An Investigation of Prime Editing Kinetics:
Physical Foundations Towards Realizing Efficient Biocomputing

Nathalie Gabriele Schäffler

Munich, 2024

An Investigation of Prime Editing Kinetics:
Physical Foundations Towards Realizing Efficient Biocomputing

Nathalie Gabriele Schäffler

Dissertation
submitted to the Faculty of Physics
Ludwig-Maximilians-Universität
Munich

by
Nathalie Gabriele Schäffler
born in Munich

Munich, 17.10.2024

Supervisor: Prof. Dr. Joachim O. Rädler

Second Examiner: Prof. Dr. Ralf Zimmer

Date of PhD Defense: 29.11.2024

Die Kinetik von Prime Editing:
Physikalische Grundlagen für realisierbare Biocomputer

Nathalie Gabriele Schäffler

München, 2024

Die Kinetik von Prime Editing:
Physikalische Grundlagen für realisierbare Biocomputer

Nathalie Gabriele Schäffler

Dissertation
an der Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Nathalie Gabriele Schäffler
aus München

München, den 17.10.2024

Erstgutachter: Prof. Dr. Joachim O. Rädler
Zweitgutachter: Prof. Dr. Ralf Zimmer
Tag der mündlichen Prüfung: 29.11.2024

Zusammenfassung

DNA spielt eine grundlegende Rolle in lebenden Systemen, indem sie genetische Informationen speichert und reguliert. Neue Entwicklungen wie Prime Editing ermöglichen präzise Modifikationen der Basenpaare des Genoms und versprechen damit verbesserte Therapien für genetische bedingte Krankheiten. Die gezielte Veränderung einer genetischen Sequenz könnte außerdem eine wichtige Rolle bei der Entwicklung DNA-basierter Rechensysteme spielen. Viele Publikationen konzentrieren sich aktuell hauptsächlich auf die Verbesserung der Effizienz von Prime Editing durch Optimierung von Proteinen, pegRNAs und Transfektionsmethoden. In diesen Studien wird die Effizienz eines Editing-Schritts meist jedoch erst nach mehreren Tagen gemessen, was für Anwendungen wie biologische Computer zu ungenau ist. Um die Kinetik von Prime Editing optimieren zu können, ist das Ziel dieser Arbeit sie erstmals mithilfe von experimentellen Methoden und mathematischen Modellen präzier zu quantifizieren.

Im ersten Teil der vorliegenden Arbeit wird die „Live Imaging on Single-Cell Arrays” -Methodik benutzt, um Prime-Editing-Ereignisse anhand der Verteilung von Einzelzelldaten zu charakterisieren. Zunächst wird gezeigt, dass der Beginn eines Fluoreszenzsignals nach Transfektion mittels mRNA-Transfektion 13 h schneller erfolgte als über Plasmid-DNA (pDNA). Mit einer Zelllinie, die einen Fluoreszenzreporter exprimiert, welcher nur nach erfolgreichem Prime Editing grünes Licht emitieren kann, wird die „time-to-edit”, die Zeit von der Transfektion der Prime-Editing-Komponenten bis zum Einsetzen der Fluoreszenz, bestimmt. Der Vergleich von mRNA- und pDNA-Transfektion des Prime Editing Systems zeigt, dass die mRNA-Transfektion 7 h schneller war als die pDNA-Transfektion, aber auch zu geringerer Effizienz führte. Die „editing time”, also die Zeit von der Expression des Prime-Editing-Komplexes bis zum Einsetzen der Fluoreszenz, ist bei mRNA-basierter Transfektion 15 mal langsamer als über pDNA. In dieser Arbeit wird außerdem gezeigt, dass in Übereinstimmung mit der ”branch-migration”-Theorie längere Edits quadratisch mit längeren Editing-Zeiten korrelieren. Durch mathematische Modellierung wurden limitierende Parameter, wie das Transfektionsverhältnis von pegRNA zu Prime-Editor-mRNA, extrahiert und die Genauigkeit verschiedener Expressionsmodelle bewertet.

Basierend auf der „search-and-replace” Operation, ermöglicht durch Prime Editing, wird abschließend ein theoretischer Rahmen für die Implementierung von Teilen des SKI-Kombinator Kalküls, eine reduzierte Form des Lambda Kalküls, vorgeschlagen. Für diese theoretische Implementierung sind jedoch eine genaue Reihenfolge und Präzision der Operationen notwendig. Inwieweit diese Voraussetzungen in einer experimentellen Implementierung sichergestellt werden können, muss verifiziert werden.

Insgesamt tragen die in dieser Arbeit erstmalig erstellten Prime Editing Modelle und Vergleiche mit experimentellen Daten zum besseren Verständnis der Kinetik bei und unterstreichen, wie wichtig es ist, bei der Optimierung dieser Technik nicht nur die Effizienz, sondern auch die Kinetik zu berücksichtigen. Die hier präsentierten Ergebnisse bieten mögliche Ansätze für Informationsverarbeitung mithilfe von DNA, was eine Grundvoraussetzung für zukünftige, DNA-basierte Computer darstellt. Darüber hinaus können diese Erkenntnisse zur Kinetik auch in weiteren Bereichen, beispielsweise in der Arzneimittelforschung, von Nutzen sein.

Summary

DNA plays a fundamental role in living systems for storing and regulating genetic information. Recent scientific advancements such as prime editing allow precise modifications of base pairs on the genome, which promises to improve gene therapy. The possibility of precisely changing DNA sequences may also play a significant role in the development of advanced biological computing systems. Current research focuses primarily on prime editing efficiency through better protein engineering, pegRNA design, and delivery methods. In these studies, a single editing step is measured only after several days, which is to inaccurate for applications like biocomputing. Quantifying the kinetics of prime editing more precisely is therefore important for optimizing this process. This thesis aims to address this gap using a combination of experimental methods and mathematical models.

In the first study, the “Live Imaging on Single-cell Arrays” assay is used to characterize prime editing events based on the distribution of single-cell data. First, it is demonstrated that transfection of fluorescent reporters via mRNA is 13 h faster than using plasmid DNA (pDNA). This approach is then used to monitor the kinetics of prime editing in real-time by employing a modified cell line, capable of expressing fluorescence only after successful prime editing. The time-to-edit is defined as the time from transfection of the prime editing components to the onset of fluorescence. The comparison between mRNA and pDNA delivery shows that mRNA delivery is 7 h faster than pDNA but resulted in reduced efficiency. Measuring the time from the expression of the Prime Editing complex to the onset of fluorescence, so the editing time itself, showed that this is 15 times slower for mRNA transfection than for pDNA transfection. This study also indicates that longer edits correlated with longer editing times, consistent with branch migration theory. Through mathematical modeling, limiting parameters were extracted for mRNA-based delivery, such as the initial ratio of delivered pegRNA to prime editor mRNA, and the accuracy of different expression models was evaluated.

Finally, based on the “search-and-replace” operation provided by prime editing, a theoretical framework for the implementation of parts of the SKI combinator calculus, a reduced form of the lambda calculus, is proposed. However, this theoretical implementation requires exact sequence and precision of operations. Whether these requirements can be ensured in an experimental implementation should be verified.

In conclusion, the novel prime editing models developed in this study, alongside comparisons with experimental data, contribute to a better understanding of prime editing and highlight the importance of considering not only efficiency but also kinetics when optimizing this technique. The results presented here offer possible approaches for information processing using DNA, which is a prerequisite for future DNA-based computers. In addition, these findings on kinetics can also be useful in other areas, such as drug delivery research.

Contents

List of Publications and Manuscripts

[P1] **Nathalie G. Schäffler**, DJ.J. Truong, Julian Geilenkeuser, Gil G. Westmeyer, and Joachim O. Rädler. "Investigating the Kinetics of Prime Editing". In: *Molecular Therapy* 31.4 (May 2023). Publisher: Elsevier, Nr 965, p468. DOI: 10.1016/j.ymthe.2023.04.017.

[P2] **Nathalie G. Schäffler**, DJ.J. Truong, Julian Geilenkeuser, Gil G. Westmeyer, and Joachim O. Rädler. "Quantifying Prime Editing Kinetics - Timelapse Fluorescence Data and Model Inference". In: *Molecular Therapy* 32.4 (Apr. 2024), Nr 686, p345. DOI: 10.1016/j.ymthe.2024.04.020.

[P3] Judith A. Müller, **Nathalie Schäffler**, Thomas Kellerer, Gerlinde Schwake, Thomas S. Ligon, and Joachim O. Rädler. "Kinetics of RNA-LNP delivery and protein expression". In: *European Journal of Pharmaceutics and Biopharmaceutics* 197 (Apr. 2024), p. 114222. DOI: 10.1016/j.ejpb.2024.114222.

[M4] **Nathalie G. Schäffler**, DJ.J. Truong, Julian Geilenkeuser, Gil G. Westmeyer, and Joachim O. Rädler. "Prime Editing Kinetics of pDNA- versus mRNA-based Delivery". in preparation for Molecular Therapy.

[M5] Judith A. Müller, **Nathalie G. Schäffler**, and Joachim O. Rädler. "Stoichiometric siRNA/mRNA lipid nanoparticles mediate self-regulated protein expression via incoherent feed-forward loop". to be submitted to Nature Communications.

[M6] Judith A. Müller, Manuel S. Torres Matsumoto, **Nathalie G. Schäffler**, Konstantin Steppe, and Joachim O. Rädler. "Ribosomal Competition in mRNA Codelivery".

1. Introduction

"DNA is the software and basis of all life."
J. Craig Venter

DNA, often referred to as the "software of life" [1–3], plays a central role in living systems. It acts as a storage medium for genetic information, regulates the transfer of this information, and serves as a blueprint for cellular functions [4]. DNA has the highest information density in the known universe, approximately 10^{19} bits/cm^3, which is significantly higher than standard flash memory at around 10^{16} bits/cm^3 [5]. DNA also functions as a blueprint for the synthesis of proteins, which are essential for the structure and function of living organisms. The transcription and translation processes of these DNA sequences are collectively known as gene expression, a fundamental process tightly regulated by cells [4].

The ability to influence gene expression has opened new avenues for controlling cellular processes [6], since the use of synthetic DNA enables the encoding of digital data in living organisms [7]. Researchers can, for example, introduce plasmids to deliver new genetic material into cells, allowing the temporary expression of fluorescent proteins or the development of resistance to specific antibiotics [8]. Similarly, mRNA transfection can effectively silence genes or express proteins without permanently altering the host genome [9]. These methods allow researchers to manipulate and study specific pathways or cellular behaviors.

Nonetheless, cells extensively regulate their own gene expression. For example, the presence of specific RNA molecules can lead to the suppression of particular genes [10]. This has been likened to the function of logical gates. Scientists have replicated these logic gate computations using strand displacement, mimicking how silicon-based computers process information [11, 12]. Simple genetic computers and synthetic cells have been constructed [13–15], laying the foundation for the design of more complex systems.

In parallel, researchers have developed techniques to edit the genome of living systems directly, such as zinc finger nucleases, base pair editing, and CRISPR, opening up new possibilities for designing biological computers [16]. However, these methods face specific challenges, including double-strand break stress, off-target effects, low efficiency, and logistical challenges in delivery [17]. Anzalone et al. proposed a new method called prime editing, "a versatile and precise genome editing method that directly writes new genetic information into a specified DNA site" [18]. This expanded possibilities for designing biological computers [19]. Still, efficiency remains one of the main challenges [20]. This has been addressed in various studies, such as improving pegRNA design [21], developing different prime editors [22] or using nicking single gRNAs in combination with pegRNAs [23].

Despite these improvements, most time frames reported for prime editing are on the order of days [18, 22, 24]. When a single computation step on a biological computer using prime editing takes a day to compute, as noted in [19], scaling this up does not compare well with contemporary high-performance computing systems. These can achieve performance levels exceeding 10^{18} computations per second [25]. So, increasing the speed of prime editing is crucial, but understanding the kinetics of prime editing and developing methods to manipulate and optimise them is a prerequisite for making biological computers feasible. Having control over these kinetics can also be advantageous for the design of therapeutic treatments for gene therapies. It has been shown that CRISPR systems tend to accumulate in the liver and have a high chance of off-target effects [26], therefore, some therapies need short-lived yet efficient prime editing [27]. However, this area of prime editing kinetics has not been extensively studied.

With respect to kinetics, prime editing can be simplified as a standard transfer of information: a process from DNA to RNA to protein, followed by the editing process as described by Anzalone et al. [18]. This then leads to a second cascade of DNA-to-RNA-to-protein. Both of these cascades are standard gene expression systems, similar to the expression of fluorescent proteins. The kinetics of these reporters have already been studied using empirical data and mathematical modelling [28–31]. Studies have targeted different parts of the information transfer machinery. For example, transcription speeds can vary widely [32], slow codons in mRNA can reduce the ribosome translation speed [33], and proteins differ in maturation times [30]. These systems are also highly regulated by sequences such as promoters for DNA and untranslated regions (UTRs) for mRNA. Given the complexity of these systems, it is necessary to simplify them and study only small variations.

In this thesis, our aim is to better understand the kinetics of prime editing using a delivery-based as well as a modelling approach. Specifically, we compare the delivery of prime editing components via plasmid DNA (pDNA) and mRNA. To do this, we use a green fluorescent protein encoded in the genome of our cell line as a reporter for successful editing. First, we estimate the parameters of this protein starting from the protein level up to the delivery via pDNA. This allows us to determine delivery times for our experimental setup and delivery methods. Next, we investigate the overall kinetics of prime editing and define the "time-to-edit". This time-to-edit comprises the delivery time, which includes the delivery and expression of the prime editing system, and the editing time itself. We then image the editing time to observe the differences between mRNA and pDNA delivery using a second fluorescent reporter co-expressing with the prime editor. Additionally, we examine the impact of editing length on the editing time. Using similar models as for the parameter estimation of mGL, we model prime editing expression and assess how single parameter variations influence the expression kinetics of our prime editing system. Finally, we propose ideas for establishing a computational system in living cells using DNA and prime editing on the basis of the computational model of the Lambda calculus.

This thesis is structured to systematically build knowledge about the kinetics of fluorescent reporters and prime editing, culminating in the proposal of a DNA-based biological computational system.

Chapter 2 explains the fundamental concepts, covering prime editing and how it differs from the established CRISPR system. The chapter also discusses the general structure of genes and their regulation, followed by an exploration of expression kinetics using fluorescent reporter proteins. In addition, a brief introduction to chemical reactions is provided. The chapter concludes with a short introduction to computer science, offering definitions of key concepts such as computers, computation, and the Lambda calculus.

Chapter 3 describes the experimental methods used in this thesis, including bright-field, phase and fluorescence microscopy, nucleic acid delivery into cells, and a short introduction to Live Imaging on Single-Cell Arrays (LISCA).

Chapter 4 provides an overview of kinetic studies, using the fluorescent protein mGreenLantern (mGL) as an example to introduce kinetic rate equations in the form of ordinary differential equations (ODEs) and the parameter estimation problem. It demonstrates how parameters such as protein maturation and degradation can be extracted using LISCA. The following section applies this approach to extract the translation and mRNA degradation parameters by using mRNA transfection of mGL. Then, pDNA transfection and LISCA are used to extract the transcription and pDNA degradation rates. This allows us to determine and compare the delivery times for pDNA and mRNA. The second half of the chapter explains how the LISCA approach can be adapted to measure the kinetics of prime editing. We introduce a cell line that stably expresses a variant of mGL, which is blue-shifted due to a mutation in its DNA sequence. This change is targeted by our prime editing system and is first validated using bulk measurements. The single-cell assay is then adapted to image prime editing, and a new algorithm is developed to determine the onset of prime editing.

Chapter 5 delves into the kinetics of prime editing, using the assay developed in the previous chapter to compare pDNA and mRNA-based delivery of the prime editing system. It then examines the differences in editing time for both pDNA and mRNA using a second fluorescent protein, which is co-expressed with the Prime Editor (PE). The chapter introduces a mathematical model to explain the previously measured distributions and provides the first model for prime editing expression. Additionally, it explores the correlation between the length of the editing frame and the editing time using pDNA delivery.

Chapter 6 elaborates on the modelling approach for prime editing expression. It begins with the simplest model and discusses its implications for the experimental setup. Two additional model variations are then introduced, demonstrating their influence on the kinetics of prime editing expression.

Chapter 7 lays the foundation for a living computational system, outlining the necessary requirements and challenges that remain to be addressed. It builds an analogy between Lambda calculus, as explained by Alonzo Church, and how this computational model can be translated into DNA-based machinery using prime editing and yeast cultures.

Chapter 8 provides an outlook for future research and proposes directions for studies that build on this thesis.

Additional details on experimental protocols, material production, gene sequences, and data evaluation are provided in **Appendix A**. **Appendix B** contains background data and more detailed figures.

2. Fundamental Concepts

This chapter introduces essential background information used in Chapter 4 to Chapter 6 to study the kinetics of prime editing and their implications for biocomputing as discussed in Chapter 7. First, prime editing and gene expression concepts are introduced and followed by some basic concepts of computer science.

2.1. Gene Expression and Prime Editing

All living organisms rely on deoxyribonucleic acid (DNA), “without DNA there is no life” [34]. DNA is used as a blueprint and regulator for ribonucleic acid (RNA) and protein synthesis, for information storage and biological memory [35, 36]. Bacteria, for example, store segments of DNA from previous viral infections — called the protospacer — in their own genome in a sequence called Clustered Regularly Interspaced Short Palindromic Repeat (CRISPR) [37]. These form part of the bacterial immune system together with a CRISPR associated protein (Cas) and allow bacteria to recognize and target specific viral sequences, enhancing their ability to neutralize subsequent viral infections of the same or similar viruses [38, 39]. The principles underlying this natural defense system are foundational to contemporary gene editing techniques and will be thoroughly examined in the following.

2.1.1. Inherent and Exogenous Genetic Information Transfer

Cells and other organisms rely on functioning proteins, which are produced and regulated by gene expression. These expression processes are explained by the central dogma of molecular biology, which can be modeled by kinetic rate equations as previous studies have shown [P3, 40, 41]. Before scientists were able to even manipulate cells by changing its genome directly, they used various delivery methods to introduce exogenous genetic information in form of plasmid, RNA and Ribonucleoproteins (RNPs) [8].

The Central Dogma of Molecular Biology

The central dogma of molecular biology describes the directional flow of genetic information within biological systems (Figure 2.1). It proposes that genetic information is transferred from DNA to RNA to proteins, showing the sequential process fundamental to gene expression [35]. In a first step, DNA inside the nucleus is transcribed into its RNA counterpart, which then diffuses into the cytosol. There it can get translated by ribosomes to produce amino acids—folding into functional proteins. Each step can

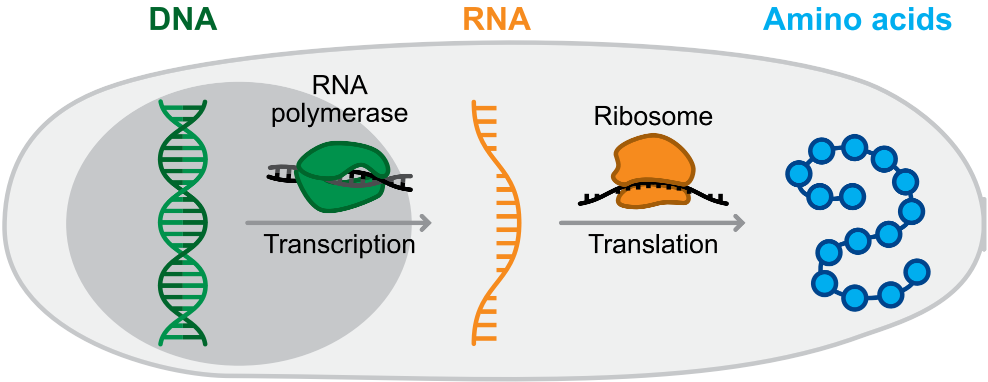


Figure 2.1: The central dogma of molecular biology describes the directional flow of genetic information from DNA over RNA to proteins.

be described by kinetic rates, but for various genes these can range over several orders of magnitude, depending on factors like cell type, promoters, enhancers/silencers, and more.

Modeling the Central Dogma

Previous studies showed that it is possible to model part of the central dogma with kinetic rate equations and extract the rates on a single-cell level using transfected fluorescent reporter [P3, 28, 41–43]. In a simplified view mRNA translation can be described by two equations:

$$\frac{dRNA}{dt} = -\delta RNA(t) \tag{2.1}$$

$$\frac{dP}{dt} = k_{tl} \cdot RNA(t) - \beta P(t) \tag{2.2}$$

Protein (P) is produced from the current amount of mRNA present in the cell with rate k_{tl}. δ describes the degradation rate of transfected mRNA in the cytosol, where as protein degrades with β. Solving these equations with m_0 as starting concentration of mRNA leads to the rate of protein production in a single cell:

$$P(t) = \frac{m_0 \cdot k_{tl}}{\delta - \beta} \cdot (1 - e^{-(\delta-\beta)(t-t_0)}) \cdot e^{-\beta(t-t_0)} \tag{2.3}$$

With these equations and experimental single cell assays, we can estimate the parameters of different proteins, for example, fluorescent reporters.

2.1.2. Core Structures of Genes and Regulation of their Expression Kinetics

Regulatory networks controlling the expression of genes and production of proteins, enzymes and other important parts of the cell machinery, are vast and complex. Yet those networks — either as DNA or mRNA — are usually build with a overarching structure of three key regions. A starting region to bind the transcription or translation machinery, the gene itself and then an ending region. Additionally, genes may include other sequences that further contribute to the regulation and modification of the gene expression.

DNA

Genes start with a promoter region, which always consists of a core promoter sequence. This core promoter sequence offers a binding site for the RNA polymerase and different sequences exhibit varying levels of transcription initiation activity, thus influencing the overall transcriptional output of the gene. Additional sequences like enhancers, silencers, insulators and repressors exist, which fine tune the regulation [44]. The to-be-transcribed gene is located within the open reading frame (ORF), which can usually be categorised in two types of sequences: Exons and introns. Exons are the regions coding for the actual amino acids of a protein and are transcribed into mRNA, depending on length and sequence varying widely in their kinetics [45]. Introns, which are non-coding sequences, also have an influence on the kinetics by intervening with the correct splicing of the DNA, but are removed during mRNA processing. The transcription process concludes at the terminator region, a sequence that signals the RNA polymerase to cease transcription. The sequence of these regions can also have a significant effect on the efficiency of the overall transcriptional output [46].

When transferring genetic information using plasmids rather than stable integration into the genome, additional parts are needed for upkeep of the plasmid, such as the origin of replication (ORI) and selection markers. The ORI is essential for the replication of the plasmid, offering the starting point for DNA replication and therefore for duplication of the plasmids during cell division. Selection markers—often carrying an antibiotic resistance—allow for the selection of cells that have successfully incorporated the plasmid.

mRNA

Transcribed mRNA undergoes several modifications until it becomes mature mRNA. After maturation it includes several distinct regions playing crucial roles in the regulation of gene expression at the post-transcriptional level. As first modification a 5'cap is added to the front of the primary mRNA transcript. This protects the mRNA from degradation, facilitates the export of the mRNA from the nucleus to the cytosol and assists in ribosome binding during translation. Next to the 5'cap and before the start codon is the 5' UTR, a non-coding sequence regulating translation [47, 48] and mRNA stability [49]. The following coding region, flanked by the start and stop codons, is the sequence that is translated into amino acids by the ribosome, forming the final protein. Downstream of this is the 3' UTR, which affects mRNA stability, localization and translational efficiency, by offering binding sites for RNA-binding proteins or small non-coding RNAs [50, 51]. The 5'cap together with the 3' poly-A tail, a stretch of adenine nucleotides, greatly influence the half-life of mRNA [29, 52].

Since all these structural elements and regulatory sequences have different influences on the kinetics, it is crucial to maintain consistency for accurate comparison of gene expression kinetics. Using fluorescent reporters provides a straightforward and effective method to measure and visualize these kinetics by enabling real-time monitoring of gene expression dynamics in systems like LISCA as mentioned in previous studies, which are explained in the following section.

2.1.3. Measuring Kinetic Rates Using Fluorescent Reporter Proteins

Fluorescent reporter proteins are integral tools in biological applications to visualize and track patterns of expressed proteins within living cells and organisms. These proteins emit fluorescence when exposed to specific wavelengths of light, making them crucial for a wide range of applications, including studying gene expression, protein localization, and cellular dynamics. One well-known reporter protein is the green fluorescent protein (GFP), originally derived from the jellyfish *Aequorea victoria* [53–55]. GFP has a barrel structure composed of eleven β-sheets with an α-helix inside that houses the chromophore. By altering this chromophore, scientists were able to develop a range of differently colored fluorescent proteins, excited by different specific wavelengths compared to GFP, like mScarlet, emitting a red fluorescence [56]. These modifications involve changes in the amino acid residues surrounding the chromophore, leading to shifts in the emission spectrum. These different amino acids are produced by small edits in the DNA or mRNA sequence. Additionally modifying certain other parts of the sequences can enhance its brightness and stability, like with mGL, which shines 6x brighter than enhanced GFP (eGFP) [57, 58].

Leonhard et al. [59] compared mRNA and pDNA transfections of eGFP and showed that the onset time for mRNA was usually within 5 h of transfection, where for pDNA it ranged from 2 h up to 20 h. They introduced the simple reaction model already mentioned in Equation 2.2 and the solution Equation 2.3. For eGFP mRNA they extracted the average mRNA degradation rate of $6.2 \times 10^{-2}\,\mathrm{h}^{-1}$ and protein degradation of $5.6 \times 10^{-2}\,\mathrm{h}^{-1}$. Ferizi et al. [51] compared different UTR combinations and their influence on transgene expression. They used a d2eGFP construct, which produces a destabilized eGFP with a faster protein degradation than normal eGFP. Using UTR-stabilized mRNAs of this d2eGFP lead to higher expression levels and longer functional half-lives. Reiser [29] then used a three-stage model for eGFP, where a maturation step for the protein was introduced, and found that longer polyA tail lengths increased the expression of fluorescent protein. Finally Krzysztoń et al. [28] also used the three-stage model to fit fluorescence traces of eGFP and CayenneRFP mRNA transfections into cells to extract the mRNA expression kinetics. They found mean protein degradation rates for eGFP and CayenneRFP to be $5.22 \times 10^{-3}\,\mathrm{h}^{-1}$ and $4.21 \times 10^{-2}\,\mathrm{h}^{-1}$ and maturation rates of $1.28\,\mathrm{h}^{-1}$ and $0.3\,\mathrm{h}^{-1}$ respectively. Kinetics like these can be modeled using the *Catalyst.jl* package in the programming language

Julia [60], which will be applied for mGL in Chapter 4.

2.1.4. Chemical Reactions and their Kinetics

Chemical reactions and their rates provide critical insights into how processes unfold over time. The rate of a chemical reaction refers to the speed at which the concentration of reactants or products change over time. Different reaction types are defined as follows, based on [61, 62].

First-Order Reactions

A first-order reaction is based on the conversion of substrate A into substrate B. For example, the maturation of an amino acid chain (AA chain) into a functioning protein is a first-order reaction. In this process, substrate A is consumed to produce substrate B. Here, the production rate v_{mat} of the protein is dependent on a rate constant k_{mat} and the concentration of available protein A.

$$A \rightarrow B \tag{2.4}$$

$$v_{mat} = k_{mat} \cdot A(t) \quad or \quad -\frac{dA}{dt} = k \cdot A(t) \tag{2.5}$$

The concentration of A decreases over time as it is used up to form B. The system explained in Equation 2.2—for example—is set up of 3 first-order reactions.

Pseudo-First-Order Reactions

A pseudo-first-order reaction is similar to a first-order reaction, but the initial substrate A is not used up to produce B and stays constant throughout the reaction. Simplified to a single step of transcription, the central dogma can be described as such a pseudo-first-order reaction rate law. The presence of the DNA A leads to production of RNA B without the DNA being used up. The rate of production v_{tc} of B is still proportional to the concentration of A and has a rate constant k_{tc}.

$$A \rightarrow A + B \tag{2.6}$$

$$v_{tc} = k_{tc} \cdot A(t) \tag{2.7}$$

Second-Order Reactions

Second-order reactions are given in a system, where two substrates (A, B) combine to form a new product C. The two initial substrates are being depleted, leading to a rate v, which is dependant on the concentrations of the two reactants and a rate constant k:

$$A + B \rightarrow C \tag{2.8}$$

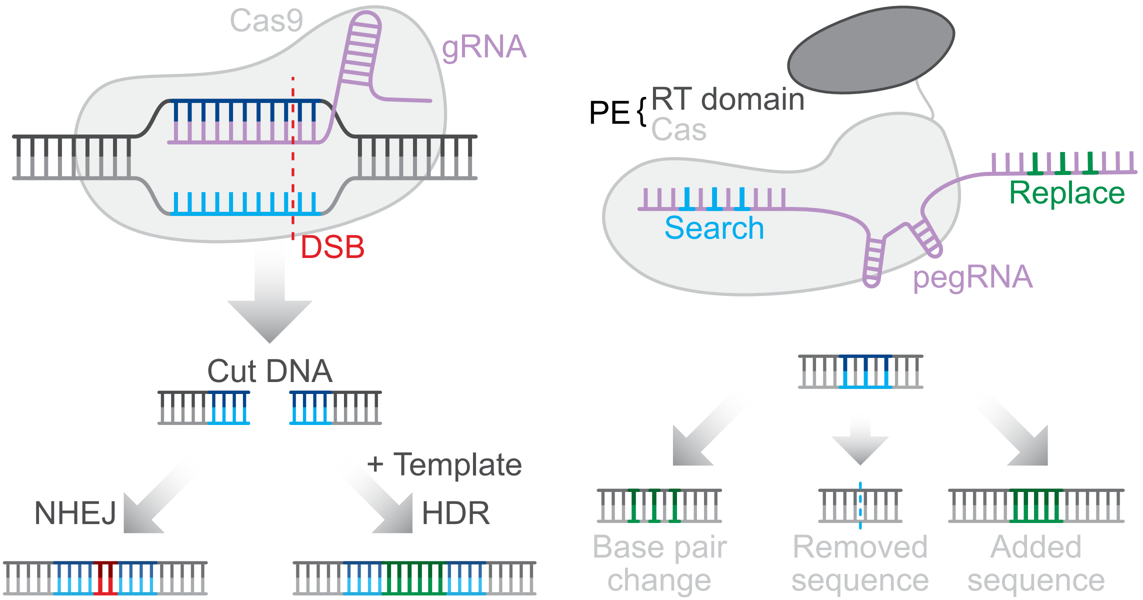
$$v = k \cdot A(t) \cdot B(t) \tag{2.9}$$

Catalytic Reactions

Generally, when a substance's A presence (often enzymes) leads to the production of a second substance B, one speaks of a catalytic reaction. A is called the catalyst and is not used up by the production of B. Rate laws for one and two substrates are similar to first and second order reactions, but A is generally constant, when not depleted through another path.

2.1.5. Gene Editing

By leveraging the "search-and-cut" technique derived from bacteria [63] and combining it with the knowledge about repair mechanisms inside different organisms, researchers developed gene editing tools that enable precise modifications to an organism's DNA [64]. This advancement allows scientists to correct genetic defects, enhance traits, and study gene function [16]. Techniques such as CRISPR/Cas and prime editing have revolutionized gene editing by providing a highly efficient and versatile method for targeted genetic alterations in a wide range of organisms [65].



A: The CRISPR/Cas system consist of a gRNA and a Cas protein to enable it to cut DNA. After the DNA has been cut, the cellular machinery repairs it again. This can either be done by non-homologous end joining (NHEJ) to create a frame-shift by indels or homology-directed repair (HDR) when provided with some template DNA shown in green.

B: In prime editing, the Cas protein is fused to a reverse transcriptase (RT) domain to form the PE. The second end of the gRNA contains the edit. Using prime editing we can produce all base pair changes, remove and add extra sequences very specifically.

Figure 2.2: Comparison of CRISPR/Cas and prime editing

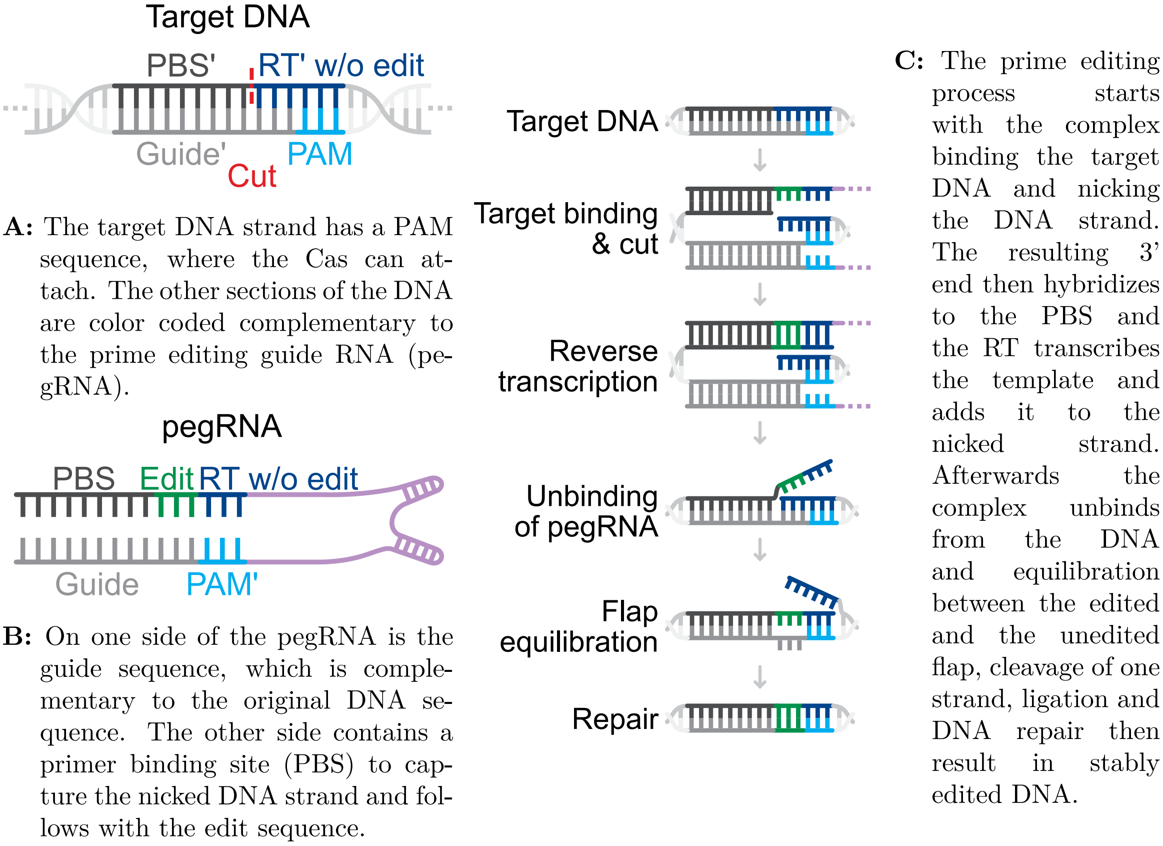
CRISPR/Cas

As previously mentioned, the CRISPR/Cas system consists of two components. The first component is the guide RNA (gRNA), which is produced from the CRISPR region and is complementary to a specific DNA sequence. The second component is the RNA-guided DNA-nicking domain or Cas protein, which has the ability to cut DNA strands. This Cas protein contains nuclear localization signal (NLS) sequences facilitating the transport of the protein into the nucleus towards the genome of cells. There the Cas protein can bind to a protospacer adjacent motive (PAM) sequence, which enables the gRNA to split open the target DNA into two single strands. The Cas can then cut either both or just a single strand, depending on the specific protein used (Figure 2.2A). The CRISPR/Cas complex then releases the DNA again and the cellular machinery tries to repair the DNA. For double strand break (DSB), HDR is performed, when the cell is supplied with a template DNA which it incorporates, this way long edits with specific sequences can be accomplished. Otherwise the cell will try to repair the DSB and in the process might loose or gain some base pairs at the break, which leads to a frame-shift and can then disable following genes. This is called non-homologous end joining (NHEJ).

Prime Editing

Prime editing is a new evolution of the CRISPR/Cas system developed by Anzalone et al. [18]. Unlike traditional CRISPR/Cas techniques, prime editing creates a single strand break (SSB) instead of a DSB. Additionally, it provides a template along with the gRNA to enable every base pair edit, including the adding of missing sequences at predetermined loci and the accurate deletion of specific sequences.

