PENSim: A Toolkit for PEN-DNA Systems

Gwendal Ducloz
École Normale Supérieure de Lyon, LIP (UMR 5668, Équipe MC2), France

Nicolas Schabanel
CNRS, École Normale Supérieure de Lyon, LIP (UMR 5668, Équipe MC2) and IXXI, France

Abstract

The PEN-DNA toolbox has emerged as a versatile framework for implementing chemical reaction networks with DNA and enzymes, enabling applications ranging from molecular sensing to neural computation. A simulation tool, DACCAD, has provided a standardized approach for translating network schematics into dynamical models, facilitating *in silico* design and validation. However, the evolution of the PEN-DNA toolbox – particularly the transition from inhibitor-based regulation to drain-template-based inactivation – introduces mechanisms that are only partially captured by existing simulation frameworks, limiting the ability to reliably simulate and design recent PEN-DNA systems.

In this context, we introduce PENSim, a simulation package that aims to realign computational modeling with current experimental practices. By extending the repertoire of simulated reactions and incorporating thermodynamic dependencies into kinetic rate calculations via NUPACK, PENSim enables a more faithful representation of modern PEN-DNA systems. In particular, accounting for thermodynamic effects is essential for accurately modeling inactivation via drain templates.

PENSim is validated through qualitative reproduction of existing experimental results, including inactivation-driven bistability, microRNA detection circuits, and linear classifiers based on neural networks. By bridging legacy modeling approaches with recent experimental advances, PENSim provides a step toward more realistic and flexible *in silico* design of systems based on the PEN-DNA toolbox.

2012 ACM Subject Classification Computer systems organization → Molecular computing; Computing methodologies → Molecular simulation; Applied computing → Chemistry

Keywords and phrases PEN-DNA toolbox, DNA circuit simulation, DNA reaction kinetics, Enzymatic DNA reaction kinetics

Digital Object Identifier 10.4230/LIPIcs.DNA.32.4

Supplementary Material *Software (Source Code)*: `https://github.com/GwendalDucloz/PENSim` archived at `swh:1:dir:ca74bb35d34748ecdb887f1601f05221aabd8233`

Funding This work was supported by the French government under the "France 2030" programme ANR-24-RRII-0001 and ANR-22-CE45-0020-01.

Acknowledgements We thank Guillaume Ginès, Antoine Anastassiades, Quentin Mosagna, Alexis Autour and Nathanaël Aubert-Kato for stimulating discussions.

1 Introduction

Nucleic acid strands – both DNA and RNA – are harnessed to perform a wide range of tasks, spanning chemical [11, 32, 40], mechanical [7, 34], biomedical [12, 18, 27], and computational [5, 21, 36] applications. In particular, their ability to form base pairs enables the design of 2D or 3D nanostructures, which in turn confer precise functional behaviors.

In the context of computing, several paradigms have emerged. Some rely exclusively on DNA, as for instance, tile and brick assembly models [9, 36], which perform computation by dynamically expanding a DNA-based network. Alternatively, another DNA-only approach relies on chemical reaction networks, where computation is achieved through toehold-mediated strand displacement [5, 13].

In recent years, a novel paradigm has emerged: the *PEN-DNA toolbox* [4, 10, 20], which integrates DNA with enzymatic processes to implement chemical reaction networks. Specifically, this framework exploits three types of enzymes, forming the so-called *Polymerase-Exonuclease-Nickase Dynamic Network Assembly (PEN-DNA)* toolbox (see Section 2 for details). This framework has been successfully applied to engineer biochemical oscillators [20], model predator-prey dynamics [10], and, more recently, construct small neural networks [21]. The ability to implement neural networks is a proof of concept that the PEN-DNA toolbox is functionally complete and may be capable of Turing-complete computations under idealized conditions (e.g. unbounded precision and infinite memory). Beyond computational applications, the PEN-DNA toolbox also shows applications in biomedical contexts, such as the detection of microRNAs, enzymes, and other small molecules [11, 12, 18, 32].

When working with any DNA model, *in silico* simulation is a critical first step to assess computational systems before advancing to *in vitro* validation. Therefore, each paradigm is supported by a dedicated simulation framework: kTAM for tile assembly models [35], systems based on ordinary differential equations or continuous-time Markov chains for toehold-mediated strand displacement systems [15, 39], and DACCAD for the PEN-DNA toolbox [2, 3]. These frameworks not only enable the prediction and optimization of nucleic acid-based systems but also bridge the gap between theoretical design and experimental implementation. Developing and refining such tools is therefore essential, as it allows simulations to increasingly reflect biochemical reality and better guide experimental outcomes.

Contribution

This work presents PENSim, a new toolkit for the PEN-DNA toolbox that addresses limitations of the existing DACCAD framework. PENSim enhances the simulation capabilities for systems utilizing the latest PEN-DNA toolbox features with two main contributions.

First, it expands the scope of simulable reactions by implementing mechanisms introduced in recent versions of the PEN-DNA toolbox, including reporter templates, EvaGreen fluorescence, inactivation via drain templates, irreversible conversion, and support for multiple versions of the same signal strand [19, 21, 29]. These mechanisms are central to modern PEN-DNA systems – such as neural networks and microRNA sensing – but are not supported in available versions of DACCAD, thereby limiting the simulation and design of such systems. Furthermore, PENSim includes experimental code modules supporting additional reactions, such as those involving non-phosphorylated and non-protected templates, reflecting a design choice toward extensibility and future integration of emerging mechanisms. Importantly, PENSim is designed to complement rather than replace DACCAD, as it does not implement inhibitor-based regulation from earlier versions of the toolbox, which is no longer used in recent studies.

Second, PENSim improves the fidelity of simulations by introducing a *thermodynamically grounded approach to kinetic rate computation*. Rather than relying on uniform or user-defined kinetic parameters, the simulator derives reaction rates directly from biochemical properties using NUPACK and a kinetics nearest neighbor model [26, 38]; including sequence, temperature, enzyme, and buffer conditions. This enables a more accurate representation of DNA hybridization and enzymatic processes, which is critical for capturing behaviors driven by inactivation mechanisms and for modeling complex systems such as enzymatic neural networks [19, 21].

By enabling simulation from DNA sequences and experimental parameters, PENSim provides a practical tool for experimentalists designing new PEN-DNA circuits. The simulator, along with validation notebooks and usage examples, is openly accessible in the accompanying

Git repository (`https://github.com/GwendalDucloz/PENSim`). It is validated through the qualitative reproduction of experimentally observed behaviors of previously implemented systems spanning dynamical circuits, molecular sensing, and neural computation [19, 21, 29].

Section 2 introduces the mechanisms of the PEN-DNA toolbox and their representation as a graph-like schematic. DACCAD, and related work on simulating PEN-DNA toolbox systems are reviewed in Section 3. Our contribution begins in Section 4, where we detail the dynamics of key subreactions – DNA hybridization and enzymatic reactions – and present our method for computing their kinetics. Section 5 outlines the code architecture, provides usage tutorials, and describes the species considered (DNA single strands and complexes), along with the construction of their governing equations. Validation of the simulator is presented in Section 6. An appendix includes additional features of the PEN-DNA toolbox, supplementary figures, and an overview of the set of equations solved by PENSim.

2 The PEN-DNA toolbox

A system built using the PEN-DNA toolbox, called *PEN-DNA system*, is composed of DNA strands and three types of enzymes: Polymerase, Exonuclease, and Nickase(s). All these components interact simultaneously through various reactions that produce, transform, and degrade DNA strands. Generally, PEN-DNA systems are designed so that the concentrations of all DNA species either reach a periodic behavior [10, 20] or equilibrium [19, 21].

2.1 Reactions of the PEN-DNA toolbox

Five types of mechanisms are designed within the PEN-DNA toolbox, described in Figure 1. They are the *conversion*, the *reporting*, the *inactivation*, the *degradation*, and the *killing conversion*. Each reaction, except degradation, needs a specific DNA strand, called *template*.

Conversion

A **converter template (cT)** enables the production of a new DNA strand β, called the *output strand* or *output signal*, from an initial DNA strand α, called *input strand* or *input signal*. The conversion $\alpha \xrightarrow{\text{cT}\alpha\beta} \alpha + \beta$, happens as illustrated in Figure 1:

① The strand α binds to the converter template cT$\alpha\beta$.
② The polymerase binds to the resulting complex and extends α according to the complementary sequence of the template cT$\alpha\beta$.
③ The nickase binds to the complex on its recognition sequence and cleaves the extended strand at a specific position, separating α from the newly synthesized part β.
④ At the working temperature, both strands α and β can unbind from the template cT$\alpha\beta$.

After step 3, if α and β are still attached to the template, the polymerase can bind again to the 3' end of α and extend it, dislocating the strand β. This intermediate process is equivalent to step 2, but provides an output signal β by polymerase displacement.

When the output signal of a converter template is the same as its input signal, we refer to it as an *autocatalytic template*.

Some converter templates are special, extending first their input signal into a longer one. Since this new input has a higher melting temperature, we consider that it never unbinds from the template at the working temperature, as illustrated in Figure 6 (Appendix). We refer to this kind of template as *irreversible template*.

