Biomolekulare Reaktionssysteme in Mikrokompartimenten

Ein automatisiertes Analyseframework

Korbinian Kapsner

Technische Universität München
Fakultät für Physik
Lehrstuhl für Experimentalphysik – E14

Biomolekulare Reaktionssysteme in Mikrokompartimenten

- Ein automatisiertes Analyseframework -

Korbinian Kapsner

Vollständiger Abdruck der von der Fakultät für Physik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (rer. nat.)

genehmigten Dissertation.

Vorsitzende(r): Univ.-Prof. Dr. Martin Zacharias

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Friedrich C. Simmel
2. Univ.-Prof. Dr. Ulrich Gerland

Die Dissertation wurde am 10. März 2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Physik am 7. Mai 2015 angenommen.

Inhaltsverzeichnis

Biomolekulare Reaktionssysteme in Mikrokompartimenten

Ein automatisiertes Analyseframework

Gegenstand der Arbeit ist die Dynamik synthetischer biologischer Systeme, die in mikrometergroße Emulsionströpfchen eingeschlossen wurden. Experimentell wurden dazu tausende von Tröpfchen mittels automatisierter Fluoreszenzmikroskopie über Zeiträume von bis zu 24 Stunden verfolgt. Die resultierenden Videos wurden durch eine speziell für diese Aufgabe erstellte Softwaresuite analysiert, mit deren Hilfe stochastische biochemische Partitionierungseffekte sowie die Kommunikation zwischen kompartimentierten Gruppen von Bakterien nachgewiesen wurden.

Stichwörter: biomolekulare Reaktionssysteme, Mikrokompartimente, synthetische Biologie, in vitro, in vivo, Oszillator, Bakterien, Kommunikation, Partitionierung, Videoanalyse

Biomolecular reaction systems in microcompartments

An automated analysis framework

This thesis deals with the dynamics of synthetic biological systems that were encapsulated into micrometer-sized emulsion droplets. Experimentally, thousands of these droplets were tracked using automated fluorescence microscopy for periods of up to 24 hours. The resulting videos were analyzed with a dedicated software suite specifically created for this task. With the aid of the software, stochastic biochemical partitioning effects as well as the communication between compartmentalized groups of bacteria could be demonstrated.

Keywords: biomolecular reaction systems, microcompartments, synthetic biology, in vitro, in vivo, oscillator, bacteria, communication, partitioning, video analysis

1. Einleitung

Die Menschheit interessierte sich schon immer dafür, wie das Leben auf der Erde entstanden ist. Dass Lebewesen aus Zellen bestehen, wurde im 17. Jahrhundert von Robert Hooke[1] festgestellt, und wie aus einfachen Einzellern komplexe Organismen entstehen können, ist seit der Entwicklung der Evolutionstheorie durch Charles Darwin[2] bekannt. Wie aber die *ersten* Zellen entstanden und funktionierten, ist noch nicht ganz geklärt.

Die Entstehung einfacher organischer Verbindungen in der Uratmosphäre der Erde kann inzwischen erklärt werden.[3–5] Auch wie sich die ersten RNA-Moleküle bildeten[6] und sich selbst reproduzierten[7], kann man bereits ansatzweise verstehen. Um nun zu ergründen, wie sich diese zu den Vorläufern der Zellen zusammenschlossen und im Zusammenspiel funktionierten, versucht man diese sogenannten Protozellen künstlich herzustellen. Dabei gibt es zwei grundlegend unterschiedliche Herangehensweisen: Die erste geht von einfachen existierenden Organismen, z. B. Bakterien, aus und versucht sie noch weiter zu vereinfachen.[8] Dabei wird das Genom und damit die Funktionalität so lange reduziert, bis nur noch die Minimalvoraussetzungen für das Überleben erfüllt sind. Dieser Weg nennt sich „top-down“. Der zweite Weg – im Kontrast dazu „bottom-up“ genannt – geht das Problem von der anderen Seite an, indem versucht wird, eine Protozelle von Grund auf neu zu entwickeln und aufzubauen.[9, 10]

Solche rein künstlichen Zellen würden nicht nur die wissenschaftliche Neugierde stillen, sondern können durchaus auch technische Anwendungen finden: Sie könnten z. B. für die Produktion von Chemikalien oder als autonome chemische Sensoriksysteme eingesetzt werden. Dabei könnte man sie so gestalten, dass sie nur in einer bestimmten Umgebung funktionieren, die in der freien Natur nicht vorkommt. Somit hätten sie den großen Vorteil gegenüber der herkömmlichen Gentechnik, dass sie sich nicht unkontrolliert verbreiten könnten, falls sie aus Versehen in die Umwelt gelangten. Auch therapeutische und diagnostische Anwendungen sind damit durchaus denkbar.[11]

Es gibt einige Definitionen, die festlegen, welche Eigenschaften solch ein Minimalsystem vorweisen muss, um als lebend angesehen zu werden.[12–14] Der theoretische Biologe Tibor Gánti definierte folgende Bedingungen:[13]

1. Die Funktionalität der Protozelle muss einem bestimmten Programm unterliegen.
2. Die Protozelle muss sich reproduzieren.
3. Die Protozelle muss sich von ihrer Umgebung separieren.

Bei der Entwicklung von Systemen, die diese Bedingungen erfüllen, wurden bereits einige Meilensteine erreicht. So kann man schon künstliche Vesikel erzeugen, die sich selbst teilen und somit eine einfache Form von Replikation zeigen.[15–19] Auch selbst replizierende RNA-Moleküle wurden schon gefunden[20] und in Vesikeln von der Umgebung

separiert[21]. Nicht nur die Reproduktionsfähigkeit solcher Liposome, sondern auch mechanische Dynamik konnte man künstlich erzeugen[22] oder auch Proteinproduktionslösungen in sie einschließen[23]. Meist werden diese Vesikel aus Lipiden aufgebaut, aber auch Lehm-Vesikel, die eventuell den echten ersten Zellen ähnlicher sind, wurden auf ihren Einfluss auf die RNA-Replikation hin untersucht.[24]

Der notwendige und wichtige Schritt auf dem Weg zu künstlichen Zellen, nichttriviale chemische Funktionen in zellgroßen Reaktionskompartimenten zu realisieren[10, 14, 25], ist Ausgangspunkt dieser Dissertation. Dazu werden Emulsionströpfchen – Wasser in Öl –, die bereits bei Versuchen mit künstlichen Leben angewendet wurden, als Reaktionsbereiche mit passender Größe verwendet.[26, 27] Zwar sind die Experimente in dieser Arbeit zunächst eher grundlagenorientiert motiviert, aber es existieren auch schon technologische Anwendungen solcher Tröpfchen, wie z. B. als Plattform für Einzelenzymexperimente[28], für Poylmerasekettenreaktionen (PCR)[29] oder für Evolutionsexperimente[30, 31].

In dieser Arbeit sollen drei Experimente auf Basis dieser Mikrokompartimente vorgestellt werden:

- Das erste Experiment sollte die Auswirkungen des Partitionierens und des kleinen Volumens auf ein komplexes, nichtlineares System untersuchen.
- Im zweiten Experiment wurde die Emulsion nicht nur zur Bereitstellung der kleinen Volumina verwendet, sondern war selbst Teil des Systems. Dabei ermöglichte sie den Transport von Signalstoffen zu bzw. zwischen Bakterien, die in den Tröpfchen eingeschlossen waren – es wurde also die Kommunikation zwischen kleinen Bakterienkolonien beobachtet.
- Das letzte Experiment hat wieder einen ähnlichen Hintergrund wie das erste. Nur wurde dabei absichtlich das System sehr stark vereinfacht, um keine Nichtlinearität zu beobachten, sondern eine bessere und direkte Betrachtung der Vorgänge während des Partitionierens zu ermöglichen.

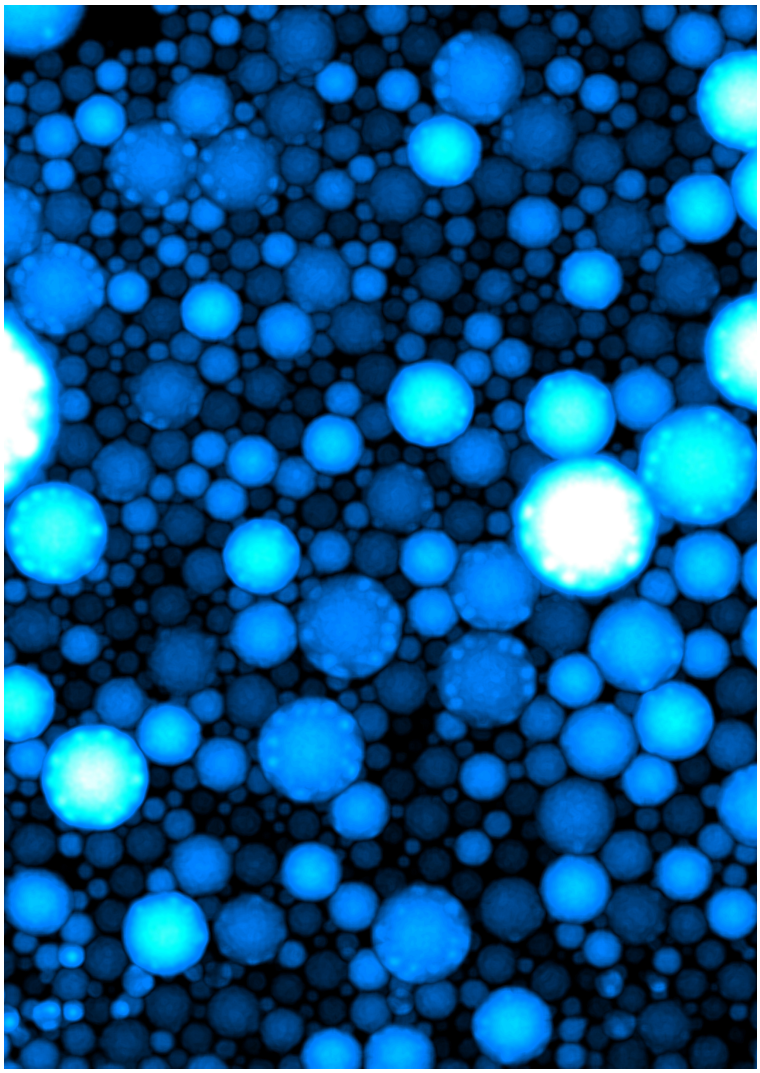
Um nun diese Tröpfchen in den Experimenten untersuchen und charakterisieren zu können, müssen sie unter dem Mikroskop betrachtet werden. Dabei kann die Anzahl der Tröpfchen pro Bildausschnitt durchaus im Bereich von Hunderten oder gar Tausenden liegen. Somit wird nicht nur *ein* kleines Volumen betrachtet, sondern *viele* kleine Volumina.

Bei den heutigen technischen Möglichkeiten werden die Experimente natürlich nicht mehr per Hand aufgezeichnet, sondern durch Computer, die dann ein Video der Reaktionen liefern. Solche Videos haben meist ein recht ästhetisches Erscheinungsbild (siehe nächste Seite), können aber nur mit sehr hohem Aufwand durch einen Menschen ausgewertet werden. Ein Computer ist zwar dem Menschen in Bezug auf Berechnungsgeschwindigkeit und Fehleranfälligkeit überlegen, kann aber zunächst nicht entscheiden, welche Informationen in den Videos relevant sind. So muss ihm erst beigebracht werden, wie die Videos zu behandeln sind: zuerst welche Teile des Videos Informationen

liefern und welche nur Hintergrund sind (Datenextraktion), und anschließend wie die interessanten Eigenschaften dieser Teile zu messen sind (Datenreduktion). So kann es passieren, dass die Auswertung eines Videos mit einer Dauer von mehreren Stunden und mit Millionen Bildpixeln am Ende nur einen Datensatz von hundert Zahlen liefert.

Die Entwicklung geeigneter Algorithmen zur Datenextraktion und -reduktion sowie deren Implementierung und Verknüpfung sind das Leitthema dieser Dissertation. Dabei wurde MATLAB®[1] als Plattform verwendet und auch der Großteil des Quelltextes ist darin geschrieben. Nur Programmteile, die sehr häufig ausgeführt werden oder sehr rechenintensiv sind, wurden in C++ verfasst und dann in MATLAB integriert.

Grundlagen und Abläufe der Experimente, die Funktionsweise des Softwarepakets sowie die gewonnenen Erkenntnisse und Schlussfolgerungen sollen in der vorliegenden Arbeit eingehend beschrieben werden.



[1]Die ersten Entwicklungen wurden in Version R2012a durchgeführt – zuletzt erfolgte eine Portierung nach R2014b. Um das Softwarepaket zu benutzen, wird die „Image processing“-Toolbox benötigt.

2. Grundlagen

2.1. Fluoreszenz[32, S. 233 ff]

Die Systeme, die in den Tröpfchen untersucht wurden, mussten in Echtzeit ausgelesen werden, da ihre Zustandsänderungen im zeitlichen Verlauf interessierten. Eine einfache Methode dafür ist die Verwendung von fluoreszierenden Farbstoffen. Diese können Licht mit einer bestimmten Wellenlänge absorbieren und emittieren dann Licht einer anderen Wellenlänge.

2.1.1. Quantenmechanische Theorie

Jedes Molekül stellt ein quantenmechanisches System dar, das in verschiedenen energetischen Zuständen – sogenannten Niveaus – vorliegen kann. Wenn nun ein Photon mit der richtigen Wellenlänge auftrifft, die genau der Energiedifferenz zwischen zwei dieser Niveaus entspricht ($\Delta E = \hbar\omega = \frac{hc}{\lambda}$), kann das System dieses absorbieren und vom niedrigenergetischen Zustand in den höherenergetischen übergehen. Das System ist dann angeregt. Wegen der Heisenbergschen Unschärferelation[33] und der Verbreiterung der Energiezustände durch thermische Vibrationszustände kann die Wellenlänge auch etwas vom Idealwert abweichen. Somit ist der Vorgang im Absorptionsspektrum auch als Maximum mit endlicher Breite zu erkennen.

Da Moleküle aus schweren Atomkernen und viel leichteren Elektronen bestehen, kann man die Zustände der beiden trennen, indem man ausnutzt, dass Elektronen viel schneller als die Kerne beschleunigen. Damit kann man die Schrödingergleichung für Elek-

Tabelle 2.1.: Liste der verwendeten Fluoreszenzfarbstoffe

Name	Anregungswellenlänge	Emissionswellenlänge
Alexa 488	490 nm	525 nm
TAMRA	559 nm	581 nm
Es werden stark unterschiedliche Werte in der Literatur angegeben. Die hier angegeben Werte sind selbst gemessen.		
Texas Red	596 nm	615 nm
Atto 655	663 nm	684 nm
Besitzt eine Seitenmode im UV.		

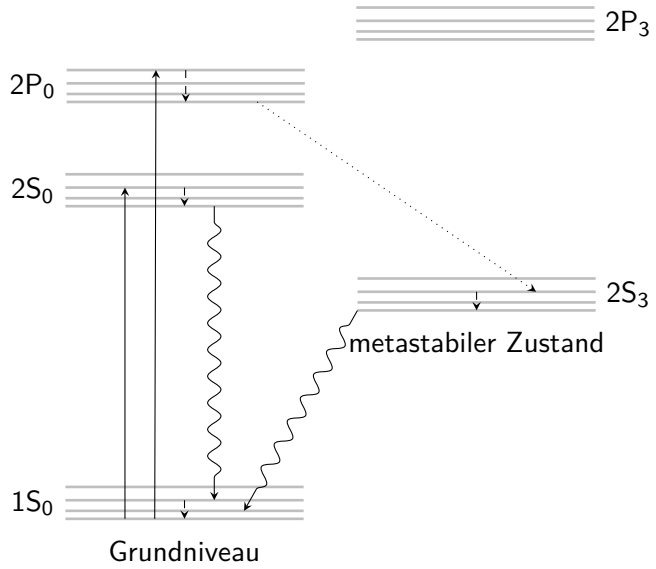


Abbildung 2.1.: Schematische Darstellung eines möglichen Fluoreszenz- und Phosphoreszenzzyklus. Die geraden Linien symbolisieren Absorptionsübergänge, die kleinen Pfeile innerhalb der Elektronenniveaus stehen für die Relaxationen, ein „verbotener“ Übergang ist gepunktet eingezeichnet und die geschwungenen Pfeile sind die Übergänge, die ein Photon erzeugen. Der Übergang von $2S_3$ nach $1S_0$ ist auch „verboten“ und deswegen handelt es sich hierbei um Phosphoreszenz.

tronen und Kerne trennen (Born-Oppenheimer-Näherung bzw. Franck-Condon-Prinzip) und sieht, dass zu jedem Elektronenzustand viele Vibrationszustände der Kerne gehören. Bei der Absorption des Photons ist es wahrscheinlicher, dass das Molekül sich danach nicht im Grundzustand der Kerne befindet. Diese Vibrationszustände können über thermische Stöße in den Grundzustand der Kerne relaxieren. Damit das System jetzt wieder in seinen Grundzustand übergeht, kann ein Photon emittiert werden, das die Energiedifferenz mitnimmt – das wird Lumineszenz genannt.

Wenn bei jedem Übergang alle Erhaltungsgrößen (z. B. Parität oder Spin) ohne zusätzlichen Stoßpartner erhalten bleiben, ist die Lebensdauer der Zustände im Bereich von ns und man spricht von Fluoreszenz. Durch Übergänge, die eine oder mehere Erhaltungssätze verletzen und damit „verboten“ sind, kann die Lebensdauer aber stark erhöht werden und man spricht von Phosphoreszenz.

Die Wahrscheinlichkeit, dass ein System vom Zustand $|i\rangle$ in den Zustand $|f\rangle$ übergeht, ist dann[1]:

$$p \propto |\langle i|\mu|f\rangle|^2 \tag{2.1.1}$$

[1] μ ist der elektromagnetische Dipoloperator.

Tabelle 2.2.: Liste der verwendeten Quencher

Name	Quenchbereich	Absorptionsmaximum
Black Hole Quencher 2 (BHQ-2)	550 – 650 nm	580 nm
Iowa Black RQ (IBRQ)	500 – 700 nm	667 nm

Das Fluoreszenzlicht hat folgende Eigenschaften:

1. Das Fluoreszenzlicht hat eine größere Wellenlänge als das absorbierte Licht, da durch die Relaxation der Kernzustände Energie verloren geht (Stokes Shift).
2. Der Aufbau der Vibrationszustände der Kerne ist bei den verschiedenen Elektronenzuständen meist recht ähnlich. Deswegen hat das Emissionspektrum eine Form, die dem Spiegelbild des Absorptionsspektrums ähnelt.
3. Die Richtung der ausgesendeten Photonen korreliert nicht mit der Richtung des Anregungslichtes.
4. Bei jedem Schritt im Lumineszenzkreislauf gibt es alternative Wege, wodurch die Wahrscheinlichkeit, dass ein absorbiertes Photon auch zu einem emittierten führt, – die Quanteneffizienz – reduziert wird.

2.1.2. Quencher

Ein einzelnes Farbstoffmolekül kann man aber nicht als isoliertes quantenmechanisches System betrachten, sondern es muss zusammen mit seiner Umgebung betrachtet werden. So gibt es Stoffe, die die Fluoreszenzeigenschaften verändern können: z. B. beeinflusst Sauerstoff die Lebensdauer des metastabilen Tripletzustandes mancher Farbstoffe[34]. Es kann aber auch die Fluoreszenz reduziert werden, indem in das System ein alternativer Abregungspfad mit höherer Wahrscheinlichkeit eingeführt wird. So kann man ein anderes geeignetes Molekül in die Nähe des Farbstoffes[2] bringen, damit die Anregungsenergie strahlungslos auf dieses übertragen werden kann.[35] Wenn dieses zweite Molekül die Energie wieder als Fluoreszenzlicht aussendet, spricht man von Fluoreszenzresonanzenergietransfer (FRET). Es ist aber auch möglich, dass es die Energie auf einem anderen Weg dissipieren kann, z. B. über thermische Stöße. Somit wird gar kein Photon emittiert. Solche Moleküle nennt man Quencher (s. Tab. 2.2) und sie werden in den Experimenten verwendet, um die Stärke des Fluoreszenzsignals der Farbstoffe (s. Tab. 2.1) zu verändern, indem der mittlere Abstand zwischen Farbstoff und Quencher variiert wird.

[2]Die Effizienz des strahlungslosen Energietransfers ist proportional zu r^{-6}.

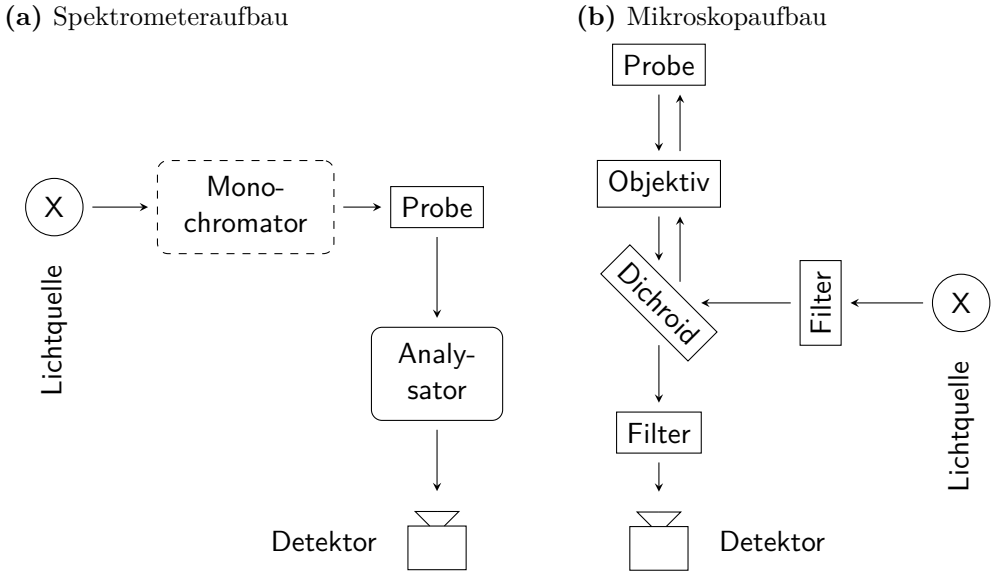


Abbildung 2.2.: Schema des Experimentalaufbaus für Fluoreszenzbetrachtungen

2.1.3. Prinzipieller Experimentalaufbau

Zuerst muss Licht mit der richtigen Anregungswellenlänge erzeugt werden. Das kann über einen Laser, eine LED oder eine Lichtquelle mit möglichst breitem Spektrum (z. B. Hg-Dampflampen oder Xe-Lampen) zusammen mit einem Monochromator (z. B. einem Gittermonochromator) realisiert werden.

Dieses Licht fällt auf die Probe und Fluoreszenz wird angeregt. Dabei wird das Fluoreszenzlicht in alle Raumrichtungen abgestrahlt. Um möglichst wenig Anregungslicht bei der späteren Detektion zu registrieren, wird der Strahlengang zum Detektor entweder rechtwinklig zur Einstrahlrichtung aufgebaut – so bei einem Fluoreszenzspektrometer – oder das Fluoreszenzlicht geht den Weg zurück und die beiden optische Wege werden mit einem dichroiden Spiegel, der Wellenlängen über einem gewissen Schwellwert durchlässt und alle anderen spiegelt, geteilt – so wird Fluoreszenz in Mikroskopen betrachtet.

Nach der örtlichen Separierung des Anregungslichtes und des Fluoreszenzlichtes wird das Fluoreszenzlicht mit einem Filter oder Monochromator noch spektral eingegrenzt, so dass man nur den Frequenzbereich betrachtet, der auch dem Fluoreszenzübergang entspricht. Dadurch kann man den Einfluss alternativer Abregungsübergänge oder störender Hintergrundlichtquellen reduzieren und somit ein Signal mit stärkerem Kontrast erzielen.

Am Ende können die Photonen dann mit jedem Gerät, das Photonen in elektronische Signale umwandeln kann (z. B. Photomultiplier, Photodiode, CCD-Pixel), detektiert werden.

2.2. Nukleinsäuren[36]

Nukleinsäuren erfüllen in der Natur viele Aufgaben, von denen die Funktion als Speicher- und Nachrichtenmedium als erste entdeckt wurde.[37] Die meisten Organismen nutzen dabei Ribonukleinsäure (im Weiteren mit RNA – **r**ibo**n**ucleic**a**cid – abgekürzt) für Nachrichten und Desoxyribonukleinsäure (im Weiteren mit DNA – **d**esoxyribo**n**ucleic**a**cid – abgekürzt) zur Langzeitspeicherung[3]. Nur einige Viren (Retroviren, z. B. HIV) verwenden RNA als Speichermedium. So werden bei Eukaryoten die Erbinformationen in Form von DNA im Zellkern gespeichert und die Kommunikation mit dem Cytoplasma findet über RNA statt.

RNA kann aber – neben der Nachrichtenübermittelung – auch vielseitige andere Funktionen erfüllen. Die drei bekanntesten Funktionen (in absteigender Häufigkeit) sind:

rRNA: Ribosomale RNAs sind integrale Bestandteile der Ribosomen, die die Proteinketten synthetisieren. Dabei erfüllen sie sowohl strukturbildende als auch katalytische Funktionen.

tRNA: Um den RNA-Code in Proteine zu übersetzen, wird die Transfer-RNA von der Zelle bereitgestellt.

mRNA: Die klassische Nachrichten (messenger) RNA. Sie dient als Vorlage für die Proteinsynthese, stellt aber nur ca. 3 – 5 % aller RNA.

Neben diesen drei Typen gibt es noch eine große Menge anderer RNA-Varianten. So existiert eine große Anzahl von regulatorischen RNA-Klassen wie z. B. miRNAs[4][38], die eine Reihe von Aufgaben erfüllen – von Zelldifferenzierung über Virenabwehr bis Zelltot[39]. Da diese miRNAs in verschiedenen menschlichen Zellen in unterschiedlichen Konzentrationsverhältnissen produziert werden, können darüber sogar auch Krebszellen identifiziert werden.[40]

Ein weiteres Beispiel sind asRNAs[5].[41–43] Auch diese können verschiedene Aufgaben erfüllen, z. B. mRNA-Deaktivierung, sodass die kodierten Proteine nicht mehr produziert werden, oder DNA-Replikationsinhibierung.[44] Ihre Mechanismen kommen aber nicht nur in der Natur vor, sondern können auch künstlich entwickelt und verwendet werden.[45, 46]

Auch in den später vorgestellten Experimenten werden solche kurzen RNA-Stränge, die aber rein künstlich sind und in der Natur keine Funktion erfüllen, zur Reaktionskontrolle verwendet. Ihre Funktionsweise ist dabei ähnlich zu asRNA.

[3] DNA hat eine höhere chemische Stabilität und benötigt deswegen weniger Reparaturen.
[4] Abkürzung für „micro RNA“, da diese RNA-Stränge nur 18 – 24 Nukleotide lang sind.
[5] Abkürzung für „antisense RNA“, da diese RNA-Stränge komplementär (s. 2.2.2) zu einem mRNA-Strang sind. Auch genannt „micRNA“: mRNA-interfering complementary RNA.

(a) Ribonukleotid **(b)** Desoxyribonukleotid

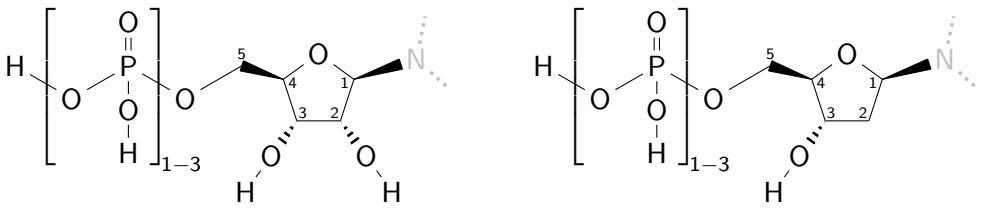


Abbildung 2.3.: Chemischer Aufbau der Nukleotide. Das hellgraue N repräsentiert den Verbindungspunkt zu den verschiedenen Basen, die dort gebunden sind. Dazu wird der Ribosering anstatt der fetten Wasserstoffatome in Abb. 2.4 an die Base gebunden. Die kleinen Zahlen sind die Nummerierung der Kohlenstoffatome. Durch die verschiedene Anzahl von Phosphaten am fünften Kohlenstoffatom (5'-Ende) des Zuckerrings unterscheiden sich die Nukleosidmono-/-bi- und -triphosphate. Der Oberbegriff über alle verschiedenen Typen ist Nukleotid.

2.2.1. Aufbau

Nulkeinsäuren sind Polymere der Ribonukleotide bzw. Desoxyribonukleotide. Deswegen wird zuerst der Aufbau der Nukleotide und dann die Polymerisation beschrieben.

2.2.1.1. Nukleotide

Der Aufbau der Nukleotide ist in Abb. 2.3 gezeigt. Diese bestehen aus bis zu drei Phosphatgruppen, einem Zuckerring – Ribose bzw. Desoxyribose – und einer der Basen. Bei DNA werden die vier Basen Adenin, Guanin, Thymin und Cytosin verwendet und bei RNA wird anstatt des Cytosins Uracil verwendet (s. Abb. 2.4). Das zentrale Element bildet der Zuckerring. An dessen erstes Kohlenstoffatom sind die Basen über eine CN-Bindung gebunden. Die Phosphatgruppe(n) bindet an das fünfte Kohlenstoffatom, weswegen das Phosphat auch als 5'-Ende der Nukleinsäure bezeichnet wird.

2.2.1.2. Polymerisation

Die Polymerisation findet nur mit Nukleotidtriphosphaten statt. Dabei bindet das erste Phosphat an das dritte Kohlenstoffatom[6] des letzten Zuckerrings im schon bestehenden Polymer[7] unter Abspaltung eines Diphosphats (s. Abb. 2.5). Da ein Nukleotid um so mehr Energie besitzt, je mehr Phosphate gebunden sind[8], wird dabei Energie frei.

Theoretisch könnte am 5'-Ende des bestehenden Polymers auch ein Triphosphat gebunden sein und somit dort die Polymerisation stattfinden. Das ist in der Natur aber nicht

[6] Auch das 3'-Ende genannt.

[7] Ein einzelnen Nukleotid kann auch schon als Polymer mit Länge 1 angesehen werden.

[8] Zum Beispiel ist ΔG beim Übergang von Adenosintriphosphat (ATP) zu Adenosindiphosphat (ADT) ca. $-12\,\mathrm{kcal\,mol^{-1}}$ und von ADT zu Adenosinmonophosphat (AMP) ca. $-13\,\mathrm{kcal\,mol^{-1}}$.

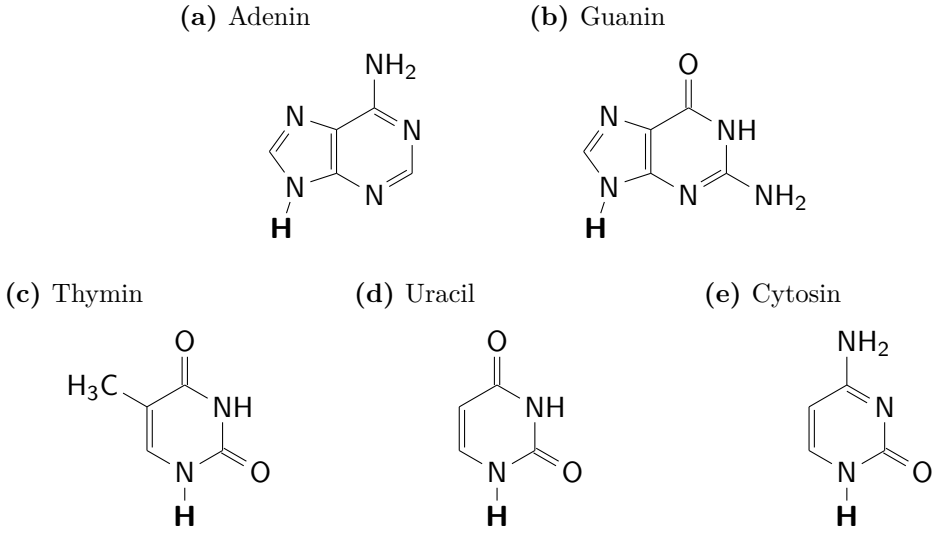


Abbildung 2.4.: Chemischer Aufbau der Basen. Das jeweils fett eingezeichnete Wasserstoffatom wird bei der Bildung der Nukleotide entfernt und an seiner Stelle bindet der Ribosering mit seinem C^1-Atom (s. Abb. 2.3).

der Fall: Die Informationen und Funktionalitäten sind in der exakten Reihenfolge der Basen gespeichert. Wenn nun durch Zufall eine falsche Base durch die Zellmaschinerie einbaut wird, muss dieser Fehler behoben werden, indem die falsche Base wieder entfernt wird. Dadurch würde aber am verkürzten Polymer nur ein Monophosphat übrig bleiben und die Polymerisation wäre erst wieder nach aufwändiger Wiederherstellung des Triphosphats möglich. Dieses Problem ergibt sich nicht, wenn immer das neu hinzuzupolymerisierende Nukleotid das Triphosphat – und damit die Energie – für die Reaktion direkt liefert.

Wegen dieser Wuchsweise wird die Basenreihenfolge – die sogenannte Sequenz – üblicherweise von 5'- zum 3'-Ende gelesen und notiert.

2.2.2. Doppelstrangbildung und Sekundärstruktur

Da die Basen beim Polymerisationsmechanismus nicht direkt involviert sind, können sie in beliebiger Reihenfolge vorliegen. Das ermöglicht die interessante Eigenschaft von Nukleinsäuren, dass zwei antiparallele Polymerstränge aneinander binden können – man spricht von hybridisieren –, wenn die Reihenfolgen der Basen zueinander passen.

2.2.2.1. Basenpaarung

Zwischen zwei Basen können sich Wasserstoffbrückenbindungen ausbilden. Dies kann prinzipiell bei allen Basenkombinationen geschehen, aber sowohl bei der Bindungsgeometrie als auch bezüglich des Abstands zwischen den beiden Zuckerrückgraten passen

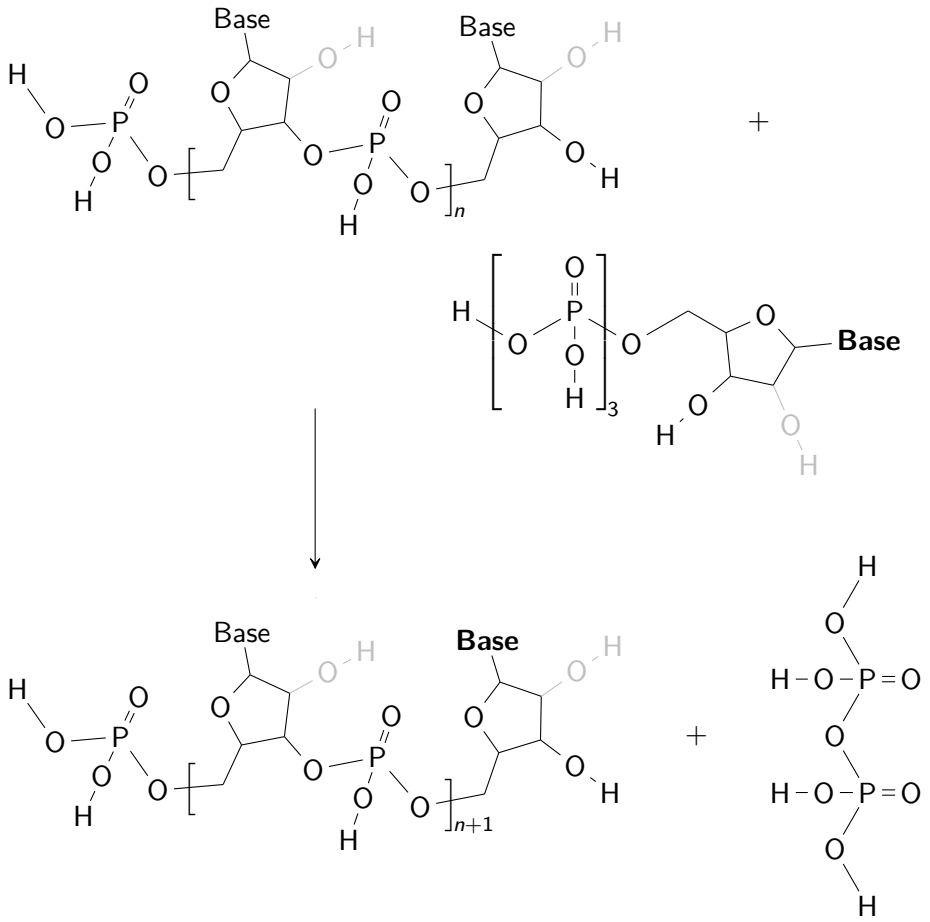
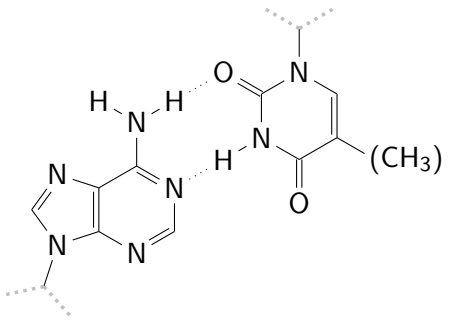


Abbildung 2.5.: Polymerisationsreaktion der Nukleinsäuren. Das Nukleotidtriphosphat bindet an das 3. Kohlenstoffatom (3'-Ende) des letzten Nukleotids im bestehenden Polymer ($n \in \mathbb{N}_0$) und spaltet dabei ein Diphosphat ab. Die hellgrauen Hydroxygruppen sind beim Rückgrat der Desoxyribonukleinsäure nicht vorhanden.

(a) AT-Basenpaarung



(b) GC-Basenpaarung

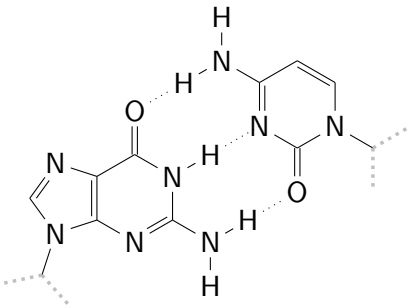


Abbildung 2.6.: Watson-Crick-Basenpaarungen. Die grau gepunkteten Linien am Rand repräsentieren den Übergang zu den Zuckerringen. Die Wasserstoffbrückenbindungen sind durch die gepunkteten Linien in der Mitte dargestellt.

Tabelle 2.3.: Basenpaarungsenergien für DNA, gemessen bei 37 °C und 1 M NaCl.[47]

Basen 5' zu 3' / 3' zu 5'	ΔH kcal mol^{-1}	ΔS cal K^{-1} mol^{-1}	ΔG kcal mol^{-1}
AA/TT	−7.6	−21.3	−1.0
AT/TA	−7.2	−20.4	−0.9
TA/AT	−7.2	−21.3	−0.6
CA/GT	−8.5	−22.7	−1.5
GT/CA	−8.4	−22.4	−1.4
CT/GA	−7.8	−21.0	−1.3
GA/CT	−8.2	−22.2	−1.3
CG/GC	−10.6	−27.2	−2.2
GC/CG	−9.8	−24.4	−2.2
GG/CC	−8.0	−19.9	−1.8
Initiation	0.2	−5.7	2.0
End-AT-Strafe	2.2	6.9	0.1
Symmetriekorrektur	0.0	−1.4	0.4

die beiden Paarungen **G**uanin-**C**ytosin und **A**denin-**U**racil/**T**hymin am besten zueinander (s. Abb. 2.6). Dadurch existiert zu jedem Nukleinsäurepolymer eine eindeutige komplementäre Sequenz, die auf ganzer Länge diese Basenpaarungen ausbilden kann. Wenn nun eine Sequenz $b_i \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^n$ mit $i \in [1, n] \cap \mathbb{N}$ der Länge n vorliegt, ergibt sich die komplementäre Sequenz folgendermaßen:

Die Operation $k(b)$, die jede Base in die am besten bindende Base umwandelt, ist definiert durch:

$$k(k(b)) = b \tag{2.2.1}$$
$$k(\mathrm{A}) = \mathrm{T} \tag{2.2.2}$$
$$k(\mathrm{G}) = \mathrm{C} \tag{2.2.3}$$

Dann ergibt sich die komplementäre Sequenz c_i:

$$c_i = k(b_{n-i+1}) \tag{2.2.4}$$

Die Sequenz wird also von hinten gelesen und jede Base wird durch die am besten Bindende ersetzt.

Da die GC-Basenpaarung eine Wasserstoffbrückenbindung mehr aufbaut als die AT-Basenpaarung, ist sie stabiler. Die Bindungsenergie ist aber nicht nur vom jeweiligen Basenpaar – im Folgenden öfters mit Bp abgekürzt – abhängig, sondern auch noch von den benachbarten Paaren (s. Tab. 2.3), da sich die Zuckerringe wie in einem Stapel übereinander anordnen und beeinflussen. Dieses sogenannte Basestacking hat einen viel höheren Beitrag zur Bindungsenergie als die reinen Wasserstoffbrückenbindungen, die in manchen Modellen sogar destabilisierend gewertet werden.[48–51]

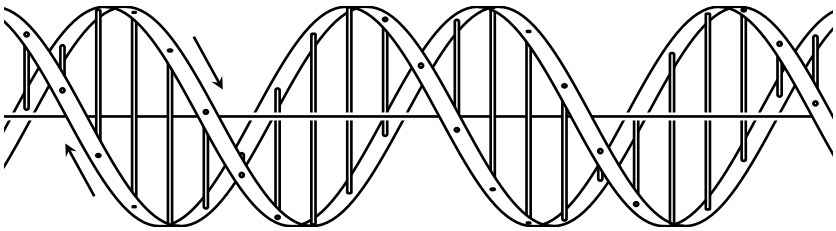


Abbildung 2.7.: Schematische Darstellung der Doppelhelix ([52] nachempfunden). Die beiden Bänder repräsentieren das Zuckerrückgrat, wobei die beiden Pfeile vom 5'- zum 3'-Ende zeigen. Die Stäbe repräsentieren die Basenpaarungen.

Tabelle 2.4.: Doppelhelixparameter für die verschiedenen kristallinen DNA Typen.[53–58]

Name	Händigkeit	Ganghöhe ($\text{Å}\,\text{Bp}^{-1}$)	Bp pro Windung
A-DNA	rechts	2.6	11.0
B-DNA	rechts	3.4	10.1
Z-DNA	links	3.7	12.0
DNA in Lösung	rechts	—	10.4
A-RNA	rechts	2.7	11.0

2.2.2.2. Physikalische Eigenschaften

Die beiden gebundenen Stränge sind aber nicht einfach antiparallel angeordnet, sondern winden sich in einer Doppelhelix umeinander (s. Abb. 2.7).[52] Dabei kann diese, abhängig von der Umgebung und der Basensequenz, verschiedene Formen annehmen (s. Tab. 2.4). Da die Einzelstränge aber flexibel sind[9], können Stränge auch auf sich selbst zurückfalten und komplizierte Sekundärstrukturen bilden, die erst einige der erwähnten Funktionalitäten ermöglichen.[10] Auch kann man künstlich Sekundärstrukturen entwickeln, die vielfältige Formen annehmen[61, 62], Funktionen ausführen[63] oder spezielle physikalische Materialeigenschaften ermöglichen[64] können.

2.2.2.3. Replikation

Da die Basensequenz die Information speichert bzw. die Funktionalität des Moleküls definiert, muss diese irgendwie kommuniziert werden. In der reinen Polymerisationsreaktion spielen die Basen keine Rolle. Deswegen werden in lebenden Organismen DNA und RNA über Proteine – die sogenannten Polymerasen (s. 2.3.1) – produziert, die einen DNA-Doppelstrang als Vorlage benutzen. Dabei wird dieser lokal aufgeschmolzen[11] und an einem oder beiden der nun frei liegenden Einzelsträngen wird die Komplementärsequenz erzeugt.

[9] Einzelsträngige DNA/RNA hat eine Persistenzlänge von 0.75 – 4.0 nm[59] – stark abhängig von den Salzkonzentrationen – und doppelsträngige DNA 30 – 80 nm[60].
[10] So bildet z. B. tRNA eine kleeblattähnliche Struktur aus.
[11] Das heißt, die Basenpaarungen werden geöffnet.

2.3. Proteine

Proteine sind eine zweite wichtige Klasse von biologischen Molekülen und sind, wie die Nukleinsäuren, Polymere. Peptide sind die Monomere und der Polymeraufbau ist in Abb. 2.8 dargestellt. Auch hier können die einzelnen Bausteine unterschiedlich aussehen – die 20 verschiedenen Seitengruppen sind in Abb. 2.9 gezeigt – ohne die grundlegende Polymerstruktur zu ändern.

Da auch hier die Reihenfolge der verschiedenen Monomere, die die Funktionalität des Proteins bestimmt und sicherstellt, die Grundstruktur der Polymerisation nicht beeinflusst, wird auch diese von einer Vorlage – der mRNA (s. 2.2) – abgelesen. Dabei kann die Sequenz nicht – wie bei der Duplikation von DNA oder der Produktion von RNA – direkt übertragen werden, sondern sie muss übersetzt werden.[12] Hierbei kodieren drei Basen auf der mRNA eine Aminosäure (s. Tab. 2.5). Die Synthese findet in den sogenannten Ribosomen statt, wobei die tRNA die Übersetzerrolle übernimmt.

Durch die extreme Flexibilität der Peptide und die stark unterschiedlichen Eigenschaften der verschiedenen Seitengruppen können Proteine sehr vielfältige Formen annehmen und dadurch auch sehr verschiedene Funktionen erfüllen. Im Folgenden sollen einige Proteintypen, die in den Experimenten eine zentrale Rolle spielen, beschrieben werden. Die meisten davon sind sogenannte Enzyme, die eine bestimmte Reaktion katalysieren.

2.3.1. RNA-Polymerase

Die Polymerisation von RNA aus den rNTPs[13] wird in Zellen von Enzymen katalysiert, die zudem noch sicherstellen, dass die produzierte RNA die richtige Sequenz enthält. Dazu bindet die RNA-Polymerase bei der DNA-Vorlage an eine bestimmte Stelle – die sogenannte Promotorregion –, schmilzt sie lokal auf, bindet an einen der beiden nun freien Einzelstränge und benutzt diesen als Vorlage – d. h. es wird die komplementäre Sequenz als RNA produziert. Der Einzelstrang, an den die Polymerase bindet, wird

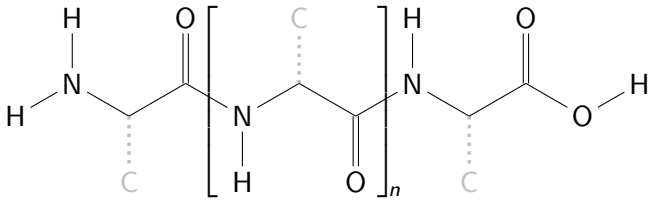


Abbildung 2.8.: Chemischer Aufbau der Polypeptidkette. Die hellgrauen Kohlenstoffatome repräsentieren die Verbindungspunkte zu den verschiedenen Seitengruppen.

[12] Der Prozess der Proteinsynthese heißt deswegen auch Translation.
[13] **R**ibo**n**ukloetid**t**ri**p**hosphate

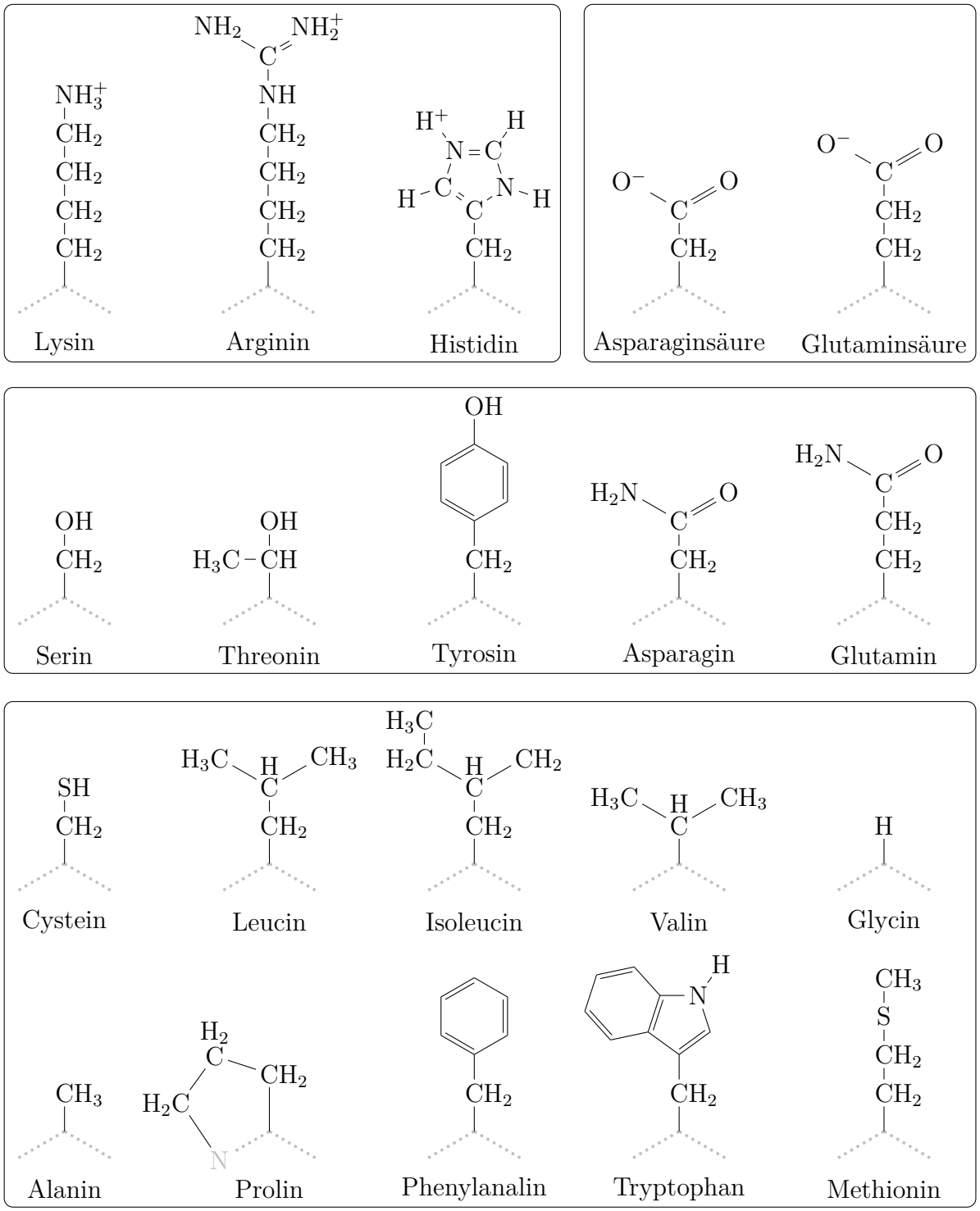


Abbildung 2.9.: Chemischer Aufbau der verschiedenen Aminosäuren. Die hellgrau gepunkteten Linien repräsentieren das Peptidrückgrat. Die Gruppierungen unterteilen die Seitengruppen in (beginnend oben links) basisch, sauer, ungeladen polar und unpolar.

Tabelle 2.5.: Übersetzungtabelle der RNA-Basen zu Aminosäuren.[36]

Aminosäure	Abkürzung	Basencode
Alanin	(Ala, A)	GCA, GCC, GCG, GCU
Arginin	(Arg, R)	AGA, AGG, CGA, CGC, CGG, CGU
Asparaginsäure	(Asp, D)	GAC, GAU
Asparagin	(Asn, N)	AAC, AAU
Cystein	(Cys, C)	UGC, UGU
Glutaminsäure	(Glu, E)	GAA, GAG
Glutamin	(Gln, Q)	CAA, CAG
Glycin	(Gly, G)	GGA, GGC, GGG, GGU
Histidin	(His, H)	CAC, CAU
Isoleucin	(Ile, I)	AUA, AUC, AUU
Leucin	(Leu, L)	UUA, UUG, CUA, CUC, CUG, CUU
Lysin	(Lys, K)	AAA, AAG
Methionin	(Met, M)	AUG
Phenylalanin	(Phe, F)	UUC, UUU
Prolin	(Pro, P)	CCA, CCC, CCG, CCU
Serin	(Ser, S)	AGC, AGU, UCA, UCC, UCG, UCU
Threonin	(Thr, T)	ACA, ACC, ACG, ACU
Tryptophan	(Trp, W)	UGG
Tyrosin	(Tyr, Y)	UAC, UAU
Valin	(Val, V)	GUA, GUC, GUG, GUU
Stop		UAA, UAG, UGA

antisense-Strang, der andere sense-Strang[14] genannt. Dieser Vorgang wird auch Transkription genannt, da die Basensequenz von der DNA auf die RNA hinübergeschrieben wird.

Da die RNA-Produktion in Zellen ein zentraler Bestandteil der Maschinerie ist, werden die RNA-Polymerasen von vielen Mechanismen kontrolliert und sind deswegen kompliziert zu handhaben. So müssen z. B. oft zuerst andere Proteine an den Promotorregionen anbinden, damit die Polymerase binden kann. Aber einige Viren, für deren Evolution es nur wichtig ist, dass sie sich schnell vermehren – ungeachtet der Konsequenzen für die Wirtszelle –, haben eine eigene RNA-Polymerase, die dann meist einfacher und leichter handzuhaben ist. So auch der T7-Phage, ein Virus, das das Escherichia Coli-Bakterium befällt[65, 66].

[14]Die Sequenz dieses Stranges liegt am Ende als RNA vor.

2.3.1.1. T7-RNA-Polymerase

Die T7-RNA-Polymerase – im Folgenden nur noch T7 genannt – besteht nur aus einer Untereinheit (s. Abb. 2.10) und benötigt als Cofaktor nur Mg^{++}-Ionen[70]. Dadurch ist sie zum einen schnell und zum anderen benötigt man in den Experimenten eine geringe Anzahl von Komponenten in der Reaktionslösung – die RNA-Polymerase des Escherichia Coli-Bakterium hat im Vergleich dazu 5 Untereinheiten und benötigt noch ein Protein als Cofaktor[71]. Die Bindestelle der T7 an der DNA ist alleine durch eine spezielle DNA-Sequenz markiert, die doppelsträngig vorliegen muss, damit die Bindung ideal ist[72, 73]:

```
5'-TAATA CGAC TCAC TATA (GGG)
```

Die drei Guanosine sind dabei der Anfang der produzierten RNA und sind im Endprodukt enthalten.

Weitere Eigenschaften der Polymerase sind[16]:

- Ideale Reaktionstemperatur: 37 °C
- Molekulargewicht: 98 kDa
- Spezifische Aktivität: 300 – 1200 units/mg[17]

2.3.2. Nukleasen

Das Gegenteil von Nukleinsäure-Polymerasen sind Nukleasen, die bestehende Nukleotid-Stränge in ihre Einzelteile zerlegen. Da jede Nuklease nur entweder DNA oder RNA zerlegen kann, kann man sie in Ribonukleasen – RNasen – und Desoxyribonukleasen – DNasen – unterteilen. Zusätzlich gibt es noch zwei verschiedene Klassen: Endonukleasen und Exonukleasen.

Exonukleasen beginnen an einem spezifischen Ende der Nukleinsäure und lösen sie von dort aus sequenziell auf.

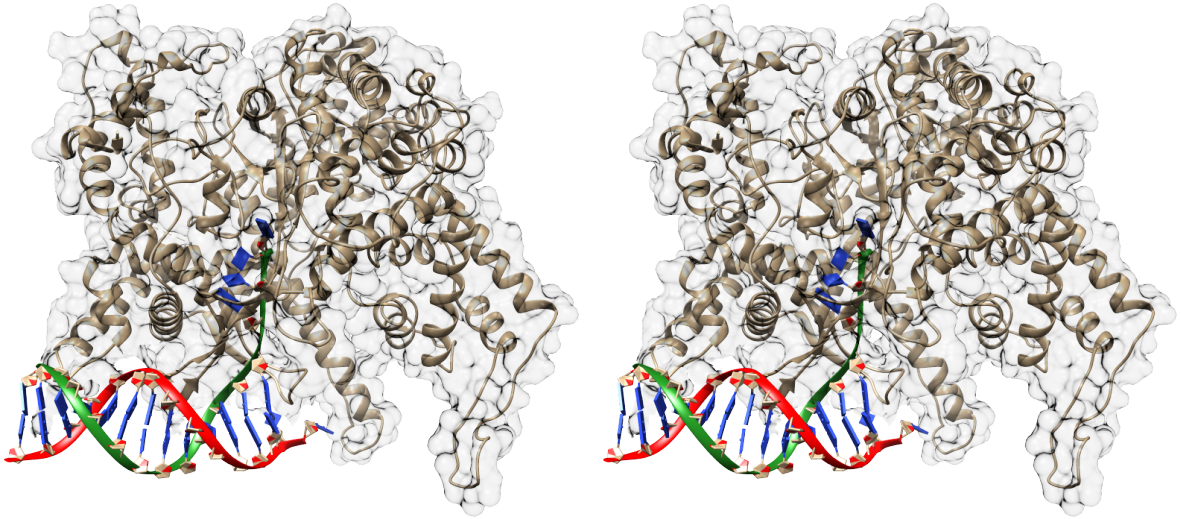
Endonukleasen spalten das Zucker-Phosphat-Rückgrat der Nukleinsäure im Inneren und fallen dann, je nach genauem Typ, ab oder lösen die Nukleinsäure in eine bestimmte Richtung auf.

[15] Die Molekülansicht wurde mit dem UCSF Chimera[67] Paket erstellt. Chimera wurde entwickelt von the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco (unterstützt von NIGMS P41-GM103311).

[16] Laut Hersteller NEB.

[17] Ein unit ist die Menge an Enzym, die nötig ist, um 1 nmol ATP in ein säureunlösliches Material in einem Reaktionsvolumen von 50 µl innerhalb einer Stunde bei 37 °C in 1X Reaktionspuffer einzubauen.

(a) Kreuzstereoansicht[15] der Promotor-Anbindung (PDB 1cez)[68]. Der Sense-Strang ist rot und der Antisense-Strang grün eingefärbt.



(b) Kreuzstereoansicht[15] der Elongation (PDB 1h38)[69]. Der Sense-Strang ist rot, der Antisense-Strang grün und das RNA-Transkript gelb eingefärbt.

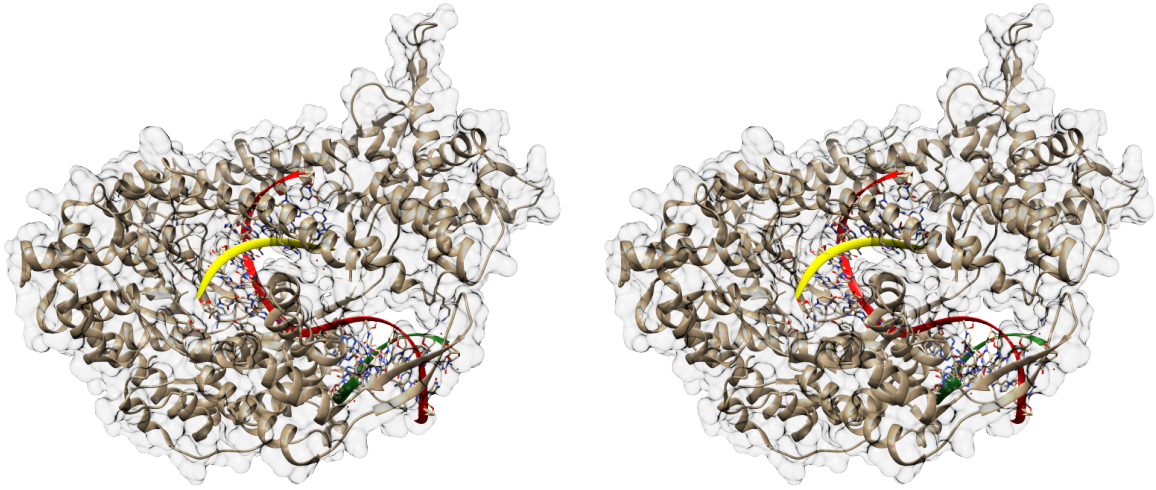


Abbildung 2.10.: Struktur der T7-RNA-Polymerase.

2.3.2.1. RNase H

Eine spezielle RNase, die auch in den Experimenten verwendet wird, ist die RNase H. Sie spielt unter anderem bei der DNA-Replikation eine wichtige Rolle, indem sie bei DNA-RNA-**H**ybridsträngen[18] – und nur bei diesen – die RNA abbaut. Dadurch werden die RNA-Primer, die zur Duplikation nötig sind, und fälschlicherweise eingebaute Ribonukleotide entfernt. Dieses muss erkannt und entfernt werden, da sonst die Langzeitstabilität der DNA gefährdet wäre. Auch bei Vermehrung von Retroviren spielt sie eine wichtige Rolle. In den Experimenten soll sie auch gezielt RNA abbauen, die an DNA gebunden ist.

2.3.2.2. Restriktionsenzyme

Restriktionsenzyme bilden eine bestimmte Art von DNasen/Endonukleasen. Sie binden bei der doppelsträngigen DNA an bestimmten Sequenzen und schneiden sie dort oder in der nächsten Nähe auseinander. Dabei muss die Schneidestelle auf den beiden Strängen nicht direkt gegenüber sitzen, sondern kann auch etwas versetzt sein. Die vier verwendeten Restriktionsenzyme mit ihren jeweiligen Erkennungssequenzen und Schneidestellen[74–78] sind:

EcoRI
5'-G▼AATT C-3'
3'-C TTAA▲G-5'

PstI
5'-C TGCA▼G-3'
3'-G▲ACGT C-5'

SpeI
5'-A▼CTAG T-3'
3'-T GATC▲A-5'

XbaI
5'-T▼CTAG A-3'
3'-A CATC▲T-5'

Hierbei ist zu beachten, dass die Erkennungssequenzen immer Palindrome sind. Das heißt, die Sequenz ist komplementär zu sich selbst.[19] Außerdem ist die vierbasige Sequenz, die beim Schneiden einzelsträngig übrig bleibt, bei SpeI und XbaI identisch. Dadurch können DNA-Stränge, die durch diese beiden Enzyme geschnitten wurden, gemischt hybridisieren. Dabei entseht eine Sequenz, die weder von SpeI noch von XbaI geschnitten wird. Diese Besonderheit wird bei sogenannten BioBricks™ (s. 2.4.2.2) ausgenutzt.

[18] Es ist also ein DNA-Strang an einen RNA-Strang gebunden.
[19] In der Notation von 2.2.2: $b_i = k(b_{n-i+1})$.

Tabelle 2.6.: Eigenschaften der fluoreszierenden Proteine[79, 80]

Name	$\lambda_{\text{Anregung}}$	$\lambda_{\text{Emission}}$
natürliches GFP	395 nm	508 nm
GFPmut3b	501 nm	511 nm
DsRed	558 nm	583 nm
RFP	584 nm	607 nm

2.3.3. DNA-Ligasen

Um DNA-Stränge mit zerschnittenem Zuckerrückgrat zu „reparieren“, gibt es DNA-Ligasen. In der Natur entstehen bei der Replikation und Reparatur von DNA solche nicht zusammenhängenden Stränge und diese müssen verknüpft werden, um die einwandfreie Funktionalität der DNA zu garantieren.

2.3.4. Fluoreszierende Proteine

Um dynamische Vorgänge in Zellen – im Besonderen in Bakterien – zu beobachten, ist es hilfreich, wenn diese Vorgänge direkt ein beobachtbares Signal erzeugen. Das kann mit Hilfe von fluoreszierenden (s. 2.1) Proteinen realisiert werden, die während der Vorgänge durch die Zelle selbst hergestellt werden.

2.3.4.1. GFP

Das bekannteste fluoreszierende Protein ist GFP – green fluorescent protein. Dieses wurde aus der Qualle Aequorea victoria extrahiert[82, 83]. Es wird aber nicht das natürliche GFP verwendet, sondern eine mutierte Version (GFPmu3b). Diese hat den Hauptvorteil, dass sie ca. 100-mal hellere Fluoreszenz zeigt, wenn sie von E. coli Bakterien produziert wird. Das liegt zum einen daran, dass die Produktionsrate höher ist, und zum anderen, dass das Protein selbst ca. 20-mal heller leuchtet. Auch haben sich die Anregungs- und Emissionswellenlängen bei der Mutation etwas geändert (s. Tab. 2.6).[79]

2.3.4.2. RFP

Als zweites leuchtendes Protein wird RFP – red fluorescent protein – verwendet. Dieses wurde aus DsRed, einem rot fluoreszierenden Protein aus der Koralle Discosoma[85], auch durch Mutation erstellt. Das war nötig, da DsRed Tetramere bildet (s. Abb. 2.12), um zu leuchten. Leider ist das Protein als Monomer etwas dunkler und weniger photostabil. Durch die fehlende Tetramerisierung fängt es aber früher an zu leuchten und die Intensität in den Zellen ist in etwa gleich.[80]

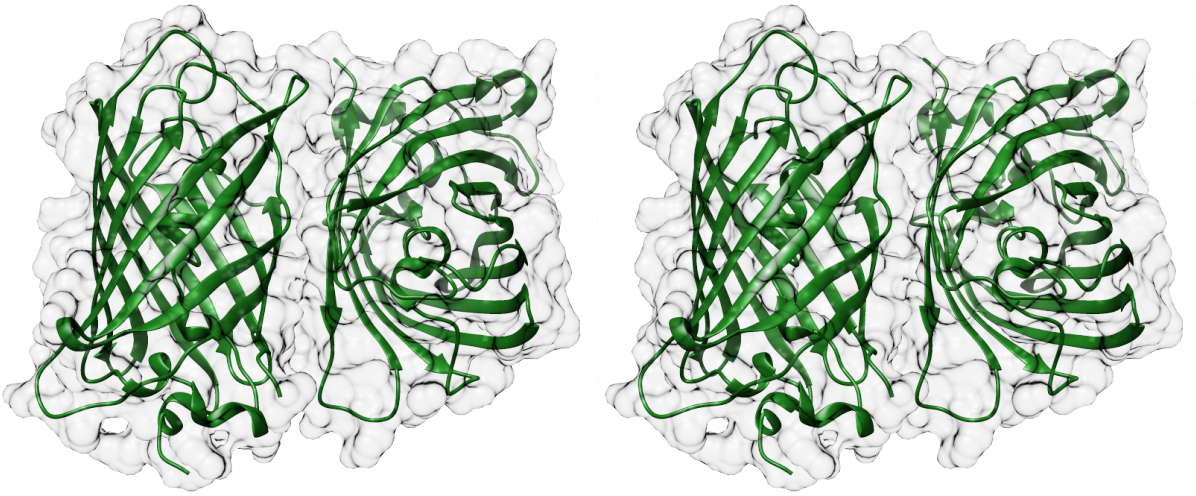


Abbildung 2.11.: Kreuzstereoansicht[15] der Struktur des GFP-Dimers (PDB 1GFL)[81]. Natürliches GFP hat eine leichte Dimerisierungstendenz.

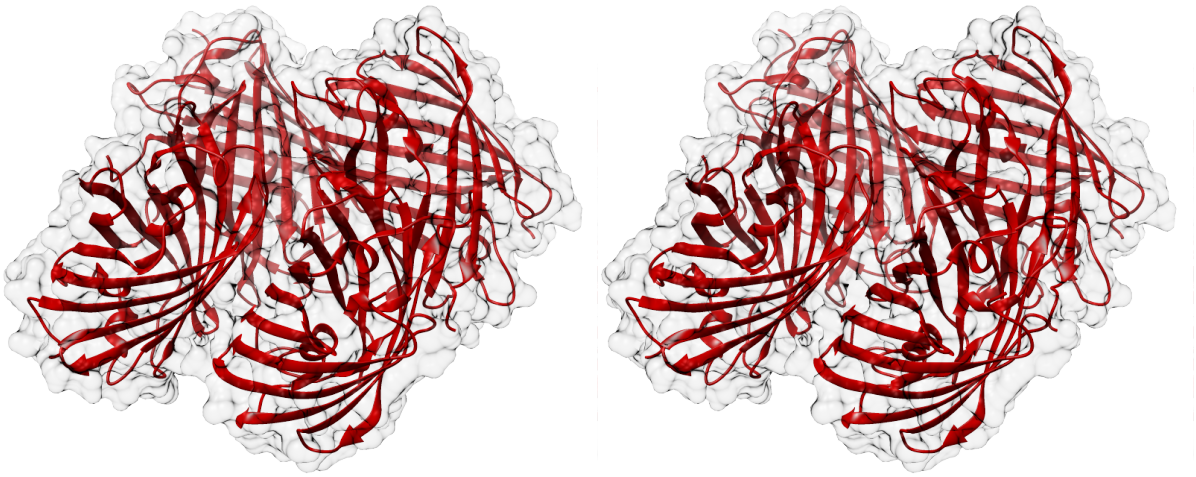


Abbildung 2.12.: Kreuzstereoansicht[15] der Struktur von DsRed (PDB 1G7K)[84].

2.4. Escherichia coli

Escherichia coli ist ein Bakterium, das im Grimmdarm[20] – auch genannt Colon – von Wirbeltieren zu finden ist.[86] Dort lebt es in Symbiose mit dem Wirt und hilft bei der Nahrungsaufnahme. Es ist eines der am meisten verwendeten Modellorganismen, an und mit dem geforscht wird. Das ist vor allem seiner einfachen Handhabung im Labor und seiner Flexibilität geschuldet. So passt es sich an verschiedene chemische Umgebungen an, kann sich schnell reproduzieren und entwickelt sich durch Mutation/Selektion rasch weiter.[36]

2.4.1. Mutanten

Da E. coli auch Durchfall, Harnwegsinfektionen oder Meningitis auslösen können[87], wurden für die Laborarbeit sogenannte Sicherheitsstämme durch Mutationen erstellt. Diese haben ein reduziertes Funktionenarsenal und stellen deswegen ein akzeptabel niedriges Gesundheitsrisiko dar. Sie müssen folgende Kriterien erfüllen:

1. Sie dürfen keinen Biofilm erzeugen.
2. Sie dürfen nicht beweglich sein, d. h. die natürlich vorkommenden Fortbewegungsmaschinerien wie z. B. Flagellen müssen deaktiviert sein.
3. Der Plasmidaustausch zwischen Bakterien muss unterbunden/deaktiviert sein.

2.4.1.1. DH5α

Um die Klonierung und Plasmidproduktion durchzuführen, wurde der DH5α Stamm von Life Technologies verwendet. Dieser hat folgende Eigenschaften[88, 89]:

1. Er stammt von Hoffman-Berling 1100 Stamm ab.
2. Deaktivierte Endonuklease ermöglicht starke Plasmidproduktion.
3. Er kann keine Lactose abbauen und produziert keinen Lac-Repressor.

Die eigentlichen Experimente wurden mit den zwei folgenden E. coli-Stämmen durchgeführt, die dafür besser geeignet sind.

2.4.1.2. DH5αZi

Die -Zi[21] Weiterentwicklung des DH5α-Stammes führt die basale Produktion des Lac-Repressors (s. 2.4.3.1) ein, wodurch dieser Stamm einfacher zu handhaben ist, wenn man auf dem zugeführten Plasmid einen Lac-Promotor verwenden möchte.[90]

[20] Der Grimmdarm ist der mittlerer Abschnitt des Dickdarms.
[21] Bezogen von der Firma ExpressSys.

2.4.1.3. BL21(DE3)pLysS

BL21(DE3) ist ein spezieller Expressionsstamm, der von B834 abstammt. Er produziert Proteine, die auf einem hinzugefügten Plasmid unter der Kontrolle des T7-Promotors (s. 2.3.1.1) stehen. Die Produktion der T7-RNA-Polymerase – und damit die Produktion des gewünschten Proteins – ist aber nicht immer aktiviert, sondern steht unter der Kontrolle eines Lac-Promotors (s. 2.4.3.1) und kann deswegen gezielt eingeschaltet werden.[91]

Die Untergruppe BL21(DE3)pLysS[22] ist ein besonderer Unterstamm, der schon ein Plasmid (pLysS) enthält. Dieses bewirkt, dass die Produktion von Proteinen, die unter der Kontrolle des T7-Promotors stehen, noch stärker reduziert ist, wenn die Zellen nicht mit IPTG induziert werden.[92]

2.4.2. Plasmide

Um nun die Bakterien bestimmte Aufgaben ausführen zu lassen, kann man zusätzliche DNA – sogenannte Plasmide – einbringen, die die Funktionalität codiert.

2.4.2.1. Eigenschaften

Es können aber nicht beliebige DNA-Stücke eingebracht werden, sondern sie müssen bestimmte Eigenschaften besitzen:

1. Sie müssen zirkular und doppelsträngig sein, um nicht von den Nukleasen im Zytoplasma abgebaut zu werden.
2. Es muss ein sogenannter origin of replication – Replikationsursprung – vorhanden sein. Dieser veranlasst die Zelle dazu, die DNA zu replizieren, damit sie auch noch nach der Zellteilung in beiden Tochterzellen vorhanden ist. Dabei gibt es unterschiedlich starke Versionen, die zwischen ein paar und tausenden Kopien variieren.
3. Da durch die Funktionalität, die auf dem Plasmid codiert ist, und die Replikation des Plasmids selbst der Metabolismus des Bakteriums beansprucht wird, hätten Bakterien, die kein Plasmid enthalten, einen Vorteil und würden sich stärker vermehren. Um diese Bakterien zu töten, setzt man dem Nährmedium ein Antibiotikum zu. Da dieses aber auch die erwünschten Bakterien töten würde, muss auf dem Plasmid eine Antibiotikumresistenz codiert sein, die die Bakterien immun macht.

Es gibt sogenannte Vektoren, die alle diese Eigenschaften besitzen, aber keinerlei sonstige Funktionalität bereitstellen. In diesen gibt es fest definierte Schneidestellen, an denen die DNA aufgeschnitten und zusätzliche DNA, die dann die Funktionalität codiert, ein gefügt werden kann.

[22] Bezogen von der Firma Promega.

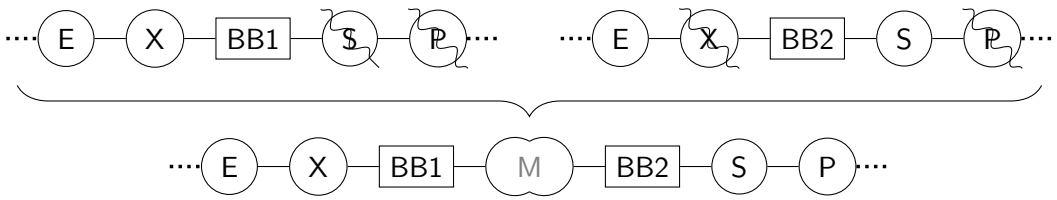


Abbildung 2.13.: Schema einer BioBricks™-Operation. E, X, S und P sind die Restriktionssequenzen für EcoRI, XbaI, SpeI bzw. PstI. BB1 und BB2 sind die funktionalen Sequenzen, die kombiniert werden. M ist die neu entstandene, nicht mehr schneidbare Sequenz. Die beiden Blöcke werden separat an den durchgestrichenen Stellen geschnitten. Anschließend werden die Fragmente vermischt und es wird eine DNA-Ligase zugesetzt, die die Fragmente dann fest verbindet. Man beachte, dass das Schema E–X–· · ·–S–P nach der Operation wieder hergestellt ist.

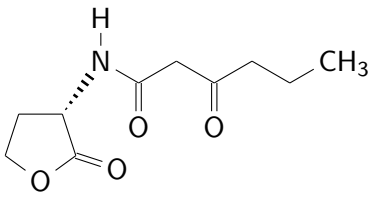
2.4.2.2. BioBricks™[93]

Ein BioBrick™ ist ein Vektor mit einer speziellen Sequenz an Schneidepositionen für verschiedene Restriktionsenzyme (s. 2.3.2.2). Die funktionale DNA-Sequenz wird am 5'-Ende von einer EcoRI- und XbaI-Erkennungssequenz und am 3'-Ende von einer SpeI- und PstI-Erkennungssequenz flankiert. Durch die Kompatibilität von SpeI und XbaI können BioBricks™ nach einem Baukastenprinzip zusammengesetzt werden. Wenn man die Schneideregeln befolgt, ist jedes Plasmid, das so zusammengesetzt wird, wieder ein BioBrick™ und kann weiter zusammengebaut werden (s. z. B. Abb. 2.13).

2.4.3. Regulationsmechanismen

Lebende Zellen haben sehr viele Mechanismen, um die Transkription bestimmter DNA-Sequenzen zu kontrollieren. Nur so können sie verschiedene Zellstadien durchlaufen und auf verschiedene Umweltbedingungen reagieren. Da diese Mechanismen fast beliebig kompliziert sein können, werden nur die zwei Signalwege beschrieben, die in dieser Arbeit verwendet wurden.

(a) AHL (3OC6HSL, CAS 143537-62-6)



(b) IPTG (CAS 367-93-1)

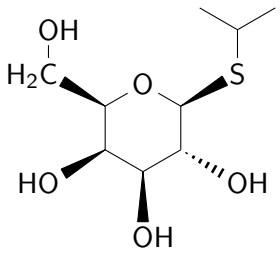


Abbildung 2.14.: Chemische Struktur der Signalstoffe.

2.4.3.1. Lac-Promotor

Um in unterschiedlichen Umgebungen überleben zu können, müssen Bakterien verschiedene Nahrungsquellen verarbeiten können. So können E. coli unter anderem auch Lactose verarbeiten. Dafür ist aber ein spezieller Proteinapparat vonnöten, der natürlich nicht bereitgestellt werden muss, wenn in der Umgebung gar keine Lactose vorhanden ist. Deswegen existiert der sogenannte Lac-Promotor, der in Abwesenheit von Lactose (bzw. der Abwesenheit von Allolactose, in die Lactose umgewandelt werden kann) durch den Lac-Repressor – das Protein LacI – blockiert ist. Allolactose bindet an das Dimer von Dimeren, den LacI im gebundenen Status bildet, und löst diesen von der DNA. Ein Analogon zu Allolactose ist Isopropyl-β-D-thiogalactopyranosid – im Folgenden mit IPTG abgekürzt. Dieses kann auch an LacI binden und es ablösen, kann aber vom Stoffwechsel der Bakterien nicht verarbeitet werden und wird deswegen zur Steuerung verwendet, da dadurch die Konzentration stabil ist und besser kontrolliert werden kann.[94–96]

2.4.3.2. Quorum-Sensing

Neben Nahrungsquellen sind auch Artgenossen ein Umgebungsfaktor, an den sich Bakterien anpassen. Dazu kommunizieren die Einzeller miteinander, um sich an die verschiedenen Anforderungen unterschiedlicher Bewuchsdichten anpassen zu können. Dabei dienen chemische Botenstoffe, die die Zellen produzieren und aussondern, als Kommunikationsmittel. Wenn sich keine Artgenossen in der Umgebung befinden, verdünnt sich die Stoffkonzentration durch Diffusion und bleibt lokal gering. Bei einer dichten Kolonie produziert jede Zelle den Stoff und die Konzentration steigt lokal an – abhängig von der Koloniedichte. Die Organismen können diese Konzentrationsteigerung messen und entsprechend reagieren. Wenn der Botenstoff ständig produziert wird, ist er ein einfaches Mittel zur Wachstumsdichtebestimmung, wenn er aber nur abhängig von bestimmten Faktoren – z. B. Stress oder Trockenheit – hergestellt wird, dient er als Abstimmungsmedium. Das heißt, es wird die Dichte der Zellen, bei denen der Faktor zutrifft, gemessen. Diesen Mechanismus nennt man Quorum-Sensing.[36]

Einer dieser Signalstoffe ist *N*-(3-Oxohexanoyl)-L-Homoserine Lactone (s. Abb. 2.14a) – im Folgenden mit AHL abgekürzt[23]. Er wird in der Natur vom Bakterium Vibrio fischeri verwendet[97] und induziert dort Biolumineszenz. Er wird vom Protein LuxI produziert und kann an LuxR binden[98], das dann dimerisiert[99], wodurch eine Promotorregion – der Lux-Promotor – für die RNA-Polymerase aktiviert wird.

[23] Die Offizielle Abkürzung ist eigentlich 3OC6HSL und AHL steht für die Obergruppe der [*N*-]Acyl-Homoserine Lactone. Da aber nur dieses eine verwendet und erwähnt wird, steht hier der Oberbegriff für das spezielle Molekül.

2.4.3.3. Anwendung

Beide Systeme – das Lac-System und Quorum-Sensing – steuern die Aktivität einer Promotorregion unabhängig von den Proteinen, deren Produktion durch die Promotoraktivität reguliert wird, und werden durch Moleküle gesteuert, die die Zellmembran durchdringen können. Deswegen kann man die Expression beliebiger Proteine steuern, indem man das Botenmolekül zur Nährlösung hinzugibt. Dadurch sind sie leicht zu handhaben und Experimente sind damit einfach zu designen.

2.5. Surfactant

Ein Surfactant[24] – auch besser bekannt als Emulgator oder Tensid – ist eine chemische Verbindung, die die Oberflächenenergie und damit auch die Oberflächenspannung zwischen zwei verschiedenen Stoffen herabsetzt, indem er sich an die Grenzfläche zwischen den verschiedenen Bereichen setzt und somit die beiden Stoffe im Idealfall keine direkten Kontaktpunkte haben, sondern nur mit dem Surfactant in Berührung kommen. Der Aufbau des Surfactanten ist mindestens zweigeteilt – ein Teil, der gerne in Verbindung mit dem einen Stoff ist, und einer, der gerne in Verbindung mit dem anderen Stoff ist – er ist also amphiphil.

Die bekanntesten Emulatoren sind Wasser-Öl-Emulgatoren, die aus einem hydrophilen (polar oder geladen) Teil und einem lipophilen (unpolar und ungeladen) Teil bestehen. Diese sind z. B. in Milch enthalten und halten die Fetttröpfchen stabil.

In den Experimenten werden Wasser-in-Öl-Tröpfchen verwendet. Das heißt, im Gegensatz zu Milch wird hier weniger wässrige Lösung – die Reaktionslösung in Puffer – in viel Öl verteilt und es bilden sich von Öl umgebene Tröpfchen, die im Inneren die wässrige Lösung enthalten. Als Öl wird ein Fluorcarbonöl – FC-40 Öl (s. Abb. 2.15) – verwendet,

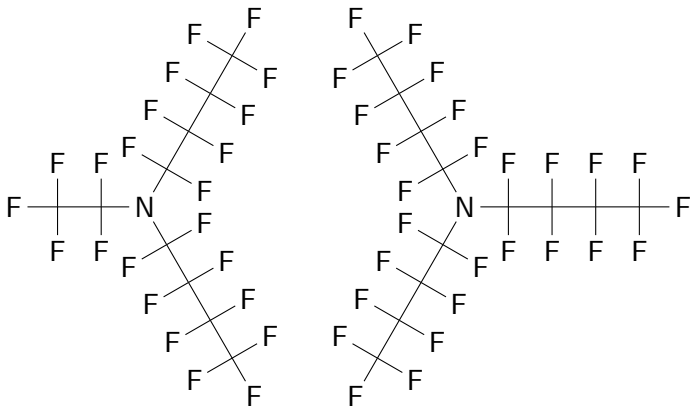


Abbildung 2.15.: Chemischer Aufbau des Fluorinert® FC-40 (CAS 51142-49-5).

[24]Kunstwort auf dem Englischen: **surf**ace **act**ive **a**ge**nt**

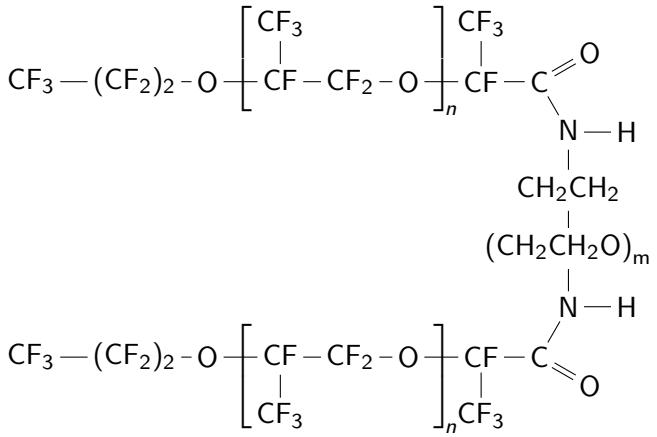


Abbildung 2.16.: Chemischer Aufbau des Surfactanten (E2K0660 RainDance Technologies).

für welches der verwendete Surfactant entwickelt wurde. Der Surfactant ist speziell für biologische Anwendungen ausgelegt worden und hat folgende Eigenschaften:[100]

1. Er besteht aus zwei lipophilen Endketten (perfluorierte Polyether PFPE) und einem hydrophilen Mittelteil (Polyethylenglycol PEG).
2. Die Tröpfchen sind sehr stabil. Sie koaleszieren[25] sehr selten und können sogar gequetscht, inkubiert oder eingefroren werden.
3. Er ist speziell nach Biokompatibilitätskriterien entwickelt worden, so dass Proteine an der Grenzfläche Wasser-Öl möglichst wenig denaturieren.

[25] verschmelzen

3. Theorie

3.1. Diffusion[101, S. 208 ff]

Man stelle sich ein Glas Wasser vor, in das man einen Tropfen Tinte gibt. Wenn man dabei vorsichtig vorgeht, gibt es am Anfang eine klare Abgrenzung zwischen der Tinte und dem Wasser. Nach einiger Zeit hat sich aber die Farbe in dem kompletten Glas ausgebreitet und das gesamte Wasser eingefärbt. Der starke Konzentrationsunterschied von Pigmenten zwischen Tinte und Wasser am Anfang hat sich also passiv von allein ausgeglichen. Diesen Vorgang nennt man Diffusion.