For this to work the Cas—here specifically SpCas9 (Streptococcus pyogenes Cas9)—is fused to a reverse transcriptase (RT) domain (see Figure 2.2B) to form the PE. The gRNA is expanded to include the template for the edit (Figure 2.3), and this modified RNA is referred to as the prime editing guide RNA (pegRNA). Together, the PE and the pegRNA assemble to build the prime editing complex. To perform an edit, the following five steps are then carried out (Figure 2.3C). First the PE-pegRNA complex binds the target DNA. The guide sequence of the pegRNA, which is complementary to the part of the DNA, where the edit is supposed to happen, hybridizes to it. This enables the Cas to cut the unbound strand. The primer binding site (PBS) of the pegRNA binds the nicked strand and the template, following the PBS on the pegRNA, is then reverse transcribed onto this nicked strand. Once the PE-pegRNA complex is unbound again, the flap equilibration takes place. This is a thermodynamical process and can be described as branch migration as explained in the following paragraph. DNA repair then resolves the double flap and SSB to produce either an stably edited DNA or the original sequence. Since the original sequence can be bound again by the guide, if the edit was not incorporated and PE-pegRNA complex is still available, the whole process can start anew, which results in high editing efficiencies over time [18].



A: The target DNA strand has a PAM sequence, where the Cas can attach. The other sections of the DNA are color coded complementary to the prime editing guide RNA (pegRNA).

B: On one side of the pegRNA is the guide sequence, which is complementary to the original DNA sequence. The other side contains a primer binding site (PBS) to capture the nicked DNA strand and follows with the edit sequence.

C: The prime editing process starts with the complex binding the target DNA and nicking the DNA strand. The resulting 3' end then hybridizes to the PBS and the RT transcribes the template and adds it to the nicked strand. Afterwards the complex unbinds from the DNA and equilibration between the edited and the unedited flap, cleavage of one strand, ligation and DNA repair then result in stably edited DNA.

Figure 2.3: Schematic overview of prime editing

Branch Migration

Branch migration is part of the strand displacement theory, which describes an input and an output DNA strand competing for hybridization to a third complementary strand. Branch migration specifically refers to the movement of the crossover point during the base-pair-wise displacement of the incumbent strand by the invader strand. Thompson et al. [66] described branch migration as a one-dimensional random walk problem, where the branch point moves randomly in either direction with equal probability. Over N steps, this can be modeled by a binomial distribution:

$$P(k, N, p) = \binom{N}{k} p^k (1-p)^{N-k} \tag{2.10}$$

with k as the number of successes (e.g. displacing the incumbent strand) in N independent steps, with each having the probability p of success. For large N, this distribution approaches a normal distribution due to the *Central Limit Theorem*, highlighting the stochastic nature of branch migration. The mean squared displacement of the branch point, $\langle x^2 \rangle$, scales linearly with time t, following the relation $\langle x^2 \rangle = 2Dt$, where D is the diffusion coefficient. The time t required for the branch point to

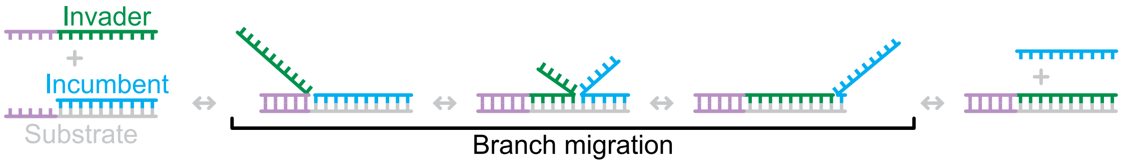


Figure 2.4: Schematic of strand displacement and branch migration: The first base pairs of an invader (green) strand can bind to a complementary region of the target strand (purple), which is currently bound to an incumbent strand (blue). Afterwards the invader strand pushes the incumbent strand away through a process called branch migration. Adapted from [67].

migrate a specific length L can be expressed as:

$$t(L) = \frac{L^2}{2D} \tag{2.11}$$

Thus, the time required for the branch point to migrate a specific length scales linearly with that length squared.

2.2. Definitions and Basics of Computer-Science

As we approach the physical limits of traditional silicon-based computing, as evidenced by the slowing progress of Moore's Law — the observation that the number of transistors on a chip doubles roughly every two years — new, innovative alternatives are being sought [68, 69]. Two approaches currently being pursued are quantum computing and biology-based computing [70, 71]. The latter, which leverages biological components like DNA and cellular machinery, offers the potential for massive parallelism, energy efficiency, and the ability to solve complex problems where classical computers struggle. In this thesis, we aim to provide an outlook on how prime editing could be the missing link in building a biocomputer. To set the stage, we will introduce a few basic concepts of computer science.

2.2.1. Defining a Computer

Traditionally, the term "computer" first referred to humans performing calculations [72], then to an electronic device designed to process, store and execute instructions to perform a wide range of tasks. These "digital computers" operate by manipulating data based on programmed instructions, enabling them to solve problems, perform calculations, and manage information.

A computer can be divided into two main parts: hardware — the tangible parts that make up the machinery and physical aspects of the device — and software — the intangible programs and operating systems that run on the hardware.

Software relies on algorithms, which are well-defined, step-by-step instructions or sets of rules designed to perform a specific task or solve a particular problem. Algorithms have a finite number of steps that must be clearly and unambiguously defined. They may have zero or more inputs, but they must produce at least one output. In principle, they should be simple enough, that a person with a pencil could reproduce them.

Each step of an algorithm that manipulates data to achieve the desired output is called an “operation”, such as arithmetic (addition, subtraction) or logical operations (and, or).

2.2.2. Theory of Computation

When considering a biocomputer, one naturally encounters the theory of computation and its central question: “What are the fundamental capabilities and limitations of computers?” [73]. This theory examines how efficiently problems can be solved using algorithms within a given model of computation. It is divided into three main branches: automata theory, computability theory, and complexity theory. In the following, we will focus on the first two:

Automata Theory

Automata theory studies abstract machines — automata — and provides a formal framework for modeling and analyzing the behavior of systems and processes. The most powerful type of automaton is the Turing machine, which can simulate any algorithmic process and forms the foundation of the concept of Turing completeness.

Turing Completeness

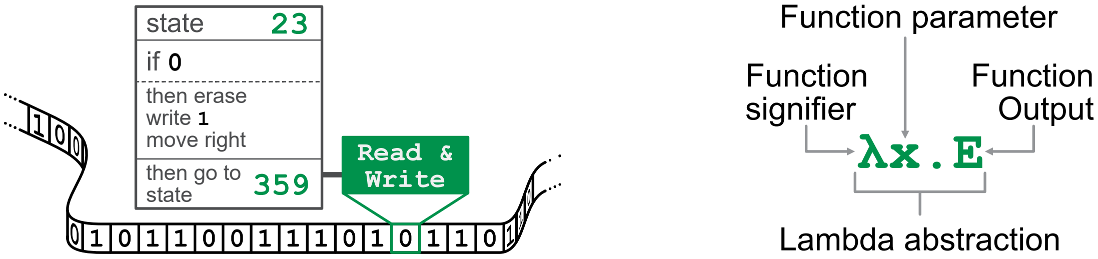
A system is considered “Turing complete” if it can perform any computation that can be described algorithmically, given enough time and resources. If it can simulate a Turing machine and execute any algorithm that a Turing machine could, it is deemed Turing complete.

2.2.3. The Lambda Calculus — A Model of Computation

A model of computation is a formal system used to describe how computation is performed. It provides a framework for understanding the steps, rules and resources required to carry out computation [74]. Models of computation can be classified in three categories: sequential models, functional models, and concurrent models. Examples of the first two will be discussed below:

Turing Machines

A Turing machine, introduced by Alan Turing in 1936, belongs to the category of sequential models. It consists of an infinite tape divided into discrete cells, a tape head that can read, write, and move along the tape, and a finite set of rules that dictate the machine’s operations based on the current symbol being read [75, 76]. At each step, the machine reads the symbol, modifies or leaves it unchanged according to the rules, and moves to the next location. The Church-Turing thesis states, that any



A: A Turing machine consists of an infinite tape divided into cells, a read/write head, and a set of rules. The tape is used to store and recover information, while the head manipulates the symbols on the tape according to the given set of rules.

B: A Lambda calculus abstraction, defining the function signifier λ, function parameter x, and the function output E.

Figure 2.5: Two models of computation

function that can be computed algorithmically can be computed by a Turing machine, linking the concepts of computation and decidability [77].

Basics of Lambda Calculus

As a function model, the Lambda calculus was first introduced in 1936 by Alonzo Church, shortly before Turing machines [78, 79]. It is is based on function abstraction and application and consists of three main components, also showed in Figure 2.5B:

- **Variables** are symbols that represent values or act as placeholders, usually denoted by x, y, or symbols like $\star$

- **Lambda abstractions** represent function definitions. A variable is bound to an expression and the general form of this is shown in Equation 2.12.

  $$\lambda x.E \tag{2.12}$$

  λ signifies the function, serving as a visual marker for defining a function. x is the function's parameter, and E is the expression that defines the output of the function. The dot just separates the input from the function body.

- **Function application** refers to applying a function to an argument and evaluating it by substituting the function parameter with the input argument. This substitution is called beta reduction. For example, when an abstraction is applied to a number:

  $$\lambda x.x + 1(2)$$

  We replace every occurrence of x in the function definition with the input, performing beta-reduction.

  $$2 + 1 \rightarrow 3$$

The Lambda calculus is a highly minimalistic model of computation, yet it is capable of expressing any computable function and is therefore Turing complete. Lambda abstractions, or functions, are of a higher order, meaning that functions can be passed as inputs to other functions and also returned as output.

So far, we have only discussed single inputs, but Lambda calculus can also handle multiple inputs through a process called "currying". Currying transforms a function that takes multiple arguments into a sequence of functions that handle these arguments one at a time. Instead of providing all arguments at once, we provide them sequentially. For example, a function $f(x, y)$ is then written as $f(x) \to g \Rightarrow g(y) \to h$. In Lambda calculus, a Lambda abstraction with multiple inputs can be written as:

$$\lambda x.\lambda y.x + y \tag{2.13}$$

To evaluate this expression, we can write it as nested functions and then sequentially apply beta reduction.

$$\begin{aligned} \lambda x.\lambda y.x + y \;\; \to \;\; & \lambda x.(\lambda y.(x + y)) \\ & \lambda x.(\lambda y.(x + y))(1)(2) \\ & \lambda y.(1 + y)(2) \\ & 1 + 2 \\ & 3 \end{aligned}$$

Church Numerals in Lambda Calculus

In 1941 Church represented natural numbers in his Lambda calculus in the following way, therefore named Church numerals [80, 81]: the natural number n is represented by a function which recursively applies to itself n times, similar to the factorial function, which is defined as $n! = n \cdot (n-1)!$. In Lambda calculus this then looks something like this:

$$\begin{aligned} 0: &\qquad \lambda f.\lambda x.x \\ 1: &\qquad \lambda f.\lambda x.fx \\ 2: &\qquad \lambda f.\lambda x.f(fx) \end{aligned}$$

The church numeral 2 is not exactly the same as the number 2, it simply states that any given function is applied 2 times to a value. It therefore only returns the number 2 if the applied function is the successor function $(x + 1)$ and the value is 0.

2.2.4. SKI Combinator Calculus

Derived from Lambda calculus, the SKI combinator calculus is a simplified model of computation that eliminates the need for variables, instead representing all functions using three basic combinators: S, K and I [82]. Together, these combinators can reproduce any Lambda expression. Since this calculus is Turing complete, it can serve as the foundation of any computation [83].

The combinators are specifically:

- **I, the Identify function,** which returns its argument:

$$\lambda x.x$$

- **K, the Kestrel function,** a constant function that takes two arguments and always returns the first:

$$\lambda x.\lambda y.x \tag{2.14}$$

- **S, the Stirling function,** also called the "substitution" function. It takes three arguments and currys them by applying the first to the third, and this then to the results of the second applied to the third:

$$\lambda x.\lambda y.\lambda z.(xz(yz)) \tag{2.15}$$

In SKI combinator calculus, variables are not needed. Everything can be expressed using the three combinators. For 0, this can be represented as follows:

$$0: \quad KI \tag{2.16}$$

At this point one might wonder whether KI works, as K is typically defined for two inputs. However, in Lambda calculus, the number 0 is defined as a function that is applied zero times to a value x. In SKI calculus, we therefore need a way to ignore the function f and return x. This is achieved by using K with only one input. The subsequent numbers are then encoded as follows:

$$1: \; I \tag{2.17}$$

$$2: \; S(S(KS)K)I \tag{2.18}$$

For a deeper understanding of the formulation of these numbers, Section B.2 provides the explicit calculations for the Church numeral 2.

2.2.5. Current Biocomputing Approaches

Bellia et al. [71] described a theoretical system that uses a version of the SKI combinator calculus implemented with DNA-tiles. Each tile has a single-stranded DNA (ssDNA) at each corner and can form complexes through hybridizing these to others. The computation of a problem is carried out through the order of tile assembly. This is similar to previous studies that used DNA strand displacement and logic circuits to perform computations [84].

While Shapiro et al. [85] described DNA computers based on Turing machines, they discussed utilizing various biological components with their inherent characteristics — for example, using DNA as both a storage medium and a computational tool, akin to the tape in a Turing machine. They proposed using cellular machinery like the Fokl enzyme to recognize and cut DNA. However, this approach has only been demonstrated in a cell-free system and not withing a living cell.

Siuti et al. [86] demonstrated that DNA-encoded Boolean logic can be implemented in *Escherichia coli* cells, with polymerase chain reaction and fluorescent reporters as output. Although Boolean logic is commonly used, for example, in hardware like central processing units, it is not Turing complete and is thus limited in its applications.

Despite various approaches to biocomputing, including those using Lambda calculus, many do not fully exploit the potential of both Lambda calculus and biological systems, as most of these systems operate in cell-free systems. Additionally, many approaches require predefined problems that are translated into predetermined DNA strands, limiting their flexibility.

3. Methods

3.1. Experimental Setup

3.1.1. Microscopy

Microscopes are key tools in biology and other research areas, particularly in studies using living organisms. By allowing scientist to observe the structure, dynamics and behaviour of cells, one can for instance gain insights into the kinetics of gene expression [87]. There are a range of different microscopes and techniques depending on the specific application and model system.

Brightfield and Phase Contrast Microscopy

A basic light microscope uses visible light to illuminate and a series of lenses to magnify the desired sample. Here brightfield (BF) microscopy is the most common techniques, which illuminates from the top or bottom of the sample. By guiding the light through the specimen and focusing it with the use of lenses, the magnified image is projected onto the eyepiece or camera. The real image can be further magnified by the ocular and the contrast seen is dependant on the absorption of light of the sample. BF microscopy is one of the simplest methods, but limited by low contrast in weakly absorbing samples and blurring of 3D materials, when out-of-focus [89, 90]. Phase contrast (PC) microscopy addresses this issue by using non- and phase shifted light to create constructive interference for converting those phase into amplitude

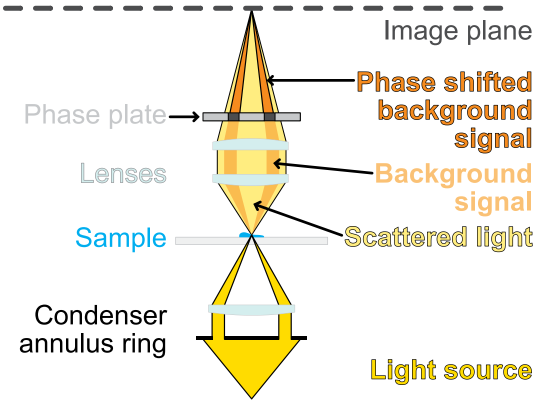


Figure 3.1: Phase contrast (PC) microscope guides light through an annular ring before passing through the sample. The sample leads to diffracted and phase shifted light. Both the background light and the diffracted are collected by a phase plate and projected onto the image plane. Adapted from [88].

shifts. Here light is directed through a condenser annulus positioned in the front focal plane (Figure 3.1). Light then either passes through the sample undeviated creating a background signal or is diffracted and phase-shifted by structures in the sample. The contrast is then enhanced by passing the background light through a phase ring, which shifts this light by one-quarter wavelength to create constructive interference between undeviated and diffracted light. The image plane now shows the sample with increased contrast depending on the phase-shift properties of the sample.

Fluorescence Microscopy

Fluorescence microscopy takes this a step further by using fluorescent dyes and proteins to visualize specific structures or molecules within a sample. Light from a white light source is passed through excitation filters, allowing only wavelengths capable of exciting the fluorophore to reach the specimen. The light is then reflected by a dichroic mirror and focused onto the sample via the objective lens. Upon excitation, the fluorescent molecules within the sample temporarily transition to a higher energy state. Once returning to the ground state, photons of a longer wavelengths are emitted. A portion of this emitted light is collected by the objective lens again, passes through the dichroic mirror, is filtered through emission filters to remove residual excitation light and passed to the camera or eyepiece for visualization. With a broad spectrum of fluorescent proteins available and as many different excitation and emission filters to excite and capture their emitted light, multiplexing of these proteins and their simultaneous observation within the same sample enables for live imaging of cellular dynamics [91, 92].

Using PC and fluorescence microscopy techniques provides a robust framework for studying cellular and molecular processes. PC offers the possibility to visualize transparent, live cells with enhanced contrast without a need for staining or similar techniques, while fluorescence microscopy enables the specific labeling and tracking of molecules through their unique excitation and emission properties. Then capturing sequences of images of the same cells over an extended period of time, called time-lapse microscopy, allows for insights into dynamic cellular behaviours [93, 94]. This approach is particularly valuable for studies investigating gene expression and their associated kinetics, since they allow for monitoring of dynamic changes in gene activity and protein localization in real time [95].

3.1.2. Delivery of Nucleic Acids

To study the function of specific genes and their production, such as fluorescent proteins, there often is a need to introduce this genetic information into the cells, this is called transfection. There are various highly efficient transfection methods including physical methods like electroporation and microinjection, chemical methods such as liposomes and lipid nanoparticles (LNPs) or biological methods using viral and non-viral vectors. The goal of transfection can be categorized into two types: for stable integration or for transient expression. Stable transfection involves integrating the genetic material into the host genome and therefore leading to continuous or switchable, long-term expression [8, 96–98]. In contrast, transient transfection involves temporary expression of the introduced nucleic acids, commonly used with pDNA or messenger RNA (mRNA).

Each transfection method has its own advantages, applications and limitations [98]. Physical methods, such as microinjection have efficiencies of almost 100 % and can introduce any type of molecule (DNA, RNA, RNPs), but are limited by low cell numbers, high costs and are very labor intensive. Virus-based methods like using adeno-associated viruses are limited to DNA transfection and offer not only no prolonged expression, but also don't integrate into the host genome. Chemical methods are the most widely used methods. They rely on positively charged chemicals like LNPs, which form complexes with the negatively charged nucleic acids. These complexes are then attracted to the also negatively charged cell membrane, where the uptake is likely due to processes including endocytosis and phagocytosis. The transfection efficiency is highly dependant on factors such as the ratio between nucleic acid and chemicals, solution pH, and cell membrane conditions. However compared to virus-mediated methods, they offer more flexibility with cargo size, lower cytotoxicity and no mutagenesis. Because of this high flexibility different chemical methods for pDNA and mRNA transfection were chosen in this thesis.

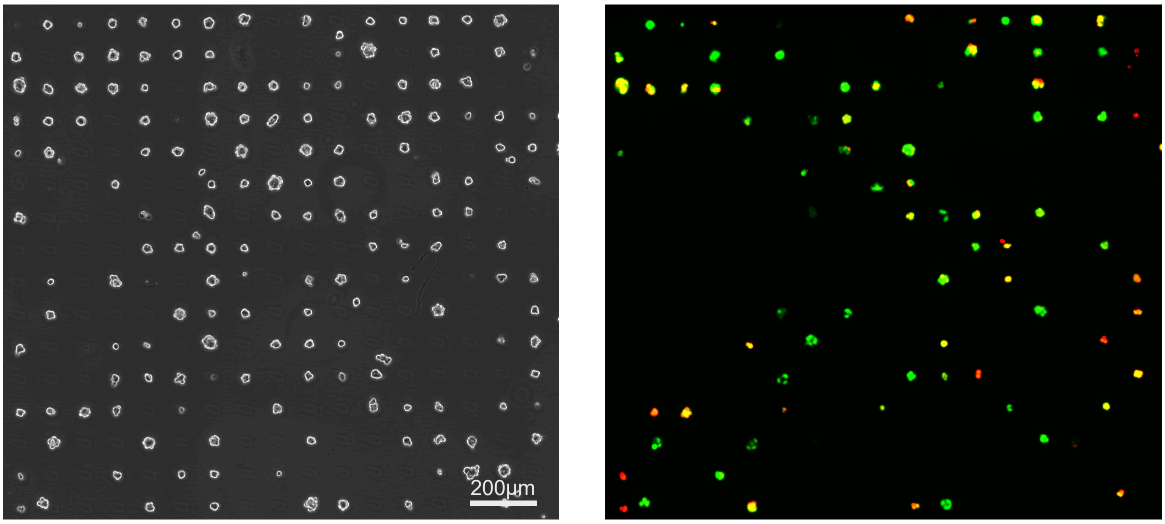
In transient transfection the choice of nucleic acids also determines the transfection method and the resulting expression. pDNA transfection involves the need for delivery into the nucleus, which often leads to a delayed and spread out (2 to 20 h) but long-term, high-level expression of target genes [P3, 59]. In contrast, mRNA transfection introduces mRNA directly into the cytosol, allowing for rapid expression of the encoded protein (mainly within 5 h after transfection). However, mRNA is prone to fast degradation with rates for eGFP in the scale of 0.30 h^{-1}, resulting in a transient expression profile [28]. More details about this are discussed in the following chapters 4-6.

3.1.3. Patterning

Cells live in a complex environment inside the human body, so to investigate single parameters scientist strive to replicate these *in vitro* in a controlled, but simplified manner by creating micro-arrays where adherent cell lines are restricted in their movement [99–101]. These controlled micro-environments allow for the precise study of cellular processes such as migration, cell-cell interaction, and gene expression at the single-cell level [102, 103]. Spatially constraining individual cells can significantly reduce noise in single-cell assays and facilitate the acquisition of sufficient statistical data, which is often challenging with bulk measurements due to the inherent movement and variability of cells [51, 104].

In this thesis, a photopatterning approach was utilized to achieve a precise placement and easy trackability of cells as previously reported in [105] — termed LISCA. Initially, the slide is coated with a cell-repellent layer or purchased precoated. A photoactive compound such as PLPP is then applied to selectively remove this coating upon activation by UV illumination. To attain a specific pattern, a mask is used; for example, for single-cell adherence, squares of a size comparable to individual cells arranged in a lattice grid can be used. After UV illumination, the resulting slide predominantly features a cell-repellent surface with small, uncoated islands. These islands can then be coated with various substances, such as laminin, poly-l-lysin, and fibronectin, to provide tailored environments for different assays and cell types through different shapes and coatings. Once cells are seeded into the slide's channels, they migrate along the substrate and adhere to the exposed, coated areas. Non-adherent cells can be removed after an optimal adherence time specific to the cell line.

A photopatterning approach like this allows for fast, reliable sample preparations and ease of reproducability, making it a valuable tool for creating consistent and controlled patterns for single-cell studies. Although it offers high flexibility in using various coatings, this technique requires a premade mask and quick handling of the substances during the patterning process. It is also limited to adherent cell lines. Despite these challenges it remains a powerful method predominantly used in this thesis.



A: Brightfield

B: Fluorescence channels

Figure 3.2: Cells on a single-cell pattern 50h after seeding. The pattern can be seen through the empty spaces in Figure 3.2A. Not all cells were successfully transfected or edited as seen in Figure 3.2B.

3.2. High-Throughput Processing of Time-Lapse Microscopy Data

Studying gene expression on a single-cell level, a high-throughput of data is crucial to detect subtle changes in cellular responses [P3, 105, 106]. Therefore one strives to automate most of the evaluation process, for instance the background correction and cell tracking, then store data in an easy to use format for subsequent analysis. The in-house software PyAMA [107] automatically converts raw microscopy data into quantifiable fluorescence and cell area metrics. First fluorescence data is background corrected to remove any noise sources such as fluctuations of illumination, based on a published algorithm for fluorescence time-lapse data [108]. Afterwards cells are automatically tracked with a nearest neighbour tracking, made possible through the patterned surface. Finally area, fluorescence signal and image data is shown for each captured position and the user can deselect cells with unwanted behaviour through a graphical interface (Figure 3.3). This data is then stored in an easy format for further evaluation.

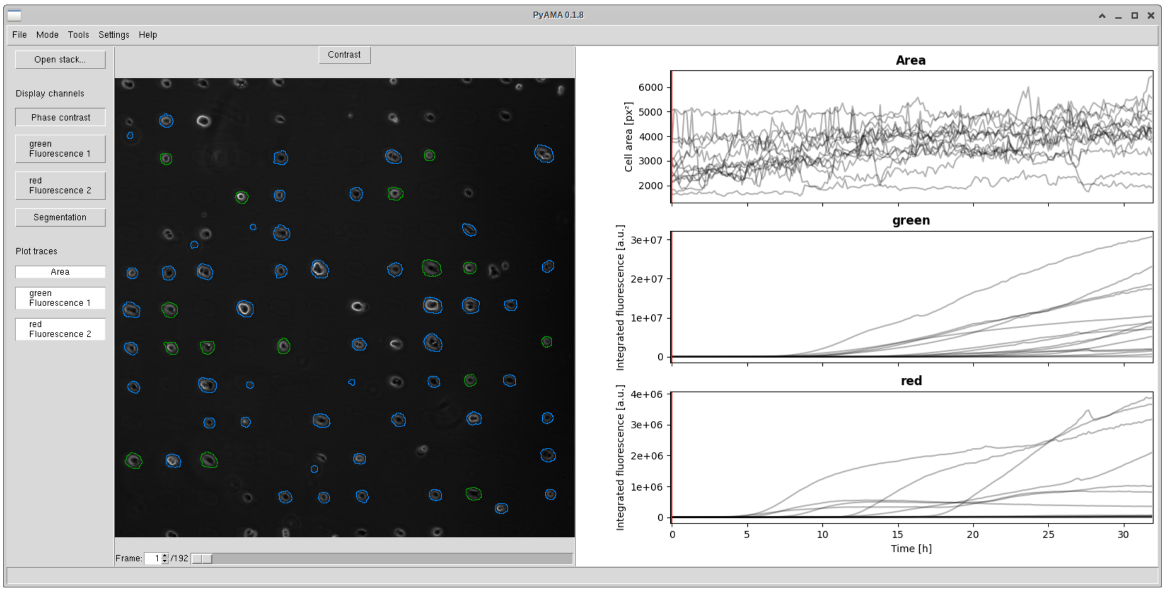


Figure 3.3: Graphical user interface of PyAMA. Channels are chosen on the left side. In the middle is the corresponding image of cells, showing selected cells as green and deselected as blue. Area and both fluorescence channels for all selected cells are shown on the right.

4. Measuring Kinetics of Gene Expression and Prime Editing

Gene expression dynamics have been extensively researched over the last few years [105], as understanding these dynamics leads to deeper insights into cellular processes, the ability to predict and manipulate biological responses, and the development of targeted therapies for complex diseases such as cancer and genetic disorders. This control and understanding becomes even more important when using gene editing tools, due to their potential harmful off-target effects. To study kinetics, simple models are often employed to simulate the production and flow of information by using ODEs [109]. This approach has already been used in previous studies, highlighting various factors that influence the kinetics of fluorescent reporter expression, as mentioned in Subsection 2.1.3.

In this chapter, we first aim to reliably determine the kinetic parameters of the fluorescent reporter mGL. Given that cell populations are inherently heterogeneous, a substantial amount of cells is needed to capture the whole population through the distribution of the kinetic parameters. This is achieved similarly to [28, 29, 105] by using micro patterned single cell arrays and kinetic rate equations in the form of ODEs to fit the extracted fluorescence traces. From these fits, we can then extract onset times for two different transfection methods used in the prime editing comparison studies later on.

Next, we introduce a new prime editing system, in which a HEK293T cell line, hereafter referred to as “HEK”, stably expresses a modified mGL protein, expressing blue fluorescence due to a change in its DNA sequence. Our prime editing system targets this section in the genome and reverts it to the original mGL sequence, enabling the cell to express green fluorescence. We first quantify this in bulk measurements and then adapt the previously employed single-cell assays to image the prime editing event. Since prime editing is a more complex process compared to gene expression studies mentioned earlier, a new approach of determining the onset time of fluorescence was necessary, as explained in the final section.

4.1. Measuring mGL Kinetics

To reduce the complexity of our pDNA transfection system, we step-wise measure and model the central dogma. First, by using translation inhibition, we can determine the maturation and degradation constant of the protein mGL. Secondly, when transfecting mGL as mRNA, we can fit the extracted fluorescence traces with a slightly more complex model and receive distributions for translation and mRNA degradation. Lastly, using all the previously attained parameter values, we can use them to fit the fluorescence traces from pDNA transfection experiments, which will also be the structure of the following sections.

4.1.1. Extracting Maturation and Degradation Rates of the mGL Protein through Translation Inhibition Experiments

To measure the degradation of mGL, we used HEK-mGL cells. For this, HEK-bsmGL cells (see Appendix A.2.2) were prime edited previously and then seeded on a slide patterned for single-cell adherence as described in Subsection 3.1.3. To measure the degradation of the protein, cells were treated with cycloheximide (CHX), which leads to a complete stop of translation. Fully assembled AA chains can still fold and mature into a functional protein, but no new AA chains are produced. Both, matured and non-matured protein degrades as outlined in Figure 4.1. t_0 here is the time point of addition of the CHX. We extracted the fluorescence data using the software PyAMA as described in Section 3.2 and then modeled the kinetics in *Julia* using mainly the package *Catalyst* and it's dependencies [60]. We used the package *PEtab.jl* [110] to fit our experimental fluorescence traces with the model shown in Figure 4.1 and used values from literature as starting parameters [28, 29, 59]. Here and in the following, solving ODEs was done with the Rodas5P algorithm [111] and the parameter estimation with the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm [112]. In the *PEtab* model, β_{Prot} was set to be always smaller than k_{mat}, as from a mathematical stand point both parameters could be switched (seen in [29]). Since *PEtab* does not allow for direct dependencies between parameters, we defined $\beta_{Prot} = k_{mat} \cdot x$ with $0 < x < 1$ and later calculated the true β_{Prot} value with the same equation. The corresponding code can be found in [113].

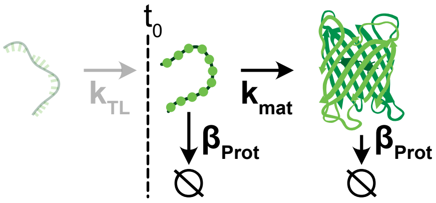


Figure 4.1: Model sketching protein maturation with k_{mat} and degradation with β_{Prot} for mGL. Translation was stopped through the addition of CHX, indicated by t_0.

Figure 4.2 shows a set of single-cell fluorescence data with some exemplary cells highlighted. In the first hour of the experiment the fluorescence increases slightly, since the non-matured AA chains fold into matured protein as seen in Figure 4.2. Afterwards, mGL protein degrades and therefore the fluorescence signal decreases over time. The same selected cells from the top are shown in the middle, overlapped with their respective fits. Here we can see that the fits of the model align well with the experimental data, with reasonable variations in their residuals. We started with cells stably expressing the fluorescent protein from its genome and therefore at a lower

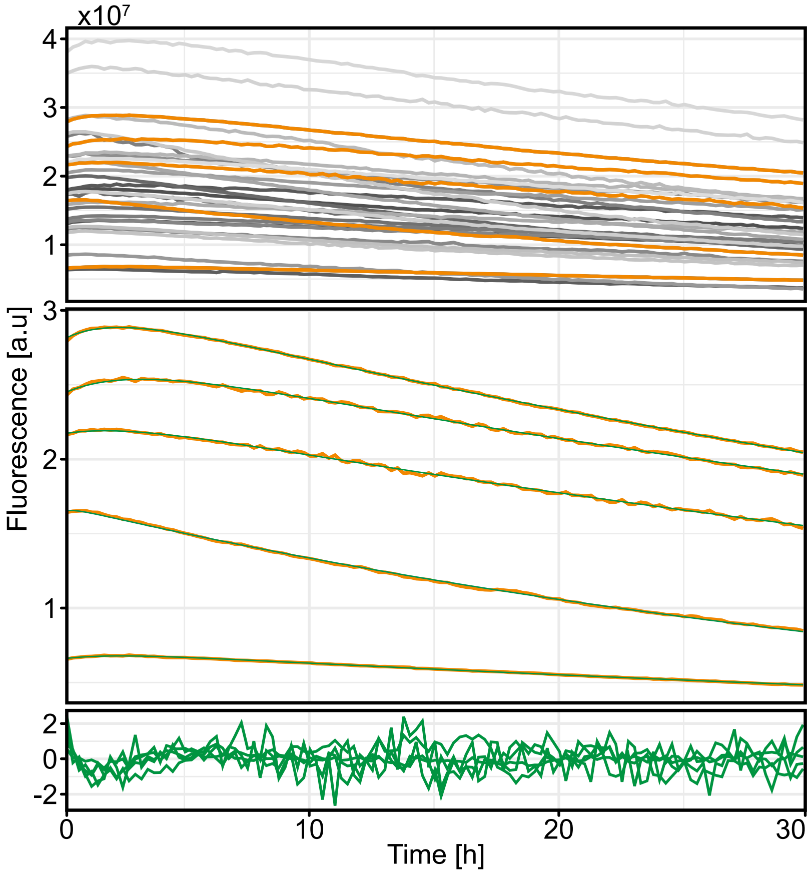
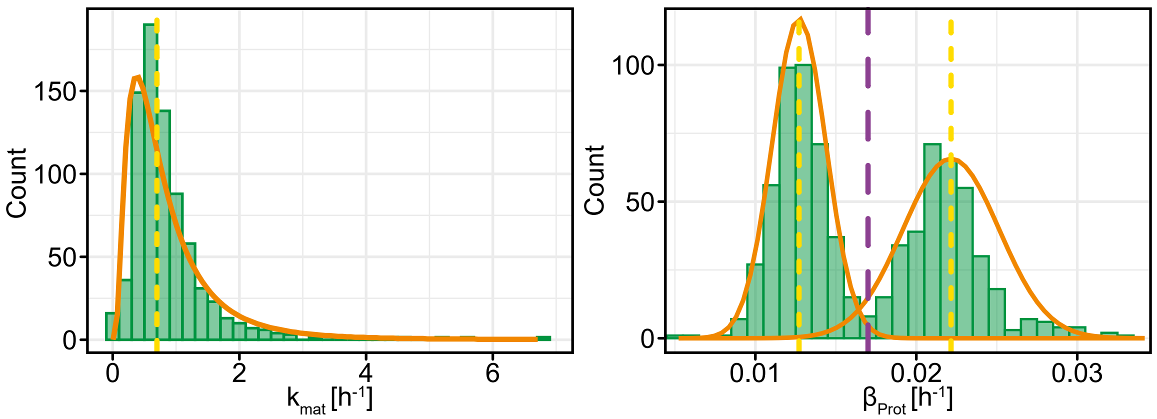


Figure 4.2: Single-cell fluorescence data from stably expressed mGL in HEK cells. Top shows exemplary experimental data with 5 representative cells highlighted in orange. These are shown with their corresponding green fits and residuals.

level than when starting from mRNA or pDNA transfection. This leads to a less significant rise of the fluorescence signal at the start of the imaging period before it starts decreasing than compared to previous research [28], which could explain why the median maturation value of 0.698 h^{-1} is lower compared to similar measurements done for eGFP of 1.28 h^{-1}. The distribution of the maturation parameter is shown in Figure 4.3A and the degradation in Figure 4.3B. Here, the histogram shows the distribution of the extracted parameters from the fit, median and means respectively. However, statistical tests for maturation and degradation parameters were performed to compare the distributions of the fitted parameter values to log-normal, Gaussian and Weibull distributions, but all tests rejected the null hypothesis, therefore only the median and mean values are discussed further. The distribution of the degradation parameter of mGL shows two populations, which suggests two different degradation pathways with two mean values of 0.013 h^{-1} and 0.022 h^{-1} and respective standard deviations of $1.68 \times 10^{-3}\,\mathrm{h}^{-1}$ and 0.003 h^{-1}.