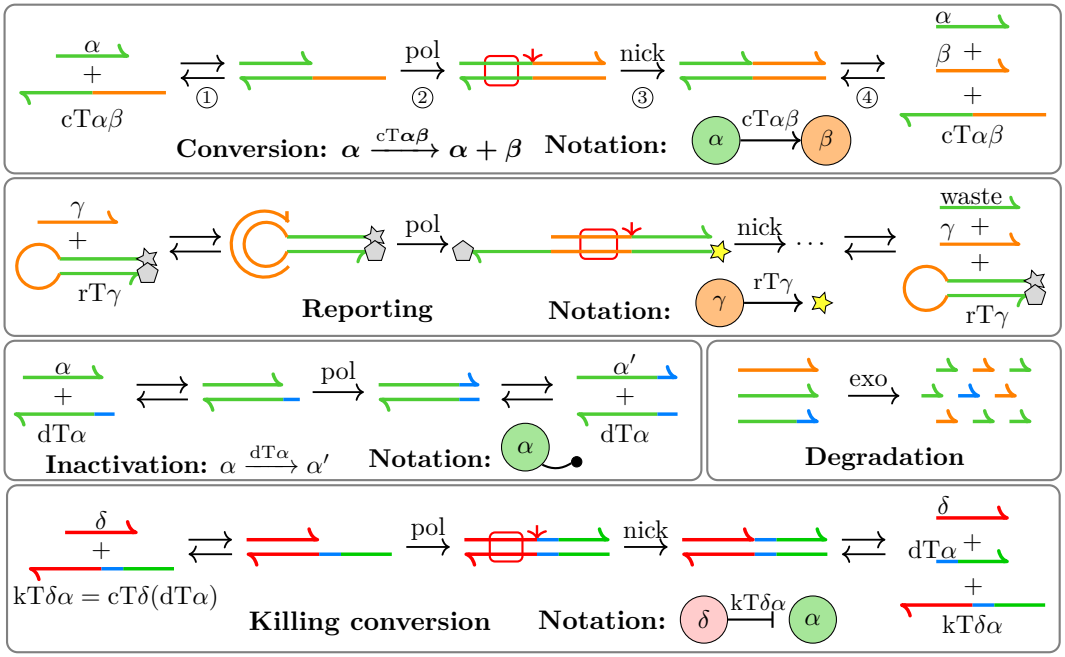


Figure 1 Schematic of the main reactions underlying PEN-DNA systems: conversion, reporting, inactivation, degradation, and killing conversion. Abbreviations pol, nick and exo denote the enzymatic actions of polymerase, nickase and exonuclease. A red arrow ↓ indicates the cleavage position of a nickase enzyme, and a red rectangle its recognition sequence. A star ☆ represents an emitting fluorophore, while a pentagon ⬠ represents its associated quencher.

Some converter templates may also feature a truncated 5' end to address thermodynamic or kinetic constraints, as illustrated in Figure 7 (Appendix). This truncation introduces a bias: when a signal has two binding options – one with a full-length template and one with a shortened template – the interaction with the full-length version will be thermodynamically favored [19].

Inactivation and killing conversion

A **drain template (dT)** inactivates a strand α by converting it into an extended form, denoted α', via the reaction $\alpha \xrightarrow{\mathrm{dT}\alpha} \alpha'$. While α' can still bind to the converter templates, it cannot be extended by the polymerase anymore because its 3' end is misaligned with the template, thereby preventing further conversion, as illustrated in Figure 8 (Appendix).

A converter template whose output strand is a drain template is called **killer template (kT)**. A schematic is presented in Figure 1.

Reporting

A **reporter template (rT)** functions as a molecular beacon, designed to signal the presence of a specific strand [14]. As illustrated in Figure 1, it adopts a hairpin secondary structure at the working temperature. In this conformation, the quencher is positioned adjacent to the fluorophore, effectively preventing fluorescence. However, when a reported strand binds to the hairpin loop, it is extended by the polymerase. This extension opens the hairpin, spatially separating the quencher from the fluorophore and enabling fluorescence.

The reporter template can also switch off if the reported strand is no longer present in solution. To achieve this, the extended reported strand is cleaved by the nickase into two short fragments, which are sufficiently small to unbind at the working temperature. Once detached, these fragments are degraded by the exonuclease, allowing the reporter template to refold into its hairpin loop. This refolding repositions the quencher near the fluorophore, quenching fluorescence once again.

The reporting process is not the only mechanism for tracking the evolution of a PEN-DNA system. *EvaGreen*, a green fluorescent nucleic acid dye, is also commonly used to quantify the amount of double-stranded DNA species present in the system.

Degradation

All DNA strands in PEN-DNA systems are subject to degradation by exonuclease, which cleaves nucleotides sequentially from the strand ends. The ttRecJ enzyme is commonly employed, degrading strands from their 5' ends [23]. To prevent degradation, DNA strands can be chemically protected using three phosphorothioate bases at their 5' ends. This modification replaces one of the oxygen atoms in the DNA backbone with a sulfur atom, making the strand less recognizable and resistant to enzymatic cleavage [33].

2.2 Graph representation

To represent a PEN-DNA system, a graph-like schematic is used. The notation – illustrated in Figure 1 – is inspired by gene regulatory networks. A simple arrow ($\rightarrow$) denotes a conversion, a black dot ($\multimap$) represents a drain template, a T-shaped arrow ($\dashv$) indicates a killing conversion, and an arrow pointing to a star ($\rightarrow\!\!\star$) refers to a reporting.

3 Related work

PEN-DNA systems are simulated by modeling the interactions between all species over time using ordinary differential equations (ODEs). Specifically, the general reactions described in Section 2 can be decomposed into two fundamental sub-reactions: *DNA hybridization* and *enzymatic processes*.

While many articles on PEN-DNA systems include supplementary sections on differential equations for simulation purposes, these sections are often dense and difficult to interpret due to different and complex notations [19, 20, 21].

To standardize the simulation of any PEN-DNA system, Aubert-Kato et al. developed *DACCAD* (DNA Artificial Circuit Computer-Assisted Design) [2]. This tool automates the generation of differential equations from the graph representation of a PEN-DNA system and solves them, significantly accelerating the design process. DACCAD has proven to be both consistent and reliable, qualitatively reproducing experimental results such as oscillators and switchable memory systems [2, 20, 23].

However, since the development of DACCAD, the PEN-DNA toolbox has evolved into a new version, replacing inhibitor-based regulation with drain template inactivation. Drain templates are preferred because they irreversibly convert active strands into non-functional species and provide a tunable degradation pathway that improves control over network dynamics [19].

It has been reported that DACCAD is extensible to support advanced features like reporter or drain templates. To our knowledge, although some of the corresponding equations have been described in detail [3], their practical implementation remains unavailable. Additionally,

a previous attempt to simulate drain templates in DACCAD revealed an unexpected behavior: a bistable system switched off only after spiking a large concentration of drain species over several minutes, rather than through a single spike of 60 nM of drain [3].

Furthermore, the latest version of the toolbox introduces additional complexities, including the use of two distinct nicking enzymes (Nb.BsmI and Nt.BstNBI), irreversible templates, and different versions of the same signal strand [12, 21]. These innovations represent new reactions that would be valuable to simulate, further highlighting the need for updated simulation capabilities.

DACCAD faces another limitation in its computation of hybridization rates. Initially, it assigns a uniform binding rate to all signals and derives unbinding rates from the dissociation constant, as detailed in Section 4. The tool requires users to input the dissociation constant for each signal, assuming that all rates associated with a given signal are identical. This simplification was primarily due because all converter templates and signals were used in their full-length version, which was sufficient to simulate systems based on the first version of the PEN-DNA toolbox. However, unbinding rates depend on the binding sequence length, leading to differences between full-length converter templates and their 3' shortened counterparts. This distinction has been identified as a key factor in implementing inactivation processes: the dissociation rate between a signal and its drain template differs from that between the same signal and its converter template [19]. Moreover, even the same signal can be present under shortened or elongated forms, to speed up certain reactions, as it has been used in some studies [12, 29]. To ensure consistency with the latest experimental findings, simulations should now account for these discrepancies by defining a specific kinetic rate for each pair of interacting DNA strands in the system.

A final limitation pertains to the kinetics rates of enzymatic processes. In DACCAD, users can adjust the Michaelis-Menten constants of all enzymatic processes. Nevertheless, these constants are not derived from an enzyme concentration, which could be more intuitive and user-friendly. Additionally, these rates should be derived from multiple parameters, including salt concentrations, temperature, and template sequences.

In the following sections, we present PENSim, a toolkit for generating and solving the ordinary differential equations (ODEs) of PEN-DNA systems. This tool addresses most of the limitations of previous approaches and enables qualitative reproduction of the behavior of complex and recent systems, which are based on inactivation mechanisms.

4 Kinetics and rates computation

In this section, we describe how the kinetic rates used in PENSim are computed. We first present DNA hybridization dynamics, then enzymatic reactions, and finally discuss their limitations.

4.1 DNA hybridization

The overall chemical equation for DNA hybridization between strands A and B is

$$\mathrm{A} + \mathrm{B} \underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} \mathrm{AB}$$

where AB is the DNA duplex formed by A and B, and k_{on} and k_{off} are the *association* and *dissociation* rates.

The association reaction is first-order in each reactant, described by $\frac{d[\mathrm{AB}]}{dt} = k_{\mathrm{on}}[\mathrm{A}][\mathrm{B}]$ with k_{on} typically ranging between 7×10^5 and 3×10^7 $\mathrm{M}^{-1} \cdot \mathrm{s}^{-1}$ [1, 39].

The dissociation reaction is also first-order, described by $\frac{d[\mathrm{AB}]}{dt} = -k_{\mathrm{off}}[\mathrm{AB}]$ with k_{off} typically ranging between 6×10^{-6} and 4×10^{3} s^{-1} [1, 39].

At equilibrium, the concentration of AB is constant and $k_{\mathrm{on}}[\mathrm{A}][\mathrm{B}] = k_{\mathrm{off}}[\mathrm{AB}]$. The *dissociation constant* K_D, which quantifies the equilibrium binding affinity, is given by:

$$K_D := \frac{[\mathrm{A}][\mathrm{B}]}{[\mathrm{AB}]} = \frac{k_{\mathrm{off}}}{k_{\mathrm{on}}} \tag{1}$$

Under the quasi-equilibrium hypothesis, k_{on} can be derived from k_{off} (or vice versa), if the dissociation rate is known. In DACCAD, a global k_{on} value of $3.3 \times 10^{6}\,\mathrm{M}^{-1} \cdot \mathrm{s}^{-1}$ and a signal-specific K_D of $74.328 \times 10^{-9}\,\mathrm{M}$ are used by default, unless otherwise specified by the user.