3.1.1. Herleitung

Die Herleitung dieses Vorgangs ist intuitiv recht gut verständlich. So nimmt man an, dass sich jedes Molekül frei bewegen kann und seine Richtung nur ändert, wenn es an ein anderes Molekül stößt. Die Bewegung eines Teilchens ist also nicht gerichtet, sondern rein zufällig. Deswegen ist die Wahrscheinlichkeit, dass sich ein **bestimmtes** Teilchen von A nach B bewegt, genauso hoch, wie dass es sich von B nach A bewegt. Wenn sich nun aber bei A mehr Teilchen befinden als bei B, ist die Wahrscheinlichkeit, dass sich **irgendein** Teilchen von A nach B bewegt, höher als umgekehrt. So gleicht sich die Teilchenanzahl so lange aus, bis die Konzentration überall gleich groß ist. Daraus folgt dann das 1. Fick'sche Gesetz, das besagt, dass der Teilchenstrom $\vec{j}$ direkt proportional zum Grandienten der Konzentration c ist:

$$\vec{j} = -D \cdot \nabla c \tag{3.1.1}$$

Die Proportionalitätskonstante D heißt Diffusionskonstante.

Zusammen mit der Kontinuumsgleichung, die ihre Grundlage im Teilchenerhalt hat,

$$\frac{\partial c}{\partial t} = -\nabla \vec{j} \tag{3.1.2}$$

ergibt sich die Diffusionsgleichung:

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \cdot \nabla c) \tag{3.1.3}$$

Diese lässt sich für spezielle Sonderfälle vereinfachen:

Isotrope Diffusionskonstante: Wenn nun die Diffusionskonstante D nicht explizit vom Ort abhängt, ergibt sich:

$$\frac{\partial c}{\partial t} = D \cdot \nabla^2 c \tag{3.1.4}$$

Im Folgenden wird dieser Sonderfall angenommen und Gleichung (3.1.4) weiter betrachtet.

Kugelsymmetrie: Wenn nun ein kugelsymmetrisches Problem vorliegt – also c nur vom Abstand zum Koordinatenursprung abhängt –, ergibt sich folgende vereinfachte Gleichung:

$$\frac{\partial c(r,t)}{\partial t} = D \cdot \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \cdot \frac{\partial}{\partial r} c(r,t)\right) = D \cdot \frac{1}{r}\frac{\partial^2}{\partial r^2}\left(r \cdot c(r,t)\right) \tag{3.1.5}$$

2D-Rotationssymmetrie: In einem System, das nur in zwei Dimensionen diffundieren kann und rotationssymmetrisch ist, wird die Diffusion beschrieben durch:

$$\frac{\partial c(r,t)}{\partial t} = D \cdot \frac{1}{r}\frac{\partial}{\partial r}\left(r \cdot \frac{\partial}{\partial r} c(r,t)\right) \tag{3.1.6}$$

3.1.2. Eigenschaften

Die Diffusionsgleichung ist linear[1]. Deswegen können zwei Lösungen addiert werden und ergeben wieder eine Lösung der Differentialgleichung. So können Anfangswertprobleme in kleine lösbare Unterprobleme zerlegt werden. Wenn man z. B. die Lösung der Differentialgleichung für das Problem eines Tropfens Tinte in einem unendlich großen Gefäß hat, kann man das Problem zweier örtlich separierter Tropfen einfach durch Addition lösen.

Durch die unterschiedliche Ordnung der Ableitung in Ort und Zeit sind die Lösungen in Ort und Zeit verschieden.

3.1.3. Diffusionskoeffizient

Der Diffusionskoeffizient ist nun aber keine Naturkonstante, sondern hängt von vielen Faktoren ab. So muss er für jedes diffundierende Teilchen oder Molekül in Abhängigkeit des umgebenden Materials und der Temperatur bestimmt werden.

[1] Das heißt, es kommt keine höher Potenz als 1 von c vor.

3.1.3.1. Kugelförmige Teilchen

Für kugelförmige Teilchen gibt es eine Formel – die Stokes-Einstein-Gleichung –, die den Diffusionskoeffizienten in Abhängigkeit von der Temperatur T, der Viskosität des Mediums η und dem Radius des Teilchens r angibt[102]:

$$D = \frac{k_B \cdot T}{6 \cdot \pi \cdot \eta \cdot r} \tag{3.1.7}$$

Diese Formel kann für beliebige Formen generalisiert werden, indem man statt eines realen Radius den hydrodynamischen Radius r_H einsetzt. Wie dieser exakt aussieht, hängt dann von der wirklichen Form des Teilchens und der Interaktion mit dem umgebenden Medium ab.

3.1.3.2. Zylinderförmige Teilchen

So kann r_H für zylinderförmige Teilchen angegeben werden durch[103]:

$$r_H = \frac{l}{2 \cdot \left(\ln\left(\frac{l}{2r}\right) + \gamma\right)} \tag{3.1.8}$$

Wobei l die Länge und r der Radius des Zylinders ist. Der Korrekturfaktor γ muss eingefügt werden, um Effekte der Enden des Zylinders zu berücksichtigen. Dafür existieren einige unterschiedliche Modelle, die zu verschiedenen Endergebnissen führen.[104]

3.1.3.3. Polymere

Für Polymere, die sich in einem zufälligen Knäuel anordnen, kann der hydrodynamische Radius nach der Zimm-Theorie bestimmt werden[105]. Dabei unterscheidet man den Fall, in dem das Lösungsmittel frei durch das Knäuel fließen kann:

$$r_H = \frac{N \cdot \rho}{6 \cdot \pi \cdot \eta} \tag{3.1.9}$$

und in dem es nicht frei fließen kann:

$$r_H = \frac{\sqrt{N} \cdot b}{0.188 \cdot 6 \cdot \pi} \tag{3.1.10}$$

wobei N die Anzahl der Monomere im Polymer, ρ der Reibungskoeffizient eines einzelnen Monomers[2] und b die mittlere Länge eines Monomers ist.

[2] Im Fall von kugelförmigen Monomeren ist das $6 \cdot \pi \cdot \eta \cdot r$.

3.1.3.4. DNA

DNA-Stränge sind etwas komplizierter zu beschreiben, aber Modelle für zylinderförmige Objekte können recht gut auf kurze doppelsträngige DNA angewendet werden[106, 107] und es existiert eine grobe heuristische Abschätzungsformel[107]:

$$D = 4.9 \times 10^2 \,\mu\mathrm{m}^2\,\mathrm{s}^{-1} \cdot bp^{-0.72} \tag{3.1.11}$$

wobei *bp* die Anzahl der Basenpaare in der DNA ist.

Diese Formel ist natürlich nur eine grobe Näherung, da die Diffusionskonstante von der Temperatur, dem Salzgehalt[108] und anderen Inhaltstoffen in der wässrigen Lösung abhängt. Auch ist der Skalierungsexponent umstritten und man findet in der Literatur Werte von -0.57[109] über -0.68[106, 110] bis zu -0.72[107] für doppelsträngige DNA. Einzelsträngige DNA diffundiert wegen ihrer höheren Flexibilität schneller[111] und es wurde ein Skalierungsfaktor von -0.68[109] gefunden.

3.1.3.5. Diffusion in strukturierten Materialien

Wenn die Umgebung, in der das Teilchen diffundiert, nun aber nicht uniform, sondern strukturiert ist, ist die Diffusion im Prinzip komplizierter. Man kann aber die Bewegung der Teilchen im Ensemble durch eine reduzierte effektive Diffusionskonstante D_{eff} beschreiben.

Wird z. B. ein System aus periodischen Membranen im Abstand a und mit Permeabilität κ betrachtet, wird die effektive Diffusionskonstante beschrieben durch [112, 113]:

$$\frac{1}{D_{\mathrm{eff}}} = \frac{1}{D_0} + \frac{1}{\kappa \cdot a} \tag{3.1.12}$$

Diese Formel wird natürlich komplexer, wenn sich zwischen den Membranen unterschiedliche Materialien mit ungleicher Diffusivität befinden oder wenn man allgemeinere Geometrien betrachtet.[114, 115]

Ein anderes Beispiel sind poröse Materialien, bei denen viele kleine Kammern, in denen freie Diffusion möglich ist, durch kleine Durchgänge miteinander verbunden sind. Die effektive Diffusionskonstante, mit der sich dann eine Substanz in dem gesamten Material ausbreitet, ist[116]:

$$D_{\mathrm{eff}} = \frac{D_0 \cdot d}{r_d} \tag{3.1.13}$$

wobei d der Radius der Kontaktfläche und r_d der Radius der Tröpfchen ist.

3.1.4. Konkrete Lösungen der Diffusionsgleichung

Die Diffusionsgleichung kann für bestimmte Randbedingungen analytisch gelöst werden. So breitet sich eine punktförmige Konzentration aller Teilchen an einem Ort zum Zeitpunkt Null in einer unendlich großen Umgebung als gaußförmiges Konzentrationsprofil aus:

$$c(\vec{r},t) = \frac{N}{(4\cdot\pi\cdot D\cdot t)^{\frac{3}{2}}}\cdot\exp\left(\frac{-\vec{r}^2}{4\cdot D\cdot t}\right) \tag{3.1.14}$$

Eine allgemeinere Lösung der Diffusionsgleichung bekommt man durch eine Separierung der Variablen. So nimmt man an, dass die Lösung ein Produkt aus Funktionen, die nur von jeweils einer Dimension abhängig sind, ist.

3.1.4.1. Allgemeiner Ansatz in 3D

Durch den Ansatz

$$c(x,y,z,t) = X(x)\cdot Y(y)\cdot Z(z)\cdot T(t) \tag{3.1.15}$$

vereinfacht sich die Diffusionsgleichung dann zu:

$$X\cdot Y\cdot Z\cdot\frac{\partial}{\partial t}T = D\cdot T\cdot\left(Y\cdot Z\cdot\frac{\partial^2}{\partial x^2}X + X\cdot Z\cdot\frac{\partial^2}{\partial y^2}Y + X\cdot Y\cdot\frac{\partial^2}{\partial z^2}Z\right) \tag{3.1.16}$$

und als Lösung ergibt sich:

$$T(t) = A\cdot e^{-\lambda\cdot t} \tag{3.1.17}$$
$$X(x) = e^{\alpha\cdot x} \tag{3.1.18}$$
$$Y(y) = e^{\beta\cdot y} \tag{3.1.19}$$
$$Z(z) = e^{\gamma\cdot z} \tag{3.1.20}$$
$$\tag{3.1.21}$$

wobei Folgendes erfüllt sein muss:

$$-\frac{\lambda}{D} = \alpha^2+\beta^2+\gamma^2 \tag{3.1.22}$$

3.1.4.2. Kugelsymmetrische Probleme

Auch für den radialsymmetrischen Fall kann man die Separierung durchführen:

$$c(r,t) = R(r)\cdot T(t) \tag{3.1.23}$$

mit der vereinfachten Diffusionsgleichung:

$$R\cdot\frac{\partial}{\partial t}T = D\cdot T\cdot\frac{1}{r}\frac{\partial^2}{\partial r^2}(r\cdot R) \tag{3.1.24}$$

Als Lösung ergibt sich:

$$T(t) = A \cdot e^{-\lambda \cdot t} \tag{3.1.25}$$

$$R(r) = \frac{1}{r} e^{\alpha \cdot r} \tag{3.1.26}$$

wobei Folgendes erfüllt sein muss:

$$-\frac{\lambda}{D} = \alpha^2 \tag{3.1.27}$$

3.1.4.3. 2D-Rotationssymmetrie

Wenn die Diffusion im Raum nur zweidimensional ist, ergibt sich auch:

$$c(r,t) = R(r) \cdot T(t) \tag{3.1.28}$$

Nur die Differentialgleichung sieht etwas anders aus:

$$\begin{aligned} R \cdot \frac{\partial}{\partial t} T &= D \cdot T \cdot \frac{1}{r} \frac{\partial}{\partial r} \left(r \cdot \frac{\partial}{\partial r} R \right) \\ &= D \cdot T \cdot \left(\frac{1}{r} \frac{\partial}{\partial r} R + \frac{\partial^2}{\partial r^2} R \right) \end{aligned} \tag{3.1.29}$$

Für die Zeitabhängigkeit ergibt sich wieder die gleiche Form:

$$T(t) = A \cdot e^{-\lambda \cdot t} \tag{3.1.30}$$

aber die Differentialgleichung im Radius hat die Form der Besseldifferentialgleichung nullter Ordnung:

$$-\lambda \cdot R = D \cdot \left(\frac{1}{r} \frac{\partial}{\partial r} R + \frac{\partial^2}{\partial r^2} R \right) \Leftrightarrow r^2 \cdot \frac{\partial^2}{\partial r^2} R + r \cdot \frac{\partial}{\partial r} R + r^2 \cdot \frac{\lambda}{D} R = 0 \tag{3.1.31}$$

wobei die Besselfunktionen erster Gattung $J_0\left(\sqrt{\frac{\lambda}{D}} r\right)$ und zweiter Gattung $Y_0\left(\sqrt{\frac{\lambda}{D}} r\right)$ die beiden linear unabhängigen Lösungen dafür sind.

3.1.4.4. Beliebige Anfangsbedingungen

Um nun ein Anfangswertproblem mit gegebener beliebiger Anfangkonzentrationsverteilung zu lösen, kann man die verschiedenen Lösungen so addieren, bis sich die Anfangskonzentration zum Zeitpunkt $t = 0$ ergibt.

3.2. Strangreaktionen

Ein zentraler Punkt der Experimente sind Strangreaktionen, bei denen ein oder mehrere Nukleotidstränge miteinander reagieren. Bei diesen Reaktionen werden keine kovalenten Bindungen gebildet oder aufgebrochen, sondern nur die Wasserstoffbrückenbindungen und Basestackinginteraktionen der Basenbindungen (s. 2.2.2.1). Deswegen sind die Aktivierungsenergien, die zum Reaktionsablauf nötig sind, klein genug, um die Reaktionen bei Raumtemperatur (oder etwas darüber – z. B. 37 °C) zu ermöglichen. Hierbei werden nur kurze DNA-Stränge betrachtet ($N < 200$), da in den Experimenten auch nur so kurze Stränge verwendet werden.

3.2.1. Stranghybridisierung

Die einfachste Art, wie zwei DNA-Stränge miteinander reagieren können, ist die Hybridisierung. Es liegen also die beiden komplementären Stränge als Einzelstrang vor und treffen sich durch Diffusion. Nun müssen sie sich aneinander orientieren, so dass die richtigen Basenregionen miteinander reagieren können. Wenn sich zwei Regionen, die komplementär zueinander sind, gefunden haben, dienen diese als Nukleationskeime und die beiden Stränge hybridisieren von dieser Stelle in beide Richtungen wie ein Reißverschluss. Dabei ist die Rate, mit der sich ein Basenpaar ausbildet, im Bereich $k_f = 10^6 - 10^7\,\mathrm{s}^{-1}$.[117] Da die Basenpaarbildung im Vergleich zum Prozess des Suchens schnell ist[3], ist die Hybridisierungsrate nur sehr schwach von der Stranglänge abhängig und beträgt in etwa $k_a = 10^6\,\mathrm{M}^{-1}\,\mathrm{s}^{-1}$[118, 119].

Da einzelsträngige DNA recht flexibel ist und im Grunde genommen alle Basen miteinander hybridisieren können, kann es viele lokale Minima geben, in denen die Stränge nicht vollständig hybridisiert sind. So kann es passieren, dass die Einzelstränge an sich selbst anbinden oder die falschen Sequenzabschnitte miteinander hybridisieren. Um das globale Minimum in der Bindungsenergielandschaft zu erreichen, werden Stränge, die für die Experimente am Anfang schon doppelsträngig vorliegen müssen, über eine Temperaturrampe, die über der Schmelztemperatur (s. 3.2.2) der Bindung (meistens um die 90 °C) beginnt und langsam (über mindestens 2 h) unter der Schmelztemperatur (5 – 20 °C) endet, getempert.

[3] Die diffusive Zeitkonstante für ein Quadrat mit Kantenlänge 35 nm, was in etwa die Konturlänge einer 100 Bp langen DNA entspricht, ist ca. 65 µs für eine 100 Bp lange doppelsträngige DNA – und das ist erst die laterale Suche, da die Stränge danach auch noch entsprechen rotieren müssen. Um einen bei Raumtemperatur stabilen (ca. 20 Bp) Doppelstrang auszubilden, werden in etwa 10 µs benötigt.

3.2.2. Strangdissoziation

Der inverse Vorgang der kompletten Strangdissoziation ist stark von der Temperatur abhängig.[118] Somit existiert eine Temperatur, bei der im Gleichgewicht zwischen Hybridisierung und Dissoziation 50 % der Stränge dissoziiert vorliegen – die sogenannte Schmelztemperatur. Sie ist von der Bindungsenergie der Basenpaarungen (s. 2.2.2.1) und dem Salzgehalt in der Lösung abhängig.[47]

Die Dissoziationsrate lässt sich in Abhängigkeit der Schmelztemperatur T_m, der Enthalpie der Strangbindung ΔH, der Umgebungstemperatur T, der Bindungsrate eines Basenpaares k_f, der Stranglänge N und der Mindestlänge für eine stabile Strangbindung n ausdrücken[120]:

$$k_{\text{diss}} = 2k_f \left(N - n + 1\right) \exp\left(\frac{\Delta H \left(T - T_m\right)}{R \cdot T \cdot T_m}\right) \tag{3.2.1}$$

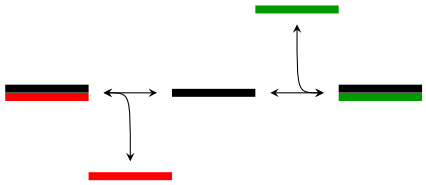
Aber ein DNA-Doppelstrang kann auch nur teilweise aufgeschmolzen vorliegen, so dass sich die beiden Einzelstränge noch nicht separieren können. So kann man davon ausgehen, dass sich bei Raumtemperatur die Enden des Doppelstrangs durch thermische Aktivierung ständig öffnen und schließen – man nennt dies „fraying“.

3.2.3. Strangverdrängung

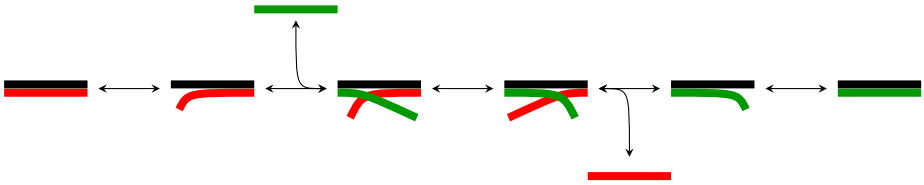
Als nächstes werden die Interaktionen zwischen einem DNA-Doppelstrang – seine beiden Stränge werden A und B genannt –, der schon so weit wie möglich hybridisiert ist, und einem Einzelstrang C, der komplementär zu A ist, betrachtet. Der Einzelstrang C hat also in einem Bereich exakt die gleiche Sequenz wie B. Es gibt nun Mechanismen, durch die Strang C den Strang B verdrängen kann, so dass am Ende der Doppelstrang AC vorliegt. Dieser Vorgang kann sequenziell oder parallel ablaufen. Die Rate, mit der der Komplex AB aufgelöst wird, ergibt sich aus der Summe der beiden Einzelraten:

$$\frac{d\left[\text{AB}\right]}{dt} = R_{\text{sequenziell}} + R_{\text{parallel}} \tag{3.2.2}$$

(a) Schema der sequenziellen Strangverdrängung



(b) Schema der parallelen Strangverdrängung



(c) Schema der irreversiblen, parallelen Strangverdrängung

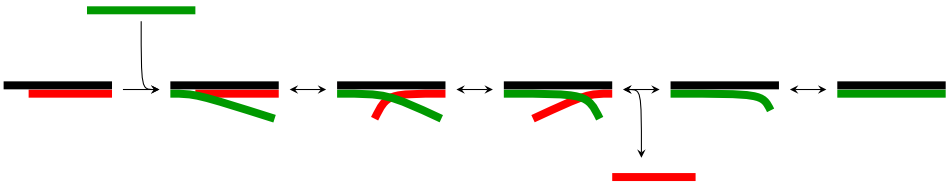


Abbildung 3.1.: Schemata der verschiedenen Verdrängungsmechanismen.

3.2.3.1. Sequenzielle Verdrängung

Die erste Möglichkeit ist, dass die beiden Prozesse – die Dissoziation des Stranges B und die Hybridisierung des Stranges C – hintereinander geschehen (s. Abb. 3.1a). Da die Dissoziationsrate unterhalb der Schmelztemperatur des AB-Komplexes um einiges langsamer ist als die Hybridisierung, wird dieser Vorgang durch k_{diss} dominiert.[120] Dabei ist aber zu beachten, dass nach der Dissoziation auch Strang B wieder hybridisieren könnte. B und C stehen dann in direkter Konkurrenz – welcher Strangtyp die höhere Hybridisierungswahrscheinlichkeit hat, hängt in erster Linie von der Konzentration der beiden Spezies ab. Wenn man also annimmt, dass C in einer viel höheren Konzentration als B vorliegt, ist die Rate der sequenziellen Strangverdrängung durch die Dissoziationsrate k_{diss} bestimmt:

$$R_{\text{sequenziell}} = -[\text{AB}] \cdot k_{\text{diss}} \tag{3.2.3}$$

3.2.3.2. Parallele Verdrängung

Bei der parallelen Verdrängung spielt das oben erwähnte partielle Aufschmelzen der Doppelstränge am Rand eine Rolle. So wird Komplex AB an den Enden oft ein paar ungebundene Basen besitzen. An diese Basen kann nun C anbinden und so weit eine Doppelhelix bilden, bis er zu der letzten gebundenen Base zwischen AB kommt. Dieser Treffpunkt kann jetzt wandern, indem einer der beiden Stränge eine Base öffnet und der andere diese im Gegenzug bindet. Da die Energie- und Entropieänderungen dabei sehr klein sind[4], kann dieser Vorgang durch eine zufällige, ungerichtete Wanderung beschrieben werden. Wenn nun der Treffpunkt weit an ein Ende des Stranges A gewandert ist und deswegen entweder B oder C nur noch so wenig Basenpaarungen besitzt, dass die Bindung nicht mehr stabil ist, fällt dieser Strang ab und der verbleibende Strang kann vollständig binden (s. Abb. 3.1b).

Die Rate, mit der Strang B verdrängt wird, kann in Abhängigkeit der Anzahl der Basenpaare, die am Anfang geöffnet werden müssen, n, der zugehörigen Bindungsenthalpie ΔH und Schmelztemperatur T_m, der Assoziationsrate k_a, der Stranglänge N und der Temperatur T beschrieben werden[120]:

$$R_{\text{parallel}} = -\left[\text{AB}\right] \cdot \left[\text{C}\right] \cdot \frac{2k_a}{N - 2n + 2} \exp\left(\frac{\Delta H\,(T - T_m)}{R \cdot T \cdot T_m}\right) \tag{3.2.4}$$

3.2.3.3. Irreversible Verdrängung

Wenn die Stränge B und C gleich viele Basenpaarungen mit A ausbilden können, ist die Situation komplett symmetrisch und wird nur durch die Konzentrationsdifferenzen in eine bestimmte Richtung getrieben. Wenn nun aber C an einem Ende von A so viele Basenpaarungen ausbilden kann, dass er stabil anbindet, ohne B verdrängen zu müssen, wird der Prozess in die Richtung AC getrieben. So ist die Dissoziationsrate durch die höhere Schmelztemperatur und Bindungsenthalpie von AC kleiner als die von AB und die sequenzielle Verdrängung findet stärker in Richtung AC statt.

Noch stärker ist der Effekt bei der parallelen Verdrängung, da der Strang C immer noch stabil binden kann, auch wenn B alle Basenpaarungen bildet. So ist die Wahrscheinlichkeit, dass Strang C „gewinnt“, viel höher (s. Abb. 3.1c). Auch muss der AB-Komplex nicht mehr partiell aufgeschmolzen werden. Die Verdrängungsrate ist durch die Assoziationsrate k_a bestimmt[121]:

$$R_{\text{parallel, irreversibel}} = -\text{AB} \cdot \text{C} \cdot k_a \tag{3.2.5}$$

[4]Sie sind nicht Null, da die Bindungsenergie auch immer von den Nachbarbasen abhängt und man davon ausgehen kann, dass zwischen den beiden Strängen immer ein paar Basen nicht gebunden sind. Außerdem ändert sich beim Binden einer Base die Länge des verbleibenden Einzelstrangs. Die dadurch entstehende Entropieänderung bei einem Strang muss nicht immer exakt der Entropieänderung beim anderen entsprechen.

Das Ende von A, an das nur C binden kann, wird dabei „Überhang“ oder englisch „toehold“ genannt und man bezeichnet den kompletten Vorgang mit „toehold-mediated strand displacement“[5].

Die obige Gleichung gilt aber nur, wenn die Dissoziationsrate des Überhangs kleiner ist als die Rate, mit der der Treffpunkt der drei Stränge von einer Seite zur anderen wandert.[122] Wenn die initiale Bindung zwischen A und B schwächer ist, ist die Verdrängungsrate kleiner und abhängig von der Bindungsenergie des Überhangs – der Logarithmus von $R_{\text{parallel, irreversibel}}$ ist linear. Diese Abhängigkeit kann durch eine genaue Analyse und Simulation der Zwischenstufen, die der Dreistrangkomplex durchläuft, erklärt und reproduziert werden.[123]

Bei der Auslegung der hier vorgestellten Experimente wurde darauf geachtet, dass alle Überhangsequenzen so lang sind, dass Gleichung (3.2.5) gilt.

3.2.3.4. Bemerkung zur Einstufenvereinfachung

Bei den obigen Diskussionen der Strangverdrängung wurde diese jeweils als ein einstufiger Prozess angenommen. Wie man aber in Abb. 3.1 erkennen kann, bestehen die Vorgänge jedoch aus mehreren Einzelschritten, die man nur unter bestimmten Bedingungen zusammenfassen kann. Generell kann man Reaktionen, die aus mehreren Schritten bestehen, dann als Einstufenvorgang beschreiben, wenn es einen Schritt gibt, der langsamer als alle anderen ist und damit die Gesamtkinetik dominiert. So ist im Fall der irreversiblen Strangverdrängung die Vereinfachung nicht korrekt, wenn die Konzentration von AB und/oder C über einem gewissen Schwellwert liegt. Dieser beträgt ca. 30 nM, wenn Gleichung (3.2.5) valide ist.[122]

Diese Konzentrationsgrenze wurde in den hier vorgestellten Experimenten teilweise überschritten. Die Modelle wurden aber trotzdem mit Einstufenprozessen beschrieben, um die Anzahl der Parameter möglichst gering zu halten. Auch zeigten stichprobenartige Einführungen von Zweistufenprozessen keine bessere Modellierung der Realdaten.

[5] Frei übersetzt: durch einen Ansatzpunkt vermittelte Strangverdrängung.

3.3. Partitionierung

Bei der Emulsifizierung werden die Stoffe, die in der Lösung enthalten sind, auf die entstehenden Tröpfchen verteilt. Wie dies genau geschieht, beeinflusst die Statistik der Molekülanzahl und damit die Variabilität der beobachteten Reaktionen in den Tröpfchen. Im Folgenden werden drei verschiedene Schemata, nach denen die Moleküle aufgeteilt werden können, beschrieben.

3.3.1. Unabhängige Teilchenaufteilung

Der erste Ansatz ist, dass die einzelnen Teilchen vollständig unabhängig voneinander verteilt werden. Um diesen Verteilungsprozess zu modellieren, werden die Stoffe molekülweise auf N Tröpfchen verteilt. Es ist ein Anfangsvolumen V_0 vorhanden und die Endtröpfchen[6] haben jeweils das Volumen V_t:

$$V_t = \frac{V_0}{N} \ll V_0 \tag{3.3.1}$$

Somit hat jedes Molekül die Wahrscheinlichkeit p ins Tröpfchen i zu gelangen:

$$p = \frac{V_t}{V_0} = \frac{1}{N} \ll 1 \tag{3.3.2}$$

Trotzdem ist die durchschnittliche Molekülanzahl λ bei einer Molekülkonzentration c in den Tröpfchen nicht Null (N_A ist die Avogadrokonstante):

$$\lambda = \langle n \rangle = N_A \cdot c \cdot V_t \tag{3.3.3}$$

Die Verteilungsfunktion der Molekülanzahlen in den Tröpfchen wird dann durch eine Poissonverteilung (s. A.1) beschrieben:

$$p\left(n = x\right) = \frac{\lambda^x}{x!} e^{-\lambda} \tag{3.3.4}$$

Eine Besonderheit dieser Verteilung ist, dass die Varianz den gleichen Wert wie der Mittelwert besitzt:

$$\mathrm{Var}\left(n\right) = \left\langle \left(n - \langle n \rangle\right)^2 \right\rangle = \lambda = \langle n \rangle \tag{3.3.5}$$

Somit ist das Verhältnis zwischen der Varianz und dem Mittelwert einer Verteilung ein Maß dafür, wie ähnlich sie einer Poissonverteilung ist.

[6] Es wird der Einfachheit halber angenommen, dass die Tröpfchen alle gleich groß sind.

3.3.2. Abhängige Teilchenaufteilung

Wenn die Verteilung der Moleküle nun aber nicht unabhängig voneinander ist, wird die Beschreibung etwas komplexer.

3.3.2.1. Volumeneffekte

Die einfachste Art der Abhängigkeit sind einfache Volumeneffekte: Jedes Molekül, das verteilt werden soll, hat ein Volumen von $V_{\text{Molekül}}$. Damit verringert jedes verteilte Molekül das freie Volumen und die Verteilungswahrscheinlichkeiten ändern sich. Insgesamt kann das Volumen V_0 die maximale Molekülanzahl M aufnehmen – man kann es also in M Teilvolumina, die von Molekülen besetzt werden können, einteilen:

$$M = \left\lfloor \frac{V_0}{V_{\text{Molekül}}} \right\rfloor \tag{3.3.6}$$

Wenn der Stoff mit einer Konzentration c vorliegt, befinden sich N_m Moleküle im Gesamtvolumen:

$$N_m = \lfloor N_A \cdot c \cdot V_0 \rfloor \tag{3.3.7}$$

Dadurch ist die Wahrscheinlichkeit, dass ein bestimmtes Teilvolumen überhaupt besetzt ist, gegeben durch:

$$p_{\text{besetzt}} = \frac{N_m}{M} \tag{3.3.8}$$

Äquivalent zum Gesamtvolumen hat ein Tröpfchen mit Volumen V_t insgesamt K Teilvolumina:

$$K = \left\lfloor \frac{V_t}{V_{\text{Molekül}}} \right\rfloor \tag{3.3.9}$$

Damit ist die Wahrscheinlichkeit, dass sich in dem Tröpfchen N_t Moleküle befinden bzw. N_t Teilvolumina besetzt sind:

$$p\left(N_t\right) = \binom{K}{N_t} \cdot p_{\text{besetzt}}^{N_t} \cdot \left(1 - p_{\text{besetzt}}\right)^{K - N_t} \tag{3.3.10}$$

Diese Binomialverteilung (s. A.2) hat den erwarteten Mittelwert von

$$\langle N_t \rangle = K \cdot p_{\text{besetzt}} \approx \frac{V_t}{V_{\text{Molekül}}} \cdot \frac{N_A \cdot c \cdot V_0}{\frac{V_0}{V_{\text{Molekül}}}} = N_A \cdot c \cdot V_t \tag{3.3.11}$$

und die Varianz

$$\begin{aligned} \text{Var}\left(N_t\right) &= K \cdot p_{\text{besetzt}} \cdot \left(1 - p_{\text{besetzt}}\right) = \langle N_t \rangle \cdot \left(1 - p_{\text{besetzt}}\right) \\ &\approx N_A \cdot c \cdot V_t \cdot \left(1 - N_A \cdot c \cdot V_{\text{Molekül}}\right) \end{aligned} \tag{3.3.12}$$

Die Varianz wird also um so kleiner, je mehr Volumen das Molekül im Vergleich zum Tröpfchen benötigt. Bei einem sehr kleinen Verhältnis zwischen $V_{\text{Molekül}}$ und V_t geht das System in den Fall unabhängiger Verteilung über und wird durch eine Poissonverteilung beschrieben.

In den Experimenten haben die Tröpfchen Radien zwischen 2 µm und 20 µm (s. Abb. 4.8) – V_0 liegt also im Bereich von $34 - 34\,000\,\mu m^3$. Das ist sehr viel größer als das Volumen der zu verteilenden Komponenten – DNA und Proteine –, die ein Volumen $V_{\text{Molekül}}$ von ca. $100\,nm^3$ haben. Somit können hierbei Volumeneffekte mögliche Abweichungen von einer Poissonverteilung nicht erklären. Erst etwa ab dem Volumen von E. coli – ca. $1\,\mu m^3$ – könnten Volumeneffekte merkliche Abweichungen hervorrufen.

3.3.2.2. Aggregatbildung

Eine andere Möglichkeit, wie eine abhängige Molekülverteilung erzeugt werden kann, ist die Aggregatbildung. Das heißt, die Moleküle liegen nicht separiert in der Lösung vor, sondern als lose Aggregate, die aber während der Aufteilung nur als Gesamteinheit verteilt werden können.

Die Anzahl von Aggregaten innerhalb eines Tröpfchens folgt also wie im unabhängigen Fall der Poissonverteilung und $p_A(n = x)$ sei die Wahrscheinlichkeit, dass ein Tröpfchen x Aggregate enthält. Innerhalb eines Aggregats hingegen kann die Anzahl der Moleküle sehr verschiedene Verteilungen annehmen[124, 125] – z. B. eine Exponentialverteilung (s. A.4 und B.3). So sei $p_a(n = x)$ die Wahrscheinlichkeit, dass ein Aggregat aus x Molekülen besteht. Die Wahrscheinlichkeit, dass sich eine bestimmte Anzahl von Molekülen in dem Tröpfchen befindet, kann folgendermaßen bestimmt werden:

Sei $\Omega(n)$ die Menge, die alle Zahlentupel $\in \mathbb{N}^{1 \text{ bis } n}$ enthält, die sich zu n addieren.[7] Wenn nun ω ein Tupel aus dieser Menge ist, sei $|\omega|$ die Anzahl der Zahlen im Tupel. Damit lässt sich die gesuchte Wahrscheinlichkeit wie folgt darstellen:

$$p(n = x) = \sum_{\omega \in \Omega(x)} \left(p_A(n = |\omega|) \cdot \prod_{i \in \omega} p_a(n = i) \right) \tag{3.3.13}$$

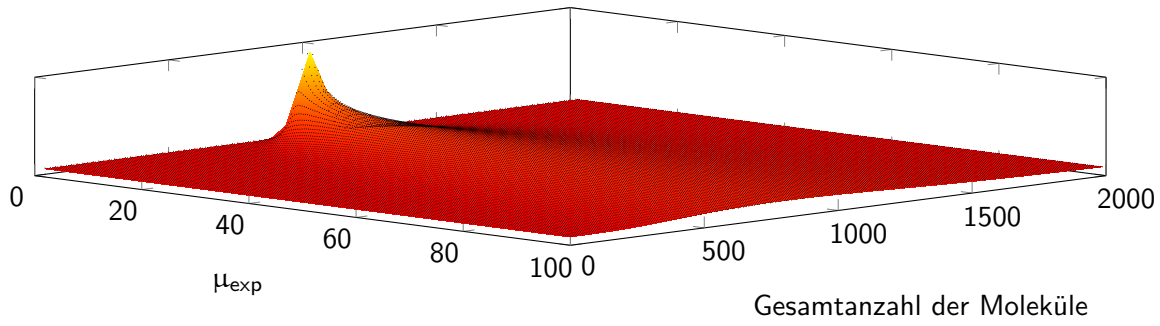
Da dieser Ausdruck extrem komplex ist, wird die Verteilung mit der Annahme, dass p_a eine Exponentialverteilung ist, mit verschiedenen Mittelwerten der Aggregatsgröße μ_{exp} simuliert:

$$p_a(n = x) = \begin{cases} \dfrac{1}{\mu_{\text{exp}} - 1} \cdot \left(1 - \dfrac{1}{\mu_{\text{exp}}}\right)^x & x \in \mathbb{N} \\ 0 & \text{sonst} \end{cases} \tag{3.3.14}$$

Die mittlere Anzahl von Einzelmolekülen in einem Tröpfchen μ_{ges} wurde auch verändert und dazu wurde für die Poissonverteilung der Aggregtsanzahl $\lambda = \frac{\mu_{\text{ges}}}{\mu_{\text{exp}}}$ verwendet.

[7] Beispiele: $\Omega(1) = \{(1)\}$, $\Omega(2) = \{(2), (1,1)\}$, $\Omega(3) = \{(3), (2,1), (1,2), (1,1,1)\}$

(a) Verteilungsfunktion der simulierten Daten für $\mu_{\text{ges}} = 1000$ und mit einem Binning mit Größe 20. Die farbige Fläche ist ein Fit mit einer Gammaverteilung, wobei die Varianz als einziger freier Parameter zu $\beta = 2.03 \cdot (\mu_{\text{exp}} - 1) + 1$ angefittet ist.



(b) Stochastische Parameter der simulierten Verteilungsfunktion.

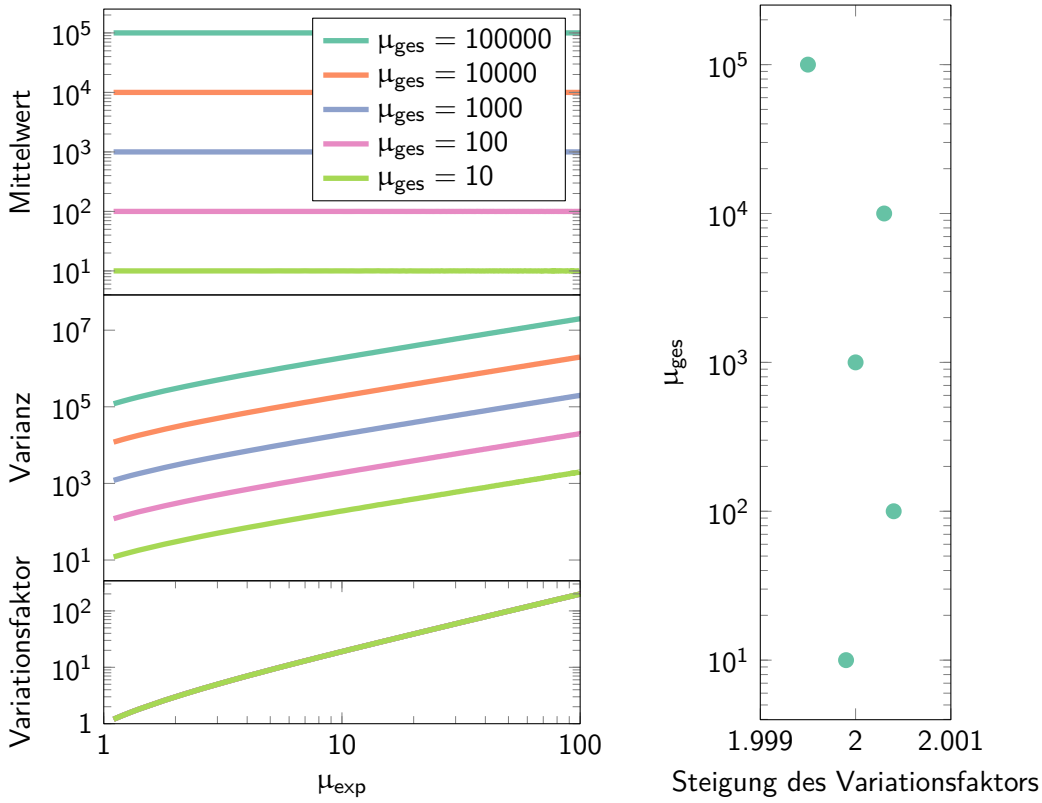


Abbildung 3.2.: Simulation der Molekülanzahlverteilung durch Aggregation. Die Aggregatsgröße folgte dabei einer Exponentialverteilung mit Mittelwert μ_{exp}. Die Population betrug 10^6 Tröpfchen.

Die Ergebnisse sind in Abb. 3.2 dargestellt. So sieht man, dass der Mittelwert den gewünschten Wert annimmt. Die Varianz σ^2 der Verteilung ist linear von μ_{exp} abhängig und die Betrachtung des Variationsfaktors[8] zeigt, dass die Abhängigkeit folgender Gesetzmäßigkeit folgt:

$$\sigma^2 = (a \cdot (\mu_{\text{exp}} - 1) + 1) \cdot \mu_{\text{ges}} \tag{3.3.15}$$

Ein Fit an die Daten ergibt ein $a = 2$ und damit:

$$\sigma^2 = (2 \cdot \mu_{\text{exp}} - 1) \cdot \mu_{\text{ges}} \tag{3.3.16}$$

Auch der Grenzfall der durchschnittlichen Aggregatsgröße $\mu_{\text{exp}} = 1$ – wenn also keine Aggregate vorliegen und die Verteilungsfunktion eine Poissonverteilung ist – wird durch diesen Fit abgedeckt und die Daten konvergieren korrekt.

Wenn man nicht nur Mittelwert und Varianz der Verteilungen, sondern die komplette Verteilungsfunktion bei einem festen μ_{ges} betrachtet (s. Abb. 3.2a), sieht man, dass die Verteilung gut durch eine Gammaverteilung (s. A.3) angenähert werden kann, solange μ_{ges} mindestens eine Größenordnung größer als μ_{exp} ist. Anderenfalls wird der Anteil der Tröpfchen, die kein einziges Molekül erhalten, von der Gammaverteilung nicht richtig erfasst.

Breitere Verteilungen als Poisson sind in der Zellbiologie durchaus bekannt.[126] So wird auch die Zell-zu-Zell-Variation von Proteinkonzentrationen in Experimenten und Modellen oft durch eine Gammaverteilung[9] beschrieben.[129, 130] Dabei ist aber noch nicht vollständig geklärt, ob die experimentell beobachteten Fluktuationen durch stochastische Genexpression oder Partitionierungseffekte entstehen.[127, 131, 132]

Das hier vorgestellte Modell der Aggregatsaufteilung zeigt aber grundlegende Unterschiede zu diesen etablierten Modellen. So findet zwischen den Teilungsprozessen keine Proteinproduktion statt, sondern die Anzahl der zu verteilenden Moleküle ist konstant. Auch Partitionierungsmodelle, die Proteine über Vesikel verteilen und somit ähnlich zur Aggregatsverteilung erscheinen,[131] können nicht angewendet werden, da die Prozesse, die die Proteine auf die Vesikel verteilen, andere stochastische Eigenschaften besitzen als die Aggregatbildung. Zusätzlich kann solch ein Vesikel auch kein einziges Protein enthalten, was in der Aggregatsinterpretation keinerlei Sinn ergibt. Das hier vorgestellte Verteilungsmodell von Aggregaten kann aber auch in Situationen, in denen keine Proteinproduktion oder klassische Zellteilung vorliegt, Verteilungsfunktionen erklären, die einen starken Unterschied zwischen Varianz und Mittelwert aufweisen.

[8] Der Variationsfaktor ist das Verhältnis zwischen Varianz und Mittelwert: $\phi = \frac{\sigma^2}{\mu}$. Da bei der Poissonverteilung (s. A.1) dieser einen Wert von 1 besitzt, ist er ein Maß dafür, wie „Poisson-ähnlich“ eine Verteilung ist – hier also ein Maß, wie unabhängig die Aufteilung ist.

[9] Manchmal wird auch eine negative Binomialverteilung[127] oder eine logarithmische Normalverteilung[128] verwendet. Diese unterscheiden sich aber nur in Details von der Gammaverteilung und sind oft austauschbar.

4. Experimente

Es wurden drei verschiedene Experimente unter dem gemeinsamen Aspekt „Tröpfchen“ durchgeführt. Im Rahmen dieser Arbeiten entstand die Notwendigkeit, experimentell gewonnene Videos auszuwerten. Das existierende Programm CellEvaluator[133], das am Anfang zur Auswertung der Mikroskopvideos verwendet wurde, stellte sich als zu langsam[1] und ungeeignet für die speziellen Anforderungen der Experimente[2] heraus. Deswegen wurden nicht nur die experimentspezifischen Auswertungsalgorithmen entwickelt, sondern auch eine Software für die allgemein Analyse von Tröpfchenvideos.

4.1. Design

Die drei Experimente, an denen die Analysemethoden entwickelt, verfeinert und angewendet wurden, sind in chronlogischer Reihenfolge:

- Ein DNA/RNA-Oszillator[134] wurde in den kleinen Volumina von Emulsionströpfchen untersucht. Dabei war vor allem die Abhängigkeit des Oszillationsverhaltens von der Tröpfchengröße von Interesse.[135][3]
- Bakterien, die auf ein oder zwei amphiphile Chemikalien reagieren, wurden in Emulsionströpfchen gebracht und die Diffusion der Chemikalien zwischen den Tröpfchen untersucht.[136][3]
- Mittels einer einfachen Transkriptionsreaktion wurden die stochastischen Eigenschaften der Partitionierung näher untersucht und quantifiziert. Diese Experimente wurden als Fortsetzung der Oszillatorexperimente geplant um nähere Erkenntnisse über Partitionierungseffekte zu erlangen.[137]

Nach ihrem Hauptthema werden diese drei im Folgenden über die Stichworte „Oszillator“, „Bakterien“ und „Stochastik“ referenziert.

[1] Allein die Datenextraktion aus den Videos dauerte bis zu einen Tag und manche Videos konnten nur ausschnittsweise analysiert werden.

[2] Man konnte z. B. nicht festlegen, dass ein Tröpfchen zwischen zwei Videobildern seine Größe nicht ändert.

[3] Die Darstellungen, Ideen und Bilder der beiden schon veröffentlichen Experimente sind natürlich auch in dieser oder ähnlicher Form in den Veröffentlichungen vorhanden. Um eine ständige Wiederholung der Referenz zu vermeiden, wird hier einmalig darauf verwiesen.

4.1.1. Oszillator in Tröpfchen

Die unten näher beschriebene Reaktionslösung, die ein oszillatorisches Verhalten zeigen kann und im Folgenden „Oszillator“ genannt wird, wird emulsifiziert. Bei der Partitionierung in die kleinen Kompartimente können die einzelnen Komponenten verschieden aufgeteilt werden. Diese Konzentrationsvariationen können das Verhalten des Oszillators beeinflussen. Die beobachtete Variabilität lässt dann Rückschlüsse auf die Art und Weise zu, wie die einzelnen Moleküle auf die Kompartimente aufgeteilt werden.

Um das Verhalten des Oszillators ohne Aufteilungseffekte zu bestimmen, wird die Reaktionslösung als Referenz zusätzlich in einem makroskopischen Volumen betrachtet.

4.1.1.1. Funktionsweise

Im Grunde besteht der Oszillator aus zwei Schaltern – SW_{12} und SW_{21} –, die durch DNA-Doppelstränge implementiert sind.[4] Die Schalter dienen als Templat für eine RNA-Polymerase[5] und können jeweils einen angeschalteten Zustand – hohe RNA-Transkriptionsrate – und einen ausgeschalteten Zustand – niedrige RNA-Transkriptionsrate – einnehmen. Die beiden DNA-Schalter sind in einer negativen Rückkopplungsschleife miteinander verbunden, indem die RNA-Spezies rA_1, die in SW_{12} kodiert ist, SW_{21} aktiviert und die RNA-Spezies rI_2 welche anhand von SW_{21} produziert wird, SW_{12} deaktiviert (s. Abb. 4.1). Dadurch können bei geeigneter Parameterwahl Oszillationen entstehen.

Die konkrete Implementierung ist komplexer (s. Abb. 4.2). Die beiden schaltbaren DNA-Komplexe bestehen zum einen aus einem DNA-Doppelstrang, der am Anfang einen nicht

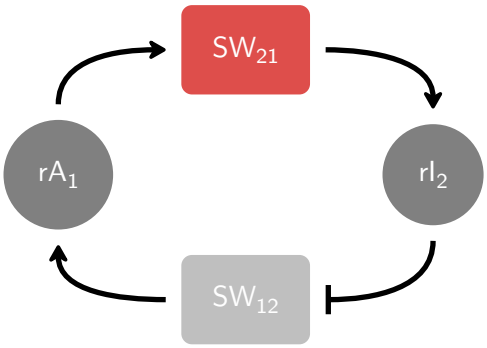


Abbildung 4.1.: Einfaches Oszillatorschema. SW_{12} und SW_{21} sind DNA-Konstrukte, welche die rA_1- bzw. rI_2-RNA kodieren. Diese RNA-Moleküle können die Transkription der jeweils anderen DNA an- bzw. ausschalten.

[4] SW steht für englisch Switch. Die erste Zahl steht für das Ausgangssignal und die zweite Zahl für das Eingangssignal, das den Schalter kontrolliert.
[5] Es wird die T7-RNA-Polymerase verwendet (s. 2.3.1.1).

vollständigen T7-Promotor hat. Das heißt, dass der sense-Strang die vollständige Promotorsequenz enthält, aber der antisense-Strang den Promotor nur teilweise zum Doppelstrang ergänzt. Dieser Doppelstrang heißt T_{12} für SW_{12} bzw. T_{21} für SW_{21}. Um den Schalter zu aktivieren, muss an den sense-Strang ein sogenannter Aktivator (A_2 für SW_{12} und A_1 für SW_{21}) binden, der den Promotor dann vervollständigt. Damit der Aktivator stabil anbindet, ist der sense-Strang vor der Promotorsequenz noch um 22 Nukleotide verlängert. Um den Schalter auch wieder deaktivieren zu können, ist der Aktivator länger als die zu bindende Sequenz auf dem sense-Strang. An diesen Überhang von ca. 20 Nukleotiden kann dann ein passender Nukleotidstrang anbinden und den Aktivator durch irreversible Strangverdrängung (s. 3.2.3.3) ablösen. Bei SW_{12} kann direkt das RNA-Produkt von SW_{21} den Aktivator entfernen und somit SW_{12} deaktivieren. Doch bei der Aktivierung von SW_{21} durch rA_1 muss noch ein Zwischenschritt eingeführt werden: neben dem Aktivator A_1 existiert ein weiterer DNA-Strang (dI_2), der den Aktivator von T_{21} ablösen kann. Durch das RNA-Produkt von SW_{12} (rA_1), das an den Inhibitor dI_1 bindet, wird der Aktivator freigesetzt und aktiviert SW_{21}.

Um das System dynamisch reversibel zu machen, dürfen sich die RNA-Produkte nicht in der Lösung ansammeln, da sonst nach einer gewissen Zeit alle A_2 und dI_1 in einem DNA-RNA-Hybridkomplex vorliegen. Deswegen wird eine RNase, die bei diesen Hybridsträngen die RNA abbaut – die RNase H (s. 2.3.2.1) –, hinzugefügt.

Es werden also zwei Proteine und sieben DNA-Einzelstränge für die Experimente benötigt.

4.1.1.2. Auslese

Um den Zustand des Oszillators auslesen zu können, wird der sense-Strang des T_{21} mit einem fluoreszenten Farbstoff (Texas Red) am 5'-Ende versehen. Dieser leuchtet nur, wenn SW21 inaktiv ist, denn am 3'-Ende vom Aktivator A_1 ist ein Quencher (Iowa Black RQ) befestigt, der die Fluoreszenz reduziert, wenn er sich in der Nähe des Farbstoffes befindet – wenn also A_1 an T_{21} gebunden ist. Auch der Aktivator A_2 besitzt einen Quencher, der aber keine Funktion erfüllt, sondern nur aus dem Originaldesign[134] noch vorhanden war.

Zusätzlich wird in die Tröpfchen ein Referenzfarbstoff (Alexa 488) gegeben, um die Fluoreszenzdaten normieren zu können.

4.1.1.3. Stimmungen

Wie jeder Oszillator kann auch dieser Oszillator in verschiedenen Stimmungen betrieben werden. So soll er mit drei verschiedenen Konzentrationsverhältnissen zwischen der RNA-Polymerase und der RNase untersucht werden:

Stabile Oszillation: Die Mischung, die im makroskopischen Experiment die stabilsten und am längsten andauernden Oszillationen zeigt, wird als Referenzpunkt verwendet.

Gedämpfte Oszillation: Etwas abseits der idealen Parameter werden gedämpfte Oszillationen beobachtet.

Stark gedämpfte Oszillation: Weit außerhalb des Parameterbereichs mit stabilen Oszillationen sind in makroskopischen Messungen durch die starke Dämpfung nur kurz Oszillationen erkennbar.

Durch die Partitionierung in die Tröpfchen entstehen Variationen der Konzentrationsverhältnisse. Diese wirken sich dann unterschiedlich auf das Oszillatorverhalten aus. So wird z. B. erwartet, dass bei der stark gedämpften Stimmung trotzdem ein paar Tröpfchen stabile Oszillationen zeigen.

4.1.1.4. Tröpfchen

Eine einfache Methode, kleine Volumina zu erzeugen, ist das Herstellen einer Emulsion aus einem großen Anteil Öl und ein wenig wässriger Reaktionslösung. Dabei bilden sich Wasser-in-Öl-Tröpfchen, die durch einen Emulgator (s. 2.5) stabilisiert werden müssen, damit sie während des Experiments nicht verschmelzen. Um eine möglichst große Variation des Oszillatorverhaltens innerhalb einer Stimmung beobachten zu können, wird dabei eine möglichst breite Verteilung der Tröpfchengrößen benötigt, die auch kleinen Radien beinhaltet.

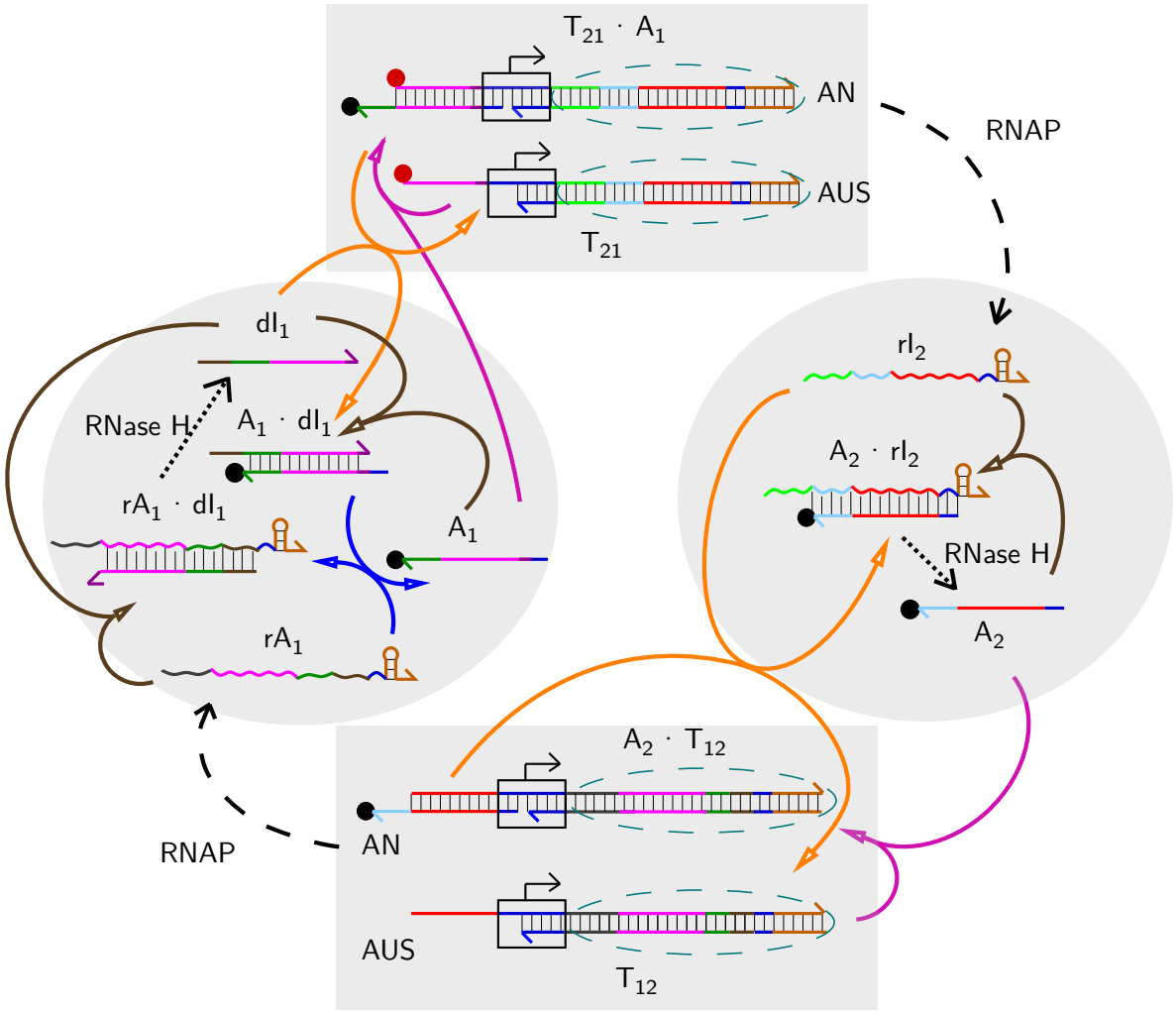


Abbildung 4.2.: Reaktionsschema des Oszillators. Die geraden Linien symbolisieren DNA-Stränge, die gewellten RNA-Stränge. Die gestrichelten Pfeile repräsentieren die Transkriptionen der RNA aus den aktivierten Schaltern und die gepunkteten stellen die Degradation der RNA aus den Hybridsträngen dar. Wenn zwei Pfeile zusammenlaufen oder sich berühren, symbolisieren sie zusammen eine Strangreaktion – eine Hybridisierungs- und Strangverdrängungsreaktion. Die orangen Pfeile stehen für die Deaktivierung der Schalter und die pinken für die Aktivierung. Braune Pfeile symbolisieren einfache Hybridisierungsreaktionen von einzelsträngig vorliegenden Strängen. Der Ablösevorgang des Aktivators A_1 ist mit blauen Pfeilen dargestellt. Schwarze Kreise symbolisieren den Quencher (IBRQ) und rote den Farbstoff (Texas Red).

4.1.2. Bakterien in Tröpfchen

Sowohl AHL als auch IPTG (s. 2.4.3) haben hydrophobe und hydrophile Teilbereiche, wodurch sie sowohl in Wasser als auch in Öl löslich sind. Somit können diese beiden Signalmoleküle verwendet werden, um Kommunikation zwischen Bakterienkolonien in separierten Wasser-in-Öl-Tröpfchen zu realisieren.

Als Teil des Quorum-Sensing-Systems (s. 2.4.3.2) kann die Kommunikation zwischen Tröpfchen mittels AHL in zwei Varianten erfolgen. Zum einen können AHL-gefüllte Tröpfchen und Tröpfchen, die AHL-empfindliche Bakterien enhalten, untersucht werden. Zum anderen können die Tröpfchen, die AHL enthalten, durch Tröpfchen mit AHL-produzierenden Bakterien ersetzt werden. Da IPTG hingegen nicht von Bakterien synthetisiert werden kann, wird es nur in Reservoirtröpfchen bereitgestellt. Um ein gleichmäßiges Arrangement der Tröpfchen zu erhalten, werden gleich große Tröpfchen benötigt.

Zuerst wird untersucht, wie die Signalstoffe AHL und IPTG zwischen den Tröpfchen diffundieren. Dazu werden Tröpfchen produziert, die schon eine bestimmte Signalstoffkonzentration enthalten. Diese werden dann mit Tröpfchen gemischt, in denen sich Bakterien mit einem Quorum-Sensing- bzw. Lac-Repressor-System befinden. Über die zeitliche Antwort der Bakterien in Abhängigkeit von ihrem Abstand zu den Signalstoff-Tröpfchen lässt sich dann die Diffusionsgeschwindigkeit des Signalstoffs in der Emulsion modellieren.

Um dann die Kommunikation zwischen zwei verschiedenen Bakterientypen zu untersuchen, werden Senderbakterien, die AHL produzieren, in eine Tröpfchensorte gebracht. Die in den AHL-Diffusionsexperimenten verwendeten Bakterien werden in andere Tröpfchen gegeben und ihre Antwort beobachtet.

Die beiden Signalstoffregulationsmechanismen können auch kombiniert werden und ein System bilden, in dem die Bakterien nur dann ein Signal produzieren, wenn beide Stoffe in ausreichender Konzentration vorhanden sind. Dieses Verhalten kann man als UND-Gatter interpretieren. Dazu werden Tröpfchen, die Bakterien mit einen entsprechenden genetischen Schaltkreis enthalten, erzeugt und mit AHL- und/oder IPTG-Tröpfchen gemischt; nur wenn beide Signalstofftröpfchensorten vorhanden sind, wird eine Antwort der Bakterien erwartet.

4.1.2.1. Plasmid-Design

Für die Experimente werden demnach vier verschiedene Bakterientypen benötigt: AHL-Empfänger, IPTG-Empfänger, AHL-Sender und UND-Gatter (s. Abb. 4.4). Die genauen Konstrukte der Plasmide sind in Abb. 4.5 dargestellt und die Sequenzen können in D.3 nachgeschlagen werden.

Die **AHL-Empfänger** sollen GFP (s. 2.3.4.1) in Abhängigkeit der AHL-Konzentration produzieren. Das Fluoreszenzsignal der GFP ist damit ein Maß für die Aktivierung durch das AHL. Dafür gibt es schon ein fertiges BioBrick-Bauteil (BBa_T9002), das sowohl die

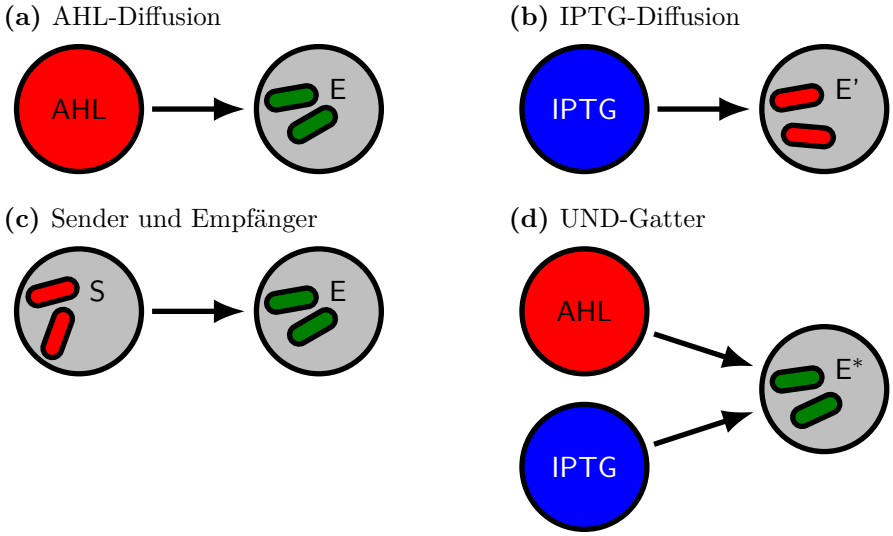


Abbildung 4.3.: Schemata der vier Bakterienexperimente. Kreise symbolisieren Tröpfchen und Stäbchen Bakterien.

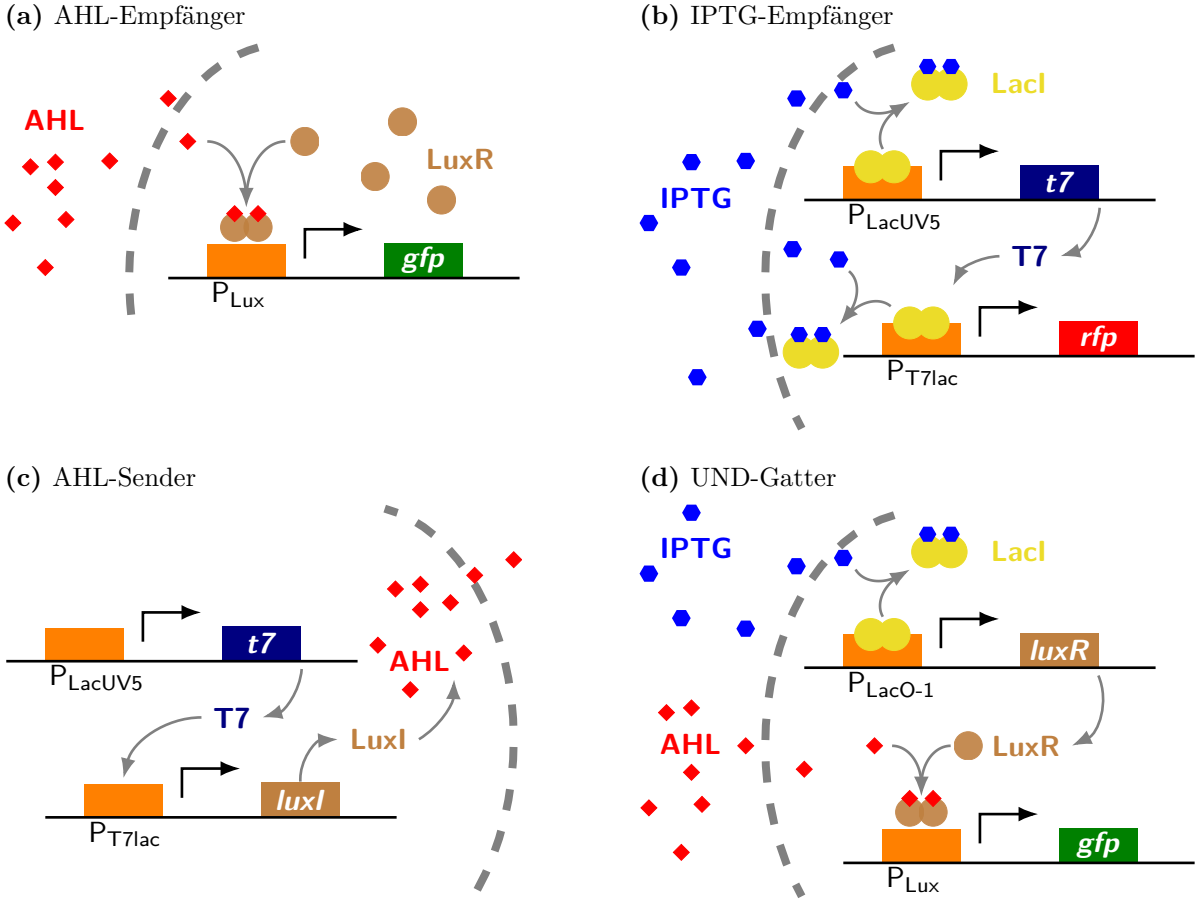
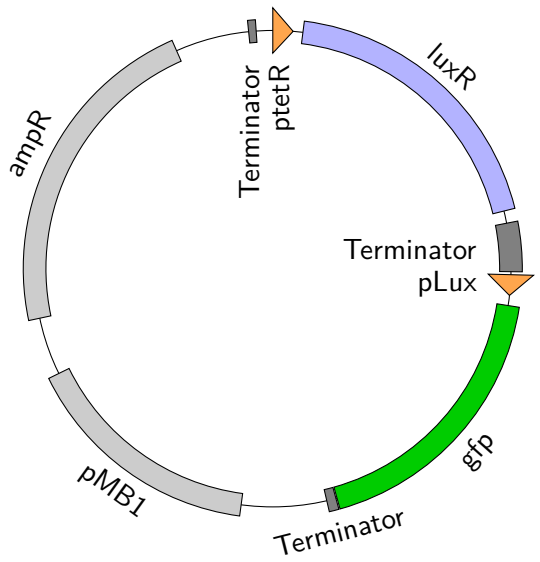
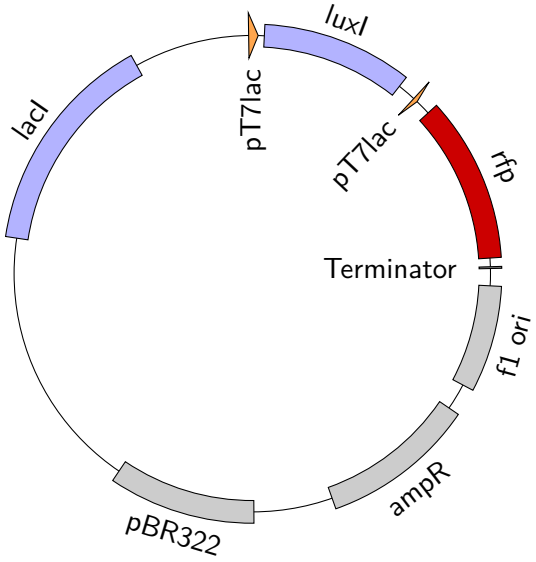


Abbildung 4.4.: Schemata der vier Bakterientypen.

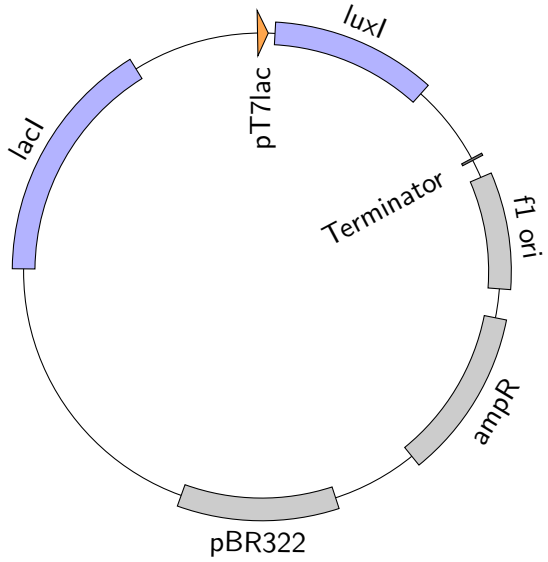
(a) AHL-Empfänger: pSB1A3 – 3960 Bp



(b) IPTG-Empfänger: pETDuet-1 – 6570 Bp



(c) AHL-Sender: pETDuet-1 – 5986 Bp



(d) UND-Gatter: pSB1A3 – 3964 Bp

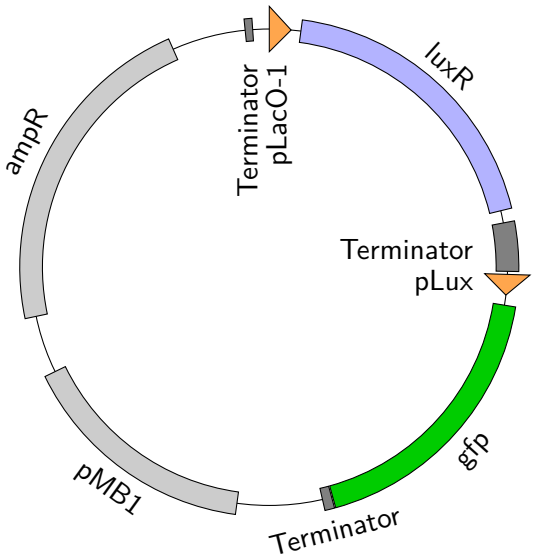


Abbildung 4.5.: Plasmidkarten. Die orangen Dreiecke stellen Promotoren dar und die dunkelgrauen Kreissegmente RNA-Polymerase-Terminatoren. Rote und grüne Segmente symbolisieren die Plasmidbereiche, die die fluoreszenten Proteine kodieren, und blaue die Proteine, die zum Quorum-Sensing- bzw. Lac-Repressor-System gehören. Sequenzen, die schon Teil des Originalvektors sind und dessen Funktion garantieren, sind in den hellgrau eingefärbten Bereichen gespeichert.

basale Produktion des LuxR als auch die Produktion von GFP in Abhängigkeit eines Lux-Promotors enthält. Dieser BioBrick wird in den pSB1A3-Vektor gebracht und in E. coli BL21(DE3)pLysS transformiert.

Um in den **AHL-Sendern** den Signalstoff zu produzieren, wird das Gen für LuxI (BioBrick BBa_C0261) in den Vektor pETDuet-1 kloniert. Dieser Vektor ist so aufgebaut, dass direkt vor dem eingefügten BioBrick ein T7-Promotor unter der Kontrolle von IPTG liegt. Dadurch wird die AHL-Produktion durch IPTG steuerbar. Der verwendete E. coli-Stamm (BL21(DE3)pLysS) stellt die T7-RNA-Polymerase selbst bereit, weswegen keine weitere Veränderung an Plasmid oder Bakterium durchgeführt werden muss.

Das **IPTG-Empfänger**-Plasmid ist eine Variation des AHL-Senders. So wird hinter das Gen für LuxI ein weiterer IPTG-induzierbarer T7-Promotor mit dem Gen für mRFP (BBa_E1010) platziert. Da die T7-RNA-Polymerase auch in Abhängigkeit von IPTG produziert wird, ist die Produktion des fluoreszierenden Proteins doppelt von IPTG abhängig.

Um das **UND-Gatter** zu realisieren, wird das AHL-Empfänger-Plasmid so verändert, dass die Produktion des LuxR unter der Kontrolle eines Lac-Promotors liegt. Dadurch kann der Lux-Promotor nur aktiviert werden, wenn vorher eine Aktivierung des Lac-Promotors durch AHL stattgefunden hat. Hierbei wird der Bakterienstamm DH5αZi verwendet.

4.1.3. Stochastik in Tröpfchen

Um die stochastischen Eigenschaften der Molekülverteilungen genauer zu untersuchen, wird ein weniger kompliziertes System als der Oszillator verwendet. Dieses einfache Transkriptionssystem (s. Abb. 4.6) besteht nur aus drei verschiedenen DNA-Strängen und der T7-RNA-Polymerase.

Der eine DNA-Strang ist ähnlich wie ein „molecular Beacon“[138] aufgebaut und dient als Reporter. So besitzt er am 3’-Ende einen Fluorophor (TAMRA) und am 5’-Ende einen Quencher (BHQ-2) und ist partiell selbstkomplementär. Dadurch kann er an sich selbst hybridisieren und den Quencher nahe am Farbstoff positionieren. Im Vergleich zu einem „molecular Beacon“ ist aber der selbstkomplementäre Teil um einiges länger – 12 anstatt 5 Basenpaare –, die Schleife, durch die sich der Strang auf sich selbst zurückfaltet, ist kürzer – 6 anstatt 15 Basen — und am 5’-Ende besitzt er eine Verlängerung, die als Ansatzpunkt für eine irreversible Strangverdrängung dienen kann (s. 3.2.3.3). Durch diese Struktur liegen Fluorophor und Quencher in gleicher Menge in der Lösung vor und im Anfangszustand ist die Fluoreszenz des Reporters reduziert.

Die anderen beiden DNA-Stränge bilden einen Templatdoppelstrang für die T7-RNA-Polymerase – er beginnt also mit dem T7-Promotor (s. 2.3.1.1). Die Sequenz des produzierten RNA-Strangs ist dabei so gewählt, dass sie teilweise komplemetär zum Reporter ist. Wenn diese RNA dann über irreversible Strangverdrängung an den Reporter bindet,

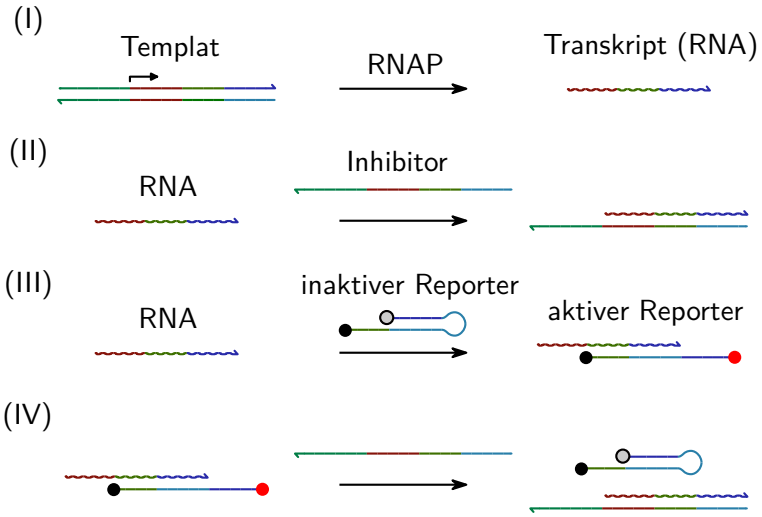


Abbildung 4.6.: Schema der Reaktionspfade für das Stochastikexperiment. Die farbigen geraden Linien symbolisieren DNA-Stränge, die gewellten RNA-Stränge. Der schwarze Kreis stellt den Quencher (BHQ-2) und der graue bzw. rote Kreis den Farbstoff (TAMRA) im gequenchten bzw. fluoreszierenden Zustand dar.

wird der Abstand zwischen Fluorophor und Quencher vergrößert und die Fluoreszenz steigt.

Um am Anfang des Experiments für Vorbereitungen Zeit zur Verfügung zu haben, wird der antisense-Strang im Überschuss zur Lösung gegeben. An diesen bindet die produzierte RNA stärker, da sie mehr Basenpaarungen ausbilden kann. Dies bewirkt zusätzlich, dass freie antisense-Stränge auch RNA-Reporter-Komplexe auflösen. So wird gewährleistet, dass die RNA zuerst an den Überschuss-Strang (Inhibitor) bindet und erst anschließend an den Reporter.

Die Reaktionslösung wird dann in möglichst verschieden große Tröpfchen gebracht, um die Radiusabhängigkeit der Verteilungsfunktionen der einzelnen Reaktionskomponenten zu untersuchen. Zusätzlich wird die Konzentration des DNA-Templates variiert, um den Einfluss der verschiedenen Reaktionsbestandteile trennen zu können. Auch hier wird die Reaktion zeitgleich in einem makroskopischen Volumen beobachtet, um die Reaktion ohne stochastische Effekte beurteilen zu können.

4.2. Geräte

4.2.1. Mikroskop

Die Emulsionen wurden unter einem Mokroskop betrachtet und Videos des Durchlicht- und Fluoreszenzbildes aufgenommen, um die zeitliche Entwicklung zu untersuchen.

4.2.1.1. Grundgehäuse

Es wurden zwei verschiedene Mikroskope von Olympus verwendet: ein IX71 und ein IX81. Diese sind vom Grundaufbau identisch und unterscheiden sich nur in ihrer Ausstattung und Ansteuerung. Für die Aufnahmen wurde ein Objektiv mit zehnfacher Vergrößerung und chromatischer Korrektur (Olympus UIS2 10x UPlanSApo) verwendet.

Das **IX71** ist ab Werk ein rein manuelles Mikroskop. Um aber Langzeitmessungen durchführen zu können, wurde eine Automatisierung hinzugefügt. So wurden ein automatischer Filterwürfelwechsler (Olympus IX2-RFACA), eine motorisierte Probenhalterung (Prior Scientific), eine fernsteuerbare Beleuchtung (Durchlicht-LED von Prior und ein vierfach Fluoreszenz-LED von Thorlabs (LED4D067 zusammen mit DC4104)) und ein Fokussiermotor (Prior PS3H122) ergänzt. Dadurch war das Mikroskop fast vollständig[6] über einen Computer fernsteuerbar und die Datenaufnahme konnte automatisiert durchgeführt werden.

Hingegen ist das **IX81** von Anfang an automatisiert. Als Lichtquellen dienten eine Quecksilberdampflampe (Olympus) für das Durchlicht und eine X-cite series 120 für das Fluoreszenzlicht. Da diese Lampen nicht automatisiert ein- und ausgeschaltet werden konnten, musste die Lichtzufuhr über automatisierte Shutter geregelt werden.

4.2.1.2. Filter

Um die Probe mit der richtigen Wellenlänge zu beleuchten und dann nur das Fluoreszenzlicht auf die Kamera zu senden, wurden Filterwürfel (Schematischer Aufbau s. Abb. 2.2b) verwendet – für jeden betrachteten Kanal ein separater Würfel (s. Tab. 4.1). Dadurch konnten die verschiedenen Aufnahmekanäle – Durchlicht und bis zu drei Fluoreszenzsignale – getrennt voneinander aufgenommen werden.

4.2.1.3. Kameras

Die Videos der Tröpfchen unter dem Mikroskop wurden von Kameras, die Graustufenbilder aufnehmen konnten, aufgezeichnet. Aus technischen und organisatorischen Gründen wurde für jedes Experiment eine andere Kamera – CoolsnapHQ, iXon3 und LucaR (s. Tab. 4.2) – verwendet.

[6] Die Objektive konnten nicht automatisch gewechselt werden, was aber auch nicht nötig war.

Tabelle 4.1.: Eigenschaften der Filterwürfel

Kanal	Mikroskop	Beleuchtung	Anregungsfilter	Dichroid	Emissionsfilter
Durchlicht	IX81	weiß	undurchlässig	—	—
GFP	IX81	weiß	457 – 487 nm	495 nm	503 – 538 nm
RFP	IX81	weiß	532 – 554 nm	562 nm	573 – 613 nm
Atto655	IX81	weiß	361 – 371 nm	427 nm	422 – 459 nm
Durchlicht	IX71	weiß	undurchlässig	—	—
RFP	IX71	530 nm	530 – 550 nm	576 nm	590 nm – ∞
Texas Red	IX71	590 nm	579 – 595 nm	593 nm	608 – 668 nm

Tabelle 4.2.: Eigenschaften der verschiedenen Kameras

	Hersteller	Typ	Auflösung	Pixelgröße	Maximale Ausleserate
CoolSNAP_{HQ}	Photometrics	CCD	1392×1040	6.45 µm	20 MHz
LucaR DL-604M-#VP	Andor	EMCCD	1004×1002	8 µm	12.4 MHz
iXon3 DU-888E-C00-#BV	Andor	EMCCD	1024×1024	13 µm	10 MHz

4.2.1.4. Temperaturkontrolle

Für die Reaktionen und Bakterien ist eine definierte und kontrollierte Temperatur nötig. Deswegen befand sich das IX81 in einer Temperaturbox, die die Reaktionsumgebungsluft auf einer konstanten Temperatur hielt. Beim IX71 wurde in die Probenhalterung eine durchsichtige Heizplatte (TOKAI HIT MATS_UAXKP-D) montiert, die die Probe temperiert. So konnten alle Experimente bei 37 °C durchgeführt werden.

4.2.1.5. Experimentzusammensetzung

Die vorgestellten Mikroskopkomponenten wurden nach Tab. 4.3 für die unterschiedlichen Experimente kombiniert. Zusätzlich wurden zwei verschiedene Aufnahmeprogramme verwendet: µManager 1.4[139] und Olympus xcellence pro 1.2.

Tabelle 4.3.: Zusammensetzung der Mikroskopkomponenten bei den Experimenten

Experiment	Mikroskop	Kamera	Programm	Kanäle
Oszillator	IX81	CoolSNAP	µManager	Durchlicht, GFP und Texas Red
Bakterien	IX81	iXon3	Olympus	Durchlicht, GFP, RFP und Atto 655
Stochastik	IX71	LucaR	µManager	Durchlicht, Texas Red und RFP

4.2.2. Spektrometer

Um für die in-vitro-Experimente eine Referenzmessung, die in einem makroskopischen Volumen stattfindet, durchzuführen, wurden zwei Spektrometer (schematischer Aufbau s. Abb. 2.2a), mit denen man den zeitlichen Verlauf der relevanten Fluoreszenzintensitäten aufnehmen kann, verwendet: ein Horiba/Jobin Yvon Fluorolog 3 System für die Oszillatorexperimente und ein Cary Eclipse der Firma Varian (jetzt Agilent) für die Stochastikexperimente. Beide haben den gleichen Grundaufbau (s. 2.1) aber unterscheiden sich in kleinen Details. So hat das Fluorolog einen klassisch integrierenden Photodetektor mit einer ständig leuchtenden Lampe. Das Cary verfügt hingegen über eine Blitzlichtlampe, was einen mittelnden Photodetektor ermöglicht.

Die Proben wurden in Hellma Präzisions-Küvetten aus Quarzglas SUPRASIL (105.254-QS) betrachtet und über ein Heizbad (Fluorolog) bzw. einen Peltierblock (Cary) auf die notwendigen 37 °C gebracht.

4.2.3. Mikrofluidik

Für die Mikrofluidikbauelemente mussten zuerst Negativvorlagen auf Siliziumbasis hergestellt werden. Dazu wurde das negative Design der Mikrofluidik auf eine hochauflösende Folie im Maßstab 1:1 gedruckt. Durch diese wurde dann ein mit negativem Photolack (AZ nLOF (1 µm)) beschichteter Siliziumwafer belichtet, so dass die Mikrofluidikstruktur als Erhebung aus Photolack nach dem Entwickeln übrig blieb.

Diese Vorlagen wurden dann mit flüssigem und entgastem PDMS (SYLGARD®184 silicone elastomer) übergossen, noch einmal entgast und im Ofen bei 80 °C ausgehärtet. Danach wurden die PDMS-Bauteile separiert – es wurden immer mehrere Bauelemente gleichzeitig gegossen –, nachbearbeitet und gereinigt. Anschließend wurden sie in einem Sauerstoffplasma auf einem Glasplättchen befestigt.

Da eine Benetzung der Kanaloberfläche die Tröpfchenproduktion unterbindet, wurde die Oberfläche hydrophob gemacht, indem das PDMS für mindestens 3 h auf 200 °C erhitzt wurde.[140]

4.2.4. Tröpfchenproduktion

Die Emulsionströpfchen wurden mit zwei verschiedenen Methoden erzeugt: durch Mikrofluidik und durch Schütteln. Beiden ist gemeinsam, dass sich in der Ölphase 1.8 % (Gewicht) Surfactant befanden. Unter dem Mikroskop wurden die Tröpfchen in einem ibidi µ-Slides VI$^{0.4}$ betrachtet.

4.2.4.1. Schütteln

In einem 200 µl-Tube wurden 45 µl Öl und 10 µl Reaktionslösung vermischt, indem für 30 – 60 s mit einem Vortex Genie 2 auf Höchststufe (2700 Hz) geschüttelt wurde. Es wurde immer so lange geschüttelt, bis die Lösung leicht trüb, aber noch nicht milchig erschien. Dann ist die Dichte der Tröpfchen unter dem Mikroskop ideal: dicht genug, so dass man viele Tröpfchen in einem Bildausschnitt sieht, aber nicht so viele, dass sich mehrere übereinander liegende Schichten von Tröpfchen ausbilden.

