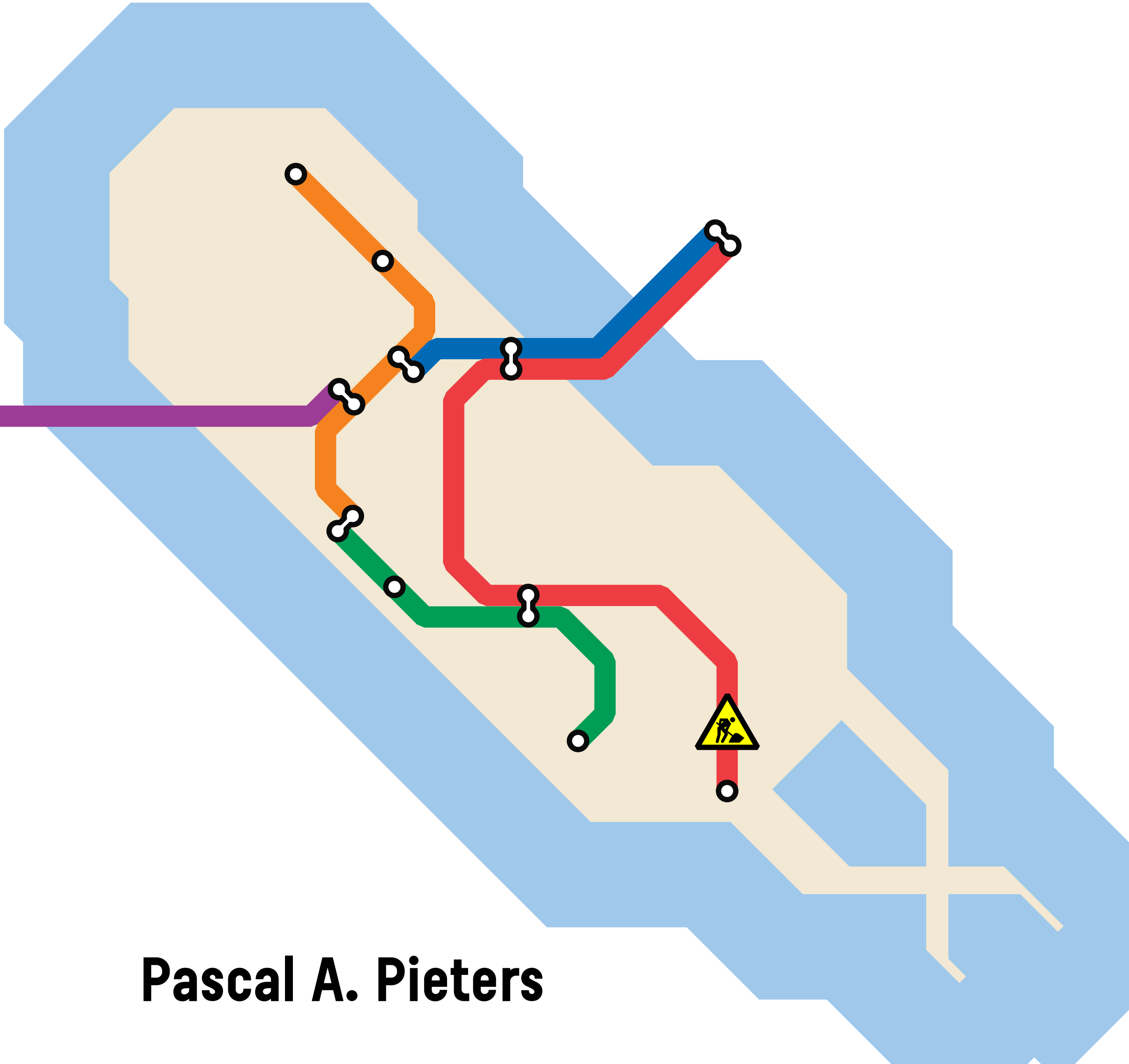
Engineering microcompartmentalized cell-free synthetic circuits

ENGINEERING
MICROCOMPARTMENTALIZED
CELL-FREE
SYNTHETIC CIRCUITS



Pascal A. Pieters

Engineering microcompartmentalized cell-free synthetic circuits

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op donderdag 9 september 2021 om 16:00 uur

door

Pascal Aldo Pieters

geboren te Leeuwarden

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. M. Merkx
1e promotor:	prof.dr.ir. T.F.A. de Greef
2e promotor:	prof.dr. P.A.J. Hilbers
leden:	prof.dr.ir. N.A.W. van Riel
	prof. O. Ces (Imperial College London)
	prof. T.M. Hermans (Université de Strasbourg)
	prof.dr. E. Mastrobattista (Universiteit Utrecht)
	A.S. Meyer, PhD (University of Rochester)
adviseur:	D. Tang, PhD (Max Planck Institute of Molecular Cell Biology and Genetics)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Engineering microcompartmentalized cell-free synthetic circuits
Pascal A. Pieters
Eindhoven University of Technology, The Netherlands, 2021

Printed by Printenbind

A catalogue record is available from the
Eindhoven University of Technology Library

ISBN: 978-90-386-5358-7

This work has been financially supported by the European Research Council
(ERC project no. 677313 BioCircuit)

Contents

1

Introduction

Part of this work has been published as:

Emilien Dubuc, Pascal A. Pieters, Ardjan J. van der Linden, Jan C.M. van Hest, Wilhelm T.S. Huck, Tom F.A. de Greef. *Current Opinion in Biotechnology* **58**, 72-80 (2019)

1.1 Genetic Regulatory Networks

Understanding the complex regulatory networks of cells is one of the key goals of the fields of synthetic biology and systems biology. Due to the multitude of interacting components in these networks, this has remained a challenging undertaking. Synthetic biology has been successful in implementing novel functionalities in natural cells, starting with the construction of a genetic oscillator[1] and toggle switch (Figure 1.1a, b)[2] in Escherichia coli. This research demonstrated that existing genetic elements could be engineered to form artificial genetic circuits that exhibit specific dynamical behavior, such as oscillations and multi-stability. This has paved the way for the creation of increasingly complex artificial circuitry in living cells. For example, artificial enzymatic pathways have been constructed in organisms to produce chemical compounds of great societal interest, such as an antimalarial drug precursor.[3,4]

Building upon the initial work by Gardner et al.,[2] the concept of artificial toggle switches in cells has been further extended to Latch gates, which were used to construct finite state machines, borrowing key concepts from automata theory.[5] To achieve this, a large collection of prokaryotic repressors of the same family as the TetR repressor used by Gardner *et al.*[2] was mined from genomic data. The transfer function of all repressors and their cross-reactivity were determined in a uniform manner, yielding a library of well-characterized NOT/NOR-gates (Figure 1.1c). A bistable Set/Reset (SR) Latch was subsequently engineered by taking two NOR gates and connecting the output of each gate to the input of the other gate, whilst adjusting the ribosome binding site (RBS) strength to create a system with a phase diagram with two distinct stable steady states (Figure 1.1d, e, f). This work expertly demonstrates how well-characterized synthetic genetic circuits can be combined to construct complex genetic networks and develop novel cellular functionalities.

As the complexity of synthetic gene networks increases, more resources are shared between synthetic circuits and their host. As a result, cells experience a decrease in fitness, which limits their growth and the efficiency of synthetic circuits.[6] Furthermore, implementing synthetic circuits within cellular hosts exposes these networks to interactions with endogenous pathways, negatively influencing the behavior of both host and synthetic circuitry. Various attempts have been made to address and

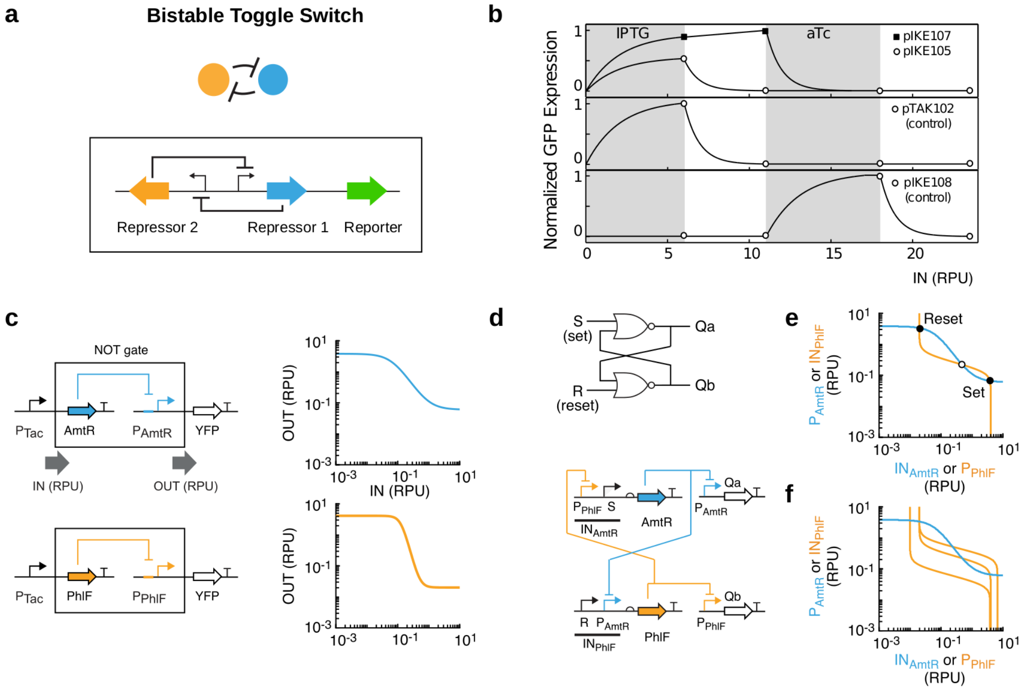


Figure 1.1: a) Schematic depicting a toggle switch and a corresponding genetic circuit. Two repressor proteins repress the promoter of the other gene to create two states in which either one of the proteins is abundant. b) Fluorescence output of the bistable switch developed by Gardner and coworkers[2], where the state of the system is switched by application of inducer molecules. Reprinted with permission from Gardner et al.[2] c) Schematics and response functions of two NOT gates used to construct complex latches and state machines. From Andrews et al.[5] Reprinted with permission from AAAS. d) Genetic implementation of a set-reset latch using two simple gates and reporter constructs. Reprinted with permission from Andrews et al.[5] e) Phase plane analysis of the latch using the response functions as nullclines. The stable (solid) and unstable (open) steady states are given by the intersections of the nullclines. From Andrews et al.[5] Reprinted with permission from AAAS. f) Demonstration of the construction of a set-reset latch through optimization of the response functions by adjusting the strength of the ribosome binding site. From Andrews et al.[5] Reprinted with permission from AAAS.

circumvent this issue. Notably, a capacity monitor was developed by Ceroni *et al.* to analyze the burden on a host cell imposed by a synthetic genetic circuit.[7] The capacity monitor made use of a genomically integrated cassette of an unregulated endogenous promoter, which was sensitive to general changes in cell burden, and a GFP reporter protein. This setup was demonstrated to be a better predictor for cell burden than the growth rate and was utilized to develop a feedback mechanism that inhibits expression from synthetic circuits upon excessive cell burden.[8]

Nevertheless, cell burden and resource sharing effects remain prominent in synthetic genetic networks that are implemented *in vivo*. Therefore, bottom-up synthetic biology approaches have been used to construct cell-free biological systems that have well-defined behavior. Cell-free synthetic biology has emerged as a powerful tool for testing new genetic networks in a controlled biochemical context, as well as for predicting cellular responses to such networks.[9,10] These minimal models of biological systems can be used to study regulatory circuits in isolation, eliminating interference with a host cell. Using this approach, general principles and key parameters of these circuits can be identified.

1.1.1 Enzyme and Protein Reaction Networks

To achieve a well-defined environment to design and analyze regulatory networks, to limiting the number of unique components in the reactions is highly beneficial. For example, Semenov et al. used the trypsin enzyme and an inhibitor to create and study the behavior of an oscillating circuit.[11] The minimal nature of the system allowed the monitoring of key molecules over time and the reconstruction of the oscillatory trajectory in a 3-dimensional phase diagram. Many studies have focused on the reconstruction of existing regulatory networks in a cell-free environment. Notably, the KaiC phosphorylation cycle, which regulates the circadian rhythm in cyanobacteria, was successfully reconstituted *in vitro* and demonstrated to be self-sustained and stable under temperature changes. This result consolidated the notion that key regulatory processes can be studied in isolation using purified components. The benefits of this approach were exploited effectively by Coyle *et al.* through the reconstitution and elaborate functional mapping of signaling circuit of Ras-superfamily GTPases.[12] They reveal the influence of many of the up- and downstream proteins on the signaling cascade by utilizing the controllable constitution of the cell-free reactions to create various configurations of the Ras circuit and map the dynamic response of the system.

1.1.2 DNA Strand Displacement

The excellent predictability of interactions between DNA strands has resulted in the emergence of DNA strand displacement as a key platform to perform molecu-

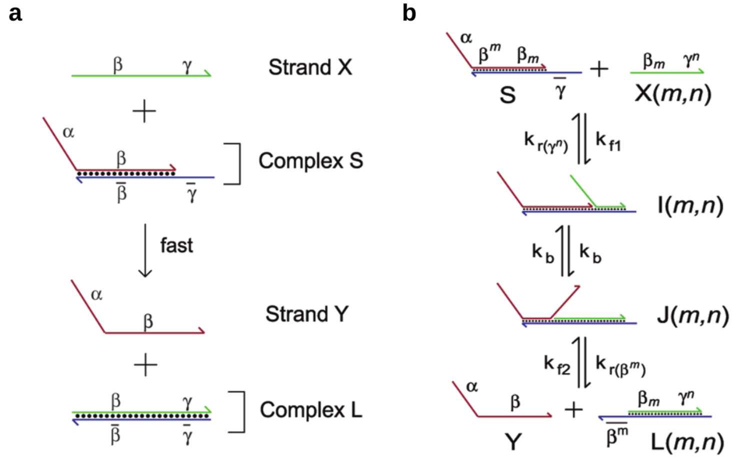


Figure 1.2: a) Schematic of a toehold-mediated DNA strand displacement reaction and the different DNA regions involved. Strand Y is displaced by strand X through, which is mediated by the toehold domain γ. Reprinted with permission from Zhang and Winfree.[14] *Copyright 2009 American Chemical Society. b) The mechanism of a toehold-mediated strand displacement reaction. Strand X binds the toehold γ to form intermediate I and initiate displacement of strand Y (intermediate J). Strand Y subsequently dissociates from the complex, completing the strand exchange reaction. Reprinted with permission from Zhang and Winfree.*[14] *Copyright 2009 American Chemical Society.*

lar computations, and various complex functions have been implemented as DNA chemical reaction networks.[13] In these reaction networks, the kinetics of DNA strand displacement reactions are modulated by toeholds, single-stranded DNA segments at the extremes of the DNA strands (Figure 1.2).[14] An important goal of the field of DNA nanotechnology is the construction of precisely defined structures[15] and DNA computers that replicate classical computing principles, such as a four-bit square-root circuit[16] and neural network circuits.[17] Nevertheless, various circuits that exhibit functions found in biological systems have been implemented as DNA-based chemical circuits, including signal restoration, amplification, feedback, and cascading.[18]

1.1.3 Genelets

Although DNA strand displacement reactions form a well-defined minimal system to mimic cellular processes, the uniformity of the reactions does not resemble the multimodal reactions governing genetic regulatory networks. DNA, RNA, and proteins provide multiple levels of regulation, which are all utilized in cells. Most notably, transcription factor proteins can activate and repress transcription from promoters on the DNA. This process was emulated in the transcription-based “genelet” system, which provides a versatile yet minimal platform to implement biochemical reaction networks.[19] Genelets are small dsDNA constructs that consist of a promoter and RNA gene. In the OFF-state, the promoter region of each genelet is partially single-stranded, which results in a slow transcription rate (Figure 1.3a). Genelets are activated by the addition of a ssDNA activator that completes the promoter region and restores transcriptional activity. Inhibitory interactions between genelets can be achieved by designing the coding sequence of the RNA genes such that the resulting RNA can release and bind the activator ssDNA of a different genelet through toehold-mediated strand displacement. Kim *et al.* developed the genelet system and utilized it to construct a bistable switch, for which they could experimentally map the bifurcation diagram (Figure 1.3b).[19] Additionally, they demonstrate that the system can be modeled computationally to great precision, which aides the understanding of the behavior of complex circuits. Capitalizing on the simplicity and molecularity of the genelet system, Kim and coworkers implemented an incoherent feed-forward loop using genelets and achieved exact adaptation and fold-change detection of signals.[20]

1.1.4 PEN Toolbox

Padirac *et al.* similarly developed a model system based on DNA replication, nicking, and degradation, using slightly more components, increasing the types of interactions that can be modeled using this system.[21] The system makes use of a DNA polymerase to assemble the complementary strand of a template segment of ssDNA, initiated by a primer (Figure 1.3c). The resulting DNA remains double-stranded until it is nicked by a nickase, creating two ssDNA fragments that can dissociate from the template and function as a primer for a new reaction. A bistable toggle switch was implemented on top of this system by Padirac et al., and they demonstrated that the synthetic toggle switch could be switched two times consecutively before running out of resources. Meijer and coworkers used this bistable switch design to reveal retroactive effects of the coupling of downstream modules to the circuit and devised a translator module that minimizes these undesired effects.[22]

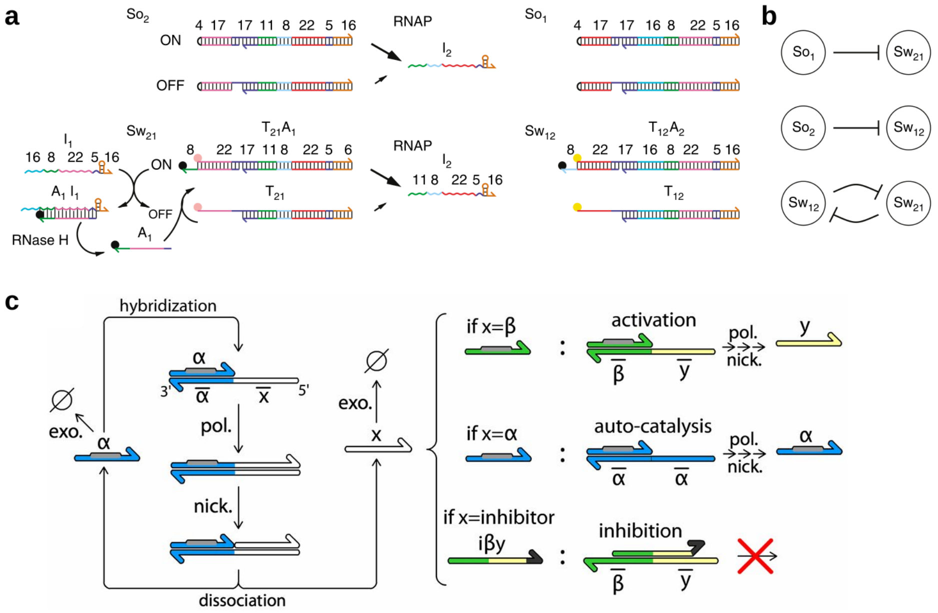


Figure 1.3: Genelets and the PEN toolbox. a) Schematic of the DNA sequences and mechanisms used to create genelet systems. Switches (Sw$_{12}$ and Sw$_{21}$) have an ON state, in which the promoter sequence is complete, and an OFF state, where the promoter is missing five bases. Source templates (So$_1$ and So$_2$) have an ON state with a complete promoter sequence with a nick. The OFF state source is also missing five bases of the promoter. The product of transcription of the active (ON state) switches and sources can interact with switches (not sources) to complete their promoter region and activate transcription. Inhibitor constructs (I$_1$) sequester and degrade activator strands, facilitated by RNase H. Reprinted with permission from Kim et al.[19] *b) Schematics of circuits constructed by Kim and coworkers using the genelet system. Reprinted with permission from Kim et al.*[19] c) *Schematic depiction of the PEN toolbox The system uses DNA templates and polymerase, nickase, and exonuclease to construct circuits. Bottom strands are the templates that bind inputs on the 3' end and release outputs at the 5'. Input α can hybridize to a template and is subsequently elongated by a polymerase. The complex is nicked at the interface between the input and output. Both the input and output are then released from the template and can participate in subsequent reactions. Exonuclease is used to degrade outputs and create dynamic circuits. Using this principle, activator, inhibitor, and auto-catalytic modules can be con*structed. Reprinted with permission from Padirac et al. [21]

1.2 Cell-free TXTL

Proteins and other components of the transcription and translation machinery can be extracted from cells as a lysate or purified and reconstituted (PURE) in order to perform gene expression *in vitro* with good control over the biochemical composition. Cell-free transcription and translation (TXTL) provides a basis to construct synthetic genetic networks that closely resembles bacterial cells, whilst retaining the minimal nature of *in vitro* experiments. Therefore, combined with compartmentalization methods, TXTL constitutes an ideal model system for deducing the rules of network composition, developing new communication pathways between cellular mimics and living cells,[23] and creating genetic devices for the implementation of synthetic communication between cells of different species.[24]

1.2.1 Component and Circuit Characterization

The development of complex genetic circuits requires the use of well-characterized regulatory modules. Cell-free TXTL allows the rapid and thorough testing of new genetic parts, thus enabling such characterization (Figure 1.4b: upper panel). Recently, a cell-free synthetic biology approach for the testing of novel regulatory elements with further *in vivo* applications has been established, leveraging the strong potential of riboregulation,[25–28] and dCas9-based repression.[29]

After characterizing regulatory building blocks, novel genetic circuits can be designed and assembled (Figure 1.4b: lower panel). *In vitro* characterization is often done under batch conditions, but these conditions do not ensure a constant supply of substrate, as well as the removal of by-products, and the renewal of information, which are prerequisites for the implementation of higher-order regulatory behaviors. In contrast, flow reactors enable the testing of regulatory modules and the implementation of complex networks in conditions mimicking cellular homeostasis,[30,31] as well as the forward engineering of such networks into bacterial hosts.[32] In addition, TXTL in flow reactors constitutes an effective method for approaching synthetic biology from control theory perspectives, where compositional context, cell heterogeneity, and division become controlled parameters instead of poorly defined variables.[33] These flow reactors could serve as a platform for studying network effects, such as retroactivity (Figure 1.4c: upper panel) between modules or circuits sharing common resources and test different compositions of a molecular circuit (Figure 1.4c: lower panel).

Synthetic gene networks mobilize resources essential to the cell, which are often found in limited supply (Figure 1.4d: left panel). In order to rapidly estimate the resource cost of integrating novel synthetic circuits, Borkowski *et al.* coupled a TXTL batch system to a capacitor circuit, which facilitated circuit transposition

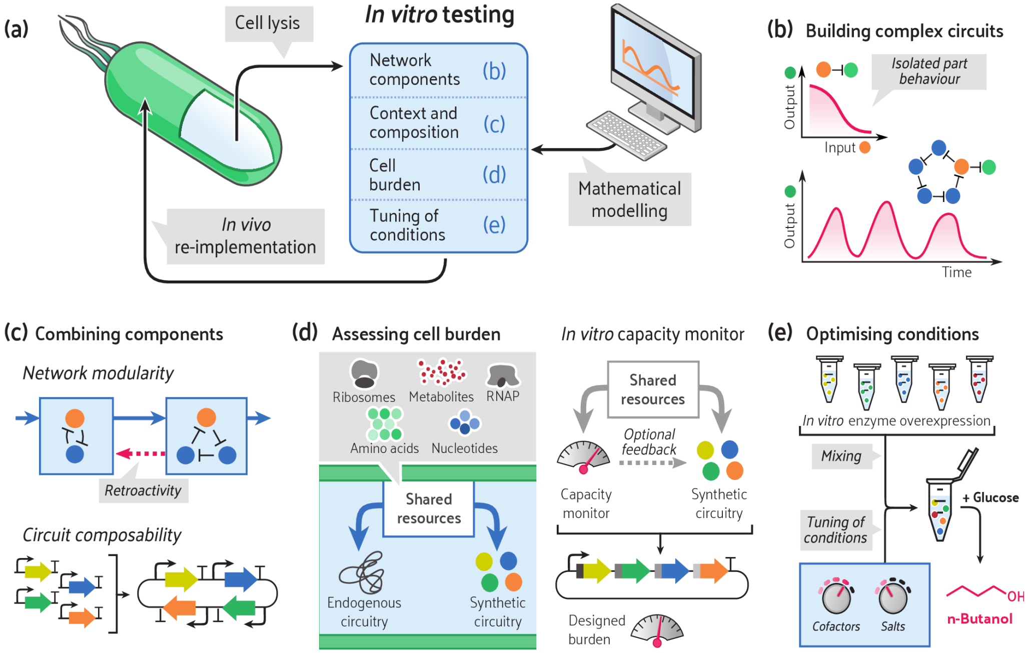


Figure 1.4: a) Cell-free TXTL reactions can be used in combination with mathematical modeling to test complex circuits and identify optimal conditions for the implementation of novel biological functions in vivo. b) Cell-free reactions are used to characterize the behavior of novel isolated circuit parts[32] *(upper panel)) and test the behavior of circuits combining novel parts (lower panel). c) Various versions of similar networks can rapidly be tested in TXTL, which can be used to reveal retroactive interactions between modules (upper panel), as well as unexpected effects when combining components into a single construct (lower panel). d) Cell burden occurs when a synthetic circuit excessively mobilizes resources that are also necessary for endogenous circuits (left panel). A capacity monitor reports possible cell burden during the implementation of novel circuits using TXTL and allows the design of networks generating minimal burden (right panel).*[34] *e) TXTL allows the testing of various biochemical conditions, such as cofactor, salt, and enzyme concentrations, in order to optimize a TXTL-based reaction.*[42]

into cells (Figure 1.4d: right panel).[34] By leveraging the possibility to tune parameters critical to the TXTL reaction, such as DNA instructions and DNA concentration, multiple other systems have also been used to screen and identify resource competition.[32,30,35]

1.2.2 Tuneability, context, and composition

One major advantage of cell-free TXTL reactions is the possibility to adjust the parameters influencing the performance of a synthetic circuit, whereas *in vivo* methods rely predominantly on host resources and characteristics. Egbert and Klavins established that, in living cells, the performance of a non-endogenous genetic circuit is highly context-dependent, as a network can yield its designed function in one host strain but fails to achieve this function in another strain.[36] The choice of a bacterial host could be guided by identifying *in vitro* which parameters are essential for the desired behavior and selecting a strain matching these context requirements. To further optimize the host context, many regulatory elements critical for protein synthesis can be added or expressed *in situ*, such as MazF ribonuclease,[37] GamS[38,] and Chi6[39] DNase inhibitors, GreA/B transcription elongation factors,[40] and ClpXP protease.[41] Karim *et al*. composed a mixture of lysates enriched in specific enzymes or with enzymes expressed *in situ* in order to identify major directions for improving the yield of *n*-butanol synthesis (Figure 1.4e).[42] Furthermore, Yeung *et al*. leveraged the tuneability of TXTL systems in order to show that supercoiling is one of the main factors explaining differences in expression levels within an intergenic context.[43]

1.3 TXTL in Microcompartments

Cells display a high intrinsic concentration of biomolecules, a reduced volume, and an amphiphilic interface between their internal volume and external environment. Engineering synthetic compartments of controlled size and composition for the conduction of TXTL reaction enables the study of gene expression at the relevant scale. We distinguish three experimental microcompartments that have been employed to perform TXTL or prototype synthetic genetic networks based on cell-free TXTL, first, PDMS-based compartments, second, lipid-based compartments, and third, coacervates, also known as complex aqueous two-phase systems (ATPS).

1.3.1 PDMS-based microcompartments

In contrast with methods using free-floating DNA gene templates, Bar-Ziv *et al.* proposed a method for performing TXTL reaction on a chip in which the gene template DNA was attached to a functionalized surface, first under batch conditions using wheat-germ extract,[44] and subsequently in microfluidic flow reactors using *E. coli* cell extract.[31] The latter, alongside the work of Niederholtmeyer *et al.* (Figure 1.5a),[30] were amongst the first examples of long-term TXTL and demonstrated the implementation of advanced synthetic gene networks such as oscillators. Although challenging to fabricate and use, the devices developed by Niederholtmeyer *et al.* are a major technological breakthrough that will facilitate the development of numerous synthetic gene circuits in the future.

Gene expression-pattern formation plays an essential role in morphogenesis and is a relevant case-study for cell-free synthetic biology. In 2005, Isalan *et al.* implemented a coarse mimic of a *Drosophila melanogaster* morphogenic gene network.[45] Three genes were coupled to paramagnetic beads at defined locations inside a cm-long batch chamber containing wheat germ lysate-based TXTL, yielding the emergence of a reaction-diffusion network allowing the formation of gene expression patterns. This work highlighted important criteria for the implementation of pattern-forming gene networks, first, control over the sublocation of network components, second, control over resource competition and depletion, and third, control over protein degradation. The previously described microfluidic system developed by the Bar-Ziv group provided a platform within which each of these critical criteria can be controlled (Figure 1.5b). Tayar *et al.* leveraged this technical improvement to study the emergence of expression patterns from out-of-equilibrium gene networks, first using a bistable switch,[46] and later a series of oscillators coupled in space.[47] In a follow-up study, Pardatscher *et al.* developed a lithography technique in order to functionalize a surface with up to three distinct DNA strands on a chip supporting both lysate-based and PURE-based TXTL reaction.[48] This technique will allow the study of complex interactions between multiple, spatially resolved genes and the formation of more complex spatial patterns.

1.3.2 Lipid-based microcompartments

The cell membrane constitutes an amphiphilic interface between the interior of the cell and its environment. This interface can be mimicked by encapsulating TXTL reactions inside single (water-in-oil) or double (water-in-oil-in-water) emulsions in order to study a large number of these reactions at a cellular scale.

Large numbers of TXTL microdroplets in oil with a controlled dispersity are easy to generate, store, and remain stable over hours, which makes them particularly interesting for screening large numbers of parameters influencing TXTL reactions

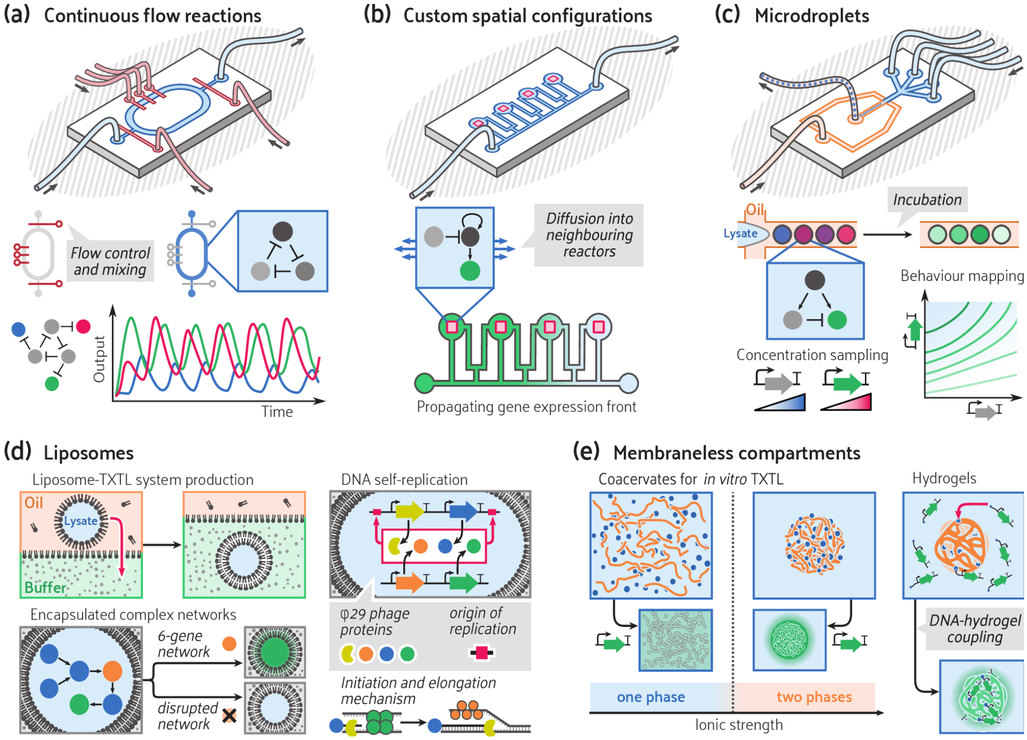


Figure 1.5: a) Continuous flow reactors allow the implementation of out-of-equilibrium gene networks in TXTL. Valves precisely control addition and mixing of fresh TXTL reagents, enabling the implementation of complex networks such as oscillators.[32] *b) Control over geometry and diffusion allows the study of gene expression propagation and pattern formation in compartmentalized flow reactors using DNA brushes.*[46] *c) Microdroplets generated on a microchip allow the screening of a vast range of conditions influencing network behavior in TXTL, such as DNA template concentrations, shown here for the implementation of an incoherent feed-forward loop.*[49] *d) Liposomes encapsulating TXTL can be obtained by double-emulsion techniques (top-left panel). Liposomes are used for isolating genetic cascades*[82] *(bottom-left panel) and creating units containing self-replicating genetic information (right panel).*[58] *e) Upon increase of ionic strength, TXTL mixture can phase separate, resulting in a highly active microcompartment (left panel).*[62] *DNA-functionalised hydrogels are used to create membraneless compartments capable of gene expression (right panel).*

(Figure 1.5c). This feature was elegantly employed by Hori *et al.*, who optimized a cell-free genetic circuit based on an incoherent feed-forward topology using fluorescently barcoded droplets in which lysate-based TXTL reactions can take place.[49]

The reduced size of microdroplets is particularly interesting to study physical effects such as confinement. Guan *et al*. encapsulated Xenopus egg extract in microdroplets of various sizes. The authors showed that the extract could undergo several mitotic oscillations and described the influence of the size of the compartments on the period.[50] Sakamoto *et al*. investigated the influence of the surface-to-volume ratio of microcompartments on the efficiency of lysate-based TXTL, suggesting a deleterious interaction between membrane lipids and translation machinery.[51]

Macromolecular crowding effects, which emerge in highly concentrated media such as the cytosol, can be artificially induced in TXTL mixtures using crowding additives inside microdroplets. Molecular crowding has been shown to influence the spatial segregation of biomolecules and kinetics of cell-free TXTL reactions.[52,53] In addition, Norred *et al*. showed that noise in transcription reaction in systems displaying reduced diffusion resulted in gene expression bursts, which are likely to occur in cells but are not observable in bulk PURE-based TXTL reactions.[54]

Given that droplets are intrinsically closed systems, they cannot be utilized for long-term protein expression nor as analogues of semipermeable lipid bilayers for mimicking the cell membrane. Noireaux *et al*. first implemented lysate-based TXTL reactions in a liposome permeated by hemolysin pores, which enabled the exchange of nutrients and by-products with a feeding solution, resulting in long-term gene expression.[55] Later, Garamella *et al*. tested various transcriptional cascades inside liposomes (Figure 1.5d: left panel).[38] Majumder *et al*. implemented a lysate-based TXTL liposome sensitive to osmotic changes via expression of the MscL calcium channel.[56] Moreover, lysate encapsulating liposomes were recently used by Krinsky *et al*. to serve as containers for the production and intratumoral delivery of a toxin protein.[57] Finally, Van Nies *et al*. recently described the first liposomes capable of isothermally replicating DNA using self-encoded proteins in PURE system, constituting a major step towards the construction of a true synthetic cell capable of full replication (Figure 1.5d: right panel).[58]

1.3.3 Membraneless compartments

Under critical ionic concentrations, biomolecules displaying a high electrical charge or high multivalency can undergo complex liquid-liquid phase separation, which yields their partition into membraneless compartments known as coacervates (Figure 1.5e: left panel). Coacervates obtained from TXTL mixture are therefore highly enriched in TXTL machinery. There is growing evidence that phase separation plays a major role in the regulation of protein activity, spatial segregation of nucleic acids, and functional organization of the cell.[59–61] Phase separation also gives rise to the emergence of partitioning, confinement, and crowding effects influencing cell-free gene expression.[62–64] TXTL was first combined with hydrogels by Park *et al.*, who employed a micropad of DNA hydrogel incubated in TXTL mixture.[65] The DNA hydrogel, consisting of genes coding for a reporter protein, was not permeable to TXTL machinery, so protein expression occurred at the surface of the gel but not inside. Thiele *et al.* used hyaluronic acid gel beads in which DNA template was covalently attached and incubated the gel beads in the presence of TXTL mixture inside a microdroplet.[66] In contrast to the gel used by Park *et al.*, gel beads used in this study were porous enough to allow TXTL to occur inside the gel beads. The authors showed that transcription and translation reactions were confined inside the gel as the mRNA remained trapped inside the beads. Finally, Zhou *et al.* produced hydrogel particles containing DNA template, ribosomes, and His-tagged TXTL proteins from PURE system. All biomolecules remained trapped inside the hydrogel particles, so TXTL could occur over several days by the continuous supply of feeding buffer (Figure 1.5d: left panel).[67]

1.4 TXTL and molecular communication

Interactions between gut microbiota and the human body recently gained in interest as studies highlighted the impact of microorganisms on the metabolism,[68] immune responses,[69] and the recurrence of cancer.[70] The development of new biomolecular tools inspired by natural quorum sensing systems is therefore of particular relevance in order to understand multicellular communication[24] and to engineer synthetic communication devices.[71,72] Furthermore, to avoid the difficultly of engineering organisms with increasingly complex genetic circuits, the development of orthogonal communication channels that enable distributed functions within a bacterial community has recently received particular attention.[24,73] Nevertheless, the prototyping of synthetic and highly orthogonal molecular communication channels that mediate collective behavior in populations of living cells remains challenging. Compartmentalized TXTL systems can play an important role

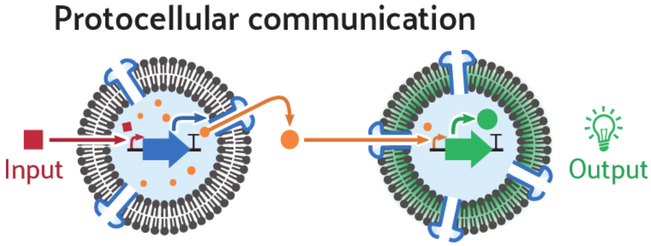


Figure 1.6: Through the exchange of chemical signaling molecules, liposomes containing TXTL mixture and quorum sensing circuits can be used as artificial cells to establish communication with other artificial cells

in the development of such pathways, as they can mimic communication between cells whilst remaining modular and easy to control.

Quorum sensors are naturally occurring genetic communication systems found in many bacterial species in a variety of chemical messenger-transcription factor-operator systems. Quorum sensors are essential building blocks for engineering molecular communication channels in synthetic biology.[24,71] Halleran and Murray recently characterized a series of quorum sensors using lysate-based TXTL, prior to their implementation *in vivo*, enabling them to accurately predict crosstalk between different systems.[74] In addition, Wen *et al*. used lysate-based TXTL reactions to identify the presence of quorum sensing signals in patients suffering from lung infection by *Pseudomonas aeruginosa*, demonstrating the therapeutic relevance of quorum sensing detection and characterization.[75]

Combination of microfluidic technologies with TXTL allows the generation of cell mimics capable of communication processes based on gene instructions. Booth *et al*. engineered a tissue mimic by 3D-printing TXTL-containing droplets into organized layers.[76] The PURE-based TXTL reactions produced α-hemolysin upon light-activation in order to create communication channels between individual droplets in a programmable manner. In a related approach, Findlay *et al*. programmed specific communication between two droplets by implementing the active transport of a signaling molecule through LacY transporter proteins produced *in situ* using PURE-based TXTL reaction.[77] Alternatively, communication was implemented between sender TXTL-liposomes and receiver-proteinosomes.[78] By establishing communication between mammalian and bacterial synthetic cell models (synells), Adamala *et al*. provided the first example of diffusive one-way communication between fully synthetic cells of different organisms (Figure 1.6).[79] Such synells comprised liposomes containing either HeLa or *E. coli* extract, as well as quorum sensing DNA circuits. However, to engineer true collective behavior in populations of synells requires further development of this technology to allow bidirectional exchange of information.

1.5 Conclusion

Cell-free TXTL reactions are becoming an essential tool to prototype novel genetic circuits and can facilitate their integration into living hosts. In combination with microfluidic technologies and the development of novel semipermeable microcapsules, TXTL reactions constitute a unique platform for the study of gene expression and coupling to other cellular processes in a biochemically relevant microenvironment.

Current efforts are exploring possibilities to establish synthetic molecular communication between artificial and natural systems, working towards therapeutic applications wherein information exchange between living cells are mediated or altered. We anticipate that micro-compartmentalized TXTL - by accelerating the characterization of novel genetic circuits and orthogonal molecular communication channels - will have a compelling contribution to the field of synthetic biology.

1.6 Aim of this Thesis

The aim of this thesis is to construct and characterize complex circuitry with interesting dynamic behavior *in vitro* employing micro-compartmentalized reactions.

In order to construct complex circuits in vitro, we mainly utilize cell-free transcription and translation (TXTL) as a biochemical breadboarding environment. Furthermore, topologically complex circuits require a multitude of regulatory components. Synthetic riboregulators that have high sequence flexibility, such as toehold switches, provide a virtually unlimited toolbox of RNA-based regulatory components. Therefore, **Chapter 2** describes the optimization of a TXTL system for RNA-based regulators and the implementation of toehold switches in TXTL.

Synthetic circuits implemented in TXTL bulk reactions can provide valuable information on the function of circuit components. However, most biologically relevant genetic networks display characteristic dynamic behavior that cannot be captured in bulk reactions. In **Chapter 3**, we describe the construction of microfluidic flow reactors to maintain prolonged TXTL reactions. We use these reactors to establish protocols to provide time-varying inputs to genetic circuits and analyze temporal behavior.

Chapter 4 focuses on a specific genetic circuit, the coherent feed-forward loop (CFFL), that was demonstrated to be able to decode temporal signals. The synthetic CFFL circuit makes use of toehold switches and we verify its behavior in TXTL reactions. Subsequently, we employ microfluidic flow reactors to observe the circuit under dynamic conditions and characterize its response to inputs of varying durations.

Building towards more topologically complex circuits, we construct different variants of the CFFL circuit and combine these variants in a composite CFFL. **Chapter 5** describes the construction and thorough characterization of this circuit in TXTL reactions.

Next to topological complexity, more complex functionality can be achieved by distributed circuits. **Chapter 6** describes a different model system that employs DNA strand displacement reactions to perform computations and exchange signals between artificial protocells (BIO-PC). Using light-activated signaling in DNA-encoded sender-receiver architectures, we investigate the length scale of the propagation under various conditions. To reveal the effect of activating synthetic circuits in a distributed fashion, we analyze the behavior of an AND logic gate in a sender-receiver architecture.

1.7 References

(1) Elowitz, M. B., and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature 403,* 335–338.

(2) Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000) Construction of a genetic toggle switch in Escherichia coli. *Nature 403,* 339–342.

(3) Ro, D.-K., Paradise, E. M., Ouellet, M., Fisher, K. J., Newman, K. L., Ndungu, J. M., Ho, K. A., Eachus, R. A., Ham, T. S., Kirby, J., Chang, M. C. Y., Withers, S. T., Shiba, Y., Sarpong, R., and Keasling, J. D. (2006) Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature 440,* 940–943.

(4) Martin, V. J. J., Pitera, D. J., Withers, S. T., Newman, J. D., and Keasling, J. D. (2003) Engineering a mevalonate pathway in Escherichia coli for production of terpenoids. *Nature Biotechnology 21,* 796–802.

(5) Andrews, L. B., Nielsen, A. A. K., and Voigt, C. A. (2018) Cellular checkpoint control using programmable sequential logic. *Science 361,* eaap8987.

(6) Borkowski, O., Ceroni, F., Stan, G.-B., and Ellis, T. (2016) Overloaded and stressed: whole-cell considerations for bacterial synthetic biology. *Current Opinion in Microbiology 33,* 123–130.

(7) Ceroni, F., Algar, R., Stan, G.-B., and Ellis, T. (2015) Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nature Methods 12,* 415–418.

(8) Ceroni, F., Boo, A., Furini, S., Gorochowski, T. E., Borkowski, O., Ladak, Y. N., Awan, A. R., Gilbert, C., Stan, G.-B., and Ellis, T. (2018) Burden-driven feedback control of gene expression. *Nature Methods 15,* 387–393.

(9) Salehi-Reyhani, A., Ces, O., and Elani, Y. (2017) Artificial cell mimics as simplified models for the study of cell biology. *Exp Biol Med (Maywood) 242,* 1309–1317.

(10) Jia, H., Heymann, M., Bernhard, F., Schwille, P., and Kai, L. (2017) Cell-free protein synthesis in micro compartments: building a minimal cell from biobricks. *New Biotechnology 39,* 199–205.

(11) Semenov, S. N., Wong, A. S. Y., van der Made, R. M., Postma, S. G. J., Groen, J., van Roekel, H. W. H., de Greef, T. F. A., and Huck, W. T. S. (2015) Rational design of functional and tunable oscillating enzymatic networks. *Nature Chemistry 7,* 160–165.

(12) Coyle, S. M., and Lim, W. A. (2016) Mapping the functional versatility and fragility of Ras GTPase signaling circuits through in vitro network reconstitution. *eLife* (Cole, P. A., Ed.) *5,* e12435.

(13) Zhang, D. Y., and Seelig, G. (2011) Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry 3,* 103–113.

(14) Zhang, D. Y., and Winfree, E. (2009) Control of DNA Strand Displacement Kinetics Using Toehold Exchange. *J. Am. Chem. Soc. 131,* 17303–17314.

(15) Chen, Y.-J., Groves, B., Muscat, R. A., and Seelig, G. (2015) DNA nanotechnology from the test tube to the cell. *Nature Nanotechnology 10,* 748–760.

(16) Qian, L., and Winfree, E. (2011) Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science 332,* 1196–1201.

(17) Qian, L., Winfree, E., and Bruck, J. (2011) Neural network computation with DNA strand displacement cascades. *Nature 475*, 368–372.

(18) Seelig, G., Soloveichik, D., Zhang, D. Y., and Winfree, E. (2006) Enzyme-Free Nucleic Acid Logic Circuits. *Science 314*, 1585–1588.

(19) Kim, J., White, K. S., and Winfree, E. (2006) Construction of an in vitro bistable circuit from synthetic transcriptional switches. *Molecular Systems Biology 2*, 68.

(20) Kim, J., Khetarpal, I., Sen, S., and Murray, R. M. (2014) Synthetic circuit for exact adaptation and fold-change detection. *Nucleic Acids Research 42*, 6078–6089.

(21) Padirac, A., Fujii, T., and Rondelez, Y. (2012) Bottom-up construction of in vitro switchable memories. *PNAS 109*, E3212–E3220.

(22) Meijer, L. H. H., Joesaar, A., Steur, E., Engelen, W., van Santen, R. A., Merkx, M., and de Greef, T. F. A. (2017) Hierarchical control of enzymatic actuators using DNA-based switchable memories. *Nature Communications 8*, 1117.

(23) Lentini, R., Yeh Martín, N., and Mansy, S. S. (2016) Communicating artificial cells. *Current Opinion in Chemical Biology 34*, 53–61.

(24) Scott, S. R., and Hasty, J. (2016) Quorum Sensing Communication Modules for Microbial Consortia. *ACS Synth. Biol. 5*, 969–977.

(25) Pardee, K., Green, A. A., Ferrante, T., Cameron, D. E., DaleyKeyser, A., Yin, P., and Collins, J. J. (2014) Paper-Based Synthetic Gene Networks. *Cell 159*, 940–954.

(26) Pardee, K., Green, A. A., Takahashi, M. K., Braff, D., Lambert, G., Lee, J. W., Ferrante, T., Ma, D., Donghia, N., Fan, M., Daringer, N. M., Bosch, I., Dudley, D. M., O’Connor, D. H., Gehrke, L., and Collins, J. J. (2016) Rapid, Low-Cost Detection of Zika Virus Using Programmable Biomolecular Components. *Cell 165*, 1255–1266.

(27) Hu, C. Y., Takahashi, M. K., Zhang, Y., and Lucks, J. B. (2018) Engineering a Functional Small RNA Negative Autoregulation Network with Model-Guided Design. *ACS Synth. Biol. 7*, 1507–1518.

(28) Senoussi, A., Lee Tin Wah, J., Shimizu, Y., Robert, J., Jaramillo, A., Findeiss, S., Axmann, I. M., and Estevez-Torres, A. (2018) Quantitative Characterization of Translational Riboregulators Using an in Vitro Transcription–Translation System. *ACS Synth. Biol. 7*, 1269–1278.

(29) Marshall, R., Maxwell, C. S., Collins, S. P., Jacobsen, T., Luo, M. L., Begemann, M. B., Gray, B. N., January, E., Singer, A., He, Y., Beisel, C. L., and Noireaux, V. (2018) Rapid and Scalable Characterization of CRISPR Technologies Using an E. coli Cell-Free Transcription-Translation System. *Molecular Cell 69*, 146-157.e3.

(30) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *PNAS 110*, 15985–15990.

(31) Karzbrun, E., Tayar, A. M., Noireaux, V., and Bar-Ziv, R. H. (2014) Programmable on-chip DNA compartments as artificial cells. *Science 345*, 829–832.

(32) Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., and Maerkl, S. J. (2015) Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* (Simmel, F., Ed.) *4*, e09771.

(33) Del Vecchio, D., Dy, A. J., and Qian, Y. (2016) Control theory meets synthetic biology. *Journal of The Royal Society Interface 13*, 20160380.

(34) Borkowski, O., Bricio, C., Murgiano, M., Rothschild-Mancinelli, B., Stan, G.-B., and Ellis, T. (2018) Cell-free prediction of protein expression costs for growing cells. *Nature Communications 9*, 1457.

(35) Moore, S. J., MacDonald, J. T., Wienecke, S., Ishwarbhai, A., Tsipa, A., Aw, R., Kylilis, N., Bell, D. J., McClymont, D. W., Jensen, K., Polizzi, K. M., Biedendieck, R., and Freemont, P. S. (2018) Rapid acquisition and model-based analysis of cell-free transcription–translation reactions from nonmodel bacteria. *PNAS 115*, E4340–E4349.

(36) Egbert, R. G., and Klavins, E. (2012) Fine-tuning gene networks using simple sequence repeats. *PNAS 109*, 16817–16822.

