(target_dataframe.loc[target_dataframe.time == i, ['X',
            ↪  'Y', 'Z']])))

        #target_dataframe.loc[target_dataframe.time == i, ['Xcorr', 'Ycorr', 'Zcorr']]]




###### RIGID TRANSFORMATION with Kabasch algorithm

elif correction_you_desire == [1]:
    print ("rigid")
```

```
import numpy as np
import numpy.linalg
import pandas as pd


# Rigidly (+scale) aligns two point clouds with know point-to-point correspondences
# with least-squares error.
# Returns (scale factor c, rotation matrix R, translation vector t) such that
# Q = P*cR + t
# if they align perfectly, or such that
# SUM over point i ( | P_i*cR + t - Q_i |^2 )
# is minimised if they don't align perfectly.


def umeyama (t1, t2):
    assert t1.shape == t2.shape, "t1 and t2 do not have the same shape" #tests if
        ↪ both datasets have the same number of row/collums
    n, dim = t1.shape

    centered_t1 = t1 - t1.mean (axis=0) #calculates centeroids by subtracting the
        ↪ means of tp1 from tp1
    centered_t2 = t2 - t2.mean (axis=0)#calculates centeroids by subtracting the
        ↪ means of tp2 from tp2

    C = np.dot(np.transpose(centered_t1), centered_t2) / n #dot multiplicates

    V, S, W = np.linalg.svd(C)
    d = (np.linalg.det(V) * np.linalg.det(W)) < 0.0

    if d:
        S[-1] = -S[-1]
        V[:, -1] = -V[:, -1]

    R = np.dot(V, W) #.dot calculates the product of V & W

    t = t2.mean(axis=0) - t1.mean(axis=0).dot(R)

    return R, t


# Testing

np.set_printoptions(precision=3)

a1 = np.array([
```

```
    [0, 0, −1],
    [0, 0, 0],
    [0, 0, 1],
    [0, 1, 0],
    [1, 0, 0],
])

a2 = np.array([
    [0, 0, 1],
    [0, 0, 0],
    [0, 0, −1],
    [0, 1, 0],
    [−1, 0, 0],
])
a2 *= 2 # for testing the scale calculation
a2 += 3 # for testing the translation calculation


R, t = umeyama(a1, a2)
print ("R = \n", R)

print ("t = \n", t)
print ("Check: a1*cR + t = a2 is", np.allclose(a1.dot(R) + t, a2))
err = ((a1.dot(R) + t − a2) ** 2).sum()
print ("Residual error", err)



for i in range(int(timemin) + 1, int(timemax)+1):

    tp1 = target_dataframe_cell.loc[target_dataframe_cell.time == i − 1] #calls
        ↪ time row in target_dataframe_cell from timepoint 0 −−> t=+1−1+0
    tp2 = target_dataframe_cell.loc[target_dataframe_cell.time == i] #calls
        ↪ timepoint 1 −−> t0+1


    # eliminate all points in tp2, that are not present in the tp1
    #isin checks bolean (true/false) and gives this as table
    tp2 = tp2.loc[tp2.TrackID.isin(tp1.TrackID)] #is tp2 TrackID also in tp1 TrackID
        ↪ ?
    tp2 = tp2.loc[tp2.prevID.isin(tp1.ID)]

    tp2 = tp2.sort_values(by=['prevID'])
    tp2 = tp2.loc[tp2.prevID != 0]
    print('____________________tp2_____________________')
```

```
print (tp2)

r,c = tp2.shape
print(r)
tp1_new = np.zeros((r,c))

print('____________________tp1new_____________________________')
print (tp1_new)
print('_____________________________________________________')

tp1_new = pd.DataFrame(tp1_new, columns = list(tp2.columns.values))

print('____________________tp1new_after header_____________________________')
print (tp1_new)
print('_____________________________________________________')

if tp1_new.shape != tp2.shape:

    print('____________________shape_missmatch_____________________________')
    print (tp1_new.shape)
    print (tp2.shape)
    print('_____________________________________________________')
    break


prevID_change = tp2.prevID
#prevID_change = target_dataframe_cell.loc[target_dataframe_cell.time == i, ['
    ↪ prevID']]
#prevID_change = prevID_change.rename(index=str, columns={"prevID": "ID"})
    ↪


#print('____________________Id_change_____________________________')
#print (prevID_change)
#print('_____________________________________________________')

for j in range (0,len(prevID_change)):
    prevID = prevID_change.iloc[j]
    tp1_new.loc[j,:] = np.array(tp1.loc[tp1.ID == prevID, :])
    #print('____________________j_____________________________')
    #print(j)
    #print('_____________________________________________________')


#print('____________________tp2_____________________________')
```

```
#print (tp2)
#print('_____________________________________________')
#print('____________________tp1new_alarm_____________________________')
#print (tp1_new)
#print('_____________________________________________')



