Politecnico di Torino

Politecnico di Torino

Mathematical Engineering

A.y. 2025/2025

Graduation Session March-April 2026

A Statistical Framework for Protein Degradation Analysis in Cell-Free Systems

Supervisors:
Enrico Bibbona
Taishi Tonooka

Candidate:
Silvia Piatino

Table of Contents

Chapter 1

Introduction

1.1 Molecular Robotics and Cell-Free Systems

Molecular robotics is an emerging research field that studies the design of functional systems operating at the molecular scale. In analogy with macroscopic robots, such systems are conceived as devices capable of sensing signals, processing information, and producing a controlled response. In molecular robotics these functions are implemented through biological components and networks of biochemical reactions, allowing the construction of programmable systems capable of reproducing, in simplified form, some typical behaviors of cellular systems.

In this context, a central role is played by **artificial cells**, biochemical systems designed to imitate specific functions of living cells in a controlled environment. Artificial cells represent simplified models of natural biological systems and make it possible to study complex processes, such as the regulation of gene expression or the response to external stimuli, while maintaining a high degree of experimental control.

Beyond their relevance as models for studying fundamental biological processes, artificial cells also represent a promising platform for numerous applications in the biomedical and biotechnological fields. Thanks to the possibility of programming their functions through synthetic genetic circuits, these systems can be designed to perform specific tasks in response to particular chemical or biological stimuli. Possible applications include, for example, intelligent drug delivery systems, in which programmable artificial cells could release therapeutic molecules only in the presence of specific biological signals. Another area of interest is the development of biosensors capable of detecting target molecules or particular environmental conditions and converting these signals into a measurable response, such as the production of a fluorescent protein.

One of the most widely used platforms for the study and design of such systems is represented by **cell-free gene expression systems**. These systems make it possible to reconstruct gene expression processes outside living cells. In particular, the molecular machinery responsible for transcription and translation is extracted from cells and used *in vitro* to synthesize proteins in a controlled biochemical environment.

Cell-free systems are generally based on cellular extracts, often derived from *Escherichia coli*, which contain ribosomes, RNA polymerases, translation factors, and metabolic enzymes required for protein synthesis. When these extracts are supplemented with an energy mixture and with a DNA template encoding the protein of interest, transcription–translation reactions can occur directly in solution, enabling rapid protein production and the direct observation of the system dynamics [1].

Compared with systems based on living cells, cell-free platforms present several experimental advantages. Since the reactions take place in an open environment, it is possible to precisely control the composition of the system and modulate the concentration of individual components. Moreover, the absence of cell growth processes and the complex regulatory networks present in living cells simplifies the analysis of biochemical processes and allows experiments to be performed on significantly shorter timescales. For these reasons, cell-free systems are widely used in synthetic biology as platforms for studying and *prototyping* genetic circuits.

However, conventional cell-free expression systems present an important limitation. In natural cellular systems, protein levels are dynamically regulated by the balance between synthesis and degradation. In cell-free systems, instead, proteins produced during the reaction tend to accumulate over time, since the mechanisms responsible for protein turnover are often absent or strongly reduced.

This lack of degradation limits the ability of cell-free systems to reproduce the dynamic behavior observed in living cells, where protein concentrations are continuously modulated by processes of production and removal. For this reason, the possibility of introducing and controlling mechanisms of **protein degradation** within cell-free systems represents a key element for the realization of more realistic synthetic systems and for the implementation of dynamic genetic circuits.

1.2 Protein Degradation in Cell-Free Systems

In natural biological systems, the regulation of gene expression results from the interaction of numerous molecular processes acting in a coordinated manner within the cell. Among these processes, a central role is played by the dynamic balance between protein synthesis and protein degradation. The concentration of a protein over time does not depend solely on its production, but rather on the balance between the processes that generate it and those responsible for its removal. This equilibrium allows cells to modulate protein levels, respond to environmental variations, and coordinate complex regulatory networks.

Protein degradation therefore plays a fundamental role in the temporal control of gene regulatory networks. Through the selective removal of specific proteins, cells can limit the duration of biological responses, prevent the accumulation of unnecessary protein products, and regulate the dynamics of intracellular processes. In many genetic circuits, the temporal evolution of protein concentrations is determined precisely by the interplay between synthesis and degradation, which together define the overall dynamics of the system.

Reproducing such mechanisms in synthetic systems represents a major challenge in the fields of synthetic biology and molecular robotics. In particular, the realization of artificial genetic circuits requires the ability to precisely control both the production and the removal of proteins. In this context, *cell-free* gene expression systems provide an extremely useful experimental platform, as they allow gene expression processes to be reconstructed outside living cells while enabling direct manipulation of the biochemical composition of the system.

As discussed in the previous section, *cell-free* systems make it possible to reproduce protein synthesis relatively easily thanks to the presence, within cellular extracts, of the molecular machinery responsible for transcription and translation. However, in conventional systems protein degradation is often absent or strongly reduced. As a consequence, proteins produced during the reaction tend to accumulate over time, limiting the ability of the system to reproduce the dynamic behaviour observed in natural biological systems.

To introduce protein degradation mechanisms into *cell-free* systems, several studies have employed specific proteolytic systems derived from bacterial organisms. One of the most widely used systems is the **ClpXP** proteolytic complex, originally found in the bacterium *Escherichia coli.* This complex belongs to the family of AAA+ proteases and consists of two main components that perform complementary roles in the degradation process.

The protein **ClpX** is an ATPase that recognizes specific degradation sequences present in target proteins. Once bound to the substrate protein, ClpX uses the energy derived from ATP hydrolysis to unfold the protein structure and translocate the polypeptide chain toward the catalytic compartment of the protease. The **ClpP** component forms a cylindrical structure composed of multiple subunits that enclose an internal proteolytic chamber. Inside this chamber, the unfolded protein is degraded into smaller peptide fragments through the catalytic activity of the protease.
In order for a protein to be recognized by this degradation system, a short peptide sequence known as the **ssrA tag** can be engineered at the end of the protein. This sequence acts as a recognition signal for ClpX and allows the degradation process to be selectively directed toward specific proteins produced in the *cell-free* system. In experimental setups, this approach is often used in combination with fluorescent reporter proteins, such as variants of the **green fluorescent protein (GFP)**. The fluorescence emitted by the reporter protein enables the concentration of the protein to be monitored over time, allowing the activity of the degradation process to be indirectly observed.

The efficiency of protein degradation mediated by the ClpXP system depends strongly on the biochemical conditions of the *cell-free* reaction. Since ClpX requires energy derived from the hydrolysis of **ATP** to unfold and translocate target proteins, the availability of ATP represents a crucial factor for the functioning of the system. The concentration of $\mathbf{Mg^{2+}}$ ions can also influence the activity of the proteolytic complex, as these ions participate in numerous biochemical reactions and are required for the proper functioning of many proteins involved in gene expression processes.

In addition to the components directly involved in enzymatic activity, the molecular environment of the reaction can significantly affect the behaviour of the system. *Cell-free* reactions often include molecular crowding agents such as **polyethylene glycol (PEG)**, which help recreate physicochemical conditions closer to those found inside living cells. These molecules can influence protein diffusion, protein stability, and the efficiency of enzyme–substrate interactions, thereby affecting the overall dynamics of the biochemical reactions.

The simultaneous presence of these different factors makes the behaviour of the degradation system highly dependent on the experimental conditions. Variations in the concentrations of components such as ClpX, ClpP, ATP, Mg^{2+}, or PEG can influence the rate of the proteolytic process, the efficiency with which proteins are degraded, and the total amount of protein removed from the system over time. Consequently, even when focusing only on the degradation process, the overall behaviour of the system depends on the interaction between multiple biochemical parameters.

Fully characterizing all the interactions among the components of the system represents an extremely complex problem. For this reason, studies of *cell-free* systems often focus on identifying experimental conditions that allow desired behaviours of the system to be achieved.

In the context of protein degradation, this means determining conditions under which the degradation process can occur efficiently. A natural way to quantify the efficiency of the process is through the *degradation rate*, which describes how quickly the concentration of the target protein decreases over time. Identifying experimental conditions that lead to high degradation rates therefore represents an important step toward the development of more controllable *cell-free* platforms and toward the realization of synthetic systems capable of reproducing, at least in part, the dynamic regulation of protein levels observed in natural biological systems.

1.3 Research Objective

As discussed in the previous section, the efficiency of protein degradation in a *cell-free* system can be naturally quantified in terms of the *degradation rate*, which describes how rapidly the concentration of the target protein decreases over time. Achieving high degradation rates is therefore a key requirement for obtaining an efficient degradation process and, more generally, for enabling the dynamic control of protein levels in synthetic biological systems.

The objective of this thesis is to identify experimental conditions that enable the most efficient possible protein degradation in a *cell-free* system. In particular, the goal is to determine a combination of the concentrations of the main components of the system — including **ClpX**, **ClpP**, **ATP**, **Mg^{2+}** and **PEG** — that maximizes the **degradation rate** of the target protein. The space of possible parameter combinations is, however, large, and a systematic exploration of all experimental configurations would require a very high number of experiments. Moreover, experimental measurements are costly both in terms of time and resources. Each experimental condition requires the preparation of a specific *cell-free* reaction mixture, including the preparation and handling of several biochemical components, as well as the execution of degradation assays and the measurement of the resulting degradation dynamics. In particular, the use of reagents such as **ClpX** represents a significant experimental cost, making extensive experimental campaigns impractical.

For this reason, in addition to identifying conditions that promote efficient protein degradation, an important aspect of the problem addressed in this work is to limit the number of experiments required to discover such conditions. To tackle this challenge, a **computational–experimental pipeline** tailored to the studied system is developed, with the aim of guiding the exploration of the parameter space and identifying promising combinations while limiting the total number of experiments required.

1.4 Structure of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 describes the experimental framework and the components of the *cell-free* system used in the study, including the biochemical setup employed to investigate protein degradation.

Chapter 3 presents a qualitative analysis of the preliminary experimental observations and discusses the initial experimental design used to explore the behaviour of the system.