(37) Shin, J., and Noireaux, V. (2010) Study of messenger RNA inactivation and protein degradation in an Escherichia coli cell-free expression system. *Journal of Biological Engineering 4*, 9.

(38) Garamella, J., Marshall, R., Rustad, M., and Noireaux, V. (2016) The All E. coli TX-TL Toolbox 2.0: A Platform for Cell-Free Synthetic Biology. *ACS Synth. Biol. 5*, 344–355.

(39) Marshall, R., Maxwell, C. S., Collins, S. P., Beisel, C. L., and Noireaux, V. (2017) Short DNA containing χ sites enhances DNA stability and gene expression in E. coli cell-free transcription–translation systems. *Biotechnology and Bioengineering 114*, 2137–2141.

(40) Maddalena, L. L. de, Niederholtmeyer, H., Turtola, M., Swank, Z. N., Belogurov, G. A., and Maerkl, S. J. (2016) GreA and GreB Enhance Expression of Escherichia coli RNA Polymerase Promoters in a Reconstituted Transcription–Translation System. *ACS Synth. Biol. 5*, 929–935.

(41) Shi, X., Wu, T., M. Cole, C., K. Devaraj, N., and Joseph, S. (2018) Optimization of ClpXP activity and protein synthesis in an E. coli extract-based cell-free expression system. *Scientific Reports 8*, 3488.

(42) Karim, A. S., Heggestad, J. T., Crowe, S. A., and Jewett, M. C. (2018) Controlling cell-free metabolism through physiochemical perturbations. *Metabolic Engineering 45*, 86–94.

(43) Yeung, E., Dy, A. J., Martin, K. B., Ng, A. H., Del Vecchio, D., Beck, J. L., Collins, J. J., and Murray, R. M. (2017) Biophysical Constraints Arising from Compositional Context in Synthetic Gene Networks. *Cell Systems 5*, 11-24.e12.

(44) Buxboim, A., Bar-Dagan, M., Frydman, V., Zbaida, D., Morpurgo, M., and Bar-Ziv, R. (2007) A Single-Step Photolithographic Interface for Cell-Free Gene Expression and Active Biochips. *Small 3*, 500–510.

(45) Isalan, M., Lemerle, C., and Serrano, L. (2005) Engineering Gene Networks to Emulate Drosophila Embryonic Pattern Formation. *PLOS Biology 3*, e64.

(46) Tayar, A. M., Karzbrun, E., Noireaux, V., and Bar-Ziv, R. H. (2015) Propagating gene expression fronts in a one-dimensional coupled system of artificial cells. *Nature Physics 11*, 1037–1041.

(47) Tayar, A. M., Karzbrun, E., Noireaux, V., and Bar-Ziv, R. H. (2017) Synchrony and pattern formation of coupled genetic oscillators on a chip of artificial cells. *PNAS 114*, 11609–11614.

(48) Pardatscher, G., Schwarz-Schilling, M., Daube, S. S., Bar-Ziv, R. H., and Simmel, F. C. (2018) Gene Expression on DNA Biochips Patterned with Strand-Displacement Lithography. *Angewandte Chemie International Edition 57,* 4783–4786.

(49) Hori, Y., Kantak, C., Murray, R. M., and Abate, A. R. (2017) Cell-free extract based optimization of biomolecular circuits with droplet microfluidics. *Lab Chip 17,* 3037–3042.

(50) Guan, Y., Li, Z., Wang, S., Barnes, P. M., Liu, X., Xu, H., Jin, M., Liu, A. P., and Yang, Q. (2018) A robust and tunable mitotic oscillator in artificial cells. *eLife* (Barkai, N., Ed.) *7,* e33549.

(51) Sakamoto, R., Noireaux, V., and Maeda, Y. T. (2018) Anomalous Scaling of Gene Expression in Confined Cell-Free Reactions. *Scientific Reports 8,* 7364.

(52) Hansen, M. M. K., Paffenholz, S., Foschepoth, D., Heus, H. A., Thiele, J., and Huck, W. T. S. (2016) Cell-Like Nanostructured Environments Alter Diffusion and Reaction Kinetics in Cell-Free Gene Expression. *ChemBioChem 17,* 228–232.

(53) Sokolova, E., Spruijt, E., Hansen, M. M. K., Dubuc, E., Groen, J., Chokkalingam, V., Piruska, A., Heus, H. A., and Huck, W. T. S. (2013) Enhanced transcription rates in membrane-free protocells formed by coacervation of cell lysate. *PNAS 110,* 11692–11697.

(54) Norred, S. E., Caveney, P. M., Chauhan, G., Collier, L. K., Collier, C. P., Abel, S. M., and Simpson, M. L. (2018) Macromolecular Crowding Induces Spatial Correlations That Control Gene Expression Bursting Patterns. *ACS Synth. Biol. 7,* 1251–1258.

(55) Noireaux, V., and Libchaber, A. (2004) A vesicle bioreactor as a step toward an artificial cell assembly. *PNAS 101,* 17669–17674.

(56) Majumder, S., Garamella, J., Wang, Y.-L., DeNies, M., Noireaux, V., and Liu, A. P. (2017) Cell-sized mechanosensitive and biosensing compartment programmed with DNA. *Chem. Commun. 53,* 7349–7352.

(57) Krinsky, N., Kaduri, M., Zinger, A., Shainsky-Roitman, J., Goldfeder, M., Benhar, I., Hershkovitz, D., and Schroeder, A. (2018) Synthetic Cells Synthesize Therapeutic Proteins inside Tumors. *Advanced Healthcare Materials 7,* 1701163.

(58) van Nies, P., Westerlaken, I., Blanken, D., Salas, M., Mencía, M., and Danelon, C. (2018) Self-replication of DNA by its encoded proteins in liposome-based synthetic cells. *Nature Communications 9,* 1583.

(59) Banani, S. F., Lee, H. O., Hyman, A. A., and Rosen, M. K. (2017) Biomolecular condensates: organizers of cellular biochemistry. *Nature Reviews Molecular Cell Biology 18,* 285–298.

(60) Alberti, S. (2017) Phase separation in biology. *Current Biology 27,* R1097–R1102.

(61) Boeynaems, S., Alberti, S., Fawzi, N. L., Mittag, T., Polymenidou, M., Rousseau, F., Schymkowitz, J., Shorter, J., Wolozin, B., Van Den Bosch, L., Tompa, P., and Fuxreiter, M. (2018) Protein Phase Separation: A New Phase in Cell Biology. *Trends in Cell Biology 28,* 420–435.

(62) Sokolova, E., Spruijt, E., Hansen, M. M. K., Dubuc, E., Groen, J., Chokkalingam, V., Piruska, A., Heus, H. A., and Huck, W. T. S. (2013) Enhanced transcription rates in membrane-free protocells formed by coacervation of cell lysate. *PNAS 110,* 11692–11697.

(63) Torre, P., Keating, C. D., and Mansy, S. S. (2014) Multiphase Water-in-Oil Emulsion Droplets for Cell-Free Transcription–Translation. *Langmuir 30,* 5695–5699.

(64) Tang, T.-Y. D., Swaay, D. van, deMello, A., Anderson, J. L. R., and Mann, S. (2015) In vitro gene expression within membrane-free coacervate protocells. *Chem. Commun. 51,* 11429–11432.

(65) Park, N., Um, S. H., Funabashi, H., Xu, J., and Luo, D. (2009) A cell-free protein-producing gel. *Nature Materials 8,* 432–437.

(66) Thiele, J., Ma, Y., Foschepoth, D., Hansen, M. M. K., Steffen, C., Heus, H. A., and Huck, W. T. S. (2014) DNA-functionalized hydrogels for confined membrane-free in vitro transcription/translation. *Lab Chip 14,* 2651–2656.

(67) Zhou, X., Wu, H., Cui, M., Lai, S. N., and Zheng, B. (2018) Long-lived protein expression in hydrogel particles: towards artificial cells. *Chem. Sci. 9,* 4275–4279.

(68) Brown, J. M., and Hazen, S. L. (2018) Microbial modulation of cardiovascular disease. *Nature Reviews Microbiology 16,* 171–181.

(69) Clemente, J. C., Ursell, L. K., Parfrey, L. W., and Knight, R. (2012) The Impact of the Gut Microbiota on Human Health: An Integrative View. *Cell 148,* 1258–1270.

(70) Bullman, S., Pedamallu, C. S., Sicinska, E., Clancy, T. E., Zhang, X., Cai, D., Neuberg, D., Huang, K., Guevara, F., Nelson, T., Chipashvili, O., Hagan, T., Walker, M., Ramachandran, A., Diosdado, B., Serna, G., Mulet, N., Landolfi, S., Cajal, S. R. y, Fasani, R., Aguirre, A. J., Ng, K., Élez, E., Ogino, S., Tabernero, J., Fuchs, C. S., Hahn, W. C., Nuciforo, P., and Meyerson, M. (2017) Analysis of Fusobacterium persistence and antibiotic response in colorectal cancer. *Science 358,* 1443–1448.

(71) Din, M. O., Danino, T., Prindle, A., Skalak, M., Selimkhanov, J., Allen, K., Julio, E., Atolia, E., Tsimring, L. S., Bhatia, S. N., and Hasty, J. (2016) Synchronized cycles of bacterial lysis for in vivo delivery. *Nature 536,* 81–85.

(72) Kim, S., Kerns, S. J., Ziesack, M., Bry, L., Gerber, G. K., Way, J. C., and Silver, P. A. (2018) Quorum Sensing Can Be Repurposed To Promote Information Transfer between Bacteria in the Mammalian Gut. *ACS Synth. Biol. 7,* 2270–2281.

(73) Kylilis, N., Tuza, Z. A., Stan, G.-B., and Polizzi, K. M. (2018) Tools for engineering coordinated system behaviour in synthetic microbial consortia. *Nature Communications 9,* 2677.

(74) Halleran, A. D., and Murray, R. M. (2018) Cell-Free and In Vivo Characterization of Lux, Las, and Rpa Quorum Activation Systems in E. coli. *ACS Synth. Biol. 7,* 752–755.

(75) Wen, K. Y., Cameron, L., Chappell, J., Jensen, K., Bell, D. J., Kelwick, R., Kopniczky, M., Davies, J. C., Filloux, A., and Freemont, P. S. (2017) A Cell-Free Biosensor for Detecting Quorum Sensing Molecules in P. aeruginosa-Infected Respiratory Samples. *ACS Synth. Biol. 6,* 2293–2301.

(76) Booth, M. J., Schild, V. R., Graham, A. D., Olof, S. N., and Bayley, H. (2016) Light-activated communication in synthetic tissues. *Science Advances 2,* e1600056.

(77) Findlay, H. E., Harris, N. J., and Booth, P. J. (2016) In vitro synthesis of a Major Facilitator Transporter for specific active transport across Droplet Interface Bilayers. *Scientific Reports 6,* 39349.

(78) Tang, T.-Y. D., Cecchi, D., Fracasso, G., Accardi, D., Coutable-Pennarun, A., Mansy, S. S., Perriman, A. W., Anderson, J. L. R., and Mann, S. (2018) Gene-Mediated

Chemical Communication in Synthetic Protocell Communities. *ACS Synth. Biol. 7*, 339–346.

(79) Adamala, K. P., Martin-Alarcon, D. A., Guthrie-Honea, K. R., and Boyden, E. S. (2017) Engineering genetic circuit interactions within and between synthetic minimal cells. *Nature Chemistry 9*, 431–439.

(80) Schwarz-Schilling, M., Aufinger, L., Mückl, A., and Simmel, F. C. (2016) Chemical communication between bacteria and cell-free gene expression systems within linear chains of emulsion droplets. *Integr. Biol. 8*, 564–570.

(81) Lentini, R., Martín, N. Y., Forlin, M., Belmonte, L., Fontana, J., Cornella, M., Martini, L., Tamburini, S., Bentley, W. E., Jousson, O., and Mansy, S. S. (2017) Two-Way Chemical Communication between Artificial and Natural Cells. *ACS Cent. Sci. 3*, 117–123.

(82) Shin, J., and Noireaux, V. (2012) An E. coli Cell-Free Expression Toolbox: Application to Synthetic Gene Circuits and Artificial Cells. *ACS Synth. Biol. 1*, 29–41.

2

Cell-Free Transcription and Translation for the Construction of RNA-based Circuits

Part of this work has been published as:

Pascal A. Pieters, Bryan L. Nathalia, Ardjan J. van der Linden, Peng Yin, Jongmin Kim, Wilhelm T. S. Huck, and Tom F. A. de Greef. *ACS Synthetic Biology*, **10**, 1406-1416 (2021)

2.1 Introduction

When implementing synthetic networks of increasing sizes, undesired interactions with host organism machinery and excessive load on the host can impede the function of synthetic circuits.[1] The use of *in vitro* transcription and translation (TXTL) reactions eliminates the need for a host organism and provides a biomolecular breadboarding environment to rapidly construct synthetic genetic networks.[2–5]

TXTL reaction mixtures can be fabricated using various methods, which can be divided into two classes: reconstituted (PURE) TXTL and lysate-based TXTL. PURE TXTL reactions were developed by Shimizu *et al.*, who purified the proteins and ribosome essential for transcription and translation and used these components to reconstitute a reaction mixture capable of transcription and translation.[6] The use of purified reagents results in a well-defined environment but renders the production of the reaction mixture costly and time-consuming. Recently, this has partially been mitigated through the formulation of protocols that describe the purification of all proteins from a single mixed coculture.[7,8] To simplify the composition of the reactions and increase protein expression yield, PURE TXTL reactions typically make use of the T7 phage-derived RNA polymerase (T7RNAP). In contrast to the multi-subunit native E. coli RNA polymerase, the T7RNAP is not reliant on a transcription factor or sigma-factor to initiate transcription, disqualifying a range of E. coli regulators from being used in PURE TXTL, unless the E. coli RNAP is introduced through different means.[9]

Lysate-based TXTL reaction systems contain crude cell lysate that contains the transcription and translation machinery instead of the purified components and ribosomes. Lysate TXTL reactions typically consist of three main components: the *E. coli* lysate, the energy mixture, and DNA constructs encoding the genetic circuits. To produce the lysate for the TXTL systems, cells are lysed through sonication,[10] freeze-thaw cycles,[11] bead-beating[4] or homogenization.[12] Bead-beating and homogenization using a cell press have been used extensively by Noireaux et al., who developed a toolbox of genetic components that can be used in combination with lysate-based TXTL systems. Homogenization required a less time-intensive protocol and resulted in highly potent TXTL mixtures,[12] but bead-beating resulted in less batch-to-batch variability.[5] On the other hand, sonication protocols lead to an increased temperature during cell lysis, degrading the TXTL components, and lysis using freeze-thaw cycles has not been extensively characterized as a platform for synthetic genetic circuits.

TXTL reactions provide a flexible environment to dynamically vary inputs and circuit parameters without the need for extensive bacterial cloning and culture cycles.[3] A toolbox of genetic elements has been created through extensive characterization of the TXTL reaction mixture alongside E. coli and phage-derived transcriptional regulators.[2,5] This TXTL toolbox has been successfully utilized to rapidly construct and study gene cascades,[2] incoherent feed-forward loops,[2,13] a negative feedback loop,[14] and oscillators.[15–18] Although the toolbox extends the range of available genetic elements through the inclusion of native E. coli transcription factors, which would severely interfere with an E. coli host cell when utilized *in vivo*, the number of available regulatory elements is still limited and does not scale up easily for the construction of topologically complex genetic networks. To resolve this, an additional regulatory layer can be introduced by utilizing post-transcriptional interactions such as riboswitches. In bacteria, small RNAs form a class of regulators that extend the complexity of genetic networks beyond transcriptional regulation.[19]

2.2 Toehold Switches

For synthetic circuits, toehold switch riboregulators, which can be forward-engineered, provide a wide dynamic range, and are highly orthogonal, offer the potential to construct larger synthetic genetic networks.[20–22] Toehold switches resemble naturally occurring trans-acting riboregulators that control translation by blocking the ribosome binding site (RBS) using an RNA stem-loop. The loop can be unfolded through toehold-mediated strand displacement with a trans-acting RNA trigger (Figure 2.1a). The forward-engineered toehold switches displayed a high dynamic range (Figure 2.1b), could be designed to be highly orthogonal and adapted to activation by the desired RNA sequence. This flexibility was achieved by excluding the fixed RBS and start codon sequences from the duplexing nucleotides in the RNA stem-loop, creating freedom in the choice of RNA trigger sequence. As a consequence, the trigger sequence could be adapted to biologically relevant sequences and detect biomarkers. Pardee *et al.* combined paper-based cell-free TXTL reactions with forward-engineered toehold switches to detect the Ebola virus and distinguish between different strains of the virus.[23] Similarly, Pardee *et al.* developed a Zika virus sensor based on toehold switches, which they

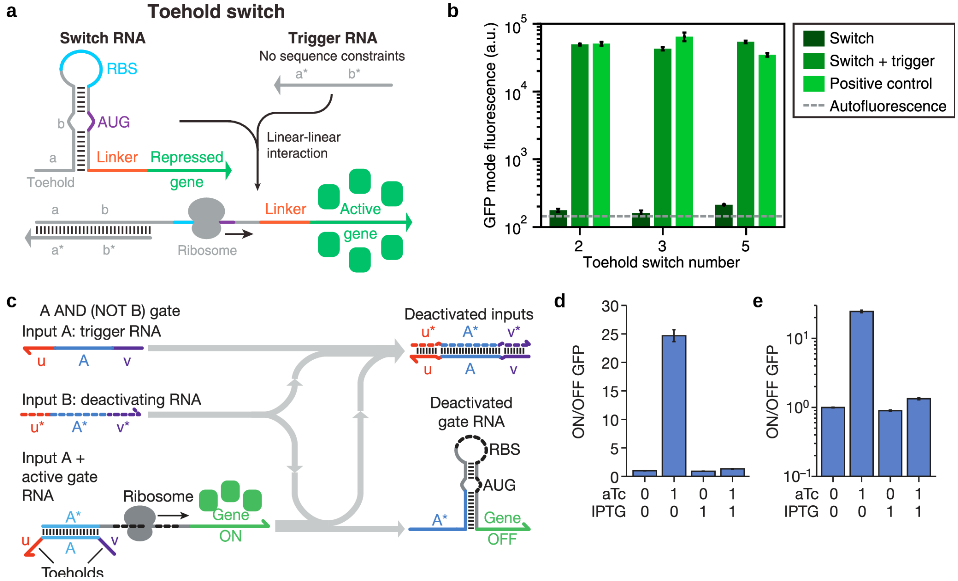


Figure 2.1: Toehold switches and ribocomputing. a) Schematic of a toehold switch and trigger forming the active riboregulator complex. The variable, trigger-binding sequence of the switch is shown in gray, where the toehold that initiates strand displacement is indicated as region a. Constrained and functional sequences are shown in various colors. Reprinted with permission from Green et al.[20] *b) The GFP fluorescence levels of three toehold switches expressed in vivo. Error bars represent the standard deviation of three or more biological replicates. Reprinted with permission from Green et al.*[20] *c) Schematic of the extension of the toehold switch design to perform a negating operation (NOT gate). The complete system performs an A AND (NOT B) operation with respect to RNA inputs A and B. Reprinted with permission from Green et al.*[26] *d) and e) Linear and logarithmic GFP output fluorescence for combinations of the two inputs, measured using flow cytometry. Reprinted with permission from Green et al.*[26]

augmented with an isothermal RNA amplification technique (nucleic acid sequence-based amplification; NASBA) and CRISPR/Cas9-based target cleavage to accurately detect different Zika virus strains in biological samples.[24] Later, Takahashi and coworkers extended this method to detect 16S ribosomal RNA (rRNA) of various gut microbia and analyze fecal samples.[25]

In addition to its use as a biosensor, toehold switches have repeatedly been demonstrated to have great potential in designing synthetic circuits that display complex molecular computations. This application was consolidated by the development of a range of ribocomputing devices based on the initial toehold switch design, comprising AND, OR, and NOT logic toehold switches.[26] NOT logic toehold switches are similar to the original toehold switch and trigger design, with additional toehold regions at both trigger ends. These toehold regions were used to initiate strand displacement of the RNA trigger by an orthogonal, deactivating, input RNA trigger (Figure 2.1c). Expression of both triggers resulted in severe repression of the output gene compared to expression of only the activating trigger (Figure 2.1d). Kim *et al.* further expanded the scope of toehold switch-based logic devices by developing toehold switches with a sharp response to inputs through the introduction of an inhibitory hairpin before the RNA stem-loop that contains the RBS.[27]

The flexibility in choice of toehold switch sequence and orthogonality between switches provides a basis for developing topologically complex synthetic genetic circuits. In contrast to toehold switches, novel protein-based regulators cannot readily be developed, and their availability is limited. Therefore, the use of toehold switches in the flexible circuit prototyping environment of TXTL will enable the development of large and complex synthetic genetic circuits. However, toehold switches have not been extensively characterized in combination with the cell-free TXTL toolbox developed by Noireaux and coworkers.[2] Lehr *et al.* demonstrated that AND logic gates can be created in TXTL by controlling gene expression on the transcription level using small transcription activating RNAs (STARs) and on the translation level using toehold switches. [28,29] However, the behavior of the toehold switches was not extensively quantified. In addition, a more straightforward AND gate design can be achieved using a toehold switch, since individual control over the expression of the switch and trigger creates AND-like behavior. This design could be further extended to multi-input AND gates using the ribocomputer designs by Green and coworkers.[26] Therefore, we optimized the cell-free TXTL system developed by Noireaux and coworkers to accommodate expression regulation using toehold switches. A method was implemented for rapid construction of toehold switch and trigger DNA construct for use in TXTL, and we demonstrated the desired functionality of toehold switches in TXTL. To show that more complex ribocomputing devices can be utilized in TXTL, we implemented a toehold switch-based NOT logic gate using cell-free TXTL reactions.

2.3 TXTL Optimization

To establish efficient post-transcriptional regulation using toehold switches, and potentially other RNA-based regulators, we considered the various components of the TXTL mixture for possible interfering factors. We identified that RNA hybridization and degradation are the most prominent factors affecting toehold switch functioning in TXTL reactions. Similar to DNA hybridization, RNA-RNA hybridization is influenced by presence of salts, particularly sodium and magnesium ions.[30] These salts were present in the TXTL reaction buffer, and their concentrations could be adjusted to facilitate RNA strand displacement reactions better. Nevertheless, these concentrations also influence transcription and translation rates, in addition to transcription factor activity.[5] Similarly, some crowding agents have been demonstrated to stabilize RNA hairpin formation, particularly high-molecular-weight polyethylene glycol (PEG),[31] while also affecting transcription and translation. Therefore, we optimized TXTL expression yield by sampling various salt and PEG-8000 concentrations to determine an optimal window of expression yields (Figure 2.2a: BL21 Rosetta 2). Moreover, naturally occurring trans-acting sRNA regulators are often mediated by the RNA chaperone protein Hfq.[32] Since toehold switches were not designed to utilize the Hfq chaperone, we assume the protein is not required for efficient toehold switch activation, but the addition of Hfq to TXTL reactions might be beneficial for other RNA regulators.

RNA degradation is an essential modality in all RNA-based regulatory components. RNAse E is the most prominent endoribonuclease in bacteria, and is responsible for the majority of RNA degradation and RNA processing.[33] In cells, RNAse E exhibits autoregulation through degradation of its own mRNA (*rne*), resulting in tight regulation of RNAse activity with respect to overall RNA levels.[34] Nevertheless, E. coli strains with reduced RNAse E activity have been engineered through mutation of the *rne* gene to prevent the formation of a degradosome, but retaining essential RNA processing activity.[35] Toehold switches were shown to behave comparably in RNAse E deficient strains compared to wildtype RNAse E strains.[20] When moving to cell-free TXTL, however, the degradation rate of RNA does not decrease by a similar factor as other reaction rates. The global mRNA mean lifetime increases by a factor of 1.65, whereas the rate of translation from mRNA decreases ~10x.[2] Lower levels of RNAse E activity can potentially compensate for this bias. We, therefore, constructed an RNAse E deficient E. coli strain suitable for the production of lysate for TXTL reactions. To achieve this, we used the BL21 STAR (DE3) strain, which harbors the RNAse E mutation that leads to lower ribonuclease activity, and transformed it with the pRARE vector obtained from BL21 Rosetta cells. This vector supplies tRNAs for codons that are rare in E. coli, which enables the expression of a broader range of proteins without codon optimization, and transfers chloramphenicol resistance to the strain, which aides

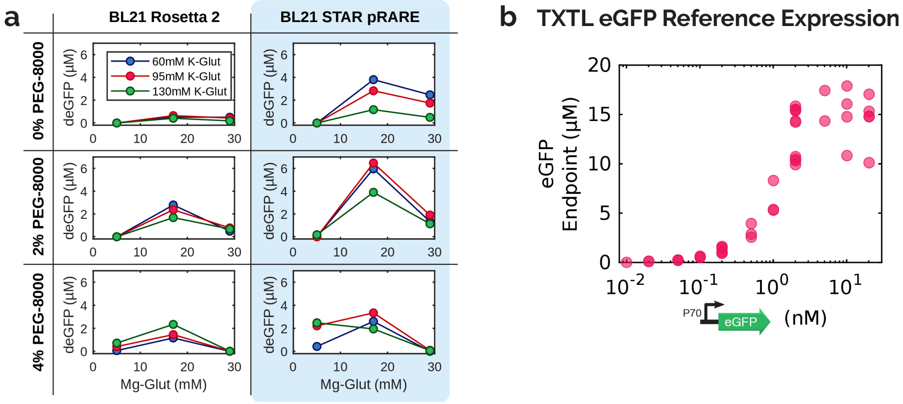


Figure 2.2: a) Expression yields after 6 h expression of 20 nM P70a-eGFP DNA construct using wildtype (BL21 Rosetta 2) and RNAse E deficient (BL21 STAR pRARE) lysate with various salt and crowding agent conditions. Each reaction contained at least 4.7 mM Magnesium L-glutamate (Mg-Glut) and 50 mM Potassium L-glutamate, which is present in the cell lysate solution (see Methods). Stock solutions of 200 mM Mg-Glut, 1 M K-Glut, and 40% PEG-8000 were used to supplement the reaction components to the final concentrations denoted in the figure. b) Endpoint eGFP output concentrations after 14 h of expression of a titration of the P70a-eGFP DNA construct, which functions as an expression positive control construct. Expression from this construct plateaus for DNA concentrations above ~5 nM.

initial culturing steps in the lysate preparation protocol. We produced lysate using a homogenizer according to standard protocols and performed a salt and crowding agent scan (Figure 2.2a: BL21 STAR pRARE). The results reveal that the expression yields of this new strain outperformed the lysate with wildtype RNAse E, more so than typical batch-to-batch variability.

Next, we varied the concentration of the reference eGFP construct to determine a working range for subsequent TXTL experiments (Figure 2.2b). The eGFP expression increased for increasing DNA concentrations, reaching a plateau around 5 nM. This observation is congruent with previous studies that use the same promoter.[2,36]

2.4 Toehold Switches in TXTL

Toehold switches provide a great potential to construct topologically complex genetic networks *in vitro* using TXTL. To aid the rapid design of DNA constructs that combine toehold switches and triggers with various promoters and coding sequences (CDS), we employed a modular and flexible cloning method based on Golden Gate Assembly (GGA).[3] This method makes use of five unique, four nucleotide, overhang sequences to assemble five fragments into a plasmid: the plasmid backbone, promoter, 5’ untranslated region (UTR), CDS and terminator. The overhangs were created through restriction with BsaI, which cleaves outside of its recognition sequence, leaving only a four-base scar sequence between each fragment. Furhtermore, since the overhang sequence between the UTR and CDS and between the CDS and terminator include the start and stop codon, respectively, the effective scar between those regions was reduced further.

Using this cloning method, we constructed toehold switch and trigger DNA constructs with the pBEST backbone, lambda phage OR2-OR1-PR (P70a) promoter, deGFP reporter protein[5] and T500 terminator (pBEST-P70a-Trigger-T500 and pBEST-P70a-Switch-deGFP-T500). Linear DNA templates for expression in TXTL were created from the assembled vectors through PCR. The switch construct was expressed in TXTL both in the presence and absence of trigger construct, which resulted in only slight activation (~2x) and a high background expression without trigger construct (Figure 2.3: P70a-Switch$_{\text{initial}}$). We hypothesized that the high leakage was caused by suboptimal folding of the toehold switch stem-loop or the translation initiation from the start codon introduced by the GGA cloning method. Therefore, toehold switch constructs lacking the 5’UTR cloning scar and alternative start codon were created and compared to the initial switch design (Figure 2.3: P70a-Switch$_{\text{optimized}}$). The updated switch design indeed decreased background expression (OFF state) and increased activated (ON state) expression yields, possibly caused by increased RNA stability due to changes in the 5’UTR. The toehold switch was further characterized in TXTL by varying the concentration of switch and trigger DNA (Figure 2.4). A monotonal increase was observed for an increase in both trigger and switch DNA concentrations, resulting in behavior that resembles an AND gate. This set of DNA constructs provided a good starting point to implement more complex synthetic genetic networks based on toehold switches.

The new design scheme was integrated into the GGA-based cloning method by defining new overhang sequences that omit the start codon (Figure 2.5). Additionally, we treat the promoter and switch combination as one fragment, which decreases the flexibility of the method, but is a straightforward method to eliminate the cloning scar at a crucial position for expression efficiency. Moreover, we uti-

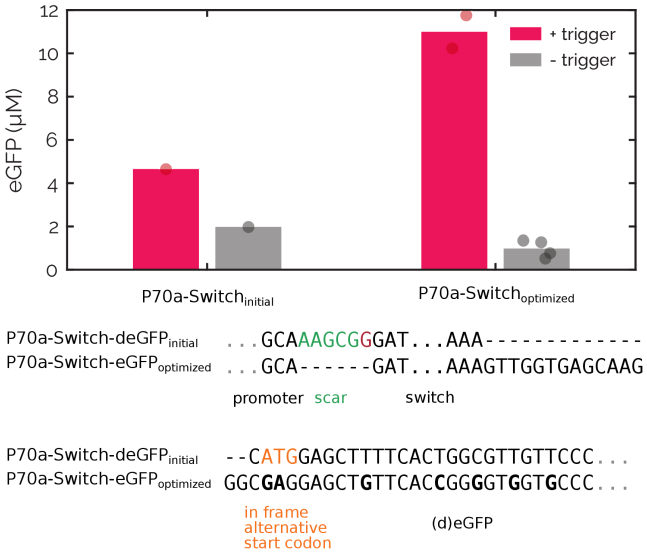


Figure 2.3: Comparison of the expression levels (see Methods) and sequence of an initial toehold switch-bearing DNA construct design and an optimized version. The differences in sequence at the promoter-switch transition and switch-CDS transition are displayed using sequence alignment at the aforementioned positions. Deletions are shown as a dash (-) and mutations are displayed in bold. The most notable changes are the deletion of a cloning scar between the promoter and switch and the omission of the start codon at the start of the eGFP sequence, which only leaves the toehold switch-regulated start codon for initiation of translation. The expression levels with (pink) and without (gray) toehold trigger encoding DNA construct are shown in the graph. The bars represent the average expression after 14 hours, and the points represent individual experiments. In all experiments 10 nM switch construct and 10 nM trigger DNA constructs were used (seeSupplementary Table 2.1).

lized the new cloning scheme to construct a reference eGFP expression construct that is highly similar to the P70a-Switch-eGFP construct but lacks the upstream side of the stem-loop. Interestingly, this construct produced high eGFP expression levels in our TXTL system, almost two times higher than the already optimized pTXTL-P70a-deGFP construct (data not shown).[5]

2.5 Exploring more components in TXTL

To extend the toolbox of functional components in our TXTL system and to demonstrate the practicality of the modular cloning method, we implemented a set

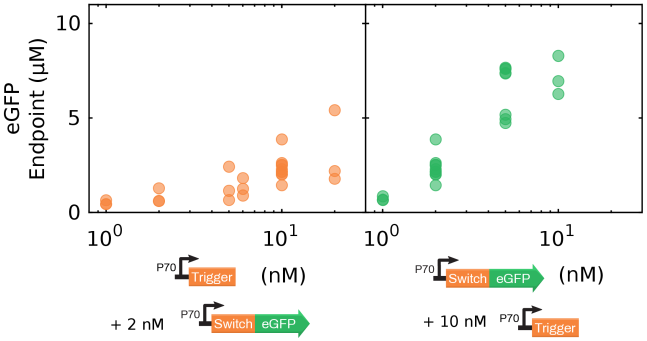


Figure 2.4: Endpoint eGFP concentrations (see Methods) for a titration of trigger (left) and switch DNA constructs (right). Both experiments display an increase in eGFP output for increasing concentrations of the DNA constructs, reaching expression levels close to the positive control (Figure 2.2b). N=3 for all experiments, except 10 nM trigger DNA + 2 nM Switch-eGFP DNA, for which N=10, and 10 nM trigger DNA + 5 nM Switch-eGFP DNA, for which N=8. All DNA constructs and concentrations are summarized in Supplementary Table 2.1.

of additional genetic components *in vitro*. Firstly, we constructed a genetic cascade using the σ^{28} factor. Sigma factors are bacterial proteins that recognize DNA motifs and bind the RNA polymerase apoenzyme to initiate transcription. The housekeeping σ^{70} factor is abundantly present in TXTL lysate, which results in constitutive expression from its cognate promoters such as the previously utilized P70a promoter. Promoters recognized by other sigma factors are not intrinsically active in TXTL reactions, enabling these factors - such as σ^{28} - to function as transcription initiation factor.

Various synthetic genetic networks have been implemented in TXTL using the σ^{28} factor,[2,15,18] making it a good candidate for testing the feasibility of implementing transcription factor-based genetic networks in our TXTL system. A σ^{28} deGFP reporter construct using the pTar (P28a) promoter (pTXTL-P28a-UTR1-deGFP-T500) was obtained from Arbor Biosciences. Additionally, a σ^{28} factor expression construct under control of the toehold switch was created using our GGA-based cloning scheme (pBEST-P70a-Switch-S28-T500). The resulting genetic cascade only produced eGFP output fluorescence when the cognate trigger for the toehold switch controlling the sigma factor was present (Figure 2.6a). When we sampled the σ^{28} factor DNA construct concentration (DNA Y1), we found an increase in

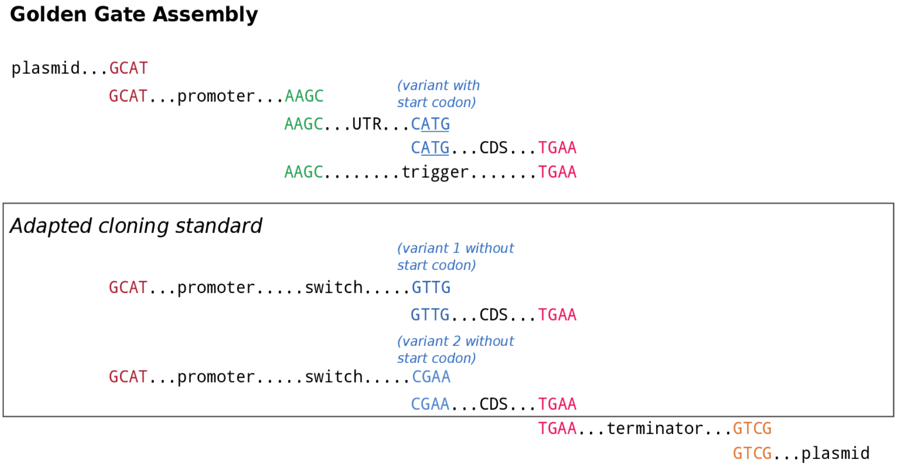

Golden Gate Assembly
plasmid...GCAT
GCAT...promoter...AAGC
(variant with start codon)
AAGC...UTR...CATG
CATG...CDS...TGAA
AAGC.......trigger.......TGAA
Adapted cloning standard
(variant 1 without start codon)
GCAT...promoter.....switch.....GTTG
GTTG...CDS...TGAA
(variant 2 without start codon)
GCAT...promoter.....switch.....CGAA
CGAA...CDS...TGAA
TGAA...terminator...GTCG
GTCG...plasmid

Figure 2.5: Schematic of the type of DNA components and their combinations to create the relevant DNA constructs.[1] *The 5- to 3-prime four nucleotide Golden Gate Assembly (GGA) overhang sequences are shown in distinct colors, whereas the construct types (plasmid, promoter, UTR, switch, trigger, CDS, terminator) are displayed in black in between the overhangs. The start codon in the concerned overhang sequence is underlined. Two variants of the blue overhang sequence were used in combination with the toehold switches to decrease background expression by omitting the obsolete start codon in the original sequence.*

output when more σ^{28} was produced, accompanied by a faster increase in output (Figure 2.6b). Taken together, we have demonstrated the feasibility of integrating toehold switches with the TXTL toolbox of genetic components and characterized a simple cascade circuit based on a toehold switch.

Toehold switches are capable of performing ribocomputing operations more complex than the activation of a gene, which provides the potential to implement a larger variety of genetic networks.[26] Therefore, we extended the toehold switch implementation in TXTL to include a NOT logic gate. The NOT gate makes use of an additional RNA trigger that can displace an activating trigger from the toehold switch and sequester it. Hereto, the previously used trigger construct was padded with toehold regions, according to the design by Green and coworkers.[26] A complementary NOT trigger construct was created and activating and deactivating triggers for a different toehold switch sequence were constructed as off-target controls. When expressed in TXTL reactions, we observed that the newly created trigger with additional toehold regions was similarly potent in activating the toehold switch as the original trigger construct (Figure 2.7a). As expected, the cognate

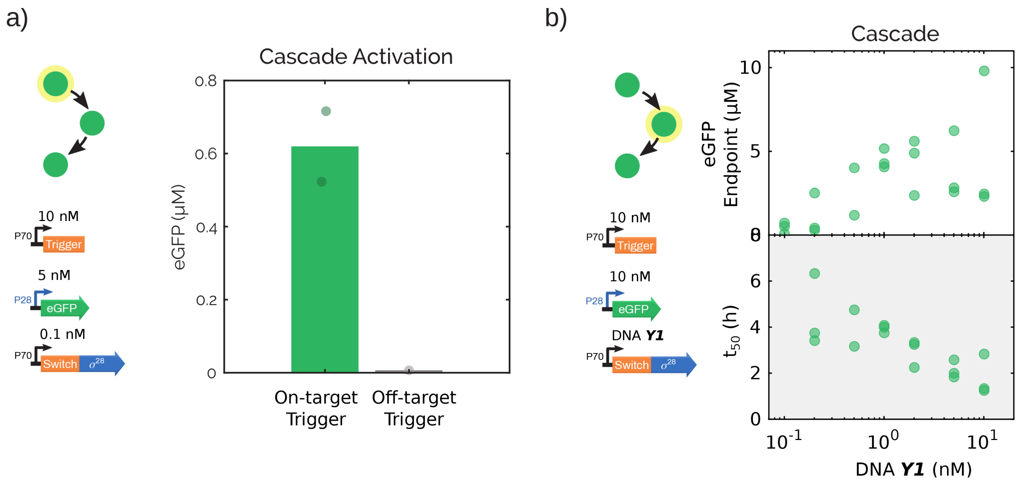
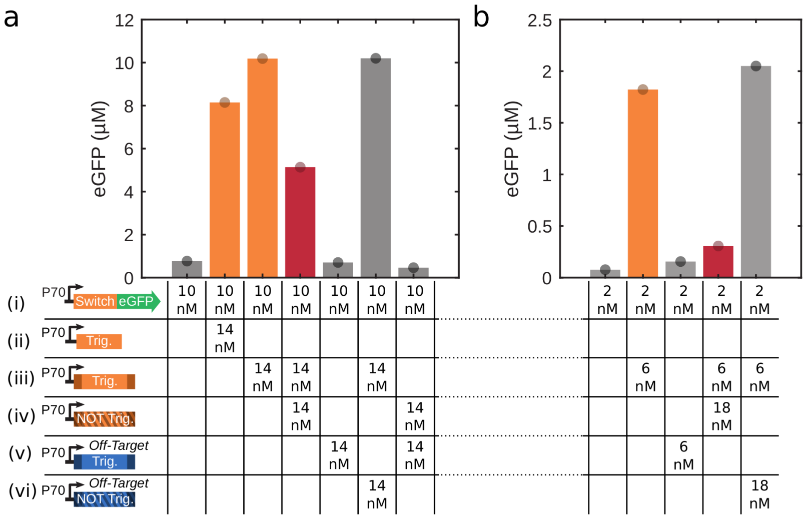


Figure 2.6: a) Endpoint expression (see Methods) of the RNA cascade consisting of a toehold trigger (10 nM) that activates a toehold switch regulating the translation of σ^{28} (0.1 nM), which activates the transcription of an eGFP output gene (5 nM). The cascade produced eGFP when an on-target toehold trigger was used, while expression was absent when an off-target trigger was used instead (N=2). b) The endpoint eGFP concentration (see Methods) and time to half of the maximum output (t_{50}) for a range of σ^{28} construct concentrations (N=3). Expression levels increased for increasing σ^{28} DNA concentrations, while the t_{50} steadily dropped. All DNA constructs and concentrations are summarized in Supplementary Table 2.1.

NOT trigger reduced the eGFP output compared to the activated trigger state while the output remained the same when an off-target NOT trigger was added, which indicates that resource sharing was not a key determinant in the experiment. However, expression from the toehold switch construct was not completely eliminated in the presence of both the activating and deactivating trigger. This indicated that the equimolar concentrations of activating and deactivating trigger DNA did not produce sufficient NOT trigger to sequester all activating trigger RNA. An excess of NOT trigger was indeed able to diminish output expression close to background levels (Figure 2.7b). In summary, as a proof-of-concept that ribocomputing operations can be conducted in TXTL, we have demonstrated that a toehold switch-based NOT gate can be implemented in our TXTL system.



(i)	10 nM	10 nM	10 nM	10 nM	10 nM	10 nM	10 nM	2 nM	2 nM	2 nM	2 nM	2 nM
(ii)		14 nM										
(iii)			14 nM	14 nM		14 nM			6 nM		6 nM	6 nM
(iv)				14 nM			14 nM				18 nM	
(v)					14 nM		14 nM			6 nM		
(vi)						14 nM						18 nM

Figure 2.7: eGFP endpoint concentrations of expression of a toehold switch with activating and deactivating triggers in TXTL (see Methods). a) Activation of the toehold switch (i) by the original trigger (ii) and adapted trigger with toehold regions (iii) and deactivation by the NOT trigger (iv). Controls for both triggers are indicated by the off-target activating trigger (v) and off-target deactivating trigger (vi). All trigger DNA concentrations were equal. b) Similar to a), but instead of equimolar trigger DNA concentrations, an excess of NOT trigger DNA was used to deactivate translation of the output eGFP gene further.

2.6 Conclusion

In this chapter, a cell-free TXTL system was optimized to express RNA-based regulatory elements. We demonstrated the activation of a toehold switch in TXTL reactions and established a suitable cloning method to construct toehold switch and trigger DNA templates. A toehold switch was combined with the σ^{28} factor, a regulatory protein widely used in TXTL circuits, to create a genetic cascade controlled by an RNA input. Furthermore, the potential of performing complex computations based on riboregulators in TXTL was demonstrated by the implementation of a toehold switch-based NOT gate. Taken together, these findings show that complex RNA-based synthetic genetic circuits can be constructed and characterized in TXTL.

2.7 Methods

2.7.1 Preparation of DNA Templates

DNA constructs were created with golden gate assembly (GGA) using the overlapping sequences adapted from Sun *et al.*[35] (Figure 2.5). Promoters, UTR1, coding sequences and terminators were ordered from IDT as gBlock fragments or amplified from the pBEST vector using PCR. Toehold switch and trigger sequences were taken from previous studies in the group of Dr. P. Yin (unpublished, related toehold switch plasmids can be obtained from Addgene (https://www.addgene.org/Alexander_Green/) and PCR amplified. PCR products were gel purified using the QIAquick Gel Extraction Kit (Qiagen) and added in equimolar amounts to GGA assembly reactions with BSAI-HF (NEB), T4 ligase (Promega) and the T4 ligase buffer. GGA reactions were performed in a thermocycler according to a standard GGA protocol.[66] The GGA products were transformed into NovaBlue cells (Merck), from which the plasmids were purified using the QIAprep Spin Miniprep Kit (Qiagen) and the DNA sequences were confirmed using Sanger sequencing.

Linear DNA templates for expression in TXTL reactions were created by PCR using Phusion High-Fidelity DNA Polymerase (NEB) with primers pBEST_LinL_F (GGCGAATCCTCTGACCAGC) and pBEST_LinL_R (CCAAGCTGGACTG-TATGCACG) and subsequent purification using QIAquick PCR Purification Kit (Qiagen).

2.7.2 Preparation of Cell Lysate

The *E. coli* cell lysate was initially prepared from BL21 Rosetta 2. Subsequent batches were based on the RNase E deficient BL21 STAR (DE3) (ThermoFisher Scientific) cells that were transformed with the pRARE vector from BL21 Rosettta (Merck) (Figure 2.2). The lysate was prepared according to previously published protocols, with slight adaptations.[34,36] The *E. coli* strain was grown in 2xYT medium supplemented with 40 mM potassium phosphate dibasic and 22 mM potassium phosphate monobasic until an OD_{600} of 1.7 was reached. The cultures were spun down and washed thoroughly with S30A buffer (14 mM Magnesium L-glutamate, 60 mM Potassium L-glutamate, 50 mM Tris, titrated to pH 8.2 using glacial acetic acid), before being resuspended in 0.9 mL S30A buffer per gram of dry pellet. The cell suspension was lysed using a French press at 16000 lb pressure in two passes and spun down. The supernatant was incubated at 37 °C for 1.5 hours and spun down. The supernatant was dialyzed into S30B buffer (14 mM Magnesium L-glutamate, 150 mM Potassium L-glutamate, titrated to pH 8.2 using 2 M Tris) in two steps for 3 hours total and spun down again. The supernatant was aliquotted, snap-frozen in liquid nitrogen and stored at -80 °C.

The energy mixture was prepared according to the protocol previously described by Sun *et al.*[36] and a constant distribution amino acid solution was prepared.[67]

2.7.3 Preparation of TXTL Reactions

The cell lysate (33% of total reaction volume) was combined with the energy mixture, amino acid solution (37.5 mM), Magnesium L-glutamate (8 mM), PEG-8000 (2%), GamS protein (3 µM; prepared as described by Sun *et al.*[35]) and MilliQ to form the 1.54x TXTL reaction mixture (65% of total reaction volume). For the lysate exploration experiment in Figure 2.2a, Magnesium L-glutamate, Potassium L-glutamate, and PEG-8000 concentrations were varied to achieve the final concentrations shown in the figure, which include the 4.7 mM Magnesium L-glutamate and 50 mM Potassium L-glutamate obtained by diluting buffer S30B three times. The remaining volume (35%) of the reactions was used to add the linear DNA constructs of the gene networks and supplemented to the final volume with MilliQ. The DNA constructs and their concentrations in each experiment are summarized in Supplementary Table 2.1.

2.7.4 Batch TXTL Reactions

Batch TXTL reactions were prepared in total volumes of 10 µL and transferred to 384-wells Nunc plates. The reactions were incubated at 29 °C and eGFP was measured on a Saffire II (Tecan), Spark 10M (Tecan) or Synergy H1M (Biotek) plate reader for at least 14 hours. The concentration measured after 14 h of incubation was taken as endpoint concentration for all experiments. The plate readers were calibrated using a titration range of purified eGFP protein.

2.8 Supplementary Figures and Tables

Figure	Samples	DNA	DNA Name	Concentration (nM)
2.2a	All		P70a-eGFP	5
2.2b	All		P70a-eGFP	0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20
2.3	Initial: All		P70a-SCAR-SwitchA-eGFP	10
	Optimized: All		P70a-SwitchA-eGFP	10
2.4	Left (orange)		P70a-SwitchA-eGFP	2
	Left (orange)	X1/X3	P70a-TriggerA	1, 2, 5, 6, 10, 20
	Right (green)	X1/X3	P70a-TriggerA	10
	Right (green)		P70a-SwitchA-eGFP	1, 2, 5, 10
2.6a	On-target	X1/X3	P70a-TriggerA	10
	Off-target		P70a-TriggerC	10
	All		P28a-eGFP	5
	All	Y1/Y3	P70a-SwitchB-S28	0.1
2.6b	All	X1/X3	P70a-TriggerA	10
	All		P28a-eGFP	10
	All	Y1/Y3	P70a-SwitchB-S28	0.1, 0.2, 0.5, 1, 2, 5, 10
2.7	All	Constructs and concentrations are explicitly stated in the figure.		

Supplementary Table 2.1: DNA construct concentrations of all batch, flow and in silico experiments.

2.9 References

(1) Borkowski, O., Ceroni, F., Stan, G.-B., and Ellis, T. (2016) Overloaded and stressed: whole-cell considerations for bacterial synthetic biology. *Current Opinion in Microbiology 33*, 123–130.

(2) Garamella, J., Marshall, R., Rustad, M., and Noireaux, V. (2016) The All E. coli TX-TL Toolbox 2.0: A Platform for Cell-Free Synthetic Biology. *ACS Synth. Biol. 5*, 344–355.

(3) Sun, Z. Z., Yeung, E., Hayes, C. A., Noireaux, V., and Murray, R. M. (2014) Linear DNA for Rapid Prototyping of Synthetic Biological Circuits in an Escherichia coli Based TX-TL Cell-Free System. *ACS Synth. Biol. 3*, 387–397.

(4) Sun, Z. Z., Hayes, C. A., Shin, J., Caschera, F., Murray, R. M., and Noireaux, V. (2013) Protocols for Implementing an Escherichia coli Based TX-TL Cell-Free Expression System for Synthetic Biology. *JoVE (Journal of Visualized Experiments)* e50762.

(5) Shin, J., and Noireaux, V. (2012) An E. coli Cell-Free Expression Toolbox: Application to Synthetic Gene Circuits and Artificial Cells. *ACS Synth. Biol. 1*, 29–41.

(6) Shimizu, Y., Inoue, A., Tomari, Y., Suzuki, T., Yokogawa, T., Nishikawa, K., and Ueda, T. (2001) Cell-free translation reconstituted with purified components. *Nature Biotechnology 19*, 751–755.

(7) Lavickova, B., and Maerkl, S. J. (2019) A Simple, Robust, and Low-Cost Method To Produce the PURE Cell-Free System. *ACS Synth. Biol. 8*, 455–462.