#check = tp2.loc[tp2.prevID == 0] #all rows in tp2 with previousID = 0 in the
    ↪ check table

#if check.size != 0:
# tp1= tp1.loc[not tp1.ID.isin(check.ID)]
# tp2 = tp2.loc[not tp2.ID.isin(check.ID)] #kicks all TrackID of var tp2 with no
    ↪ previousID



# check for same dataframe shape
if tp1_new.shape != tp2.shape:
    print('Houston␣we␣have␣a␣problem!')
    print (tp1_new.shape + tp2.shape)
    break

#here we pick just X,Y,Z from the timepoints to later form products with R
tp1_new = tp1_new.loc[:,['X', 'Y', 'Z']]
tp2 = tp2.loc[:,['X', 'Y', 'Z']]


# calculate R & t for all point in tp1 and tp2
R, t = umeyama(tp1_new, tp2)

tobechanged_cell = target_dataframe_cell.loc[target_dataframe_cell.time == i,
    ↪ ['X', 'Y', 'Z']]


for k in range (0,len(tobechanged_cell)):
    print ("tp1shape")
    print(tp1_new.shape)
    print("tp2.shape")
    print(tp2.shape)
    print ("Rshape")
    print(R.shape)

    temp_cell = tobechanged_cell.iloc[k,:]
```

```
        print (temp_cell.shape)
        temp_cell = temp_cell.dot(R) + t
        tobechanged_cell.iloc[k,:] = temp_cell

    print('tada_cell' + str(i))
    print (tobechanged_cell)

    if eye_data_present:
        tobechanged_eye = target_dataframe_eye.loc[target_dataframe_eye.time
            ↪ == i, ['X', 'Y', 'Z']]
        for l in range (0,len(tobechanged_eye)):
            temp_eye = tobechanged_eye.iloc[l,:]
            temp_eye = temp_eye.dot(R) + t
            tobechanged_eye.iloc[l,:] = temp_eye

        print('tada_eye' + str(i))
        print(tobechanged_eye)
        target_dataframe_eye.loc[target_dataframe_eye.time == i, ['Xcorr', 'Ycorr',
            ↪ 'Zcorr']] = np.array(tobechanged_eye)
        target_dataframe_eye.loc[target_dataframe_eye.time == i, ['Xcorr', 'Ycorr',
            ↪ 'Zcorr']] = transform(np.array(target_dataframe_eye.loc[
            ↪ target_dataframe_eye.time == i, ['X', 'Y', 'Z']]))

    target_dataframe_cell.loc[target_dataframe_cell.time == i, ['Xcorr', 'Ycorr', '
        ↪ Zcorr']] = np.array(tobechanged_cell)

    target_dataframe_cell.tail()

    #Transformation of the actual datapoints and adding to the dataframe
    #target_dataframe_cell.loc[target_dataframe_cell.time == i, ['Xcorr', 'Ycorr', '
        ↪ Zcorr']] = transform(np.array(target_dataframe_cell.loc[
        ↪ target_dataframe_cell.time == i, ['X', 'Y', 'Z']]))


        # Troubleshooting

        #print("Target:")
        #print(secondary)
        #print("Result:")
        #print(transform(primary))
        #print("Max error:", np.abs(secondary - transform(primary)).max())
        #A[np.abs(A) < 1e-10] = 0 # set really small values to zero
        #print(A)
        #print(np.array(target_dataframe.loc[target_dataframe.time == i, ['X', 'Y', '
            ↪ Z']]))
```

```
            #print(transform(np.array(target_dataframe.loc[target_dataframe.time == i,
                ↪ ['X', 'Y', 'Z']])))

            #target_dataframe.loc[target_dataframe.time == i, ['Xcorr', 'Ycorr', 'Zcorr
                ↪ ']]

elif correction_you_desire == [2]:

    target_dataframe_cell = target_dataframe_cell
    if eye_data_present:
        target_dataframe_eye = target_dataframe_eye
```

```
if correction_you_desire == 2:
    len(tp2)
```

```
if correction_you_desire == 2:
    len(tp1)
```

```
target_dataframe_cell.head()
```

```
target_dataframe_cell.tail()
```

```
if eye_data_present:
    target_dataframe_eye.head()
```

```
if eye_data_present:
    target_dataframe_eye.tail()
```

```
target_dataframe_cell.loc[target_dataframe_cell.TrackID==51]
```

```
if eye_data_present:
    target_dataframe_eye.loc[target_dataframe_eye.time == 13]
```

```
if correction_you_desire == 2:
    if eye_data_present:
        target_dataframe_eye.loc[target_dataframe_eye.time == i]
```