4.1.1 An automatized computation

Decades of research into the mechanism of hybridization have led to a deep understanding of its thermodynamics, resulting in the development of detailed computational models [16, 30, 38]. However, many questions remain regarding its kinetics and dynamics.

Recent advances include coarse-grained computational models that describe hybridization as a two-step process: nucleation of a critical base-pair stretch followed by rapid zippering [8, 22]. Tools like Multistrand, which simulates nucleic acid kinetics using continuous-time Markov chains, enable the exploration of folding pathways and reaction rates [17, 31].

Another approach involves developing nearest-neighbor (NN) thermodynamic models to compute association and dissociation rates directly from the sequence, similar to how free energy is calculated [26, 30]. This approach is based on the *Eyring equation*, which computes the rate of a chemical reaction:

$$k = \frac{k_B T}{h} \exp\left(-\frac{\Delta G^{\ddagger}}{RT}\right) \tag{2}$$

where:

- T is the temperature,
- k_B is the Boltzmann constant,
- h is the Planck constant,
- R is the gas constant,
- $\Delta G^{\ddagger}$ is the Gibbs energy of activation.

The Gibbs energy of activation can be computed using the thermodynamic equation:

$$\Delta G^{\ddagger} = \Delta H^{\ddagger} - T\Delta S^{\ddagger} \tag{3}$$

where $\Delta H^{\ddagger}$ is the enthalpy of activation and $\Delta S^{\ddagger}$ is the entropy of activation.

The nearest-neighbor model designed by Rejali et al. [26] allows the computation of the enthalpy and entropy of activation from the DNA sequence. The article provides two NN models for association and two for dissociation, each corresponding to one of two buffer conditions: 1.0 M NaCl or 2.2mM of $MgCl_2$.

For dissociation, the models demonstrated high accuracy: in 1.0 M NaCl, predicted dissociation rates were within 3-fold of literature data. However, for association, predictions were less accurate, with NN predictions differing from experiments by $35 \pm 30\%$ even on the training data. This discrepancy is attributed to weak temperature dependence and overparameterization of the NN terms.

Given that the thermodynamics of hybridization are well-studied and models are well-established, we aim to ensure that our computed rates are consistent with the dissociation constant K_D. To achieve this, we use only one of the nearest-neighbor (NN) models – either for association or dissociation – and derive the complementary rate from the K_D value computed using NUPACK [38].

Although NN models for association are less accurate, and exhibit weaker dependence on thermodynamic parameters – such as sequence length and temperature – we chose to derive k_{off} from k_{on} in PENSim. When the reverse approach is taken, even small errors in the computation of k_{off} are amplified upon division by K_D, rendering k_{on} overly sensitive to variations in sequence length and temperature. A detailed comparison of these approaches is available in the file `resources/comparison_computation_kon_koff.ipynb` in the Git repository and in Figure 9 (Appendix).

4.1.2 Coaxial stacking slowdown

At a nick, adjacent bases undergo collaborative stabilization through *coaxial stacking*, which slows the release of both input and output strands. This effect depends on the bases located at the nick site. Early studies in DACCAD estimated realistic slowdown values based on the data available at the time [2, 24, 37]. However, due to experimental constraints, these studies could only measure stacking interactions between base *pairs*, not between individual bases. This has prevented knowing stacking energies between A and C for example, because in the context of a duplex this would be necessarily paired with stacking of the hybridized bases (T and G in this example).

Recently, a study using a centrifuge force microscope advanced our understanding of coaxial stacking energy by directly measuring the unbinding slowdown [25]. Building on these findings, we incorporated the updated values into our model, automating the computation to account for temperature and nick sequence using the formula:

$$\mathsf{stack\_slowdown} = \exp\left(\frac{\Delta G_{\text{stack}}}{\text{RT}}\right)$$

This refinement shifts the reference slowdown value in DACCAD from 0.2 to 0.035, as initial works had underestimated the coaxial stacking energy.

4.1.3 Limitations

Despite the improved accuracy of our hybridization kinetic rate computations, several uncertainties need to be pointed out. First, the NN model was calibrated specifically for 1.0 M NaCl, which does not match the buffer conditions used in PEN-DNA systems – which usually contain 0.07 M monovalent ions and 0.0125 M bivalent ions. Note that although these buffer conditions do not match, this is not a major concern for qualitative predictions, since all association rates k_{on} are relatively close from each other. System behavior is primarily determined by the dissociation rates, which are computed through NUPACK using the actual buffer concentrations. For example, a DNA strand will bind at nearly the same speed to a shortened converter template and to its full-length counterpart, but it will unbind much more quickly from the shortened version.

Second, regarding again association rates, the NN model neglects the impact of secondary structures on kinetics, prioritizing computational speed over accuracy. More sophisticated approaches, such as Multistrand or molecular dynamics simulations, could refine these computations [1, 17].

Finally, chemical modifications to the DNA backbone – such as 5' phosphorothioate – are known to affect both kinetic and thermodynamic constants [1, 6]. Since these modifications are employed in the PEN-DNA toolbox to prevent degradation, the release of output signals may be accelerated. Additionally, the presence of netropsin, used to stabilize PEN-DNA systems, may further alter all kinetic rates.

4.2 Enzymatic reactions

All enzymatic processes in this study are described using Michaelis–Menten kinetics, under the quasi-equilibrium hypothesis [28]. In the case of several reactions using the same enzyme ($S_i \xrightarrow{E} P_i$), their competitive behavior is modeled by the rate equation $\frac{d[P_i]}{dt} = \text{enz}_i \cdot [S_i]$, where the *effective enzymatic activity*, enz_i, is given by:

$$\text{enz}_i = \frac{k_{\text{cat},i} \cdot [E]_{\text{total}}}{K_i \cdot \left(1 + \sum_j \frac{[S_j]}{K_j}\right)}$$

where:

- $[P_i]$ and $[S_i]$ are the product and substrate concentrations (in nM),
- $k_{\text{cat,i}}$ is the *catalytic rate constant* associated to the i-th enzymatic process,
- K_i is the *Michaelis constant* (in nM) associated to the i-th enzymatic process,
- $[E]_{\text{total}}$ is the total enzyme concentration (in nM, or in units/mL).

For simplicity, PENSim assumes that all catalytic rate constants ($k_{\text{cat,i}}$) and Michaelis constants (K_i) are identical for a given enzyme, except for the polymerase. For this enzyme, we distinguish two cases: whether it is displacing the output strand or not. Rates associated with this special case are denoted with the suffix `_displ` in subsequent sections.

4.2.1 Rates computation

In PENSim, we adopted the same enzymatic kinetic constants as those used in DACCAD processes involving polymerase, exonuclease, and the Nt.BstNBI nickase enzyme, based on the study by Padirac et al. [2, 23]. However, we modified the approach to compute $V_{\max} = k_{\text{cat}} \cdot [E]_{\text{total}}$ directly from the enzyme concentration, enhancing user-friendliness.

Additionally, we incorporated the distinct kinetic constants of the Nb.BsmI nicking enzyme, which differ from those of Nt.BstNBI, as measured by Montagne et al. [19]. All kinetic rates are summarized in Table 1 (Appendix).

4.2.2 Limitations

All enzymatic rates depend on temperature, salt conditions and template sequence [11, 18]. While we currently use constant rates for simplicity, this approach will be refined in future work, as precise method for measuring enzymatic rates are available [11]. Additionally, the polymerase rates described here apply to the Bst DNA polymerase. To our knowledge, no measurements exist for the Vent(exo–) polymerase, which is also used in PEN-DNA systems [18, 21].

Intuitively, these remarks primarily limit quantitative predictions. However, one should also note that, since enzymatic parameters are fixed as constants, this likely impacts qualitative explorations of temperature and buffer conditions as well.

5 Overview of PENSim

5.1 Installation and code organization

PENSim is a piece of software developed in Python. It relies on basic packages but also on NUPACK for computing the kinetic constants [38]. PENSim is publicly available on this Git repository: `https://github.com/GwendalDucloz/PENSim`.

PENSim can be installed by following the instruction in the README. The main code is organized into three files:

- `strands.py` defines the DNA strands classes, as described in the following subsection.
- `rates.py` manages all the kinetics rates, as detailed in Section 4.
- `PEN_System.py` is the core file for system setup, ODEs generation, simulation execution, and result visualization (including graph-like representations).

5.2 Species and equations

As previously described, the system is simulated using an ordinary differential equation (ODE) of the form: $\frac{dx}{dt} = f(x)$, where each coordinate x_i of the variable vector x represents the concentration of a species within the PEN-DNA system. This species may be a single strand or a complex of strands.

Four classes of species are defined in the `strands.py` file:

- **Signal**: Can exist in two states:
  - alone: Free in solution.
  - drained: The Signal once inactivated by a drain template, free in solution.
- **Drain**: Can exist in three states:
  - alone: Free in solution.
  - in: Bound to the target signal.
  - ext: Bound to the inactivated signal.
- **Template**: Can exist in six states:
  - alone: Free in solution.
  - in: Bound to its input signal.
  - out: Bound to its output signal.
  - ext: Bound to its extended input signal (via polymerase activity).
  - both: Bound to both input and output signals.
  - in_drained: Bound to an inactivated input strand.

  Note 1: The state where the template is bound to an inactivated output strand was not implemented, as it is suspected not to significantly inhibit the system.

  Note 2: Irreversible Template includes an additional state called load. It corresponds to the initial binding step of the input signal to the template before irreversible extension. The in state refers to the irreversible binding of the extended input (illustrated in Figure 6).
- **Reporter Template**: Shares the same states as reversible templates, except for in_drained, which cannot open the hairpin structure of reporter templates.