(8) Villarreal, F., Contreras-Llano, L. E., Chavez, M., Ding, Y., Fan, J., Pan, T., and Tan, C. (2018) Synthetic microbial consortia enable rapid assembly of pure translation machinery. *Nature Chemical Biology 14*, 29–35.

(9) Asahara, H., and Chong, S. (2010) In vitro genetic reconstruction of bacterial transcription initiation by coupled synthesis and detection of RNA polymerase holoenzyme. *Nucleic Acids Research 38*, e141–e141.

(10) Kwon, Y.-C., and Jewett, M. C. (2015) High-throughput preparation methods of crude extract for robust cell-free protein synthesis. *Scientific Reports 5*, 8663.

(11) Didovyk, A., Tonooka, T., Tsimring, L., and Hasty, J. (2017) Rapid and Scalable Preparation of Bacterial Lysates for Cell-Free Gene Expression. *ACS Synth. Biol. 6*, 2198–2208.

(12) Caschera, F., and Noireaux, V. (2014) Synthesis of 2.3 mg/ml of protein with an all Escherichia coli cell-free transcription–translation system. *Biochimie 99*, 162–168.

(13) Guo, S., and Murray, R. M. (2019) Construction of Incoherent Feedforward Loop Circuits in a Cell-Free System and in Cells. *ACS Synth. Biol. 8*, 606–610.

(14) Patel, A., Murray, R. M., and Sen, S. (2020) Assessment of Robustness to Temperature in a Negative Feedback Loop and a Feedforward Loop. *ACS Synth. Biol. 9*, 1581–1590.

(15) Karzbrun, E., Tayar, A. M., Noireaux, V., and Bar-Ziv, R. H. (2014) Programmable on-chip DNA compartments as artificial cells. *Science 345*, 829–832.

(16) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *PNAS 110*, 15985–15990.

(17) Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., and Maerkl, S. J. (2015) Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* (Simmel, F., Ed.) *4*, e09771.

(18) Yelleswarapu, M., van der Linden, A. J., van Sluijs, B., Pieters, P. A., Dubuc, E., de Greef, T. F. A., and Huck, W. T. S. (2018) Sigma Factor-Mediated Tuning of Bacterial Cell-Free Synthetic Genetic Oscillators. *ACS Synth. Biol. 7*, 2879–2887.

(19) Wagner, E. G. H., and Romby, P. (2015) Chapter Three - Small RNAs in Bacteria and Archaea: Who They Are, What They Do, and How They Do It, in *Advances in Genetics* (Friedmann, T., Dunlap, J. C., and Goodwin, S. F., Eds.), pp 133–208. Academic Press.

(20) Green, A. A., Silver, P. A., Collins, J. J., and Yin, P. (2014) Toehold Switches: De-Novo-Designed Regulators of Gene Expression. *Cell 159*, 925–939.

(21) Green, A. A., Kim, J., Ma, D., Silver, P. A., Collins, J. J., and Yin, P. (2017) Complex cellular logic computation using ribocomputing devices. *Nature 548*, 117–121.

(22) Lehr, F.-X., Hanst, M., Vogel, M., Kremer, J., Göringer, H. U., Suess, B., and Koeppl, H. (2019) Cell-Free Prototyping of AND-Logic Gates Based on Heterogeneous RNA Activators. *ACS Synth. Biol. 8*, 2163–2173.

(23) Pardee, K., Green, A. A., Ferrante, T., Cameron, D. E., DaleyKeyser, A., Yin, P., and Collins, J. J. (2014) Paper-Based Synthetic Gene Networks. *Cell 159*, 940–954.

(24) Pardee, K., Green, A. A., Takahashi, M. K., Braff, D., Lambert, G., Lee, J. W., Ferrante, T., Ma, D., Donghia, N., Fan, M., Daringer, N. M., Bosch, I., Dudley, D. M., O'Connor, D. H., Gehrke, L., and Collins, J. J. (2016) Rapid, Low-Cost Detection of Zika Virus Using Programmable Biomolecular Components. *Cell 165*, 1255–1266.

(25) Takahashi, M. K., Tan, X., Dy, A. J., Braff, D., Akana, R. T., Furuta, Y., Donghia, N., Ananthakrishnan, A., and Collins, J. J. (2018) A low-cost paper-based synthetic biology platform for analyzing gut microbiota and host biomarkers. *Nature Communications 9*, 3347.

(26) Green, A. A., Kim, J., Ma, D., Silver, P. A., Collins, J. J., and Yin, P. (2017) Complex cellular logic computation using ribocomputing devices. *Nature 548*, 117–121.

(27) Kim, S.-J., Leong, M., Amrofell, M. B., Lee, Y. J., and Moon, T. S. (2019) Modulating Responses of Toehold Switches by an Inhibitory Hairpin. *ACS Synth. Biol. 8*, 601–605.

(28) Lehr, F.-X., Hanst, M., Vogel, M., Kremer, J., Göringer, H. U., Suess, B., and Koeppl, H. (2019) Cell-Free Prototyping of AND-Logic Gates Based on Heterogeneous RNA Activators. *ACS Synth. Biol. 8*, 2163–2173.

(29) Chappell, J., Takahashi, M. K., and Lucks, J. B. (2015) Creating small transcription activating RNAs. *Nature Chemical Biology 11*, 214–220.

(30) SantaLucia, J. (1998) A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences 95*, 1460–1465.

(31) Gao, M., Gnutt, D., Orban, A., Appel, B., Righetti, F., Winter, R., Narberhaus, F., Müller, S., and Ebbinghaus, S. (2016) RNA Hairpin Folding in the Crowded Cell. *Angewandte Chemie International Edition 55*, 3224–3228.

(32) Wagner, E. G. H., and Romby, P. (2015) Small RNAs in Bacteria and Archaea, in *Advances in Genetics*, pp 133–208. Elsevier.

(33) Mackie, G. A. (2013) RNase E: at the interface of bacterial RNA processing and decay. *Nature Reviews Microbiology 11*, 45–57.

(34) Mudd, E. A., and Higgins, C. F. (1993) Escherichia coli endoribonuclease RNase E: autoregulation of expression and site-specific cleavage of mRNA. *Mol Microbiol 9*, 557–568.

(35) Lopez, P. J., Marchand, I., Joyce, S. A., and Dreyfus, M. (1999) The C-terminal half of RNase E, which organizes the Escherichia coli degradosome, participates in mRNA degradation but not rRNA processing in vivo. *Molecular Microbiology 33*, 188–199.

(36) Siegal-Gaskins, D., Tuza, Z. A., Kim, J., Noireaux, V., and Murray, R. M. (2014) Gene Circuit Performance Characterization and Resource Usage in a Cell-Free "Breadboard." *ACS Synth. Biol. 3*, 416–425.

3

Time-Varying Inputs in Multi-Layer Microfluidics for Prolonged TXTL Reactions

Part of this work has been published as:

Pascal A. Pieters, Bryan L. Nathalia, Ardjan J. van der Linden, Peng Yin, Jongmin Kim, Wilhelm T. S. Huck, and Tom F. A. de Greef. *ACS Synthetic Biology*, ***10**, 1406-1416 (2021)*

and

Ardjan J. van der Linden, Maaruthy Yelleswarapu, Pascal A. Pieters, Zoe Swank, Wilhelm T. S. Huck, Sebastian J. Maerkl, and Tom F. A. de Greef, Journal of Visualized Experiments **152**, e59655 (2019)

3.1 Introduction

Microfluidic devices have become a powerful lab technique to manipulate fluid flows and droplets on a small scale. Performing biochemical reactions at a nanoliter scale using microfluidics provides a cost-effective manner to scan many reaction conditions and biophysical parameters. For example, microfluidic reactors have been utilized to map the phase behavior of a protein to optimize its x-ray crystallography conditions[1] and scan combinations of reagents to determine RNA-ribosome binding kinetics.[2] In addition, droplet microfluidics has provided a platform to perform a high-throughput evaluation of cell-free reaction conditions, including a genetic circuit.[3]

In particular, the development of microfluidics based on poly(dimethylsiloxane) (PDMS), which is low-cost and has a short design iteration cycle.[4] PDMS devices are made by mixing two components, the base (dimethylvinyl-terminated dimethyl siloxane) and a curing agent (dimethyl methylhydrogen siloxane). To create patterns in PDMS layers, the mixture is poured onto a master mold, which is typically generated using photolithography on a silicon wafer. After curing at elevated temperatures, an elastic and transparent material is formed. The PDMS layers are subsequently bonded onto a glass slide or PDMS slabs to form the channels.

The adoption of this soft lithography method in biochemical research has enabled the conduction of complex experiments using small sample sizes. Valves and fluid pumps can be implemented on microfluidic devices using multiple PDMS layers patterned with channels.[5] Typically, one of the PDMS layers in the microfluidic device functions as flow layer, containing the fluid channels that transport reagent,

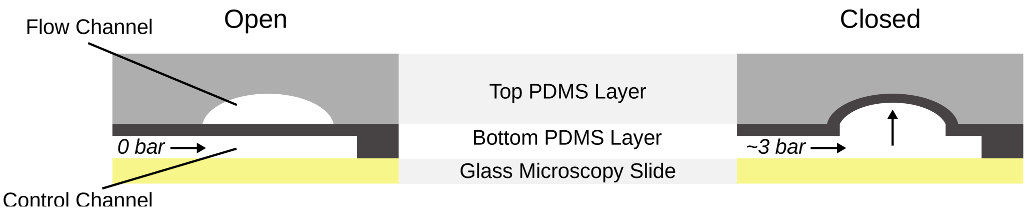


Figure 3.1: Operating principle of microfluidic push-up valves in a two-layer microfluidic device. The bottom PDMS layer contains the control channels that can be pressurized to push up a PDMS membrane to seal the flow channel it intersects with. Fluid flow in the channel in the upper PDMS layer is subsequently impeded, enabling regulation of fluid flow, isolation of reaction chambers and peristaltic pumps.

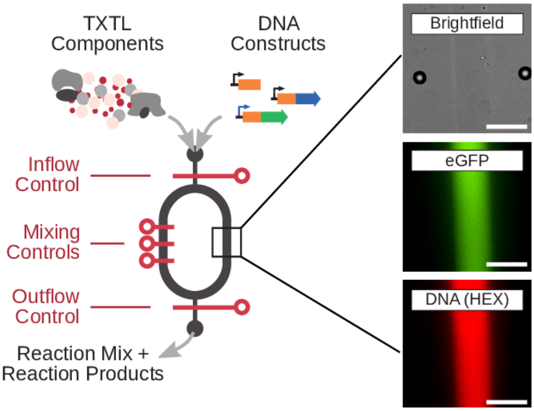


Figure 3.2: Schematic drawing of a microfluidic semi-continuous flow reactor and its operation. In addition, an exemplar brightfield image and fluorescence micrographs of a channel in the reactor are shown. Scale bars are 100 µm.

while the other layer is used to control the fluid flow. By pressurizing channels in the control layer, the thin (~5µm) PDMS membrane separating both layers can deform and block flow in select flow channels (Figure 3.1). This process can readily be automated using computer-controlled solenoid valves and a pressure regulator. Further integration with microscopy or other readout methods allows fully automated experiments to be performed.

This chapter describes the fabrication and verification of microfluidic devices to sustain out-of-equilibrium *in vitro* transcription and translation (TXTL) reactions for prolonged periods. Furthermore, we adapt the corresponding protocols to provide pulse inputs of varying lengths to a synthetic circuit.

3.2 Semi-Continuous Flow Reactors

A significant application of multilayer microfluidic devices is the creation of reaction chambers that can be controlled and continuously monitored. This idea was first explored by Estéves-Torres et al., who miniaturized an open reactor on a microfluidic chip to create a micro-continuous stirred tank reactor (µCSTR).[6] This design was further developed by Niederholtmeyer et al. to a microfluidic device that contains eight µCSTRs and nine reagent input channels that is suitable to conduct TXTL reactions.[7] Using a semi-continuous flow reactor, they replenished all TXTL reactants over time and eliminated reaction products by replacing a fraction of the reactor with fresh reactants (Figure 3.2). As a result, an approximately constant expression rate could be achieved over prolonged periods and a degradation rate was introduced for all reaction products, which enables the implementation of dynamic, out-of-equilibrium circuits that cannot operate under batch conditions. Niederholtmeyer et al. demonstrated that the device could be used to characterize a 3-node oscillator in PURE TXTL[7] and prototype 3- and 5-node oscillators in lysate-based TXTL.[8]

Recently, Yelleswarapu et al. utilized this device to implement a cell-free 2-node oscillator.[9] They demonstrated the value of a well-controlled reactor and accompanying computational methods to determine the oscillatory regimes of the genetic circuit. In addition, they employed the device to analyze a more complex setup, consisting of two oscillators, and showed that the oscillators were coupled through the shared use of the E. coli RNA polymerase by sigma-factors employed in the reactions. This work, taken together with the thorough characterization of oscillators by Niederholtmeyer et al.,[7,8] has solidified the use of microfluidic flow reactors as a tool to prototype and analyze synthetic genetic circuits.

3.3 Setup and Operation

We adopted the design of Niederholtmeyer et al.[7] and applied slight simplifications to remove manufacturing complexities and arrive with a design that can be produces using standard soft lithography techniques (Figure 3.3). The device has 9 reactant inlets, which can be selectively activated through a multiplexer. The inflow pumps, in combination with a pressure of 300 mbar on the inflow solutions, enable the fluid from the selected inlet to enter the device. From the multiplexer, there are two routes for the fluid to flow to the device outlet. First, a selection of reactors can be opened for the fluid to enter the reactors (*loading*). Alternatively, all reactors can remain closed, and the bypass channels can be utilized to bypass the reaction chambers and flush the contents of all channels, except the reactors, to the outlet (*flushing*). Bypassing the reactors allows for the dead volume between the multiplexer and reactor inlets to be replaced by a new input solution and assures that complete control over the reactor contents can be achieved.

All processes of the protocol, including *loading* and *flushing* operations, are automated via a computer-controlled programmable logic controller that regulates the state of pneumatic valves connected to the control channels on the microfluidic device (Figure 3.4). The software used to define protocols and communicate with

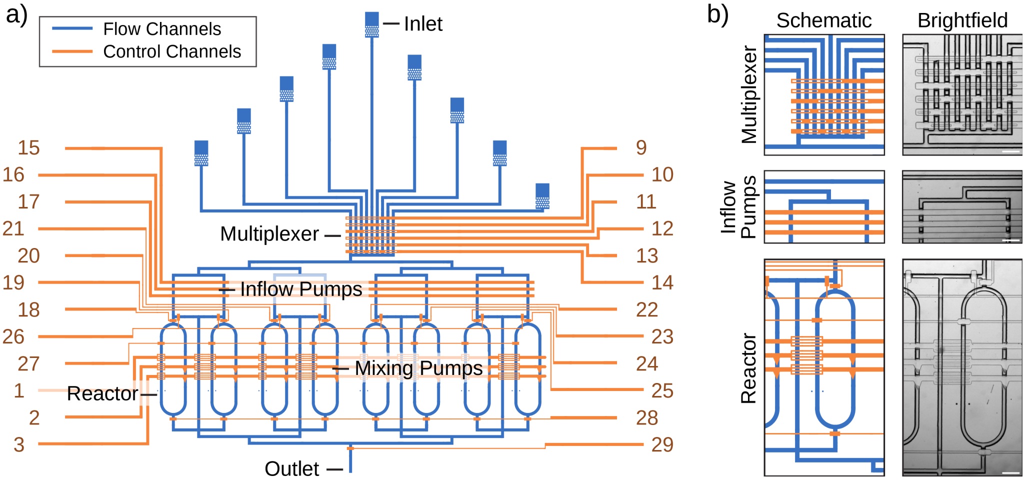


Figure 3.3: Microfluidic flow reactor used in this research. a) Schematic depicting the two layers of channels that constitute the flow reactor. Flow channels are shown in blue and control channels in orange. The device consists of 8 reactors, each with a volume of 10.7 nL and a channel with of 100 µm. The layer height of the control and flow layers were 45 µm and 25 µm, respectively. The 9 inlets lead to a multiplexer, which can be used to select the solution of single inlet to flow into the device by opening a combination of control channels 9-11 and 12-14. Control channels 15-17 form a peristaltic pump that is used to pump the selected solution through the device. The inlets of reactors are individually controlled by channels 18-25, whereas the outlets are under control of a single channel (28). Another peristaltic pump is used to mix the contents of the reactors (channels 1-3). Channel 27 can be used to asymmetrically load reactor rings, but is typically left depressurized during experiments. The reactors can be bypassed, which is regulated by channel 26. Finally, channel 29 controls the overall outlet of the device. b) The key modules of the device. Brightfield images are taken of a device of which the flow channels were filled with air and the control channels are filled with water and pressurized at 3 bar. Scale bars are 250 µm.

the controllers was custom-made in-house using Labview (National Instruments). This program also directs the timing of the image acquisition by the microscope, which monitors the fluorescence of the reactor channels. As a result, the software can fully control the device operation and readout and provides a single entry point to implement protocols.

Image acquisition was performed in bright field and two fluorescence channels using an inverted microscope with an automated stage. The computer-controlled automated stage was configured to image a straight part of the channels of each reactor per image cycle, which can be activated from the custom control software. Bright field images were taken as references for the position of the reactor chan-

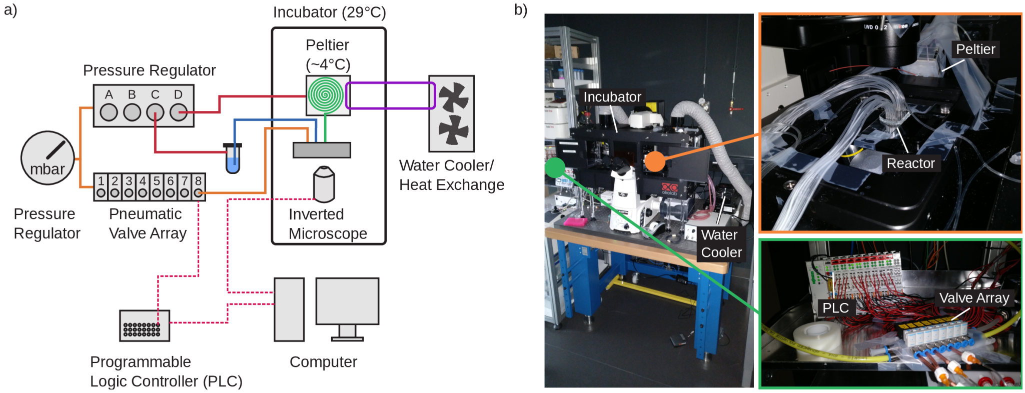


Figure 3.4: a) Schematic of the setup surrounding the microfluidic flow reactor. High pressure lines are shown in orange, low pressure lines in red, the TXTL reaction mixture in green, other reagents in blue and water cooling lines in purple. The dashed pink lines represent electronic connections between controllers. b) Photographs of the complete setup required to perform semi-continuous flow reactions using TXTL reagents. Close-ups of the inside of the incubator and the valves that control the control channels of a device are shown as outsets.

nels and to provide information to deduce the cause of possible device failures. eGFP fluorescence was measured as the preferred output of the genetic circuits developed in this work. An additional channel was recorded to be used for device calibration and reference dyes in experiments.

All microscopy images were processed using a custom MATLAB (Mathworks) script to automatically detect the flow channels and determine the average fluorescence intensity of a channel. First, a channel was detected through convolution of a channel image with a rectangular mask of the size of a channel. The average intensity was subsequently determined for a 50x100 pixel patch at the center of the detected channel.

3.3.1 Calibration

Preceding any experiments in the microfluidic flow reactors, the device integrity had to be verified and the reactors calibrated. Hereto, all control channels of a device were connected to the pneumatic valve array, and pressure was applied to detect leakage and device bonding issues. Subsequently, the fluid channels of the device were flushed with water and the reactors loaded with a fluorophore (FITC-dextran) solution. Finally, reactors were closed and the remaining channels were flushed with water. Observation of the device over 10 minutes or more would reveal any leakage of the reactor rings at the reactor inlets, outlets, or control chan-

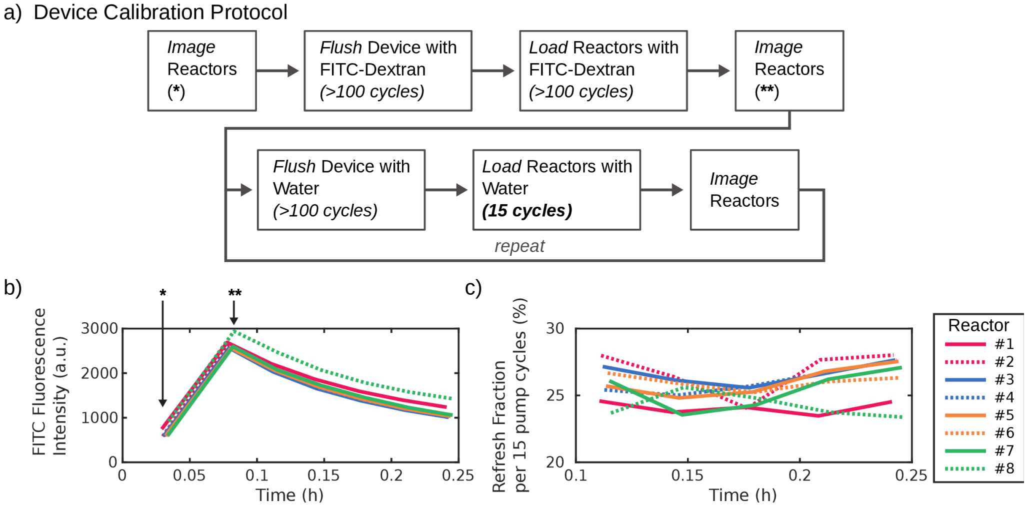


Figure 3.5: a) Protocol for a reactor calibration experiment for a microfluidic flow reactor. In short, the reactors are filled with a 25 µM FITC-dextran solution and subsequently diluted using a fixed number of pump cycles per iteration (15 cycles). b) The fluorescent intensity over time per reactor for a calibration protocol for one of the devices used in this research. The reactors are initially filled with water, which results in a background intensity measurement (*). Next, the reactors are loaded with a fluorophore solution (**) and diluted using water. c) The refresh rate per 15 pump cycles for each reactor in a microfluidic device, obtained from the dilution data in b). The average refresh fraction per reactor was used in subsequent experiment to translate refresh fractions to pump cycles.

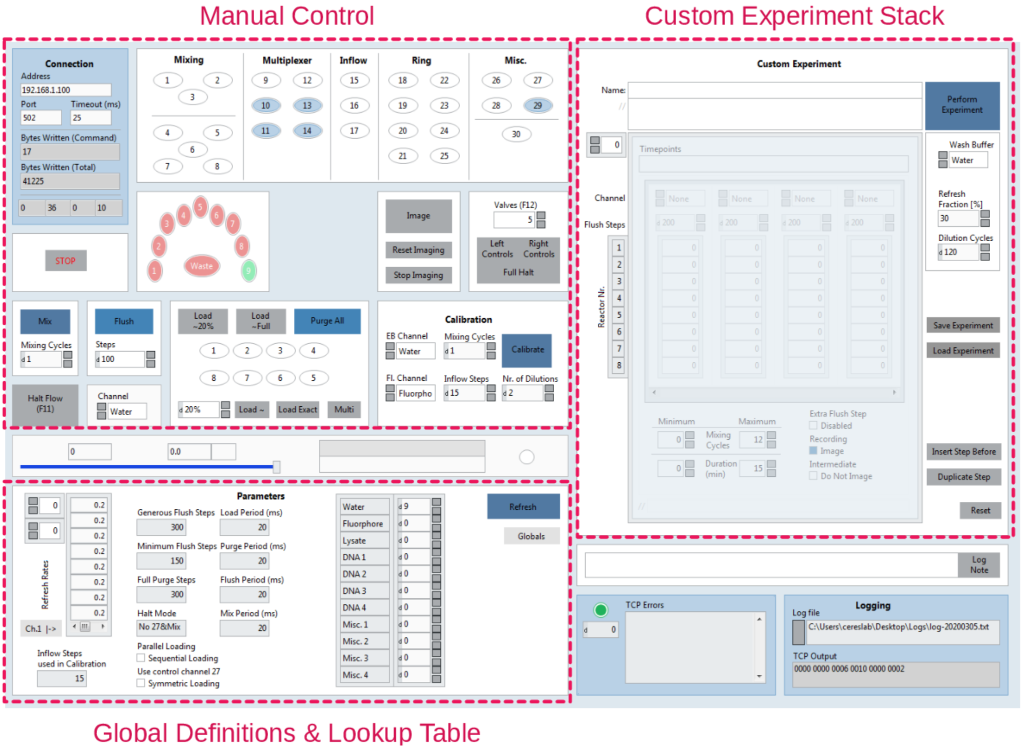
nels. Repeated observation of a decrease in fluorescence at the intersection of the reactor rings with the mixing control channels led us to increase the membrane layer thickness between the control and flow channels.

To precisely control the reactor contents, a relation was established between the *loading* protocol and the fraction of each reactor that it replaces. As part of this calibration protocol, the reactors were loaded with a fluorophore solution and sequentially diluted with water using a fixed number of pump cycles per iteration (typically 15), after which the reactors were thoroughly mixed. A refresh rate was subsequently deduced from the fractional decrease in fluorescence (Figure 3.5).

3.3.2 Custom Protocols

With the properties of a device verified, TXTL expression experiments could be performed. Previous work has focused on maintaining a constant composition of TXTL reaction mixture and DNA solution in the reactors over time to study circuits in out-of-equilibrium conditions and identify characteristic behavior, such as sustained oscillations.[9] However, most circuits do not exhibit such self-contained behavior and are characteristic in their transfer function, i.e., the relation between an input given to the circuit and its output. Various types of inputs need to be provided to such circuits to study their characteristic behavior. In microfluidic flow reactors, this corresponds to varying the composition of the fresh TXTL reactants that are supplied to the reactors each refresh cycle.

To facilitate the creation of modular and flexible protocols that contain arbitrary inputs and can vary the composition of a reactor in a user-defined way, we defined a generalized experiment unit. This unit consists of *flushing* the device with a user-defined inlet channel, *loading* a subset of reactors with a user-defined fraction of the same solution, *flushing* with a wash buffer to keep the device clean, *mixing* the



Address 192.168.1.100
Port 502
Timeout (ms) 25
Bytes Written (Command) 17
Bytes Written (Total) 41225
Mixing Cycles 1
Steps 100
Channel Water
EB Channel Water
Mixing Cycles 1
Fl. Channel Fluorpho
Inflow Steps 15
Nr. of Dilutions 2
Name:
Timepoints
Channel
Flush Steps
Reactor Nr.
Wash Buffer Water
Refresh Fraction [%] 30
Dilution Cycles 120
Mixing Cycles 12
Duration (min) 15
Extra Flush Step □ Disabled
Recording ■ Image
Intermediate □ Do Not Image
Refresh Rates
Inflow Steps used in Calibration 15
Generous Flush Steps 300
Load Period (ms) 20
Minimum Flush Steps 150
Purge Period (ms) 20
Full Purge Steps 300
Flush Period (ms) 20
Halt Mode No 27&Mix
Mix Period (ms) 20
Parallel Loading
□ Sequential Loading
Use control channel 27
□ Symmetric Loading
Water 9
Fluorphore 0
Lysate 0
DNA 1 0
DNA 2 0
DNA 3 0
DNA 4 0
Misc. 1 0
Misc. 2 0
Misc. 3 0
Misc. 4 0
Log file C:\Users\cereslab\Desktop\Logs\log-20200305.txt
TCP Output 0000 0000 0006 0010 0000 0002

Figure 3.6: Interface of the custom control software for the setup surrounding the microfluidic flow reactors. Three main modules of the software are highlighted: First, the manual control panes enable direct control over control valves and basic control sequences. In the custom experiment stack, complex experimental setups can be defined, saved and loaded. The global definitions module functions as central access to all device relates parameters that are set during the setup of experiments.

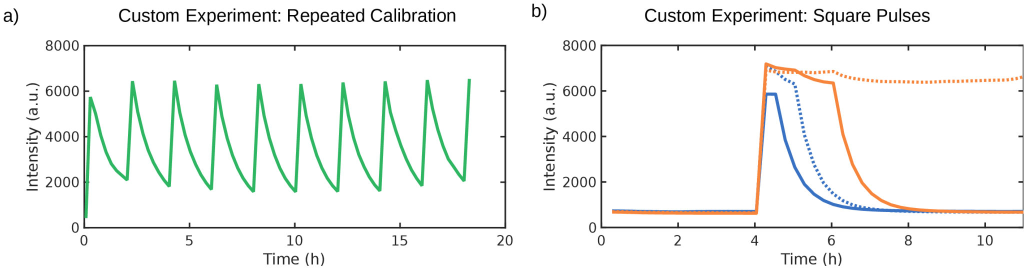


Figure 3.7: Output fluorescence traces of custom experiments that demonstrate the ability to control the composition of reagents in the flow reactors over time. a) A repeated calibration experiment (Figure 3.5), where the reactors are filled with 25 µm FITC-dextran every 2 h and subsequently diluted using water. b) The construction of square pulses using DNA-hexachlorofluorescein conjugate (see Methods). At t=4h, pulses of 30 min, 1 h, 2 h and an indefinite pulse (step function) are generated, corresponding to 2, 4, 8 and >30 experiment cycles.

reactors, and *imaging* all reactors. Per unit, the first *flush* and *load* steps can be repeated for multiple inlet channels and the duration of each unit can be controlled. An experiment consists of multiple units that are ordered in a stack, where each unit contains a sequence that matches the cycle number of an experiment. For each experiment cycle, the stack of units is matched to the current cycle number, where units higher in the stack are prioritized. The unit that matches the cycle number is executed and the experiment proceeds to the successive cycle. This setup enables the definition of a primary operating mode for the experiment (e.g. “no input”) as the bottom unit in the stack, which matches all cycle numbers. Exceptions can subsequently be defined in higher priority layers in the stack (e.g. “input” for cycles 5-10). We implemented this custom protocol definition in the Labview control software (Figure 3.6).

Additionally, we separated parameters that can be tuned per device out of the experiment definition. This includes the ordering of reactants in the device inlets, which was generalized by creating a lookup table for reagent names. Therefore, an experiment can define a reactor to be loaded with “fluorophore”, which is translated back to the correct inlet channel. Finally, the implementation of export and import of experiment definitions to and from XML files, in combination with the lookup table, provides a generalized method to set up complex experiments and reuse experiment definitions across devices.

To verify the functioning of custom protocol implementation and demonstrate its flexibility, two experiments were devised that make heavy use of a non-constant composition of the inflow reactants. Firstly, we implemented a calibration experiment using the custom protocol and repeated it every 2 hours (Figure 3.7a). The resulting calibration series remained relatively constant in an overnight experiment, indicating that the refresh rates of the device do not change significantly throughout a typical experiment. Next, we constructed an experiment that creates an approximation of square input pulses in the concentration of a fluorophore. Hereto, a high and a low concentration of fluorophore solution were connected to the device inlets. The high concentration of fluorophore was utilized to generate an initial concentration spike, while the low concentration of fluorophore could maintain the height of the initial spike. This was achieved by setting the ratio between the concentrations to the fraction of the reactor volume that is displaced in a single experiment cycle. In this experiment, the refresh fraction was set to 40% and, as a result, the high concentration of fluorophore was 2.5 times the low concentration. The onset of the resulting pulses closely resembled square pulses, and only a slight overshoot was observed for the initial spike, as indicated by the decreasing plateaus of the pulses (Figure 3.7b).

3.4 TXTL Reactions in Flow Reactors

Next, we explored the performance of TXTL reactions using in-house produced cell lysate in semi-continuous flow reactors. It has previously been demonstrated that gene expression can be achieved using TXTL reactions in the reactors.[8,10] Nevertheless, batch-to-batch variability of cell lysate is non-negligible,[11] which impacts the expression quality in flow reactors.

We expressed an eGFP gene under the control of a constitutive promoter (5 nM P70a-eGFP) in out-of-equilibrium TXTL reactions, achieved by refreshing 40% of each flow reactor per 15 minutes. We observed eGFP expression in the devices but did not readily achieve a constant expression rate over a prolonged period. After several repetitions of the experiment, we observed a few critical modes of failure (Figure 3.8). In some cases, expression ceased after a brief initial spike in fluorescence (Figure 3.8a). In other cases, non-constant expression could be observed, which could typically be attributed to device failure (e.g., increased resistance in some channels due to a blockage; Figure 3.8b). We often observed a regime of relatively constant expression, which rapidly halted after 11-14 hours (Figure 3.8c). On the other hand, in some cases, no decrease in expression rate was observed after an overnight measurement (Figure 3.8d).

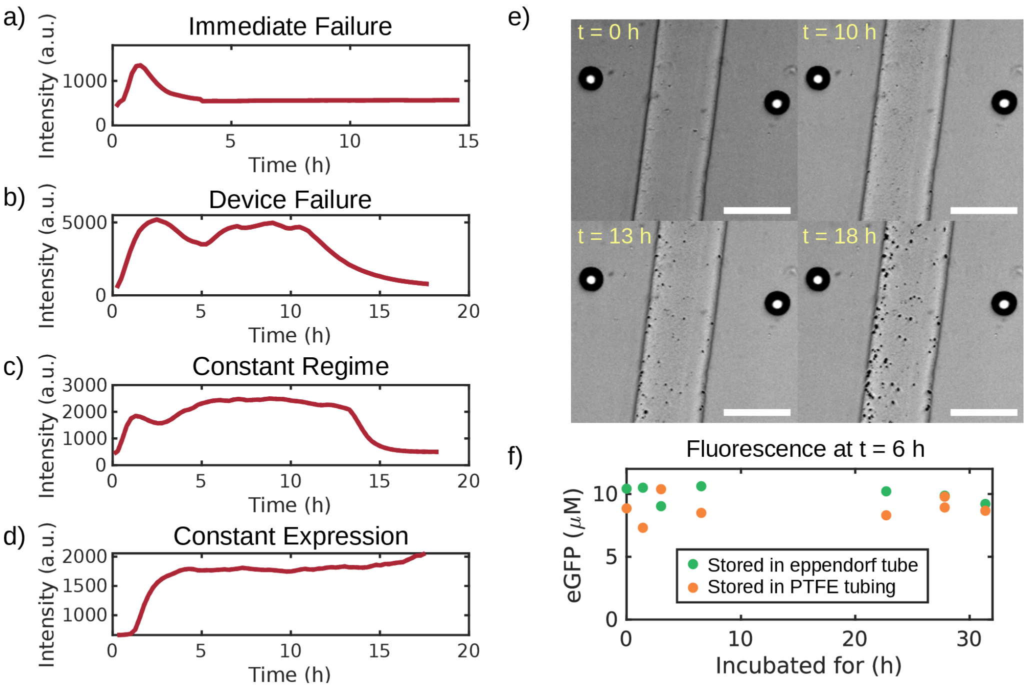


Figure 3.8: a) Example output fluorescence trace of a TXTL reaction that fails after an initial peak in output fluorescence. b) Example output fluorescence trace of a TXTL reaction where a device failure caused inconsistent refresh fractions, resulting in a non-constant expression rate. c) Example output fluorescence trace of a TXTL reaction that remained relatively consistent for several hours, except for an initial inconsistency in reactor composition and a drop-off in expression after ~13 h. d) Fluorescence output traces of a TXTL reaction in a flow reactor that resulted in constant expression for up to 15 hours, after which the setup ran out of fresh reactants, air entered the device. e) Bright field microscopy images of part of the reactor channel during the TXTL reaction shown in c). Formation of debris can be observed over time, in particular after more than 10 h of reaction time. Scale bars are 100 μm. f) Output eGFP concentrations measured from the fluorescence of batch TXTL reactions with 5 nM P70a-eGFP after a 6 h reaction using reaction mixture stored for various durations. TXTL reaction mixture was stored in an eppendorf tube or PTFE tubing at ~7°C. At various intervals, samples from the stored mixtures were taken and used for TXTL batch reactions. For both storage conditions, the TXTL reaction mixture retained its activity for more than 30 hours (see Methods).

The sudden decrease in expression observed in some cases after several hours of successful expression could hint at a reaction product that accumulates in the devices and poisons the TXTL reaction. We did observe the formation debris in the reaction chambers and TXTL lysate inlet of the microfluidic device (Figure 3.8e).

The stability of the TXTL reaction mixture in its storage device (PTFE tubing at 4-7°C) was verified by manually setting up eGFP TXTL batch expression experiments with lysate stored in PTFE tubing for varying durations (Figure 3.8f). Expression using lysate that was stored for more than 30 hours was observed to have a similar expression yield to fresh lysate, which indicates that the TXTL lysate storage method was not responsible for the expression drop-off observed in the flow reactors.

The cause of the expression failure after 11-14 hours was not elucidated in this research, but a combination of minimizing dead volume by using small diameter tubing, cooling the bacterial lysate solution, and maintaining a relatively high refresh rate (40% per 15 min) consistently resulted in >11 h of cell-free expression. Consequently, all flow reactor experiments in Chapter 4 were halted after 12 h, which is comparable to the experimental time scales of previous studies that make use of the flow reactors.[9] An additional benefit of this approach is the lack of severe debris formation at this time point, which allows for automated cleaning (using water and Terg-a-zyme) and reuse of microfluidic devices. As a consequence, the failure rate of experiments due to device manufacturing defects is lowered by reusing properly functioning devices.

3.5 Conclusion

In this chapter, we constructed and validated multilayer PDMS-based semi-continuous flow reactors. Furthermore, equipment to fully automatically control the microfluidic reactors and execute reaction protocols was set up, and custom control software was created. This setup enables the maintenance of biochemical reactions out-of-equilibrium through semi-continuous dilution of the reactions with fresh reactants. We demonstrated that time-dependent concentration patterns could be created in the reactors. Notably, a pattern that closely resembles a square pulse was created, which can be used to characterize the response of biochemical circuits to changes in input and inputs of varying durations. TXTL reactions were successfully sustained in the flow reactors for more than 11 hours. This provides an excellent operating window to characterize the dynamic behavior of most synthetic genetic circuits in TXTL.

3.6 Methods

3.6.1 Microfluidic Devices

The microfluidic semi-continuous flow reactors were produced using standard soft lithography methods.[41,63]

Semi-continuous flow reactions were performed according to the protocol previously described by van der Linden *et al.*[63], with adapted Labview control software to enable the configuration of time-varying input signals. The devices were monitored on an Eclipse Ti-E inverted microscope (Nikon). Reactor channels were automatically detected in the obtained images using a custom Matlab (Mathworks) script and the average fluorescence of a 50x100 pixel rectangle at the center of a channel was calculated to represent the output fluorescence.

3.6.2 Manual Operation

To set up the flow reactor device and its control systems, manual operation of the control channels is required. Tubing connected to the pneumatic valve array (Fig-

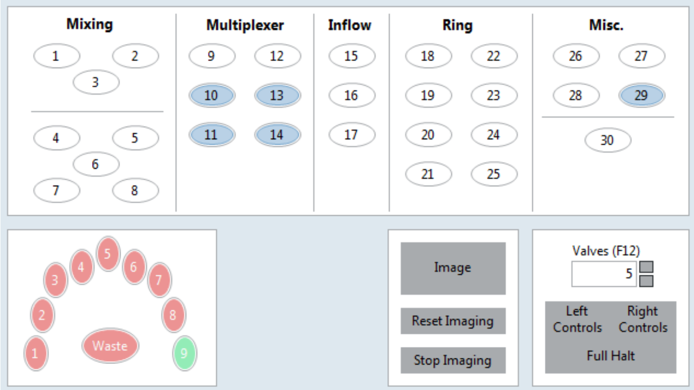


Figure 3.9: Manual control operators in the custom control software. All 30 connected valves can be directly toggled. The numbers correspond to the control channel numbering in Figure 3.2a. Currently, only control channels 10, 11, 13, 14 and 29 are pressurized, connecting inlet 9 to the rest of the device, as shown in the bottom left. This manual part of the interface furthermore contains buttons for direct signals to the imaging software and shortcuts for valve states that are helpful when connecting the device to the valve array.

ure 3.4) was connected to the device one by one and pressurized by directly toggling the corresponding valves in the control software (Figure 3.9).

In addition to direct control over individual channels, the following basic operations have been implemented (Figure 3.10):

Halt Flow: The multiplexer, reactor inlets and outlets, the bypass control channel are all closed off to completely halt the flow in the device.

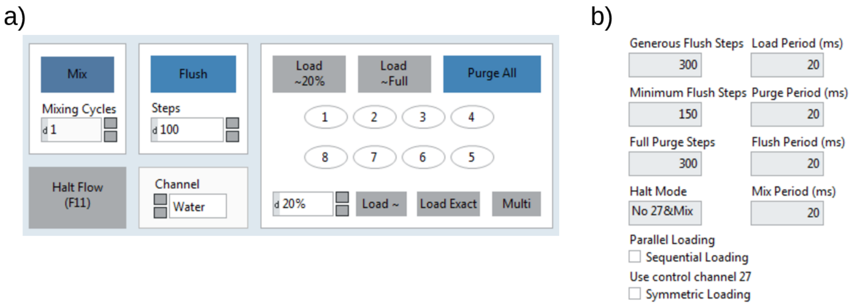
Flush: Pump the solution from the selected inlet into the device, but bypass the reactors.

Load: Sequentially pump a fixed amount of solution from the selected inlet into the reactors.

Purge: Pump an excess of the selected solution (usually water or buffer) through the reactors.

Mix: Use the peristaltic pumps in the reactors to mix its contents, no inlets or outlets are opened.

These operations can all be manually initiated, but are also part of the calibration and custom protocols.



Mixing Cycles: 1

Steps: 100

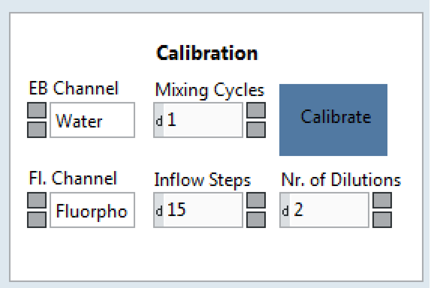
Channel: Water

20%

Generous Flush Steps	Load Period (ms)
300	20
Minimum Flush Steps	**Purge Period (ms)**
150	20
Full Purge Steps	**Flush Period (ms)**
300	20
Halt Mode	**Mix Period (ms)**
No 27&Mix	20

Parallel Loading
□ Sequential Loading
Use control channel 27
□ Symmetric Loading

Figure 3.10: Basic control operations in the custom control software. a) Buttons to initiate the various protocols. A central channel selector controls the multiplexer for the flush, load and purge sequences. Loading sequences with a tilde (e.g. "Load ~20%") can be executed without calibration since they use a the rough that 15 pump steps refresh 20% of a reactor. b) The number of steps for purging, generous flushing (water, buffers, etc.) and minimum length flushing (costly reagents) are defined as global variables and can be adjusted per microfluidic device. Similarly, the timing of the peristaltic pumps can be adjusted to account for small differences in flow resistance between devices.



Calibration

EB Channel: Water

Mixing Cycles: d 1

Fl. Channel: Fluorpho

Inflow Steps: d 15

Nr. of Dilutions: d 2

Figure 3.11: Interface for the built-in calibration protocol. The sequence of operation shown in Figure 3.5a is automatically executed using the parameters defined here.

3.6.3 Calibration

All 8 reactors on each device used in this research were calibrated to determine the fraction of the reactor volume that is displaced per pump cycle. To determine this relation, a dilution sequence of 25 µM FITC-dextran was executed using a fixed number (15) of pump cycles (Figure 3.5a). Typically, a single pump cycle replaced approximately 1-1.5% of the reactor volume. The calibration protocol was built into the custom software and merely requires setting the correct fluorophore and elution buffer inlet channels (Figure 3.11). Note that calibration is independent of exact reactor volume and fluorophore concentration, since the fraction of the reactor that is refreshed is determined based on the fractional decrease in observed fluorescence.

3.6.4 Repeated Calibration Experiment

To test the custom protocols implementation in the device control software, a protocol that repeats a dilution sequence every 2 hours was devised. Hereto, the reactor was filled with 25 µM FITC-dextran and every 15 minutes 20% of the reactor volume was replaced with water. The number of pump cycles to replace 20% of the reactor volume was determined based on an initial calibration experiment. The dilution step was executed 7 times, after which the reactor was loaded with FITC-dextran to repeat the dilution sequence several times.

3.6.5 Custom Inputs

A DNA-Hexachlorofluorescein conjugate (IDT) was used as fluorophore to demonstrate the construction of square pulses using the microfluidic reactors. Reactors were operated as an out-of-equilibrium experiment, refreshing 40% of each reactor every 15 minutes. First, all reactors, except a positive control, were loaded with water. After 4 hours of refreshing with water, pulses were created. To create an input that resembles a square pulse function, an initial step containing 2.5x the final input concentration of fluorophore (DNA-Hexachlorofluorescein conjugate) was loaded, after which all subsequent steps contained the regular concentration (1 µM). Pulses of 2, 4 and 8 cycles were created in different reactors on the device. In addition, a step function was constructed by continuing to refresh with the fluorophore, instead of switching back to water.

3.6.6 Preparation of Cell Lysate

The E. coli cell lysate was prepared from the RNase E deficient BL21 STAR (DE3) (ThermoFisher Scientific) cells that were transformed with the pRARE vector from BL21 Rosettta (Merck). The lysate was prepared according to previously published protocols,[13,21] with the adaptations described in Chapter 2. The E. coli strain was grown in 2xYT medium supplemented with 40 mM potassium phosphate dibasic and 22 mM potassium phosphate monobasic until an OD600 of 1.7 was reached. The cultures were spun down and washed thoroughly with S30A buffer (14 mM Magnesium L-glutamate, 60 mM Potassium L-glutamate, 50 mM Tris, titrated to pH 8.2 using glacial acetic acid), before being resuspended in 0.9 mL S30A buffer per gram of dry pellet. The cell suspension was lysed using a French press at 16000 lb pressure in two passes and spun down. The supernatant was incubated at 37 °C for 1.5 hours and spun down. The supernatant was dialyzed into S30B buffer (14 mM Magnesium L-glutamate, 150 mM Potassium L-glutamate, titrated to pH 8.2 using 2 M Tris) in two steps for 3 hours total and spun down again. The supernatant was aliquoted, snap-frozen in liquid nitrogen and stored at -80 °C.

The energy mixture was prepared according to the protocol previously described by Sun et al.[21] and a constant distribution amino acid solution was prepared.[22]

3.6.7 TXTL Reactions in Flow Reactors

The 1.54x TXTL reaction mixture was stored on a water-cooled peltier element (4-7 °C) during the experiment to maintain reactivity of the solution, whereas other reactants were stored in tubing in the incubation chamber (29 °C). Reactions were conducted overnight, during which 40% of each reactor was refreshed 15 minutes with 65% TXTL reaction mixture and 35% DNA (Figure 3.12 and Figure 3.13).

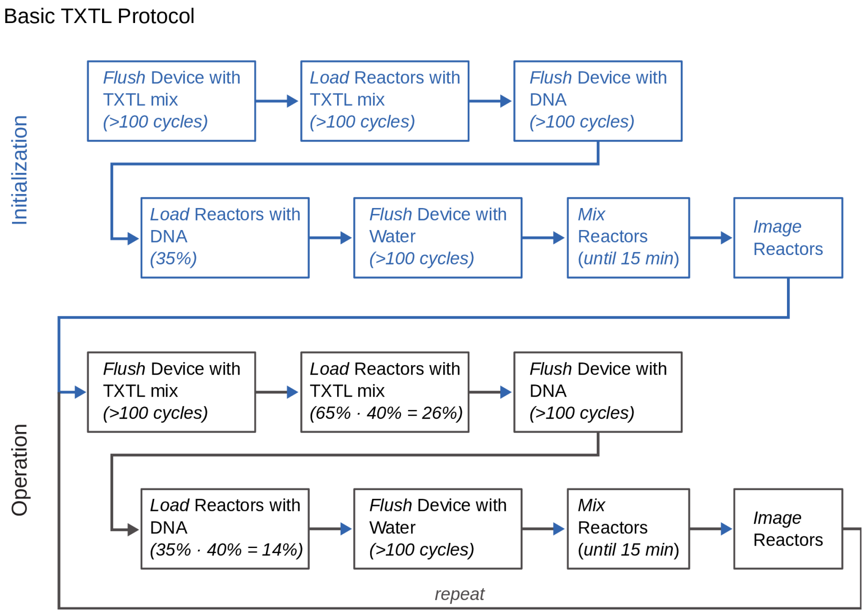
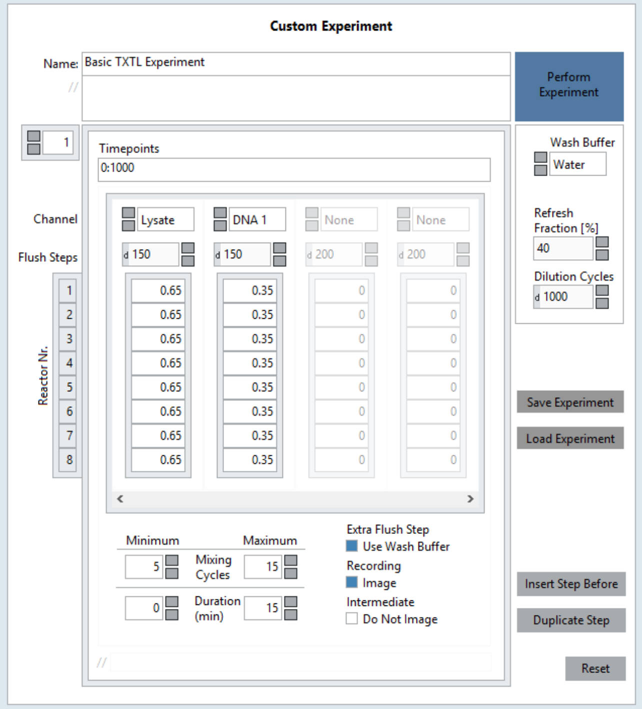

Basic TXTL Protocol
Initialization
Flush Device with TXTL mix (>100 cycles)
Load Reactors with TXTL mix (>100 cycles)
Flush Device with DNA (>100 cycles)
Load Reactors with DNA (35%)
Flush Device with Water (>100 cycles)
Mix Reactors (until 15 min)
Image Reactors
Operation
Flush Device with TXTL mix (>100 cycles)
Load Reactors with TXTL mix (65% · 40% = 26%)
Flush Device with DNA (>100 cycles)
Load Reactors with DNA (35% · 40% = 14%)
Flush Device with Water (>100 cycles)
Mix Reactors (until 15 min)
Image Reactors
repeat

Figure 3.12: Protocol for the execution of a TXTL experiment in the microfluidic flow reactors. This protocol assumes that every 15 minutes 40% of the reactor volume is displaced with 65% fresh TXTL mixture and 35% DNA. It can however easily be extended for other compositions, refresh rates and multiple DNA species. Since before the protocol is executed the reactors are filled with water, an initialization sequence is required. Initialization steps of the device are similar to normal operations steps, except that the reactors are completely filled with TXTL mixture and subsequently 35% of the TXTL mixture is replaced with DNA

Microfluidic reactors of reaction that were terminated after 11 hours could be cleaned. These devices were cleaned for reuse through repeated flushing with a Terg-a-zyme enzyme detergent solution (Alconox) and MilliQ.