Chapter 4 introduces the mathematical model developed to describe the degradation dynamics and to relate the experimental conditions to the observed degradation rates.

Chapter 5 describes the inference methodology used to estimate the model parameters from experimental data, including the implementation based on **Bayesian Inference** and **Active Learning strategy**.

Chapter 6 presents the computational–experimental pipeline developed in this work, including the active learning strategy used to guide the selection of new experiments and efficiently explore the space of experimental conditions.

Chapter 7 discusses the results obtained from the modelling and inference procedures and analyzes the experimental conditions identified as promising for efficient protein degradation.

Finally, Chapter 8 summarizes the main conclusions of the work and outlines possible directions for future research.

Chapter 2

Experimental Framework

This chapter describes the experimental framework underlying the data analyzed in this work. In particular, it presents the experimental setup used to study protein degradation in a *cell-free* system, the preparation of the reaction mixtures, and the experimental design used to explore the parameter space of the degradation process. The procedures used to measure fluorescence signals and the practical constraints associated with the experimental workflow are also discussed. Finally, a preliminary analysis of the fluorescence trajectories is presented in order to characterize the initial transient phase observed in the measurements.

2.1 Experimental Setup and Sample Preparation

The experimental measurements analyzed in this work were carried out in collaboration with Yusei Hattori at the Biomicrosystems Laboratory of the Kyoto Institute of Technology. The experiments aim to quantify the dynamics of protein degradation in a cell-free system where the ClpXP protease complex is reconstituted in vitro.

Each experiment is performed in a 384-well microplate, where every well contains a cell-free reaction mixture with a total volume of approximately $10\,\mu L$. The samples are prepared directly in the wells by dispensing the reaction components using a combination of manual pipetting and automated nanodispensing.

The biochemical components are added to the wells in the following order:

- **Biochemical system components dispensed manually**
  - **Autolysate**: cellular extract of *E. coli* containing enzymes and other cellular components;
  - **Premix**: mixture of nucleotides, amino acids and cofactors required for the activity of the cell-free system;
  - **PEG-8000**: polyethylene glycol, a neutral water-soluble polymer composed of repeating ethylene oxide units, acting as a molecular crowding agent. $(50w/v\%)$
- **Biochemical system components dispensed using the I.DOT nanodispensing system**
  - **DI water**: deionized water used to adjust the final reaction volume;
  - **Elution buffer**: buffer solution used to store purified proteins and maintain appropriate pH and ionic conditions;

- **ATP**: adenosine triphosphate, the energy source required for the ATPase activity of ClpX (stock solution 400 mM);
- **Mg-glutamate**: magnesium ions acting as essential cofactors for ATP-dependent enzymatic reactions (stock solution 400 mM);
- **Clp buffer**: buffer solution maintaining appropriate biochemical conditions for the ClpXP degradation system;
- **ClpX**: purified ClpX ATPase (472 μg/mL);
- **ClpP**: purified ClpP protease (7800 μg/mL);
- **sfGFP-ssrA**: fluorescent reporter protein carrying the ssrA degradation tag (stock solution 150 μM).

Autolysate, PEG-8000 and premix are dispensed manually, while the remaining reagents are transferred using the I.DOT nanodispensing system, which enables precise delivery of very small liquid volumes across the microplate. The fluorescent protein sfGFP-ssrA is introduced as the final component in order to minimize premature degradation before the beginning of the fluorescence measurement.

2.2 Experimental Design

The experimental design is defined by varying the concentrations of ATP, Mg, ClpX, ClpP and PEG across wells, while the remaining components are kept constant. Considering the concentration levels reported in Table 2.1, the full combinatorial design space contains a total of 4704 possible parameter combinations (excluding control conditions in which the fluorescent protein is absent). An extreamely big space to be explored experimentally.

Table 2.1: Concentration levels explored in the experimental design space.

Component	Unit	Tested concentration levels
sfGFP-ssrA*	μM	0, 1.5
ATP	mM	0, 1, 2, 4, 5.7, 8
Mg (Mg-glutamate)	mM	6, 7.5, 10, 14
ClpX	nM	0, 50, 100, 150, 200, 300, 400
ClpP	nM	0, 50, 100, 150, 200, 300, 400
PEG	% (w/v)	3, 4, 5, 6

* The fluorescent protein sfGFP-ssrA is included in the reaction mixture at a fixed concentration of 1.5 μM in all degradation experiments. In a small number of control wells the protein is omitted (0 μM) in order to measure background fluorescence; these wells will be referred to as blanks.

2.3 Fluorescence Measurement

Once the reaction mixtures are prepared, the microplate is placed in a fluorescence plate reader (Infinite Nano+ Tecan), which records the fluorescence signal over time. The instrument excites GFP at a wavelength of 485 nm and measures the emitted fluorescence at 535 nm while maintaining the plate at controlled temperature. Fluorescence measurements are acquired at regular time intervals during the experiment, producing a time series for each well.

The emitted light from fluorescent molecules passes through the optical system of the instrument and is detected by a camera. The detected photons are converted into a digital signal representing the fluorescence intensity. This signal is expressed in arbitrary units (AU), also referred to as intensity counts. The intensity counts are proportional to the number of photons captured by the detector and therefore provide a quantitative proxy for the amount of fluorescent protein present in the sample.

Each well therefore produces a fluorescence trajectory describing the evolution of the GFP signal over time. Since fluorescence intensity is approximately proportional to the concentration of the fluorescent protein, these trajectories provide an indirect measurement of the degradation dynamics of GFP-ssrA under different biochemical conditions.

2.4 Experimental Constraints

The experimental workflow is subject to several practical constraints that limit the number of experiments that can be performed within a single experimental campaign. Preparing the reaction mixtures requires multiple manual and automated dispensing steps, and each experimental run involves the preparation of the microplate, the execution of the degradation assay, and the acquisition of fluorescence measurements over time. As a result, experimental sessions are both time-consuming and resource-intensive.

A particularly important constraint arises from the preparation and stability of the **ClpX** protein. The purification of ClpX is experimentally demanding and relatively costly, and once prepared the protein gradually loses activity over time. For this reason, experiments are typically performed using a single freshly prepared batch of ClpX in order to avoid variability in enzymatic activity between different experimental sessions.

This requirement implies that experiments associated with the same batch of ClpX should be performed within a limited time window, so that biochemical conditions remain as consistent as possible across wells. Consequently, the number of reaction mixtures prepared in a single session must be limited, and the time interval between the preparation of the first and the last well should be minimized.

These constraints make it impractical to exhaustively explore all possible combinations of the concentration levels reported in Table 2.1. Instead, only a subset of the available wells of the 384-well plate is used in each experimental run. As a consequence, the overall number of experiments that can be performed is limited, motivating the need for strategies that allow the exploration of the experimental design space in an efficient manner.

Before proceeding with the quantitative data analysis, it is therefore useful to perform a preliminary inspection of the fluorescence trajectories in order to identify systematic effects present in the measurements.

2.5 Initial Transient Phase

Although each well is prepared with the same initial concentration of fluorescent protein sfGFP-ssrA, the fluorescence measured at time t_0 is not identical across wells. This variability arises from several experimental factors, the most relevant being the **inhomogeneity** within the well at the time of the first measurement.

According to the experimental procedure, the purified GFP-ssrA stored in Elution Buffer is the last component dispensed by the I.DOT system in order to minimize premature degradation. For the same reason, no mixing is performed before the first fluorescence acquisition. As a consequence, at the time of the initial measurement with the *Infinite F Nano+* plate reader (Tecan), the protein is not yet uniformly distributed throughout the well volume. The dispensed droplet initially remains localized near the liquid surface, generating a local concentration gradient.

Since the plate reader detects fluorescence mainly from a defined central optical region of the well, with higher sensitivity toward the upper layers of the liquid, the signal recorded at t_0 reflects the *local* concentration within the interrogated region rather than the average concentration in the total volume. Small differences in droplet positioning, surface geometry, or early diffusion therefore lead to differences in the measured initial fluorescence, even when the nominal concentration is the same.

Additional variability arises from dispensing uncertainty and instrumental noise. The I.DOT nanodispensing system exhibits a coefficient of variation (CV) of approximately 8%, meaning that the dispensed volume may deviate from its nominal value by about 8%. Because the initial sfGFP concentration in the well is proportional to the dispensed volume (0.10 μL to obtain 1.5 μM), this volumetric variability directly translates into variability in initial concentration and consequently in measured fluorescence. Instrumental noise from the plate reader, due to electronic fluctuations, excitation source variability, and optical background, further contributes to the observed dispersion.

This initial spatial heterogeneity explains two characteristic features observed in the fluorescence curves. First, it accounts for the non-uniform initial fluorescence peak, which differs from well to well depending on the local protein distribution at the time of measurement. Second, it explains the moderate early decrease in fluorescence that is observed even in samples where no enzymatic degradation is expected.

Immediately after dispensing, the protein is concentrated near the surface in a localized region. During subsequent shaking and iterative readings, mixing and diffusion progressively homogenize the sample. As long as the distribution remains non-uniform, the detected signal reflects local concentration effects. The observed early decrease therefore corresponds to the transition from a heterogeneous initial state to a spatially homogeneous steady state rather than to actual protein degradation.

This effect is particularly evident in samples lacking active degradation (e.g., in the absence of ClpX or ATP), where the expected kinetic profile would otherwise be approximately flat. In such cases, the early moderate decline followed by a stable plateau reflects physical and optical equilibration of the system rather than protein consumption. In samples where degradation is active, this transient phase is rapidly dominated by the exponential enzymatic decay, making the heterogeneity-induced effect less apparent.

For this reason, the first 50 minutes of acquisition, dominated by mixing and stabilization effects not described by the exponential kinetic model, were excluded from the parameter estimation procedure.

Chapter 3

Preliminary Analysis of Experimental Data

3.1 Structure of the Experimental Data

As discussed in Sec. 2.2, the experimental design space defined by the possible combinations of ATP, Mg, ClpX, ClpP and PEG concentrations contains several thousand feasible configurations. Exploring this space exhaustively through experiments would therefore require a very large number of measurements and is not practical given the experimental constraints described in the previous chapter.