A: Distribution of extracted maturation rate k_{mat} with a fitted log-normal distribution and median value of 0.698 h^{-1}

B: Distribution of degradation rate β_{Prot} with two normal distributions fitted to values split by threshold (purple) suggests two degradation paths with mean values (yellow) of 0.013 h^{-1} and 0.022 h^{-1}

Figure 4.3: Distribution of extracted rates from stably expressing mGL HEK cells after translation stop through addition of CHX.

4.1.2. Determining the Translation Rate of mGL mRNA in HEK Cells

Based on this, in the next step we want to measure the translation k_{TL} and mRNA degradation δ_{RNA} of mGL-RNA. Hence, we produced mRNA from a mGL template as explained in Appendix A.1.2 and seeded HEK-bsmGL cells onto single-cell slides. After incubation, cells were transfected with this mGL mRNA as detailed in Appendix A.2.3 and imaged over 30 h. Cells are tracked and fluorescence values extracted. The model in *Julia* (Figure 4.4) is expanded with two new parameters, k_{TL} and δ_{RNA} together with t_0, the time where the mGL-mRNA reaches the cytosol and can be translated. The previously attained parameters for k_{mat} and β_{Prot} are used as additional starting parameter. Figure 4.5A shows exemplary expression lines. Residuals can be found in Figure B.1. mRNA transfection starts in the first 5 h after transfection, but also reaches a plateau at most 10 h after fluorescence onset.

As seen in Equation 2.3, the protein level depends on the product of m_0, the starting

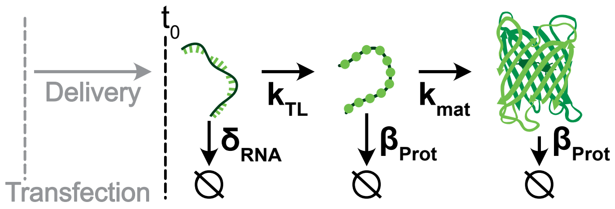
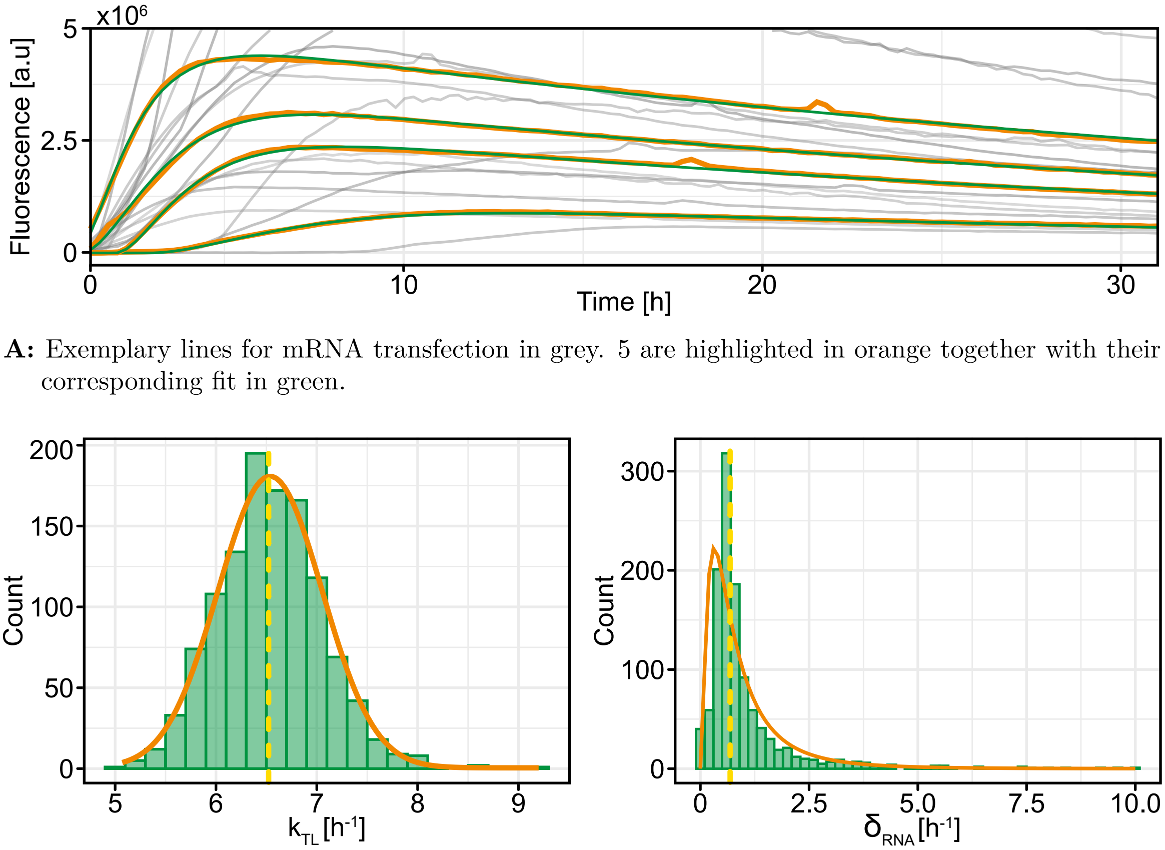


Figure 4.4: Model for gene expression from mRNA transfection. Once the RNA reaches the cytosol at t_0, AA chains are produced with rate k_{TL}. These fold into the fluorescent protein with k_{mat}. Corresponding degradation rates are δ_{RNA} and β_{Prot}.



A: Exemplary lines for mRNA transfection in grey. 5 are highlighted in orange together with their corresponding fit in green.

B: Distribution of extracted translation rate k_{TL} with a fitted normal distribution and mean value of 6.54 h^{-1}. Figures are cut off where no values exist.

C: Distribution of mRNA degradation rate δ_{RNA} with a log-normal distribution fit and a median of 0.68 h^{-1}.

Figure 4.5: Transfection of mGL mRNA into HEK-bsmGL cells. Figure B and C are cut off where no values exist.

concentration of delivered mRNA and k_{TL}. In our model, we set $m_0 = 1$ to simplify the fitting and only estimate k_{TL}. This distribution is shown in Figure 4.5B fitted with a normal distribution and a mean value of 6.54 h^{-1}. Various values for $k_{TL} \cdot m_0$ were measured previously and range from 1 up to 200 h^{-1}, therefore our value aligns with current knowledge [51, 59, 114–116]. Figure 4.5C shows the distribution of the extracted degradation parameter δ_{RNA} for the mGL RNA with a log-normal fit and a median of 0.680 h^{-1}. This is slightly higher than previously reported experiments [28], but we also measure a different fluorescent protein here — mGL, not eGFP or CayRFP — and use a different cell line than reported in those studies.

4.1.3. Estimating mGL pDNA Transcription and Degradation Rate

As a final step, we aimed to measure the transcription and degradation rate of pDNA. Therefore, the same experiments as before were carried out with mGL-pDNA as de-

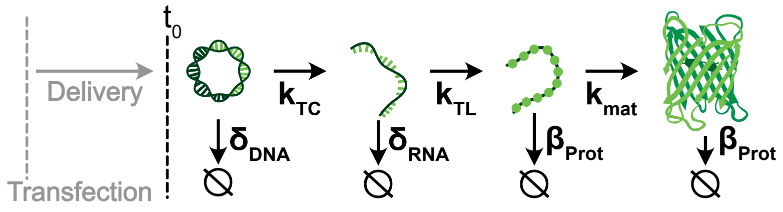
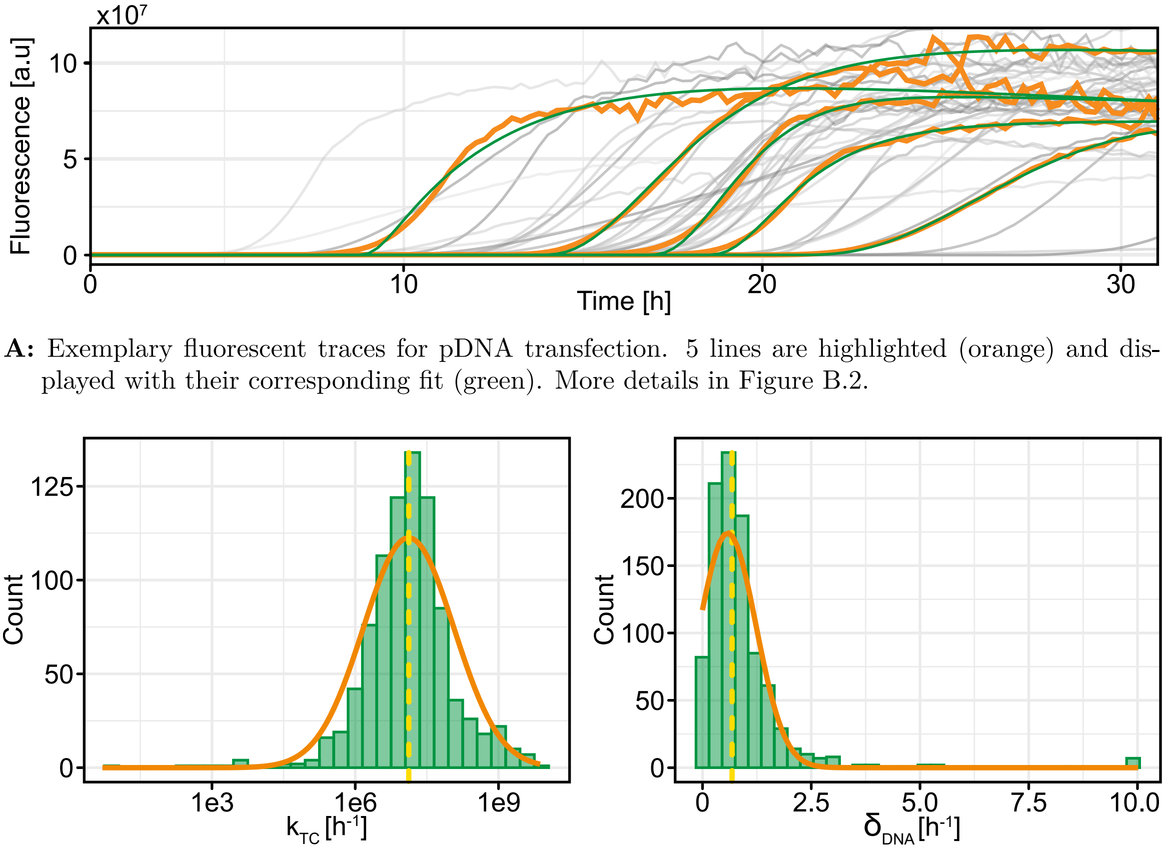


Figure 4.6: Model for gene expression from pDNA transfection. The pDNA must reach the nucleus, where it is transcribed into mRNA at t_0 with a rate constant k_{TC}. Following this, the same steps as in the mRNA model are applied. The corresponding degradation rates, δ_{DNA}, δ_{RNA} and β_{Prot}, are defined as previously.

scribed in Appendix A.2.3, and imaged for 30 h. Once again, the model was expanded to incorporate the parameters of k_{TC} for transcription from DNA to RNA and δ_{DNA} for degradation of the plasmids, as shown in Figure 4.6. In this case, t_0 was set to the time point when the plasmid reached the nucleus, as only there it could be transcribed into mRNA. We used the previously measured parameters but increased their bounds to improve the fitting results.

As seen in Figure 4.7A pDNA transfection onset times are very broad distributed and in the range of 2 to 20 h, also shown by Leonhard et al. [59]. In Figure 4.7A, we can see that the fits align less than the ones for maturation and mRNA translation (Figure B.1). This may be due to the increased number of parameters in the fit, making it more challenging for the fit to converge. Additionally, this could result from starting conditions, which were taken from the previous mRNA experiments or the fact, that the kinetics could be altered due to the cell using different UTRs. Producing the mGL-mRNA through in vitro transcription (IVT), allowed us control over the UTRs, which are influential for the measured kinetics here, as explained in Subsection 2.1.2. When transfecting pDNA, the cell itself controls the UTRs, and therefore, the UTR dependent kinetic parameters can vary. The value of k_{TL} was also estimated again using the previously obtained value from the mRNA transfection experiment as a starting parameter. In this parameter estimation, the median of k_{TL} resulted in a smaller value of $0.753\,\mathrm{h}^{-1}$, smaller than the value from mRNA transfection with $6.52\,\mathrm{h}^{-1}$. Figure 4.7A shows, that the fitted curves start later than the experimental data, but with a steeper slope, suggesting that k_{TC} could have been overestimated with a mean of $1.13 \times 10^8\,\mathrm{h}^{-1}$ (see Figure 4.7B) and therefore leading to an underestimation of k_{TL}. It is important to note here, that in the previous case of mRNA transfection in the ODE system k_{TL} and m_0 are not independent parameters and m_0 was set to 1, but in the case of pDNA transfection m_0 is now combined with the transcription k_{TC} parameter instead of the translation parameter k_{TL} and still set to 1. The fits for pDNA could potentially be improved by using a fitting algorithm that weighs different parts of the fit differently, thereby focusing more on the initial part of the curve and less on the noisy part at the end, leading to a more accurate estimate of k_{TC}. The median degradation of the pDNA from the estimated model can be seen in Figure 4.7C and is $0.680\,\mathrm{h}^{-1}$.



A: Exemplary fluorescent traces for pDNA transfection. 5 lines are highlighted (orange) and displayed with their corresponding fit (green). More details in Figure B.2.

B: Distribution of extracted transcription rate k_{TC}. x-Axis is log-transformed to fit the data with a normal distribution. Mean of this distribution is $1.13 \times 10^8\,\mathrm{h}^{-1}$

C: Distribution of pDNA degradation rate δ_{DNA} with a normal distribution and a median of $0.680\,\mathrm{h}^{-1}$.

Figure 4.7: pDNA transfection of mGL into HEK-bsmGL cells. Figure B and C are cut off where no values exist.

4.1.4. Comparison of Fluorescence Onset Distributions for pDNA and mRNA Delivery

In the next chapter we want to compare the timing of prime editing based on the encoding of its components either via pDNA or mRNA. Unlike RNA, which is translated in the cytosol, DNA has to first enter the nucleus to be transcribed. Thus the delivery time of DNA and RNA differ widely. To compare these distributions, we extracted the parameter t_0 for each fit from the mRNA transfection experiments in Subsection 4.1.2 and the pDNA experiments in Subsection 4.1.3. Figure 4.8 shows that the onset of fluorescence for mGL-mRNA transfection occurs approximately 13 h earlier than with the mGL-pDNA construct, with means of 1.4 h and 14.5 h, respectively. The large standard deviation of 5.7 h for pDNA delivery indicates a much broader distribution of fluorescence onset compared to mRNA delivery, which has a standard deviation of 0.5 h.

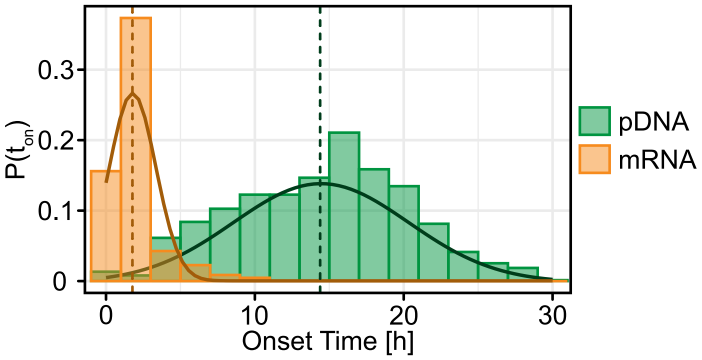


Figure 4.8: Distribution of t_0 extracted from fitting mGL fluorescence traces using pDNA and mRNA transfection. Solid lines depict Gaussian fits with marked mean onset times (1.4 h for mRNA and 14.5 h for pDNA). Figure adapted from [M4].

4.2. Measuring Prime Editing

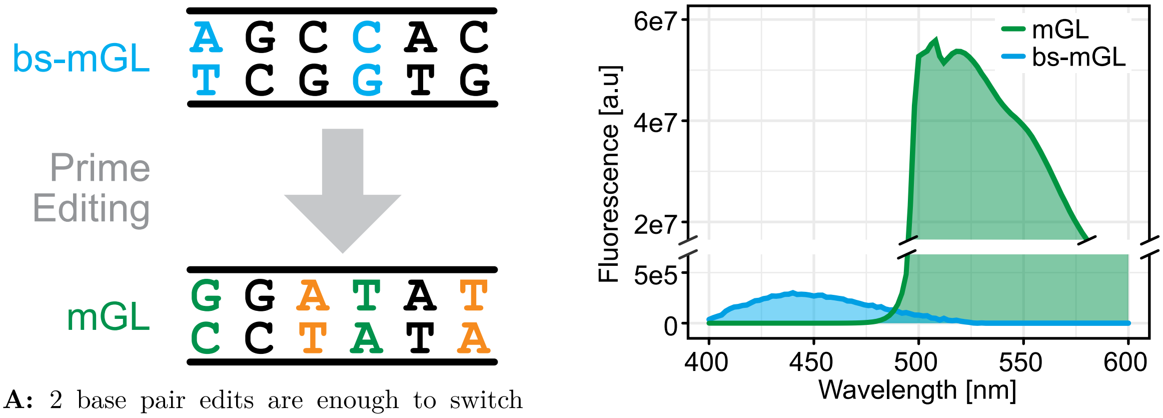
We have now established, that we can efficiently measure fluorescence data for single cells in a high-throughput manner by directly transfecting the blueprints for the fluorescent protein. In the following chapter, we introduce a new system that enables the measurement of prime editing kinetics using fluorescence as a readout. We begin by getting a rough idea of the time frames using bulk measurements and then adapt the previously used assay to better align with the time frames of prime editing and the resulting fluorescence signal. Finally, we developed an algorithm to reliably determine the onset time of these prime-editing-triggered fluorescence signals.

4.2.1. Edit Leads to Measurable Fluorescence Shift

In the following experiments, we continued to use the HEK-bsmGL cell line, because it was engineered to incorporate a variant of mGL in its ATP1A1 locus. This variant of mGL contains two base edits, as shown in Figure 4.9A and Appendix A.1.3, which shifts the fluorescence of the resulting protein “bsmGL” to blue. Our prime editing system targets these base pairs and reverts them, so that the original AA chain sequence can be produced by the cells, resulting in green fluorescence. Two additional base-pair edits were introduced, which have no effect on the resulting AA chain, but increase editing efficiency. To validate the fluorescence emission, we used a spectrofluorometer with an excitation wavelength of 385 nm to excite cell lysate from both HEK-bsmGL and HEK-mGL. Figure 4.9B shows that the fluorescence emission spectrum of bsmGL is successfully shifted to blue, but with a significant loss of intensity compared to the original mGL.

4.2.2. From Bulk to Single-Cell Resolution of Editing Events

After validating the green fluorescence in cell lysates, we proceeded to living cells. To do this, we seeded cells in a 24-well plate and transfected them with the two



A: 2 base pair edits are enough to switch the fluorescence from blue to green. Additional synonymous edits are marked in orange.

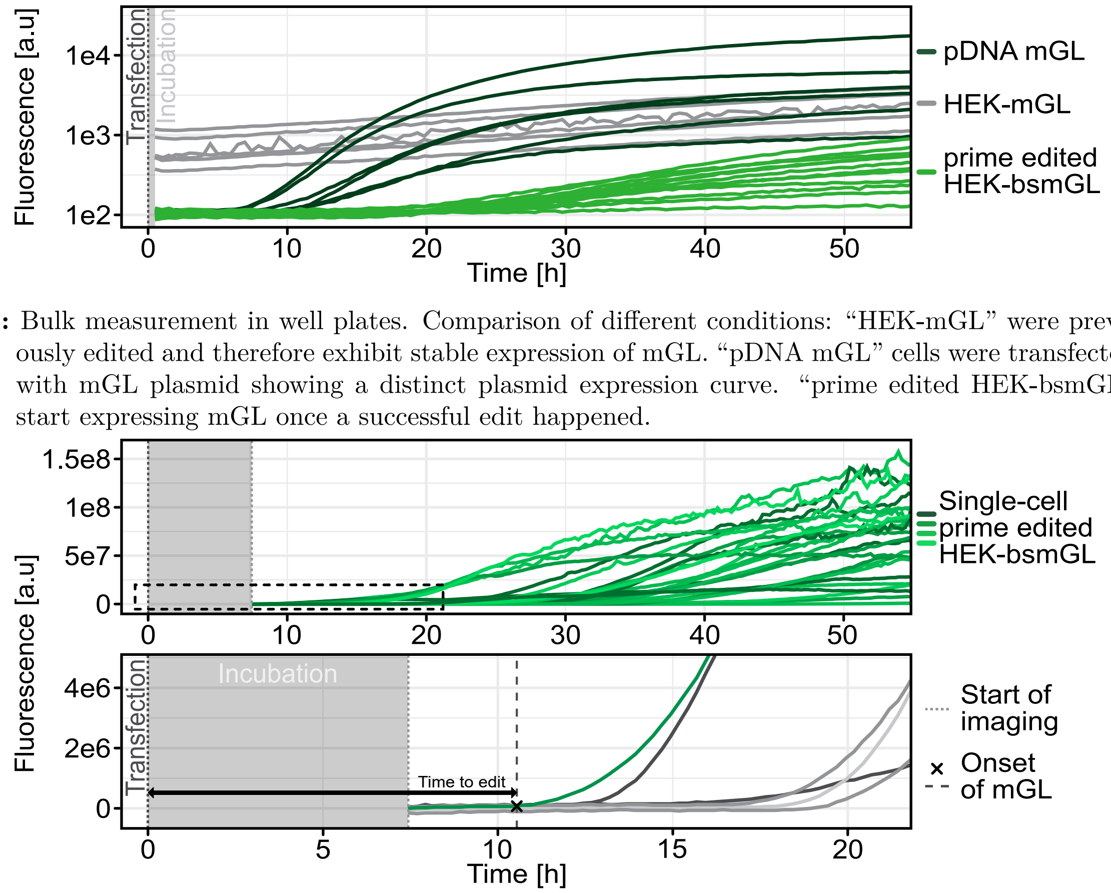
B: Fluorescence emission spectra of bsmGL and mGL HEK cell lysate background corrected by the emission of HEK wild-type cell lysate.

Figure 4.9: HEK cells with a blue-shifted mGL integrated in locus ATP1A1 can be edited to original mGL using prime editing, regaining their ability to emit green fluorescence.

prime editing components: a plasmid encoding for the PE and a plasmid encoding for the pegRNA. We chose to co-transfect two plasmids instead of using a single plasmid for both components, as this approach allows for easier pegRNA replacement in subsequent experiments, maintaining consistency with the current setup.

After a short incubation period, the fluorescence of each well was imaged over 60 h using a heated plate reader (Tecan). Figure 4.10A shows the short incubation period after transfection, followed by fluorescence traces collected from each well, with each representing a large cell population. HEK-mGL cells, which were previously edited, show only a slight increase in fluorescence, attributable to population growth. In contrast, the fluorescence of HEK-bsmGL cells transfected with pDNA exhibited a typical sigmoidal curve, with an onset of around 10 h and a plateau after another 10 to 20 h. The bulk fluorescence signal of HEK-bsmGL cells transfected with the prime editing components showed a later onset of 20 to 30 h and a more gradual increase.

These results indicate that the prime editing step takes significantly more time compared to direct transfection with mGL pDNA. In previous single-cell transfection experiments, cells were first seeded onto the pattern and then transfected within the channels. This approach was used to capture the early moments of fluorescence. However, this method did not work for prime editing. The increased stress associated with the editing event led to a significant loss of cells over time. Additionally, cells proliferated, eventually crowding the slide before a fluorescence onset could be detected. Furthermore, the experimental time window is limited by the availability of nutrients in the medium and the slide's reservoir to keep conditions stable. To address these challenges, we first seeded the cells in 24-well plates and transfected them there. To further reduce the stress on the cells, the media was exchanged after 4 h of incubation. Following another incubation period, the cells were removed from the well plate and seeded onto the slides, as detailed in Appendix A.2.5. The slide was then transferred to a microscope and imaged over 30 to 48 h (see Appendix A.2.6). This



A: Bulk measurement in well plates. Comparison of different conditions: "HEK-mGL" were previously edited and therefore exhibit stable expression of mGL. "pDNA mGL" cells were transfected with mGL plasmid showing a distinct plasmid expression curve. "prime edited HEK-bsmGL" start expressing mGL once a successful edit happened.

B: Top: Fluorescence traces of single HEK cells after transfection with the prime editing system. **Bottom:** Zoom in of the period from 0 h to 20 h. Imaging starts after a required incubation period. The time-to-edit is defined as the time from transfection until the onset of mGL as shown here.

Figure 4.10: Fluorescence data after transfection of HEK cells. Different experimental setups need different incubation periods, here shown as grey boxes.

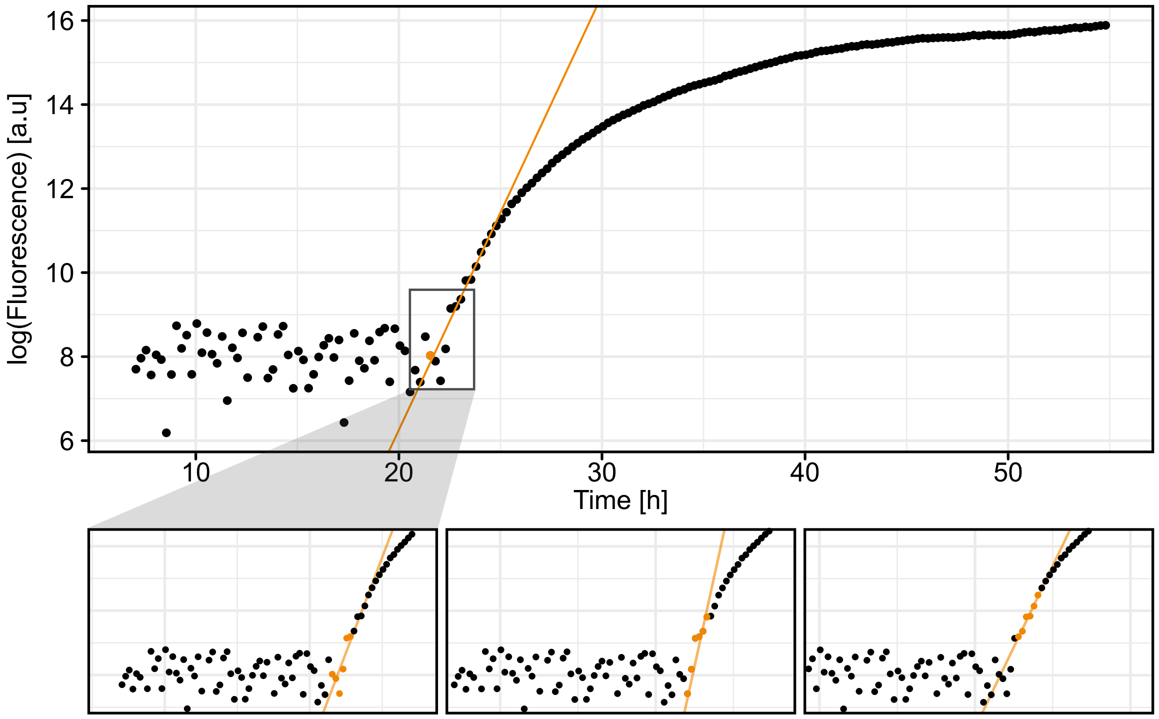
protocol reduces cellular stress, increases the likelihood of isolating single cells on the pattern without excessive growth, but also results in a longer not measurable period. Figure 4.10B shows exemplary single-cell fluorescence traces, including the incubation time for the adapted assay. From these traces, we can now define the **time-to-edit** as the time from transfection in the well plate to the onset of fluorescence signal of a single cell.

4.2.3. Reliably Determining Onset Time for Prime Editing

Fitting the data with a standard exponential curve or one of the previously established expression models turned out to be inadequate in this context because it failed to accurately capture the initial time point (t_0), leading to an incorrect shift to later time points in the onset distribution for prime editing. Additionally, the noise present at the start of the experiment, when no fluorescence signal is present complicated the fitting. This issue is particularly evident when the data is plotted on a logarithmic scale as can be seen in Figure 4.11, where the onset time point becomes more pronounced. Therefore, an alternative fitting approach was necessary to account for these factors and provide a more accurate representation of the data. We employed an algorithm written in the programming language *R* (see Appendix A.4.2), which uses a sliding window approach with three main parameters and is executed for each trace independently. The algorithm takes six consecutive values of a single-cell fluorescence trace, fitting a linear model to them and evaluating the slope of the fit against a predefined slope-threshold. Should this threshold be exceeded for six consecutive windows, the time of the first data point from the first window exceeding the threshold is recorded as t_0, otherwise the threshold is adjusted downwards, and the algorithm re-commences with the first window, as detailed below:

```
Slope-threshold = 1.0
Window-size = 6
Window-counter = 0

Loop through the data points, starting at the first:
    Window = Choose <window-size> continuous values starting at the
             current data point

    Fit a linear curve through window values

    If the slope of the fit is bigger than the Slope-threshold
        Increase Window-counter by 1
    Else
        Reset Window-counter to 0

    If Window-counter > 6
        Record the first time point of the first window
            (Window-counter = 1) as t_0
        export a plot of all data points, the first fit and t_0
        End the program for this cell
    Else
        Go to next loop iteration

If no t_0 was found and the Slope-threshold is greater than 0.1
    Decrease Slope-threshold by 0.1
    Retry with lower Slope-threshold
```

Afterwards fluorescence traces are manually checked and wrongfully detected t_0 are removed following the exclusion criteria described in Figure A.2. Figure 4.11 shows at the top the image exported by the algorithm, t_0 and the respective fit through the next 6 values. The bottom plots show intermediate iterations of the sliding window, the corresponding slope and the increasing counter.



Window	58	60	63
Slope	1.18	1.77	1.01
Count	1	3	6

Figure 4.11: Algorithm used for determining the onset time of prime edited fluorescence. **Top:** Complete fluorescence trace of a single exemplary cell. The determined t_0 is marked in orange with the corresponding slope. **Bottom:** Window 58 to 63 have a slope bigger than the threshold (0.9). After 6 consecutive windows like this t_0 is determined by the first point of the first window 58.

4.3. Discussion

In this chapter, we aimed to establish a system for tracking prime editing events at the single-cell level. To better characterize mGL, we first extracted kinetic rates for maturation and degradation of the protein, then mRNA and pDNA transfection. We observed that mGL degradation appears to have two distinct pathways, as indicated by the double distribution seen in Figure 4.3B, which may be the result of using stably expressing HEK-mGL cells rather than mRNA transfection as in previous studies [28]. This difference is noteworthy and warrants further investigation and could lead to different values of k_{mat}, as the modest fluorescence increase observed in the stably expressing cells in Figure 4.2 could have impacted our results.

The transition to *Julia* and the packages *Catalyst.jl* and *PEtab.jl* for modeling and parameter estimations has facilitated the testing of various model variations and provided greater access to established algorithms for optimization and parameter estimation. This enabled us to fit the pDNA system, though further optimization is required, and additional constraints, such as the $\beta_{Prot} < \delta_{RNA}$ used in Subsection 4.1.2, are likely necessary. Overall, these fits were instrumental in determining t_0 for mRNA and pDNA transfections. Instead of relying solely on data from previous studies, we repeated the measurements with mRNA, pDNA, and mGL, as we used different transfection reagents optimized for the prime editing system and a different fluorescent protein. Additionally, we included pDNA transfections, which have not been extensively studied in this context before.

As mentioned before, our primary interest lies in the kinetics of prime editing. To this end, we developed a system to image prime editing events at the single-cell level by targeting a blue-shifted fluorescent protein sequence in the cell genome and reverting it to its green variant. Bulk measurements indicated that green fluorescence in prime-edited cells takes significantly longer to express compared to cells transfected directly with pDNA-mGL. At the single-cell level, the previous fitting method failed to accurately identify the t_0 time point, leading us to develop a sliding window algorithm to determine it. This allowed us to then define the **time-to-edit**.

In prime editing research, the primary method for assessing successful editing is next-generation sequencing (NGS) [117], which allows for the quantification of editing frequencies, precise identification of edits and off-target effects. However, NGS cannot be performed on live single cells and is limited to a single time point, making it unsuitable for kinetic studies like ours. Previous studies have used fluorescent proteins with a fluorescence shift caused by single base edits [118, 119] or modifications of start codons [120], for validation of base editing systems, but these measurements were typically made at one specific time point and followed by NGS. Our approach, involving single-cell tracking, has not been used with prime editing before and offers the unique capability to precisely determine the onset of mGL fluorescence at the single-cell level.

5. Defining Prime Editing Time Frames

The introduction of prime editing added a new level of precision to CRISPR-based gene editing, enabling efficient "search and replace" of specific DNA sequences with minimal off-target effects [18, 65]. This technique, which uses a Cas9 nickase fused with a RT and a specialized guide RNA — the pegRNA, has expanded the potential for gene therapy, leading to the first FDA-approved treatments and several ongoing clinical trials [24, 121, 122]. The pegRNA guides the edit by providing a RT template and a primer binding site for the targeted DNA.

Prime editing is a complex cascaded process that includes the co-delivery of PE and pegRNA vectors, protein expression, complex formation, and finally the editing process itself, as shown in Figure 5.1. Although numerous studies have examined the efficiency of prime editing, it is typically evaluated from 24 to 72 h after transfection at a single time point [18], so the overall kinetics and their own specific time frames are still poorly studied.

Therefore, in this study, we analyze these editing durations and divide them into different time frames. The aspect of pDNA and mRNA-based delivery was already addressed in Subsection 4.1.4. In this chapter, we first compare the efficiency and timing when delivering the prime editing components via pDNA and mRNA. We then introduce a new PE construct that co-expresses a red fluorescent protein along with the PE, enabling direct measurement of the editing time for both pDNA and mRNA-based delivery. This data is then combined with models to explain the experimental results for mRNA transfection. Finally, we examine the editing process itself and its dependence on the length of the edited window by using different pegRNAs in our pDNA-based system. The majority of the results presented in this chapter are content of [M4].

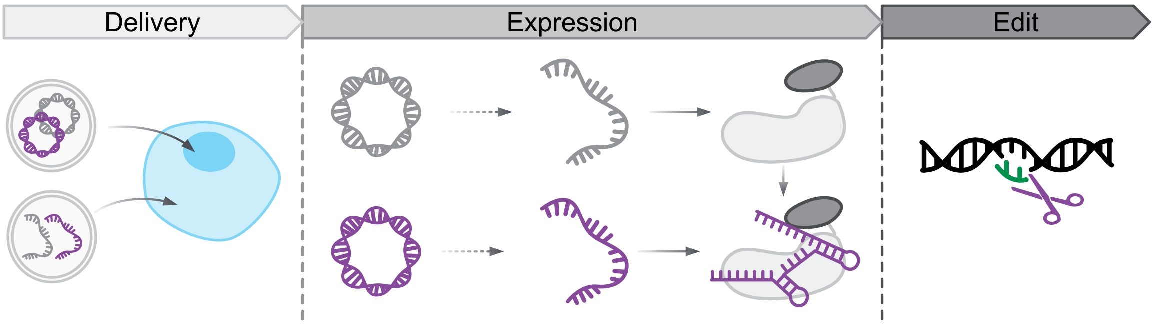


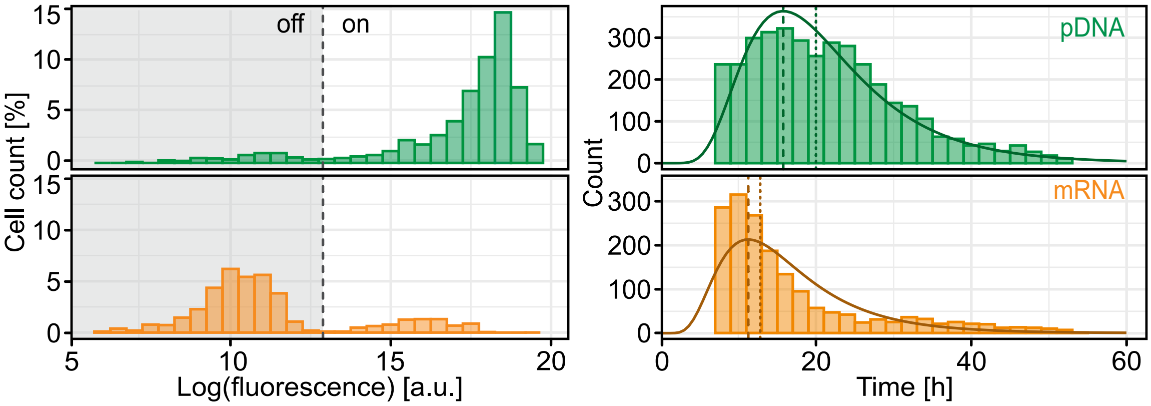
Figure 5.1: Overview of prime editing time frames. After delivery of the components for prime editing, they are expressed inside the cell. Once the prime editing complex is build and transported into the nucleus, the editing step itself can take place.