3.6.8 TXTL Stability Experiment

The cell lysate (33% of total reaction volume) was combined with the energy mixture, amino acid solution (37.5 mM), Magnesium L-glutamate (8 mM), PEG-8000 (2%), GamS protein (3 µM; prepared as described by Sun et al.[19]) and MilliQ to form the 1.54x TXTL reaction mixture (65% of total reaction volume). The remaining volume (35%) of the reactions was used to add the P70a-eGFP linear DNA (5 nM final concentration).



Custom Experiment

Name: Basic TXTL Experiment

1

Timepoints
0:1000

Reactor Nr.	Lysate	DNA 1	None	None
Flush Steps	d 150	d 150	d 200	d 200
1	0.65	0.35	0	0
2	0.65	0.35	0	0
3	0.65	0.35	0	0
4	0.65	0.35	0	0
5	0.65	0.35	0	0
6	0.65	0.35	0	0
7	0.65	0.35	0	0
8	0.65	0.35	0	0

Wash Buffer: Water

Refresh Fraction [%]: 40

Dilution Cycles: d 1000

	Minimum	Maximum
Mixing Cycles	5	15
Duration (min)	0	15

Extra Flush Step
☑ Use Wash Buffer

Recording
☑ Image

Intermediate
□ Do Not Image

Figure 3.13: Implementation of the basic TXTL protocol using the custom experiment function of the device control software. The resulting device operations are shown in the diagram in Figure 3.12. The initialization steps are automatically deduced from the protocol at timepoint 0. The reactors are filled with the first solution used at this time point (here "Lysate") and subsequent loading steps are executed with a 100% refresh ratio. Starting at time point 1, normal device operation starts and 40% of the reactor is refreshed, of which 65% with "Lysate" and 35% with "DNA 1". The duration of each experiment cycle is regulated by minimum and maximum mixing cycles and duration settings. By default, the protocol aims to reach the maximum amount of mixing cycles (15; each mixing cycle typically takes 1 minute). However, if the duration of the experiment cycle exceeds the maximum duration setting (15 minutes) and the minimum number of mixing cylces (5) has been achieved, mixing is halted and the protocol moves on to the imaging step.

Lysate an energy mixture were stored in eppendorf tubes or PTFE tubing at 7 °C. At varying intervals 6.5 µL of the TXTL reagents were combined with 3.5 µL P70a-eGFP linear DNA (5 nM final concentration). The reactions were incubated at 29 °C and eGFP was measured on a Saffire II (Tecan) plate reader for at least 6 hours.

3.7 References

(1) Hansen, C. L., Sommer, M. O. A., and Quake, S. R. (2004) Systematic investigation of protein phase behavior with a microfluidic formulator. *Proceedings of the National Academy of Sciences 101*, 14431–14436.

(2) Ridgeway, W. K., Seitaridou, E., Phillips, R., and Williamson, J. R. (2009) RNA–protein binding kinetics in an automated microfluidic reactor. *Nucleic Acids Research 37*, e142–e142.

(3) Hori, Y., Kantak, C., Murray, R. M., and Abate, A. R. (2017) Cell-free extract based optimization of biomolecular circuits with droplet microfluidics. *Lab Chip 17*, 3037–3042.

(4) Duffy, D. C., McDonald, J. C., Schueller, O. J. A., and Whitesides, G. M. (1998) Rapid Prototyping of Microfluidic Systems in Poly(dimethylsiloxane). *Anal. Chem. 70*, 4974–4984.

(5) Unger, M. A. (2000) Monolithic Microfabricated Valves and Pumps by Multilayer Soft Lithography. *Science 288*, 113–116.

(6) Galas, J.-C., Haghiri-Gosnet, A.-M., and Estévez-Torres, A. (2013) A nanoliter-scale open chemical reactor. *Lab Chip 13*, 415–423.

(7) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *PNAS 110*, 15985–15990.

(8) Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., and Maerkl, S. J. (2015) Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife 4*.

(9) Yelleswarapu, M., van der Linden, A. J., van Sluijs, B., Pieters, P. A., Dubuc, E., de Greef, T. F. A., and Huck, W. T. S. (2018) Sigma Factor-Mediated Tuning of Bacterial Cell-Free Synthetic Genetic Oscillators. *ACS Synth. Biol. 7*, 2879–2887.

(10) van der Linden, A. J., Yelleswarapu, M., Pieters, P. A., Swank, Z., Huck, W. T. S., Maerkl, S. J., and de Greef, T. F. A. (2019) A Multilayer Microfluidic Platform for the Conduction of Prolonged Cell-Free Gene Expression. *JoVE* e59655.

(11) Sun, Z. Z., Hayes, C. A., Shin, J., Caschera, F., Murray, R. M., and Noireaux, V. (2013) Protocols for Implementing an Escherichia coli Based TX-TL Cell-Free Expression System for Synthetic Biology. *JoVE (Journal of Visualized Experiments)* e50762.

4

Cell-Free Characterization of a Coherent Feed-Forward Loop

Part of this work has been published as:

Pascal A. Pieters, Bryan L. Nathalia, Ardjan J. van der Linden, Peng Yin, Jongmin Kim, Wilhelm T. S. Huck, and Tom F. A. de Greef. *ACS Synthetic Biology*, **10**, 1406-1416 (2021)

4.1 Introduction

In this chapter, we successfully construct a modular synthetic gene network based on translational regulation using the toehold switch riboregulators in TXTL that were characterized in Chapter 2. We design and build a coherent feed-forward loop (CFFL) with RNA as top regulator, modeled after sRNA-based CFFLs found in nature (Figure 4.1).[1–4] The CFFL is a network motif highly abundant in both bacterial and mammalian regulatory networks[5] and can display temporal filtering through sign-sensitive delay, where short inputs do not provoke a response while long-lived, persistent inputs are capable of generating a strong response.[6–8] This property is essential for circuits experiencing a noisy input signal but can also serve to decode the temporal information that is encoded in various stimuli in cells.[9–13] However, due to the dynamic nature of this function, it has remained difficult to systematically analyze temporal properties of CFFLs, with experimental characterization being limited to a narrow subset of its behavior.[8] Next to temporal filtering, it has been postulated that CFFLs can suppress leaky expression in a network, resulting in a higher fold-change of the circuit output.[1]

We characterize both the background suppression and temporal filtering functions of the synthetic CFFL circuit using the microfluidic semi-continuous flow reactor described in Chapter 3,[14–17] complemented with *in silico* experiments. Our analysis reveals that the synthetic CFFL can effectively reduce background expression of components, increasing the fold-change of circuits for a wide range of circuit parameters. In agreement with recent computational studies that show low robustness of temporal filtering in CFFLs,[6] the synthetic CFFL is not a potent noise filter, since the response of the circuit to time-varying inputs is similar to a reference cascade. We identify that more ultrasensitive response is required in the circuit component responsible for the delay in signal propagation, in order to develop a CFFL circuit that is able to serve as noise filter. Our results provide a foundation to construct modular synthetic gene networks based on translational regulation and offer renewed insight into the signal processing functions of feed-forward loops.

4.2 *In vitro* implementation of a synthetic CFFL

The structurally simplest CFFL motif consists of three genes that interact to enable a signal to propagate from the input either directly or via an intermediate gene to

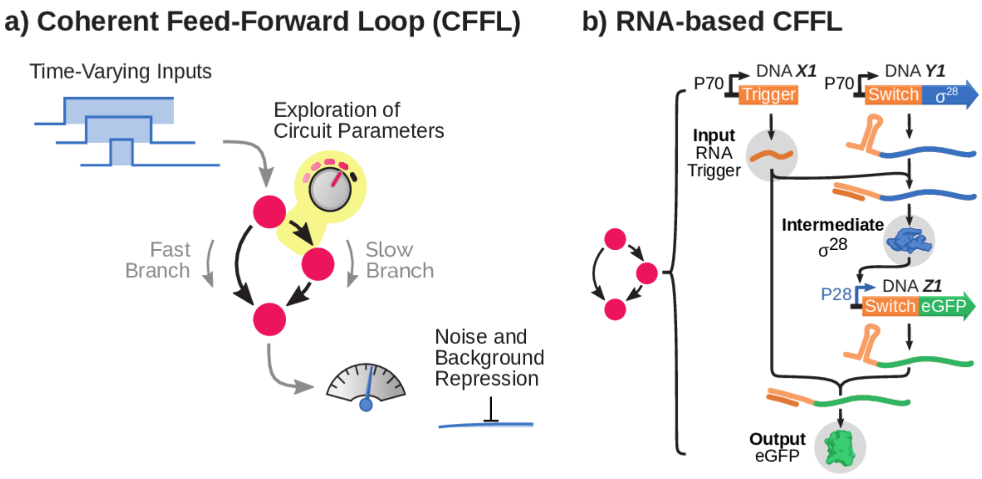


Figure 4.1: General concept of the construction and characterization of synthetic circuits based on CFFLs. a) Schematic drawings, where nodes are genes or RNA genes and arrows indicate interactions, of a CFFL. b) Schematic representation of all DNA species and the DNA, RNA and protein level interactions that constitute the CFFL. The three main components are the toehold switch, marked by the 5'-adjacent RNA stem loop, and its corresponding RNA trigger (orange), the E. coli σ^{28}-factor (blue) and fluorescent output protein (green).

the output gene (Figure 4.1a). In this circuit, the dissimilarity between the two pathways allows for propagation of the signal with different timescales. The delay generated by the presence of an intermediate gene largely determines the difference in timescales, whereas the mechanism by which the two pathways are integrated controls which aspect of the output is governed by the induced delay.[55]

We designed a synthetic type 1 coherent feed-forward loop, representing the most commonly observed subtype of CFFLs,[54] with AND-gate logic integrating the two branches of the circuit. In our design, the *E. coli* σ^{28}-factor, which has been successfully used in various TXTL-based genetic circuits,[40,43] is employed as intermediate species. The highly programmable toehold switch and trigger RNA-RNA post-transcriptional interactions are used to implement AND-type behavior (Figure 4.1b).[45] Additionally, the RNA trigger is utilized to activate translation of the sigma-factor, creating a CFFL with an RNA species as a top regulator, mimicking naturally occurring RNA regulatory circuits.[48,50]

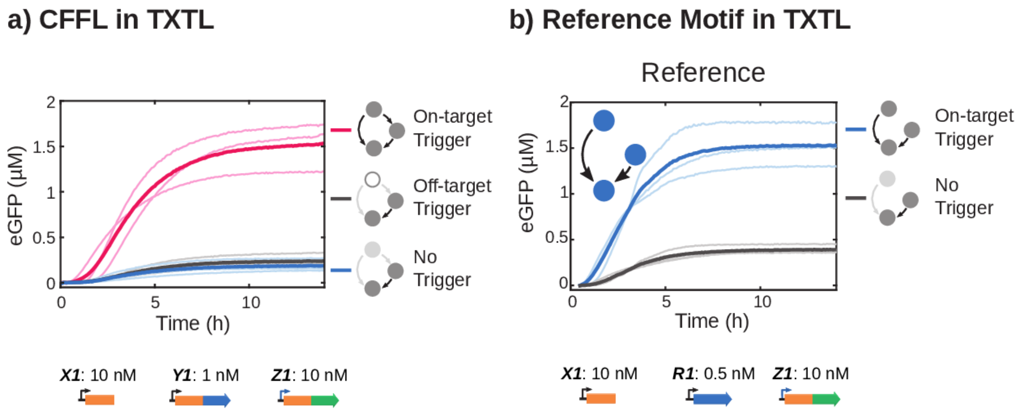


Figure 4.2: a) Time traces of eGFP production for the CFFL (pink, N=3). Expression without a trigger construct (gray, N=3) and with an off-target trigger (blue, N=3) are plotted as negative controls. Light traces are distinct experiments and the thick darker traces are the averages. Time traces of eGFP expression of the reference motif with an on-target trigger (light traces are three distinct experiments and the dark solid lines are their average) and without trigger (dark gray).

Using the cloning method and DNA constructs developed in Chapter 2, fragments of the CFFL were constructed. We extended the cascade developed in Chapter 2 to contain a toehold switch in the 5'-UTR of the eGFP output construct, resulting in the synthetic CFFL motif. When expressing the CFFL constructs in TXTL, eGFP expression only increased drastically in the presence of the cognate RNA trigger (Figure 4.2a). In the absence of trigger, some leakage from the switch was observed and slight crosstalk with a randomly selected off-target trigger was detected. In conjunction with the CFFL, a reference motif was designed by removing the interaction between the input RNA trigger and the intermediate σ^{28}-factor construct (Supplementary Figure). This circuit represents a simple signal transducing network with only a single path from input to output, to which the CFFL circuit characteristics can fairly be compared.[54] Similar to the CFFL, the reference motif was confirmed to only activate when the DNA construct of the on-target RNA trigger was present in the TXTL reactions (Figure 4.2b). These circuits collectively constitute a flexible system to build and analyze CFFL-based networks.

4.3 Characteristics of the synthetic CFFL

We have demonstrated that the synthetic CFFL can propagate an input signal and can be used to implement topologically more complex feed-forward circuits. To assess if the circuit can display information processing functionalities associated with a CFFL network motif[48,54,55] and whether these functionalities are retained in the composite CFFL network, we observed the CFFL, reference motif, and composite CFFL over a range of circuit parameters. A range of relative expression levels of the circuit components was sampled by varying the concentration of the DNA species. The response of the circuits was analyzed using four circuit characteristics (Figure 4.3). Firstly, to quantify the repression of background expression by the CFFL-based circuits, endpoint expression levels after 12h with and without input trigger were determined (*ON* and *OFF* state). Additionally, the ratio between these two measures, the *ON/OFF* ratio, was computed as a measure of the relative change in output upon circuit activation. TXTL batch reactions are unsuitable for the full assessment of temporal filtering behavior of the synthetic CFFL due to the inability to apply time-varying inputs. Nevertheless, a characteristic time scale of the response of the circuit can be determined and used to estimate for which range of input pulse durations the behavior is expected to manifest. We therefore computed, t_{50}, the time until half of the maximum output was reached as a measure of the characteristic circuit time-scale.

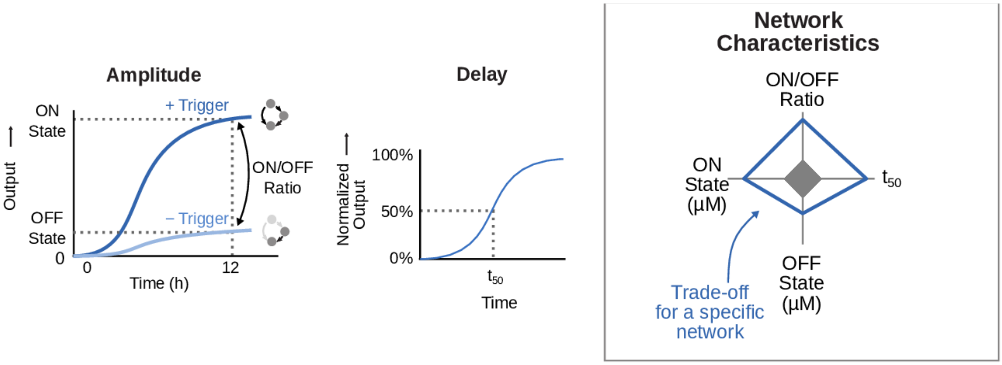


Figure 4.3: a) Schematic representation of the four characteristics determined for each circuit and parameter combination, plotted on the axes of a spider plot. The endpoint concentration of the output protein with circuit input (ON state; left axis) or without circuit input (OFF state; bottom axis). The ratio between those measurements gives the ON/OFF ratio (top axis). Lastly, the time until 50% of the endpoint concentration of output protein is reached serves as a temporal measure of the circuit (t_{50}; right axis).

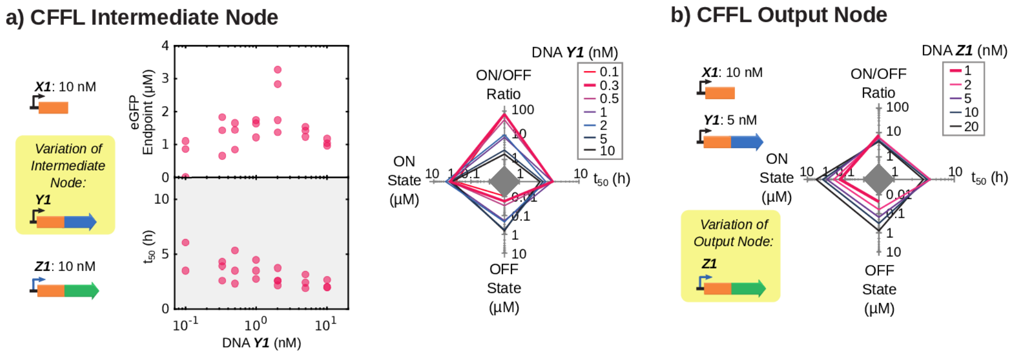


Figure 4.4: a) Endpoint concentrations, t_{50} and trade-offs for the CFFL with varying concentrations of σ^{28}-encoding DNA construct (N=3 for all concentrations in both the ON and OFF state, except the ON state with 2 nM DNA Y1, for which N=4). The highest ON/OFF ratio of 75x is reached for 0.3 nM σ^{28}-encoding DNA construct. b) Trade-offs for varying concentrations of output DNA construct of the CFFL, where all ON/OFF ratios range between 4x and 7x.

A wide range of σ^{28} expression levels was probed by varying the concentration of the DNA construct coding for the *E. coli* sigma factor (DNA **Y1**), whilst keeping other concentrations fixed. The CFFL was observed over time in the presence and absence of trigger DNA (DNA **X1**) to determine all its characteristics (Figure 4.4a). We observed a decrease in activation delay (t_{50}) for increasing DNA **Y1** concentration, suggesting that σ^{28} expression is a key parameter to determine the dynamic behavior of the circuit. Endpoint expression exhibited an increase for higher DNA **Y1** concentrations, before plateauing and subsequently slightly decreasing. This behavior suggests that the σ^{28} protein concentration reaches a saturated regime and subsequent addition of more DNA merely limits the expression capacity available to the output protein, resulting in an inefficient use of resources and a decrease in circuit output. We observed similar behavior when varying the σ^{28} expression levels in the reference motif, except that the background expression decreased less compared to the CFFL when the σ^{28} expression strength was lowered (Supplementary Figure 4.1). As a result, the CFFL motif displayed a higher *ON/OFF* ratio, reaching a value of 75x compared to 9x in the reference circuit.

Upon varying the concentration of output DNA species (DNA **Z1**), the delay in activation remained relatively constant, whilst the endpoint concentration and background endpoint concentration changed proportionally, with minimal changes to the *ON/OFF* ratio (Figure 4.4b). This further confirms that the σ^{28} DNA concentra-

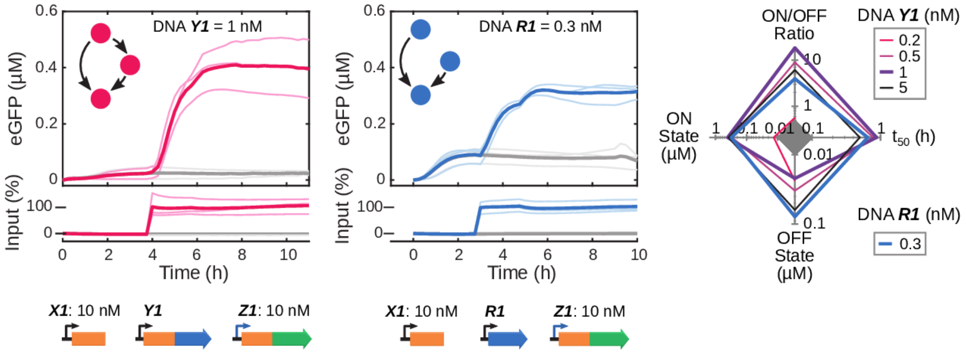


Figure 4.5: eGFP output time traces of flow reactions of the CFFL (left) and reference motif (right). Initially, no DNA encoding for the RNA input trigger was present. After either 3 or 4 hours, trigger DNA was added to the reactors to immediately reach a final concentration of 10 nM and was subsequently maintained at that concentration (colored lines). Negative controls, where no input DNA was added to the reactions are shown in gray. In addition, the trade-offs in characteristics of the flow reactions are plotted in a spider plot (CFFL in pink, reference motif in blue). The CFFL reaches a maximum ON/ OFF ratio of 19x for 1 nM σ^{28}-producing construct.

tion is the main parameter influencing the temporal behavior of the CFFL and the DNA **Z1** concentration can merely be used to scale the circuit output. Moreover, the same intermediate DNA species dictates the fold-change of the circuit, since only when background expression from the σ^{28} DNA construct does not significantly activate the cognate promoter on the output DNA construct, can the overall leakage be minimized.

4.4 Time-varying circuit inputs

Characterization of the temporal behavior of CFFLs requires the introduction of input pulses of varying durations, which requires a method to dynamically add and eliminate DNA species in TXTL reactions. Here, we utilized semi-continuous the microfluidic flow reactors explored in Chapter 3[41,42,63] in combination with TXTL reactions to implement and characterize the synthetic CFFL 1, for which we performed the most extensive characterization in batch reactions, and its reference motif, taking advantage of the controlled inflow and outflow capabilities of the reactors to automatically change the inflow composition to create variable length

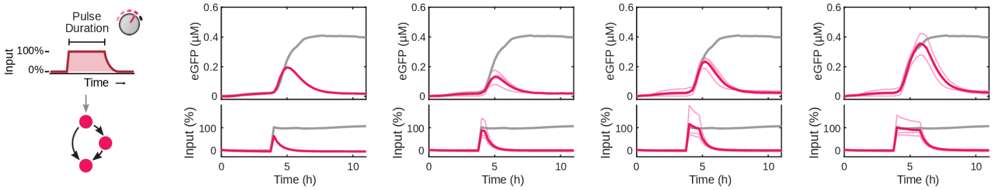


Figure 4.6: Time-varying inputs and corresponding CFFL circuit outputs (pink). A persistent input and corresponding output is shown in all plots as reference (gray).

DNA inputs. After 3 or 4 hours of pre-equilibration without input DNA species, during which all constitutive and background expression equilibrated, persistent step inputs were applied to both the CFFL and reference circuits (Figure 4.5). Like our analysis under batch conditions, endpoint and temporal characteristics were determined. We again observed that background expression in the absence of input is higher in the reference circuit, resulting in a larger *ON/OFF* ratio for the CFFL. The more efficient repression of background expression in the CFFL can be attributed to the sequential stages of repression achieved by the two toehold switches (Supplementary Figure 4.2). Subsequently, DNA input pulses of lengths ranging from 15 minutes to 2 hours were applied to the CFFL to probe for noise-filtering behavior (Figure 4.6). Square input pulses were emulated by initially supplying a high concentration of input DNA to create an immediate onset of signal, followed

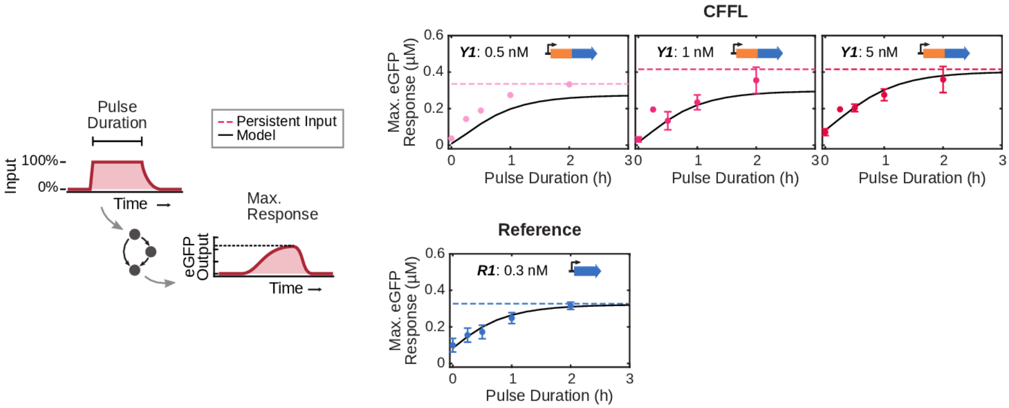


Figure 4.7: Maximum circuit responses for the CFFL (top plots) and reference motif (bottom plot) for no input, inputs of 15 min, 30 min, 1h, 2h (dots; N=3 for all experiments except Y1=0.5 nM and 15 min input pulse duration for Y1=1 nM and Y1=5 nM, for which N=1) and a constitutive input (dashed lines). Error bars represent the standard deviation of the experiments that were performed in triplicate. The ODE model fit to this flow data (Equations 4.1-4.14; Supplementary Table 4.1) is shown as black solid line.

by a variable amount of regular concentration input steps to maintain a high input (Supplementary Table 4.3). Finally, the input signal was terminated through omission of DNA **X1** from the inflow mixture, leading to a decrease of input trigger DNA concentration that was governed by the refresh rate of the microfluidic flow reactors ($t_{½}$ = 25 min).

We applied varying input pulse durations to the reference motif and CFFL, using multiple σ^{28} DNA concentrations, and determined the maximum GFP output for each input pulse as a measure of circuit response (Figure 4.7). Short inputs elicited a response in both the CFFL and reference motif and we observed no clear indication of noise-filtering, although the higher ON/OFF ratio of the CFFL results in a lower output for short inputs. Based on our observation that the σ^{28} DNA concentration is the main contributor to the dynamic behavior of the CFFL in batch TXTL reactions, we next examined the influence of DNA **Y1** concentration on the characteristics of CFFL 1 in semi-continuous flow reactions. While slightly different response dynamics were observed for varying σ^{28} expression strengths, short inputs still propagated through the CFFL motif. Taken together, these experimental results demonstrate that although the synthetic RNA-based CFFL does not display additional noise-filtering characteristics over the reference circuit for a wide range of circuit parameters, it can be utilized to suppress background expression and yield a high fold-change.

4.5 Computational analysis

To demonstrate that the observed experimental behaviors are general properties of the synthetic CFFL, we constructed ordinary differential equation (ODE) models of the CFFL and reference motif and parameterized the models using outputs of the flow reactor experiments and previously determined parameter values (solid lines in Figure 4.7, Supplementary Methods, *Supplementary Table 4.1*).[43] The models were analyzed to predict the *ON*/*OFF* ratio of the circuit output for varying concentrations of σ^{28} encoding DNA in both circuits, including low concentrations that were shown to produce low outputs under batch conditions and are therefore difficult to analyze using flow reactor experiments (Figure 4.8a). The CFFL consistently produced a higher *ON*/*OFF* ratio than the reference circuit.

We further investigated the CFFL through the ODE model and determined temporal ultrasensitivities for varying concentrations of σ^{28} DNA (Figure 4.8b). The tem-

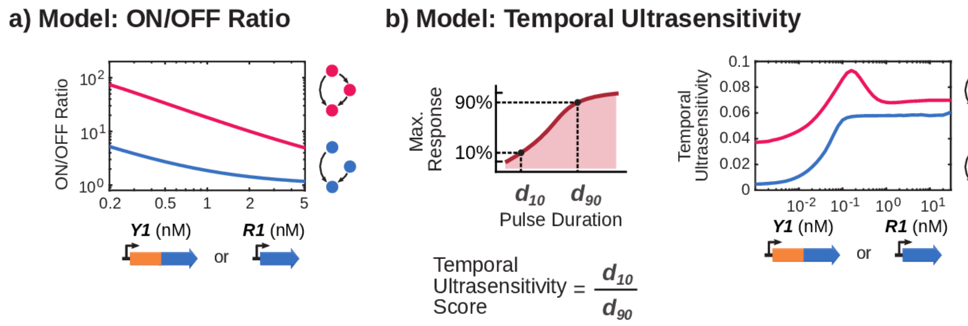


Figure 4.8: a) ODE model predictions for the ON/OFF ratio of the CFFL (pink) and reference motif (blue) under continuous flow conditions. The concentrations of the σ28-producing constructs were varied over the approximate range of experimental conditions and display comparable ON/OFF ratios to the experiments. b) ODE model predictions for the temporal ultrasensitivity of the CFFL (pink) and reference motif (blue) under continuous flow conditions. The concentration of the σ28-producing construct of each circuit was varied over a wide range of concentrations to explore the various behaviors that could be achieved using the circuits, but would be time-consuming to explore in vitro.

poral ultrasensitivity measures how sharp the transition from 10% to 90% of the maximum output is with respect to the input pulse duration (d_{10} and d_{90}, respectively; Equation 4.15).[53] Direct determination of d_{10} and d_{90} from the flow reactor experiments is hampered by the relatively low resolution of the pulse duration domain, which is resolved by utilizing the ODE model to simulate a large amount of input pulse durations input (100 values between 0.01h and 20h evenly distributed on a logarithmic scale) for each condition. Temporal sensitivity quantifies filtering of short-lived inputs, since the transition from a low to high output for a change in input duration should be sharp to create a noise filter which blocks short-lived inputs while retaining a high output for all other signals. The synthetic CFFL displayed very low levels of temporal ultrasensitivity, which were only slightly higher than the reference motif, peaking in a narrow range of σ^{28} DNA concentrations around 0.2 nM.

We explored the temporal ultrasensitivity of the CFFL circuit further by modelling the circuit for a wide range of parameter values using Latin Hypercube sampling (Figure 4.10; *Supplementary Table 4.1*). The model displayed high temporal ultrasensitivity (>0.5) for only 0.3% of the parameter samples (Figure 4.9a). This observation is in line with a recent computational analysis, which revealed that the

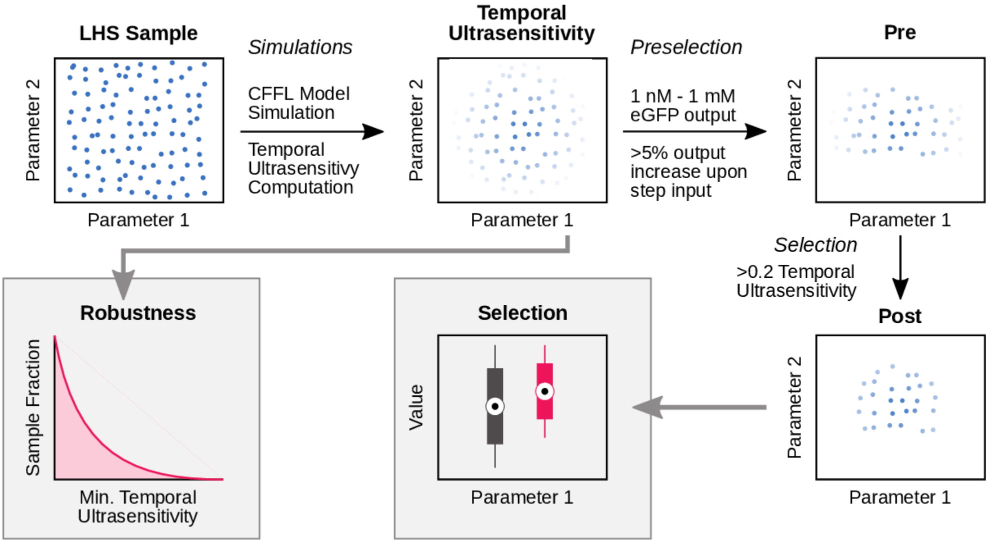


Figure 4.10: Schematic drawing of the in silico parameter sampling method and analysis procedures, shown for a 2-dimensional space for clarity. Latin hypercube sampling was used to create 105 samples of the 13-dimensional logarithmic parameter space. For each parameter sample, the model was evaluated and the temporal ultrasensitivity was computed. To determine the robustness of the temporal ultrasensitivity behavior, for each temporal ultrasensitivity value the fraction of samples that displayed temporal ultrasensitivity of at least that value was computed. Additionally, the samples were filtered based on the computed properties. An initial selection was performed to create a collection of reasonable parameter values (pre). Subsequently, the samples were selected based on the computed temporal ultrasensitivity (post). These two collections were used to analyze parameter value distributions.

temporal ultrasensitivity of CFFL motifs has low robustness and is only significant in a small subset of circuit parameters.[53] To investigate whether our synthetic CFFL could be improved to display stronger noise-filtering behavior, the sampled parameter space was filtered based on the computed temporal ultrasensitivity and statistics of the values of the selected parameter sets were determined. The parameter sets enriched for a high temporal ultrasensitivity were mainly associated with a high Hill-coefficient of the σ^{28} and DNA interaction (Figure 4.9b). Therefore, when incorporating a CFFL in synthetic genetic networks to increase tolerance to noise, one of the many known prokaryotic genetic regulators that binds more cooperatively to DNA[64] should be utilized as delay element (Figure 4.9c). Alternatively, the sharpness of the σ^{28} binding curve could be increased using molecular

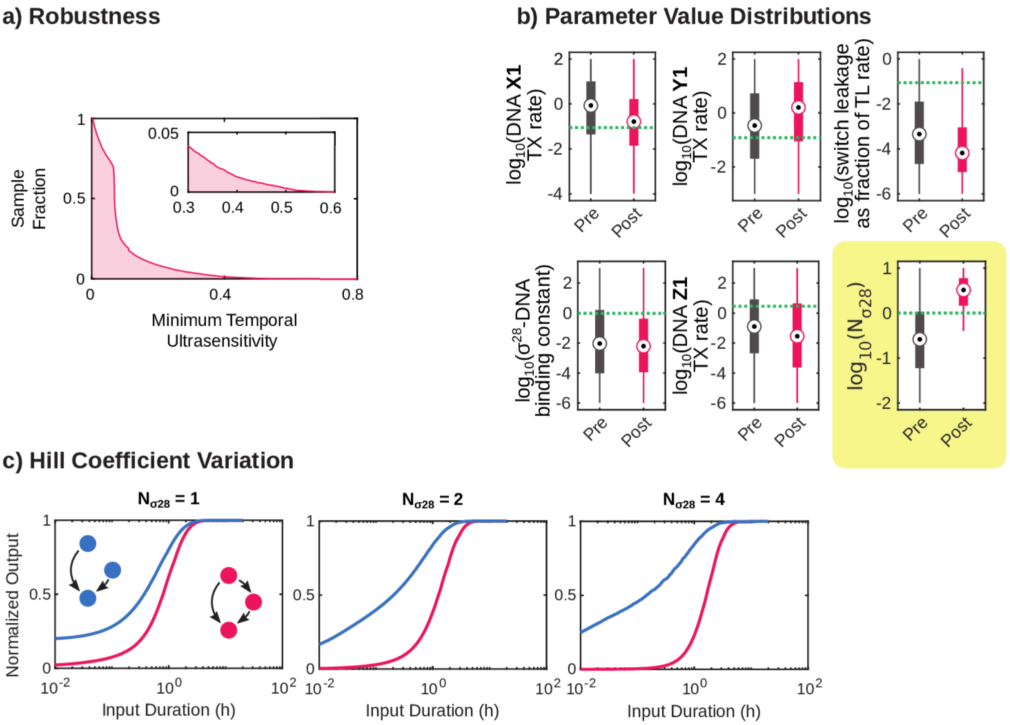


Figure 4.9: a) The fraction of parameter samples that satisfied a minimum temporal ultrasensitivity threshold. A fraction of 0.3%, 8.5% and 21% displayed a temporal ultrasensitivity of at least 0.5, 0.2 and 0.1, respectively. b) The pre and post distributions of parameter values for a selection of parameters (see Supplementary Figure for all parameters and Supplementary Table 4.1 for parameter descriptions and units). The dashed green lines show the parameter values used to simulate the experimental data in this research. A large increase in values between the pre and post sets can be observed for the σ^{28}-binding cooperativity ($N_{\sigma28}$), which indicates that there is a preference for a high Hill coefficient when the temporal ultrasensitivity is high. c) Maximum output amplitudes of simulations of the CFFL and reference motif for a range of input pulse lengths, normalized to the maximum output for a step function input. Simulations with Hill coefficients 1, 2 and 4 demonstrate that the sharpness of the pulse length response of the CFFL increases for higher Hill coefficients, which is reflected in the associated temporal ultrasensitivity values of 0.09, 0.12 and 0.19, respectively.

titration with the anti-σ^{28} factor (FlgM) to achieve a similar effect.[65] Nevertheless, adoption of our synthetic CFFL motif in synthetic circuits can prove to be beneficial in eliminating background expression and improving their fold-change.

4.6 Conclusion

In this chapter, we reveal that toehold switch post-transcriptional regulators can be used to construct modular synthetic genetic networks in TXTL. We combined toehold switches with an *E. coli* sigma-factor to build a synthetic CFFL. The characteristics of the synthetic CFFL were determined under batch TXTL conditions and in a semi-continuous microfluidic flow reactor. By comparing the circuit to a reference motif, we found that the synthetic CFFL could reduce background expression levels, thus increasing the fold-change of the circuit output. We utilized the flexibility of the microfluidic flow reactor to apply time-varying inputs to the synthetic CFFL, but could not identify temporal filtering in the circuit. *In silico* parameter sampling corroborated the observation that this behavior occurs for a small subset of circuit parameters.[53]

Whereas previous analyses of regulatory networks have focused on transcriptional interactions,[52] recent work has shown that small RNAs (sRNA) play key roles in bacterial regulatory networks[44,48,69] and sRNA-based feed-forward loops have been identified.[49,51,69,70] As such, our CFFL-based circuits with RNA species as top regulators can provide a starting point for the characterization of these naturally occurring regulatory elements. Complemented by studies of the integration of synthetic genetic circuits into topologically more complex systems[71] and the role of translational control in regulatory networks of cells,[48] this work provides insight into the function of CFFLs and their application in synthetic biology.

4.7 Methods

4.7.1 Preparation of DNA Templates

DNA constructs were created with golden gate assembly (GGA) using the overlapping sequences adapted from Sun *et al.*[35] (Chapter 2). Promoters, UTR1, coding sequences and terminators were ordered from IDT as gBlock fragments or amplified from the pBEST vector using PCR. Toehold switch and trigger sequences were taken from previous studies in the group of Dr. P. Yin (unpublished, related toehold switch plasmids can be obtained from Addgene (https://www.addgene.org/Alexander_Green/) and PCR amplified. PCR products were gel purified using the QIAquick Gel Extraction Kit (Qiagen) and added in equimolar amounts to GGA assembly reactions with BSAI-HF (NEB), T4 ligase (Promega) and the T4 ligase buffer. GGA reactions were performed in a thermocycler according to a standard GGA protocol.[66] The GGA products were transformed into NovaBlue cells (Merck), from which the plasmids were purified using the QIAprep Spin Miniprep Kit (Qiagen) and the DNA sequences were confirmed using Sanger sequencing.

Linear DNA templates for expression in TXTL reactions were created by PCR using Phusion High-Fidelity DNA Polymerase (NEB) with primers pBEST_LinL_F (GGCGAATCCTCTGACCAGC) and pBEST_LinL_R (CCAAGCTGGACTGTATGCACG) and subsequent purification using QIAquick PCR Purification Kit (Qiagen).

4.7.2 Preparation of Cell Lysate

The *E. coli* cell lysate was prepared from the RNase E deficient BL21 STAR (DE3) (ThermoFisher Scientific) cells that were transformed with the pRARE vector from BL21 Rosettta (Merck). The lysate was prepared according to previously published protocols,[34,36] with the adaptations described in Chapter 2. The *E. coli* strain was grown in 2xYT medium supplemented with 40 mM potassium phosphate dibasic and 22 mM potassium phosphate monobasic until an OD_{600} of 1.7 was reached. The cultures were spun down and washed thoroughly with S30A buffer (14 mM Magnesium L-glutamate, 60 mM Potassium L-glutamate, 50 mM Tris, titrated to pH 8.2 using glacial acetic acid), before being resuspended in 0.9 mL S30A buffer per gram of dry pellet. The cell suspension was lysed using a French press at 16000 lb pressure in two passes and spun down. The supernatant was incubated at 37 °C for 1.5 hours and spun down. The supernatant was dialyzed into S30B buffer (14 mM Magnesium L-glutamate, 150 mM Potassium L-glutamate, titrated to pH 8.2 using 2 M Tris) in two steps for 3 hours total and spun down again. The supernatant was aliquotted, snap-frozen in liquid nitrogen and stored at -80 °C.

The energy mixture was prepared according to the protocol previously described by Sun *et al.*[36] and a constant distribution amino acid solution was prepared.[67]

4.7.3 Preparation of TXTL Reactions

The cell lysate (33% of total reaction volume) was combined with the energy mixture, amino acid solution (37.5 mM), Magnesium L-glutamate (8 mM), PEG-8000 (2%), GamS protein (3 µM; prepared as described by Sun *et al.*[35]) and MilliQ to form the 1.54x TXTL reaction mixture (65% of total reaction volume). The remaining volume (35%) of the reactions was used to add the linear DNA constructs of the gene networks and supplemented to the final volume with MilliQ. The DNA constructs and their concentrations in each experiment are summarized in Supplementary Table 4.2.

4.7.4 Batch TXTL Reactions

Batch TXTL reactions were prepared in total volumes of 10 µL and transferred to 384-wells Nunc plates. The reactions were incubated at 29 °C and eGFP and optionally eCFP fluorescence was measured on a Saffire II (Tecan), Spark 10M (Tecan) or Synergy H1M (Biotek) plate reader for at least 14 hours. The plate readers were calibrated using a titration range of purified eGFP protein.

4.7.5 Microfluidic Device Fabrication and Flow TXTL Reactions

The microfluidic semi-continuous flow reactors were produced using standard soft lithography methods.[41,63]

Semi-continuous flow TXTL reactions were performed according to the protocol previously described by van der Linden *et al.*[63], with adapted Labview control software to enable configuration of time-varying input signals. The 1.54x TXTL reaction mixture was stored on a water-cooled peltier element during the experiment to maintain reactivity of the solution, whereas other reactants were stored in tubing in the incubation chamber (29 °C). The input RNA trigger DNA template was mixed with a DNA-Hexachlorofluorescein conjugate to monitor and verify the applied circuit input pulses (IDT; /5HEX/GGCGAATCCTCTGACCAGC). Concentrations of the DNA constructs used are summarized in Supplementary Table 4.2. Reactions were conducted for 11 h, during which 40% of each reactor was refreshed 15 minutes with 65% TXTL reaction mixture and 35% DNA or MilliQ. To create an input that resembles a square pulse function, an initial step containing 2.5x the final input concentration of DNA X1 was flushed in, after which all subsequent steps contained the regular input DNA concentration. The DNA solutions and operation sequences used to construct the time-dependent input pulses are described in Supplementary Table 4.3. The devices were monitored on an Eclipse Ti-E inverted microscope (Nikon). Reactor channels were automatically detected in the obtained images using a custom Matlab (Mathworks) script (available upon request) and the average fluorescence of a 50x100 pixel rectangle at the center of a channel was calculated to represent the output fluorescence. After 11 hours the reactions were terminated and the microfluidic reactors were flushed with MilliQ. Optionally, microfluidic devices were cleaned for reuse through repeated flushing with a Terg-a-zyme enzyme detergent solution (Alconox).

4.7.6 Ordinary differential equation (ODE) models

We constructed ODE models of the synthetic CFFL and reference motif to simulate flow reactor experiments. Protein species and all RNA species and complexes were taken as state variables of the ODE models. DNA species were defined as parameters that can be adjusted to emulate the dynamic inputs given in the flow experiments. Complex formation of the RNA toehold switches and triggers, constitutive transcription and translation were modeled with a mass action term, while activation of the P28 promoter by the σ^{28} factor was modeled using Hill-type kinetics. Fluorescent protein maturation was not explicitly included, but is captured in the translation rate of the protein. Since the model was exclusively used to model flow reactor experiments, depletion of resources was assumed not to play a major role and was not represented in the ODE models.

4.7.7 ODE Model of the synthetic CFFL

The CFFL model comprises a RNA trigger (T) species that is produced from DNA species **X1** (DNA_T) with rate b_T. SwitchA-S28 mRNA is produced from DNA species Y1 (DNA_{Sw1}) with rate b_{Sw1}. SwitchA-eGFP mRNA is produced by expression from DNA species **Z1** (DNA_{Sw2}) with rate BS_{w2} (background expression) and with rate b_{Sw2} upon activation by the σ^{28} factor (S28) through Hill kinetics (K_{S28} and N_{S28}). The σ^{28} factor was previously shown to behave according to Michaelis-Menten kinetics,[16] so N_{S28} was set to 1 for the parameter fitting procedure, but was varied during parameter sampling (Supplementary Table 4.1). Binding of trigger RNA (T) to SwitchA-S28 mRNA (Sw1) and SwitchA-eGFP mRNA (Sw2) to form complexes TSw1 and TSw2, respectively, was explicitly modeled using mass action kinetics ($k_{on,TSw}$ and $k_{off,TSw}$). Translation from the trigger:switch complexes was represented by rates $k_{tl,S28}$ and $k_{tl,GFP}$. Since translation from unbound switch RNA species can also occur, but at a lower rate determined by the efficiency of the toehold switch, this was included in the model using $b_{gratiosw}$. The resulting translation rate of unbound SwitchA-S28 mRNA is the product of $k_{tl,S28}$ and $b_{gratiosw}$ and the translation rate of unbound SwitchA-eGFP mRNA is the product of $k_{tl,GFP}$ and $bgratio_{sw}$. The degradation rates of all RNA species was represented by parameter aRNA and of all protein species by parameter aprotein. Since the degradation rate of proteins in the TXTL reactions was negligible, as demonstrated by the constant plateaus after 10+ hours of batch TXTL reactions, the value of aprotein was set to zero. The flow rate of the experiment can be found in all equations as parameter kflow, which was set as 1.6 h-1 for all experiments, giving a residence time of 37.5 min. The resulting ODE model of the synthetic CFFL is given by Equations 4.1-4.8, where Equation 4.5 shows a helper function f(..) which is used in the Hill equation in Equation 4.6.

$$\begin{aligned}\frac{dT(t)}{dt} &= b_T \cdot DNA_T + k_{off,TSw} \cdot TSw1(t) - k_{on,TSw} \cdot T(t) \cdot Sw1(t) + k_{off,TSw} \cdot TSw2(t) \\ &\quad - k_{on,TSw} \cdot T(t) \cdot Sw2(t) - (a_{RNA} + k_{flow}) \cdot T(t)\end{aligned} \tag{4.1}$$

$$\begin{aligned}\frac{dSw1(t)}{dt} &= b_{Sw1} \cdot DNA_{Sw1} + k_{off,TSw} \cdot TSw1(t) - k_{on,TSw} \cdot T(t) \cdot Sw1(t) \\ &\quad - (a_{RNA} + k_{flow}) \cdot Sw1(t)\end{aligned} \tag{4.2}$$

$$\frac{dTSw1(t)}{dt} = k_{on,TSw} \cdot T(t) \cdot Sw1(t) - k_{off,TSw} \cdot TSw1(t) - (a_{RNA} + k_{flow}) \cdot TSw1(t) \tag{4.3}$$

$$\frac{dS28(t)}{dt} = k_{tl,S28} * TSw1(t) + k_{tl,S28} \cdot bgratio_{sw} \cdot Sw1(t) - (a_{protein} + k_{flow}) \cdot S28(t) \tag{4.4}$$

$$f(S28(t)) = \left(\frac{S28(t)}{K_{S28}}\right)^{N_{S28}} \tag{4.5}$$

$$\begin{aligned}\frac{dSw2(t)}{dt} &= B_{Sw2} \cdot DNA_{Sw2} + b_{Sw2} \cdot DNA_{Sw2} \cdot \frac{f(S28(t))}{1 + f(S28(t))} + k_{off,TSw} \cdot TSw2(t) \\ &\quad - k_{on,TSw} \cdot T(t) \cdot Sw2(t) - (a_{RNA} + k_{flow}) \cdot Sw2(t)\end{aligned} \tag{4.6}$$

$$\frac{dTSw2(t)}{dt} = k_{on,TSw} \cdot T(t) \cdot Sw2(t) - k_{off,TSw} \cdot TSw2(t) - (a_{RNA} + k_{flow}) \cdot TSw2(t) \tag{4.7}$$

$$\frac{dGFP(t)}{dt} = k_{tl,GFP} \cdot TSw2(t) + k_{tl,GFP} \cdot bgratio_{sw} \cdot Sw2(t) - (a_{protein} + k_{flow}) \cdot GFP(t \tag{4.8}$$

Where species T is the trigger RNA, Sw1 the Switch-S28 RNA and TSw1 the complex of bound Trigger and Switch-S28 RNA. Sw2 is the Switch-eGFP RNA and TSw2 the complex of Trigger and Switch-eGFP RNA. S28 is the σ28 protein species and GFP is the eGFP protein species. DNAX is the concentrations of the

DNA coding for RNA species X (nM). Further parameter descriptions and units can be found in Supplementary Table 4.1.