4.2.4.2. Mikrofluidik

In Mikrofluidik wurden die Tröpfchen durch ein sogenanntes Flow-focus Design[100] erzeugt. Dabei handelt es sich um eine rechtwinklige Zusammenführung von drei Flüssigkeitseinlässen und einem Auslass (s. Abb. 4.7). Das Öl strömt von zwei gegenüberliegenden Einlässen und die wässrige Lösung gegenüber vom Auslass in die Kreuzung. Durch die Oberflächenspannung kommt es bei geeigneten Flussratenverhältnissen in der Kreuzung zu einer Oszillation der Druckverhältnisse und das Wasser strömt gepulst durch die Kreuzung, wodurch am Auslass Tröpfchen entstehen. Die Flussraten des Öls und der wässrigen Phase wurden durch Spritzenpumpen (TSE Systems) eingestellt. Durch verschiedene Raten und Ratenverhältnisse – typischerweise ca. 100 µl h^{-1} für die wässrige Phase und 300 – 500 µl h^{-1} für das Öl – kann man verschiedene Tröpfchengrößen herstellen.[27]

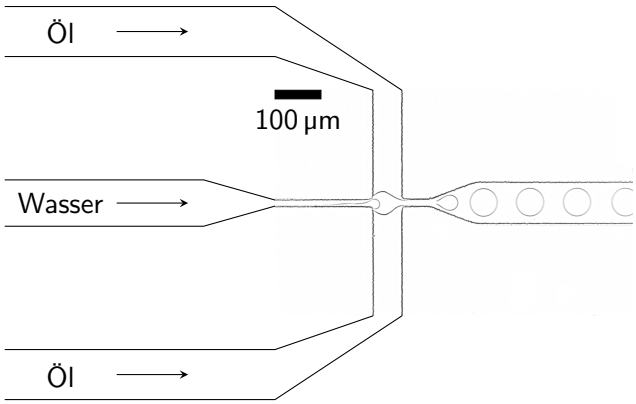


Abbildung 4.7.: Schematischer Aufbau eines Flow-focus.

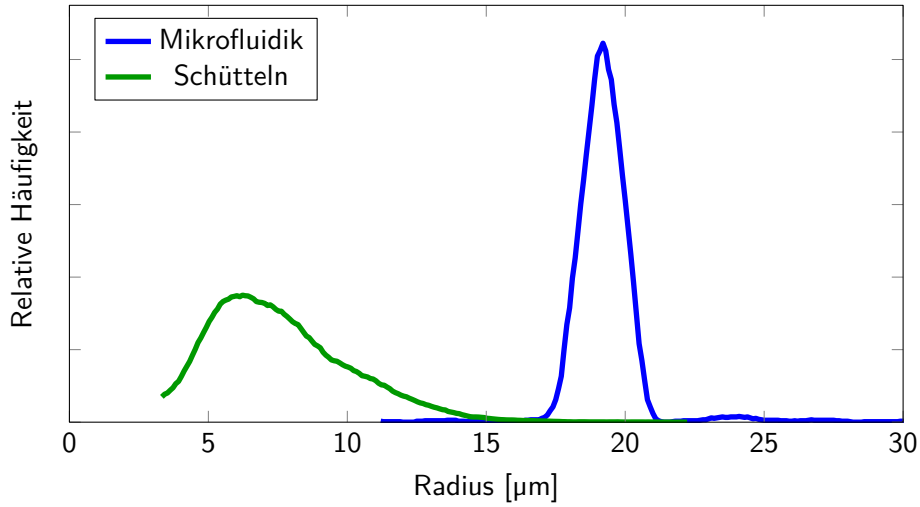


Abbildung 4.8.: Beispiele von Tröpfchengrößenverteilungen für geschüttelte und durch Mikrofluidik erzeugte Tröpfchen. Um die Verteilungsfunktion abzuschätzen, wurde die Radiuspopulation in überlappende Bereiche mit 1.5 µm Breite im Abstand von 0.1 µm eingeteilt und jeweils die Anzahl der Tröpfchen gezählt, die in den Bereich fallen. Die Verteilung der geschüttelten Tröpfchen ist bei kleinen Radien durch die Bildaufnahme und -analyse abgeschnitten und etwas zu niedrig abgeschätzt. Deswegen geht die wirkliche Verteilung noch zu kleineren Radien und auch das Maximum der Verteilung liegt wahrscheinlich bei einem etwas kleineren Radius.

4.2.4.3. Vergleich

Beide Methoden haben ihre Vor- und Nachteile und wurden in den Experimenten sorgfältig ausgewählt, um die gewünschten Ergebnisse messen zu können. Die Kategorien, in denen sich die Produktionsarten unterscheiden, sind:

Tröpfchengrößenverteilung: Der auffälligste Unterschied zwischen den beiden Methoden ist die Breite der Verteilung der Tröpfchengrößen (s. Abb. 4.8). So haben die Tröpfchen, die mit Mikrofluidik erzeugt wurden, alle eine sehr ähnliche Größe (monodispers), wohingegen das Schütteln eine sehr breite Verteilung erreicht (heterodispers).

Tröpfchengröße: Auch die erreichbare Größe der Tröpfchen ist unterschiedlich. So ist es schwierig mit Mikrofluidik Radien im Bereich von 2 µm herzustellen, was durch Schütteln ohne Probleme möglich ist. Dort können Radien in allen Größenbereichen entstehen.

Kontrollierbarkeit der Tröpfchengröße: Der große Nachteil des Schüttelns ist, dass man die Verteilung der Tröpfchengrößen kaum kontrollieren kann, da sie von extrem vielen Faktoren abhängt (z. B. genaue Temperatur des Öls, Haltung der Hand auf dem Schüttler, exakte Schüttelzeit oder sogar Unebenheiten im Schüttelgefäß). Bei der Mikrofluidik hingegen lässt sie sich gut einstellen.

Produktionszeit: Hier unterscheiden sich die Techniken um einen großen Faktor. Wie oben geschrieben, dauert die Produktion der Schütteltröpfchen ungefähr eine Minute. Die Mikrofluidik ist da um einiges langsamer, da mehr Handgriffe nötig sind, die Flussraten erst richtig eingestellt werden müssen und auch die Produktion selbst dann auch noch länger braucht. So dauert die Produktion ca. 20 – 45 Minuten.

Biokompatibilität: Bei beiden Methoden wird die wässrige Lösung verwirbelt, unter Druck gesetzt und mit Öl in Kontakt gebracht. Dadurch können Proteine denaturieren und ihre Funktionalität einbüßen.[141–145] Dieser Effekt wurde bei beiden Methoden beobachtet und es wurde versucht die Experimente dahingegen auszulegen, dass er das Ergebnis nicht beeinflusst.

4.2.4.4. Andere Methoden

Andere Methoden zur Emulsifizierung – z. B. mit Hochdruckventilhomogenisatoren oder Kolloidmühlen – wurden wegen der hohen Energiedichten, die während der Tröpfchenproduktion auftreten[146], verworfen, da diese wahrscheinlich die Proteine in den Lösungen zerstört hätten. Auch erzeugen diese Standardmethoden keine monodispersen Emulsionen und haben deswegen keinen Vorteil gegenüber dem Schütteln.

Um mikrokanalbasierte Methoden[147] durchzuführen, die monodisperse kleine Tröpchen erzeugen könnten, fehlte im Labor die nötige Expertise.

4.3. Durchführung

Im Folgenden sollen die Durchführungsprotokolle der Experimente, die teilweise sehr ähnlich sind, einzeln beschrieben werden.

4.3.1. Oszillator in Tröpfchen

Die Reaktionslösung, die in den Tröpfchen eingeschlossen wurde, setzt sich wie folgt zusammen:

1. 10 % (Volumen) 10-fach NEB-Transkriptions-Puffer
2. 80 nM T_{12}
3. 150 nM T_{21}
4. 500 nM dI_1
5. 150 nM A_1
6. 250 nM A_2
7. 24 mM $MgCl_2$
8. 7.5 mM von jedem rNTP (epicentre)
9. 1 % 100-fach Pyrophosphataselösung
10. 500 nM Alexa 488
11. 12.15 units/µl (gedämpfte und stabile Stimmung) bzw. 6.075 units/µl (stark gedämpft) epicentre T7-Polymerase
12. 60.75 units/ml (gedämpft), 48.6 units/ml (stabil) bzw. 40.1 units/ml (stark gedämpft) RNase H

Ablauf des Experiments:

1. Als erstes wurden die Küvetten gespült und getrocknet. Zusätzlich wurden die ibidi µ-Slides bereitgelegt und einer der vier Kanäle mit ca. 50 µl Öl gefüllt.
2. Dann wurde die Reaktionslösung vorbereitet, wobei die T7-RNA-Polymerase als Startsubstanz weggelassen wurde.
3. Anschließend wurden Küvetten befüllt, in das schon beheizte Spektrometer gestellt und die Messung am Spektrometer gestartet.
4. In 200 µl-Tubes wurden 45 µl Öl zur Vorbereitung gefüllt.
5. Wenn die Probe temperiert war (nach ca. 10 min), wurde sie aus dem Spektrometer genommen – die Messung lief weiter –, um die T7-RNA-Polymerase hinzuzufügen.

6. Von der so komplettierten Reaktionslösung wurde jeweils 10 µl in die vorbereiteten Tubes gegeben.

7. Die Küvetten wurden zurück in das Spektrometer gestellt und die Oberfläche mit 35 µl n-Hexadekan abgedeckt, um die Verdunstung der Probe zu verhindern.

8. Nun wurden die Tröpfchen durch 60 s Schütteln erzeugt und 45 µl der Tröpfchenlösung in die Kanäle gefüllt.

9. Damit keine Verdunstung stattfand, wurden die Tröpfchenkanäle mit einem Klebestreifen versiegelt.

10. Nun wurden der µ-Slide auf das temperierte Mikroskop gelegt, die Positionen ausgewählt und die Messung am Mikroskop gestartet.

4.3.2. Bakterien in Tröpfchen

Bei den Bakterienexperimenten wurden zuerst die genetischen Schaltkreise charakterisiert. Anschließend wurden die Experimente mit den Tröpfchen durchgeführt. Die Vorbereitung der Bakterien war bei beiden Schritten gleich:

1. Die Bakterien wurden über Nacht bei 37 °C in LB Nährmedium inkubiert, um eine ausreichende Menge Bakterien zur Verfügung zu haben. Damit nur die Bakterien überleben, die auch das Plasmid enthalten, wurden 100 µg ml^{-1} Carbenicillin (Carl Roth) bzw. 30 µg ml^{-1} Chloramphenicol (Carl Roth) hinzugefügt.

2. Anschließend wurde die Bakterienlösung auf eine optische Dichte[7] von 0.01 bei 600 nm verdünnt. Zum Verdünnen wurde auch LB-Medium verwendet – nur bei den Experimenten zum UND-Gatter wurde M9 Minimalmedium verwendet. Die Antibiotikakonzentrationen wurden dabei aber auf die Hälfte der Inkubierkonzentration reduziert, um den Stress auf die Bakterien zu minimieren.

3. Nun wurden die Bakterien noch einmal für 2 – 3 h bei 37 °C gehalten, bis sie eine optische Dichte von 0.1 – 0.2 erreichten.

Tabelle 4.4.: Verwendete Filter bei den Charakterisierungsexperimenten

Protein	Anregungsfilter	Emissionsfilter
GFP	480 – 490 nm	515 – 525 nm
RFP	579 – 589 nm	615 – 625 nm

4.3.2.1. Charakterisierung der Bakterien

Um die Antwort der Bakterien auf verschiedene Signalstoffkonzentrationen zu charakterisieren, wurden in einer 96-Kammern-Platte (ibidi 89621) verschiedene AHL- (Sigma-Aldrich K3007) und IPTG-Konzentrationen (Carl Roth CN08.2) vorbereitet und anschließend die Bakterienlösung hinzugegeben. Die Fluoreszenzantwort wurde dann in einem Plattenfluoreszenzlesegerät (FLUOstar Optima, BMG Labtech) gemessen. Dabei wurde einerseits die Expression des fluoreszierenden Proteins mit einer geeigneten Wellenlänge (s. Tab. 4.4), andererseits das Wachstum der Bakterien über die gleichzeitige Messung der Absorption bei 600 nm bestimmt.

4.3.2.2. Bakterien in Tröpfchen

Bei den Experimenten mit Bakterien in Tröpfchen gab es bis zu drei verschiedene Tröpfchensorten mit unterschiedlichem Inhalt. Die Tröpfchen, die Bakterien enthielten, waren immer ähnlich gefüllt, nur dass verschiedene Bakterien verwendet wurden. Die Tröpfchen mit Signalstoff wurden mit dem gleichen Medium gefüllt, in dem sich auch die Bakterien befanden, um gleiche Fluoreszenzhintergründe zu erhalten. Zusätzlich wurde immer ein Fluoreszenzfarbstoff hinzugegeben, um die Tröpfchen unter dem Mikroskop und in der Auswertung unterscheiden zu können. So enthielten die Signaltröpfchen in den unterschiedlichen Kommunikationsmodi Folgendes:

AHL-Diffusion: 200 nM AHL und 500 nM Alexa 488 (Life Technologies)

IPTG-Diffusion: 10 mM IPTG und 500 nM Texas Red (Life Technologies)

Sender und Empfänger: Hier ist das Signaltröpfchen auch mit Bakterien gefüllt und die Senderbakterien wurden vor der Tröpfchenproduktion mit 1 mM IPTG induziert.

UND-Gatter: Hierbei gab es zwei Signaltröpfchensorten:

- 200 nM bzw. 20 nM AHL und 1 µM TAMRA (Sigma-Aldrich)
- 10 mM bzw. 200 µM IPTG und 1 µM Atto 655 (Atto-Tec)

Dabei war die Experimentdurchführung wegen der Mikrofluidik etwas aufwändiger und zeitintensiver:

1. Nach bzw. während der Bakterienvorbereitung wurde das PDMS-Bauelement mit Öl gefüllt, damit keine Luftblasen die Tröpfchenproduktion stören, und auf dem Mikroskop platziert.
2. Öl und Bakterienlösung wurden in Spritzen gefüllt, die an das Bauelement angeschlossen und anschließend in die Spritzenpumpen eingebaut wurden.

[7] Die optische Dichte entspricht der Extinktion über 1 cm: $\mathrm{OD} = \ln\left(\frac{I_0}{I(1\,\mathrm{cm})}\right)$.

3. Die Spritzenpumpen wurden gestartet und die Flussraten so lange geändert, bis sich eine stabile Tröpfchenproduktion mit dem gewünschten Tröpfchenradius einstellte.
4. Nach der Produktion der Tröpfchen mit Bakterien wurden die Tröpfchen mit den Signalstoffen produziert. Für jede Tröpfchensorte wurde ein frisches Bauelement verwendet, um Kreuzkontamination zu vermeiden.
5. Nachdem alle Tröpfchen produziert worden waren, wurden die verschiedenen Tröpfchenlösungen in den Kanal des ibidi µ-Slides transferiert und dort durch sanftes Drehen vermischt.
6. Zum Schluss wurden der µ-Slide auf dem Mikroskop platziert, die gewünschten Positionen ausgewählt und die Messung gestartet.

4.3.3. Stochastik in Tröpfchen

Die Stochastiktröpfchen enthielten folgende Lösung (alle Prozentangaben sind Volumenprozente):

1. 10 % 10-fach Tris-Puffer
2. 10 % DTT (10 mM)
3. 4 % Reporter-DNA (400 nM)
4. 6 % Inhibitor-DNA (600 nM)
5. 5 % NEB Ribonukleotidlösung (4 mM)
6. 10 % NEB T7-RNA-Polymerase (5 units/µl)
7. 0.5 – 5 % Templat-DNA (1 – 50 nM – für kleine Konzentrationen wurde eine verdünnte Lösung verwendet.)
8. Rest: nukleasefreies Wasser

Die Durchführung fand nach folgendem Protokoll statt:

1. Als Erstes wurden die Küvetten gespült und mit 1 ml 10 %iger TWEEN 80 Lösung gefüllt. Nach mindestens 20 min wurden die Küvetten noch einmal vorsichtig gespült und getrocknet. Zusätzlich wurden die ibidi µ-Slides bereitgelegt und einer der vier Kanäle mit ca. 50 µl Öl gefüllt.
2. Anschließend wurde die Reaktionslösung vorbereitet. Dabei wurde die Templat-DNA weggelassen, um sie später als Startsubstanz hinzuzufügen.
3. Dann wurden die Küvetten befüllt, in das schon beheizte Spektrometer gestellt und die Messung am Spektrometer gestartet.
4. In 200 µl-Tubes wurden 45 µl Öl zur Vorbereitung gefüllt.

5. Wenn die Probe temperiert war (nach ca. 10 min), wurde sie aus dem Spektrometer genommen – die Messung lief weiter –, um die Templat-DNA hinzuzufügen.

6. Von der fertigen Reaktionslösung wurden 10 µl in die vorbereiteten Tubes und mindestens 60 µL in einen der Kanäle als Referenz gegeben.

7. Die Küvetten wurden anschließend zurück in das Spektrometer gestellt und die Oberfläche mit mindestens 50 µl n-Hexadekan abgedeckt, um die Verdunstung der Probe zu verhindern.

8. Nun wurden die Tröpfchen durch 30 – 60 s Schütteln erzeugt und 40 µl der Tröpfchenlösung in die Kanäle gefüllt.

9. Damit keine Verdunstung stattfand, wurden die Tröpfchenkanäle zuerst mit ca. 30 µl Wasser und zusätzlich noch mit ca. 20 µl n-Hexadekan bedeckt – der Referenzkanal wurde nur mit n-Hexadekan abgedeckt.

10. Alle Kanalöffnungen wurden dann mit einem Klebestreifen versiegelt.

11. Nun wurden der µ-Slide auf das temperierte Mikroskop gelegt, die Positionen ausgewählt und die Messung am Mikroskop gestartet.

4.4. Ausführende

Die verschiedenen Experimente wurden jeweils von unterschiedlichen Teams ausgeführt. Im Folgenden soll die Arbeit jedes Mitglieds gewürdigt werden.

Oszillator in Tröpfchen

Das Oszillatorprojekt wurde von Maximilian Weitz, Jongmin Kim, Korbinian Kapsner, Erik Winfree, Elisa Franco und Friedrich C. Simmel gestaltet. Die letzten drei entwarfen die Grundidee des Experiments. Die Hauptexperimente des Projektes wurden von M.Weitz und kleinere Versuchen auch noch von J. Kim, E. Franco und K. Kapsner durchgeführt. Für die Auswertung und Aufbereitung der Daten war vor allem K. Kapsner zuständig, wobei auch hier Bereiche von M. Weitz, J. Kim und E. Franco übernommen wurden. Die Simulationen und die Modellierung des Systems war alleine der Bereich von J. Kim.

Bakterien in Tröpfchen

Auch das Team der Bakterienexperimente war recht groß: Maximilian Weitz, Andrea Mückl, Korbinian Kapsner, Ronja Berg, Andrea Meyer und Friedrich C. Simmel. Hierbei wurde das Design und die Planung des Experiments von M. Weitz, A. Mückl und F. C. Simmel übernommen. Die Charakterisierung der genetischen Schaltkreise war Teil der Bachelorarbeit von R. Berg. Deswegen wurde die Durchführung und Auswertung dieses Bereichs von ihr übernommen. Die restlichen Tröpchenexperimente wurden von A. Mückl und M. Weitz und die Auswertung/Aufbereitung wieder von K. Kapsner ausgeführt. F. C. Simmel übernahm die Simulation und Modellierung der Ergebnisse.

Stochastik in Tröpfchen

Das Stochastikprojekt hatte das kleinste Team: Korbinian Kapsner und Friedrich C. Simmel. Dabei planten und designten beide das Konzept und K. Kapsner führte das Projekt aus.

In all diesen Teams wurden die Daten von allen – oft in einer regen Diskussion – interpretiert und Schlussfolgerungen gemeinsam gezogen.

5. Datenanalyse

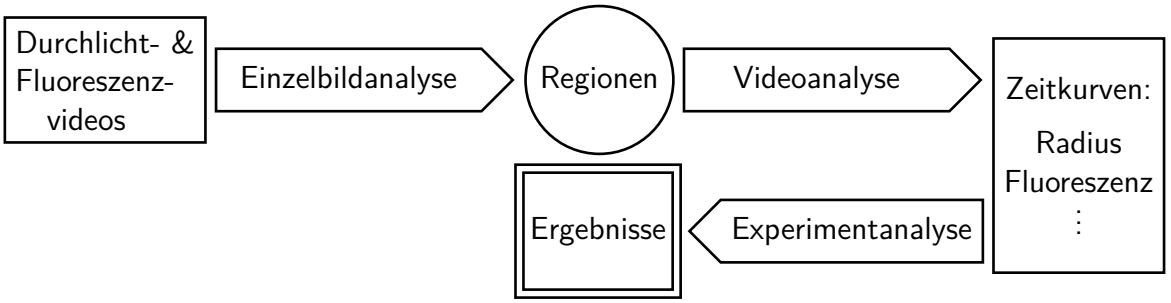


Abbildung 5.1.: Arbeitsflussdiagramm der Datenanalyse.

Die Datenanalyse unterteilt sich in drei Schritte (s. Abb. 5.1). Den Ersten bildet die Einzelbildanalyse, der zweite Schritt setzt die Daten aus dieser Analyse als Video zusammen und der dritte Schritt extrahiert dann aus den Videodaten die experimentspezifisch interessierenden Werte. Da vor allem die ersten beiden Schritte schnell auszuführen sein sollten, mussten einige Algorithmen selbst entwickelt und implementiert werden.

5.1. Einzelbildanalyse

Die Einzelbildanalyse gliedert sich in fünf Teile, die aufeinander aufbauen (s. Abb. 5.2). Als Erstes muss das Rohdatendurchlichtbild aufbereitet werden, um unerwünschte Bildbestandteile (z. B. Beleuchtungsgradienten oder Pixelrauschen) zu entfernen. Anschließend wird das Bild in ein Schwarzweißbild umgewandelt. Dieses binäre Bild wird dann weiter so bearbeitet, dass im vierten Schritt die zusammenhängenden Bildbereiche im SW-Bild segmentiert werden können. Um die experimentalrelevanten Fluoreszenzdaten dieser Bereiche zu erhalten, werden die Bereiche im letzten Schritt in dem/den Fluoreszenzbild/-ern ausgewertet.

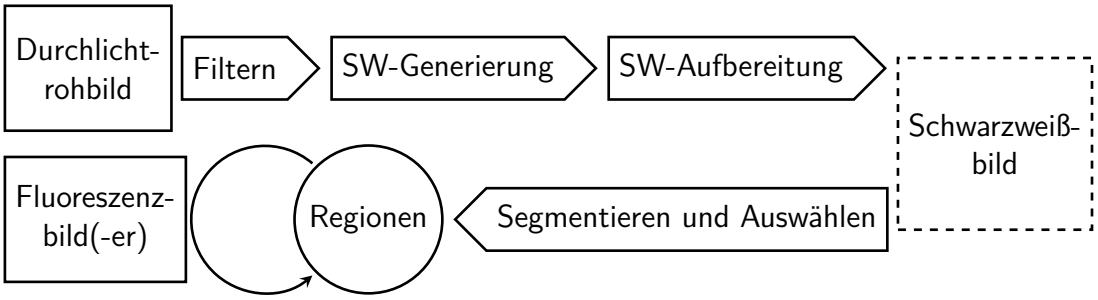


Abbildung 5.2.: Arbeitsflussdiagramm der Einzelbildanalyse.

5.1.1. Bildaufbereitung

Um das Bild in ein sauberes Schwarzweißbild umzuwandeln, müssen Pixelrauschen und jegliche ungleichmäßige Beleuchtung entfernt werden. Um das zu bewerkstelligen, wird ein Bandpassfrequenzfilter auf das Bild ($w \times h$ groß) angewendet. Dazu wird zuerst das komplexe Frequenzbild $\tilde{f}(\omega_x, \omega_y) \in \mathbb{C}$ ($\omega_x \in \mathbb{N}_0 \cap [0, w-1]$ und $\omega_y \in \mathbb{N}_0 \cap [0, h-1]$) über die zweidimensionale diskrete Fouriertransformation aus dem Originalbild $f(x, y)$ ($x \in \mathbb{N}_0 \cap [0, w-1]$ und $y \in \mathbb{N}_0 \cap [0, h-1]$) erzeugt:

$$\tilde{f}(\omega_x, \omega_y) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} f(x, y) \cdot e^{-\frac{2\pi i}{w} \cdot x \cdot \omega_x} \cdot e^{-\frac{2\pi i}{h} \cdot y \cdot \omega_y} \tag{5.1.1}$$

Dieses Frequenzbild wird dann pixelweise mit der Differenz $g(\omega_x, \omega_y)$ zwischen zwei Gaußbildern multipliziert:

$$\begin{aligned} g(\omega_x, \omega_y) = \exp\left(-\left(\frac{\omega_x}{\sigma_{h,x}}\right)^2 - \left(\frac{\omega_y}{\sigma_{h,y}}\right)^2\right) - \\ \exp\left(-\left(\frac{\omega_x}{\sigma_{l,x}}\right)^2 - \left(\frac{\omega_y}{\sigma_{l,y}}\right)^2\right) \end{aligned} \tag{5.1.2}$$

Dabei sind $\sigma_{h,x}$, $\sigma_{h,y}$, $\sigma_{l,x}$ und $\sigma_{l,y}$ die Filterfrequenzen in x- bzw. y-Richtung. Alle Frequenzen, die höher als $\sigma_{h,x}$ bzw. $\sigma_{h,y}$ und niedriger als $\sigma_{l,x}$ bzw. $\sigma_{l,y}$ sind, werden also stärker als $1/e$ gedämpft.

Man kann dies im Ortsraum so interpretieren, dass alle Strukturen, die größer als Δx_h bzw. Δy_h und kleiner als Δx_l bzw. Δy_l sind, aus dem Bild entfernt werden.

$$\begin{aligned} \Delta x_l &= \frac{w}{2\sigma_{l,x}} \quad \Delta y_l = \frac{h}{2\sigma_{l,y}} \\ \Delta x_h &= \frac{w}{2\sigma_{h,x}} \quad \Delta y_h = \frac{h}{2\sigma_{h,y}} \end{aligned} \tag{5.1.3}$$

Die σ werden jeweils so gewählt, dass die Strukturauflösung in x- und y-Richtung immer gleich ist. Die Strukturgrößen heißen dann Δl und Δh. Für Δl wurden Werte zwischen 2 und 6 Pixel und für Δh 20 Pixel gewählt.

Das multiplizierte Frequenzbild wird anschließend durch die inverse zweidimensionale diskrete Fouriertransformation in ein Realbild $f^*(x, y) \in \mathbb{R}$ umgewandelt:

$$f^*(x, y) = \frac{1}{w \cdot h} \sum_{\omega_x=0}^{w-1} \sum_{\omega_y=0}^{h-1} g(\omega_x, \omega_y) \cdot \tilde{f}(\omega_x, \omega_y) \cdot e^{\frac{2\pi i}{w} \cdot x \cdot \omega_x} \cdot e^{\frac{2\pi i}{h} \cdot y \cdot \omega_y} \tag{5.1.4}$$

Dieses Vorgehen entspricht einer Faltung des Bildes mit der Differenz zweier Gaußbilder mit den Breiten Δl und Δh.

(a) Originalbild **(b)** Gefiltertes Bild **(c)** Stark gefiltertes Bild

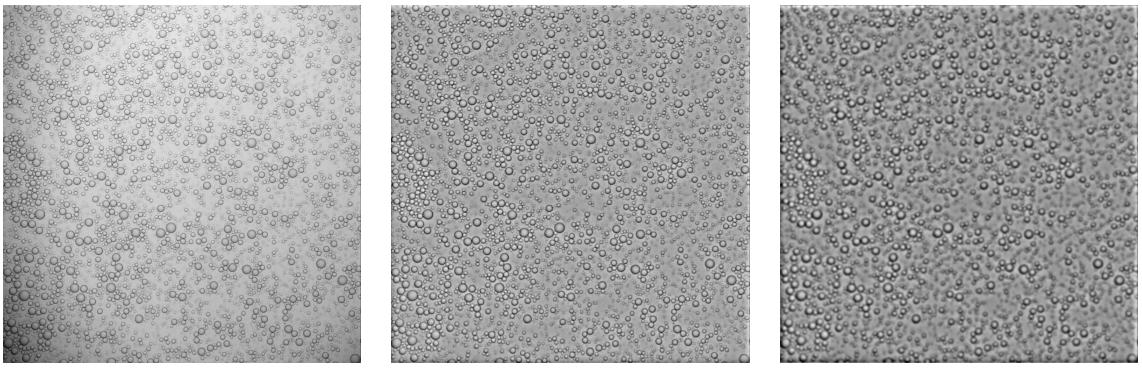


Abbildung 5.3.: Illustration der Rohbildfilterung. Es ist ein typisches Bild geschüttelter Tröpfchen in 10-facher Vergrößerung gezeigt. Die Kantenlänge ist 1024 Pixel bzw. 1.3 mm. Die verwendeten Kameras produzierten aber nicht Bilder gleicher Größe (s. Tab. 4.2).

Die so bearbeiteten Bilder haben nun einen deutlich reduzierten Beleuchtungsgradienten und zeigen weniger Rauschen (s. Abb. 5.3a und Abb. 5.3b). Bei etwas höherem Δl kann das Bild auch etwas unscharf erscheinen, was manchmal durchaus gewollt ist (s. Abb. 5.3c), weil dadurch z. B. sehr kleine Tröpfchen, die manchmal die Datenanalyse durch ihre hohe Anzahl erschweren, so „verschmiert“ werden, dass sie nicht mehr erkennbar sind.

Solche Gaußfilter sind standardmäßig in Bildanalyse- oder Bildbearbeitungsprogrammen wie z. B. ImageJ[148] oder GIMP[149] enthalten und werden in weiten Bereichen eingesetzt. Auch hat der Gaußfilter ein glattes Verhalten im Frequenz- und im Ortsraum, wodurch er wenige Artefakte erzeugt. Als Alternative zur Rauschreduktion könnte man auch einen zweidimensionalen Medianfilter verwenden, der die Kanten im Bild besser erhält.[150] Er entfernt aber keinen Beleuchtungsgradienten und kann sehr kleine Tröpfchen nicht „verschmieren“.

5.1.2. Konvertierung zu Schwarzweißbild

Das aufbereitete Graustufenbild wird nun in ein Schwarzweißbild umgewandelt. Die Grundprozedur dabei ist recht einfach. Man wählt einen Schwellwert und jedes Pixel, dessen Wert über dem Schwellwert liegt, wird in ein weißes Pixel umgewandelt. Alle anderen Pixel werden schwarz eingefärbt (s. Abb. 5.4). Komplizierter ist die Festlegung des Schwellwertes, wenn dieser automatisch erkannt werden soll. Dafür wurde die Methode von Otsu[151] verwendet[1].

Neben dieser globalen Schwellwertmethode existieren auch Methoden, die mit lokalen Schwellwerten arbeiten. Diese sind aber in der Berechnung langsamer – es müssen ja viel

[1] Es wurde die MATLAB-Funktion graythresh verwendet.

mehr Werte bestimmt werden. Dafür liefern sie manchmal etwas bessere Ergebnisse.[152] Da die Berechnungsgeschwindigkeit ein ausschlaggebender Faktor war und die Tröpfchen nach der Umwandlung gut weiterverarbeitet werden konnten, wurde Otsu verwendet.

5.1.2.1. Schwellwertbestimmung nach Otsu

Die Grauwerte der einzelnen Pixel sollen in zwei Populationen eingeordnet werden. Dazu wird die Gesamtpopulation, dargestellt durch die Wahrscheinlichkeitsdichte $\rho(x)$[2], am Schwellwert k auseinander geschnitten. Dadurch ergeben sich folgende statistischen Werte:

$$\omega_1 = \int_{-\infty}^{k} \rho(x)dx \qquad = \text{relative Häufigkeit der Population 1} \tag{5.1.5}$$

$$\mu_1 = \int_{-\infty}^{k} \frac{x}{\omega_1}\rho(x)dx \qquad = \text{Mittelwert der Population 1} \tag{5.1.6}$$

$$\sigma_1^2 = \int_{-\infty}^{k} \frac{(x-\mu_1)^2}{\omega_1}\rho(x)dx \qquad = \text{Varianz der Population 1} \tag{5.1.7}$$

$$\omega_2 = \int_{k}^{\infty} \rho(x)dx \qquad = \text{relative Häufigkeit der Population 2} \tag{5.1.8}$$

$$\mu_2 = \int_{k}^{\infty} \frac{x}{\omega_2}\rho(x)dx \qquad = \text{Mittelwert der Population 2} \tag{5.1.9}$$

$$\sigma_2^2 = \int_{k}^{\infty} \frac{(x-\mu_2)^2}{\omega_2}\rho(x)dx \qquad = \text{Varianz der Population 2} \tag{5.1.10}$$

$$\mu = \int_{-\infty}^{\infty} x\rho(x)dx \qquad = \text{Mittelwert der Gesamtpopulation} \tag{5.1.11}$$

$$\sigma = \int_{-\infty}^{\infty} (x-\mu)^2\rho(x)dx \qquad = \text{Varianz der Gesamtpopulation} \tag{5.1.12}$$

$$\sigma_{in}^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2 \qquad = \text{Gesamtvarianz innerhalb der Populationen} \tag{5.1.13}$$

$$\sigma_{zw}^2 = \omega_1\omega_2(\mu_2-\mu_1)^2 \qquad = \text{Gesamtvarianz zwischen den Populationen} \tag{5.1.14}$$

mit den Zusammenhängen:

$$\mu = \omega_1\mu_1 + \omega_2\mu_2 \tag{5.1.15}$$

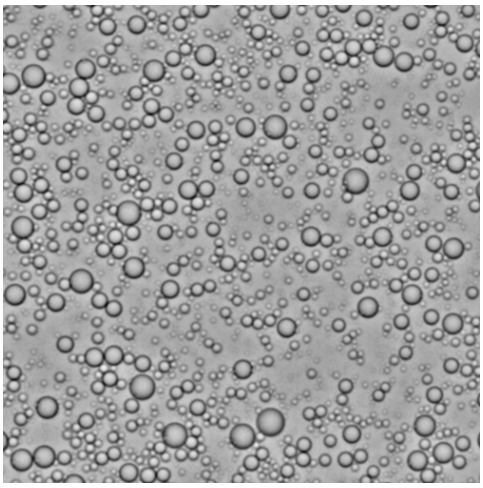
$$\sigma^2 = \omega_1\left(\sigma_1^2 + (\mu_1-\mu)^2\right) + \omega_2\left(\sigma_2^2 + (\mu_2-\mu)^2\right) = \sigma_{in}^2 + \sigma_{zw}^2 \tag{5.1.16}$$

Um nun die Gesamtvarianz innerhalb der Populationen zu minimieren und zwischen den Populationen zu maximieren, wird der Quotient λ maximiert:

$$\lambda = \frac{\sigma_{zw}^2}{\sigma_{in}^2} \tag{5.1.17}$$

[2] In der Originalversion wurde auf einer diskreten Wahrscheinlichkeitsdichte $\rho(x)$ gerechnet und argumentiert. Die Argumentation ist dann exakt die Gleiche. Man muss nur den Populationsgrenzen besondere Aufmerksamkeit schenken.

(a) Graustufenbild



(b) Schwarzweißbild

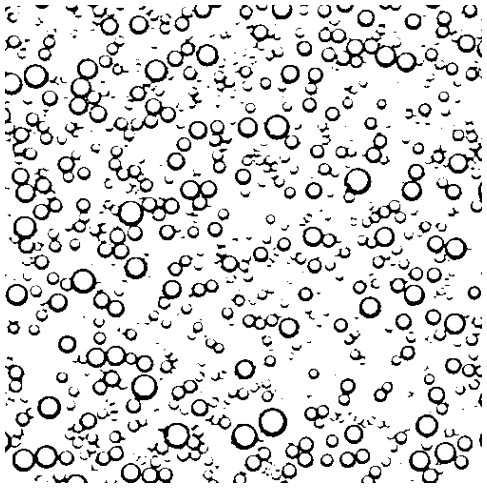


Abbildung 5.4.: Illustration der Konvertierung zu Schwarzweißbild.

Durch geschicktes Umstellen erkennt man, dass auch eine reine Maximierung von σ_{zw}^2 ausreichend ist:

$$\begin{aligned}
\sigma_{zw}^2 &= \underbrace{\sigma^2}_{\text{unabhängig von } k} \cdot \frac{\sigma_{zw}^2}{\sigma^2} \\
&= \sigma^2 \cdot \frac{\sigma_{zw}^2}{\sigma_{zw}^2 + \sigma_{in}^2} \\
&= \sigma^2 \cdot \frac{\frac{\sigma_{zw}^2}{\sigma_{in}^2}}{\frac{\sigma_{zw}^2}{\sigma_{in}^2} + 1} \\
&= \sigma^2 \cdot \frac{\lambda}{\lambda + 1}
\end{aligned} \tag{5.1.18}$$

Da σ_{zw}^2 nur von den Mittelwerten und den relativen Häufigkeiten der Populationen abhängt, wird die Berechnung dadurch schneller.

5.1.3. Schwarzweißbildaufbereitung

Das Schwarzweißbild kann schließlich noch aufbereitet werden, um ein besseres Segmentieren zu ermöglichen. Jeder dieser Aufbereitungsschritte ist optional.

5.1.3.1. Löcher auffüllen

Obwohl das Rohdatenbild gefiltert wird, kann es immer noch passieren, dass innerhalb einer zusammenhängenden Fläche im Schwarzweißbild Löcher entstehen (s. Abb. 5.5a). Diese Löcher stören bei der weiteren Analyse[3] und müssen entfernt werden. Hierbei kann eingestellt werden, wie groß die Löcher, die entfernt werden, maximal sein dürfen.

Um zu bestimmen, ob ein schwarzes Pixel weiß eingefärbt werden muss, wird festgestellt, zu welchem zusammenhängenden schwarzen Bereich es gehört[4]. Wenn dieser Bereich kleinergleich der maximalen Lochgröße ist, wird der komplette Bereich weiß gefärbt. Wenn er größer ist, bleiben die Pixel schwarz, aber alle Pixel werden so markiert, dass sie nicht noch einmal untersucht werden.

5.1.3.2. Sackgassen entfernen

Ähnlich wie Löcher können Sackgassen (s. Abb. 5.6a) in der Analyse Probleme bereiten. Um die Sackgasse zu entfernen, wird zuerst definiert, was ein Sackgassenpixel ist: es handelt sich um ein Sackgassenpixel, wenn es schwarz und maximal eines seiner acht Nachbarpixel schwarz ist.

Wenn nun solch eine Sackgasse entfernt wird, kann ein eventuell vorhandenes Nachbarpixel zu einem Sackgassenpixel werden. Deswegen werden zuerst alle schwarzen Pixel dahingegehend untersucht, ob sie Sackgassenpixel sind. Wenn eines als Ende der Sackgasse identifiziert wurde, wird es weiß eingefärbt und mit den neuen Pixeldaten die Nachbarpixel untersucht. So wird eine längere Sackgasse vom Ende her sukzessive abgebaut.

5.1.3.3. Brücken entfernen

Zwischen weißen Regionen befinden sich manchmal schmale Brücken (s. Abb. 5.7a). Diese sind meistens keine reale Verbindung zwischen den Regionen, sondern wurden durch schlechten Kontrast erzeugt. Deswegen müssen die Brücken entfernt werden, um die Regionen getrennt analysieren zu können. Wenn die Regionen doch zusammenhängen, entfernt sie ein späteres Filterkriterium (z. B. in 5.1.4.4), sodass sie in den Ergebnissen nicht erscheinen. Videos, in denen diese falsche Trennung sehr häufig auftritt, können

[3] Zum Beispiel beim Entfernen der Brücken kann ein Loch in einem schmalen Bereich die zusammenhängende Fläche in unerwünschter Weise auseinanderschneiden.
[4] Der Bereich wird über einen sogenannten „flood fill“-Algorithmus bestimmt.

(a) Bild mit Löchern **(b)** Zu füllende Löcher

Abbildung 5.5.: Illustration des Löcherfüllens mit Maximallochgröße 1 (orange), 4 (blau), 9 (grün) und 16 (cyan).

(a) Bild mit Sackgassen **(b)** Zu füllende Sackgassen (orange)

Abbildung 5.6.: Illustration der Sackgassenentfernung.

(a) Bild mit Brücken **(b)** Zu entfernende Brücken

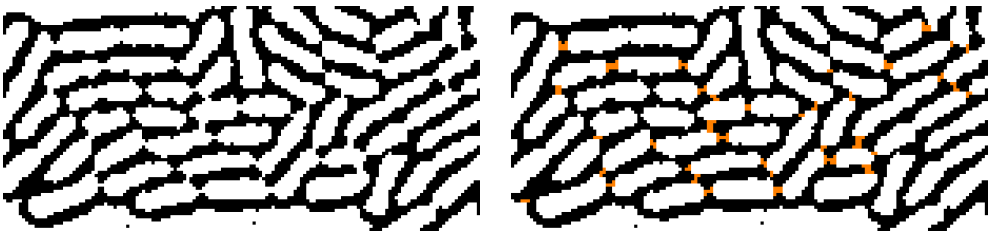


Abbildung 5.7.: Illustration des Brückenentfernens mit der Erweiterung des Algorithmus auf 2 Pixel.

(a) Bild mit breiten Rändern **(b)** Verschmälerte Ränder

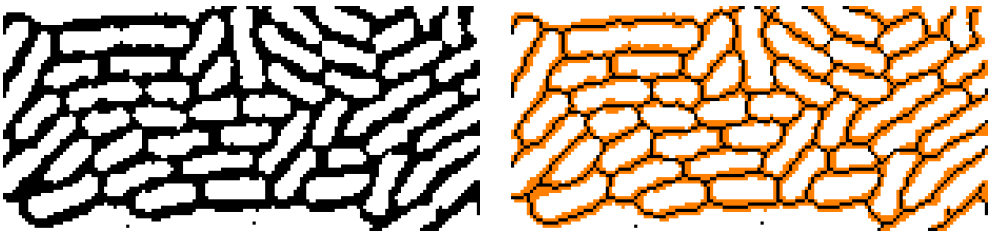


Abbildung 5.8.: Illustration des Ränderminimierens.

(a) Bild mit Randflächen **(b)** Zu entfernende Randflächen

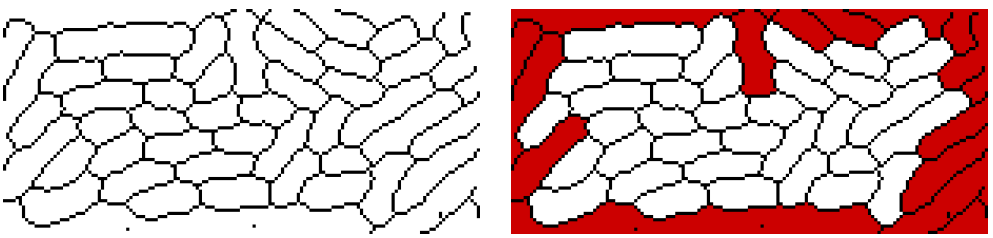


Abbildung 5.9.: Illustration des Randflächenentfernens.

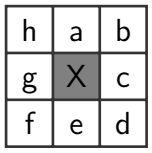


Abbildung 5.10.: Benennung der Nachbarn des Pixels X.

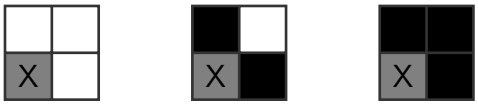


Abbildung 5.11.: Pixelkonfigurationen der Ecke ohne Wechsel.

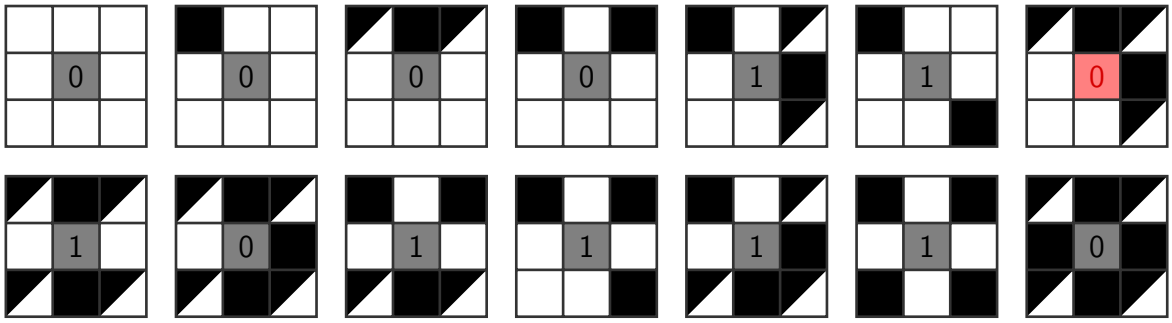


Abbildung 5.12.: Übersicht über alle möglichen Pixelmuster. Die halbgefüllten Pixel können weiß oder schwarz sein – die Eckenentscheidung, ob ein Wechsel vorliegt oder nicht, ist unabhängig davon. Die Zahl in der Mitte repräsentiert das Ergebnis des Algorithmus. Das Muster mit dem rot hinterlegten Mittelpixel zeigt die Ausnahmekonfiguration.

meist nicht vollautomatisch ausgewertet werden und der Operator kann die Fehler korrigieren.

Um die Brücken zu entfernen, wird jedes Pixel einzeln mit seiner Umgebung betrachtet (s. Abb. 5.10). Eine Brücke muss dann entfernt werden, wenn zwei gegenüberliegende Ecken einen Wechsel von weiß nach schwarz oder schwarz nach weiß aufweisen. Dabei gibt es nur drei Eckkonfigurationen, die keinen Wechsel beinhalten (s. Abb. 5.11) bzw. so definiert werden.

Ob eine Ecke – z. B. die Pixel a, b und c in Abb. 5.10 – nun einen Wechsel besitzt, kann man mit folgender logischer Operation entscheiden, wobei m der Pixelwert direkt in der Ecke (b im Beispiel) und l und r der Pixelwert der anschließenden Pixel (im Beispiel a und c) ist:

$$\text{hat Wechsel} = (\neg l \otimes \neg r) \vee (\neg m \wedge l) \tag{5.1.19}$$

Nur eine Nachbarschaftskonfiguration bricht aus dem Schema aus und muss gesondert betrachtet werden: wenn eine Ecke komplett weiß ist und die schwarzen Pixel in den anderen Ecken verbunden sind, soll das Mittelpixel nicht entfernt werden (s. Abb. 5.12).

Dieser Algorithmus schließt nur Lücken, die ein Pixel groß sind. Um auch Zwei-Pixel-Lücken zu schließen, muss er nur geringfügig erweitert werden:

Zuerst legt man fest, dass die Lücke nur zwei Pixel auf der Horizontalen oder Vertikalen groß sein darf. Eine diagonale Lücke wird nicht betrachtet, da die Realisierung viel komplexer wäre. Solche Lücken können über einen Zwei-Stufen-Prozess mit dem obigen

Algorithmus geschlossen werden, indem man bei den Nachbarpixeln a, c, e und g auch noch das Pixel eine Position weiter außen mit betrachtet. So wird z. B. a als schwarz angesehen, wenn das Pixel bei a oder das Pixel direkt darüber schwarz ist. Wenn nun das mittlere Pixel schwarz eingefärbt wird, werden die Felder, die kombiniert betrachtet wurden, auch schwarz angefärbt, wenn die Kombination schwarz ist.[5]

5.1.3.4. Ränder minimieren

Die schwarzen Ränder zwischen den weißen Flächen sind oft zu breit bzw. müssen zu breit gewählt werden, damit die weißen Flächen gut separiert werden (s. Abb. 5.8a). Ein sogenannter „thinning"-Algorithmus, der die Eulercharakteristik erhält[6], macht die Ränder so schmal wie möglich.[153] Dadurch ist die Bestimmung der Flächengröße genauer – bei Tröpfchenbildern wird also der Radius der Tröpfchen akkurater erkannt.

5.1.3.5. Randflächen entfernen

Da über den weiteren Verlauf von Flächen, die den Rand berühren, nichts bekannt ist, werden diese Flächen entfernt. Dazu wird jedes weiße Randpixel schwarz angefärbt. Alle Nachbarpixel werden danach auch als Randpixel behandelt. Es wird also ein „flood fill" vom Rand her durchgeführt. Ein positiver Nebeneffekt ist dabei, dass die Hintergrundfläche zwischen interessanten Flächen auch oft an den Rand stößt und mit diesem Schritt auch entfernt wird.

Tabelle 5.1.: Schwarzweißbildaufbereitungsoptionen

Optionsname	Verwendeter Algorithmus	Standardmäßig aktiviert	Standardparameter
Löcher auffüllen	F.4	✓	Maximalfläche = 1
Sackgassen entfernen	F.5	✗	–
Brücken entfernen	F.6 (auf Negativ)	✓	–
Ränder minimieren	MATLAB[7]	✓	–
Randflächen entfernen	F.7	✓	–

[5] Die Erweiterung auf zwei Pixel hat den unerwünschten Nebeneffekt, dass Drei-Pixel-Lücken auch geschlossen werden, obwohl diagonale Zwei-Pixel-Lücken, die kürzer sind, nicht geschlossen werden.
[6] Konkret bedeutet das, dass der Algorithmus keine Löcher erzeugt oder schließt.
[7] Mit der Funktion bwmorph aus der Image Processing Toolbox mit der „thin" Operation.

5.1.4. Segmentierung

Im nächsten Schritt müssen aus dem Schwarzweißbild die einzelnen weißen Bereiche – im Folgenden als „Regionen“ bezeichnet – als zusammenhängend erkannt und extrahiert werden. Dabei können die erkannten Bereiche noch verbessert und selektiert werden.

5.1.4.1. Segmentationsalgorithmus

Der Algorithmus, mit dem das Schwarzweißbild segmentiert wird, ist recht einfach[8]: Es werden alle weißen Pixel durchgegangen und geprüft, ob das Pixel schon eine Nummer hat. Wenn das nicht der Fall ist, wird ihm eine Nummer zugewiesen und alle damit verbundenen weißen Pixel bekommen die gleiche Nummer zugeordnet[9]. Wenn es schon eine Nummer hat, ist es bereits Teil einer Region und es muss nichts weiter unternommen werden. Dieses Verfahren wird so lange durchgeführt, bis alle weißen Pixel eine Nummer haben. Alle Pixel mit der gleichen Nummer gehören dann zur gleichen Region.

5.1.4.2. Regionsmaße

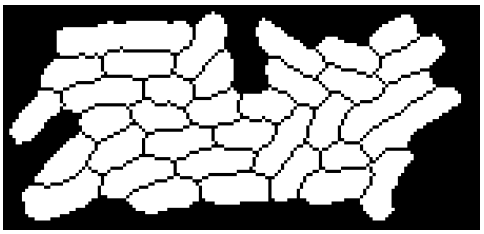
Um die Regionen in der Video- und Experimentanalyse handhaben zu können, werden jeder Region verschiedene Maße für Form und Position zugeordnet:

Fläche A**:** Anzahl der Pixel.

Umfang p**:** Länge der umlaufenden Linie. Hierbei wurde ein Algorithmus, der auf kreisförmige Flächen ausgelegt ist, verwendet. (s. F.8)

Schwerpunkt (x, y)**:** x- und y-Koordinate des Flächenschwerpunktes. Da jedes weiße Pixel gleich gewichtet wird, sind diese Koordinaten die Mittelwerte der x- bzw. y-Koordinaten aller weißen Pixel.

(a) Bild ohne Segmentierung



(b) Bild mit Segmentierung

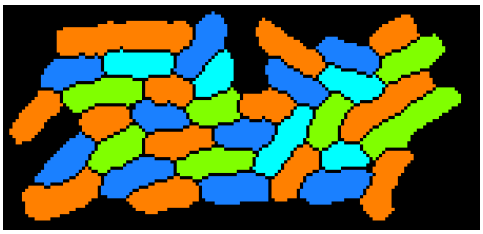


Abbildung 5.13.: Illustration der Segmentierung.

[8] Das Segmentieren wird mit der bwconncomp Funktion der Image processing toolbox von MATLAB durchgeführt.

[9] Über einen „flood fill“-Algorithmus.

Kreisförmigkeit *cy*: Maß, wie ähnlich die Fläche einem Kreis ist. Dazu wird der aus der Fläche berechnete Radius mit dem aus dem Umfang berechneten verglichen:

$$cy = \frac{\sqrt{\frac{A}{\pi}}}{\frac{p}{2\pi}} \tag{5.1.20}$$

Ein perfekter Kreis hat folglich den Wert 1 und eine unendliche Linie den Wert 0.

Konkavität *co*: Verhältnis zwischen der Fläche der Originalregion und der Fläche der Region, die entsteht, wenn alle konkaven Bereiche entfernt und somit die Fläche konvex gemacht wurde.

Ellipseneigenschaften: die Ellipse, die das gleiche zweite Moment wie die Region besitzt, hat weitere Eigenschaften:

Länge der größeren Halbachse a

Länge der kleineren Halbachse b

Orientierung *o*: Richtung, in die die größere Halbachse der Ellipse zeigt.

Exzentrizität ϵ: ähnlich wie *cy* ein Maß für die Ähnlichkeit mit einem Kreis. Durch die andere Berechnungsmethode hat sie aber nicht die gleiche Aussage:

$$\epsilon = \frac{\sqrt{a^2 + b^2}}{a} \tag{5.1.21}$$

Ein Kreis weist einen Wert von 0 und eine unendliche Linie den Wert 1 auf.

5.1.4.3. Regionstrennung

Trotz aller Aufbereitungen des Schwarzweißbildes kann es vorkommen, dass eigentlich getrennte Bereiche irrtümlich zu einer Region zusammengefasst werden (s. Abb. 5.14a). Um sie zu separieren, wird folgender Trennungsalgorithmus angewendet:

Zuerst wird aus der falsch zusammengefassten Region ein Schwarzweißbild erzeugt, in dem nur diese Region weiß dargestellt ist. Dann wird jedes schwarze Pixel, das nicht vom Rand aus erreichbar ist – also Löcher –, weiß eingefärbt. Anschließend wird jedem weißen Pixel der Abstand zum nächsten schwarzen Pixel als Wert zugewiesen und dieses Grauwertbild dann über eine Faltung mit einer zweidimensionalen Gaußfunktion (s. F.2) mit Breite σ[10] geglättet. So ergeben sich glatte lokale Maxima, die durch sogenannte Wasserscheidelinien getrennt werden[154][11]. Diese Wasserscheidelinien sind nun die Grenzen für die neuen Regionen, die weiter verwendet werden (s. Abb. 5.14b).

Da dieser Algorithmus sehr rechenintensiv und langsam ist, wird er nur dann eingesetzt, wenn es auch nötig ist. Und auch in diesen Fällen wird er nur auf Bereiche angewendet,

[10] Für σ wurde meist der Wert 8 Pixel verwendet, was meist gute Ergebnisse lieferte.
[11] Dafür wird die Funktion watershed der Image processing toolbox von MATLAB verwendet.

(a) Zu trennende Region mit $\epsilon = 0.76$.

(b) Getrennte Regionen, die anschließend unabhängig weiterverarbeitet werden.

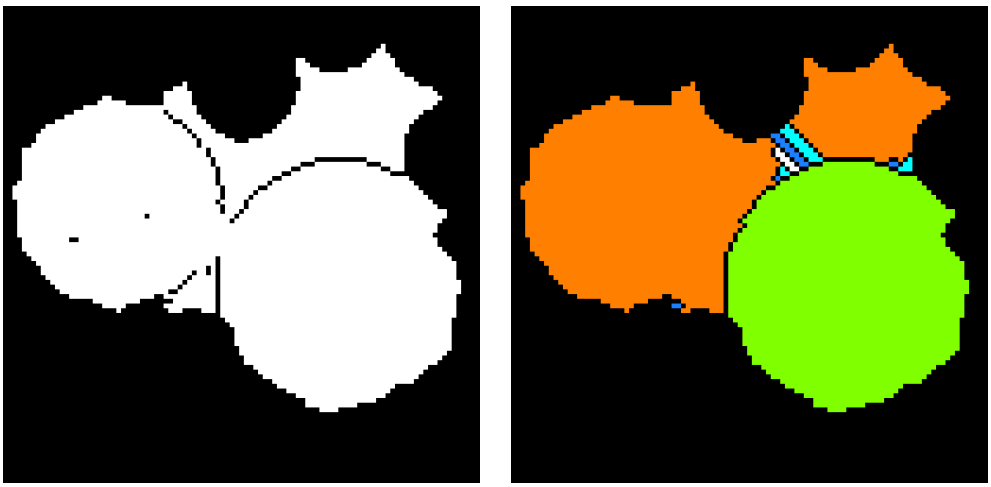


Abbildung 5.14.: Illustration der Regionstrennung.

die potentiell aus mehreren Regionen bestehen. Bei Tröpfchenregionen wird deswegen ein Schwellwert bei ϵ verwendet[12], der überschritten werden muss, damit die Regionstrennung durchgeführt wird.

5.1.4.4. Regionsauswahl

Die Maße aus 5.1.4.2 können verwenden werden, um aus der Menge der Regionen diejenigen zu entfernen, die keine validen Regionen repräsentieren und/oder nicht ausgewertet werden können.

So sind sehr kleine Regionen[13] sehr schwer in der Videoanalyse behandelbar und enthalten zu wenig Pixel, als dass sich das stochastische Rauschen der einzelnen Pixelwerte herausmitteln ließe. Sehr große Flächen hingegen sind oft unbrauchbar, da sie meist uninteressante Flächen[14] oder die Vereinigung mehrerer Flächen[15] repräsentieren.

Auch die anderen Maße können, abhängig vom Experiment, gute Klassifizierungsmaße darstellen. Wenn man z. B. Bakterien untersucht, kann man für cy eine obere Grenze und/oder für ϵ eine untere Grenze wählen, um Kreise[16] aus der Analyse auszuschließen. Bei der Arbeit mit Tröpfchen kann der komplementäre Weg gegangen werden, um alles, was nicht rund ist, zu entfernen.

[12] Als Schwellwert für ϵ wurde meist 0.6 verwendet.
[13] Ein sinnvoller Grenzwert dafür sind 5 Pixel.
[14] Zum Beispiel uninteressante Zwischenflächen.
[15] Dadurch beobachtet man nicht mehr das Verhalten der einzelnen Flächen, sondern eine Mittelung, die unbrauchbare Daten für die Experimentalanalyse erzeugen kann.
[16] Erzeugt durch Verunreinigungen auf der Kamera oder in der Reaktionskammer.

Anstatt einer einfachen Schwellwertbehandlung könnte man die Maße auch benutzen, um die Regionen in Kategorien mit ähnlichen Eigenschaften einzuteilen[155–157] und dann nur eine gewisse Kategorie weiterverwenden. Dies wurde aber in dieser Arbeit nicht angewendet, da die einfache Klassifizierung gut genug funktionierte.

5.1.5. Fluoreszenzdatenermittlung

Die Maße aus 5.1.4.2 sind gute Beschreibungen der Form und Position der Regionen, sind aber bei der endgültigen Analyse in 5.3 nur bedingt hilfreich. Da die Experimente Fluoreszenz zur Auslese verwenden, müssen die Fluoreszenzdaten der einzelnen Regionen bestimmt werden. Dazu werden die Pixel, die zu einer Region gehören, im Fluoreszenzbild jedes Fluoreszenzkanals ausgelesen. Aus diesem Datensatz werden nun Fluoreszenzmaße für jede Region und jeden Fluoreszenzkanal bestimmt:

Summe aller Pixelwerte. Diese ist meist kein gutes Maß, da sie mit der Größe der Regionen skaliert und Fluktuationen in der Fläche A, die durch das Segmentieren entstehen, direkt wiedergibt.

Mittelwert der Datenpunkte. Dieser kann direkt aus der Summe und der Fläche A berechnet werden und stellt ein stabileres Maß dar.

Minimum des Datensatzes. Hierbei wird nicht der absolute Minimalpunkt verwendet, da sonst das Pixelrauschen das Maß unbrauchbar machen würde. Deswegen wird der Mittelwert der fünf kleinsten Werte verwendet. Da diese aber meistens bei Randpixeln und/oder Hintergrundpixeln liegen, stellt dieses Maß nur eine Kontrolle dar und wird in der Analyse nicht verwendet.

Maximum der Pixelwerte. Auch hier wird – aus den selben Gründen wie beim Minimum – der Mittelwert der fünf größten Werte verwendet. In der Analyse ist dieses Maß meistens brauchbar und stabil.

„Helle“ Daten: Aus dem Datensatz der Pixel wird – mit der gleichen Methode wie in 5.1.2.1 – ein Schwellwert bestimmt, der die Pixel in „hell“ und „dunkel“ aufteilt. Die hellen Pixel stellen dann einen neuen Datensatz dar:

Helle Fläche: Die Anzahl der hellen Pixel ist die helle Fläche.

Summe der hellen Pixelwerte

Mittelwert des hellen Datensatzes: Kann – äquivalent zum normalen Mittelwert – auch direkt aus der Summe der hellen Pixelwerte und der hellen Fläche berechnet werden.

Je nach Experiment, Datenlage und Rauschverhalten muss immer individuell entschieden werden, welches Fluoreszenzmaß verwendet wird. So stellen die „hellen“ Maße bei Experimenten mit Bakterien in Tröpfchen ein gutes Maß dar, da nur die Pixel, die ein Bakterium darstellen, ausgewertet werden. Wenn in den Experimenten hingegen die Tröpfchen als Ganzes leuchten, liefern sie keine zusätzlichen Informationen.

5.2. Videoanalyse

5.2.1. Allgemeiner Ansatz

Nach der Einzelbildanalyse sind für jedes Einzelbild des Experimentvideos Regionen verfügbar. Um den zeitlichen Verlauf der einzelnen Regionen zu ermitteln, müssen Korrelationen zwischen den Regionen der verschiedenen Bilder hergestellt werden – das sogenannte „Tracking“. Dazu wird jede Region im Einzelbild A (es sei N_A die Anzahl der Regionen in A) mit allen Regionen des nächsten Einzelbildes B (N_B Stück) anhand seiner Maße $\mathcal{M} \in \mathbb{R}^n$ verglichen[17] und es wird eine Metrik erstellt, die einen abstrakten Abstand Δ zwischen den verglichenen Regionen darstellt:

$$\Delta\left(\mathcal{M}_i, \mathcal{M}_j\right) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \tag{5.2.1}$$

So kann man eine Abstandsmatrix $\Delta_{i,j}$ erstellen:

$$\Delta_{i,j} = \Delta\left(\mathcal{M}_i, \mathcal{M}_j\right) \text{ mit } \begin{array}{l} i \in \mathbb{N} \cap [1, N_A] \\ j \in \mathbb{N} \cap [1, N_B] \end{array} \tag{5.2.2}$$

Der kleinste Wert in $\Delta_{i,j}$ wird als erfolgreiche Zuordnung der Regionen i und j angesehen. Damit im Folgenden keine Mehrfachzuordnung geschehen kann, werden die Werte in Spalte j und Zeile i auf ∞ gesetzt:

$$\Delta_{i',j'} = \infty \begin{cases} \forall \quad j' = j \quad \wedge \quad i \in \mathbb{N} \cap [1, N_A] \\ \forall \quad i' = i \quad \wedge \quad j \in \mathbb{N} \cap [1, N_B] \end{cases} \tag{5.2.3}$$

Dadurch existiert wieder ein neuer kleinster Wert und die Zuordnungsprozedur wird solange wiederholt, bis alle Einträge auf ∞ gesetzt wurden.

Wenn N_A und N_B nicht gleich groß sind, was in der Regel der Fall ist, oder die Metrikdefinition so gestaltet ist, dass manche Maßkombinationen dort schon ∞ ergeben[18], gibt es immer ein paar Regionen, die nicht zugeordnet werden. Dies ist aber durchaus gewollt.

Dieser Ansatz funktioniert und ist leicht verständlich, ist aber sehr langsam in der Berechnung. Ein schnellerer Algorithmus ist folgender:

Aus $\Delta_{i,j}$ werden nur die Werte, die durch die Metrik einen endlichen Wert zugewiesen bekommen haben, betrachtet. Sie werden alle aufsteigend sortiert, wobei die Zuordnung an i und j erhalten bleibt. Diese sortierte Liste von $i \to j$ Zuweisungen wird nun von oben nach unten inspiziert und man behält nur die ersten Einträge, die ein spezifisches i *und* j enthalten. Es entsteht also am Ende eine Zuordnungsliste, in der jedes i und jedes j maximal einmal vorkommt.

[17] Wenn alle vorher erwähnten Maße verwendet werden, ist n gleich 17. Das ist aber nur bedingt sinnvoll.
[18] Zum Beispiel wenn sich die x- und y-Koordinaten zu stark unterscheiden, man also einen Schwellwert im maximalen Bewegungsabstand zwischen zwei Einzelbildern eingeführt hat.

5.2.2. Realisierung

5.2.2.1. Vollautomatisiertes Tracking

Um die Tröpfchen vollautomatisiert zu tracken, wurden nur drei Maße verwendet: die Fläche A sowie die x- und y-Koordinaten aus 5.1.4.2. Aus diesen drei Maßen berechnet man fünf abgeleitete Maße:

$$r_i = \sqrt{\frac{A_i}{\pi}} \qquad \text{Berechneter Radius der Fläche} \tag{5.2.4}$$

$$\varrho = r_j + r_i \qquad \text{Summe der Radien} \tag{5.2.5}$$

$$\Delta r = r_j - r_i \qquad \text{Änderung des Radius} \tag{5.2.6}$$

$$\Delta x = x_j - x_i \qquad \text{Änderung der } x\text{-Position} \tag{5.2.7}$$

$$\Delta y = y_j - y_i \qquad \text{Änderung der } y\text{-Position} \tag{5.2.8}$$

Um die Metrik zu bestimmen, werden zuerst Regeln aufgestellt, die festlegen, welche Regionen überhaupt zugeordnet werden dürfen.

- Der Radius darf sich maximal um den Faktor $\delta r_{\max}$ ändern. Diese Regel wird durch einen festen Grenzwert $\Delta r_{\max}$ ersetzt, wenn die absolute Änderungsgrenze kleiner als dieser ist.
- Die Zentren dürfen nicht weiter als $\varrho + \Delta p_{\max}$ voneinander entfernt sein. Bei zwei Kreisen bedeutet dies, dass sie sich überlappen oder die Ränder nicht weiter als $\Delta p_{\max}$ voneinander entfernt sind.

Dadurch ergibt sich die Formel für die Metrik:

$$\begin{aligned}\Delta\left(\mathcal{M}_i, \mathcal{M}_j\right) &= \Delta\left(\begin{pmatrix} x_i \\ y_i \\ r_i \end{pmatrix}, \begin{pmatrix} x_j \\ y_j \\ r_j \end{pmatrix}\right) = \Delta'\left(\varrho, \Delta r, \Delta x, \Delta y\right) \\ &= \begin{cases} \infty & \text{für } |\Delta r| \geq \max\left(\delta r_{\max} \cdot (\Delta r - \rho)/2, \Delta r_{\max}\right) \\ \infty & \text{für } \Delta x^2 + \Delta y^2 - \varrho^2 > \Delta p_{\max} \\ \Delta x^2 + \Delta y^2 + 4\Delta r^2 & \text{sonst} \end{cases}\end{aligned} \tag{5.2.9}$$

Typische Werte für die Schwellwertparameter, die ein gutes Ergebnis liefern, sind:

$$\begin{aligned}\delta r_{\max} &= 20\,\% \\ \Delta r_{\max} &= 2\,\text{Pixel} \\ \Delta p_{\max} &= 5\,\text{Pixel}\end{aligned}$$

Die Implementierung ist in F.9 dargestellt.

5.2.2.2. Semiautomatisches Tracking

Bei manchen Video(-typen) ist ein vollständig selbstständiges Bestimmen der Korrelationen nicht möglich. Das gilt vor allem für Videos, in denen keine Tröpfchen, sondern einzelne Bakterien segmentiert und verfolgt werden sollen[19]. Deswegen wird ein halbautomatischer Ansatz verwendet. Dabei wird versucht, bei den Bilderwechseln automatisch die Regionen zuzuordnen und zu entscheiden, ob die Auswahl verwendet werden soll. Wenn die Entscheidung negativ ausfällt, wird die gefundene Auswahl dem Nutzer vorgeschlagen. Dieser kann die Auswahl korrigieren und/oder die Einstellungen, nach denen die Regionen erzeugt werden (s. 5.1.2 und 5.1.3), ändern.

Hierbei wird für die Abstandsmetrik $\Delta_{i,j}$ die Überlappung zwischen der Region im Bild A und den Regionen in B als Maß verwendet. Die Entscheidung, ob die Zuordnung korrekt ist, wird nach zwei Kriterien gefällt:

1. Die Überlappung darf nicht kleiner sein als ein gewisser Prozentsatz $\eta_{\text{Überlapp}}$ der Fläche der Region F_A in Bild A.
2. Die Größe der Fläche der Region in Bild A F_A und in Bild B F_B darf sich nicht zu stark ändern. So muss das Verhältnis zwischen $\eta_{\min}$ und $\eta_{\max}$ liegen.

Dabei sind typische Werte:

$$\begin{aligned} \eta_{\text{Überlapp}} &= 20\,\% \\ \eta_{\min} &= 95\,\% \\ \eta_{\max} &= 110\,\% \end{aligned}$$

Diese können aber immer, wenn der Benutzer Einstellungen ändern und/oder eine Auswahl korrigieren kann, angepasst werden.

Unter anderem führt dies dazu, dass jeder Teilungsschritt von Bakterien per Hand zugeordnet werden muss.

[19] So ist der Kontrast der Bakterienränder sehr gering, weswegen die automatische Schwellwertbestimmung (s. 5.1.2.1) versagt. Auch verkompliziert die Tatsache, dass sich die Bakterien teilen und deswegen zu einem bestimmen Zeitpunkt aus einer Region zwei werden, das Automatisieren stark.

5.3. Experimentanalyse

Die zeitlich abhängigen Regionsmaße müssen für jedes Experiment unterschiedlich weiteranalysiert werden.

5.3.1. Oszillator in Tröpfchen

Der in 4.1.1 beschriebene Oszillator zeigt auch in den Tröpfchendaten oszillierende Fluoreszenzkurven. Im Folgenden wird beschrieben, wie diese Kurven weiter behandelt, verarbeitet und ausgewertet wurden.

5.3.1.1. Datensatzbestimmung

Zur Auswertung dieses Experiments werden zwei Werte benötigt: der Radius des Tröpfchens und die normierte Fluoreszenz. Der Radius wird, wie in Formel (5.2.4), für jeden Zeitpunkt berechnet und dann gemittelt. Die normierte Fluoreszenz wird aus dem Quotienten der beiden Fluoreszenzsignale errechnet. Da das Signal des Referenzfarbstoffs und des Auslesefarbstoffs über die gleiche Fläche berechnet wird, kann die Fluoreszenzsumme als gutes Maß verwendet werden.

$$x_{\text{normiert}} = \frac{x_{\text{Signal}}}{x_{\text{Referenz}}} \tag{5.3.1}$$

Um den Einfluss etwaiger Fehler bei der Bild- oder Videoanalyse zu minimieren[20], wird der Datensatz zuerst so beschnitten, dass der erste und der letzte Wert des Radius und der Referenzfluoreszenz nicht Null sind. Dann wird geprüft, ob einer dieser recht stabilen Werte eine Stufe enthält (s. F.3), und gegebenenfalls wird nur der Bereich vor der Stufe verwendet.

Da für die weitere Analyse das Signal um Null zentriert sein muss, wird eine sehr stark geglättete Version des Signals[21] abgezogen. Dies entfernt zusätzlich auch noch Langzeitschwankungen und Drift im Signal.

$$x_{\text{hintergrundkorrigiert}} = x_{\text{normiert}} - \text{Gaußfilter}\left(\sigma_{\text{Hintergrund}}, x_{\text{normiert}}\right) \tag{5.3.2}$$

Rauschen auf dem Signal wird durch eine weitere, weniger starke Filterung[22] entfernt und die Differenz zwischen dem hintergrundkorrigierten und dem gefilterten Signal wird als Rauschen deklariert.

$$x_{\text{gefiltert}} = \text{Gaußfilter}\left(\sigma_{\text{Filter}}, x_{\text{hintergrundkorrigiert}}\right) \tag{5.3.3}$$

$$x_{\text{Rauschen}} = x_{\text{hintergrundkorrigiert}} - x_{\text{gefiltert}} \tag{5.3.4}$$

[20] Am Anfang des Projektes wurde noch eine andere Software verwendet, die diese Artefakte manchmal produzierte: CellEvaluator[133].
[21] Mit dem Gaußfilter F.2 und einem hohen $\sigma_{\text{Hintergrund}}$.
[22] Mit einem σ_{Filter}, das um den Faktor 10 kleiner ist als $\sigma_{\text{Hintergrund}}$.

5.3.1.2. Extrema bestimmen

In dem gefilterten Signal können nun die zeitlichen Positionen der lokalen Extrema bestimmt werden und daraus die Periode der Oszillation abgeschätzt werden.

Die Bestimmung der Extrema erfolgt über vier Kriterien:

1. Es muss ein Vorzeichenwechsel zwischen den Extrema liegen.
2. Jedes Extremum ist das globale Extremum in seinem Vorzeichenblock.
3. Die Wertdifferenz zwischen zwei aufeinander folgenden Extrema muss größer sein als das doppelte Rauschlevel[23].
4. Wenn ein Extremum durch das vorherige Kriterium entfernt wurde, werden die beiden benachbarten „Vorzeichenblocks“ verschmolzen und als ein einziger behandelt.

Die so gefundenen Extrema werden nun verwendet, um die Periode und die Amplitude jedes Datensatzes zu berechnen. Dabei wird folgendermaßen vorgegangen:

1. Wenn sich das erste Extremum an der allerersten Zeitposition befindet, wird es verworfen, da man nicht wissen kann, ob das Extremum nicht in Wirklichkeit davor liegt und man nur die Flanke aufgezeichnet hat.
2. Analog wird mit dem letzten Extremum verfahren, wenn es sich an der allerletzten Zeitposition befindet.
3. Wenn nun weniger als fünf Extrema übrig bleiben, wird der Datensatz als nicht oszillierend angesehen und die weitere Perioden- und Amplitudenanalyse nicht durchgeführt. Da fünf Extrema zwei volle Perioden der Oszillation bedeuten und nur 1000 Minuten aufgezeichnet wurden, ist ein Datensatz mit weniger als fünf Extrema entweder nicht oszillierend oder die Periode ist länger als 500 Minuten.
4. Die Periode eines oszillierenden Datensatzes wird nur aus der Zeitdifferenz zwischen dem $n$ten und $m$ten Extremum berechnet:
$$T_{n,m} = \frac{t_m - t_n}{\frac{1}{2}(m-n)} \tag{5.3.5}$$
Es wurden $n = 1$ und $m \in [3, 4, 5]$ verwendet. Empirisch ergaben sich mit $m = 3$ die verlässlichsten Daten.
5. Die Amplitude wird über die maximale absolute Differenz zwischen zwei aufeinander folgenden Extrema bestimmt. Dabei werden aber nicht alle Extrema verwendet. So zeigt der Oszillator in den Referenzmessungen am Anfang ein sehr tiefes Tal (s. z. B. Abb. 7.1). Da dieses aber nicht bei allen Datensätzen mit erfasst wird, muss es bei den Datensätzen, die es erfassen, entfernt werden, um vergleichbare Daten zu erhalten. Deswegen wird das erste Extremum verworfen, wenn es ein Minimum ist. Insgesamt werden auch maximal die ersten fünf Extrema beachtet.

[23] Das Rauschlevel ist die Standardabweichung von x_{Rauschen}.

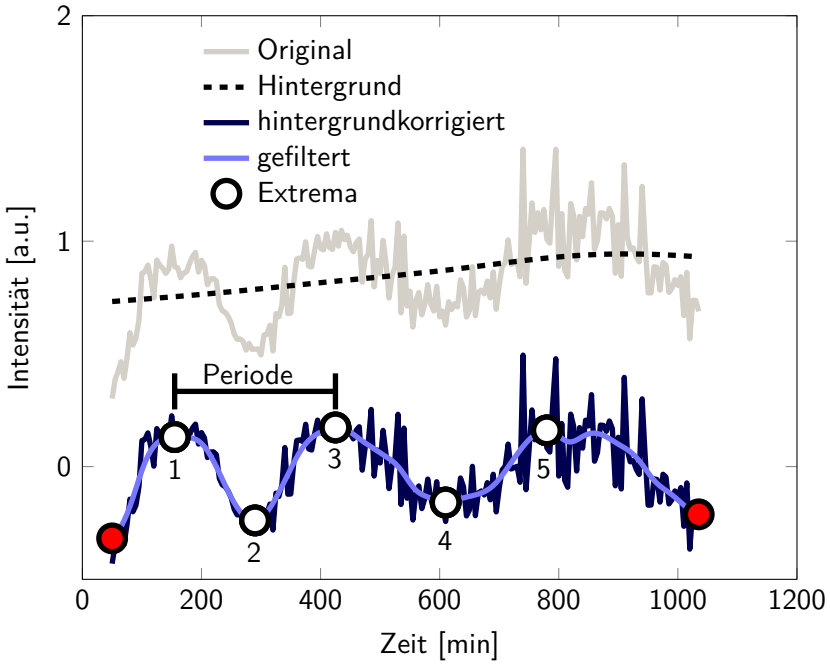


Abbildung 5.15.: Illustration der Datenverarbeitung beim Oszillator in Tröpfchen. Die durch das Referenzsignal normierten Daten (graue Linie) werden um den Hintergrund (gestrichelte schwarze Linie) verschoben, um das Signal um Null zu zentrieren (dunkelblaue Linie). Anschließend werden sie gefiltert (blaue Linie) und die Extrema bestimmt (Kreise). Das erste und das letzte Extremum werden verworfen (rot gefüllte Kreise), da nicht sicher ist, dass das echte Extremum nicht davor bzw. danach liegt. Da fünf Extrema übrig bleiben, wird die Kurve als oszillierend markiert und die zeitliche Differenz zwischen dem ersten und dritten Extremum wird als Periode verwendet. Die Wertdifferenz zwischen dem zweiten und dritten Extremum wird als Amplitude angenommen (nicht eingezeichnet).

5.3.1.3. Datensatzauswahl

Da fälschlich als oszillierend markierte Datensätze Artefakte in den endgültigen Daten erzeugen können, werden recht strikte Auswahlregeln angewendet, um diese zu entfernen:

1. Zu kurze Datensätze – mit weniger als 40 Datenpunkten bzw. kürzer als 200 min – werden verworfen.
2. Wenn zu irgendeinem Zeitpunkt der Referenzfluoreszenzwert eines Datensatzes Null ist[24], ist er auch zu ignorieren (s. Abb. 5.16a).
3. Da der Radius eines Tröpfchens eine Konstante sein sollte, werden Datensätze entfernt, bei denen dieser zu stark schwankt (Standardabweichung größer als 10 % des Mittelwerts)[25] (s. Abb. 5.16b).

[24]Das heißt, das Tröpfchen enthielt zu wenig Referenzfarbstoff oder wurde nicht richtig zugeordnet.
[25]Solche Datensätze repräsentieren meistens keine Tröpfchen, sondern Zwischenräume.

4. Ein Signal-zu-Rausch-Verhältnis[26] von weniger als 1 entfernt ebenfalls einen Datensatz. Dieses Kriterium kann auch Datensätze mit einer sehr kleinen Amplitude oder nicht oszillierende Datensätze entfernen (s. Abb. 5.16c).

5. Wenn die Oszillation eines Datensatzes mit der Zeit beschleunigen, wird er auch verworfen. Aus den Simulationen und den Ensemblemessungen ist bekannt, dass der Oszillator mit der Zeit langsamer wird. Deswegen ist eine Beschleunigung unerwartet, könnte aber durchaus real sein.

   Nähere Betrachtungen der Datensätze zeigen aber, dass sie eine stark verrauschte Region besitzen, in der normalerweise das zweite Maximum der Oszillation liegt. Bei dieser Beschleunigung handelt es sich somit um ein Artefakt. Auch für einen menschlichen Betrachter ist es nicht immer möglich, eindeutig zu klassifizieren, ob sich in der verrauschten Region ein Extremum befindet oder nicht. Um sicherzugehen, dass nur korrekt analysierte Datensätze weiterverwendet werden, werden folglich alle schneller werdenden Oszillationen entfernt (s. Abb. 5.16d).

6. Die bestimmte Periode wird mit einer Frequenzabschätzung durch eine diskrete Fouriertransformation (DFT) verglichen. Wenn das nächste lokale Maximum in der Nähe der berechneten Frequenz zu klein (kleiner als ein Drittel des globalen Maximums) oder zu weit entfernt ist, wird der Datensatz verworfen (s. Abb. 5.16e).

   Dies ist nötig, da bei einer stark verrauschten Datenlage der Extremumsalgorithmus manchmal versagt und dadurch doppelt getestet wird, ob er richtig gearbeitet hat.

7. Ein kleines Tröpfchen, das unter einem größeren Tröpfchen sitzt, wird als solches erkannt, aber das Fluoreszenzsignal wird durch das Größere dominiert. Da dies falsche Größenzuordnungen bedeuten würde, wurden diese Tröpfchen über ein heuristisches Kriterium entfernt:

   Solche Tröpfchen bilden eine erkennbar separierte Population, die eine höhere Referenzfluoreszenz hat als gleichgroße „normale“ Tröpfchen (s. Abb. 5.16f).

Dieses rigoroses Aussortieren ist eindeutig darauf fokusiert, valide und gut verarbeitete Tröpfchen zu identifizieren, und man erwartet dadurch keine Veränderungen der statistischen Eigenschaften der Tröpfchenpopulationen. So ist zu bemerken, dass nach der Filterung immer noch nicht merkbar oszillierende Datensätze vorhanden sind (s. Tab. 5.2). Aber eine Aussage über das wirkliche Verhältnis zwischen oszillierenden und nicht oszillierenden Daten kann nun nicht mehr getroffen werden, da die Kriterien nicht oszillierende eher aussortieren als oszillierende.

[26] Als Signal wurde die Standardabweichung von $x_{\text{gefiltert}}$ und als Rauschen die Standardabweichung von x_{Rauschen} definiert.

Tabelle 5.2.: Anteil der entfernten Datensätze bei jedem Kriterium und für jedes Experiment

Kriterium	Stabil	Gedämpft	Stark gedämpft
1. Zu wenige Datenpunkte	44.24 %	87.15 %	28.39 %
2. Schlechte Normierung	22.65 %	0.06 %	2.58 %
3. Schlechte Radiusdaten	5.08 %	2.21 %	2.23 %
4. Starkes Rauschen	9.61 %	0.78 %	5.46 %
5. Beschleunigung	0.14 %	0.04 %	0.30 %
6. Schlechte DFT	1.40 %	0.29 %	0.78 %
7. Unter einem größeren Tröpfchen	4.46 %	2.78 %	0.89 %
Erfüllen alle Kriterien	12.43 %	6.69 %	14.39 %
5 oder mehr Extrema: „oszillierend“	4.99 %	3.71 %	2.58 %

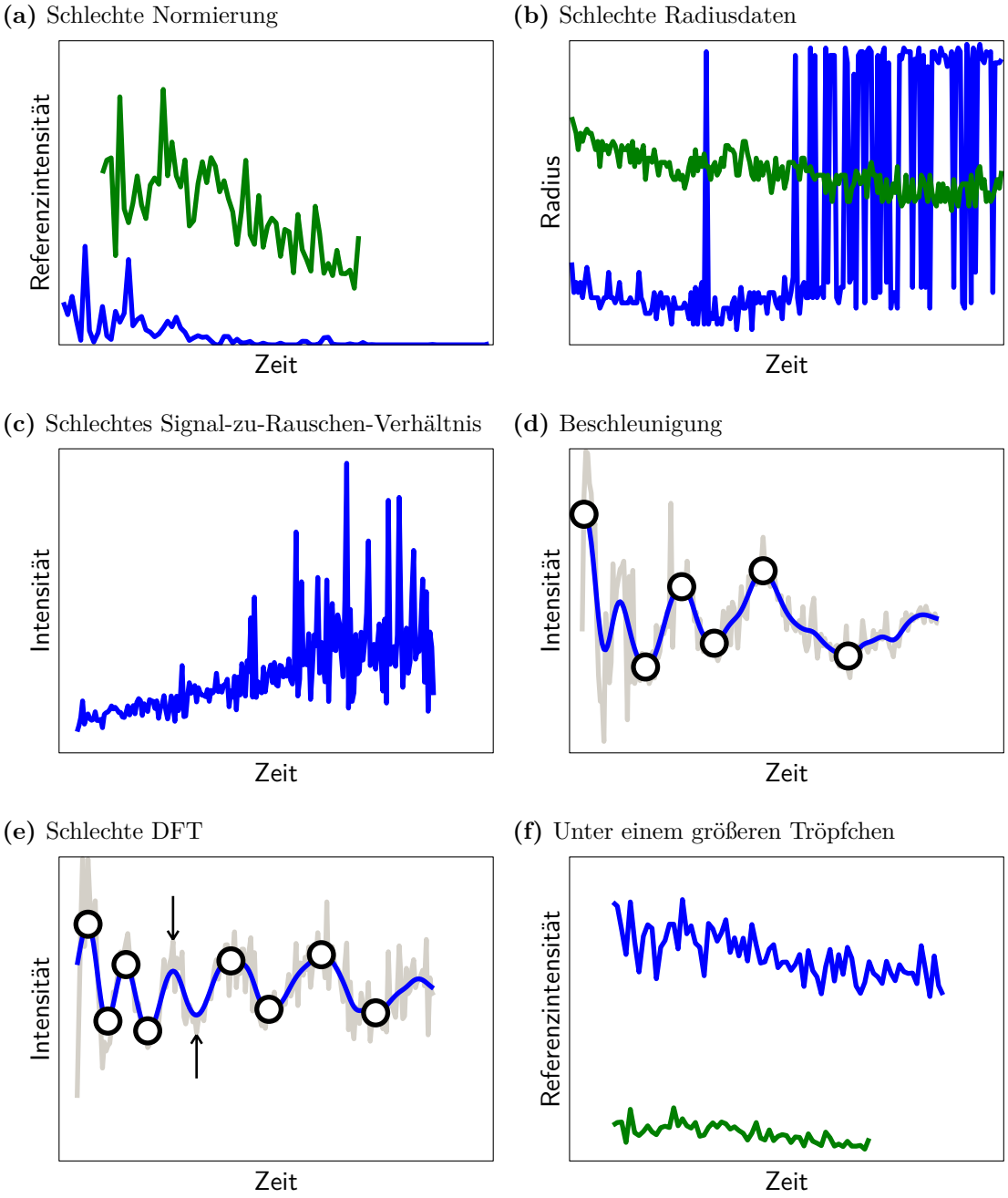


Abbildung 5.16.: Beispielkurven für jeden Filter. Grüne Linien passieren und Blaue werden verworfen. **(a)** Die invalide Kurve nimmt zwischenzeitlich den Wert Null an. **(b)** Radiusdaten mit starken Fluktuationen. Da der Radius aber recht stabil sein sollte, kann das kein gut verarbeitetes Tröpfchen sein. **(c)** Beispielkurve mit sehr viel Rauschen. **(d)** Durch das hohe Rauschen wird die erste Periode nicht gut erkannt und deswegen ist die zweite detektierte Periode kürzer als die erste. **(e)** Weiteres Beispiel, in dem die Extremadetektion wegen des relativ hohen Rauschens versagt. Deswegen wird im Fourierraum kein passender Modus erkannt. **(f)** Zwei Datensätze mit ähnlichen Radien. Da sie ähnliche Referenzintensitäten haben sollten, muss das Tröpfchen mit den höheren Werten unter einem größeren liegen.