The ODEs for all species are computed by decomposing each reaction into either (de)hybridization or enzymatic processes (see Section 4). An overview of the resulting system of equations is provided in Appendix D (Appendix), while the full set of equations is available in the `resources/` folder within the accompanying Git repository.

Importantly, in addition to hybridization and enzymatic processes, we must account for a polymerase default behavior: even in the absence of an input signal, conversion can be spontaneously initiated by spurious polymerization or impurities [19]. This phenomenon, referred to as *leak*, is modeled in PENSim by defining a leak rate, which quantifies the fraction of polymerase molecules initiating spurious reactions.

5.3 Tutorials

Tutorials of use are available in the `tutorials/` directory of the accompanying Git repository. The notebook `tutorial1_basic.ipynb` guides users through basic simulations of a PEN-DNA system, covering:

1. **System definition:** Specifying experimental conditions (temperature, salt concentrations, enzyme concentrations, and polymerase leak rate) and the main species (signals, drain templates, converter templates, and reporter templates).
2. **Simulation execution** over a predefined time range.
3. **Result visualization**: Plotting species concentrations and fluorescence (reporter templates/EvaGreen dye).
4. **System validation:** Checking kinetic rates and graph-based coherence.

This tutorial illustrates a simple system with the mechanisms presented in Figure 1 (conversions, inactivation, killing conversions, reporting).

Two additional tutorials explore other simulable scenarios, including systems with multiple versions of the same signal strands, templates degraded by exonucleases, and 3' shortened templates extended by polymerase.

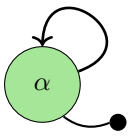
6 Validation

In this section, we validate PENSim across three representative classes of PEN-DNA systems: dynamical systems (bistability), sensing circuits (microRNA detection), and computational architectures (neural networks). Unless otherwise specified, all simulation parameters are directly sourced from the referenced articles. The only tuned parameter is the polymerase leak rate, which is generally set to 6×10^{-7}, unless specified otherwise. All simulations are available in the `validation/` folder within the Git repository.

6.1 Bistable systems and inactivation process

A key contribution of this article is the ability to simulate the inactivation process (described in Section 2). Our first validation demonstrates that the simulated process exhibits the same behavior as experimentally observed in the study from Montagne et al. [19]. The simulation is available in the notebook `validation/drain_validation.ipynb` in the accompanying Git repository.

Specifically we simulate a system comprising a single signal α, its autocatalytic template, and a drain template, as schematized below:



Our simulations qualitatively confirm the following main behaviors, as detailed in the original article:

1. **Self-reproduction and equilibrium of α**
   In the absence of drain templates, α self-reproduces through conversion until it reaches equilibrium, where its production is balanced by degradation via exonuclease. However, this amplification spontaneously starts even without α, due to polymerase leak.

Drain templates are introduced to capture and inactivate self-produced α, thereby slowing the self-activation. Beyond the *critical drain concentration*, a bifurcation occurs, stabilizing the $[\alpha] = 0$ nM state. Past this point, the system becomes *bistable*, exhibiting either no α or a stable positive concentration.

The critical drain concentration is determined by defining the *start time* (C_t) as the time at which the concentration of α reaches 20% of its maximal value. This concentration corresponds to the point where $1/C_t$ reaches 0, with $1/C_t$ set to 0 in the absence of amplification. This process is illustrated in Figure 2.

2. **Impact of drain template length**
   The longer the drain template, the slower the inactivated strand (α') dissociates, reducing the efficiency of the inactivation process. Consequently, longer drain templates require a higher critical drain concentration to achieve bistability, as revealed in Figure 10 (Appendix).
3. **Effect of shortened 3' end in the autocatalytic template**
   Thermodynamic predictions indicate that a shortened 3' end in the autocatalytic template enhances the binding affinity of α for the drain template (which exhibits a full length interaction). As a result, inactivation becomes more efficient, and the critical drain concentration decreases as the 3' end of the autocatalytic template is shortened. This behavior is presented in Figure 11 (Appendix).

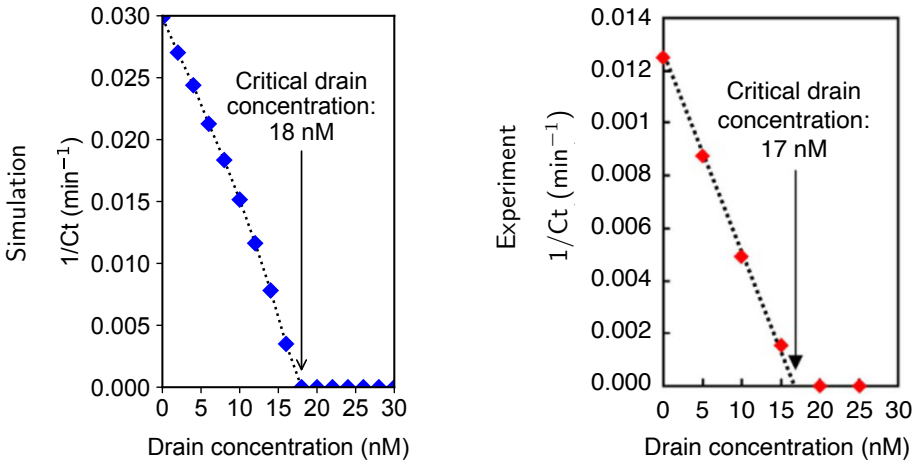


Figure 2 $1/C_t$ versus drain concentration for simulated (left, blue) **and experimental data** (right, red) adapted from Montagne et al. [19]. In the simple autocatalytic network, self-initiation occurs spontaneously at a drain concentration of 0. As the drain template concentration increases, it first delays and eventually suppresses this self-initiation. In this simulation, the leak rate has been set to 2×10^{-7}.

Furthermore, in an analog bistable system using γ instead of α, we simulated the system's switch-on and switch-off behavior by triggering either γ (on) or dTγ (off). The simulations successfully reproduced the expected γ concentration dynamics and the experimentally measured fluorescence signals (EvaGreen and red Cy3.5), as shown in Figure 3.

This result contrasts with previous attempts in DACCAD, where simulations of drain templates were reported to result in unexpected behavior: the system switched off only after spiking a large concentration of drain species over several minutes [3]. This discrepancy likely arose because (de)hybridization kinetics were modeled identically for both signal-drain template and signal-converter template interactions.

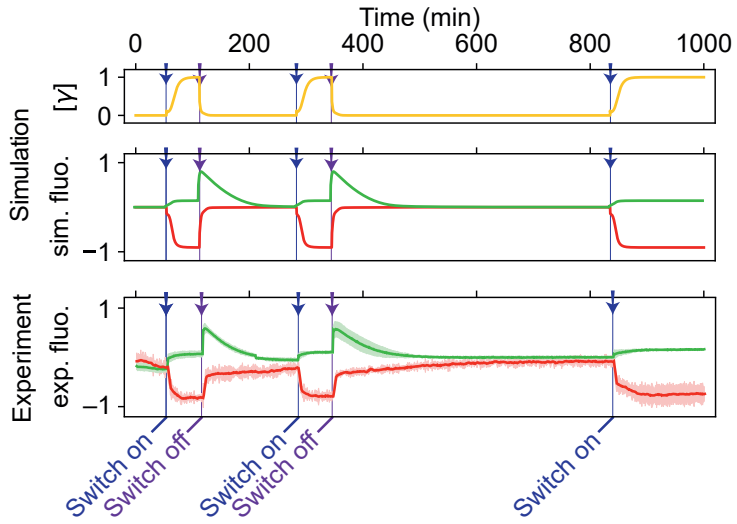
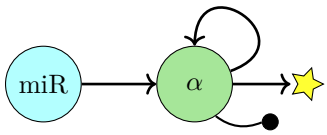


Figure 3 Simulation and experimental comparison of bistable system dynamics. The first two curves (top) are from our simulation, triggered by spiking either 20 nM of γ (switch-on) or 340 nM of drainγ (switch-off). The yellow curve represents the normalized concentration of γ, while the red and green curves correspond to normalized EvaGreen and red Cy3.5 fluorescences. The third curve (bottom) is adapted from Montagne et al. [19], where the system is toggled using 20 nM of γ and 60 nM of drainγ.

6.2 Sensing systems: microRNA detection

Ginès et al. recently developed PUMA, an isothermal, digital detection method for microRNAs (miR) using a PEN-DNA circuit [12, 18, 29]. This system integrates an irreversible converter template, an autocatalytic template, a drain template and a reporter template, as illustrated below:



Qualitatively, our simulations demonstrate the system's ability to detect microRNA concentrations as low as femtomoles, validating the use of different nickases in the same system (Nt.BstNBI and Nb.BsmI), reporting, and irreversible templates implementations. The simulation files are available in `validation/PUMA_simulation.ipynb` on the Git repository, and the result is presented in Figure 4.

Moreover, our model validates the presence of modified signal variants, where additional bases are truncated at the 5' end and extended at the 3' end, while maintaining compatibility with the autocatalytic template. Specifically, the converter template produces the strand called α+2–2, which binds to the autocatalytic template with two bases downstream of the nicking position. This modification biases the thermodynamic, favoring the input region of the autocatalytic template, relative to the drain template, thereby altering the kinetics.