4.7.8 ODE Model of the reference motif

The model of the reference motif is largely equal to the CFFL model, except for the absense of an interaction between RNA trigger T and mRNA species Sw1. In the reference motif model, RNA species Sw1 represents the S28 RNA (without toehold switch) that results from transcription of DNA species **R1** (DNA_{Sw1}). Since parameter DNA_{Sw1} now represents the concentration of a different physical DNA species (**R1** instead of **Y1**), the transcription rate of this species ($b_{Sw1,ref}$) is assumed to be different than for the CFFL model.

$$\begin{aligned}\frac{dT(t)}{dt} = {} & b_T \cdot DNA_T + k_{off,TSw} \cdot TSw2(t) - k_{on,TSw} \cdot T(t) \cdot Sw2(t) \\ & -(a_{RNA} + k_{flow}) \cdot T(t)\end{aligned} \tag{4.9}$$

$$\frac{dSw1(t)}{dt} = b_{Sw1,ref} \cdot DNA_{Sw1} - (a_{RNA} + k_{flow}) \cdot Sw1(t) \tag{4.10}$$

$$\frac{dS28(t)}{dt} = k_{tl,S28} * Sw1(t) - (a_{protein} + k_{flow}) \cdot S28(t) \tag{4.11}$$

$$\begin{aligned}\frac{dSw2(t)}{dt} = {} & B_{Sw2} \cdot DNA_{Sw2} + b_{Sw2} \cdot DNA_{Sw2} \cdot \frac{f(S28(t))}{1 + f(S28(t))} + k_{off,TSw} \cdot TSw2(t) \\ & - k_{on,TSw} \cdot T(t) \cdot Sw2(t) - (a_{RNA} + k_{flow}) \cdot Sw2(t)\end{aligned} \tag{4.12}$$

$$\frac{dTSw2(t)}{dt} = k_{on,TSw} \cdot T(t) \cdot Sw2(t) - k_{off,TSw} \cdot TSw2(t) - (a_{RNA} + k_{flow}) \cdot TSw2(t) \tag{4.13}$$

$$\frac{dGFP(t)}{dt} = k_{tl,GFP} \cdot TSw2(t) + k_{tl,GFP} \cdot bgratio_{sw} \cdot Sw2(t) - (a_{protein} + k_{flow}) \cdot GFP(t \tag{4.14}$$

Where species T is the trigger RNA and Sw1 is the S28 RNA (without any toehold switch). Sw2 is the Switch-eGFP RNA and TSw2 the complex of Trigger and Switch-eGFP RNA. S28 is the σ^{28} protein species and GFP is the eGFP protein species. DNAX is the concentrations of the DNA coding for RNA species X (nM). Further parameter descriptions and units can be found in Supplementary Table 4.1.

4.7.9 Temporal Ultrasensitivity Computation

The temporal ultrasensitivity calculations were based on the α and β measures, calculated from the maximum output amplitude per input pulse length (Figure 4C and D).[6] α is the pulse duration at which the maximum response has increased 10% of the total difference between background expression (no input) and full activation (persistent step input), computed by linearly interpolating the nearest experimental or simulation data points. Similarly, β marks the pulse duration of 90% increase in response. The temporal ultrasensitivity is defined as follows:

$$temporal\,ultrasensitivity = \frac{\alpha}{\beta} \tag{4.15}$$

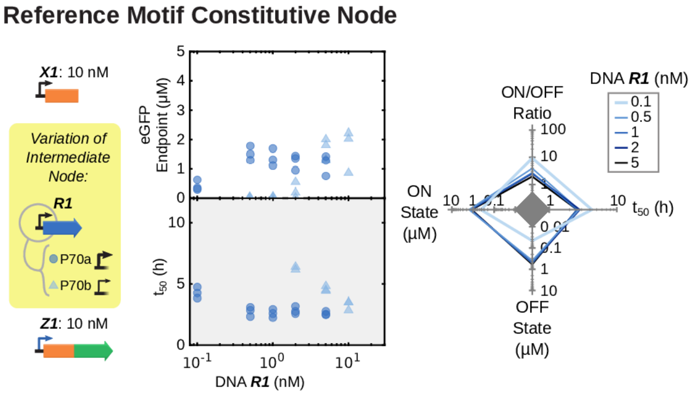
Thus, the temporal ultrasensitivity is low for a gradually increasing activation upon an increase in input pulse length and approaches 1 for a circuit that approaches immediate switch-like behavior at a given pulse length.

4.7.10 Parameter Fitting and Sampling

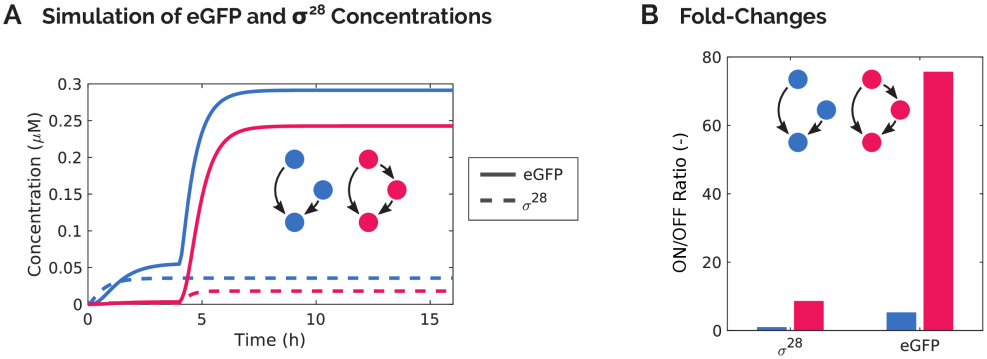
The ODE models of the CFFL and reference motif were implemented in Matlab (Mathworks) and numerically solved using the ode15s solver. We utilized the lsqnonlin solver using the trust-region-reflective algorithm to parameterize the ODE models. The model parameters were simultaneously fitted on a logarithmic scale to all flow reactor experimental data, with 10^3 Latin hypercube samples provided to the solver as initial parameter sets to prevent the fit from being only locally optimal. The fitted parameters and the resulting parameter set, that was used to perform further *in silico* experiments, is provided in Supplementary Table 4.1.

To screen the behavior of the CFFL outside of the experimental parameter regime, Latin hypercube sampling (lhsdesign) was employed to generate 10^5 parameter samples of wide range of parameter values in logarithmic space (see Supplementary Table 4.1 for the parameters and their ranges). The CFFL ODE model was evaluated for all parameter samples to map the temporal ultrasensitivity of the CFFL. The network was simulated without input for 10 h, then 20 different time durations with input (0.01 – 20 evenly distributed on a logarithmic scale) and finally 4 h without input. From the maximum eGFP outputs of these 20 simulated experiments the temporal ultrasensitivity was computed (Equation 4.15). Parameter samples that resulted in a model that could not be correctly solved by ode15s were excluded from the analysis. To determine parameter values corresponding to a high temporal ultrasensitivity score, two stages of selection of the parameter samples were applied. First, samples were selected for a maximum eGFP output between 1 nM and 1 mM and a minimum increase of 5% of the maximum output upon addition of input trigger. The subsequent selection was conducted based on a minimum temporal ultrasensitivity of 0.2. The sampling and selection procedures are illustrated in Figure 4.10 and Supplementary Figure 4.3.

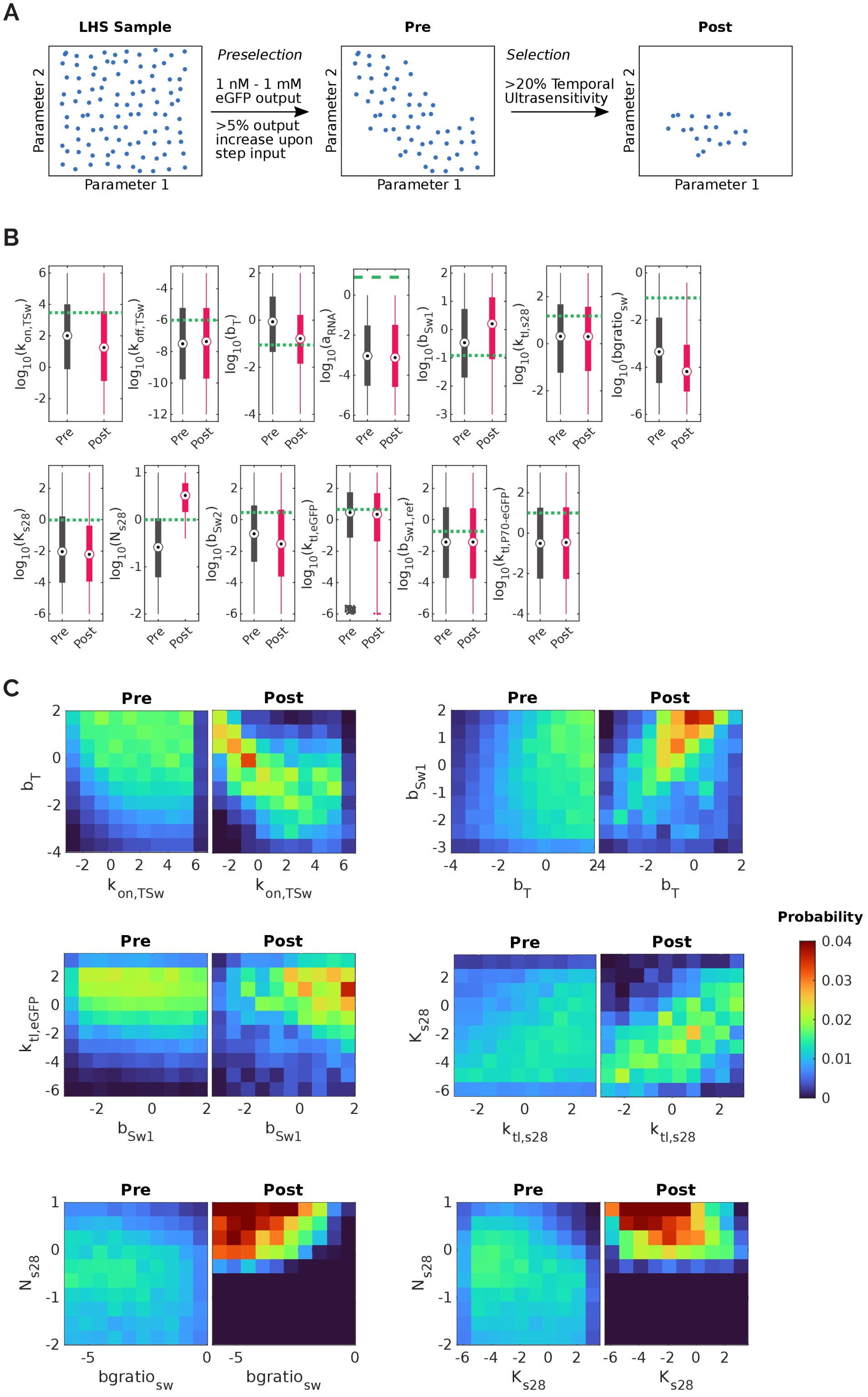
4.8 Supplementary Figures and Tables



*Supplementary Figure 4.1: Endpoint concentrations, t_{50} and trade-offs for the reference motif with varying concentrations of σ^{28}-encoding DNA construct (DNA **R1**). In addition to the default construct (circles), a construct with a point mutation in the promoter to give a low efficiency σ^{28}-producing gene was evaluated (triangles). The trade-offs are shown for the default high-yield σ^{28} DNA construct. N=3 for all conditions. All experimental conditions are summarized in Supplementary Table 4.2.*



Supplementary Figure 4.2: Simulations of the influence of the presence of toehold switches at two positions in the CFFL circuit. A) Simulated concentration traces for the CFFL (pink) and reference motif (blue) of both the eGFP output (solid) and σ^{28} protein concentration. B) The ON/OFF ratios of σ^{28} and eGFP in the CFFL and reference circuits. The σ^{28} concentration of the reference motif remains constant after addition of an input, since it is constitutively produced. The eGFP output of the reference motif displays an ON/OFF ratio of 8.6, which is caused only by the toehold switch on the output construct, since the other switch is not present in the reference motif. The fold-change in the σ^{28} concentration of the CFFL mainly represents the activation of the toehold switch on the switch-σ^{28} construct and was observed to be 5.3. Together, the activation of the switches largely accounts for the high fold-change in eGFP output of the CFFL, leaving a factor 1.7 to be accounted for by other factors, such as differences in parameter values between the reference motif and CFFL and non-linearity of the σ^{28}-DNA interaction.


A
LHS Sample
Parameter 2
Parameter 1
Preselection
1 nM - 1 mM eGFP output
>5% output increase upon step input
Pre
Selection
>20% Temporal Ultrasensitivity
Post
B
$\log_{10}(k_{on,TSw})$
$\log_{10}(k_{off,TSw})$
$\log_{10}(b_T)$
$\log_{10}(a_{RNA})$
$\log_{10}(b_{Sw1})$
$\log_{10}(k_{tl,s28})$
$\log_{10}(bgratio_{sw})$
$\log_{10}(K_{s28})$
$\log_{10}(N_{s28})$
$\log_{10}(b_{Sw2})$
$\log_{10}(k_{tl,eGFP})$
$\log_{10}(b_{Sw1,ref})$
$\log_{10}(k_{tl,P70\text{-}eGFP})$
Pre
Post
C
Pre
Post
b_T
$k_{on,TSw}$
b_{Sw1}
b_T
$k_{tl,eGFP}$
b_{Sw1}
K_{s28}
$k_{tl,s28}$
N_{s28}
$bgratio_{sw}$
N_{s28}
K_{s28}
Probability
0.04
0.03
0.02
0.01
0

Supplementary Figure 4.3: In silico selection of parameter samples of the CFFL model that display high temporal ultrasensitivity. a) Schematic that shows the selection procedure of the parameter samples. A latin hypercube sampling (LHS) of the 13-dimensional parameter space was taken in the logarithmic domain. For each parameter sample, the temporal ultrasensitivity and general output statistics of the CFFL were computed. An initial selection was performed based on the general output measures to ensure reasonable output protein concentrations (between 1 nM and 1 mM) and discernible activation when an input signal is given (at least 5%). The resulting selection of parameter samples is denoted as 'Pre'. The subsequent step is to select for parameter samples that result in temporal ultrasensitivity higher than 0.2, yielding the 'Post' parameter collection. Comparison of the parameter values found in the Pre and Post collections gives information about the preference for certain parameter values for circuits with high temporal ultrasensititivity. b) The distribution of the values of each circuit parameter in the Pre and Post collections of parameter sets visualized as box plots with a dashed green line indicating the parameter value obtained from a fit to the experimental data. For most parameters, the Pre and Post distributions are comparable, meaning that those parameter values are equally likely to be found before and after selection for a high temporal ultrasensitivity. The parameters that display the largest shift in distribution between Pre and Post are $bgratio_{sw}$, the fraction the translation rate of the bound switch:trigger complex that is observed as leakage in the unbound toehold switch, and N_{S28}, which is the Hill coefficient of the binding of σ^{28} to its promoter. A high temporal ultrasensitivity is associated with a lower $bgratio_{sw}$ and a higher N_{S28}. c) 2D distributions of parameter values of a select combination of parameters, displaying the Pre and Post collections.

Parameter Name	Unit	Description	Value	Sampling bounds
k_{flow}	h^{-1}	Microfluidic reactor flow rate	1.6	-
$k_{on,TSw}$	$\mu M^{-1}h^{-1}$	Trigger and switch association rate	$3 \cdot 10^{3}$	10^{-3} - 10^{6}
$k_{off,TSw}$	h^{-1}	Trigger and switch dissociation rate	$1 \cdot 10^{-6}$	10^{-12} - 10^{-3}
b_T	$\mu M\ h^{-1}$ $(nM\ DNA)^{-1}$	Trigger transcription rate	0.09	10^{-4} - 10^{2}
a_{RNA}	h^{-1}	RNA degradation rate	7.9	10^{-6} - 10^{0}
b_{Sw1}	$\mu M\ h^{-1}$ $(nM\ DNA)^{-1}$	Switch-S28 construct transcription rate	0.12	10^{-3} - 10^{2}
$k_{tl,S28}$	h^{-1}	S28 translation rate	15	10^{-3} - 10^{3}
$bgratio_{Sw}$	-	Fraction of translation rate observed from unbound switch	0.09	10^{-6} - 10^{0}
$a_{protein}$	h^{-1}	Protein degradation rate	0	-
K_{S28}	μM	S28 binding constant	0.96	10^{-6} - 10^{3}
N_{S28}	-	Hill coefficient	1	10^{-2} - 10^{1}
B_{Sw2}	$\mu M\ h^{-1}$ $(nM\ DNA)^{-1}$	Background expression of Switch-eGFP construct.	0	-
b_{Sw2}	$\mu M\ h^{-1}$ $(nM\ DNA)^{-1}$	Switch-eGFP construct transcription rate	2.88	10^{-6} - 10^{3}
$k_{tl,GFP}$	h^{-1}	eGFP translation rate	4.5	10^{-4} - 10^{3}
$b_{Sw1,ref}$	$\mu M\ h^{-1}$ $(nM\ DNA)^{-1}$	S28 reference construct transcription rate	0.18	10^{-6} - 10^{3}
DNA_T	nM	Trigger DNA (***X1***) concentration.	10	-
DNA_{Sw1}	nM	DNA ***Y1*** (CFFL) or ***R1*** (Reference) concentration.	1	-
DNA_{Sw2}	nM	Output DNA (***Z1***) concentration.	10	-

Supplementary Table 4.1: Model parameter, their units, used values and sampled parameter range. Sample values marked with an asterisk (*) were obtained through parameter fitting. The DNA concentrations (DNA$_T$, DNA$_{Sw1}$ and DNA$_{Sw2}$) are the concentrations used for parameter sampling. The concentrations used in other in silico experiments are given in Supplementary Table 4.2.

Figure	Type	Samples	DNA	DNA Name	Concentration (nM)
4.2a	Batch	All	Y1/Y3	P70a-SwitchA-S28	1
		All	Z1	P28a-SwitchA-eGFP	10
		On-Target	X1/X3	P70a-TriggerA	10
		Off-Target	X2	P70a-TriggerB	10
4.2b	Batch	All	R1	P70a-S28	0.5
		All	Z1	P28a-SwitchA-eGFP	10
		On-Target	X1/X3	P70a-TriggerA	10
4.4a	Batch	All	Y1/Y3	P70a-SwitchA-S28	0.1, 0.3, 0.5, 1, 2, 5
		All	Z1	P28a-SwitchA-eGFP	10
		ON State	X1/X3	P70a-TriggerA	10
4.4b	Batch	All	Y1/Y3	P70a-SwitchA-S28	5
		All	Z1	P28a-SwitchA-eGFP	1, 2, 5, 10, 20
		ON State	X1/X3	P70a-TriggerA	10
4.5	Flow	All: *ON* State	X1/X3	P70a-TriggerA	10
		All: All	Z1	P28a-SwitchA-eGFP	10
		CFFL: All	Y1/Y3	P70a-SwitchB-S28	0.2, 0.5, 1, 5
		Reference Motif: All	R1	P70a-S28	0.3
4.6	Flow	*ON* State	X1/X3	P70a-TriggerA	10
		All	Z1	P28a-SwitchA-eGFP	10
		All	Y1/Y3	P70a-SwitchB-S28	1

		Input durations: 15, 30, 60, 120, persistent			
4.7	Flow	All: *ON* State	X1/X3	P70a-TriggerA	10
		All: All	Z1	P28a-SwitchA-eGFP	10
		CFFL: All	Y1/Y3	P70a-SwitchB-S28	0
		Reference Motif: All	R1	P70a-S28	0.3
		Input durations: 0, 15, 30, 60, 120, persistent			
4.8a/b	Model	All: *ON* State	X1/X3	P70a-TriggerA	10
		All: All	Z1	P28a-SwitchA-eGFP	10
		CFFL: All	Y1/Y3	P70a-SwitchB-S28	10^{-3} - 10^{2}
		Reference Motif: All	R1	P70a-S28	10^{-3} - 10^{2}
		Input durations: logspace(-2,1.3,100)			
4.10	Model	All	X1/X3	P70a-TriggerA	10
		All: All	Z1	P28a-SwitchA-eGFP	10
		CFFL: All	Y1/Y3	P70a-SwitchB-S28	1
		Reference Motif: All	R1	P70a-S28	1
		Input durations: 10^{-2} - 10^{2}			
SFig 4.1	Batch	P70a/b: All	Z1	P28a-SwitchA-eGFP	10
		Strong Promoter	R1	P70a-S28	0.1, 0.5, 1, 2, 5
		Weak promoter	R1	P70b-S28	0.5, 1, 2, 5, 5, 10
		P70a/b: *ON* State	X1/X3	P70a-TriggerA	10
SFig 4.2	Model	All: *ON* State	X1/X3	P70a-TriggerA	10
		CFFL: All	Y1/Y3	P70a-SwitchB-S28	1
		Reference Motif: All	R1	P70a-S28	1

All: All	Z1	P28a-SwitchA-eGFP	10

Supplementary Table 4.2: DNA construct concentrations of all batch, flow and in silico experiments.

Experiment	Sample	DNA ***X1*** (nM)	DNA ***Y1*** (nM)	DNA ***R1*** (nM)	DNA ***Z1*** (nM)
CFFL, ***Y1*** = 0.2 nM	*No input*	0	0.57	0	29
	High input	71	0.57	0	29
	Low input	29	0.57	0	29
CFFL, ***Y1*** = 0.5 nM	*No input*	0	1.4	0	29
	High input	71	1.4	0	29
	Low input	29	1.4	0	29
CFFL, ***Y1*** = 1 nM	*No input*	0	2.9	0	29
	High input	71	2.9	0	29
	Low input	29	2.9	0	29
CFFL, ***Y1*** = 5 nM	*No input*	0	14	0	29
	High input	71	14	0	29
	Low input	29	14	0	29
Reference Motif, ***R1*** = 0.3 nM	*No input*	0	0	0.86	29
	High input	71	0	0.86	29
	Low input	29	0	0.86	29

Supplementary Table 4.3: Composition of the DNA solutions used in flow experiments. A flow experiment of an initial fill of the reactor with TXTL reaction mixture and 35% reactor volume No input solution. Subsequently, every 15 minutes 40% of the reactor was refreshed with a mixture consisting of 65% TXTL mixture and the remaining 45% one of the DNA solutions. The sequence of DNA solutions was: 11 or 15 steps No input; 1 step High input; 0, 1, 3 or 7 steps Low input (to create 15 min, 30 min, 1 h and 2 h input pulses) and the remaining steps No input until 11 h of reaction time were reached. Additionally, a negative control without input signal was constructed by only supplying the No input solution. The persistent input experiments were conducted using the following sequence: 11 or 15 steps No input; 1 step High input and the remaining steps Low input.

4.9 References

(1) Nitzan, M., Rehani, R., and Margalit, H. (2017) Integration of Bacterial Small RNAs in Regulatory Networks. *Annu. Rev. Biophys. 46*, 131–148.

(2) Papenfort, K., Espinosa, E., Casadesús, J., and Vogel, J. (2015) Small RNA-based feedforward loop with AND-gate logic regulates extrachromosomal DNA transfer in *Salmonella*. *Proc. Natl. Acad. Sci. 112*, E4772–E4781.

(3) Tej, S., Gaurav, K., and Mukherji, S. (2019) Small RNA driven feed-forward loop: critical role of sRNA in noise filtering. *Phys. Biol.*

(4) Thomason, M. K., Fontaine, F., De Lay, N., and Storz, G. (2012) A small RNA that regulates motility and biofilm formation in response to changes in nutrient availability in Escherichia coli: sRNA regulator of motility and biofilm formation. *Mol. Microbiol. 84*, 17–35.

(5) Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network Motifs: Simple Building Blocks of Complex Networks *298*, 5.

(6) Gerardin, J., Reddy, N. R., and Lim, W. A. (2019) The Design Principles of Biochemical Timers: Circuits that Discriminate between Transient and Sustained Stimulation. *Cell Syst. 9*, 297-308.e2.

(7) Mangan, S., and Alon, U. (2003) Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci. 100*, 11980–11985.

(8) Mangan, S., Zaslaver, A., and Alon, U. (2003) The Coherent Feedforward Loop Serves as a Sign-sensitive Delay Element in Transcription Networks. *J. Mol. Biol. 334*, 197–204.

(9) Purvis, J. E., and Lahav, G. (2013) Encoding and Decoding Cellular Information through Signaling Dynamics. *Cell 152*, 945–956.

(10) Purvis, J. E., Karhohs, K. W., Mock, C., Batchelor, E., Loewer, A., and Lahav, G. (2012) p53 Dynamics Control Cell Fate. *Science 336*, 1440–1444.

(11) Yosef, N., and Regev, A. (2011) Impulse Control: Temporal Dynamics in Gene Transcription. *Cell 144*, 886–896.

(12) Albeck, J. G., Mills, G. B., and Brugge, J. S. (2013) Frequency-Modulated Pulses of ERK Activity Transmit Quantitative Proliferation Signals. *Mol. Cell 49*, 249–261.

(13) Behar, M., Dohlman, H. G., and Elston, T. C. (2007) Kinetic insulation as an effective mechanism for achieving pathway specificity in intracellular signaling networks. *Proc. Natl. Acad. Sci. 104*, 16146–16151.

(14) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *Proc. Natl. Acad. Sci. 110*, 15985–15990.

(15) Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., and Maerkl, S. J. (2015) Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* (Simmel, F., Ed.) *4*, e09771.

(16) Yelleswarapu, M., van der Linden, A. J., van Sluijs, B., Pieters, P. A., Dubuc, E., de Greef, T. F. A., and Huck, W. T. S. (2018) Sigma Factor-Mediated Tuning of Bacterial Cell-Free Synthetic Genetic Oscillators. *ACS Synth. Biol. 7*, 2879–2887.

(17) van der Linden, A. J., Yelleswarapu, M., Pieters, P. A., Swank, Z., Huck, W. T. S., Maerkl, S. J., and de Greef, T. F. A. (2019) A Multilayer Microfluidic Platform for the Conduction of Prolonged Cell-Free Gene Expression. *J. Vis. Exp.* e59655.

(18) Mangan, S., Zaslaver, A., and Alon, U. (2003) The Coherent Feedforward Loop Serves as a Sign-sensitive Delay Element in Transcription Networks. *J. Mol. Biol. 334*, 197–204.

(19) Mangan, S., and Alon, U. (2003) Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci. 100*, 11980–11985.

(20) Karzbrun, E., Tayar, A. M., Noireaux, V., and Bar-Ziv, R. H. (2014) Programmable on-chip DNA compartments as artificial cells. *Science 345*, 829–832.

(21) Yelleswarapu, M., van der Linden, A. J., van Sluijs, B., Pieters, P. A., Dubuc, E., de Greef, T. F. A., and Huck, W. T. S. (2018) Sigma Factor-Mediated Tuning of Bacterial Cell-Free Synthetic Genetic Oscillators. *ACS Synth. Biol. 7*, 2879–2887.

(22) Green, A. A., Silver, P. A., Collins, J. J., and Yin, P. (2014) Toehold Switches: De-Novo-Designed Regulators of Gene Expression. *Cell 159*, 925–939.

(23) Nitzan, M., Rehani, R., and Margalit, H. (2017) Integration of Bacterial Small RNAs in Regulatory Networks. *Annu. Rev. Biophys. 46*, 131–148.

(24) Tej, S., Gaurav, K., and Mukherji, S. (2019) Small RNA driven feed-forward loop: critical role of sRNA in noise filtering. *Phys. Biol. 16*, 046008.

(25) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *Proc. Natl. Acad. Sci. 110*, 15985–15990.

(24) Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., and Maerkl, S. J. (2015) Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife 4*, DOI: 10.7554/eLife.09771.

(44) van der Linden, A. J., Yelleswarapu, M., Pieters, P. A., Swank, Z., Huck, W. T. S., Maerkl, S. J., and de Greef, T. F. A. (2019) A Multilayer Microfluidic Platform for the Conduction of Prolonged Cell-Free Gene Expression. *J. Vis. Exp.* e59655, DOI: 10.3791/59655.

(28) Gerardin, J., Reddy, N. R., and Lim, W. A. (2019) The Design Principles of Biochemical Timers: Circuits that Discriminate between Transient and Sustained Stimulation. *Cell Syst. 9*, 297-308.e2.

(29) Stanton, B. C., Nielsen, A. A. K., Tamsir, A., Clancy, K., Peterson, T., and Voigt, C. A. (2014) Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nat. Chem. Biol. 10*, 99–105.

(30) Buchler, N. E., and Louis, M. (2008) Molecular Titration and Ultrasensitivity in Regulatory Networks. *J. Mol. Biol. 384*, 1106–1119.

(31) Milo, R. (2002) Network Motifs: Simple Building Blocks of Complex Networks. *Science 298*, 824–827.

(32) Beisel, C. L., and Storz, G. (2011) The Base-Pairing RNA Spot 42 Participates in a Multioutput Feedforward Loop to Help Enact Catabolite Repression in Escherichia coli. *Mol. Cell 41*, 286–297.

(33) Wagner, E. G. H., and Romby, P. (2015) Small RNAs in Bacteria and Archaea, in *Advances in Genetics*, pp 133–208. Elsevier, Amsterdam.

(34) Papenfort, K., Espinosa, E., Casadesús, J., and Vogel, J. (2015) Small RNA-based feedforward loop with AND-gate logic regulates extrachromosomal DNA transfer in *Salmonella*. *Proc. Natl. Acad. Sci. 112*, E4772–E4781.

(35) Thomason, M. K., Fontaine, F., De Lay, N., and Storz, G. (2012) A small RNA that regulates motility and biofilm formation in response to changes in nutrient availability in Escherichia coli: sRNA regulator of motility and biofilm formation. *Mol. Microbiol. 84*, 17–35.

(36) Mank, N. N., Berghoff, B. A., and Klug, G. (2013) A mixed incoherent feed-forward loop contributes to the regulation of bacterial photosynthesis genes. *RNA Biol. 10*, 347–352.

(37) Purnick, P. E. M., and Weiss, R. (2009) The second wave of synthetic biology: from modules to systems. *Nat. Rev. Mol. Cell Biol. 10*, 410–422.

(38) Sun, Z. Z., Yeung, E., Hayes, C. A., Noireaux, V., and Murray, R. M. (2014) Linear DNA for Rapid Prototyping of Synthetic Biological Circuits in an *Escherichia coli* Based TX-TL Cell-Free System. *ACS Synth. Biol. 3*, 387–397.

(52) Engler, C., Kandzia, R., and Marillonnet, S. (2008) A One Pot, One Step, Precision Cloning Method with High Throughput Capability. *PLoS ONE 3*, e3647, DOI: 10.1371/journal.pone.0003647..

(40) Garamella, J., Marshall, R., Rustad, M., and Noireaux, V. (2016) The All E. coli TX-TL Toolbox 2.0: A Platform for Cell-Free Synthetic Biology. *ACS Synth. Biol. 5*, 344–355.

(18) Sun, Z. Z., Hayes, C. A., Shin, J., Caschera, F., Murray, R. M., and Noireaux, V. (2013) Protocols for Implementing an Escherichia coli Based TX-TL Cell-Free Expression System for Synthetic Biology. *J. Vis. Exp.* 50762, DOI: 10.3791/50762.

(42) Caschera, F., and Noireaux, V. (2015) Preparation of amino acid mixtures for cell-free expression systems. *BioTechniques 58*, 40–43.

5

Composite Synthetic Coherent Feed-Forward Loops

Part of this work has been published as:

Pascal A. Pieters, Bryan L. Nathalia, Ardjan J. van der Linden, Peng Yin, Jongmin Kim, Wilhelm T. S. Huck, and Tom F. A. de Greef. *ACS Synthetic Biology,* ***10****, 1406-1416 (2021)*

5.1 Introduction

The identification and study of network motifs, such as the feed-forward loop, in gene regulatory networks has provided insight in the relation between network topologies and their biological function. The *in vitro* construction genetic circuits that represent well-defined network motifs has enabled the identification and verification of behaviors their corresponding parameter ranges. For example, conditions for sustained oscillations of 2-node activator-repressor networks[1,2], 3-node repressilator networks[3,4] and larger ring oscillators[4] have successfully been identified *in vitro*. In Chapter 4, we have applied the same principles to characterize the behavior of a synthetic coherent feed-forward circuit. Thorough characterization of synthetic implementations of these network motifs provides a starting point to the incorporation of the motifs in more complex synthetic genetic networks for synthetic biology applications.

Designing larger networks with well-defined complex functionalities is nevertheless a non-trivial task. In larger networks, motifs may not retain the behavior they exhibit in isolation due to unwanted interactions, resource sharing[5] or retroactivity.[6,7] Furthermore, more complex functionalities may require genetic circuits that do not contain well-defined isolated motifs. This was illustrated by Burda et al.[8], who demonstrated that when a gene network was evolved *in silico* to a simple function, the resulting network comprised well-defined motifs. However, when more complex network functions were chosen, the motifs did not fulfill a similarly isolated function.

Nevertheless, higher-order organizations of network motifs can be identified in regulatory gene networks. Using a generalized version of the approach taken to discover small network motifs, such as the feed-forward loop, by Milo et al.[9], Shen-Orr et al.[10] identified larger motifs in the transcriptional regulation network of E. coli. Firstly, the single input module (SIM) was identified, where a single gene is the sole regulator for a group of operons. This pattern can be observed for clusters of functionally related genes, for example the ArgR transcription factor, which uniquely controls five genes related to the synthesis of arginine. The maximum size of SIM motifs is in practice limited due to the low probability that every operon merely has a single regulator.

More commonly, genes are connected in dense overlapping regulons (DOR), where multiple transcription factors regulate multiple operons in a dense layer of interactions. These DOR motifs are subsequently connected to other motifs using few interactions and thus represent operons that share a functionality, such as the σ^{38}-regulated entry to the stationary phase.

The SIM and DOR motifs are topologically simple and do not comprise feed-forward or feedback interactions. In contrast, the understanding of ordering and clustering of small motifs, such as feed-forward and feedback loops, into large structures is limited. Gorochowski et al.[11] recently elucidated the organization of feed-forward network motifs in the E. coli transcriptional regulation network. They found that clustering of feed-forward motifs is more significant than expected from random clustering of the highly abundant feed-forward loops and determined the frequency of all clusters (Figure 5.1). In transcription networks, two types of feed-forward loop clusters were overrepresented, both consisting of a shared input (transcription factor) that regulate both outputs via individual intermediates (type 6) or a shared intermediate (type 12). Structurally, these clusters are simple and similar to SIM and DOR motifs, with single transcription factors controlling multiple output genes. Moreover, the type 12 feed-forward loop organization has been demonstrated to encode the temporal ordering of expression of the output genes, based on the activation threshold of each output.[12]

Other types of feed-forward loop clusterings are not as abundant in transcription networks, but may still have characteristic functionalities associated with their topologies. In this chapter, we demonstrate that the coherent feed-forward loop design we developed in the previous chapters can function as the basis to construct and characterize higher-order organizations of coherent feed-forward loops. By combining two new variants of our initial CFFL design, we create a composite CFFL circuit (type 4) and characterize its response to various combinations of inputs using batch TXTL reactions.

5.2 CFFL Variants

To demonstrate the feasibility of implementing these CFFL-based circuits with increased topological complexity in TXTL reactions, we constructed two new CFFL variations (CFFL 2 and 3) based on our initial design (CFFL 1) (Figure 5.2). Using the modular cloning method described in Chapter 2 (see Methods), we imple-

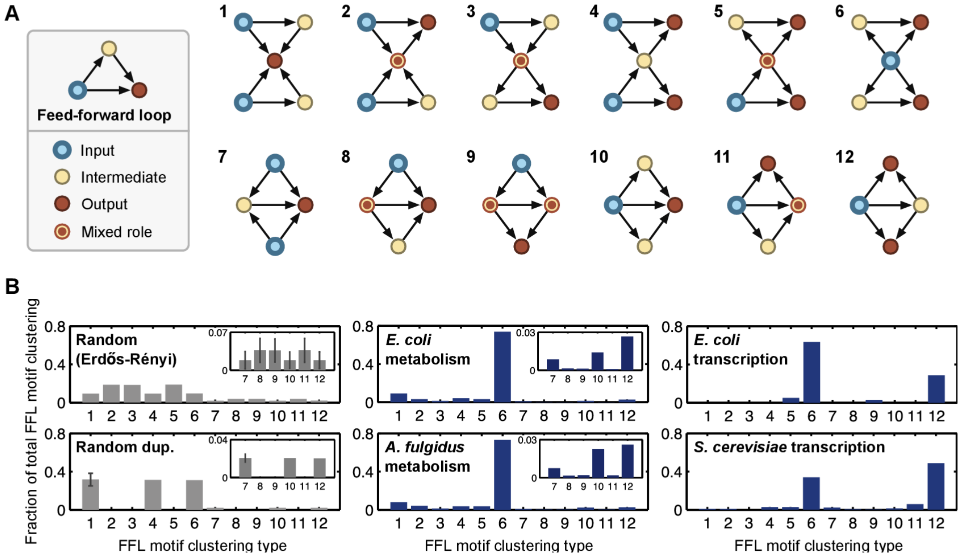


Figure 5.1: a) Schematics of the 12 types of clusters identified by Gorochowski et al.[11] in various networks. The color of each node indicates its function in the network. b) Distributions of the types of FFL clusters found in various networks, including two types of random control networks. From Gorochowski et al.[11]. Reprinted with permission from AAAS.

mented an orthogonal CFFL circuit (CFFL 2) by replacing the toehold switch and trigger (switch/trigger A) of the initial design with a largely orthogonal switch and trigger pair (switch/trigger B), selected from two switch/trigger pairs evaluated for their dynamic range in TXTL, to create trigger B input construct **X2**, switch B-σ^{28} intermediate **Y2** and switch B-eGFP output construct **Z2**. Secondly, we constructed a CFFL variant with eCFP as output protein while maintaining the switch/trigger A combination (CFFL 3), containing the switch A-eGFP output DNA construct **Z3**. Both alternative implementations achieved comparable expression levels and retained a clear distinction between on and off states (Figure 5.2).

In a similar manner, using the same design and cloning method, the σ^{28} intermediate node can be replaced by alternative sigma factors, many of which have been successfully expressed in TXTL reactions.[13] In addition, a large number of orthogonal toehold switch and trigger pairs have been engineered[14] and multiple fluorescent protein have been optimized for expression in TXTL reactions.[15] Taken together, the availability of variants of each component of the CFFL design enables the creation of a library of CFFL variants, which can be used to construct various types of composite CFFLs or larger organizations of CFFLs.

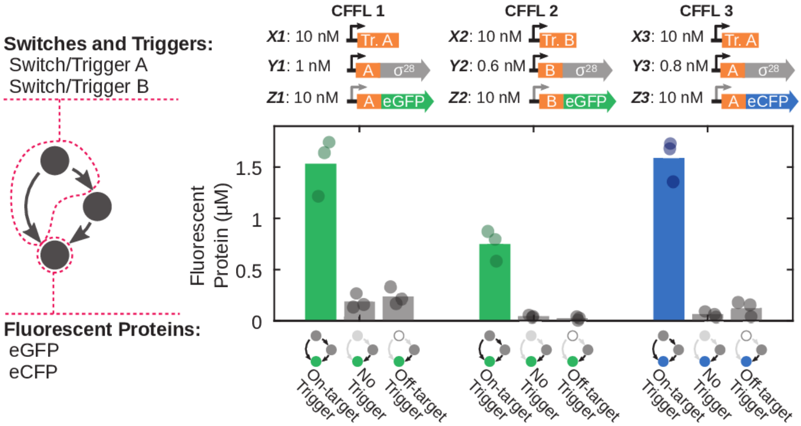


*Figure 5.2: End point fluorescent protein concentrations (after 14 h incubation; see Methods) of CFFL (CFFL 1), a variant with a distinct switch (CFFL 2), and a CFFL with eCFP as output protein (CFFL 3). For CFFL 1 and 3, which comprise switch/trigger A, trigger B (DNA **X2**) constructs were used as off-target control and for CFFL 2 trigger A (DNA **X1**) was taken as the off-target trigger. Note that the input and intermediate constructs for CFFL 1 and 3 are equal. All experiments were performed in triplicate full overview of DNA concentrations can be found in Supplementary Table 5.1.*

5.3 Construction of a Composite CFFL

Using CFFL variant 2 and 3, we constructed a composite CFFL with two inputs and outputs and with the σ^{28} gene as a shared intermediate node (a multi-input, multi-output CFFL; Figure 5.3). To verify all interactions in the circuit functioned as expected, we evaluated the composite CFFL by expressing all DNA constructs of CFFL 2 and CFFL 3 (Figure 5.4a) in a single TXTL reaction whilst monitoring the eGFP and eCFP output fluorescence. The observed expression levels mostly equaled the individual CFFL circuits, ruling out large contributions of the synergistic production of σ^{28}. The higher eCFP expression level when DNA **X2** was omitted suggests a slight depletion of resources when the full circuit and both inputs are present.[16]

When we omitted either one of the input triggers, expression of the corresponding fluorescent protein dropped significantly, while omission of both inputs yielded background levels of all outputs, indicating that the output RNA constructs are correctly activated by their cognate triggers.

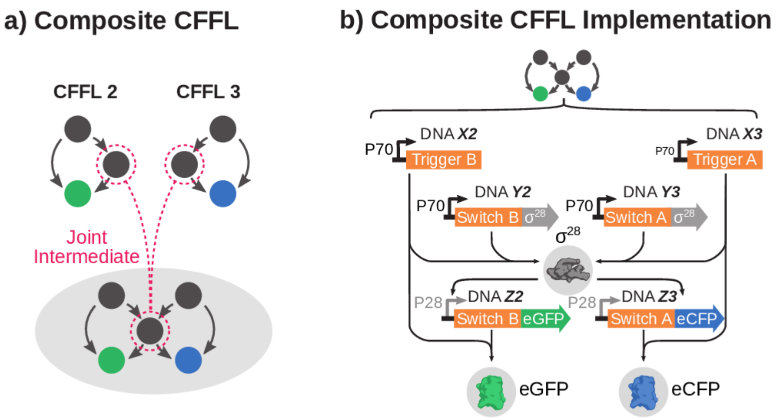


Figure 5.3: Construction of a composite CFFL using new synthetic CFFL variants. a) Schematic drawing of the formation of a composite CFFL that shares the intermediate σ^{28}-factor protein (middle), using two new CFFL variants (CFFL 2 and 3). b) Schematic representation of all DNA species and the DNA, RNA, and protein level interactions that form the composite CFFL with a shared intermediate σ^{28}-factor protein (middle), using CFFL 2 (left) and CFFL 3 (right).

The σ^{28} protein that serves as intermediate for both sides of the composite circuit was produced in excess, since removal of either of the σ^{28}-encoding DNA constructs did not lower the output levels of the circuit (Figure 5.4b). The slight increase in eCFP expression when DNA constructs **X2** and **Y2** are removed is consistent with the increase observed for omission of only **X2** in Figure 5.4a. Conversely, when the **X3** or **Y3** constructs are left out, eGFP expression remains constant compared to the full network, which implies there is a small imbalance between the two sides of the network. Nevertheless, the production of σ^{28} by one of the inputs can drive the production of RNA for the opposite output construct, as revealed by combinatorial evaluation of circuit components (Figure 5.5). In circuits where one input construct is used to activate σ^{28} production and the other input drives the activation of the output construct, expression levels of the activated output constructs are comparable to the levels of the full network. Furthermore, omission of a reporter construct (DNA **Z2** or **Z3**) completely eliminated the corresponding output and the absence of both intermediate constructs (DNA **Y2** and **Y3**) fully interrupted the progression of a signal through the circuit.

In summary, we demonstrate that composite feed-forward organizations can be readily implemented using our synthetic CFFL design and the composite CFFL with a shared intermediate node displayed selective activation of each output by its cognate input while simultaneously being coupled to the opposite input.

5.4 Dynamics of a Composite CFFL

In Chapter 4, we showed that CFFL 1 has a low background expression level compared to its maximum expression level, quantified by a high ON/OFF ratio. To investigate whether the same behavior persists in composite CFFL-based networks, we repeated the analysis on the new CFFL variants and the composite CFFL with a shared intermediate node.

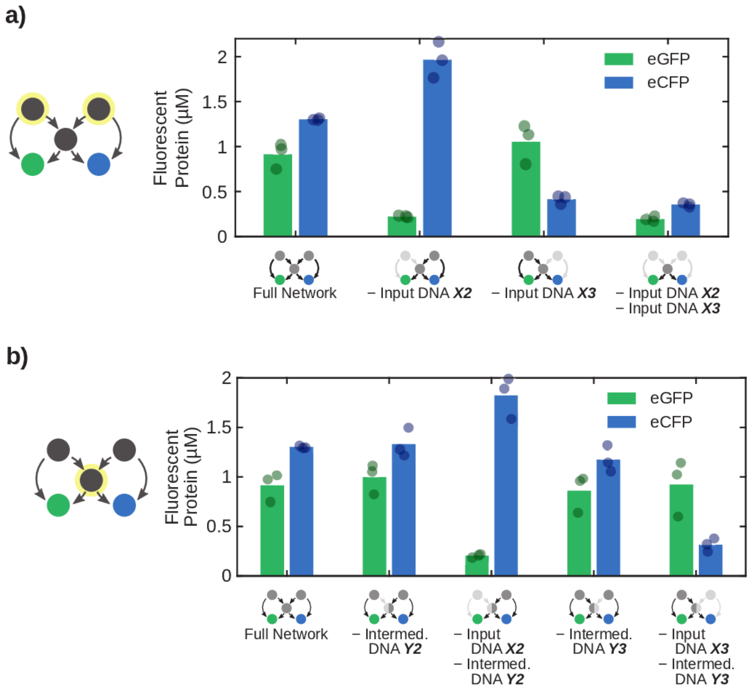


Figure 5.4: a) End point eGFP and eCFP concentrations of the composite CFFL with both input DNA constructs present, with either of the inputs or without input. (b) End point concentrations for the composite CFFL when one of the σ^{28}-producing DNA constructs is omitted. Concentrations of DNA species in panels were as shown in Figure 5.2. In case an off-target trigger was used, its DNA concentration was 10 nM, equal to the DNA concentration for on-target triggers. All experiments were performed in triplicate and a full overview of DNA concentrations can be found in Supplementary Table 5.1.

First, CFFL 2 and 3 were expressed in presence and absence of input DNA construct to determine the ON/OFF ratio and expression dynamics (Figure 5.6; solid lines). Expression dynamics and ON/OFF ratios were consistent with values observed for CFFL 1 in Chapter 4.

Since this network features two inputs and two outputs, the end point and transient characteristics were determined for each output in the presence and absence of the trigger that directly regulates the respective output (DNA **X2**, coding for trigger B, for the eGFP output and DNA **X3**, coding for trigger A, for the eCFP output; Figure 5.6). We observed that the ON/OFF ratio of both outputs of the composite circuit is greatly reduced compared to CFFL 2 and 3, since the overall background expression of σ^{28} increased due to the presence of two σ^{28}-producing constructs, which propagates into the background level of the output proteins. As a result, the background suppression behavior of the synthetic CFFL does not directly translate to the composite CFFL. Overall, our modular toehold switch-based CFFL system enabled the rapid analysis of a topologically complex synthetic CFFL circuit.

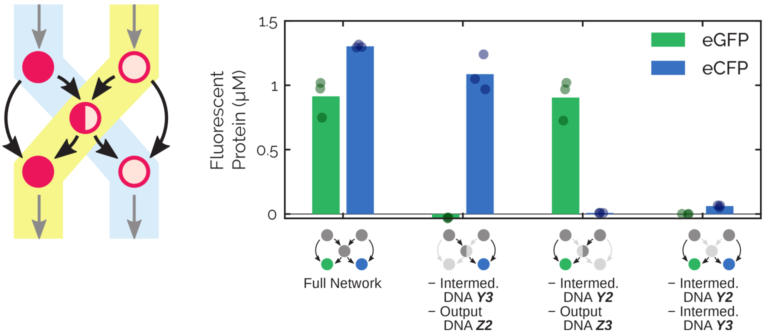


Figure 5.5: Endpoint eGFP (green) and eCFP (blue) concentrations of select subnetworks of the composite CFFL, where only the trigger and σ^{28} constructs from one side of the network and the trigger and output constructs from the other side were expressed. Concentrations of DNA species in panels are as shown in Figure 5.2. These experiments demonstrate that the two sides of the network can influence each other through the σ^{28} factor. The full composite CFFL and the composite CFFL where all σ^{28} constructs are left out are included as positive and negative control, respectively. Bars represent the average output protein endpoint concentration and the data points display the endpoint concentration of individual experiments. All experiments were performed in triplicate and a full overview of DNA concentrations can be found in Supplementary Table 5.1.

5.5 Conclusion

In this chapter, two alternative coherent feed-forward loop circuit implementations were constructed and characterized in batch TXTL reactions. Subsequently, a composite coherent feed-forward loop was implemented by combining the newly created CFFLs such that the two loops share the intermediate node. Using the flexibility of TXTL reaction, we verify all interaction by expressing subsets of the complex circuit. In Chapter 4, we found that our synthetic CFFL circuit displayed an increased ON/OFF ratio. When an equivalent analysis was applied to the composite CFFL, this behavior was not observed. Taken together, the synthetic CFFL circuit design presented in this research provides an excellent foundation to implement and systematically characterize higher-order organizations of CFFLs.

Since orthogonal alternative versions of the toehold switch[14,17] and E. coli sigma factor[13,18] are well characterized, we envision that all organizations of two CFFLs are suitable for implementation using our synthetic CFFL design, which enables characterization of topologically complex CFFL-based circuits that have as yet remained unexplored. Additionally, the modularity of the synthetic CFFL combined

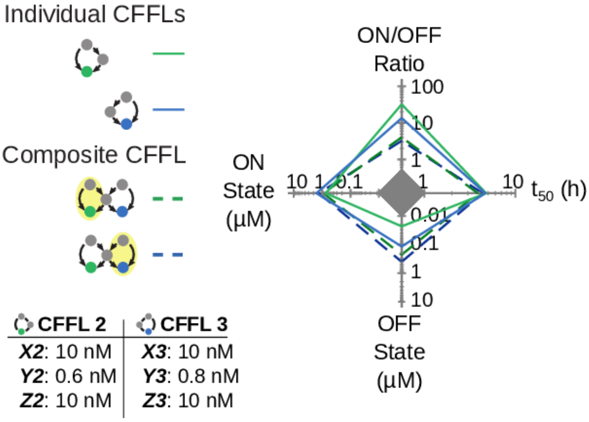


Figure 5.6: Trade-offs of the CFFL variants used in the composite CFFL (solid lines) and the trade-offs observed for both outputs of the circuit (dashed lines; see Methods). The dashed green line represents the characteristics of the eGFP output with or without the input on the same side of the network (yellow shaded side), whereas the dashed blue line shows the characteristics of the eCFP output for its corresponding input trigger.

with the use of post-transcriptional regulation facilitates integration with existing genetic networks that are based on transcriptional control.