```
if eye_data_present:
    target_dataframe_eye
```

Saving as csv

```
if correction_you_desire == [0]:
    target_dataframe_cell.to_csv(path_or_buf=output_celltrack_affine_corr)
    if eye_data_present:
        target_dataframe_eye.to_csv(path_or_buf=output_eyetrack_affine_corr)
    print ("Affine transformation executed on data!")
elif correction_you_desire == [1]:
    target_dataframe_cell.to_csv(path_or_buf=output_celltrack_rigid_corr)
    if eye_data_present:
        target_dataframe_eye.to_csv(path_or_buf=output_eyetrack_rigid_corr)
    print ("Rigid transformation executed on data!")
elif correction_you_desire == [2]:
    target_dataframe_cell.to_csv(path_or_buf=output_celltrack_no_corr)
    if eye_data_present:
        target_dataframe_eye.to_csv(path_or_buf=output_eyetrack_no_corr)
    print ("No Correction done on data!")
else:
    print ("You did not choose an Transformation Option: Please selcet one at IN[17]!
        ↪ ")
```

Calculating properties of points

This notebooks calculates the properties of points such as velocity and direction.

```
#*********************************************************

###PLEASE CHANGE TRUE TO FALSE IF NOT EYE DATA IS PRESENT!!!###

eye_data_present = True

here_your_eye_ID_data = "ID7"
# *********************************************************

#####choose your transformation (0=affine,1=rigid,2=no transformation)####

correction_you_desire = [1]
```

```
#*********************************************************

import pandas as pd
import numpy as np

if eye_data_present:

    if correction_you_desire == [0]:

        input_affine_eye = 'Data/affine/eye_track_affine_corr'+ str (
            ↪ here_your_eye_ID_data) + '.csv'
        output_affine_eye = 'Data/affine/eye_track_affine_corr_calc' + str (
            ↪ here_your_eye_ID_data) + '.csv'

    elif correction_you_desire == [1]:

        input_rigid_eye = 'Data/rigid/eye_track_rigid_corr'+ str (
            ↪ here_your_eye_ID_data) + '.csv'
        output_rigid_eye = 'Data/rigid/eye_track_rigid_corr_calc'+ str (
            ↪ here_your_eye_ID_data) + '.csv'

    elif correction_you_desire == [2]:

        input_no_corr_eye = 'Data/no_corr/eye_track_no_corr'+ str (
            ↪ here_your_eye_ID_data) + '.csv'
        output_no_corr_eye = 'Data/no_corr/eye_track_no_corr_calc'+ str (
            ↪ here_your_eye_ID_data) + '.csv'

    else:
        print ('Choose␣the␣correction␣of␣data␣you␣used␣above!')


input_affine_cell = 'Data/affine/cell_track_affine_corr'+ str (here_your_eye_ID_data)
    ↪  + '.csv'
output_affine_cell = 'Data/affine/cell_track_affine_corr_calc'+ str (
    ↪ here_your_eye_ID_data) + '.csv'

input_rigid_cell = 'Data/rigid/cell_track_rigid_corr' + str (here_your_eye_ID_data)
    ↪ + '.csv'
output_rigid_cell = 'Data/rigid/cell_track_rigid_corr_calc' + str (
    ↪ here_your_eye_ID_data) + '.csv'

input_no_corr_cell = 'Data/no_corr/cell_track_no_corr'+ str (
    ↪ here_your_eye_ID_data) + '.csv'
```

```
output_no_corr_cell = 'Data/no_corr/cell_track_no_corr_calc'+ str (
    ↪ here_your_eye_ID_data) + '.csv'


if correction_you_desire == [0]:

    original_dataframe_cell = pd.read_csv(input_affine_cell).iloc[:, 1:]
    if eye_data_present:
        original_dataframe_eye = pd.read_csv(input_affine_eye).iloc[:, 1:]
    print ("You chose to work with affine transformed data!")
elif correction_you_desire == [1]:
    original_dataframe_cell = pd.read_csv(input_rigid_cell).iloc[:, 1:]
    if eye_data_present:
        original_dataframe_eye = pd.read_csv(input_rigid_eye).iloc[:, 1:]
    print ("You chose to work with rigid transformed data!")
elif correction_you_desire == [2]:
    original_dataframe_cell = pd.read_csv(input_no_corr_cell).iloc[:, 1:]
    if eye_data_present:
        original_dataframe_eye = pd.read_csv(input_no_corr_eye).iloc[:, 1:]
    print ("You chose to work with non corrected data!")
else:
    print ("You did not choose an Transformation Option: Please selcet one above!")
```

calculating velocity

```
timemax = max(original_dataframe_cell.time)
timemax
```

```
timemin = min(original_dataframe_cell.time)
timemin
```

```
original_dataframe_cell.head()
```

```
original_dataframe_eye.head()
```

```
target_dataframe_cell = original_dataframe_cell
target_dataframe_eye = original_dataframe_eye
```

Saving as csv