5.3.2. Bakterien in Tröpfchen

In den Tröpfchenfluoreszenzkurven der Experimente mit den Bakterien (s. 4.1.2) ist kein so komplexes Verhalten zu sehen wie in den Oszillatorexperimenten. Deswegen ist die Experimentanalyse nicht so kompliziert.

Die Empfänger-Bakterien beginnen aber abhängig von ihrer Lage zu den Signalstoff- oder Sender-Bakterien-Tröpfchen zu leuchten. So muss man zwischen den verschiedenen Tröpfchensorten unterscheiden und die Abstände bestimmen. Die Abstandsabhängigkeit der Fluoreszenzantwort kann dann aus dem Gesamtdatensatz aller Fluoreszenzkurven extrahiert werden.

5.3.2.1. Datensatzauswahl

Die Kriterien für einen validen Datensatz aus einem Tröpfchen ist hierbei:

1. Die Zeitreihe muss mindestens 20 Datenpunkte lang sein.
2. Da die Tröpfchen mit der Mikrofluidik relativ monodispers hergestellt werden, wird der mittlere Radius als Selektionskriterium für ein valides Tröpfchen verwendet. Deswegen werden alle Regionen, die einen mittleren Radius außerhalb eines recht kleinen Bereichs haben, verworfen.
3. Um die restlichen Tröpfchenzwischenräume zu entfernen, wird eine Kreisförmigkeit cy von mindestens 0.8 gefordert. Da die Tröpfchen alle ähnliche Größen haben, gibt es nur sehr wenige Tröpfchen, die unter oder über einem anderen liegen. Deswegen sind die erkannten Tröpfchen alle recht kreisförmig und es wurden durch dieses Kriterium fast keine „echten“ Tröpfchen aussortiert.

5.3.2.2. Tröpfchenklassifizierung

Die verschiedenen Tröpfchenarten müssen als erstes getrennt werden. Da die AHL- bzw. IPTG-Tröpfchen auch einen Referenzfarbstoff enthielten, ist es recht einfach, diese zu selektieren. Dazu wird der Mittelwert – sowohl zeitlich als auch pixelweise – des Fluoreszenzwertes im Referenzkanal für jedes Tröpfchen genommen. Darin kann man immer einen Schwellwert definieren, der die Tröpfchensorten separiert.

In den Tröpfchensorten, die Bakterien enthielten, müssen die Tröpfchen, die bei der Tröpfchenproduktion zufällig leer blieben, aussortiert werden. Da die Produktion des fluoreszierenden Proteins nie komplett deaktiviert werden kann, zeigt auch ein Tröpfchen, das keine Signalstoffe bekommt, eine leichte Fluoreszenz im Proteinkanal. Deswegen kann meistens ebenfalls der Weg über den Mittelwert gewählt werden, um die leeren Tröpfchen zu entfernen – nur in Fällen extremen Rauschens bzw. niedriger Fluoreszenzintensität muss per Auge entschieden werden.

5.3.2.3. Abstandsbestimmung

Um die Diffusivität der Signalstoffe zu untersuchen, muss als nächstes die relative räumliche Information zwischen den Signaltröpfchen und den Empfängerbakterientröpfchen ermittelt und in eine vergleichbare Form gebracht werden. Dazu wird eine „nächster Nachbar"-Vereinfachung durchgeführt, die die Daten vergleichbar zur Simulation macht. Dabei wird für jedes Empfängertröpfchen zu jedem Zeitpunkt das nächste Sendertröpfchen ermittelt und der Abstand Δ_t zu diesem bestimmt. Da die Vorgänge in den Empfängertröpfchen von der Vergangenheit abhängen, wird nun für jeden Zeitpunkt der Mittelwert der Abstände in der Vergangenheit errechnet:

$$\bar{\Delta}_t = \frac{1}{t} \sum_{i=1}^{t} \Delta_i \tag{5.3.6}$$

Dieser kumulative Mittelwert wird nun verwendet, um die Tröpfchen zu jedem Zeitpunkt in einen Abstandsbin einzuteilen. Somit ergibt sich zu jedem Zeitpunkt t für jeden Abstandsbin einen Satz $\Omega_{\mathrm{Bin},t}$ von Tröpfchen, die sich in der Vergangenheit im Mittel innerhalb des Bins befanden.

Es ist zu beachten, dass der Einfluss von weiter entfernten Sendertröpfchen oder von Tröpfchen, die sich außerhalb des beobachteten Bildbereiches befanden, vernachlässigt wird. Dadurch können theoretisch zwei Tröpfchen, die völlig verschiedene Umgebungen haben – eines z. B. mit nur einem einzigen Sendertröpfchen in einem bestimmten Abstand und das andere mit einem kompletten Ring aus Sendertröpfchen mit dem gleichen Radius – und dadurch unterschiedliche zeitliche Verläufe der Signalkonzentration erfahren, den gleichen Abstand zugeordnet bekommen. Durch die Bildausschnittsauswahl wurde aber versucht, die Umgebung möglichst homogen auszusuchen.

5.3.2.4. Datensatzbestimmung

Die Fluoreszenzdaten für den jeweiligen Bin werden dann aus dem Summenmaß der Tröpfchen bestimmt. Dazu wird zuerst die Summe auf das Volumen des Tröpfchens normiert:

$$x_{\mathrm{normiert},t} = \frac{x_{\mathrm{Summe},t}}{\frac{4}{3}\pi r_t^3} \tag{5.3.7}$$

Um das Rauschen zu reduzieren, werden die Kurven dann noch mit einem Gaußfilter (s. F.2) gefiltert:

$$x_{\mathrm{gefiltert}} = \mathrm{Gaußfilter}(5, x_{\mathrm{normiert}}) \tag{5.3.8}$$

Die Fluoreszenzwerte der Bins werden nun durch Mittelung über die Fluoreszenzwerte der Tröpfchen, die sich zu dem Zeitpunkt im Bin befinden, bestimmt:

$$x_{\mathrm{Bin},t} = \frac{\sum_{i \in \Omega_{\mathrm{Bin},t}} x_{i,\mathrm{gefiltert},t}}{\sum_{i \in \Omega_{\mathrm{Bin},t}} 1} \tag{5.3.9}$$

Diese Binzeitkurven geben dann Auskunft über die abstandsabhängige zeitliche Fluoreszenzantwort der Bakterien.

5.3.3. Stochastik in Tröpfchen

Das einfache Reaktionsnetzwerk der Stochastikexperimente (s. 4.1.3) erzeugt ein simples Fluoreszenzsignal: zuerst einen niedrigen Wert und dann einen Anstieg bis zur Sättigung. Über ein vereinfachtes Modell (s. 7.3.2) kann dieses beschrieben werden. Im Folgenden wird dargelegt, wie die Tröpfchenfluoreszenzkurven verarbeitet, ausgewertet und ausgewählt wurden.

5.3.3.1. Datensatzbestimmung

Um die statistischen Eigenschaften der Tröpfchen zu untersuchen, werden auch hier ihr Radius und die normierte Fluoreszenz benötigt. Der Radius wird wieder nach Formel (5.2.4) berechnet, aber die Normierung wird anders durchgeführt. Für das vereinfachte Modell, das an den Fluoreszenzdatensatz jedes Tröpfchens angefittet wird, müssen die Daten bei 0 beginnen und bei 1 enden. Dazu wird das Maß des Fluoreszenzmaximums verwendet und durch den Mittelwert der 10000 dunkelsten Pixel des zugehörigen Fluoreszenzbildes geteilt, um globale Schwankungen zu reduzieren.

$$\text{Hintergrund} = \left\langle \min_{10000} (\text{Fluoreszenzbild}) \right\rangle \tag{5.3.10}$$

$$x_{\text{hintergrundkorrigiert}} = \frac{x_{\text{Signal}}}{\text{Hintergrund}} \tag{5.3.11}$$

Dann wird der Mittelwert der ersten fünf Datenpunkte als Minimalwert und der Mittelwert aller Datenpunkte nach dem vierzigsten Datenpunkt als Maximalwert genommen und der Datensatz dementsprechend normiert.

$$\text{Minimum} = \langle x_{\text{hintergrundkorrigiert}} \, (\text{Datenpunkt 1 - 5}) \rangle \tag{5.3.12}$$

$$\text{Maximum} = \langle x_{\text{hintergrundkorrigiert}} \, (\text{Datenpunkt 40 - Ende}) \rangle \tag{5.3.13}$$

$$x_{\text{normiert}} = \frac{x_{\text{hintergrundkorrigiert}} - \text{Minimum}}{\text{Maximum} - \text{Minimum}} \tag{5.3.14}$$

Hierbei ist zu beachten, dass das Minimum über 5 und das Maximum über mindestens 35 Datenpunkte bestimmt wird. Deswegen ist der Wert des Maximums verlässlicher als der des Minimums. Es können nicht mehr Datenpunkte für die Minimumsbestimmung verwendet werden, da in manchen Datensätzen der Minimumszustand bereits beim sechsten Datenpunkt verlassen wurde – in wenigen Fällen sogar noch früher.

Anschließend wird der Datensatz noch mit einem Medianfilter über ein drei Datenpunkte großes Fenster (s. F.1) gefiltert, um das Rauschen zu reduzieren:

$$x_{\text{gefiltert}} = \text{Medianfilter}(3, x_{\text{normiert}}) \tag{5.3.15}$$

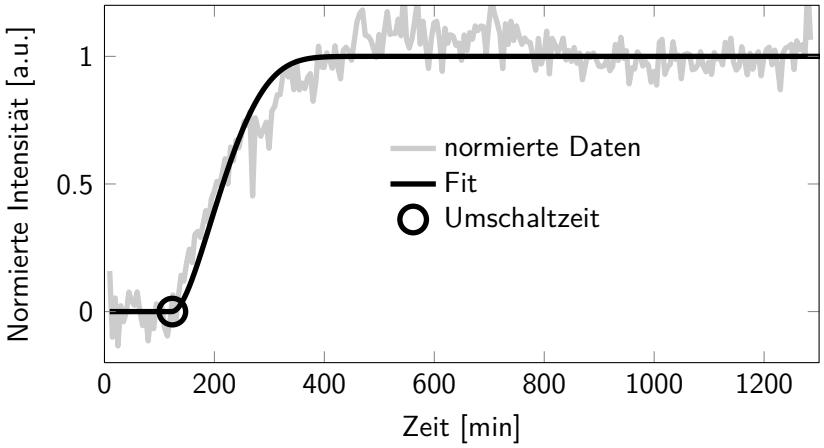


Abbildung 5.17.: Illustration der Datenverarbeitung bei Stoachstik in Tröpfchen. An die normierten Daten (graue Linie) wird die vereinfachte Modellfunktion angefittet (schwarze Linie). Der Umschaltzeitpunkt zwischen den beiden Funktionsteilen ist mit einem Kreis markiert.

5.3.3.2. Produktionsrate bestimmen

Aus diesen normierten und gefilterten Datensätzen wird jetzt die RNA-Produktionsrate k_{prod} bestimmt. Dazu wird eine gemischte Funktion an die Daten angefittet, die bis zum Zeitpunkt

$$t_1 = \frac{c_{\mathrm{Inhibitor}}}{k_{\mathrm{prod}}} \tag{5.3.16}$$

den Wert Null annimmt und anschließend mit der verschobenen Zeit

$$\tilde{t} = t - t_1 \tag{5.3.17}$$

durch die Funktion (siehe später 7.3.2 und B.2.3 für die Herleitung)

$$x(t) = 1 - \frac{2 \cdot e^{k_{\mathrm{a}} \cdot \left(-\left(\frac{k_{\mathrm{prod}} \cdot \tilde{t}^2}{2}\right) + \tilde{t} \cdot R_{\mathrm{tot}}\right)}}{2 + \sqrt{k_{\mathrm{a}} \cdot 2 \cdot \frac{\pi}{k_{\mathrm{prod}}}} \cdot R_{\mathrm{tot}} \cdot e^{\frac{k_{\mathrm{a}} \cdot R_{\mathrm{tot}}^2}{2 \cdot k_{\mathrm{prod}}}} \cdot \left[\mathrm{erf}\left(\frac{\sqrt{k_{\mathrm{a}}} \cdot \left(k_{\mathrm{prod}} \cdot \tilde{t} - R_{\mathrm{tot}}\right)}{\sqrt{2 \cdot k_{\mathrm{prod}}}}\right) + \mathrm{erf}\left(\frac{\sqrt{k_{\mathrm{a}}} \cdot R_{\mathrm{tot}}}{\sqrt{2 \cdot k_{\mathrm{prod}}}}\right)\right]} \tag{5.3.18}$$

beschrieben wird. Dabei sind R_{tot} (Reporterkonzentration) und $c_{\mathrm{Inhibitor}}$ (Inhibitorkonzentration) gegeben durch das Experiment und $k_{\mathrm{a}} = 8.7 \times 10^2\,\mathrm{M}^{-1}\,\mathrm{s}^{-1}$ wurde durch einen Reaktionsmodellfit bestimmt (s. 7.3.1). Es wird also nur die Produktionsrate k_{prod} angefittet.

Tabelle 5.3.: Anzahl der Tröpfchen für jedes Experiment

Experimentkonzentration	Valide Tröpfchen	Tröpfchen mit gutem Fit
1 nM	6226	4615
5 nM	5741	5510
10 nM	3578	3309
50 nM	3036	1144

5.3.3.3. Datensatzauswahl

Auch hier werden wieder strikte Kriterien eingeführt, die sicherstellen sollen, dass nur valide und auswertbare Tröpfchen in den endgültigen Datensatz aufgenommen werden:

1. Der Datensatz muss mindestens 75 Datenpunkte enthalten, da er sonst mit dem Modell nicht gut auswertbar ist. Vor allem der Maximalwert, nach dem normiert wird, ist sonst einer zu starken Schwankung unterworfen.
2. Die untersuchte Reaktion findet ganz am Anfang des Experiments statt. Deswegen sind auch nur Datensätze interessant, die ganz am Anfang starten und nicht mitten im Video.
3. Um nur Tröpfchen und keine Tröpfchenzwischenräume zu analysieren, wird ein Schwellwert für den Mittelwert der Kreisförmigkeit *cy* von 0.9 eingeführt. Dadurch werden auch einige Tröpfchen aussortiert, aber da die Auswahl nur von der erkannten Form des Tröpfchens abhängt, wird das Endergebnis der statistischen Analyse dadurch nicht beeinflusst.
4. In manchen Fällen kann die Normierung versagen, was meistens am unsicher bestimmten Minimalwert liegt. Das äußert sich dann entweder darin, dass der Wertebereich von x_{normiert} weit ins Negative reicht, oder dass der Startwert größer als der Endwert ist. Deswegen werden alle Datensätze entfernt, deren Mindestwert nach der Normierung kleiner als -1 ist oder deren Startwert größer als der Endwert ist.
5. Bei manchen Datensätzen kann die Fitfunktion nicht angelegt werden, weil z. B. das Rauschen zu stark ist. Das schlägt sich dann in großen Unsicherheiten der Fitparameter nieder. Deswegen wird ein Datensatz verworfen, wenn eine oder beide Grenzen des Konfidenzintervalls[27] von $k_{\text{prod}} - \left[\Delta k_{\text{prod},\downarrow}, \Delta k_{\text{prod},\uparrow}\right]$ – weiter als 50 % vom gefitteten Parameterwert entfernt sind. Es werden also alle Datensätze entfernt, die folgende Ungleichung erfüllen:

$$\max\left(k_{\text{prod}} - \Delta k_{\text{prod},\downarrow}, \Delta k_{\text{prod},\uparrow} - k_{\text{prod}}\right) > 0.5 \cdot k_{\text{prod}} \tag{5.3.19}$$

[27] Es wurde ein Konfidenzbereich von 95 % verwendet.

5.4. Bemerkungen

In allen Aspekten der Datenanalyse ist hohe Objektivität ein Hauptkriterium. Deswegen wurde die Datenverarbeitung so weit wie möglich automatisiert und Entscheidungen wurden weitestgehend von einem Computer nach objektiven Richtlinien getroffen. Diese mussten dementsprechend sehr streng und restriktiv gewählt werden, damit keine invaliden Daten in den endgültigen Datensatz aufgenommen wurden. So wurden im Zweifelsfall Daten verworfen, die von einem Menschen eventuell hätten ausgewertet werden können. Trotz dieser Verschwendung kann wegen der großen Tröpfchenanzahl eine gute Statistik erhalten werden. Auch im seltenen Fall, dass ein ungültiges „Tröpfchen“ alle Kriterien besteht, schützt die hohe Anzahl an validen Tröpfchen vor einer Verfälschung der endgültigen Aussage. Zusätzlich ermöglicht der große Gesamtdatensatz Schlussfolgerungen und Interpretationen, die aus den Einzeldaten nicht ersichtlich sind.

6. Ergebnisse

Aus den Experimentanalysen (s. 5.3) konnten für jedes Experiment verschiedene Ergebnisse extrahiert werden. Diese sollen im Folgenden dargestellt werden. Eine Interpretation erfolgt in einem späteren Kapitel (s. 7).

6.1. Oszillator in Tröpfchen

6.1.1. Stabile Oszillation

Die stabile Stimmung des Oszillators zeigte im Referenzexperiment im makroskopischen Volumen sehr ausgeprägte Oszillationen (sechs in 1000 min mit einer Periode von 134 min), deren Amplitude mit der Zeit aber trotzdem etwas abnahm. Auch die Periodenlänge vergrößert sich etwas. Solches Verhalten konnte auch in den Tröpfchendaten beobachtet (s. Abb. 6.2) werden, aber die Tröpfchen zeigten ein sehr breites Spektrum an Verhalten. So gab es zum einen sehr viele verschiedene Periodenlängen, die sich teilweise sehr stark von der Referenz unterschieden, aber auch viele Tröpfchen, die gar nicht oszillierend eingeordnet wurden[1]. Interessant ist, dass die Mittelwertskurven der Tröpfchen – egal ob die nicht Oszillierenden mitbetrachtet wurden oder nicht – sehr wenige bis gar keine Oszillationen zeigen. Das resultiert aus den stark unterschiedlichen Perioden und die einzelnen Oszillatoren kommen deswegen sehr schnell außer Phase, was effektiv die Gesamtoszillationen dämpft oder gar unterdrückt. Wenn man also die Tröpfchenvolumina gedanklich zu einem makroskopischen Volumen zusammenfasst – ein großes Volumen also durch viele kleine simuliert –, erhält man ein anderes Ergebnis, als durch direkte makroskopische Messung.

Wenn man nun die Tröpfchen nach ihrer Größe in 2 µm große Bins aufteilt, kann man eine starke Abhängigkeit der mittleren Periode und Amplitude – und auch der stochastischen Breite – dieser Subpopulationen vom Radius erkennen (s. Abb. 6.1, linke Spalte). Sowohl Periode als auch Amplitude nehmen mit steigendem Radius ab. Auch die Breite der Periodenverteilung wird kleiner, aber die Amplitudenverteilung bleibt relativ konstant.

[1] Wenn eine Tröpfchenkurve als nicht oszillierend eingestuft wurde, heißt das nicht unbedingt, dass sie nicht oszilliert hat. So sind sehr schnell oszillierende Kurven, stark verrauschte Kurven oder Kurven mit einer kleinen Amplitude nicht als oszillierend gewertet worden, um die Menge der oszillierend eingestuften Kurven möglichst rigide „sauber“ zu halten. Deswegen ist das Verhältnis zwischen oszillierenden und nicht oszillierenden Tröpfchen nicht aussagekräftig.

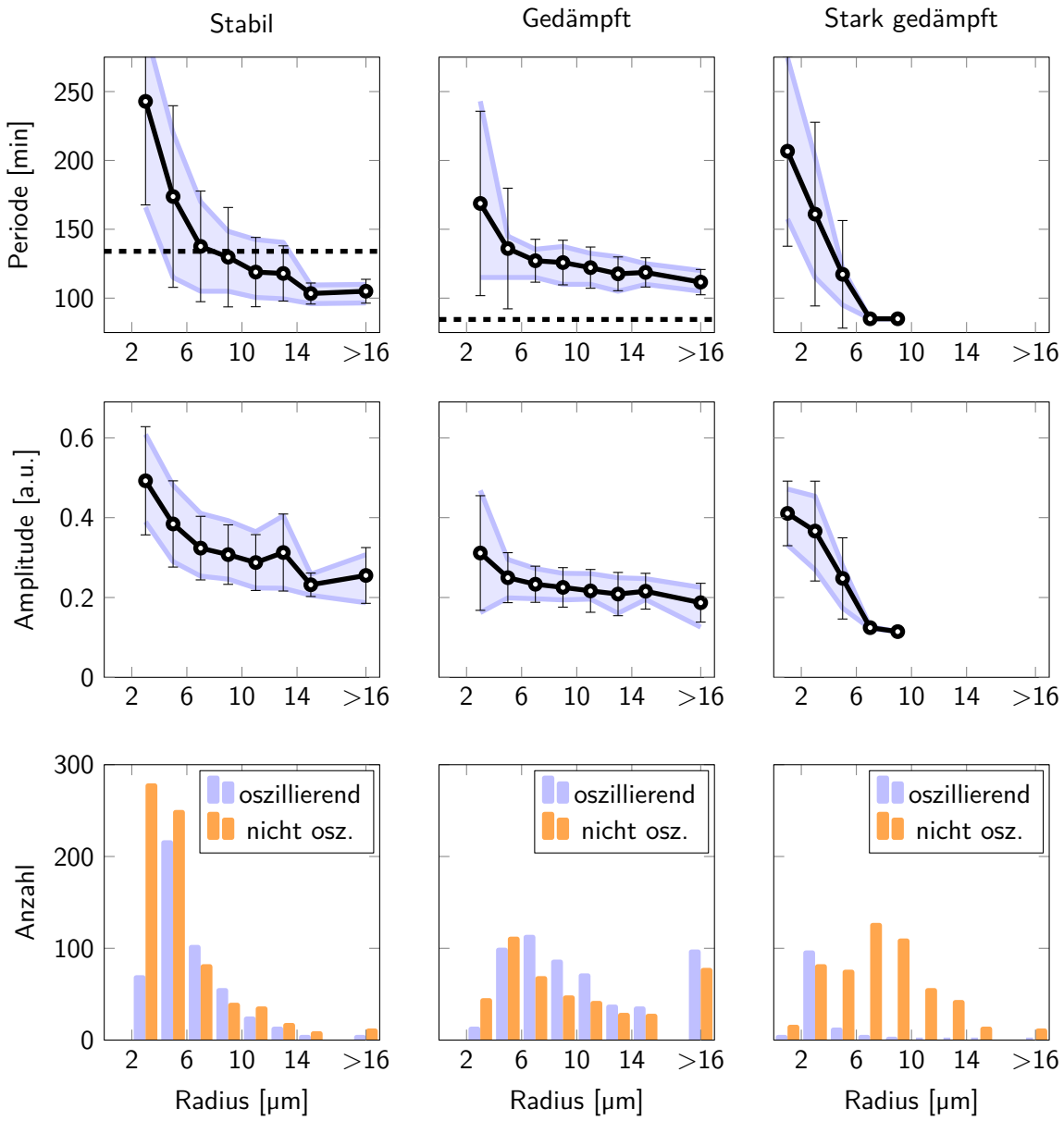
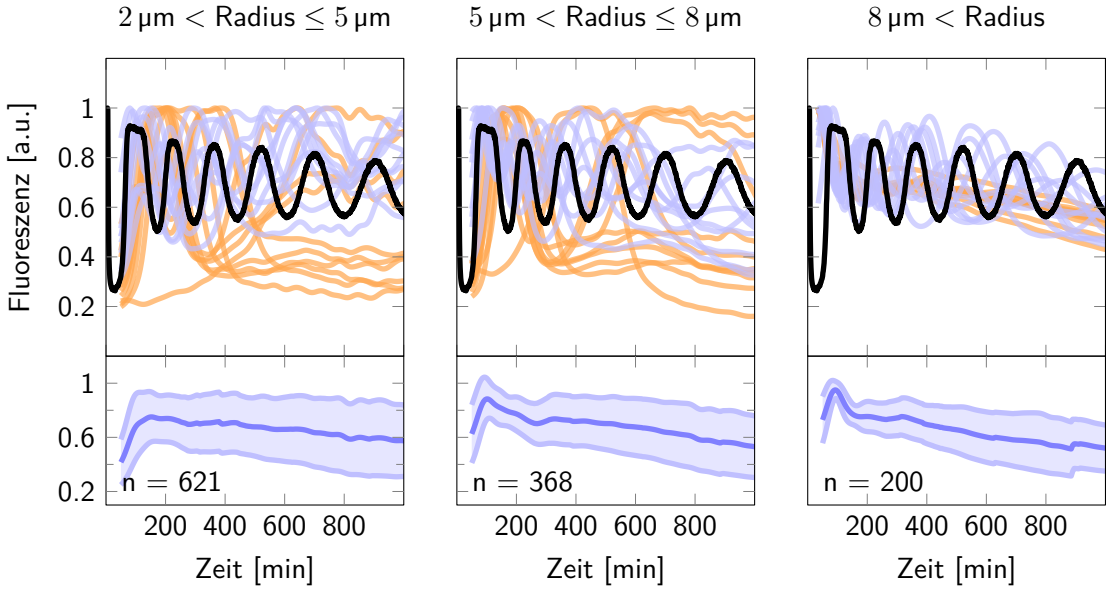


Abbildung 6.1.: Stochastische Eigenschaften der Tröpfchenkurven für alle drei Stimmungen. Die Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die stochastischen Eigenschaften dieser Populationen sind dargestellt: die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil dar. Die gestrichelte Linie ist die Periode der Oszillatorkurve im makroskopischen Volumen. Es ist zu beachten, dass im stark gedämpften Fall die Daten für den ersten Bin (0 – 2 µm) sehr wenige Datenpunkte enthalten und dass diese Stimmung eigentlich nicht als oszillierend eingestuft wird.

(a) Alle Kurven



(b) Nur Oszillierende

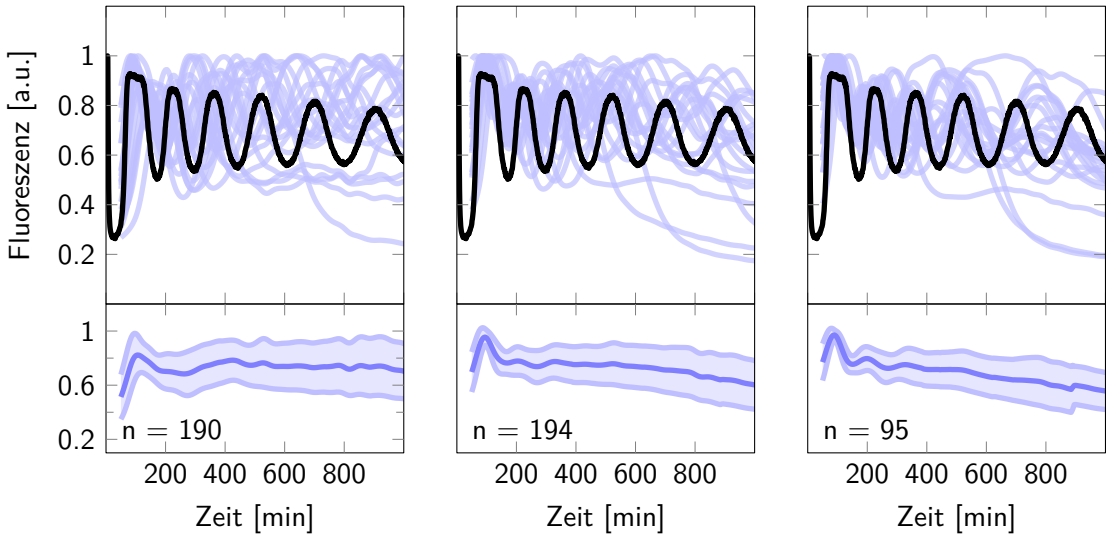


Abbildung 6.2.: Überblick über die Tröpfchenkurven bei stabiler Stimmung. Der obere Bereich stellt je 20 zufällige Beispielkurven (oszillierende sind blau und nicht oszillierende orange gezeichnet) aus dem Datensatz zusammen mit der Referenzkurve (schwarz) dar. Der untere Bereich ist die Mittelwertskurve aus allen Kurven und die umhüllende Fläche die Standardabweichungskurve. Die Tröpfchen wurden grob in Radiusbereiche eingeteilt und die Größe der Population ist n. Alle Kurven wurden auf das Maximum normiert.

6.1.2. Gedämpfte Oszillation

Wie der Name auch schon sagt, zeigt die gedämpfte Oszillatorstimmung eine gedämpfte Oszillation in der makroskopischen Referenzmessung. So oszilliert sie am Anfang sehr gut erkennbar – sogar schneller als die stabile Stimmung mit einer Periode von 84.5 min – wird aber mit der Zeit immer langsamer und weniger stark. Innerhalb der 1000 min klingt sie fast vollständig aus. Auch manche Tröpfchen zeigten solches Verhalten, aber viele Tröpfchen zeigten auch sehr stabile Oszillationen, die jedoch meist langsamer sind (s. Abb. 6.3). Interessant ist, dass hingegen die Mittelwertskurven der Tröpfchen eine stärkere Oszillation zeigen als bei der stabilen Stimmung. Die Tröpfchen haben also hier eine engere Periodenverteilung und laufen deswegen nicht so schnell außer Phase.

Diese Eigenschaft kann man auch in den gebinnten Populationen (s. Abb. 6.1, mittlere Spalte) erkennen. So sind die mittlere Periode und Amplitude nicht so stark vom Radius abhängig und die Breite der Populationen ist nicht ganz so groß. Klar erkennbar ist aber, dass die Tröpfchen viel langsamer oszillieren als die Referenz.

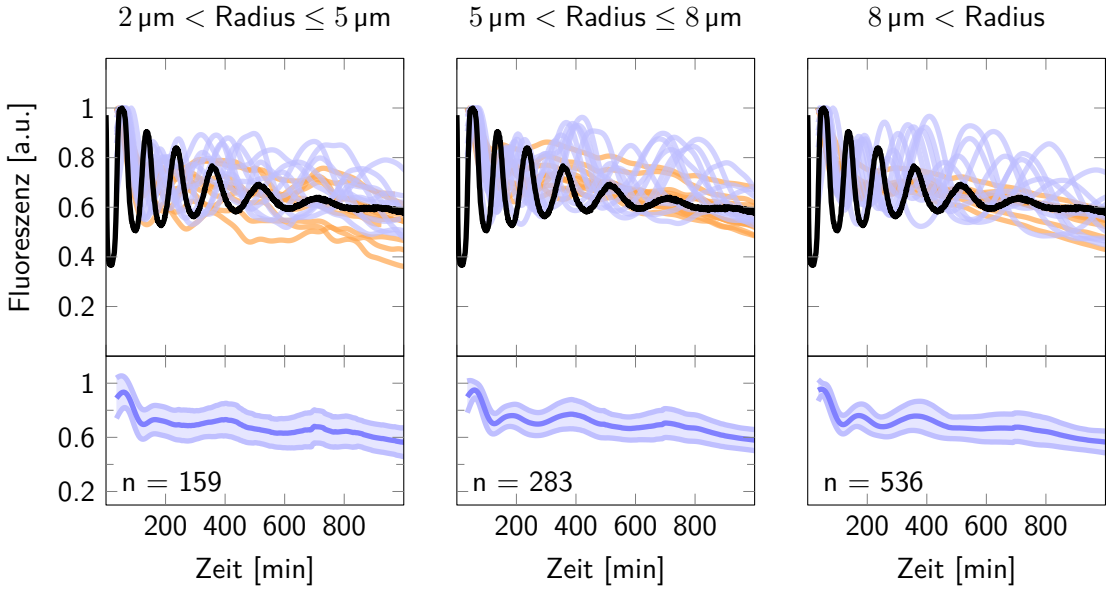
6.1.3. Stark gedämpfte Oszillation

Die Referenzmessung im makroskopischen Volumen der stark gedämpften Stimmung wird eigentlich als nicht oszillierend erkannt, da sie zum einen sehr schnell oszilliert – wenn man die Stärke der Filterung σ_{Filter} (s. 5.3.1.1) um den Faktor 10 reduziert, ergibt sich eine Periode von 56 min – und außerdem sehr schnell abklingt (s. Abb. 6.4). So ist nach 300 min eigentlich keine Oszillation mehr erkennbar. Bei den Tröpfchen sind aber erstaunlich viele als oszillierend eingestuft worden. Man hat also durch die Streuung der Molekülkonzentrationen bei der Partitionierung in Tröpfchen einige in den oszillierenden Bereich gebracht. Besonders auffällig ist, dass vor allem die kleineren Tröpfchen oszillieren, aber nur ein einziges Tröpfchen oszilliert, das größer ist als 8 µm. Dabei ist es dann auch nicht erstaunlich, dass die oszillierenden Tröpfchen sehr verschiedene Perioden haben und deswegen sehr schnell außer Phase kommen[2].

Auch die radiusabhängigen Populationen weisen sehr große Breiten auf – natürlich nur diejenigen, die auch eine signifikante Anzahl haben (s. Abb. 6.1, recht Spalte). Periode und Amplitude sind auch sehr stark vom Radius abhängig.

[2]Da es nur sechs oszillierende Tröpfchen mit einem Radius größer als 5 µm gibt, kann man ausschließlich anhand der Mittelwertskurve der kleineren Tröpfchen eine valide statistische Aussage treffen.

(a) Alle Kurven



(b) Nur Oszillierende

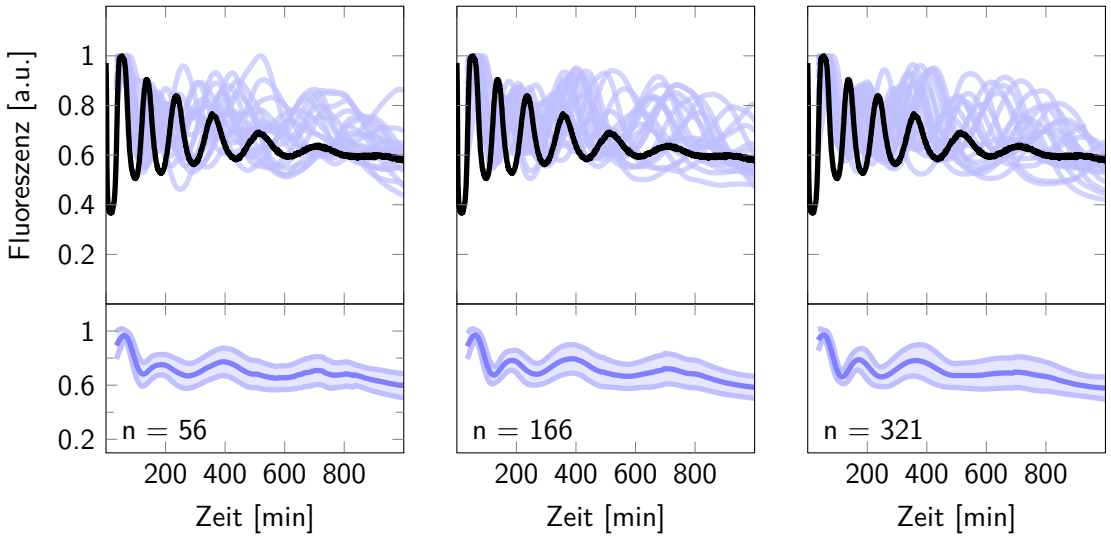
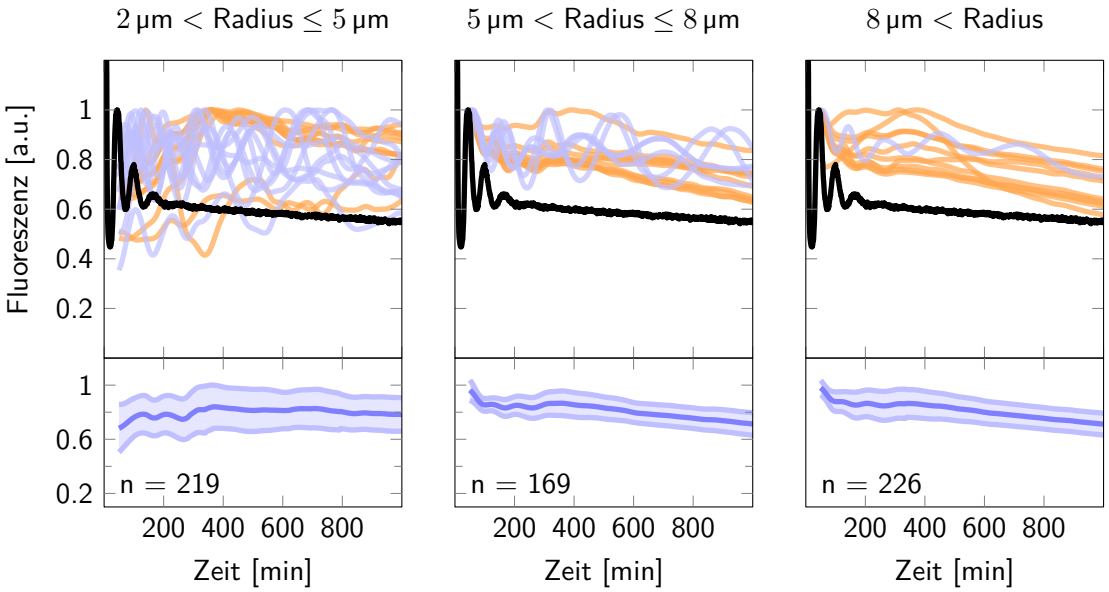


Abbildung 6.3.: Überblick über die Tröpfchenkurven bei gedämpfter Stimmung. Der obere Bereich stellt je 20 zufällige Beispielkurven (oszillierende sind blau und nicht oszillierende orange gezeichnet) aus dem Datensatz zusammen mit der Referenzkurve (schwarz) dar. Der untere Bereich ist die Mittelwertskurve aus allen Kurven und die umhüllende Fläche die Standardabweichungskurve. Die Tröpfchen wurden grob in Radiusbreiche eingeteilt und die Größe der Population ist n. Alle Kurven wurden auf das Maximum normiert.

(a) Alle Kurven



(b) Nur Oszillierende

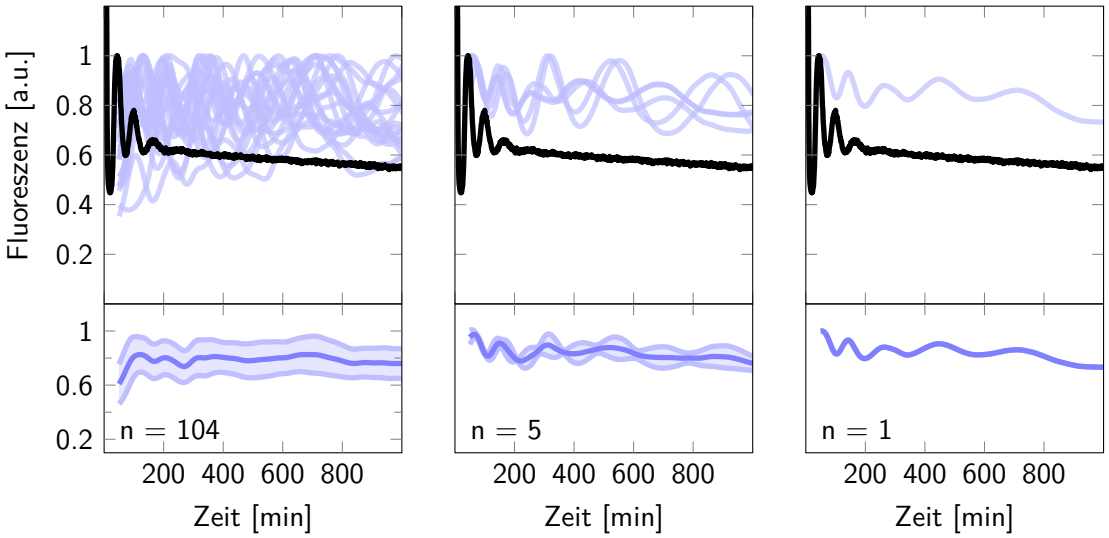


Abbildung 6.4.: Überblick über die Tröpfchenkurven bei stark gedämpfter Stimmung. Der obere Bereich stellt je 20 zufällige (oder alle, wenn weniger vorhanden sind) Beispielkurven (oszillierende sind blau und nicht oszillierende orange gezeichnet) aus dem Datensatz zusammen mit der Referenzkurve (schwarz) dar. Der untere Bereich ist die Mittelwertskurve aus allen Kurven und die umhüllende Fläche die Standardabweichungskurve. Die Tröpfchen wurden grob in Radiusbreiche eingeteilt und die Größe der Population ist n. Alle Kurven wurden auf das Maximum normiert.

6.2. Bakterien in Tröpfchen

6.2.1. AHL-Empfänger

Das Bakterienexperiment mit den AHL-Empfängerbakterien in einer Tröpfchensorte (insgesamt wurden 1405 Tröpfchen ausgewertet) und 200 nM AHL in einer anderen (91 Tröpfchen) zeigte eine recht starke Abhängigkeit der Fluoreszenz der Bakterien vom Abstand zum nächsten AHL-Tröpfchen (s. Abb. 6.5a). So haben die Tröpfchen mit dem größten Abstand am Ende des Experiments einen ähnlich geringen Fluoreszenzwert wie Bakterien, die komplett uninduziert sind. Die Tröpfchen in der direkten Umgebung zu einem AHL-Tröpfchen hingegen zeigen eine drei- bis vierfach höhere Intensität. Generell kann man bis zu einem Abstand von ungefähr 200 µm eine klar erkennbare Antwort sehen.

6.2.2. IPTG-Empfänger

Auch die äquivalenten Versuche mit IPTG (76 IPTG- und 1358 Bakterientröpfchen) zeigen eine klare Antwort der Bakterien (s. Abb. 6.5b). Diese ist aber nicht so stark vom Abstand abhängig und auch die am weitesten entfernten Tröpfchen weisn eine viel höhere Fluoreszenz als die Bakterien im Kontrollexperiment ohne IPTG auf.

6.2.3. Sender und Empfänger

Die Experimente mit Sender- und Empfängerbakterien lieferten für die Datenanalyse 13 Sender- und 1679 Empfängertröpfchen. Letzere zeigten auch eine Fluoreszenzantwort, die von der Form ähnlich zu den Experimenten mit AHL-Tröpfchen ist (s. Abb. 6.5c). Es gibt aber ein paar Unterschiede: so ist die die Abhängigkeit vom Abstand zum nächsten Sendertröpfchen viel weniger stark ausgeprägt und auch die am weitesten entfernten Tröpfchen zeigen eine ausgeprägte Fluoreszenz. Auch dauert es länger, bis sich die Antwort von der Referenzmessung unterscheidet – für die Tröpfchen mit kleinstem Abstand kann man sagen, dass die Antwort nach ca. 75 min anfängt, sich von der Referenzkurve zu unterscheiden, wohingegen bei dem Experiment mit den AHL-Tröpfchen schon nach ca. 25 min ein Unterschied erkennbar ist. Dafür dauert die Produktion des fluoreszierenden Proteins länger an – ca. 350 min im Vergleich zu 275 min.

Diese Unterschiede können sehr gut damit erklärt werden, dass das AHL in den Sendertröpfchen hierbei erst produziert werden muss, dann aber kontinuierlich die Gesamtmenge erhöht wird. So wird die Schwellwertkonzentration erst später erreicht, kann aber länger und bei größeren Abständen gehalten werden.

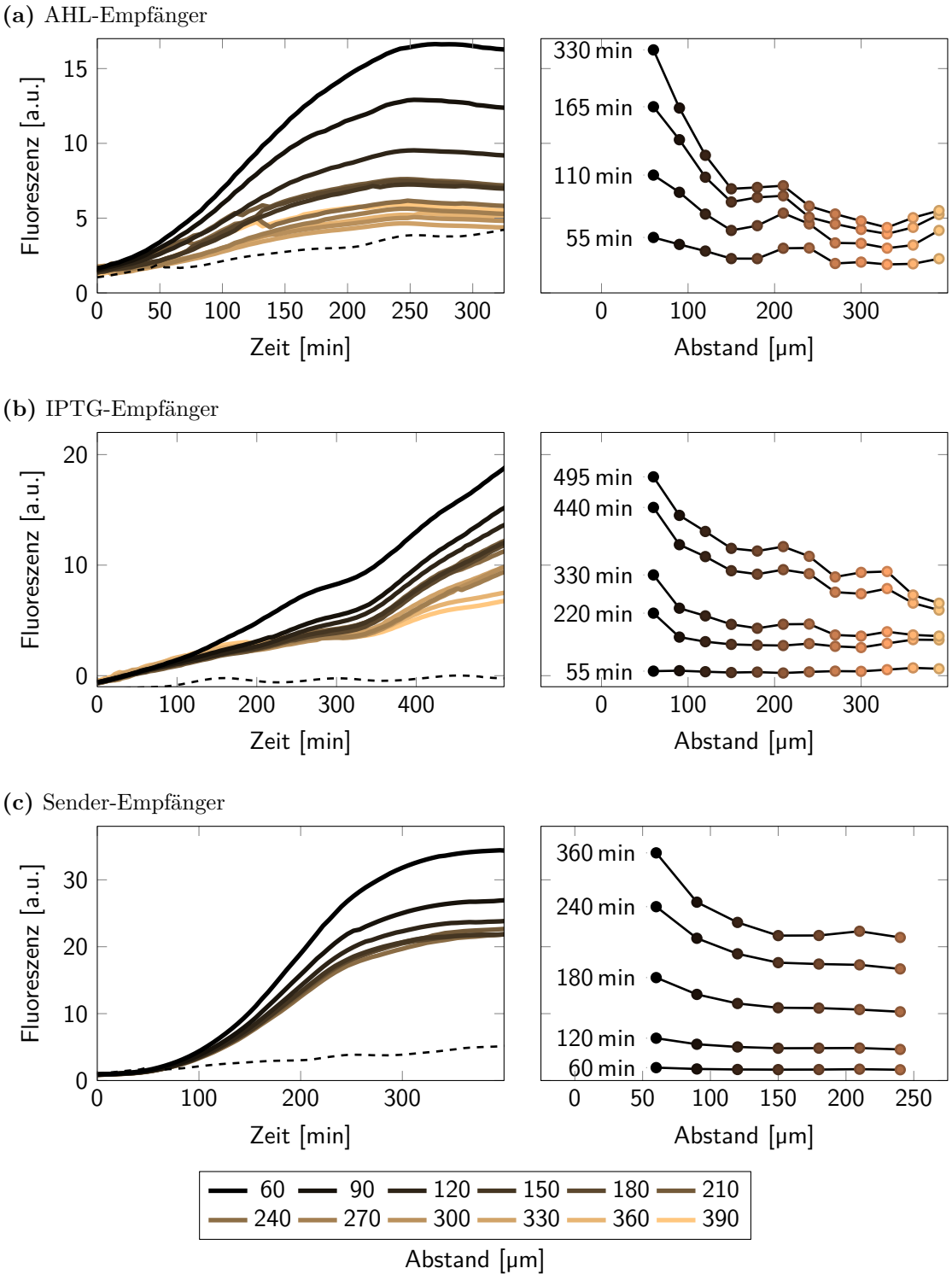


Abbildung 6.5.: Zeit- und abstandsabhängiger Verlauf der Empfängerfluoreszenz. Der Abstand ist die Entfernung zum nächsten Tröpfchen mit Senderbakterien. Die Entfernung zum übernächsten Tröpfchen wurde nicht beachtet. Die gestrichelte schwarze Linie repräsentiert das Fluoreszenzverhalten der Empfängerbakterien ohne AHL, IPTG und Senderbakterien.

(a) Mittelwerte der vier Quadranten. Die graue Linie ist der Zeitpunkt, zu dem (b) alle Tröpfchendaten zeigt. Im Inset ist der Quotient zwischen „nah-nah“ und „fern-fern“ gezeigt.

(b) Tröpfchendaten zum Zeitpunkt 600 min. Die vier farbigen Quadranten sind die Bereiche, in die die Tröpfchen für (a) eingeteilt wurden. Der Grauwert der Datenpunkte symbolisiert die Fluoreszenz des Tröpfchens.

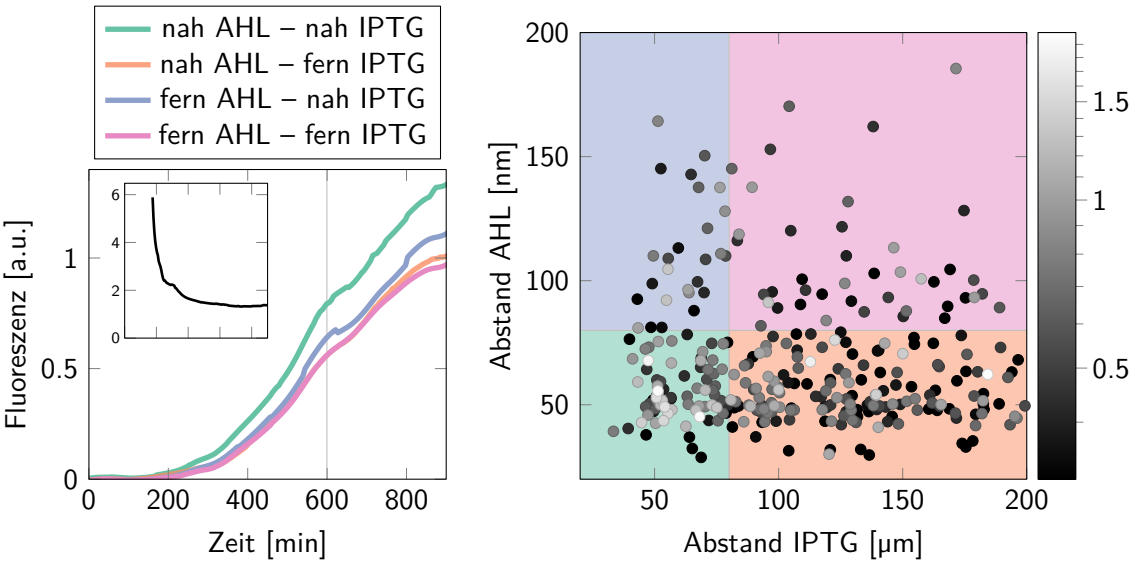


Abbildung 6.6.: Abstandsabhängige Darstellung der UND-Gatter Tröpfchendaten.

6.2.4. UND-Gatter

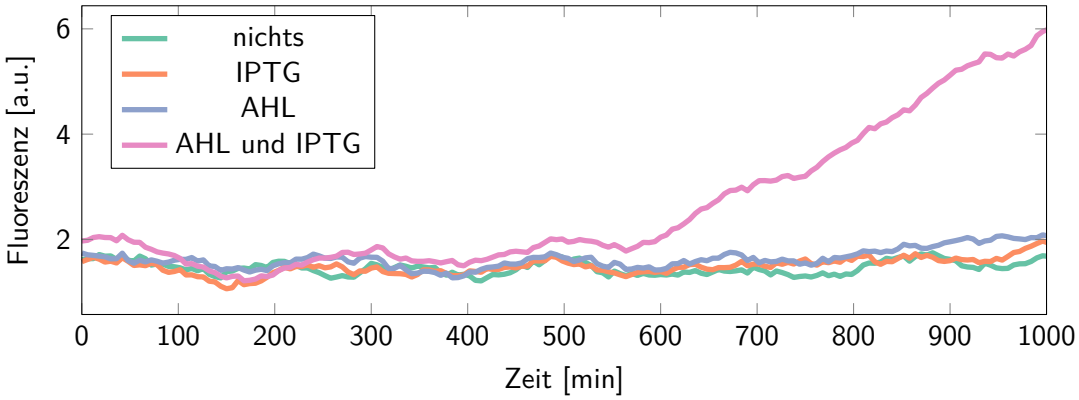
Bei den Experimenten mit drei Tröpfchensorten – AHL-, IPTG- und Empfänger-Tröpfchen – hat jedes Empfängertröpfchen nun zwei Abstände. Diese Abstände teilen jedes Tröpfchen in eine von vier Kategorien ein: nah und fern von AHL bzw. IPTG (s. Abb. 6.6b) – als Schwellwert zwischen „nah“ und „fern“ wurden 80 µm gewählt. Um Effekte von Tröpfchen außerhalb des Bildabschnitts zu reduzieren, wurden Tröpfchen, die weiter als 200 µm vom nächsten Sendertröpfchen entfernt waren, ignoriert. Diese vier Quadranten zeigen eine ähnliche Fluoreszenzantwort, aber die „nah AHL – nah IPTG“ Population beginnt früher und ihre Antwort ist auch etwas stärker (s. Abb. 6.6a).

Um noch klarer zu zeigen, dass die Bakterien nur dann fluoreszieren, wenn auch beide Senderstoffe vorhanden sind, wurden Experimente durchgeführt, bei denen die Bakterientröpfchen mit keiner, einer oder beiden Signalstofftröpfchensorten[3] gemischt wurden – also eines mit nur Empfänger-, eines mit AHL- und Empfänger-, eines mit IPTG- und Empfänger und zu guter Letzt eines mit allen Tröpfchensorten. Dabei sieht man, dass die Empfänger nur eine signifikante Antwort erzeugen, wenn auch beide Signalstofftröpfchensorten vorhanden sind (s. Abb. 6.7).

Somit konnte das UND-Gatter auch in der Tröpfchenumgebung gesteuert und sogar eine – wenn auch schwache – ortsabhängige Antwort beobachtet werden.

[3]Die Signalstoffkonzentrationen wurden hierbei um den Faktor 10 (AHL) bzw. 50 (IPTG) reduziert.

(a) Zeitlicher Verlauf der Fluoreszenzmittelwerte der Tröpfchen mit Bakterien. Die Daten wurden mit einem 35 min-Mittelwertsfilter geglättet.



(b) Bilder des Endzustandes der Tröpfchen. AHL-gefüllte Tröpfchen sind rot, IPTG blau und die Bakterien sind grün, wenn sie Fluoreszenz zeigen. Die Maßstabsskala ist 50 µm lang.

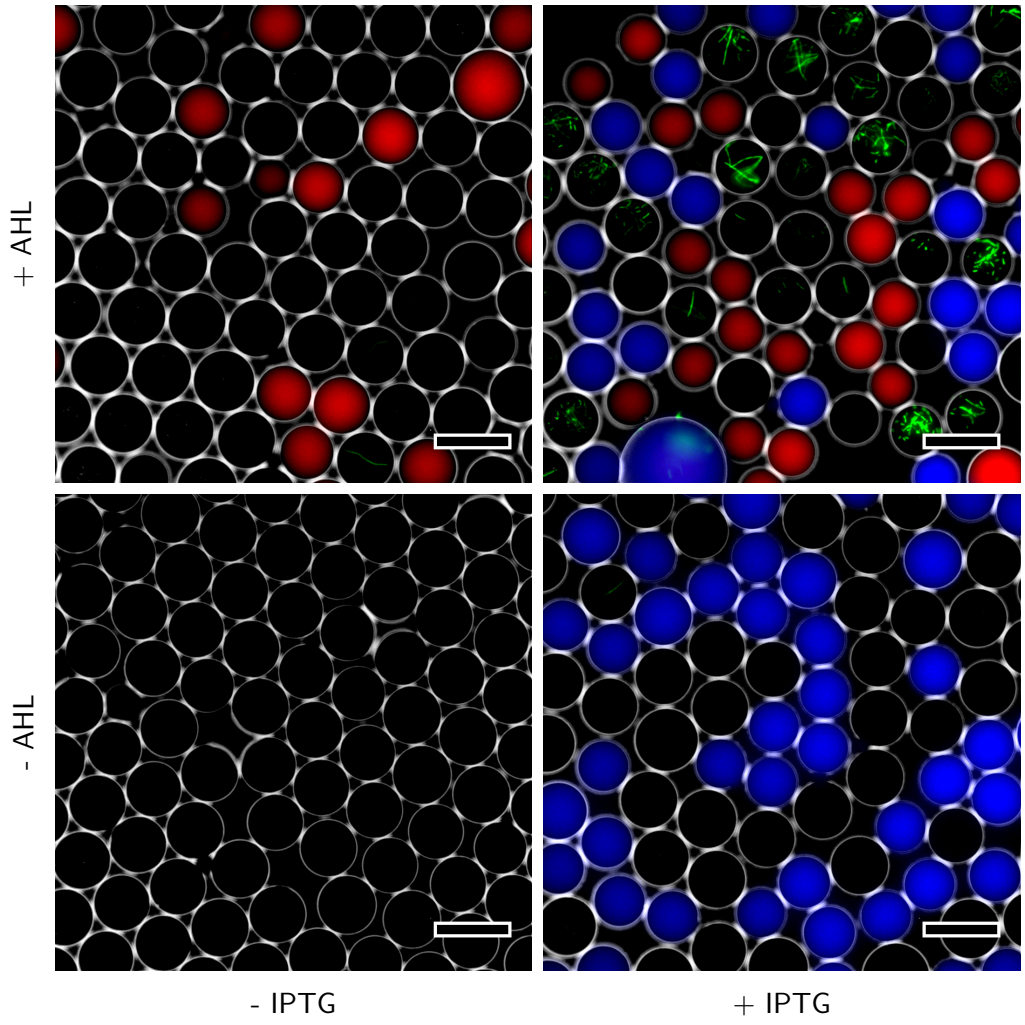


Abbildung 6.7.: Wahrheitstabelle des UND-Gatters.

6.3. Stochastik in Tröpfchen

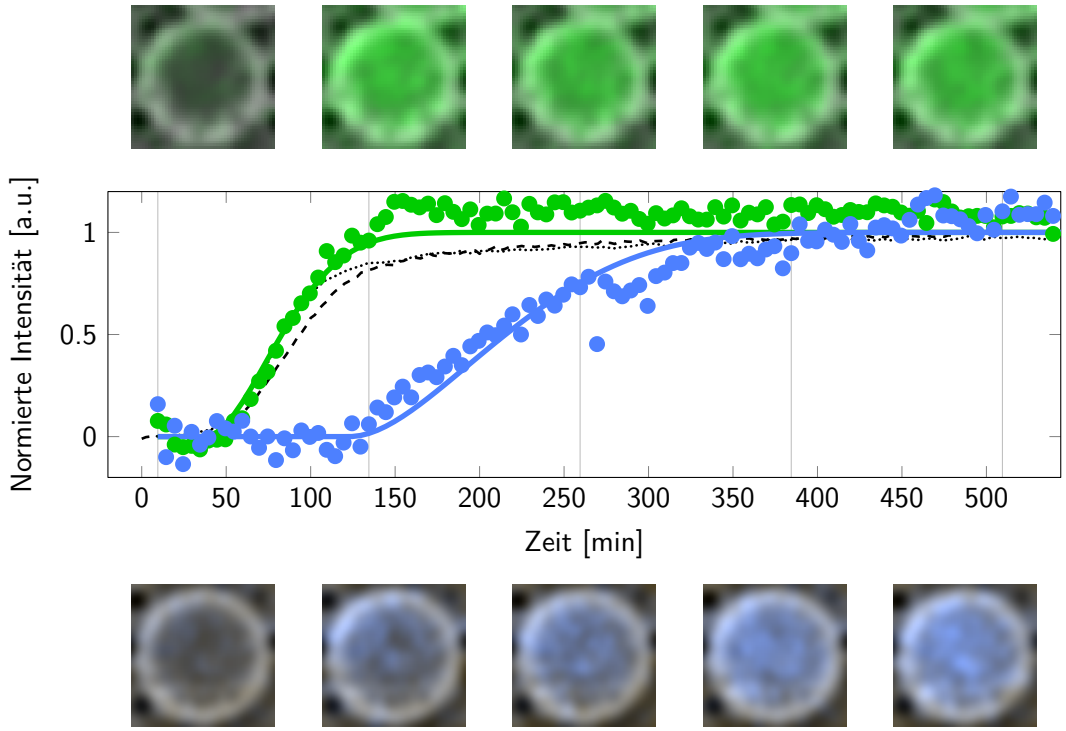


Abbildung 6.8.: Beispielkurven für Variabilität zwischen Tröpfchen. Zwei etwa gleich große (Radius 15.8 µm bzw. 15.4 µm) Tröpfchen (blau und grün – die Realdaten sind durch Punkte und die Modellfits durch Linien dargestellt) aus dem 1 nM-Experiment mit sehr unterschiedlichen Produktionsraten sind gezeigt. Die makroskopischen Experimente (schwarze Linien – Spektrometerdaten sind gestrichelt und Mikroskopdaten gepunktet) zeigen Verhalten, das zwischen den Extremen der Tröpfchen liegt. Die grauen Linien entsprechen den Zeitpunkten der Falschfarbenbilder der Tröpfchen über und unter dem Graphen.

Die Analysen der Daten der Stochastikexperimente zeigen auch, dass sich die Fluoreszenzkurven zwischen verschiedenen Tröpfchen stark unterscheiden können. So sind in Abb. 6.8 beispielhaft zwei ausgewählte Tröpfchen aus dem 1 nM-Experiment, die zwar einen ähnlichen zeitlichen normierten Fluoreszenzverlauf aufweisen aber stark unterschiedliche Produktionsraten besitzen, zusammen mit den makroskopischen Referenzkurven dargestellt.

In Abb. 6.9 sind die Mittelwerte der Produktionszeiten[4] und Produktionsraten aller Tröpfchen und der Referenzmessungen gezeigt. Vergleicht man diese, sieht man, dass die Werte bei kleinen Templatkonzentrationen zusammenfallen, dass aber für die höheren Konzentrationen die Tröpfchen langsamer sind. Die beiden Referenztypen – Mikroskop-

[4]In der Produktionszeit wird die gleiche Menge an RNA produziert, wie von Inhibitor und Reporter zusammen vorhanden ist.

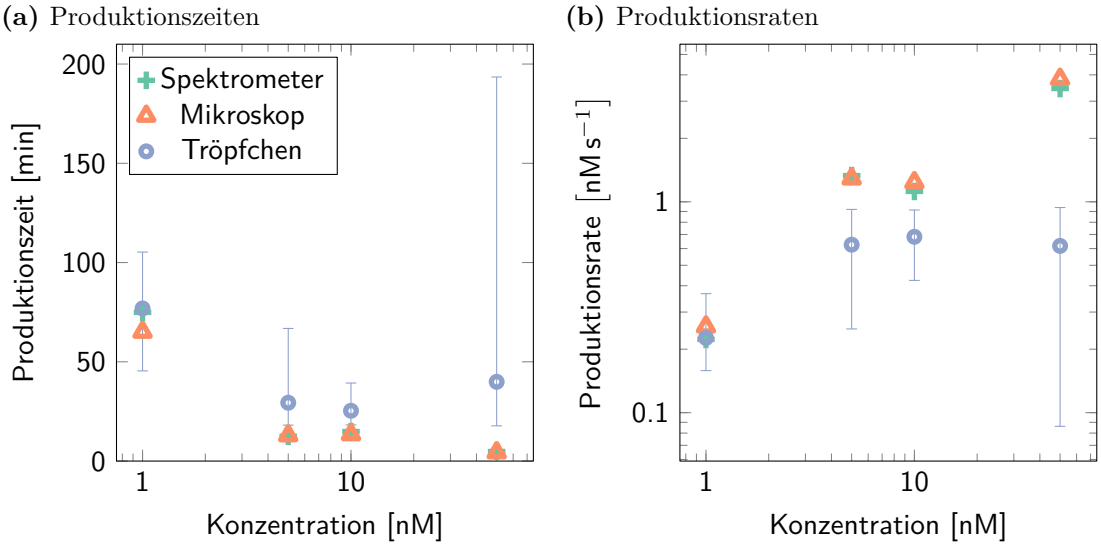


Abbildung 6.9.: Produktionszeiten und Produktionsraten für verschiedene Templatkonzentrationen. Die Datenpunkte für die Tröpfchen sind die Mittelwerte für alle Tröpfchen. Die Fehlerbalken stellen den Bereich zwischen dem 2.5 %- und 97.5 %-Quantil der Tröpfchenpopulation dar.

und Spektrometeraufnahmen – unterscheiden sich nicht stark, so dass angenommen werden kann, dass die grundlegenden Rahmenbedingungen – wie z. B. Temperatur – in Mikroskop und Spektrometer vergleichbar waren.

Der Mittelwert der Tröpfchen kann außerdem nach Tröpfchengröße spezifiziert werden. So sind in Abb. 6.10 sowohl jedes Tröpfchen als Datenpunkt als auch der Mittelwert von überlappenden Radiusbins in Abhängigkeit des Binzentrums repräsentiert. Auch dabei ist zu sehen, dass für 1 nM die Tröpfchendaten nahe bei den Referenzwerten liegen. Ferner ist bei dieser Konzentration keine Radiusabhängigkeit des Mittelwerts sichtbar. Bei höheren Konzentrationen ist zwar eine Radiusabhängigkeit erkennbar, sie ist aber nicht sehr stark ausgeprägt. Viel auffälliger hingegen ist der globale Trend zu kleineren Raten.

Da zwischen der Produktionsrate k_{prod} und der Konzentration des DNA-Protein-Komplexes eine lineare Beziehung besteht (s. Gl. (B.2.4) und 7.3.1), kann man eine Größe ξ, die proportional zur zugehörigen Teilchenanzahl ist, über $\xi = \frac{k_{\text{prod}}}{k_{\text{cat}}} \cdot r^3$ errechnen[5]. Der Skalierungsfaktor ist aber unbekannt und deswegen kann man von der Form der Verteilungen von ξ bei den verschiedenen Radien nicht darauf schließen, ob die Verteilungen der Teilchenanzahl einer Poissonverteilung folgten – also durch eine unabhängige Aufteilung entstehen – oder nicht. Wenn man aber die relativen Streuungen der Verteilungen –

[5]Die Katalyserate k_{cat} wird für jede Templatkonzentration separat mit dem Fit des Reaktionsmodells (s. 7.3.1) an die Referenzmessungen im makroskopischen Volumen bestimmt (s. Tab. 7.5).

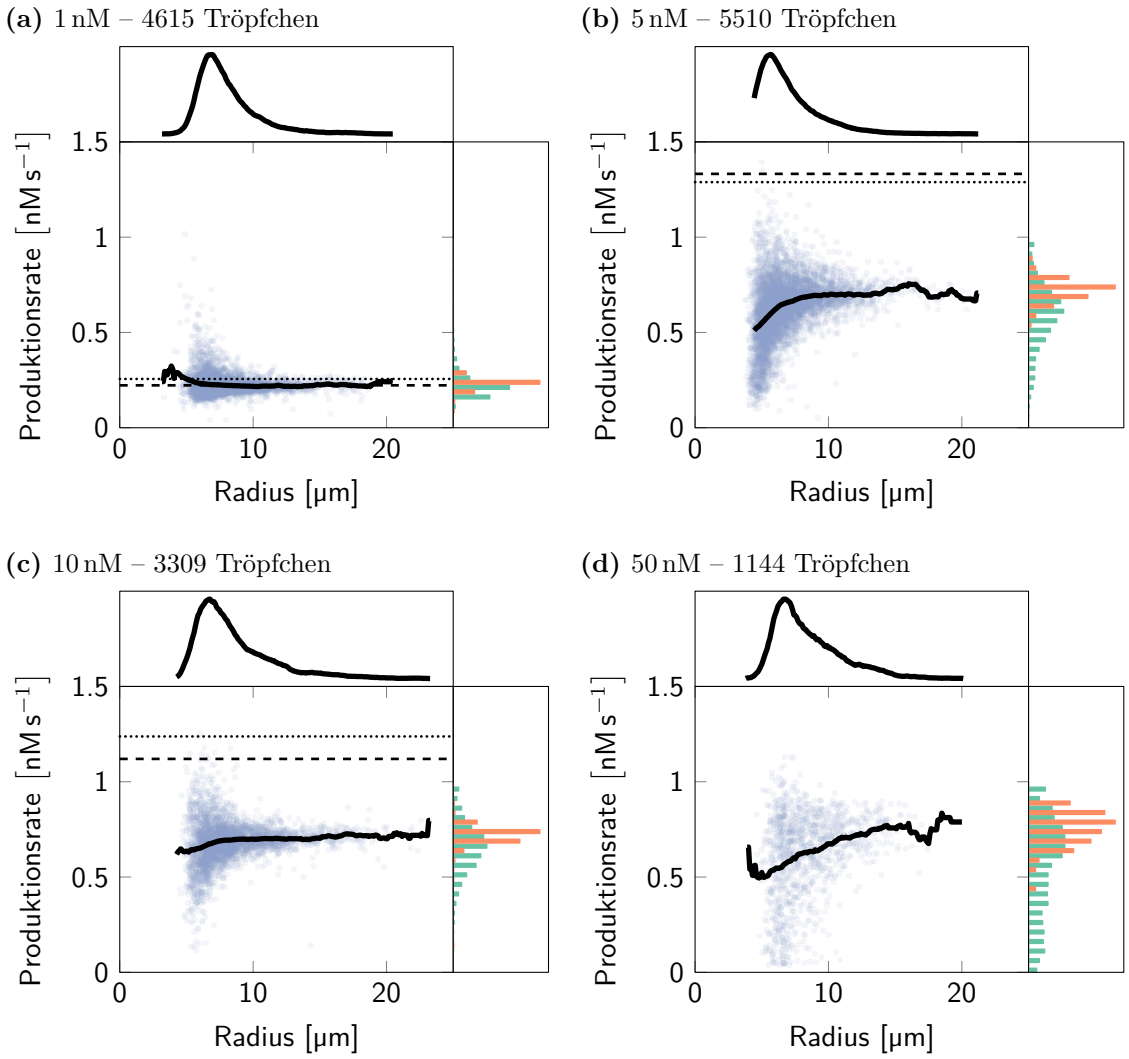


Abbildung 6.10.: Übersichten über die Produktionsraten der Tröpfchen. Die schwarzen dünnen Linien sind die Produktionsraten in den makroskopischen Volumen (Spektromenter gestrichelt und Mikroskop gepunktet). Jeder durchsichtige blaue Punkt ist der Datenpunkt eines Tröpfchens. Die dicke schwarze Linie ist der Mittelwert von 1.5 µm breiten wandernden Bins und die Linie über der Hauptdarstellung ist deren relative Größe. Die Histogramme auf der rechten Seite unterscheiden zwischen kleinen – Radius kleiner als 8 µm (grün) – und großen – Radius größer als 12 µm (orange) – Tröpfchen. Die Referenzkurvenraten im 50 nM-Experiment sind nicht gezeigt, da sie viel größer sind als die Tröpfchenraten.

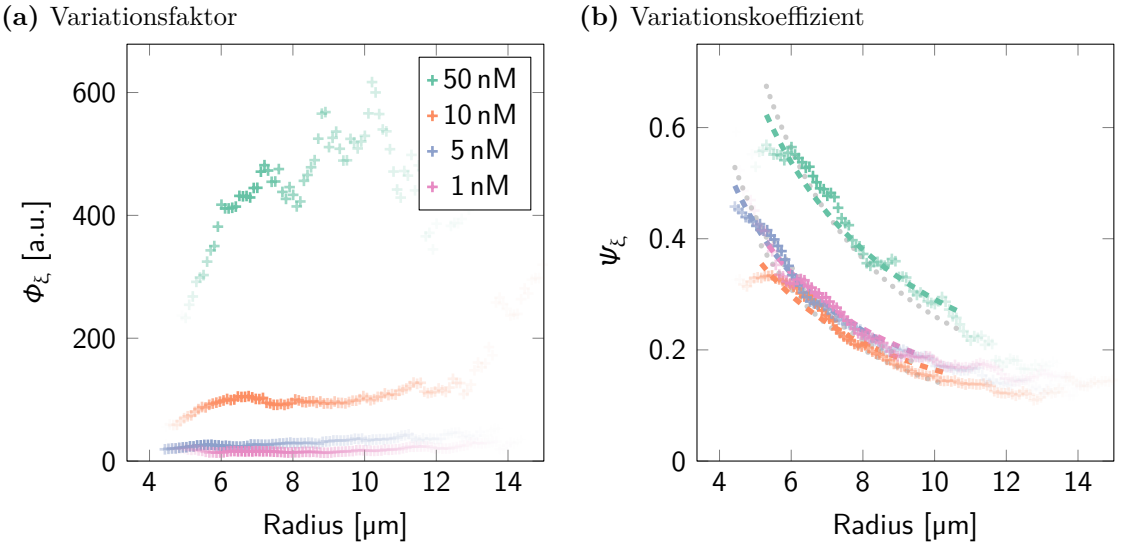


Abbildung 6.11.: Relative Streuungen der skalierten DNA-Protein-Komplexanzahl ξ. Jeder Datenpunkt repräsentiert die relative Streuung eines 1.5 µm breiten Radiusbins. Der Variationsfaktor Φ ist das Verhältnis zwischen Varianz und Mittelwert der Population, der Variationskoeffizient Ψ das Verhältnis zwischen Standardabweichung und Mittelwert. Je mehr Tröpfchen in dem Radiusbin liegen, desto kräftiger ist das gezeichnete Kreuz. Es wird dadurch also die statistische Signifikanz des Datenpunktes repräsentiert. Die gestrichelten farbigen Linien sind Fits an die gewichteten Datenpunkte des Variationskoeffizient mit $a \cdot r^b$. Die grau gepunkteten Linien sind Fits an eine Poissonverteilung: $a \cdot r^{-1.5}$.

den Variationsfaktor $\Phi = \frac{\text{Varianz}}{\text{Mittelwert}}$ und den Variationskoeffizienten $\Psi = \frac{\text{Standardabweichung}}{\text{Mittelwert}}$[6] – in Abhängigkeit des Radius betrachtet (s. Abb. 6.11), kann man die Ähnlichkeit zu Poisson beschreiben. So müsste der Variationsfaktor konstant sein und der Variationskoeffizient eine $r^{-1.5}$-Abhängigkeit zeigen (s. A.1)[7]. Der Variationskoeffizient zeigt bei allen Experimenten diese Radiusabhängigkeit, aber nur das 1 nM-Experiment zeigt einen konstanten Variationsfaktor. Zusätzlich ist zu bemerken, dass sowohl die Radiusabhängigkeit als auch die absoluten Werte des Variationsfaktors mit steigender Templatkonzentration zunehmen.

Allgemein ist zu beachten, dass bei diesen Experimenten jeweils viel mehr Tröpfchen ausgewertet werden konnten als bei den Oszillatorexperimenten. Das liegt zum einen daran, dass in den Mikroskopvideos weniger Drift zu beobachten war und deswegen pro Experiment mehr Videos, die bei unterschiedlichen Positionen aufgenommen wurden, ausgewertet werden konnten. Zum anderen reduziert die Unterscheidung zwischen oszillierenden und nicht oszillierenden Tröpfchen die Anzahl an Tröpfchen zusätzlich.

[6] Im Englischen heißt der Variationsfaktor „coefficient of variation“ (CV) und der Variationskoeffizient „coefficient of dispersion“ oder „Fano factor“.

[7] $\Psi \underbrace{=}_{\text{Gl. (A.1.7)}} \frac{1}{\sqrt{\langle n \rangle}} \underbrace{\propto}_{\langle n \rangle \propto V \propto r^3} r^{-\frac{3}{2}}$

7. Modelle

Eine weitere Abstraktion, Analyse und Interpretation dieser Ergebnisse – und manchmal sogar die Generierung der Ergebnisse selbst – ist ohne Modelle nicht möglich. Die Modelle wurden aus der abstrakten Repräsentation des Experimentlayouts, d. h. designter und bekannter Reaktionen und Geometrien, entwickelt. Wenn das so generierte Modell die experimentellen Daten nicht gut genug nachbilden konnte, wurden weitere Modelle, die zusätzliche physikalische Annahmen aufgreifen, entwickelt. Ein Vergleich ermöglichte dann die Selektion des am besten passenden Modells.

Da alle Modelle Parameter enthalten, die in einem bestimmten Rahmen frei wählbar sind, wurden die Modellkurven an die jeweiligen experimentellen Kurven angefittet.

7.1. Oszillator in Tröpfchen

Der Oszillator wird in zwei Stufen modelliert. Zuerst werden die Reaktionen, die die Oszillationen erzeugen, und dann die Verteilung der Spezies auf die Tröpfchen modelliert.

7.1.1. Reaktionsmodell

Die chemischen Reaktionen, die die einzelnen Bestandteile (Bezeichnungen s. 4.1.1 und Reaktionsschema s. Abb. 4.2) ausführen können, bestimmen das Reaktionsmodell. Dabei gibt es zum einen die Reaktionen zwischen den DNA- und RNA-Strängen – Hybridisierung und Verdrängung (s. 3.2.3) – und zum anderen die Reaktionen, an denen die Enzyme beteiligt sind.

Stranghybridisierungs- und Strangverdrängungsreaktionen

$$\begin{aligned}
\mathrm{T_{12}} + \mathrm{A_2} &\xrightarrow{k_{\mathrm{TA,12}}} \mathrm{T_{12} \cdot A_2} \\
\mathrm{T_{21}} + \mathrm{A_1} &\xrightarrow{k_{\mathrm{TA,21}}} \mathrm{T_{21} \cdot A_1} \\
\mathrm{A_1} + \mathrm{dI_1} &\xrightarrow{k_{\mathrm{AI,1}}} \mathrm{A_1 \cdot dI_1} \\
\mathrm{rA_1} + \mathrm{dI_1} &\xrightarrow{k_{\mathrm{rAI,1}}} \mathrm{rA_1 \cdot dI_1} \\
\mathrm{A_2} + \mathrm{rI_2} &\xrightarrow{k_{\mathrm{AI,2}}} \mathrm{A_2 \cdot rI_2}
\end{aligned}$$

$$\begin{aligned}
\mathrm{T_{12} \cdot A_2} + \mathrm{rI_2} &\xrightarrow{k_{\mathrm{TAI,12}}} \mathrm{T_{12} + A_2 \cdot rI_2} \\
\mathrm{T_{21} \cdot A_1} + \mathrm{dI_1} &\xrightarrow{k_{\mathrm{TAI,21}}} \mathrm{T_{21} + A_1 \cdot dI_1} \\
\mathrm{rA_1 + A_1 \cdot dI_1} &\xrightarrow{k_{\mathrm{AIrA,1}}} \mathrm{rA_1 \cdot dI_1 + A_1}
\end{aligned}$$

mit nicht vollständigen Abbauprodukten

$$\begin{aligned}
\mathrm{A_2} + \mathrm{sI_2} &\underset{k_{\mathrm{s},-}}{\overset{k_{\mathrm{s},+}}{\rightleftharpoons}} \mathrm{A_2 \cdot sI_2} \\
\mathrm{T_{12}} + \mathrm{A_2 \cdot sI_2} &\xrightarrow{k_{\mathrm{TA},12}} \mathrm{T_{12} \cdot A_2 \cdot sI_2} \\
\mathrm{T_{12} \cdot A_2} + \mathrm{sI_2} &\underset{k_{\mathrm{s},-}}{\overset{k_{\mathrm{s},+}}{\rightleftharpoons}} \mathrm{T_{12} \cdot A_2 \cdot sI_2} \\
\mathrm{rI_2} + \mathrm{A_2 \cdot sI_2} &\xrightarrow{k_{\mathrm{AI},2}} \mathrm{A_2 \cdot rI_2} + \mathrm{sI_2}
\end{aligned}$$

Enzymreaktionen

$$\begin{aligned}
\mathrm{RNAP} + \mathrm{T_{12} \cdot A_2} &\underset{k_{-,\mathrm{an},12}}{\overset{k_+}{\rightleftharpoons}} \mathrm{RNAP \cdot T_{12} \cdot A_2} \xrightarrow{k_{\mathrm{c,an},12}} \mathrm{RNAP} + \mathrm{T_{12} \cdot A_2} + \mathrm{rA_1} \\
\mathrm{RNAP} + \mathrm{T_{12}} &\underset{k_{-,\mathrm{aus},12}}{\overset{k_+}{\rightleftharpoons}} \mathrm{RNAP \cdot T_{12}} \xrightarrow{k_{\mathrm{c,aus},12}} \mathrm{RNAP} + \mathrm{T_{12}} + \mathrm{rA_1} \\
\mathrm{RNAP} + \mathrm{T_{21} \cdot A_1} &\underset{k_{-,\mathrm{an},21}}{\overset{k_+}{\rightleftharpoons}} \mathrm{RNAP \cdot T_{21} \cdot A_1} \xrightarrow{k_{\mathrm{c,an},21}} \mathrm{RNAP} + \mathrm{T_{21} \cdot A_1} + \mathrm{rI_2} \\
\mathrm{RNAP} + \mathrm{T_{21}} &\underset{k_{-,\mathrm{aus},21}}{\overset{k_+}{\rightleftharpoons}} \mathrm{RNAP \cdot T_{21}} \xrightarrow{k_{\mathrm{c,aus},21}} \mathrm{RNAP} + \mathrm{T_{21}} + \mathrm{rI_2} \\
\mathrm{RNaseH} + \mathrm{rA_1 \cdot dI_1} &\underset{k_{-,\mathrm{H},1}}{\overset{k_{+,\mathrm{H}}}{\rightleftharpoons}} \mathrm{RNaseH \cdot rA_1 \cdot dI_1} \xrightarrow{k_{\mathrm{c,H},1}} \mathrm{RNaseH} + \mathrm{dI_1}
\end{aligned}$$

mit nicht vollständigen Abbauprodukten

$$\begin{aligned}
\mathrm{RNaseH} + \mathrm{A_2 \cdot rI_2} &\underset{k_{-,\mathrm{H},2}}{\overset{k_{+,\mathrm{H}}}{\rightleftharpoons}} \mathrm{RNaseH \cdot A_2 \cdot rI_2} \\
&\xrightarrow{k_{\mathrm{c,H},2}} \mathrm{RNaseH} + \mathrm{A_2 \cdot sI_2} \\
\mathrm{RNAP} + \mathrm{T_{12} \cdot A_2 \cdot sI_2} &\underset{k_{-,\mathrm{an},12}}{\overset{k_+}{\rightleftharpoons}} \mathrm{RNAP \cdot T_{12} \cdot A_2 \cdot sI_2} \\
&\xrightarrow{k_{\mathrm{c,an},12}} \mathrm{RNAP} + \mathrm{T_{12} \cdot A_2 \cdot sI_2} + \mathrm{rA_1}
\end{aligned}$$

7.1.2. Differentialgleichungen

Diese chemischen Gleichungen können in Differentialgleichungen umgewandelt werden, mit denen das Modell simuliert werden kann (s. B.1). Von den 24 Differentialgleichungen der 24 verschiedenen Spezies können sieben durch die Erhaltung der Enzym- und DNA-Konzentrationen ersetzt werden (s. B.1.1). Somit ergibt sich ein Modell des Oszillators mit 22 Parametern – 15 anzupassende Raten und 7 Anfangskonzentrationen.

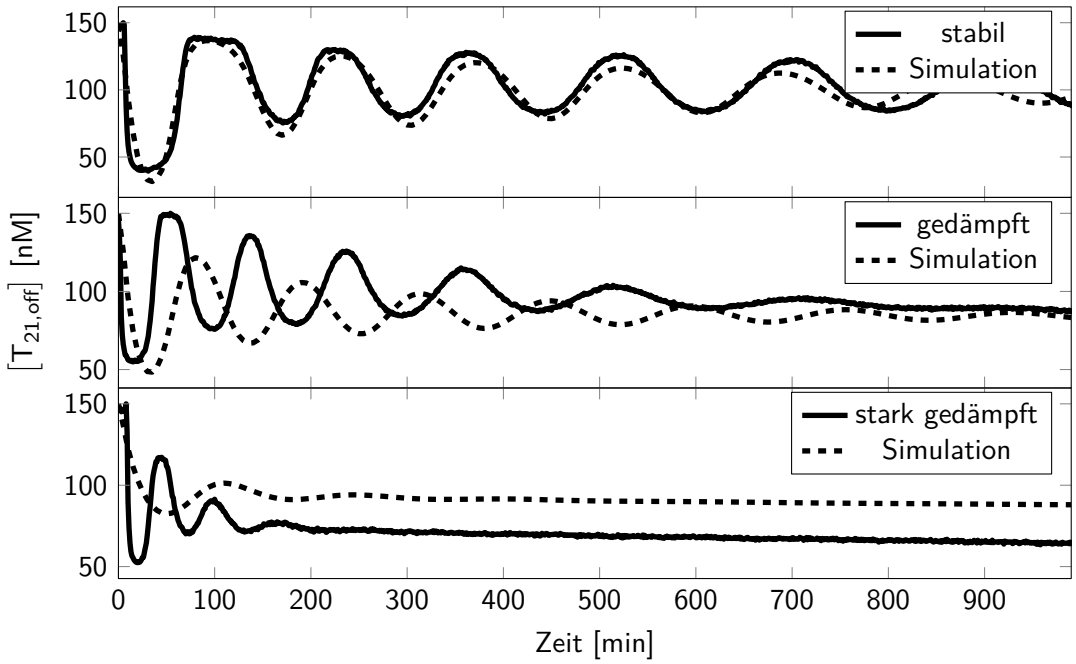


Abbildung 7.1.: Vergleich zwischen Simulation und Experiment. Die durchgezogenen Linien sind die Experimente im makroskopischen Volumen und die gestrichelten die angefitteten Simulationen. Man sieht, dass bei der stabilen Stimmung die Simulation am besten zum Experiment passt und bei der stark gedämpften am schlechtesten.