5.6 Methods

5.6.1 Preparation of DNA Constructs

DNA constructs were created with golden gate assembly (GGA) using the overlapping sequences adapted from Sun et al.[19] (see Chapter 2). Promoters, UTR1, coding sequences and terminators were ordered from IDT as gBlock fragments or amplified from the pBEST vector using PCR. Toehold switch and trigger sequences were taken from previous studies in the group of Dr. P. Yin (Switch A is unpublished and Switch B is Switch 1 of the second generation in the research by Green et al.[14]; related toehold switch plasmids can be obtained from Addgene (https://www.addgene.org/Alexander_Green/) and PCR amplified. PCR products were gel purified using the QIAquick Gel Extraction Kit (Qiagen) and added in equimolar amounts to GGA assembly reactions with BSAI-HF (NEB), T4 ligase (Promega) and the T4 ligase buffer. GGA reactions were performed in a thermocycler according to a standard GGA protocol.[20] The GGA products were transformed into NovaBlue cells (Merck), from which the plasmids were purified using the QIAprep Spin Miniprep Kit (Qiagen) and the DNA sequences were confirmed using Sanger sequencing.

Linear DNA templates for expression in TXTL reactions were created by PCR using Phusion High-Fidelity DNA Polymerase (NEB) with primers pBEST_LinL_F (GGCGAATCCTCTGACCAGC) and pBEST_LinL_R (CCAAGCTGGACTG-TATGCACG) and subsequent purification using QIAquick PCR Purification Kit (Qiagen).

5.6.2 Preparation of Cell Lysate

The E. coli cell lysate was prepared from the RNase E deficient BL21 STAR (DE3) (ThermoFisher Scientific) cells that were transformed with the pRARE vector from BL21 Rosettta (Merck). The lysate was prepared according to previously published protocols,[13,21] with the adaptations described in Chapter 2. The E. coli strain was grown in 2xYT medium supplemented with 40 mM potassium phosphate dibasic and 22 mM potassium phosphate monobasic until an OD600 of 1.7

was reached. The cultures were spun down and washed thoroughly with S30A buffer (14 mM Magnesium L-glutamate, 60 mM Potassium L-glutamate, 50 mM Tris, titrated to pH 8.2 using glacial acetic acid), before being resuspended in 0.9 mL S30A buffer per gram of dry pellet. The cell suspension was lysed using a French press at 16000 lb pressure in two passes and spun down. The supernatant was incubated at 37 °C for 1.5 hours and spun down. The supernatant was dialyzed into S30B buffer (14 mM Magnesium L-glutamate, 150 mM Potassium L-glutamate, titrated to pH 8.2 using 2 M Tris) in two steps for 3 hours total and spun down again. The supernatant was aliquotted, snap-frozen in liquid nitrogen and stored at -80 °C.

The energy mixture was prepared according to the protocol previously described by Sun et al.[21] and a constant distribution amino acid solution was prepared.[22]

5.6.3 Batch TXTL Reactions

The cell lysate (33% of total reaction volume) was combined with the energy mixture, amino acid solution (37.5 mM), Magnesium L-glutamate (8 mM), PEG-8000 (2%), GamS protein (3 µM; prepared as described by Sun et al.[19]) and MilliQ to form the 1.54x TXTL reaction mixture (65% of total reaction volume). The remaining volume (35%) of the reactions was used to add the linear DNA constructs of the gene networks and supplemented to the final volume with MilliQ.

Batch TXTL reactions were prepared in total volumes of 10 µL and transferred to 384-wells Nunc plates. The reactions were incubated at 29 °C and eGFP and optionally eCFP fluorescence was measured on a Saffire II (Tecan), Spark 10M (Tecan) or Synergy H1M (Biotek) plate reader for at least 14 hours. The concentration measured after 14 h of incubation was taken as endpoint concentration for all experiments. The plate readers were calibrated using a titration range of purified eGFP protein. For the composite CFFL, where there are two outputs, eGFP and eCFP concentration ranges were measured in both the eGFP and eCFP measurement channels to determine the crosstalk between the two measurements. Concentrations of all DNA constructs used in TXTL experiments are given in Supplementary Table 5.1.

5.6.4 Circuit Trade-Offs

The repression of background expression by the CFFL-based circuits was quantified using the endpoint expression levels after 12h with and without input trigger (ON and OFF state). Additionally, the ratio between these two measures, the ON/OFF ratio, was computed as a measure of the relative change in output upon circuit activation. A characteristic time scale of the response of the circuit was determined using the t_{50}, the time until half of the maximum output was reached as a measure of the characteristic circuit time-scale. Each characteristic measurement was plotted on an axis of a spider plot, with lines connecting the values corresponding to a single experimental condition (see also Figure 4.3). For CFFL 2 and CFFL 3, the characteristics were determined based on eGFP and eCFP output, respectively. For the composite CFFL, the characteristic measurements were determined for both eGFP and eCFP, based on the ON and OFF state of their corresponding input. The concentration of DNA species for all experimental conditions can be found in Supplementary Table 5.1.

5.7 Supplementary Tables and Figures

Figure	Samples	DNA	DNA Name	Concentration (nM)
5.2	CFFL 1: All	Y1/Y3	P70a-SwitchA-S28	1
	CFFL 2: All	Y2	P70a-SwitchB-S28	0.6
	CFFL 3: All	Y1/Y3	P70a-SwitchA-S28	0.8
	CFFL 1: All	Z1	P28a-SwitchA-eGFP	10
	CFFL 2: All	Z2	P28a-SwitchB-eGFP	10
	CFFL 3: All	Z3	P28a-SwitchA-eCFP	10
	CFFL 1 / 3: On-Target	X1/X3	P70a-TriggerA	10
	CFFL 1: Off-Target		P70a-TriggerC	10
	CFFL 3: Off-Target	X2	P70a-TriggerB	10
	CFFL 2: On-Target	X2	P70a-TriggerB	10
	CFFL 2: Off-Target	X1/X3	P70a-TriggerA	10
5.4a	All	Y2	P70a-SwitchB-S28	0.6
	All	Y1/Y3	P70a-SwitchA-S28	0.8
	All	Z2	P28a-SwitchB-eGFP	10
	All	Z3	P28a-SwitchA-eCFP	10
	Full / -X2	X1/X3	P70a-TriggerA	10
	Full / -X3	X2	P70a-TriggerB	10
5.4b	All	Z2	P28a-SwitchB-eGFP	10
	All	Z3	P28a-SwitchA-eCFP	10
	Full / -Y2 / -X2 -Y2	Y1/Y3	P70a-SwitchA-S28	0.8
	Full / -Y3 / -X3 - Y3	Y2	P70a-SwitchB-S28	0.6
	Full / -Y2	X1/X3	P70a-TriggerA	10
	Full / -Y3	X2	P70a-TriggerB	10
5.5	All	X1/X3	P70a-TriggerA	10
	All	X2	P70a-TriggerB	10
	Full / -Y3 -Z2 / -Y2 -Y3	Z3	P28a-SwitchA-eCFP	10
	Full / -Y2 -Z3 / -Y2 -Y3	Z2	P28a-SwitchB-eGFP	10
	Full / -Y3 -Z2	Y2	P70a-SwitchB-S28	0.6
	Full / -Y2 -Z3	Y1/Y3	P70a-SwitchA-S28	0.8
5.6	CFFL 2 (solid green): All / Composite: All	Y2	P70a-SwitchB-S28	0.6
	CFFL 3 (solid blue): All / Composite: All	Y1/Y3	P70a-SwitchA-S28	0.8
	CFFL 2 (solid green): All / Composite: All	Z2	P28a-SwitchB-eGFP	10

CFFL 3 (solid blue): All / Composite: All	Z3	P28a-SwitchA-eCFP	10
CFFL 2 (solid green): *ON* State / Composite left (dashed green): *ON* State	X2	P70a-TriggerB	10
CFFL 3 (solid blue): *ON* State / Composite right (dashed blue): *ON* State	X1/X3	P70a-TriggerA	10

Supplementary Table 5.1: Concentrations of DNA construcs used in TXTL experiments in this chapter, ordered by figure.

5.8 References

(1) Karzbrun, E., Tayar, A. M., Noireaux, V., and Bar-Ziv, R. H. (2014) Programmable on-chip DNA compartments as artificial cells. *Science 345,* 829–832.

(2) Yelleswarapu, M., van der Linden, A. J., van Sluijs, B., Pieters, P. A., Dubuc, E., de Greef, T. F. A., and Huck, W. T. S. (2018) Sigma Factor-Mediated Tuning of Bacterial Cell-Free Synthetic Genetic Oscillators. *ACS Synth. Biol. 7,* 2879–2887.

(3) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *PNAS 110,* 15985–15990.

(4) Niederholtmeyer, H., Sun, Z. Z., Hori, Y., Yeung, E., Verpoorte, A., Murray, R. M., and Maerkl, S. J. (2015) Rapid cell-free forward engineering of novel genetic ring oscillators. *eLife* (Simmel, F., Ed.) *4,* e09771.

(5) Borkowski, O., Ceroni, F., Stan, G.-B., and Ellis, T. (2016) Overloaded and stressed: whole-cell considerations for bacterial synthetic biology. *Current Opinion in Microbiology 33,* 123–130.

(6) Del Vecchio, D., Ninfa, A. J., and Sontag, E. D. (2008) Modular cell biology: retroactivity and insulation. *Mol Syst Biol 4,* 161.

(7) Meijer, L. H. H., Joesaar, A., Steur, E., Engelen, W., van Santen, R. A., Merkx, M., and de Greef, T. F. A. (2017) Hierarchical control of enzymatic actuators using DNA-based switchable memories. *Nature Communications 8,* 1117.

(8) Burda, Z., Krzywicki, A., Martin, O. C., and Zagorski, M. (2011) Motifs emerge from function in model gene regulatory networks. *Proceedings of the National Academy of Sciences 108,* 17263–17268.

(9) Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network Motifs: Simple Building Blocks of Complex Networks *298,* 5.

(10) Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002) Network motifs in the transcriptional regulation network of Escherichia coli. *Nat Genet 31,* 64–68.

(11) Gorochowski, T. E., Grierson, C. S., and di Bernardo, M. (2018) Organization of feed-forward loop motifs reveals architectural principles in natural and engineered networks. *Sci. Adv. 4,* eaap9751.

(12) Kashtan, N., Itzkovitz, S., Milo, R., and Alon, U. (2004) Topological generalizations of network motifs. *Phys. Rev. E 70,* 031909.

(13) Garamella, J., Marshall, R., Rustad, M., and Noireaux, V. (2016) The All E. coli TX-TL Toolbox 2.0: A Platform for Cell-Free Synthetic Biology. *ACS Synth. Biol. 5,* 344–355.

(14) Green, A. A., Silver, P. A., Collins, J. J., and Yin, P. (2014) Toehold Switches: De-Novo-Designed Regulators of Gene Expression. *Cell 159,* 925–939.

(15) Shin, J., and Noireaux, V. (2012) An E. coli Cell-Free Expression Toolbox: Application to Synthetic Gene Circuits and Artificial Cells. *ACS Synth. Biol. 1,* 29–41.

(16) Stögbauer, T., Windhager, L., Zimmer, R., and Rädler, J. O. (2012) Experiment and mathematical modeling of gene expression dynamics in a cell-free system. *Integr. Biol. 4,* 494.

(17) Green, A. A., Kim, J., Ma, D., Silver, P. A., Collins, J. J., and Yin, P. (2017) Complex cellular logic computation using ribocomputing devices. *Nature 548,* 117–121.

(18) Bervoets, I., Van Brempt, M., Van Nerom, K., Van Hove, B., Maertens, J., De Mey, M., and Charlier, D. (2018) A sigma factor toolbox for orthogonal gene expression in Escherichia coli. *Nucleic Acids Research 46*, 2133–2144.

(19) Sun, Z. Z., Yeung, E., Hayes, C. A., Noireaux, V., and Murray, R. M. (2014) Linear DNA for Rapid Prototyping of Synthetic Biological Circuits in an Escherichia coli Based TX-TL Cell-Free System. *ACS Synth. Biol. 3*, 387–397.

(20) Engler, C., Kandzia, R., and Marillonnet, S. (2008) A One Pot, One Step, Precision Cloning Method with High Throughput Capability. *PLoS ONE* (El-Shemy, H. A., Ed.) *3*, e3647.

(21) Sun, Z. Z., Hayes, C. A., Shin, J., Caschera, F., Murray, R. M., and Noireaux, V. (2013) Protocols for Implementing an Escherichia coli Based TX-TL Cell-Free Expression System for Synthetic Biology. *JoVE (Journal of Visualized Experiments)* e50762.

(22) Caschera, F., and Noireaux, V. (2015) Preparation of amino acid mixtures for cell-free expression systems. *BioTechniques 58*.

6

Light-Activated Signaling in DNA-encoded Sender-Receiver Architectures

Part of this work has been published as:

Shuo Yang, Pascal A. Pieters, Alex Joesaar, Bas W.A. Bögels, Rens Brouwers, Iuliia Myrgorodska, Stephen Mann, and Tom F.A. de Greef. *ACS Nano* **14**, 15992-16002 (2020)

6.1 Introduction

In chapter 5, we have explored the construction of higher-order architectures of a simple network motif to achieve more complex functionalities. Alternatively, cellular systems commonly utilize collective behavior that emerges from the combination of intracellular pathways of cells through cell-cell communication.[1] The resulting emergent behavior of a population is more complex than can be achieved by individual cells and is dependent on the architecture of the cell-cell communication network.[2]

Many biological systems collectively control essential cell functions by sensing signaling molecules that are secreted by itself (autocrine signaling) and neighboring cells (e.g. quorum sensing).[3] In sender-receiver topologies sender cells secrete a signaling molecule that subsequently forms a concentration gradient across a population of receiver cells, which sense and process the signaling molecule. Architectures of sender and receiver cells are abundantly present in biological systems[4] and orchestrate, among others, cell growth,[5] death,[2] and differentiation.[6]

Synthetic implementations of sender-receiver architectures have been demonstrated to capture complex multicellular functionalities, such as morphogen reconstitution[7,8] and pattern formation.[9,10] Nevertheless, the complexity of cells prohibits a quantitative, bottom-up approach to characterizing the spatiotemporal properties of sender-receiver architectures. The development of minimalistic model systems that mimic natural cells (protocells) provides a starting point to obtain such a quantitative understanding of cell-cell communication and its emergent behavior.[11] Encapsulated TXTL systems have been studied in the context of cell signaling,[12,13] but the limited number of signaling molecules combined with low tunability of other parameters has hampered thorough spatiotemporal analysis of TXTL-based communication architectures.

Previously, we have developed a protocell platform that is based on DNA strand displacement reactions (DSD) for communication and signal processing.[14] This BIO-PC (Biomolecular Implementation of Protocellular Communication) platform is minimalistic and programmable, because of its reliance on highly predictable strand displacement reactions.[15] In this platform, compartmentalization is achieved using semi-permeable cross-linked polymer-protein conjugates (proteinosomes)[16] that encapsulate streptavidin-linked DNA gate complexes, which act as signaling

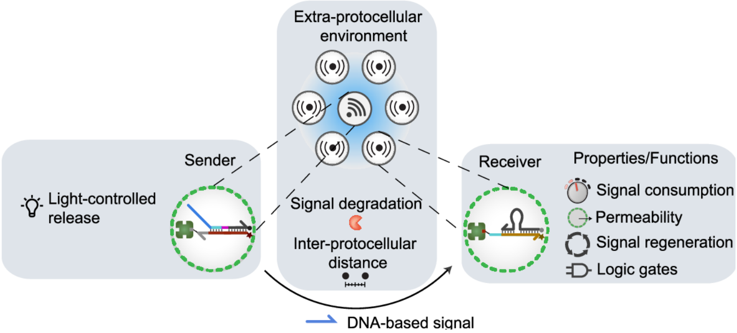


Figure 6.1: Light-activated DNA-encoded sender-receiver spatial system. Schematic of a single sender protocell and multiple receiver protocells are localized on a 2D spatial grid using a PDMS-based trapping array. Light-activated release of a ssDNA signal from the sender protocell sets up a signaling gradient which activates nearby receiver protocells. By controlling the characteristics of the protocells and environmental factors, this architecture enables quantitative analysis of diffusive signal propagation in space and time and programmable properties.

machinery, and enable exchange of short DNA signaling molecules. The flexibility and tunability of the proteinosomes, DNA gates, and DNA signals make this system an excellent candidate to mimick and characterize cell-cell communication *in vitro*. However, the BIO-PC platform does not accommodate spatiotemporal control over signal release.

Therefore, in this chapter, I collaborated with Shuo Yang, who produced the DNA-loaded proteinsomes and performed all measurements. Subsequently, I developed all image processing and data analysis methods (see Methods and Supplementary Code). We adapted and expanded the BIO-PC platform to achieve selective activation of sender protocells using light and use the updated system to study sender-receiver architectures (Figure 6.1). To systematically analyze fixed configurations of sender and receiver protocells, we used a microfluidic trapping array to trap a single sender protocell and surround it with a multitude of receiver protocells. The microfluidic setup was designed and constructed using a setup similar to the flow reactors described in Chapter 2. Activation of the sender using light irradiation resulted in a signaling gradient that could be processed by the receivers, using a diffusion-consumption mechanism.[17] We quantify this gradient by determining a characteristic signaling length-scale.

This highly controlled setup enabled us to vary the receiver density and permeability, in addition to signal consumption and extra-protocellular signal degradation rates. These system parameters were demonstrated to be key determinants of the signaling length-scale of our signaling model. Moreover, we increased the topological complexity of the protocellular system by introducing two distinct senders. The integration of these two signals in receiver cells followed Boolean AND-gate logic, which resulted in complex spatiotemporal control over receiver activation.

6.2 Construction of a light-activated spatial DNA-encoded sender-receiver system

We adapted the BIO-PC platform to function as a sender-receiver architecture by employing two types streptavidin-containing proteinosome-based semipemeable protocells which were loaded with biotinylated DNA gate complexes capable of sending or receiving short single-stranded DNA strands respectively (Figure 6.2a). Our setup is based on preparation of multimodal protocell populations consisting of a single sender and multiple receiver protocells (average protocellular diameter 33.84 µm ± 5.99 µm) using a microfluidic trapping array. Diffusive molecular communication from the sender to surrounding receiver protocells is initiated by applying laser irradiation to the sender protocell, resulting in cleavage of an internalized photo-cleavable nitrobenzyl linker, and concomitant release of DNA strand **A**. Strand **A** functions as the diffusible signal that is secreted from the sender protocell and migrates through the medium thereby activating surrounding receiver protocells resulting in a fluorescent response which can be probed with high spatiotemporal resolution. Specifically, the sender protocell contains an encapsulated DNA gate complex $\mathbf{F_1Q_1}$ consisting of a fluorophore (Cy5)-labeled gate strand $\mathbf{F_1}$ and strand $\mathbf{Q_1}$ functionalized with a quencher and photo-cleavable nitrobenzyl moiety (Figure 6.2a). Upon laser irradiation, strand $\mathbf{Q_1}$ is cleaved into two shorter ssDNA strands **A** and **B** which dissociate from the $\mathbf{F_1}$ strand at room temperature. We characterized photo-cleavage of the internalized $\mathbf{F_1Q_1}$ gate complex by localizing a single sender protocell in the trapping array followed by irradiation with laser light, resulting in an increase in Cy5 fluorescence of the sender protocell over time due to the cleavage of $\mathbf{Q_1}$ and dissociation of the quencher-labeled fragment B (Figure 6.2b). The photo-cleavage process is observed to follow first order kinetics (Figure 6.2b and Supplementary Figure 6.1). Furthermore, photocleavage of the nitrobenzyl linker inside the sender protocell is localized to the illuminated area and a specific wavelength (Supplementary Figure 6.2). Together, these results

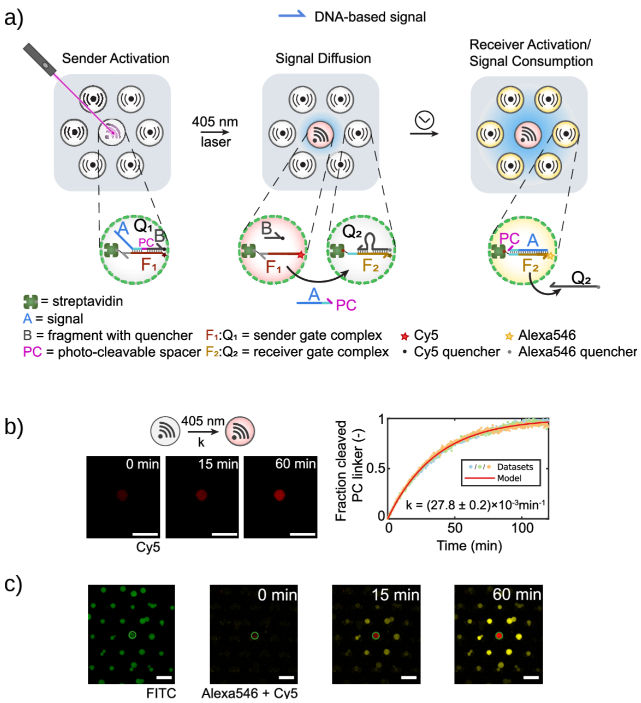


Figure 6.2: a) The sender protocell contains a fluorescently quenched internalized gate complex $\mathbf{F_1Q_1}$ *anchored to streptavidin using a biotinylated DNA gate strand (*$\mathbf{F_1}$*). Signal release from the sender protocell is triggered by laser irradiation resulting in photo-cleavage of strand* $\mathbf{Q_1}$*, concomitant dissociation of the two cleaved parts (**A** and **B**) and Cy5 fluorescence. Signaling strand A activates the surrounding receiver protocells by displacement of a quencher strand (*$\mathbf{Q_2}$*) from an internalized streptavidin-anchored gate complex* $\mathbf{F_2Q_2}$ *to produce an Alexa546 fluorescence output and consumption of strand **A**. b) Confocal micrographs of one sender protocell showing time-dependent increase in Cy5 fluorescence upon laser (405 nm) irradiation, indicating signal release. The plot shows the background-corrected fluorescence trace and exponential fit of the photo-cleavage reaction with a first-order rate constant of 0.0278 min*$^{-1}$*. Experiments were performed in independent triplicates. c) Confocal micrographs of one sender and multiple receivers (FITC-labelled proteinosome membrane, green) showing time-dependent increase in Cy5 fluorescence (red) and Alexa546 (yellow) fluorescence associated with signal release and activation, respectively. All scale bars are 100 µm.*

validate that the photo-cleavage of internalized DNA gate complexes inside a sender protocell can be achieved with a high spatial resolution.

Next, we assembled a multimodal sender-receiver population by sequential loading of a single sender and multiple (~150) receiver protocells. The receiver proto-

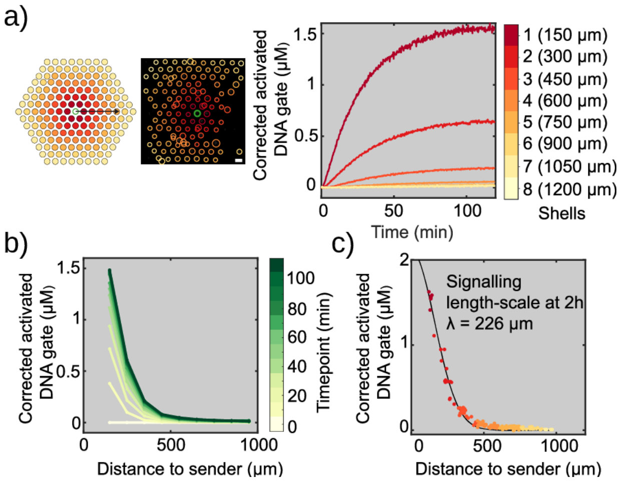


Figure 6.3: a) Binning method employed to analyze spatial receiver activation. Protocells are binned in concentric shells based on their distance from the sender (left images). Plots (right) show time-dependent changes in the concentration of activated DNA gates associated with receiver protocells in different concentric shells arranged around the central sender. Shell 1 (dark red) is closest to the sender. The upper limit of the distance from the receiver protocells to the sender protocell of each shell is listed next to the color bars. b) Concentration of activated DNA gate complexes in receiver protocells plotted for different times as a function of distance to the sender. The fluorescence intensity of each protocell at intervals of exactly 10 minutes was obtained through linear interpolation and the average of each distance bin (50 μm per bin) was plotted for these timepoints (Supplemental Methods). c) Plots of concentrations of activated DNA gates in individual receivers positioned at different distances from the sender protocell. Data collected after 2h of signal release. The color code corresponds to the different concentric shells as shown in a). Line represents fit of the data with Gaussian function. Sender protocells and receiver protocells were prepared using 10 μM and 4 μM streptavidin respectively. Experiments were performed at room temperature. Scale bar is 100 μm.

cells contain an encapsulated DNA gate complex $\mathbf{F_2Q_2}$ consisting of a fluorophore (Alexa 546)-labeled gate strand $\mathbf{F_2}$ with an exposed toehold domain and a quencher-labeled strand $\mathbf{Q_2}$ (Figure 6.2a) resulting in quenching of the Alexa fluorescence. Activation of the receiver protocells is initiated by toehold-mediated strand displacement of $\mathbf{Q_2}$ by signal strand **A** released from the sender protocell. Experiments were initiated by laser irradiation (405 nm laser, 2h) of the single sender protocell resulting in photo-cleavage of the $\mathbf{F_1Q_1}$ gate complex which could be monitored by an increase in Cy5 fluorescence (Figure 6.2c, red). We confirmed successful activation of the receiver protocells by signal strand **A** by monitoring an increase in Alexa546 fluorescence in individual receivers (Figure 6.2c, yellow). To analyze receiver activation dynamics under the signaling gradient we binned receivers in concentric shells based on their radial distance from the sender and plotted the average fluorescence traces (Figure 6.3a). Receivers in close proximity to the sender protocell are activated first and to a higher extent, confirming that the diffusible signal released from the central sender is consumed by the surrounding receivers, which therefore limits the activation of receivers at larger distances from the sender. To further quantify the spatiotemporal data, we plotted the spatial distribution of the receiver protocells' activation states at different timepoints (Figure 6.3b). The spatiotemporal data displayed an activation front stabilized after 100-120 min. This pseudo steady-state resulted from the diminished signals release from the sender protocell after 2 h of illumination (Figure 6.2b). We defined the characteristic signaling length scale λ (see Methods) as the distance at which the receiver activation has dropped off to 1/e (37%) of its maximum amplitude at pseudo steady-state. We determined λ from the image taken after 2 h of illumination and found a characteristic signaling length scale λ of approximately 226 µm (Figure 6.3c). This value is well in the range of many natural systems that communicate via soluble factors, i.e. morphogens[18] (40~200 µm), cytokines[17] (30~150 µm) and retinoic acid[19] (300~500 µm). These results establish that spatially controlled light-induced signal release from an individual sender protocell results in a distance-dependent activation of surrounding receivers in agreement with a diffusion-consumption mechanism.

6.3 Modulation of signaling range

Intercellular communication distances established by soluble factors in multicellular populations are regulated by both internal and external physicochemical factors, such as signal secretion, diffusion and consumption rates.[20,21] How each of these determinants modulates the signaling length scale has remained difficult to analyze due to the intrinsic complexity of natural sender-receiver systems. Here we employ our synthetic sender-receiver architecture and quantitatively analyze the contribution of individual determinants to the signaling length scale. Specifically, we constructed multimodal sender-receiver populations through sequential loading of a single sender and multiple (100-150) receiver protocells. Using this setup, we determined the internal and external determinants leading to changes in effective signaling length scale associated with variations in the capacity and rate

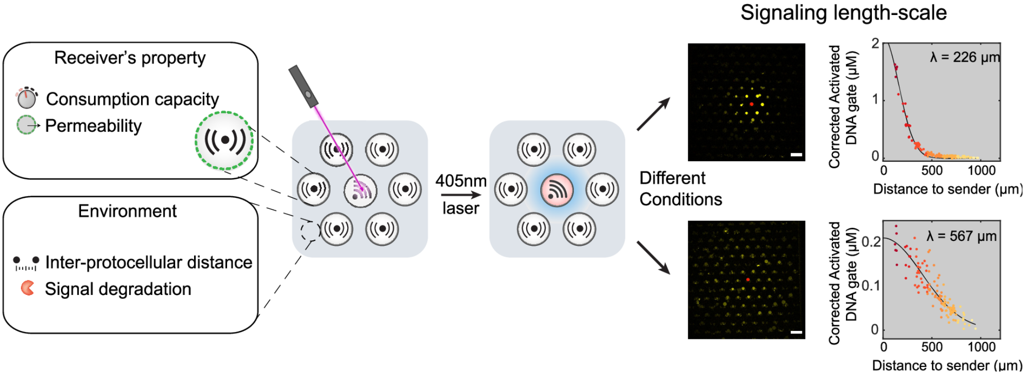


Figure 6.4: Tuning of the signaling length scale in light-activated DNA-encoded sender-receiver spatial systems. Changes in internal factors such as signal consumption capacity (receiver DNA gate complex concentration) and consumption rate (receiver membrane permeability) and external factors such as inter-protocellular distance (protocell trap density) and signal degradation (exonuclease concentration) influence the signaling length scale (left and center). Plots show typical experimental data used for the determination of the signaling length scale (right). Data collected after 2h of signal release from the sender. Scale bars 150 μm.

of signal consumption, and levels of signal degradation in the environment (Figure 6.4). To quantify the influence of the variations, we calculated the signaling length scale from images taken after 2 h illumination (see Methods).

In cellular populations, binding of soluble factors to receptors on neighboring cells results in consumption of the available signal and therefore influences the effective signaling range.[20–22] In the BIO-PC platform, the consumption capacity of individual receiver protocells can be varied by changing the concentration of encapsulated $\mathbf{F_2Q_2}$ DNA gate complex, which depletes the diffusible signal by strand-displacement. We performed the sender-receiver experiments using receiver protocells with three different DNA gate complex concentrations (Supplementary Figure 6.3) and calculated the corresponding signaling length-scales (Figure 6.5a). In general, for the 20-40 μm sized receiver protocells used in this study, the effective signaling length scale ranges between 200-500 μm. In agreement with our expectations, increasing the consumption capacity of individual receiver protocells results in lower effective communication distances due to a higher local depletion of the soluble signal. Besides consumption capacity, the consumption rate of a soluble signal can also modulate the signaling length scale. In multicellular populations, the consumption rate of a morphogen or cytokine can be regulated by controlling the rate of endocytosis.[20] Here, we modulate the consumption rate by tun-

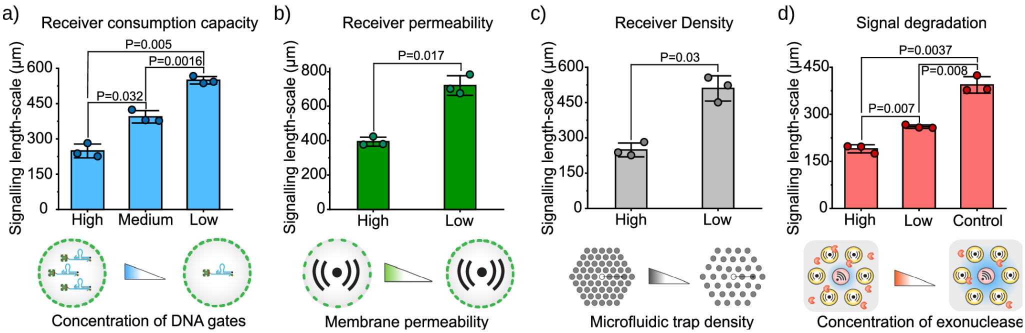


Figure 6.5: Modulation of signaling length scale at t = 2 h for changes in receiver consumption capacity (a), consumption rate (membrane permeability) (b), protocell trap density (c) and signal degradation (d). a) High, medium and low levels of the receiver protocell-entrapped DNA gate complex relate to changes in receiver-encapsulated streptavidin concentrations ([SA]) of 4, 1 and 0.4 µM, respectively. b) High (202.8 µm min^{-1}) and low (2.16 µm min^{-1}) receiver permeabilities relate to modifications in the protocell membrane crosslinking density; [SA] = 1 µM. c) High and low receiver number densities relate to the use of 90 or 70 microfluidic traps per mm2, respectively; [SA] = 4 µM. d) High and low levels of signal degradation arise from the presence of 0.1 unit/µL and 0.05 unit/µL of exonuclease I, respectively; [SA] = 1 µM. The control experiment is performed in the absence of exonuclease I. For all experiments, the concentration of encapsulated streptavidin in the sender protocells was 10 µM. All experiments were performed in independent triplicates at room temperature. All the experimental conditions are summarized in Supplementary Table 6.2. Data are presented as means±SD. A P-value less than 0.05 is considered to be statistically significant.

ing the membrane permeability of receiver protocells. We have previously shown that the permeability (P) of proteinosomes can be tuned using protein-crosslinking reagents of different length and revealed how the permeability influences the compartmentalized DSD reaction kinetics.[14] We prepared high-P and low-P receiver protocells (see Methods), quantified their permeability and confirmed they have approximately similar binding capacity for biotin-labelled DNA (Supplementary Figure 6.4). As expected, the experimentally derived signaling length scale is higher for low-P receiver protocells (Figure 6.5b) as the soluble signal is consumed at a lower rate.

Because an individual sender is surrounded by multiple receivers, the effective signaling distance not only depends on the consumption rate and consumption capacity of individual receivers but also on the cumulative signal consumption which can be varied by modulating the protocell number density in the spatial array.[17] We fabricated microfluidic trapping arrays with two different densities of protocell traps and determined the effective signaling range from the experimental data (Supplementary Figure 6.5). Our data shows that the signaling length scale increases when protocell density is decreased (Figure 6.5c). This increasing communication distance can be explained by lower total signal consumption capacity as the number density of receiver protocell decreases.

Biochemical degradation of diffusible factors is a key regulatory mechanism in morphogenesis and can control both the signaling range and sharpness of the diffusion front.[21,23,24] To mimic signal degradation, we added exonuclease I (3’ to 5’) to the trapped proteinosomes before initiating the photo-cleavage reaction. Exonuclease I selectively degrades the diffusible signal from its free 3’ end. However, the encapsulated DNA gate complexes in the sender and receivers lack a free 3’ end preventing their degradation. Laser-irradiation of the sender protocell cleaves the internalized DNA gate complex and yields a diffusible signal strand with a free 3’ end that is amenable for exonuclease degradation. Because the 3’ end of the diffusible signal contains the toehold binding domain required for strand-displacement with the receiver gate complex, enzymatic degradation strongly inhibits receiver activation. We performed sender-receiver experiments for two different concentrations of exonuclease I added to the medium (Figure 6.5d) and calculated the signaling length scale from the experimental data. As expected, higher concentrations of exonuclease result in decreasing signaling length scales. Taken together, these findings reveal how a fully synthetic sender-receiver protocell platform can be used to systematically study the effect of isolated physicochemical factors on the diffusive communication range.

6.4 Spatial integration of diffusible signals by Boolean receivers

Spatial integration of chemical signals by Boolean operations is essential to generate collective behaviour in multicellular populations as exemplified by the immune and nervous systems.[25,26] Although Boolean reaction-diffusion systems have been implemented using the Belousov-Zhabotinsky reaction,[27–31] a versatile and tunable platform based on biomolecular reactions is currently lacking. We previously showed the possibilities of implementing Boolean AND logic using BIO-PC, which relied on the sequential hybridization of two different DNA signals in

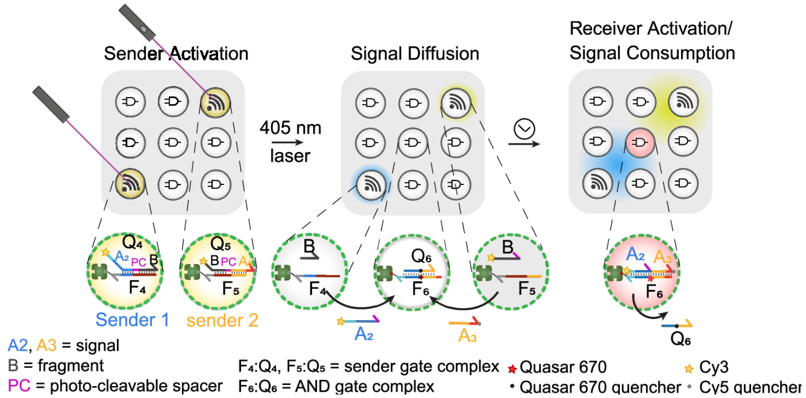


Figure 6.6: Spatial integration of non-identical signals by 2D-arrayed AND-gate protocells. Two fluorescent sender protocells (1 and 2) containing internalized gate complexes $\boldsymbol{F_4Q_4}$ *or* $\boldsymbol{F_5Q_5}$*, respectively, are embedded in a high number density of non-fluorescent AND-gate receivers. Signal release from sender protocells is triggered by laser irradiation resulting in photocleavage of* $\boldsymbol{Q_4}$ *and* $\boldsymbol{Q_5}$*, concomitant dissociation of the cleaved parts,* $\boldsymbol{A_2}$*+*$\boldsymbol{B}$ *and* $\boldsymbol{A_3}$*+*$\boldsymbol{B}$*, and loss of Cy3 fluorescence. The Cy3-labelled dissociated strand A2 and non-fluorescent strand* $\boldsymbol{A_3}$ *activate Quasar670-quenched receiver protocells containing an encapsulated AND gate (*$\boldsymbol{F_6Q_6}$*) through cooperative hybridization to produce a Cy3/Quasar670 fluorescence output.*

DSD circuitry.[14] However, this configuration does not allow distinct signal gradients to be integrated spatially, since the AND gate localized in the receiver protocells will sequester one of the signals in the absence of the other. Here, we reveal how the BIO-PC platform can be adapted to integrate non-identical gradients by localized AND operations based on a cooperative DSD mechanism. Using a sequential loading procedure, we implemented a three-population configuration consisting of two non-identical sender protocells embedded in a high density of receivers that implement Boolean AND logic (Figure 6.6). The two senders contain gate complexes $\mathbf{F_4Q_4}$ and $\mathbf{F_5Q_5}$, respectively, which upon simultaneous laser illumination (405 nm, 1.8 h), secrete two distinct Cy3-labeled signal strands $\mathbf{A_2}$ and $\mathbf{A_3}$ as monitored by the decrease in Cy3 fluorescence of the sender protocells (Figure 6.8, yellow). Receiver protocells contain an encapsulated DNA gate complex which is activated by a cooperative hybridization mechanism[32] where both $\mathbf{A_2}$ and $\mathbf{A_3}$ need to be simultaneously present to release quencher labelled strand $\mathbf{Q_6}$ resulting in an increase in Quasar 670 fluorescence (Figure 6.8, red). We analyzed spatiotemporal AND-gate receiver activation by binning receiver protocells into shells based on the maximum of the two distances to the senders and calculated

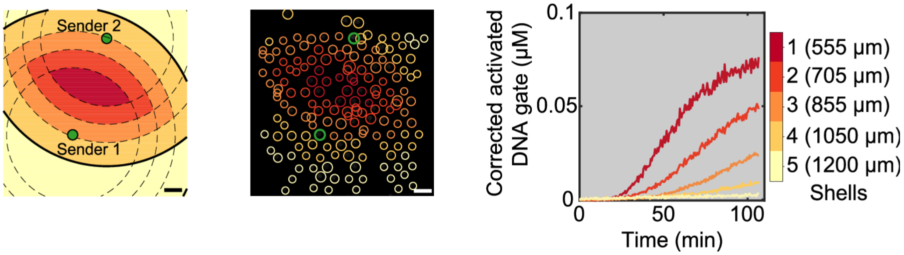


Figure 6.7: Binning method used to analyze spatiotemporal activation of receiver protocells. Protocells are binned based on the maximum of the two distances to the senders, which yields bins with outer bounds that are the intersection of the equivalent bounds of single sender systems, as illustrated by the black lines (left). Corresponding time traces of AND gate receiver activation within different bins. Shell 1 (dark red) is closest to the two senders. The maximum of the two distances from the receiver protocells to the sender protocell of each shell is listed next to the color bars. All the experimental conditions are summarized in Supplementary Table 6.2. Scale bars 150 μm.

the average fluorescence per bin over time (Figure 6.7). The experimental curves reveal that activation of the receivers is initiated at positions equidistant to the two senders, in agreement with AND-type logic. Furthermore, we plotted the response time of each receiver as a function of the distance to both senders and find the lowest response time for receivers at equidistant position of both senders (Figure 6.9a). Together, these results indicate that receiver protocells are activated by two non-identical signaling gradients of distributed spatial origins and demonstrate Boolean AND logic.

Because receiver protocells are activated by gradients from both senders, we wondered if we could spatially control initiation of receiver activation by sequential laser irradiation of the two senders. This would result in the development of a signal gradient from one of the two senders before the other gradient is established. Due to the reversible nature of the cooperative DSD mechanism,[32] signal strands from one sender that bind to the AND-gate are preferentially released in the absence of the other signal, preventing signal consumption by receivers. We first irradiated sender **1** for 18 minutes, followed by exposure of both senders for 1.5 h and calculated the response time for each receiver (Figure 6.9b). We observe a skewed activation pattern where receiver activation is initiated in receivers in close proximity to sender **2**, in agreement with the presence of spatial bias in the established signal gradients. Together, these results show that a population of pro-

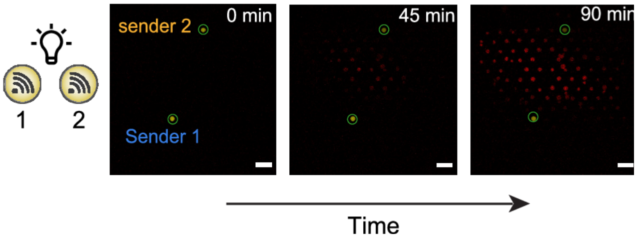


Figure 6.8: Confocal micrographs of two sender protocells (1 and 2) and multiple AND gate receivers recorded at t = 0 (left) showing Cy3 fluorescence in the spatially separated transmitters. Light-induced activation leads to a reduction in Cy3 fluorescence (yellow) in the senders, and progressive increase in Quasar 670 fluorescence (red) associated with activation of receiver protocells. All the experimental conditions are summarized in Supplementary Table 6.2. Scale bars 150 μm.

tocells could be programmed to integrate two non-identical cues and perform spatially-encoded Boolean operations using encapsulated DSD based reactions.

6.5 Conclusion

In this chapter, we demonstrate a fully synthetic soft matter system based on semipermeable microcompartments that communicate via short DNA strands under a light-induced local signaling gradient arising from a single sender protocell. We prepared multimodal protocell arrays consisting of a single sender protocell and a polydisperse receiver population using a microfluidic trapping device and systematically quantified how individual parameters control the signaling length scale typically between 100 and 700 μm. The simplicity of the system allows variation of the consumption capacity and consumption rate of receiver protocells. Building towards complex architectures of communicating protocells, we revealed how two local signaling gradients can be spatially integrated by employing receiver protocells containing Boolean AND gates. Demonstrating the high level of control over the sender-receiver protocells, we were able to manipulate the spatial output of the system through the timing of signal release from the individual senders.

As an artificial communication platform designed for simplicity and tunability, the adapted BIO-PC has its limitations. Living cells can translate extracellular signals into intracellular signals and perform subsequent processing of these signals through highly complex interaction networks of nodes, modules and pathways.[33] In contrast, there is no clear distinction between extracellular and intracellular sig-

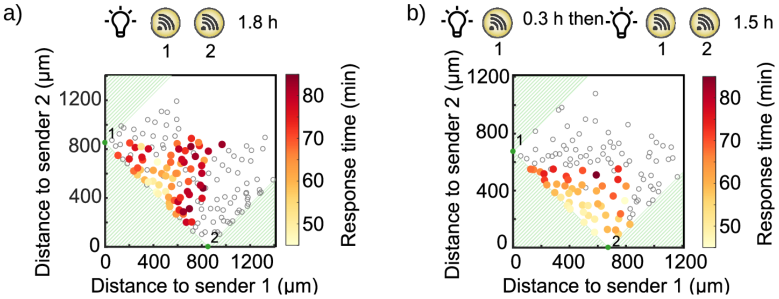


Figure 6.9: a) Response time of individual Boolean receivers upon simultaneous laser irradiation of two senders for 1.8 h plotted as a function of their distance to each of the two senders. The response time is defined as the time taken for an individual receiver to reach 50% of its final activated concentration. Two senders are marked as 1 and 2 in green. To remove background noise, any protocells with an absolute increase less than 1 RFU are excluded and labeled in hollow circles. b) Response time of individual Boolean receivers upon sequential laser irradiation showing spatial bias in activation of receiver protocells. Sender 1 is irradiated for 18 min, followed by irradiation of both senders for 1.5 h. Sender protocells and receiver protocells were prepared using 30 μM and 1 μM streptavidin respectively. All the experimental conditions are summarized in Supplementary Table 6.2.

nals in our system, which does not allow coordination of responses via combinatorial signaling. Additional research is needed to construct more complicated DNA-based networks for signal reception and processing.[34–37] Secondly, negative feedback control is often employed in multicellular organisms to regulate intercellular communication and guarantees precision, robustness and versatility.[38] Because our system is based on enzyme-free DNA circuity, it is difficult to realize negative feedback control. Negative feedback loops can be implemented in BIO-PC using enzyme-assisted DNA circuits, which could be utilized to construct protocell communities with much spatiotemporal behavior.[39] Moreover, in nature sender cells consume a fraction of the signal they produced via an autocrine signaling loop.[40] Artificial autocrine pathway could be created in our system by colocalization of signal and receiver gates inside the protocell or modification of receiver gates onto the protocellular membrane.

6.6 Acknowledgements

Shuo Yang is acknowledged for the experimental work the sender-receiver systems in this chapter. Alex Joesaar is acknowledged for his work on the BIO-PC system and microfluidic trapping devices.

6.7 Methods

6.7.1 Preparation of BS-NH2/PNIP Aam nanoconjugates

Cationized bovine serum albumin (BSA-NH_2) is synthesized via a previously reported procedure.[16] Typically, 1.5 g diaminohexane is dissolved in 10 mL MilliQ water, the resulting solution is adjusted to pH 6.5 using 5 M HCl and added dropwise to a solution of BSA (200 mg in 10 mL MilliQ water). The coupling reaction is initiated by addition of 100 mg of 1-(3-Dimethylaminopropyl)-3-ethylcarbodiimide HCl (EDC) immediately and another 50 mg after 5 h. The pH value of the solution is readjusted to pH 6.5 after EDC is added. The reaction mixture is stirred at 300 rpm using a hotplate stirrer and held at ambient temperature. The total reaction time is 11 h and then the solution is centrifuged to remove precipitate. The corresponding supernatant is dialyzed (Medicell dialysis tubing, MWCO 12-14 kDa) in MilliQ water overnight and freeze-dried.

Mercaptothiazoline-activated PNIPAAm (M_n = 9800 g mol^{-1}, 4 mg in 5 mL MilliQ water) is synthesized via a previously reported procedure[16] and added to a stirred solution of BSA-NH_2 (10 mg in 5 mL of PBS buffer, pH 8.0). The reaction mixture is stirred for 10h at ambient temperature. Then the solution is purified using a centrifugal filter (Millipore, Amicon Ultra, MWCO 50 kDa) and freeze-dried.

Fluorescent-labeled nanoconjugates BSA-NH_2/PNIPAAm are synthesized via the same method, except that Fluorescein isothiocyanate (FITC) labeled BSA is used as starting material.

6.7.2 Preparation of streptavidin-containing proteinosomes

Typically, BSA-NH_2/PNIPAAm nanoconjugates (final concentration 15 mg mL^{-1}), FITC-labeled BSA-NH2/PNIPAAm nanoconjugates (final concentration 1 mg mL^{-1}), streptavidin (final concentration: 0.4, 1, 4, 10 µM) and 1.5 mg PEG-bis(N-succinimidyl succinate) (Mw 2000) or 0.6 mg Bis(succinimidyl penta(ethylene glycol)) (BS(PEG)5, Mw 532.50) (to prepare proteinosomes with lower permeability) are mixed in 15 µL sodium carbonate buffer (50 mM, pH 9.0). The mixture is added to the oil 2-ethyl-1-hexanol (aqueous/oil volume fraction 0.06) and shaken by hand (15 s) to produce a Pickering emulsion. The upper oil layer is removed after 2 h of sedimentation. Then the emulsion is dispersed by addition of 400 µL mixture of ethanol/water (ethanol fraction 70%) and dialyzed sequentially in 70% ethanol (2 h), 50% ethanol (2 h) and water (overnight). The proteinosomes suspension is stored at 4°C.

6.7.3 DNA sequence design and synthesis

DNA sequences are designed via MATLAB script that generates random sequences with a G or C fraction between 0.3 and 0.7.

All gate complexes in protocells contain two strands, a gate strand (denoted as **F**) labeled with both a biotin group to enable binding to encapsulated streptavidin and a fluorophore to monitor signal release or receiver activation and an output strand (denoted as **Q**) functionalized with a quencher. Sender protocells used in the spatial Boolean setup consist of gate complexes in which strand **F** is only biotinylated without any other modification while the fluorophore is attached to output strand **Q**.

Sender protocells

The design of DNA gate complex localized inside sender protocells used in the sender-receiver setup is based on a biotinylated gate strand $\mathbf{F_1}$ labeled with fluorophore Cy5 to monitor the photo-cleavage reaction. Strand $\mathbf{Q_1}$ consist of three sequences ($\mathbf{A_1}$**-PC-**$\mathbf{B_1}$): strand $\mathbf{A_1}$ functions as signal for activation of receiver protocell, strand $\mathbf{B_1}$ is labeled with quencher and a photo-cleavable nitrobenzyl spacer (**PC**) linking strand $\mathbf{A_1}$ and $\mathbf{B_1}$. Before activation $\mathbf{F_1Q_1}$ form a stable duplex where Cy5 fluorescence is quenched. Upon UV irradiation $\mathbf{Q_1}$ is cleaved into two parts which both dissociate from template **F**. Strand $\mathbf{A_1}$ diffuses out from the sender protocells and activate receivers while dissociation of strand $\mathbf{B_1}$ leads to unquenching of the Cy5 fluorescence providing a direct measure for the photo-cleavage reaction. We verified if increase in Cy5 fluorescence directly correlates to signal release by labeling strand $\mathbf{A_1}$ labeled with Cy3 providing a direct measure for signal release (Supplementary Figure 6.1).