Instead, the experimental exploration is performed iteratively. At each iteration, only a limited number of reaction compositions are experimentally tested. The information obtained from these measurements is then used to guide the selection of subsequent experiments, progressively improving our understanding of the system. The details of this iterative strategy will be described in later chapters. At this stage, it is sufficient to introduce the structure of the experimental batches used throughout the study.

The general organization of the experimental batches is inspired by the Active Learning workflow proposed by Borkowski et al. in the work *"Large-scale active-learning-guided exploration for in vitro protein production optimization"* [2]. In that study, each iteration evaluates a batch with the following features:

- 102 distinct reaction compositions are examined;
- each specific composition is measured in triplicate to monitor experimental variability;
- 13 control compositions are included, measured in triplicate as well.

This corresponds to more than 300 experimental samples per iteration, allowing the authors to exploit almost the entire capacity of a 384-well plate. Such a large batch size is motivated by the enormous size of the design space considered in the reference study, which exceeds 4 million possible combinations of reaction conditions.

In the present work, the same general principles are retained but adapted to a much smaller and more constrained experimental setting. In particular, our design space is around 1000 smaller and the experimental protocol imposes additional practical limitations that prevent the full utilization of the microplate capacity.

For this reason, each experimental batch consists of only 12 distinct reaction compositions, each measured in **triplicate**. Although the experiments are performed using a 384-well plate, only two rows are used in each experimental run, corresponding to a total of 48 wells. This limitation arises from the experimental protocol: some components of the reaction mixture are dispensed manually, while the reporter protein is added sequentially using the I.DOT nanodispensing system. To avoid introducing variability due to excessive delays in plate preparation, the time interval between the preparation of the first and the last well must remain limited.

Within these 48 wells, 36 are used to evaluate the 12 candidate reaction compositions measured in triplicate. The remaining 12 wells are reserved for **control compositions**.

Similarly to the reference study, control samples are systematically included in each experimental batch. 4 control compositions are used: two contain the fluorescent reporter protein, while the other two correspond to the same biochemical compositions but without the fluorescent protein. Each control composition is measured in triplicate.

These control wells serve several purposes. First, they provide a consistency check on the experimental procedure, allowing possible anomalies in reagent dispensing or fluorescence acquisition to be detected. Second, the reference trajectories obtained from the control wells containing the fluorescent protein are later used to estimate day-dependent batch effects and to make measurements collected on different experimental days more comparable. Finally, the wells lacking the fluorescent protein allow the background fluorescence signal of the measurement system to be quantified.

Summarily, in our work, the batch set at each iteration is composed of:

- 12 Distinct new design compositions;
- 4 Control compositions, fixed across iterates, divided into:
  - 2 **Reference** compositions: with specific stable design;
  - 2 **Blank** compositions: reference compositions but without protein sfGFP-ssrA to be degraded.

Control samples are treated with more details below:

3.1.1 Reference compositions

As described in Sec. 3.1, each experimental plate includes a set of control reactions used to monitor the consistency of the experimental measurements. Among these controls, two reference reactions contain the fluorescent reporter protein *sfGFP-ssrA*. These reference compositions serve as internal benchmarks that allow the behaviour of the system to be compared across different experimental days.

Two reference compositions, denoted as **RefA** and **RefB**, are included in every plate. Their biochemical compositions are reported in Table 3.1.

Table 3.1: Reference compositions used as experimental controls in every plate.

Component	RefA	RefB
ATP (mM)	4	2
Mg (mM)	10	10
ClpX (nM)	300	400
ClpP (nM)	300	400
sfGFP-ssrA (μM)	1.5	1.5
PEG (%)	5	5

These compositions were selected in order to represent stable operating conditions within the design space. In particular, the concentrations were chosen to lie in the interior of the admissible region rather than at its boundaries, thereby avoiding extreme conditions that could lead to unstable experimental behaviour.

For consistency, the concentrations of ClpX and ClpP were chosen to be equal within each reference composition. Preliminary experiments conducted by Hattori [1] indicated that relatively high concentrations of ClpX combined with non-zero ATP levels typically produce a clear and reproducible degradation signal. For this reason, the reference designs were selected in a region of the design space where protein degradation is expected to occur reliably.

The main purpose of these reference reactions is to verify that the experimental conditions of a given day are consistent with those observed in previous experiments. In practice, the fluorescence trajectories of the references are inspected after each experimental run. If the reference curves deviate significantly from the behaviour observed in earlier measurements, this may indicate potential problems in the experimental preparation, such as inaccuracies in reagent dispensing or anomalies in fluorescence acquisition.

In addition to these reference reactions, corresponding control wells without the fluorescent protein, named as "blank" are also included in each experimental plate. These wells will be used to estimate the background fluorescence of the system, as described in the following section.

3.1.2 Blank samples

In addition to the reference reactions introduced in Sec. 3.1.1, each experimental plate also includes blank samples. In our experiments, blanks correspond to reaction mixtures that contain all components of the cell-free system except the fluorescent reporter protein (sfGFP). They therefore represent the intrinsic fluorescence of the system in the absence of the reporter protein and provide an empirical measurement of the background signal.

The fluorescence measured in these blanks originates from several non-specific sources, including intrinsic fluorescence of buffer components, plate autofluorescence, instrumental drift, and other experimental factors that may vary across experimental days and plates. For this reason, the blank signal captures a day-specific background contribution that cannot be attributed to sfGFP and constitutes an important source of inter-day variability.

Blank samples associated with the reference conditions (RefA and RefB) are used to monitor and correct this background effect throughout the experiment. Because a composition-matched blank (i.e., an identical mixture without protein) is not available for every individual design, the reference blanks are used as a proxy for the background fluorescence of all wells measured on the same experimental day. Importantly, this approach allows the background to be treated as a time-dependent signal rather than as a constant additive offset.

Including a composition-matched blank for each individual design would effectively double the number of experimental conditions. Given the objective of exploring the design space while maintaining a limited number of experiments, such a strategy would substantially reduce the number of informative measurements that could be collected. The use of reference blanks therefore represents a practical compromise between experimental feasibility and background correction accuracy.

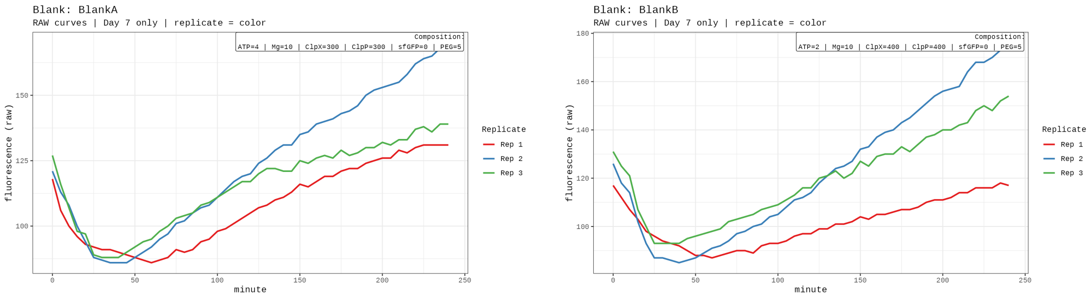
As shown in Fig. 3.1, the blank fluorescence exhibits a clearly non-constant temporal behaviour, reflecting the dynamic nature of the background contribution. Although the overall magnitude of the signal remains relatively low throughout the experiment (typically below approximately 300 arbitrary units), it displays a structured temporal pattern that cannot be attributed to random measurement noise alone.

During the first $\sim$ 30–60 minutes, a pronounced decrease is consistently observed across replicates.

A minimum is typically reached around $\sim$ 50–70 minutes, after which a slow monotonic increase becomes visible. As discussed in Section 2.5, the initial decrease is likely associated with a transient phase during which the reaction mixture is not yet fully equilibrated. Incomplete mixing, temperature stabilization, or photophysical adaptation effects may therefore lead to unstable fluorescence measurements in the early stage of the experiment.

The subsequent gradual increase observed at later time points does not have a definitive mechanistic explanation. Instrumental factors such as detector drift, slow thermal equilibration of the plate reader, or cumulative optical effects due to repeated excitation may contribute to a time-dependent background component. Biological contributions cannot be excluded either, as slow physicochemical changes within the cell-free extract may alter the optical properties of the reaction mixture over time.

Despite its relatively small amplitude compared with the reporter signal, the background fluorescence is not negligible when estimating kinetic parameters, particularly at early time points or in low-signal regimes. For this reason, a time-dependent blank correction is necessary in order to prevent systematic bias in the quantitative analysis of protein degradation dynamics.



(a) Blank A

(b) Blank B

Figure 3.1: Fluorescence trajectories for blank reactions measured on Day 7. The choice of Day 7 is purely illustrative, as the same qualitative behaviour is observed across all experimental days. Colors indicate replicates, and the composition of each blank is reported within the corresponding panel.

Now the initial Batch set is presented and analyzed from a qualitative point of view.

3.1.3 Initial Experimental Designs

The first set of experimental measurements consists of an initial batch of 12 candidate reaction compositions. Each composition is measured in triplicate according to the experimental batch structure described in Sec. 3.1.