7.1.3. Datenfit

Dieser Satz von Differentialgleichungen wurde verwendet, um die freien Parameter an die Experimente anzufitten. Dazu wurde das Modell simuliert und mit dem Experiment verglichen. Da oszillierende Kurven schwierig zu fitten sind, wurde eine spezielle Fehlerfunktion verwendet, die beim Fitten minimiert wurde:

$$F = \frac{1}{N}\sum_{t=0}^{N-1}\left(x(t)_{\mathrm{Exp}} - x(t)_{\mathrm{Sim}}\right)^2 + (A_{\mathrm{Exp}} - A_{\mathrm{Sim}})^2 + (f_{\mathrm{Exp}} - f_{\mathrm{Sim}})^2 + (\lambda_{\mathrm{Exp}} - \lambda_{\mathrm{Sim}})^2 \tag{7.1.1}$$

Dabei sind $x(t)$ die Werte zum Zeitpunkt t, A ist die maximale Differenz zwischen zwei benachbarten Extrema, f die Frequenz und λ die Dämpfung der Kurve. Dadurch wurden auch die grundlegenden Eigenschaften der Oszillation angefittet. Bei den Ergebnissen (s. Abb. 7.1) sieht man z. B., dass die Kurve für die gedämpfte Stimmung nicht perfekt auf die Daten passt – sie scheint etwas zu langsam zu oszillieren –, aber das Verhalten wird doch recht gut beschrieben. Die stabile Stimmung konnte sehr gut an die Experimente angefittet werden und die stark gedämpfte am schlechtesten – so wird die experimentelle Kurve als nicht oszillierend angesehen, da sie zu schnell oszilliert, aber die simulierte wird gerade noch als oszillierend klassifiziert.

Tabelle 7.1.: Fitparameter des Oszillatormodells

Parameter	$i=2,\ j=1$	$i=1,\ j=2$	Fitbereich	Andere Studien
$k_{+}\,[\mathrm{M^{-1}\,s^{-1}}]$	1.90×10^{5}	–	$10^{5}-10^{7}$	–
$k_{-,\mathrm{an,ij}}\,[\mathrm{s^{-1}}]$	0.0446	0.0634	0.01 – 0.3	–
$k_{\mathrm{c,an,ij}}\,[\mathrm{s^{-1}}]$	0.0186	0.0471	0.01 – 0.1	0.73 – 1.12
$k_{-,\mathrm{aus,ij}}\,[\mathrm{s^{-1}}]$	0.35	0.388	0.01 – 1	–
$k_{\mathrm{c,aus,ij}}\,[\mathrm{s^{-1}}]$	0.0033	0.0131	0.001 – 0.03	0.11 – 0.18
$k_{+,\mathrm{H}}\,[\mathrm{M^{-1}\,s^{-1}}]$	7.13×10^{5}	–	$10^{5}-10^{7}$	–
$k_{-,\mathrm{H,j}}\,[\mathrm{s^{-1}}]$	0.125	0.0692	0.01 – 1	–
$k_{\mathrm{c,H,j}}\,[\mathrm{s^{-1}}]$	0.554	0.463	0.01 – 1	0.02 – 0.6
$k_{\mathrm{TA,ij}}\,[\mathrm{M^{-1}\,s^{-1}}]$	1.59×10^{5}	9.47×10^{3}	$3\times10^{3}-3\times10^{5}$	$0-3\times10^{6}$
$k_{\mathrm{AI,j}}\,[\mathrm{M^{-1}\,s^{-1}}]$	4.69×10^{3}	5.66×10^{4}	$3\times10^{3}-3\times10^{5}$	$0-3\times10^{6}$
$k_{\mathrm{TAI,ij}}\,[\mathrm{M^{-1}\,s^{-1}}]$	1.92×10^{4}	5.05×10^{3}	$3\times10^{3}-3\times10^{5}$	$0-3\times10^{6}$
$k_{\mathrm{rAI,j}}\,[\mathrm{M^{-1}\,s^{-1}}]$	1.50×10^{4}	–	$3\times10^{3}-3\times10^{5}$	$0-3\times10^{6}$
$k_{\mathrm{AIrA,j}}\,[\mathrm{M^{-1}\,s^{-1}}]$	1.65×10^{4}	–	$3\times10^{3}-3\times10^{5}$	$0-3\times10^{6}$
$k_{\mathrm{s,+}}\,[\mathrm{M^{-1}\,s^{-1}}]$	1.65×10^{4}	–	$3\times10^{3}-3\times10^{5}$	–
$k_{\mathrm{s,-}}\,[\mathrm{s^{-1}}]$	0.0525	–	0.01 – 1	–

7.1.4. Proteinscan

Mit diesem Modell wurden dann die Proteinkonzentrationen durchgescannt (s. Abb. 7.2), d. h. das Modell wurde mit verschiedenen Anfangskonzentrationen für die zwei Proteine simuliert und ausgewertet. Dabei sieht man gut, dass die stabile Stimmung in einem Bereich niedriger Dämpfung, aber in einem Bereich mit veränderlicher Periode liegt. Die gedämpfte Stimmung liegt hingegen in einem Bereich, in dem sich die Periode nicht so stark ändert. Die stark gedämpfte Stimmung liegt am Rand des oszillierenden Bereichs, wodurch kleine Änderungen einen großen Einfluss haben.

7.1.5. Verteilungsmodell

Um die Verteilung der Spezies auf die Tröpfchen zu beschreiben, werden verschiedene Ansätze verfolgt, die am Ende mit den experimentell beobachteten Ergebnissen verglichen werden. So wird zuerst eine unabhängige Verteilung – also eine Poissonverteilung (s. A.1) – der Moleküle untersucht, dann zwei Gammaverteilungen (s. A.3) mit verschiedenen Breiten, $\beta = 10$ und $\beta = 100$. Als Letztes werden diese beiden Gammaverteilungen noch mit einem Verlustterm versehen.

Dabei wird jede beteiligte Molekülart separat aufgeteilt. Dazu wird zuerst die durchschnittliche Molekülanzahl in einem Tröpfchen mit festgelegtem Radius aus der Anfangskonzentration berechnet, die dann als Mittelwert der Verteilungsfunktion eingesetzt wird. Anschließend wird – entsprechend dieser Verteilungsfunktionen – ein Satz

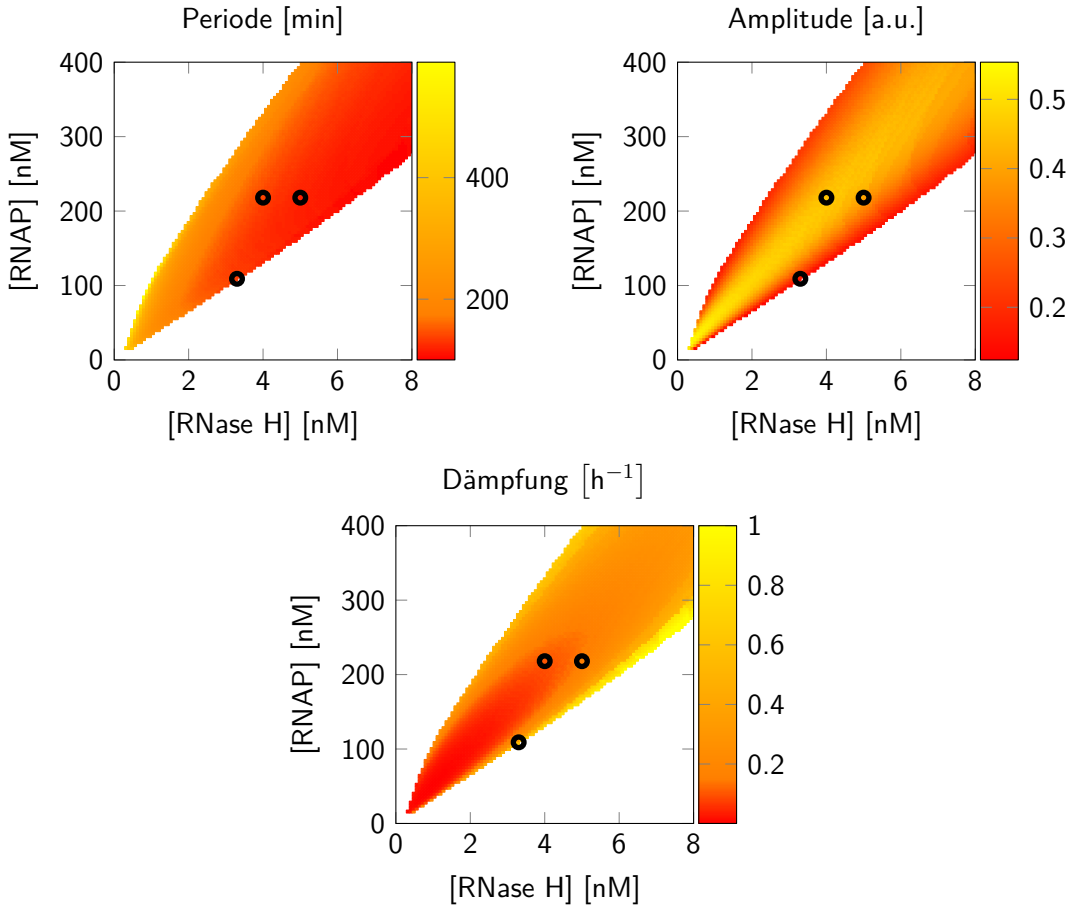


Abbildung 7.2.: Simulierter Scan durch verschiedene Proteinkonzentrationen. Die weißen Flächen sind Proteinkonstellationen, die nicht als oszillierend klassifiziert werden. Die drei schwarzen Kreise sind die Positionen der drei Stimmungen. Im Urzeigersinn links oben startend: stabil, gedämpft und stark gedämpft.
Es ist zu beachten, dass die Farbskala nur bei der Amplitude linear ist. Bei der Periode und der Dämpfung ist die Farbänderung im unteren Bereich größer als im oberen.

von Tröpfchen mit zufälligen Molekülanzahlen generiert. Diese Anzahlen werden dann in effektive Konzentrationen umgerechnet und das Modell mit diesen simuliert.

Der zusätzliche Verlustterm wird vor der Bestimmung des Mittelwertes der Proteinanzahlen angewendet. So wird dieser durch folgenden heuristischen[1] Term herabgesetzt:

$$[\mathrm{RNAP}]_{\mathrm{Verlust}}(r) = \left(\frac{0.0278 + 0.0107 \cdot \ln\left(\frac{r}{\mu\mathrm{m}}\right)}{0.083}\right) \cdot [\mathrm{RNAP}] \tag{7.1.2}$$

$$[\mathrm{RNase\ H}]_{\mathrm{Verlust}} = 0.9 \cdot [\mathrm{RNase\ H}] \tag{7.1.3}$$

[1] Um den Rahmen dieser Arbeit nicht zu sprengen, wird für nähere Informationen auf die Veröffentlichung und deren Zusatzmaterial verwiesen.[135]

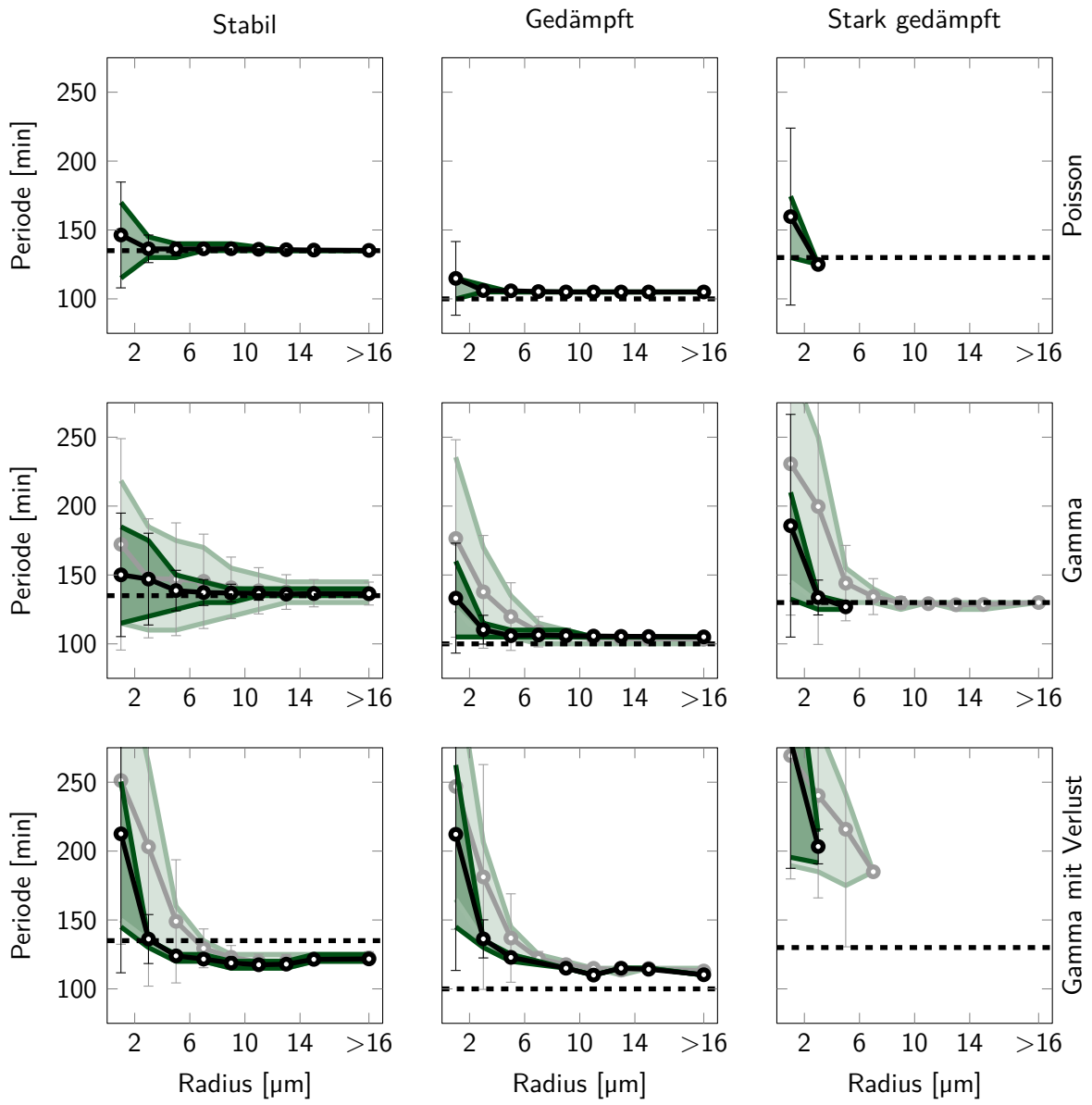


Abbildung 7.3.: Ergebnisse der Simulation mit verschiedenen Verteilungsfunktionen. Die Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die stochastischen Eigenschaften dieser Populationen sind dargestellt: die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil dar. Die gestrichelte Linie ist die Periode der Oszillatorkurve im makroskopischen Volumen. Die oberste Zeile zeigt die Simulation mit einer Poissonverteilung. Die zweite mit einer Gammaverteilung, wobei die Daten im Vordergrund für $\beta = 10$ und die transparenten Daten im Hintergrund für $\beta = 100$ sind. In der letzten Zeile ist eine Gammaverteilung mit Verlust gezeigt.

Im Phasendiagramm des Proteinscans (s. Abb. 7.2) verschiebt sich somit der Punkt der Stimmung in Richtung Ursprung. Die Verteilungsfunktionen kann man sich dann als Ellipsen um den Mittelwert vorstellen.

Die Auswertungen der Simulationen sind in Abb. 7.3 und Anhang C dargestellt.

7.1.6. Interpretation

Ein Vergleich zwischen Abb. 6.1 (Seite 96) und Abb. 7.3 lässt erkennen, dass die Poissonverteilung nicht ausreicht, um die Experimentergebnisse zu erklären. Alleine das Verhalten des Mittelwerts der verschiedenen Radiuspopulationen ist nicht stark genug vom Radius abhängig und auch die Breite der Population ist nicht groß genug. Die Gammaverteilungen hingegen passen besser auf die experimentellen Daten. So hat der Verlauf des Mittelwerts eine ähnliche Tendenz und auch die Breiten sind besser getroffen.

Das legt den Schluss nahe, dass Partitionierungseffekte die Variabilität erklären können, dass also keine unabhängige Aufteilung (s. 3.3.1) der Moleküle stattfindet.

Die Mittelwerte der Verteilung bei den Simulationen mit Gammaverteilung nähern sich aber zu schnell an den Referenzwert an. Das Annäherungsverhalten an den Referenzwert wird jedoh bei den „Gamma mit Verlust“-Simulationen recht gut getroffen und die Mittelwerte liegen auch näher an den Experimentwerten. Nur die Breiten sind etwas zu klein. Dennoch beschreibt ein Modell mit Verlust und einer Gammaverteilung, die β zwischen 10 und 100 verwendet, die Experimentaldaten am besten.

Somit kann man aus den Vergleichen zwischen den experimentellen und den modellierten Daten schließen, dass während der Tröpfchenproduktion die Proteine entweder denaturieren oder in der Ölphase gelöst werden. Der Verlust an Proteinaktivität ist von Protein zu Protein unterschiedlich. Das weist darauf hin, dass der Verlustterm durch verschiedene Prozesse zustande kommt: So scheint die RNase H während der Prozedur immer gleichförmig reduziert zu werden, z. B. indem ein Teil in der Ölphase gelöst wird. Da hingegen die Polymeraseaktivitätsreduktion radiusabhängig ist, wird sie entweder durch die genauen Vorgänge während der Tröpfchenproduktion, z. B. während der Tröpfchenteilungsereignisse, oder durch die Tröpfchengeometrie, z. B. Denaturierung an der Tröpfchenoberfläche, hervorgerufen.

7.2. Bakterien in Tröpfchen

Um das System der Bakterien zu beschreiben, wurde ein dreiteiliges Modell verwendet. Es wurden das Bakterienwachstum, die Ausbreitung des Signalstoffs und die Produktion der (fluoreszierenden) Proteine betrachtet.

7.2.1. Ausbreitung des Signalstoffs

Die Signalmoleküle wurden als einzige Molekülart als frei diffusiv betrachtet und durch die Diffusionsgleichung beschrieben. Es wurde angenommen, dass sich die Signalstoffe in der wässrigen Lösung viel schneller ausbreiten als im Öl, und da die Tröpfchen alle in einer Ebene liegen, kann man das Problem als zweidimensional ansehen. Obwohl es in der Ebene Tröpfchen, Ölzwischenräume und Grenzflächen gibt, wurde die Umgebung als isotrop mit einer effektiven Diffusionskonstante D_{eff} angesehen.

Das Koordinatensystem wurde so gewählt, dass sich das Tröpfchen mit dem Signalstoff im Ursprung befindet. Somit ist das Problem rotationssymmetrisch und kann durch Gleichung (3.1.6) (Seite 30) beschrieben werden.

Als Anfangsbedingung wurde ein Tröpfchen mit 25 µm Radius in der Mitte platziert. Somit ist die Konzentrationsverteilung zum Zeitpunkt $t = 0\,\text{s}$:

$$c(r,0) = \begin{cases} c_0 & \text{für } r < 25\,\mu\text{m} \\ 0 & \text{sonst} \end{cases} \tag{7.2.1}$$

wobei c_0 die Anfangskonzentration ist, die in die Signaltröpfchen gebracht wurde – bei AHL also 200 nM, bei IPTG 10 mM.

Bei der Simulation der Tröpfchen mit Bakterien, die AHL produzieren, wurde $c_0 = 0\,\text{nM}$ gewählt und die Produktion wurde als konstant im zentralen Tröpfchen angesehen. Dadurch ergibt sich ein ortsabhängiger Produktionsterm:

$$p(r) = \begin{cases} p & \text{für } r < 25\,\mu\text{m} \\ 0 & \text{sonst} \end{cases} \tag{7.2.2}$$

Der Abbau der Signalstoffe wurde als Vorgang erster Ordnung angesehen und somit ergibt sich die Abbaurate d zu:

$$d = -\delta c \tag{7.2.3}$$

Dabei wurde $\delta_{\text{AHL}} = 10^{-5}\,\text{s}^{-1}$[158] und $\delta_{\text{IPTG}} = 0\,\text{s}^{-1}$ angenommen.

Tabelle 7.2.: Transferfunktionsparameter

Molekül	y_0 [a.u. min^{-1}]	$y_{\max}$ [a.u. min^{-1}]	K [M]	n
AHL	0.29 ± 2.43	179.4 ± 6.2	$14.90 \pm 0.72 \times 10^{-9}$	1.62 ± 0.13
IPTG	0.29 ± 0.73	73.4 ± 1.5	$0.26 \pm 0.01 \times 10^{-3}$	1.32 ± 0.07

Tabelle 7.3.: Verwendete Skalierungsparameter der Transferfunktionen

Protein	y_0	$y_{\max}$
GFP	0	1
RNAP	0.02	0.1
RFP	0.02	10

7.2.2. Bakterienwachstum

Die Bakterien konnten in den Tröpfchen nicht frei wachsen, da sowohl die Nährstoffe als auch der Platz begrenzt waren. Ein Tröpfchen mit einem Radius von 25 µm hat etwa ein Volumen von 64 000 fl und somit könnten rein rechnerisch 64000 Bakterien mit einem Eigenvolumen von 1 fl hineinpassen. Dies wurde aber nie beobachtet, sondern die Bakterien wuchsen fünf bis zehn Stunden und formten dann einen lose zusammenhängenden Klumpen, dessen Radius kleiner als der halbe Tröpfchenradius war. Deswegen wurde die Maximalkapazität mit

$$N_{\max} \approx 100 - 1000 \tag{7.2.4}$$

abgeschätzt. Das Wachstum wurde dann mit einer logistischen Funktion modelliert:

$$N(t) = \frac{N_{\max} \cdot N_0 \cdot e^{\gamma \cdot t}}{N_{\max} + N_0 \cdot (e^{\gamma \cdot t} - 1)} \tag{7.2.5}$$

Dabei ist γ die Zellteilungsrate und N_0 die Anzahl der Bakterien, die sich am Anfang in den Tröpfchen befunden haben. Für die Simulationen wurden $N_{\max} = 100$, $N_0 = 1$ und $\gamma = 2\,\mathrm{h}^{-1}$ (AHL-sensitive Bakterien) bzw. $\gamma = 0.5\,\mathrm{h}^{-1}$ (IPTG-sensitive Bakterien) verwendet. Dieser Ratenunterschied ist nicht wegen verschiedener Bakterienstämme oder Nährmedien nötig, sondern nur wegen der verschiedenen Plasmide, die in die Bakterien transformiert wurden. So ist das Plasmid der IPTG-sensitiven Bakterien größer und veranlasst die Zelle, die AHL-Produktionsmaschinerie herzustellen. Dadurch stehen weniger Ressourcen für die Zellteilung zur Verfügung, was sich direkt in der Zellteilungsrate niederschlägt.

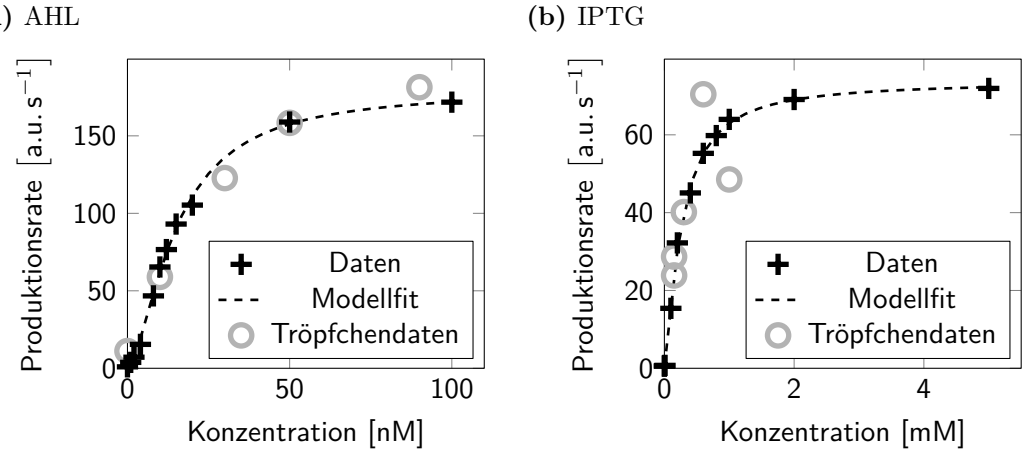


Abbildung 7.4.: Transferfunktionen für die Signalstoffinduzierung. Die Kreuze sind die Datenpunkte und die gestrichelten Linien die angefitteten Modellfunktionen. Die grauen Kreise sind Transferexperimente, die die Antwort in Tröpfchen maßen. Ihre Skalierung – $y_{\max}$ und y_0 – wurde an die Modellfunktion angefittet.

7.2.3. Proteinproduktion

Um die Antwort der Bakterien auf das AHL bzw. IPTG zu beschreiben, wurden Referenzmessungen in größeren Volumina durchgeführt. Die maximale Steigung der auf die Absorption normierten Fluoreszenz wurde dann gegen die Signalmolekülkonzentration aufgetragen und als Transferfunktion eine Hillfunktion angefittet:

$$y(c) = y_0 + \frac{y_{\max}}{1 + \left(\frac{K}{c}\right)^n} \tag{7.2.6}$$

wobei y_0 die basale und $y_{\max}$ die maximale Produktionsrate bei voller Induzierung ist. K ist die Konzentration, bei der die Produktion halb aktiviert ist, und n beschreibt, wie scharf der Übergang erfolgt.

Die gefitteten Werte für K und n (s. Tab. 7.2 und Abb. 7.4) liegen dabei in dem Bereich, der in der Literatur angegeben wird.[159, 160] Die Werte für y_0 und $y_{\max}$ sind beliebig skalierbar, da z. B. die Beleuchtungsintensität und -dauer diese beeinflussen. Deswegen können sie frei gewählt werden und es wurden die Werte aus Tab. 7.3 verwendet.

Damit kann die Produktion der Proteine in einem Tröpfchen in Abhängigkeit von der Bakterienanzahl und der Signalstoffkonzentration in dem Tröpfchen beschrieben werden. So wurde angenommen, dass die Proteinproduktion proportional zur Bakteriumswachstumsrate R ist:

$$R(t) = \frac{d}{dt}N(t) = \gamma \cdot \left(N(t) - \frac{N(t)^2}{N_{\max}}\right) \tag{7.2.7}$$

Um aber die Konzentration der fluoreszierenden Proteine in einem Tröpfchen zu beschreiben, müssen noch zwei weitere Vorgänge berücksichtigt werden: Degradation und Maturierung.

Degradation

Der Abbau der Proteine wird als Vorgang erster Ordnung angesehen. Somit ist die Abbaurate proportional zur aktuellen Proteinkonzentration [P]:

$$d = -\delta \cdot [\mathrm{P}] \tag{7.2.8}$$

Konkret wurde GFP und RFP als relativ stabil ($\delta = 2 \times 10^{-4}\,\mathrm{s}^{-1}$) und die T7-RNA-Polymerase als noch stabiler ($\delta = 10^{-5}\,\mathrm{s}^{-1}$) angenommen.

Maturierung

Wenn Proteine produziert werden, sind sie nicht sofort funktional und müssen erst reifen. Dieser Vorgang ist nicht vernachlässigbar, da er auf der gleichen Zeitskala wie die Experimente stattfindet. Deswegen werden zwei Proteinzustände angenommen – frisch produziert und maturiert –, deren Übergang in erster Ordnung stattfindet:

$$m = \alpha_{\mathrm{mat}} \cdot [\mathrm{P}] \tag{7.2.9}$$

Für die fluoreszierenden Proteine wurde eine Maturationszeit von 5 min angesetzt[161], woraus sich ein $\alpha_{\mathrm{mat}} = 3.3 \times 10^{-3}\,\mathrm{s}^{-1}$ ergibt.

GFP

Die Produktion von GFP ist im Prinzip nicht nur von der AHL-Konzentration abhängig, sondern auch von der Präsenz des LuxR-Proteins. Dieses wird aber konstitutiv produziert und es wurde deswegen angenommen, dass es seine Gleichgewichtskonzentration erreicht hat. Somit wurde dessen Kinetik nicht mitmodelliert.

RFP

RFP hingegen wird abhängig von zwei Faktoren produziert, da es durch einen IPTG-induzierbaren T7-RNA-Polymerase-Promotor kontrolliert wird. So wird dieser direkt über IPTG aktiviert, aber auch die Produktion der T7-RNA-Polymerase (RNAP), die dann die mRNA herstellt, wird durch IPTG angeschaltet. Deswegen muss hier noch eine weitere Stufe in das Produktionsmodell eingeführt werden. Deswegen wird die Produktion der Polymerase durch die oben beschriebene Transferfunktion beschrieben und RFP wird in Abhängigkeit von der Polymerasekonzentration produziert:

$$y_{\mathrm{RFP}}(c) = y_0 + \frac{y_{\mathrm{max,\,RFP}}}{1 + \left(\frac{K}{c}\right)^n} \cdot \frac{[\mathrm{RNAP}]}{[\mathrm{RNAP}] + K_{\mathrm{RNAP}}} \tag{7.2.10}$$

Dabei wird $K_{\mathrm{RNAP}} = 1\,\mathrm{nM}$ angenommen.

7.2.4. Resultierende Differentialgleichungssysteme

Wenn man nun alles zusammensetzt, erhält man für jeden Simulationstyp einen Satz von Differentialgleichungen.

AHL

Im Falle des AHL-Tröpfchens in der Mitte müssen die Konzentrationen von AHL, nicht maturiertem GFP und maturiertem GFP betrachtet werden. Am Ende ergeben sich dann folgende Differentialgleichungen:

$$\frac{\partial}{\partial t}\left[\mathrm{AHL}\right](r,t) = D_{\mathrm{eff}}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\left[\mathrm{AHL}\right]\right) - \delta_{\mathrm{AHL}}\left[\mathrm{AHL}\right] \tag{7.2.11}$$

$$\frac{d}{dt}\left[\mathrm{GFP}\right](r,t) = R\cdot\frac{y_{\max}\cdot\left[\mathrm{AHL}\right]^n}{\left[\mathrm{AHL}\right]^n + K^n} - \delta_{\mathrm{GFP}}\left[\mathrm{GFP}\right] \tag{7.2.12}$$

$$\frac{d}{dt}\left[\mathrm{GFP,mat}\right](r,t) = \alpha_{\mathrm{mat}}\cdot\left[\mathrm{GFP}\right] - \delta_{\mathrm{GFP}}\left[\mathrm{GFP,mat}\right] \tag{7.2.13}$$

Sender-Bakterien

Wenn das AHL in dem zentralen Tröpfchen nun durch Bakterien, die AHL produzieren, ausgetauscht wird, erweitert sich im vorherigen Differentialgleichungssystem nur Gleichung (7.2.11) um den Produktionsterm $p(r)$ aus Gleichung (7.2.2):

$$\frac{\partial}{\partial t}\left[\mathrm{AHL}\right](r,t) = D_{\mathrm{eff}}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\left[\mathrm{AHL}\right]\right) + p(r) - \delta_{\mathrm{AHL}}\left[\mathrm{AHL}\right] \tag{7.2.14}$$

IPTG

Das Differentialgleichungssystem für die IPTG-Tröpfchensystem braucht eine Gleichung mehr, da zusätzlich die T7-RNA-Polymerase betrachtet werden muss:

$$\frac{\partial}{\partial t}\left[\mathrm{IPTG}\right](r,t) = D_{\mathrm{eff}}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\left[\mathrm{IPTG}\right]\right) \tag{7.2.15}$$

$$\frac{d}{dt}\left[\mathrm{RNAP}\right](r,t) = R\cdot y_0 + R\cdot\frac{y_{\max,\ \mathrm{RNAP}}\cdot\left[\mathrm{IPTG}\right]^n}{\left[\mathrm{IPTG}\right]^n + K^n} - \delta_{\mathrm{RNAP}}\left[\mathrm{RNAP}\right] \tag{7.2.16}$$

$$\begin{aligned}\frac{d}{dt}\left[\mathrm{RFP}\right](r,t) = {} & R\cdot y_0 + R\cdot\frac{y_{\max,\ \mathrm{RFP}}\cdot\left[\mathrm{IPTG}\right]^n}{\left[\mathrm{IPTG}\right]^n + K^n}\cdot\frac{\left[\mathrm{RNAP}\right]}{\left[\mathrm{RNAP}\right] + K_{\mathrm{RNAP}}} \\ & - \delta_{\mathrm{RFP}}\left[\mathrm{RFP}\right]\end{aligned} \tag{7.2.17}$$

$$\frac{d}{dt}\left[\mathrm{RFP,mat}\right](r,t) = \alpha_{\mathrm{mat}}\cdot\left[\mathrm{RFP}\right] - \delta_{\mathrm{RFP}}\left[\mathrm{RFP,mat}\right] \tag{7.2.18}$$

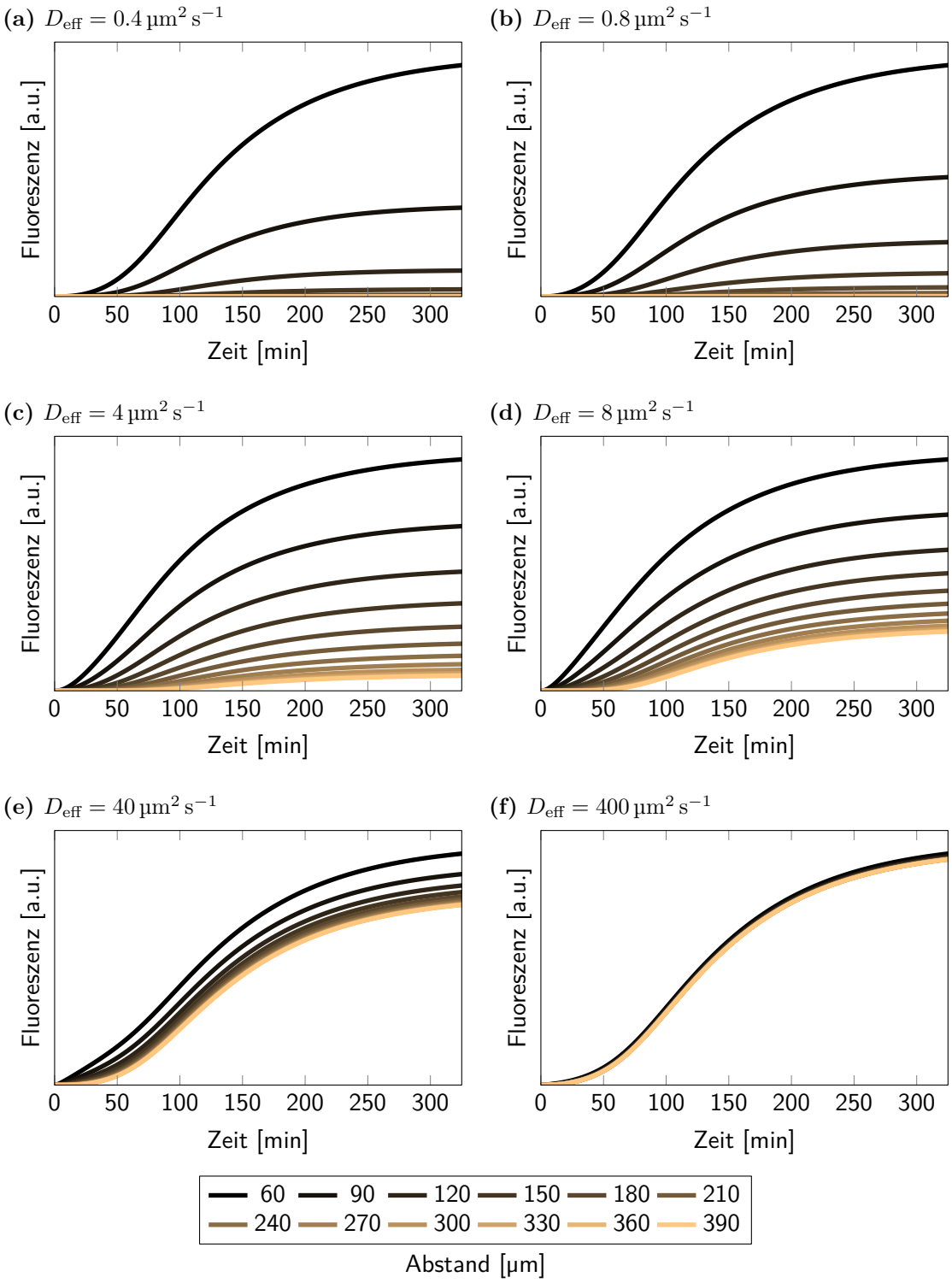


Abbildung 7.5.: Simulation der AHL-Empfänger bei verschiedenen effektiven Diffusionskonstanten.

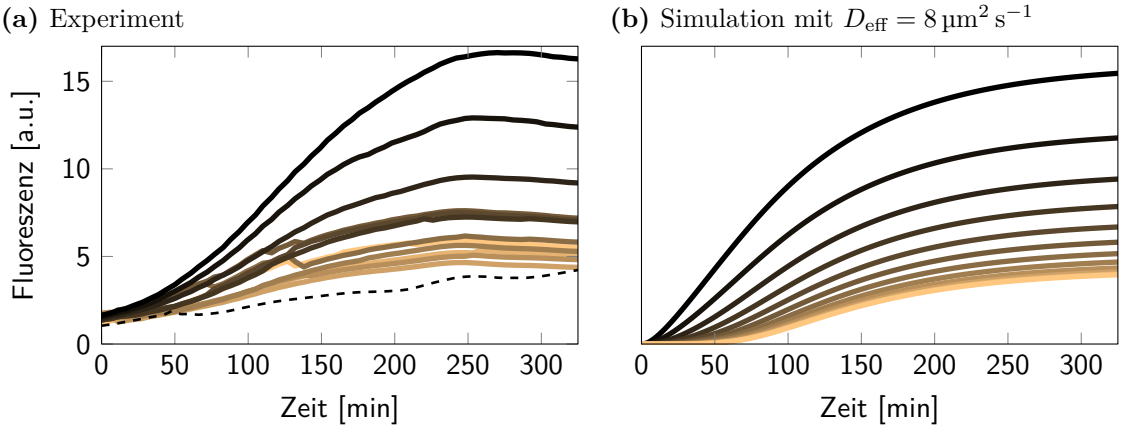


Abbildung 7.6.: Vergleich zwischen Simulation und Experiment der AHL-Empfänger.

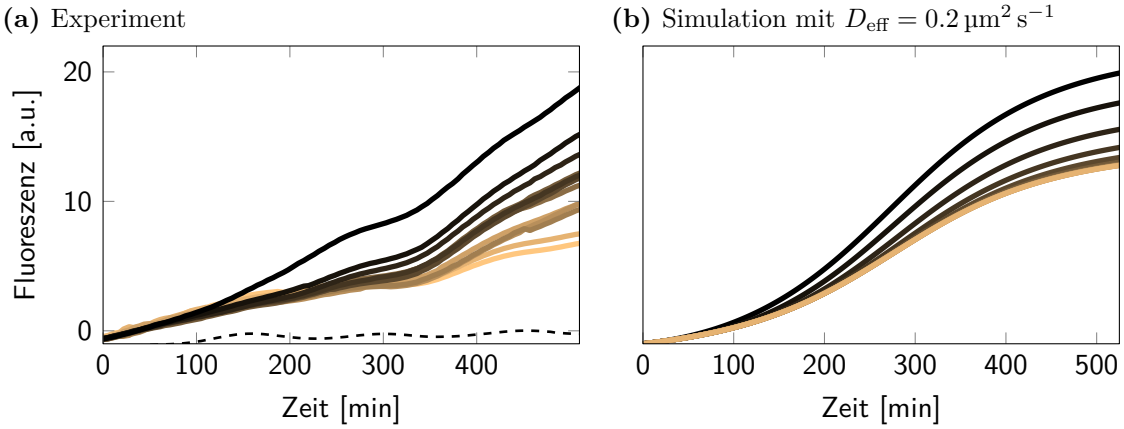


Abbildung 7.7.: Vergleich zwischen Simulation und Experiment der IPTG-Empfänger.

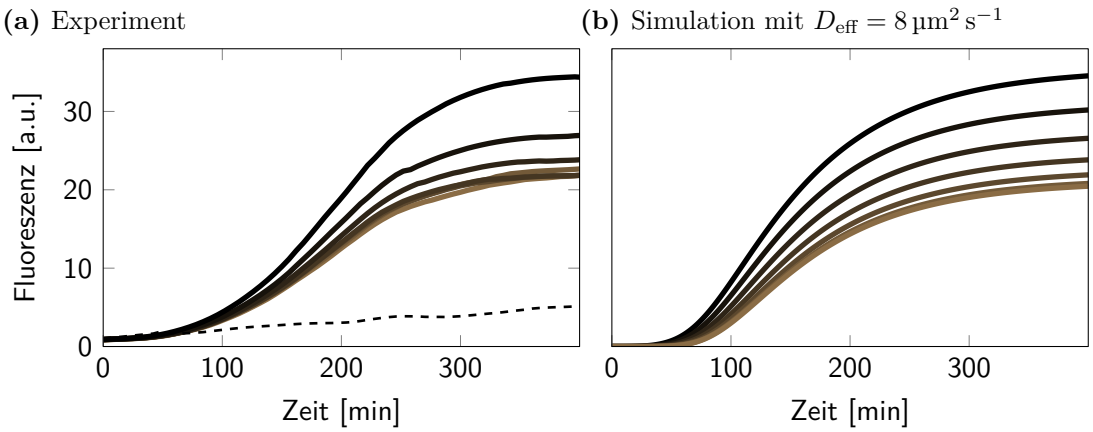
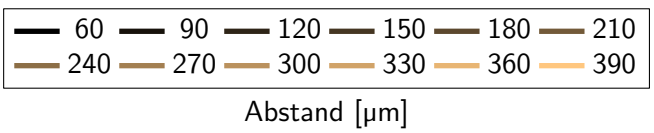


Abbildung 7.8.: Vergleich zwischen Simulation und Experiment des Sender-Empfänger-Systems.



Simulationen

Wenn man die Modelle mit verschiedenen effektiven Diffusionskonstanten simuliert[2], erkennt man unterschiedliche Abhängigkeiten für den Abstand zwischen dem Tröpfchen, in dem der Signalstoff bereitgestellt oder produziert wird, und dem Tröpfchen mit den Empfängerbakterien (s. Abb. 7.5). Durch Vergleich mit den experimentellen Daten sieht man, dass eine effektive Diffusionskonstante von $8\,\mu m^2\,s^{-1}$ für AHL und $0.2\,\mu m^2\,s^{-1}$ für IPTG das Verhalten qualitativ reproduziert (s. Abb. 7.6, Abb. 7.7 und Abb. 7.8).

7.2.5. Interpretation

Die experimentellen Daten werden durch die Simulation recht gut wiedergegeben, jedoch sind die benötigten effektiven Diffusionskonstanten viel kleiner als erwartet. So haben kleine Moleküle in Wasser typischerweise eine Diffusionskonstante im Bereich von $D_0 = 100 - 1000\,\mu m^2\,s^{-1}$. Die viermal höhere Viskosität des Öls ($\eta \approx 4.1\,mPa\,s$ bei $25\,°C$), die D_0 um diesen Faktor reduziert (s. 3.1.3), kann dieses Ergebnis nicht ganz erklären.

Wenn man annimmt, dass die Grenzfläche zwischen zwei Tröpfchen für die Signalstoffe frei passierbar ist, kann die reduzierte effektive Diffusionskonstante auch über Gleichung (3.1.13) beschieben werden. Um die beobachtete Diffusion zu erreichen, müsste der Radius der frei passierbaren Grenzfläche mindestens um den Faktor 10 kleiner sein als der Tröpfchenradius. Da aber Tröpfchen, die keine direkte Verbindung zu anderen Tröpfchen haben, kein signifikant anderes Verhalten zeigen, scheinen die direkten Grenzflächen zwischen Tröpfchen keine anderen Eigenschaften als die Tröpfchen-Öl-Grenzflächen zu besitzen.

Wenn die Grenzflächen aber nicht frei passierbar sind, kann über ihre Permeabilität κ und Gleichung (3.1.12) eine Reduktion der Diffusion beschrieben werden. Um die beobachteten effektiven Diffusionskonstanten zu erlangen, müsste die Permeabilität im Bereich $4 - 200\,nm\,s^{-1}$ liegen. Verglichen mit Zellwänden ist dieser Wert klein[112, 113] aber durchaus realistisch[162]. Somit kann die stark reduzierte Diffusionskonstante dadurch gut erklärt werden. Des Weiteren erscheint es einleuchtend, dass die Grenzfläche für die beiden Signalstoffe unterschiedliche Permeabilitäten aufweist. So kann auch der starke Unterschied zwischen $D_{\text{eff, AHL}}$ und $D_{\text{eff, IPTG}}$ erklärt werden.

[2]Da das Problem sowohl mehr als eine Dimension – Radius und Zeit – als auch höhere Ableitungen zeigt, kann kein einfacher Differentialgleichungslösungsalgorithmus verwendet werden, um die Modelle zu simulieren. Deswegen wurde dafür die Funktion pdepe, die solche Konstellationen handhaben kann, von MATLAB verwendet.

Tabelle 7.4.: Parameter der Transferfunktion des UND-Gatters

Parameter	Wert
y_0	4.4 ± 3.2 a.u. min^{-1}
$y_{\max}$	150.4 ± 8.9 a.u. min^{-1}
K_{AHL}	1.8 ± 0.2 nM
n_{AHL}	1.5 ± 0.2
K_{IPTG}	10.9 ± 1.9 µM
n_{IPTG}	1.1 ± 0.3

7.2.6. UND-Gatter

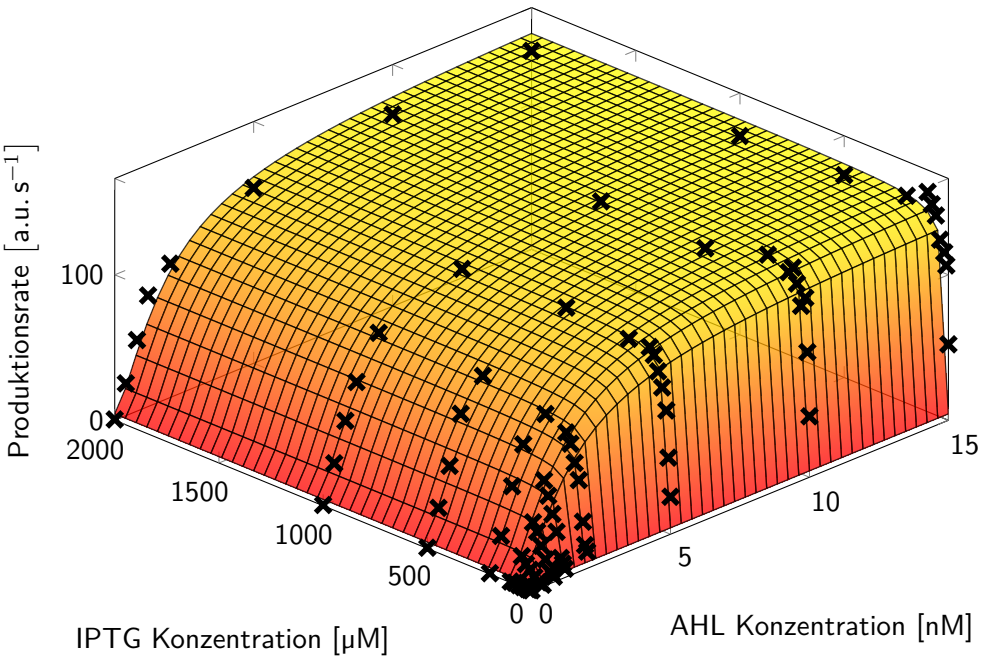


Abbildung 7.9.: Transferfunktionen für das UND-Gatter. Die Kreuze sind die Datenpunkte und die Fläche ist die angefittete Modellfunktion.

Die Experimente mit dem UND-Gatter wurden nicht so detailliert modelliert, da die geometrische Komplexität den Rahmen der Arbeit gesprengt hätte und wahrscheinlich kein signifikanter Wissensgewinn dadurch entstanden wäre. So wurde nur die zweidimensionale Antwortfunktion auf AHL und IPTG charakterisiert. Dabei wurde angenommen, dass die beiden Signalstoffe unabhängig voneinander agieren:[163, 164]

$$c([\mathrm{AHL}], [\mathrm{IPTG}]) = y_0 + \frac{y_{\max}}{\left(1 + \left(\frac{K_{\mathrm{AHL}}}{[\mathrm{AHL}]}\right)^{n_{\mathrm{AHL}}}\right) \cdot \left(1 + \left(\frac{K_{\mathrm{IPTG}}}{[\mathrm{IPTG}]}\right)^{n_{\mathrm{IPTG}}}\right)} \tag{7.2.19}$$

Die Daten sind in Abb. 7.9 und die gefitteten Parameter in Tab. 7.4 gezeigt. Dabei sieht man, dass die Unabhängigkeitsannahme bestätigt wird, da Funktion (7.2.19) die Datenpunkte gut beschreibt. Ein Vergleich mit Tab. 7.2 zeigt, dass die Kooperativitätskoeffizienten n_{AHL} und n_{IPTG} ähnlich sind – die Schwellwerte K_{AHL} und K_{IPTG} hingegen sind um eine Größenordnung kleiner.

7.3. Stochastik in Tröpfchen

Das Reaktionsnetzwerk in den Stochastikversuchen ist um einiges kürzer und weniger komplex als das Reaktionssystem des Oszillators. So kann es auch durch einen kleineren Satz von Differentialgleichungen beschrieben werden. Trotzdem wird ein vereinfachtes Modell benötigt, um die hohe Anzahl von Datenkurven bewältigen zu können.

7.3.1. Reaktionsmodell

Das chemische Reaktionsschema (vgl. Abb. 4.6) wird folgendermaßen aufgestellt:

$$\begin{aligned}
\mathrm{RNAP} + \mathrm{Templat} &\underset{k_-}{\overset{k_+}{\rightleftharpoons}} \mathrm{RNAP} \cdot \mathrm{Templat} \\
&\xrightarrow{k_{\mathrm{cat}}} \mathrm{RNAP} + \mathrm{Templat} + \mathrm{RNA} \\
\mathrm{RNA} + \mathrm{Reporter} &\xrightarrow{k_{\mathrm{activate}}} \mathrm{RNA} \cdot \mathrm{Reporter} \\
\mathrm{RNA} + \mathrm{Inhibitor} &\xrightarrow{k_{\mathrm{waste}}} \mathrm{RNA} \cdot \mathrm{Inhibitor} \\
\mathrm{RNA} \cdot \mathrm{Reporter} + \mathrm{Inhibitor} &\xrightarrow{k_{\mathrm{heal}}} \mathrm{RNA} \cdot \mathrm{Inhibitor} + \mathrm{Reporter}
\end{aligned}$$

Auch hier können die acht Differentialgleichungen (s. B.2) durch die Gesamtkonzentrationen, die Erhaltungsgrößen sind, reduziert werden (s. B.2.2).

Um die Parameter des Modells (s. Tab. 7.5) zu bestimmen, wird das Modell an die experimentellen Daten angefittet[3]. Die Genauigkeit des Fits wird dadurch noch gesteigert, dass sowohl die Originaldaten als auch die Ableitung[4] gefittet wurden. Da die Ableitung um eine Größenordnung kleiner ist, wurde sie um den Faktor 10 stärker gewichtet. Wie man in Abb. 7.10 sieht, lässt sich das Modell relativ gut an die Daten anfitten, allerdings nur unter der Annahme, dass die Polymeraseaktivität – beschrieben durch k_{cat} – zwischen den Experimenten stark schwankt. Solch unerwartet starke Schwankungen können auftreten, wenn verschiedene Proteinchargen verwendet werden und/oder sich die Präparationsbedingungen (Raumtemperatur, Luftfeuchtigkeit etc.) unterscheiden.

[3] Zum Simulieren des Modells wurde die Funktion ode23s von MATLAB verwendet.
[4] Die experimentelle Ableitung wurde mit der Methode aus F.10 mit einer Fensterbreite von 8 Datenpunkten berechnet.

(a) Konzentrationsverlauf

(b) Ableitung des Konzentrationsverlaufs

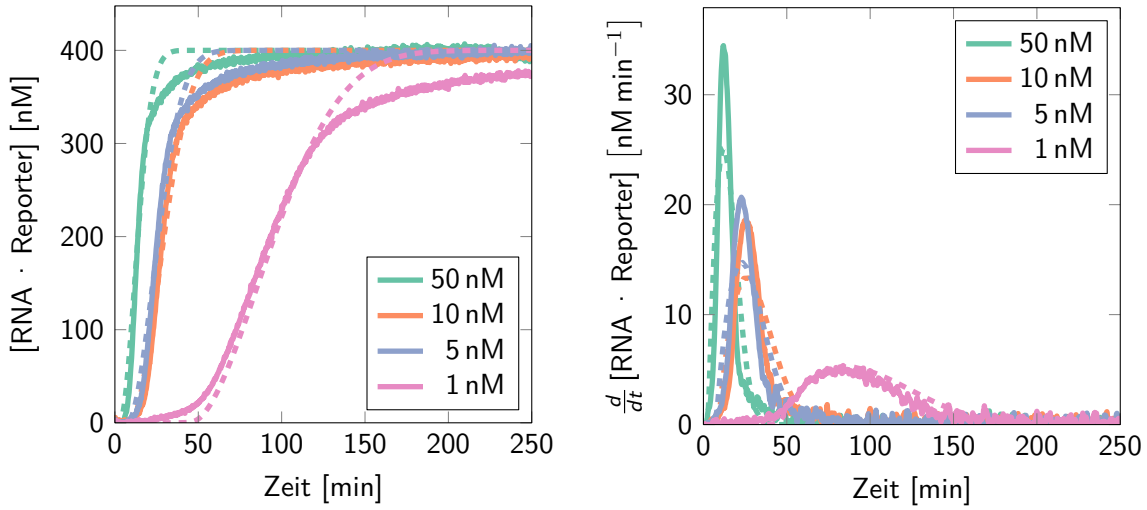


Abbildung 7.10.: Vergleich zwischen Simulation und Experiment. Die durchgezogenen Linien sind die Referenzkurven der Experimente im makroskopischen Volumen und die gestrichelten die angefitteten Simulationen. Man sieht, dass sowohl der zeitliche Verlauf der Konzentration des RNA-Reporter-Komplexes als auch die Ableitung recht gut getroffen werden.

Tabelle 7.5.: Parameter des Stochastikmodells. Die Konzentration der T7-RNA-Polymerase musste auf einen festen Wert eingestellt werden, da das System sonst überbestimmt gewesen wäre.

Parameter	Wert				Fitbereich
k_+ [M^{-1} s^{-1}]	5.15×10^6				$10^5 - 10^7$
k_- [s^{-1}]	1.06×10^{-2}				$10^{-2} - 10^0$
k_{activate} [M^{-1} s^{-1}]	8.69×10^2				$10^2 - 10^6$
k_{waste} [M^{-1} s^{-1}]	2.55×10^5				$10^3 - 10^6$
k_{heal} [M^{-1} s^{-1}]	2.98×10^5				$10^3 - 10^6$
[RNAP] [nM]	100				
[Inhibitor] [nM]	600				
[Reporter] [nM]	400				
[Templat] [nM]	50	10	5	1	
k_{cat} [s^{-1}]	0.101	0.155	0.619	0.404	$10^{-2} - 2$

Aber das Modell scheint die Wirklichkeit noch nicht gänzlich zu beschreiben, da die langsame Annäherung der experimentellen Kurven an 400 nM dadurch gar nicht erfasst wird. Welche Reaktionen genau hinzugefügt werden müssen, damit das Modell die experimentellen Kurven besser beschreibt, ist nicht ganz klar. Ein Ansatzpunkt könnte sein, dass die produzierte RNA durch ihre selbstkomplementäre Natur an sich selbst bindet und eine so stabile Sekundärstruktur bildet, dass der Reporter oder Inhibitor viel langsamer gebunden werden. Aber ein einfaches Modell mit einem gefalteten Zustand der RNA, in dem sie um einen konstanten Faktor langsamer reagiert, ergab auch keine erkennbar bessere Übereinstimmung zwischen Simulation und Experiment. Auch eine Erweiterung der Strangverdrängungsreaktion am Reporter auf einen Zweistufenprozess (s. 3.2.3.4) verbesserte das Modell nicht. Eventuell müsste zusätzlich jeder Einzelschritt der RNA-Polymerisation aus den rNTPs getrennt modelliert werden oder müssten die Strangreaktionen durch ein Molekularmodell (z. B. wie in [123] beschrieben) dargestellt werden. Solch komplexe Modelle hätten aber den Rahmen der Arbeit gesprengt und ein signifikanter Wissensgewinn ist ungewiss.

7.3.2. Vereinfachtes Modell

Das oben erläuterte Reakionsmodell ist zu komplex, um es an die Kurven jedes Tröpfchens (zwischen 3000 und 6000 pro Experiment – s. Tab. 5.3) anzufitten. So hätte alleine die Fitberechnung pro Experiment ca. 6 – 12 h in Anspruch genommen und hätte wahrscheinlich wegen der Sensitivität des Fits bezüglich der Anfangsparameter öfters wiederholt werden müssen. Um die Modellfunktion robuster gegenüber der Anfangsparameterwahl zu machen und die Fitprozedur zu beschleunigen, wurden Annahmen getroffen, die das Modell vereinfachen:

1. Die Produktionsrate der RNA $\frac{d}{dt}\left[\mathrm{RNA}^{\mathrm{tot}}\right]$ ist konstant k_{prod}. Da die RNA nur über die Polymerase produziert und nicht abgebaut wird – sie wird nur an verschiedene Stränge gebunden – ist diese Annahme äquivalent dazu, dass [RNAP · Templat] konstant ist.
2. Die RNA bindet instantan an den Inhibitor. Es wird also angenommen, dass k_{waste} unendlich groß ist.
3. Das Ablösen der RNA von einem Reporter durch einen freien Inhibitorstrang ist auch instantan. k_{heal} ist folglich ebenso unendlich groß.

Dadurch dauerte der Fitvorgang nur noch ca. 2 – 5 min pro Experiment.

Diese Annahmen sind durchaus realistisch. So sind die Raten k_{waste} und k_{heal} um drei Größenordnungen größer als k_{activate} (s. Tab. 7.5). Diese langsame Rate bleibt Teil des Modells und limitiert die Geschwindigkeit.

Aus diesen Annahmen ergibt sich folgender Reaktionsverlauf: solange freie Inhibitorstränge vorhanden sind, sieht man im Fluoreszenzsignal keinerlei Änderung, da alle Reporterstränge ungebunden vorliegen. Sobald alle Inhibitorstränge verbraucht sind –

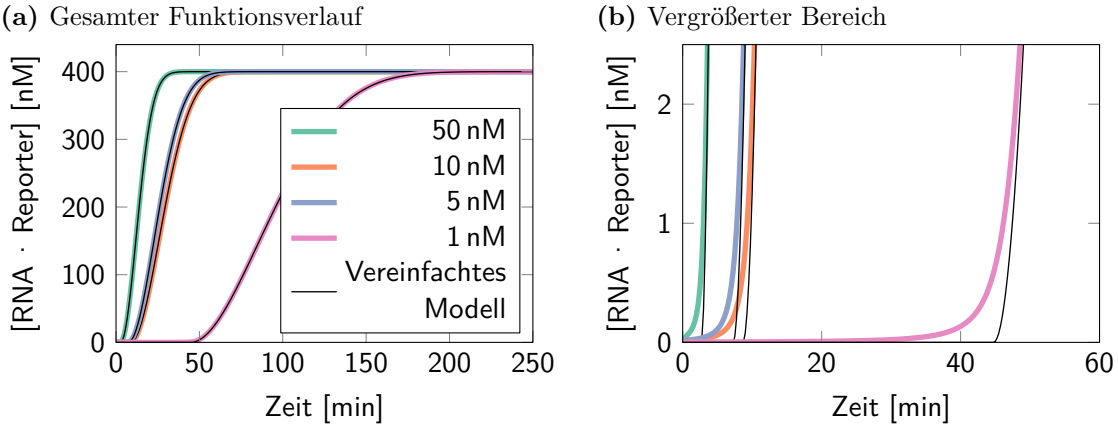


Abbildung 7.11.: Vergleich zwischen vollem und vereinfachtem Modell. Das volle Modell ist farbig und das vereinfachte Modell als dünne schwarze Linie darüber gezeichnet.

also nach $t_1 = \frac{[\text{Inhibitor}^{\text{tot}}]}{k_{\text{prod}}}$ –, kann die ab diesem Zeitpunkt produzierte RNA an den Reporter binden. Dabei wird der Wert von $k_{\text{activate}} = 8.7 \times 10^2\,\text{M}^{-1}\,\text{s}^{-1}$ aus dem Fit des Reaktionsmodells verwendet (s. Tab. 7.5). Somit ergibt sich folgendes Differentialgleichungssystem:

$$\frac{d}{dt}[\text{RNA}] + \frac{d}{dt}[\text{RNA} \cdot \text{Reporter}] = k_{\text{prod}} \tag{7.3.1}$$

$$\begin{aligned}\frac{d}{dt}[\text{RNA} \cdot \text{Reporter}] &= -\frac{d}{dt}[\text{Reporter}] = \\ &= k_{\text{activate}} \cdot [\text{Reporter}] \cdot [\text{RNA}]\end{aligned} \tag{7.3.2}$$

Mit den Randbedingungen, dass am Anfang keine RNA vorhanden ist und dass alle Reporter im ungebundenen Zustand vorliegen, lassen sich die Differentialgleichungen analytisch lösen (s. B.2.3):

$$[\text{RNA} \cdot \text{Reporter}](t) = R_{\text{tot}} - \frac{2 \cdot e^{k_a \cdot \left(-\left(\frac{k_{\text{prod}} \cdot t^2}{2}\right) + t \cdot R_{\text{tot}}\right)} \cdot R_{\text{tot}}}{2 + \sqrt{k_a \cdot 2 \cdot \frac{\pi}{k_{\text{prod}}}} \cdot R_{\text{tot}} \cdot e^{\frac{k_a \cdot R_{\text{tot}}^2}{2 \cdot k_{\text{prod}}}} \cdot \left[\text{erf}\left(\frac{\sqrt{k_a} \cdot (k_{\text{prod}} \cdot t - R_{\text{tot}})}{\sqrt{2 \cdot k_{\text{prod}}}}\right) + \text{erf}\left(\frac{\sqrt{k_a} \cdot R_{\text{tot}}}{\sqrt{2 \cdot k_{\text{prod}}}}\right)\right]} \tag{7.3.3}$$

Dabei ist aus Platzgründen $R_{\text{tot}} = \left[\text{Reporter}^{\text{tot}}\right]$ und $k_a = k_{\text{activate}}$ gewählt.

Ein Vergleich mit dem vollen Reaktionsmodell zeigt nur minimale Abweichungen – am Anfang der Fluoreszenzantwort – (s. Abb. 7.11) und rechtfertigt somit die Annahmen. Aus dem Modell mit 10 Parametern – 6 zu bestimmende Raten und 4 Anfangskonzentrationen – wird so ein vereinfachtes Modell mit 4 Parametern – eine feste Rate, eine zu bestimmende Rate und 2 Anfangskonzentrationen –, das die Ergebnisse ähnlich gut reproduziert.

Deshalb wurden die Fluoreszenzkurven der Tröpfchen mit diesem zweigeteilten Modell – zuerst keine Fluoreszenzänderung und anschließend Beschreibung des Fluoreszenzverlaufs mit Formel (7.3.3) – gefittet und die Produktionsrate k_{prod} verwendet (s. 5.3.3.2).

7.3.3. Verteilungsmodell

Um die Konzentrationsverteilungen in den Tröpfchen zu beschreiben, wird folgendes Modell verwendet:

Die anfänglich vorhandenen Molekülarten – die DNA, die T7-RNA-Polymerase und der Komplex aus diesen beiden – werden stochastisch auf die Tröpfchen aufgeteilt. Dabei wird angenommen, dass sie sich vor dem Partitionieren im chemischen Gleichgewicht (s. B.4) befinden. Während des Partitionierens werden die Molekültypen unabhängig voneinander aufgeteilt und danach stellt sich ein neues chemisches Gleichgewicht ein. Die dadurch modellierte Verteilung der Endkonzentration [DNA · RNAP] kann dann mit dem experimentell bestimmten $\frac{d}{dt}\left[\text{RNA}^{\text{tot}}\right]$ qualitativ verglichen werden.

Als Verteilungsfunktionen der DNA und des Komplexes werden Poissonverteilungen (s. A.1) gewählt – es wird also eine unabhängige Partitionierung angenommen (s. 3.3.1). Dies ist hingegen für die Aufteilung der T7-RNA-Polymerase nicht möglich, da man in Simulationen damit keine Radiusabhängigkeit des Variationsfaktors $\Phi_{\text{DNA · RNAP}}$ beobachtet (s. Abb. 7.12 – „Reine Poissonverteilung“). Wenn man aber in Betracht zieht, dass die Polymerase aggregieren kann[165, 166], kann ihre Aufteilung durch eine Gammaverteilung (s. A.3) beschrieben werden (s. 3.3.2.2 und Abb. 3.2a). Der Verteilungsparameter β, der die Varianz vom Mittelwert entkoppelt, hängt dabei von der mittleren Aggregatsgröße während der Tröpfchenproduktion ab.

Angenommen, das Schütteln beeinflusst die Reaktionskinetik des Anbindens und Ablösens von Einzelmolekülen an bzw. von den Aggregaten – es verändert sich z. B. das Verhältnis von k_{on} und k_{off} in Gleichung (B.3.1). Dadurch ändert sich der Gleichgewichtszustand und damit auch die mittlere Aggregatsgröße μ_{Agg} – ohne Schütteln μ_0 und mit Schütteln μ_∞. Wie der zeitliche Übergang zwischen den beiden Gleichgewichtszuständen genau aussieht, hängt stark von den Aggregationsmechanismen ab. Deswegen wird hier der Einfachheit halber ein exponentieller Abfall angenommen:

$$\mu_{\text{Agg}}(t) = (\mu_0 - \mu_\infty) \cdot e^{-\frac{t}{\tau_{\text{Agg}}}} + \mu_\infty \tag{7.3.4}$$

Damit hätte β nach Formel (3.3.16) eine zeitlich Abhängigkeit von:

$$\begin{aligned} \beta(t) = \frac{\sigma^2}{\mu} &= 2 \cdot \mu_{\text{Agg}} - 1 = \\ &= 2 \cdot (\mu_0 - \mu_\infty) \cdot e^{-\frac{t}{\tau_{\text{Agg}}}} + 2 \cdot \mu_\infty - 1 = \\ &= (\beta_0 - \beta_\infty) \cdot e^{-\frac{t}{\tau_{\text{Agg}}}} + \beta_\infty \end{aligned} \tag{7.3.5}$$

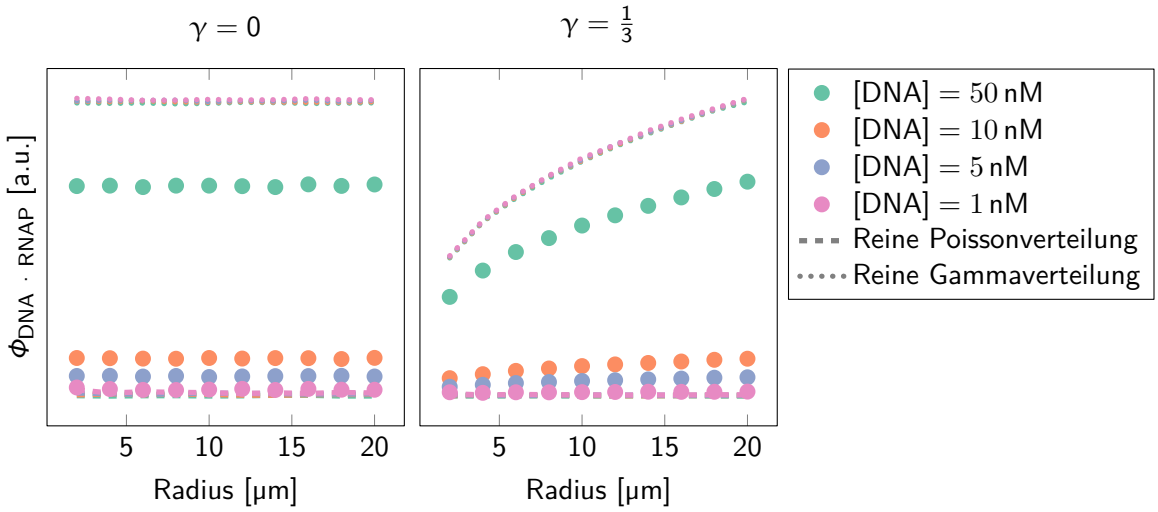


Abbildung 7.12.: Simulierte Abhängigkeit des Variationsfaktors von Tröpfchenradius und DNA-Konzentration, einmal mit $\gamma = \frac{1}{3}$ und einmal mit $\gamma = 0$. Dabei wurden jeweils 10^6 Tröpfchen simuliert. Die Konzentration der Polymerase wurde auf 100 nM gesetzt und $K_d = 65.5\,\text{nM}$. Die gestrichelten Linien sind Simulationen, in denen alle Molekülsorten mit einer Poissonverteilung aufgeteilt wurden, und die gepunkteten sind Simulationen, in denen alle mit einer Gammaverteilung aufgeteilt wurden[5]. Es sind hierbei auch jeweils vier Kurven – eine für jede Templatkonzentration – gezeichnet, die aber so dicht übereinander liegen, dass man sie kaum unterscheiden kann.

Zusätzlich wird angenommen, dass der Radius eines Tröpfchens Auskunft darüber gibt, wann es ungefähr erzeugt wurde – größere Tröpfchen entstehen tendenziell früher als kleinere, da letztere aus Teilungsprozessen der ersteren hervorgehen und der Surfactant eine kleine Tröpfchenfusionsrate zeigt. Auch hier wird naiv angenommen, dass der Radius über einen exponentiellen Abfall von der Zeit abhängt:

$$r(t) = r_0 \cdot e^{-\frac{t}{\tau_r}} \tag{7.3.6}$$

Wenn man nun Formel (7.3.6) nach t auflöst und in Gleichung (7.3.5) einsetzt, erhält man die Relation zwischen β und r:

$$\beta(r) = (\beta_0 - \beta_\infty) \cdot \left(\frac{r}{r_0}\right)^{\frac{\tau_r}{\tau_{\text{Agg}}}} + \beta_\infty \tag{7.3.7}$$

Im Folgenden wird das Verhältnis zwischen den zwei Zeitkonstanten mit $\gamma = \frac{\tau_r}{\tau_{\text{Agg}}}$ bezeichnet.

Um abzuschätzen, welchen Wert γ annimmt, wurde die beschriebene Teilungsprozedur simuliert. Dabei wurde willkürlich $r_0 = 1\,\mu\text{m}$, $\beta_0 = 10$ und $\beta_\infty = 1$ gewählt.[6] Die

[5] Aus Darstellungsgründen wurde hierbei die y-Achse reskaliert $\tilde{y} = \frac{y-1.25}{7.5} + 0.7$. Das ändert nichts an der Aussage, da hier nur der Trend der Kurve gezeigt werden soll.

[6] Simulationen mit anderem r_0, β_0 und/oder β_∞ erzeugten qualitativ sehr ähnliche Ergebnisse, so dass diese hier nicht gezeigt werden.

Variation von γ (in Abb. 7.12 sind $\gamma = 0$ und $\gamma = \frac{1}{3}$ gezeigt) und ein Vergleich mit den Realdaten zeigt, dass γ kleiner 1 ist. Der genaue Wert von γ ist aber weniger von Interesse als die quantitative Ausprägung der Variationsfaktoren. So ist die Tatsache, dass man im Variationsfaktor die gleiche Radiusabhängigkeit wie $\beta(r)$ sieht, bemerkenswert. Am erstaunlichsten ist aber, dass man verschieden starke Ausprägungen der Radiusabhängigkeit für die unterschiedlichen Templatkonzentrationen erhält. Bei [DNA] = 1 nM ist fast keine Änderung über den Radiusbereich sichtbar, aber bei [DNA] = 50 nM schon.

Wenn man hingegen alle drei Molekülsorten mit einer Gammaverteilung gleicher Radiusabhängigkeit beschreibt, zeigen sich zwischen den verschiedenen Templatkonzentrationen nur minimale Unterschiede im Variationsfaktorverlauf (s. Abb. 7.12 – „Reine Gammaverteilung“). Folglich ist die Aufteilung der Moleküle auf die Tröpfchen nicht für alle Molekülarten gleich.

Einige Annahmen und Aspekte dieses Modells sind experimentell leider kaum zugänglich. So gibt es keine Untersuchungen über die Größenverteilung solcher Proteinaggretate auf so kleinen Zeitskalen – nur die Aggregation durch stundenlanges Schütteln wurde untersucht[167]. Auch ist die genaue Relation zwischen Entstehungszeitpunkt und Radius rein heuristisch gewählt und nicht experimentell untermauert. Deswegen wird dieses Modell auch nur als mögliche Erklärung der experimentellen Ergebnisse gewertet und erhebt keinerlei Anspruch auf Gültigkeit oder Universalität.

7.3.4. Interpretation

Sowohl das Reaktionsmodell als auch seine Vereinfachung (s. 7.3.1 und 7.3.2) können verwendet werden, um die experimentellen Daten auszuwerten. Da sie beide im Fall der makroskopischen Referenzmessungen fast identisch aussehen (s. Abb. 7.11), kann das vereinfachte Modell ohne Nachteile für die Bestimmung der Produktionsraten in den Tröpfchen verwendet werden. Aber die Details der Dynamik der Messungen können durch beide Modelle nicht perfekt reproduziert werden (s. Abb. 7.10). Somit muss es noch Reaktionen geben, die im Reaktionsmodell nicht erfasst werden.

Des Weiteren kann man die Schlussfolgerung ziehen, dass die Dynamik der Reporteraktivierung durch die RNA-Produktion und die langsame Reaktion der RNA mit dem Reporterstrang dominiert wird. So führen die Seitenstrangreaktionen der RNA mit dem Inhibitor-Strang nur zu einer Verzögerung des Fluoreszenzsignals. Dadurch kann bei festem k_{activate} aus dem einfachen System die RNA-Produktionsrate extrahiert werden.

Das vorgestellte Aufteilungsmodell ist – wie schon oben erwähnt – nur eine mögliche Erklärung der Beobachtungen. Aber ein Vergleich zwischen den experimentell gemessenen Abhängigkeiten des Variationsfaktors vom Radius (s. Abb. 6.11a) und der Simulation (s. Abb. 7.12) zeigt eine gewisse qualitative Ähnlichkeit. Die beiden alternativen Aufteilungsmodelle – nur Poissonverteilungen oder nur Gammaverteilungen – haben hingegen eine ganz andere qualitative Aussage. Dies ist ein Hinweis darauf, dass die DNA tatsächlich einen anderen Aufteilungsmechanismus aufweist als die T7-RNA-Polymerase.

8. Schlussfolgerungen und Ausblick

8.1. Experimente

Zum Schluss sollen die wichtigsten Ergebnisse der drei Experimente rekapituliert, zusammengefasst und mögliche Anwendungsbereiche skizziert werden.

8.1.1. Oszillator in Tröpfchen

Die Ergebnisse der Oszillatorexperimente waren dahingehend bemerkenswert, dass die Tröpfchendaten zum einen signifikante Abweichungen von den Referenzmessungen zeigten und zum anderen eine breitere Verteilung des dynamischen Verhaltens aufwiesen als erwartet. Dies kann durch eine Variation der Anfangskonzentrationen in den Tröpfchen, die dann über das Reaktionsnetzwerk verstärkt und in ihrer Dynamik sichtbar wird, erklärt werden. Diese Verstärkung konnte auch in Simulationen mit variierten Anfangskonzentrationen reproduziert werden. Die Simulationen mit den nach der Poissonverteilung erwarteten Breiten konnten die Variabilitätsbreiten nicht erzeugen. Wenn hingegen eine Gammaverteilung, die einen höheren Variationsfaktor besitzt, verwendet wurde, wurden die Breiten gut reproduziert. Mit zusätzlichem Verlust von Proteinaktivität während der Tröpchenproduktion wurde qualitativ das gleiche Verhalten wie bei den experimentellen Ergebnissen erzeugt. Die genauen Auswirkungen dieser Konzentrationsvariabilität auf die Dynamik des Oszillators hängt stark davon ab, unter welchen Rahmenbedingungen – in welcher Stimmung – der Oszillator betrieben wird.

So weisen die Experimente darauf hin, dass Aufteilungseffekte einen starken Einfluss haben und damit eine Herausforderung bei der Entwicklung und Realisierung von künstlichen Protozellen mit robustem und reproduzierbarem Verhalten darstellen. Somit spielt die Empfindlichkeit synthetischer biologischer Systeme auf Partitionierungsrauschen eine wichtige Rolle. Das genauere Verständnis der Strategien, wodurch die Zellen dieses Rauschen unterdrücken oder Rauschunempfindlichkeit realisieren[168–170], kann dabei helfen, geeignete Lösungsansätze zu entwickeln.

Als Anwendungbereich dieser breiten Partitionierungsvariabilität wäre denkbar, dass man durch sie das mögliche dynamische Verhalten von nichtlinearen biochemischen Systemen erkundet und charakterisiert. Wenn man zusätzlich die Anfangskonzentrationen der wichtigsten Bestandteile des Systems direkt in jedem Tröpfchen messen könnte, hätte man eine Möglichkeit die Dynamik direkt damit in Bezug zu setzen und in einem einzigen Experiment das Phasendiagramm zu erstellen. Im Falle des Oszillators würde man die T7-RNA-Polymerase und die RNase H fluoreszent markieren[171] und –

mit verbesserter Messgenauigkeit – aus den Mikroskopvideos experimentell bestimmte Phasendiagramme in Abhängigkeit vom Radius erstellen.

8.1.2. Bakterien in Tröpfchen

Bei den Bakterienexperimenten wurde gezeigt, dass die beiden Signalstoffe AHL und IPTG Kommunikation zwischen chemischen Reservoirs und kleinen Bakterienkolonien, die in Wasser-in-Öl-Emulsionströpfchen eingebracht wurden, ermöglichen können. Da Quorum-Sensing normalerweise nur in komplexen Umgebungen – wie Biofilmen – auftritt, können Emulsionssysteme ein Modellsystem für Bakterienkommunikation in heterogenen Medien sein.

Zusätzlich kann die beobachtete geringe effektive Diffusionskonstante generell für das Feld der synthetischen Biologie nützlich sein. So haben andere Studien genetische Schaltkreise vorgestellt, die selbstbildende Strukturen mit Mustern im Millimeterbereich erzeugen.[172, 173] Durch die geringe effektive Diffusionskonstante können die Längenskalen um den Faktor $\sqrt{D_0/D_{\text{eff}}}$ reduziert werden.

Einen möglichen Anwendungsfall für Bakterien in Tröpfchen stellt die Realisierung von Kommunikation zwischen Bakterien, die unterschiedliche Umgebungsbedingungen – z. B. pH-Wert oder Salzkonzentration – benötigen, dar. Jeder Bakterientyp könnte sein ideales Nährmedium im Tröpfchen vorfinden und trotzdem mit den anderen über die Ölphase kommunizieren.

Die Tröpfchen bieten auch eine Möglichkeit, agentenbasierte Berechnungen durchzuführen, wobei aber jeder Agent aus einer kleinen Bakterienkolonie besteht. Es können also Hybridanwendungen mit separierten Berechnungseinheiten – Agenten – realisiert werden, wobei aber in jeder Einheit die Berechnung durch mehrere Bakterien ausgeführt wird. Somit werden die Fluktuationen der Einzelzellen[174] gemittelt, wodurch der Agent verlässlicher wird.

8.1.3. Stochastik in Tröpfchen

Im Gegensatz zu den Oszillatorexperimenten zeigte die globale Dynamik des Systems – also der Mittelwert der Produktionsraten – keine starke Abhängigkeit vom Radius der Tröpfchen. Da es sich beim Reaktionsnetzwerk um ein lineares System handelt, entsprach das den Erwartungen. Beim Variabilitätskoeffizienten hingegen konnte teilweise eine starke Abhängigkeit festgestellt werden. Die Stärke der Abhängigkeit korreliert mit der verwendeten Templat-DNA-Konzentration – je höher die Konzentration, desto ausgeprägter die Abhängigkeit. Ein solches Verhalten wurde von dem System nicht erwartet und ist kontraintuitiv.

In Simulationen, die für alle Bestandteile des Systems – DNA-Stränge und Protein – gleiche Verteilungsfunktionen verwendeten, konnte diese doppelte Abhängigkeit nicht reproduziert werden. So zeigte eine Poissonverteilung keinerlei Abhängigkeit vom Radius und eine Gammaverteilung keine Korrelation mit der Templatkonzentration. Wenn man aber für die DNA-Stränge und das Protein verschiedene Verteilungsfunktionen verwendete, war qualitativ ein ähnliches Verhalten wie in den Experimenten zu sehen.

Ähnlich wie beim Oszillator müssen diese Ergebnisse bei der Entwicklung und Realisierung von künstlichen Zellen beachtet werden. Es muss also für jeden Bestandteil separat geprüft werden, wie er auf die verschiedenen Kompartimente verteilt wird, um dann adäquat darauf zu reagieren.

Wenn es aber möglich wäre, die Verteilungsfunktionen der Bestandteile unabhängig voneinander zu beeinflussen und zu variieren – z. B. durch genetische Modifikation der Proteinoberfläche und damit Änderung des Aggregationsverhaltens –, könnte das die Kontrollierbarkeit der oben erwähnten Experimente zur Erkundung und Charakterisierung des dynamischen Verhaltens synthetischer Systeme erhöhen. Dadurch könnte man dann z. B. die Genauigkeit, mit der ein bestimmter Parameterbereich charakterisiert wird, einstellen – also ob ein weiter Parameterbereich abgedeckt wird oder ob ein kleinerer Bereich mit einer erhöhten Statistik untersucht wird.

8.2. Datenanalyse

Im Rahmen der beschriebenen Experimente und deren Auswertung sind die in dieser Arbeit vorgestellten Algorithmen (s. Anhang F) und Auswertungsideen entstanden. Dabei wurde ein MATLAB-Softwarepaket entwickelt, das u. a. eine schnelle Auswertung von Tröpfchenvideos ermöglicht. So benötigte die Einzelbild- und Videoanalyse (s. 5.1 und 5.2) meist 2 – 20 min pro Video. Das zuerst verwendete Programm – CellEvaluator[133] – hingegen hat dazu teilweise bis zu einen Tag in Anspruch genommen und konnte manche Videos auch nur ausschnittsweise auswerten.

Das Paket kann von Github als MATLAB-App für Windows 64bit und Mac OS X 64bit heruntergeladen werden. Unter https://github.com/kkapsner/Matlab kann der vollständige Quelltext[1] der Algorithmen und der Benutzeroberfläche eingesehen und verwendet werden. Somit ist auch eine Nutzung unter anderen Betriebssystemen möglich.

Die Software wurde so modular gestaltet, dass einzelne Teile verändert, erweitert oder sogar ganz ausgetauscht werden können. So konnten damit auch schon Videos ausgewertet werden, die keine Tröpfchen zeigten, sondern Bakterien. Dazu wurde das in 5.2.2.2 vorgestellte semiautomatische Tracking verwendet – es wurde also das Trackingmodul ausgetauscht. Die Methode kann aber noch optimiert werden und ihre genauere Beschreibung hätte den Rahmen der Arbeit gesprengt.

Die Nützlichkeit der Auswertsoftware über diese drei Experimente hinaus zeigt sich darin, dass sie auch schon für andere Projekte verwendet wurde und wird. Bisher wurde sie in der Bachelorarbeit von Lukas Aufinger[175] eingesetzt. Laufende Projekte mit dem Paket sind:

1. FRET an Multiaptamer-RNA in Tröpfchen – Matthaeus Schwarz-Schilling
2. Kommunikation zwischen Tröpfchen über Membranproteine – Aurore Dupin
3. Bakterienwachstum in Mikrofluidikfallen – Korbinian Kapsner mit Andrea Meyer

Man sieht also, dass die Methoden, Algorithmen und Implementierungen für die Analyse von Mikroskopvideos nützlich waren, sind und sein werden. Außerdem wird durch den wachsenden Fokus der synthetischen Biologie auf Einzelzell-, Tröpfchen- oder Vesikelexperimente die Analyse solcher Videodaten noch an Bedeutung gewinnen.

[1] Der ganze Quelltext ist mit der MIT-Lizenz lizenziert.

9. Danksagungen

Abschließend möchte ich mich bei allen bedanken, die diese Dissertation ermöglicht haben. Der erste Dank gilt meinem Doktorvater Prof. Dr. Friedrich C. Simmel, der die Rahmenbedingungen geschaffen und mir mit Rat und Tat zur Seite gestanden hat. Auch meinem Zweitgutachter gilt ein herzliches „Dankeschön“.