For the spatial Boolean setup a configuration of two non-identical sender protocells is used, where sender 1 is loaded with gate complex $\mathbf{F_4Q_4}$ and sender 2 is loaded with gate complex $\mathbf{F_5Q_5}$ (Figure 6.6). The corresponding biotinylated gate strands $\mathbf{F_4}$ and $\mathbf{F_5}$ are designed to bind Cy3-modified strands $\mathbf{Q_4}$ ($\mathbf{A_2}$**-PC-**$\mathbf{B_2}$) and $\mathbf{Q_5}$ ($\mathbf{A_3}$**-PC-**$\mathbf{B_3}$) respectively. Upon light activation, two non-identical signal strands $\mathbf{A_2}$ and $\mathbf{A_3}$ diffuse out and activate receiver protocells when the two strands are simultaneously present (AND logic).

The sequences of all gate complexes in sender protocells including their modification are summarized in Supplementary Table 6.1.

Receiver protocells

The design of DNA gate complex localized inside receiver protocells is based on the "seesaw" gate motif. Instead of using separate output and reporter complexes, the gate strands **F** are directly labelled with fluorophore Alexa546 at one end and a biotin group at the other end while the output strand **Q** is functionalized with a quencher. In the initial inactive state, **QF** forms a stable complex where the fluorophore is quenched. When signal strands enter receiver protocells and activate the gate complex by toehold-mediated DSD reaction, the fluorescence is turned on.

For the spatial Boolean setup, cooperative DSD is used to integrate two non-identical signals in receiver protocells (Figure 6.6). Biotinylated gate strand $\mathbf{F_6}$ is internally modified with a Quasar670 fluorophore whose fluorescence is quenched by binding to quencher-modified output strand $\mathbf{Q_6}$ in the initial inactive state. Upon light activation two non-identical strands $\mathbf{A_2}$ and $\mathbf{A_3}$ released form two distant senders are able to unquench the dye through cooperative DSD.

The sequences including chemical modification of all strands are listed in Supplementary Table 6.1. All sequences where screened using NUPACK[41] to detect any possible undesired interactions.

All DNA are synthesized by Integrated DNA Technologies (IDT) and BioSearch Technologies. The DNA strands are dissolved in TE buffer (10 mM Tris, 0.1 mM EDTA, pH 8.0, nuclease-free, IDT) as stock solutions (100 µM and 10 µM), which are stored at -20°C for later use.

6.7.4 Localization of DNA gate complexes in streptavidin-containing proteinosomes

The localization of DNA gate complexes in proteinosomes is performed in 10 mM Tris buffer with 12 mM Mg^{2+} and 0.1% v/v Tween 20. Typically, a dispersion of streptavidin-containing proteinosomes (10 µL), 4X buffer (5 µL) and biotinylated DNA gate strand **F** (2 µL from a 10 µM stock) are gently mixed and incubated at room temperature for 1 h, then 3 µL of the corresponding output strand **Q** (10 µM stock) is added to the mixture which is incubated at 4°C overnight. After incubation, the supernatant (10 µL) is removed carefully from the top and 400 µL buffer is added. The proteinosomes suspension is allowed to sediment for 5 h and 400 µL of supernatant is removed. This process is repeated and the resulting DNA gate-containing proteinosomes are stored at 4°C.

6.7.5 Design and fabrication of microfluidic trapping devices

A two-layer microfluidic chip is used to physically trap the proteinosomes. The chip is prepared using a previously reported procedure,[14] and contains a localization chamber with PDMS pillars, a filtering chamber, inlet channels with pneumatically actuated push-up valves and outlet channels. Master molds for the bottom and top layers are fabricated on separate silicon wafers (Silicon Materials) using photolithography techniques.[42] The bottom and top layer's mold is made by spin-coating SU8-3050 (50 µm in height) and spin-coating AZ 40xt (40 µm in height) respectively. After development the AZ 40xt mold is reflowed, resulting in rounded channels (60 µm in height) at the center. The two layers are assembled and then plasma bonded to circular #1.5 glass coverslips. Two chips with different density (270 and 210 in 1.5 mm X 2 mm) of traps are fabricated, with the use for density-dependent experiment.

6.7.6 Proteinosome trapping and activation

The microfluidic chip is installed on the stage of a confocal laser scanning microscope (Leica SP8). Water-filled control channels are actuated through pneumatic valve array (FESTO) which is connected to a programmable logic controlled (PLC, WAGO Kontakttechnik GmbH). The PLC is then linked to a PC and controlled by a custom Matlab GUI. The pressure in the control channels is 2 bar. The pressure of the inlet channels can be adjusted using pressure regulators (Flow-EZ, Fluigent). Typically, an experiment starts with removing air bubbles and filling all the chambers and channels with buffer solution by pressurizing the buffer channel. Then sender and receiver protocells are loaded sequentially. One DNA gate-localized sender protocell is loaded into the trapping chamber from inlet port 2 (pressure 10 mbar, port 2 is washed with buffer every time before loading proteinosomes). The sender protocell is first trapped at the first row of the chamber, then is pushed through several trapping rows by applying a pulse of flow with high pressure (200 mbar) and eventually locate at the chamber's center approximately. The trapping chamber is washed through flowing in buffer solution (10 mbar) for 5 min to remove any unbound DNA. Next, receiver protocells are loaded into the trapping array from inlet port 2 and a flow of buffer (20 mbar) is applied to remove any unbound DNA and proteinosomes that are not properly trapped, resulting in spatial distribution of protocells contains one sender protocell with surrounding receiver protocells (6-8 shells). In the spatial Boolean setup (Figure 6.6), sender 1 and 2 are loaded sequentially with a distance of 6 to 8 rows, followed by loading of receiver protocells. Solution of fuel strand or exonuclease (if used) with desired concentration is flowing into the trap array (10 mbar) via inlet port 3. Confocal microscope is focused on the trapping chamber, in which the sender protocell is illuminated with light (405 um laser, 100 ms every 1s) and time-lapse imaging (488, 552 and 638 nm lasers, 100 ms every 18s) is started. All the experiments are measured for 2 h and performed at room temperature.

To quantify the permeability of proteinosomes, a solution of Cy3-labeled strand (0.5 µM) is flowed into the chamber loaded with proteinosomes. The diffusion of fluorescent strand is measured over time inside and outside the proteinosomes and the permeability constant is estimated via a previously reported method.5

All of the experiments of sender-receiver system were performed in independent triplicates.

6.7.7 Data acquisition and analysis

Fluorescence data are obtained using confocal microscope with solid state lasers (488 nm for FITC, 552 nm for Cy3 and Alexa546, 638 nm for Cy5 and Quasar670). The objective used is 10X/0.40NA (1.55X1.55mm field of view, 7 µm slice thickness) with a resolution of 512 × 512 pixels (or alternatively 20X/0.75NA objective with a resolution of 1024 × 1024 pixels for permeability measurement). A hybrid detector (photon counting mode) is used. Typically, one sender and approximately 150 receiver protocells are recorded for each experimental condition. For spatial Boolean logic setup, two senders are imaged. The relative fluorescence units (RFU) is measured and is used to calculate the concentration of activated DNA gate. For all experiments the baseline valve is subtracted and labeled with "corrected" in the data analysis. The RFU-to-concentration conversion factor is determined by measuring the average RFU (value across a horizontal line through the device) of specific activated DNA gate complex (0.1 µM, 0.5 µM, 1 µM) that is flown into the device. The conversion factor is then determined by plotting the RFU vs the concentration of activated DNA gate complex.

Data processing

All image analysis was performed using MATLAB programming environment and its imaging toolbox (Mathworks). The image processing and data analysis scripts are listed in Section 6.9 (Supplementary Code).

Protocell detection: Microscopy images were imported, the FITC channel (label on the membrane of the protocells) was selected and the intensity was adjusted. Noise was removed using a median filter. Subsequently, circular objects (protocells) were detected using a two-stage circular Hough transform using a sensitivity of 0.9. Protocells intersecting with the edge of the image were ignored automatically and obvious misclassifications were discarded manually.

Protocell fluorescence quantification: Protocell fluorescence intensities for each fluorophore channel were determined as the average intensity of all pixels for each protocell, ignoring an outer band of pixels to minimize effects of slight discrepancies between the protocells and the Hough transform fitted circles. Fluorescence intensities were converted to fluorophore concentrations using a calibrated reference curve. All data was corrected by background subtraction. Sender and receiver protocells were automatically distinguished based on their fluorescence: Cy5 for the sender and Alexa546 for the receivers in sender-receiver and Cy3 for the senders and Qusaar670 for receivers in the spatial Boolean setup.

Quantitative analysis of sender-receiver system

To analyze receiver activation in space and time, receivers were binned into different concentric shells based on their distance toward the sender. The mean value of receiver fluorescence intensity per shell was used to represent time-dependent activation of the corresponding shell. Typically, 6~8 shells consisting of around 150 receiver protocells in total were tracked and used for data analysis.

To analyze the activation over time of the protocells with varying distance to the sender, the protocells were binned into bins of 50 µm. The fluorescence intensity of each protocell at intervals of exactly 10 minutes was obtained through linear interpolation and the average of each distance bin was plotted for these timepoints.

The characteristic length-scale is defined as the distance from the sender protocell at which receiver activation has dropped off to 1/e (37%) of its maximum value. In this research, the length scale is always determined after 2h of illumination and incubation. To estimate the characteristic length scale λ, the diffusion front was approximated by the fundamental solution to Fick's second law:[43]

$$C(r)=A\,\mathrm{e}^{-(\frac{r}{\lambda})} \qquad (1)$$

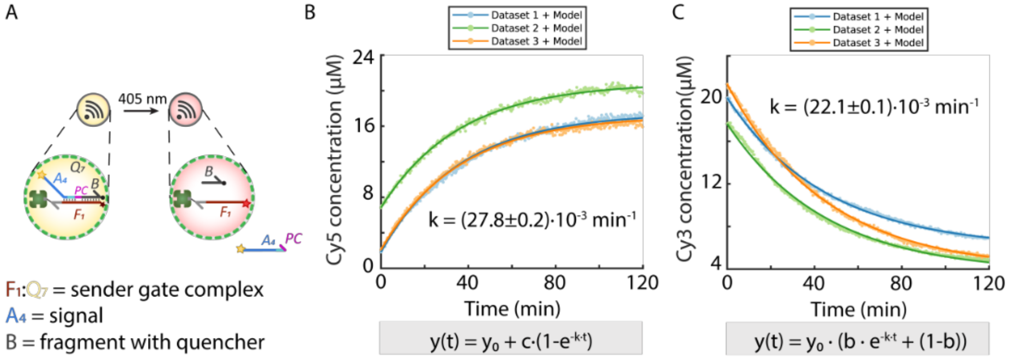
Where C (µM) is the concentration of activated gate complex, r (µm) is the distance of individual receiver to the sender, A (µM) is the maximum activation and λ is the length scale (µm). Filling in the definition of λ shows that it is indeed the distance at which the gradient concentration has dropped to 1/e of the maximum amplitude A:

$$C(\lambda)=A\,\mathrm{e}^{-(\frac{\lambda}{\lambda})}=\frac{A}{e} \qquad (2)$$

Quantitative analysis of Boolean receivers

To analyze activating dynamics of the spatial Boolean setup where two senders are embedded in a high density of AND receivers, receivers were binned into shells based on their maximum distance to one of the senders. The method of quantifying the average activation of protocells per shell remains the same. In addition, response time of each activated AND gate receiver was defined as the timepoint at which the fluorescence reaches 50% of its final value. To remove background noise a cutoff of minimum increase of 1 RFU was set.

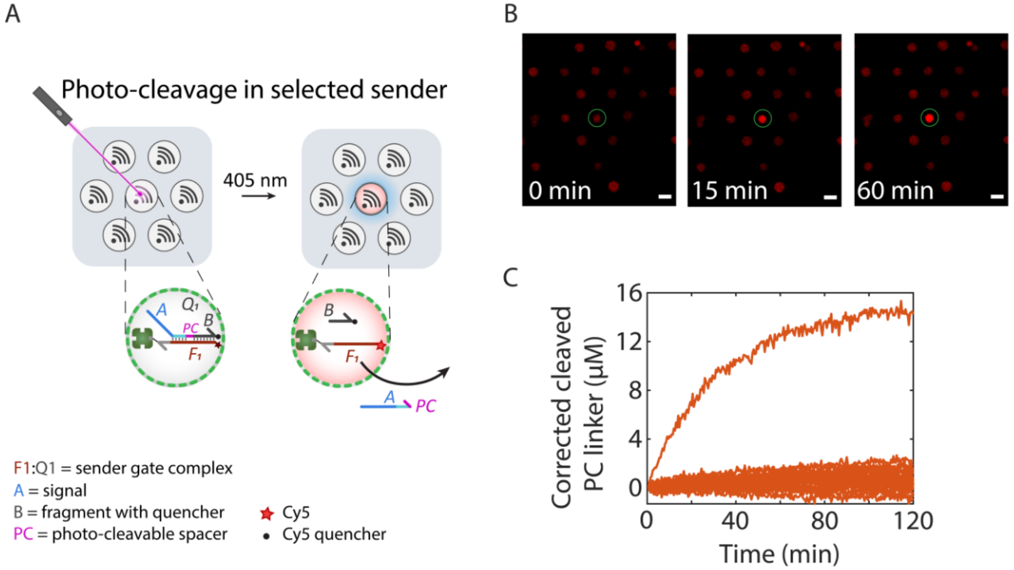
6.8 Supplementary Figures and Tables



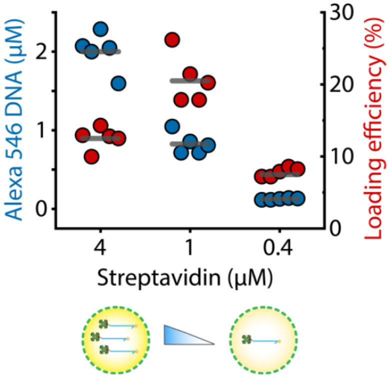
Dataset #	y_0 (μM)	c (μM)
1	1.83	15.7
2	6.83	14.1
3	2.08	15.1

Dataset #	y_0 (μM)	b (-)
1	20.1	0.704
2	17.6	0.794
3	21.4	0.817

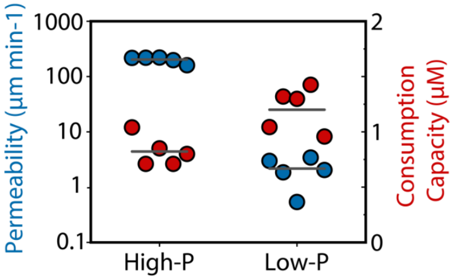
*Supplementary Figure 6.1: A) Upon cleavage of the photo-cleavable spacer (**PC**) by laser light irradiation (405 nm), the signal (strand **A_4**) and Cy5 quencher (strand **B**) are released. This reaction was monitored through unquenching of Cy5 and release of Cy3 from the proteinosomes. B) Cy5 fluorescence in three distinct proteinosomes over time upon irradiation by laser (405 nm, 2 h). The photo-cleavage was modeled as a first-order reaction, yielding the exponential equation shown in the figure. Since proteinosomes are heterogeneous in size, background fluorescence and DNA loading capacity, the background (y_0) was determined from the first timepoint for each proteinosome individually. In addition, a uniqueparameter c (maximum released concentration of reactive sender gate complex) was fitted for each proteinosome, whilst the rate of release (k), which should be independent of proteinosome characteristics, was fitted as a single parameter for all datasets. C) Similarly, the release of the Cy3 labeled signal strand was modeled using first-order kinetics. Now the fluorescence at the initial timepoint (y_0) indicates the total concentration of complex plus background fluorescence. The parameter b, which is fitted as unique parameter for eachdataset, signifies the fraction of the total fluorescence that is accounted for by the sender gate complex, making 1-b the fraction originating from background fluorescence. Experiments were performed in independent triplicates*



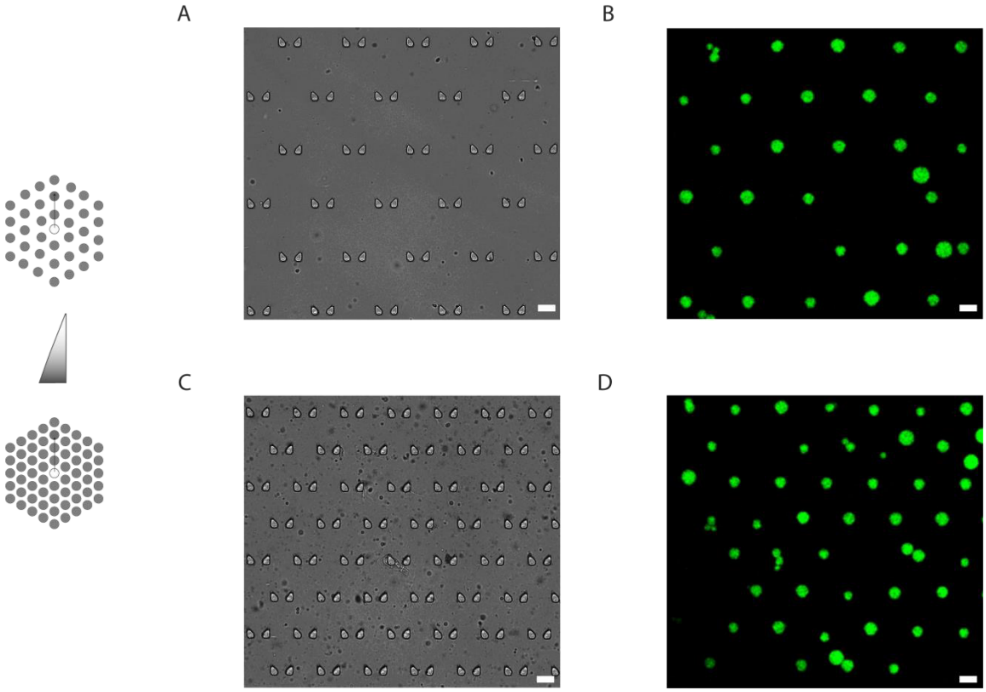
Supplementary Figure 6.2: A) A population of sender protocells were loaded into the trapping device and a single sender protocell was illuminated with laser light (405 nm, 100 ms every 1 s). B) Confocal micrographs of sender protocells showing time-dependent increase in Cy5 fluorescence of the irradiated sender due to the cleavage of $\boldsymbol{Q_1}$ *and dissociation of quencher-modified strand* ***B****. Scale bar 50 µm. C) Time-traces of cleaved* ***PC*** *linker in sender protocells through selective light irradiation. The concentration of cleaved* ***PC*** *linker was obtained through conversion of measured Cy5 fluorescence using calibration values of* $\boldsymbol{F_1}$ *strand. The data shows that only the irradiated sender protocells shows a time-dependent increase in concentration of cleaved duplex, indicating selective photo-cleavage. Sender protocells were prepared using 10 µM streptavidin.*



Supplementary Figure 6.3: DNA localization capacity of proteinosomes (Alexa 546-DNA concentration, blue dots) prepared using arange of streptavidin concentrations. The streptavidin concentration corresponds to the concentration of streptavidin present in the aqueous phase during proteinosomes assembly. In addition, loading efficiencies (red dots) were obtained under assumption that each streptavidin can bind four biotinylated ssDNA strands at maximum.



Supplementary Figure 6.4: Permeability (blue dots) and consumption capacity (red dots) of high-P and low-P proteinosomes. Permeability constants were estimated as reported previously.[14] The consumption capacity was determined through fluorescence measurements of proteinosomes that were loaded with Alexa 546-modified DNA.



Supplementary Figure 6.5: Bright-field micrograph of low-density (A) and high-density (C) trapping array. Fluorescence micrograph of FITC-labeled proteinosomes in a low-density (B) and high-density (D) trapping array. Scale bar 50 µm. The high and low density device contain 270 and 210 traps, respectively, in a 2 mm × 1.5 mm localization chamber.

Name	Sequence	Length	5’ Mod.	3’ Mod.
Figure 6.2-Figure 6.5, Supplementary Figure 6.2				
Q_1	CGA ACG AAC GAC CAT GAT AGA CTA ATG CAC TAC TAC /PC/ TAA CTA GC	44		Iowa Black
F_1	GCT AGT TAT TGT AGT AGT TTT TTT TT	26	Cy5	Biotin-TEG
Q_2	CGA ACG AAC GAC CAT GCG TGA AAC ATA GAC TAA TGC	36	Iowa Black	Phosphate
F_2	GTA GTA GTG CAT TAG TCT ATC ATG GTC GTT CGT TCG	36	Biotin-TEG	Alexa546
Supplementary Figure 6.1				
Q_7	CGA ACG AAC GAC CAT GAT AGA CTA ATG CAC TAC TAC /PC/ TAA CTA GC	44	Cy3	Iowa Black
F_1	GCT AGT TAT TGT AGT AGT TTT TTT TT	26	Cy5	Biotin-TEG
Figure 6.6-Figure 6.9				
Q_4	CAC ACA TCT ATA CAA CCA CTT ACT T /PC/ TA ACT ACC	33	Cy3	
F_4	GGT AGT TAT TAA GTA AGT TTT TTT TT	26		Biotin-TEG
Q_5	TAA CTA CC /PC/ G CCA TCA GAA CTT AAC CTA ACT CCT	33	Cy3	Iowa Black
F_5	TGA TGG CTT GGT AGT TAT TTT TTT T	25		Biotin-TEG
Q_6	CTA TAC AAC CAC TTA CTT /BHQ/ GC CAT CAG AAC TTA ACC	35		
F_6	AGG AGT TAG GTT AAG TTC TGA TGG C /Quasar670/ AAG TAA GTG GTT GTA TAG ATG TGT GTT TTT TTT	58		Biotin-TEG

Supplementary Table 6.1: Summary of DNA sequences used in this chapter.

Figures	Streptavidin Sender (μM)	Binding Capacity (μM)	Streptavidin Receiver (μM)	Consumption Capacity (μM)	Receiver Permeability (μm/min)	Receiver Density (traps/mm^2)	Exonuclease (unit/μL)
6.5a (high), 6.5c (high)	10	19.3	4	2	202.8	90	0
6.5a (med.), 6.5b (high), 6.5d (control)	10	19.3	1	0.82	202.8	90	0
6.5a (low)	10	19.3	0.4	0.25	202.8	90	0
6.5b (low)	10	19.3	4	0.82	2.16	90	0
6.5c (low)	10	19.3	4	2	202.8	70	0
6.5d (high)	10	19.3	1	0.82	202.8	90	0.1
6.5d (low)	10	19.3	1	0.82	202.8	90	0.05
6.8, 6.7, 6.9	30	40	1	0.82	202.8	90	0

Supplementary Table 6.2: Summary of experimental conditions used in this chapter, ordered by figure.

6.9 Supplementary Code

6.9.1 Introduction

This is the code that is used for data analysis in this thesis and the corresponding paper.[44] Proteinosome detection was based on previous work from our group.[14] The data processing performed here was based on and inspired by the work of Li et al.[7] The code was written for MATLAB 2019b.

Concepts

Experiments performed on a Leica confocal microscope were exported to TIFF files (one file per channel) and metadata XML files. To go from this data to the analysis performed in the paper, two struct are defined:

- **S** for settings. This holds all settings and datapaths defined before the analysis and is not changed during data processing.
- **D** for data. This struct is filled up with data during data processing and analysis and holds amongst others proteinosome positions, sizes and fluorescence data.

Data processing functions typically take **S** and **D** as function arguments and return an updated struct **D**, e.g.: `D = Processing.findPsomes(S, D);`

To define and start an analysis, define all settings in struct S, using *+Analysis/settings_defaul.m* as template. Subsequently run *+Analysis/calculate.m* in order to perform the data processing and create struct **D**.

Plotting

Some of the processed data for figure 4 is included in the *data* folder as example. This data can be loaded by loading the .mat file:

```
load('data/AND.mat')
```

To get an overview of the detected proteinosomes, run:

```
p1_image_processing
```

The fluorescence traces for all proteinosomes can be plotted using:

```
p2_psome_traces
```

To verify whether the correct proteinsomes have been identified as trigger, run:

```
p3_trigger_psome
```

In order to obtain a color-coded delay image from the data, similar to Figure 2D, execute:

```
p4_delay_image
```

The plots in Figure 4C can be reproduced using the following plot scripts:

```
p5_shells_part_1
p5_shells_part_2
p5_shells_schematic
```

Artificial color endpoint fluorescence graphs can be plotted using the script:

```
p6_endpoint_image
```

The signalling gradient at various timepoints (as in Figure 1G) can be plotted by executing:

```
p10_temp_integrated_signal
```

The characteristic signalling length scale can be calculated and visualized using the script (this is not very meaningful for the AND-gate example system):

```
p11_diffusion_gradient
```

In order to reproduce Figure 4D, the following script can be run:

```
p12_two_trigger_delay
```

A similar plot, showing the endpoint fluorescence instead of response time is generated by:

```
p12_two_trigger_endpoint
```

6.9.2 Code

+p1_image_processing.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Basic image analysis plots
f1 = figure;
imshow(D.firstFITC, [min(min(D.firstFITC)), max(max(D.firstFITC))]);
Plot.overlay_circles(S, D);
figure(f1);
Plot.overlay_circles(S, D, 'scale', S.radiusFactor, 'color', 'b');
figure(f1);
title('FITC, find circles');

f2 = figure;
imshow(D.firstFITC, [min(min(D.firstFITC)), max(max(D.firstFITC))]);
figure(f2);
Plot.overlay_numbers(S, D);
figure(f2);
title('FITC, find circles');
```

+p2_psome_traces.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Psome intensity plots
figure;
subplot(1, 3, 1);
Plot.intensities(S, D, 1); %, 'smooth', 20
title('FITC');
subplot(1, 3, 2);
Plot.intensities(S, D, 2); %, 'smooth', 20
title('Cy5');
subplot(1, 3, 3);
Plot.intensities(S, D, 3); %, 'smooth', 20
title('Alexa');

%% Normalized intensity plots
figure;
subplot(1, 3, 1);
Plot.intensities(S, D, 1, 'norm', true); %, 'smooth', 20
ylabel('Normalized Intensity (-)');
title('FITC');
subplot(1, 3, 2);
Plot.intensities(S, D, 2, 'norm', true); %, 'smooth', 20
ylabel('Normalized Intensity (-)');
title('Cy5');
subplot(1, 3, 3);
Plot.intensities(S, D, 3, 'norm', true); %, 'smooth', 20
ylabel('Normalized Intensity (-)');
title('Alexa');
```

+p3_trigger_psome.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Find Trigger Psome
figure(6);
subplot(1, 2, 1);
triggerImage = D.images(end, S.chTrigger);
Plot.imagech(triggerImage{1}, S.chTrigger);
Plot.overlay_circles(S, D, 'select', D.triggerPsome, 'scale', 1.5);
title('Trigger Psome');

subplot(1, 2, 2);
Plot.imagech(D.firstFITC, S.chFITC);
Plot.overlay_circles(S, D, 'select', D.triggerPsome, 'scale', 1.5);
title('Trigger Psome');
```

+p4_delay_image.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Delay Image
cm = Plot.cbrewer.cbrewer('div', 'RdYlBu', 100);
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);

Plot.delay_image(S, D, 'colormap', cm, 'nancolor', [0.5, 0.5, 0.5], ...
    'colorbardelta', 8, ... % Choose such that the ticks on the colorbar are nice
and round.
    'triggercolor', qualcolors(4, :) ...
    );
% Plot.overlayCircles(S, D, 'select', D.triggerPsome, ...
%     'scale', 1.5, 'color', qualcolors(4, :), 'lineWidth', 2);
hold on;
Plot.create_scale_bar(D, 100);
hold off;
```

+p5_shells_part_1.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Shells
figure;
shellcolors = flipud(Plot.cbrewer.cbrewer('seq', 'YlOrRd', D.nrShells));
dists = D.maxDists;
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);

for i = 1:D.nrShells
    means = mean(D.intensities(:,D.shells==i,S.chSignal) - D.intensi-
ties(1,D.shells==i,S.chSignal),2);
    plot(D.timeStamps, means, 'Color', shellcolors(i, :), 'LineWidth', 2);
    hold on;
end
set(gca, 'Color', [1,1,1].*0.8);
```

```
xlabel('Time (min)');
ylabel(D.intensityUnits{S.chSignal});
hold off;

colormap(shellcolors);
c = colorbar('Direction', 'reverse');

c.TickDirection = 'out';
ticks = linspace(0,1,D.nrShells+1);
ticks = ticks(2:end) - (ticks(2)/2);
c.Ticks = ticks;
c.TickLabels = arrayfun(@(x)(sprintf('%d', x)), 1:D.nrShells, 'UniformOutput',
false);
c.Label.String = 'Shell #';
```

+p5_shells_part_2.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Shells
figure;
imshow(D.firstFITC);
shellcolors = flipud(Plot.cbrewer.cbrewer('seq', 'YlOrRd', D.nrShells));
dists = D.maxDists;
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);

for i = 1:D.nrShells
    Plot.overlay_circles(S, D, 'select', D.shells == i, ...
        'scale', 1/S.radiusFactor, 'color', shellcolors(i, :));
    means = mean(D.intensities(:,D.shells==i,S.chSignal) - D.intensi-
ties(1,D.shells==i,S.chSignal),2);
end
Plot.overlay_circles(S, D, 'select', D.triggerPsome, ...
    'scale', 2, 'color', qualcolors(4, :), 'lineWidth', 2);

hold on;
Plot.create_scale_bar(D, 100);
hold off;
```

+p5_shells_schematic.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Shells
f = figure;
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);
shellcolors = flipud(Plot.cbrewer.cbrewer('seq', 'YlOrRd', D.nrShells));
examplebound = 4;

for p = 0:length(D.triggerPsome)
    [X,Y] = meshgrid(1:D.pixelLength, 1:D.pixelLength);
    dists = NaN(D.pixelLength, D.pixelLength, length(D.triggerPsome));
    if p == 0
        ps = 1:length(D.triggerPsome);
    else
        ps = p;
    end
    for i = ps
        tp = D.psomeCenters(D.triggerPsome(i),:);
        dist = sqrt((X-tp(1)).^2 + (Y-tp(2)).^2);
        dists(:,:,i) = dist;
    end
    maxdists = max(dists,[],3);

    minmd = min(min(maxdists));
    maxmd = max(max(maxdists));
    contouri = (D.shellBounds(:,1) >= minmd) & (D.shellBounds(:,1) <= maxmd);
    contouri(max(1, find(contouri, 1, 'first')-1)) = true;

    figure(f);
    hold on;
    if p == 0
        contourf(flipud(maxdists), D.shellBounds(:,1)', 'LineStyle', 'none', 'LineWidth', 0.5);
        cmorig = shellcolors(contouri,:);
        lsp = linspace(0, 1, size(cmorig, 1)-1);
        lsp = lsp(2:end) - (lsp(2) - lsp(1))/2;
        cm = interp1(lsp, cmorig(2:end-1,:), linspace(0, 1, 62), 'nearest', 'extrap');
        cm = [cmorig(1,:); cm; cmorig(end,:)];
        colormap(cm);
    else
        contour(flipud(maxdists), D.shellBounds(1:end,1), 'LineStyle', '--', 'LineWidth', 0.5, 'LineColor', 'k');
        if examplebound
            contour(flipud(maxdists), repmat(D.shellBounds(examplebound,1), 1, 2), 'LineStyle', '-', 'LineWidth', 2, 'LineColor', 'k');
        end
    end
    xlim([1,D.pixelLength]);
    ylim([1,D.pixelLength]);
    axis square;
    axis off;
    plot(D.psomeCenters(D.triggerPsome(ps),1), 1 + D.pixelLength - D.psomeCenters(D.triggerPsome(ps),2), 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', qualcolors(4,:), 'MarkerSize', 14, 'LineWidth', 1)
    hold off;
end

hold on;

sz = 100;
edgeSpacing = 20;
dims = D.imageDims;
pixelSize = sz / Basic.pix2dist(D, 1);
plot([dims(2) - pixelSize, dims(2)] - edgeSpacing + 0.5, ...
    [0,0]+edgeSpacing, 'Color', 'k', ...
    'LineWidth', 4);
text(dims(2) - pixelSize - edgeSpacing, ...
    edgeSpacing + 15, ...
    [num2str(sz) ' {\mu}m'], 'Color', 'k',...
    'FontUnits', 'normalized' , 'FontSize', 0.02, ...
    'FontName', 'Droid Sans');
hold off;
```

+p6_endpoint_image.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Endpoint Intensity Image
cm = Plot.cbrewer.cbrewer('seq', 'YlOrRd', 100);

figure;
Plot.intensity_image(S, D, 'colormap', cm, 'ColorBarDelta', 3);
title('Endpoint Intensities');
hold on;
Plot.create_scale_bar(D, 100);
hold off;
```

+p7_distance_plot.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Distance plot
cm = Plot.cbrewer.cbrewer('seq', 'YlOrRd', 100);
markercolor = cm(end, :);

dists = D.maxDists;%sqrt(sum((D.psomeCenters-D.psomeCenters(D.triggerPsome,:)).^2,
2));
[dists, dsorti] = sort(dists);

delays = D.delays;
timeupsSort = delays(dsorti);

figure;
plot(Basic.pix2dist(D, dists), timeupsSort, 'LineStyle', 'none', 'Marker', '.',
'MarkerSize', 10, 'Color', markercolor);
xlabel('Distance to sender ({\mu}m)');
ylabel('Delay (min)');
```

+p10_temp_integrated_signal.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Distance plot for various timepoints using average of region
timePoints = 0:10:110; % min
distPoints = 50:50:1000; % um
cm = Plot.cbrewer.cbrewer('seq', 'YlGn', length(timePoints));
intensities = interp1(D.timeStamps, D.intensities(:,:,S.chSignal), timePoints);

dists = D.maxDists;%sqrt(sum((D.psomeCenters-D.psomeCenters(D.triggerPsome,:)).^2,
2));
```

```
[dists, dsorti] = sort(dists);
dists = Basic.pix2dist(D, dists);

intensities = intensities(:,dsorti);
avintensities = nan(length(timePoints), length(distPoints) - 1);
for i = 2:length(distPoints)
    sel = (dists >= distPoints(i-1)) & (dists < distPoints(i));
    avintensities(:, i-1) = mean(intensities(:, sel) - intensities(1, sel), 2);
end

figure;
p = plot((distPoints(2)-distPoints(1))/2 + distPoints(1:(end-1)), avintensities',
'LineStyle', '-', 'Marker', '.', 'MarkerSize', 12, 'LineWidth', 2);
set(gca, 'Color', [1,1,1].*0.8);
xlabel('Distance to sender ({\mu}m)');
ylabel(D.intensityUnits{S.chSignal});
%xlim([0 600])

for pi = 1:length(p)
    p(pi).Color = cm(pi, :);
end

colormap(cm);
cb = colorbar;
ticks = linspace(0, 1, length(timePoints) + 1);
ticks = ticks(1:(end-1)) + (ticks(2) - ticks(1))/2;
cb.Ticks = ticks;
cb.TickLabels = num2cell(timePoints);

cb.TickLabels(2:2:end) = {''};

ylabel(cb, 'Timepoint (min)');
```

+p11_diffusion_gradient.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Shells
figure;
shellcolors = flipud(Plot.cbrewer.cbrewer('seq', 'YlOrRd', D.nrShells));
dists = Basic.pix2dist(D, D.maxDists);
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);

[w, w_ci, f, gof] = Processing.length_scale(D);

fitx = linspace(0, max(max(dists)), 100);
plot(fitx, f(fitx), 'Color', 'k', 'LineStyle', '-', 'LineWidth', 1.25);
hold on;
for i = 1:D.nrShells
    plot(dists(D.shells==i), D.intensities(end,D.shells==i,S.chSignal) - D.intensities(1,D.shells==i,S.chSignal), 'Color', shellcolors(i,:), 'LineStyle', 'none',
'Marker', '.', 'MarkerSize', 10);
    hold on;
end
set(gca, 'Color', [1,1,1].*0.8);
```

```
xlabel('Distance ({\mu}m)');
ylabel(['Endpoint ' D.intensityUnits{S.chSignal}]);
hold off;

w_ci
w
```

+p12_two_trigger_delay.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Distance plot
cm = Plot.cbrewer.cbrewer('seq', 'YlOrRd', 100);
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);
triggercolor = qualcolors(4, :);
markercolor = cm(end, :);
colorbartickdiff = 10;

dists = D.dists;
interdist = Basic.pix2dist(D, D.dists(D.triggerPsome(2),1));
delays = D.delays;
mindel = min(delays);
maxdel = max(delays);

maxdist = Basic.pix2dist(D, max(max(dists)));maxdist100 = 100*ceil(maxdist/100);

delayColors = interp1(linspace(mindel, maxdel, size(cm, 1)), cm, delays);

figure;
axis equal;
fakezero = 0;
extradist = maxdist100-interdist;
ph = patch([fakezero, fakezero, extradist, fakezero, interdist, interdist+ex-
tradist, interdist+extradist], [fakezero, interdist+extradist, interdist+ex-
tradist, interdist, fakezero, extradist, fakezero], 'white', 'EdgeColor',
brighten(triggercolor, 0.85));
Plot.stripes(ph, 45, 4, 1, '-');
hold on;

for i = 1:length(delays)
    if any(i == D.triggerPsome)
        clr = triggercolor;
        msz = 7;
        plot(Basic.pix2dist(D, dists(i,1)), Basic.pix2dist(D, dists(i,2)),
'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', msz, 'Color', clr, 'MarkerFace-
Color', clr);
    elseif isnan(delays(i))
        clr = [0.5, 0.5, 0.5];
        msz = 7;
        plot(Basic.pix2dist(D, dists(i,1)), Basic.pix2dist(D, dists(i,2)),
'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', msz, 'Color', clr);
    end
end
for i = 1:length(delays)
    if ~isnan(delays(i))
```

```
        clr = delayColors(i,:);
        msz = 10;
        plot(Basic.pix2dist(D, dists(i,1)), Basic.pix2dist(D, dists(i,2)),
'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', msz, 'Color', clr, 'MarkerFace-
Color', clr);
    end
end
hold off;
xlabel('Distance to sender 1 ({\mu}m)');
ylabel('Distance to sender 2 ({\mu}m)');
axis equal;
xlim([0, maxdist100]);
ylim([0, maxdist100]);

set(gca,'Layer','top');

colormap(cm);
cb = colorbar;

ticks = 0:colorbartickdiff:maxdel;
ticks = ticks(ticks >= mindel);
cb.Ticks = interp1([mindel, maxdel], [0, 1], ticks);
cb.TickLabels = ticks;
ylabel(cb, 'Delay (min)');
```

+p12_two_trigger_endpoint.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Distance plot
cm = Plot.cbrewer.cbrewer('seq', 'YlOrRd', 100);
qualcolors = Plot.cbrewer.cbrewer('qual', 'Paired', 12);
triggercolor = qualcolors(4, :);
markercolor = cm(end, :);
colorbartickdiff = 0.01;

dists = D.dists;
interdist = Basic.pix2dist(D, D.dists(D.triggerPsome(2),1));
endpoints = D.intensities(end,:,S.chSignal);
minep = min(endpoints);
maxep = max(endpoints);

maxdist = Basic.pix2dist(D, max(max(dists)));
maxdist100 = 100*ceil(maxdist/100);

epColors = interp1(linspace(minep, maxep, size(cm, 1)), cm, endpoints);

figure;
axis equal;
fakezero = 0;
extradist = maxdist100-interdist;
ph = patch([fakezero, fakezero, extradist, fakezero, interdist, interdist+ex-
tradist, interdist+extradist], [fakezero, interdist+extradist, interdist+ex-
tradist, interdist, fakezero, extradist, fakezero], 'white', 'EdgeColor',
brighten(triggercolor, 0.85));
```

```
Plot.stripes(ph, 45, 4, 1, '-');
hold on;

for i = 1:length(endpoints)
    if any(i == D.triggerPsome)
        clr = triggercolor;
        msz = 7;
        plot(Basic.pix2dist(D, dists(i,1)), Basic.pix2dist(D, dists(i,2)),
'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', msz, 'Color', clr, 'MarkerFace-
Color', clr);
    elseif isnan(endpoints(i))
        clr = [0.5, 0.5, 0.5];
        msz = 7;
        plot(Basic.pix2dist(D, dists(i,1)), Basic.pix2dist(D, dists(i,2)),
'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', msz, 'Color', clr);
    end
end
for i = 1:length(endpoints)
    if ~isnan(endpoints(i))
        clr = epColors(i,:);
        msz = 10;
        plot(Basic.pix2dist(D, dists(i,1)), Basic.pix2dist(D, dists(i,2)),
'LineStyle', 'none', 'Marker', 'o', 'MarkerSize', msz, 'Color', clr, 'MarkerFace-
Color', clr);
    end
end
hold off;
xlabel('Distance to sender 1 ({\mu}m)');
ylabel('Distance to sender 2 ({\mu}m)');
axis equal;
xlim([0, maxdist100]);
ylim([0, maxdist100]);

set(gca,'Layer','top')

colormap(cm);
cb = colorbar;

ticks = 0:colorbartickdiff:maxep;
ticks = ticks(ticks >= minep);
cb.Ticks = interp1([minep, maxep], [0, 1], ticks);
cb.TickLabels = ticks;
ylabel(cb, D.intensityUnits{S.chSignal});
```

+Analysis/setttings_default.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Settings

%% Input
S = struct();
S.dataPath = '/path/to/data/';
S.expName = 'experiment_folder';
% To merge multiple experiments: {'Exp1','Exp2'}
```

```
%% Psome detection
S.radiusMin = 5; % Minimum radius for classification as proteinosome
S.radiusMax = 40; % Maximum radius for classification as proteinosome
S.radiusFactor = 0.5; % Scaling factor used for plotting

% Which psomes to ignore; empty list ([]) will result in an input prompt;
% Zero (0 or [0]) will not ignore any psomes.
S.ignorePsomes = [];

%% Channels
S.chFITC = 1;
S.chTrigger = 3;
S.chSignal = 2;

%% Sorting
S.yCutoffs = [200 300 400 550 650 800 900 1200];
% Cutoffs are used to get more human readable numbering.
% Does not influence the results.

%% Shells
S.shellThresholds = 50:50:350;

%% Delays
% Channel in which to detect a delay.
S.delayChannel = 2;
% Whether to use normalized data to detect the delay.
S.delayOnNormalizedData = true;
% The (normalized) intensity threshold for delay detection.
S.delayIntensityThreshold = 0.5;
% Minimum absolute increase in signal.
S.delayMinimumIncrease = 20;
% Intensity reference function (intensity to concentration in microM)
% per channel.
S.refFunc = {[], @(x)(x/40), []}; % 0.5% laser power
%S.refFunc = {[], [], []}; % No reference done
```

+Analysis/calculate.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

%% Data input
% Optionally load a cached analysis
if iscell(S.expName)
    cachefile = [S.dataPath S.expName{1} '/MetaData/datacache.mat'];
else
    cachefile = [S.dataPath S.expName '/MetaData/datacache.mat'];
end
if exist(cachefile, 'file')
    if strcmpi(input('Do you want to [R]ecalculate or load from [c]ache?, [R]:
','s'), 'c')
        load(cachefile, 'D', 'S');
        return;
    end
end

D = Basic.get_images_and_timestamps(S);
D = Processing.clip_timeseries(D);

%% Image analysis
% Identify the initial location of the psomes from the first FITC image
D.firstImages = D.images(1, :);
D.firstFITC = cell2mat(D.firstImages(1, S.chFITC));

D = Processing.find_psomes(S, D);
D = Processing.remove_edge_psomes(S, D);
D = Processing.order_psomes(S, D);

% Ignore some proteinosomes
D.ignorePsomes = S.ignorePsomes;
if isempty(S.ignorePsomes)
    f = figure;
    imshow(D.firstFITC);
    figure(f);
    Plot.overlay_numbers(S, D);
    title('FITC, find circles');
    igps = input('Proteinosomes to ignore (e.g. [34, 56, 78]): ');
    D.ignorePsomes = igps;
end
if ~isempty(D.ignorePsomes) && ~(isscalar(D.ignorePsomes) && D.ignorePsomes == 0)
    indices = true(D.nrPsomes, 1);
    indices(D.ignorePsomes) = false;
    D.psomeCenters = D.psomeCenters(indices, :);
    D.psomeRadii = D.psomeRadii(indices, :);
    D.psomeMetric = D.psomeMetric(indices, :);
    D.nrPsomes = size(D.psomeCenters, 1);
end

%% Proteinosome Intensities
D = Basic.detect_proteinosome_intensity( S, D );

D.intensitiesNorm = D.intensities;
D.intensitiesNorm(:,:,1) = D.intensitiesNorm(:,:,1) ./ max(D.intensities-
Norm(:,:,1));
D.intensitiesNorm(:,:,2) = D.intensitiesNorm(:,:,2) ./ max(D.intensities-
Norm(:,:,2));
D.intensitiesNorm(:,:,3) = D.intensitiesNorm(:,:,3) ./ max(D.intensities-
Norm(:,:,3));

%% Calculate concentration
D.rawIntensities = D.intensities;
D.intensityUnits = {'Intensity (a.u.)', 'Intensity (a.u.)', 'Intensity (a.u.)'};
for i = 1:3
    func = S.refFunc{i};
    if ~isempty(func)
        D.intensityUnits{i} = {'Corrected activated', 'DNA gate (M)'};
        D.intensities(:,:,i) = func(D.rawIntensities(:,:,i));
    end
```

```
end

%% Find Trigger Proteinosome(s)
triggerData = D.intensities(end,:,S.chTrigger);
D.triggerPsome = find(abs(triggerData-mean(triggerData)) > std(triggerData));

%% Delays
D = Processing.calculate_delays(S, D);

%% Find Shells
D = Processing.find_shells(S, D);

%% Cache Analysis
save(D.datacache, 'D', 'S','-v7.3');
```

+Basic/detect_proteinosome_intensity.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = detect_proteinosome_intensity( S, D )

nrPsomes = D.nrPsomes;
nrImages = D.nrImages;
psomeCenters = D.psomeCenters;
psomeRadii = D.psomeRadii;

disp('Loading images into memory');
images = D.images;

intensities = zeros(nrImages, nrPsomes, 3);

imgRes = D.imageDims(1, 1); %Images must be square

[rr, cc] = meshgrid(1:imgRes);

for i=1:nrPsomes
        C(:,:,i) = sqrt((rr-psomeCenters(i,1)).^2+(cc-psomeCenters(i,2)).^2)<=psomeRadii(i)*S.radiusFactor;
end
tic;
for k=1:nrImages
    fprintf('Image %d\n', k)

    for i=1:nrPsomes
        intensities(k,i,1) = mean(images{k,1}(C(:,:,i)));
        intensities(k,i,2) = mean(images{k,2}(C(:,:,i)));
        intensities(k,i,3) = mean(images{k,3}(C(:,:,i)));
    end
end
toc

D.intensities = intensities;

end
```

+Basic/get_images_and_timestamps.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function [ D ] = get_images_and_timestamps( S )
if ~iscell(S.expName)
    expNames = {S.expName};
else
    expNames = S.expName;
end

D = struct();
D.datacache = [];
D.images = [];
D.timeStamps = [];
D.expTimeAndDate = [];

for i = 1:length(expNames)
    expName = expNames{i};
    dpath = strcat(S.dataPath, expName);
    datacache = [dpath '/MetaData/datacache.mat'];

    dirData = dir(dpath);
    dirIndex = [dirData.isdir];  % Find the index of directories
    fileList = {dirData(~dirIndex).name}';  %remove directories from the filelist
    timeStampList = [];
    imageList = {};

    for j=1:length(fileList)
        filename1 = char(fileList(j));
        filename = [dpath '/' filename1];
        %imageList{i} = imread([dpath '/' filename]);
        fileinfo = imfinfo(filename);
        num_images = numel(fileinfo);
        for k = 1:num_images
            A = imread(filename, k, 'Info', fileinfo);
            imageList{k,j} = A;
        end
    end

    %get timing information from metadata
    fname_meta = filename1(1:end-9);  %% change this to 8 if less than 100 images
    fname_meta = [dpath '/MetaData/' fname_meta '_Properties.xml'];
    xDoc = xmlread(fname_meta);
    import javax.xml.xpath.*
    factory = XPathFactory.newInstance;
    xpath = factory.newXPath;
    expression = xpath.compile('Data/Image/ImageDescription/Dimensions/Dimension-
Description[@DimID=''X'']/@Length');
    node = expression.evaluate(xDoc, XPathConstants.NODE);
    realLength = str2double(strrep(char(node.getTextContent), ',', ''));
    expression = xpath.compile('Data/Image/ImageDescription/Dimensions/Dimension-
Description[@DimID=''X'']/@NumberOfElements');
```

```
    node = expression.evaluate(xDoc, XPathConstants.NODE);
    pixelLength = str2double(strrep(char(node.getTextContent), ',', ''));
    expression = xpath.compile('Data/Image/TimeStampList');
    node = expression.evaluate(xDoc, XPathConstants.NODE);
    inode = node.getFirstChild;
    expTimeAndDate = [char(node.getAttribute('FirstTimeStampDate')) ' '
char(node.getAttribute('FirstTimeStampTime'))];

    timeStampList = [];
    while ~isempty(inode)
        if strcmp(inode.getNodeName, 'TimeStamp')
            timeStampList(end + 1) = str2double(char(inode.getAttribute('Relative-
Time')));
        end
        inode = inode.getNextSibling;
    end

    expression = xpath.compile('Data/Image/ImageDescription/Channels');
    node = expression.evaluate(xDoc, XPathConstants.NODE);
    inode = node.getFirstChild;

    nrChannels = 0;
    while ~isempty(inode)
        if strcmp(inode.getNodeName, 'ChannelDescription')
            nrChannels = nrChannels + 1;
        end
        inode = inode.getNextSibling;
    end
    timeStamps1 = 1:nrChannels:size(timeStampList, 2);
    timeStampList = timeStampList(1, timeStamps1);
    if isempty(D.images)
        D.datacache = datacache;
        D.images = imageList;
        D.timeStamps = timeStampList;
        D.expTimeAndDate = expTimeAndDate;
        D.realLength = realLength;
        D.pixelLength = pixelLength;
    else
        minDim = min(size(D.images, 2), size(imageList, 2));
        D.images = D.images(:, 1:minDim);
        D.images = [D.images; imageList(:, 1:minDim)];
        D.timeStamps = [D.timeStamps (D.timeStamps(end) + timeStampList)];
        if ~(D.realLength == realLength && D.pixelLength == pixelLength)
            error('Cannot merge experiments!');
        end
    end
end
save(datacache, 'D', '-v7.3');
end
```

+Basic/pix2dist.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef
```

```
function [ dist ] = pix2dist( D, pix )
dist = pix .* (D.realLength / (D.pixelLength - 1));
end
```