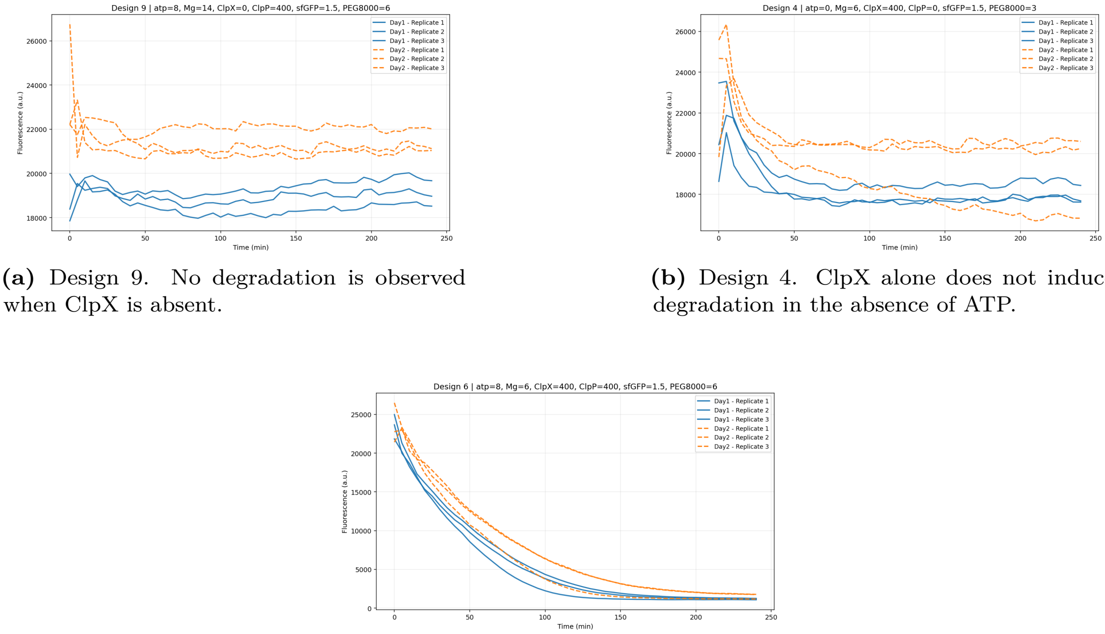
The selection of these initial designs follows a strategy inspired by the Active Learning study of Borkowski et al. [2]. In that work, a subset of compositions is deterministically selected in order to probe extreme regions of the design space before the iterative exploration begins.

Following the same principle, the initial designs used in this study correspond to combinations located at the boundaries of the design space. In practice, this means that individual components are set to their minimum or maximum concentration levels while the remaining variables are fixed at the opposite edge. The goal of this choice is to generate an initial set of experiments that span qualitatively different regimes of the system, despite the limited number of available measurements.

The compositions of the 12 initial designs are reported in Table 3.2.

Table 3.2: Initial experimental designs used to construct the starting dataset for the Active Learning procedure. Each design corresponds to a specific combination of biochemical component concentrations in the reaction mixture.

Design	ATP (mM)	Mg (mM)	ClpX (nM)	ClpP (nM)	PEG (%)	sfGFP-ssrA (μM)
1	0	14	0	0	3	1.5
2	0	6	0	0	6	1.5
3	8	6	0	0	3	1.5
4	0	6	400	0	3	1.5
5	0	6	0	400	3	1.5
6	8	6	400	400	6	1.5
7	8	14	400	400	3	1.5
8	0	14	400	400	6	1.5
9	8	14	0	400	6	1.5
10	8	14	400	0	6	1.5
11	0	6	0	0	3	1.5
12	8	14	400	400	6	1.5



(a) Design 9. No degradation is observed when ClpX is absent.

(b) Design 4. ClpX alone does not induce degradation in the absence of ATP.

(c) Design 6. Clear exponential decay indicating active ClpXP degradation.

Figure 3.2: Representative fluorescence trajectories for selected designs from the initial dataset measured on two experimental days. Each panel shows the three technical replicates recorded on Day 1 and Day 2 for the same reaction composition. The plots highlight the qualitative differences between conditions where degradation is inactive and conditions where the ClpXP system is active.

The initial set of designs was evaluated over two experimental days in order to obtain a first qualitative assessment of the degradation dynamics and to investigate the presence of possible day-dependent batch effects.

Inspection of the fluorescence trajectories associated with these designs already reveals several qualitative patterns in the degradation behaviour of the system (see Fig. 3.2).

- **Presence of two distinct dynamical regimes.** The observed fluorescence trajectories can be broadly divided into two categories. Some designs exhibit approximately flat curves, indicating that no significant degradation of the fluorescent protein occurs during the experiment. In contrast, other designs show a clear exponential decay of the fluorescence signal, consistent with active degradation mediated by the ClpXP system.
- **Central role of ClpX in enabling degradation.** Across the tested designs, the presence of ClpX appears to be the main discriminating factor between these two regimes. When ClpX is absent, the fluorescence trajectories remain essentially flat even when the concentrations of the other components are varied. This suggests that the unfoldase activity provided by ClpX is necessary for initiating the degradation process.
- **ATP dependence of the degradation process.** Even when ClpX is present at high concentration, the degradation dynamics remain very weak in the absence of ATP. This behavior is consistent with the ATP-dependent mechanism of the ClpXP system, where ATP hydrolysis provides the energy required for substrate unfolding and translocation.
- **Emergence of exponential degradation profiles.** When both ClpX and ATP are present, several designs exhibit trajectories characterized by a clear exponential decay. This behavior supports the modeling assumption adopted later in the analysis, where the degradation dynamics are approximated through an exponential kinetic model.

Finally, a qualitative discussion on batch effect is presented below

3.2 Qualitative analysis of the batch effect

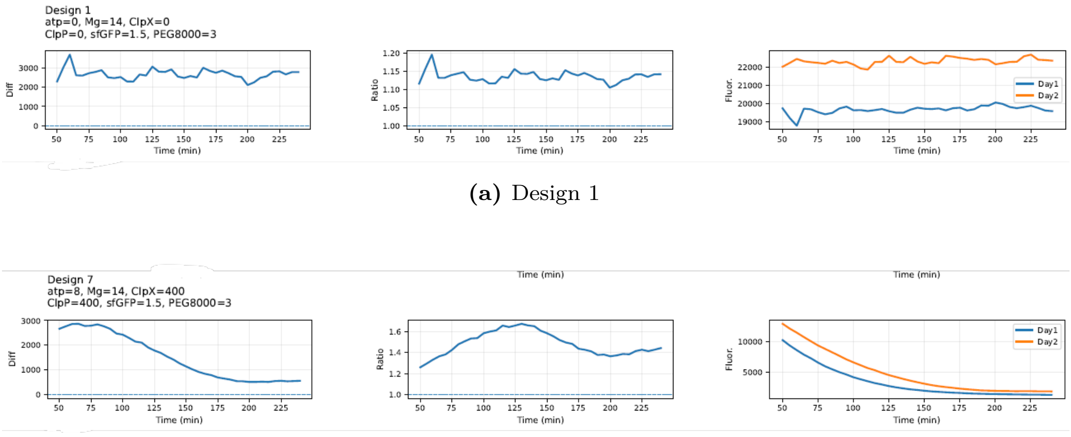
In experimental studies based on fluorescence measurements, variability between repeated experiments can arise from several sources related to both biological and technical factors. Even when the same experimental protocol is followed, measurements collected on different days or in different plates may exhibit systematic differences in signal intensity or degradation dynamics. These differences are commonly referred to as *batch effects*.

In the present experimental setup, two main types of variability can be distinguished. The first is *intra–day variability*, which arises from technical replicates performed within the same experimental session. This variability can be caused by factors such as pipetting inaccuracies, slight differences in well position within the plate, local temperature variations, or small fluctuations in fluorescence detection.

The second source is *inter–day variability*, which reflects differences between experiments performed on different days. In cell–free systems, this variability may originate from variations in reagent preparation, differences in the activity of enzymes such as ClpX and ClpP, changes in extract quality, or other experimental conditions that are difficult to control perfectly across separate experimental sessions.

In the modeling framework adopted in this work, the batch effect is primarily associated with this second source of variability, namely differences between experimental days. Replicate–level variability within the same day is instead treated as part of the residual experimental noise, while day–to–day differences are explicitly modeled as a systematic effect acting on the fluorescence trajectories. This choice reflects the empirical observation that trajectories collected on the same day tend to be relatively consistent, whereas more structured discrepancies are often observed between measurements performed on different days.

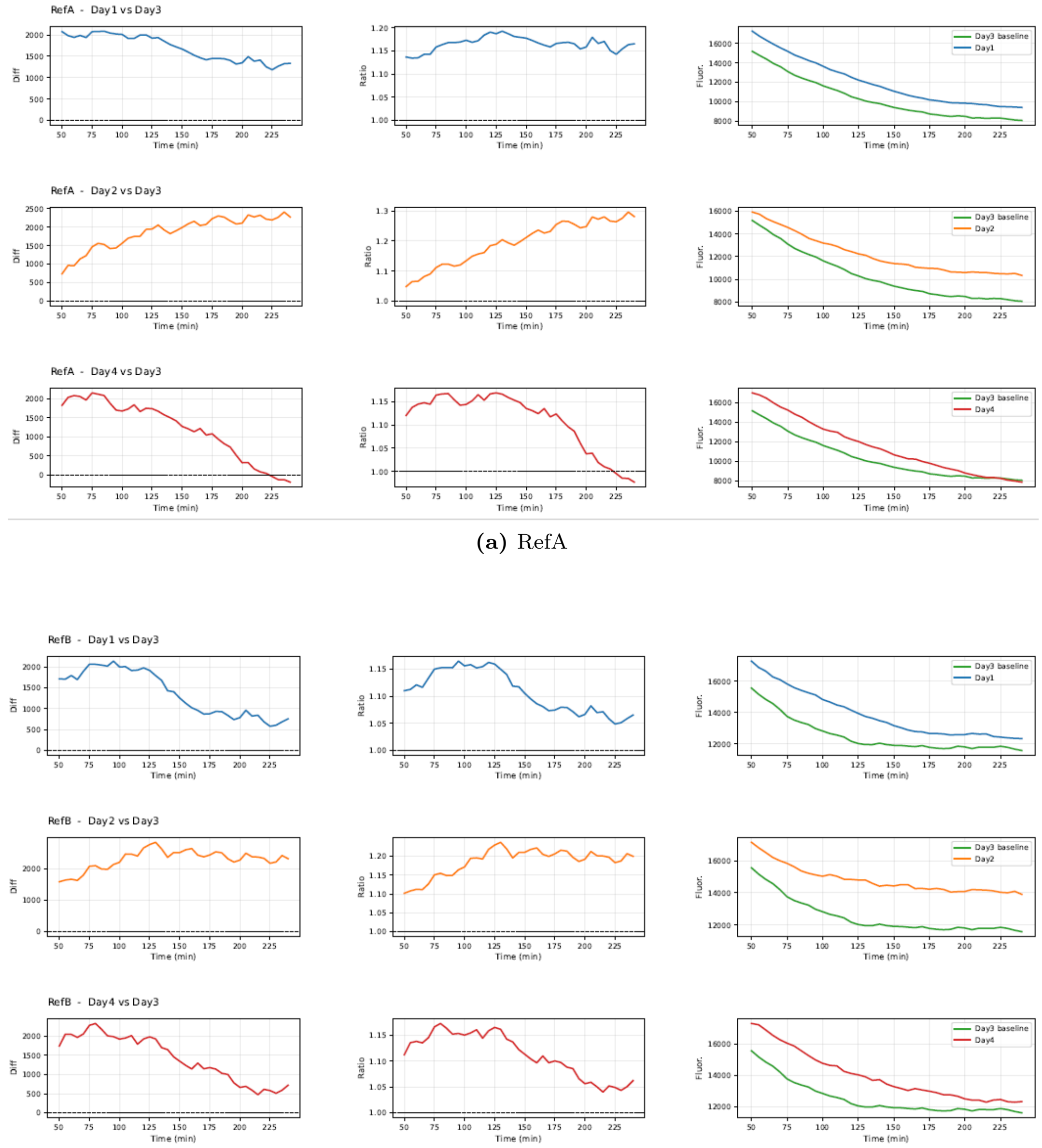
Understanding the qualitative structure of this day–dependent variability is therefore an important preliminary step before introducing the statistical model used for batch correction.
For this reason, an exploratory analysis of the fluorescence trajectories was carried out using the data collected in the initial experimental batch. In particular, for each design (including the reference conditions) the median fluorescence trajectory across replicates was computed for each experimental day.
The pointwise difference and ratio between these median curves were then calculated as functions of time and inspected in order to characterize how trajectories vary across experimental days. The corresponding plots are reported in Figures 3.3 and 3.4 and