Ohne meine Kollaborationspartner Ass. Prof. Dr. Elisa Franco, Dr. Jongmin Kim, Andrea Meyer, Andrea Mückl, Dr. Maximilian Weitz und Prof. Dr. Erik Winfree wären die Arbeiten über den Oszillator und die Bakterien wahrscheinlich gar nicht entstanden. Besonders Dr. M. Weitz und A. Mückl haben mir als Lehrstuhlkollegen mit wertvollen Diskussionen sehr geholfen.

Generell haben mich die Denkanstöße, Tipps und Diskussionen aller Kollegen des Lehrstuhls E14 weitergebracht. Deswegen auch an sie alle ein herzliches „Dankeschön“.

Zu guter Letzt und ganz besonders möchte ich mich bei meiner Frau und meiner Familie bedanken, die mich während meiner Doktorandenzeit stets unterstützten und bestärkten.

A. Eigenschaften verschiedener Verteilungsfunktionen

A.1. Poissonverteilung

Eine Poissonverteilung wird durch einen freien Parameter $\lambda \in \mathbb{R}_+$ festgelegt.

Form:

$$p\left(n=x\right) = \begin{cases} \dfrac{\lambda^x}{x!} e^{-\lambda} & x \in \mathbb{N}_0 \\ 0 & \text{sonst} \end{cases} \tag{A.1.1}$$

Normierung:

$$\sum_{x=0}^{\infty} p\left(n=x\right) = \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} e^{-\lambda} = e^{-\lambda} \cdot \underbrace{\sum_{x=0}^{\infty} \frac{\lambda^x}{x!}}_{=e^{\lambda}} = 1 \tag{A.1.2}$$

Mittelwert:

$$\begin{aligned} \langle n \rangle &= \sum_{x=0}^{\infty} x \cdot p\left(n=x\right) = \\ &= \sum_{x=0}^{\infty} x \cdot \frac{\lambda^x}{x!} e^{-\lambda} = \\ &= \sum_{x=1}^{\infty} x \cdot \frac{\lambda^x}{x!} e^{-\lambda} = \\ &= e^{-\lambda} \cdot \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-1)!} = \\ &= e^{-\lambda} \cdot \sum_{x=1}^{\infty} \lambda \cdot \frac{\lambda^{x-1}}{(x-1)!} = \\ &= \lambda \cdot e^{-\lambda} \cdot \underbrace{\sum_{k=0}^{\infty} \frac{\lambda^k}{k!}}_{=e^{\lambda}} = \lambda \end{aligned} \tag{A.1.3}$$

Zweites Moment:

$$\begin{aligned}
\left\langle n^2 \right\rangle &= \sum_{x=0}^{\infty} x^2 \cdot p\left(n = x\right) = \\
&= \sum_{x=0}^{\infty} x^2 \cdot \frac{\lambda^x}{x!} e^{-\lambda} = \\
&= e^{-\lambda} \cdot \sum_{x=1}^{\infty} x \cdot \frac{\lambda^x}{(x-1)!} = \\
&= e^{-\lambda} \cdot \left(\sum_{x=1}^{\infty} \frac{\lambda^x}{(x-2)!} + \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-1)!} \right) = \\
&= e^{-\lambda} \cdot \left(\sum_{x=2}^{\infty} (x-1) \cdot \frac{\lambda^x}{(x-1)!} + \sum_{k=0}^{\infty} \lambda \frac{\lambda^k}{k!} \right) = \\
&= e^{-\lambda} \cdot \left(\lambda^2 \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} + \lambda \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \right) = \\
&= e^{-\lambda} \cdot \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \cdot \left(\lambda^2 + \lambda \right) = \lambda^2 + \lambda
\end{aligned} \tag{A.1.4}$$

Varianz:

$$\begin{aligned}
\left\langle \left(n - \langle n \rangle\right)^2 \right\rangle &= \left\langle n^2 - 2n \langle n \rangle + \langle n \rangle^2 \right\rangle = \\
&= \left\langle n^2 \right\rangle - 2 \langle n \rangle \langle n \rangle + \langle n \rangle^2 = \\
&= \left\langle n^2 \right\rangle - \langle n \rangle^2 = \lambda
\end{aligned} \tag{A.1.5}$$

Variationsfaktor Φ:

$$\Phi = \frac{\text{Varianz}}{\text{Mittelwert}} = \frac{\lambda}{\lambda} = 1 \tag{A.1.6}$$

Variationskoeffizient Ψ:

$$\Psi = \frac{\text{Standardabweichung}}{\text{Mittelwert}} = \frac{\sqrt{\lambda}}{\lambda} = \frac{1}{\sqrt{\lambda}} \tag{A.1.7}$$

Wenn nun der Mittelwert durch

$$\langle n \rangle = c \cdot \frac{4}{3} \cdot \pi \cdot r^3 \tag{A.1.8}$$

gegeben ist, ergibt sich Folgendes:

$$\Psi = \frac{1}{\sqrt{c \cdot \frac{4}{3} \cdot \pi}} \cdot r^{-1.5} \tag{A.1.9}$$

Somit wird eine Poissonverteilung allein durch die Angabe des Mittelwertes λ komplett bestimmt und die Breite der Verteilung kann nicht gesondert eingestellt werden.

A.2. Binomialverteilung

Eine Binomialverteilung wird durch eine Wahrscheinlichkeit $p \in [0,1]$ und durch eine Anzahl $N \in \mathbb{N}_0$ festgelegt.

Um die Eigenschaften zu berechnen, werden zwei mathematische Aussagen benötigt. Eine ist die binomische Formel:

$$(a+b)^c = \sum_{i=0}^{c} \binom{c}{i} a^i \cdot b^{c-i} \tag{A.2.1}$$

und die zweite ist eine Umformung des Binomialkoeffizienten:

$$\binom{a}{b} = \frac{a!}{b!\,(a-b)!} = \frac{a}{b} \cdot \frac{(a-1)!}{(b-1)!\,(a-b)!} = \frac{a}{b} \cdot \binom{a-1}{b-1} \tag{A.2.2}$$

Form:

$$p\,(n=x) = \begin{cases} \binom{N}{x} \cdot p^x \cdot (1-p)^{N-x} & x \in \mathbb{N}_0 \\ 0 & \text{sonst} \end{cases} \tag{A.2.3}$$

Normierung:

$$\begin{aligned} \sum_{x=0}^{N} p\,(n=x) &= \sum_{x=0}^{N} \binom{N}{x} \cdot p^x \cdot (1-p)^{N-x} \underbrace{=}_{\text{(A.2.1)}} \\ &= (p+(1-p))^N = 1 \end{aligned} \tag{A.2.4}$$

Mittelwert:

$$\begin{aligned} \langle n \rangle &= \sum_{x=0}^{N} x \cdot p\,(n=x) = \\ &= \sum_{x=0}^{N} x \cdot \binom{N}{x} \cdot p^x \cdot (1-p)^{N-x} \underbrace{=}_{\text{(A.2.2)}} \\ &= \sum_{x=1}^{N} x \cdot \frac{N}{x} \binom{N-1}{x-1} \cdot p^x \cdot (1-p)^{N-x} = \\ &= \sum_{x=1}^{N} Np \cdot \binom{N-1}{x-1} \cdot p^{x-1} \cdot (1-p)^{N-x} \underbrace{=}_{y=x-1} \\ &= Np \cdot \sum_{y=0}^{N-1} \binom{N-1}{y} \cdot p^y \cdot (1-p)^{N-1-y} \underbrace{=}_{\text{(A.2.1)}} \\ &= Np \cdot (p+(1-p))^{N-1} = Np \end{aligned} \tag{A.2.5}$$

Zweites Moment:

$$\begin{aligned}
\left\langle n^2 \right\rangle &= \sum_{x=0}^{N} x^2 \cdot p\,(n = x) = \\
&= \sum_{x=0}^{N} x^2 \cdot \binom{N}{x} \cdot p^x \cdot (1-p)^{N-x} \underbrace{=}_{\text{(A.2.2)}} \\
&= \sum_{x=1}^{N} x^2 \cdot \frac{N}{x} \binom{N-1}{x-1} \cdot p^x \cdot (1-p)^{N-x} = \\
&= Np \cdot \sum_{x=1}^{N} (x-1) \cdot \binom{N-1}{x-1} \cdot p^{x-1} \cdot (1-p)^{N-x} \\
&\quad + Np \cdot \sum_{x=1}^{N} \binom{N-1}{x-1} \cdot p^{x-1} \cdot (1-p)^{N-x} = \\
&= Np \cdot \sum_{x=2}^{N} (x-1) \cdot \frac{N-1}{x-1} \binom{N-2}{x-2} \cdot p^{x-1} \cdot (1-p)^{N-x} + Np \underbrace{=}_{y=x-2} \\
&= Np^2\,(N-1) \cdot \sum_{y=0}^{N-2} \binom{N-2}{y} \cdot p^y \cdot (1-p)^{N-2-y} + Np \underbrace{=}_{\text{(A.2.1)}} \\
&= Np^2\,(N-1) \cdot (p + (1-p))^{N-2} + Np = N^2p^2 - Np^2 + Np
\end{aligned} \tag{A.2.6}$$

Varianz:

$$\begin{aligned}
\left\langle (n - \langle n \rangle)^2 \right\rangle &= \left\langle n^2 \right\rangle - \langle n \rangle^2 = \\
&= N^2p^2 - Np^2 + Np - N^2p^2 = \\
&= Np - Np^2 = Np\,(1-p)
\end{aligned} \tag{A.2.7}$$

Bei bekanntem Mittelwert μ und Standardabweichung σ ergibt sich somit:

$$N = \frac{\mu^2}{\mu - \sigma^2} \tag{A.2.8}$$

$$p = 1 - \frac{\sigma^2}{\mu} \tag{A.2.9}$$

Variationsfaktor:

$$\Phi = 1 - p \tag{A.2.10}$$

Variationskoeffizient:

$$\Psi = \frac{\sqrt{Np\,(1-p)}}{Np} = \sqrt{\frac{1-p}{Np}} \tag{A.2.11}$$

A.3. Gammaverteilung

Eine Gammaverteilung wird durch die Parameter α und $\beta \in \mathbb{R}_+$ definiert.

Form:

$$p(n=x) = \begin{cases} \dfrac{1}{\beta^\alpha \cdot \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{A.3.1}$$

Wobei $\Gamma(\alpha)$ die Eulersche Gammafunktion ist:

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx \tag{A.3.2}$$

Normierung:

$$\begin{aligned} \int_0^\infty p(n=x)dx &= \int_0^\infty \frac{1}{\beta^\alpha \cdot \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} dx = \\ &= \frac{\int_0^\infty x^{\alpha-1} e^{-\frac{x}{\beta}} dx}{\beta^\alpha \cdot \Gamma(\alpha)} \underbrace{=}_{\frac{x}{\beta}=y} \\ &= \frac{\int_0^\infty y^{\alpha-1} \cdot \beta^{\alpha-1} \cdot e^{-y} \cdot \beta \cdot dy}{\beta^\alpha \cdot \Gamma(\alpha)} = \\ &= \frac{\beta^\alpha \cdot \int_0^\infty y^{\alpha-1} e^{-y} dy}{\beta^\alpha \cdot \int_0^\infty x^{\alpha-1} e^{-x} dx} = 1 \end{aligned} \tag{A.3.3}$$

Mittelwert:

$$\begin{aligned} \langle n \rangle &= \int_0^\infty x \cdot \frac{1}{\beta^\alpha \cdot \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} dx = \\ &= \frac{\int_0^\infty x^\alpha e^{-\frac{x}{\beta}} dx}{\beta^\alpha \cdot \Gamma(\alpha)} \underbrace{=}_{\frac{x}{\beta}=y} \\ &= \frac{\int_0^\infty y^\alpha \cdot \beta^\alpha \cdot e^{-y} \cdot \beta \cdot dy}{\beta^\alpha \cdot \Gamma(\alpha)} = \\ &\qquad = \alpha \int_0^\infty y^{\alpha-1} e^{-y} dy - \overbrace{\left[y^\alpha e^{-y} \right]_0^\infty}^{=0} \\ &= \frac{\beta^{\alpha+1} \cdot \overbrace{\int_0^\infty y^\alpha e^{-y} dy}}{\beta^\alpha \cdot \Gamma(\alpha)} = \\ &= \frac{\alpha\beta \int_0^\infty y^{\alpha-1} e^{-y} dy}{\int_0^\infty x^{\alpha-1} e^{-x} dx} = \alpha\beta \end{aligned} \tag{A.3.4}$$

Zweites Moment:

$$\begin{aligned}\left\langle n^2 \right\rangle &= \int_0^\infty x^2 \cdot \frac{1}{\beta^\alpha \cdot \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}} dx = \\ &= \frac{\int_0^\infty x^{\alpha+1} e^{-\frac{x}{\beta}} dx}{\beta^\alpha \cdot \Gamma(\alpha)} \underbrace{=}_{\frac{x}{\beta}=y} \\ &= \frac{\int_0^\infty y^{\alpha+1} \cdot \beta^{\alpha+1} \cdot e^{-y} \cdot \beta \cdot dy}{\beta^\alpha \cdot \Gamma(\alpha)} = \\ &= \frac{\beta^{\alpha+2} \cdot \overbrace{\int_0^\infty y^{\alpha+1} e^{-y} dy}^{=(\alpha+1)\alpha \int_0^\infty y^{\alpha-1} e^{-y} dy}}{\beta^\alpha \cdot \Gamma(\alpha)} = \\ &= \frac{(\alpha^2+\alpha) \cdot \beta^2 \cdot \int_0^\infty y^{\alpha-1} e^{-y} dy}{\int_0^\infty x^{\alpha-1} e^{-x} dx} = \left(\alpha^2 + \alpha\right) \cdot \beta^2\end{aligned} \tag{A.3.5}$$

Varianz:

$$\begin{aligned}\left\langle (n - \langle n \rangle)^2 \right\rangle &= \left\langle n^2 - 2n \langle n \rangle + \langle n \rangle^2 \right\rangle = \\ &= \left\langle n^2 \right\rangle - 2 \langle n \rangle \langle n \rangle + \langle n \rangle^2 = \\ &= \left\langle n^2 \right\rangle - \langle n \rangle^2 = \alpha \cdot \beta^2\end{aligned} \tag{A.3.6}$$

Wenn nun Mittelwert und Varianz der Verteilung vorgegeben sind, müssen die Parameter folgendermaßen gewählt werden:

$$\alpha = \frac{\langle n \rangle^2}{\text{Varianz}} \tag{A.3.7}$$

$$\beta = \frac{\text{Varianz}}{\langle n \rangle} \tag{A.3.8}$$

Variationsfaktor:

$$\Phi = \beta \tag{A.3.9}$$

Variationskoeffizient:

$$\Psi = \frac{1}{\sqrt{\alpha}} \tag{A.3.10}$$

A.4. Exponentialverteilung

Eine Exponentialverteilung wird durch einen freien Parameter $\alpha \in \mathbb{R}_+$ festgelegt.

Form:

$$p\left(n = x\right) = \begin{cases} \dfrac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot e^{-\alpha \cdot x} & x \in \mathbb{N} \\ 0 & \text{sonst} \end{cases} \tag{A.4.1}$$

Um die Eigenschaften dieser Verteilung zu berechnen, müssen erst drei unendliche Summen bestimmt werden.

$$\begin{aligned} \sum_{i=0}^{\infty} e^{-\alpha \cdot i} &= 1 + \sum_{i=1}^{\infty} e^{-\alpha \cdot i} \underbrace{=}_{j=i-1} \\ &= 1 + \sum_{j=0}^{\infty} e^{-\alpha \cdot (j+1)} = \\ &= 1 + e^{-\alpha} \cdot \sum_{j=0}^{\infty} e^{-\alpha \cdot j} \\ \Rightarrow \sum_{i=0}^{\infty} e^{-\alpha \cdot i} &= \frac{1}{1 - e^{-\alpha}} \end{aligned} \tag{A.4.2}$$

$$\begin{aligned} \sum_{i=0}^{\infty} i \cdot e^{-\alpha \cdot i} &= \sum_{i=0}^{\infty} (i - 1 + 1) \cdot e^{-\alpha \cdot i} = \\ &= \sum_{i=0}^{\infty} (i-1) \cdot e^{-\alpha \cdot i} + \sum_{i=0}^{\infty} e^{-\alpha \cdot i} = \\ &= -1 + \sum_{i=1}^{\infty} (i-1) \cdot e^{-\alpha \cdot i} + \frac{1}{1 - e^{-\alpha}} \underbrace{=}_{j=i-1} \\ &= \frac{1}{1 - e^{-\alpha}} - 1 + \sum_{j=0}^{\infty} j \cdot e^{-\alpha \cdot (j+1)} = \\ &= \frac{e^{-\alpha}}{1 - e^{-\alpha}} + e^{-\alpha} \sum_{j=0}^{\infty} j \cdot e^{-\alpha \cdot j} \\ \Rightarrow \sum_{i=0}^{\infty} i \cdot e^{-\alpha \cdot i} &= \frac{e^{-\alpha}}{\left(1 - e^{-\alpha}\right)^2} = \sum_{i=1}^{\infty} i \cdot e^{-\alpha \cdot i} \end{aligned} \tag{A.4.3}$$

$$\begin{aligned}
\sum_{i=0}^{\infty} i^2 \cdot e^{-\alpha \cdot i} &= \sum_{i=0}^{\infty} \left(i^2 - 2i + 1 + 2i - 1 \right) \cdot e^{-\alpha \cdot i} = \\
&= \sum_{i=0}^{\infty} (i-1)^2 \cdot e^{-\alpha \cdot i} + 2 \sum_{i=0}^{\infty} i \cdot e^{-\alpha \cdot i} - \sum_{i=0}^{\infty} e^{-\alpha \cdot i} = \\
&= 1 + \sum_{i=1}^{\infty} (i-1)^2 \cdot e^{-\alpha \cdot i} + 2 \frac{e^{-\alpha}}{\left(1 - e^{-\alpha}\right)^2} - \frac{1}{1 - e^{-\alpha}} \underbrace{=}_{j=i-1} \\
&= \frac{2e^{-\alpha} - (1 - e^{-\alpha}) + (1 - e^{-\alpha})^2}{(1 - e^{-\alpha})^2} + \sum_{j=0}^{\infty} j^2 \cdot e^{-\alpha \cdot (j+1)} = \\
&= \frac{e^{-\alpha} + e^{-2\alpha}}{(1 - e^{-\alpha})^2} + e^{-\alpha} \sum_{j=0}^{\infty} j^2 \cdot e^{-\alpha \cdot j} \\
\Rightarrow \sum_{i=0}^{\infty} i^2 \cdot e^{-\alpha \cdot i} &= \frac{e^{-\alpha} + e^{-2\alpha}}{(1 - e^{-\alpha})^3} = \sum_{i=1}^{\infty} i^2 \cdot e^{-\alpha \cdot i}
\end{aligned} \tag{A.4.4}$$

Normierung:

$$\begin{aligned}
\sum_{x=1}^{\infty} p\,(n = x) &= \sum_{x=1}^{\infty} \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot e^{-\alpha \cdot x} \underbrace{=}_{y=x-1} \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot \sum_{y=0}^{\infty} e^{-\alpha \cdot (y+1)} \\
&= \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot e^{-\alpha} \sum_{y=0}^{\infty} e^{-\alpha \cdot y} = 1
\end{aligned} \tag{A.4.5}$$

Mittelwert:

$$\begin{aligned}
\langle n \rangle &= \sum_{x=1}^{\infty} x \cdot p\,(n = x) = \sum_{x=1}^{\infty} x \cdot \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot e^{-\alpha \cdot x} = \\
&= \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot \sum_{x=0}^{\infty} x \cdot e^{-\alpha \cdot x} = \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot \frac{e^{-\alpha}}{(1 - e^{-\alpha})^2} = \frac{1}{1 - e^{-\alpha}}
\end{aligned} \tag{A.4.6}$$

Bei bekanntem Mittelwert μ ist also α gegeben durch:

$$\alpha = -\ln\left(1 - \frac{1}{\mu}\right) \tag{A.4.7}$$

Deswegen kann der Mittelwert keine Werte kleiner gleich 1 annehmen.

Mit Gleichung (A.4.7) kann man die Form umschreiben zu:

$$p\,(n = x) = \begin{cases} \dfrac{1}{\mu - 1} \cdot \left(1 - \dfrac{1}{\mu}\right)^x & x \in \mathbb{N} \\ 0 & \text{sonst} \end{cases} \tag{A.4.8}$$

Zweites Moment:

$$\begin{aligned}\left\langle n^2 \right\rangle &= \sum_{x=1}^{\infty} x^2 \cdot p\left(n = x\right) = \sum_{x=0}^{\infty} x^2 \cdot \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot e^{-\alpha \cdot x} = \\ &= \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot \sum_{x=0}^{\infty} x^2 \cdot e^{-\alpha \cdot x} = \frac{1 - e^{-\alpha}}{e^{-\alpha}} \cdot \frac{e^{-\alpha} + e^{-2\alpha}}{\left(1 - e^{-\alpha}\right)^3} = \frac{1 + e^{-\alpha}}{\left(1 - e^{-\alpha}\right)^2}\end{aligned} \tag{A.4.9}$$

Varianz:

$$\left\langle \left(n - \langle n \rangle\right)^2 \right\rangle = \left\langle n^2 \right\rangle - \langle n \rangle^2 = \frac{e^{-\alpha}}{\left(1 - e^{-\alpha}\right)^2} = \mu \cdot (\mu - 1) \tag{A.4.10}$$

Variationsfaktor:

$$\Phi = \frac{e^{-\alpha}}{1 - e^{-\alpha}} = \mu - 1 \tag{A.4.11}$$

Variationskoeffizient:

$$\Psi = e^{-\frac{\alpha}{2}} = \sqrt{1 - \frac{1}{\mu}} \tag{A.4.12}$$

B. Differentialgleichungssysteme

B.1. Oszillator

$$\frac{d\left[\mathrm{A}_1\right]}{dt} = -k_{\mathrm{TA},21}\left[\mathrm{T}_{21}\right]\left[\mathrm{A}_1\right] - k_{\mathrm{AI},1}\left[\mathrm{A}_1\right]\left[\mathrm{dI}_1\right] + k_{\mathrm{AIrA},1}\left[\mathrm{rA}_1\right]\left[\mathrm{A}_1 \cdot \mathrm{dI}_1\right] \tag{B.1.1}$$

$$\frac{d\left[\mathrm{A}_1 \cdot \mathrm{dI}_1\right]}{dt} = +k_{\mathrm{AI},1}\left[\mathrm{A}_1\right]\left[\mathrm{dI}_1\right] + k_{\mathrm{TAI},21}\left[\mathrm{T}_{21} \cdot \mathrm{A}_1\right]\left[\mathrm{dI}_1\right] - k_{\mathrm{AIrA},1}\left[\mathrm{rA}_1\right]\left[\mathrm{A}_1 \cdot \mathrm{dI}_1\right] \tag{B.1.2}$$

$$\frac{d\left[\mathrm{A}_2\right]}{dt} = -k_{\mathrm{TA},12}\left[\mathrm{T}_{12}\right]\left[\mathrm{A}_2\right] - k_{\mathrm{AI},2}\left[\mathrm{A}_2\right]\left[\mathrm{rI}_2\right] - k_{\mathrm{s},+}\left[\mathrm{A}_2\right]\left[\mathrm{sI}_2\right] + k_{\mathrm{s},-}\left[\mathrm{A}_2 \cdot \mathrm{sI}_2\right] \tag{B.1.3}$$

$$\begin{aligned}\frac{d\left[\mathrm{A}_2 \cdot \mathrm{rI}_2\right]}{dt} = {} & +k_{\mathrm{AI},2}\left[\mathrm{A}_2\right]\left[\mathrm{rI}_2\right] + k_{\mathrm{TAI},12}\left[\mathrm{T}_{12} \cdot \mathrm{A}_2\right]\left[\mathrm{rI}_2\right] \\ & + k_{\mathrm{AI},2}\left[\mathrm{rI}_2\right]\left[\mathrm{A}_2 \cdot \mathrm{sI}_2\right] - k_{+,\mathrm{H}}\left[\mathrm{RNaseH}\right]\left[\mathrm{A}_2 \cdot \mathrm{rI}_2\right] \\ & + k_{-,\mathrm{H},2}\left[\mathrm{RNaseH} \cdot \mathrm{A}_2 \cdot \mathrm{rI}_2\right]\end{aligned} \tag{B.1.4}$$

$$\begin{aligned}\frac{d\left[\mathrm{A}_2 \cdot \mathrm{sI}_2\right]}{dt} = {} & +k_{\mathrm{s},+}\left[\mathrm{A}_2\right]\left[\mathrm{sI}_2\right] - k_{\mathrm{s},-}\left[\mathrm{A}_2 \cdot \mathrm{sI}_2\right] \\ & - k_{\mathrm{TA},12}\left[\mathrm{T}_{12}\right]\left[\mathrm{A}_2 \cdot \mathrm{sI}_2\right] - k_{\mathrm{AI},2}\left[\mathrm{rI}_2\right]\left[\mathrm{A}_2 \cdot \mathrm{sI}_2\right] \\ & + k_{\mathrm{c,H},2}\left[\mathrm{RNaseH} \cdot \mathrm{A}_2 \cdot \mathrm{rI}_2\right]\end{aligned} \tag{B.1.5}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\right]}{dt} = {} & -k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12} \cdot \mathrm{A}_2\right] + k_{-,\mathrm{an},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12} \cdot \mathrm{A}_2\right] \\ & + k_{\mathrm{c,an},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12} \cdot \mathrm{A}_2\right] - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\right] \\ & + k_{-,\mathrm{aus},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12}\right] + k_{\mathrm{c,aus},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12}\right] \\ & - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{21} \cdot \mathrm{A}_1\right] + k_{-,\mathrm{an},21}\left[\mathrm{RNAP} \cdot \mathrm{T}_{21} \cdot \mathrm{A}_1\right] \\ & + k_{\mathrm{c,an},21}\left[\mathrm{RNAP} \cdot \mathrm{T}_{21} \cdot \mathrm{A}_1\right] - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{21}\right] \\ & + k_{-,\mathrm{aus},21}\left[\mathrm{RNAP} \cdot \mathrm{T}_{21}\right] + k_{\mathrm{c,aus},21}\left[\mathrm{RNAP} \cdot \mathrm{T}_{21}\right] \\ & - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12} \cdot \mathrm{A}_2 \cdot \mathrm{sI}_2\right] \\ & + k_{-,\mathrm{an},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12} \cdot \mathrm{A}_2 \cdot \mathrm{sI}_2\right] \\ & + k_{\mathrm{c,an},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12} \cdot \mathrm{A}_2 \cdot \mathrm{sI}_2\right]\end{aligned} \tag{B.1.6}$$

$$\frac{d\left[\mathrm{RNAP} \cdot \mathrm{T}_{12}\right]}{dt} = +k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\right] - k_{-,\mathrm{aus},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12}\right] - k_{\mathrm{c,aus},12}\left[\mathrm{RNAP} \cdot \mathrm{T}_{12}\right] \tag{B.1.7}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\right]}{dt} &= +k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\\ &\quad - k_{-,\mathrm{an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\\ &\quad - k_{\mathrm{c},\mathrm{an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\end{aligned} \tag{B.1.8}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]}{dt} &= +k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]\\ &\quad - k_{-,\mathrm{an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]\\ &\quad - k_{\mathrm{c},\mathrm{an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]\end{aligned} \tag{B.1.9}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\right]}{dt} &= +k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{21}\right] - k_{-,\mathrm{aus},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\right]\\ &\quad - k_{\mathrm{c},\mathrm{aus},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\right]\end{aligned} \tag{B.1.10}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\cdot\mathrm{A}_1\right]}{dt} &= +k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\\ &\quad - k_{-,\mathrm{an},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\\ &\quad - k_{\mathrm{c},\mathrm{an},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\end{aligned} \tag{B.1.11}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNaseH}\right]}{dt} &= -k_{+,\mathrm{H}}\left[\mathrm{RNaseH}\right]\left[\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\\ &\quad + k_{-,\mathrm{H},1}\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\\ &\quad + k_{\mathrm{c},\mathrm{H},1}\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\\ &\quad - k_{+,\mathrm{H}}\left[\mathrm{RNaseH}\right]\left[\mathrm{A}_2\cdot\mathrm{rI}_2\right]\\ &\quad + k_{-,\mathrm{H},2}\left[\mathrm{RNaseH}\cdot\mathrm{A}_2\cdot\mathrm{rI}_2\right]\\ &\quad + k_{\mathrm{c},\mathrm{H},2}\left[\mathrm{RNaseH}\cdot\mathrm{A}_2\cdot\mathrm{rI}_2\right]\end{aligned} \tag{B.1.12}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNaseH}\cdot\mathrm{A}_2\cdot\mathrm{rI}_2\right]}{dt} &= +k_{+,\mathrm{H}}\left[\mathrm{RNaseH}\right]\left[\mathrm{A}_2\cdot\mathrm{rI}_2\right]\\ &\quad - k_{-,\mathrm{H},2}\left[\mathrm{RNaseH}\cdot\mathrm{A}_2\cdot\mathrm{rI}_2\right]\\ &\quad - k_{\mathrm{c},\mathrm{H},2}\left[\mathrm{RNaseH}\cdot\mathrm{A}_2\cdot\mathrm{rI}_2\right]\end{aligned} \tag{B.1.13}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]}{dt} &= +k_{+,\mathrm{H}}\left[\mathrm{RNaseH}\right]\left[\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\\ &\quad - k_{-,\mathrm{H},1}\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\\ &\quad - k_{\mathrm{c},\mathrm{H},1}\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\end{aligned} \tag{B.1.14}$$

$$\begin{aligned}\frac{d\left[\mathrm{T}_{12}\right]}{dt} &= -k_{\mathrm{TA},12}\left[\mathrm{T}_{12}\right]\left[\mathrm{A}_2\right] + k_{\mathrm{TAI},12}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\left[\mathrm{rI}_2\right]\\ &\quad - k_{\mathrm{TA},12}\left[\mathrm{T}_{12}\right]\left[\mathrm{A}_2\cdot\mathrm{sI}_2\right] - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\right]\\ &\quad + k_{-,\mathrm{aus},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\right] + k_{\mathrm{c},\mathrm{aus},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\right]\end{aligned} \tag{B.1.15}$$

$$\begin{aligned}\frac{d\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]}{dt} &= +k_{\mathrm{TA},12}\left[\mathrm{T}_{12}\right]\left[\mathrm{A}_2\right] - k_{\mathrm{TAI},12}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\left[\mathrm{rI}_2\right] \\ &\quad - k_{\mathrm{s},+}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\left[\mathrm{sI}_2\right] + k_{\mathrm{s},-}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right] \\ &\quad - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right] + k_{-,\mathrm{an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\right] \\ &\quad + k_{\mathrm{c,an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\end{aligned} \tag{B.1.16}$$

$$\begin{aligned}\frac{d\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]}{dt} &= +k_{\mathrm{TA},12}\left[\mathrm{T}_{12}\right]\left[\mathrm{A}_2\cdot\mathrm{sI}_2\right] + k_{\mathrm{s},+}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\left[\mathrm{sI}_2\right] \\ &\quad - k_{\mathrm{s},-}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right] - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right] \\ &\quad + k_{-,\mathrm{an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right] \\ &\quad + k_{\mathrm{c,an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]\end{aligned} \tag{B.1.17}$$

$$\begin{aligned}\frac{d\left[\mathrm{T}_{21}\right]}{dt} &= -k_{\mathrm{TA},21}\left[\mathrm{T}_{21}\right]\left[\mathrm{A}_1\right] + k_{\mathrm{TAI},21}\left[\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\left[\mathrm{dI}_1\right] \\ &\quad - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{21}\right] + k_{-,\mathrm{aus},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\right] \\ &\quad + k_{\mathrm{c,aus},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\right]\end{aligned} \tag{B.1.18}$$

$$\begin{aligned}\frac{d\left[\mathrm{T}_{21}\cdot\mathrm{A}_1\right]}{dt} &= +k_{\mathrm{TA},21}\left[\mathrm{T}_{21}\right]\left[\mathrm{A}_1\right] - k_{\mathrm{TAI},21}\left[\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\left[\mathrm{dI}_1\right] \\ &\quad - k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{T}_{21}\cdot\mathrm{A}_1\right] + k_{-,\mathrm{an},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\cdot\mathrm{A}_1\right] \\ &\quad + k_{\mathrm{c,an},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\end{aligned} \tag{B.1.19}$$

$$\begin{aligned}\frac{d\left[\mathrm{dI}_1\right]}{dt} &= -k_{\mathrm{AI},1}\left[\mathrm{A}_1\right]\left[\mathrm{dI}_1\right] - k_{\mathrm{rAI},1}\left[\mathrm{rA}_1\right]\left[\mathrm{dI}_1\right] \\ &\quad - k_{\mathrm{TAI},21}\left[\mathrm{T}_{21}\cdot\mathrm{A}_1\right]\left[\mathrm{dI}_1\right] + k_{\mathrm{c,H},1}\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\end{aligned} \tag{B.1.20}$$

$$\begin{aligned}\frac{d\left[\mathrm{rA}_1\right]}{dt} &= -k_{\mathrm{rAI},1}\left[\mathrm{rA}_1\right]\left[\mathrm{dI}_1\right] - k_{\mathrm{AIrA},1}\left[\mathrm{rA}_1\right]\left[\mathrm{A}_1\cdot\mathrm{dI}_1\right] \\ &\quad + k_{\mathrm{c,an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\right] + k_{\mathrm{c,aus},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\right] \\ &\quad + k_{\mathrm{c,an},12}\left[\mathrm{RNAP}\cdot\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right]\end{aligned} \tag{B.1.21}$$

$$\begin{aligned}\frac{d\left[\mathrm{rA}_1\cdot\mathrm{dI}_1\right]}{dt} &= +k_{\mathrm{rAI},1}\left[\mathrm{rA}_1\right]\left[\mathrm{dI}_1\right] + k_{\mathrm{AIrA},1}\left[\mathrm{rA}_1\right]\left[\mathrm{A}_1\cdot\mathrm{dI}_1\right] \\ &\quad - k_{+,\mathrm{H}}\left[\mathrm{RNaseH}\right]\left[\mathrm{rA}_1\cdot\mathrm{dI}_1\right] \\ &\quad + k_{-,\mathrm{H},1}\left[\mathrm{RNaseH}\cdot\mathrm{rA}_1\cdot\mathrm{dI}_1\right]\end{aligned} \tag{B.1.22}$$

$$\begin{aligned}\frac{d\left[\mathrm{rI}_2\right]}{dt} &= -k_{\mathrm{AI},2}\left[\mathrm{A}_2\right]\left[\mathrm{rI}_2\right] - k_{\mathrm{TAI},12}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\left[\mathrm{rI}_2\right] \\ &\quad - k_{\mathrm{AI},2}\left[\mathrm{rI}_2\right]\left[\mathrm{A}_2\cdot\mathrm{sI}_2\right] + k_{\mathrm{c,an},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\cdot\mathrm{A}_1\right] \\ &\quad + k_{\mathrm{c,aus},21}\left[\mathrm{RNAP}\cdot\mathrm{T}_{21}\right]\end{aligned} \tag{B.1.23}$$

$$\begin{aligned}\frac{d\left[\mathrm{sI}_2\right]}{dt} &= -k_{\mathrm{s},+}\left[\mathrm{A}_2\right]\left[\mathrm{sI}_2\right] + k_{\mathrm{s},-}\left[\mathrm{A}_2\cdot\mathrm{sI}_2\right] \\ &\quad - k_{\mathrm{s},+}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\right]\left[\mathrm{sI}_2\right] + k_{\mathrm{s},-}\left[\mathrm{T}_{12}\cdot\mathrm{A}_2\cdot\mathrm{sI}_2\right] \\ &\quad + k_{\mathrm{AI},2}\left[\mathrm{rI}_2\right]\left[\mathrm{A}_2\cdot\mathrm{sI}_2\right]\end{aligned} \tag{B.1.24}$$

B.1.1. Erhaltungsgrößen

$$\begin{aligned}\left[\mathrm{A_1^{tot}}\right] &= [\mathrm{A_1}] + [\mathrm{T_{21}} \cdot \mathrm{A_1}] + [\mathrm{A_1} \cdot \mathrm{dI_1}] + [\mathrm{rA_1}] + [\mathrm{rA_1} \cdot \mathrm{dI_1}] \\ &\quad + [\mathrm{RNAP} \cdot \mathrm{T_{21}} \cdot \mathrm{A_1}] + [\mathrm{RNaseH} \cdot \mathrm{rA_1} \cdot \mathrm{dI_1}]\end{aligned} \tag{B.1.25}$$

$$\begin{aligned}\left[\mathrm{A_2^{tot}}\right] &= [\mathrm{A_2}] + [\mathrm{T_{12}} \cdot \mathrm{A_2}] + [\mathrm{A_2} \cdot \mathrm{rI_2}] + [\mathrm{A_2} \cdot \mathrm{sI_2}] + [\mathrm{T_{12}} \cdot \mathrm{A_2} \cdot \mathrm{sI_2}] \\ &\quad + [\mathrm{RNAP} \cdot \mathrm{T_{12}} \cdot \mathrm{A_2}] + [\mathrm{RNaseH} \cdot \mathrm{A_2} \cdot \mathrm{rI_2}] \\ &\quad + [\mathrm{RNAP} \cdot \mathrm{T_{12}} \cdot \mathrm{A_2} \cdot \mathrm{sI_2}]\end{aligned} \tag{B.1.26}$$

$$\begin{aligned}\left[\mathrm{RNAP^{tot}}\right] &= [\mathrm{RNAP}] + [\mathrm{RNAP} \cdot \mathrm{T_{12}} \cdot \mathrm{A_2}] + [\mathrm{RNAP} \cdot \mathrm{T_{12}}] \\ &\quad + [\mathrm{RNAP} \cdot \mathrm{T_{21}} \cdot \mathrm{A_1}] + [\mathrm{RNAP} \cdot \mathrm{T_{21}}] \\ &\quad + [\mathrm{RNAP} \cdot \mathrm{T_{12}} \cdot \mathrm{A_2} \cdot \mathrm{sI_2}]\end{aligned} \tag{B.1.27}$$

$$\left[\mathrm{RNaseH^{tot}}\right] = [\mathrm{RNaseH}] + [\mathrm{RNaseH} \cdot \mathrm{rA_1} \cdot \mathrm{dI_1}] + [\mathrm{RNaseH} \cdot \mathrm{A_2} \cdot \mathrm{rI_2}] \tag{B.1.28}$$

$$\begin{aligned}\left[\mathrm{T_{12}^{tot}}\right] &= [\mathrm{T_{12}}] + [\mathrm{T_{12}} \cdot \mathrm{A_2}] + [\mathrm{T_{12}} \cdot \mathrm{A_2} \cdot \mathrm{sI_2}] + [\mathrm{RNAP} \cdot \mathrm{T_{12}} \cdot \mathrm{A_2}] \\ &\quad + [\mathrm{RNAP} \cdot \mathrm{T_{12}}] + [\mathrm{RNAP} \cdot \mathrm{T_{12}} \cdot \mathrm{A_2} \cdot \mathrm{sI_2}]\end{aligned} \tag{B.1.29}$$

$$\left[\mathrm{T_{21}^{tot}}\right] = [\mathrm{T_{21}}] + [\mathrm{T_{21}} \cdot \mathrm{A_1}] + [\mathrm{RNAP} \cdot \mathrm{T_{21}} \cdot \mathrm{A_1}] + [\mathrm{RNAP} \cdot \mathrm{T_{21}}] \tag{B.1.30}$$

$$\left[\mathrm{dI_1^{tot}}\right] = [\mathrm{dI_1}] + [\mathrm{A_1} \cdot \mathrm{dI_1}] + [\mathrm{rA_1} \cdot \mathrm{dI_1}] + [\mathrm{RNaseH} \cdot \mathrm{rA_1} \cdot \mathrm{dI_1}] \tag{B.1.31}$$

B.2. Stochastiksystem

B.2.1. Reaktionsmodell

$$\begin{aligned}\frac{d\,[\mathrm{RNA}]}{dt} &= +k_{\mathrm{cat}}\,[\mathrm{RNAP} \cdot \mathrm{Templat}] \\ &\quad - k_{\mathrm{activate}}\,[\mathrm{RNA}]\,[\mathrm{Reporter}] \\ &\quad - k_{\mathrm{waste}}\,[\mathrm{RNA}]\,[\mathrm{Inhibitor}]\end{aligned} \tag{B.2.1}$$

$$\begin{aligned}\frac{d\,[\mathrm{RNA} \cdot \mathrm{Reporter}]}{dt} &= +k_{\mathrm{activate}}\,[\mathrm{RNA}]\,[\mathrm{Reporter}] \\ &\quad - k_{\mathrm{heal}}\,[\mathrm{RNA} \cdot \mathrm{Reporter}]\,[\mathrm{Inhibitor}]\end{aligned} \tag{B.2.2}$$

$$\begin{aligned}\frac{d\,[\mathrm{RNA} \cdot \mathrm{Inhibitor}]}{dt} &= +k_{\mathrm{waste}}\,[\mathrm{RNA}]\,[\mathrm{Inhibitor}] \\ &\quad + k_{\mathrm{heal}}\,[\mathrm{RNA} \cdot \mathrm{Reporter}]\,[\mathrm{Inhibitor}]\end{aligned} \tag{B.2.3}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNA}^{\mathrm{tot}}\right]}{dt} &= +\frac{d\left[\mathrm{RNA}\right]}{dt} \\ &\quad + \frac{d\left[\mathrm{RNA}\cdot\mathrm{Reporter}\right]}{dt} \\ &\quad + \frac{d\left[\mathrm{RNA}\cdot\mathrm{Inhibitor}\right]}{dt} = \\ &= k_{\mathrm{cat}}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right]\end{aligned} \tag{B.2.4}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\right]}{dt} &= -k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{Templat}\right] \\ &\quad + k_{-}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right] \\ &\quad + k_{\mathrm{cat}}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right]\end{aligned} \tag{B.2.5}$$

$$\begin{aligned}\frac{d\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right]}{dt} &= +k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{Templat}\right] \\ &\quad - k_{-}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right] \\ &\quad - k_{\mathrm{cat}}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right]\end{aligned} \tag{B.2.6}$$

$$\begin{aligned}\frac{d\left[\mathrm{Reporter}\right]}{dt} &= -k_{\mathrm{activate}}\left[\mathrm{RNA}\right]\left[\mathrm{Reporter}\right] \\ &\quad + k_{\mathrm{heal}}\left[\mathrm{RNA}\cdot\mathrm{Reporter}\right]\left[\mathrm{Inhibitor}\right]\end{aligned} \tag{B.2.7}$$

$$\begin{aligned}\frac{d\left[\mathrm{Templat}\right]}{dt} &= -k_{+}\left[\mathrm{RNAP}\right]\left[\mathrm{Templat}\right] \\ &\quad + k_{-}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right] \\ &\quad + k_{\mathrm{cat}}\left[\mathrm{RNAP}\cdot\mathrm{Templat}\right]\end{aligned} \tag{B.2.8}$$

$$\begin{aligned}\frac{d\left[\mathrm{Inhibitor}\right]}{dt} &= -k_{\mathrm{waste}}\left[\mathrm{RNA}\right]\left[\mathrm{Inhibitor}\right] \\ &\quad - k_{\mathrm{heal}}\left[\mathrm{RNA}\cdot\mathrm{Reporter}\right]\left[\mathrm{Inhibitor}\right]\end{aligned} \tag{B.2.9}$$

B.2.2. Erhaltungsgrößen

$$\left[\mathrm{RNAP}^{\mathrm{tot}}\right] = \left[\mathrm{RNAP}\right] + \left[\mathrm{RNAP}\cdot\mathrm{Templat}\right] \tag{B.2.10}$$

$$\left[\mathrm{Reporter}^{\mathrm{tot}}\right] = \left[\mathrm{Reporter}\right] + \left[\mathrm{RNA}\cdot\mathrm{Reporter}\right] \tag{B.2.11}$$

$$\left[\mathrm{Templat}^{\mathrm{tot}}\right] = \left[\mathrm{Templat}\right] + \left[\mathrm{RNAP}\cdot\mathrm{Templat}\right] \tag{B.2.12}$$

$$\left[\mathrm{Inhibitor}^{\mathrm{tot}}\right] = \left[\mathrm{Inhibitor}\right] + \left[\mathrm{RNA}\cdot\mathrm{Inhibitor}\right] \tag{B.2.13}$$

B.2.3. Vereinfachtes Modell

Sei $r(t)$ die Konzentration an freier RNA, $y(t)$ die Konzentration der inaktiven Reporter und $x(t)$ die Konzentration der aktiven Reporter. Die Gesamtkonzentration an Reportern ist konstant

$$y(t) + x(t) = R_{\text{tot}} = \text{konstant} \tag{B.2.14}$$

und $r+x$ ist die Gesamtkonzentration an RNA, die durch die konstante RNA-Produktion beschrieben wird:

$$r(t) + x(t) = k_{\text{prod}} \cdot t \tag{B.2.15}$$

Die Aktivierung des Reporters ist dann gegeben durch:

$$\begin{aligned}
\frac{d}{dt}y(t) &= -\frac{d}{dt}x(t) \\
&= -k_{\text{activate}} \cdot y(t) \cdot r(t) \\
&= -k_{\text{activate}} \cdot y(t) \cdot \Big(k_{\text{prod}} \cdot t - x(t)\Big) \\
&= -k_{\text{activate}} \cdot y(t) \cdot \Big(k_{\text{prod}} \cdot t - (R_{\text{tot}} - y(t))\Big) \\
&= -k_{\text{activate}} \cdot k_{\text{prod}} \cdot y(t) \cdot t + k_{\text{activate}} \cdot R_{\text{tot}} \cdot y(t) - k_{\text{activate}} \cdot y(t)^2
\end{aligned} \tag{B.2.16}$$

Mit den Rahmenbedingungen

$$r(t=0) = 0 \tag{B.2.17}$$

$$x(t=0) = 0 \tag{B.2.18}$$

$$y(t=0) = R_{\text{tot}} \tag{B.2.19}$$

ergibt sich die Lösung (aus Platzgründen wurde k_{activate} durch k_{a} ersetzt)[1]:

$$x(t) = R_{\text{tot}} - \frac{2 \cdot e^{k_{\text{a}} \cdot \left(-\left(\frac{k_{\text{prod}} \cdot t^2}{2}\right) + t \cdot R_{\text{tot}}\right)} \cdot R_{\text{tot}}}{2 + \sqrt{k_{\text{a}} \cdot 2 \cdot \frac{\pi}{k_{\text{prod}}}} \cdot R_{\text{tot}} \cdot e^{\frac{k_{\text{a}} \cdot R_{\text{tot}}^2}{2 \cdot k_{\text{prod}}}} \cdot \left[\text{erf}\left(\frac{\sqrt{k_{\text{a}}} \cdot \left(k_{\text{prod}} \cdot t - R_{\text{tot}}\right)}{\sqrt{2 \cdot k_{\text{prod}}}}\right) + \text{erf}\left(\frac{\sqrt{k_{\text{a}}} \cdot R_{\text{tot}}}{\sqrt{2 \cdot k_{\text{prod}}}}\right)\right]} \tag{B.2.20}$$

Hierbei ist $\text{erf}(x)$ die Fehlerfunktion:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\tau^2} d\tau \tag{B.2.21}$$

[1] Die Differentialgleichung wurde mit Mathematica 10.0 über den Befehl DSolve[x[0] == RTot, x'[t] == -kActivate*kProd*t*x[t] + kActivate*RTot*x[t] - kActivate*x[t]*x[t], x, t] gelöst.

B.3. Aggregation

Das einfachste System der Aggregation ist das, bei dem die Wachstums- und Schrumpfungsrate in einer festen Potenz von der Größe des Aggregats abhängig ist:

$$\mathrm{A_n} + \mathrm{A_1} \underset{n^d \cdot k_{\mathrm{off}}}{\overset{n^d \cdot k_{\mathrm{on}}}{\rightleftharpoons}} \mathrm{A_{n+1}} \quad \forall n \in \mathbb{N}$$

Diese allgemeine Formel kann man für D-dimensionale Aggregation näherungsweise verwenden, indem man $d = \frac{D-1}{D}$ setzt.

Dadurch ergibt sich das folgende Differentialgleichungssystem:

$$\frac{d}{dt}\left[\mathrm{A_1}\right] = k_{\mathrm{off}} \cdot \sum_{i=2}^{\infty} (i-1)^d \cdot \left[\mathrm{A_i}\right] - k_{\mathrm{on}}\left[\mathrm{A_1}\right] \cdot \sum_{i=1}^{\infty} i^d \left[\mathrm{A_i}\right] \tag{B.3.1}$$

$$\begin{aligned}\frac{d}{dt}\left[\mathrm{A_n}\right] = {} & k_{\mathrm{off}} \cdot \left(n^N \cdot \left[\mathrm{A_{n+1}}\right] - (n-1)^d \cdot \left[\mathrm{A_n}\right]\right) \\ & - k_{\mathrm{on}}\left[\mathrm{A_1}\right] \cdot \left(n^d \cdot \left[\mathrm{A_n}\right] - (n-1)^d \cdot \left[\mathrm{A_{n-1}}\right]\right)\end{aligned} \tag{B.3.2}$$

Die Anzahl der Monomere bildet dabei eine Erhaltungsgröße:

$$\sum_{i=1}^{\infty} i \left[\mathrm{A_i}\right] = N \tag{B.3.3}$$

Der Gleichgewichtszustand wird dadurch definiert, dass die Ableitungen aller Konzentrationen nach der Zeit gleich Null sind:

$$\frac{d}{dt}\left[\mathrm{A_i}\right] = 0 \quad \forall i \in \mathbb{N} \tag{B.3.4}$$

Als Lösung des Gleichungssystems nehmen wir an, dass die Konzentrationen einer Exponentialfunktion folgen:

$$\left[\mathrm{A_i}\right] = A_0 \cdot e^{-\alpha \cdot i} \tag{B.3.5}$$

Dann ergeben sich folgende Rechenregeln:

$$\left[\mathrm{A_1}\right] \cdot \left[\mathrm{A_n}\right] = A_0 \cdot e^{-\alpha} \cdot A_0 \cdot e^{-\alpha \cdot n} = A_0^2 \cdot e^{-\alpha \cdot (n+1)} = A_0 \cdot \left[\mathrm{A_{n+1}}\right] \tag{B.3.6}$$

$$\begin{aligned}\sum_{i=\beta}^{\infty} \left[\mathrm{A_i}\right] = \sum_{i=\beta}^{\infty} A_0 \cdot e^{-\alpha \cdot i} &= A_0 \cdot e^{-\alpha \cdot \beta} \cdot \sum_{i=\beta}^{\infty} e^{-\alpha \cdot (i-\beta)} \underbrace{=}_{j=i-\beta} \\ &= A_0 \cdot e^{-\alpha \cdot \beta} \cdot \sum_{j=0}^{\infty} e^{-\alpha \cdot j} \underbrace{=}_{\text{(A.4.2)}} \frac{A_0 \cdot e^{-\alpha \cdot \beta}}{1 - e^{-\alpha}}\end{aligned} \tag{B.3.7}$$

Aus der Gleichung für A_1 ergibt sich damit:

$$\begin{aligned}
\frac{d}{dt}\left[\mathrm{A}_1\right] = 0 &= k_{\mathrm{off}} \cdot \sum_{i=2}^{\infty} (i-1)^d \left[\mathrm{A_i}\right] - k_{\mathrm{on}} \left[\mathrm{A}_1\right] \cdot \sum_{i=1}^{\infty} i^d \left[\mathrm{A_i}\right] = \\
&= k_{\mathrm{off}} \cdot \sum_{i=2}^{\infty} (i-1)^d \left[\mathrm{A_i}\right] - k_{\mathrm{on}} \cdot A_0 \cdot \sum_{i=1}^{\infty} i^d \left[\mathrm{A_{i+1}}\right] \underbrace{=}_{j=i+1} \\
&= k_{\mathrm{off}} \cdot \sum_{i=2}^{\infty} (i-1)^d \left[\mathrm{A_i}\right] - k_{\mathrm{on}} \cdot A_0 \cdot \sum_{j=2}^{\infty} (j-1)^d \left[\mathrm{A_j}\right] = \\
&= (k_{\mathrm{off}} - k_{\mathrm{on}} \cdot A_0) \cdot \sum_{j=2}^{\infty} (j-1)^d \left[\mathrm{A_j}\right] \\
\Rightarrow A_0 &= \frac{k_{\mathrm{off}}}{k_{\mathrm{on}}}
\end{aligned} \tag{B.3.8}$$

Wenn man diese Relation in die Gleichungen für $[\mathrm{A_n}]$ einsetzt, werden diese auch gelöst:

$$\begin{aligned}
\frac{d}{dt}\left[\mathrm{A_n}\right] &= k_{\mathrm{off}} \cdot \left(n^d \cdot \left[\mathrm{A_{n+1}}\right] - (n-1)^d \cdot \left[\mathrm{A_n}\right]\right) \\
&\quad - k_{\mathrm{on}} \left[\mathrm{A}_1\right] \cdot \left(n^d \cdot \left[\mathrm{A_n}\right] - (n-1)^d \cdot \left[\mathrm{A_{n-1}}\right]\right) = \\
&= k_{\mathrm{off}} \cdot \left(n^d \cdot \left[\mathrm{A_{n+1}}\right] - (n-1)^d \cdot \left[\mathrm{A_n}\right]\right) \\
&\quad - k_{\mathrm{on}} \cdot A_0 \cdot \left(n^d \cdot \left[\mathrm{A_{n+1}}\right] - (n-1)^d \cdot \left[\mathrm{A_n}\right]\right) = \\
&= \left(n^d \cdot \left[\mathrm{A_{n+1}}\right] - (n-1)^d \cdot \left[\mathrm{A_n}\right]\right) \cdot (k_{\mathrm{off}} - k_{\mathrm{on}} \cdot A_0) = \\
&= \left(n^d \cdot \left[\mathrm{A_{n+1}}\right] - (n-1)^d \cdot \left[\mathrm{A_n}\right]\right) \cdot \left(k_{\mathrm{off}} - k_{\mathrm{on}} \cdot \frac{k_{\mathrm{off}}}{k_{\mathrm{on}}}\right) = 0
\end{aligned} \tag{B.3.9}$$

Somit ergibt sich der übrige Parameter α aus der Erhaltungsgröße N:

$$N = \sum_{i=1}^{\infty} i \left[\mathrm{A_i}\right] = \sum_{i=0}^{\infty} i \cdot A_0 \cdot e^{-\alpha \cdot i} \underbrace{=}_{(\mathrm{A.4.3})} A_0 \cdot \frac{e^{-\alpha}}{\left(1 - e^{-\alpha}\right)^2} \tag{B.3.10}$$

$$\left(1 - e^{-\alpha}\right)^2 = \frac{A_0}{N} \cdot e^{-\alpha} \tag{B.3.11}$$

$$1 - 2e^{-\alpha} + e^{-2\alpha} = \frac{A_0}{N} \cdot e^{-\alpha} \tag{B.3.12}$$

$$e^{\alpha} + e^{-\alpha} = \frac{A_0}{N} + 2 \tag{B.3.13}$$

$$\cosh \alpha = \frac{A_0}{2N} + 1 \tag{B.3.14}$$

$$\alpha = \cosh^{-1}\left(\frac{A_0}{2N} + 1\right) = \cosh^{-1}\left(\frac{k_{\mathrm{off}}}{2k_{\mathrm{on}}N} + 1\right) \tag{B.3.15}$$

B.4. Chemisches Gleichgewicht

Zwei chemische Spezies können sich zu einem Komplex verbinden und dissoziieren:

$$\mathrm{A} + \mathrm{B} \underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} \mathrm{A} \cdot \mathrm{B} \tag{B.4.1}$$

Dadurch ergibt sich folgendes Differentialgleichungssystem:

$$\frac{d\,[\mathrm{A}]}{dt} = -k_{\mathrm{on}}\,[\mathrm{A}]\,[\mathrm{B}] + k_{\mathrm{off}}\,[\mathrm{A} \cdot \mathrm{B}] \tag{B.4.2}$$

$$\frac{d\,[\mathrm{A} \cdot \mathrm{B}]}{dt} = +k_{\mathrm{on}}\,[\mathrm{A}]\,[\mathrm{B}] - k_{\mathrm{off}}\,[\mathrm{A} \cdot \mathrm{B}] \tag{B.4.3}$$

$$\frac{d\,[\mathrm{B}]}{dt} = -k_{\mathrm{on}}\,[\mathrm{A}]\,[\mathrm{B}] + k_{\mathrm{off}}\,[\mathrm{A} \cdot \mathrm{B}] \tag{B.4.4}$$

Dabei sind die Gesamtkonzentrationen gegeben durch:

$$\left[\mathrm{A}^{\mathrm{tot}}\right] = [\mathrm{A}] + [\mathrm{A} \cdot \mathrm{B}] \tag{B.4.5}$$

$$\left[\mathrm{B}^{\mathrm{tot}}\right] = [\mathrm{B}] + [\mathrm{A} \cdot \mathrm{B}] \tag{B.4.6}$$

Das Gleichgewicht stellt sich dann dar durch:

$$\left.\begin{aligned} \frac{d\,[\mathrm{A}]}{dt} &= 0 \\ \frac{d\,[\mathrm{A} \cdot \mathrm{B}]}{dt} &= 0 \\ \frac{d\,[\mathrm{B}]}{dt} &= 0 \end{aligned}\right\} \Rightarrow \frac{[\mathrm{A}]\,[\mathrm{B}]}{[\mathrm{A} \cdot \mathrm{B}]} = \frac{k_{\mathrm{off}}}{k_{\mathrm{on}}} = K_d \tag{B.4.7}$$

Dieses quadratische Gleichungssystem lässt sich auflösen zu:

$$\begin{aligned} [\mathrm{A}] = \frac{1}{2}\Big(&\left[\mathrm{A}^{\mathrm{tot}}\right] - K_d - \left[\mathrm{B}^{\mathrm{tot}}\right] \\ &+\sqrt{\left(\left[\mathrm{B}^{\mathrm{tot}}\right] + K_d - \left[\mathrm{A}^{\mathrm{tot}}\right]\right)^2 + 4\left[\mathrm{A}^{\mathrm{tot}}\right] K_d}\Big) \end{aligned} \tag{B.4.8}$$

$$\begin{aligned} [\mathrm{A} \cdot \mathrm{B}] &= \left[\mathrm{A}^{\mathrm{tot}}\right] - [\mathrm{A}] \\ &= \frac{1}{2}\Big(\left[\mathrm{A}^{\mathrm{tot}}\right] + K_d + \left[\mathrm{B}^{\mathrm{tot}}\right] \\ &\quad -\sqrt{\left(\left[\mathrm{B}^{\mathrm{tot}}\right] + K_d - \left[\mathrm{A}^{\mathrm{tot}}\right]\right)^2 + 4\left[\mathrm{A}^{\mathrm{tot}}\right] K_d}\Big) \end{aligned} \tag{B.4.9}$$

$$\begin{aligned} [\mathrm{B}] = \left[\mathrm{B}^{\mathrm{tot}}\right] - [\mathrm{A} \cdot \mathrm{B}] \\ = \frac{1}{2}\Big(&\left[\mathrm{B}^{\mathrm{tot}}\right] - K_d - \left[\mathrm{A}^{\mathrm{tot}}\right] \\ &+\sqrt{\left(\left[\mathrm{B}^{\mathrm{tot}}\right] + K_d - \left[\mathrm{A}^{\mathrm{tot}}\right]\right)^2 + 4\left[\mathrm{A}^{\mathrm{tot}}\right] K_d}\Big) \end{aligned} \tag{B.4.10}$$

C. Ergebnisse der Oszillatorsimulation

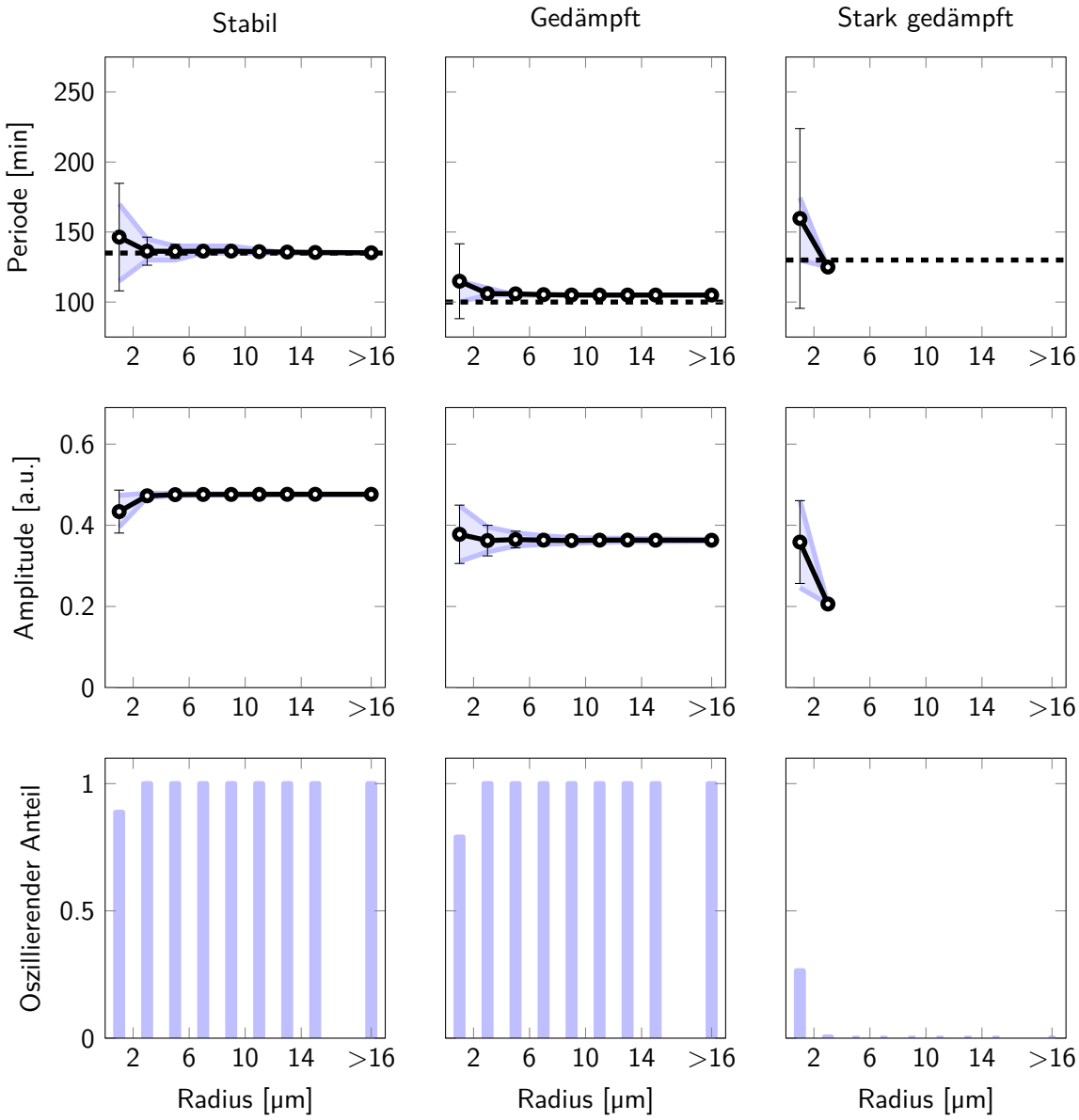


Abbildung C.1.: Ergebnisse der Simulation mit Poissonverteilung. Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil der Population dar. Die gestrichelte Linie ist die Periode im makroskopischen Volumen.

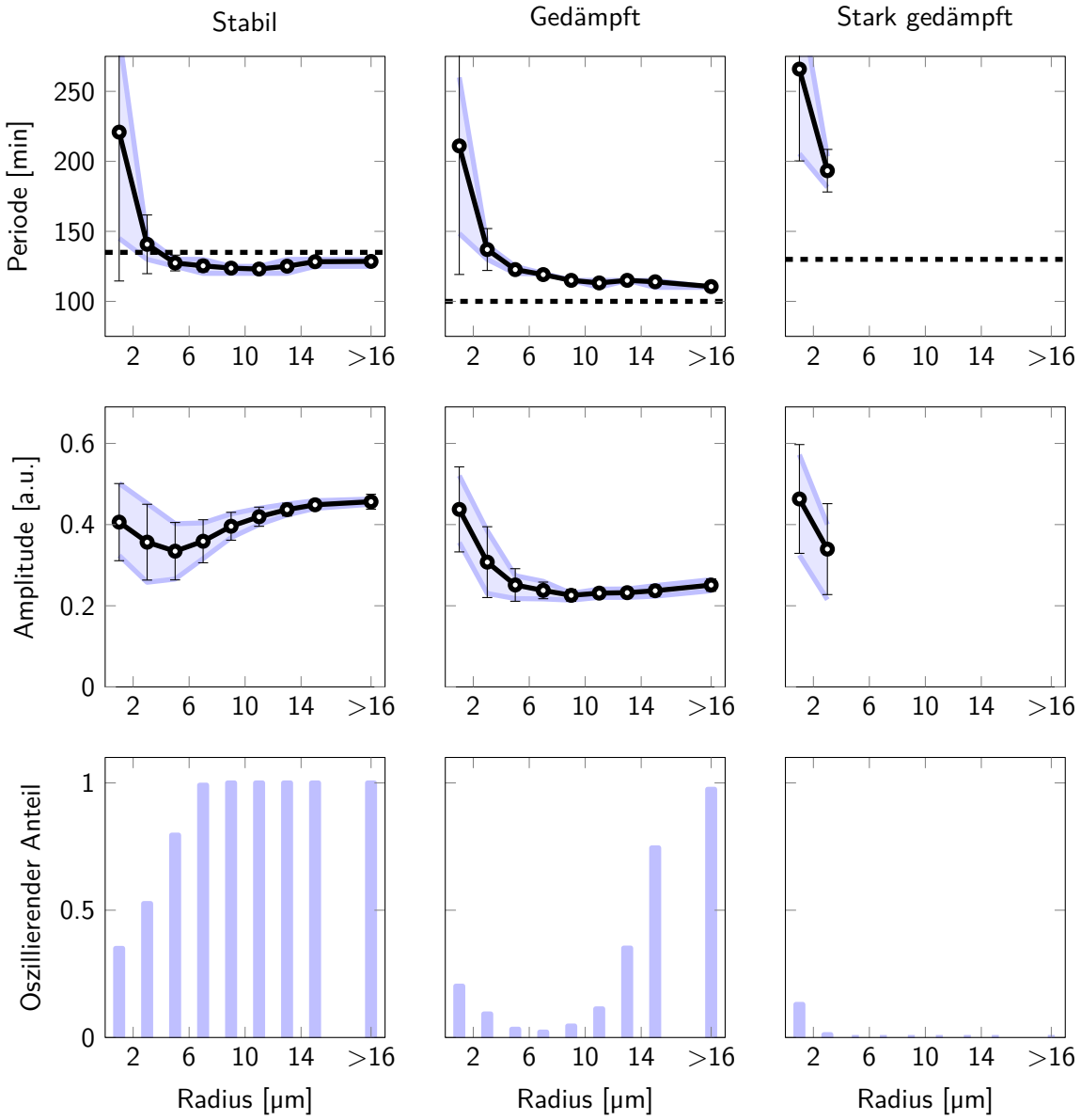


Abbildung C.2.: Ergebnisse der Simulation mit Proteinverlust während der Tröpfchenbildung. Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil der Population dar. Die gestrichelte Linie ist die Periode im makroskopischen Volumen.

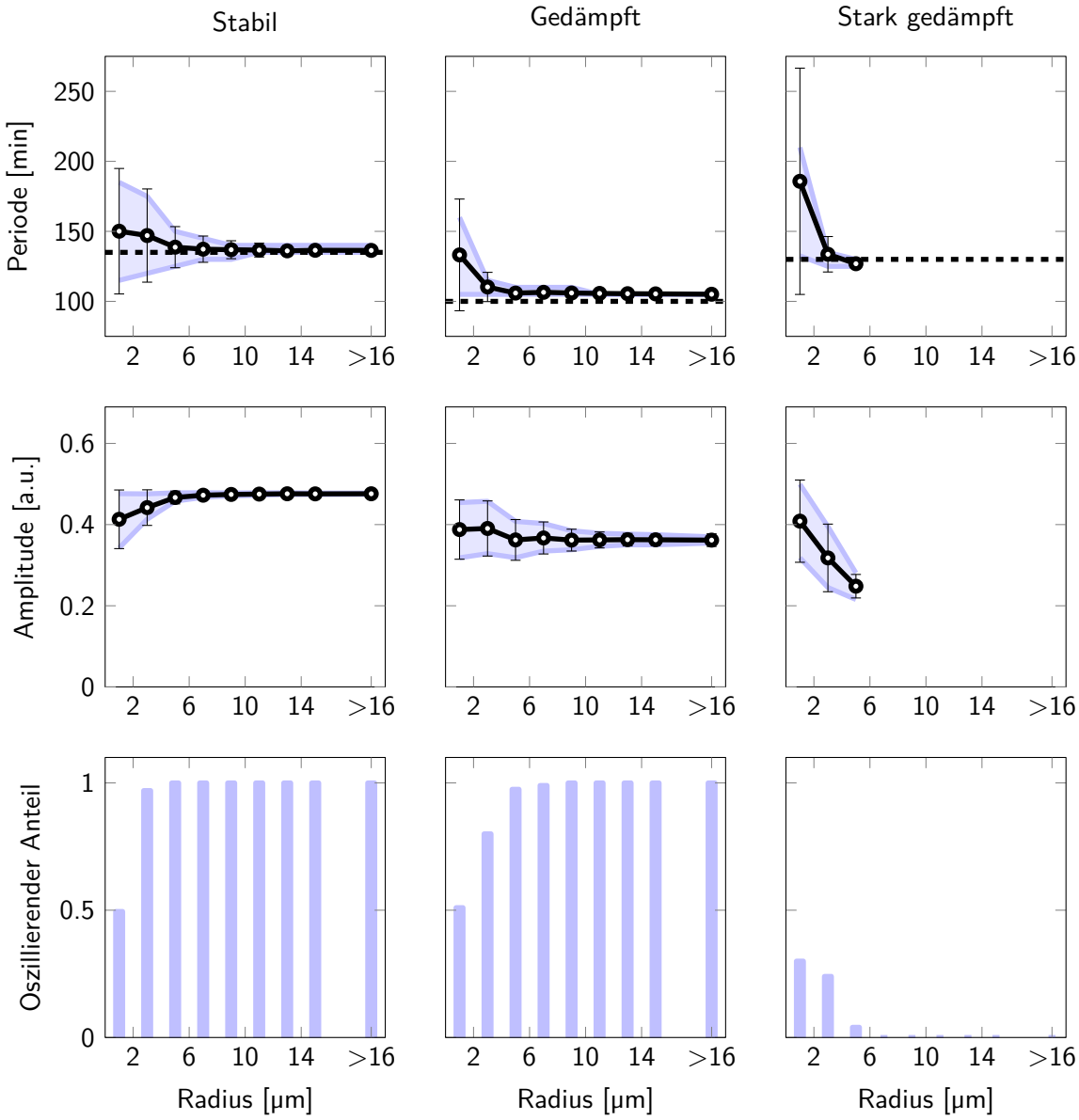


Abbildung C.3.: Ergebnisse der Simulation mit Gammaverteilung und $\beta = 10$. Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil der Population dar. Die gestrichelte Linie ist die Periode im makroskopischen Volumen.

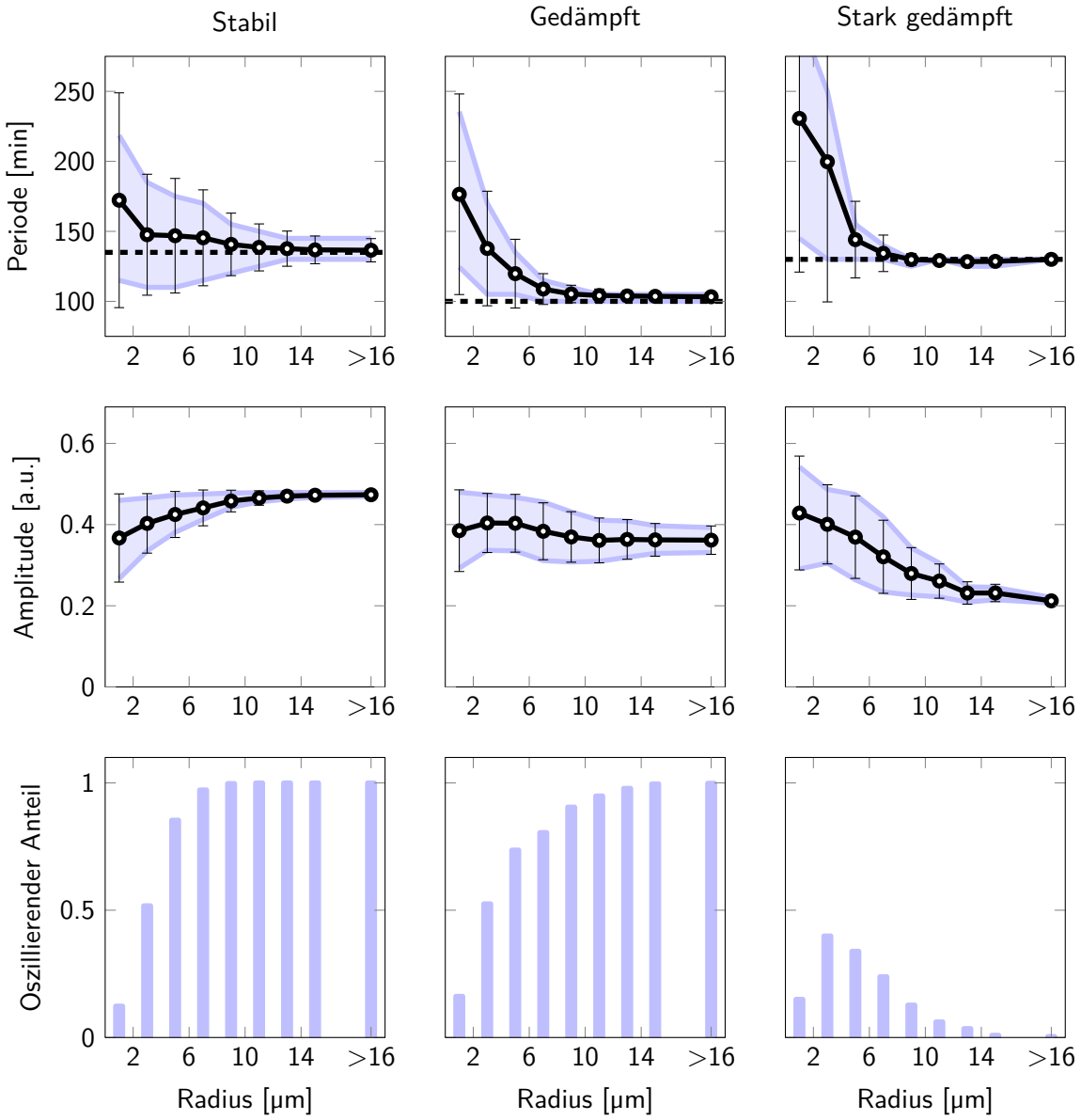


Abbildung C.4.: Ergebnisse der Simulation mit Gammaverteilung und $\beta = 100$. Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil der Population dar. Die gestrichelte Linie ist die Periode im makroskopischen Volumen.

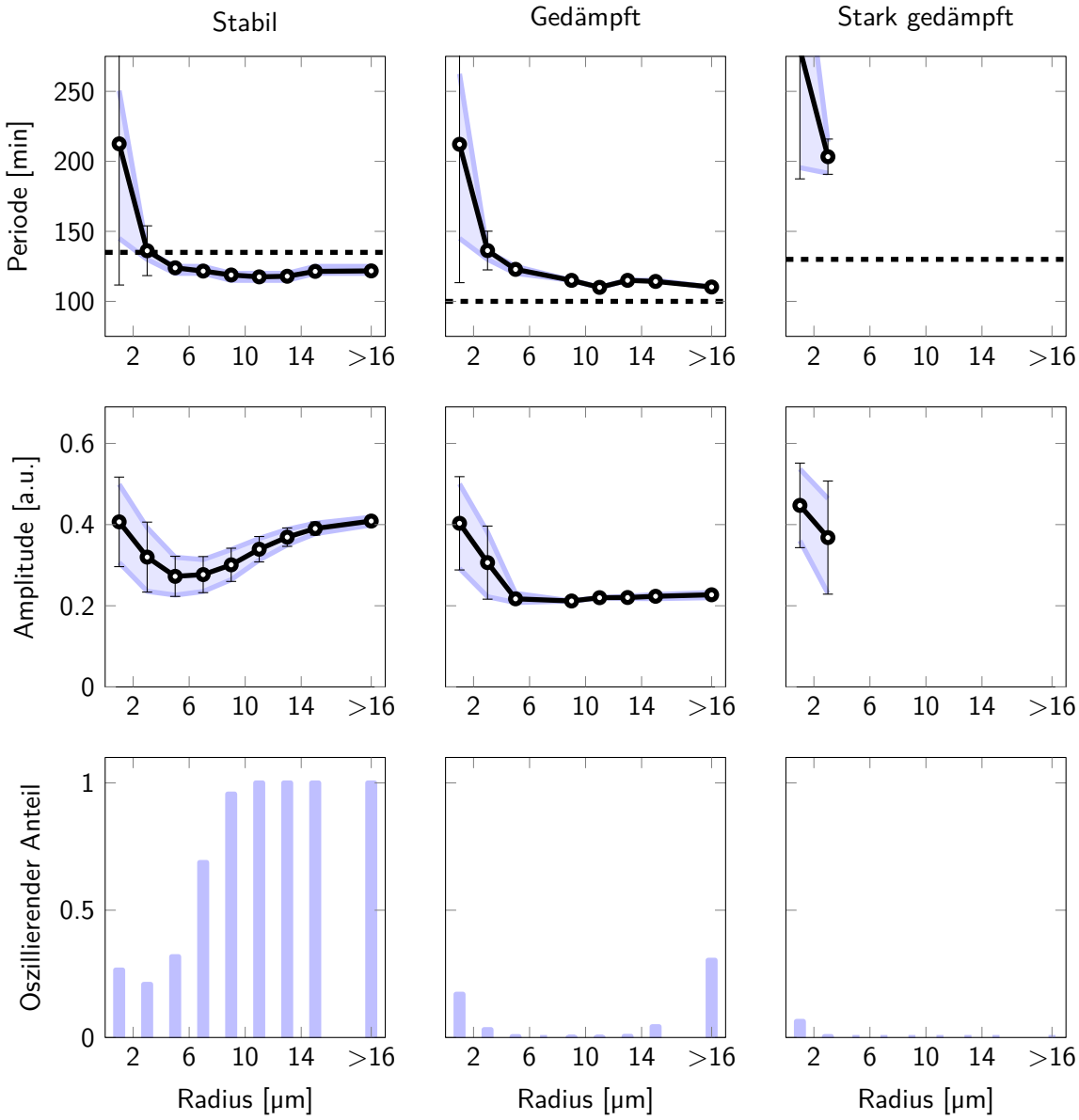


Abbildung C.5.: Ergebnisse der Simulation mit Gammaverteilung und $\beta = 10$ und Proteinverlust. Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil der Population dar. Die gestrichelte Linie ist die Periode im makroskopischen Volumen.

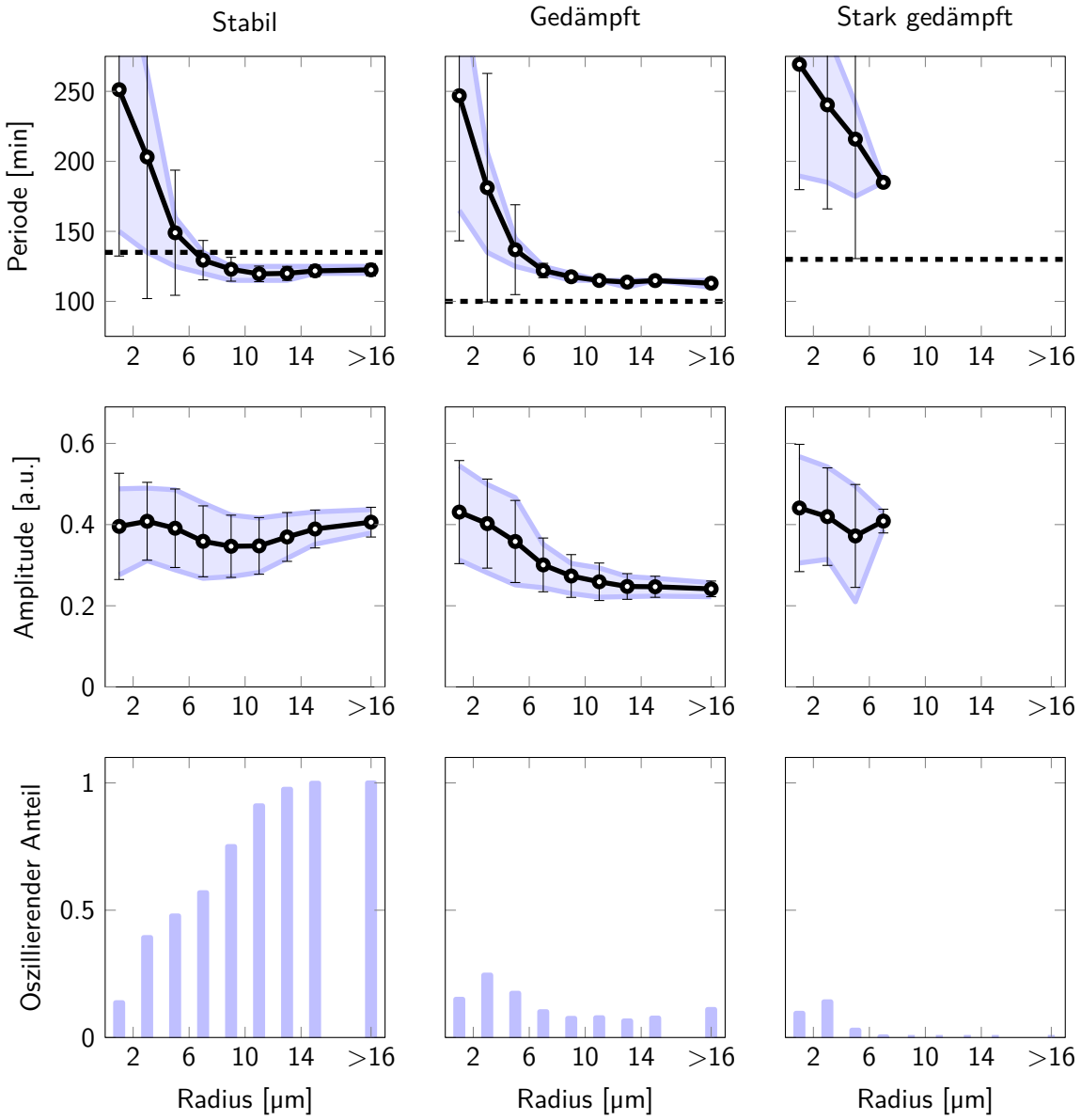


Abbildung C.6.: Ergebnisse der Simulation mit Gammaverteilung und $\beta = 100$ und Proteinverlust. Tröpfchen kleiner als 16 µm wurden nach ihrem Radius in 2 µm breite Bins eingeteilt und alle größeren in einem Bin zusammengefasst. Die Kreise sind die Mittelwerte, die Fehlerbalken die Standardabweichung und die umhüllende Fläche stellt den Bereich zwischen dem 20 %- und 80 %-Quantil der Population dar. Die gestrichelte Linie ist die Periode im makroskopischen Volumen.