```
#target_dataframe_cell.to_csv(path_or_buf=output_cell)
#target_dataframe_eye.to_csv(path_or_buf=output_eye)

if correction_you_desire == [0]:
    target_dataframe_cell.to_csv(path_or_buf=output_affine_cell)
    if eye_data_present:
        target_dataframe_eye.to_csv(path_or_buf=output_affine_eye)
    print ("Affine data saved with calculated informations!")
elif correction_you_desire == [1]:
    target_dataframe_cell.to_csv(path_or_buf=output_rigid_cell)
    if eye_data_present:
        target_dataframe_eye.to_csv(path_or_buf=output_rigid_eye)
    print ("Rigid data saved with calculated informations!")
elif correction_you_desire == [2]:
    target_dataframe_cell.to_csv(path_or_buf=output_no_corr_cell)
    if eye_data_present:
        target_dataframe_eye.to_csv(path_or_buf=output_no_corr_eye)
    print ("Non corrected data saved with calculated informations!")
else:
    print ("You did not choose an Transformation Option: Please selcet one at IN[3]!"
        ↪ )
```

3D Visualization of points using matplotlib

Loading the dataframe

```
import pandas as pd
import numpy as np

#************************************************************

###PLEASE CHANGE TRUE TO FALSE IF NOT EYE DATA IS PRESENT!!!###

## The eye data part is not finished in this script! Right now it only loads eye data an
    ↪ stores it into variables!

eye_data_present = False

here_your_eye_ID_data = "ID7"

# ************************************************************
```

```
#####choose your transformation (0=affine,1=rigid,2=no transformation)####

correction_you_desire = [2]


#*********************************************************

if eye_data_present:

    if correction_you_desire == [0]:

        input_eye = 'Data/affine/eye_track_affine_corr_calc'+ str (
            ↪ here_your_eye_ID_data) + '.csv'
        print ("you␣loaded␣eye␣data␣aswell")

    elif correction_you_desire == [1]:

        input_eye = 'Data/rigid/eye_track_rigid_corr_calc'+ str (
            ↪ here_your_eye_ID_data) + '.csv'
        print ("you␣loaded␣eye␣data␣aswell")

    elif correction_you_desire == [2]:

        input_eye = 'Data/no_corr/eye_track_no_corr_calc'+ str (
            ↪ here_your_eye_ID_data) + '.csv'
        print ("you␣loaded␣eye␣data␣aswell")
    else:
        print ('Choose␣the␣correction␣of␣data␣you␣used␣above!')

if correction_you_desire == [0]:

    input_cell = 'Data/affine/cell_track_affine_corr_calc'+ str (
        ↪ here_your_eye_ID_data) + '.csv'
    original_dataframe = pd.read_csv(input_cell).iloc[:, 1:]
    print ("affine␣data␣loaded")

elif correction_you_desire == [1]:

    input_cell = 'Data/rigid/cell_track_rigid_corr_calc'+ str (here_your_eye_ID_data
        ↪ ) + '.csv'
    original_dataframe = pd.read_csv(input_cell).iloc[:, 1:]
    print ("rigid␣data␣loaded")

elif correction_you_desire == [2]:
```

```
input_cell = 'Data/no_corr/cell_track_no_corr_calc'+ str (
    ↪ here_your_eye_ID_data) + '.csv'
original_dataframe = pd.read_csv(input_cell).iloc[:, 1:]
print ("non corrected data loaded")
```

```
original_dataframe.head()
```

Visualization

Importing libraries

```
from mpl_toolkits import mplot3d
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.widgets import Slider
```

Getting first and last timepoint with data

```
timemin = min(original_dataframe.time)
print('First timepoint: ', timemin)
timemax = max(original_dataframe.time)
print('Last timepoint: ', timemax)
```

Function for getting the data for the current timepoint

```
if correction_you_desire == [0]:
    def gettimepointdata(tp):
        xdata = original_dataframe.loc[original_dataframe.time == tp, 'Xcorr']
        ydata = original_dataframe.loc[original_dataframe.time == tp, 'Ycorr']
        zdata = original_dataframe.loc[original_dataframe.time == tp, 'Zcorr']
        track = original_dataframe.loc[original_dataframe.time == tp, 'TrackID']
        return xdata, ydata, zdata, track

if correction_you_desire == [1]:
    def gettimepointdata(tp):
        xdata = original_dataframe.loc[original_dataframe.time == tp, 'Xcorr']
        ydata = original_dataframe.loc[original_dataframe.time == tp, 'Ycorr']
        zdata = original_dataframe.loc[original_dataframe.time == tp, 'Zcorr']
        track = original_dataframe.loc[original_dataframe.time == tp, 'TrackID']
        return xdata, ydata, zdata, track
```

```
if correction_you_desire == [2]:
    def gettimepointdata(tp):
        xdata = original_dataframe.loc[original_dataframe.time == tp, 'X']
        ydata = original_dataframe.loc[original_dataframe.time == tp, 'Y']
        zdata = original_dataframe.loc[original_dataframe.time == tp, 'Z']
        track = original_dataframe.loc[original_dataframe.time == tp, 'TrackID']
        return xdata, ydata, zdata, track
```

Getting initial data points for the first time point