(a) Design 1

(b) Design 7

Figure 3.3: Qualitative comparison of batch effects for two representative experimental designs. Design 1 shows nearly flat ratio and difference curves, consistent with negligible degradation dynamics, whereas Design 7 exhibits a stronger time-dependent batch effect, visible in both transformed trajectories and median curves.



(a) RefA

(b) RefB

Figure 3.4: Qualitative comparison of batch effects for the two reference conditions. Each row reports the pointwise difference, the ratio, and the corresponding median fluorescence trajectories across experimental days.

If batch variability were limited to a simple additive offset in fluorescence values or to a time–independent multiplicative scaling of the signal, these transformed curves would be expected to remain approximately constant over time.

However, the empirical analysis reveals that this is generally not the case. In many comparisons the ratio and difference curves exhibit clear time–dependent patterns. Ratios are flat only where original trajectories are. In the other cases, a unique pattern for ratios is not identifiable: a non monotonic behaviour is more frequent, with a bell-like profile but also increasing or decreasing trend is observable. A closer inspection of the reference trajectories reveals slightly different behaviours between the two reference designs. For the RefA condition, the ratio and difference curves appear somewhat irregular across time, making it difficult to identify a simple functional form describing the batch effect. In contrast, the RefB comparisons display a more structured behaviour: curves are more stable across days and this makes the second reference to look more reliable.

Another common feature highlighted by the plots is that both ratio and difference curves tend to stabilize at later time points. This behaviour reflects the fact that fluorescence trajectories progressively approach a plateau, where the signal becomes less sensitive to small variations in the degradation dynamics. As a result, batch differences tend to be more pronounced during the intermediate phase of the trajectory and less visible once the curves converge toward their final levels.

More generally, batch effects appear to be more evident in designs characterized by strong degradation dynamics, particularly those involving high concentrations of ClpX. In these cases the fluorescence signal decays rapidly, and even small variations between experimental days can produce noticeable deviations between trajectories during the transient phase of the curve.

Conversely, in designs where little or no degradation occurs, the fluorescence trajectories remain nearly flat over time and the corresponding ratio and difference curves are also approximately constant. In this situation the batch effect appears much less pronounced. A possible interpretation is that part of the day–to–day variability acts through mechanisms affecting the effective degradation rate. When the degradation process is essentially inactive, variations in this rate have little impact on the observed trajectories, resulting in minimal differences between experimental days.

Taken together, these observations suggest that the batch effect is not simply a constant perturbation of the fluorescence signal but is instead linked to the dynamical component of the trajectories.

Overall, these exploratory comparisons suggest that batch variability does not act as a simple constant perturbation of the fluorescence signal. Rather, it appears to be linked to the dynamical component of the trajectories, becoming more evident when degradation is active and diminishing when decay dynamics are weak. These observations motivate the modeling framework introduced in the following sections.

Chapter 4

Mathematical Model

4.1 Introduction

Based on the considerations discussed in Section 2.5, we restrict our analysis to time points satisfying $t \geq t_0 = 50$ minutes, in order to describe the dynamics of protein degradation in cell-free systems. However, the input vector $\mathbf{x}_j$ considered in the analysis is defined by components that are experimentally set at $t = 0$.

The objective of the present mathematical formulation is to construct a quantitative and reproducible description of the degradation dynamics that allows the extraction of a single kinetic descriptor for each design. In particular, for every design j, we aim to estimate a degradation rate parameter k_j summarizing the temporal evolution of fluorescence and, more specifically, the speed of the degradation process.

The logarithm of this scalar quantity will constitute the response variable used to build the dataset through the Active Learning (AL) procedure, whose goal is to identify combinations of initial concentrations of the cell-free buffer that maximize the protein degradation log-rate.

The proposed model is not intended to provide a fully mechanistic representation of the biochemical system. Instead, it serves as a parsimonious and effective approximation of the dominant behavior observed in the experimental data. Several simplifying assumptions are therefore introduced:

- The analysis is restricted to the post-transient regime ($t \geq t_0 = 50$ minutes), and the initial transitory phase is not modeled.
- The fluorescence decay is approximated by a single-exponential function characterized by a single degradation rate parameter k_j.
- The input vector is assumed to be deterministic.
- The baseline fluorescence level is assumed to be constant over time and is represented by a parameter b.
- Experimental measurements are assumed to be affected by independent Gaussian noise.
- The model aims at capturing the dominant trend of the degradation dynamics rather than providing a detailed mechanistic description of the underlying biochemical processes.
- Noise is additive in the model of observed data.
- No batch effect on the residual fluorescence at long time

These assumptions are motivated by preliminary inspection of the experimental trajectories. As shown in Fig. 4.1, the first set of tested designs exhibits two clearly distinguishable behaviors. Some designs display a pronounced monotonic decrease in fluorescence, consistent with exponential decay and substantial protein degradation. Other designs show nearly flat trajectories, indicating negligible degradation. In the latter case, the data can still be consistently represented within the same modeling framework by an exponential function with rate parameter close to zero.

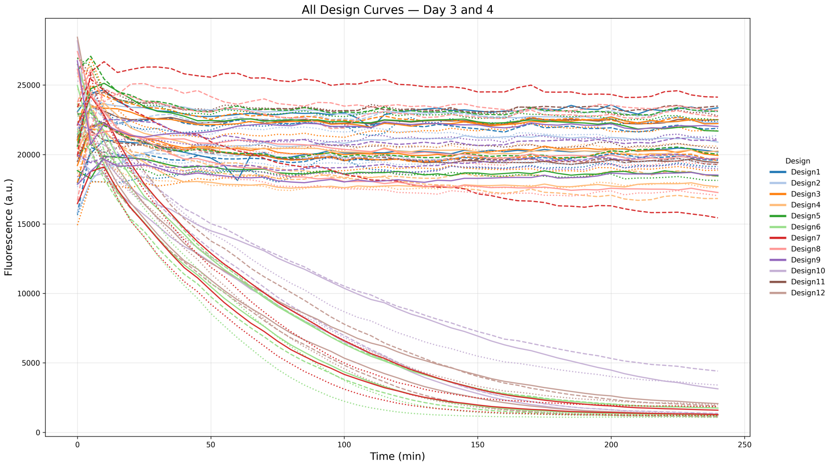


Figure 4.1: Representative fluorescence trajectories for designs exhibiting sporadic degradation failures. Colors indicate experimental day and transparency indicates replicate.

Although simplified, this formulation is sufficiently expressive to discriminate between high-degradation and low-degradation regimes, which is the primary requirement for constructing a reliable dataset for subsequent optimization.

4.2 Experimental observations and Statistical setup

In this section we introduce the statistical framework used to describe the experimental data and to connect the observed fluorescence trajectories to the experimental conditions defining each design. In particular, we identify the input variables of the system, the observed outputs, and the quantity of interest that the model aims to estimate.

4.2.1 Predictors

Each experimental configuration, hereafter referred to as a *design* and indexed by j, is characterized by a vector of covariates $\mathbf{x}_j \in \mathcal{X}$ whose components $x_{j,k}$ correspond to the concentrations of biochemical elements present in the cell-free reaction mixture, set experimentally at time $t = 0$. The design vector then, is defined as:

$$\mathbf{x}_j = (\mathrm{atp}_j^0,\ \mathrm{mg}_j^0,\ \mathrm{clpx}_j^0,\ \mathrm{clpp}_j^0,\ \mathrm{sfgfp}_j^0,\ \mathrm{peg}_j^0) \tag{4.1}$$

The vector $\mathbf{x}_j$ therefore encodes the experimental conditions defining design j. Each design corresponds to a point in the discrete design space $\mathcal{X}$, whose cardinality is $|\mathcal{X}| = 4704$.