+Basic/sort_psomes.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef
function [order, initialCenters] = sort_psomes(initialCenters, yCutoffs)
order = 1:length(initialCenters);
% Selection sort
n = length(initialCenters);
for j = 1:(n - 1)
    % Find jth smallest element
    imin = j;
    for i = (j + 1):n
        if (psomeIsLowerPos(initialCenters(i,:),initialCenters(imin,:),yCutoffs))
            imin = i;
        end
    end
    % Put jth smallest element in place
    if (imin ~= j)
        initialCenters = swap(initialCenters,imin,j);
        order = swap1(order,imin,j);
    end
end
end

function x = swap(x,i,j)
% Swap x(i) and x(j)
% Note: In practice, x xhould be passed by reference

val = x(i,:);
x(i,:) = x(j,:);
x(j,:) = val;
end

function x = swap1(x,i,j)
% Swap x(i) and x(j)
% Note: In practice, x xhould be passed by reference
val = x(i);
x(i) = x(j);
x(j) = val;
end

function answ = psomeIsLowerPos(a,b,yCutoffs)  %1 if a is lower, 0 if b
    aYpos = find(yCutoffs>a(2),1);
    bYpos = find(yCutoffs>b(2),1);

    if (aYpos < bYpos)
        answ = 1;
    elseif (aYpos > bYpos)
        answ = 0;
    else
```

```
        if(a(1)<b(1))
            answ = 1;
        else
            answ = 0;
        end
    end

end
```

+Processing/calculate_delays.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = calculate_delays( S, D )

delayIntensityThreshold = S.delayIntensityThreshold;
delayChannel = S.delayChannel;

nrPsomes = D.nrPsomes;
timeStamps = D.timeStamps;

absoluteIntensities = D.rawIntensities(:, :, delayChannel);
if ~S.delayOnNormalizedData
    intensityProbe = absoluteIntensities;
else
    intensityProbe = D.intensitiesNorm(:, :, delayChannel);
end

deltas = find((absoluteIntensities(end, :) - absoluteIntensities(1, :)) >= S.delayMinimumIncrease);

delays = nan(1, nrPsomes);
for k = deltas
    tup = timeStamps(find(intensityProbe(:, k) > delayIntensityThreshold, 1, 'first'));

    if ~isempty(tup)
        delays(k) = tup;
    end
end

D.delays = delays;

end
```

+Processing/clip_timeseries.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = clip_timeseries( D )

infile = @(var)ismember(var, fields(D));

ydim = size(cell2mat(D.images(1,1)), 1);
```

```
spacer = {255.*ones(ydim, 5, 'uint8')};
nrImages = size(D.images, 1);
D.startPos = GUI.crop_slider( ...
    @(i)(cell2mat([D.images(i,1) spacer D.images(i,2) spacer D.images(i,3)])), ...
    nrImages, 1, ...
    'Select reaction start time');
D.endPos = GUI.crop_slider( ...
    @(i)(cell2mat([D.images(i,1) spacer D.images(i,2) spacer D.images(i,3)])), ...
    nrImages, nrImages, ...
    'Select reaction end time');

if D.startPos ~= 1 || D.endPos ~= nrImages
    D.images = D.images(D.startPos:D.endPos, :);
end

D.timeStamps = (D.timeStamps(D.startPos:D.endPos))';
D.timeStamps = D.timeStamps./60;
D.nrImages = size(D.timeStamps, 1);

D.imageDims = size(cell2mat(D.images(1,1)));

end
```

+Processing/find_psomes.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = find_psomes( S, D )

figure(1);
subplot(2, 2, 1);
imshow(D.firstFITC);
title('FITC original');

subplot(2, 2, 2);
imshow(cell2mat(D.firstImages(1, S.chTrigger)));
title('Probe original');

J = imadjust(D.firstFITC); % adjust the image so that the whole 8 bit dynamic range is used
subplot(2, 2, 3);
imshow(J);
title('FITC, contrast stretched')

K = medfilt2(J);  %median filter to remove noise
subplot(2, 2, 4);
imshow(K);
title('FITC, median filter');

[initialCenters, initialRadii, metric] = imfindcircles(K, [S.radiusMin S.radiusMax], 'Method','TwoStage','Sensitivity',0.9);

D.psomeCenters = initialCenters;
D.psomeRadii = initialRadii;
D.psomeMetric = metric;
```

```
D.nrPsomes = length(initialRadii);

end
```

+Processing/find_shells.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = find_shells( S, D )

dists = NaN(length(D.psomeCenters), length(D.triggerPsome));
for i = 1:length(D.triggerPsome)
    dist = sqrt(sum((D.psomeCenters-D.psomeCenters(D.triggerPsome(i),:)).^2, 2));
    dists(:,i) = dist;
end

maxdists = max(dists,[],2);

shells = zeros(length(maxdists), 1);

thr = S.shellThresholds;
thr = [0 thr inf];
for i = 1:(length(thr) -1)
    shells((maxdists > thr(i)) & (maxdists <= thr(i+1))) = i;
end

startShell = 1;
if length(D.triggerPsome) > 1
    startShell = min(shells);
end

shells = shells + 1 - startShell;
bounds = [thr(1:end-1)' thr(2:end)'];
bounds = bounds(startShell:end, :);
bounds(1, 1) = 0;

D.shells = shells;
D.shellBounds = bounds;
D.nrShells = size(D.shellBounds, 1);
D.dists = dists;
D.maxDists = maxdists;

end
```

+Processing/length_scale.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function [w, w_ci, f, gof] = length_scale(D)
diffexpr = @(c, d, x)(c.*exp(-(10.^d).*x.^2));

%% Calculate distances and convert to um
% dists = sqrt(sum((D.psomeCenters-D.psomeCenters(D.triggerPsome,:)).^2, 2));
```

```
dists = D.maxDists;
dists = Basic.pix2dist(D, dists);

%% Get psome intensities and substract background
fl = D.intensities(end, :, 3)';
fl = fl - D.intensities(1, :, 3)';

fl(D.triggerPsome) = [];
dists(D.triggerPsome) = [];

[f, gof] = fit(dists, fl, diffexpr, 'StartPoint', [0.3,-4]);

scale = @(d)(sqrt(1/(10.^d)));

w = scale(f.d);

ci = confint(f, 0.95);
w_ci = [scale(ci(1,2)), scale(ci(2,2))];
end
```

+Processing/order_psomes.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = order_psomes( S, D )

[psomeOrder, ~] = Basic.sort_psomes( D.psomeCenters, S.yCutoffs );

D.psomeCenters = D.psomeCenters(psomeOrder,:);
D.psomeRadii = D.psomeRadii(psomeOrder,:);

end
```

+Processing/remove_edge_psomes.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function D = remove_edge_psomes( ~, D )

lim = D.psomeRadii;

goodPsomes = (D.psomeCenters(:, 1) > lim) & ...
    (D.psomeCenters(:, 2) > lim) & ...
    ((D.imageDims(1) - D.psomeCenters(:, 1)) > lim) & ...
    ((D.imageDims(2) - D.psomeCenters(:, 2)) > lim);

lst = 1:D.nrPsomes;
lst(~goodPsomes)

D.psomeCenters = D.psomeCenters(goodPsomes, :);
D.psomeRadii = D.psomeRadii(goodPsomes, 1);
D.psomeMetric = D.psomeMetric(goodPsomes, :);
D.nrPsomes = sum(goodPsomes, 1);
```

```
end
```

+GUI/crop_slider.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function [ pos ] = crop_slider( imfun, imax, istart, name )

f = createGUI();
uiwait(f);

    function hFig = createGUI()
        hFig = figure('Position',[100 100 1000 500],'Units','normalized', 'Color', 'white', 'Name', name);
        pos = istart;
        handles.axes1 = axes('Units','normalized','Position',[.1 .1 .8 .9]);

        %// Create slider and listener object for smooth visualization
        handles.SliderFrame = uicontrol('Style','slider','Position',[60 20 600 30],'Min',1,'Max',imax,'Value',pos,'SliderStep',[1/imax 2/imax],'Callback',@XSliderCallback);
        %handles.SliderxListener = addlistener(handles.SliderFrame,'Value','PostSet',@(s,e) XSliderCallback);

        handles.Text1 = uicontrol('Style','Text','Position',[720 20 60 30],'String','Current frame');
        handles.Edit1 = uicontrol('Style','Edit','Position',[790 20 100 30],'String','1');
        handles.Button1 = uicontrol('Style','PushButton','Position',[910 20 80 30],'String','Continue','Callback',@(s,e)close(hFig));

        %// Display 1st frame
        imshow(imfun(pos))
        setappdata(hFig,'pos',pos);

        %// IMPORTANT. Update handles structure.
        guidata(hFig,handles);
        XSliderCallback([],[]);

        %// Slider callback; executed when the slider is release or you press
        %// the arrows.
            function XSliderCallback(~,~)

                handles = guidata(gcf);

                pos = round((get(handles.SliderFrame,'Value')));
                set(handles.Edit1,'String',num2str(pos));
                imshow(imfun(pos),'Parent',handles.axes1);

                guidata(hFig,handles);
            end
    end
end
```

+Plot/create_scale_bar.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function create_scale_bar( D, sz )

edgeSpacing = 20;

dims = D.imageDims;

pixelSize = sz / Basic.pix2dist(D, 1);

plot([dims(2) - pixelSize, dims(2)] - edgeSpacing + 0.5, ...
    [dims(1), dims(1)]-edgeSpacing, 'Color', 'w', ...
    'LineWidth', 4);
text(dims(2) - pixelSize - edgeSpacing, ...
    dims(1) - edgeSpacing + 10, ...
    [num2str(sz) ' {\mu}m'], 'Color', 'w',...
    'FontUnits', 'normalized' , 'FontSize', 0.02, ...
    'FontName', 'Droid Sans');

end
```

+Plot/delay_image.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function [ h ] = delay_image( ~, D, varargin )
p = inputParser;
addParameter(p, 'original', [], @ismatrix);
addParameter(p, 'colormap', 'jet');
addParameter(p, 'nancolor', [0.5 0.5 0.5]);
addParameter(p, 'colorbardelta', 6, @isnumeric);
addParameter(p, 'triggercolor', [1,1,1]);
parse(p, varargin{:});

f = gcf;

delays = D.delays;
psomeCenters = D.psomeCenters;
psomeRadii = D.psomeRadii;
triggerPsome = D.triggerPsome;

if isempty(p.Results.original)
    timeim = zeros(D.imageDims, 'uint8');
else
    timeim = p.Results.original;
end

maxTime = max(delays);
maxTimeFl = floor(maxTime);
maxTimeFl = floor(maxTimeFl/(p.Results.colorbardelta - 1))*(p.Results.colorbardelta - 1);

cm = colormap(p.Results.colormap);
cmfunc = @(x)(interp1(linspace(0,maxTime,size(cm, 1)), cm, x));
cdata = ones(length(delays), 3) .* p.Results.nancolor .* 255;
cdata(~isnan(delays),:) = cmfunc(delays(~isnan(delays))).*255;

cdata(triggerPsome,:) = repmat(p.Results.triggercolor .* 255, length(triggerPsome), 1);

timeim = insertShape(timeim,'filledcircle',[psomeCenters
psomeRadii],'color',cdata, 'opacity', 1);

figure(f);
imshow(timeim);
im = gca;
figure(f);
cbcm = interp1(linspace(0,95,size(cm, 1)), cm, 0:95);
cbcm = [cbcm; repmat(p.Results.nancolor, 5, 1)];
colormap(cbcm);
figure(f);
cb = colorbar('Ticks',[linspace(0,1,p.Results.colorbardelta).*0.95.*(maxTimeFl/
maxTime), 1],...
         'TickLabels',[num2cell(linspace(0,maxTimeFl,p.Results.colorbardelta)),
{'n.d.'}]);
ylabel(cb, 'Delay (min)');

end
```

+Plot/imagech.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function [ h ] = imagech( im, ch )

if ch == 1
    cm = interp1([0 1], [0 0 0; 0 1 0], linspace(0, 1, 256));
elseif ch == 2
    cm = interp1([0 1], [0 0 0; 1 1 0], linspace(0, 1, 256));
elseif ch == 3
    cm = interp1([0 1], [0 0 0; 1 0 0], linspace(0, 1, 256));
else
    cm = interp1([0 1], [0 0 0; 1 1 1], linspace(0, 1, 256));
end

h = imshow(im, cm);

end
```

+Plot/intensities.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef
```

```
function h = intensities( ~, D, channel, varargin )
p = inputParser;
addRequired(p, 'channel', @isnumeric);
addParameter(p,'smooth', 0, @isnumeric);
addParameter(p,'norm', false, @islogical);
parse(p, channel, varargin{:});

if ~p.Results.norm
    data = D.intensities(:,:,channel);
else
    data = D.intensitiesNorm(:,:,channel);
end

if ~p.Results.smooth
    h = plot(D.timeStamps, data);
else
    h = plot(D.timeStamps, movmean(data, p.Results.smooth));
end
ylabel(D.intensityUnits{channel});
xlabel('Time (min)');

end
```

+Plot/intensity_image.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function intensity_image( S, D, varargin )
p = inputParser;
addParameter(p, 'colormap', 'jet');
addParameter(p, 'colorbardelta', 6, @isnumeric);
parse(p, varargin{:});

f = gcf;

iend = D.rawIntensities(end,:,S.delayChannel);
psomeCenters = D.psomeCenters;
psomeRadii = D.psomeRadii;

im = zeros(D.imageDims, 'uint8');

maxi = max(iend);
maxiFl = floor(maxi*1000)/1000;
cm = colormap(p.Results.colormap);
cmfunc = @(x)(interp1(linspace(0,maxi,size(cm, 1)), cm, x));
cdata = cmfunc(iend).*255;

im = insertShape(im,'filledcircle',[psomeCenters psomeRadii],'color',cdata, 'opac-
ity', 1);

figure(f);
imshow(im);
cb = colorbar('Ticks',linspace(0, 1, p.Results.colorbardelta) .* (maxiFl/maxi),...
         'TickLabels', num2cell(linspace(0, maxiFl, p.Results.colorbardelta)));
ylabel(cb, 'Endpoint Intensity (-)');
```

```
end
```

+Plot/overlay_circles.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function overlay_circles( ~, D, varargin )
p = inputParser;
addParameter(p, 'scale', 1.0, @isnumeric);
addParameter(p, 'color', 'r');
addParameter(p, 'lineWidth', 1.0, @isnumeric);
addParameter(p, 'select', [], @isvector);
parse(p, varargin{:});
f = gcf;
if isempty(p.Results.select)
    psomeCenters = D.psomeCenters;
    psomeRadii = D.psomeRadii;
else
    psomeCenters = D.psomeCenters(p.Results.select, :);
    psomeRadii = D.psomeRadii(p.Results.select, 1);
end

figure(f);
viscircles(psomeCenters, psomeRadii .* p.Results.scale, 'DrawBackgroundCircle',
false,'EdgeColor',p.Results.color,'LineWidth',p.Results.lineWidth);
end
```

+Plot/overlay_numbers.m

```
% Copyright (C) 2020
% Pascal A. Pieters, Alex Joesaar, Shuo Yang and T.F.A. de Greef

function overlay_numbers( ~, D )

f = gcf;

psomeCenters = D.psomeCenters;
nrPsomes = D.nrPsomes;

for i=1:nrPsomes
    figure(f);
    text(psomeCenters(i,1), psomeCenters(i,2), sprintf('%d', i), 'HorizontalAlign-
ment', 'center','VerticalAlignment', 'middle','Color','red','FontSize',14);
end

end
```

6.10 References

(1) Toda, S., Frankel, N. W., and Lim, W. A. (2019) Engineering cell–cell communication networks: programming multicellular behaviors. *Current Opinion in Chemical Biology 52*, 31–38.

(2) Hart, Y., Antebi, Y. E., Mayo, A. E., Friedman, N., and Alon, U. (2012) Design principles of cell circuits with paradoxical components. *Proceedings of the National Academy of Sciences 109*, 8346–8351.

(3) Doğaner, B. A., Yan, L. K. Q., and Youk, H. (2016) Autocrine Signaling and Quorum Sensing: Extreme Ends of a Common Spectrum. *Trends in Cell Biology 26*, 262–271.

(4) Barcena Menendez, D., Senthivel, V. R., and Isalan, M. (2015) Sender–receiver systems and applying information theory for quantitative synthetic biology. *Current Opinion in Biotechnology 31*, 101–107.

(5) Miller, M. B., and Bassler, B. L. (2001) Quorum Sensing in Bacteria. *Annu. Rev. Microbiol. 55*, 165–199.

(6) Zhu, J., Yamane, H., and Paul, W. E. (2010) Differentiation of Effector CD4 T Cell Populations. *Annu. Rev. Immunol. 28*, 445–489.

(7) Li, P., Markson, J. S., Wang, S., Chen, S., Vachharajani, V., and Elowitz, M. B. (2018) Morphogen gradient reconstitution reveals Hedgehog pathway design principles. *Science 360*, 543–548.

(8) Carvalho, A., Menendez, D. B., Senthivel, V. R., Zimmermann, T., Diambra, L., and Isalan, M. (2014) Genetically Encoded Sender–Receiver System in 3D Mammalian Cell Culture. *ACS Synth. Biol. 3*, 264–272.

(9) Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., and Weiss, R. (2005) A synthetic multicellular system for programmed pattern formation. *Nature 434*, 1130–1134.

(10) Ortiz, M. E., and Endy, D. (2012) Engineered cell-cell communication via DNA messaging. *J Biol Eng 6*, 16.

(11) Salehi-Reyhani, A., Ces, O., and Elani, Y. (2017) Artificial cell mimics as simplified models for the study of cell biology. *Exp Biol Med (Maywood) 242*, 1309–1317.

(12) Findlay, H. E., Harris, N. J., and Booth, P. J. (2016) In vitro synthesis of a Major Facilitator Transporter for specific active transport across Droplet Interface Bilayers. *Scientific Reports 6*, 39349.

(13) Tang, T.-Y. D., Cecchi, D., Fracasso, G., Accardi, D., Coutable-Pennarun, A., Mansy, S. S., Perriman, A. W., Anderson, J. L. R., and Mann, S. (2018) Gene-Mediated Chemical Communication in Synthetic Protocell Communities. *ACS Synth. Biol. 7*, 339–346.

(14) Joesaar, A., Yang, S., Bögels, B., van der Linden, A., Pieters, P., Kumar, B. V. V. S. P., Dalchau, N., Phillips, A., Mann, S., and de Greef, T. F. A. (2019) DNA-based communication in populations of synthetic protocells. *Nat. Nanotechnol. 14*, 369–378.

(15) Zhang, D. Y., and Winfree, E. (2009) Control of DNA Strand Displacement Kinetics Using Toehold Exchange. *J. Am. Chem. Soc. 131*, 17303–17314.

(16) Huang, X., Li, M., Green, D. C., Williams, D. S., Patil, A. J., and Mann, S. (2013) Interfacial assembly of protein–polymer nano-conjugates into stimulus-responsive biomimetic protocells. *Nat Commun 4*, 2239.

(17) Oyler-Yaniv, A., Oyler-Yaniv, J., Whitlock, B. M., Liu, Z., Germain, R. N., Huse, M., Altan-Bonnet, G., and Krichevsky, O. (2017) A Tunable Diffusion-Consumption Mechanism of Cytokine Propagation Enables Plasticity in Cell-to-Cell Communication in the Immune System. *Immunity 46*, 609–620.

(18) Müller, P., and Schier, A. F. (2011) Extracellular Movement of Signaling Molecules. *Developmental Cell 21*, 145–158.

(19) Shimozono, S., Iimura, T., Kitaguchi, T., Higashijima, S., and Miyawaki, A. (2013) Visualization of an endogenous retinoic acid gradient across embryonic development. *Nature 496*, 363–366.

(20) Altan-Bonnet, G., and Mukherjee, R. (2019) Cytokine-mediated communication: a quantitative appraisal of immune complexity. *Nat Rev Immunol 19*, 205–217.

(21) Ashe, H. L., and Briscoe, J. (2006) The interpretation of morphogen gradients. *Development 133*, 385–394.

(22) Francis, K., and Palsson, B. O. (1997) Effective intercellular communication distances are determined by the relative time constants for cyto/chemokine secretion and diffusion. *Proceedings of the National Academy of Sciences 94*, 12258–12262.

(23) Green, J. B. A., and Sharpe, J. (2015) Positional information and reaction-diffusion: two big ideas in developmental biology combine. *Development 142*, 1203–1211.

(24) Briscoe, J., and Small, S. (2015) Morphogen rules: design principles of gradient-mediated embryo patterning. *Development 142*, 3996–4009.

(25) Gidon, A., Zolnik, T. A., Fidzinski, P., Bolduan, F., Papoutsi, A., Poirazi, P., Holtkamp, M., Vida, I., and Larkum, M. E. (2020) Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science 367*, 83–87.

(26) Altan-Bonnet, G., Mora, T., and Walczak, A. M. (2020) Quantitative immunology for physicists. *Physics Reports 849*, 1–83.

(27) Parrilla-Gutierrez, J. M., Sharma, A., Tsuda, S., Cooper, G. J. T., Aragon-Camarasa, G., Donkers, K., and Cronin, L. (2020) A programmable chemical computer with memory and pattern recognition. *Nat Commun 11*, 1442.

(28) Taylor, A. F., Tinsley, M. R., Wang, F., Huang, Z., and Showalter, K. (2009) Dynamical Quorum Sensing and Synchronization in Large Populations of Chemical Oscillators. *Science 323*, 614–617.

(29) Adamatzky, A., and De Lacy Costello, B. (2002) Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond. *Phys. Rev. E 66*, 046112.

(30) Tóth, Á., and Showalter, K. (1995) Logic gates in excitable media. *The Journal of Chemical Physics 103*, 2058–2066.

(31) Tompkins, N., Li, N., Girabawe, C., Heymann, M., Ermentrout, G. B., Epstein, I. R., and Fraden, S. (2014) Testing Turing’s theory of morphogenesis in chemical cells. *Proceedings of the National Academy of Sciences 111*, 4397–4402.

(32) Zhang, D. Y. (2011) Cooperative Hybridization of Oligonucleotides. *J. Am. Chem. Soc. 133*, 1077–1086.

(33) Papin, J. A., Hunter, T., Palsson, B. O., and Subramaniam, S. (2005) Reconstruction of cellular signalling networks and analysis of their properties. *Nat Rev Mol Cell Biol 6*, 99–111.

(34) Chen, Y.-J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., and Seelig, G. (2013) Programmable chemical controllers made from DNA. *Nature Nanotech 8*, 755–762.

(35) Seelig, G., Soloveichik, D., Zhang, D. Y., and Winfree, E. (2006) Enzyme-Free Nucleic Acid Logic Circuits. *Science 314*, 1585–1588.

(36) Qian, L., Winfree, E., and Bruck, J. (2011) Neural network computation with DNA strand displacement cascades. *Nature 475*, 368–372.

(37) Qian, L., and Winfree, E. (2011) Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science 332*, 1196–1201.

(38) Freeman, M. (2000) Feedback control of intercellular signalling in development. *Nature 408*, 313–319.

(39) Padirac, A., Fujii, T., and Rondelez, Y. (2012) Bottom-up construction of in vitro switchable memories. *Proceedings of the National Academy of Sciences 109*, E3212–E3220.

(40) Berg, H. C., and Purcell, E. M. (1977) Physics of chemoreception. *Biophysical Journal 20*, 193–219.

(41) Zadeh, J. N., Steenberg, C. D., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., and Pierce, N. A. (2011) NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem. 32*, 170–173.

(42) Unger, M. A. (2000) Monolithic Microfabricated Valves and Pumps by Multilayer Soft Lithography. *Science 288*, 113–116.

(43) Crank, J. (2009) The mathematics of diffusion 2. ed., reprinted. Oxford Univ. Press, Oxford.

(44) Yang, S., Pieters, P. A., Joesaar, A., Bögels, B. W. A., Brouwers, R., Myrgorodska, I., Mann, S., and de Greef, T. F. A. (2020) Light-Activated Signaling in DNA-Encoded Sender–Receiver Architectures. *ACS Nano 14*, 15992–16002.

7

Epilogue

7.1 Introduction

Synthetic biology has proven to be a valuable tool to gain an understanding of the intricate interactions in gene regulatory pathways of cells. In particular, the reconstruction of regulatory circuits in minimalistic model systems has enabled novel approaches to characterize complex networks of interacting genes. By taking a well-defined and highly tunable environment to implement synthetic circuits, environment and circuit parameters can be sampled systematically, without influencing unrelated pathways. The bottom-up construction of synthetic genetic circuits based on recurring motifs in regulatory networks of cells can provide insight into their corresponding behavior and the parameter regimes required to implement the circuits in novel contexts. Additionally, the field of bottom-up synthetic biology is concerned with the construction of synthetic cells that exhibit life-like features. Hereto, complex networks of interacting regulatory genes need to be implemented and compartmentalized to form a coherent unit (protocell), containing genetic information and circuitry. Another layer of complexity is achieved through communication between protocells, which mimics the coordination between cells that is observed in multicellular organisms and populations of single-cell organisms.

This thesis explored the use of *in vitro* transcription and translation (TXTL) and DNA and RNA nanotechnology to construct synthetic circuits. We demonstrated that the complexity of the behavior of the circuits could be increased through the construction of higher-order organizations of simple circuit motifs and the implementation of spatiotemporal organizations. This epilogue will give an overview of the critical findings of the work described in this thesis and outline future directions.

7.2 RNA Interactions in TXTL

In Chapter 2, the production of TXTL reagents suitable for RNA-based gene regulation was described, and simple genetic circuits using toehold-mediated RNA strand displacement were implemented. The subsequent development of larger circuits based on the same regulatory elements in Chapters 4 and 5 demonstrates the flexibility and scalability of RNA-based synthetic genetic circuits. However, due to the abundance of knowledge on transcription factors, repressor proteins, and other transcriptional regulators, most previously developed synthetic circuits have largely omitted post-transcriptional RNA-RNA interactions. Nevertheless, post-transcriptional regulation is featured prominently in various forms in prokaryotes (e.g., sRNA) and eukaryotes (e.g., miRNA). Therefore, to achieve the implementation circuitry of a complexity that approaches living systems, it is essential for the field of synthetic biology to embrace this additional layer of regulation. The RNA interactions in TXTL described here provide a starting point, but may not capture other common forms of RNA-based post-transcriptional regulation. Protein-mediated RNA-RNA binding (e.g., by the Hfq protein in bacteria[1] and Argonaute proteins in eukaryotes) and RNA-protein interactions (by RNA-binding proteins) are not captured yet in our TXTL system.

7.3 Temporal Signals in Flow Reactors

The semi-continuous flow reactors we descibed in Chapter 3 have been demonstrated to maintain TXTL reactions for prolonged periods.[2] However, we found that our TXTL system quickly lost activity after more than ~10 hours of continuous gene expression. Interestingly, the TXTL reagents were shown to retain their activity for at least 30 hours, leaving the cause of the drop in expression yield open for further investigation. For our current application, ~10 hours of constant expression rates provided a sufficient window to observe the response of circuits to varying inputs. Nevertheless, longer reaction times would enable the measurement of the frequency response of synthetic circuits or advanced experiments where multiple reaction conditions are analyzed consecutively in a single reactor. Combined with continuous computational model selection[3] and parameter identifiability analysis[4], this setup could perform fully automated model selection and parameter estimation for novel synthetic circuits.

Using our custom device control software, we were able to introduce an almost instantaneous change in concentration at the start of the square pulse, but the reactor refresh rates dictated the reverse step. Perhaps, the off-step of the square pulse could be made more direct by introducing selective RNA degradation[5] or obfuscation by antisense RNA,[6] at the cost of increased experimental complexity. Using our customizable protocols, a more diverse range of signal shapes could be generated, and we envision that the computer-aided identification of circuit-defining inputs can be utilized to optimally characterize complex synthetic genetic circuits.

7.4 Feed-Forward Loop-Based Circuits

In Chapters 4 and 5, we focused on constructing and characterizing synthetic circuits based on the coherent feed-forward loop (CFFL), a network motif abundantly present in the gene regulatory circuits of various organisms. Remarkably, we could not identify the dynamic noise-filtering behavior that was speculated to contribute to the prominence of CFFLs in cell regulatory networks. Gerardin et al.[7] similarly observed noise-filtering behavior could only be observed in CFFLs for limited circuit parameters. As a result, it remains unclear how closely intertwined the function and topology of genetic networks are.[8,9] To further elucidate this relation for CFFLs, further research should focus on the determination of the parameter regimes of naturally occurring CFFLs and alternative evolutionary preferences for the topology.

In contrast, repression of background activity of the synthetic CFFL circuit was observed, which could be beneficial in biological regulatory networks or the design of synthetic genetic circuits. We envision that the CFFL could convey this benefit to existing genetic circuits by integrating the topologies and compensate for leaky expression of circuit components. However, when we combined two CFFLs into a composite CFFL circuit, the background repression behavior was not retained. Likely, the shared use of an intermediate node between the two loops in the composite circuit caused the loss of background repression, and connecting components to the output node of the CFFL will not result in retroactive effects.

7.5 Spatiotemporal Distribution of Circuits

Finally, in Chapter 6, we looked at the spatiotemporal integration of simple circuits by exchanging a diffusive signaling molecule. Ideally, the sender-receiver architectures described in Chapter 6 are extended to protocells capable of sustaining TXTL reactions and facilitating the use of the complete range of transcriptional and translational interactions present in natural cells. This setup would enable the study of distributed circuits with more complex dynamic behavior than the enzyme-free DNA strand-displacement reactions. Communication between encapsulated TXTL reactions has been demonstrated,[10,11] but require physical contact between the TXTL-containing droplets or are limited to a single communication channel.[12] Perhaps, to maintain the programmability and flexibility of nucleic acid-based diffusive signals but exploit the full capabilities of TXTL reactions, RNA could be used as a signaling modality between TXTL-based protocells. Selective export of signal RNA strands and retention of mRNA used in intraprotocellular circuits could then be achieved using mimics of RNA-containing exosomes[13] or the anchoring of circuit mRNA by RNA-binding proteins, similar to how streptavidin was used to retain the DNA circuitry inside proteinosomes.

7.6 References

(1) Wagner, E. G. H., and Romby, P. (2015) Small RNAs in Bacteria and Archaea, in *Advances in Genetics*, pp 133–208. Elsevier.

(2) Niederholtmeyer, H., Stepanova, V., and Maerkl, S. J. (2013) Implementation of cell-free biological networks at steady state. *PNAS 110*, 15985–15990.

(3) Ghosh, J. K., and Samanta, T. (2001) Model selection – An overview. *Current Science 80*, 1135–1144.

(4) Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. (2009) Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics 25*, 1923–1929.

(5) Zhang, Y., Zhang, J., Hoeflich, K. P., Ikura, M., Qing, G., and Inouye, M. (2003) MazF Cleaves Cellular mRNAs Specifically at ACA to Block Protein Synthesis in Escherichia coli. *Molecular Cell 12*, 913–923.

(6) Green, A. A., Kim, J., Ma, D., Silver, P. A., Collins, J. J., and Yin, P. (2017) Complex cellular logic computation using ribocomputing devices. *Nature 548*, 117–121.

(7) Gerardin, J., Reddy, N. R., and Lim, W. A. (2019) The Design Principles of Biochemical Timers: Circuits that Discriminate between Transient and Sustained Stimulation. *Cell Systems 9*, 297-308.e2.

(8) Ahnert, S. E., and Fink, T. M. A. (2016) Form and function in gene regulatory networks: the structure of network motifs determines fundamental properties of their dynamical state space. *J. R. Soc. Interface 13*, 20160179.

(9) Payne, J. L., and Wagner, A. (2015) Function does not follow form in gene regulatory circuits. *Sci Rep 5*, 13015.

(10) Booth, M. J., Schild, V. R., Graham, A. D., Olof, S. N., and Bayley, H. (2016) Light-activated communication in synthetic tissues. *Science Advances 2*, e1600056.

(11) Findlay, H. E., Harris, N. J., and Booth, P. J. (2016) In vitro synthesis of a Major Facilitator Transporter for specific active transport across Droplet Interface Bilayers. *Scientific Reports 6*, 39349.

(12) Tang, T.-Y. D., Cecchi, D., Fracasso, G., Accardi, D., Coutable-Pennarun, A., Mansy, S. S., Perriman, A. W., Anderson, J. L. R., and Mann, S. (2018) Gene-Mediated Chemical Communication in Synthetic Protocell Communities. *ACS Synth. Biol. 7*, 339–346.

(13) Gill, S., Catchpole, R., and Forterre, P. (2019) Extracellular membrane vesicles in the three domains of life and beyond. *FEMS Microbiology Reviews 43*, 273–303.

Summary

This thesis describes the construction and characterization of complex synthetic circuitry using cell-free micro-compartmentalized reactions. In **Chapter 1**, we review the implementation of cell-free synthetic circuits that mimic natural regulatory circuits and discuss recent developments in the microcompartmentalization of cell-free transcription and translation (TXTL) reactions. We identify that the incorporation of RNA-based regulation in TXTL reaction could extend the toolbox of components available to construct synthetic genetic circuits. Therefore, in **Chapter 2**, we produce TXTL reagents suitable for RNA-based gene regulation and demonstrate post-transcriptional regulation in TXTL reactions using synthetic riboswitch activated by toehold-mediated RNA strand-displacement. The functionality of this mode of regulation is further established by utilizing it to construct a gene cascade and implement an RNA-based NOT logic gate.

The characteristic response of cell-free synthetic circuits to time-varying signals not been studied extensively, since many genetic circuits were developed to have well-defined steady-state behavior, such as oscillations. To construct a platform to analyze the response of genetic circuits to various input signals, we employ microfluidic semi-continuous flow reactors to compartmentalize TXTL reactions, which is described in **Chapter 3**. Using custom control software, various time-varying input signals are generated, and prolonged out-of-equilibrium TXTL reactions are performed. We achieve more than 10 hours of constant gene expression rates in the flow reactors, which provides a sufficient window to observe the response of circuits to varying inputs.

In **Chapter 4**, we construct a synthetic implementation of a coherent feed-forward loop (CFFL) motif, which is known to mainly propagate long-lived input signals and effectively filter out short, noisy inputs. To study its dynamic behavior, we supplement extensive characterization of the circuit in batch TXTL reactions with temporal characterization using microfluidic flow reactors. Our work reveals that the synthetic CFFL does not display dynamic noise-filtering, indicating that this behavior may not be a robust property of CFFLs. Conversely, we demonstrate that the synthetic circuit can repress background activity, a property beneficial for generating distinct ON and OFF states of a circuit.

Subsequently, we explore the construction of larger circuit architectures based on CFFLs in **Chapter 5**. We combine two CFFLs into a composite CFFL circuit and evaluate all circuit interactions in batch TXTL reactions. This work displays the flexibility of the synthetic CFFL design, which facilitates the straightforward implementation of a topologically complex circuit. Interestingly, we reveal that the background repression observed in a single CFFL is not retained in the composite circuit.

Chapter 6 focuses on forming higher-order spatiotemporal organizations by implementing synthetic circuits compartmentalized in protocells. We employ the previously developed BIO-PC platform, which utilizes DNA strand-displacement reactions encapsulated in crosslinked protein-polymer conjugates (proteinosomes) to analyze populations of communicating protocells. Using light-induced release of signals, we reveal the crucial determinants of the signaling length scale of protocell-based sender-receiver architectures. Furthermore, we demonstrate spatiotemporal control over the activation of a distributed AND logic gate using two distinct sender protocells.

In conclusion, this thesis explores the cell-free characterization of microcompartmentalized synthetic circuits, with an emphasis on a synthetic CFFL circuit. We utilize minimalistic model systems to reveal fundamental characteristics of synthetic circuits, and demonstrate the potential of combining DNA and RNA nanotechnology with microcompartmentalized cell-free reactions.

Curriculum Vitae

Pascal Aldo Pieters was born on March 12, 1993, in Leeuwarden, the Netherlands. He finished secondary education in 2011 at the Christelijk Gymnasium Beyers Naudé in Leeuwarden, the Netherlands, receiving the Jan Kommandeur prize for best secondary education research paper in the field of nature, science, and medicine. Subsequently, he moved to Eindhoven, the Netherlands, to study Biomedial Engineering at the Eindhoven University of Technology. In 2014 he completed his Bachelor of Science within the Computational Biology group, and a minor in Embedded Systems, *cum laude.* During his studies, Pascal was team captain of a team competing in the 2013 International Genetically Engineered Machine (iGEM) competition and completed the Eindhoven University of Technology Honors program. He continued his education in Eindhoven with a Master's degree under supervision of prof.dr.ir. Tom F.A. de Greef and prof.dr.ir. Luc Brunsveld, during which he used his Amandus H. Lundqvist Scholarship to performed an internship in the group of prof. Peng Yin at the Wyss Institute at Harvard University, Massachusetts, USA. In 2016 he graduated *cum laude* within the Chemical Biology group on the implementation of cell-free genetic circuits and completed the Institute for Complex Molecular Systems Graduate Program. In 2017 he started a PhD project at the Institute for Complex Molecular Systems of the Eindhoven University of Technology under supervision of prof.dr.ir. Tom F.A. de Greef. He focused on combining cell-free transcription and translation methods with microfluidic techniques and computational methods to characterize complex synthetic genetic circuits.

List of Publications

1. **P.A. Pieters**, B.L. Nathalia, A.J. van der Linden, J. Kim, P. Yin, W.T.S. Huck and T.F.A. de Greef. *Cell-Free Characterization of Coherent Feed-Forward Loop-Based Synthetic Genetic Circuits. ACS Synthetic Biology, 6, 1406-1416 (2021).*
2. S. Yang, **P.A. Pieters**, A. Joesaar, B.W. A. Bögels, R. Brouwers, I. Myrgorodska, S. Mann, and T.F.A. de Greef. Light-activated signaling in DNA-encoded sender-receiver architectures. *ACS Nano*, **14**, 15992-16002 (2020).
3. A.J. van der Linden, **P.A. Pieters**, M. Yelleswarapu, Z. Swank, W.T.S. Huck, S. Maerkl, and T.F.A. de Greef. A multilayer microfluidic platform for the conduction of prolonged cell-free gene expression. *JoVE*, **152**, e59655 (2019).
4. E. Dubuc, **P.A. Pieters**, A.J. van der Linden, J.C.M. van Hest, W.T.S. Huck, and T.F.A. de Greef. Cell-free microcompartmentalised transcription-translation for the prototyping of synthetic communication networks. *Current Opinion in Biotechnology*. **58**, 72-80 (2019).
5. A.H. Joesaar, S. Yang, B.W.A. Bögels, A.J. van der Linden, **P.A. Pieters**, P. Kumar, N. Dachau, A. Phillips, S. Mann, and T.F.A. de Greef. DNA-based communication in populations of synthetic protocells. *Nature Nanotechnology*. **14**, 369-378 (2019).
6. M. Yelleswarapu, A.J. van der Linden, **P.A. Pieters**, E. Dubuc, T.F.A. de Greef, and W.T.S. Huck. Sigma factor-mediated tuning of bacterial cell-free synthetic genetic oscillators. *ACS Synthetic Biology.* **7**, 2879-288 (2018).
7. M. van Rosmalen, B.M.G. Janssen, N. Hendrikse, A.J. van der Linden, **P.A. Pieters**, D. Wanders, T.F.A. de Greef & M. Merkx. Affinity maturation of a cyclic peptide handle for therapeutic antibodies using deep mutational scanning. *Journal of Biological Chemistry* **292**, 1477–1489 (2017).

8. D. Van der Zwaag, **P.A. Pieters**, P.A. Korevaar, A.J. Markvoort, A.J.H. Spiering, T.F.A. de Greef, and E.W. Meijer. Kinetic Analysis as a Tool to Distinguish Pathway Complexity in Molecular Assembly. *JACS*, **137**, 12677–12688 (2015).
9. S.J.A. Aper, A.C.C. van Spreeuwel, M.C. van Turnhout, A.J. van der Linden, **P.A. Pieters**, N.L.L. van der Zon, S.L. de la Rambelje, C.V.C. Bouten, M. Merkx. Colorful Protein-Based Fluorescent Probes for Collagen Imaging. *PLoS ONE*, **9**, e114983 (2014).

Acknowledgments Dankwoord

De weg naar het voltooien van deze thesis begon voor mij jaren geleden toen ik vanuit Friesland naar Eindhoven verhuisde om Biomedische Technologie te studeren. Sindsdien hebben veel mensen mij geholpen, gesteund en gemotiveerd in mijn academische projecten en daaromheen. In dit hoofdstuk wil ik graag al deze mensen persoonlijk bedanken.

Studerend aan de TU/e was ik al snel aangetrokken tot het Synthetische Biologie studieveld en besloot mee te doen aan de iGEM competitie. Rond deze tijd ontmoette ik voor het eerst mijn latere promotor. Tom, jouw enthousiasme en brede interesse heeft mij altijd gemotiveerd. Je bent door de jaren heen een mentor voor me geweest met veel vertrouwen in mij en hebt mijn inbreng altijd zeer gewaardeerd. Ik ben zeer dankbaar voor je begeleiding door de jaren heen en alles wat je me in die tijd hebt geleerd.

De basis voor het vele labwerk heb ik te danken aan Maarten en Stijn. Maarten, bedankt voor je betrokkenheid bij het biolab, de verscheidene projecten die ik in jouw groep heb kunnen doen en tevens je colleges, die ik, samen met die van Luc, als interessantste van mijn hele studie heb ervaren. Stijn, het solide fundament dat jij hebt gelegd voor mijn labervaring heeft mij in latere jaren nog veel profijt opgeleverd. Ik heb leuke tijden met je gehad in het lab en een veel te duur hotel in Boston. Daan, Martijn en Sam, jullie wil ik ook bedanken voor de vele aspecten van onderzoek die ik van jullie heb geleerd tijdens projecten.

During the summer of 2015, I completed an internship in the group of Peng Yin at the Wyss Institute in Boston. The experience with toehold switches and DNA and RNA nanotechnology I acquired there has proven exteremely useful in the completion of my thesis. I would like to thank Peng Yin for the amazing environment he created in his group with extremely motivated people. Jongmin, you were a wonderful supervisor for the internship and I would furthermore like to thank you for your help with the completion of my thesis project and concomitant research paper. I wish you all the best with the next steps in your academic career in Korea! Anita, Cameron, Hiroshi, and the other lab member, thank you, I had a great time working with you!

Furthermore, I would like to thank all members of my defense committee for their enthusiasm in accepting this role and the subsequent time and effort put in to review this dissertation and be part of the defense ceremony.

Peter, bedankt voor je geduld met mij door de jaren heen. Het is altijd fijn geweest om het onderzoek in jouw groep te presenteren, aangezien jouw commentaar altijd erg nuttig was en je vaak net wat anders tegen zaken aankeek. Natal, als autoriteit op het gebied van data fitting is je commentaar altijd zeer gewaardeerd!

Oscar, Thomas, Enrico, Anne and Dora, I look forward to meeting you at the defense ceremony!

Ik moet hier natuurlijk ook nog iemand bedanken waarmee ik voor sommigen op de universiteit onlosmakelijk verbonden ben en wiens naam ik verscheidene keren genoemd ben. Ardjan, partner in crime vanaf de bachelor. We zijn behoorlijk verschillend, de een houdt van pinguïns, de ander niet, maar juist daarom hebben we altijd veel lol gehad. We hebben samen honours gedaan, op een of andere manier een lastige minor voltooid tegelijkertijd met zijn van team captains voor iGEM, we waren meubilair in het Ceres gebouw en promoveren nu op vergelijkbare onderwerpen. Misschien is het goed dat onze professionele wegen eindelijk scheiden, zolang we maar samen een erg onsuccesvol AIVD kerstpuzzel team blijven vormen. Natuurlijk heb ik je wetenschappelijke inbreng tijdens het PhD project ook zeer gewaardeerd en je vaardigheden met betrekking tot microfluidics zijn erg nuttig geweest voor mij, net als voor vele anderen. Bedankt paranimf!

Alex, dr. Joesaar, as fellow member of the early de Greef Ceres crew, I’ve had a lot of office fun with you! Additionally, you have been extremely helpful in debugging experiments and brainstorming ideas, as well as solving crossword puzzles. Joining somewhat later, Emilien, you have been a great addition to the group. I have thoroughly enjoyed working with you and visiting you in Nijmegen! Thanks for all the help setting up the cell-free expression system.

Shuo, working with you has been a delight. Your work has been instrumental in the construction of Chapter 6 of this dissertation. Thank you for everything and I hope you, your wife, and the little one find an exciting next step in your careers!

Glenn, ook al zat je niet in het kantoor in Ceres, je bent wel altijd een essentieel onderdeel van de groep geweest. Tevens was ons kantoor een tweede thuis voor jou, of in ieder geval je golf clubs. Je wetenschappelijke vaardigheden zijn benijdbaar en projecten en resultaten met jou bespreken was altijd uitermate nuttig. Bedankt voor het regelen van een leuke videoboodschap voor mijn bruiloft!

Job, ook jij huisde niet in het Ceres kantoor, maar hoorde wel bij de groep. Samen met jou en Bart heb ik het altijd erg leuk gevonden om tutor te zijn bij Programming en Genomics!

Bas, ik heb jouw open houding en bijdrage aan de goede sfeer in de groep altijd zeer geapprecieerd. Je kwam in een kantoor terecht bij een paar eigenzinnige oudjes, maar hebt altijd goed stand gehouden en laat niet over je heen lopen. Ik heb er daarom ook alle vertrouwen in dat je je PhD project zeer succesvol zult afronden!

Anna, you had an even tougher job, joining an office of solely guys. This turned out to be no issue for you at all. I have had a great time with you the past years! I really admire your work ethic too, working with those mammalian cells can be pain. It's a shame the pandemic put a hold on your big fat Greek wedding. Being a παράνυμφος at a defense is just not the same. Hopefully you can celebrate your wedding soon!

Bryan, als master student heb je uitstekend werk geleverd, zoals je kunt zien aan de paper en Hoofdstuk 5 van deze thesis. We hebben altijd veel lol gehad op het lab en ik hoop dat ik een goede bijdrage heb kunnen leveren aan jouw enthousiasme voor de wetenschap!

The work in this thesis benefited greatly from close collaboration with the Huck group in Nijmegen. Wilhelm, your feedback on the initial drafts of the feed-forward loop paper was crucial to improving its quality. Maaruthy and Roel, you are one of the only people who truly understand the struggles of doing the flow experiments. Thank you for always sharing your findings and helping each other improve the experimental setup. Additionally, together with the rest of the Huck group, it has been great partying with you during conferences.

Wanneer ik aan het ICMS denk, waar ik al deze jaren heb gewerkt, denk ik nog steeds als eerst aan jou, Sagitta. Bedankt voor al je hulp, je uitstekende leiderschap en gezelligheid. Monique, Jolanda, Cindy, Wendy en Carla, bedankt voor al het werk dat jullie hebben verzet om van Ceres een uitmuntend instituut te maken! Koen, bedankt voor je begrip voor het draaien van Linux op m'n computer, die instelling hoort thuis op een technische universiteit!

In addition to the Ceres building, the Helix Biolab has played an essential role in my PhD dissertation. This lab is extremely well organized (even though we like to pretend it isn't) and the atmosphere is great! Peggy, you are one of the main contributors to the success of this lab. Thank you for everything, all the help you provided me starting as second year bachelor student.

I would like to thank all (former) members of the Biolab for the friendly environment they created. Stijn, Remco, Martijn, Anniek, the friday-curry-gang: Eline, Eva, Sebastian, Amy, Alex, Wiggert, Bastiaan, Jurgen, Sam, and many more, thank you all!

Tijdens mijn studietijd in Eindhoven is een vriendengroep gevormd waar ik nog altijd veel plezier mee heb. Genaamd 'de Fustjes', hebben we goed van het studentenleven genoten. Moniek, Meike, Denise, Rick, Stan, Stefan, Ilona, Rowin, Rens, Jacques en Tim, bedankt voor alle lol tijdens, borrels, feestjes, weekendjes weg, et cetera. Moniek, het is fijn geweest om af en toe de PhD struggles met jou te kunnen bespreken. Wij kennen elkaar al het langst, wat betekend dat jij waarschijnlijk de meeste beschamende foto's van mij hebt. Vrienden voor het leven dus.

Bernard, vrienden sinds de brugklas, best man bij mijn bruiloft. Wat wij hebben uitgevreten zal ik hier niet beschrijven, maar ik wil je vooral bedanken voor dat je er altijd voor me bent en we altijd weer veel plezier kunnen beleven, ook al hebben we elkaar een tijdje niet gezien. Ik wens jou ook veel succes bij het afronden van je PhD project. De online Portal spelletjes en later Valheim zijn altijd mijn broodnodige afleiding geweest. Bedankt voor al die jaren vriendschap!

Priscilla, als jonger zusje moest je je soms een beetje uit mijn schaduw knokken. Dat je je inmiddels fantastisch gelukt, ik ben super trots op je en wat je allemaal doet! Ik ben super blij dat je weer naar Nederland kunt komen om bij mijn promotie aanwezig te zijn. Nog dankbaarder ben ik dat we sinds we beide uit huis zijn gegaan nog steeds goed contact hebben. Een betere zus kan ik me niet wensen!

Mama, papa, jullie wil ik ervoor bedanken dat jullie er altijd voor me zijn geweest, ook al liet ik niet altijd even vaak van me horen uit Eindhoven. Jullie hebben van jongs af aan in me geloofd en me gemotiveerd om te gaan doen waar ik passie voor had. Het is ook altijd weer fijn om een weekendje thuis te komen in de Westereen. Ook mijn schoonfamilie, in het bijzonder Lennard en Petra, bedankt voor alle support!

En natuurlijk als belangrijkste, Saskia. Sinds kort mijn vrouw, als een waar hoogtepunt in de corona periode. Je bent de liefde van mijn leven en bent er altijd voor me. Het laatste deel van de PhD ben ik soms wat gestrest geweest en heb ik het je niet altijd makkelijk gemaakt, maar je hebt me altijd gesteund. Bedankt schat! We hebben samen al veel beleefd en zijn op veel bijzondere plekken geweest. Ik kan niet wachten om meer avonturen met jou mee te maken! Ik hou van je.

~ Pascal