```
xdata_init, ydata_init, zdata_init, track_init = gettimepointdata(timemin)
```

Defining colormap for plotting

```
Colormap = 'Spectral'
```

Plotting

```
%matplotlib notebook

background_color = (0.6, 0.6, 0.6, 1.0)

fig, ax = plt.subplots(figsize = (9,8))
plt.subplots_adjust(left=0.25, bottom=0.25)

ax3d = plt.axes(projection='3d')

ax3d.w_xaxis.set_pane_color(background_color)
ax3d.w_yaxis.set_pane_color(background_color)
ax3d.w_zaxis.set_pane_color(background_color)

# Function for setting the axes to the maximum range that is present in the data

if correction_you_desire == [0]:
    def updateaxes():
        ax3d.set_xlim(min(original_dataframe.Xcorr), max(original_dataframe.Xcorr))
        ax3d.set_ylim(min(original_dataframe.Ycorr), max(original_dataframe.Ycorr))
        ax3d.set_zlim(min(original_dataframe.Zcorr), max(original_dataframe.Zcorr))

if correction_you_desire == [1]:
    def updateaxes():
        ax3d.set_xlim(min(original_dataframe.Xcorr), max(original_dataframe.Xcorr))
        ax3d.set_ylim(min(original_dataframe.Ycorr), max(original_dataframe.Ycorr))
        ax3d.set_zlim(min(original_dataframe.Zcorr), max(original_dataframe.Zcorr))
```

```
if correction_you_desire == [2]:
    def updateaxes():
        ax3d.set_xlim(min(original_dataframe.X), max(original_dataframe.X))
        ax3d.set_ylim(min(original_dataframe.Y), max(original_dataframe.Y))
        ax3d.set_zlim(min(original_dataframe.Z), max(original_dataframe.Z))

updateaxes()

# Initial plotting
ax3d.scatter3D(xdata_init, ydata_init, zdata_init, c=track_init, cmap=Colormap)
ax3d.set_xlabel('x-axis [\my m]')
ax3d.set_ylabel('y-axis [\my m]')
ax3d.set_zlabel('z-axis [\my m]')

# Adding TimeSlider
timeslideax = plt.axes([0.25, 0.1, 0.65, 0.03])
timeslide = Slider(timeslideax, label = 'Time', valmin = timemin, valmax = timemax,
    ↪ valstep = 1, valinit = timemin)

# Defining TimeSlider update function using previously defined functions
def update(val):
    xdata_temp, ydata_temp, zdata_temp, track_temp = gettimepointdata(val)
    ax3d.clear()
    updateaxes()
    ax3d.scatter3D(xdata_temp, ydata_temp, zdata_temp, c=track_temp, cmap=
        ↪ Colormap)
    ax3d.set_xlabel('x-axis [\my m]')
    ax3d.set_ylabel('y-axis [\my m]')
    ax3d.set_zlabel('z-axis [\my m]')
timeslide.on_changed(update)

plt.show()
```

Saving rotated versions of the plot

```
### set the elevation and azimuth:
elev = -103
azi = -89

### set the timepoint
tp = 0

### load data according to timepoint
xdata_time, ydata_time, zdata_time, track_time = gettimepointdata(tp)
```

```
%matplotlib inline
plt.figure(figsize = (9,8))

ax3d_static = plt.axes(projection='3d')

### setting the background color:
background_color = (0.6, 0.6, 0.6, 1.0)
ax3d_static.w_xaxis.set_pane_color(background_color)
ax3d_static.w_yaxis.set_pane_color(background_color)
ax3d_static.w_zaxis.set_pane_color(background_color)

updateaxes()

# Initial plotting
ax3d_static.view_init(elev, azi)
ax3d_static.scatter3D(xdata_time, ydata_time, zdata_time, c=track_time, cmap=
    ↪ Colormap)
ax3d_static.set_xlabel('x-axis [\my m]')
ax3d_static.set_ylabel('y-axis [\my m]')
ax3d_static.set_zlabel('z-axis [\my m]')
plt.savefig('3dplots/'+ here_your_eye_ID_data + '/' + str(tp) + '_elev_' + str(elev) +
    ↪ '_azi_' + str(azi) +'.pdf')
plt.savefig('3dplots/'+ here_your_eye_ID_data + '/' + str(tp) + '_elev_' + str(elev) +
    ↪ '_azi_' + str(azi) +'.png')
plt.show()
```

save all timepoints for the orientation

```
### set the elevation and azimuth:
elev = -173
azi = -89

for tp in range(int(timemin), int(timemax)):

### load data according to timepoint
    xdata_time, ydata_time, zdata_time, track_time = gettimepointdata(tp)

    %matplotlib inline
    plt.figure(figsize = (9,8))

    ax3d_static = plt.axes(projection='3d')