4.2.2 Discretization of the design space

The biochemical variables defining each design correspond to the initial concentrations of several species in the cell–free reaction mixture. From a physical perspective, these concentrations are continuous quantities and can in principle take values in a subset of $\mathbb{R}^p_+$.
However, the experimental protocol does not explore the full continuous domain. Instead, a finite set of concentration levels is specified for each biochemical component. Let $\mathcal{G}_k$ denote the set of admissible levels for component k.
In the present study the grids are defined as

$$\mathcal{G}_{ATP} = \{0{,}1{,}2{,}4{,}5.7{,}8\}, \quad \mathcal{G}_{Mg} = \{6{,}7.5{,}10{,}14\}, \tag{4.2}$$

$$\mathcal{G}_{ClpX} = \mathcal{G}_{ClpP} = \{0{,}50{,}100{,}150{,}200{,}300{,}400\}, \quad \mathcal{G}_{PEG} = \{3{,}4{,}5{,}6\}. \tag{4.3}$$

The design space is therefore defined as the Cartesian product

$$\mathcal{X} = \mathcal{G}_{ATP} \times \mathcal{G}_{Mg} \times \mathcal{G}_{ClpX} \times \mathcal{G}_{ClpP} \times \mathcal{G}_{PEG}, \tag{4.4}$$

which forms a finite subset of $\mathbb{R}^5$.
Each design j is thus represented by a vector $\mathbf{x}_j \in \mathcal{X}$.
It is important to note that the exact value $x_{j,k}$ does not correspond exactly to the true physical concentration realized in the experiment. Due to pipetting variability and measurement uncertainty, the effective concentration can be interpreted as a random perturbation around the true value,

$$\tilde{x}_{j,k} = x_{j,k} + \varepsilon_{j,k}. \tag{4.5}$$

In practice, the uncertainty associated with the dispensing procedure is commonly characterized in terms of a coefficient of variation (CV) of the delivered volumes. In the experimental setup considered here, the relative variability of pipetting operations is approximately in the range $5\% - 8\%$.
Under this assumption, the perturbation $\varepsilon_{j,k}$ can be interpreted as a relative fluctuation around the true value, satisfying approximately

$$\mathrm{sd}(\varepsilon_{j,k}) \approx \mathrm{CV}_k \, x_{j,k}. \tag{4.6}$$

Consequently, the realizable concentrations associated with the nominal level $x_{j,k}$ lie approximately in the interval

$$I_{j,k} = [x_{j,k}(1 - 2\mathrm{CV}_k),\ x_{j,k}(1 + 2\mathrm{CV}_k)], \tag{4.7}$$

which corresponds roughly to a tolerance region of two standard deviations around the true value. Consequently, each discrete level represents in practice a small neighbourhood in the continuous concentration space. In other words, the design point $\mathbf{x}_j$ can be interpreted as a representative element of a region of the continuous domain corresponding to the experimental tolerance associated with the dispensing procedure.
This consideration was also taken into account when defining the discrete grids $\mathcal{G}_k$. In particular, the concentration levels were selected so that the uncertainty regions associated with different levels are sufficiently separated, thereby avoiding configurations that would be experimentally indistinguishable within the typical dispensing precision.
For modelling purposes, the covariates $\mathbf{x}_j$ are treated as deterministic inputs defined on the discrete design space $\mathcal{X}$. In this framework the uncertainty associated with the dispensing

procedure is not explicitly modelled at the covariate level; instead, its effect is incorporated into the stochastic component of the statistical model describing the observed fluorescence data.

4.2.3 Observed fluorescence trajectories

For each design j, identified by the input vector $\mathbf{x}_j$, fluorescence is monitored over time on one or more experimental days, with technical replicates collected for each day. Accordingly, each design j is associated with R_j fluorescence time series. Each of these consists of a sequence of fluorescence observations recorded at equally spaced time points during the experiment. More precisely, fluorescence measurements are collected every 5 minutes up to a final time of 240 minutes, so that

$$t_i \in \{0{,}5{,}10, \dots ,240\}. \tag{4.8}$$

Although the experimental acquisition starts at $t = 0$, the first part of the experiment is affected by transient mixing and stabilization effects that are not described by the kinetic model adopted in this work. For this reason, the subsequent statistical analysis will be restricted to the post-transient regime starting at

$$t_0 = 50 \text{ minutes}. \tag{4.9}$$

All subsequent modelling steps will therefore focus on the fluorescence dynamics observed for $t \geq t_0$.
Accordingly, each trajectory is represented by the sequence

$$y_{jdr}^{\text{obs}}(t_0), y_{jdr}^{\text{obs}}(t_1), \dots , y_{jdr}^{\text{obs}}(t_T), \tag{4.10}$$

where the three indices j, d, r identify, respectively, the design, the experimental day, and the technical replicate.
In most cases, the measurements associated with a given design j are collected on a single experimental day and repeated $R = 3$ times. However, some relevant exceptions must be noted:

- for the reference designs, observations are collected on all available experimental days;
- for the first 12 designs, fluorescence trajectories are available from two distinct experimental days;
- Design 7 and Design 32 include 36 additional measurements in order to monitor the success or failure of the protein degradation process.

Each observation $y_{jdr}^{\text{obs}}(t_i)$ is interpreted as a realization of a random variable $Y_{jdr}^{\text{obs}}(t_i)$. Hence, for each fixed triple (j, d, r), the observed fluorescence trajectory is viewed as a realization of the stochastic process

$$\{Y_{jdr}^{\text{obs}}(t_i)\}_{i=1}^{T}. \tag{4.11}$$

Sources of variability

Although the experimental conditions associated with a given design are nominally fixed, the observed fluorescence trajectories exhibit several sources of variability. These sources can be conceptually decomposed into three levels:

- **Inter-day variability (batch effect):** systematic differences between experimental days, affecting all wells measured on the same day. These effects may influence both the background fluorescence component $g_d(t)$ and the effective scale of the protein-related signal.
- **Intra-day variability (replicate variability):** well-to-well fluctuations within the same day, due for instance to pipetting variability, local plate conditions, or micro-environmental factors. These effects may induce replicate-specific differences in the observed fluorescence trajectories.
- **Instrumental noise:** random measurement error affecting each fluorescence observation independently in time, modeled through the Gaussian noise term $\varepsilon_{jdr}(t)$.

As a consequence, the fluorescence associated with a given design j may vary across experimental days and technical replicates.

In order to capture these sources of variability within a statistical framework, the observed fluorescence measurements are modeled as noisy observations of an underlying deterministic signal.

At this stage, before any pre-processing step, the raw fluorescence dynamics are described in a general form as

$$Y_{jdr}^{\text{obs}}(t_i) = h(t_i; \psi_{jdr}) + \varepsilon_{jdr,i}, \tag{4.12}$$

where $h(t_i; \psi_{jdr})$ denotes the expected fluorescence level at time t_i, ψ_{jdr} is the vector of trajectory-specific parameters, and $\varepsilon_{jdr,i}$ represents measurement noise. Unless otherwise specified, the measurement errors are assumed to be independent and Gaussian, with mean zero and variance σ_y^2, that is,

$$\varepsilon_{jdr,i} \sim \mathcal{N}(0, \sigma_y^2). \tag{4.13}$$

Decomposition of the fluorescence signal

The function $h(t; \psi_{jdr})$ introduced in the previous section represents the expected fluorescence level associated with the trajectory identified by design j, experimental day d, and replicate r.

$$h(t; \psi_{jdr}) = \mathbb{E}[Y_{jdr}^{\text{obs}}(t)] \tag{4.14}$$

From an experimental perspective, the fluorescence measured in each well is not generated by a single physical mechanism. Instead, the observed signal results from the superposition of multiple contributions.

In particular, two main components can be distinguished. The first is a *background fluorescence* component, which originates from the instrument, the plate, and the biochemical buffer used in the reaction. This component is present even in the absence of the reporter protein and can be experimentally observed through blank measurements (see Section 3.1.2). The second component is the fluorescence emitted by the sfGFP reporter protein, whose intensity evolves over time as a consequence of the degradation process.

Motivated by this physical interpretation, we assume that the expected fluorescence signal can be decomposed into a background-related term and a protein-related term. More precisely, the deterministic component of the fluorescence dynamics is written as

$$h(t; \psi_{jdr}) = g_{jdr}(t) + s_{jdr}(t), \tag{4.15}$$

where $g_{jdr}(t)$ denotes the background fluorescence contribution and $s_{jdr}(t)$ represents the fluorescence associated with the reporter protein.

This decomposition should be interpreted as a modeling assumption reflecting the experimental structure of the system. In particular, the following assumptions are adopted:

- **Additive structure of the fluorescence signal.** The total fluorescence measured in each well is assumed to be the sum of a background component and a protein-related component.
- **Separation of background and protein dynamics.** The background fluorescence and the reporter-related fluorescence are assumed to arise from distinct physical mechanisms and can therefore be modeled as separate contributions.
- **Background variability across experimental days.** The background signal may vary across experimental days due to instrumental conditions or plate-specific effects, while being largely independent of the specific biochemical design.
- **Design-dependent protein signal.** The component $s_{jdr}(t)$ captures the fluorescence emitted by the reporter protein and therefore depends on the biochemical configuration $\mathbf{x}_j$ defining design j.

Under these assumptions, the observed fluorescence can be written as

$$Y_{jdr}^{\text{obs}}(t) = g_{jdr}(t) + s_{jdr}(t) + \varepsilon_{jdr}(t), \tag{4.16}$$

where $\varepsilon_{jdr}(t)$ represents measurement noise.
In the following sections, the background component will be estimated using blank measurements, while the protein-related fluorescence will be modeled through a parametric function describing the degradation dynamics of the reporter protein.

4.3 Background and Protein Degradation

4.3.1 Background fluorescence

The fluorescence signal measured in each well contains contributions that are not directly related to the sfGFP reporter protein. These contributions arise from several experimental sources, including intrinsic fluorescence of buffer components, plate autofluorescence, and instrumental effects and they can be modeled with a background function $g_{jdr}(t)$.

In principle, it could depend on the specific design j, the experimental day d, and the replicate r. However, composition-matched blank measurements are not available for every individual design. Consequently, estimating a design-specific background function would not be feasible from the available data. (See Section 3.1.2)
To reduce the complexity of the model, we therefore introduce the following approximation:

$$g_{jdr}(t) \approx g_d(t). \tag{4.17}$$

This modeling assumption states that, within the same experimental day, the dominant background contribution is shared across all wells. In other words, potential well-specific background fluctuations are neglected, while day-to-day differences are assumed to capture the main source of systematic variability in the background signal.
The day-specific background function $g_d(t)$ is not specified through an explicit parametric model. Instead, it is estimated directly from blank measurements. Blank samples correspond to reaction mixtures containing all components of the cell-free system except the fluorescent reporter protein, and therefore provide an empirical observation of the background fluorescence of the system.