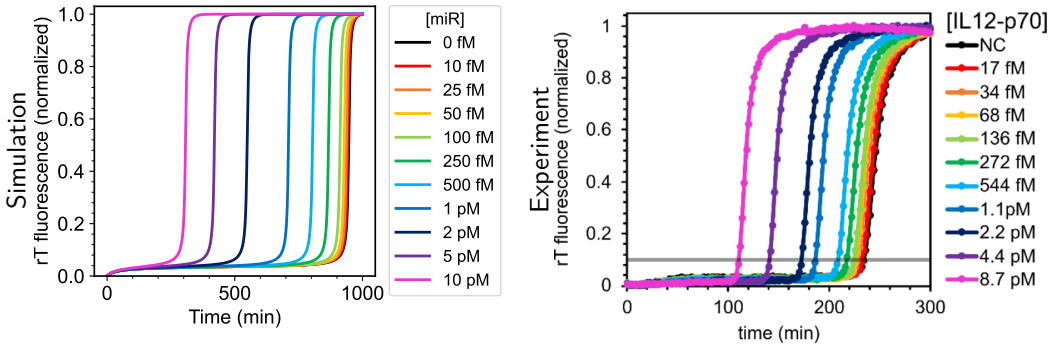


Figure 4 Simulated (left) and experimental (right) PUMA system. The experimental data are adapted from Masurier et al. [18]. The simulation was conducted at 50°C using concentrations of 0.5 nM converter template, 50 nM autocatalytic template, 9.66 nM drain template (compared to 6 nM in the experimental data), and 50 nM reporter template. Measurements were taken over a time range of 1000 minutes (extended from 300 minutes in the experiment). In both cases, the detection limit of the system is approximately 100 fM.

6.3 Neural network simulation

The PEN-DNA toolbox has been demonstrated to enable the implementation of neural networks [21]. From a theoretical perspective, this proof of concept is significant, as it establishes that PEN-DNA systems are functionally complete and may be capable of Turing-complete computations under idealized conditions (e.g. unbounded precision and infinite memory). Thus, the ability to accurately simulate neural networks serves as a critical validation for our simulator.

The capability of PENSim to simulate neural networks is demonstrated in the notebook `validation/neural_network_simulation.ipynb` (available in the Git repository) and is illustrated in Figure 5. The simulator qualitatively reproduces the behavior of positive-weighted linear classifiers (exhibiting a negative slope) with respect to converter and drain template concentrations. It also approximates the behavior of negative-weighted classifiers (exhibiting a positive slope) with respect to temperature, aligning with experimental results reported by Okumura et al. [21]. From a logic gate perspective, positive-weighted classifiers function as AND gates, while negative-weighted classifiers behave as AND-NOT gates.

7 Conclusion

The PENSim package enables the simulation of complex reactions based on the PEN-DNA toolbox, expanding the scope of previously simulable systems on DACCAD by incorporating new mechanisms used in recent studies. It also enhances quantitative precision through a novel method for computing kinetic rates, grounded in thermodynamic properties such as DNA sequences, temperature, and salt concentrations. This advancement addresses many of the limitations inherent in previous tools.

PENSim has been validated by simulating previously experimentally characterized systems, including bistable systems [19], microRNA detection circuits [29], and linear classifiers based on neural network architectures [21]. In all cases, PENSim produced qualitative results that faithfully replicate experimental behaviors. This makes it a valuable tool for designing and validating experiments *in silico* before laboratory implementation, while also allowing users to explore the impact of critical parameters such as DNA lengths and temperature.

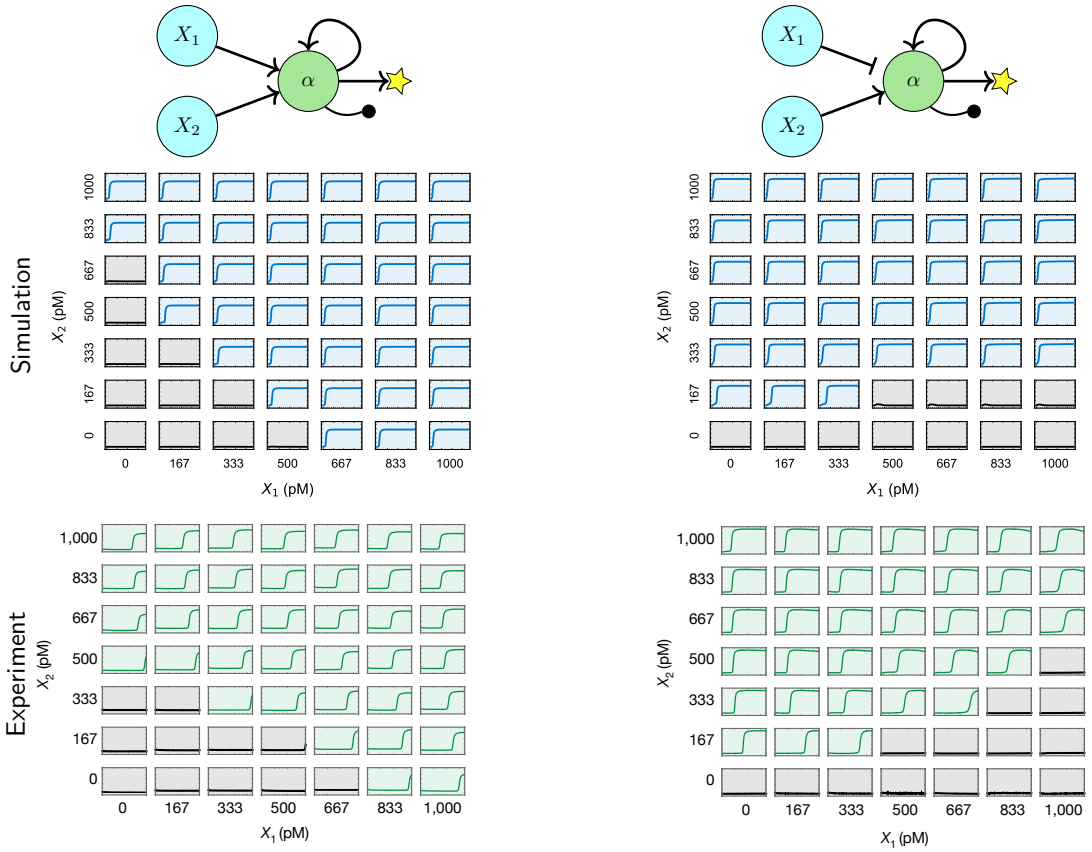


Figure 5 Simulation (top, blue) and experimental (bottom, green) results for linear classifiers. Simulations were conducted using the same configuration as the experiments, from which the bottom figure is adapted [21]. The only deviation is the polymerase concentration, which was set to half the experimental value, as PENSim was calibrated using Bst polymerase rates, whereas the experiments used Vent (exo–) polymerase – known to be slower. Another difference is that the right system was simulated at 46°C instead of 45°C. The **left** system represents a positive-weighted classification, while the **right** system represents a negative-weighted classification. The underlying PEN-DNA systems are schematized at the top. Qualitative behaviors align between simulations and experiments, with the positive-weighted classification showing strong quantitative agreement. However, the negative-weighted classification exhibits less quantitative agreement, as the slope is significantly flatter than in experiments.

However, PENSim is not yet a quantitative oracle. While it significantly improves qualitative accuracy and modeling scope, several challenges remain to achieve fully quantitative simulations. Intuitively, the major obstacles, from most to least important, are:

1. Enzymatic reactions are insufficiently characterized. In our simulator, kinetic rates are currently modeled as dependent solely on enzyme concentrations, despite their known dependence on temperature and salt conditions. The existing rates were measured under different experimental conditions (e.g., buffer, temperature) than those used in recent systems. Additionally, enzymes such as Vent(exo–) polymerase (now often used instead of Bst) have not been measured in any PEN-DNA buffer. Enzymatic activity is also highly sequence-dependent, with rates varying by up to threefold between templates [18].
2. To our knowledge, polymerase leak rates are currently uncharacterized. In the previous simulations, this rate was set to 6×10^{-7}, as this value yields convincing results with respect to enzymatic activities. While this parameter did not affect the qualitative agreement of the simulations, it remains a major obstacle for achieving quantitative accuracy.
3. Hybridization rates are still not fully understood. Our approach does not account for secondary structures or chemical modifications, which may significantly affect reaction dynamics. Moreover, the computation of association rates in PENSim was calibrated for a specific buffer condition, differing from the usual conditions in PEN-DNA systems.
4. Non-specific interactions, such as cross-talks and secondary structures, are not currently modeled, further limiting quantitative accuracy.

Future work

To enhance the capabilities of PENSim, we plan to pursue the following improvements:

- Develop a web-based graphical user interface to improve accessibility for non-expert users.
- Refine enzymatic reaction kinetics to improve quantitative predictions and temperature dependencies. Additionally, conduct a study of the polymerase leak rate to properly define and model it.
- Perform experiments to calculate quantitative indicators comparing simulation results with wet lab data. In particular, focus on kinetic predictions and sensitivity limits as key targets.
- Optimize the ODE solver to accelerate simulations for large-scale systems, enabling rapid exploration of parameter spaces and experimental conditions. Additionally, conduct experiments on complex architectures to assess simulation scalability.
- Enable automated species definition from input sequences, allowing users to simulate systems directly from raw DNA sequences.
- Integrate sequence design tools to facilitate seamless transitions from theoretical models to experimental implementation.
- Incorporate additional side effects, such as exonuclease inhibition by free templates [23], and support for new reactions to align with further updates in the PEN-DNA toolbox.

References

1 Brennan Ashwood and Andrei Tokmakoff. Kinetics and dynamics of oligonucleotide hybridization. *Nature Reviews Chemistry*, 9(5):305–327, May 2025. doi:10.1038/s41570-025-00704-8.

2 Nathanaël Aubert, Clément Mosca, Teruo Fujii, Masami Hagiya, and Yannick Rondelez. Computer-assisted design for scaling up systems based on DNA reaction networks. *Journal of The Royal Society Interface*, 11(93):20131167, April 2014. doi:10.1098/rsif.2013.1167.

3 Nathanael Aubert-Kato and Leo Cazenille. Designing dynamical molecular systems with the PEN toolbox. *New Generation Computing*, 38(2):341–366, May 2020. doi:10.1007/s00354-020-00089-w.