### setting the background color:
    background_color = (0.6, 0.6, 0.6, 1.0)
    ax3d_static.w_xaxis.set_pane_color(background_color)
```

```
    ax3d_static.w_yaxis.set_pane_color(background_color)
    ax3d_static.w_zaxis.set_pane_color(background_color)

    # Function for setting the axes to the maximum range that is present in the data

    if correction_you_desire == [0]:
        def updateaxes():
            ax3d_static.set_xlim(min(original_dataframe.Xcorr), max(
                ↪ original_dataframe.Xcorr))
            ax3d_static.set_ylim(min(original_dataframe.Ycorr), max(
                ↪ original_dataframe.Ycorr))
            ax3d_static.set_zlim(min(original_dataframe.Zcorr), max(
                ↪ original_dataframe.Zcorr))

    if correction_you_desire == [1]:
        def updateaxes():
            ax3d_static.set_xlim(min(original_dataframe.Xcorr), max(
                ↪ original_dataframe.Xcorr))
            ax3d_static.set_ylim(min(original_dataframe.Ycorr), max(
                ↪ original_dataframe.Ycorr))
            ax3d_static.set_zlim(min(original_dataframe.Zcorr), max(
                ↪ original_dataframe.Zcorr))

    if correction_you_desire == [2]:
        def updateaxes():
            ax3d_static.set_xlim(min(original_dataframe.X), max(original_dataframe.X
                ↪ ))
            ax3d_static.set_ylim(min(original_dataframe.Y), max(original_dataframe.Y
                ↪ ))
            ax3d_static.set_zlim(min(original_dataframe.Z), max(original_dataframe.Z)
                ↪ )

    updateaxes()

# Initial plotting
    ax3d_static.view_init(elev, azi)
    ax3d_static.scatter3D(xdata_time, ydata_time, zdata_time, c=track_time, cmap=
        ↪ Colormap)
    ax3d_static.set_xlabel('x-axis [\my m]')
    ax3d_static.set_ylabel('y-axis [\my m]')
    ax3d_static.set_zlabel('z-axis [\my m]')
    plt.savefig('3dplots/'+ here_your_eye_ID_data + '/' + str(tp) + '_elev_' + str(
        ↪ elev) + '_azi_' + str(azi) +'.pdf')
    plt.savefig('3dplots/'+ here_your_eye_ID_data + '/' + str(tp) + '_elev_' + str(
        ↪ elev) + '_azi_' + str(azi) +'.png')
```

```
    plt.show()
```

This notebook creates xyz.files for chimera

Loading csv affine corrected

```
##here you import the uncorrected files
#type [0] for corrected (affine) or [1] rigid or [2] for non corrected
corr_non_corr = [1]

if corr_non_corr == [0]:
    input_celltrack = 'Data/affine/cell_track_affine_corrID7.csv'

if corr_non_corr == [1]:
    input_celltrack = 'Data/rigid/cell_track_rigid_corrID7.csv'

if corr_non_corr == [2]:
    input_celltrack = 'Data/no_corr/cell_track_no_corrID7.csv'



import pandas as pd
import numpy as np

cell_tracks = pd.read_csv(input_celltrack).iloc[:, 1:]
```

```
cell_tracks.head()
```

```
cell_tracks.tail()
```

Building point cloud vectors

```
timemin = min(cell_tracks.time)
timemin
```

```
timemax = max(cell_tracks.time)
timemax
```

Creating txt with XYZ coordinates of every timepoint to load into chimera

```
import numpy as np
import pandas as pd
import os.path as path



for i in range(int(timemin) + 1, int(timemax)+1):


    ### IMPORTANT --> change folder and create it by your own in the directiory
        ↪ you want

    f = open(r"D:\Chimera_XYZ\tp" + str(i) + ".xyz", "w") #creates a new txt.file for
        ↪ each timepoint


    tp = cell_tracks.loc[cell_tracks.time == i - 1] #selects one timepoint

    sort = tp.sort_values(by=['TrackID'])
    #print (sort)
    #print ("tp" +str(i))
    if corr_non_corr == [0]:
        coordinates = sort.loc[:,('Xcorr', 'Ycorr', 'Zcorr')]
    if corr_non_corr == [1]:
        coordinates = sort.loc[:,('Xcorr', 'Ycorr', 'Zcorr')]
    if corr_non_corr == [2]:
        coordinates = sort.loc[:,('X', 'Y', 'Z')]



    print (coordinates)




    for row in range (1,len(coordinates)):

        #print (row[str('X')], row[str('Y')])

        line_X = coordinates.iloc [row,0]
        line_Y = coordinates.iloc [row,1]
        line_Z = coordinates.iloc [row,2]
```

```
        #print(line_X)
        f.write("C" + "," + str(line_X) + "," + str(line_Y) +"," + str(line_Z)+"\n") #
            ↪ write to the text file

    f.close()
```

3D Visualization of single track points using matplotlib

Loading the dataframe from the xml file

setting the options