5.1. Measuring Delivery Dependant Time-to-Edit

To compare pDNA- and mRNA-based delivery for prime editing, we used our developed protocol as explained in Subsection 4.2.2. For pDNA, the prime editor and pegRNA were encoded on different plasmids, pre-mixed before addition of the transfection reagents, and then added to the cells. mRNA was produced in advance as detailed in Appendix A.1.2 and both parts, PE mRNA and pegRNA, were also pre-mixed, similarly to the pDNA solution, before the addition of the transfection reagent. For comparison purposes, pDNA and mRNA transfection experiments were always performed together. The extracted fluorescence traces were sorted by hand following the exclusion criteria described in Appendix A.3.

We compared the efficiency of our transfection using the last recorded fluorescence value for all tracked cells. Figure 5.2A shows this for pDNA and mRNA. Two distinct populations are evident: "on" for successfully transfected cells at higher fluorescence values and "off" for unedited cells. Accumulative prime editing was more efficient using pDNA delivery with 79 % fluorescing cells, compared to 15 % fluorescing cells for mRNA delivery. Cells transfected with pDNA also showed an overall higher mean logarithmic fluorescence value of 17.7 compared to 15.8 for mRNA. This could be attributed to the more stressful transfection process for mRNA. After three passages, cells were indistinguishable by fluorescence intensity.

We previously established that the onset of mRNA-based delivery occurs within 2 h, while the onset of pDNA-based delivery is more widely distributed, as seen in Subsection 4.1.4. We expected a similar time difference for prime editing because, once the complex is assembled, there should be no difference in editing time between both



A: Fluorescence intensity distribution of 10.000 cells 60 h after transfection showing two clear populations of "on" and "off". pDNA delivery has a higher efficiency with 79 % compared to mRNA with 15 % fluorescing cells.

B: Time-to-edit distribution. The dashed lines indicate the peak of a log-normal fit, and the dotted lines the median. The peak of the time-to-edit fits are 15.8 h and 11.3 h, respectively. mRNA shows an 7 h earlier median onset of fluorescence.

Figure 5.2: Prime editing efficiency and kinetics of pDNA- vs. mRNA-based delivery. Figure adapted from [M4].

delivery methods. To compare the time-to-edit between pDNA- and mRNA-based delivery, we extracted the time-to-edit as detailed in Subsection 4.2.3 and fitted it with a log-normal distribution, as shown in Figure 5.2B. For pDNA, this resulted in a median onset of 20 h, and for mRNA-based delivery, 12.9 h. The peak of the distribution shows a difference of only 4.5 h and the median difference is 7 h. Although mRNA-based delivery was earlier than pDNA, the difference was less than the previously measured median onset difference for direct mGL transfection for pDNA and mRNA delivery, which was 13.1 h. We also found that the overall efficacy of the editing, as shown by the area under the distribution curve, is lower for mRNA, consistent with the lower efficiency shown in Figure 5.2A.

5.2. Comparing Delivery Dependant Editing Time

We first quantified the delivery timings of our transfection reagents by directly transfecting fluorescent protein pDNA and mRNA to compare it to literature [59], confirming the previously reported difference of around 13 h. We then measured the time-to-edit for these two delivery methods using our prime editing system and observed that the difference of the median was smaller than expected, at only 7 h. As explained previously in Figure 5.1 the time-to-edit consists of delivery, expression, and editing. We have already measured the delivery time of our transfection agents in Subsection 4.1.4, and now we proceed to quantify the editing time. To this end, we implemented another PE construct, first published by Anzalone et al. [18] the plasmid pCMV-PE2-P2A-GFP (Addgene #132776), but replaced the GFP sequence with mScarlet3 (Addgene #189753). The P2A sequence between PE2 and mScarlet is a short self-cleaving peptide that allows co-expression of multiple proteins from a single ORF [123, 124]. By enabling "ribosome skipping" during translation, multiple

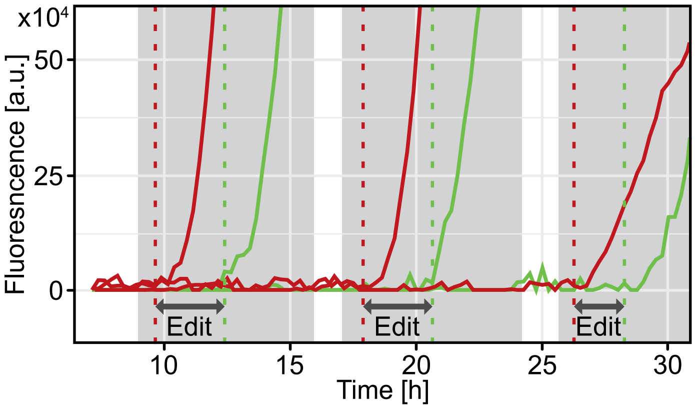


Figure 5.3: Exemplary fluorescence traces for three cells transfected with PE2-P2A-mScarlet. Cells co-express mScarlet (red) and PE leading to an additional red fluorescence signal, which can be tracked simultaneously to mGL (green). Subtracting the mScarlet onset time from the mGL onset time allows one to determine the time it took from expressed PE to a successful edit. Figure adapted from [M4].

AA chains can be translated without being attached to each other, leading to the co-expression of the PE protein and the mScarlet fluorescent protein. This approach allows us to determine the time point at which PE is expressed by measuring the fluorescence of mScarlet. We produced the plasmid pCMV-PE2-P2A-mScarlet and the mRNA PE2-P2A-mScarlet as detailed in Appendix A.1.1 and Appendix A.1.2. To compare, if the co-expression of mScarlet influenced the time-to-edit, we performed control experiments with the original PE2 and the co-expressing PE2-P2A-mScarlet plasmids and mRNAs, finding no notable differences in the mGL onset distributions (see Figure B.3).

With these new constructs, the editing time can be defined as the time from the onset of PE expression $t_{0mScarlet}$ to the onset of mGL expression t_{0mGL}. Figure 5.3 shows three exemplary cells that first express mScarlet and therefore PE, followed by mGL expression. The onset times of mScarlet and mGL were determined using the same algorithm as before, and the editing time was calculated for each cell independently:

$$t_{editing} = t_{0mGL} - t_{0mScarlet}$$

We repeated the experiments as explained in Section 5.1 using the PE2-P2A-mScarlet constructs, but included a second fluorescence channel to image the red fluorescence. Cells with a negative editing time were removed from the data after confirming that

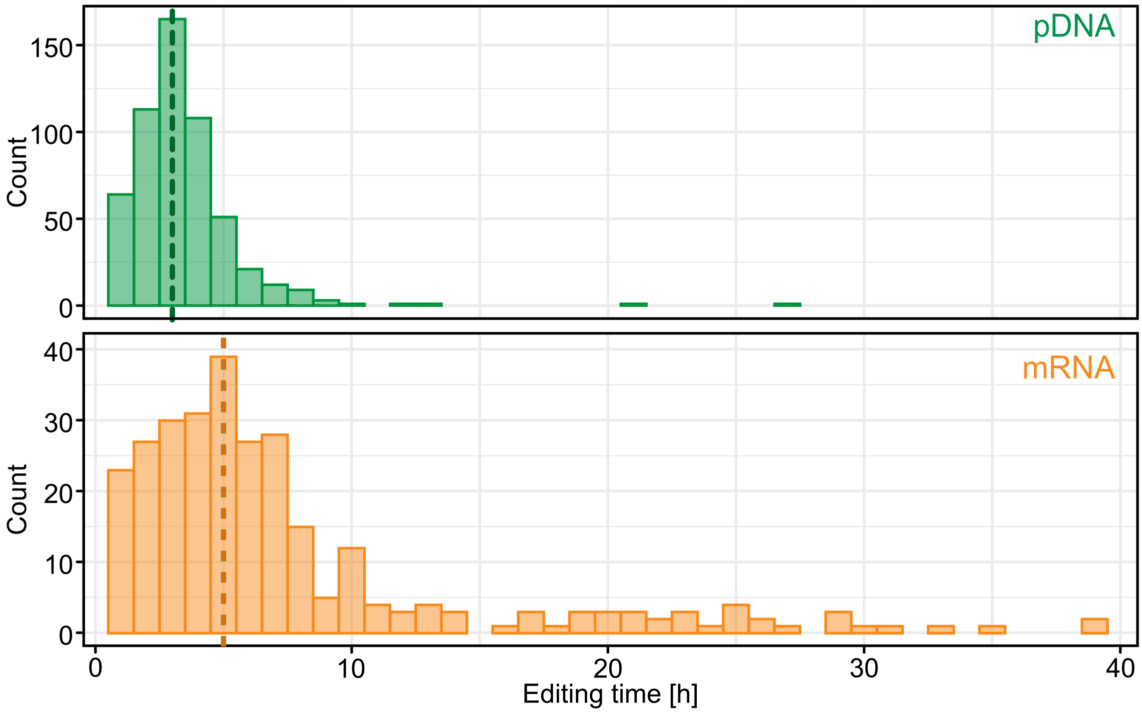


Figure 5.4: Distribution of editing times for pDNA and mRNA transfection using the PE2-P2A-mScarlet sequence and the same pegRNA design. Dashed lines denote the median editing time, showing that mRNA editing is slower with a median of 5 h compared to pDNA with 3 h. Figure adapted from [M4].

our algorithm had not identified the correct onset time by checking the images extracted from the algorithm as explained in Subsection 4.2.3. This experimental setup and evaluation resulted in a median editing time of 3 h for pDNA, which increased to 5 h for mRNA, as shown in Figure 5.4. This result was unexpected and will be discussed in further detail in the next section.

5.3. Missing Delay in mRNA Time-to-Edit

With our findings on the delivery and editing time frames for pDNA- and mRNA-based delivery, we now aim to reproduce our measured time-to-edit. We hypothesize that the time-to-edit, defined as the period from delivery to the expression of the edited fluorescent protein, can be mathematically represented as a convolution of the two previously measured distributions, as shown in Equation 5.1.

$$P_{Edit}(t) = C_{Delivery}(t) \otimes K_{Editing}(t) \tag{5.1}$$

The delivery time, measured in Subsection 4.1.4, represents the period from transfection to the beginning of protein expression in cells. The editing time begins when PE is expressed for the first time and continues until editing occurs and mGL is expressed, as described in Section 5.2. The convolution of these two distributions should result in the observed time-to-edit and is shown in Figure 5.5. The experimental values are the same as in Figure 5.2B, but with a higher bin resolution of 1 h. The solid line represents the convolution P_{Edit} of the histogram data of Figure 4.8 ($C_{Delivery}$) and Figure 5.4 ($K_{Editing}$). This convolution was calculated using the *conv()* function from the *Julia* package *DSP.jl*. The convolution aligns well with the time-to-edit

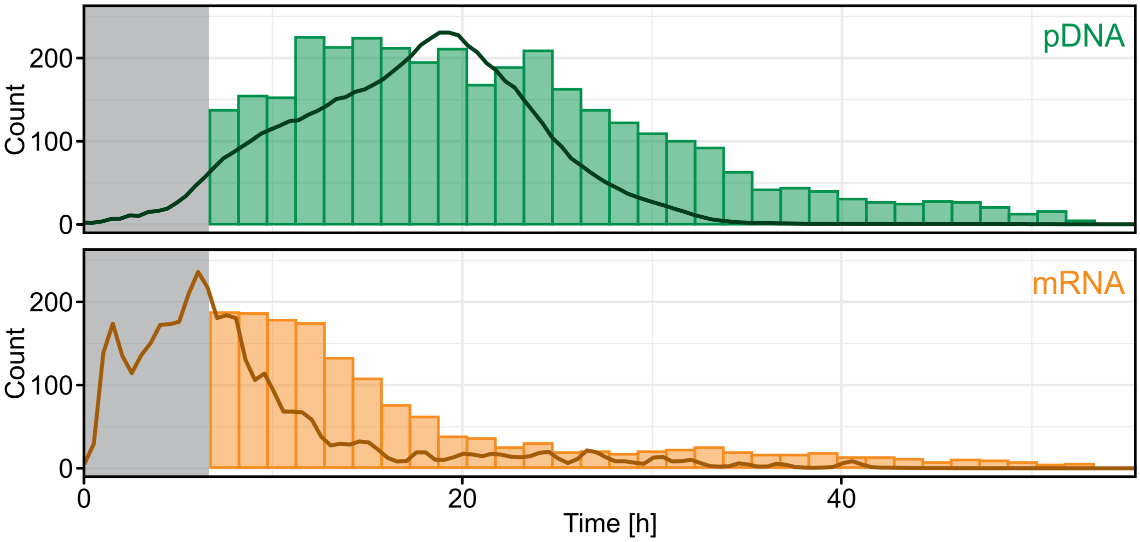


Figure 5.5: Experimental time-to-edit distributions for pDNA and mRNA delivery. Solid line denotes theoretical model. Grey areas show incubation period outside of the microscope. Figure adapted from [M4].

distribution for pDNA delivery, but not as closely with the mRNA-delivery distribution. To simulate possible differences, we built a expression model similar to the one used previously for mGL expression. More details about the model will be explained in Chapter 6. Values from earlier experiments and literature were used for the model parameters (see Table B.2), and we compared the expression model starting the expression from either the pDNA or mRNA stages. In both models, t_0 is set to the specific time point when either pDNA or mRNA have reached their processing location, so for pDNA the nucleus and mRNA the cytosol.

In Figure 5.6, we can see that after pDNA enters the nucleus and transcription begins, both PE and pegRNA are produced extremely quickly and in such large quantities that complex formation occurs rapidly, primarily due to sheer numbers. This is especially evident in the top frames of Figure 5.6, where the scale of the y-axis shows that the PE complex from pDNA reaches up to 1×10^6 counts, while the count starting from mRNA peaks at 0.5×10^5 counts. When zoomed in (bottom of Figure 5.6), we observe that it takes 2 h to reach the same concentration of complexes after mRNA transfection, while for pDNA it takes only 8 min, after which the concen-

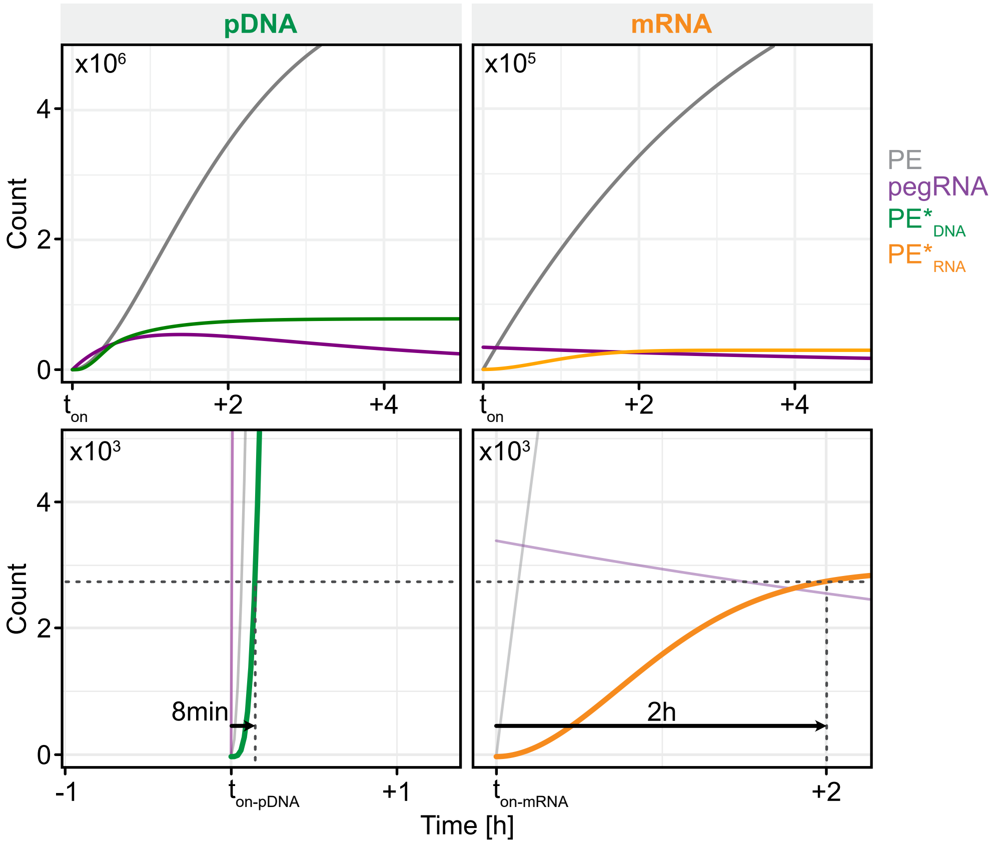


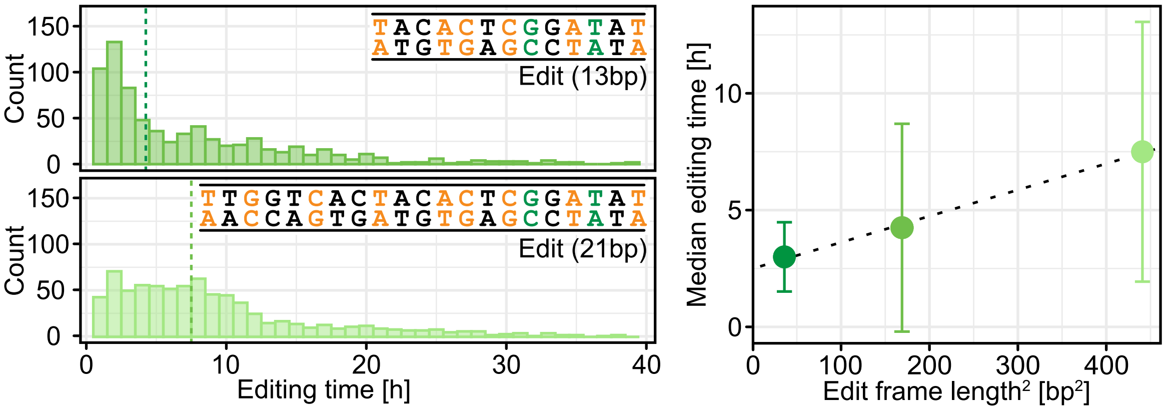
Figure 5.6: Model of PE complex concentration over time starting from mRNA(left) or pDNA(right). The scale of the y-axis is shown in the top left corner. **Bottom:** Zoom into concentrations up to count = 5×10^3 shows, that mRNA takes 15 times longer to reach the same concentration of assembled PE complex than pDNA. Figure adapted from [M4].

tration of complexes continues to increase rapidly. This measurable delay, either due to complexation or slower target loci finding caused by low concentrations of assembled complex, following mRNA transfection will be further discussed in Chapter 6.

From this model, we can also extract the following: one of the limiting factors for the maximal concentration of complex after mRNA transfection is the amount of pegRNA transfected as seen in Figure 5.6, bottom right. We hypothesize that the starting ratio of PE mRNA to pegRNA may influence this and will be discussed in the following chapter.

5.4. Editing Time, Branch Migration and their Consequences on Edit Length

In the previous section, we observed how quickly the prime editing complex forms when starting from pDNA transfection. This setup, involving a short and highly efficient pegRNA using pDNA delivery, resulted in an editing time of around 3 h. This editing time likely represents a combined duration for the transport of the complex into the nucleus, the identification and binding of the target sequence, and the editing process itself, including reverse transcription, flap exchange, and repair as explained in subsubsection 2.1.5. When using the same PE and comparable pegRNA but with a longer editing frame, the transport and target identification times should be similar, so we assume that the time this takes remains constant. However, the flap exchange process is not the same when modeled as a strand displacement process, as explained in subsubsection 2.1.5. This suggests that the editing time depends on the length of the edit.



A: Editing time distribution of two pegRNAs with increased edit frame. The median editing time is marked by the dashed lines (13 bp=4.25 h and 21 bp = 7.5 h).

B: Median editing time versus edit frame length squared shows a positive correlation for pDNA delivery. The dashed line denotes a linear fit. Error bars show the median absolute deviation.

Figure 5.7: pegRNA variations show a correlation between median editing time and edit frame length. Figure adapted from [M4].

To test this hypothesis, we designed two additional pegRNAs. We maintained the original editing sequence to target the bsmGL-to-mGL edit but extended it with synonymous edits in the direction of the previous PBS. The new PBS therefore shifted to a complementary sequence further upstream of the 5' genome sequence, but retained the same length as the original pegRNA's PBS, leading to a longer RT template and a longer editing frame. Since we used synonymous edits, the produced AA chain and therefore the fluorescent protein remain the same.

We repeated our single-cell prime editing assay by transfecting the pCMV-PE2-P2A-mScarlet plasmid and each of the three different pegRNA plasmids to extract their respective editing time distributions. Figure 5.7A shows the editing time distributions for medium and long pegRNAs. The editing frame is depicted in the small insets of both plots. Comparing these data with Figure 5.4, we observe that the median editing time for an editing frame of 6 bp is 3.0 h, increasing to 4.25 h for 13 bp, and then 7.5 h for 21 bp.

Following the strand displacement theory described in subsubsection 2.1.5, the time should scale with the square of the length of the edit. Therefore, we plotted the editing time against the edit frame length squared and fitted the results, as shown in Figure 5.7B. The data and fit align well, supporting our hypothesis. We also note that the median absolute deviation increases with longer editing length.

5.5. Discussion

This chapter focused on the differences in time-to-edit between pDNA- and mRNA-based delivery methods. Using our previously established LISCA approach, we observed that the median time-to-edit for mRNA-based delivery was 7 h earlier than for pDNA. While this result was expected, the mRNA editing process was slower than anticipated. To further investigate this, we designed a co-expression plasmid that enabled the imaging of PE expression using an mScarlet protein. With this, we measured the editing time for both pDNA and mRNA delivery and demonstrated that the median mRNA-based editing was 2 h slower than pDNA-based editing.

We then introduced a mathematical model that represents the time-to-edit as a convolution of the delivery and editing times. Although the model accurately predicted the time-to-edit for pDNA delivery, it revealed a discrepancy for mRNA-based delivery. To address this, we set up a prime editing expression model that highlighted a difference in complexation and concentration dynamics, showing that the mRNA-based prime editing system is approximately 15 times slower in reaching the same complex concentration as the pDNA-based system.

Given the simplicity of our model — lacking, for example, compartmentalization, relying on experimental ratios, and a limited set of known kinetic values for prime editing from the literature (see Table B.2) —– further studies are required to specifically extract kinetic parameters for prime editing. Different model setups, such as compartmentalization could improve the precision of our model and will be discussed in more detail in the next chapter. However, the kinetics of mRNA-based prime editing could potentially already be improved by employing mixed or sequential transfections of mRNA and pDNA, as suggested by Nasr et al. [125]. Hybrid delivery strategies could provide an effective way to improve both the speed and efficiency of prime editing.

The timing of prime editing is particularly relevant for researchers targeting specific tissues or regions of the body. Especially since studies have shown that CRISPR systems tend to accumulate in the liver and therefore could lead to increased off-target effects [26, 126]. Alternative delivery agents, such as LNPs, are also intensively researched. LNPs can carry larger cargo and may offer distinct delivery kinetics, such as pH-dependent or sequential release for mRNAs of different sizes, while also providing the potential for tissue-specific targeting and reduced off-target effects [127, 128]. In contrast, certain gene therapies can require sustained, long-term recurring edits to maintain therapeutic efficacy [129], further underscoring the importance of knowledge about kinetics when designing CRISPR-based systems.

In our work, we explored the kinetics of pDNA transfection and demonstrated a correlation between editing length and editing time. Based on strand displacement theory, one could argue that the editing time also depends on the specific sequence of the edit [130]. Additional experiments targeting different loci, with varying lengths or synonymous edits, could provide further insight into the accuracy of strand displacement as a mechanism for determining editing time.

This consideration is particularly important when designing prime editing systems, raising the question of whether to perform multiple edits at once or sequentially. Choi et al. [19] introduced the concept of a “DNA typewriter”, where sequential edits are used to track cellular lineage, with an estimated prime editing rate on the order of days, aligning with our observations. They suggested that their typewriter — using prime editing — is more suitable to track processes that unfold over several days or weeks. In comparison, our pDNA system showed a median total editing time of around 20 h, which increased to 30 h for edits of 21 bp. In such cases, a single pegRNA would be advantageous over sequential edits for longer editing frames. However, we did not perform NGS, and therefore cannot confirm whether the full sequence was incorporated. It is possible that two smaller edits might result in higher efficiency than a single larger one.

For the medium-length pegRNA, we observed an accumulation of editing times similar to that of the shortest pegRNA, along with a higher onset count in the earlier hours. This may indicate that the full edit was not incorporated, but only the first few base pairs including the bsmGL-to-mGL shift. Since we did not perform NGS to assess the extent of the editing, our system classified an edit as successful when the blue-to-green edits were present, without confirming the incorporation of the entire sequence. Future research could explore the use of NGS or a system where the blue-green sequence is located at the end of the editing frame to better assess the length and completeness of the incorporated edits.

6. Modeling Beyond the Experimental Results

Numerous computational models and simulations for prime editing have been developed, offering various tools for designing pegRNAs [131, 132], as well as optimizing PBS and RT lengths [133], and predicting editing efficiencies [134]. However, models describing the kinetics of prime editing have not yet been introduced. Previous research has demonstrated that gene expression kinetics for fluorescent proteins can be modeled using kinetic rate equations and ODEs [29, 61, 62]. In Chapter 4, we showed that these equations can be applied to fit experimental fluorescence traces of single cells using *Julia*, primarily utilizing the packages *Catalyst.jl* and *PEtab.jl*.

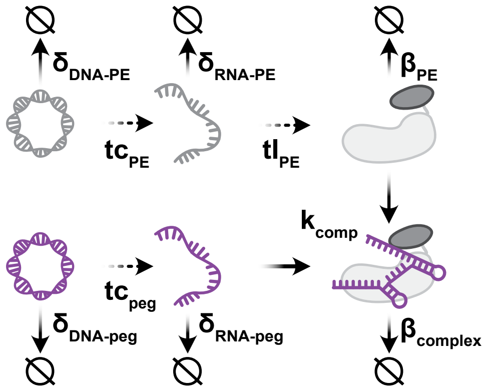
As previously mentioned, *Catalyst* facilitates easy model setup and modification of parameters. In this chapter, we leverage this "ease of use" to model the kinetics of prime editing expression and explore insights gained from our *in silico* approach for our *in vitro* experiments [135]. We begin with the simplest model introduced earlier (see Section 5.3), and by varying different parameters, we aim to guide our experimental design.

Next, we examine the difference between a catalytic system and a first- or second-order reaction system. This distinction relates to whether the resources are consumed in the different steps of the prime editing process or merely act as catalysts for the production of subsequent reactants as explained in Subsection 2.1.4.

Finally, we extend the simple model by introducing a barrier, representing the nuclear membrane, and incorporating a diffusion/transport parameter. We then perform parameter variations and again compare the kinetics related to mRNA- and pDNA-delivery.

6.1. Leveraging Prime Editing Expression Models to Guide Experimental Design

We designed the simplest model of prime editing expression, as shown in Figure 6.1A. The two components of prime editing are encoded on two separate plasmids in our experiments; therefore, we implemented two expression cascades for eachseparately. PE is transcribed from pDNA to RNA with a rate tc_{PE}, then translated into protein with tl_{PE}. The pegRNA is only transcribed with a rate tc_{peg}. Both components differ significantly in their DNA as well as RNA sequence length: PE-DNA is approximately 6000 bp long, while pegRNA-DNA has 200 bp. As a result, we used different transcription rates for each. The pegRNA and PE protein then assemble to form the editing complex with a rate k_{comp}. All reactants degrade at their respective



A: Graphical representation

```
PE_DNA = @reaction_network begin
  tc_PE,   PE_DNA --> PE_DNA + PE_RNA
  tl_PE,   PE_RNA --> PE_RNA + PE_prot
  tc_peg,  peg_DNA --> peg_DNA + peg_RNA
  k_comp,  PE_prot + peg_RNA --> PE_complex

  d_DNA_PE,  PE_DNA --> 0
  d_RNA_PE,  PE_RNA --> 0
  b_PE,      PE_prot --> 0
  d_DNA_peg, peg_DNA --> 0
  d_RNA_peg, peg_RNA --> 0
  b_complex, PE_complex --> 0
end
```

B: Julia Catalyst implementation

Figure 6.1: Basic prime editing expression system starting from pDNA with separate rates for transcription, translation and degradation of PE and pegRNA. The complex is build with PE and pegRNA. Dashed arrows represent production, which does not consume the initial reactants. Rates as explained in Table B.2.

rates δ and β. Processes shown with dashed arrows in Figure 6.1A do not consume the initial reactants, meaning RNA is produced from pDNA without reducing the pDNA concentration. Solid arrows depict processes where the reactants are consumed to form new reactants. Figure 6.1B shows how this is implemented in *Julia*. All reactants are set to an initial concentration of 0, except for the starting concentrations of the plasmids $PE\_DNA(t=0)$ and $peg\_DNA(t=0)$. All parameters used for this model are shown in Table B.2. This implementation leads to the following ODE system:

$$\frac{dPE_{DNA}(t)}{dt} = -\delta_{DNA-PE}PE_{DNA}(t) \tag{6.1}$$

$$\frac{dPE_{RNA}(t)}{dt} = -\delta_{RNA-PE}PE_{RNA}(t) + tc_{PE}PE_{DNA}(t) \tag{6.2}$$

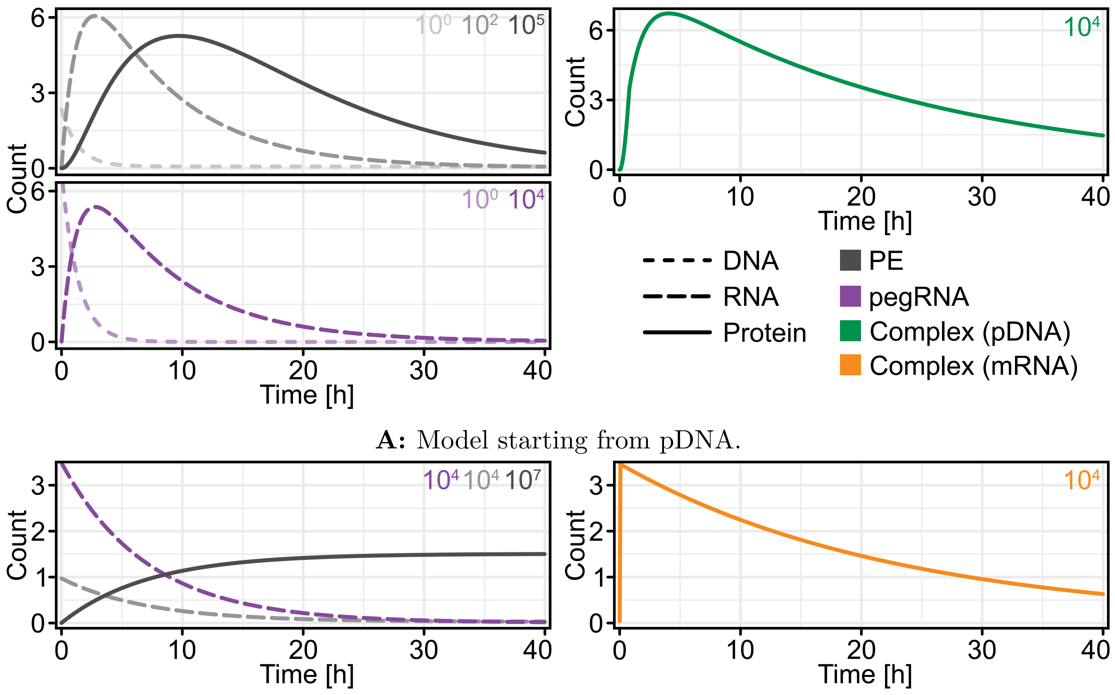
$$\frac{dPE_{prot}(t)}{dt} = -\beta_{PE}PE_{prot}(t) + tl_{PE}PE_{RNA}(t) - k_{comp}peg_{RNA}(t)PE_{prot}(t) \tag{6.3}$$

$$\frac{dpeg_{DNA}(t)}{dt} = -\delta_{DNA-peg}peg_{DNA}(t) \tag{6.4}$$

$$\frac{dpeg_{RNA}(t)}{dt} = -\delta_{RNA-peg}peg_{RNA}(t) + tc_{peg}peg_{DNA}(t) - k_{comp}peg_{RNA}(t)PE_{prot}(t) \tag{6.5}$$

$$\frac{dPE_{complex}(t)}{dt} = -\beta_{complex}PE_{complex}(t) + k_{comp}peg_{RNA}(t)PE_{prot}(t) \tag{6.6}$$

Figure 6.2A shows how the concentrations of the different reactants evolve over time. The scale of the y-axis is displayed in the top right corner of each plot. The time point $t = 0$ marks the moment when pDNA has entered the nucleus and becomes



A: Model starting from pDNA.

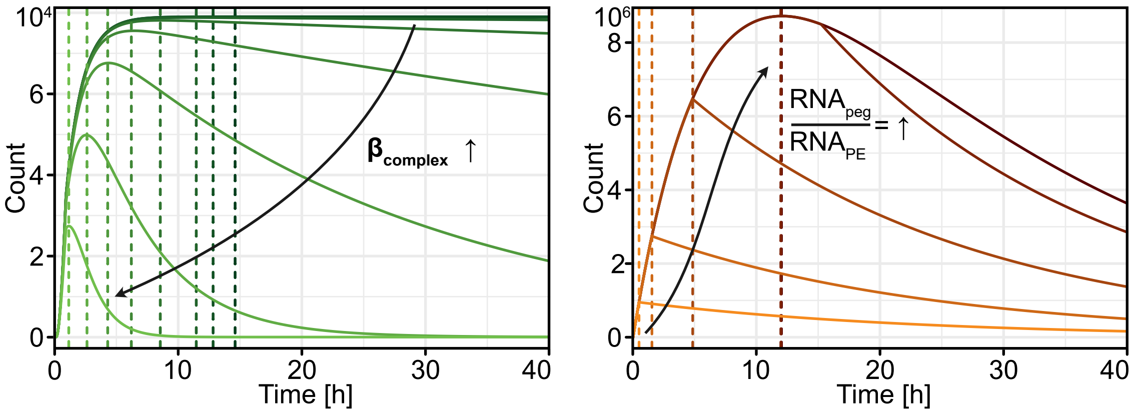
B: Model starting from mRNA.

Figure 6.2: Concentrations extracted from model for the different reactants. y-Axis scaling shown in the corner for each plot. Left shows the production of the two components for prime editing: PE and pegRNA. Right shows the concentration of assembled complex over time.

available for transcription. Our model shows that it takes approximately 4 h to reach the maximum concentration of the assembled prime editor complex, starting from the expression of pDNA. We observe that the pDNA concentrations are governed solely by pDNA degradation, as described in Equation 6.1 and Equation 6.4. RNA concentrations are initially determined by transcription from DNA. Once the pDNA is fully degraded, PE-RNA is governed by RNA degradation, as shown in Equation 6.2. The pegRNA does not reach the same maximum concentration as PE-RNA, since its concentration is also affected by its use in complex formation (see Equation 6.5). In the first few hours, the production of the complex is controlled by the production rates of pegRNA and PE protein, as well as the complexation parameter. At later times, it is governed only by the degradation parameter $\beta_{complex}$.

To adapt this model for RNA delivery, the plasmid steps are removed from the implementation, corresponding to the reactions written in line 2, 4, 7 and 10 in Figure 6.1B. Figure 6.2B shows how the kinetics change in this setup. PE-RNA and pegRNA are only degraded from their initial concentrations. The production of PE protein is slower, but reaches a higher maximum, as the transfected pegRNA is limited to the starting concentration. This also causes the maximum concentration of the assembled complex to occur within the first hour after delivery and to be only about half that of the pDNA system.

We then examined both delivery systems independently, varying one parameter at



A: pDNA

B: mRNA

Figure 6.3: The most influential parameter for the time of maximal assembled complex. Dashed lines mark the peak for each curve t_{peak}. Data for these simulations can be found in Table B.1.

a time to determine which had the greatest impact on the timing of the maximum assembled complex.