D. DNA-Sequenzen

D.1. Oszillator

Tabelle D.1.: Oszillatorsequenzen

Name	Typ	Sequenz	Länge
T12-t	DNA	5'-TTT CTG ACT TTG TCA GTA TTA GTG TGT AGT AGT AGT TCA TTA GTG TCG TTC GTT CTT TGT TTC TCC CTA TAG TGA GTC G	79
T12-nt	DNA	5'-AAG CAA GGG TAA GAT GGA ATG ATA ATA CGA CTC ACT ATA GGG AGA AAC AAA GAA CGA ACG ACA CTA ATG AAC TAC TAC TAC ACA CTA ATA CTG ACA AAG TCA GAA A	106
T21-t	DNA	5'-TTT CTG ACT TTG TCA GTA TTA TCA TTC CAT CTT ACC CTT GCT TCA ATC CGT TTT ACT CTC CCT ATA GTG AGT CG	74
T21-nt	DNA	5'-**Texas Red** CAT TAG TGT CGT TCG TTC ACA GTA ATA CGA CTC ACT ATA GGG AGA GTA AAA CGG ATT GAA GCA AGG GTA AGA TGG AAT GAT AAT ACT GAC AAA GTC AGA AA	101
dI1	DNA	5'-GTG TGT AGT AGT AGT TCA TTA GTG TCG TTC GTT CAC AG	38
A1	DNA	5'-TAT TAC TGT GAA CGA ACG ACA CTA ATG AAC TAC TAC **Iowa Black RQ**	36
A2	DNA	5'-TAT TAT CAT TCC ATC TTA CCC TTG CTT CAA TCC GT **Iowa Black RQ**	35
rA1	RNA	5'-GGG AGA AAC AAA GAA CGA ACG ACA CUA AUG AAC UAC UAC UAC ACA CUA AUA CUG ACA AAG UCA GAA A	67
rI2	RNA	5'-GGG AGA GUA AAA CGG AUU GAA GCA AGG GUA AGA UGG AAU GAU AAU ACU GAC AAA GUC AGA AA	63

D.2. Stochastiksystem

Tabelle D.2.: Sequenzen des Stochastiksystems

Name	Typ	Sequenz	Länge
Reporter	DNA	5'-**TAMRA** CGT ACA CTC AGC GAT TAA GCT GAG TGT ACG TAA TGA TG **BHQ-2**	38
Sense	DNA	5'-ATT CAA TTA ATA CGA CTC ACT ATA GGG TCT GAT CGA TTC TAT CAT CAT TAC GTA CAC TCA GCT TAA TC	68
Antisense/ Inhibitor	DNA	5'-GAT TAA GCT GAG TGT ACG TAA TGA TGA TAG AAT CGA TCA GAC CCT ATA GTG AGT CGT ATT AAT TGA AT	68
RNA Produkt	RNA	5'-GGG UCU GAU CGA UUC UAU CAU CAU UAC GUA CAC UCA GCU UAA UC	44

D.3. Bakterienplasmide

D.3.1. AHL-Empfänger – 3960 Bp

Vektor: pSB1A3
Inserts: BBa_T9002

TCCCTATCAGTGATAGAGATTGACATCCCTATCAGTGATAGAGATACTGAGCACTACTAGAGAAAGAGGAGAAATAC
TAGATGAAAAACATAAATGCCGACGACACATACAGAATAATTAATAAAATTAAAGCTTGTAGAAGCAATAATGATAT
TAATCAATGCTTATCTGATATGACTAAAATGGTACATTGTGAATATTATTTACTCGCGATCATTTATCCTCATTCTA
TGGTTAAATCTGATATTTCAATCCTAGATAATTACCCTAAAAAATGGAGGCAATATTATGATGACGCTAATTTAATA
AAATATGATCCTATAGTAGATTATTCTAACTCCAATCATTCACCAATTAATTGGAATATATTTGAAAACAATGCTGT
AAATAAAAAATCTCCAAATGTAATTAAAGAAGCGAAAACATCAGGTCTTATCACTGGGTTTAGTTTCCCTATTCATA
CGGCTAACAATGGCTTCGGAATGCTTAGTTTTGCACATTCAGAAAAAGACAACTATATAGATAGTTTATTTTTACAT
GCGTGTATGAACATACCATTAATTGTTCCTTCTCTAGTTGATAATTATCGAAAAATAAATATAGCAAATAATAAATC
AAACAACGATTTAACCAAAAGAGAAAAAGAATGTTTAGCGTGGGCATGCGAAGGAAAAAGCTCTTGGGATATTTCAA
AAATATTAGGTTGCAGTGAGCGTACTGTCACTTTCCATTTAACCAATGCGCAAATGAAACTCAATACAACAAACCGC
TGCCAAAGTATTTCTAAAGCAATTTTAACAGGAGCAATTGATTGCCCATACTTTAAAAATTAATAACACTGATAGTG
CTAGTGTAGATCACTACTAGAGCCAGGCATCAAATAAAACGAAAGGCTCAGTCGAAAGACTGGGCCTTTCGTTTTAT
CTGTTGTTTGTCGGTGAACGCTCTCTACTAGAGTCACACTGGCTCACCTTCGGGTGGGCCTTTCTGCGTTTATATAC
TAGAGACCTGTAGGATCGTACAGGTTTACGCAAGAAAATGGTTTGTTATAGTCGAATAAATACTAGAGTCACACAGG
AAAGTACTAGATGCGTAAAGGAGAAGAACTTTTCACTGGAGTTGTCCCAATTCTTGTTGAATTAGATGGTGATGTTA
ATGGGCACAAATTTTCTGTCAGTGGAGAGGGTGAAGGTGATGCAACATACGGAAAACTTACCCTTAAATTTATTTGC
ACTACTGGAAAACTACCTGTTCCATGGCCAACACTTGTCACTACTTTCGGTTATGGTGTTCAATGCTTTGCGAGATA
CCCAGATCATATGAAACAGCATGACTTTTTCAAGAGTGCCATGCCCGAAGGTTATGTACAGGAAAGAACTATATTTT
TCAAAGATGACGGGAACTACAAGACACGTGCTGAAGTCAAGTTTGAAGGTGATACCCTTGTTAATAGAATCGAGTTA
AAAGGTATTGATTTTAAAGAAGATGGAAACATTCTTGGACACAAATTGGAATACAACTATAACTCACACAATGTATA
CATCATGGCAGACAAACAAAAGAATGGAATCAAAGTTAACTTCAAAATTAGACACAACATTGAAGATGGAAGCGTTC
AACTAGCAGACCATTATCAACAAAATACTCCAATTGGCGATGGCCCTGTCCTTTTACCAGACAACCATTACCTGTCC
ACACAATCTGCCCTTTCGAAAGATCCCAACGAAAAGAGAGACCACATGGTCCTTCTTGAGTTTGTAACAGCTGCTGG
GATTACACATGGCATGGATGAACTATACAAATAATACTAGTAGCGGCCGCTGCAGTCCGGCAAAAAAGGGCAAGGTG
TCACCACCCTGCCCTTTTTCTTTAAAACCGAAAAGATTACTTCGCGTTATGCAGGCTTCCTCGCTCACTGACTCGCT
GCGCTCGGTCGTTCGGCTGCGGCGAGCGGTATCAGCTCACTCAAAGGCGGTAATACGGTTATCCACAGAATCAGGGG
ATAACGCAGGAAAGAACATGTGAGCAAAAGGCCAGCAAAAGGCCAGGAACCGTAAAAAGGCCGCGTTGCTGGCGTTT
TTCCACAGGCTCCGCCCCCCTGACGAGCATCACAAAAATCGACGCTCAAGTCAGAGGTGGCGAAACCCGACAGGACT
ATAAAGATACCAGGCGTTTCCCCCTGGAAGCTCCCTCGTGCGCTCTCCTGTTCCGACCCTGCCGCTTACCGGATACC
TGTCCGCCTTTCTCCCTTCGGGAAGCGTGGCGCTTTCTCATAGCTCACGCTGTAGGTATCTCAGTTCGGTGTAGGTC
GTTCGCTCCAAGCTGGGCTGTGTGCACGAACCCCCCGTTCAGCCCGACCGCTGCGCCTTATCCGGTAACTATCGTCT
TGAGTCCAACCCGGTAAGACACGACTTATCGCCACTGGCAGCAGCCACTGGTAACAGGATTAGCAGAGCGAGGTATG
TAGGCGGTGCTACAGAGTTCTTGAAGTGGTGGCCTAACTACGGCTACACTAGAAGAACAGTATTTGGTATCTGCGCT
CTGCTGAAGCCAGTTACCTTCGGAAAAAGAGTTGGTAGCTCTTGATCCGGCAAACAAACCACCGCTGGTAGCGGTGG
TTTTTTTGTTTGCAAGCAGCAGATTACGCGCAGAAAAAAAGGATCTCAAGAAGATCCTTTGATCTTTTCTACGGGGT
CTGACGCTCAGTGGAACGAAAACTCACGTTAAGGGATTTTGGTCATGAGATTATCAAAAAGGATCTTCACCTAGATC

CTTTTAAATTAAAAATGAAGTTTTAAATCAATCTAAAGTATATATGAGTAAACTTGGTCTGACAGTTACCAATGCTT
AATCAGTGAGGCACCTATCTCAGCGATCTGTCTATTTCGTTCATCCATAGTTGCCTGACTCCCCGTCGTGTAGATAA
CTACGATACGGGAGGGCTTACCATCTGGCCCCAGTGCTGCAATGATACCGCGAGACCCACGCTCACCGGCTCCAGAT
TTATCAGCAATAAACCAGCCAGCCGGAAGGGCCGAGCGCAGAAGTGGTCCTGCAACTTTATCCGCCTCCATCCAGTC
TATTAATTGTTGCCGGGAAGCTAGAGTAAGTAGTTCGCCAGTTAATAGTTTGCGCAACGTTGTTGCCATTGCTACAG
GCATCGTGGTGTCACGCTCGTCGTTTGGTATGGCTTCATTCAGCTCCGGTTCCCAACGATCAAGGCGAGTTACATGA
TCCCCCATGTTGTGCAAAAAAGCGGTTAGCTCCTTCGGTCCTCCGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTT
ATCACTCATGGTTATGGCAGCACTGCATAATTCTCTTACTGTCATGCCATCCGTAAGATGCTTTTCTGTGACTGGTG
AGTACTCAACCAAGTCATTCTGAGAATAGTGTATGCGGCGACCGAGTTGCTCTTGCCCGGCGTCAATACGGGATAAT
ACCGCGCCACATAGCAGAACTTTAAAAGTGCTCATCATTGGAAAACGTTCTTCGGGGCGAAAACTCTCAAGGATCTT
ACCGCTGTTGAGATCCAGTTCGATATAACCCACTCGTGCACCCAACTGATCTTCAGCATCTTTTACTTTCACCAGCG
TTTCTGGGTGAGCAAAAACAGGAAGGCAAAATGCCGCAAAAAAGGGAATAAGGGCGACACGGAAATGTTGAATACTC
ATACTCTTCCTTTTTCAATATTATTGAAGCATTTATCAGGGTTATTGTCTCATGAGCGGATACATATTTGAATGTAT
TTAGAAAAATAAACAAATAGGGGTTCCGCGCACATTTCCCCGAAAAGTGCCACCTGACGTCTAAGAAACCATTATTA
TCATGACATTAACCTATAAAAATAGGCGTATCACGAGGCAGAATTTCAGATAAAAAAAATCCTTAGCTTTCGCTAAG
GATGATTTCTGGAATTCGCGGCCGCTTCTAGAG

D.3.2. IPTG-Empfänger – 6570 Bp

Vektor: pETDuet-1
Inserts: BBa_C0261 & BBa_E1010

GGGGAATTGTGAGCGGATAACAATTCCCCTCTAGAAAAGAGGAGAAATACTAGATGACTATAATGATAAAAAAATCG
GATTTTTTGGCAATTCCATCGGAGGAGTATAAAGGTATTCTAAGTCTTCGTTATCAAGTGTTTAAGCAAAGACTTGA
GTGGGACTTAGTTGTAGAAAATAACCTTGAATCAGATGAGTATGATAACTCAAATGCAGAATATATTTATGCTTGTG
ATGATACTGAAAATGTAAGTGGATGCTGGCGTTTATTACCTACAACAGGTGATTATATGCTGAAAAGTGTTTTTCCT
GAATTGCTTGGTCAACAGAGTGCTCCCAAAGATCCTAATATAGTCGAATTAAGTCGTTTTGCTGTAGGTAAAAATAG
CTCAAAGATAAATAACTCTGCTAGTGAAATTACAATGAAACTATTTGAAGCTATATATAAACACGCTGTTAGTCAAG
GTATTACAGAATATGTAACAGTAACATCAACAGCAATAGAGCGATTTTTAAAGCGTATTAAAGTTCCTTGTCATCGT
ATTGGAGACAAAGAAATTCATGTATTAGGTGATACTAAATCGGTTGTATTGTCTATGCCTATTAATGAACAGTTTAA
AAAAGCAGTCTTAAATGCTGCAAACGACGAAAACTACGCTTTAGTAGCTTAATAACTCTGATAGTGCTAGTGTAGAT
CTCCTGCAGGTCGACAAGCTTGCGGCCGCATAATGCTTAAGTCGAACAGAAAGTAATCGTATTGTACACGGCCGCAT
AATCGAAATTAATACGACTCACTATAGGGGAATTGTGAGCGGATAACAATTCCCCATCTTAGTATATTAGTTAAGTA
TAAGAAGGAGATATACATATGATGGCTTCCTCCGAAGACGTTATCAAAGAGTTCATGCGTTTCAAAGTTCGTATGGA
AGGTTCCGTTAACGGTCACGAGTTCGAAATCGAAGGTGAAGGTGAAGGTCGTCCGTACGAAGGTACCCAGACCGCTA
AACTGAAAGTTACCAAAGGTGGTCCGCTGCCGTTCGCTTGGGACATCCTGTCCCCGCAGTTCCAGTACGGTTCCAAA
GCTTACGTTAAACACCCGGCTGACATCCCGGACTACCTGAAACTGTCCTTCCCGGAAGGTTTCAAATGGGAACGTGT
TATGAACTTCGAAGACGGTGGTGTTGTTACCGTTACCCAGGACTCCTCCCTGCAAGACGGTGAGTTCATCTACAAAG
TTAAACTGCGTGGTACCAACTTCCCGTCCGACGGTCCGGTTATGCAGAAAAAAACCATGGGTTGGGAAGCTTCCACC
GAACGTATGTACCCGGAAGACGGTGCTCTGAAAGGTGAAATCAAAATGCGTCTGAAACTGAAAGACGGTGGTCACTA
CGACGCTGAAGTTAAAACCACCTACATGGCTAAAAAACCGGTTCAGCTGCCGGGTGCTTACAAAACCGACATCAAAC
TGGACATCACCTCCCACAACGAAGACTACACCATCGTTGAACAGTACGAACGTGCTGAAGGTCGTCACTCCACCGGT
GCTTAATAACGCTGATAGTGCTAGTGTAGATCGCTTAATTAACCTAGGCTGCTGCCACCGCTGAGCAATAACTAGCA

TAACCCCTTGGGGCCTCTAAACGGGTCTTGAGGGGTTTTTTGCTGAAAGGAGGAACTATATCCGGATTGGCGAATGG
GACGCGCCCTGTAGCGGCGCATTAAGCGCGGCGGGTGTGGTGGTTACGCGCAGCGTGACCGCTACACTTGCCAGCGC
CCTAGCGCCCGCTCCTTTCGCTTTCTTCCCTTCCTTTCTCGCCACGTTCGCCGGCTTTCCCCGTCAAGCTCTAAATC
GGGGGCTCCCTTTAGGGTTCCGATTTAGTGCTTTACGGCACCTCGACCCCAAAAAACTTGATTAGGGTGATGGTTCA
CGTAGTGGGCCATCGCCCTGATAGACGGTTTTTCGCCCTTTGACGTTGGAGTCCACGTTCTTTAATAGTGGACTCTT
GTTCCAAACTGGAACAACACTCAACCCTATCTCGGTCTATTCTTTTGATTTATAAGGGATTTTGCCGATTTCGGCCT
ATTGGTTAAAAAATGAGCTGATTTAACAAAAATTTAACGCGAATTTTAACAAAATATTAACGTTTACAATTTCTGGC
GGCACGATGGCATGAGATTATCAAAAAGGATCTTCACCTAGATCCTTTTAAATTAAAAATGAAGTTTTAAATCAATC
TAAAGTATATATGAGTAAACTTGGTCTGACAGTTACCAATGCTTAATCAGTGAGGCACCTATCTCAGCGATCTGTCT
ATTTCGTTCATCCATAGTTGCCTGACTCCCCGTCGTGTAGATAACTACGATACGGGAGGGCTTACCATCTGGCCCCA
GTGCTGCAATGATACCGCGAGACCCACGCTCACCGGCTCCAGATTTATCAGCAATAAACCAGCCAGCCGGAAGGGCC
GAGCGCAGAAGTGGTCCTGCAACTTTATCCGCCTCCATCCAGTCTATTAATTGTTGCCGGGAAGCTAGAGTAAGTAG
TTCGCCAGTTAATAGTTTGCGCAACGTTGTTGCCATTGCTACAGGCATCGTGGTGTCACGCTCGTCGTTTGGTATGG
CTTCATTCAGCTCCGGTTCCCAACGATCAAGGCGAGTTACATGATCCCCCATGTTGTGCAAAAAAGCGGTTAGCTCC
TTCGGTCCTCCGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTATCACTCATGGTTATGGCAGCACTGCATAATTC
TCTTACTGTCATGCCATCCGTAAGATGCTTTTCTGTGACTGGTGAGTACTCAACCAAGTCATTCTGAGAATAGTGTA
TGCGGCGACCGAGTTGCTCTTGCCCGGCGTCAATACGGGATAATACCGCGCCACATAGCAGAACTTTAAAAGTGCTC
ATCATTGGAAAACGTTCTTCGGGGCGAAAACTCTCAAGGATCTTACCGCTGTTGAGATCCAGTTCGATGTAACCCAC
TCGTGCACCCAACTGATCTTCAGCATCTTTTACTTTCACCAGCGTTTCTGGGTGAGCAAAAACAGGAAGGCAAAATG
CCGCAAAAAAGGGAATAAGGGCGACACGGAAATGTTGAATACTCATACTCTTCCTTTTTCAATCATGATTGAAGCAT
TTATCAGGGTTATTGTCTCATGAGCGGATACATATTTGAATGTATTTAGAAAAATAAACAAATAGGTCATGACCAAA
ATCCCTTAACGTGAGTTTTCGTTCCACTGAGCGTCAGACCCCGTAGAAAAGATCAAAGGATCTTCTTGAGATCCTTT
TTTTCTGCGCGTAATCTGCTGCTTGCAAACAAAAAAACCACCGCTACCAGCGGTGGTTTGTTTGCCGGATCAAGAGC
TACCAACTCTTTTTCCGAAGGTAACTGGCTTCAGCAGAGCGCAGATACCAAATACTGTCCTTCTAGTGTAGCCGTAG
TTAGGCCACCACTTCAAGAACTCTGTAGCACCGCCTACATACCTCGCTCTGCTAATCCTGTTACCAGTGGCTGCTGC
CAGTGGCGATAAGTCGTGTCTTACCGGGTTGGACTCAAGACGATAGTTACCGGATAAGGCGCAGCGGTCGGGCTGAA
CGGGGGGTTCGTGCACACAGCCCAGCTTGGAGCGAACGACCTACACCGAACTGAGATACCTACAGCGTGAGCTATGA
GAAAGCGCCACGCTTCCCGAAGGGAGAAAGGCGGACAGGTATCCGGTAAGCGGCAGGGTCGGAACAGGAGAGCGCAC
GAGGGAGCTTCCAGGGGGAAACGCCTGGTATCTTTATAGTCCTGTCGGGTTTCGCCACCTCTGACTTGAGCGTCGAT
TTTTGTGATGCTCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCAGCAACGCGGCCTTTTTACGGTTCCTGGCCTTT
TGCTGGCCTTTTGCTCACATGTTCTTTCCTGCGTTATCCCCTGATTCTGTGGATAACCGTATTACCGCCTTTGAGTG
AGCTGATACCGCTCGCCGCAGCCGAACGACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAAGAGCGCCTGATGC
GGTATTTTCTCCTTACGCATCTGTGCGGTATTTCACACCGCATATATGGTGCACTCTCAGTACAATCTGCTCTGATG
CCGCATAGTTAAGCCAGTATACACTCCGCTATCGCTACGTGACTGGGTCATGGCTGCGCCCCGACACCCGCCAACAC
CCGCTGACGCGCCCTGACGGGCTTGTCTGCTCCCGGCATCCGCTTACAGACAAGCTGTGACCGTCTCCGGGAGCTGC
ATGTGTCAGAGGTTTTCACCGTCATCACCGAAACGCGCGAGGCAGCTGCGGTAAAGCTCATCAGCGTGGTCGTGAAG
CGATTCACAGATGTCTGCCTGTTCATCCGCGTCCAGCTCGTTGAGTTTCTCCAGAAGCGTTAATGTCTGGCTTCTGA
TAAAGCGGGCCATGTTAAGGGCGGTTTTTTCCTGTTTGGTCACTGATGCCTCCGTGTAAGGGGGATTTCTGTTCATG
GGGGTAATGATACCGATGAAACGAGAGAGGATGCTCACGATACGGGTTACTGATGATGAACATGCCCGGTTACTGGA
ACGTTGTGAGGGTAAACAACTGGCGGTATGGATGCGGCGGGACCAGAGAAAAATCACTCAGGGTCAATGCCAGCGCT
TCGTTAATACAGATGTAGGTGTTCCACAGGGTAGCCAGCAGCATCCTGCGATGCAGATCCGGAACATAATGGTGCAG
GGCGCTGACTTCCGCGTTTCCAGACTTTACGAAACACGGAAACCGAAGACCATTCATGTTGTTGCTCAGGTCGCAGA
CGTTTTGCAGCAGCAGTCGCTTCACGTTCGCTCGCGTATCGGTGATTCATTCTGCTAACCAGTAAGGCAACCCCGCC
AGCCTAGCCGGGTCCTCAACGACAGGAGCACGATCATGCTAGTCATGCCCCGCGCCCACCGGAAGGAGCTGACTGGG

TTGAAGGCTCTCAAGGGCATCGGTCGAGATCCCGGTGCCTAATGAGTGAGCTAACTTACATTAATTGCGTTGCGCTC
ACTGCCCGCTTTCCAGTCGGGAAACCTGTCGTGCCAGCTGCATTAATGAATCGGCCAACGCGCGGGGAGAGGCGGTT
TGCGTATTGGGCGCCAGGGTGGTTTTTCTTTTCACCAGTGAGACGGGCAACAGCTGATTGCCCTTCACCGCCTGGCC
CTGAGAGAGTTGCAGCAAGCGGTCCACGCTGGTTTGCCCCAGCAGGCGAAAATCCTGTTTGATGGTGGTTAACGGCG
GGATATAACATGAGCTGTCTTCGGTATCGTCGTATCCCACTACCGAGATGTCCGCACCAACGCGCAGCCCGGACTCG
GTAATGGCGCGCATTGCGCCCAGCGCCATCTGATCGTTGGCAACCAGCATCGCAGTGGGAACGATGCCCTCATTCAG
CATTTGCATGGTTTGTTGAAAACCGGACATGGCACTCCAGTCGCCTTCCCGTTCCGCTATCGGCTGAATTTGATTGC
GAGTGAGATATTTATGCCAGCCAGCCAGACGCAGACGCGCCGAGACAGAACTTAATGGGCCCGCTAACAGCGCGATT
TGCTGGTGACCCAATGCGACCAGATGCTCCACGCCCAGTCGCGTACCGTCTTCATGGGAGAAAATAATACTGTTGAT
GGGTGTCTGGTCAGAGACATCAAGAAATAACGCCGGAACATTAGTGCAGGCAGCTTCCACAGCAATGGCATCCTGGT
CATCCAGCGGATAGTTAATGATCAGCCCACTGACGCGTTGCGCGAGAAGATTGTGCACCGCCGCTTTACAGGCTTCG
ACGCCGCTTCGTTCTACCATCGACACCACCACGCTGGCACCCAGTTGATCGGCGCGAGATTTAATCGCCGCGACAAT
TTGCGACGGCGCGTGCAGGGCCAGACTGGAGGTGGCAACGCCAATCAGCAACGACTGTTTGCCCGCCAGTTGTTGTG
CCACGCGGTTGGGAATGTAATTCAGCTCCGCCATCGCCGCTTCCACTTTTTCCCGCGTTTTCGCAGAAACGTGGCTG
GCCTGGTTCACCACGCGGGAAACGGTCTGATAAGAGACACCGGCATACTCTGCGACATCGTATAACGTTACTGGTTT
CACATTCACCACCCTGAATTGACTCTCTTCCGGGCGCTATCATGCCATACCGCGAAAGGTTTTGCGCCATTCGATGG
TGTCCGGGATCTCGACGCTCTCCCTTATGCGACTCCTGCATTAGGAAGCAGCCCAGTAGTAGGTTGAGGCCGTTGAG
CACCGCCGCCGCAAGGAATGGTGCATGCAAGGAGATGGCGCCCAACAGTCCCCCGGCCACGGGGCCTGCCACCATAC
CCACGCCGAAACAAGCGCTCATGAGCCCGAAGTGGCGAGCCCGATCTTCCCCATCGGTGATGTCGGCGATATAGGCG
CCAGCAACCGCACCTGTGGCGCCGGTGATGCCGGCCACGATGCGTCCGGCGTAGAGGATCGAGATCGATCTCGATCC
CGCGAAATTAATACGACTCACTATA

D.3.3. AHL-Sender – 5986 Bp

Vektor: pETDuet-1
Inserts: BBa_C0261

GGGGAATTGTGAGCGGATAACAATTCCCCTCTAGAAAAGAGGAGAAATACTAGATGACTATAATGATAAAAAAATCG
GATTTTTTGGCAATTCCATCGGAGGAGTATAAAGGTATTCTAAGTCTTCGTTATCAAGTGTTTAAGCAAAGACTTGA
GTGGGACTTAGTTGTAGAAAATAACCTTGAATCAGATGAGTATGATAACTCAAATGCAGAATATATTTATGCTTGTG
ATGATACTGAAAATGTAAGTGGATGCTGGCGTTTATTACCTACAACAGGTGATTATATGCTGAAAAGTGTTTTTCCT
GAATTGCTTGGTCAACAGAGTGCTCCCAAAGATCCTAATATAGTCGAATTAAGTCGTTTTGCTGTAGGTAAAAATAG
CTCAAAGATAAATAACTCTGCTAGTGAAATTACAATGAAACTATTTGAAGCTATATATAAACACGCTGTTAGTCAAG
GTATTACAGAATATGTAACAGTAACATCAACAGCAATAGAGCGATTTTTAAAGCGTATTAAAGTTCCTTGTCATCGT
ATTGGAGACAAAGAAATTCATGTATTAGGTGATACTAAATCGGTTGTATTGTCTATGCCTATTAATGAACAGTTTAA
AAAAGCAGTCTTAAATGCTGCAAACGACGAAAACTACGCTTTAGTAGCTTAATAACTCTGATAGTGCTAGTGTAGAT
CTCCTGCAGGTCGACAAGCTTGCGGCCGCATAATGCTTAAGTCGAACAGAAAGTAATCGTATTGTACACGGCCGCAT
AATCGAAATTAATACGACTCACTATAGGGGAATTGTGAGCGGATAACAATTCCCCATCTTAGTATATTAGTTAAGTA
TAAGAAGGAGATATACATATGGCAGATCTCAATTGGATATCGGCCGGCCACGCGATCGCTGACGTCGGTACCCTCGA
GTCTGGTAAAGAAACCGCTGCTGCGAAATTTGAACGCCAGCACATGGACTCGTCTACTAGCGCAGCTTAATTAACCT
AGGCTGCTGCCACCGCTGAGCAATAACTAGCATAACCCCTTGGGGCCTCTAAACGGGTCTTGAGGGGTTTTTTGCTG
AAAGGAGGAACTATATCCGGATTGGCGAATGGGACGCGCCCTGTAGCGGCGCATTAAGCGCGGCGGGTGTGGTGGTT
ACGCGCAGCGTGACCGCTACACTTGCCAGCGCCCTAGCGCCCGCTCCTTTCGCTTTCTTCCCTTCCTTTCTCGCCAC

GTTCGCCGGCTTTCCCCGTCAAGCTCTAAATCGGGGGCTCCCTTTAGGGTTCCGATTTAGTGCTTTACGGCACCTCG
ACCCCAAAAAACTTGATTAGGGTGATGGTTCACGTAGTGGGCCATCGCCCTGATAGACGGTTTTTCGCCCTTTGACG
TTGGAGTCCACGTTCTTTAATAGTGGACTCTTGTTCCAAACTGGAACAACACTCAACCCTATCTCGGTCTATTCTTT
TGATTTATAAGGGATTTTGCCGATTTCGGCCTATTGGTTAAAAAATGAGCTGATTTAACAAAAATTTAACGCGAATT
TTAACAAAATATTAACGTTTACAATTTCTGGCGGCACGATGGCATGAGATTATCAAAAAGGATCTTCACCTAGATCC
TTTTAAATTAAAAATGAAGTTTTAAATCAATCTAAAGTATATATGAGTAAACTTGGTCTGACAGTTACCAATGCTTA
ATCAGTGAGGCACCTATCTCAGCGATCTGTCTATTTCGTTCATCCATAGTTGCCTGACTCCCCGTCGTGTAGATAAC
TACGATACGGGAGGGCTTACCATCTGGCCCCAGTGCTGCAATGATACCGCGAGACCCACGCTCACCGGCTCCAGATT
TATCAGCAATAAACCAGCCAGCCGGAAGGGCCGAGCGCAGAAGTGGTCCTGCAACTTTATCCGCCTCCATCCAGTCT
ATTAATTGTTGCCGGGAAGCTAGAGTAAGTAGTTCGCCAGTTAATAGTTTGCGCAACGTTGTTGCCATTGCTACAGG
CATCGTGGTGTCACGCTCGTCGTTTGGTATGGCTTCATTCAGCTCCGGTTCCCAACGATCAAGGCGAGTTACATGAT
CCCCCATGTTGTGCAAAAAAGCGGTTAGCTCCTTCGGTCCTCCGATCGTTGTCAGAAGTAAGTTGGCCGCAGTGTTA
TCACTCATGGTTATGGCAGCACTGCATAATTCTCTTACTGTCATGCCATCCGTAAGATGCTTTTCTGTGACTGGTGA
GTACTCAACCAAGTCATTCTGAGAATAGTGTATGCGGCGACCGAGTTGCTCTTGCCCGGCGTCAATACGGGATAATA
CCGCGCCACATAGCAGAACTTTAAAAGTGCTCATCATTGGAAAACGTTCTTCGGGGCGAAAACTCTCAAGGATCTTA
CCGCTGTTGAGATCCAGTTCGATGTAACCCACTCGTGCACCCAACTGATCTTCAGCATCTTTTACTTTCACCAGCGT
TTCTGGGTGAGCAAAAACAGGAAGGCAAAATGCCGCAAAAAAGGGAATAAGGGCGACACGGAAATGTTGAATACTCA
TACTCTTCCTTTTTCAATCATGATTGAAGCATTTATCAGGGTTATTGTCTCATGAGCGGATACATATTTGAATGTAT
TTAGAAAAATAAACAAATAGGTCATGACCAAAATCCCTTAACGTGAGTTTTCGTTCCACTGAGCGTCAGACCCCGTA
GAAAAGATCAAAGGATCTTCTTGAGATCCTTTTTTTCTGCGCGTAATCTGCTGCTTGCAAACAAAAAAACCACCGCT
ACCAGCGGTGGTTTGTTTGCCGGATCAAGAGCTACCAACTCTTTTTCCGAAGGTAACTGGCTTCAGCAGAGCGCAGA
TACCAAATACTGTCCTTCTAGTGTAGCCGTAGTTAGGCCACCACTTCAAGAACTCTGTAGCACCGCCTACATACCTC
GCTCTGCTAATCCTGTTACCAGTGGCTGCTGCCAGTGGCGATAAGTCGTGTCTTACCGGGTTGGACTCAAGACGATA
GTTACCGGATAAGGCGCAGCGGTCGGGCTGAACGGGGGGTTCGTGCACACAGCCCAGCTTGGAGCGAACGACCTACA
CCGAACTGAGATACCTACAGCGTGAGCTATGAGAAAGCGCCACGCTTCCCGAAGGGAGAAAGGCGGACAGGTATCCG
GTAAGCGGCAGGGTCGGAACAGGAGAGCGCACGAGGGAGCTTCCAGGGGGAAACGCCTGGTATCTTTATAGTCCTGT
CGGGTTTCGCCACCTCTGACTTGAGCGTCGATTTTTGTGATGCTCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCA
GCAACGCGGCCTTTTTACGGTTCCTGGCCTTTTGCTGGCCTTTTGCTCACATGTTCTTTCCTGCGTTATCCCCTGAT
TCTGTGGATAACCGTATTACCGCCTTTGAGTGAGCTGATACCGCTCGCCGCAGCCGAACGACCGAGCGCAGCGAGTC
AGTGAGCGAGGAAGCGGAAGAGCGCCTGATGCGGTATTTTCTCCTTACGCATCTGTGCGGTATTTCACACCGCATAT
ATGGTGCACTCTCAGTACAATCTGCTCTGATGCCGCATAGTTAAGCCAGTATACACTCCGCTATCGCTACGTGACTG
GGTCATGGCTGCGCCCCGACACCCGCCAACACCCGCTGACGCGCCCTGACGGGCTTGTCTGCTCCCGGCATCCGCTT
ACAGACAAGCTGTGACCGTCTCCGGGAGCTGCATGTGTCAGAGGTTTTCACCGTCATCACCGAAACGCGCGAGGCAG
CTGCGGTAAAGCTCATCAGCGTGGTCGTGAAGCGATTCACAGATGTCTGCCTGTTCATCCGCGTCCAGCTCGTTGAG
TTTCTCCAGAAGCGTTAATGTCTGGCTTCTGATAAAGCGGGCCATGTTAAGGGCGGTTTTTTCCTGTTTGGTCACTG
ATGCCTCCGTGTAAGGGGGATTTCTGTTCATGGGGGTAATGATACCGATGAAACGAGAGAGGATGCTCACGATACGG
GTTACTGATGATGAACATGCCCGGTTACTGGAACGTTGTGAGGGTAAACAACTGGCGGTATGGATGCGGCGGGACCA
GAGAAAAATCACTCAGGGTCAATGCCAGCGCTTCGTTAATACAGATGTAGGTGTTCCACAGGGTAGCCAGCAGCATC
CTGCGATGCAGATCCGGAACATAATGGTGCAGGGCGCTGACTTCCGCGTTTCCAGACTTTACGAAACACGGAAACCG
AAGACCATTCATGTTGTTGCTCAGGTCGCAGACGTTTTGCAGCAGCAGTCGCTTCACGTTCGCTCGCGTATCGGTGA
TTCATTCTGCTAACCAGTAAGGCAACCCCGCCAGCCTAGCCGGGTCCTCAACGACAGGAGCACGATCATGCTAGTCA
TGCCCCGCGCCCACCGGAAGGAGCTGACTGGGTTGAAGGCTCTCAAGGGCATCGGTCGAGATCCCGGTGCCTAATGA
GTGAGCTAACTTACATTAATTGCGTTGCGCTCACTGCCCGCTTTCCAGTCGGGAAACCTGTCGTGCCAGCTGCATTA
ATGAATCGGCCAACGCGCGGGGAGAGGCGGTTTGCGTATTGGGCGCCAGGGTGGTTTTTCTTTTCACCAGTGAGACG

GGCAACAGCTGATTGCCCTTCACCGCCTGGCCCTGAGAGAGTTGCAGCAAGCGGTCCACGCTGGTTTGCCCCAGCAG
GCGAAAATCCTGTTTGATGGTGGTTAACGGCGGGATATAACATGAGCTGTCTTCGGTATCGTCGTATCCCACTACCG
AGATGTCCGCACCAACGCGCAGCCCGGACTCGGTAATGGCGCGCATTGCGCCCAGCGCCATCTGATCGTTGGCAACC
AGCATCGCAGTGGGAACGATGCCCTCATTCAGCATTTGCATGGTTTGTTGAAAACCGGACATGGCACTCCAGTCGCC
TTCCCGTTCCGCTATCGGCTGAATTTGATTGCGAGTGAGATATTTATGCCAGCCAGCCAGACGCAGACGCGCCGAGA
CAGAACTTAATGGGCCCGCTAACAGCGCGATTTGCTGGTGACCCAATGCGACCAGATGCTCCACGCCCAGTCGCGTA
CCGTCTTCATGGGAGAAAATAATACTGTTGATGGGTGTCTGGTCAGAGACATCAAGAAATAACGCCGGAACATTAGT
GCAGGCAGCTTCCACAGCAATGGCATCCTGGTCATCCAGCGGATAGTTAATGATCAGCCCACTGACGCGTTGCGCGA
GAAGATTGTGCACCGCCGCTTTACAGGCTTCGACGCCGCTTCGTTCTACCATCGACACCACCACGCTGGCACCCAGT
TGATCGGCGCGAGATTTAATCGCCGCGACAATTTGCGACGGCGCGTGCAGGGCCAGACTGGAGGTGGCAACGCCAAT
CAGCAACGACTGTTTGCCCGCCAGTTGTTGTGCCACGCGGTTGGGAATGTAATTCAGCTCCGCCATCGCCGCTTCCA
CTTTTTCCCGCGTTTTCGCAGAAACGTGGCTGGCCTGGTTCACCACGCGGGAAACGGTCTGATAAGAGACACCGGCA
TACTCTGCGACATCGTATAACGTTACTGGTTTCACATTCACCACCCTGAATTGACTCTCTTCCGGGCGCTATCATGC
CATACCGCGAAAGGTTTTGCGCCATTCGATGGTGTCCGGGATCTCGACGCTCTCCCTTATGCGACTCCTGCATTAGG
AAGCAGCCCAGTAGTAGGTTGAGGCCGTTGAGCACCGCCGCCGCAAGGAATGGTGCATGCAAGGAGATGGCGCCCAA
CAGTCCCCCGGCCACGGGGCCTGCCACCATACCCACGCCGAAACAAGCGCTCATGAGCCCGAAGTGGCGAGCCCGAT
CTTCCCCATCGGTGATGTCGGCGATATAGGCGCCAGCAACCGCACCTGTGGCGCCGGTGATGCCGGCCACGATGCGT
CCGGCGTAGAGGATCGAGATCGATCTCGATCCCGCGAAATTAATACGACTCACTATA

D.3.4. UND-Gatter – 3964 Bp

Vektor: pSB1A3
Inserts: placO-1 & BBa_T9002

CTGATAAATGTGAGCGGATAACATTGACATTGTGAGCGGATAACAAGATACTGAGCACTACTAGAGAAAGAGGAGAA
ATACTAGATGAAAAACATAAATGCCGACGACACATACAGAATAATTAATAAAATTAAAGCTTGTAGAAGCAATAATG
ATATTAATCAATGCTTATCTGATATGACTAAAATGGTACATTGTGAATATTATTTACTCGCGATCATTTATCCTCAT
TCTATGGTTAAATCTGATATTTCAATCCTAGATAATTACCCTAAAAAATGGAGGCAATATTATGATGACGCTAATTT
AATAAAATATGATCCTATAGTAGATTATTCTAACTCCAATCATTCACCAATTAATTGGAATATATTTGAAAACAATG
CTGTAAATAAAAAATCTCCAAATGTAATTAAAGAAGCGAAAACATCAGGTCTTATCACTGGGTTTAGTTTCCCTATT
CATACGGCTAACAATGGCTTCGGAATGCTTAGTTTTGCACATTCAGAAAAAGACAACTATATAGATAGTTTATTTTT
ACATGCGTGTATGAACATACCATTAATTGTTCCTTCTCTAGTTGATAATTATCGAAAAATAAATATAGCAAATAATA
AATCAAACAACGATTTAACCAAAAGAGAAAAAGAATGTTTAGCGTGGGCATGCGAAGGAAAAAGCTCTTGGGATATT
TCAAAAATATTAGGTTGCAGTGAGCGTACTGTCACTTTCCATTTAACCAATGCGCAAATGAAACTCAATACAACAAA
CCGCTGCCAAAGTATTTCTAAAGCAATTTTAACAGGAGCAATTGATTGCCCATACTTTAAAAATTAATAACACTGAT
AGTGCTAGTGTAGATCACTACTAGAGCCAGGCATCAAATAAAACGAAAGGCTCAGTCGAAAGACTGGGCCTTTCGTT
TTATCTGTTGTTTGTCGGTGAACGCTCTCTACTAGAGTCACACTGGCTCACCTTCGGGTGGGCCTTTCTGCGTTTAT
ATACTAGAGACCTGTAGGATCGTACAGGTTTACGCAAGAAAATGGTTTGTTATAGTCGAATAAATACTAGAGTCACA
CAGGAAAGTACTAGATGCGTAAAGGAGAAGAACTTTTCACTGGAGTTGTCCCAATTCTTGTTGAATTAGATGGTGAT
GTTAATGGGCACAAATTTTCTGTCAGTGGAGAGGGTGAAGGTGATGCAACATACGGAAAACTTACCCTTAAATTTAT
TTGCACTACTGGAAAACTACCTGTTCCATGGCCAACACTTGTCACTACTTTCGGTTATGGTGTTCAATGCTTTGCGA
GATACCCAGATCATATGAAACAGCATGACTTTTTCAAGAGTGCCATGCCCGAAGGTTATGTACAGGAAAGAACTATA
TTTTTCAAAGATGACGGGAACTACAAGACACGTGCTGAAGTCAAGTTTGAAGGTGATACCCTTGTTAATAGAATCGA

GTTAAAAGGTATTGATTTTAAAGAAGATGGAAACATTCTTGGACACAAATTGGAATACAACTATAACTCACACAATG
TATACATCATGGCAGACAAACAAAAGAATGGAATCAAAGTTAACTTCAAAATTAGACACAACATTGAAGATGGAAGC
GTTCAACTAGCAGACCATTATCAACAAAATACTCCAATTGGCGATGGCCCTGTCCTTTTACCAGACAACCATTACCT
GTCCACACAATCTGCCCTTTCGAAAGATCCCAACGAAAAGAGAGACCACATGGTCCTTCTTGAGTTTGTAACAGCTG
CTGGGATTACACATGGCATGGATGAACTATACAAATAATACTAGTAGCGGCCGCTGCAGTCCGGCAAAAAAGGGCAA
GGTGTCACCACCCTGCCCTTTTTCTTTAAAACCGAAAAGATTACTTCGCGTTATGCAGGCTTCCTCGCTCACTGACT
CGCTGCGCTCGGTCGTTCGGCTGCGGCGAGCGGTATCAGCTCACTCAAAGGCGGTAATACGGTTATCCACAGAATCA
GGGGATAACGCAGGAAAGAACATGTGAGCAAAAGGCCAGCAAAAGGCCAGGAACCGTAAAAAGGCCGCGTTGCTGGC
GTTTTTCCACAGGCTCCGCCCCCCTGACGAGCATCACAAAAATCGACGCTCAAGTCAGAGGTGGCGAAACCCGACAG
GACTATAAAGATACCAGGCGTTTCCCCCTGGAAGCTCCCTCGTGCGCTCTCCTGTTCCGACCCTGCCGCTTACCGGA
TACCTGTCCGCCTTTCTCCCTTCGGGAAGCGTGGCGCTTTCTCATAGCTCACGCTGTAGGTATCTCAGTTCGGTGTA
GGTCGTTCGCTCCAAGCTGGGCTGTGTGCACGAACCCCCCGTTCAGCCCGACCGCTGCGCCTTATCCGGTAACTATC
GTCTTGAGTCCAACCCGGTAAGACACGACTTATCGCCACTGGCAGCAGCCACTGGTAACAGGATTAGCAGAGCGAGG
TATGTAGGCGGTGCTACAGAGTTCTTGAAGTGGTGGCCTAACTACGGCTACACTAGAAGAACAGTATTTGGTATCTG
CGCTCTGCTGAAGCCAGTTACCTTCGGAAAAAGAGTTGGTAGCTCTTGATCCGGCAAACAAACCACCGCTGGTAGCG
GTGGTTTTTTTGTTTGCAAGCAGCAGATTACGCGCAGAAAAAAAGGATCTCAAGAAGATCCTTTGATCTTTTCTACG
GGGTCTGACGCTCAGTGGAACGAAAACTCACGTTAAGGGATTTTGGTCATGAGATTATCAAAAAGGATCTTCACCTA
GATCCTTTTAAATTAAAAATGAAGTTTTAAATCAATCTAAAGTATATATGAGTAAACTTGGTCTGACAGTTACCAAT
GCTTAATCAGTGAGGCACCTATCTCAGCGATCTGTCTATTTCGTTCATCCATAGTTGCCTGACTCCCCGTCGTGTAG
ATAACTACGATACGGGAGGGCTTACCATCTGGCCCCAGTGCTGCAATGATACCGCGAGACCCACGCTCACCGGCTCC
AGATTTATCAGCAATAAACCAGCCAGCCGGAAGGGCCGAGCGCAGAAGTGGTCCTGCAACTTTATCCGCCTCCATCC
AGTCTATTAATTGTTGCCGGGAAGCTAGAGTAAGTAGTTCGCCAGTTAATAGTTTGCGCAACGTTGTTGCCATTGCT
ACAGGCATCGTGGTGTCACGCTCGTCGTTTGGTATGGCTTCATTCAGCTCCGGTTCCCAACGATCAAGGCGAGTTAC
ATGATCCCCCATGTTGTGCAAAAAAGCGGTTAGCTCCTTCGGTCCTCCGATCGTTGTCAGAAGTAAGTTGGCCGCAG
TGTTATCACTCATGGTTATGGCAGCACTGCATAATTCTCTTACTGTCATGCCATCCGTAAGATGCTTTTCTGTGACT
GGTGAGTACTCAACCAAGTCATTCTGAGAATAGTGTATGCGGCGACCGAGTTGCTCTTGCCCGGCGTCAATACGGGA
TAATACCGCGCCACATAGCAGAACTTTAAAAGTGCTCATCATTGGAAAACGTTCTTCGGGGCGAAAACTCTCAAGGA
TCTTACCGCTGTTGAGATCCAGTTCGATATAACCCACTCGTGCACCCAACTGATCTTCAGCATCTTTTACTTTCACC
AGCGTTTCTGGGTGAGCAAAAACAGGAAGGCAAAATGCCGCAAAAAAGGGAATAAGGGCGACACGGAAATGTTGAAT
ACTCATACTCTTCCTTTTTCAATATTATTGAAGCATTTATCAGGGTTATTGTCTCATGAGCGGATACATATTTGAAT
GTATTTAGAAAAATAAACAAATAGGGGTTCCGCGCACATTTCCCCGAAAAGTGCCACCTGACGTCTAAGAAACCATT
ATTATCATGACATTAACCTATAAAAATAGGCGTATCACGAGGCAGAATTTCAGATAAAAAAAATCCTTAGCTTTCGC
TAAGGATGATTTCTGGAATTCGCGGCCGCTTCTAGAG

E. Chemikalien

NEB T7-RNA-Polymerase von NEB (M0251S)

- 50 units/µl T7-RNA-Polymerase
- 50 mM Tris−HCl
- 100 mM NaCl
- 20 mM β-Mercaptoethanol
- 1 mM EDTA
- 50 % Glycerol
- 0.1 % Triton®X-100
- pH 7.9 bei 25 °C

10-fach NEB-Transkriptions-Puffer von NEB (B9012S)

- 400 mM Tris−HCl
- 60 mM $MgCl_2$
- 100 mM DTT[1]
- 20 mM Spermidin
- pH 7.9 bei 25 °C

10-fach Tris-Puffer selbst hergestellt

- 400 mM Tris−HCl
- 400 mM MgCl
- Mit Salzsäure bei 37 °C auf pH 7.1 eingestellt

NEB Ribonukleotidlösung von NEB (N0466L)

- 20 mM Tris−HCl
- 10 mM DTT
- 80 mM von jedem Nukleotidtriphosphat (ATP, GTP, CTP und UTP) als Natriumsalz
- pH 7.8

[1] Dithiothreitol

epicentre Ribonukleotidlösung von epicentre (RN02825)

- 100 mM ATP, GTP, CTP oder UTP in getrennten Tubes
- Mit NaOH auf pH 7.0 eingestellt

epicentre T7-RNA-Polymerase von epicentre (TM910K)

- 200 units/µl T7-RNA-Polymerase
- 50 mM Tris−HCl
- 100 mM NaCl
- 1 mM DTT
- 0.1 mM EDTA
- 50 % Glycerol
- 0.1 % Triton®X-100
- pH 7.5

RNase H von life technologies (AM2292)

- 10 units/µl[2] RNase H
- 100 mM NaCl
- 1 mM DTT
- 0.1 mM EDTA
- 50 % Glycerol
- 0.1 % Triton®X-100

100-fach Pyrophosphataselösung selbst hergestellt

- 100 µg ml^{-1} inorganicshe Pyrophosphatase (Sigma-Aldrich I1891-100UN)
- 20 mM Tris−HCl
- 1 mM $MgCl_2$
- 50 % Glycerol
- Mit Salzsäure bei 25 °C auf pH 7.2 eingestellt

LB-Medium von Carl Roth (X968.1)

- 10 g l^{-1} Trypton

[2]Ein unit ist die Menge an Enzym, die nötig ist, um die Fluoreszenz um den Faktor 1.5 pro Sekunde bei 37 °C zu erhöhen, wenn 20 pmol RNaseAlert® mit 1000 pmol komplementärem Oligonukleotid als Substrat verwendet wird.

- $5\,\mathrm{g\,l^{-1}}$ Hefeextrakt
- $10\,\mathrm{g\,l^{-1}}$ NaCl
- pH 7.0

5-fach M9 Minimalsalze von Sigma-Aldrich (M6030)

- $33.9\,\mathrm{g\,l^{-1}}$ Dinatriumhydrogenphosphat
- $15\,\mathrm{g\,l^{-1}}$ Kaliumdihydrogenphosphat
- $5\,\mathrm{g\,l^{-1}}$ Ammoniumchlorid
- $2.5\,\mathrm{g\,l^{-1}}$ NaCl

M9 Minimalmedium selbst hergestellt

- 20 % 5-fach M9 Minimalsalze
- 20 mM Glukose
- $300\,\mathrm{\mu g\,ml^{-1}}$ Thiamin (Sigma-Aldrich T1270)

F. Algorithmen

F.1. Medianfilter

F.1.1. Median

Ein Median ist der Wert eines Datensatzes, der den Datensatz in zwei Hälften teilt. Das heißt, mindestens 50% der Datenpunkte sind größer oder gleich groß wie der Median und mindestens 50% der Datenpunkte sind kleiner oder gleich groß wie der Median. Der Median wird auch 0.5-Quantil genannt, da er den Datensatz bei 50% „auseinanderschneidet“.

F.1.2. Funktionsweise

Ein Medianfilter ist ein nichtlinearer Filter, bei dem jeder Datenpunkt durch den Median seiner Umgebung ersetzt wird. Die Größe des Umgebungsfensters bestimmt die Stärke des Filters.

F.1.3. Beispiel

Man nehme an, dass man einen Datensatz mit einem starken Ausreißer hat. Durch einen Medianfilter kann man den Ausreißer entfernen, ohne die restlichen Daten stark zu verfälschen. In dem Beispiel wird ein Umgebungsfenster von insgesamt drei Datenpunkten angenommen:

Tabelle F.1.: Beispieldaten für den Medianfilter

Originaldaten	5	4	2	**24**	3	1	2
gefiltert	4.5	4	4	3	3	2	1.5
Fehler	0.5	0	2		0	1	0.5

F.1.4. Eigenschaften

- Wenn das globale Signal-zu-Rausch-Verhältnis klein ist, werden Kanten in den Daten sehr wenig abgeschwächt[176].

- Kurze Ausreißer in Datensätzen beeinflussen die Umgebung viel weniger, als das bei linearen Filtern der Fall ist. Kurz heißt hier weniger als das halbe Umgebungsfenster.
- Durch seine nichtlineare Natur wird das gefilterte Signal nicht „glatt“ im differentialistischen Sinne. So ist die Ableitung eines median gefilterten Signals meist nicht verwendbar.

F.1.5. Algorithmuseigenschaften

- An den Enden des Datensatzes sollen die fehlenden Punkte einfach ignoriert werden. Das heißt, das Umgebungsfenster wird kleiner und unsymmetrisch. Dies produziert Artefakte, wie jede Entscheidung am Rand, ist aber unabhängig vom Absolutwert und reduziert die Filterstärke nur um ein Mindestmaß.
- Wenn der Datensatz n Punkte lang ist und der Filter ein Fenster der Größe k hat, soll er einen Speicherverbrauch von $\mathcal{O}(n) + \mathcal{O}(k)$ und eine Zeitabhängigkeit von $\mathcal{O}(n \cdot k)$ haben[1].

F.1.6. Algorithmus

Die Hauptidee hinter dem Algorithmus ist, dass viele Berechnungsschritte, die für die Berechnung des Medians für einen Datenpunkt nötig sind, für seine Nachbarpunkte identisch sind. So kann man den Median leicht aus einer sortierten Liste auslesen, da es der Wert ist, der sich in der Mitte der Liste befindet. Die Listen, die sich aus dem Umgebungsfenster ergeben, unterscheiden sich zwischen zwei Nachbarn aber nur um einen Datenpunkt, der herausfällt, und einen, der hinzukommt. Einen Datenpunkt aus einer Liste zu entfernen ist trivial und einen Datenpunkt in eine schon sortiert Liste einzufügen kann leicht in linearer Zeit erfolgen: man beginnt am Anfang der Liste und geht so lange einen Datenpunkt weiter, bis der einzufügende Datenpunkt größer als der gerade Angesehene ist (oder man am Ende der Liste ist, dann muss man den Datenpunkt einfach am Ende einfügen). Wenn man die Liste als binären Baum speichert, könnte man diese Suche des richtigen Platzes für den Datenpunkt in $\mathcal{O}(\log(k))$ durchführen. Ein möglicher Kandidat wäre ein „black and red tree“ (BRT), der garantiert, dass der Baum gut genug balanciert ist[177]. Da k nicht sehr groß ist und ein BRT programmiertechnisch komplex ist, wurde die lineare Suche implementiert.

Wenn man nun von einem Datenpunkt zum nächsten geht, muss man auch noch wissen, welcher Datenpunkt aus dem Umgebungsfenster entfernt werden muss. Deswegen muss in der Liste nicht nur die aufsteigende Sortierung nach den Werten, sondern auch die aufsteigende Sortierung nach der Reihenfolge, in der die Datenpunkte in die Liste kommen, gespeichert werden. Man hat also eine doppelt verlinkte Liste.

[1] Das theoretische Optimum ist ein Zeitverbrauch von $\mathcal{O}(n \cdot \log(k))$.

Da bei einer verlinkten Liste, im Gegensatz zu einem Array, die Auslese eines Elementes nach dem Index nicht in konstanter Zeit erfolgen kann[2], muss die Liste noch erweitert werden, damit der Median, der in der Mitte der Liste ist, in konstanter Zeit ausgelesen werden kann. Dazu wird beim Aufbauen und Verändern der Liste immer gespeichert, welcher Datenpunkt gerade der Median ist. Zusätzlich wird bei jedem Datenpunkt auch noch gespeichert, ob er sich vor oder hinter dem Median in der sortierten Verlinkung befindet. Wenn nun ein Datenpunkt aus der Liste entfernt wird, kann in konstanter Zeit entschieden werden, ob der Median einen Datenpunkt nach vorne oder hinten wandert oder sich nicht bewegt. Beim Einfügen eines Datenpunktes ist das äquivalent zu handhaben.

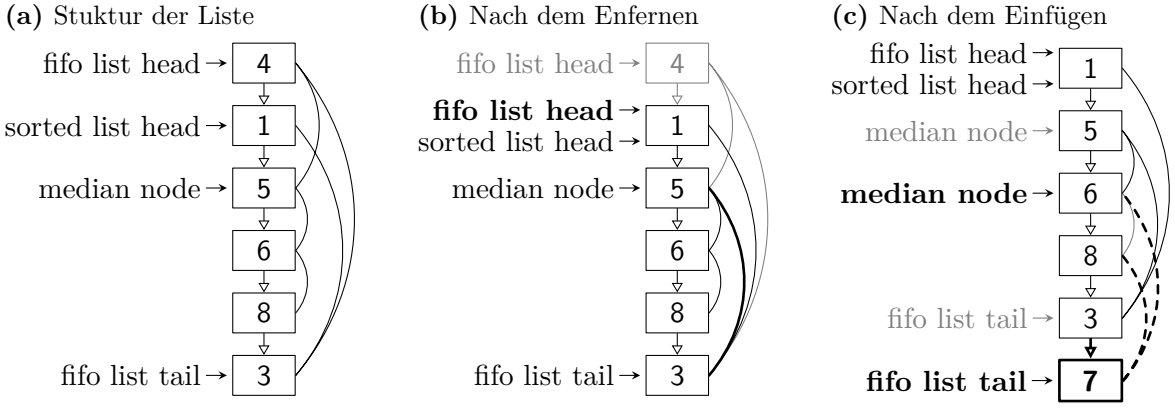


Abbildung F.1.: Illustration eines Schritts im Medianfilter. **(a)** Die horizontale Reihenfolge ist die Verlinkung, die die Verlinkung durch die Reihenfolge des Vorkommens repräsentiert. Die gebogenen Verbindungslinien auf der rechten Seite repräsentieren die sortierte Verlinkung. Da die Liste eine gerade Anzahl von Datenpunkten enthält, zeigt der Medianpfeil nicht auf den realen Medianwert, sondern auf den Datenpunkt, der direkt darüber liegt. Der Median ist dann der Wert zwischen diesem Datenpunkt und dem Punkt direkt davor. **(b)** Der Datenpunkt, der nicht mehr im Umgebungsfenster ist, wird entfernt. Änderungen in der Liste sind fett markiert und können in konstanter Zeit errechnet werden. Da die Liste nun eine ungerade Anzahl von Elementen enthält, zeigt der Medianpfeil jetzt auch auf den Wert des Medians. **(c)** Der Datenpunkt, der neu in das Umgebungsfenster hinzukommt, wird in die Liste eingefügt. Die gestrichelte Änderung muss in linearer Zeit errechnet werden. Die Fetten sind dann wieder in konstanter Zeit errechenbar.

[2]Eine Erweiterung des BRT kann in $\mathcal{O}(\log(k))$ auf den Index zugreifen.

F.1.7. median1d.m

```
function filtered = median1d(data, windowSize, dim)
%MEDIAN1D Performs a median filter
%   this function does a similar task than medfilt1 BUT it has
%   a less insane handling of the data borders.
%   Additional it runs faster (depending on data and window
%   size between 10 an 100-fold). It also consumes less memory.
%
%       FILTERED = median1d(DATA, WINDOWSIZE) filters the DATA
%           with the given WINDOWSIZE.
%       FILTERED = median1d(..., dim) spezifies the dimension
%       that should be filtered. Default is the first non
%       singleton dimension.
%
%   See also MEDFILT1, MEDIAN1D_CPP

    if (windowSize <= 1)
        filtered = data;
    elseif (nargin < 3)
        [data, nshifts] = shiftdim(data);

        filtered = shiftdim( ...
            Filter.median1d_cpp(data, windowSize), ...
            -nshifts ...
        );
    else
        ndim = ndims(data);
        assert(isscalar(dim) && dim > 0 && dim <= ndim, ...
            'Filter:median1d:InvalidDimensions');
        perm = [dim, 1:(dim-1), (dim+1):ndim];
        filtered = ipermute( ...
            Filter.median1d_cpp( ...
                permute(data, perm), ...
                windowSize ...
            ), ...
            perm ...
        );
    end
end
```

F.1.8. median_cpp.cpp

```
#include "mex.h"
#include "median1d/List.cpp"

void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{
  double *window, *data, *filtered;
  int windowSize[2], dataSize[2];

  /* Check for proper number of arguments. */
  if(nrhs != 2){
    mexErrMsgIdAndTxt(
        "MATLAB:Filter:median1d:invalidNumInputs",
        "Two input arguments required."
    );
  }
  else if(nlhs > 1){
    mexErrMsgIdAndTxt(
        "MATLAB:Filter:median1d:maxlhs",
        "Too many output arguments."
    );
  }

  /* The input must be a noncomplex double.*/
  if(
    !mxIsDouble(prhs[0]) || mxIsComplex(prhs[0]) ||
    !mxIsDouble(prhs[1]) || mxIsComplex(prhs[1])
  ){
    mexErrMsgIdAndTxt(
        "MATLAB:Filter:median1d:inputNotRealDouble",
        "Input must be a noncomplex double."
    );
  }

  dataSize[0] = (int) mxGetM(prhs[0]);
  dataSize[1] = (int) mxGetN(prhs[0]);
  windowSize[0] = (int) mxGetM(prhs[1]);
  windowSize[1] = (int) mxGetN(prhs[1]);

  if(
    windowSize[0] != 1 ||
    windowSize[1] != 1
  ){
    mexErrMsgIdAndTxt(
        "MATLAB:Filter:median1d:windowSizeNoScalar",
        "Window input must be a scalar."
    );
  }
```

```
    /* Create matrix for the return argument. */
    plhs[0] = mxCreateDoubleMatrix(dataSize[0], dataSize[1], mxREAL);
    mxSetDimensions(plhs[0], mxGetDimensions(prhs[0]),
        mxGetNumberOfDimensions(prhs[0]));

    /* Assign pointers to each input and output. */
    data = mxGetPr(prhs[0]);
    window = mxGetPr(prhs[1]);
    filtered = mxGetPr(plhs[0]);

    long windowS = (long) *window;
    if (windowS <= 0){
        windowS = 1;
    }

    /* Call the MedianFilter::filter subroutine. */
    if (dataSize[0] != 1 && dataSize[1] != 1){
        int i;
        for (i = 0; i < dataSize[1]; i++){
            MedianFilter::filter(
                &data[i * dataSize[0]],
                &filtered[i * dataSize[0]],
                dataSize[0],
                windowS
            );
        }
    }
    else {
        int size = dataSize[0] > dataSize[1]? dataSize[0]: dataSize[1];
        MedianFilter::filter(
                data, filtered, size,
                windowS
        );
    }
}
```

F.1.9. median1d/List.h

```
namespace MedianFilter{
    class Node{
        public:
            // enum values for the relative position to the median node
            static signed char const BEFORE_MEDIAN = -1;
            static signed char const IS_MEDIAN = 0;
            static signed char const AFTER_MEDIAN = 1;
        private:
            /**
             * *private* constructor with the value of the node
             * Only List can create new nodes.
             *
             * @param double value
             */
            Node(double value);
            // the nodes value
            double value;
            // the relative position to the median node
            signed char position;
            // the next node in the FIFO list
            Node *next;
            // the next node in the sorted list
            Node *sortedNext;
            // the previous node in the sorted list
            Node *sortedPrev;

        friend class List;
    };

    class List{
        private:
            // the head node of the FIFO list
            Node *head;
            // the tail node of the FIFO list
            Node *tail;
            // the head node of the sorted list
            Node *sortedHead;
            // the median node
            Node *medianNode;
            // size of the list
            unsigned long size;

            /**
             * performs the shift operation
             *
             * @return Node the shifted node
             */
            Node *internalShift();
```

```
50              /**
                 * performs the push operation
                 *
                 * @param Node newNode
                 */
55              void internalPush(Node *newNode);

            public:
                // constructor
                List();
60              // destructor
                ~List();

                /**
                 * calculates the median from the median node.
65               * This is fast.
                 *
                 * @return double
                 */
                double getMedian();
70
                /**
                 * calculates the median from the entire list.
                 * This is slow.
                 *
75               * @return double
                 */
                double getMedian2();

                /**
80               * pushes a value to the list.
                 *
                 * @param double value
                 */
                void push(double value);
85
                /**
                 * shifts the first node of the list and pushes
                 * a value to the list.
                 * This is slightly faster than calling shift() and
90               * then push().
                 *
                 * @param double value
                 */
                void pushAndShift(double value);
95
                /**
                 * shifts the first node of the list.
                 */
                void shift();
100
```

```
            /**
             * Debug function to inspect the list and some of
             * its properties.
             */
105         void outputInfo();

            /**
             * Getter for the lists size.
             *
110          * @retrun long
             */
            long getSize();
    };

115 /**
     * performs the filtering
     *
     * @param double* data pointer to the data array.
     * @param double* pointer to the filtered data array.
120  *  The two data arrays have to have the same size.
     * @param long dataSize the size of the data array.
     *        IMPORTANT: there is no validation for this value!
     * @param long windowSize the size of the filter window
     */
125 void filter(double* data, double* filtered, long dataSize, long
        windowSize);
}
```

F.1.10. median1d/List.cpp

```
#include "iostream"
#include "List.h"
#ifndef NULL
#define NULL 0
#endif

using namespace MedianFilter;

Node::Node(double value):
        next(NULL),
        sortedNext(NULL),
        sortedPrev(NULL),
        value(value),
        position(0)
        {};

List::List():
    head(NULL),
    tail(NULL),
    sortedHead(NULL),
    medianNode(NULL),
    size(0)
    {};

List::~List(){
    Node *current;
    // free all nodes
    current = head;
    while(current != NULL){
        Node *c = current;
        current = current->next;
        delete c;
    }
}

double List::getMedian2(){
    if (size == 0){
        return 0;
    }
    if (size == 1){
        return sortedHead->value;
    }
    Node *current;
    current = sortedHead;
    unsigned int i = 0;
    while (i < size){
        current = current->sortedNext;
        i += 2;
    }
```

```
50      if (i == size){
            return (
                current->value + current->sortedPrev->value
            ) / 2.;
        }
55      else {
            return current->sortedPrev->value;
        }
    }

60  double List::getMedian(){
        if (size % 2 == 0){
            return (
                medianNode->value + medianNode->sortedPrev->value
            ) / 2.;
65      }
        else {
            return medianNode->value;
        }
    }
70
    void List::internalPush(Node* newNode){
        double value = newNode->value;

        size++;
75
        if (size == 1){
            // if the new node is the first in the list
            head = newNode;
            tail = newNode;
80          medianNode = newNode;
            sortedHead = newNode;
            newNode->position = 0;
        }
        else {
85          // append to the FIFO linking
            tail->next = newNode;
            tail = newNode;

            // find right position in sorted linking
90          Node *current;
            current = sortedHead;

            bool run = true;
            while (run){
95              if (current->value >= value){
                    // found correct position

                    if (current->sortedPrev == NULL){
                        // if current is the sortedHead redirect
100                     // to the new node
```

```
                    sortedHead = newNode;
                }
                else {
                    current->sortedPrev->sortedNext = newNode;
                }
                newNode->sortedPrev = current->sortedPrev;
                newNode->sortedNext = current;
                current->sortedPrev = newNode;

                // set the relative position to the median node
                // for the new node
                if (current->position == Node::AFTER_MEDIAN){
                    newNode->position = Node::AFTER_MEDIAN;
                }
                else {
                    newNode->position = Node::BEFORE_MEDIAN;
                }
                run = false;
            }
            else if (current->sortedNext == NULL){
                // there was no node with a bigger value
                // --> append
                current->sortedNext = newNode;
                newNode->sortedPrev = current;
                newNode->position = Node::AFTER_MEDIAN;
                run = false;
            }

            current = current->sortedNext;
        }
    }

    // move median node if neccessary
    switch (newNode->position){
        case Node::BEFORE_MEDIAN:
            if (size % 2 == 1){
                medianNode->position = Node::AFTER_MEDIAN;
                medianNode = medianNode->sortedPrev;
                medianNode->position = Node::IS_MEDIAN;
            }
            break;
        case Node::AFTER_MEDIAN:
            if (size % 2 == 0){
                medianNode->position = Node::BEFORE_MEDIAN;
                medianNode = medianNode->sortedNext;
                medianNode->position = Node::IS_MEDIAN;
            }
            break;
    }
}
```

```
Node *List::internalShift(){
    Node *toRemove = head;

    head = toRemove->next;

    if (toRemove->sortedNext != NULL){
        toRemove->sortedNext->sortedPrev = toRemove->sortedPrev;
    }
    if (toRemove->sortedPrev != NULL){
        toRemove->sortedPrev->sortedNext = toRemove->sortedNext;
    }
    else {
        sortedHead = toRemove->sortedNext;
    }
    size--;

    // move median node if neccessary
    switch (toRemove->position){
        case Node::BEFORE_MEDIAN:
            if (size % 2 == 0){
                medianNode->position = Node::BEFORE_MEDIAN;
                medianNode = medianNode->sortedNext;
                medianNode->position = Node::IS_MEDIAN;
            }
            break;
        case Node::IS_MEDIAN:
            if (size % 2 == 0){
                medianNode = toRemove->sortedNext;
                medianNode->position = Node::IS_MEDIAN;
            }
            else {
                medianNode = toRemove->sortedPrev;
                medianNode->position = Node::IS_MEDIAN;
            }
            break;
        case Node::AFTER_MEDIAN:
            if (size % 2 == 1){
                medianNode->position = Node::AFTER_MEDIAN;
                medianNode = medianNode->sortedPrev;
                medianNode->position = Node::IS_MEDIAN;
            }
            break;
    }

    return toRemove;
}


void List::push(double value){
    Node *newNode = new Node(value);
    internalPush(newNode);
```

```
}
void List::pushAndShift(double value){
    Node *newNode = internalShift();
    // recycle the shifted Node
    newNode->value = value;
    newNode->next = NULL;
    newNode->sortedNext = NULL;
    newNode->sortedPrev = NULL;
    newNode->position = 0;
    internalPush(newNode);
}
void List::shift(){
    Node *toRemove = internalShift();
    delete toRemove;
}

void List::outputInfo(){
    std::cout << std::endl << "Size: " << size << std::endl;
    if (size != 0){
        std::cout << "Main chain: ";
        Node *current = head;
        while (current != NULL){
            std::cout << current->value << ' ';
            current = current->next;
        }
        std::cout << std::endl << "Sorted chain: ";

        current = sortedHead;
        while (current != NULL){
            if (current == medianNode){
                std::cout << ">" << current->value << "< ";
            }
            else {
                std::cout << current->value << ' ';
            }
            current = current->sortedNext;
        }
        std::cout << std::endl
            << "Median:  " << getMedian()  << std::endl
            << "Median2: " << getMedian2()
            << std::endl << std::endl;
    }

}

long List::getSize(){
    return size;
}


void MedianFilter::filter(
```

```
    double* data,
    double* filtered,
    long dataSize,
    long windowSize
){
    // double* filtered = new double[dataSize];
    // number of data points on the left of the current
    // data point
    long leftWindow = windowSize / 2;
    // number of data points on the right of the current
    // data point BUT including the data point itself
    long rightWindow = windowSize - leftWindow;
    List list;

    if (dataSize < leftWindow){
        // if the the filter window will allways include
        // all data points

        // fill list
        for (long i = 0; i < dataSize; i++){
            list.push(data[i]);
        }

        // get median
        double median = list.getMedian();

        // assign this value to all filtered data points
        for (long i = 0; i < dataSize; i++){
            filtered[i] = median;
        }
    }
    else if (dataSize < windowSize){
        // if the filter window will always hit one of
        // the borders

        // initialise list
        for (long i = 0; i < rightWindow - 1; i++){
            list.push(data[i]);
        }

        // until filter window hits the end
        for (long i = 0; i < dataSize - rightWindow + 1; i++){
            list.push(data[i + rightWindow - 1]);
            filtered[i] = list.getMedian();
        }

        // until filter window leaves the beginning
        for (long i = dataSize - rightWindow + 1; i < leftWindow + 1; i
           ++){
            filtered[i] = list.getMedian();
        }
```

```
305             // until the end of the data
                for (long i = leftWindow + 1; i < dataSize; i++){
                    list.shift();
                    filtered[i] = list.getMedian();
                }
310
        }
        else {
            // if the filter window fits into the data completely

315             // initialise list
                for (long i = 0; i < rightWindow - 1; i++){
                    list.push(data[i]);
                }

320             // until filter window leaves the beginning
                for (long i = 0; i < leftWindow + 1; i++){
                    list.push(data[i + rightWindow - 1]);
                    filtered[i] = list.getMedian();
                }
325
                // until filter window hits the end
                for (long i = leftWindow + 1; i < dataSize - rightWindow + 1; i
                   ++){
                    list.pushAndShift(data[i + rightWindow - 1]);
                    filtered[i] = list.getMedian();
330             }

                //until the end of the data
                for (long i = dataSize - rightWindow + 1; i < dataSize; i++){
                    list.shift();
335                 filtered[i] = list.getMedian();
                }
        }
    }
```

F.2. Gaußfilter

Um Daten zu glätten wurde öfters ein Gaußfilter verwendet, der eine diskrete normierte Faltung des Rohdatensignals mit einer Gaußfunktion darstellt:

$$x'_i = \frac{\sum_{k=1}^{i-1} x_k \cdot w_{i-k} + x_i \cdot w_0 + \sum_{k=i+1}^{n} x_k \cdot w_{k-i}}{\sum_{k=1}^{i-1} w_{i-k} + w_0 + \sum_{k=i+1}^{n} w_{k-i}} \tag{F.2.1}$$

$$w_{\sigma,j} = e^{-\frac{(j \cdot \delta t)^2}{2\sigma^2}} \tag{F.2.2}$$

F.2.1. gauss.m

```
function filtered = gauss(data, sigma, dim)
%GAUSS performs a filter with a symmetric gaussian filter
%   The data is filtered by a weighted moving average. The
%   weighting function is a gaussian with width sigma.
%
%   FILTERED = gauss(DATA)
%       filters the DATA with SIGMA = 3
%   FILTERED = gauss(DATA, SIGMA)
%       filters the DATA with SIGMA
%   FILTERED = gauss(DATA, SIGMA, DIM)
%       filters along dimension DIM

    if (nargin < 2)
        % default value for sigma
        sigma = 3;
    end


    if (nargin < 3)
        % no DIM argument provided -> shift the matrix to the
        % first non singular dimension
        [data, nshifts] = shiftdim(data);

        filtered = shiftdim( ...
            performGauss(data, sigma), ...
            -nshifts ...
        );
    else
        ndim = ndims(data);
        assert(isscalar(dim) && dim > 0 && dim <= ndim, ...
            'Filter:gauss:InvalidDimensions');
        perm = [dim, 1:(dim-1), (dim+1):ndim];
        filtered = ipermute( ...
            performGauss( ...
                permute(data, perm), ...
                sigma ...
```

```
            ), ...
            perm ...
        );
    end
end

function filtered = performGauss(data, sigma)
% the actual filter function
    if (sigma == 0)
        % no filtering necessary
        filtered = data;
    else
        threeSigma = ceil(3*sigma);
        g = exp(-(-threeSigma:threeSigma).^2/2/sigma^2);

        filtered = zeros(size(data));
        for i = 1:size(data, 2)
            % perform the convolution for every column
            filtered(:, i) = conv(data(:, i), g, 'same') ./ ...
                conv(ones(size(data(:, i))), g, 'same');
        end
    end

end
```

F.2.2. gauss2d.m

```
function filtered = gauss2D(data, sigma, dim)
%GAUSS2D performs a filter with a symmetric gaussian in 2D
%   The data is filtered by a weighted moving average.
%   The weighting function is a  two dimensonal gaussian
%   with width sigma.
%
%   FILTERED = gauss2D(DATA)
%       filters the DATA with SIGMA = 3
%   FILTERED = gauss2D(DATA, SIGMA)
%       filters the DATA with SIGMA
%   FILTERED = gauss2D(DATA, SIGMA, DIM)
%       filters along dimension DIM
%
%   SEE ALSO: Filter.gauss

    if (nargin < 2 || numel(sigma) < 1)
        % default value for sigma
        sigma = [3 3];
    elseif numel(sigma) < 2
        % if only one sigma is provided
        sigma = [sigma sigma];
    elseif numel(sigma) > 2
        % if too many sigmas are provided
```

```
        sigma = sigma(1:2);
    end

    if (nargin < 3)
        % no DIM argument provided -> shift the matrix to the
        % first non singular dimension
        [data, nshifts] = shiftdim(data);

        filtered = shiftdim( ...
            performGauss(data, sigma), ...
            -nshifts ...
        );
    else
        ndim = ndims(data);
        assert(isscalar(dim) && dim > 0 && dim <= ndim, ...
            'Filter:gauss2D:InvalidDimensions');
        if (dim == 1)
            filtered = performGauss(data, sigma);
        else
            perm = [dim, 1:(dim-1), (dim+1):ndim];
            filtered = ipermute( ...
                performGauss( ...
                    permute(data, perm), ...
                    sigma ...
                ), ...
                perm ...
            );
        end
    end

    function filtered = performGauss(data, sigma)
    % the actual filter function
        if (all(sigma == 0))
            % no filtering necessary
            filtered = data;
        else
            threeSigma = ceil(3*sigma);
            g1 = exp(-(-threeSigma(1):threeSigma(1)).^2/2/sigma(1)^2);
            g2 = exp(-(-threeSigma(2):threeSigma(2)).^2/2/sigma(2)^2);

            % perform the 2D convolution
            filtered = conv2(g1, g2, data, 'same') ./ conv2(g1, g2, ones
                (size(data)), 'same');
        end
    end
end
```

F.3. DetectSteps

Wenn eine Datenreihe relativ konstant sein sollte, kann eine abrupte Änderung – eine Stufe – ein Problem oder einen Fehler anzeigen (s. Abb. F.2). Um diese zu registrieren, benötigt man einen Erkennungsalgorithmus:

1. Zuerst werden die Daten durch einen Medianfilter (s. F.1) mit einer Filterfestergröße $2n + 1$[3] geglättet.
2. Dann wird der Absolutwert der diskreten Ableitung der gefilterten Daten berechnet und die ersten n und die letzten n Datenpunkte entfernt, da der Medianfilter an den Rändern Artefakte, die die Detektion beeinträchtigen können, erzeugt.
3. Stufen werden dann über Datenpunkte mit signifikant größerer Ableitung detektiert. Heuristisch wird hier der fünffache Wert des 90 %-Quantils als Schwellwert verwendet. Wenn ein Datenpunkt also eine Ableitung hat, die fünf mal größer ist als die Maximalableitung der kleinsten 90 %, ist an dieser Stelle eine Stufe.

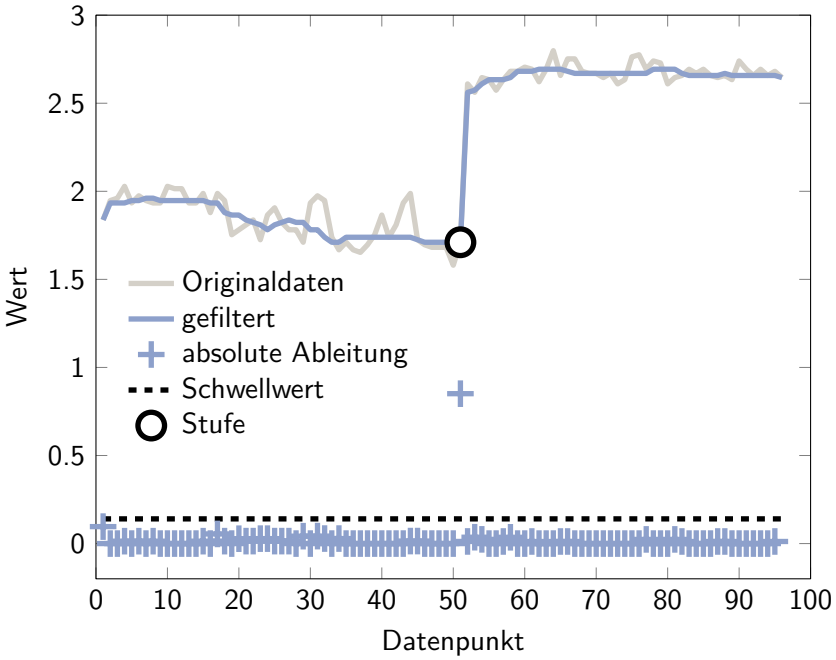


Abbildung F.2.: Die Daten (graue Linie) werden gefiltert (blaue Linie) und der Absolutwert der diskreten Ableitung (blaue Kreuze) berechnet. Jeder Ableitungspunkt, der größer ist als fünf mal die 90 % der ganzen Population (gestrichelte schwarze Linie) wird als Stufe detektiert (Kreise).

[3] Meistens wird $n = 5$ verwendet.