```
import pandas as pd
import numpy as np

# where is the data for import?
dataID = 6
input_cell = 'Data/trackdata/dataID' + str(dataID) + '_cell_track.xml'

# which track should be visualized?
trackID_tobevisualized = "Track_9"

# what should be color coded
# elements of choice as of now: time, subtrack
color_code = "time"

# how should it be saved?
output = '3dplots/single_tracks/ID' + str(dataID) + '_' + trackID_tobevisualized + '_'

# how should the time be corrected? (enter the timestep in minutes)
timestep = 20

# how should z be corrected? (enter the z-step in um)
zstep = 2
```

loading only this track

```
from xml.dom import minidom

#parse data
mydoc_cell = minidom.parse(input_cell)
```

```
spots_cell = mydoc_cell.getElementsByTagName('Spot')
tracks_cell = mydoc_cell.getElementsByTagName('Track')

# get matching track
for element in tracks_cell:
    if element.getAttribute("name") == trackID_tobevisualized:
        track_of_interest = element
        break

number_spots = track_of_interest.getAttribute("NUMBER_SPOTS")
number_splits = track_of_interest.getAttribute("NUMBER_SPLITS")

# creating pandas dataframe to be filled
index = range(int(number_spots) + int(number_splits))
columns = ["spot_source_id", "spot_target_id", "x", "y", "z", "time", "subtrack", "color"]
df = pd.DataFrame(index = index, columns = columns, dtype = 'float')

# fill df with edges
i = 0
for element in track_of_interest.getElementsByTagName('Edge'):
    df.iloc[i,0] = element.getAttribute("SPOT_SOURCE_ID")
    df.iloc[i,1] = element.getAttribute("SPOT_TARGET_ID")
    i += 1

# filling missing begin and end spots
missing_spots_end = df["spot_target_id"][~df["spot_target_id"].isin(df["spot_source_id"
    ↪ ])].drop_duplicates()
missing_spots_end = missing_spots_end.dropna()
for i in range(len(missing_spots_end)):
        row = 1 + i
        df.iloc[-row, 0] = missing_spots_end.iloc[i]

# fill track data frame with x,y,z,t
for row in df.itertuples(index=True, name='Pandas'):
    currid = getattr(row, "spot_source_id")
    currindex = getattr(row, "Index")
    ## find corresponding ID in spots and fill x,y,z,t
    for element in spots_cell:
        if element.getAttribute("ID") == currid:
            df.iloc[currindex, 2] = float(element.attributes['POSITION_X'].value)
            df.iloc[currindex, 3] = float(element.attributes['POSITION_Y'].value)
            df.iloc[currindex, 4] = float(element.attributes['POSITION_Z'].value)
            df.iloc[currindex, 5] = float(element.attributes['POSITION_T'].value)
            break
# delete NaN rows
```

```
# df = df.dropna(thresh=3)

# correct data types
df.spot_source_id = df.spot_source_id.astype(float)
df.spot_target_id = df.spot_target_id.astype(float)

# correct z and time
df.z = df.z * zstep
df.time = df.time * timestep / 60
```

```
df.head()
```

```
df.tail()
```

establishing subtracks

so far only working with one division

```
import random
# finding origin
## initialize at a random position
start_position = random.randint(0,len(df.index))

## starting at the random position trace back to origin
origin = -1
not_found = True
previous_ID = df.iloc[start_position,0]
while not_found:
    if previous_ID in df.spot_target_id.unique():
        previous_ID = df[df.spot_target_id == previous_ID].iloc[0,0]
    else:
        origin = int(previous_ID)
        not_found = False

# finding split events (only supports one division so far)
## starting at origin
split_event = df.spot_source_id.value_counts()
split_event = split_event[split_event > 1].index.astype(int)

# tracing along branches and assing numbers for subtracks
## origin subtrack:
next_ID = origin
```

```
df.subtrack[df.spot_source_id == origin] = −1

while next_ID != split_event:
    next_ID = df.spot_target_id[df.spot_source_id == next_ID].iloc[0]
    df.subtrack[df.spot_source_id == next_ID] = −1

## determining both branch points
branch_points = df.spot_target_id[df.spot_source_id == split_event[0]]

## go through branches:
for i in range(len(branch_points)):
    next_ID = branch_points.iloc[i]
    next_ID_present = True
    while next_ID_present:
        if next_ID in df.spot_source_id.unique():
            df.subtrack[df.spot_source_id == next_ID] = i
            next_ID = df.spot_target_id[df.spot_source_id == next_ID].iloc[0]
        else:
            next_ID_present = False
```

```
df[df.spot_source_id == origin]
```

assigning color code

```
import matplotlib.colors
Colormap = 'Spectral'

if color_code == "time":
    df.color = df.time
if color_code == "subtrack":
    df.color = df.subtrack
```

Visualization

Importing libraries