4 Alexandre Baccouche, Kevin Montagne, Adrien Padirac, Teruo Fujii, and Yannick Rondelez. Dynamic DNA-toolbox reaction circuits: A walkthrough. *Methods*, 67(2):234–249, 2014. doi:10.1016/j.ymeth.2014.01.015.

5 Shulian Bai, Bangtian Xu, Jiangling Wu, and Guoming Xie. Series or parallel toehold-mediated strand displacement and its application in circular RNA detection and logic gates. *Biosensors and Bioelectronics*, 241:115677, 2023. doi:10.1016/j.bios.2023.115677.

6 Antonio Carlesso, Johanna Hörberg, Giuseppe Deganutti, Anna Reymer, and Pär Matsson. Structural dynamics of therapeutic nucleic acids with phosphorothioate backbone modifications. *NAR Genomics and Bioinformatics*, 6(2):lqae058, May 2024. doi:10.1093/nargab/lqae058.

7 Mathias Centola, Erik Poppleton, Sujay Ray, Martin Centola, Robb Welty, Julián Valero, Nils G. Walter, Petr Šulc, and Michael Famulok. A rhythmically pulsing leaf-spring DNA-origami nanoengine that drives a passive follower. *Nature Nanotechnology*, 19(2):226–236, February 2024. doi:10.1038/s41565-023-01516-x.

8 M. E. Craig, D. M. Crothers, and P. Doty. Relaxation kinetics of dimer formation by self complementary oligonucleotides. *Journal of Molecular Biology*, 62(2):383–401, December 1971. doi:10.1016/0022-2836(71)90434-7.

9 Constantine G. Evans and Erik Winfree. Physical principles for DNA tile self-assembly. *Chem. Soc. Rev.*, 46:3808–3829, 2017. doi:10.1039/C6CS00745G.

10 Teruo Fujii and Yannick Rondelez. Predator–prey molecular ecosystems. *ACS Nano*, 7(1):27–34, 2013. doi:10.1021/nn3043572.

11 Guillaume Gines, Rocío Espada, Adèle Dramé-Maigné, Alexandre Baccouche, Nicolas Larrouy, and Yannick Rondelez. Functional analysis of single enzymes combining programmable molecular circuits with droplet-based microfluidics. *Nature Nanotechnology*, 19(6):800–809, June 2024. doi:10.1038/s41565-024-01617-1.

12 Guillaume Gines, Roberta Menezes, Kaori Nara, Anne-Sophie Kirstetter, Valerie Taly, and Yannick Rondelez. Isothermal digital detection of microRNAs using background-free molecular circuit. *Science Advances*, 6(4):eaay5952, 2020. doi:10.1126/sciadv.aay5952.

13 Robert F. Johnson and Lulu Qian. Simplifying chemical reaction network implementations with two-stranded DNA building blocks. In Cody Geary and Matthew J. Patitz, editors, *26th International Conference on DNA Computing and Molecular Programming (DNA 26)*, volume 174 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DNA.2020.2.

14 Ievgen Kurylo, Guillaume Gines, Yannick Rondelez, Yannick Coffinier, and Alexis Vlandas. Spatiotemporal control of DNA-based chemical reaction network via electrochemical activation in microfluidics. *Scientific Reports*, 8(1):6396, 2018. doi:10.1038/s41598-018-24659-7.

15 Matthew R. Lakin, Simon Youssef, Filippo Polo, Stephen Emmott, and Andrew Phillips. Visual DSD: a design and analysis tool for DNA strand displacement systems. *Bioinformatics*, 27(22):3211–3213, October 2011. doi:10.1093/bioinformatics/btr543.

16 Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011. doi:10.1186/1748-7188-6-26.

17 Jordan Lovrod, Boyan Beronov, Chenwei Zhang, Erik Winfree, and Anne Condon. Revisiting hybridization kinetics with improved elementary step simulation. In Ho-Lin Chen and Constantine G. Evans, editors, *29th International Conference on DNA Computing and Molecular Programming (DNA 29)*, volume 276 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:24, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DNA.29.5.

18 Antoine Masurier, Rémi Sieskind, Guillaume Gines, and Yannick Rondelez. DNA circuit-based immunoassay for ultrasensitive protein pattern classification. *Analyst*, 149:5052–5062, 2024. doi:10.1039/D4AN00728J.

19 Kevin Montagne, Guillaume Gines, Teruo Fujii, and Yannick Rondelez. Boosting functionality of synthetic DNA circuits with tailored deactivation. *Nature Communications*, 7(1):13474, 2016. doi:10.1038/ncomms13474.

20 Kevin Montagne, Raphael Plasson, Yasuyuki Sakai, Teruo Fujii, and Yannick Rondelez. Programming an in vitro DNA oscillator using a molecular networking strategy. *Molecular Systems Biology*, 7(1):466, 2011. doi:10.1038/msb.2010.120.

21 Shu Okumura, Guillaume Gines, Nicolas Lobato-Dauzier, Alexandre Baccouche, Robin Deteix, Teruo Fujii, Yannick Rondelez, and Anthony J Genot. Nonlinear decision-making with enzymatic neural networks. *Nature*, 610(7932):496–501, 2022. doi:10.1038/s41586-022-05218-7.

22 Thomas E. Ouldridge, Petr Sulc, Flavio Romano, Jonathan P. K. Doye, and Ard A. Louis. DNA hybridization kinetics: zippering, internal displacement and sequence dependence. *Nucleic Acids Research*, 41(19):8886–8895, October 2013. doi:10.1093/nar/gkt687.

23 Adrien Padirac, Teruo Fujii, and Yannick Rondelez. Bottom-up construction of in vitro switchable memories. *Proceedings of the National Academy of Sciences*, 109(47):E3212–E3220, 2012. doi:10.1073/pnas.1212069109.

24 Ekaterina Protozanova, Peter Yakovchuk, and Maxim D. Frank-Kamenetskii. Stacked-unstacked equilibrium at the nick site of DNA. *Journal of Molecular Biology*, 342(3):775–785, September 2004. doi:10.1016/j.jmb.2004.07.075.

25 Jibin Abraham Punnoose, Kevin J. Thomas, Arun Richard Chandrasekaran, Javier Vilcapoma, Andrew Hayden, Kacey Kilpatrick, Sweta Vangaveti, Alan Chen, Thomas Banco, and Ken Halvorsen. High-throughput single-molecule quantification of individual base stacking energies in nucleic acids. *Nature Communications*, 14(1):631, February 2023. doi:10.1038/s41467-023-36373-8.

26 Nick A. Rejali, Felix D. Ye, Aisha M. Zuiter, Caroline C. Keller, and Carl T. Wittwer. Nearest-neighbour transition-state analysis for nucleic acid kinetics. *Nucleic Acids Research*, 49(8):4574–4585, May 2021. doi:10.1093/nar/gkab205.

27 Anna Romanov, Grant A. Knappe, Larance Ronsard, Christopher A. Cottrell, Yiming J. Zhang, Heikyung Suh, Lauren Duhamel, Marjan Omer, Asheley P. Chapman, Katie Spivakovsky, Patrick Skog, Claudia T. Flynn, Jeong Hyun Lee, Oleksandr Kalyuzhniy, Alessia Liguori, Molly F. Parsons, Vanessa R. Lewis, Josue Canales, Boris Reizis, Ryan D. Tingle, Torben Schiffner, William R. Schief, Daniel Lingwood, Mark Bathe, and Darrell J. Irvine. DNA origami vaccines program antigen-focused germinal centers. *Science*, 391(6785):eadx6291, 2026. doi:10.1126/science.adx6291.

28 Yannick Rondelez. Competition for catalytic resources alters biological network dynamics. *Phys. Rev. Lett.*, 108:018102, January 2012. doi:10.1103/PhysRevLett.108.018102.

29 Yannick Rondelez and Guillaume Gines. Programmable ultrasensitive molecular amplifier for digital and multiplex microRNA quantification. *Methods in Molecular Biology*, 2630:89–102, 2023. Chapter. doi:10.1007/978-1-0716-2982-6_7.

30 John SantaLucia Jr and Donald Hicks. The thermodynamics of DNA structural motifs. *Annual Review of Biophysics*, 33:415–440, 2004. doi:10.1146/annurev.biophys.32.110601.141800.

31 Joseph M. Schaeffer, Chris Thachuk, and Erik Winfree. Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In *DNA*, 2013. URL: https://api.semanticscholar.org/CorpusID:17591933.

32 Vasily Shenshin, Camille Lescanne, Guillaume Gines, and Yannick Rondelez. A small-molecule chemical interface for molecular programs. *Nucleic Acids Research*, 49(13):7765–7774, 2021. doi:10.1093/nar/gkab470.

33 C.A. Stein, Chrisanthi Subasinghe, Kazuo Shinozuka, and Jack S. Cohen. Physicochemical properties of phospborothioate oligodeoxynucleotides. *Nucleic Acids Research*, 16(8):3209–3221, April 1988. doi:10.1093/nar/16.8.3209.

34 Y. Wang, I. Baars, I. Berzina, I. Rocamonde-Lago, B. Shen, Y. Yang, M. Lolaico, J. Waldvogel, I. Smyrlaki, K. Zhu, R. A. Harris, and B. Högberg. A DNA robotic switch with regulated autonomous display of cytotoxic ligand nanopatterns. *Nature Nanotechnology*, 19(9):1366–1374, September 2024. doi:10.1038/s41565-024-01676-4.

35 Erik Winfree and Renat Bekbolatov. Proofreading tile sets: Error correction for algorithmic self-assembly. In Junghuei Chen and John Reif, editors, *DNA Computing*, pages 126–144, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. doi:10.1007/978-3-540-24628-2_13.

36 Damien Woods, David Doty, Cameron Myhrvold, Joy Hui, Felix Zhou, Peng Yin, and Erik Winfree. Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature*, 567(7748):366–372, March 2019. doi:10.1038/s41586-019-1014-9.