Figure 6.3A shows the variation of the $\beta_{complex}$ parameter and the corresponding concentrations of the assembled complex over time. The specific values of $\beta_{complex}$ and the times of the peaks are listed in Table B.1a. We found that this parameter had the greatest influence on the timing of the maximum. For $\beta_{complex} = 1$, the maximum complex concentration occurs at 1 h, but the peak value decreases to approximately 3×10^4 counts.

In the mRNA model, the most influential parameter was the initial ratio of PE-RNA and pegRNA (see Figure 6.3B). The higher the initial pegRNA concentration relative to PE-RNA, the later the maximum complex concentration is reached. This is logical, as complex formation takes time, and during this period, the pegRNA already degrades. Therefore, when more pegRNA is available, more complexes can form over time.

6.2. Influence of Resource Depletion in Reaction Networks

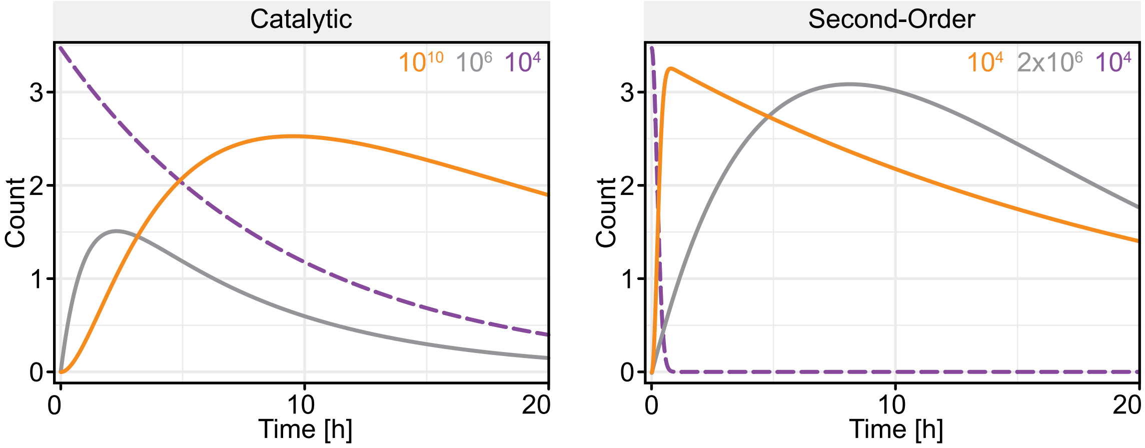
The previously introduced model of prime editing expression naturally assumed, that assembling the prime editing complex consumes both components, PE and pegRNA. As a result, we implemented this process as a second-order reaction. However, this complicates the solution of the ODE system. To address this, we conducted an *in silico* experiment to explore the possibility of using a simplified model. By considering different chemical reaction types, as explained in Subsection 2.1.4, we simplified the first- and second-order reactions to pseudo-first-order or catalytic reactions.

This modification is easily implemented in *Julia* by altering line 5 of Figure 6.1B. Figure 6.4A shows the catalytic system implementation on the left and the second-

```
k_comp, PE_prot + peg_RNA -->
                            PE_prot + peg_RNA + PE_complex
```

```
k_comp, PE_prot + peg_RNA -->
                            PE_complex
```

A: PE$_{\text{complex}}$ implementation in Julia



B: Graphical concentration

Figure 6.4: Prime editing expression from RNA. **Left:** Catalytic system. **Right:** Second-order system with k_{comp} set to $1 \times 10^{-5}\,\text{h}^{-1}$ for better illustration. pegRNA is shown as dashed purple line, PE protein as grey line and the complex as orange line. y-Axis scaling shown in the corner for each plot.

order reaction on the right.

For the mRNA model, the kinetics change significantly, as seen in Figure 6.4B. The t_{peak}) of the complex for the catalytic reaction shifts to later time points, around 9 h, compared to the previous first hour peak. However, the amplitude increases dramatically, reaching approximately 2.5×10^{10} counts compared to 3×10^{4} counts. This is expected because, in the second-order reaction, the limiting factor quickly becomes one of the two prime editing components, as shown in Figure 6.3B. This is particularly problematic in mRNA transfection. While PE is produced, pegRNA can only degrade, as mentioned in Section 5.3, causing t_{peak} to be primarily determined by the initial pegRNA concentration, as discussed in Section 6.1. In the catalytic reaction for the mRNA model, pegRNA is no longer consumed but acts as a catalyst for complex formation, reducing its influence on t_{peak}.

The initial ratio of PE to pegRNA was previously identified as a highly influential parameter, so we now revisit this for both the catalytic and second-order systems, comparing their effects on the pDNA and mRNA models. Figure 6.5 shows the concentration of the complex for different ratios, as listed in Table B.1b. For both models — pDNA and mRNA — we keep the starting concentration of PE, either as plasmid or RNA, constant and vary the corresponding concentration of pegDNA/RNA. The plots use the same y-axis scale but have different x-axis scales to better illustrate the early kinetics. The dashed line represents the second-order DNA maximum for the previously used ratio and is extended across all plots.

In the catalytic pDNA system, shown in Figure 6.5 (top left), the dashed line is

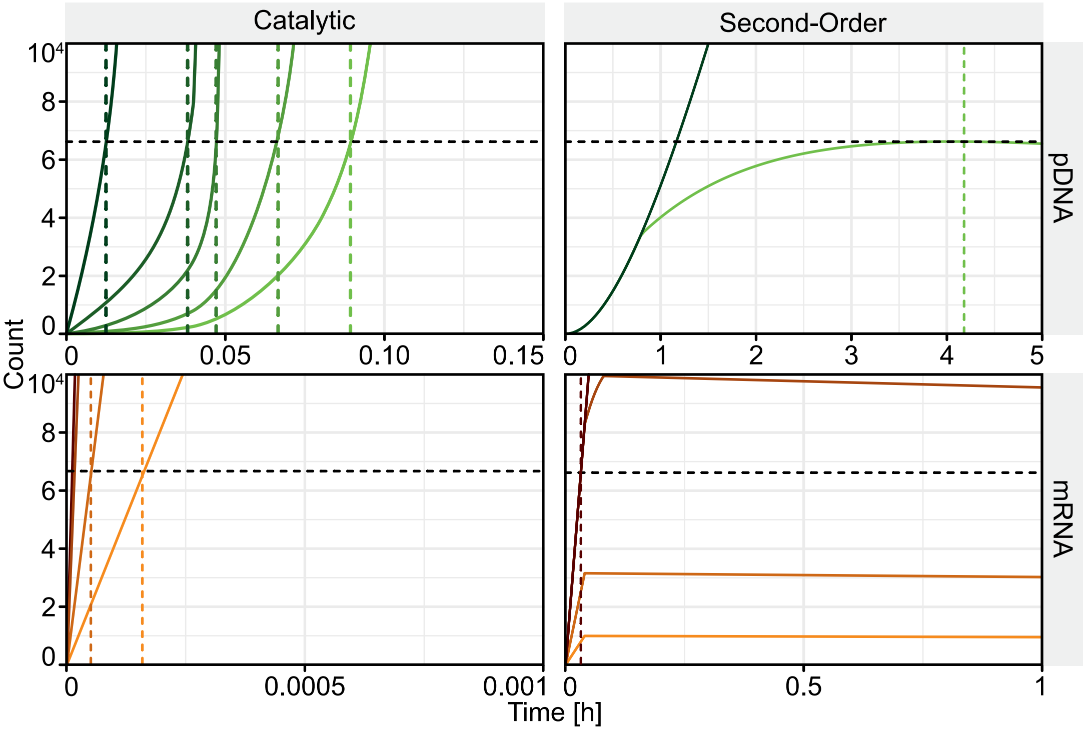


Figure 6.5: $PE_{complex}$ for different implementations of the prime editing expression. Different colored lines depict different starting ratios of PE:pegRNA as listed in Table B.1b. The horizontal black dashed line marks the maximum of the second-order pDNA system. Colored dashed lines mark the time different implementations take to reach this level.

reached extremely quickly for the lowest concentration, after 54 min, and continues to rise, as pegRNA and PE protein are produced almost exponentially. Higher pegRNA concentrations have little effect on this time. The second-order system (top right) reaches its maximum PE complex concentration at around 4 h for the experimental ratio, but higher pegDNA levels yield more complex, with the peak occurring at roughly the same time. The prime editing components are produced in such large amounts that the ratio becomes less critical in both pDNA models. The abundance of assembled complex leads to a high likelihood of rapid edits. Using a catalytic reaction instead of a second-order one in the pDNA model simply results in more material being produced over time (see additional Figure B.4).

As we know, the mRNA system forms the complex more quickly due to the absence of the transcription step. This leads to extremely high early concentrations in the catalytic mRNA system (bottom left). pegRNA is no longer a limiting factor and can continuously “catalyse” PE protein until the pegRNA is degraded. In contrast, the bottom right plot shows that pegRNA remains the limitation in the second-order system. Some ratios do not even reach the threshold set by the pDNA model. In summary, a catalytic system would distort the actual process by removing the pegRNA limitation for both delivery models.

6.3. Incorporating Nuclear Barriers: Delayed Prime Editing Complexation for mRNA Expression

In Section 5.3 we observed a delay between our mathematical model and the experimental values for mRNA transfection. We previously introduced a simple model and demonstrated that a measurable delay exists, concluding that it could be due to either complexation or slower target site localization, resulting from low concentrations of assembled complex. In this section, we propose a third possible cause for this delay: the nuclear barrier, a key difference between pDNA and mRNA delivery.

Each cell can be simplified into two compartments: the cytosol, which includes everything inside the cellular membrane, and the nucleus, enclosed by a barrier that not all molecules can pass through. Molecules can cross this barrier through various mechanics, such as diffusion or active transport [136]. We modeled this by introducing additional reactants and corresponding parameters. As shown in Figure 6.6, for PE and pegRNA, we now have two RNA reactants: RNA_{nuc} inside the nucleus and RNA_{cyto} in the cytosol. These can be converted by the rate *ntc* in a first-order reaction. PE-RNA and pegRNA have different transfer rates, similar to their transcription rates, due to their different sizes. The PE protein and the complex are also divided into cytosolic and nuclear reactants and can transfer into the nucleus at the rate *ctn*. In this setup, the complex can form either in the cytosol or the nucleus. Since the editing occurs inside the nucleus, we focus on the concentration of assembled complex in the nucleus for this model. The additional transfer parameters were taken from literature and are listed in Table B.3.

We now compare the nuclear barrier model to our simple model. Figure 6.7 shows the concentration of the prime editing complex for pDNA and mRNA, with the simple model represented by a dashed line and the nuclear barrier model by a solid line. On the left, we see that after 10 h, there is no longer a difference between the two models. However, the early time points are more interesting. The right plot, showing the time from 0 to 6 h, includes a dashed horizontal line as a visual reference to better compare

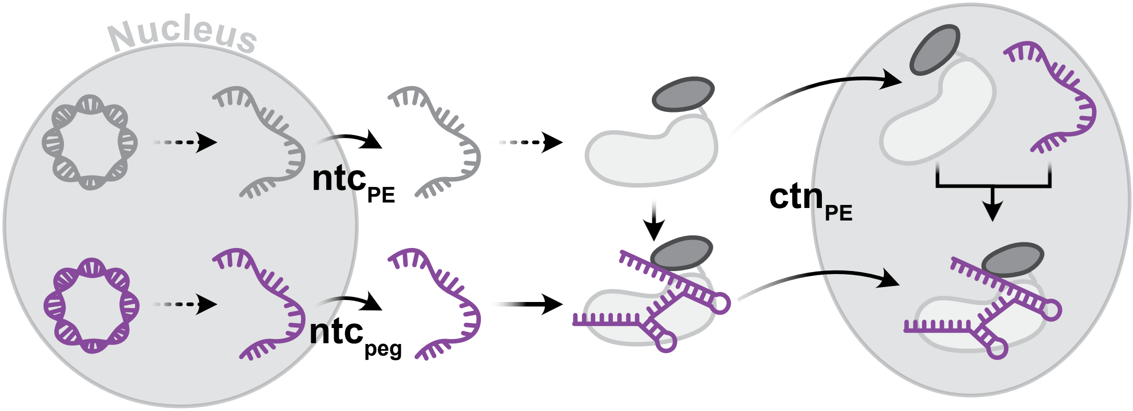


Figure 6.6: Model starting from pDNA introducing nuclear barriers and therefore transition rates ntc_{PE}, ntc_{peg} and ctn_{PE}.

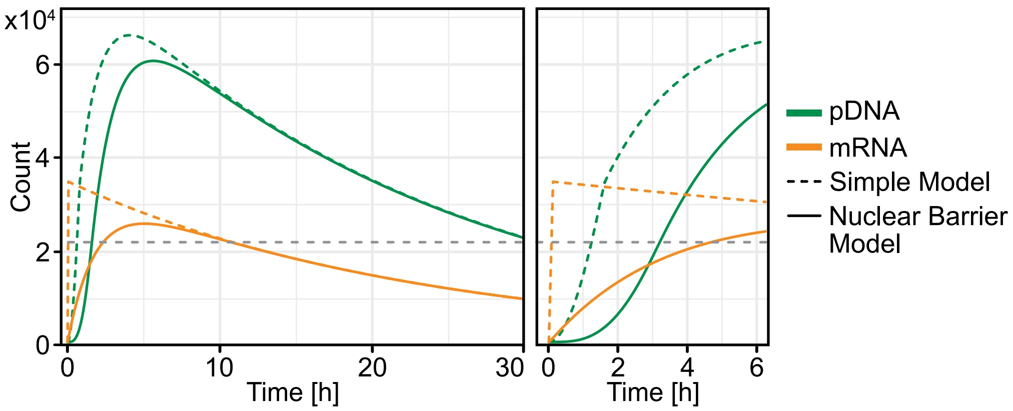


Figure 6.7: Comparison of simple model with the nuclear barrier model for pDNA and mRNA delivery. The dashed horizontal line serves as a visual reference to facilitate comparison of the different models.

the models.

In the simple model, mRNA leads to faster complex formation than pDNA, as expected. The nuclear barrier model, however, results in a slower onset of complex formation for pDNA. This is logical, as some complex forms inside the nucleus, but others still need to be transported there, slowing the overall increase in nuclear complex concentration and slightly reducing the peak concentration. However, the difference for the pDNA model is minimal.

In contrast, the nuclear barrier model significantly impacts the mRNA system, as shown in Figure 6.7. Whereas the complex was previously almost immediately available and then degraded as pegRNA was used up, it now takes approximately 4.5 h longer to reach the dashed line. This delay is explained by the complex being formed in the cytosol and then transported into the nucleus, as well as by PE protein entering the nucleus without a pegRNA. These proteins cannot perform any edits on their own and are thus unavailable for complexation, leading to a notable decrease in peak complex concentration.

With this new model, we again tested parameter variations, focusing on the newly established parameters ctn_{PE}, ntc_{PE}, and ntc_{peg}. The parameters and corresponding t_{peak} values are listed in Table B.4. Varying the parameter ctn_{PE} shows similar behavior in both mRNA and pDNA models. The lower the rate of transport into the nucleus, the less complex accumulates inside the nucleus, as some of the complex degrades before being transported into the nucleus when $ctn_{PE} < \beta_{complex}$. This leads to a shift in peak concentration times to later time points for both models, indicating that a high transport rate is beneficial for greater efficiency. This can be achieved, for example, by using multiple NLS sequences for the PE protein.

The ntc parameters can only be varied in the pDNA system, as shown in Figure 6.8. On the left, we see that increasing the rate ntc_{PE} shifts the peak of the complex concentration to earlier, from 24.6 h to as early as 5.4 h, while also increasing peak concentration levels. This suggests that the concentration of PE-RNA in the cytosol can limit the kinetics of complex formation. Interestingly, in the pDNA model,

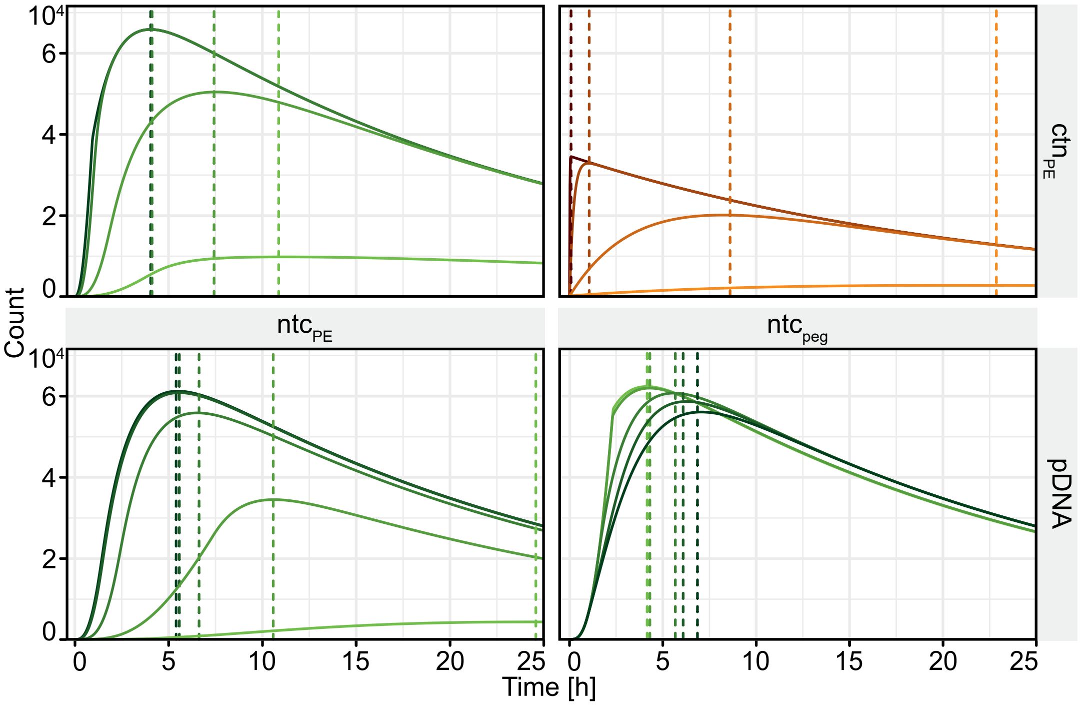


Figure 6.8: Parameter variations for nuclear barrier model. Green lines depict pDNA model, orange ones mRNA model. Three different parameters are varied. Line color indicates a increase in the value of the varied parameter, with lighter colors representing smaller values, and darker colors representing larger values. Additional data is listed in Table B.4.

pegRNA levels have less influence on the peak concentration and timing of the complex concentration, as seen on the right. A further range of ntc_{peg} values was tested compared to ntc_{PE}, but the model showed only a slight shift, from 6.87 h to 4.17 h. This is likely because pegRNA is produced in greater quantities than PE-RNA, as it is significantly smaller.

6.4. Discussion

This chapter explored simulating the kinetics of prime editing expression using a series of models implemented in *Julia*. We began with a coarse-grained, simple model, which helped identify key factors in experimental design. Notably, we found that the initial ratio of transfected pegRNA to PE-RNA plays a crucial role in mRNA delivery. This significantly impacted the efficiency and timing of complex formation.

Next, we explored the effects of different reaction types — catalytic and first/second-order. Although catalytic reactions simplify ODE systems, they heavily distorted the behaviour of our system. The expression dynamics and limitations inherent to a first-order reaction were eliminated, highlighting the need to accurately define the underlying chemical kinetics and processes.

Finally, we refined the model by introducing a nuclear barrier and transport parameters. For the mRNA model, this addition shifted the timing of complex formation, delaying the peak concentration compared to earlier models. The nuclear barrier added complexity, as PE and pegRNA must pass through the nuclear membrane, altering the overall kinetics.

The simple model allowed us to explore the basic dynamics of prime editing expression but did not capture the complexity needed to fully explain experimental results, particularly for mRNA delivery. In contrast, while the catalytic model reduced the complexity of the ODE system, it severely distorted the expression dynamics by removing key limitations present in real systems. This indicates that catalytic models are not suitable for this context.

The nuclear barrier model, while more accurate, introduced additional parameters related to nuclear transport that increased system complexity. Although this model provided more detailed insights into the dynamics of complex formation, it requires further refinement by measuring transport parameters, which were only estimated from literature for the scope of this thesis.

To better model our system, more of the parameters used here should be measured directly in this system. For example, fluorescence resonance energy transfer could be used to determine complex formation [137] or GFP-tagged Cas proteins for nuclear transport rates [138].

As proposed in the experimental chapters, a hybrid system of pegRNA as pDNA and PE as mRNA could first be tested *in silico* to guide design of such experiments and determine whether sequential or simultaneous transfection would yield higher efficiency and shorter timing. This demonstrates the usefulness of simulations in guiding experimental design before running time- and resource-intensive experiments.

In conclusion, while modeling gene expression with differential equations is well-established [40], the use of *Julia* offers a more efficient approach for implementing chemical rate equations in the form of ODE systems [135]. *Julia's* flexibility allowed us to study the system at all stages of the experiment, from design to data analysis. By iteratively using simulations to test hypotheses and guide experimental setups, creating a feedback loop where experimental data informs simulations and vice versa. This iterative approach offers a powerful method for understanding and optimizing prime editing systems.

7. Search-and-Replace Prime Editing as Implementation for Biocomputing

Current implementations of biocomputers are based on Boolean logic, approximating the operation of silicon-based computers. However, this approach "fundamentally limits the types of computation that may be engineered inside the cells and fails to exploit the rich and diverse functionality available in natural living system" [139]. One notable advantage, for instance, is DNA's information density, which is orders of magnitude greater than that of conventional storage technologies [140]. As early as 1994, Fontana and Buss [141] proposed a biocomputer using the Lambda calculus, which was explained in Subsection 2.2.3.

In the following, we present our approach on implementing a version of Lambda calculus, specifically the SKI combinator calculus, using prime editing. We begin by defining general requirements for the DNA system and demonstrate how the Lambda calculus components can be translated into cellular machinery. Here, a computation is encoded in the DNA as a Lambda expression. For evaluating (or beta-reducing) the function applications within these expressions multiple predefined pegRNA sequences together with PE are used. We then propose a system for implementing the I combinator, but note that this quickly necessitates the introduction of a garbage collection operator. Subsequently, we introduce a version of the K combinator. Finally, we conclude with a discussion and outlook on the potential of a SKI combinator-based cellular computer.

7.1. Defining the SKI Combinator Calculus on DNA

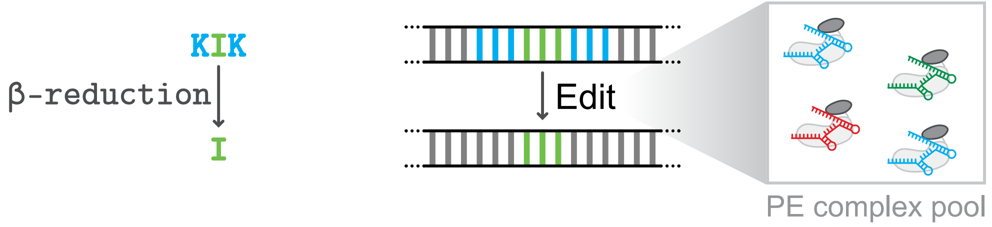
In Subsection 2.2.3, we introduced the Lambda calculus and its three components: variables, Lambda abstractions and function application. In the following, we propose a theoretical framework for implementing the SKI combinator calculus, as shown in Subsection 2.2.4, derived from Lambda calculus. In this framework, the need for variables is eliminated and replaced by three combinators: the identity function I, the K combinator, and the S combinator with the following applications:

$$\mathtt{I}\ \mathtt{x} \rightarrow \mathtt{x} \tag{7.1}$$

$$\mathtt{K}\ \mathtt{x}\ \mathtt{y} \rightarrow \mathtt{x} \tag{7.2}$$

$$\mathtt{S}\ \mathtt{x}\ \mathtt{y}\ \mathtt{z} \rightarrow \mathtt{x}\ \mathtt{z}\ (\mathtt{y}\ \mathtt{z}) \tag{7.3}$$

Manzoni et al. [142] proposed that a biocomputer requires three main elements: inputs, computational module and an output. First, the system must be able to read



A: In the SKI combinator calculus, the computation is represented by the expression KIK. This expression is evaluated through beta reduction, resulting in I.

B: The computation is encoded in the DNA. Colors are used to represent the different combinators. Prime editing, using a pool of various PE-pegRNA complexes, is employed to beta reduce the expression to its beta-normal form.

Figure 7.1: Transfer from SKI combinator calculus to a DNA/PE-pegRNA system

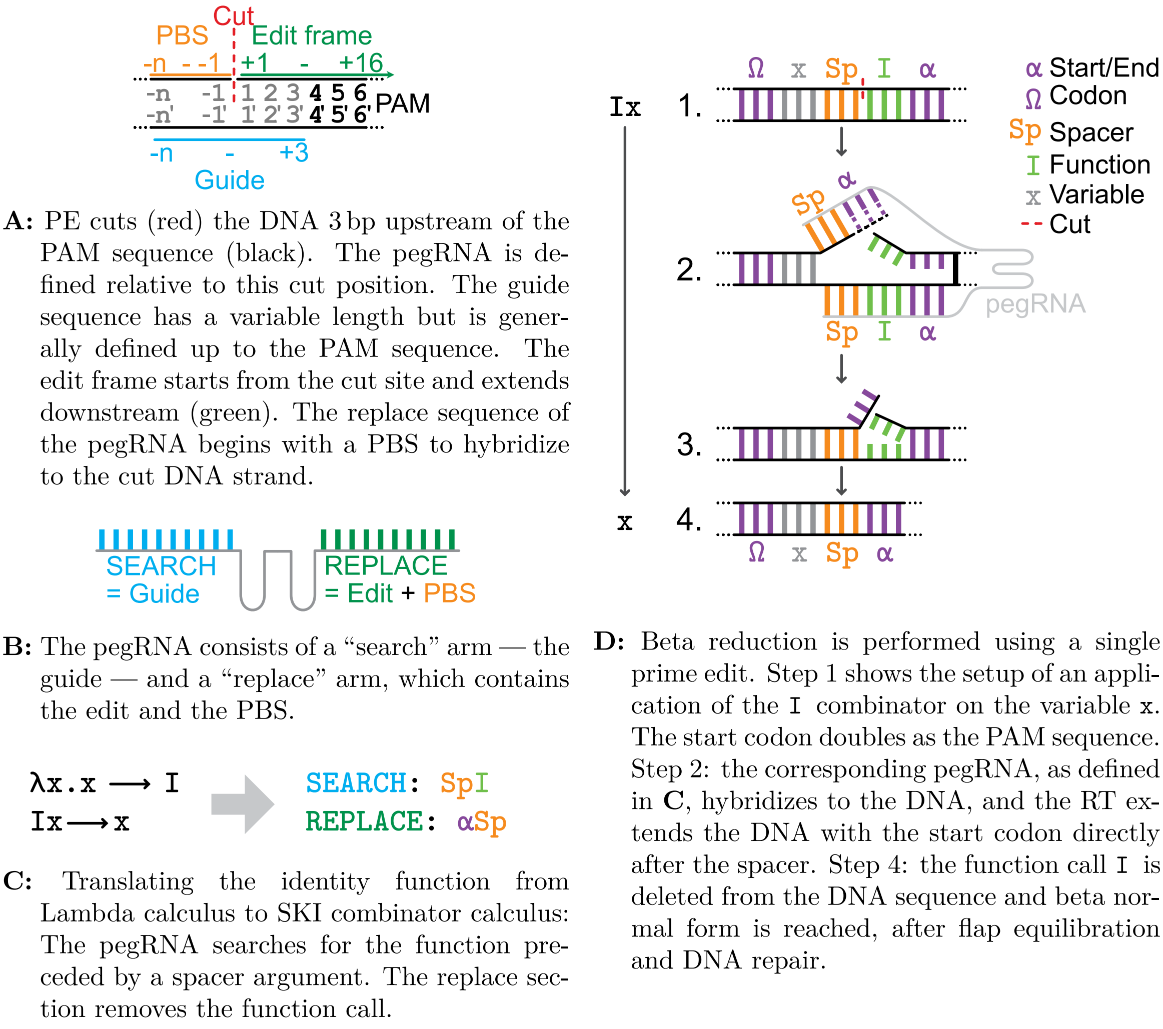
and react to inputs, for example, using cellular sensors such as calcium ion channels like TRPV1, which responds to temperature and chemical stimuli [143]. Second, a computational module is needed to perform a desired function based on the input, producing a change in the system. Lastly, the system must have a way to express the result of the function application, the output.

Using the SKI combinator calculus, the input is defined by the expression to be evaluated. This expression is stored as a sequence of combinators. `x`, `y` and `z`, as seen in the definitions of the combinators, are variables, but they can only be drawn from the set of combinators {`I`, `K`, `S`}. In Figure 7.1A, we see such a combination of combinators (input), while Figure 7.1B shows the encoding of these combinators on DNA. Function application and beta reduction provide the computational module, reducing the expression `KIK` to a single `I`. To perform this beta reduction using the biocomputer, we utilize the previously studied prime editing system. By designing different pegRNAs, we form a PE-pegRNA pool that reduces the expression to its beta-normal form, which can no longer be reduced.

This approach defines DNA as the storage medium for the expression to be evaluated, the desired computation encoded as a specific DNA sequence. The DNA must be stored in a way that allows editing by the PE-pegRNA complexes to perform beta reduction. We, therefore, propose integrating it stably into the system, likely through cloning, similar to how our fluorescent reporter was implemented in the HEK cell line. The expression of the PE-pegRNA complex needs to be stable enough to ensure high-efficiency edits, as the beta reduction depends on this process.

7.2. Implementing the I Combinator Utilizing Prime Editing

We propose that a combinator can be encoded by a specific DNA sequence. The design of our functions and edit lengths is guided by the requirements for prime editing and



A: PE cuts (red) the DNA 3 bp upstream of the PAM sequence (black). The pegRNA is defined relative to this cut position. The guide sequence has a variable length but is generally defined up to the PAM sequence. The edit frame starts from the cut site and extends downstream (green). The replace sequence of the pegRNA begins with a PBS to hybridize to the cut DNA strand.

B: The pegRNA consists of a "search" arm — the guide — and a "replace" arm, which contains the edit and the PBS.

C: Translating the identity function from Lambda calculus to SKI combinator calculus: The pegRNA searches for the function preceded by a spacer argument. The replace section removes the function call.

D: Beta reduction is performed using a single prime edit. Step 1 shows the setup of an application of the I combinator on the variable x. The start codon doubles as the PAM sequence. Step 2: the corresponding pegRNA, as defined in **C**, hybridizes to the DNA, and the RT extends the DNA with the start codon directly after the spacer. Step 4: the function call I is deleted from the DNA sequence and beta normal form is reached, after flap equilibration and DNA repair.

Figure 7.2: The I combinator is translated into DNA, and the corresponding beta reduction is performed using prime editing.

pegRNA design. Section 2.1 briefly explained pegRNA design. Figure 7.2A shows a target dsDNA with base pairs numbered from $-n$ to 6. The PAM sequence, recognized by the Cas protein, is located at base pairs $4-6$. Three base pairs upstream, between -1 and 1, the Cas protein cuts the upper strand of the DNA. The pegRNA has two arms: one for the search operation and one for the replace operation (Figure 7.2B). The guide/search sequence of the pegRNA is complementary to the bottom strand of the target DNA from position $-n$ up to, and possibly including, the PAM sequence. A PBS, complementary to the $-n$ to -1 bp of the upper strand, and the desired edit sequence comprise the replace strand of the pegRNA. The edit frame is always downstream of the cut site. To perform a deletion in this system, the edit sequence needs to omit the desired base pairs from the DNA target sequence.

Figure 7.2C depicts the identity combinator in Lambda calculus, which returns the input variable, meaning the application of Ix returns x. When translating this into

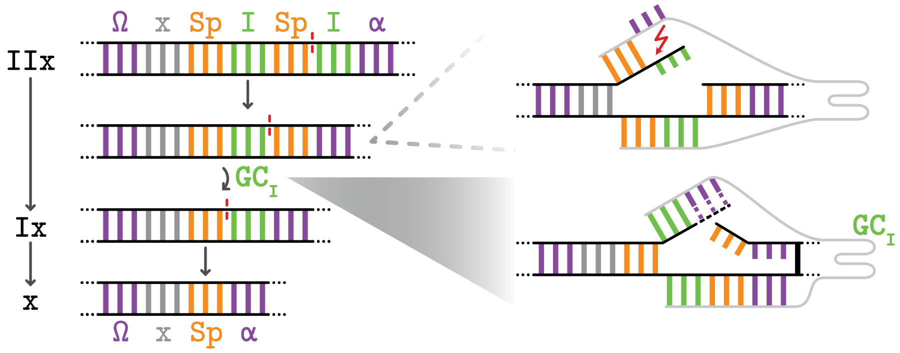


Figure 7.3: Applying the identity function twice to a variable leaves a spacer between the next identity operator and the variable. Even if the pegRNA hybridizes to this DNA sequence, reverse transcription may be inhibited by the remaining function sequence. To resolve this, we introduce a set of pegRNAs to clean up such remnants. After this step, the final beta reduction can proceed.

an I combinator pegRNA, we propose that the search sequence be encoded as SpIα and the replace sequence as αSp. This leads to a simple deletion of the I combinator from the DNA expression, independent of the true combinator for the x variable. We introduced α and Sp here, which are encoded sequences used for hybridization of the pegRNA to corresponding sequences in the target DNA. The reasoning for these will be explained in the following.

α and Ω define the start and end of our expression to be evaluated and can be used as primers when performing NGS as readout, for example. Figure 7.2D shows the simple application of the identity operator on a variable x from the set of combinators {I, K, S}, as encoded by ΩxSpIα (Figure 7.2D **1.**). α also encodes for a PAM sequence, which serves as the binding site for the PE and defines the cut site. The expressions encoded in DNA are backwards compared to the standard SKI combinator calculus because the DNA edit frame is always downstream of the cut site, toward the PAM. The pegRNA hybridizes to the bottom DNA strand in step 2 (in Figure 7.2D **2**), allowing the Cas protein to cut the upper strand. This, in turn, enables the second half of the pegRNA to bind to the now exposed flap. These binding dynamics are the reason for including Sp. The flap is then extended with α by the RT fused to the Cas protein, as shown in Figure 7.2D **3.**. After DNA repair, the I combinator is successfully removed, and the variable x remains.

The I combinator is designed to remain completely flexible to any input, as the input is not part of the pegRNA. Therefore, only one pegRNA is needed. Incorporating the variable x into the search sequence would instead require three different pegRNAs, one for each combinator. While this is still feasible for the I combinator, applying this approach to the S combinator would require at least $3^3 = 27$ versions, accounting for only the basic operators and no combinations. This would significantly increase the required pegRNA pool, making it impractical.

We previously saw a single application of the I combinator translated to DNA. Now, we will run a few test cases to verify our implementation. First, we apply the I combinator multiple times to a variable x , as shown in Figure 7.3. The initial step mirrors the previous beta-reduction, where the I between Sp and α is removed. However, we then observe a leftover spacer in front of the next I. Even when the Cas protein can bind to the DNA and the pegRNA hybridizes to it, the reverse transcription of α may be blocked by the remaining combinator. Therefore, we introduce additional pegRNAs to remove any leftover Sp between a combinator and the start codon α:

$$\texttt{GC}_{\texttt{I}}\texttt{: I Sp } \alpha \rightarrow \texttt{I } \alpha$$
$$\texttt{GC}_{\texttt{K}}\texttt{: K Sp } \alpha \rightarrow \texttt{K } \alpha$$
$$\texttt{GC}_{\texttt{S}}\texttt{: S Sp } \alpha \rightarrow \texttt{S } \alpha$$

In computer science, **garbage collection** refers to the process of automatically identifying and removing unused or redundant data from memory, ensuring efficient system operation and preventing clutter [144]. Similarly, since we are left with the residual sequence Sp or “garbage”, which impedes the next evaluation steps, we term these pegRNAs garbage collectors (GCs). Just as garbage collectors in computer systems free up memory to allow for new operations, these GCs clear away unnecessary or obstructive DNA sequences, enabling the next combinator to be evaluated. We can now test three more variations:

Ω I x Sp α → Ω I x α : In this case, the previously introduced GCs remove the spacer, depending on the variable x.

Ω I α → Ω I α : Since beta-normal form is already reached, no further action occurs. The I combinator might bind to the DNA, but without a Sp, no edit is happening.

Ω I Sp I x α → ? : This expression depends on the variable x, as the calculus is a right-associative in our case, meaning x is evaluated first.