```
from mpl_toolkits import mplot3d
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.cm as cm
```

Plotting

```
%matplotlib notebook

background_color = (0.6, 0.6, 0.6, 1.0)

plt.figure(figsize = (9,8))

ax3d = plt.axes(projection='3d')

ax3d.w_xaxis.set_pane_color(background_color)
ax3d.w_yaxis.set_pane_color(background_color)
ax3d.w_zaxis.set_pane_color(background_color)
ax3d.set_xlim(min(df.x), max(df.x))
ax3d.set_ylim(min(df.y), max(df.y))
ax3d.set_zlim(min(df.z), max(df.z))

# for time have a legend with colorbar:
if color_code == "time":
    ## plotting
    ax3d.scatter3D(df.x, df.y, df.z, c=df.color, cmap=Colormap, label = df.color)

    ## setup the normalization and the colormap
    normalize = mcolors.Normalize(vmin=df.color.min(), vmax=df.color.max())
    colormap = cm.jet

    ## setup the colorbar
    scalarmappaple = cm.ScalarMappable(norm=normalize, cmap=Colormap)
    scalarmappaple.set_array(df.color)
    plt.colorbar(scalarmappaple)

# for subtracks have a legend with points:
if color_code == "subtrack":
    ## plotting
    labels = ['origin', 'lineage 1', 'lineage 2']
    ax3d.scatter3D(df.x, df.y, df.z, c=df.color, cmap=Colormap, label = labels)

ax3d.set_xlabel('x-axis [\my m]')
ax3d.set_ylabel('y-axis [\my m]')
ax3d.set_zlabel('z-axis [\my m]')

plt.show()
```

Plotting both in the same pane and save the image

```
from matplotlib import gridspec

# set the elevation and azimuth:
azi = 52
elev = 51

%matplotlib inline

fig = plt.figure(figsize = (15,7.5))
gs = gridspec.GridSpec(1, 2, width_ratios=[1.13, 1])
ax3d_time = fig.add_subplot(gs[0], projection='3d')

ax3d_time.w_xaxis.set_pane_color(background_color)
ax3d_time.w_yaxis.set_pane_color(background_color)
ax3d_time.w_zaxis.set_pane_color(background_color)
ax3d_time.set_xlim(min(df.x), max(df.x))
ax3d_time.set_ylim(min(df.y), max(df.y))
ax3d_time.set_zlim(min(df.z), max(df.z))

# for time have a legend with colorbar:
## plotting
ax3d_time.view_init(elev, azi)
plot1 = ax3d_time.scatter3D(df.x, df.y, df.z, c=df.time, cmap=Colormap, label = df.time)

## setup the normalization and the colormap
normalize = mcolors.Normalize(vmin=df.time.min(), vmax=df.time.max())
colormap = cm.jet

## setup the colorbar with both labels
scalarmappaple = cm.ScalarMappable(norm=normalize, cmap=Colormap)
scalarmappaple.set_array(df.time)
cbar = fig.colorbar(scalarmappaple, fraction = 0.05)
cbar.set_label('time [h]')
cbar.ax.yaxis.set_label_position('left')
cbar.ax.set_aspect('auto')

# create a second axes instance and set the limits you need
ax2 = cbar.ax.twinx()
ax2.set_yticks((0,max(df.time)/2,max(df.time)))
ax2.set_yticklabels(('origin', 'lineage 1', 'lineage 2'))



ax3d_time.set_xlabel('x-axis [\my m]')
ax3d_time.set_ylabel('y-axis [\my m]')
ax3d_time.set_zlabel('z-axis [\my m]')
```

```
ax3d_subtrack = fig.add_subplot(gs[1], projection='3d')

ax3d_subtrack.w_xaxis.set_pane_color(background_color)
ax3d_subtrack.w_yaxis.set_pane_color(background_color)
ax3d_subtrack.w_zaxis.set_pane_color(background_color)
ax3d_subtrack.set_xlim(min(df.x), max(df.x))
ax3d_subtrack.set_ylim(min(df.y), max(df.y))
ax3d_subtrack.set_zlim(min(df.z), max(df.z))

# for subtracks have a legend with points:
## plotting
labels = ['origin', 'lineage 1', 'lineage 2']
ax3d_subtrack.view_init(elev, azi)
plot2 = ax3d_subtrack.scatter3D(df.x, df.y, df.z, c=df.subtrack, cmap=Colormap, label =
    ↪ labels)

ax3d_subtrack.set_xlabel('x-axis [\my m]')
ax3d_subtrack.set_ylabel('y-axis [\my m]')
ax3d_subtrack.set_zlabel('z-axis [\my m]')

plt.savefig(output + 'both_elev_' + str(elev) + '_azi_' + str(azi) +'.pdf')
plt.savefig(output + 'both_elev_' + str(elev) + '_azi_' + str(azi) +'.png')
plt.show()
```