F.3.1. detectSteps.m

```
function steps = detectSteps(data, varargin)
%DETECTSTEPS performs a step detection
%
% STEPS = detectSteps(DATA)
%   detects the steps in the DATA vector and stores the
%   positions of the steps in the vector STEPS.
% STEPS = detectSteps(..., paramKey, paramValue)
%
% Parameter:
%   medianLength  (3)     : the median filter half window size
%   comparePoint  (0.9)   : the compare quantile
%   compareFactor (5)     : the compare factor
%   debug         (false): flag for debug mode

    %% get input parameter
    p = inputParser;
    p.addParamValue('medianLength', 3, @(x)x==round(x)&&x>0);
    p.addParamValue('comparePoint', 0.9, @isnumeric);
    p.addParamValue('compareFactor', 5, @isnumeric);
    p.addParamValue('debug', false, @islogical);
    p.parse(varargin{:});

    % the filter length
    medianLength = p.Results.medianLength;
    % the point in the population to compare with
    % (i.e. the quantile)
    comparePoint = p.Results.comparePoint;
    % the factor by which the compare point has to be
    % mulipied to get the threshold
    compareFactor = p.Results.compareFactor;

    %% data too short
    if (numel(data) < 2*medianLength + 1)
        steps = [];
        return
    end

    %% validate data input
    assert( ...
        isvector(data), ...
        'detectSteps:dataNoVector', ...
        'Data has to be a vector.' ...
    );

    %% filter data, calculate derivative & generate threshold
    filteredData = medfilt1(data, 2*medianLength + 1);
    dFilteredData = abs(diff(filteredData));

    % remove artefacts from median filtering
```

```
50      trimmedDFilteredData = ...
            dFilteredData(medianLength:(end-medianLength));
        % calculate threshold
        compareValue = ...
            quantile(trimmedDFilteredData, comparePoint) * ...
55          compareFactor;
        % get step positions
        steps = find(trimmedDFilteredData > compareValue) ...
            + medianLength - 1;

60      %% debug output
        if (p.Results.debug)
            fprintf(['Parameter:' ...
                '\n\tmedian window size: %d' ...
                '\n\tcompare point: %.2f' ...
65              '\n\tcompare factor: %.2f' ...
                '\n' ...
                '\n\tcompare value: %.2f\n' ], ...
                2*medianLength + 1, ...
                comparePoint, ...
70              compareFactor, ...
                compareValue...
            );
            dataLength = numel(data);
            plot(1:dataLength, data, '-', ...
75              1:dataLength, filteredData, '-', ...
                1:(dataLength - 1), dFilteredData, '-', ...
                [1 dataLength], [compareValue compareValue], '--', ...
                steps, data(steps), '+' ...
            );
80      end
    end
```

F.4. Fill

F.4.1. fill.cpp

```
#include "mex.h"
#include <algorithm>
#include <math.h>
#include <iostream>
#include <queue>
#define addToQueue(x, y) if ( \
    !floodedImage[(x) * height + (y)] && \
    !image[(x) * height + (y)] \
){ \
    floodedImage[(x) * height + (y)] = 1;\
    xQueue.push(x);\
    yQueue.push(y);\
}

/*
 * fill.cpp
 */

void floodFill(
    const mxLogical *image,
    mxLogical *floodedImage,
    const unsigned int width,
    const unsigned int height,
    unsigned int x,
    unsigned int y,
    unsigned int &minX,
    unsigned int &maxX,
    unsigned int &minY,
    unsigned int &maxY,
    unsigned int &count
)
{
    std::queue<unsigned int> xQueue;
    std::queue<unsigned int> yQueue;

    addToQueue(x, y);

    while (xQueue.size()){
        x = xQueue.front();
        xQueue.pop();
        y = yQueue.front();
        yQueue.pop();
        count += 1;

        if (x < minX){
            minX = x;
        }
```

```
        if (x > maxX){
            maxX = x;
        }
        if (y < minY){
            minY = y;
        }
        if (y > maxY){
            maxY = y;
        }

        if (x > 0){
            addToQueue(x - 1, y);
            if (y > 0){
                addToQueue(x - 1, y - 1);
            }
            if (y < height - 1){
                addToQueue(x - 1, y + 1);
            }
        }
        if (y > 0){
            addToQueue(x, y - 1);
        }
        if (y < height - 1){
            addToQueue(x, y + 1);
        }
        if (x < width - 1){
            addToQueue(x + 1, y);
            if (y > 0){
                addToQueue(x + 1, y - 1);
            }
            if (y < height - 1){
                addToQueue(x + 1, y + 1);
            }
        }
    }
}

void copyFilling(
    const mxLogical *src,
    mxLogical *dest,
    const unsigned int width,
    const unsigned int height,
    const unsigned int minX,
    const unsigned int maxX,
    const unsigned int minY,
    const unsigned int maxY
)
{
    for (unsigned int x = minX; x <= maxX; x += 1){
        for (unsigned int y = minY; y <= maxY; y += 1){
            unsigned int idx = x * height + y;
```

```
                dest[idx] = src[idx] | dest[idx];
            }
        }
    }
    void clearFilling(
        mxLogical *image,
        const unsigned int width,
        const unsigned int height,
        const unsigned int minX,
        const unsigned int maxX,
        const unsigned int minY,
        const unsigned int maxY
    )
    {
        for (unsigned int x = minX; x <= maxX; x += 1){
            for (unsigned int y = minY; y <= maxY; y += 1){
                unsigned int idx = x * height + y;
                image[idx] = 0;
            }
        }
    }

    void mexFunction( int nlhs, mxArray *plhs[],
                      int nrhs, const mxArray *prhs[] )
    {
        unsigned int maxHoleSize;
        mxLogical *image, *fillImage, *outImage;
        mwSize w, h;

        if (nrhs == 0){
            mexErrMsgIdAndTxt( "MATLAB:fill:wrongInputCount",
                    "At least one input (image) expected.");
        }
        if (nrhs == 1 || mxGetM(prhs[1]) * mxGetN(prhs[1]) == 0){
            maxHoleSize = 5;
        }
        else {
            maxHoleSize = (unsigned int) mxGetScalar(prhs[1]);
        }

        if (nrhs > 2){
            mexErrMsgIdAndTxt( "MATLAB:fill:wrongInputCount",
                    "Too many input parameter.");
        }

        /* The input must be a logical.*/
        if(!mxIsLogical(prhs[0])){
            mexErrMsgIdAndTxt( "MATLAB:fill:inputNotLogical",
                    "Input must be a logical.");
        }
```

```
    h = mxGetM(prhs[0]);
    w = mxGetN(prhs[0]);
    image = mxGetLogicals(mxDuplicateArray(prhs[0]));

    /* Create matrix for the return argument. */
    plhs[0] = mxDuplicateArray(prhs[0]);
    outImage = mxGetLogicals(plhs[0]);

    fillImage = mxGetLogicals(mxCreateLogicalMatrix(h, w));

    for (unsigned int x = 0; x < w; x += 1){
        unsigned int x_ = x * h;
        for (unsigned int y = 0; y < h; y += 1){
            unsigned int idx = x_ + y;

            if (!image[idx]){
                unsigned int
                    minX = w,
                    maxX = 0,
                    minY = h,
                    maxY = 0,
                    count = 0;

                floodFill(
                    image, fillImage,
                    w, h, x, y,
                    minX, maxX, minY, maxY,
                    count
                );

                if (count <= maxHoleSize){
                    copyFilling(
                        fillImage, outImage,
                        w, h,
                        minX, maxX, minY, maxY
                    );
                }
                copyFilling(
                    fillImage, image,
                    w, h,
                    minX, maxX, minY, maxY
                );
                clearFilling(
                    fillImage,
                    w, h,
                    minX, maxX, minY, maxY
                );
            }
        }
    }
}
```

F.5. RemoveDeadEnds

F.5.1. removeDeadEnds.cpp

```
#include "mex.h"
#include <algorithm>
#include <math.h>
#include <iostream>
#define image(x, y) image[(x) * height + (y)]

/*
 * fill.cpp
 */

void checkDeadEnd(
    mxLogical *image,
    const unsigned int width,
    const unsigned int height,
    const unsigned int x,
    const unsigned int y
)
{
    if (!image(x, y)){
        unsigned int neighbors = 0;
        bool border = false;
        if (x > 0){
            neighbors += !image(x - 1, y + 0);
            if (y > 0){
                neighbors += !image(x - 1, y - 1);
            }
            else {
                border = true;
            }
            if (y < height - 1){
                neighbors += !image(x - 1, y + 1);
            }
            else {
                border = true;
            }
        }
        else {
            border = true;
        }

        if (y > 0){
            neighbors += !image(x - 0, y - 1);
        }
        else {
            border = true;
        }
        if (y < height - 1){
```

```
            neighbors += !image(x - 0, y + 1);
        }
        else {
            border = true;
        }

        if (x < width - 1){
            neighbors += !image(x + 1, y + 0);
            if (y > 0){
                neighbors += !image(x + 1, y - 1);
            }
            else {
                border = true;
            }
            if (y < height - 1){
                neighbors += !image(x + 1, y + 1);
            }
            else {
                border = true;
            }
        }
        else {
            border = true;
        }


        // std::cout << x << "/" << y << " > " << neighbors << std::endl
            ;

        if ((!border && neighbors < 2) || (border && neighbors < 2)){
            image(x, y) = 1;
            if (x > 0){
                checkDeadEnd(image, width, height, x - 1, y + 0);
                if (y > 0){
                    checkDeadEnd(image, width, height, x - 1, y - 1);
                }
                if (y < height - 1){
                    checkDeadEnd(image, width, height, x - 1, y + 1);
                }
            }

            if (y > 0){
                checkDeadEnd(image, width, height, x + 0, y - 1);
            }
            if (y < height - 1){
                checkDeadEnd(image, width, height, x + 0, y + 1);
            }

            if (x < width - 1){
                checkDeadEnd(image, width, height, x + 1, y + 0);
                if (y > 0){
```

```
                    checkDeadEnd(image, width, height, x + 1, y - 1);
                }
                if (y < height - 1){
                    checkDeadEnd(image, width, height, x + 1, y + 1);
                }
            }
        }
    }
}

void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{
    mxLogical *image;
    mwSize w, h;

    if (nrhs != 1){
        mexErrMsgIdAndTxt( "MATLAB:fill:wrongInputCount",
                "One input (image) expected.");
    }

    /* The input must be a logical.*/
    if(!mxIsLogical(prhs[0])){
        mexErrMsgIdAndTxt( "MATLAB:fill:inputNotLogical",
                "Input must be a logical.");
    }

    h = mxGetM(prhs[0]);
    w = mxGetN(prhs[0]);

    /* Create matrix for the return argument. */
    plhs[0] = mxDuplicateArray(prhs[0]);
    image = mxGetLogicals(plhs[0]);

    for (unsigned int x = 0; x < w; x += 1){
        for (unsigned int y = 0; y < h; y += 1){
            checkDeadEnd(image, w, h, x, y);
        }
    }

}
```

F.6. Bridge

F.6.1. bridge.cpp

```
#include "mex.h"
#define isChange(l, m, r) (l ^ r) | (m & !l)

/*
 * bridge.cpp
 */


void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{
    mxLogical *image, *outImage;
    mwSize width, height;

    if (nrhs != 1){
        mexErrMsgIdAndTxt( "MATLAB:bridge:wrongInputCount",
                "One input (image) expected.");
    }

    /* The input must be a logical.*/
    if(!mxIsLogical(prhs[0])){
        mexErrMsgIdAndTxt( "MATLAB:bridge:inputNotLogical",
                "Input must be a logical.");
    }

    height = mxGetM(prhs[0]);
    width = mxGetN(prhs[0]);
    image = mxGetLogicals(prhs[0]);

    /* Create matrix for the return argument. */
    plhs[0] = mxDuplicateArray(prhs[0]);
    outImage = mxGetLogicals(plhs[0]);

    for (int x = 2; x < width - 2; x++){
        int x_ = x * height;
        for (int y = 2; y < height - 2; y++){
            int idx = x_ + y;
            if (!outImage[idx]){
                mxLogical a, a1, b, c, c1, d, e, e1, f, g, g1, h;
                /*
                 * the neighbourhood of the current pixel (indicated by
                 * an X) look as follows:
                 *       a1
                 *    h  a  b
                 * g1 g X c c1
                 *    f  e  d
                 *       e1
```

```
                 * a, c, e and g are a combined with a1, c1, e1 and g1
                 * in a second check round. This makes the algorithm a
50               * TWO pixel bridge algorithm.
                 */
                a  = image[idx - 1];
                a1 = a|image[idx - 2];

55              b  = image[idx - 1 + height];

                c  = image[idx + height];
                c1 = c|image[idx + 2*height];

60              d  = image[idx + 1 + height];

                e  = image[idx + 1];
                e1 = e|image[idx + 2];

65              f  = image[idx + 1 - height];

                g  = image[idx - height];
                g1 = g|image[idx - 2*height];

70              h  = image[idx - 1 - height];

                /*
                 * The bridge has to be build if there is a color change
                 * on two opposit corner of the 3x3 matrix shown above.
75               *
                 * A color change is detected as the following patterns
                 * (the X indicates the center pixel a hyphen ("-")
                 * indicates either "0" or "1":
                 *  0 -  1 -  0 1
80               *  X 1  X 0  X 0
                 *
                 * Therefore the "not color change" patterns are:
                 *  0 0  1 1  1 0
                 *  X 0  X 1  X 1
85               */
                outImage[idx] = (mxLogical)
                    (
                        (
                            (
90                              (isChange(a, b, c)) &
                                (isChange(e, f, g))
                            ) |
                            (
                                (isChange(c, d, e)) &
95                              (isChange(g, h, a))
                            )
                        ) &
                        !(
```

```
                                (a & !f & (a == c) & (e == g) & (a != g)) |
                                (e & !b & (a == c) & (e == g) & (a != g)) |
                                (c & !h & (c == e) & (g == a) & (a != c)) |
                                (g & !d & (c == e) & (g == a) & (a != c))
                            )

                        ) |
                        (
                            (a || b || c || d || e || f || g || h) &
                            (
                                (
                                    (isChange(a1, b, c1)) &
                                    (isChange(e1, f, g1))
                                ) |
                                (
                                    (isChange(c1, d, e1)) &
                                    (isChange(g1, h, a1))
                                )
                            ) &
                            !(
                                (a1 & !f & (a1 == c1) & (e1 == g1) & (a1 !=
                                    g1)) |
                                (e1 & !b & (a1 == c1) & (e1 == g1) & (a1 !=
                                    g1)) |
                                (c1 & !h & (c1 == e1) & (g1 == a1) & (a1 !=
                                    c1)) |
                                (g1 & !d & (c1 == e1) & (g1 == a1) & (a1 !=
                                    c1))
                            )

                        );
                    if (outImage[idx]){
                        if (a1){
                            outImage[idx - 1] = 1;
                        }
                        if (c1){
                            outImage[idx + height] = 1;
                        }
                        if (e1){
                            outImage[idx + 1] = 1;
                        }
                        if (g1){
                            outImage[idx - height] = 1;
                        }
                    }
                }
            }
        }
    }
```

F.7. BorderFill

F.7.1. borderFill.cpp

```
#include "mex.h"
#include "borderFill/List.cpp"
#define idx(dx, dy) (x + dx) * h + y + dy
#define check(dx, dy){\
    idx = idx(dx, dy);\
    if ((image[idx] == image[idx0]) && !outImage[idx]){\
        outImage[idx] = 1;\
        pixelList.push(x + dx, y + dy);\
    }\
}

/*
 * borderFill.cpp
 */


void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{
    mxLogical *image, *outImage;
    mwSize w, h;
    borderFill::List pixelList;

    if (nrhs != 1){
        mexErrMsgIdAndTxt( "MATLAB:borderFill:wrongInputCount",
                "One input (image) expected.");
    }

    /* The input must be a logical.*/
    if(!mxIsLogical(prhs[0])){
        mexErrMsgIdAndTxt( "MATLAB:borderFill:inputNotLogical",
                "Input must be a logical.");
    }

    h = mxGetM(prhs[0]);
    w = mxGetN(prhs[0]);
    image = mxGetLogicals(prhs[0]);

    /* Create matrix for the return argument. */
    plhs[0] = mxCreateLogicalMatrix(h, w);
    outImage = mxGetLogicals(plhs[0]);



    /* first and last column */
    for (int y = 0; y < h; y++){
        outImage[y] = 1;
```

```
        pixelList.push(0, y);
        outImage[h * (w - 1) + y] = 1;
        pixelList.push(w - 1, y);
    }

    /* first and last row */
    for (int x = 1; x < w - 1; x++){
        outImage[x * h] = 1;
        pixelList.push(x, 0);
        outImage[x * h + h - 1] = 1;
        pixelList.push(x, h - 1);
    }

    while (pixelList.head){
        int x = pixelList.head->x;
        int y = pixelList.head->y;
        int idx0 = idx(0, 0);
        int idx;
        pixelList.shift();

        if (y > 0){
            if (x > 0){
                check(-1, -1);
            }
            check(0, -1);
            if (x < w - 1){
                check(1, -1);
            }
        }
        if (x > 0){
            check(-1, 0);
        }
        if (x < w - 1){
            check(1, 0);
        }
        if (y < h - 1){
            if (x > 0){
                check(-1, 1);
            }
            check(0, 1);
            if (x < w - 1){
                check(1, 1);
            }
        }
    }
}
```

F.7.2. borderFill/List.h

```
1  namespace borderFill{
       class Node{
           public:
               int x;
5              int y;
           private:
               /**
                * *private* constructor with the value of the node
                * Only List can create new nodes.
10              *
                * @param double value
                */
               Node(int x, int y);
               // the next node in the FIFO list
15             Node *next;

           friend class List;
       };

20     class List{
           public:
               // the head node of the FIFO list
               Node *head;
           private:
25             // the tail node of the FIFO list
               Node *tail;

               /**
                * performs the shift operation
30              *
                * @return Node the shifted node
                */
               Node *internalShift();

35             /**
                * performs the push operation
                *
                * @param Node newNode
                */
40             void internalPush(Node *newNode);

           public:
               // constructor
               List();
45             // destructor
               ~List();

               /**
                * pushes a value to the list.
```

```
50              *
                * @param double value
                */
               void push(int x, int y);

55             /**
                * shifts the first node of the list.
                */
               void shift();
    };
60 }
```

F.7.3. borderFill/List.cpp

```
 1 #include "List.h"
   #ifndef NULL
   #define NULL 0
   #endif
 5
   using namespace borderFill;

   Node::Node(int x, int y):
           next(NULL),
10         x(x),
           y(y)
           {};

   List::List():
15     head(NULL),
       tail(NULL)
       {};

   List::~List(){
20     Node *current;
       // free all nodes
       current = head;
       while(current != NULL){
           Node *c = current;
25         current = current->next;
           delete c;
       }
   }

30 void List::internalPush(Node* newNode){
       if (head == NULL){
           // if the new node is the first in the list
           head = newNode;
           tail = newNode;
35     }
       else {
```

```
        // append to the FIFO linking
        tail->next = newNode;
        tail = newNode;
    }
}

Node *List::internalShift(){
    Node *toRemove = head;

    head = toRemove->next;

    return toRemove;
}


void List::push(int x, int y){
    Node *newNode = new Node(x, y);
    internalPush(newNode);
}
void List::shift(){
    Node *toRemove = internalShift();
    delete toRemove;
}
```

F.8. GetPerimeter

Um den Umfang einer Region zu bestimmen, wird der Mittelpunk jeder Pixelkante, die zwischen einem weißen und einem schwarzen Pixel liegt, mit dem Mittelpunkt der direkt benachbarten Pixelkanten, die auch an einem Wechsel liegen, verbunden (s. Abb. F.3). Da diese Kontur den Umfang etwas zu hoch abschätzt, muss noch ein Korrekturfaktor von $\frac{\pi}{8} \cdot \left(1 + \sqrt{2}\right)$[178] multipliziert werden.

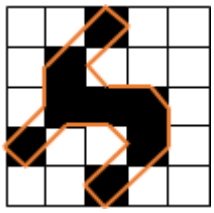


Abbildung F.3.: Illustration der „Mid Crack“-Methode.

Um den Algorithmus zu testen, wurden Kreise mit verschiedenen Radien gezeichnet und der gemessene Umfang p mit dem errechnetem verglichen:

$$\Delta p_{\text{rel}} = 1 - \frac{p}{2\pi \cdot r} \tag{F.8.1}$$

In Abb. F.4 sieht man, dass der Algorithmus gute Ergebnisse liefert. So ist der Maximalfehler bei kleinen Radien nur 6 % und bei großen Radien konvergiert er zu Null.

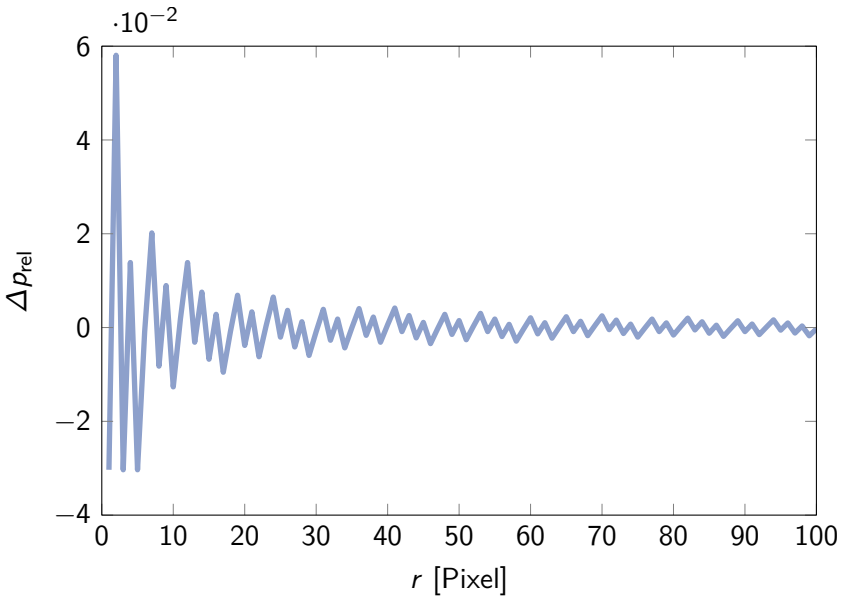


Abbildung F.4.: Relativer Fehler der Umfangabschätzung.

F.8.1. getPerimeter.cpp

```
#include "mex.h"
//#include "math.h"
#define idx(dx, dy) (x + dx) * h + y + dy
#define img(dx, dy) image[idx(dx, dy)]
#define DIAG p = p + d
#define ALONG p = p + 1

/*
 * getPerimeter.cpp
 */


void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{
    mxLogical *image;
    mwSize w, h;
    /* the perimeter */
    double p;

    /* size of a diagonal connection*/
    double d;

    p = 0;
    d = 0.70710678118654752440084436210485;//1/sqrt(2.0);

    if (nrhs != 1){
        mexErrMsgIdAndTxt( "MATLAB:getPerimeter:wrongInputCount",
                "One input (image) expected.");
    }

    h = mxGetM(prhs[0]);
    w = mxGetN(prhs[0]);

    if (h < 2 || w < 2){

        plhs[0] = mxCreateDoubleScalar(0);
        return;
        mexWarnMsgIdAndTxt( "MATLAB:getPerimeter:inputNotMatrix",
                "Input must be a matrix not a vector or scalar.");
    }

    /* The input must be a logical.*/
    if (!mxIsLogical(prhs[0])){
        image = mxGetLogicals(mxCreateLogicalMatrix(h, w));
        double *data = mxGetPr(prhs[0]);
        for (int x = 0; x < w; x += 1){
            for (int y = 0; y < h; y += 1){
                image[x * h + y] = (mxLogical) data[x * h + y];
```

```
            }
        }
//          mxArray *rhs[1];
//          mexCallMATLAB(1, rhs, 1, prhs, "logical");
//          image = mxGetLogicals(rhs[0]);
//      mexErrMsgIdAndTxt( "MATLAB:getPerimeter:inputNotLogical",
//                "Input must be a logical.");
    }
    else {
        image = mxGetLogicals(prhs[0]);
    }

    /* corners */
    if (image[0]){
        DIAG;
        DIAG;
        if (image[h]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[1] & !image[h + 1]){
                ALONG;
            }
            else {
                DIAG;
            }
        }
        if (image[1]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[h] & !image[h + 1]){
                ALONG;
            }
            else {
                DIAG;
            }
        }
    }
    if (image[h - 1]){
        DIAG;
        DIAG;
        if (image[h - 2]){
            ALONG;
        }
        else {
            DIAG;
```

```
            DIAG;
            if (image[h + h - 1] & !image[h + h - 2]){
                ALONG;
            }
            else {
                DIAG;
            }
        }

        if (image[h + h - 1]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[h - 2] & !image[h + h - 2]){
                ALONG;
            }
            else {
                DIAG;
            }
        }
    }
    if (image[(w - 1) * h]){
        DIAG;
        DIAG;
        if (image[(w - 2) * h]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[(w - 1) * h + 1] & !image[(w - 2) * h + 1]){
                ALONG;
            }
            else {
                DIAG;
            }
        }

        if (image[(w - 1) * h + 1]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[(w - 2) * h] & !image[(w - 2) * h + 1]){
                ALONG;
            }
            else {
                DIAG;
```

```
            }
        }
    }
    if (image[w*h - 1]){
        DIAG;
        DIAG;
        if (image[w*h - 2]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[w*h - 1 - h] & !image[(w - 2) * h + h - 2]){
                ALONG;
            }
            else {
                DIAG;
            }
        }

        if (image[w*h - 1 - h]){
            ALONG;
        }
        else {
            DIAG;
            DIAG;
            if (image[w*h - 2] & !image[(w - 2) * h + h - 2]){
                ALONG;
            }
            else {
                DIAG;
            }
        }
    }

    /* first row */
    for (int x = 1, y = 0; x < w - 1; x++){
        if (img(0, 0)){
//              if (!img(0, -1)){
                if (img(-1, 0)/* & !img(-1, -1)*/){
                    ALONG;
                }
                else {
                   DIAG;
                }

                if (img(1, 0)/* & !img(1, -1)*/){
                    ALONG;
                }
                else {
                    DIAG;
```

```
                }
//              }

            if (!img(0, 1)){
                if (img(-1, 0) & !img(-1, 1)){
                    ALONG;
                }
                else {
                    DIAG;
                }

                if (img(1, 0) & !img(1, 1)){
                    ALONG;
                }
                else {
                    DIAG;
                }
            }

            if (!img(-1, 0)){
                /*if (img(0, -1) & !img(-1, -1)){
                    ALONG;
                }
                else */{
                    DIAG;
                }

                if (img(0, 1) & !img(-1, 1)){
                    ALONG;
                }
                else {
                    DIAG;
                }
            }

            if (!img(1, 0)){
                /*if (img(0, -1) & !img(1, -1)){
                    ALONG;
                }
                else */{
                    DIAG;
                }

                if (img(0, 1) & !img(1, 1)){
                    ALONG;
                }
                else {
                    DIAG;
                }
            }
        }
```

```
    }

    /* last row */
    for (int x = 1, y = h - 1; x < w - 1; x++){
        if (img(0, 0)){
            if (!img(0, -1)){
                if (img(-1, 0) & !img(-1, -1)){
                    ALONG;
                }
                else {
                   DIAG;
                }

                if (img(1, 0) & !img(1, -1)){
                    ALONG;
                }
                else {
                    DIAG;
                }
            }

//              if (!img(0, 1)){
                if (img(-1, 0)/* & !img(-1, 1)*/){
                    ALONG;
                }
                else {
                    DIAG;
                }

                if (img(1, 0)/* & !img(1, 1)*/){
                    ALONG;
                }
                else {
                    DIAG;
                }
//              }

            if (!img(-1, 0)){
                if (img(0, -1) & !img(-1, -1)){
                    ALONG;
                }
                else {
                    DIAG;
                }

                /*if (img(0, 1) & !img(-1, 1)){
                    ALONG;
                }
                else */{
                    DIAG;
                }
```

```
305                 }

                    if (!img(1, 0)){
                        if (img(0, -1) & !img(1, -1)){
                            ALONG;
310                     }
                        else {
                            DIAG;
                        }

315                     /*if (img(0, 1) & !img(1, 1)){
                            ALONG;
                        }
                        else */{
                            DIAG;
320                     }
                    }
                }
            }

325         /* first column */
            for (int x = 0, y = 1; y < h - 1; y++){
                if (img(0, 0)){
                    if (!img(0, -1)){
                        /*if (img(-1, 0) & !img(-1, -1)){
330                         ALONG;
                        }
                        else */{
                           DIAG;
                        }
335
                        if (img(1, 0) & !img(1, -1)){
                            ALONG;
                        }
                        else {
340                         DIAG;
                        }
                    }

                    if (!img(0, 1)){
345                     /*if (img(-1, 0) & !img(-1, 1)){
                            ALONG;
                        }
                        else */{
                            DIAG;
350                     }

                        if (img(1, 0) & !img(1, 1)){
                            ALONG;
                        }
355                     else {
```

```
                    DIAG;
                }
            }

//              if (!img(-1, 0)){
                if (img(0, -1)/* & !img(-1, -1)*/){
                    ALONG;
                }
                else {
                    DIAG;
                }

                if (img(0, 1)/* & !img(-1, 1)*/){
                    ALONG;
                }
                else {
                    DIAG;
                }
//              }

            if (!img(1, 0)){
                if (img(0, -1) & !img(1, -1)){
                    ALONG;
                }
                else {
                    DIAG;
                }

                if (img(0, 1) & !img(1, 1)){
                    ALONG;
                }
                else {
                    DIAG;
                }
            }
        }
    }

    /* last column */
    for (int x = w - 1, y = 1; y < h - 1; y++){
        if (img(0, 0)){
            if (!img(0, -1)){
                if (img(-1, 0) & !img(-1, -1)){
                    ALONG;
                }
                else {
                   DIAG;
                }

                /*if (img(1, 0) & !img(1, -1)){
                    ALONG;
```

```
                    }
                    else */{
                        DIAG;
                    }
                }

                if (!img(0, 1)){
                    if (img(-1, 0) & !img(-1, 1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }

                    /*if (img(1, 0) & !img(1, 1)){
                        ALONG;
                    }
                    else */{
                        DIAG;
                    }
                }

                if (!img(-1, 0)){
                    if (img(0, -1) & !img(-1, -1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }

                    if (img(0, 1) & !img(-1, 1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }
                }

//                  if (!img(1, 0)){
                    if (img(0, -1)/* & !img(1, -1)*/){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }

                    if (img(0, 1)/* & !img(1, 1)*/){
                        ALONG;
                    }
                    else {
                        DIAG;
```

```
                }
//              }
        }
    }

    /* inner matrix */
    for (int x = 1; x < w - 1; x++){
        for (int y = 1; y < h - 1; y++){
            if (img(0, 0)){
                if (!img(0, -1)){
                    if (img(-1, 0) & !img(-1, -1)){
                        ALONG;
                    }
                    else {
                       DIAG;
                    }

                    if (img(1, 0) & !img(1, -1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }
                }

                if (!img(0, 1)){
                    if (img(-1, 0) & !img(-1, 1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }

                    if (img(1, 0) & !img(1, 1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }
                }

                if (!img(-1, 0)){
                    if (img(0, -1) & !img(-1, -1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }

                    if (img(0, 1) & !img(-1, 1)){
                        ALONG;
```

```
                    }
                    else {
                        DIAG;
                    }
                }

                if (!img(1, 0)){
                    if (img(0, -1) & !img(1, -1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }

                    if (img(0, 1) & !img(1, 1)){
                        ALONG;
                    }
                    else {
                        DIAG;
                    }
                }
            }
        }
    }


    /* Create matrix for the return argument. */
    /* very edge is counted twice therefore p has to be divided by 2 */
    /* the correction factor taken from Z. Kulpa, Area and perimeter
     * measurement of blobs in discrete binary pictures. Comput. Graph.
     * Image Process, 6:434-451, 1977,
     * doi:10.1016/s0146-664X(77)80021-x */
//      plhs[0] = mxCreateDoubleMatrix(1, 1, mxREAL);
//      mxGetPr(plhs[0])[0] = p/2 * 0.948059448968519935684815546667 52;
    //pi/8*(1+sqrt(2))
    plhs[0] = mxCreateDoubleScalar(p/2 *
        0.948059448968519935684815546667 52);
}
```

F.9. Track

F.9.1. track.m

```
function assignment = track(tracker, p1, r1, p2, r2)
%TRACKER.TRACK tracks droplets between frames
%
%   ASSIGMENT = TRACKER.TRACK(P1, R1, P2, R2)
%       Droplets with positions P1 (DIM1x2) and radii R1
%       (DIM1x1) will be tracked on droplets with positions
%       P2 (DIM2x2) and radii R2 (DIM2x2). The assigment
%       matrix ASSIGMENTS (Nx2) contains all N assigments
%       where the first column contains the indices of the
%       first droplet set and the second column the indices
%       of the second set. The relation N <= min(DIM1, DIM2)
%       always hold true.

    dim1 = numel(r1);
    dim2 = numel(r2);

    if (dim1 == 0 || dim2 == 0)
        % one droplet set contains no droplets
        assignment = zeros(0, 2);
        return
    end

    % reshape input to proper dimensions
    x1 = reshape(p1(:, 1), dim1, 1);
    y1 = reshape(p1(:, 2), dim1, 1);
    r1 = reshape(r1, dim1, 1);
    x2 = reshape(p2(:, 1), 1, dim2);
    y2 = reshape(p2(:, 2), 1, dim2);
    r2 = reshape(r2, 1, dim2);

    % vectors for matrix generation
    ones1 = ones(dim1, 1);
    ones2 = ones(1, dim2);

    % create radii matrices
    r1 = r1 * ones2;
    r2 = ones1 * r2;

    % calculate distances
    dists2 = ...
        (x1 * ones2 - ones1 * x2).^2 + ...
        (y1 * ones2 - ones1 * y2).^2;

    % calculate radii distances and sums
    dr = abs(r1 - r2);
    sr = r1 + r2 + tracker.maxBorderDistance;
```

```
    % filter too big distances and too big radii changes
    distFilter = ...
        dists2 <= sr.^2 & ...
        dr < max( ...
            tracker.minMaxRadiusChange, ...
            tracker.maxRadiusChange * r1 ...
        );

    % create metric
    dists2 = dists2(distFilter) + (2*dr(distFilter)).^2;
    indexTranslator = find(distFilter);

    if (isempty(dists2))
        % no droplet pair holds the threshold constrains
        assignment = zeros(0, 2);
        return;
    end

    % reshape matrix for sorting
    dists2 = reshape(dists2, [], 1);
    % sort distances
    [~, sortedIdx] = sort(dists2);
    % get sorted index pairs
    [sortedIdx1, sortedIdx2] = ind2sub2D( ...
        [dim1, dim2], ...
        indexTranslator(sortedIdx) ...
    );

    % find double indices
    doubleIdx = orUnique([sortedIdx1, sortedIdx2]);
    uniqueIdx = doubleIdx == 0;
    % return only the unique
    assignment = ...
        [sortedIdx1(uniqueIdx), sortedIdx2(uniqueIdx)];

end
```

F.9.2. orUnique.m

```
%ORUNIQUE searches for double entries in a matrix
%   ASSIGNMENT = orUnique(A) searches along the first dimension of A for
%   double entires in the remaining dimensions.
%   ASSIGNMENT is a vector with length size(A, 1).
%   It is zero for a "row" that is unique in A or is the first entry of
%   a double entry. Otherwise it contains the index of the first entry
%   of the doubles.
%
%   Example:
%       >> A = [1 2; 1 1; 2 2; 3 3]
%
%       A =
%
%            1     2
%            1     1
%            2     2
%            3     3
%
%       >> Ass = orUnique(A)
%
%       Ass =
%
%            0
%            1
%            1
%            0
```

F.9.3. orUnique.cpp

```
#include "mex.h"

void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] ){
    double *data, *indices;
    int dataSize[2];

    /* Check for proper number of arguments. */
    if(nrhs != 1) {
        mexErrMsgIdAndTxt( "MATLAB:DropletTracker:orUnique:
            invalidNumInputs",
                "One input argument required.");
    } else if(nlhs != 1) {
        mexErrMsgIdAndTxt( "MATLAB:DropletTracker:orUnique:maxlhs",
                "Only one and exactly one output argument supported.");
    }

    /* The input must be a noncomplex double.*/
    if(!mxIsDouble(prhs[0]) || mxIsComplex(prhs[0])) {
        mexErrMsgIdAndTxt( "MATLAB:DropletTracker:orUnique:
            inputNotRealDouble",
                "Input must be a noncomplex double.");
    }

    dataSize[0] = (int) mxGetM(prhs[0]);
    dataSize[1] = (int) mxGetN(prhs[0]);

    if (dataSize[1] == 0){
        dataSize[0] = 0;
    }

    /* Create matrix for the return argument. */
    plhs[0] = mxCreateDoubleMatrix(dataSize[0], 1, mxREAL);

    /* Assign pointers to each input and output. */
    data = mxGetPr(prhs[0]);
    indices = mxGetPr(plhs[0]);

    for (int y = 0; y < dataSize[0]; y++){
        double index = 0;
        for (int y_ = 0; y_ < y; y_++){
            if (indices[y_]){
                /* pair y_ is not unique itself */
                continue;
            }
            for (int i = 0; i < dataSize[1]; i++){
                /* iterate through all additional dimensions */
                if (data[i * dataSize[0] + y] == data[i * dataSize[0] +
                    y_]){
```

```
                        /* found one matching index*/
                        index = y_ + 1;
                        y_ = y; // stop loop on y_
                        break;
                    }
                }
            }
            indices[y] = index;
        }
    }
```

F.10. Ableiten diskreter Zeitserien

Der Übergang von kontinuierlichen zu diskreten Zeitintervallen ist bei jeder Messung notwendig, da man nicht unendlich viele Datenpunkte speichern kann.

F.10.1. Einfacher Ansatz

Prinzipiell ist die Ableitung diskreter Werte keine schwierige mathematische Sache. Aus der infinitesimalen Ableitung mit $dx \to 0$:

$$\begin{aligned} f'(x) &= \frac{df(x))}{dx} = \\ &= \frac{f(x+dx) - f(x)}{dx} = \\ &= \frac{f(x+dx) - f(x)}{dx} = \\ &= \frac{f(x) - f(x-dx)}{dx} = \\ &= \frac{f(x+dx) - f(x-dx)}{2 \cdot dx} \end{aligned} \tag{F.10.1}$$

wird das dx durch ein endliches Δx ersetzt:

$$\begin{aligned} f'(x) &= \frac{\Delta f(x))}{\Delta x} = \\ &= \frac{f(x+\Delta x) - f(x-\Delta x)}{2 \cdot \Delta x} \end{aligned} \tag{F.10.2}$$

Dabei ist zu beachten, dass bei endlichem Δx Vorwärts- und Rückwärtsableitung nicht gleich sind. Um das Ergebnis nicht um ein halbes Zeitintervall zu verschieben, wird das Mittel der Vorwärts- und Rückwärtsableitung verwendet. An den Enden der Datenreihe, wo nur eine Vorwärts- bzw. Rückwärtsableitung existiert, wird nur die existierende verwendet.

Für glatte Kurven – z. B. aus Simulationen – funktioniert das auch sehr gut, aber bei Realdaten sitzt auf dem Signal oft ein Rauschen, das die Ableitung – wegen seiner kleinen Zeitskala – dominiert oder wenigstens so stört, dass man die Ableitung des Signals nicht erkennen kann (s. Abb. F.5). Um dennoch mit der Ableitung arbeiten zu können, kann man entweder die Daten vor dem Ableiten glätten – z. B. über F.2 – oder man berechnet die Ableitung über einen anderen Ansatz.

F.10.2. Ansatz über lineare Regression

Ein Ansatz, der sehr gute Ergebnisse liefert, ist, dass man für jeden Datenpunkt ein Datenfenster der Größe $N \in 2\mathbb{N}+1$ verwendet[4]. In die Datenpunkte y_i zu den Zeitpunkten t_i wird jetzt eine Ausgleichsgerade gelegt:

$$y(t) = a \cdot t + b \tag{F.10.3}$$

Bevor man den Zusammenhang mit den real gemessenen Daten darstellen kann, muss man einige Vektoren und Matrizen definieren:

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \tag{F.10.4}$$

$$A = \begin{bmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_N & 1 \end{bmatrix} \tag{F.10.5}$$

$$\vec{\beta} = \begin{pmatrix} a \\ b \end{pmatrix} \tag{F.10.6}$$

$$\vec{\epsilon} = \begin{pmatrix} y_1 - y(t_1) \\ y_2 - y(t_2) \\ \vdots \\ y_N - y(t_n) \end{pmatrix} = \text{Rauschen} \tag{F.10.7}$$

Dadurch kann man den Zusammenhang zwischen y_i und t_i darstellen:

$$\vec{y} = A \cdot \vec{\beta} + \vec{\epsilon} \tag{F.10.8}$$

Die Ausgleichsgerade soll nun so gewählt werden, dass die absolute Länge von $\vec{\epsilon}$ minimal wird. Das ist äquivalent dazu, dass das Betragsquadrat $|\vec{\epsilon}|^2$ des Vektors minimiert wird.

$$|\vec{\epsilon}|^2 = \vec{\epsilon}'\vec{\epsilon} = \left(\vec{y} - A\vec{\beta}\right)' \left(\vec{y} - A\vec{\beta}\right) = \tag{F.10.9}$$

$$= \left(\vec{y}' - \vec{\beta}'A'\right)\left(\vec{y} - A\vec{\beta}\right) = \tag{F.10.10}$$

$$= \vec{y}'\vec{y} + \vec{\beta}'A'A\vec{\beta} - \vec{y}'A\vec{\beta} - \underbrace{\vec{\beta}'A'\vec{y}}_{=\left(\vec{y}'A\vec{\beta}\right)'=\vec{y}'A\vec{\beta}} = \tag{F.10.11}$$

$$= \vec{y}'\vec{y} + \vec{\beta}'A'A\vec{\beta} - 2 \cdot \vec{y}'A\vec{\beta} \tag{F.10.12}$$

[4]Die Randpunkte, bei denen das Fenster nicht mehr ganz mit Daten gefüllt werden kann, da sie am Rand liegen, werden nur mit den verfügbaren Punkten berechnet. So haben die Datenpunkte ganz am Rand ein Fenster mit einer Größe von nur $\frac{N-1}{2}+1$.

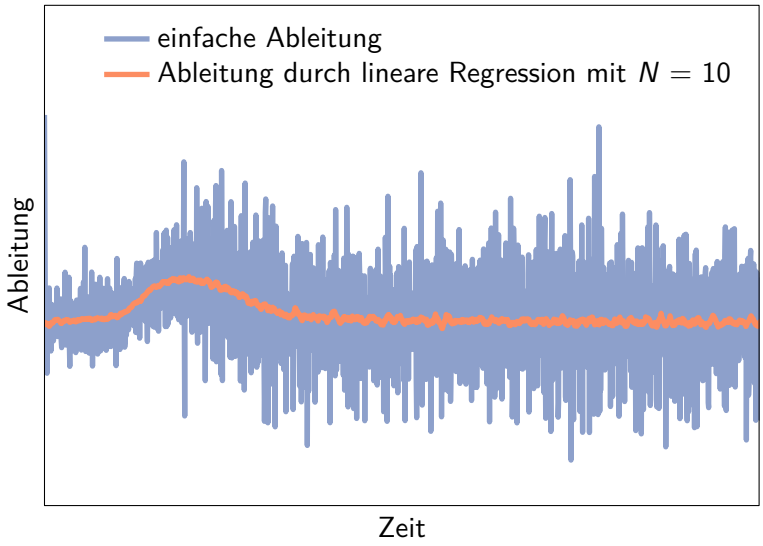


Abbildung F.5.: Vergleich der diskreten Ableitungstypen.

Die Minimierung wird über $\vec{\beta}$ ausgeführt. Deswegen muss die Ableitung von $|\vec{\epsilon}|^2$ nach $\vec{\beta}$ gleich Null sein:

$$\frac{\partial |\vec{\epsilon}|^2}{\partial \vec{\beta}} = -2\vec{y}A + 2\vec{\beta}'A'A \stackrel{!}{=} 0 \tag{F.10.13}$$

Somit ergibt sich als Zusammenhang für $\vec{\beta}$:

$$\vec{y}'A = \vec{\beta}'A'A \tag{F.10.14}$$

$$\vec{\beta} = A^{-1}\vec{y} \tag{F.10.15}$$

Wobei hier A^{-1} die pseudoinverse Matrix von A ist, die folgende Eigenschaften hat:

$$A \cdot A^{-1} \cdot A = A \tag{F.10.16}$$

$$A^{-1} \cdot A \cdot A^{-1} = A^{-1} \tag{F.10.17}$$

$$A \cdot A^{-1} \text{ ist hermitisch} \tag{F.10.18}$$

$$A^{-1} \cdot A \text{ ist hermitisch} \tag{F.10.19}$$

Somit erhält man für jeden Datenpunkt die Steigung der Ausgleichsgeraden und kann diese als Ableitung verwenden. Das Ergebniss ist natürlich davon abhängig, wie groß N gewählt wird, aber wie man in Abb. F.5 sehen kann, ist das Ergebnis in diesem Fall besser als das des einfachen Ansatzes.

Da der einfache Ansatz aber viel schneller berechnet werden kann, wird er immer dort verwendet, wo man mit seiner Ausgabe weiterarbeiten kann.

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[1] R. Hooke. *Micrographia, or, Some physiological descriptions of minute bodies made by magnifying glasses: with observations and inquiries thereupon*. Printed by Jo. Martyn und Ja. Allestry, printers to the Royal Society, 1665. DOI: 10.5962/bhl.title.904.

[2] C. Darwin. *On the origin of species*. D. Appleton und Co., 1871. DOI: 10.5962/bhl.title.28875.

[3] S. L. Miller und H. C. Urey. „Organic Compound Synthesis on the Primitive Earth: Several questions about the origin of life have been answered, but much remains to be studied.“ In: *Science* 130.3370 (Juli 1959), S. 245–251. ISSN: 1095-9203. DOI: 10.1126/science.130.3370.245.

[4] M. W. Powner, B. Gerland und J. D. Sutherland. „Synthesis of activated pyrimidine ribonucleotides in prebiotically plausible conditions“. In: *Nature* 459.7244 (Mai 2009), S. 239–242. ISSN: 1476-4687. DOI: 10.1038/nature08013.

[5] M. W. Powner, J. D. Sutherland und J. W. Szostak. „Chemoselective Multicomponent One-Pot Assembly of Purine Precursors in Water“. In: *Journal of the American Chemical Society* 133.11 (März 2011), S. 4149–4150. ISSN: 1520-5126. DOI: 10.1021/ja2002023.

[6] J. P. Ferris. „Montmorillonite catalysis of 30-50 mer oligonucleotides: laboratory demonstration of potential steps in the origin of the RNA world.“ In: *Orig Life Evol Biosph* 32.4 (Aug. 2002), S. 311–32. ISSN: 0169-6149. PMID: 12458736.

[7] J. W. Szostak. „The eightfold path to non-enzymatic RNA replication“. In: *Journal of Systems Chemistry* 3.1 (Feb. 2012), S. 2. ISSN: 1759-2208. DOI: 10.1186/1759-2208-3-2.

[8] P. L. Luisi. „Toward the engineering of minimal living cells“. In: *The Anatomical Record* 268.3 (Okt. 2002), S. 208–214. ISSN: 1097-0185. DOI: 10.1002/ar.10155.

[9] M. L. Simpson. „Cell-free synthetic biology: a bottom-up approach to discovery by design“. In: *Molecular Systems Biology* 2 (Dez. 2006). ISSN: 1744-4292. DOI: 10.1038/msb4100104.

[10] J. W. Szostak, D. P. Bartel und P. L. Luisi. „Synthesizing life“. In: *Nature* 409.6818 (Jan. 2001), S. 387–390. ISSN: 0028-0836. DOI: 10.1038/35053176.

[11] A. Pohorille und D. Deamer. „Artificial cells: prospects for biotechnology“. In: *Trends in Biotechnology* 20.3 (März 2002), S. 123–128. ISSN: 0167-7799. DOI: 10.1016/s0167-7799(02)01909-1.

[12] P. L. Luisi. „About Various Definitions of Life“. In: *Origins of Life and Evolution of the Biosphere* 28.4/6 (Okt. 1998), S. 613–622. ISSN: 0169-6149. DOI: 10.1023/a:1006517315105.

[13] T. Ganti. *The Principles of Life*. Hrsg. von E. Szathmary und J. Griesemer. Oxford University Press, Sep. 2003. ISBN: 978-0-198-50726-0. DOI: 10.1093/acprof:oso/9780198507260.001.0001.

[14] S. Rasmussen, M. A. Bedau, L. Chen u. a., Hrsg. *Protocells: Bridging Nonliving and Living Matter*. The MIT Press, Nov. 2008. ISBN: 0-262-18268-8.

[15] P. A. Bachmann, P. Walde, P. L. Luisi u. a. „Self-replicating reverse micelles and chemical autopoiesis“. In: *Journal of the American Chemical Society* 112.22 (Okt. 1990), S. 8200–8201. ISSN: 0002-7863. DOI: 10.1021/ja00178a073.

[16] J. Käs und E. Sackmann. „Shape transitions and shape stability of giant phospholipid vesicles in pure water induced by area-to-volume changes“. In: *Biophysical Journal* 60.4 (Okt. 1991), S. 825–844. ISSN: 0006-3495. DOI: 10.1016/s0006-3495(91)82117-8.

[17] K. Kurihara, M. Tamura, K.-i. Shohda u. a. „Self-reproduction of supramolecular giant vesicles combined with the amplification of encapsulated DNA“. In: *Nature Chemistry* 3.10 (Sep. 2011), S. 775–781. ISSN: 1755-4349. DOI: 10.1038/nchem.1127.

[18] Y. Sato, K. Yasuhara, J.-i. Kikuchi u. a. „Synthetic cell division system: Controlling equal vs. unequal divisions by design“. In: *Sci. Rep.* 3 (Dez. 2013). ISSN: 2045-2322. DOI: 10.1038/srep03475.

[19] L. Giomi und A. DeSimone. „Spontaneous Division and Motility in Active Nematic Droplets“. In: *Physical Review Letters* 112.14 (Apr. 2014). ISSN: 1079-7114. DOI: 10.1103/physrevlett.112.147802.

[20] A. Wochner, J. Attwater, A. Coulson u. a. „Ribozyme-Catalyzed Transcription of an Active Ribozyme“. In: *Science* 332.6026 (Apr. 2011), S. 209–212. ISSN: 1095-9203. DOI: 10.1126/science.1200752.

[21] K. Adamala und J. W. Szostak. „Nonenzymatic Template-Directed RNA Synthesis Inside Model Protocells“. In: *Science* 342.6162 (Nov. 2013), S. 1098–1100. ISSN: 1095-9203. DOI: 10.1126/science.1241888.

[22] F. C. Keber, E. Loiseau, T. Sanchez u. a. „Topology and dynamics of active nematic vesicles“. In: *Science* 345.6201 (Sep. 2014), S. 1135–1139. ISSN: 1095-9203. DOI: 10.1126/science.1254784.

[23] V. Noireaux und A. Libchaber. „A vesicle bioreactor as a step toward an artificial cell assembly“. In: *Proc. Natl. Acad. Sci. U.S.A.* 101.51 (Dez. 2004), S. 17669–17674. ISSN: 1091-6490. DOI: 10.1073/pnas.0408236101.

[24] M. M. Hanczyc, S. M. Fujikawa und J. W. Szostak. „Experimental models of primitive cellular compartments: encapsulation, growth, and division.“ In: *Science* 302.5645 (Okt. 2003), S. 618–22. ISSN: 1095-9203. PMID: 14576428.

[25] D. S. Tawfik und A. D. Griffiths. „Man-made cell-like compartments for molecular evolution“. In: *Nature Biotechnology* 16.7 (Juli 1998), S. 652–656. ISSN: 1087-0156. DOI: 10.1038/nbt0798-652.

[26] T. K. De und A. Maitra. „Solution behaviour of Aerosol OT in non-polar solvents“. In: *Advances in Colloid and Interface Science* 59 (Aug. 1995), S. 95–193. ISSN: 0001-8686. DOI: 10.1016/0001-8686(95)80005-n.

[27] A. B. Theberge, F. Courtois, Y. Schaerli u. a. „Microdroplets in Microfluidics: An Evolving Platform for Discoveries in Chemistry and Biology“. In: *Angewandte Chemie International Edition* 49.34 (Juni 2010), S. 5846–5868. ISSN: 1433-7851. DOI: 10.1002/anie.200906653.

[28] B. Rotman. „Measurement of activity of single molecules of beta-D-galactosidase.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 47 (Dez. 1961), S. 1981–91. ISSN: 0027-8424. PMID: 14038788.

[29] M. Nakano, J. Komatsu, S.-i. Matsuura u. a. „Single-molecule PCR using water-in-oil emulsion“. In: *Journal of Biotechnology* 102.2 (Apr. 2003), S. 117–124. ISSN: 0168-1656. DOI: 10.1016/s0168-1656(03)00023-3.

[30] O. J. Miller, K. Bernath, J. J. Agresti u. a. „Directed evolution by in vitro compartmentalization“. In: *Nature Methods* 3.7 (Juli 2006), S. 561–570. ISSN: 1548-7105. DOI: 10.1038/nmeth897.

[31] W.-C. Lu und A. D. Ellington. „In vitro selection of proteins via emulsion compartments“. In: *Methods* 60.1 (März 2013), S. 75–80. ISSN: 1046-2023. DOI: 10.1016/j.ymeth.2012.03.008.

[32] W. Demtröder. *Experimentalphysik 3*. Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-03911-9. DOI: 10.1007/978-3-642-03911-9.

[33] W. Heisenberg. „Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik“. In: *Zeitschrift für Physik* 43.3-4 (März 1927), S. 172–198. ISSN: 1434-601X. DOI: 10.1007/bf01397280.

[34] M. Geissbuehler, T. Spielmann, A. Formey u. a. „Triplet Imaging of Oxygen Consumption during the Contraction of a Single Smooth Muscle Cell (A7r5)“. In: *Biophysical Journal* 98.2 (Jan. 2010), S. 339–349. ISSN: 0006-3495. DOI: 10.1016/j.bpj.2009.10.006.

[35] T. Förster. „Zwischenmolekulare Energiewanderung und Fluoreszenz“. In: *Annalen der Physik* 437.1-2 (1948), S. 55–75. ISSN: 1521-3889. DOI: 10.1002/andp.19484370105.

[36] J. Boyle. „Molecular biology of the cell, 5th edition by B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter“. In: *Biochem. Mol. Biol. Educ.* 36.4 (Juli 2008), S. 317–318. ISSN: 1539-3429. DOI: 10.1002/bmb.20192.

[37] A. D. Hershey. „Independent functions of viral protein and nucleic acid in growth of bacteriophage“. In: *The Journal of General Physiology* 36.1 (Sep. 1952), S. 39–56. ISSN: 1540-7748. DOI: 10.1085/jgp.36.1.39.

[38] R. C. Lee, R. L. Feinbaum und V. Ambros. „The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14“. In: *Cell* 75.5 (Dez. 1993), S. 843–854. ISSN: 0092-8674. DOI: 10.1016/0092-8674(93)90529-y.

[39] I. Alvarez-Garcia und E. A. Miska. „MicroRNA functions in animal development and human disease“. In: *Development* 132.21 (Sep. 2005), S. 4653–4662. ISSN: 1477-9129. DOI: 10.1242/dev.02073.

[40] J. Lu, G. Getz, E. A. Miska u. a. „MicroRNA expression profiles classify human cancers“. In: *Nature* 435.7043 (Juni 2005), S. 834–838. ISSN: 1476-4679. DOI: 10.1038/nature03702.

[41] T. Mizuno, M. Y. Chou und M. Inouye. „A unique mechanism regulating gene expression: translational inhibition by a complementary RNA transcript (micRNA).“ In: *Proc. Natl. Acad. Sci. U.S.A.* 81.7 (Apr. 1984), S. 1966–70. ISSN: 0027-8424. PMID: 6201848.

[42] J. Tomizawa, T. Itoh, G. Selzer u. a. „Inhibition of ColE1 RNA primer formation by a plasmid-specified small RNA.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 78.3 (März 1981), S. 1421–5. ISSN: 0027-8424. PMID: 6165011.

[43] P. Stougaard, S. Molin und K. Nordström. „RNAs involved in copy-number control and incompatibility of plasmid R1.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 78.10 (Okt. 1981), S. 6008–12. ISSN: 0027-8424. PMID: 6171808.

[44] S. Brantl. „Antisense-RNA regulation and RNA interference“. In: *Biochimica et Biophysica Acta (BBA) - Gene Structure and Expression* 1575.1-3 (Mai 2002), S. 15–25. ISSN: 0167-4781. DOI: 10.1016/s0167-4781(02)00280-4.

[45] C. G. Hebert, J. J. Valdes und W. E. Bentley. „Beyond silencing—engineering applications of RNA interference and antisense technology for altering cellular phenotype“. In: *Current Opinion in Biotechnology* 19.5 (Okt. 2008), S. 500–505. ISSN: 0958-1669. DOI: 10.1016/j.copbio.2008.08.006.

[46] R. A. Sanders und W. Hiatt. „Tomato transgene structure and silencing“. In: *Nature Biotechnology* 23.3 (März 2005), S. 287–289. ISSN: 1087-0156. DOI: 10.1038/nbt0305-287b.

[47] J. SantaLucia und D. Hicks. „The thermodynamics of DNA structural motifs“. In: *Annu. Rev. Biophys. Biomol. Struct.* 33.1 (Juni 2004), S. 415–440. ISSN: 1545-4266. DOI: 10.1146/annurev.biophys.32.110601.141800.

[48] M. Petersheim und D. H. Turner. „Base-stacking and base-pairing contributions to helix stability: thermodynamics of double-helix formation with CCGG, CCGGp, CCGGAp, ACCGGp, CCGGUp, and ACCGGUp“. In: *Biochemistry* 22.2 (Jan. 1983), S. 256–263. ISSN: 1520-4995. DOI: 10.1021/bi00271a004.

[49] E. Protozanova, P. Yakovchuk und M. D. Frank-Kamenetskii. „Stacked-Unstacked Equilibrium at the Nick Site of DNA“. In: *Journal of Molecular Biology* 342.3 (Sep. 2004), S. 775–785. ISSN: 0022-2836. DOI: 10.1016/j.jmb.2004.07.075.

[50] A. Krueger, E. Protozanova und M. D. Frank-Kamenetskii. „Sequence-Dependent Basepair Opening in DNA Double Helix“. In: *Biophysical Journal* 90.9 (Mai 2006), S. 3091–3099. ISSN: 0006-3495. DOI: 10.1529/biophysj.105.078774.

[51] P. Yakovchuk. „Base-stacking and base-pairing contributions into thermal stability of the DNA double helix“. In: *Nucleic Acids Research* 34.2 (Jan. 2006), S. 564–574. ISSN: 1362-4962. DOI: 10.1093/nar/gkj454.

[52] J. D. Watson und F. H. C. Crick. „Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid“. In: *Nature* 171.4356 (Apr. 1953), S. 737–738. ISSN: 0028-0836. DOI: 10.1038/171737a0.

[53] J. C. Wang. „Helical repeat of DNA in solution.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 76.1 (Jan. 1979), S. 200–3. ISSN: 0027-8424. PMID: 284332.

[54] R. Wing, H. Drew, T. Takano u. a. „Crystal structure analysis of a complete turn of B-DNA“. In: *Nature* 287.5784 (Okt. 1980), S. 755–758. ISSN: 0028-0836. DOI: 10.1038/287755a0.

[55] R. Langridge, H. Wilson, C. Hooper u. a. „The molecular configuration of deoxyribonucleic acid“. In: *Journal of Molecular Biology* 2.1 (Apr. 1960), 19–IN11. ISSN: 0022-2836. DOI: 10.1016/s0022-2836(60)80004-6.

[56] F. M. Pohl und T. M. Jovin. „Salt-induced co-operative conformational change of a synthetic DNA: equilibrium and kinetic studies with poly (dG-dC).“ In: *J. Mol. Biol.* 67.3 (Juni 1972), S. 375–96. ISSN: 0022-2836. PMID: 5045303.

[57] A. H.-J. Wang, G. J. Quigley, F. J. Kolpak u. a. „Molecular structure of a left-handed double helical DNA fragment at atomic resolution“. In: *Nature* 282.5740 (Dez. 1979), S. 680–686. ISSN: 0028-0836. DOI: 10.1038/282680a0.

[58] S. Arnott, M. Wilkins, W. Fuller u. a. „Molecular and crystal structures of Double-helical RNA“. In: *Journal of Molecular Biology* 27.3 (Aug. 1967), S. 535–548. ISSN: 0022-2836. DOI: 10.1016/0022-2836(67)90057-5.

[59] C. Rivetti, C. Walker und C. Bustamante. „Polymer chain statistics and conformational analysis of DNA molecules with bends or sections of different flexibility“. In: *Journal of Molecular Biology* 280.1 (Juli 1998), S. 41–59. ISSN: 0022-2836. DOI: 10.1006/jmbi.1998.1830.

[60] S. Brinkers, H. R. C. Dietrich, F. H. de Groote u. a. „The persistence length of double stranded DNA determined using dark field tethered particle motion“. In: *The Journal of Chemical Physics* 130.21 (Juni 2009), S. 215105. ISSN: 0021-9606. DOI: 10.1063/1.3142699.

[61] P. W. K. Rothemund. „Folding DNA to create nanoscale shapes and patterns“. In: *Nature* 440.7082 (März 2006), S. 297–302. ISSN: 1476-4679. DOI: 10.1038/nature04586.

[62] C. E. Castro, F. Kilchherr, D.-N. Kim u. a. „A primer to scaffolded DNA origami“. In: *Nature Methods* 8.3 (März 2011), S. 221–229. ISSN: 1548-7105. DOI: 10.1038/nmeth.1570.

[63] J. Tang und R. R. Breaker. „Structural diversity of self-cleaving ribozymes“. In: *Proc. Natl. Acad. Sci. U.S.A.* 97.11 (Mai 2000), S. 5784–5789. ISSN: 1091-6490. DOI: 10.1073/pnas.97.11.5784.

[64] A. Kuzyk, R. Schreiber, Z. Fan u. a. „DNA-based self-assembly of chiral plasmonic nanostructures with tailored optical response“. In: *Nature* 483.7389 (März 2012), S. 311–314. ISSN: 1476-4687. DOI: 10.1038/nature10889.

[65] M. Demerec und U. Fano. „Bacteriophage-Resistant Mutants in Escherichia Coli.“ In: *Genetics* 30.2 (März 1945), S. 119–36. ISSN: 0016-6731. PMID: 17247150.

[66] M. Chamberlin, J. McGrath und L. Waskell. „New RNA Polymerase from Escherichia coli infected with Bacteriophage T7“. In: *Nature* 228.5268 (Okt. 1970), S. 227–231. ISSN: 0028-0836. DOI: 10.1038/228227a0.

[67] E. F. Pettersen, T. D. Goddard, C. C. Huang u. a. „UCSF Chimera–a visualization system for exploratory research and analysis.“ In: *J Comput Chem* 25.13 (Okt. 2004), S. 1605–12. ISSN: 0192-8651. PMID: 15264254.

[68] T. A. Steitz, G. M. T. Cheetham und D. Jeruzalmi. „Structural basis for initiation of transcription from an RNA polymerase–promoter complex“. In: *Nature* 399.6731 (Mai 1999), S. 80–83. ISSN: 0028-0836. DOI: 10.1038/19999.

[69] T. H. Tahirov, D. Temiakov, M. Anikin u. a. „Structure of a T7 RNA polymerase elongation complex at 2.9 Å resolution“. In: *Nature* 420.6911 (Nov. 2002), S. 43–50. ISSN: 0028-0836. DOI: 10.1038/nature01129.

[70] M. Chamberlin und J. Ring. „Characterization of T7-specific ribonucleic acid polymerase. 1. General properties of the enzymatic reaction and the template specificity of the enzyme.“ In: *J. Biol. Chem.* 248.6 (März 1973), S. 2235–44. ISSN: 0021-9258. PMID: 4570474.

[71] S. A. Darst, E. W. Kubalek und R. D. Kornberg. „Three-dimensional structure of Escherichia coli RNA polymerase holoenzyme determined by electron crystallography“. In: *Nature* 340.6236 (Aug. 1989), S. 730–732. ISSN: 0028-0836. DOI: 10.1038/340730a0.

[72] C. T. Martin und J. E. Coleman. „Kinetic analysis of T7 RNA polymerase-promoter interactions with small synthetic promoters“. In: *Biochemistry* 26.10 (Mai 1987), S. 2690–2696. ISSN: 1520-4995. DOI: 10.1021/bi00384a006.

[73] H. L. Osterman und J. E. Coleman. „T7 Ribonucleic acid polymerase-promoter interactions“. In: *Biochemistry* 20.17 (Aug. 1981), S. 4884–4892. ISSN: 1520-4995. DOI: 10.1021/bi00520a013.

[74] R. J. Roberts. „Restriction enzymes and their isoschizomers.“ In: *Nucleic Acids Res.* 18 Suppl (Apr. 1990), S. 2331–65. ISSN: 0305-1048. PMID: 2159140.

[75] J. Hedgpeth, H. M. Goodman und H. W. Boyer. „DNA nucleotide sequence restricted by the RI endonuclease.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 69.11 (Nov. 1972), S. 3448–52. ISSN: 0027-8424. PMID: 4343974.

[76] B. Polisky, P. Greene, D. E. Garfin u. a. „Specificity of substrate recognition by the EcoRI restriction endonuclease.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 72.9 (Sep. 1975), S. 3310–4. ISSN: 0027-8424. PMID: 242001.

[77] R. Y. Walder, J. A. Walder und J. E. Donelson. „The organization and complete nucleotide sequence of the PstI restriction-modification system.“ In: *J. Biol. Chem.* 259.12 (Juni 1984), S. 8015–26. ISSN: 0021-9258. PMID: 6330092.

[78] B. Zain und R. J. Roberts. „A new specific endonuclease from Xanthomonas badrii“. In: *Journal of Molecular Biology* 115.2 (Sep. 1977), S. 249–255. ISSN: 0022-2836. DOI: 10.1016/0022-2836(77)90101-2.

[79] B. P. Cormack, R. H. Valdivia und S. Falkow. „FACS-optimized mutants of the green fluorescent protein (GFP)“. In: *Gene* 173.1 (Jan. 1996), S. 33–38. ISSN: 0378-1119. DOI: 10.1016/0378-1119(95)00685-0.

[80] R. E. Campbell, O. Tour, A. E. Palmer u. a. „A monomeric red fluorescent protein“. In: *Proc. Natl. Acad. Sci. U.S.A.* 99.12 (Juni 2002), S. 7877–7882. ISSN: 1091-6490. DOI: 10.1073/pnas.082243699.

[81] F. Yang, L. G. Moss und G. N. Phillips. „The molecular structure of green fluorescent protein“. In: *Nature Biotechnology* 14.10 (Okt. 1996), S. 1246–1251. ISSN: 1087-0156. DOI: 10.1038/nbt1096-1246.

[82] O. Shimomura, F. H. Johnson und Y. Saiga. „Extraction, Purification and Properties of Aequorin, a Bioluminescent Protein from the Luminous Hydromedusan,Aequorea“. In: *J. Cell. Comp. Physiol.* 59.3 (Juni 1962), S. 223–239. ISSN: 1553-0809. DOI: 10.1002/jcp.1030590302.

[83] O. Shimomura. „The discovery of aequorin and green fluorescent protein“. In: *J Microsc* 217.1 (Jan. 2005), S. 3–15. ISSN: 1365-2818. DOI: 10.1111/j.0022-2720.2005.01441.x.

[84] D. Yarbrough, R. M. Wachter, K. Kallio u. a. „Refined crystal structure of DsRed, a red fluorescent protein from coral, at 2.0-A resolution“. In: *Proc. Natl. Acad. Sci. U.S.A.* 98.2 (Jan. 2001), S. 462–467. ISSN: 1091-6490. DOI: 10.1073/pnas.98.2.462.

[85] M. V. Matz, A. F. Fradkov, Y. A. Labas u. a. „Fluorescent proteins from nonbioluminescent Anthozoa species.“ In: *Nat. Biotechnol.* 17.10 (Okt. 1999), S. 969–73. ISSN: 1087-0156. PMID: 10504696.

[86] J. Hacker und G. Blum-Oehler. „In appreciation of Theodor Escherich“. In: *Nature Reviews Microbiology* 5.12 (Dez. 2007), S. 902–902. ISSN: 1740-1534. DOI: 10.1038/nrmicro1810.

[87] J. B. Kaper, J. P. Nataro und H. L. T. Mobley. „Pathogenic Escherichia coli“. In: *Nature Reviews Microbiology* 2.2 (Feb. 2004), S. 123–140. ISSN: 1740-1534. DOI: 10.1038/nrmicro818.

[88] B. R. Laboratories. „BRL pUC host: E. coli DH5α competent cells“. In: *Focus* 8 (1986), S. 9–12.

[89] S. G. Grant, J. Jessee, F. R. Bloom u. a. „Differential plasmid rescue from transgenic mouse DNAs into Escherichia coli methylation-restriction mutants.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 87.12 (Juni 1990), S. 4645–9. ISSN: 0027-8424. PMID: 2162051.

[90] R. Lutz und H. Bujard. „Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements“. In: *Nucleic Acids Research* 25.6 (März 1997), S. 1203–1210. ISSN: 1362-4962. DOI: 10.1093/nar/25.6.1203.

[91] F. Studier und B. A. Moffatt. „Use of bacteriophage T7 RNA polymerase to direct selective high-level expression of cloned genes“. In: *Journal of Molecular Biology* 189.1 (Mai 1986), S. 113–130. ISSN: 0022-2836. DOI: 10.1016/0022-2836(86)90385-2.

[92] B. A. Moffatt und F. W. Studier. „T7 lysozyme inhibits transcription by T7 RNA polymerase.“ In: *Cell* 49.2 (Apr. 1987), S. 221–7. ISSN: 0092-8674. PMID: 3568126.

[93] R. P. Shetty, D. Endy und T. F. Knight. „Engineering BioBrick vectors from BioBrick parts“. In: *J Biol Eng* 2.1 (Apr. 2008), S. 5. ISSN: 1754-1611. DOI: 10.1186/1754-1611-2-5.

[94] G. C. Chamnes und C. D. Willson. „An unusual lac repressor mutant“. In: *Journal of Molecular Biology* 53.3 (Nov. 1970), S. 561–565. ISSN: 0022-2836. DOI: 10.1016/0022-2836(70)90084-7.

[95] M. Pfahl. „lac repressor-operator interaction. Analysis of the X86 repressor mutant“. In: *Journal of Molecular Biology* 106.3 (Sep. 1976), S. 857–869. ISSN: 0022-2836. DOI: 10.1016/0022-2836(76)90269-2.

[96] W. Gilbert und B. Müller-Hill. „Isolation of the Lac Repressor“. In: *Proc. Natl. Acad. Sci. U.S.A.* 56.6 (Dez. 1966), S. 1891–1898. ISSN: 1091-6490. DOI: 10.1073/pnas.56.6.1891.

[97] A. Eberhard, A. L. Burlingame, C. Eberhard u. a. „Structural identification of autoinducer of Photobacterium fischeri luciferase“. In: *Biochemistry* 20.9 (Apr. 1981), S. 2444–2449. ISSN: 1520-4995. DOI: 10.1021/bi00512a013.

[98] J. Engebrecht und M. Silverman. „Identification of genes and gene products necessary for bacterial bioluminescence.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 81.13 (Juli 1984), S. 4154–8. ISSN: 0027-8424. PMID: 6377310.

[99] K. A. Egland und E. P. Greenberg. „Quorum sensing in Vibrio fischeri: elements of the luxI promoter“. In: *Mol Microbiol* 31.4 (Feb. 1999), S. 1197–1204. ISSN: 1365-2958. DOI: 10.1046/j.1365-2958.1999.01261.x.

[100] C. Holtze, A. C. Rowat, J. J. Agresti u. a. „Biocompatible surfactants for water-in-fluorocarbon emulsions“. In: *Lab Chip* 8.10 (Sep. 2008), S. 1632. ISSN: 1473-0189. DOI: 10.1039/b806706f.

[101] W. Demtröder. *Experimentalphysik*. Springer Berlin Heidelberg, 2003. ISBN: 978-3-662-08597-4. DOI: 10.1007/978-3-662-08597-4.

[102] A. Einstein. „Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen [AdP 17, 549 (1905)]“. In: *Annalen der Physik* 14.S1 (Feb. 2005), S. 182–193. ISSN: 1521-3889. DOI: 10.1002/andp.200590005.

[103] M. M. Tirado und J. G. de la Torre. „Translational friction coefficients of rigid, symmetric top macromolecules. Application to circular cylinders“. In: *The Journal of Chemical Physics* 71.6 (Sep. 1979), S. 2581. ISSN: 0021-9606. DOI: 10.1063/1.438613.

[104] M. M. Tirado, C. L. Martínez und J. G. de la Torre. „Comparison of theories for the translational and rotational diffusion coefficients of rod-like macromolecules. Application to short DNA fragments“. In: *The Journal of Chemical Physics* 81.4 (Aug. 1984), S. 2047. ISSN: 0021-9606. DOI: 10.1063/1.447827.

[105] B. H. Zimm. „Dynamics of Polymer Molecules in Dilute Solution: Viscoelasticity, Flow Birefringence and Dielectric Loss“. In: *The Journal of Chemical Physics* 24.2 (Feb. 1956), S. 269. ISSN: 0021-9606. DOI: 10.1063/1.1742462.

[106] S. S. Sorlie und R. Pecora. „A dynamic light scattering study of four DNA restriction fragments“. In: *Macromolecules* 23.2 (März 1990), S. 487–497. ISSN: 1520-5835. DOI: 10.1021/ma00204a022.

[107] G. L. Lukacs, P. Haggie, O. Seksek u. a. „Size-dependent DNA Mobility in Cytoplasm and Nucleus“. In: *Journal of Biological Chemistry* 275.3 (Jan. 2000), S. 1625–1629. ISSN: 1083-351X. DOI: 10.1074/jbc.275.3.1625.

[108] E. Stellwagen und N. C. Stellwagen. „Probing the electrostatic shielding of DNA with capillary electrophoresis.“ In: *Biophys. J.* 84.3 (März 2003), S. 1855–66. ISSN: 0006-3495. PMID: 12609887.

[109] A. E. Nkodo, J. M. Garnier, B. Tinland u. a. „Diffusion coefficient of DNA molecules during free solution electrophoresis.“ In: *Electrophoresis* 22.12 (Aug. 2001), S. 2424–32. ISSN: 0173-0835. PMID: 11519946.

[110] N. C. Stellwagen, S. Magnusdottir, C. Gelfi u. a. „Measuring the translational diffusion coefficients of small DNA molecules by capillary electrophoresis.“ In: *Biopolymers* 58.4 (Apr. 2001), S. 390–7. ISSN: 0006-3525. PMID: 11180052.

[111] E. Stellwagen und N. C. Stellwagen. „Determining the electrophoretic mobility and translational diffusion coefficients of DNA molecules in free solution.“ In: *Electrophoresis* 23.16 (Aug. 2002), S. 2794–803. ISSN: 0173-0835. PMID: 12210184.

[112] F. Crick. „Diffusion in Embryogenesis“. In: *Nature* 225.5231 (Jan. 1970), S. 420–422. ISSN: 0028-0836. DOI: 10.1038/225420a0.

[113] J. E. Tanner. „Transient diffusion in a system partitioned by permeable barriers. Application to NMR measurements with a pulsed field gradient“. In: *The Journal of Chemical Physics* 69.4 (Aug. 1978), S. 1748. ISSN: 0021-9606. DOI: 10.1063/1.436751.

[114] L. L. Latour, K. Svoboda, P. P. Mitra u. a. „Time-dependent diffusion of water in a biological model system.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 91.4 (Feb. 1994), S. 1229–1233. ISSN: 1091-6490. DOI: 10.1073/pnas.91.4.1229.

[115] P. N. Sen. „Time-dependent diffusion coefficient as a probe of the permeability of the pore wall“. In: *The Journal of Chemical Physics* 119.18 (Nov. 2003), S. 9871. ISSN: 0021-9606. DOI: 10.1063/1.1611477.

[116] A. M. Berezhkovskii, V. Y. Zitserman und S. Y. Shvartsman. „Diffusivity in periodic arrays of spherical cavities“. In: *The Journal of Chemical Physics* 118.15 (Apr. 2003), S. 7146. ISSN: 0021-9606. DOI: 10.1063/1.1561615.

[117] V. V. Anshelevich, A. V. Vologodskii, A. V. Lukashin u. a. „Slow relaxational processes in the melting of linear biopolymers: A theory and its application to nucleic acids“. In: *Biopolymers* 23.1 (Jan. 1984), S. 39–58. ISSN: 1097-0282. DOI: 10.1002/bip.360230105.

[118] M. E. Craig, D. M. Crothers und P. Doty. „Relaxation kinetics of dimer formation by self complementary oligonucleotides“. In: *Journal of Molecular Biology* 62.2 (Dez. 1971), S. 383–401. ISSN: 0022-2836. DOI: 10.1016/0022-2836(71)90434-7.

[119] D. Pörschke und M. Eigen. „Co-operative non-enzymatic base recognition III. Kinetics of the helix—coil transition of the oligoribouridylic · oligoriboadenylic acid system and of oligoriboadenylic acid alone at acidic pH“. In: *Journal of Molecular Biology* 62.2 (Dez. 1971), S. 361–381. ISSN: 0022-2836. DOI: 10.1016/0022-2836(71)90433-5.

[120] L. P. Reynaldo, A. V. Vologodskii, B. P. Neri u. a. „The kinetics of oligonucleotide replacements“. In: *Journal of Molecular Biology* 297.2 (März 2000), S. 511–520. ISSN: 0022-2836. DOI: 10.1006/jmbi.2000.3573.

[121] A. Turberfield, J. Mitchell, B. Yurke u. a. „DNA Fuel for Free-Running Nanomachines“. In: *Physical Review Letters* 90.11 (März 2003). ISSN: 1079-7114. DOI: 10.1103/physrevlett.90.118102.

[122] D. Y. Zhang und E. Winfree. „Control of DNA Strand Displacement Kinetics Using Toehold Exchange“. In: *Journal of the American Chemical Society* 131.47 (Dez. 2009), S. 17303–17314. ISSN: 1520-5126. DOI: 10.1021/ja906987s.

[123] N. Srinivas, T. E. Ouldridge, P. Sulc u. a. „On the biophysics and kinetics of toehold-mediated DNA strand displacement“. In: *Nucleic Acids Research* 41.22 (Dez. 2013), S. 10641–10658. ISSN: 1362-4962. DOI: 10.1093/nar/gkt801.

[124] G. M. Rather, J. Mukherjee, P. J. Halling u. a. „Activation of Alpha Chymotrypsin by Three Phase Partitioning Is Accompanied by Aggregation“. In: *PLoS ONE* 7.12 (Dez. 2012). Hrsg. von R. H. Khan, e49241. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0049241.

[125] A. K. Upadhyay, A. Murmu, A. Singh u. a. „Kinetics of Inclusion Body Formation and Its Correlation with the Characteristics of Protein Aggregates in Escherichia coli“. In: *PLoS ONE* 7.3 (März 2012). Hrsg. von C. Herman, e33951. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0033951.

[126] O. G. Berg. „A model for the statistical fluctuations of protein numbers in a microbial population“. In: *Journal of Theoretical Biology* 71.4 (Apr. 1978), S. 587–603. ISSN: 0022-5193. DOI: 10.1016/0022-5193(78)90326-0.

[127] D. Huh und J. Paulsson. „Non-genetic heterogeneity from stochastic partitioning at cell division“. In: *Nature Genetics* 43.2 (Feb. 2011), S. 95–100. ISSN: 1546-1718. DOI: 10.1038/ng.729.

[128] C. Furusawa, T. Suzuki, A. Kashiwagi u. a. „Ubiquity of log-normal distributions in intra-cellular reaction dynamics“. In: *BIOPHYSICS* 1 (Apr. 2005), S. 25–31. ISSN: 1349-2942. DOI: 10.2142/biophysics.1.25.

[129] A. Tiessen, P. Pérez-Rodríguez und L. Delaye-Arredondo. „Mathematical modeling and comparison of protein size distribution in different plant, animal, fungal and microbial species reveals a negative correlation between protein size and protein number, thus providing insight into the evolution of proteomes“. In: *BMC Res Notes* 5.1 (Feb. 2012), S. 85. ISSN: 1756-0500. DOI: 10.1186/1756-0500-5-85.

[130] N. Friedman, L. Cai und X. Xie. „Linking Stochastic Dynamics to Population Distribution: An Analytical Framework of Gene Expression“. In: *Physical Review Letters* 97.16 (Okt. 2006). ISSN: 1079-7114. DOI: 10.1103/physrevlett.97.168302.

[131] D. Huh und J. Paulsson. „Random partitioning of molecules at cell division“. In: *Proc. Natl. Acad. Sci. U.S.A.* 108.36 (Aug. 2011), S. 15004–15009. ISSN: 1091-6490. DOI: 10.1073/pnas.1013171108.

[132] A. Singh. „Transient Changes in Intercellular Protein Variability Identify Sources of Noise in Gene Expression“. In: *Biophysical Journal* 107.9 (Nov. 2014), S. 2214–2220. ISSN: 0006-3495. DOI: 10.1016/j.bpj.2014.09.017.

[133] S. Youssef, S. Gude und J. O. Rädler. „Automated tracking in live-cell time-lapse movies“. In: *Integr. Biol.* 3.11 (Sep. 2011), S. 1095. ISSN: 1757-9708. DOI: 10.1039/c1ib00035g.

[134] J. Kim und E. Winfree. „Synthetic in vitro transcriptional oscillators“. In: *Molecular Systems Biology* 7.1 (Jan. 2011), S. 465–465. ISSN: 1744-4292. DOI: 10.1038/msb.2010.119.

[135] M. Weitz, J. Kim, K. Kapsner u. a. „Diversity in the dynamical behaviour of a compartmentalized programmable biochemical oscillator“. In: *Nature Chemistry* 6.4 (Feb. 2014), S. 295–302. ISSN: 1755-4349. DOI: 10.1038/nchem.1869.

[136] M. Weitz, A. Mückl, K. Kapsner u. a. „Communication and Computation by Bacteria Compartmentalized within Microemulsion Droplets“. In: *Journal of the American Chemical Society* 136.1 (Jan. 2014), S. 72–75. ISSN: 1520-5126. DOI: 10.1021/ja411132w.

[137] K. Kapsner und F. C. Simmel. „Partitioning variability of a compartmentalized transcriptional thresholding circuit“. In: *ACS Synth. Biol.* (in Vorbereitung).

[138] S. Tyagi und F. R. Kramer. „Molecular Beacons: Probes that Fluoresce upon Hybridization“. In: *Nature Biotechnology* 14.3 (März 1996), S. 303–308. ISSN: 1087-0156. DOI: 10.1038/nbt0396-303.

[139] A. Edelstein, N. Amodaj, K. Hoover u. a. „Computer Control of Microscopes Using µManager“. In: *Current Protocols in Molecular Biology* (Mai 2001). DOI: 10.1002/0471142727.mb1420s92.

[140] J. Quan und J. Tian. „Circular Polymerase Extension Cloning of Complex Gene Libraries and Pathways“. In: *PLoS ONE* 4.7 (Juli 2009). Hrsg. von P. L. Ho, e6441. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0006441.

[141] A. Oliva, A. Santoveña, J. Fariña u. a. „Effect of high shear rate on stability of proteins: kinetic study“. In: *Journal of Pharmaceutical and Biomedical Analysis* 33.2 (Sep. 2003), S. 145–155. ISSN: 0731-7085. DOI: 10.1016/s0731-7085(03)00223-1.

[142] E. Di Stasio und R. De Cristofaro. „The effect of shear stress on protein conformation“. In: *Biophysical Chemistry* 153.1 (Dez. 2010), S. 1–8. ISSN: 0301-4622. DOI: 10.1016/j.bpc.2010.07.002.

[143] P. W. Bridgman. „The coagulation of albumen by pressure“. In: *Journal of Biological Chemistry* 19.4 (Dez. 1914), S. 511–512. eprint: http://www.jbc.org/content/19/4/511.full.pdf+html.

[144] H.-J. Kim, E. A. Decker und D. J. McClements. „Impact of Protein Surface Denaturation on Droplet Flocculation in Hexadecane Oil-in-Water Emulsions Stabilized by β-Lactoglobulin“. In: *J. Agric. Food Chem.* 50.24 (Nov. 2002), S. 7131–7137. ISSN: 1520-5118. DOI: 10.1021/jf020366q.

[145] Y. Liu, S.-Y. Jung und C. P. Collier. „Shear-Driven Redistribution of Surfactant Affects Enzyme Activity in Well-Mixed Femtoliter Droplets“. In: *Analytical Chemistry* 81.12 (Juni 2009), S. 4922–4928. ISSN: 1520-6882. DOI: 10.1021/ac900624h.

[146] H. Karbstein und H. Schubert. „Developments in the continuous mechanical production of oil-in-water macro-emulsions“. In: *Chemical Engineering and Processing: Process Intensification* 34.3 (Juni 1995), S. 205–211. ISSN: 0255-2701. DOI: 10.1016/0255-2701(94)04005-2.

[147] G. T. Vladisavljević, I. Kobayashi und M. Nakajima. „Production of uniform droplets using membrane, microchannel and microfluidic emulsification devices“. In: *Microfluid Nanofluid* 13.1 (Feb. 2012), S. 151–178. ISSN: 1613-4990. DOI: 10.1007/s10404-012-0948-0.

[148] C. A. Schneider, W. S. Rasband und K. W. Eliceiri. „NIH Image to ImageJ: 25 years of image analysis“. In: *Nature Methods* 9.7 (Juni 2012), S. 671–675. ISSN: 1548-7105. DOI: 10.1038/nmeth.2089.

[149] The GIMP team. *GIMP: GNU Image Manipulation Program*. http://gimp.net/. 1997-2015.

[150] T. S. Huang, Hrsg. *Two-Dimensional Digital Signal Prcessing II: Transforms and Median Filters*. Springer-Verlag, 1981. ISBN: 3-540-10359-7. DOI: 10.1007/bfb0057592.

[151] N. Otsu. „A Threshold Selection Method from Gray-Level Histograms“. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (Jan. 1979), S. 62–66. ISSN: 0018-9472. DOI: 10.1109/tsmc.1979.4310076.

[152] P. Stathis, E. Kavallieratou und N. Papamarkos. „An Evaluation Technique for Binarization Algorithms“. In: *Journal of Universal Computer Science* 14.18 (Okt. 2008), S. 3011–3030. DOI: 10.3217/jucs-014-18-3011.

[153] L. Lam, S.-W. Lee und C. Suen. „Thinning methodologies-a comprehensive survey“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.9 (Sep. 1992), S. 869–885. ISSN: 0162-8828. DOI: 10.1109/34.161346.

[154] F. Meyer. „Topographic distance and watershed lines“. In: *Signal Processing* 38.1 (Juli 1994), S. 113–125. ISSN: 0165-1684. DOI: 10.1016/0165-1684(94)90060-4.

[155] S. Lloyd. „Least squares quantization in PCM“. In: *IEEE Trans. Inform. Theory* 28.2 (März 1982), S. 129–137. ISSN: 0018-9448. DOI: 10.1109/tit.1982.1056489.

[156] R. Xu und D. Wunsch II. „Survey of Clustering Algorithms“. In: *IEEE Transactions on Neural Networks* 16.3 (Mai 2005), S. 645–678. ISSN: 1045-9227. DOI: 10.1109/tnn.2005.845141.

[157] A. Rodriguez und A. Laio. „Clustering by fast search and find of density peaks“. In: *Science* 344.6191 (Juni 2014), S. 1492–1496. ISSN: 1095-9203. DOI: 10.1126/science.1242072.

[158] B. A. Hense, C. Kuttler, J. Müller u. a. „Does efficiency sensing unify diffusion and quorum sensing?“ In: *Nat. Rev. Microbiol.* 5.3 (März 2007), S. 230–9. ISSN: 1740-1534. PMID: 17304251.

[159] M. L. Urbanowski, C. P. Lostroh und E. P. Greenberg. „Reversible acyl-homoserine lactone binding to purified Vibrio fischeri LuxR protein.“ In: *J. Bacteriol.* 186.3 (Feb. 2004), S. 631–7. ISSN: 0021-9193. PMID: 14729687.

[160] B. Canton, A. Labno und D. Endy. „Refinement and standardization of synthetic biological parts and devices.“ In: *Nat. Biotechnol.* 26.7 (Juli 2008), S. 787–93. ISSN: 1546-1696. PMID: 18612302.

[161] R. Iizuka, M. Yamagishi-Shirasaki und T. Funatsu. „Kinetic study of de novo chromophore maturation of fluorescent proteins.“ In: *Anal. Biochem.* 414.2 (Juli 2011), S. 173–8. ISSN: 1096-0309. PMID: 21459075.

[162] S. Matsumoto und M. Kohda. „The viscosity of W/O/W emulsions: An attempt to estimate the water permeation coefficient of the oil layer from the viscosity changes in diluted systems on aging under osmotic pressure gradients“. In: *Journal of Colloid and Interface Science* 73.1 (Jan. 1980), S. 13–20. ISSN: 0021-9797. DOI: 10.1016/0021-9797(80)90115-0.

[163] G. K. Ackers, A. D. Johnson und M. A. Shea. „Quantitative model for gene regulation by lambda phage repressor.“ In: *Proc. Natl. Acad. Sci. U.S.A.* 79.4 (Feb. 1982), S. 1129–33. ISSN: 0027-8424. PMID: 6461856.

[164] U. Gerland, J. D. Moroz und T. Hwa. „Physical constraints and functional characteristics of transcription factor-DNA interaction“. In: *Proc. Natl. Acad. Sci. U.S.A.* 99.19 (Sep. 2002), S. 12015–12020. ISSN: 1091-6490. DOI: 10.1073/pnas.192693599.

[165] S. S. Sastry und B. M. Ross. „Nuclease activity of T7 RNA polymerase and the heterogeneity of transcription elongation complexes.“ In: *J. Biol. Chem.* 272.13 (März 1997), S. 8644–52. ISSN: 0021-9258. PMID: 9079696.

[166] K. Finan, J. P. Torella, A. N. Kapanidis u. a. „T7 RNA Polymerase Functions In Vitro without Clustering“. In: *PLoS ONE* 7.7 (Juli 2012). Hrsg. von J. Langowski, e40207. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0040207.

[167] S. Kiese, A. Papppenberger, W. Friess u. a. „Shaken, not stirred: Mechanical stress testing of an IgG1 antibody“. In: *J. Pharm. Sci.* 97.10 (Okt. 2008), S. 4347–4366. ISSN: 1520-6017. DOI: 10.1002/jps.21328.

[168] D. Gonze, J. Halloy und A. Goldbeter. „Robustness of circadian rhythms with respect to molecular noise“. In: *Proc. Natl. Acad. Sci. U.S.A.* 99.2 (Jan. 2002), S. 673–678. ISSN: 1091-6490. DOI: 10.1073/pnas.022628299.

[169] J. M. G. Vilar, H. Y. Kueh, N. Barkai u. a. „Mechanisms of noise-resistance in genetic oscillators“. In: *Proc. Natl. Acad. Sci. U.S.A.* 99.9 (Apr. 2002), S. 5988–5992. ISSN: 1091-6490. DOI: 10.1073/pnas.092133899.

[170] S. Kar, W. T. Baumann, M. R. Paul u. a. „Exploring the roles of noise in the eukaryotic cell cycle“. In: *Proc. Natl. Acad. Sci. U.S.A.* 106.16 (Feb. 2009), S. 6471–6476. ISSN: 1091-6490. DOI: 10.1073/pnas.0810034106.

[171] M. Modesti. „Fluorescent Labeling of Proteins“. In: *Single Molecule Analysis* (Juli 2011), S. 101–120. ISSN: 1940-6029. DOI: 10.1007/978-1-61779-282-3_6.

[172] S. Basu, Y. Gerchman, C. H. Collins u. a. „A synthetic multicellular system for programmed pattern formation“. In: *Nature* 434.7037 (Apr. 2005), S. 1130–1134. ISSN: 1476-4679. DOI: 10.1038/nature03461.

[173] J. J. Tabor, H. M. Salis, Z. B. Simpson u. a. „A Synthetic Genetic Edge Detection Program“. In: *Cell* 137.7 (Juni 2009), S. 1272–1281. ISSN: 0092-8674. DOI: 10.1016/j.cell.2009.04.048.

[174] A. Raj und A. van Oudenaarden. „Nature, Nurture, or Chance: Stochastic Gene Expression and Its Consequences“. In: *Cell* 135.2 (Okt. 2008), S. 216–226. ISSN: 0092-8674. DOI: 10.1016/j.cell.2008.09.050.

[175] L. Aufinger. „Dynamics of Nucleic-Acid-Based In Vitro Reaction Circuits“. Bachelorarbeit. Technische Universität München, Juli 2014.

[176] E. Arias-Castro und D. L. Donoho. „Does median filtering truly preserve edges better than linear filtering?“ In: *Ann. Statist.* 37.3 (Juni 2009), S. 1172–1206. ISSN: 0090-5364. DOI: 10.1214/08-aos604.

[177] R. Bayer. „Symmetric binary B-Trees: Data structure and maintenance algorithms“. In: *Acta Informatica* 1.4 (Dez. 1972), S. 290–306. ISSN: 1432-0525. DOI: 10.1007/bf00289509.

[178] Z. Kulpa. „Area and perimeter measurement of blobs in discrete binary pictures“. In: *Computer Graphics and Image Processing* 6.5 (Okt. 1977), S. 434–451. ISSN: 0146-664X. DOI: 10.1016/s0146-664x(77)80021-x.