7.3. Designing a K Combinator Variant

The K combinator, compared to I , is slightly more complex as it takes two inputs, x and y . It is typically defined to select the first of two inputs, but in our system, we instead select the second input, reducing the K combinator to a deletion operation using prime editing. As shown in Figure 7.4, Kyx reduces to x. This is encoded as Ω x Sp y Sp K α in the DNA.

The K combinator encodes for a different PAM sequence than α, so to shift the cut site further upstream, as illustrated in Figure 7.4. However, instead of simply removing SpK, we replace it with a sequence $\texttt{K}_2$. This sequence $\texttt{K}_2$ is crucial because, only removing the spacer and K combinator would lead to ΩxSpyα. The system would evaluate y next and fail to remove it, thus not selecting x. For the first evaluation step of K, we need the replace arm of the pegRNA to hybridize to the variable y, leading us

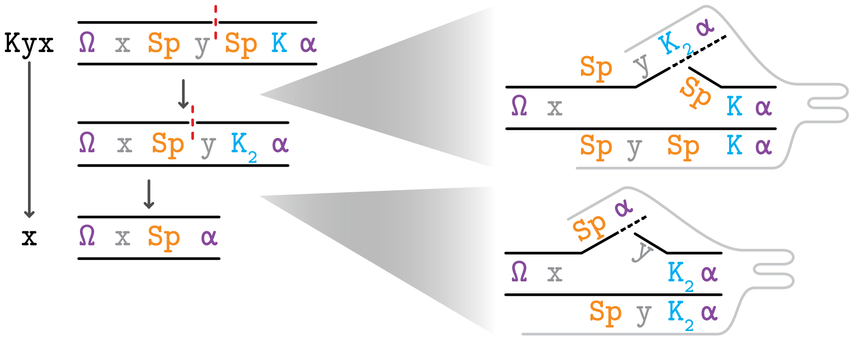


Figure 7.4: The K combinator chooses one of two arguments, in our case always the second one. The application uses a K combinator which contains a PAM sequence, therefore shifting the cut site upstream (red). 3 different versions of K exist — K_{1I}, K_{1S}, and K_{1K} depending on what the variable y stands for. To then completely remove y a second edit step using the operator K_2 is needed. There are 3 different versions of K_2 as well. The right side depicts, how the combinators K and K_2 are implemented as pegRNAs.

to a set of K combinators: $\{K_{1I}, K_{1S}, K_{1K}\}$. The K_2 sequence must also include a PAM sequence to shift the cut site to the left of the variable y, ensuring the independence of it. The K_2 pegRNA then removes yK_2 and reduces the expression to the beta-normal form, x.

In the case of only getting one input, such as Kx, the computation becomes incomplete.

$$\Omega \mathrm{xSpK}\alpha \rightarrow \Omega \mathrm{xK}_2\alpha \rightarrow ?$$

Here we see that our version of the K combinator breaks down when only one input is provided. The first editing step is still possible, but the computation halts at the second step due to the missing Sp. This issue will need to be addressed in further research, as will be the implementation of the S combinator.

7.4. Discussion

In this chapter, we proposed a computational model based on the SKI combinator calculus, implemented in DNA using prime editing for the crucial step of beta reduction. Similar to the CRISPR sequences found in bacteria and archaea, we introduced a pool of pegRNAs to represent the three combinators, as summarized in Table 7.1.

The single application of the I combinator required one pegRNA, with the “search”-sequence of $\mathrm{SpI}\alpha$ and “replace” of $\mathrm{Sp}\alpha$, resulting in a single editing step. However, we observed that when an expression contains two applications of the I combinator, additional pegRNAs are needed to clean up the DNA after each editing step. We termed these additional pegRNAs garbage collectors (GCs).

Next, we introduced the K combinator, which requires two sequential editing steps. Since the first step could not be designed with maximal flexibility, we opted to instead

Family	Name	Search		Replace
I combinator	$\texttt{I}$	$\texttt{SpI}\alpha$	→	$\texttt{Sp}\alpha$
K combinator	$\texttt{K}_{1\texttt{I}}$	$\texttt{ISpK}\alpha$	→	$\texttt{IK}_2\alpha$
	$\texttt{K}_{1\texttt{K}}$	$\texttt{KSpK}\alpha$	→	$\texttt{KK}_2\alpha$
	$\texttt{K}_{1\texttt{S}}$	$\texttt{SSpK}\alpha$	→	$\texttt{SK}_2\alpha$
	$\texttt{K}_2$	$\texttt{SpxK}_2\alpha$	→	$\texttt{Sp}\alpha$
Garbage collectors	$\texttt{GC}_{\texttt{I}}$	$\texttt{ISp}\alpha$	→	$\texttt{I}\alpha$
	$\texttt{GC}_{\texttt{K}}$	$\texttt{KSp}\alpha$	→	$\texttt{K}\alpha$
	$\texttt{GC}_{\texttt{S}}$	$\texttt{SSp}\alpha$	→	$\texttt{S}\alpha$

Table 7.1: Overview of pegRNAs for the SKI combinator calculus.

provide three pegRNAs with a similar setup for each possible combinator. The second editing step of the K combinator, however, only required one pegRNA.

Both a double application of the I combinator and the two-step K combinator edits need to happen sequentially. As discussed in previous chapters, assuming PE and pegRNAs are stably expressed from the genome, a single edit step could take approximately 3 h, leading to a total computation time of around 6 h. This time increases significantly if transfection, instead of stable integration, is used. For instance, based on our results in Chapter 5, transfection with pDNA could take between 10 h and 80 h, with efficiencies around 60 %.

These estimates highlight the importance of efficiency and timing. While a standard Lambda calculus implementation may not be highly dependent on timing due to its ability to parallelize operations, our system requires strict left-association. This limitation might be addressed by omitting the start-codon to increase flexibility and enable parallelism. However, without this start-codon, ensuring the correct order of computations remains necessary.

We designed our system using base pair sets of three, aligning with both the natural codon structure and the cut-PAM distance of SpCas9. However, this is not strictly required. Other Cas proteins with different distances could also be used for prime editing, potentially simplifying the K combinator and aiding in the design of an S combinator [145]. When implementing this system *in vitro*, these properties would need to be carefully determined and optimized. As seen in Chapter 5, not only does the delivery system matter for the duration of an edit, but also the length of the edit frame. Using two editing steps for a single combinator could be reduced to one with a Cas protein that has a longer cut-PAM distance.

In designing our pegRNA pool, we made several assumptions regarding how deletions work and how the RT could be hindered. These assumptions should be verified in a lab setting to ensure accurate design before further development.

Our experiments were conducted in mammalian cells, a well-established model for prime editing. However, often integrating longer DNA sequences into these cells can be challenging [146]. A better approach for *in vitro* experiments could be to use *Saccharomyces cerevisiae*, as it is easier to maintain and integrates given DNA stably into its genome [147]. Creating a biocomputing yeast strain that constantly expresses the PE protein and the pool of pegRNAs would be advantageous. A designated homology region in the yeast genome could allow easy insertion of different encoded expression, followed by NGS after a few generations to extract the solution.

There is also evidence suggesting that prime editing is dependant on the cell cycle phase [23, 148]. Yeast has a shorter doubling time (90 −120 min under optimal conditions) compared to mammalian cells like HEK293, which take 34 −36 h [147, 149]. This could further reduce computation times and should be considered when designing an *in vitro* biocomputer.

Once a yeast system is established, cell-free yeast systems could also be considered. There are already synthetic yeast systems available for different purposes [150], which may be even easier to maintain and eliminate the need for cloning, as editing could be performed directly on added dsDNA.

Another critical aspect is the readout process, we used a fluorescent reporter system, which is limited to Boolean “on” and “off” results. While NGS was mentioned as a possible readout, it could lead to multiple or undefined results unless a single-cell isolation is performed beforehand. Alternatively, for a limited number of combinator permutations, one could design a DNA origami with ssDNA docking stations that emit different fluorescence signals when hybridized to a complementary strand. Adding a solution of DNA in supposedly beta-normal form, one could then image the results and determine their distribution [151]. This method could be faster and more precise than a single fluorescent reporter [151].

On a more abstract note, Grozinger et al. [139] pointed out that the inherent stochasticity and determinism of biological systems could be advantageous for specific computational tasks. However, we must be careful not to oversimplify these processes or equate them with electronic systems, as they are inherently less reliable and predictable. Therefore, extensive knowledge about the robustness, efficiency, timing of these systems and a proper readout for biocomputers is essential.

8. Conclusion and Outlook

In this thesis, we set out to investigate and learn to control the kinetics of prime editing to achieve faster genomic edits without a loss of efficiency. This is critical for the development of therapies for genetic diseases and the foundations of biocomputers. The ability to precisely control the timing and efficiency of prime editing is essential, as some applications require rapid, accurate edits, while others benefit from prolonged exposure [26, 27, 126].

Our study made several contributions to the field of prime editing. We began by developing a method to monitor prime editing events at the single-cell level using a HEK cell line stably expressing a blue-shifted fluorescent reporter. A successful prime edit in the genomic sequence of this reporter restored the fluorescence to its original green color. By adapting the LISCA assay, we were able to define the time-to-edit, from transfection to the onset of green fluorescence, with a median of 20 h for pDNA transfection. Knowing from literature that mRNA-based delivery of fluorescent reporters is typically faster than pDNA-based delivery, we compared the two approaches to determine whether mRNA-delivery could speed up prime editing. While mRNA-based prime editing was several hours faster than pDNA, it exhibited significantly lower efficiency, and the time difference was not as pronounced as expected. Although pDNA offers higher efficiency, it leads to longer exposure and carries the risk of off-target effects. mRNA, though less efficient and potentially requiring re-dosing, is comparable to pDNA on the time scale for edits. Therefore, the choice between pDNA and mRNA should prioritize factors like delivery vehicle or off-target effects, rather than time-to-edit.

To investigate the unexpectedly slow mRNA time-to-edit further, we used a plasmid that co-expresses a red fluorescent protein alongside the prime editor, allowing us to track the expression of editing components and measure the editing time. We found that median mRNA-based editing was two hours slower than pDNA-based editing. We hypothesized that the time-to-edit depends on two separate time scales: from delivery to the expression of the prime editing components, and the editing process until the expression of the fluorescent reporter. We estimated this using a mathematical convolution of these two time distributions. While our model accurately predicted the pDNA time-to-edit, it showed a discrepancy for mRNA-based delivery, indicating that reaching similar complex concentration takes 15 times longer starting from the mRNA model compared to the pDNA model. Further simulations revealed that mRNA-based prime editing is constrained by the initial ratio of prime editor mRNA to pegRNA, limiting the speed of editing. This suggests that strategies like dual transfection or re-dosing could improve the kinetics of mRNA-based editing.

We also explored the relationship between edit length and editing kinetics in the pDNA system, where longer edits correlated with slower editing times. This aligns with the strand displacement theory, offering insights for designing more efficient editing strategies, such as determining whether two sequential edits might be more time efficient than a single long edit. More studies about the kinetics of strand displacement, also including mismatches like in the work of Irmisch et al. [130], could improve models of timing and efficiency.

Our investigation into the use of mathematical models with ODEs to simulate prime editing kinetics highlighted the power of simulations in experimental design. We demonstrated that even simple expression models reveal the differences between mRNA and pDNA delivery but require a nuanced understanding of reaction dynamics and their mathematical implementation. These models are valuable tools for guiding experimental setups but must be grounded on robust parameter estimations from literature or independent experiments to avoid over-complicating or oversimplifying the system.

Lastly, we proposed a theoretical framework for implementing parts of the SKI combinator calculus using DNA and prime editing as a beta reduction machinery. Although this concept demonstrates the potential of prime editing in synthetic biology, the complexity of the kinetics indicates that more detailed knowledge of the editing process is necessary before such systems can be reliably realized.

The experimental scope of this thesis was limited to a single editing locus in one cell line to ensure consistency and comparability across delivery systems. However, it is known that prime editing efficiency can vary widely between different loci, likely affecting the kinetics as well. Design choices such as PBS length, guide sequence and edit length, influence efficiency and likely kinetics, too.

Future research, particularly with the goal of drug delivery in mind, should explore how these kinetic principles apply across a broader range of loci, cell types, and editing designs. This could include delivery of the PE-pegRNA complex as fully assembled RNPs or studying the impact of different UTR sequences on the expression kinetics of the prime editing components delivered by mRNA. Hybrid delivery systems, combining pDNA- and mRNA-delivery or using sequential transfection, could also be of interest for drug delivery approaches.

Before testing the proposed biocomputer system, further studies are needed to understand the impact of variables such as mismatches, PAM sequences, and the use of alternative Cas proteins for prime editing. We also proposed yeast as a potential model organism for biocomputers. Thus, future research should repeat these kinetic studies in yeast to validate our findings and explore the development of a yeast-based biocomputer.

In conclusion, this thesis presents a kinetic study of prime editing at the single-cell level. We initially explored the dynamics of different delivery systems for the expression of the prime editing components, revealing an unexpected discrepancy, when transfecting mGL-mRNA directly compared to the mRNA-based prime editing time-to-edit. We introduced a new fluorescent reporter to measure the expression of the PE protein and examined the kinetics of the editing time for both mRNA delivery and different editing lengths in the pDNA system. Through the use of mathematical models, we evaluated their accuracy and the influence of model parameters, showing that mRNA delivery is not always the best option for prime editing and that pDNA should also be considered. Lastly, we proposed a framework for implementing part of the SKI combinator calculus in DNA using prime editing for beta reduction.

A. Experimental Protocols and Data Evaluation Scripts

A.1. Manufacturing of Material

A.1.1. Plasmids

Plasmids were stored in NEB 5-alpha Competent *E. coli* (C2987H, NEB) cells at −70 °C. These cells were transformed following the manufacturers instructions. For plasmid multiplication, a single colony was picked from an LB agar plate and grown over night in 5ml LB supplemented with the corresponding antibiotics at 37 °C shaking with 250 rpm to ensure optimal growth and plasmid replication. The antibiotic used was either ampicillin at a final concentration of 100 µg/mL or kanamycin at a final concentration of 50 µg/mL, depending on the plasmid's resistance marker. The next day a plasmid extraction was performed using standard miniprep kits (Quiagen, Thermo Fisher, NEB) accodring to the manufacturer's instructions. The concentration of the final pDNA was measured using a spectrophotometer (Nanodrop) and diluted to the specific stock solution required and stored at −20 °C.

Plasmid	**Promoter**	**ORF**	**Antibiotic**	**From**
mGL	CMV	mGreenLantern	KanR	Westmeyer Lab
mScarlet3_C1	CMV	mScarlet	KanR	Addgene #189753
pCMV-PE2	CMV	PE2	AmpR	Addgene: #132775
pCMV-PE2-P2A-mScarlet	CMV	PE2 + P2A + mScarlet	AmpR	Cloned by vectorbuilder
GA890	U6	pegRNA(short)	AmpR	Westmeyer Lab
L1135	U6	pegRNA(middle)	AmpR	Westmeyer Lab
L908	U6	pegRNA(long)	AmpR	Westmeyer Lab

Table A.1: Overview of plasmids for experiments requiring pDNA transfection

A.1.2. mRNA/IVT

The plasmids for IVT were kindly provided by the Westmeyer lab, transformed and stored like the other plasmids. After using a maxi prep kit (Quiagen) following the manufacturers instructions 3 µg of template plasmid were linearized by digestion with AsiSI and purified by gel electrophoresis. IVT was performed using the Kit XY with the following modifications:

- **mGL/mScarlet:** HiScribe T7 mRNA Kit with CleanCap Reagent AG (NEB: E2080S), substituting UTP with Pseudo-UTP from Jena Bioscience (NU-1139S) for increased stability of the resulting RNA

- **PE2, PE2-P2A-mScarlet:** HiScribe T7 mRNA Kit with CleanCap Reagent AG (NEB: E2080S) with Pseudo-UTP from Jena Bioscience, also for increased stability, since the resulting RNA is of a larger order than usual RNAs

- **pegRNA:** HiScribe T7 Quick High Yield RNA Synthesis Kit (NEB: E2050S) was used without modified NTPs

The DNA template was removed by DpnI digestion (NEB). mRNA isolation was conducted using the Monarch RNA Cleanup Kit (NEB #T2050) following the manufacturer's instructions. The concentration of the RNA was then measured with a nanodrop, diluted to the desired stock concentration and stored at −70 °C.

Plasmid	**Promoter**	**ORF**	**Antibiotic**	**From**
IVT-mGL	T7	mGL	AmpR	Westmeyer Lab
IVT-PE2	CAG	PE2	AmpR	
IVT-PE2-P2A-mScarlet	CAG	PE2, P2A, mScarlet	AmpR	
IVT-pegRNA	T7	GA890	AmpR	

Table A.2: Overview of plasmids for IVT for experiments requiring mRNA transfection

A.1.3. mGreenLantern DNA Sequence

```
  1 ATGGT GAGCA AGGGC GAGGA GCTGT TCACC GGGGT GGTGC CCATC CTGGT CGAGC
 56 TGGAC GGCGA CGTAA ACGGC CACAA GTTCA GCGTC CGCGG CGAGG GCGAG GGCGA
111 TGCCA CCAAC GGCAA GCTGA CCCTG AAGTT CATCT GCACC ACCGG CAAGC TGCCC
166 GTGCC CTGGC CCACC CTCGT GACCA CCTTA GGCTA CGGCG TGGCC TGCTT CGCCC
221 GCTAC CCCGA CCACA TGAAG CAGCA CGACT TCTTC AAGTC CGCCA TGCCC GAAGG
276 CTACG TCCAG GAGCG CACCA TCTCT TTCAA GGACG ACGGT ACCTA CAAGA CCCGC
331 GCCGA GGTGA AGTTC GAGGG CGACA CCCTG GTGAA CCGCA TCGTG CTGAA GGGCA
386 TCGAC TTCAA GGAGG ACGGC AACAT CCTGG GGCAC AAGCT GGAGT ACAAC TTCAA
441 CAGCC ACAAG GTCTA TATCA CGGCC GACAA GCAGA AGAAC GGCAT CAAGG CTAAC
496 TTCAA GACCC GCCAC AACGT TGAGG ACGGC GGCGT GCAGC TCGCC GACCA CTACC
551 AGCAG AACAC CCCCA TCGGC GACGG CCCCG TGCTG CTGCC CGACA ACCAC TACCT
606 GAGCC ATCAG TCCAA GCTGA GCAAA GACCC CAACG AGAAG CGCGA TCACA TGGTC
661 CTGAA GGAGA GGGTG ACCGC CGCCG GGATT ACACA TGACA TGGAC GAGCT GTACA
716 AGTAA
```

Original DNA sequence of mGreenLantern, specific location for shift to blue is marked in green.

A.2. Protocols

A.2.1. Patterning and Coating

Passivation A bioinert μ-slide VI 0.4 (ibidi) is positioned on top of a mask, which is patterned with squares with a side length of 25 µm and a lattice distance of 90 µm. Then 100 µL of 2 % low-melting agarose (Roth) are mixed with 100 µL PLPP (Alveole) with a concentration of 100 mM and stored in a 42 °C heat block. Calciumperoxid (Roth) is diluted to a 6 M concentration with RNase free water and 62.5 µL are mixed with the agar-PLPP mix. Each channel is then filled with 40 µL of patterning solution and illuminated with UV-light on highest settings for 8 rounds with each 1 min illumination and 5 s break to allow for cooling of the slide and patterning solution. The PLPP is activated by UV illumination and allows for removal of the polyol-based coating of the slide, revealing the underlying periodic square-grid pattern provided by the mask. All channels are then washed twice with 25 mL double distilled H2O (ddH2O). To remove waste and residual chemicals the channels are additional washed with 200 µL of 0.5 M HCl and again with 25 mL ddH2O. Consecutively all channels are dried with pressured air and then sterilised by UV illumination for 20 min.

Laminin Coating The patterning removes the passivation to reveal a lattice of squares, which can now be coated with cell adhesive substances. For the experiments performed in this thesis, Biolaminin 521 LN (Lamina) was used. After transfecting the cells the slides are coated by diluting 44 µL of laminin from a 100 ng/µL stock with 176 µL of PBS. Each channel of the slide is washed with 5 mL of ddH2O and 35 µL of laminin mix are added to the channels. After incubation at 37 °C for 1 h the channels are washed with 5 mL of PBS, filled with the respective medium and then further incubated until seeding of the cells.

A.2.2. Cell Culture

A HEK293T cell line, which stably express a blue-shifted variant of mGreenLantern in the ATP1A1 locus, was kindly provided by Prof. Dr. Gil Westmeyer from the Institute for Synthetic Biology, Helmholtz Zentrum München. These cells are cultured at 5% CO2 at 37 °C in DMEM medium (Gibco), supplemented with 10% fetal bovine serum. For passaging cells are being washed and treated with trypsin for 1 min.

A.2.3. mGL Transfection

For fluorescent reporter experiments, cells from culture were treated with accutase for 1 min, centrifuged at 800 r c f and resuspended in medium. Approximately 25 000 cells were seeded into a micro-patterned μ-slide and incubated for 30 min for adherence to the pattern. Cells were then washed with medium to remove debree and not adhered cells and again incubated for 30 min. After addition of the transfection mix cells were incubated for 45 min to allow for endocytosis. Subsequently medium was changed to phenol red free L-15 medium supplemented with 10% fetal bovine serum and transferred onto the microscope.

pDNA 37.5 µL of jetOPTIMUS buffer (polyplus) are mixed with 3.75 µL of mGL plasmid with a stock concentration of 100 ng/µL. 0.375 µL of jetOPTIMUS are added, mixed and incubated for 10 min at room temperature to allow for complex formation. Next the lipoplex mix is diluted with 80 µL of DMEM and 20 µL of this is added to each channel and mixed carefully by pipetting repeatedly from the back to the front.

mRNA Here, 0.9 µL of Lipofectamine MessengerMAX (LMRNA001, Thermo Fisher) are diluted in 30 µL of OptiMEM (Gibco) and incubated for 10 min. During this time 0.6 µL of mGL RNA are diluted from a stock concentration of 500 ng/µL in 30 µL OptiMEM. After sufficient incubation 30 µL of RNA mix are transferred to 30 µL of Lipofectamine MessengerMax mix and allowed to perform complexation into lipoplexes by incubation at room temperature for another 5 min. This, too, is then diluted with 60 µL OptiMEM and 20 µL are added per channel and mixed carefully.

A.2.4. Translation Inhibition Using Cycloheximide

The protocol for inhibiting translation was established previously and slightly adapted for the HEK cell line [29]. HEK-bsmGL cells are prime edited before start of the experiment to attain green fluorescing cells. Cells are then cultured for at least 3 passages to recover from the possible transfection and editing stress on the cells, before being used for the following experiments. These cells are seeded on a slide as explained previously in Subsection A.2.3. Before the second wash cycle, the settings for the microscope and the positions are adjusted. The second wash cycle is then performed with L15 supplemented with 10% fetal bovine serum and 15 µg/mL CHX(Sigma Aldrich).

A.2.5. Prime Editing Transfection

For prime editing experiments, 120 000 cells were seeded with 500 mL medium in 24 ibi-Treat μ-plates (ibidi). About 16 h later cells when cells reached 70 % confluency, they were transfected as explained below and incubated. After 4 h medium was changed to remove remaining transfection mix and dead cells. Cells were then incubated another hour to allow for sufficient recovery time and afterwards treated carefully with accutase for 1 min. The cell solution was centrifuged at 800 r c f for 3 min and the cells resuspended in medium. Approximately 25 000 cells are added into a micro-patterned μ-slide and incubated for adherence for 45 min. A first washing step is performed with medium to remove the non-adhered cells and any cell debree and the cells are incubated another 30 min. The medium is then changed to phenol red free L-15 medium supplemented with 10% fetal bovine serum for imaging.

pDNA For one well of transfected cells 50 µL of jetOPTIMUS buffer were mixed with 2.5 µL PE plasmid (Stock 0.1 µg/µL) and 2.5 µL pegRNA plasmid (Stock 0.1 µg/µL). After mixing 0.5 µL jetOPTIMUS were added and incubated for 10 min for complexation. 55 µL were then added to a well containing cells in standard medium.

mRNA For mRNA transfection 22.5 µL Lipofectamine MessengerMAX was diluted in 75 µL OptiMEM (Gibco) and incubated for 10 min to allow for building of liposomes. During this time 10.58 µL of PE RNA (Stock 500 ng/µL) were mixed with 1.18 µL of pegRNA (Stock 500 ng/µL) and diluted in 75 µL OptiMEM. Afterwards 75 µL of mRNA mix were added to 75 µL of Lipofectamine mix and incubated for another 5 min at room temperature to allow for the formation of lipoplexes. 50 µL were then added to cells in a well and carefully shaken for even distribution.

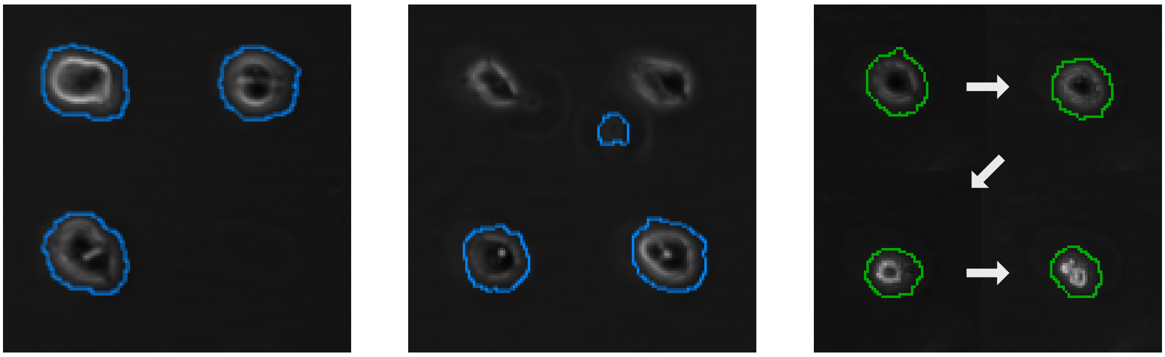
A.2.6. Time-Lapse Microscopy

Time-lapse measurements were conducted over a 30 to 48 h-period using a inverted Nikon Eclipse Ti-E microscope with a 10x objective. The samples were maintained in a heated chamber from Okolabs at 37 °C. Images were captured every 15 min consisting of BF and the needed fluorescence channels. BF images were illuminated with a LED 100 warm white (MHLED100W) and taken with a CMOS camera (PCO edge 4.2). Fluorescence - either mGL, mScarlet or Cy5 - was captured using filter sets for the corresponding wavelengths. Multiple positions per channel and slide were captured in this 15 min window to facilitate high-throughput. The time-lapse was set up and controlled using NIS-Elements Advanced Research software (Nikon) and the perfect focus system (PFS) was used to keep the sample in focus.

A.3. Image Processing with PyAMA

As explained in Section 3.2 fluorescence images are background corrected, cells are then tracked via the BF images and using PyAMA area and fluorescence curves are plotted. Cells are then deselected by the user for the following reasons (see Figure A.1):

- multiple cells on one spot
- empty squares
- as cell detected dirt
- dirt on cells
- cells dying during the measurement



A: Multiple cells

B: Dirt

C: A dying cell at multiple time points.

Figure A.1: Example pictures for exclusion parameters of cells

A.4. Data Evaluation Scripts

Subsequent data evaluation was performed with *R* or *Julia*. Here presented are code snippets of evaluation code.

A.4.1. Deterministic Models Coded in Julia

DNA-to-Protein Model in Julia

```julia
#Model in Catalyst
DNA_to_Protein_model = @reaction_network begin
    @parameters t0
    tc, DNA --> DNA + RNA
    tl, RNA --> RNA + Protein_unfold
    km, Protein_unfold --> Protein_matured

    delta_DNA, DNA --> 0
    delta_RNA, RNA --> 0
    beta, (Protein_unfold, Protein_matured) --> 0
end

#Initial Conditions
u0 = [:DNA => 0.0, :RNA => 0.0, :Protein_unfold => 0.0, :Protein_matured => 0.0]
```

RNA-to-Protein Model in Julia

```julia
#Model in Catalyst
RNA_to_Protein_model = @reaction_network begin
    @parameters t0
    tl, RNA --> RNA + Protein_unfold
    km, Protein_unfold --> Protein_matured
    delta, RNA --> 0
    beta, (Protein_unfold, Protein_matured) --> 0
end

#Initial Conditions
u0 = [:RNA => 0.0, :Protein_unfold => 0.0, :Protein_matured => 0.0]
```

Fitting Experimental Data Using PEtab

```
#Observables
    @unpack Protein_matured = DNA_to_Protein_model
    obs_Pm = PEtabObservable(Protein_matured, 1)
    observables = Dict("obs_Pm" => obs_Pm)
#Parameters
    par_tc = PEtabParameter(:tc, lb=1e-10, ub=2e2, scale=:lin)
    par_tl = PEtabParameter(:tl, lb=1e5, ub=1e10, scale=:lin)
    par_km = PEtabParameter(:km, lb = 1e-10, ub=1e3, scale=:lin)
    par_delta_DNA = PEtabParameter(:delta_DNA, lb=1e-10, ub=10.0, scale=:lin)
    par_delta_RNA = PEtabParameter(:delta_RNA, lb=1e-10, ub=10.0, scale=:lin)
    par_beta = PEtabParameter(:beta, lb=1e-10, ub=20.0, scale=:lin)
    par_t0 = PEtabParameter(:t0, lb= 0.0, ub=30.0, scale=:lin)
    params = [par_tc, par_tl, par_km, par_delta_DNA, par_delta_RNA, par_beta, par_t0]
#Measurements
    measurements = DataFrame(obs_id="obs_Pm", time=t_vector,
                    measurement=convert(Array{Float64,1}, measured_parameter)
#Events
    @unpack DNA = DNA_to_Protein_model
    eventt0 = PEtabEvent(:t0, DNA + 1, DNA)
    event = [eventt0]
#PEtabModel
    petab_model = PEtabModel(DNA_to_Protein_model, observables, measurements, params;
                        state_map=u0, events=event, verbose=false)
    petab_problem = PEtabODEProblem(petab_model, ode_solver = ODESolver(Rodas5P(), verbose=false),
                                verbose=false)
#Solution
    res_ms = calibrate_model_multistart(petab_problem, LBFGS(), runs, optim_folder_path,
                                    save_trace=true)
    fitted_sol = get_odesol(res_ms.xmin, petab_problem)
    residuals = petab_problem.compute_residuals(res_ms.xmin; as_array=true)
    chi2 = petab_problem.compute_chi2(res_ms.xmin)
```

A.4.2. Algorithm for t_0 in R

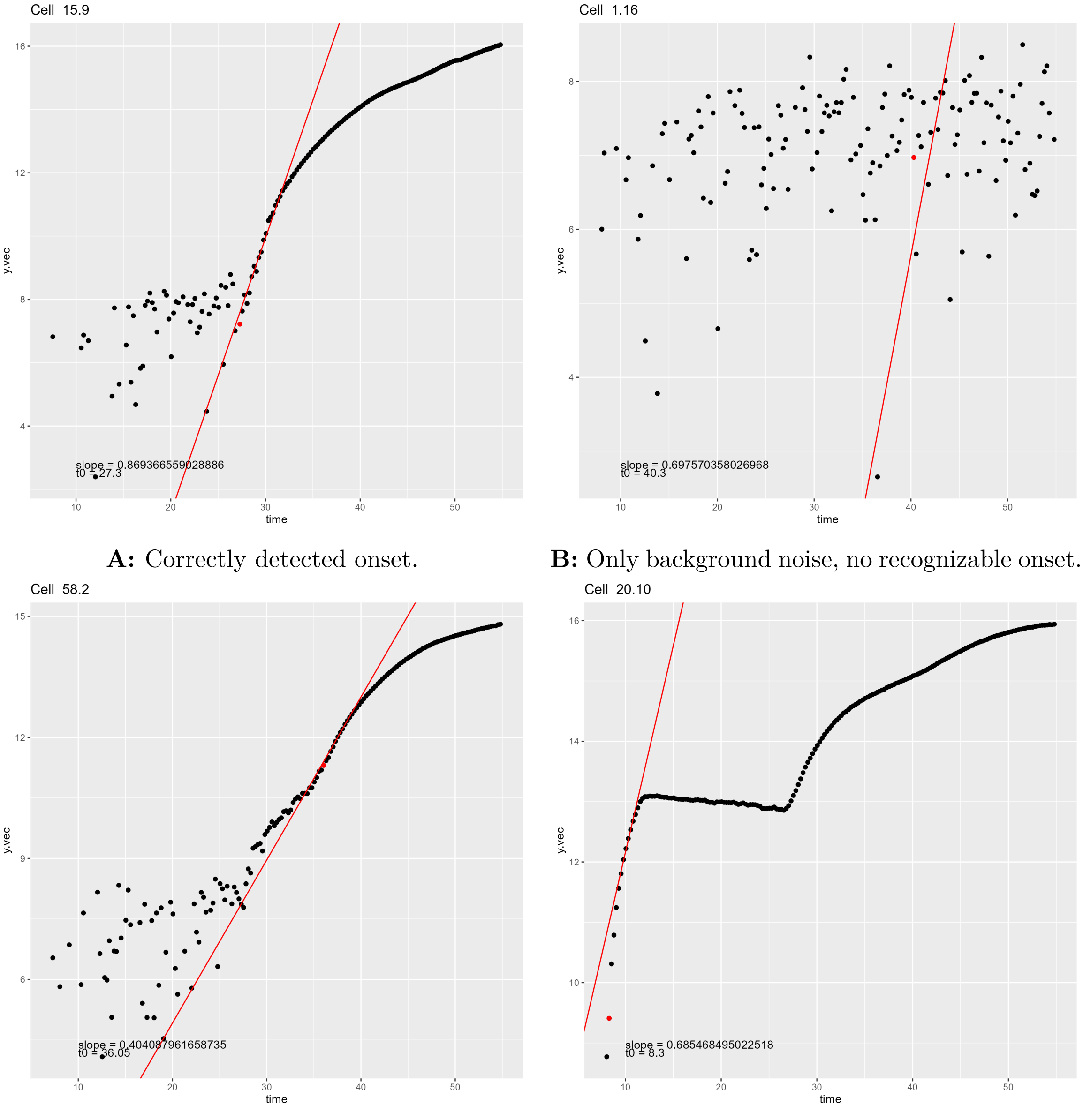
```
# Searches for t0 for one cell and plots the data with the fitted line
plot_t0_with_slope <- function(time, y.vec, slope.th, count.th, window.width, cellnr)
     {
  df <- data.frame(time = time, y = y.vec)
  df <- na.omit(df)
  time <- df$time
  y.vec <- df$y
  t0 <- NA
  count <- 0
  # sliding window
  for (i in 1:(length(y.vec) - window.width + 1)) {
    y_sub <- y.vec[i:(i + window.width - 1)]
    time_sub <- time[i:(i + window.width - 1)]
    fit <- lm(y_sub ~ time_sub)
    slope <- coef(fit)[2]
    #threshold check
    if (slope > slope.th) {
      count <- count + 1
      if (count == 2) {t0 <- time[i]}
    } else {
      count <- 0
      t0 <- NA
    }
    # plot t0 and fit
    if (count > count.th) {
      p <- ggplot() + geom_point(aes(x = time, y = y.vec)) +
        geom_abline(slope = slope, intercept = coef(fit)[1], color = "red") +
```

```r
        geom_text(aes(x = 10, y = min(y.vec), label = paste("t0 =", t0)), hjust = 0, vjust = 0)+
        geom_text(aes(x = 10, y = min(y.vec), label = paste("slope =", slope)), hjust = 0, vjust = -1)+
        labs(title = paste("Cell ", cellnr))+
        geom_point(aes(x = t0, y= y.vec[which(time == t0)]), color = "red")
      return(list(t0 = t0, plot = p))
    }
  }
  # when t0 is not found
  if (is.na(t0)) {
    p <- ggplot() + geom_point(aes(x = time, y = y.vec)) + labs(title = paste("Cell ", cellnr))
    return(list(t0 = t0, plot = p))
  }
}

# calls plot_t0_with_slope for all cells in the dataset and saves plots as *.png and t0 values in a *.xlsx in output_dir
plot_all_t0_with_slope_wouser <- function(time, df.y, cell_group, count.th, window.width, count.exp, v.names, output_dir) {
  df.on <- data.frame(Cell = character(), Time = numeric(), Group = character())
  if (!dir.exists(output_dir)) { dir.create(output_dir) }
  group <- NA
  for (i in 1:ncol(df.y)) {
    print(paste("Cell number: ", colnames(df.y)[i]))
    if (max(df.y[,i], na.rm=TRUE) < 7) {
      next
    }
    group <- cell_group$Group[which(cell_group$Cell==colnames(df.y)[i])]
    for (slope.th in seq(1.0, 0.1, by = -0.1)) {
      list <- plot_t0_with_slope(time, df.y[ ,i], slope.th, count.th, window.width, colnames(df.y)[i])
      if (is.na(list$t0) || is.null(list$t0)) { next } else {
        png_filename <- paste0(output_dir, "/", colnames(df.y)[i], ".png")
        ggsave(png_filename, list$plot)
        df.on <- rbind(df.on, data.frame(cellnr = colnames(df.y)[i], t0 = list$t0, group = group))
        print(list$t0)
        break
      }
    }
  }
  xlsx_filename <- paste0(output_dir, "/df_on.xlsx")
  openxlsx::write.xlsx(df.on, xlsx_filename)
  return(df.on)
}
```

Algorithm for determining t_0 written in R

Exclusion Parameters

The presented algorithm saves pictures for each cell line in an output directory. The user then has the possibility to check for wrongfully detected onsets and exclude them from the data set for the following criteria:



A: Correctly detected onset.