Let $y^{obs}_{\text{Blank},d,r}(t_i)$ denote the fluorescence measured in blank wells on day d, replicate r, at time t_i. The background function is estimated pointwise at the observed time points by

$$\widehat{g}_d(t_i) = \text{median}_r \left(y^{obs}_{\text{Blank},d,r}(t_i) \right), \qquad i = 0, \ldots, 38. \tag{4.18}$$

The median is used instead of the mean in order to obtain a robust estimate of the background signal in the presence of small well-to-well fluctuations or occasional measurement irregularities. This estimator provides a data-driven approximation of the time-dependent background fluorescence associated with each experimental day.

Finally, the estimated background curve $\widehat{g}_d(t)$ is used as a plug-in approximation of $g_d(t)$ in the subsequent analysis, allowing the observed fluorescence trajectories to be corrected for day-specific background effects before fitting the degradation model.

When valid blank measurements were available, Blank B was used to estimate the background curve for all design wells measured on the same experimental day and for the reference condition RefB, whereas Blank A was used exclusively for the reference condition RefA. This rule ensures that each fluorescence trajectory is corrected using blank measurements obtained under consistent reference conditions.

Once the background contribution has been characterized, the remaining component of the fluorescence signal can be attributed to the reporter protein. This component reflects the degradation dynamics of sfGFP and constitutes the primary signal of interest in the present study.

4.3.2 Protein-related Fluorescence

After accounting for the background contribution, the dominant component of the measured fluorescence is attributed to the sfGFP reporter protein. This signal component determines the overall shape of the observed trajectories and reflects the degradation dynamics of sfGFP mediated by the enzymatic complex ClpXP in the cell-free system, operating under specific initial concentrations of ATP, Mg^{2+} and PEG8000.

Preliminary inspection of the experimental trajectories indicates that, for $t \geq t_0$, curves exhibiting substantial degradation display a clear monotonic decrease compatible with an exponential-like decay, whereas trajectories associated with negligible degradation remain approximately flat. In order to describe both regimes within a unified modeling framework, the reporter-related fluorescence is represented through a single-exponential function, allowing the degradation rate to approach zero in the absence of observable degradation.

This modeling choice is consistent with the widely used assumption of *first-order degradation kinetics*, according to which the rate of decrease of the protein concentration is proportional to the current amount of protein present in the system. Under this assumption, the resulting temporal evolution follows an exponential decay law, a model commonly adopted to describe protein degradation dynamics in biochemical systems (see for example [**alberts_molecular_2015**, **nelson_lehninger_2017**]).

Formally, for a given design j, measured on experimental day d and replicate r, the reporter-related fluorescence component is modeled as

$$s_{j,d,r}(t) = b_{j,d,r} + a_{j,d,r} \exp\big(- k_{j,d,r}(t - t_0) \big), \qquad t \geq t_0. \tag{4.19}$$

This expression provides a phenomenological description of the dominant degradation dynamics while intentionally neglecting higher-order kinetic effects or potential multi-exponential behaviors that may arise in more complex biochemical settings.

The parameter $k_{j,d,r} \geq 0$ represents the degradation rate associated with the specific fluorescence trajectory corresponding to design j, day d, and replicate r. It quantifies the speed of the protein degradation process observed in that experimental curve.

The remaining parameters characterize the overall scale and baseline of the fluorescence signal. In particular, $a_{j,d,r}$ denotes the portion of fluorescence that decays over time due to protein degradation; while $b_{j,d,r}$ represents the asymptotic fluorescence level reached at long times once degradation has occurred. Variability in the pair $(a_{j,d,r}, b_{j,d,r})$ primarily captures inter-day batch effects and intra-day well-to-well fluctuations that manifest as changes in signal scale or baseline, whereas variation in $k_{j,d,r}$ reflects differences in the observed degradation kinetics.

Combining the background component and the protein-related signal, the fluorescence associated with a given trajectory can be described at the level of random variables as

$$Y^{obs}_{j,d,r}(t) = g_{j,d,r}(t) + b_{j,d,r} + a_{j,d,r} \exp\big(-k_{j,d,r}(t - t_0)\big) + \varepsilon_{j,d,r}(t), \qquad t \geq t_0, \tag{4.20}$$

where $\varepsilon_{j,d,r}(t)$ denotes the measurement noise term introduced previously.

In practice, the background function $g_{j,d,r}(t)$ is not directly observable and is approximated through the estimator $\hat{g}_d(t)$ derived from blank measurements. Consequently, the observed fluorescence data satisfy

$$y^{obs}_{j,d,r}(t) = \hat{g}_d(t) + b_{j,d,r} + a_{j,d,r} \exp\big(-k_{j,d,r}(t - t_0)\big) + r^{obs}_{j,d,r}(t), \qquad t \geq t_0, \tag{4.21}$$

where the residual term

$$r^{obs}_{j,d,r}(t) = \varepsilon_{j,d,r}(t) + \big(g_{j,d,r}(t) - \hat{g}_d(t)\big) \tag{4.22}$$

accounts both for instrumental noise and for the approximation error introduced by replacing the true background component with its data-driven estimate.

This formulation provides a trajectory-level representation of the experimental fluorescence measurements, explicitly separating background fluorescence, protein-related signal dynamics, and residual variability. In particular, it introduces the degradation rate $k_{j,d,r}$ as the key kinetic descriptor associated with each experimental curve.

Since the objective of this work is to associate each design j, characterized by its covariate vector $\mathbf{x}_j$, with a single kinetic descriptor, the next step consists in deriving a design-level degradation rate k_j from the collection of trajectory-level rates $k_{j,d,r}$ varying across days and replicates.

4.3.3 Batch Effect explained by the model

In order to better understand the role of day-to-day and replicate variability in the observed fluorescence trajectories, it is useful to consider the protein-related signal component in isolation. For this discussion, and only for interpretative purposes, we neglect the background term and the measurement noise, and approximate the fluorescence trajectories by the signal component

$$s_{jdr}(t) = b_{jdr} + a_{jdr} \exp\big(-k_{jdr}(t - t_0)\big). \tag{4.23}$$

Consider first two trajectories associated with the same design j and the same experimental day d, but with different replicates $r_1 \neq r_2$. Their difference is given by

$$s_{jdr_1}(t) - s_{jdr_2}(t) = \big(b_{jdr_1} - b_{jdr_2}\big) + a_{jdr_1} e^{-k_{jdr_1}(t-t_0)} - a_{jdr_2} e^{-k_{jdr_2}(t-t_0)}, \tag{4.24}$$

which is not constant in time in general. Likewise, the ratio

$$\frac{s_{jdr_1}(t)}{s_{jdr_2}(t)} = \frac{b_{jdr_1} + a_{jdr_1} e^{-k_{jdr_1}(t-t_0)}}{b_{jdr_2} + a_{jdr_2} e^{-k_{jdr_2}(t-t_0)}} \tag{4.25}$$

is not constant in time unless additional restrictive assumptions are imposed.

The same conclusion holds when comparing trajectories associated with the same design j but acquired on different experimental days. In that case, differences between curves may arise not only from replicate-specific fluctuations, but also from day-dependent effects acting on the background component, on the overall signal scale, on the asymptotic fluorescence level, and potentially on the degradation rate itself. This qualitative behaviour is also supported by an empirical inspection of the experimental trajectories. In particular, pairwise differences and ratios between fluorescence curves corresponding to the same design and experimental day were computed across technical replicates. As discussed in Section 3.2, these transformed trajectories are generally not constant in time and exhibit distinct shapes depending on the pair of replicates considered. This empirical observation confirms that replicate variability cannot be described solely through a simple additive or multiplicative correction acting uniformly over time.

These observations clarify where the batch effect may enter the general model. In the formulation

$$Y_{jdr}^{\text{obs}}(t) = g_{jdr}(t) + s_{jdr}(t) + \varepsilon_{jdr}(t), \tag{4.26}$$

day-to-day variability may affect the background contribution $g_{jdr}(t)$, the parameters a_{jdr} and b_{jdr}, and, in principle, also the kinetic parameter k_{jdr}. Consequently, batch effects are not expected to appear as a simple constant shift or as a time-independent multiplicative factor.

This motivates the preprocessing strategy adopted in the following sections. Background subtraction removes the dominant day-specific additive component of the signal, while normalization reduces trajectory-to-trajectory variability in scale. Together, these steps simplify the statistical problem and make the degradation rate k_{jdr} more directly comparable across trajectories.

4.4 Pre-processing for rate estimation

The raw fluorescence measurements $y_{j,d,r}^{raw}(t)$ recorded over the experimental time interval contain several sources of variability, including background fluorescence and differences in signal scale across trajectories. As discussed in the previous sections, these effects may obscure the kinetic information carried by the protein-related signal.

In order to obtain a robust estimate of the degradation rate, the observed trajectories are therefore transformed through a sequence of model-consistent preprocessing steps. These transformations are designed to reduce background contributions and scale-related variability while preserving the underlying exponential degradation dynamics.

The preprocessing procedure consists of two main steps:

- subtraction of the estimated background signal;
- normalization with respect to the fluorescence level at the initial modeling time t_0.

Together, these operations isolate the protein-related fluorescence component and yield a representation of the trajectories that is more suitable for the estimation and comparison of degradation rates across designs.

4.4.1 Background subtraction

Let us consider again the approximation of the observed fluorescence introduced in Eq. 4.21. Since the background component $g_{j,d,r}(t)$ is not directly observable, it is approximated through the day-specific estimator $\hat{g}_d(t)$ obtained from blank measurements.