37 Peter Yakovchuk, Ekaterina Protozanova, and Maxim D. Frank-Kamenetskii. Base-stacking and base-pairing contributions into thermal stability of the DNA double helix. *Nucleic Acids Research*, 34(2):564–574, January 2006. doi:10.1093/nar/gkj454.

38 John N. Zadeh, Christopher D. Steenberg, Justin S. Bois, Blake R. Wolfe, Matthew B. Pierce, Abbas R. Khan, Raymond M. Dirks, and Niles A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, 2011. doi:10.1002/JCC.21596.

39 David Yu Zhang and Erik Winfree. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–17314, December 2009. doi:10.1021/ja906987s.

40 Fangzhou Zhao, Ziyi Sun, and Hanadi F. Sleiman. DNA-templated assembly of metalloprotein mimics with hydrolytic activity. *Journal of the American Chemical Society*, 147(40):36365–36375, October 2025. doi:10.1021/jacs.5c10056.

A Additional mechanisms and figures about the PEN-DNA toolbox

The mechanisms presented in Section 2 can incorporate additional features. For example, the converter template can act as an *irreversible* template, if it includes an extra domain between its input and output region (depicted in violet in Figure 6). In this configuration, once the polymerase extends the input strand, the template becomes "loaded". Since the nickase cleaves after the extended domain, the resulting extended α strand (comprising green and violet domains) remains stably bound to the template at the working temperature. This irreversible template continuously produces the β output signal.

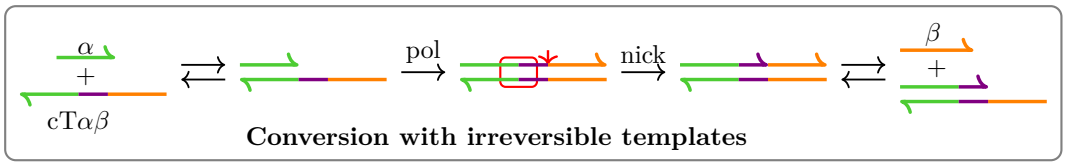


Figure 6 Schematic of the conversion with irreversible template. The converter template contains an extra domain (violet) that irreversibly binds the input once it is extended.

Another feature of the converter template involves its 3' end, which can be shortened by a couple of bases. While unbinding from shortened templates occurs faster than from standard templates, the overall binding is less thermodynamically favorable. Such shortened templates are typically used in inactivation processes, to favor pairing with the drain template. It is important to note that the converter template is usually protected by a 3' phosphorylation (represented by a "P" in Figure 7) to prevent the polymerase from re-extending the strand to its full length.

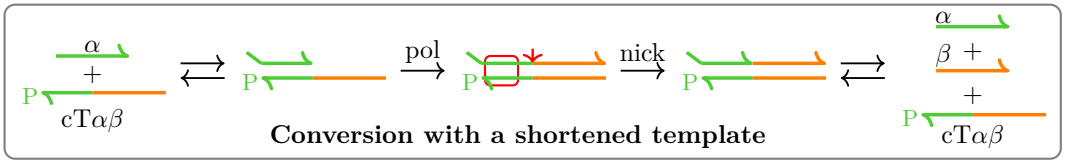


Figure 7 Schematic of the conversion with a shortened template. A P signifies a 3' phosphorylation, which prevents the polymerase from extending the template.

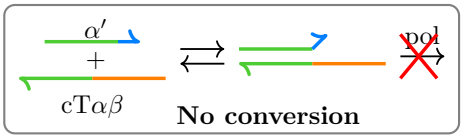


Figure 8 Schematic of the interaction between an inactivated strand and a converter template. Even though the strand can bind to the template, it cannot be extended by the polymerase since its 3' end (in blue) does not properly match the template.

B Kinetics supplement

As discussed in Section 4, deriving association rates by deriving them from dissociation rates via NUPACK may introduce errors. Given that K_D and k_{off} exhibit an exponential dependence on thermodynamic parameters, this approach can lead to incorrect behavior in k_{on}. For example, Figure 9 illustrates this exponential temperature dependency. Further exploration of these behaviors is available in the notebook `comparison_computation_kon_koff.ipynb` in the `resources/` folder in the Git repository.

C Validation supplement

As explained in Section 6.1 and illustrated in Figure 10, the critical drain concentration required to achieve bistability increases with the drain template length. Experimentally, however, this behavior does not fully align with theory, as a single mismatched nucleotide may not be sufficient to completely prevent polymerase extension of the input signal [19].

The simulation closely matches the measured behavior for 3'-shortened converter templates, as shown in Figure 11: the critical drain concentration decreases as the shortening size increases. However, even at 43°C (instead of 45°C), the system fails to self-start in the absence of the drain template for a shortening of 3 nucleotides. This is primarily because the input signal unbinds faster than the polymerase can extend it. This effect may be addressed in future work, once temperature-dependent enzyme activities will be incorporated.

D ODE construction

Following the schematic in Figure 1, the ODE function $f(x)$ is computed by decomposing each reaction into either (de)hybridization or enzymatic processes (see Section 4). This decomposition results in a system of equations, which are briefly outlined here. The rest of the full set of reactions is provided in the `resources/` folder in the Git repository, along with its modifications for complex features.

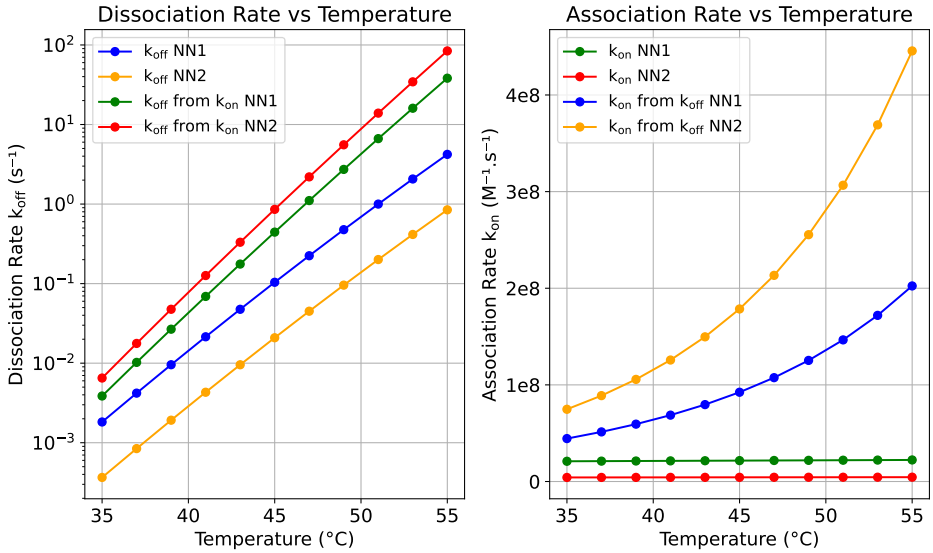


Figure 9 Comparison of the four ways to compute the hybridization rates applied to the DNA sequence `TAATCGGCAGGT`. For dissociation (and association), two nearest-neighbor (NN) models from Rejali et al. [26] are employed, along with two additional models derived from the NN models for association (and dissociation), as explained in Section 4. NN1 refers to the model designed for 1 M NaCl, while NN2 refers to the model developed for 2.2 mM $MgCl_2$. The association rates, when derived from the dissociation ones, suffer from an exponential dependency on the temperature.

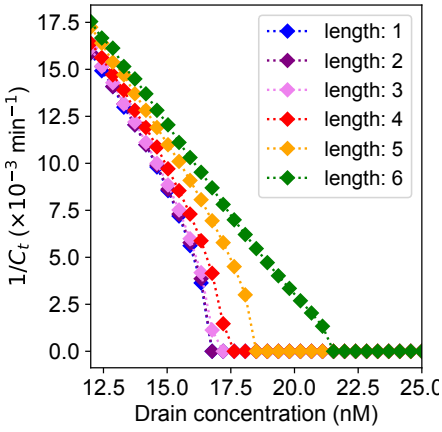


Figure 10 Simulated behavior of $\mathbf{1/C_t}$ **versus drain concentration for drain templates of varying lengths.** The legend indicates the length of the inactivation tail of the drain templates. The critical drain concentration increases with drain template length, indicating reduced inactivation efficiency for longer templates.