B: Only background noise, no recognizable onset.

C: Faulty detection of onset.

D: Two cells with different onsets in one spot.

Figure A.2: Example pictures of fluorescence signal for individual cells. Detected onset point is marked red, fit shows first fit from sliding window. B-D were excluded from the dataset for stated reasons.

B. Supplementary Figures and Tables

B.1. Additional Experimental Figures

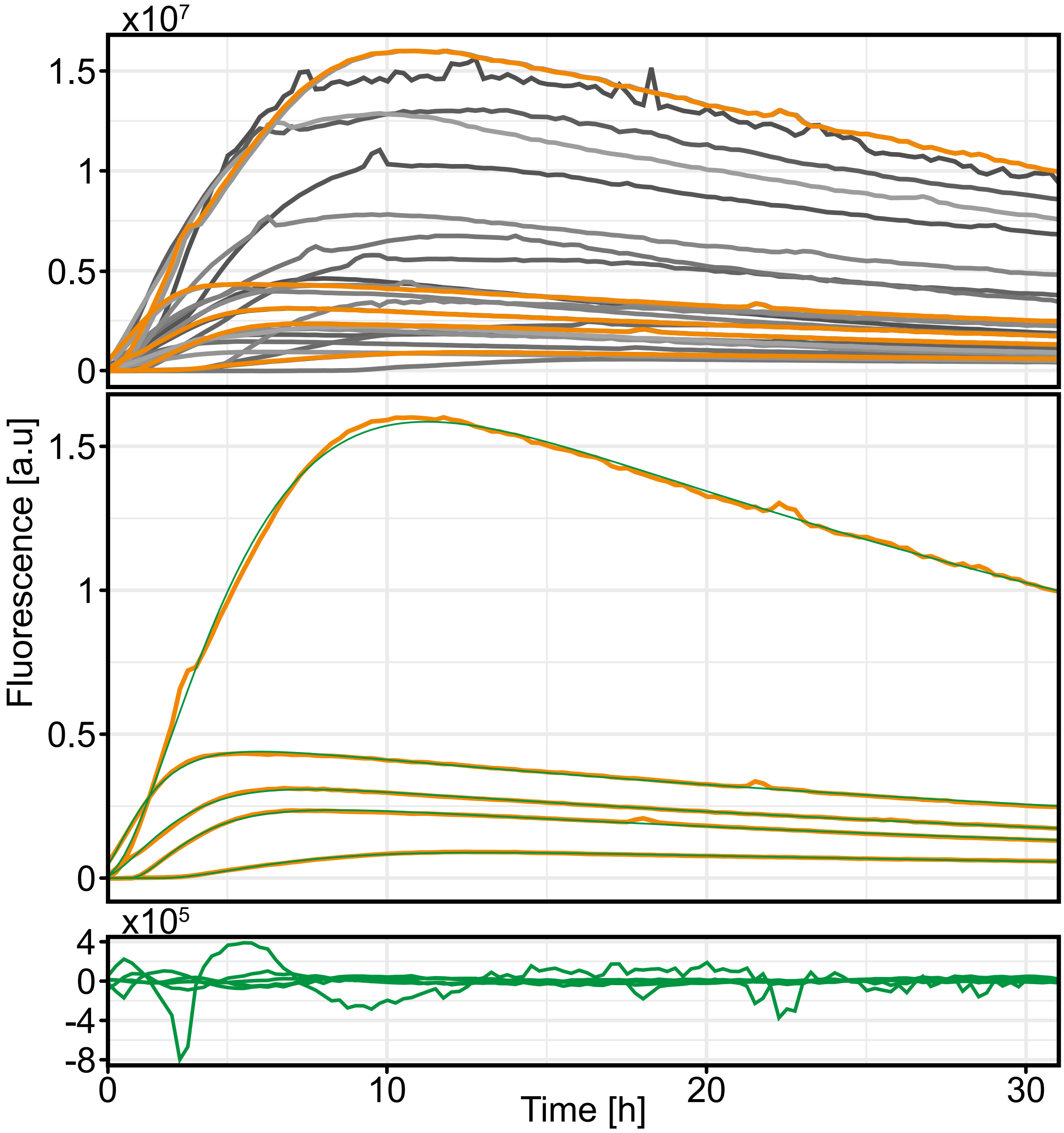


Figure B.1: Fit of single-cell fluorescence data from bs-mGL HEK293T cells transfected with mGL RNA. Top shows exemplary experimental data. 5 cells are highlighted. These 5 are shown with their corresponding green fits and residuals.

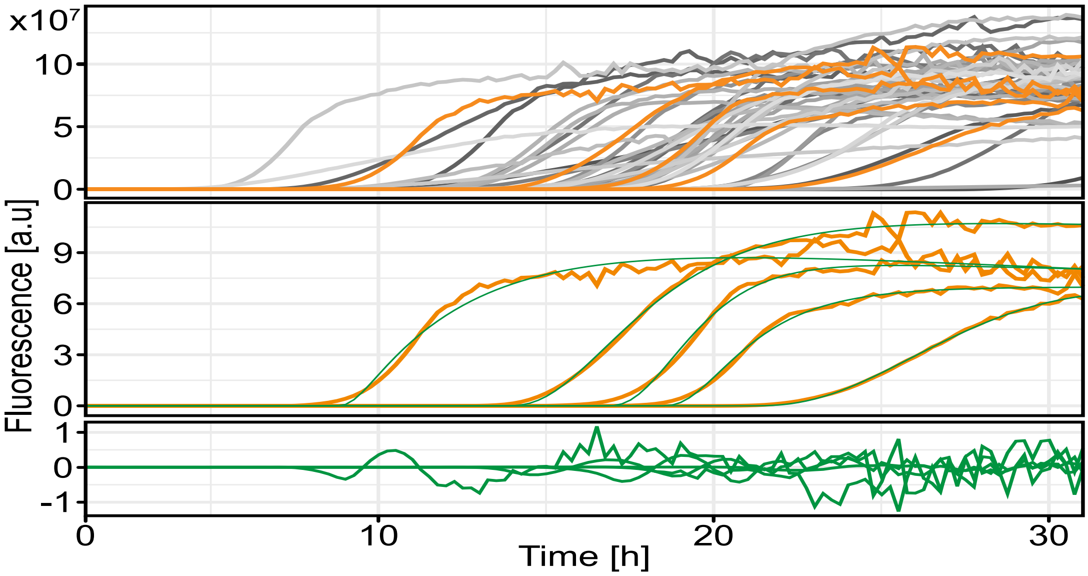
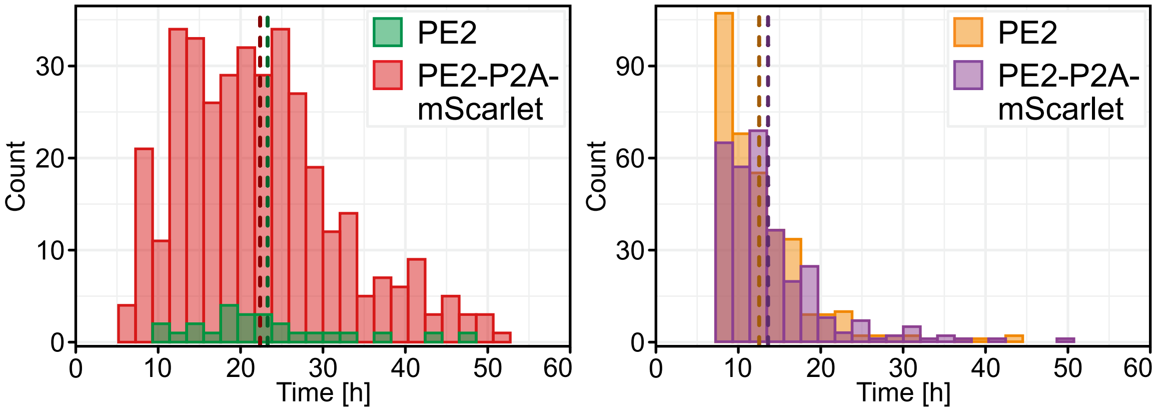


Figure B.2: Fit of single-cell fluorescence data from bs-mGL HEK293T cells transfected with mGL DNA. Top shows exemplary experimental data. 5 cells are highlighted. These 5 are shown with their corresponding green fits and residuals.



A: pDNA

B: mRNA

Figure B.3: Comparing onset distribution of mGL for the PE2 and PE2-P2A-mScarlet construct. The corresponding lines show the mean of each distribution.

B.2. SKI Combinator Calculus - Example Calculations

In Subsection 2.2.4 we showed how Church numerals are defined in SKI calculus. To better showcase the truth of our definitions, we here want to evaluate these expressions. We start with the successor function defined in the SKI calculus as follows:

$$S(S(KS)K)$$

Now using that the SKI calculus is left-associative and we know that any number is defined as $n \cdot function f$ applied on our value x (so nfx), we can apply it to three inputs and evaluate as seen on the left:

Successor function applied to three inputs:

$$
\begin{aligned}
&S(S(KS)K)nfx\\
&S(KS)Kf(nf)x\\
&KSf(Kf)(nf)x\\
&S(Kf)(nf)x\\
&Kfx(nfx)\\
&f(nfx)
\end{aligned}
$$

Successor function applied to Church numeral 2:

$$
\begin{aligned}
&S(S(KS)K)Ix\\
&S(KS)Kx(IX)\\
&KSxKx(Ix)\\
&SKx(Ix)\\
&KIxxIx\\
&IxIx\\
&xx
\end{aligned}
$$

The succ(nfx) is function f applied to nfx again. Knowing this, we can evaluate our definition of the Church numeral 2 on the right, by applying it to Ix, with I being the representation of 1 and x our value.

B.3. Additional Figures and Tables for Julia Implementation

Value $[h^{-1}]$	$t_{peak}[h]$
$\beta_{complex}$ for pDNA prime editor expression	
1e-5	14.5
5.18e-5	12.8
2.68e-4	11.4
1.39e-3	8.46
7.2e-3	6.11
3.73e-2	4.19
1.93e-1	2.51
1	1

(a) pDNA expression

Value $[h^{-1}]$	$t_{peak}[h]$
$\frac{pegRNA_0}{PE-RNA_0}$ ratio for mRNA prime editor expression	
100	0.48
316	1.52
1000	4.82
3160	11.6
10000	11.8

(b) mRNA expression

Table B.1: Values and t_{peak} from parameter variations as plotted in Figure 6.3.

Parameter	Description	Value	Units	Source/Ref
pDNA-mediated PE expression				
PE_DNA(0)	initial pDNA released	2.4		Estimated from experiments
tc_PE	transcription factor for PE	278	1/h	estimated from literature
deg_PE_DNA	degradation of PE plasmid	0.74	1/h	From mGL experiments
pDNA-mediated pegRNA expression				
peg_DNA(0)	initial pDNA released	6.8		Calculated from mixing ratio in experiment in relation to PE_DNA(0)
tc_peg	transcription factor for pegRNA	8685	1/h	estimated from literature
deg_peg_DNA	degradation of pegRNA plasmid	0.74	1/h	From mGL experiments
mRNA-mediated PE expression				
PE_mRNA(0)	initial mRNA released	10000		Estimated from experiments
k_tl	Translation rate	207.7	1/h	[28, 114, 152]
delta_RNA	RNA degradation rate	0.14	1/h	Endogenous RNA [59]
delta_PE	PE protein degradation rate	0.108	1/h	[153]
pegRNA lifetime				
pegRNA(0)	initial mRNA released	34000		Calculated from mixing ratio in experiment in relation to PE_mRNA(0)
delta_RNA	RNA degradation rate	0.14	1/h	Endogenous RNA [59]
Complex formation				
k_find	Complexation of pegRNA and Prime Editor	1	1/h	[154]
d_complex	PE-pegRNA complex degradation rate	0.044	1/h	[115, 116]

Table B.2: Parameter as used in Section 6.1

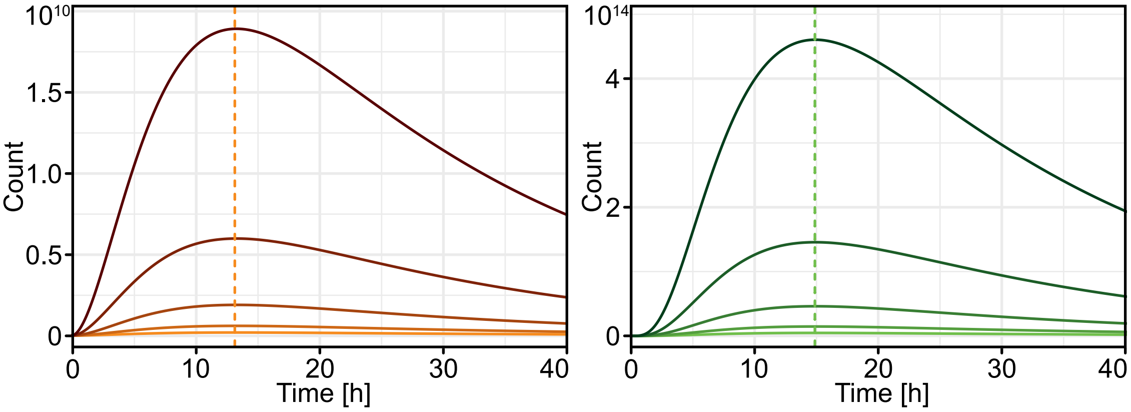


Figure B.4: Catalytical implementation of prime editing expression. Lines show concentration of complex for different ratios of PE_0:$pegRNA_0$. Left: mRNA. Right: pDNA

Parameter	Description	Value	Units	Source/Ref
pDNA-mediated diffusion from nucleus to cytosol				
ntc_PE	for produced PE RNA	10	1/h	Estimated from [155]
ntc_peg	for produced pegRNA	120	1/h	[155]
Transport from cytosol to nucleus				
ctn_PE	PE protein or complex	0.5	1/h	[156]

Table B.3: Additional parameters for Section 6.3 combined with Table B.2.

	$t_{peak}[h]$	
$ctn_{PE}[h^{-1}]$	pDNA	mRNA
0.01	10.9	22.8
0.215	7.47	8.58
4.64	4.15	1.05
100	4.08	0.08

(a)

$ntc_{PE}[h^{-1}]$	$t_{peak}[h]$
1e-3	24.6
2.51e-2	10.6
6.31e-1	6.64
1.58e1	5.58
3.98e2	5.41

(b) pDNA

$ntc_{peg}[h^{-1}]$	$t_{peak}[h]$
1e-4	4.17
6.31e-2	4.32
3.98e1	5.68
2.51e4	6.1
1.58e7	6.87

(c) pDNA

Table B.4: Values and t_{peak} from parameter variations for the nuclear barrier model as plotted in Figure 6.8.

List of Abbreviations

AA chain amino acid chain. 9, 26, 28, 32, 42, 46

BF brightfield. 19, 75

Cas CRISPR associated protein. 5, 10, 11, 12

CHX cycloheximide. 26, 28, 74

CRISPR Clustered Regularly Interspaced Short Palindromic Repeat. 5, 11, 64

ddH2O double distilled H2O. 73

DNA deoxyribonucleic acid. 5, 10, 11

DSB double strand break. 11

eGFP enhanced GFP. 8

GC garbage collector. 63, 64

GFP green fluorescent protein. 8

gRNA guide RNA. 11

HDR homology-directed repair. 10, 11

IVT in vitro transcription. 30, 71

LISCA Live Imaging on Single-Cell Arrays. 3, 8, 21, 46, 67

LNP lipid nanoparticle. 20, 21, 47

mGL mGreenLantern. 3, 8, 9, 25, 32, 74

mRNA messenger RNA. 21

NGS next-generation sequencing. 37, 47, 62, 66

NHEJ non-homologous end joining. 10, 11

NLS nuclear localization signal. 11, 56

ODE ordinary differential equation. 3, 25, 26, 30, 49, 50, 52, 57, 58, 68

ORF open reading frame. 7, 41

ORI origin of replication. 7

PAM protospacer adjacent motive. 11, 12, 61, 62

PBS primer binding site. 11, 12, 46, 49, 61, 68

PC phase contrast. 19, 20

pDNA plasmid DNA. xi, 2, 21

PE Prime Editor. 3, 10, 11, 33

pegRNA prime editing guide RNA. 11, 12

RNA ribonucleic acid. 5

RNP Ribonucleoprotein. 5, 68

RT reverse transcriptase. 10, 11, 12, 39, 46, 49, 62, 65

SSB single strand break. 11

ssDNA single-stranded DNA. 17, 66

UTR untranslated region. 2, 7, 30, 68

List of Figures

List of Tables

References

[1] J. Craig Venter. *Life at the Speed of Light: From the Double Helix to the Dawn of Digital Life*. Google-Books-ID: GWDD9yMb33cC. Penguin, Oct. 17, 2013. 239 pp.

[2] Donald E. Johnson. *Programming of Life*. Google-Books-ID: AO8wXVxh1qMC. Big Mac Publishers, 2010. 136 pp.

[3] Amand Lucas. "DNA: hardware and software of life". In: *Journal of Computational Electronics* 13.4 (Dec. 1, 2014), pp. 781–793. DOI: `10.1007/s10825-014-0570-3`.

[4] Bruce Alberts, Hopkin Karen, Johnson D Alexander, Morgan David, Raff Martin, Roberts Keith, and Walter Peter. *Essential Cell Biology: Fifth International Student Edition*. Google-Books-ID: u8WaDwAAQBAJ. W.W. Norton & Company, Nov. 19, 2018. 23 pp.

[5] Jing-Yuan Ma, Kai-Xuan Wang, Ming Ma, and Yu Zhang. "Magnetic Microsphere/Silica Nanoparticle Composite Structures for Switchable DNA Storage". In: *ACS Applied Nano Materials* 5.10 (Oct. 28, 2022). Publisher: American Chemical Society, pp. 15619–15628. DOI: `10.1021/acsanm.2c03686`.

[6] Roy J. Britten and Eric H. Davidson. "Gene Regulation for Higher Cells: A Theory". In: *Science* 165.3891 (July 25, 1969). Publisher: American Association for the Advancement of Science, pp. 349–357. DOI: `10.1126/science.165.3891.349`.

[7] Seth L. Shipman, Jeff Nivala, Jeffrey D. Macklis, and George M. Church. "CRISPR–Cas encoding of a digital movie into the genomes of a population of living bacteria". In: *Nature* 547.7663 (July 2017). Publisher: Nature Publishing Group, pp. 345–349. DOI: `10.1038/nature23017`.

[8] Tae Kyung Kim and James H. Eberwine. "Mammalian cell transfection: the present and the future". In: *Analytical and Bioanalytical Chemistry* 397.8 (2010), pp. 3173–3178. DOI: `10.1007/s00216-010-3821-6`.

[9] Yoshiyuki Hattori, Yuki Yoshiike, Maho Honda, Hiroaki Ohno, and Hiraku Onishi. "Evaluation of Small Interfering RNA Delivery into Cells by Reverse Transfection in Suspension with Cationic Liposomes". In: *Pharmacology &amp; Pharmacy* 08.5 (2017), pp. 129–139. DOI: `10.4236/pp.2017.85009`.

[10] David P. Bartel. "MicroRNAs: target recognition and regulatory functions". In: *Cell* 136.2 (Jan. 23, 2009), pp. 215–233. DOI: `10.1016/j.cell.2009.01.002`.

[11] Shuang Zhao, Lianyu Yu, Sha Yang, Xiaoqi Tang, Kai Chang, and Ming Chen. "Boolean logic gate based on DNA strand displacement for biosensing: current and emerging strategies". In: *Nanoscale Horizons* 6.4 (2021). Publisher: Royal Society of Chemistry, pp. 298–310. DOI: 10.1039/D0NH00587H.

[12] Friedrich C. Simmel, Bernard Yurke, and Hari R. Singh. "Principles and Applications of Nucleic Acid Strand Displacement Reactions". In: *Chemical Reviews* 119.10 (May 22, 2019). Publisher: American Chemical Society, pp. 6326–6369. DOI: 10.1021/acs.chemrev.8b00580.

[13] Tianhe Wang, Henning Hellmer, and Friedrich C Simmel. "Genetic switches based on nucleic acid strand displacement". In: *Current Opinion in Biotechnology* 79 (Feb. 1, 2023), p. 102867. DOI: 10.1016/j.copbio.2022.102867.

[14] Wakana Sato, Tomasz Zajkowski, Felix Moser, and Katarzyna P. Adamala. "Synthetic cells in biomedical applications". In: *WIREs Nanomedicine and Nanobiotechnology* 14.2 (2022), e1761. DOI: 10.1002/wnan.1761.

[15] Deboki Chakravarti, Jang Hwan Cho, Benjamin H. Weinberg, Nicole M. Wong, and Wilson W. Wong. "Synthetic biology approaches in cancer immunotherapy, genetic network engineering, and genome editing". In: *Integrative Biology* 8.4 (Apr. 18, 2016), pp. 504–517. DOI: 10.1039/c5ib00325c.

[16] Ahmad M. Khalil. "The genome editing revolution: review". In: *Journal of Genetic Engineering and Biotechnology* 18.1 (Dec. 1, 2020), p. 68. DOI: 10.1186/s43141-020-00078-y.

[17] Mario Bonillo, Julia Pfromm, and M. Dominik Fischer. "Challenges to Gene Editing Approaches in the Retina". In: *Klinische Monatsblätter für Augenheilkunde* 239 (Mar. 22, 2022). Publisher: Georg Thieme Verlag KG, pp. 275–283. DOI: 10.1055/a-1757-9810.

[18] Andrew V. Anzalone, Peyton B. Randolph, Jessie R. Davis, Alexander A. Sousa, Luke W. Koblan, Jonathan M. Levy, Peter J. Chen, Christopher Wilson, Gregory A. Newby, Aditya Raguram, and David R. Liu. "Search-and-replace genome editing without double-strand breaks or donor DNA". In: *Nature* 576.7785 (Dec. 2019). Publisher: Nature Publishing Group, pp. 149–157. DOI: 10.1038/s41586-019-1711-4.

[19] Junhong Choi, Wei Chen, Anna Minkina, Florence M. Chardon, Chase C. Suiter, Samuel G. Regalado, Silvia Domcke, Nobuhiko Hamazaki, Choli Lee, Beth Martin, Riza M. Daza, and Jay Shendure. "A time-resolved, multi-symbol molecular recorder via sequential genome editing". In: *Nature* 608.7921 (Aug. 2022). Publisher: Nature Publishing Group, pp. 98–107. DOI: 10.1038/s41586-022-04922-8.

[20] Zhangrao Huang and Gang Liu. "Current advancement in the application of prime editing". In: *Frontiers in Bioengineering and Biotechnology* 11 (Feb. 16, 2023). Publisher: Frontiers. DOI: 10.3389/fbioe.2023.1039315.

[21] Xiangyang Li, Xin Wang, Wenjun Sun, Shisheng Huang, Mingtian Zhong, Yuan Yao, Quanjiang Ji, and Xingxu Huang. "Enhancing prime editing efficiency by modified pegRNA with RNA G-quadruplexes". In: *Journal of Molecular Cell Biology* 14.4 (Apr. 1, 2022), mjac022. DOI: `10.1093/jmcb/mjac022`.

[22] Peter J. Chen, Jeffrey A. Hussmann, Jun Yan, Friederike Knipping, Purnima Ravisankar, Pin-Fang Chen, Cidi Chen, James W. Nelson, Gregory A. Newby, Mustafa Sahin, Mark J. Osborn, Jonathan S. Weissman, Britt Adamson, and David R. Liu. "Enhanced prime editing systems by manipulating cellular determinants of editing outcomes". In: *Cell* 184.22 (Oct. 28, 2021). Publisher: Elsevier, 5635–5652.e29. DOI: `10.1016/j.cell.2021.09.018`.

[23] I. F. Schene, I. P. Joore, J. H. L. Baijens, R. Stevelink, G. Kok, S. Shehata, E. F. Ilcken, E. C. M. Nieuwenhuis, D. P. Bolhuis, R. C. M. van Rees, S. A. Spelier, H. P. J. van der Doef, J. M. Beekman, R. H. J. Houwen, E. E. S. Nieuwenhuis, and S. A. Fuchs. "Mutation-specific reporter for optimization and enrichment of prime editing". In: *Nature Communications* 13.1 (Mar. 1, 2022). Publisher: Nature Publishing Group, p. 1028. DOI: `10.1038/s41467-022-28656-3`.

[24] Zhihan Zhao, Peng Shang, Prarthana Mohanraju, and Niels Geijsen. "Prime editing: advances and therapeutic applications". In: *Trends in Biotechnology* 41.8 (Aug. 1, 2023). Publisher: Elsevier, pp. 1000–1012. DOI: `10.1016/j.tibtech.2023.03.004`.

[25] William Andrew Simon, Yasir Mahmood Qureshi, Marco Rios, Alexandre Levisse, Marina Zapater, and David Atienza. "BLADE: An in-cache computing architecture for edge devices". In: *IEEE Transactions on Computers* 69.9 (2020). Publisher: IEEE, pp. 1349–1363.

[26] Yongchun Pan, Jingjing Yang, Xiaowei Luan, Xinli Liu, Xueqing Li, Jian Yang, Ting Huang, Lu Sun, Yuzhen Wang, Youhui Lin, and Yujun Song. "Near-infrared upconversion–activated CRISPR-Cas9 system: A remote-controlled gene editing platform". In: *Science Advances* 5.4 (Apr. 3, 2019). Publisher: American Association for the Advancement of Science, eaav7199. DOI: `10.1126/sciadv.aav7199`.

[27] Tanja Rothgangl, Melissa K. Dennis, Paulo J. C. Lin, Rurika Oka, Dominik Witzigmann, Lukas Villiger, Weihong Qi, Martina Hruzova, Lucas Kissling, Daniela Lenggenhager, Costanza Borrelli, Sabina Egli, Nina Frey, Noëlle Bakker, John A. Walker, Anastasia P. Kadina, Denis V. Victorov, Martin Pacesa, Susanne Kreutzer, Zacharias Kontarakis, Andreas Moor, Martin Jinek, Drew Weissman, Markus Stoffel, Ruben van Boxtel, Kevin Holden, Norbert Pardi, Beat Thöny, Johannes Häberle, Ying K. Tam, Sean C. Semple, and Gerald Schwank. "In vivo adenine base editing of PCSK9 in macaques reduces LDL cholesterol levels". In: *Nature Biotechnology* 39.8 (Aug. 2021). Publisher: Nature Publishing Group, pp. 949–957. DOI: `10.1038/s41587-021-00933-4`.

[28] Rafał Krzysztoń, Daniel Woschée, Anita Reiser, Gerlinde Schwake, Helmut H. Strey, and Joachim O. Rädler. "Single-cell kinetics of siRNA-mediated mRNA degradation". In: *Nanomedicine: Nanotechnology, Biology and Medicine* 21 (Oct. 1, 2019), p. 102077. DOI: 10.1016/j.nano.2019.102077.

[29] Anita Reiser. "Single-cell time courses of mRNA transport and translation kinetics". PhD thesis. Ludwig-Maximilians-Universität München, July 18, 2019.

[30] Enrique Balleza, J. Mark Kim, and Philippe Cluzel. "Systematic characterization of maturation time of fluorescent proteins in living cells". In: *Nature Methods* 15.1 (Jan. 2018). Publisher: Nature Publishing Group, pp. 47–51. DOI: 10.1038/nmeth.4509.

[31] Michał Komorowski, Bärbel Finkenstädt, and David Rand. "Using a Single Fluorescent Reporter Gene to Infer Half-Life of Extrinsic Noise and Other Parameters of Gene Expression". In: *Biophysical Journal* 98.12 (June 16, 2010). Publisher: Elsevier, pp. 2759–2769. DOI: 10.1016/j.bpj.2010.03.032.

[32] Lisa Muniz, Estelle Nicolas, and Didier Trouche. "RNA polymerase II speed: a key player in controlling and adapting transcriptome composition". In: *The EMBO Journal* 40.15 (Aug. 2, 2021). Publisher: John Wiley & Sons, Ltd, e105740. DOI: 10.15252/embj.2020105740.

[33] Daniel Tarrant and Tobias von der Haar. "Synonymous codons, ribosome speed, and eukaryotic gene expression regulation". In: *Cellular and Molecular Life Sciences* 71.21 (Nov. 1, 2014), pp. 4195–4206. DOI: 10.1007/s00018-014-1684-2.

[34] Nikola Mićić and Bogdana Mićić. "God is Dead, Long Live DNA". In: *AGRO - KNOWLEDGE JOURNAL* 18.2 (Dec. 18, 2017). Number: 2, pp. 143–146. DOI: 10.7251/AGREN1702143M.

[35] Francis Crick. "Central Dogma of Molecular Biology". In: *Nature* 227.5258 (Aug. 1970). Publisher: Nature Publishing Group, pp. 561–563. DOI: 10.1038/227561a0.

[36] Sethuramasundaram Pitchiaya and Yamuna Krishnan. "First blueprint, now bricks: DNA as construction material on the nanoscale". In: *Chemical Society Reviews* 35.11 (Oct. 23, 2006). Publisher: The Royal Society of Chemistry, pp. 1111–1121. DOI: 10.1039/B602886C.

[37] Yoshizumi Ishino, Mart Krupovic, and Patrick Forterre. "History of CRISPR-Cas from Encounter with a Mysterious Repeated Sequence to Genome Editing Technology". In: *Journal of Bacteriology* 200.7 (Mar. 12, 2018), e00580–17. DOI: 10.1128/JB.00580-17.

[38] Kira S. Makarova, Nick V. Grishin, Svetlana A. Shabalina, Yuri I. Wolf, and Eugene V. Koonin. “A putative RNA-interference-based immune system in prokaryotes: computational analysis of the predicted enzymatic machinery, functional analogies with eukaryotic RNAi, and hypothetical mechanisms of action”. In: *Biology Direct* 1.1 (Mar. 16, 2006), p. 7. DOI: `10.1186/1745-6150-1-7`.

[39] Rodolphe Barrangou, Christophe Fremaux, Hélène Deveau, Melissa Richards, Patrick Boyaval, Sylvain Moineau, Dennis A Romero, and Philippe Horvath. “CRISPR provides acquired resistance against viruses in prokaryotes”. In: *Science* 315.5819 (2007). Publisher: American Association for the Advancement of Science, pp. 1709–1712.

[40] Ting Chen, Hongyu L. He, and George M. Church. “Modeling gene expression with differential equations”. In: *Biocomputing ’99.* 1999, pp. 29–40.

[41] A. Reiser, D. Woschée, N. Mehrotra, R. Krzysztoń, H. H. Strey, and J. O. Rädler. “Correlation of mRNA delivery timing and protein expression in lipid-based transfection”. In: *Integrative Biology: Quantitative Biosciences from Nano to Macro* 11.9 (Dec. 31, 2019), pp. 362–371. DOI: `10.1093/intbio/zyz030`.

[42] Fabian Fröhlich, Anita Reiser, Laura Fink, Daniel Woschée, Thomas Ligon, Fabian Joachim Theis, Joachim Oskar Rädler, and Jan Hasenauer. “Multi-experiment nonlinear mixed effect modeling of single-cell translation kinetics after transfection”. In: *npj Systems Biology and Applications* 4.1 (Dec. 10, 2018). Publisher: Nature Publishing Group, pp. 1–12. DOI: `10.1038/s41540-018-0079-7`.

[43] Thomas S. Ligon, Carolin Leonhardt, and Joachim O. Rädler. “Multi-Level Kinetic Model of mRNA Delivery via Transfection of Lipoplexes”. In: *PLOS ONE* 9.9 (Sept. 19, 2014). Publisher: Public Library of Science, e107148. DOI: `10.1371/journal.pone.0107148`.

[44] Anil Panigrahi and Bert W. O’Malley. “Mechanisms of enhancer action: the known and the unknown”. In: *Genome Biology* 22.1 (Apr. 15, 2021), p. 108. DOI: `10.1186/s13059-021-02322-1`.

[45] David M. Suter, Nacho Molina, David Gatfield, Kim Schneider, Ueli Schibler, and Felix Naef. “Mammalian Genes Are Transcribed with Widely Different Bursting Kinetics”. In: *Science* 332.6028 (Apr. 22, 2011). Publisher: American Association for the Advancement of Science, pp. 472–474. DOI: `10.1126/science.1198817`.

[46] Dongjun Kim, Jeong Do Kim, Kwanghee Baek, Yeup Yoon, and Jaeseung Yoon. “Improved Mammalian Expression Systems by Manipulating Transcriptional Termination Regions”. In: *Biotechnology Progress* 19.5 (2003), pp. 1620–1622. DOI: `10.1021/bp0341186`.

[47] Songkhla Chulakasian, Tien-Jye Chang, Ching-Hsiu Tsai, Min-Liang Wong, and Wei-Li Hsu. "Translational enhancing activity in 5 UTR of peste des petits ruminants virus fusion gene". In: *The FEBS Journal* 280.5 (2013), pp. 1237–1248. DOI: 10.1111/febs.12115.

[48] Jose Luis Ortega, Olivia L. Wilson, and Champa Sengupta-Gopalan. "The 5' untranslated region of the soybean cytosolic glutamine synthetase beta1 gene contains prokaryotic translation initiation signals and acts as a translational enhancer in plants". In: *Molecular Genetics and Genomics* 287.11 (Dec. 1, 2012), pp. 881–893. DOI: 10.1007/s00438-012-0724-6.

[49] Michael J. Hansen, Li-How Chen, Mariena L. S. Fejzo, and Joel G. Beiasco. "The ompA 5 untranslated region impedes a major pathway for mRNA degradation in Escherichia coli". In: *Molecular Microbiology* 12.5 (1994), pp. 707–716. DOI: 10.1111/j.1365-2958.1994.tb01058.x.

[50] Marco Mangone, Arun Prasad Manoharan, Danielle Thierry-Mieg, Jean Thierry-Mieg, Ting Han, Sebastian D. Mackowiak, Emily Mis, Charles Zegar, Michelle R. Gutwein, Vishal Khivansara, Oliver Attie, Kevin Chen, Kourosh Salehi-Ashtiani, Marc Vidal, Timothy T. Harkins, Pascal Bouffard, Yutaka Suzuki, Sumio Sugano, Yuji Kohara, Nikolaus Rajewsky, Fabio Piano, Kristin C. Gunsalus, and John K. Kim. "The Landscape of C. elegans 3'UTRs". In: *Science* 329.5990 (July 23, 2010). Publisher: American Association for the Advancement of Science, pp. 432–435. DOI: 10.1126/science.1191244.

[51] Mehrije Ferizi, Carolin Leonhardt, Christian Meggle, Manish K. Aneja, Carsten Rudolph, Christian Plank, and Joachim O. Rädler. "Stability analysis of chemically modified mRNA using micropattern-based single-cell arrays". In: *Lab on a Chip* 15.17 (2015). Publisher: Royal Society of Chemistry, pp. 3561–3571. DOI: 10.1039/C5LC00749F.

[52] Alexandra E. Grier, Stephen Burleigh, Jaya Sahni, Courtnee A. Clough, Victoire Cardot, Dongwook C. Choe, Michelle C. Krutein, David J. Rawlings, Michael C. Jensen, Andrew M. Scharenberg, and Kyle Jacoby. "pEVL: A Linear Plasmid for Generating mRNA IVT Templates With Extended Encoded Poly(A) Sequences". In: *Molecular Therapy. Nucleic Acids* 5.4 (Apr. 19, 2016), e306. DOI: 10.1038/mtna.2016.21.