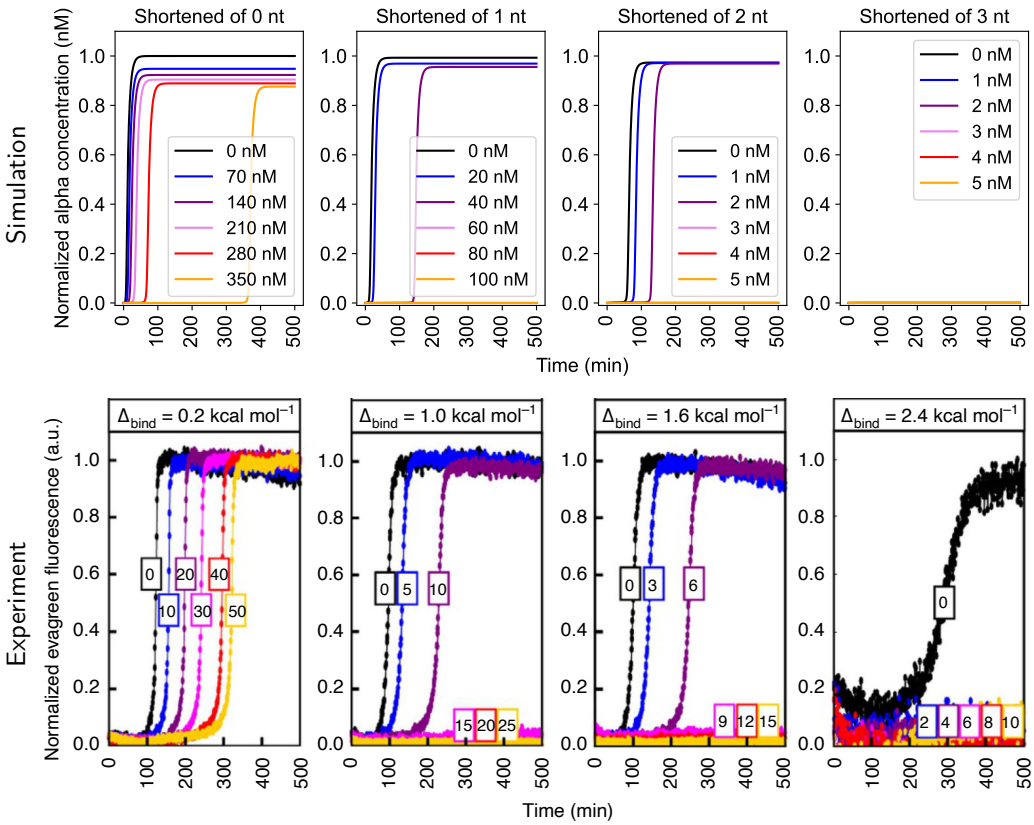


Figure 11 Normalized α concentration over time for templates of varying shortened lengths. The top panel shows results from our simulations (at 43°C), while the bottom panel presents experimental data adapted from Montagne et al. [19] (at 45°C). The critical drain concentration decreases with template length, reflecting enhanced inactivation efficiency for shorter templates.

We use the following notation:

- `s` denotes standalone signals, $\texttt{s}_{\texttt{dr}}$ inactivated signals, `d` drain, and `tmp` stands for template, with $\texttt{s}_{\texttt{in}}$ and $\texttt{s}_{\texttt{out}}$ as its input and output signals.
- [X] denotes the concentration of species X.
- $\texttt{k}_{\texttt{on}}(\texttt{X},\texttt{Y})$ and $\texttt{k}_{\texttt{off}}(\texttt{X},\texttt{Y})$ are the binding and unbinding rates between species X and Y.
- `leak(tmp)` is the leak rate of template tmp.
- `stack(tmp)` is the unbinding penalty coming from the stability increase from the coaxial base-stacking interaction between the input and output signals. `stack` is computed as detailed in Section 4.
- `exo`, `nick`, and `pol` represent the effective enzymatic activities of exonuclease, nickase, and polymerase enzymes, respectively. `pol_displ` is the specific effective activity of the polymerase when displacing the output signal strand.
- $\texttt{polV}([\texttt{polymerase}],\texttt{T}) = [\text{polymerase}] \cdot k_{\text{cat}}^{\text{pol}}$, where $k_{\text{cat}}^{\text{pol}}$ is the catalytic rate constant of the polymerase.
- `polK` is the Michaelis constant of the polymerase and `polK_displ` the one for the polymerase when displacing the output signal strand.
- All concentrations and enzyme activities depend on time, but we omit the (`t`) term for readability.

alone signal equation:

$$\begin{aligned}\text{flux\_in}(s) = \sum_{\substack{\text{tmp whose}\\ \text{input is s}}} \Bigg(& k_{\text{off}}(\text{s},\text{tmp}) \cdot [\text{tmp}_{\text{in}}(\text{s})] && \text{(s unbinds from in)}\\ & + k_{\text{off}}(\text{s},\text{tmp}) \cdot \text{stack}(\text{tmp}) \cdot [\text{tmp}_{\text{both}}(\text{s})] && \text{(s unbinds from both)}\\ & - k_{\text{on}}(\text{s},\text{tmp}) \cdot [\text{s}] \cdot [\text{tmp}_{\text{alone}}] && \text{(s binds to alone)}\\ & - k_{\text{on}}(\text{s},\text{tmp}) \cdot [\text{s}] \cdot [\text{tmp}_{\text{out}}] \Bigg) && \text{(s binds to out)}\end{aligned}$$

$$\begin{aligned}\text{flux\_out}(s) = \sum_{\substack{\text{tmp whose}\\ \text{output is s}}} \Bigg(& k_{\text{off}}(\text{s},\text{tmp}) \cdot [\text{tmp}_{\text{out}}(\text{s})] && \text{(s unbinds from out)}\\ & + k_{\text{off}}(\text{s},\text{tmp}) \cdot \text{stack}(\text{tmp}) \cdot [\text{tmp}_{\text{both}}(\text{s})] && \text{(s unbinds from both)}\\ & - k_{\text{on}}(\text{s},\text{tmp}) \cdot [\text{s}] \cdot [\text{tmp}_{\text{alone}}] && \text{(s binds to alone)}\\ & - k_{\text{on}}(\text{s},\text{tmp}) \cdot [\text{s}] \cdot [\text{tmp}_{\text{in}}] && \text{(s binds to in)}\\ & + \text{pol\_displ} \cdot [\text{tmp}_{\text{both}}] && \text{(pol displacement)}\\ & + \text{leak}(\text{tmp}) \cdot \text{polV}([\text{pol}],\text{T}) \cdot [\text{tmp}_{\text{alone}}] \Bigg) && \text{(polymerase leak)}\end{aligned}$$

$$\begin{aligned}\frac{d[s]}{dt} = \; & \text{flux\_in}(\text{s}) + \text{flux\_out}(s) && \text{(interactions with templates)}\\ & + \sum_{\substack{\text{d whose}\\ \text{input is s}}} \Bigg(k_{\text{off}}(\text{s},\text{d}) \cdot [\text{d}_{\text{in}}] && \text{(s unbinds from the drain)}\\ & \qquad - k_{\text{on}}(\text{s},\text{d}) \cdot [\text{s}] \cdot [\text{d}_{\text{alone}}] \Bigg) && \text{(s binds to the drain)}\\ & - \mathbb{1}_{\{\text{s not protected}\}} \cdot \text{exo} \cdot [\text{s}] && \text{(exonuclease degradation)}\end{aligned}$$

alone drain equation:

$$\begin{aligned}\frac{d}{dt}[\mathrm{d_{alone}}] = \quad & k_{\mathrm{off}}(\mathrm{s_{dr}}, \mathrm{d}) \cdot [\mathrm{d_{ext}}] && \text{(inactivated signal unbinding from drain)}\\ & + k_{\mathrm{off}}(\mathrm{s}, \mathrm{d}) \cdot [\mathrm{d_{in}}] && \text{(signal unbinding from in)}\\ & - k_{\mathrm{on}}(\mathrm{s}, \mathrm{d}) \cdot [\mathrm{s}] \cdot [\mathrm{d_{alone}}] && \text{(signal binding to alone)}\\ & - k_{\mathrm{on}}(\mathrm{s_{dr}}, \mathrm{d}) \cdot [\mathrm{d_{alone}}] \cdot [\mathrm{s_{dr}}] && \text{(inactivated signal binding to alone)}\\ & + \mathrm{flux\_out(d)} && \text{(created drains by conversion)}\\ & - \mathbb{1}_{\{\mathrm{d\ not\ protected}\}} \cdot \mathrm{exo} \cdot [\mathrm{d_{alone}}] && \text{(exonuclease degradation)}\end{aligned}$$

both template equation (for reversible or irreversible):

$$\begin{aligned}\frac{d}{dt}[\mathrm{tmp_{both}}] = \quad & k_{\mathrm{on}}(\mathrm{s_{in}}, \mathrm{tmp}) \cdot [\mathrm{s_{in}}] \cdot [\mathrm{tmp_{out}}] && \text{(input binding to out)}\\ & + k_{\mathrm{on}}(\mathrm{s_{out}}, \mathrm{tmp}) \cdot [\mathrm{s_{out}}] \cdot [\mathrm{tmp_{in}}] && \text{(output binding to in)}\\ & - k_{\mathrm{off}}(\mathrm{s_{in}}, \mathrm{tmp}) \cdot \mathrm{stack(tmp)} \cdot [\mathrm{tmp_{both}}] && \text{(input unbinding from both)}\\ & - k_{\mathrm{off}}(\mathrm{s_{out}}, \mathrm{tmp}) \cdot \mathrm{stack(tmp)} \cdot [\mathrm{tmp_{both}}] && \text{(output unbinding from both)}\\ & - \mathrm{pol\_displ} \cdot [\mathrm{tmp_{both}}] && \text{(polymerase binding to both)}\\ & + \mathrm{nick} \cdot [\mathrm{tmp_{ext}}] && \text{(nicking enzyme binding to ext)}\end{aligned}$$

Effective polymerase activity:

$$\mathrm{pol} = \frac{\mathrm{polV}([\mathrm{polymerase}], \mathrm{T})}{\mathrm{polK(T)} \cdot \left(1 + \sum_{\mathrm{tmp}} \frac{[\mathrm{tmp_{in}}]}{\mathrm{polK(T)}} + \sum_{\mathrm{d}} \frac{[\mathrm{d_{in}}]}{\mathrm{polK(T)}} + \sum_{\mathrm{tmp}} \frac{[\mathrm{tmp_{both}}]}{\mathrm{polK\_displ(T)}}\right)}$$

Enzymatic kinetic constants

Table 1 Table of the enzymatic kinetic constants used in PENSim, along with the temperature they have been measured at. (u.a.) means $\mathrm{nM} \cdot \mathrm{min}^{-1} \cdot (\mathrm{units/mL})^{-1}$.

Enzymes	k_{cat}	K (nM)	Reference Temperature
pol	41.1 (u.a.)	80	42°C
pol_displ	8.2 (u.a.)	5.5	42°C
exo	6 min^{-1}	440	42°C
Nt.BstNBI	1.8 (u.a.)	30	42°C
Nb.BsmI	0.19 (u.a.)	9	45°C