[53] Osamu Shimomura. "Discovery of green fluorescent protein (GFP)(Nobel Lecture)". In: *Angewandte Chemie International Edition* 48.31 (2009). Publisher: WILEY-VCH Verlag Weinheim, pp. 5590–5602.

[54] Marc Zimmer. "Green Fluorescent Protein (GFP): Applications, Structure, and Related Photophysical Behavior". In: *Chemical Reviews* 102.3 (Mar. 1, 2002). Publisher: American Chemical Society, pp. 759–782. DOI: 10.1021/cr010142r.

[55] Fan Yang, Larry G. Moss, and George N. Phillips. “The molecular structure of green fluorescent protein”. In: *Nature Biotechnology* 14.10 (Oct. 1996). Publisher: Nature Publishing Group, pp. 1246–1251. DOI: 10.1038/nbt1096-1246.

[56] Theodorus W. J. Gadella, Laura van Weeren, Jente Stouthamer, Mark A. Hink, Anouk H. G. Wolters, Ben N. G. Giepmans, Sylvain Aumonier, Jérôme Dupuy, and Antoine Royant. “mScarlet3: a brilliant and fast-maturing red fluorescent protein”. In: *Nature Methods* 20.4 (Apr. 2023). Publisher: Nature Publishing Group, pp. 541–545. DOI: 10.1038/s41592-023-01809-y.

[57] Benjamin C. Campbell, Maria G. Paez-Segala, Loren L. Looger, Gregory A. Petsko, and Ce Feng Liu. “Chemically stable fluorescent proteins for advanced microscopy”. In: *Nature Methods* 19.12 (Dec. 2022). Publisher: Nature Publishing Group, pp. 1612–1621. DOI: 10.1038/s41592-022-01660-7.

[58] Benjamin C. Campbell, Elisa M. Nabel, Mitchell H. Murdock, Cristina Lao-Peregrin, Pantelis Tsoulfas, Murray G. Blackmore, Francis S. Lee, Conor Liston, Hirofumi Morishita, and Gregory A. Petsko. “mGreenLantern: a bright monomeric fluorescent protein with rapid expression and cell filling properties for neuronal imaging”. In: *Proceedings of the National Academy of Sciences* 117.48 (Dec. 2020). Publisher: Proceedings of the National Academy of Sciences, pp. 30710–30721. DOI: 10.1073/pnas.2000942117.

[59] Carolin Leonhardt, Gerlinde Schwake, Tobias R. Stögbauer, Susanne Rappl, Jan-Timm Kuhr, Thomas S. Ligon, and Joachim O. Rädler. “Single-cell mRNA transfection studies: Delivery, kinetics and statistics by numbers”. In: *Nanomedicine: Nanotechnology, Biology and Medicine* 10.4 (May 1, 2014), pp. 679–688. DOI: 10.1016/j.nano.2013.11.008.

[60] Torkel E. Loman, Yingbo Ma, Vasily Ilin, Shashi Gowda, Niklas Korsbo, Nikhil Yewale, Chris Rackauckas, and Samuel A. Isaacson. “Catalyst: Fast and flexible modeling of reaction networks”. In: *PLOS Computational Biology* 19.10 (Oct. 18, 2023). Publisher: Public Library of Science, e1011530. DOI: 10.1371/journal.pcbi.1011530.

[61] Claire Vallance. “Rate laws: relating the reaction rate to reactant concentrations”. In: *An Introduction to Chemical Kinetics*. Morgan & Claypool Publishers, Sept. 1, 2017, pp. 2–1 –2–5.

[62] Athel Cornish-Bowden. “Fundamentals of Enzyme Kinetics”. In: *Wiley.com*. 4th ed. Vol. 1. 1 vols. Wiley, pp. 1–24.

[63] Alexander Bolotin, Benoit Quinquis, Alexei Sorokin, and S Dusko Ehrlich. “Clustered regularly interspaced short palindrome repeats (CRISPRs) have spacers of extrachromosomal origin”. In: *Microbiology* 151.8 (2005). Publisher: Microbiology Society, pp. 2551–2561.

[64] Le Cong, F Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D Hsu, Xuebing Wu, Wenyan Jiang, Luciano A Marraffini, et al. "Multiplex genome engineering using CRISPR/Cas systems". In: *Science* 339.6121 (2013). Publisher: American Association for the Advancement of Science, pp. 819–823.

[65] Peter J. Chen and David R. Liu. "Prime editing for precise and highly versatile genome manipulation". In: *Nature Reviews Genetics* 24.3 (Mar. 2023). Publisher: Nature Publishing Group, pp. 161–177. DOI: 10.1038/s41576-022-00541-1.

[66] B. J. Thompson, M. N. Camien, and R. C. Warner. "Kinetics of branch migration in double-stranded DNA". In: *Proceedings of the National Academy of Sciences of the United States of America* 73.7 (July 1976), pp. 2299–2303. DOI: 10.1073/pnas.73.7.2299.

[67] Francesca G. Smith, John P. Goertz, Molly M. Stevens, and Thomas E. Ouldridge. *Strong sequence dependence in RNA/DNA hybrid strand displacement kinetics*. Pages: 2023.11.14.567030 Section: New Results. Nov. 14, 2023. DOI: 10.1101/2023.11.14.567030.

[68] Gordon E Moore. "Cramming More Components onto Integrated Circuits". In: *PROCEEDINGS OF THE IEEE* 86.1 (1998).

[69] Tao Li, Jie Hou, Jinli Yan, Rulin Liu, Hui Yang, and Zhigang Sun. "Chiplet Heterogeneous Integration Technology—Status and Challenges". In: *Electronics* 9.4 (Apr. 2020). Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 670. DOI: 10.3390/electronics9040670.

[70] Banani Chakraborty, Natasha Jonoska, and Nadrian C. Seeman. "A programmable transducer self-assembled from DNA". In: *Chemical Science* 3.1 (Nov. 28, 2011). Publisher: The Royal Society of Chemistry, pp. 168–176. DOI: 10.1039/C1SC00523E.

[71] Marco Bellia and M. Eugenia Occhiuto. "DNA Tiles, Wang Tiles and Combinators". In: *Fundamenta Informaticae* 133.2 (2014). Publisher: IOS Press, pp. 105–121. DOI: 10.3233/fi-2014-1065.

[72] Alan M. Turing. "Computing Machinery and Intelligence". In: *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. by Robert Epstein, Gary Roberts, and Grace Beber. Dordrecht: Springer Netherlands, 2009, pp. 23–65. DOI: 10.1007/978-1-4020-6710-5_3.

[73] Rebecca Weber. *Computability Theory*. Google-Books-ID: YaECAQAAQBAJ. American Mathematical Soc., 2012. 218 pp.

[74] Maribel Fernández. "Introduction". In: *Models of Computation: An Introduction to Computability Theory*. Ed. by Maribel Fernández. London: Springer, 2009, pp. 1–7. DOI: 10.1007/978-1-84882-434-8_1.

[75] Maribel Fernández. “Automata and Turing Machines”. In: *Models of Computation: An Introduction to Computability Theory*. Ed. by Maribel Fernández. London: Springer, 2009, pp. 11–32. DOI: `10.1007/978-1-84882-434-8_2`.

[76] Harold S. Stone. *Introduction to Computer Organization and Data Structures*. Google-Books-ID: TfsmAAAAMAAJ. McGraw-Hill, 1972. 346 pp.

[77] Paula Quinon. “Implicit and Explicit Examples of the Phenomenon of Deviant Encodings”. In: *Studies in Logic, Grammar and Rhetoric* 63.1 (Nov. 4, 2020), pp. 53–67. DOI: `10.2478/slgr-2020-0027`.

[78] Maribel Fernández. “The Lambda Calculus”. In: *Models of Computation: An Introduction to Computability Theory*. Ed. by Maribel Fernández. London: Springer, 2009, pp. 33–53. DOI: `10.1007/978-1-84882-434-8_3`.

[79] J. Roger Hindley and Jonathan P. Seldin. *Lambda-Calculus and Combinators: An Introduction*. Google-Books-ID: 9fhujocrM7wC. Cambridge University Press, July 24, 2008. 346 pp.

[80] Ralf Hinze. “THEORETICAL PEARL Church numerals, twice!” In: *Journal of Functional Programming* 15.1 (Jan. 2005), pp. 1–13. DOI: `10.1017/S0956796804005313`.

[81] Isamu Furuya and Takuya Kida. “Compaction of Church Numerals”. In: *Algorithms* 12.8 (Aug. 2019). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 159. DOI: `10.3390/a12080159`.

[82] Moses Schönfinkel. “On the building blocks of mathematical logic”. In: *From Frege to Gödel* (1967). Publisher: Harvard University Press Cambridge MA, pp. 355–366.

[83] Mike Stay and Lucius Gregory Meredith. *Logic as a distributive law*. Oct. 16, 2016. DOI: `10.48550/arXiv.1610.02247`. arXiv: `1610.02247[cs]`.

[84] Lulu Qian and Erik Winfree. “Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades”. In: *Science* 332.6034 (June 3, 2011). Publisher: American Association for the Advancement of Science, pp. 1196–1201. DOI: `10.1126/science.1200520`.

[85] Ehud Shapiro and Yaakov Benenson. “Bringing Dna Computers to Life”. In: *Scientific American* 294.5 (2006). Publisher: Scientific American, a division of Nature America, Inc., pp. 44–51.

[86] Piro Siuti, John Yazbek, and Timothy K. Lu. “Synthetic circuits integrating logic and memory in living cells”. In: *Nature Biotechnology* 31.5 (May 2013). Publisher: Nature Publishing Group, pp. 448–452. DOI: `10.1038/nbt.2510`.

[87] Kurt Thorn. “A quick guide to light microscopy in cell biology”. In: *Molecular Biology of the Cell* 27.2 (Jan. 15, 2016). Publisher: American Society for Cell Biology (mboc), pp. 219–222. DOI: `10.1091/mbc.e15-02-0088`.

[88] Egelberg. *Working principle of phase contrast microscopy.* Wikimedia Commons. Nov. 15, 2012. URL: `https://commons.wikimedia.org/wiki/File:Working_principle_of_phase_contrast_microscopy.gif` (visited on 08/07/2024).

[89] P. J. Evennett and C. Hammond. "MICROSCOPY — Overview". In: *Encyclopedia of Analytical Science (Second Edition).* Ed. by Paul Worsfold, Alan Townshend, and Colin Poole. Oxford: Elsevier, Jan. 1, 2005, pp. 32–41. DOI: `10.1016/B0-12-369397-7/00376-9`.

[90] Javier Ramón-Azcón and Artur Rydosz. "Fundamentals". In: *Human Organs-on-a-Chip: Novel Organ-on-a-Chip Techniques in Medicine.* Google-Books-ID: JK0NEQAAQBAJ. Elsevier, Nov. 15, 2023, pp. 105–135.

[91] Gert-Jan Kremers, Sarah G. Gilbert, Paula J. Cranfill, Michael W. Davidson, and David W. Piston. "Fluorescent proteins at a glance". In: *Journal of Cell Science* 124.2 (Jan. 15, 2011), pp. 157–160. DOI: `10.1242/jcs.072744`.

[92] Jörg Wiedenmann, Franz Oswald, and Gerd Ulrich Nienhaus. "Fluorescent proteins for live cell imaging: Opportunities, limitations, and challenges". In: *IUBMB Life* 61.11 (2009), pp. 1029–1042. DOI: `10.1002/iub.256`.

[93] Jonathan W. Young, James C. W. Locke, Alphan Altinok, Nitzan Rosenfeld, Tigran Bacarian, Peter S. Swain, Eric Mjolsness, and Michael B. Elowitz. "Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy". In: *Nature Protocols* 7.1 (Jan. 2012). Publisher: Nature Publishing Group, pp. 80–88. DOI: `10.1038/nprot.2011.432`.

[94] Dale Muzzey and Alexander van Oudenaarden. "Quantitative Time-Lapse Fluorescence Microscopy in Single Cells". In: *Annual Review of Cell and Developmental Biology* 25 (Volume 25, 2009 Nov. 10, 2009). Publisher: Annual Reviews, pp. 301–327. DOI: `10.1146/annurev.cellbio.042308.113408`.

[95] Richard N. Day and Fred Schaufele. "Fluorescent protein tools for studying protein dynamics in living cells: a review". In: *Journal of Biomedical Optics* 13.3 (May 2008). Publisher: SPIE, p. 031202. DOI: `10.1117/1.2939093`.

[96] Lauren R. Polstein, Mark Juhas, Gabi Hanna, Nenad Bursac, and Charles A. Gersbach. "An Engineered Optogenetic Switch for Spatiotemporal Control of Gene Expression, Cell Differentiation, and Tissue Morphogenesis". In: *ACS Synthetic Biology* 6.11 (Nov. 17, 2017). Publisher: American Chemical Society, pp. 2003–2013. DOI: `10.1021/acssynbio.7b00147`.

[97] Sae Shimizu-Sato, Enamul Huq, James M. Tepperman, and Peter H. Quail. "A light-switchable gene promoter system". In: *Nature Biotechnology* 20.10 (Oct. 2002). Publisher: Nature Publishing Group, pp. 1041–1044. DOI: `10.1038/nbt734`.

[98] Agnieszka Fus-Kujawa, Pawel Prus, Karolina Bajdak-Rusinek, Paulina Teper, Katarzyna Gawron, Agnieszka Kowalczuk, and Aleksander L. Sieron. “An Overview of Methods and Tools for Transfection of Eukaryotic Cells in vitro”. In: *Frontiers in Bioengineering and Biotechnology* 9 (July 20, 2021). Publisher: Frontiers. DOI: 10.3389/fbioe.2021.701031.

[99] Christoph Schreiber, Felix J. Segerer, Ernst Wagner, Andreas Roidl, and Joachim O. Rädler. “Ring-Shaped Microlanes and Chemical Barriers as a Platform for Probing Single-Cell Migration”. In: *Scientific Reports* 6.1 (May 31, 2016). Publisher: Nature Publishing Group, p. 26858. DOI: 10.1038/srep26858.

[100] Elisavet I. Chatzopoulou, Claudia C. Roskopf, Farzad Sekhavati, Todd A. Braciak, Nadja C. Fenn, Karl-Peter Hopfner, Fuat S. Oduncu, Georg H. Fey, and Joachim O. Rädler. “Chip-based platform for dynamic analysis of NK cell cytolysis mediated by a triplebody”. In: *Analyst* 141.7 (2016). Publisher: Royal Society of Chemistry, pp. 2284–2295. DOI: 10.1039/C5AN02585K.

[101] Peter J. F. Röttgermann, Kenneth A. Dawson, and Joachim O. Rädler. “Time-Resolved Study of Nanoparticle Induced Apoptosis Using Microfabricated Single Cell Arrays”. In: *Microarrays* 5.2 (June 2016). DOI: 10.3390/microarrays5020008.

[102] Johannes C. J. Heyn, Joachim O. Rädler, and Martin Falcke. “Mesenchymal cell migration on one-dimensional micropatterns”. In: *Frontiers in Cell and Developmental Biology* 12 (Apr. 16, 2024). Publisher: Frontiers. DOI: 10.3389/fcell.2024.1352279.

[103] Manuel Théry. “Micropatterning as a tool to decipher cell morphogenesis and functions”. In: *Journal of Cell Science* 123.24 (Dec. 15, 2010), pp. 4201–4213. DOI: 10.1242/jcs.075150.

[104] Anita Reiser, Matthias Lawrence Zorn, Alexandra Murschhauser, and Joachim Oskar Rädler. “Single Cell Microarrays Fabricated by Microscale Plasma-Initiated Protein Patterning (μPIPP)”. In: *Cell-Based Microarrays: Methods and Protocols*. Ed. by Peter Ertl and Mario Rothbauer. New York, NY: Springer, 2018, pp. 41–54. DOI: 10.1007/978-1-4939-7792-5_4.

[105] Anita Reiser, Daniel Woschée, Simon Maximilian Kempe, and Joachim Oskar Rädler. “Live-cell Imaging of Single-Cell Arrays (LISCA) - a Versatile Technique to Quantify Cellular Kinetics”. In: *JoVE (Journal of Visualized Experiments)* 169 (Mar. 18, 2021), e62025. DOI: 10.3791/62025.

[106] Judith A. Müller, Gerlinde Schwake, and Joachim O. Rädler. “Einzelzellmikroskopie im Hochdurchsatz auf Mikrostrukturen”. In: *BIOspektrum* 28.7 (Nov. 1, 2022), pp. 723–725. DOI: 10.1007/s12268-022-1857-8.

[107] *SoftmatterLMU-RaedlerGroup/pyama*. Apr. 20, 2021.

[108] Michael Schwarzfischer, Carsten Marr, Jan Krumsiek, Philipp Hoppe, T Schroeder, and Fabian Theis. "Efficient fluorescence image normalization for time lapse movies". In: Jan. 1, 2011.

[109] Edda Klipp, Ralf Herwig, Axel Kowald, Christoph Wierling, and Hans Lehrach. *Systems Biology in Practice: Concepts, Implementation and Application*. Google-Books-ID: bLhw9l1OJOUC. John Wiley & Sons, May 6, 2005. 496 pp.

[110] Sebastian Persson. *sebapersson/PEtab.jl*. original-date: 2022-12-01T16:22:49Z. Sept. 2, 2024.

[111] Gerd Steinebach. "Construction of Rosenbrock–Wanner method Rodas5P and numerical benchmarks within the Julia Differential Equations package". In: *BIT Numerical Mathematics* 63.2 (Apr. 17, 2023), p. 27. DOI: 10.1007/s10543-023-00967-x.

[112] Dong C. Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1 (Aug. 1, 1989), pp. 503–528. DOI: 10.1007/BF01589116.

[113] Nathalie Gabriele Schäffler. *Gene Expression Fitting in Julia · GitLab*. Sept. 2, 2024.

[114] David T. Gonzales, Naresh Yandrapalli, Tom Robinson, Christoph Zechner, and T-Y. Dora Tang. "Cell-Free Gene Expression Dynamics in Synthetic Cell Populations". In: *ACS Synthetic Biology* 11.1 (Jan. 21, 2022). Publisher: American Chemical Society, pp. 205–215. DOI: 10.1021/acssynbio.1c00376.

[115] Gerlinde Schwake, Simon Youssef, Jan-Timm Kuhr, Sebastian Gude, Maria Pamela David, Eduardo Mendoza, Erwin Frey, and Joachim O. Rädler. "Predictive modeling of non-viral gene transfer". In: *Biotechnology and Bioengineering* 105.4 (2010), pp. 805–813. DOI: 10.1002/bit.22604.

[116] Andrea Sacchetti, Tarek El Sewedy, Ashraf F Nasr, and Saverio Alberti. "Efficient GFP mutations profoundly affect mRNA transcription and translation rates". In: *FEBS Letters* 492.1 (2001), pp. 151–155. DOI: 10.1016/S0014-5793(01)02246-3.

[117] Xu Tang, Simon Sretenovic, Qiurong Ren, Xinyu Jia, Mengke Li, Tingting Fan, Desuo Yin, Shuyue Xiang, Yachong Guo, Li Liu, Xuelian Zheng, Yiping Qi, and Yong Zhang. "Plant Prime Editors Enable Precise Gene Editing in Rice Cells". In: *Molecular Plant* 13.5 (May 4, 2020). Publisher: Elsevier, pp. 667–670. DOI: 10.1016/j.molp.2020.03.010.

[118] Matthew A. Coelho, Songyuan Li, Luna Simona Pane, Mike Firth, Giovanni Ciotta, Jonathan D. Wrigley, Maria Emanuela Cuomo, Marcello Maresca, and Benjamin J. M. Taylor. "BE-FLARE: a fluorescent reporter of base editing activity reveals editing characteristics of APOBEC3A and APOBEC3B". In: *BMC Biology* 16 (Dec. 28, 2018), p. 150. DOI: 10.1186/s12915-018-0617-1.

[119] Kylie Standage-Beier, Stefan J Tekel, Nicholas Brookhouser, Grace Schwarz, Toan Nguyen, Xiao Wang, and David A Brafman. “A transient reporter for editing enrichment (TREE) in human cells”. In: *Nucleic Acids Research* 47.19 (Nov. 4, 2019), e120. DOI: `10.1093/nar/gkz713`.

[120] A. St. Martin, D. J. Salamango, A. A. Serebrenik, N. M. Shaban, W. L. Brown, and R. S. Harris. “A panel of eGFP reporters for single base editing by APOBEC-Cas9 editosome complexes”. In: *Scientific Reports* 9 (Jan. 24, 2019), p. 497. DOI: `10.1038/s41598-018-36739-9`.

[121] Mohadeseh Khoshandam, Hossein Soltaninejad, Marziyeh Mousazadeh, Amir Ali Hamidieh, and Saman Hosseinkhani. “Clinical applications of the CRISPR/Cas9 genome-editing system”. In: *Genes & Diseases* 11.1 (Jan. 1, 2024), pp. 268–282. DOI: `10.1016/j.gendis.2023.02.027`.

[122] Dan Budwick. *Prime Medicine Announces FDA Clearance of Investigational New Drug (IND) Application for PM359 for the Treatment of Chronic Granulomatous Disease (CGD) - Prime Medicine, Inc.* news release details, prime medicine. Apr. 29, 2024. URL: `https://investors.primemedicine.com/news-releases/news-release-details/prime-medicine-announces-fda-clearance-investigational-new-drug/` (visited on 09/04/2024).

[123] Garry A. Luke and Martin D. Ryan. ““Therapeutic applications of the ‘NPGP’ family of viral 2As””. In: *Reviews in Medical Virology* 28.6 (2018), e2001. DOI: `10.1002/rmv.2001`.

[124] Jin Hee Kim, Sang-Rok Lee, Li-Hua Li, Hye-Jeong Park, Jeong-Hoh Park, Kwang Youl Lee, Myeong-Kyu Kim, Boo Ahn Shin, and Seok-Yong Choi. “High Cleavage Efficiency of a 2A Peptide Derived from Porcine Teschovirus-1 in Human Cell Lines, Zebrafish and Mice”. In: *PLOS ONE* 6.4 (Apr. 29, 2011). Publisher: Public Library of Science, e18556. DOI: `10.1371/journal.pone.0018556`.

[125] Sarah S. Nasr, Pascal Paul, Brigitta Loretz, and Claus-Michael Lehr. “Realizing time-staggered expression of nucleic acid-encoded proteins by co-delivery of messenger RNA and plasmid DNA on a single nanocarrier”. In: *Drug Delivery and Translational Research* (July 15, 2024). DOI: `10.1007/s13346-024-01668-w`.

[126] Yamin Li, Zhongfeng Ye, Hanyi Yang, and Qiaobing Xu. “Tailoring combinatorial lipid nanoparticles for intracellular delivery of nucleic acids, proteins, and drugs”. In: *Acta Pharmaceutica Sinica B* 12.6 (June 1, 2022), pp. 2624–2639. DOI: `10.1016/j.apsb.2022.04.013`.

[127] Pardis Kazemian, Si-Yue Yu, Sarah B. Thomson, Alexandra Birkenshaw, Blair R. Leavitt, and Colin J. D. Ross. “Lipid-Nanoparticle-Based Delivery of CRISPR/Cas9 Genome-Editing Components”. In: *Molecular Pharmaceutics* 19.6 (June 6, 2022). Publisher: American Chemical Society, pp. 1669–1686. DOI: `10.1021/acs.molpharmaceut.1c00916`.

[128] Daniel Rosenblum, Anna Gutkin, Ranit Kedmi, Srinivas Ramishetti, Nuphar Veiga, Ashley M. Jacobi, Mollie S. Schubert, Dinorah Friedmann-Morvinski, Zvi R. Cohen, Mark A. Behlke, Judy Lieberman, and Dan Peer. "CRISPR-Cas9 genome editing using targeted lipid nanoparticles for cancer therapy". In: *Science Advances* 6.47 (Nov. 18, 2020). Publisher: American Association for the Advancement of Science, eabc9450. DOI: 10.1126/sciadv.abc9450.

[129] Dong Hyun Jo, Taeyoung Koo, Chang Sik Cho, Jin Hyoung Kim, Jin-Soo Kim, and Jeong Hun Kim. "Long-Term Effects of In Vivo Genome Editing in the Mouse Retina Using Campylobacter jejuni Cas9 Expressed via Adeno-Associated Virus". In: *Molecular Therapy* 27.1 (Jan. 2, 2019). Publisher: Elsevier, pp. 130–136. DOI: 10.1016/j.ymthe.2018.10.009.

[130] Patrick Irmisch, Thomas E. Ouldridge, and Ralf Seidel. "Modeling DNA-Strand Displacement Reactions in the Presence of Base-Pair Mismatches". In: *Journal of the American Chemical Society* 142.26 (July 1, 2020). Publisher: American Chemical Society, pp. 11451–11463. DOI: 10.1021/jacs.0c03105.

[131] Jinman Park, Goosang Yu, Sang-Yeon Seo, Jinyeong Yang, and Hyongbum Henry Kim. "SynDesign: web-based prime editing guide RNA design and evaluation tool for saturation genome editing". In: *Nucleic Acids Research* 52 (W1 July 5, 2024), W121–W125. DOI: 10.1093/nar/gkae304.

[132] Jonathan Y. Hsu, Julian Grünewald, Regan Szalay, Justine Shih, Andrew V. Anzalone, Kin Chung Lam, Max W. Shen, Karl Petri, David R. Liu, J. Keith Joung, and Luca Pinello. "PrimeDesign software for rapid and simplified design of prime editing guide RNAs". In: *Nature Communications* 12.1 (Feb. 15, 2021). Publisher: Nature Publishing Group, p. 1034. DOI: 10.1038/s41467-021-21337-7.

[133] Jonathan Y. Hsu, Kin Chung Lam, Justine Shih, Luca Pinello, and J. Keith Joung. *MOSAIC enables in situ saturation mutagenesis of genes and CRISPR prime editing guide RNA optimization in human cells.* Pages: 2024.04.25.591078 Section: New Results. Apr. 26, 2024. DOI: 10.1101/2024.04.25.591078.

[134] Yichao Li, Jingjing Chen, Shengdar Q. Tsai, and Yong Cheng. "Easy-Prime: a machine learning–based prime editor design tool". In: *Genome Biology* 22.1 (Aug. 19, 2021), p. 235. DOI: 10.1186/s13059-021-02458-0.

[135] Elisabeth Roesch, Joe G. Greener, Adam L. MacLean, Huda Nassar, Christopher Rackauckas, Timothy E. Holy, and Michael P. H. Stumpf. "Julia for biologists". In: *Nature Methods* 20.5 (May 2023). Publisher: Nature Publishing Group, pp. 655–664. DOI: 10.1038/s41592-023-01832-z.

[136] Juane Lu, Tao Wu, Biao Zhang, Suke Liu, Wenjun Song, Jianjun Qiao, and Haihua Ruan. "Types of nuclear localization signals and mechanisms of protein import into the nucleus". In: *Cell Communication and Signaling* 19.1 (May 22, 2021), p. 60. DOI: 10.1186/s12964-021-00741-y.

[137] Austin T. Raper, Anthony A. Stephenson, and Zucai Suo. “Functional Insights Revealed by the Kinetic Mechanism of CRISPR/Cas9”. In: *Journal of the American Chemical Society* 140.8 (Feb. 28, 2018). Publisher: American Chemical Society, pp. 2971–2984. DOI: `10.1021/jacs.7b13047`.

[138] Noelia Lander, Miguel A. Chiurillo, and Roberto Docampo. “Genome Editing by CRISPR/Cas9: A Game Change in the Genetic Manipulation of Protists”. In: *Journal of Eukaryotic Microbiology* 63.5 (2016). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jeu.12338, pp. 679–690. DOI: `10.1111/jeu.12338`.

[139] Lewis Grozinger, Martyn Amos, Thomas E. Gorochowski, Pablo Carbonell, Diego A. Oyarzún, Ruud Stoof, Harold Fellermann, Paolo Zuliani, Huseyin Tas, and Angel Goñi-Moreno. “Pathways to cellular supremacy in biocomputing”. In: *Nature Communications* 10.1 (Nov. 20, 2019). Publisher: Nature Publishing Group, p. 5250. DOI: `10.1038/s41467-019-13232-z`.

[140] Zhi Ping, Dongzhao Ma, Xiaoluo Huang, Shihong Chen, Longying Liu, Fei Guo, Sha Joe Zhu, and Yue Shen. “Carbon-based archiving: current progress and future prospects of DNA-based data storage”. In: *GigaScience* 8.6 (June 1, 2019), giz075. DOI: `10.1093/gigascience/giz075`.

[141] Walter Fontana and Leo W. Buss. ““The arrival of the fittest”: Toward a theory of biological organization”. In: *Bulletin of Mathematical Biology* 56.1 (Jan. 1, 1994), pp. 1–64. DOI: `10.1007/BF02458289`.

[142] Romilde Manzoni, Arturo Urrios, Silvia Velazquez-Garcia, Eulàlia de Nadal, and Francesc Posas. “Synthetic biology: insights into biological computation”. In: *Integrative Biology* 8.4 (Apr. 18, 2016), pp. 518–532. DOI: `10.1039/c5ib00274e`.

[143] Maja Meško, Tina Lebar, Petra Dekleva, Roman Jerala, and Mojca Benčina. “Engineering and Rewiring of a Calcium-Dependent Signaling Pathway”. In: *ACS Synthetic Biology* 9.8 (Aug. 21, 2020). Publisher: American Chemical Society, pp. 2055–2065. DOI: `10.1021/acssynbio.0c00133`.

[144] Harold Abelson and Gerald Jay Sussman. *Structure and interpretation of computer programs*. The MIT Press, 1996.

[145] Felix Wolter and Holger Puchta. “In planta gene targeting can be enhanced by the use of CRISPR/Cas12a”. In: *The Plant Journal* 100.5 (2019). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tpj.14488, pp. 1083–1094. DOI: `10.1111/tpj.14488`.

[146] Albert Blanch-Asensio, Catarina Grandela, Karina O. Brandão, Tessa de Korte, Hailiang Mei, Yavuz Ariyurek, Loukia Yiangou, Mervyn P.H. Mol, Berend J. van Meer, Susan L. Kloet, Christine L. Mummery, and Richard P. Davis. “STRAIGHT-IN enables high-throughput targeting of large DNA payloads in human pluripotent stem cells”. In: *Cell Reports Methods* 2.10 (Sept. 22, 2022), p. 100300. DOI: `10.1016/j.crmeth.2022.100300`.

[147] David C. Amberg, Dan Burke, and Jeffrey N. Strathern. *Methods in Yeast Genetics: A Cold Spring Harbor Laboratory Course Manual.* Google-Books-ID: 47zspjkKkCwC. CSHL Press, 2005. 250 pp.

[148] Anila George, Nithin Sam Ravi, Kirti Prasad, Lokesh Panigrahi, Sanya Koikkara, Vignesh Rajendiran, Nivedhitha Devaraju, Joshua Paul, Aswin Anand Pai, Yukio Nakamura, Ryo Kurita, Poonkuzhali Balasubramanian, Saravanabhavan Thangavel, Srujan Marepally, Shaji R. Velayudhan, Alok Srivastava, and Kumarasamypet M. Mohankumar. "Efficient and error-free correction of sickle mutation in human erythroid cells using prime editor-2". In: *Frontiers in Genome Editing* 4 (Dec. 20, 2022). Publisher: Frontiers. DOI: 10.3389/fgeed.2022.1085111.

[149] *HEK293 Cells: Background, Applications, Protocols, and More.* Synthego. URL: https://www.synthego.com/hek293 (visited on 09/25/2024).

[150] C. Eric Hodgman and Michael C. Jewett. "Cell-free synthetic biology: Thinking outside the cell". In: *Metabolic Engineering.* Synthetic Biology: New Methodologies and Applications for Metabolic Engineering 14.3 (May 1, 2012), pp. 261–269. DOI: 10.1016/j.ymben.2011.09.002.

[151] Johannes B. Woehrstein, Maximilian T. Strauss, Luvena L. Ong, Bryan Wei, David Y. Zhang, Ralf Jungmann, and Peng Yin. "Sub–100-nm metafluorophores with digitally tunable optical properties self-assembled from DNA". In: *Science Advances* 3.6 (June 21, 2017). Publisher: American Association for the Advancement of Science, e1602128. DOI: 10.1126/sciadv.1602128.

[152] David Shore and Benjamin Albert. "Ribosome biogenesis and the cellular energy economy". In: *Current Biology* 32.12 (June 20, 2022). Publisher: Elsevier, R611–R617. DOI: 10.1016/j.cub.2022.04.083.

[153] Eva K. Brinkman, Tao Chen, Marcel de Haas, Hanna A. Holland, Waseem Akhtar, and Bas van Steensel. "Kinetics and Fidelity of the Repair of Cas9-Induced Double-Strand DNA Breaks". In: *Molecular Cell* 70.5 (June 7, 2018). Publisher: Elsevier, 801–813.e6. DOI: 10.1016/j.molcel.2018.04.016.

[154] Vladimir Mekler, Leonid Minakhin, Ekaterina Semenova, Konstantin Kuznedelov, and Konstantin Severinov. "Kinetics of the CRISPR-Cas9 effector complex assembly and the role of 3'-terminal segment of guide RNA". In: *Nucleic Acids Research* 44.6 (Apr. 7, 2016), pp. 2837–2845. DOI: 10.1093/nar/gkw138.

[155] Hisashi Tadakuma, Yo Ishihama, Toshiharu Shibuya, Tokio Tani, and Takashi Funatsu. "Imaging of single mRNA molecules moving within a living cell nucleus". In: *Biochemical and Biophysical Research Communications* 344.3 (June 9, 2006), pp. 772–779. DOI: 10.1016/j.bbrc.2006.03.202.

[156] Giulia Paci, Tiantian Zheng, Joana Caria, Anton Zilman, and Edward A Lemke. "Molecular determinants of large cargo transport into the nucleus". In: *eLife* 9 (July 21, 2020). Ed. by Robert H Singer and Suzanne R Pfeffer. Publisher: eLife Sciences Publications, Ltd, e55963. DOI: 10.7554/eLife.55963.

Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, mich bei all denen zu bedanken, die mich während der Erstellung dieser Dissertation und meiner Forschung unterstützt und begleitet haben.

Zunächst gilt mein aufrichtiger Dank meinem Betreuer, **Prof. Dr. Joachim Rädler**. Danke für die Möglichkeit, in deiner Arbeitsgruppe zu forschen, dadurch konnte ich wertvolle wissenschaftliche Erfahrungen sammeln und mich in meinem Fachbereich weiterentwickeln. Dein Vertrauen in meine Arbeit hat mir die Freiheit gegeben, eigene Ideen zu verfolgen und mein Forschungsvorhaben eigenständig umzusetzen.

Ein herzliches Dankeschön richte ich auch an **Prof. Dr. Ralf Zimmer**, **Prof. Dr. Gil Westmeyer** und **Dr. Jona Kayser**, deren wertvolle Ratschläge und anregenden Diskussionen diese Arbeit bereichert haben.

Ich möchte auch allen am Lehrstuhl Rädler danken. Die Zeit mit euch war geprägt durch viele Mittagsessens-, Kaffee- und Kuchensessions bei dem sowohl wissenschaftliche, aber auch so manche nicht sehr wissenschaftliche Diskussionen und Ideen entstanden sind. Besonders danken möchte ich Gerlinde, Charlott und Susi, ohne die das Labor einfach nicht funktionieren würde.

Judith, Emily, Chris, Stefan und Johannes. Danke für all unsere Aperol-Abende, Konferenzen- und Antholz-Fahrten, euer Feedback und eure Unterstützung.

Meinen fleißigen Korrekturlesern Max, Alex, Friede, Katja und Dan, danke ich ebenfalls.

Meinen **StrickOmis** möchte ich für all die Unterstützung danken, die ich während meiner Masterzeit und dann auch der Dissertation bekommen habe. Ich habe es so genossen mit euch zu basteln, kochen und plauschen und hoffe das hält noch ewig an!

Meiner Familie: Danke, dass ihr immer für mich da wart! Ich weiß, dass egal was kommt ich immer auf euch zählen kann und das ist mir mehr wert als ich jemals in Worte fassen könnte. Und außerdem für meinen Bruder: **Ceterum censeo Carthaginem esse delendam!**

Thomas (and Mochi) for always being there for me, supporting me (all the Flauschies), and cheering me up on my bad days. Life is perfect with you (two) and I couldn't imagine it without you 😘.