Subtracting this estimated background from the raw fluorescence observations yields the quantity

$$y^{sub}_{j,d,r}(t) = y^{obs}_{j,d,r}(t) - \hat{g}_d(t), \qquad t \geq t_0. \tag{4.27}$$

Using the decomposition introduced previously, this expression can be written as

$$y^{sub}_{j,d,r}(t) = s_{j,d,r}(t) + r^{obs}_{j,d,r}(t), \tag{4.28}$$

where $s_{j,d,r}(t)$ denotes the protein-related fluorescence component and $r^{obs}_{j,d,r}(t)$ collects both the instrumental noise and the approximation error arising from replacing the true background $g_{j,d,r}(t)$ with its estimate $\hat{g}_d(t)$.

Consequently, the background-corrected observations can be interpreted as noisy measurements of the protein-related signal,

$$y^{sub}_{j,d,r}(t) \approx s_{j,d,r}(t). \tag{4.29}$$

This operation removes the additive component of inter-day variability associated with the background fluorescence and isolates the part of the signal that carries information about the protein degradation dynamics.

In the next step, an additional normalization procedure is introduced in order to reduce remaining variability in signal scale across trajectories.

4.4.2 Data Normalization

As discussed previously, fluorescence measurements collected during the first part of the experiment are affected by transient mixing and stabilization effects that are not described by the exponential kinetic model adopted in this work. For this reason, the analysis is restricted to the post–transient regime starting at

$$t_0 = 50 \text{ minutes}. \tag{4.30}$$

Starting from the background–corrected observations introduced in Eq. 4.27, an additional preprocessing step is applied in order to reduce trajectory-to-trajectory variability in signal scale. This is achieved by normalizing each curve with respect to its value at the initial modeling time t_0.

Formally, the normalized observations are defined as

$$y^{norm}_{j,d,r}(t) = \frac{y^{sub}_{j,d,r}(t)}{y^{sub}_{j,d,r}(t_0)} = \frac{y^{raw}_{j,d,r}(t) - \widehat{g}_d(t)}{y^{raw}_{j,d,r}(t_0) - \widehat{g}_d(t_0)}, \qquad t \geq t_0. \tag{4.31}$$

This transformation is applied directly to the observed trajectories and forces all normalized curves to satisfy

$$y^{norm}_{j,d,r}(t_0) = 1, \tag{4.32}$$

thereby removing variability associated with the absolute fluorescence scale.

To interpret this transformation in terms of the underlying signal model, recall that after background subtraction the observations can be written as

$$y^{sub}_{j,d,r}(t) = s_{j,d,r}(t) + r^{obs}_{j,d,r}(t). \tag{4.33}$$

If the residual variability is moderate compared to the signal magnitude—particularly at the reference time t_0—the normalized trajectory can be approximated by the normalized deterministic component,

$$y^{norm}_{j,d,r}(t) \approx \frac{s_{j,d,r}(t)}{s_{j,d,r}(t_0)}, \qquad t \geq t_0. \tag{4.34}$$

Using the exponential model introduced in Section 4.3.2,

$$s_{j,d,r}(t) = b_{j,d,r} + a_{j,d,r} \exp\big(- k_{j,d,r}(t - t_0)\big), \tag{4.35}$$

the normalized deterministic component becomes

$$\frac{s_{j,d,r}(t)}{s_{j,d,r}(t_0)} = \frac{b_{j,d,r} + a_{j,d,r} \exp\big(- k_{j,d,r}(t - t_0)\big)}{b_{j,d,r} + a_{j,d,r}}. \tag{4.36}$$

Introducing the normalized asymptotic parameter

$$B_{j,d,r} := \frac{b_{j,d,r}}{a_{j,d,r} + b_{j,d,r}}, \tag{4.37}$$

the previous expression can be rewritten as

$$\frac{s_{j,d,r}(t)}{s_{j,d,r}(t_0)} = B_{j,d,r} + (1 - B_{j,d,r}) \exp\big(- k_{j,d,r}(t - t_0)\big). \tag{4.38}$$

This representation highlights an important consequence of the normalization procedure. The amplitude parameter $a_{j,d,r}$ no longer appears as an independent degree of freedom, since the signal is rescaled by the initial fluorescence level $a_{j,d,r} + b_{j,d,r}$. Instead, the normalized dynamics are fully described by the pair $(B_{j,d,r}, k_{j,d,r})$.

Importantly, the degradation rate parameter $k_{j,d,r}$ remains unchanged under this transformation. Indeed, normalization rescales the signal amplitude but does not modify the exponential term $\exp(-k_{j,d,r}(t - t_0))$, which determines the temporal decay rate. Consequently, the kinetic information encoded in $k_{j,d,r}$ is preserved.

Finally, defining the normalized error term

$$\tilde{\varepsilon}_{j,d,r}(t) = y^{norm}_{j,d,r}(t) - \frac{s_{j,d,r}(t)}{s_{j,d,r}(t_0)}, \qquad t \geq t_0, \tag{4.39}$$

the normalized observations can be written as

$$y^{norm}_{j,d,r}(t) = B_{j,d,r} + (1 - B_{j,d,r}) \exp\big(- k_{j,d,r}(t - t_0)\big) + \tilde{\varepsilon}_{j,d,r}(t), \qquad t \geq t_0. \tag{4.40}$$

4.4.3 Role of the asymptotic parameter

The normalized representation introduced in Eq. 4.40 shows that each trajectory is characterized by two parameters: the degradation rate $k_{j,d,r}$ and the normalized asymptotic level $B_{j,d,r}$.

While both parameters contribute to the description of the fluorescence dynamics, their roles within the present study are different.

The parameter $k_{j,d,r}$ governs the exponential decay term and therefore determines the temporal kinetics of protein degradation. In contrast, $B_{j,d,r}$ controls the relative asymptotic fluorescence level of the normalized trajectory and mainly affects the vertical offset of the curves.

Since the primary objective of this work is to extract a single kinetic descriptor of degradation associated with each experimental design, the degradation rate represents the quantity of primary interest, whereas the asymptotic level mainly plays a secondary role in the description of the fluorescence trajectories.

For this reason, the statistical analysis focuses primarily on the estimation and interpretation of the degradation rates $k_{j,d,r}$, while the asymptotic parameters $B_{j,d,r}$ are treated as trajectory-specific nuisance parameters required to properly describe the observed dynamics.

4.5 Statistical model for degradation rates

The normalized fluorescence trajectories introduced in the previous section are described through a trajectory-level exponential model of the form

$$y^{norm}_{j,d,r}(t) = B_{j,d,r} + (1 - B_{j,d,r}) \exp\big(- k_{j,d,r}(t - t_0)\big) + \tilde{\varepsilon}_{j,d,r}(t), \qquad t \geq t_0. \tag{4.41}$$

where:

- $B_{j,d,r} \in [0,1]$ denotes the normalized asymptotic fluorescence level associated with trajectory (j, d, r),
- $k_{jdr} > 0$ is the degradation rate of the trajectory identified by design j, experimental day d, and replicate r,
- $\tilde{\varepsilon}_{jdr}(t_i)$ represents a residual error term accounting for measurement noise and modeling misspecification.

Logarithmic parametrization of degradation rates Since degradation rates are strictly positive quantities, $k_{jdr} > 0$, it is convenient to model them on the logarithmic scale. Introducing the transformed variable

$$\ell_{jdr} := \log k_{jdr}, \qquad \ell_{jdr} \in \mathbb{R} \tag{4.42}$$

the parameter space becomes the entire set $\mathbb{R}$. This transformation allows the degradation rates to be described through an additive linear structure and ensures that the corresponding rates remain positive on the original scale.
The statistical model for the log-rates is therefore written as

$$\log k_{jdr} = \mu_j + \alpha_d + u_{jdr}, \tag{4.43}$$

where

- μ_j is the design-specific effect associated with the experimental configuration $\mathbf{x}_j$;
- α_d represents the effect of experimental day d, capturing systematic batch-to-batch variability;
- u_{jdr} is a residual trajectory-specific term accounting for remaining variability between trajectories.

The residual component is assumed to satisfy

$$u_{jdr} \sim \mathcal{N}(0, \sigma_k^2), \tag{4.44}$$

independently across trajectories. The parameter σ_k^2 therefore quantifies the residual dispersion of log-rates around their expected value.

Interpretation of model parameters Within this formulation, the quantity

$$k_j := \exp(\mu_j) \tag{4.45}$$

represents the design-level degradation rate. It can be interpreted as the intrinsic degradation speed associated with the biochemical configuration encoded by the vector of covariates $\mathbf{x}_j$.

Similarly, the day effect

$$R_d := \exp(\alpha_d) \tag{4.46}$$

represents a multiplicative batch factor acting on the degradation rates. Moreover, values $R_d > 1$ indicate experimental days associated with systematically faster degradation rates, while $R_d < 1$ correspond to days producing lower apparent rates.
Combining these terms, the trajectory-specific rate can be written as

$$k_{jdr} = k_j \, R_d \, \exp(u_{jdr}), \tag{4.47}$$

showing that the observed degradation rate of each trajectory results from the interaction of three components: a design-specific rate, a day-dependent multiplicative factor, and a residual stochastic fluctuation.

Model structure The resulting formulation can be interpreted as a linear mixed model on the logarithmic scale of the degradation rate. The parameters μ_j and α_d play the role of fixed effects describing the systematic contribution of design and experimental day, whereas the terms u_{jdr} represent trajectory-specific random deviations.
This hierarchical structure allows the model to separate the intrinsic kinetic properties of each design from day-dependent experimental effects and residual replicate variability.

Aim of the model The primary purpose of this statistical model is to extract, for each design j, a single kinetic descriptor k_j summarizing the speed of protein degradation. This quantity will later serve as the response variable in the subsequent Active Learning procedure aimed at exploring the design space.

4.5.1 Model limitations

Despite its simplicity and interpretability, the model relies on several simplifying assumptions.

1. The trajectory-level dynamics are approximated by a single exponential decay, which may not capture multi-phase degradation processes or more complex kinetic behaviors.

2. The log-rate model assumes that the effects of design and experimental day combine additively on the logarithmic scale, implying a multiplicative interaction on the original rate scale.

3. the residual variability u_{jdr} is assumed to be homoscedastic and Gaussian across trajectories.

4. The asymptotic fluorescence levels B_{jdr} are treated as trajectory-specific parameters without explicitly modeling possible systematic effects associated with experimental day or batch. In principle, the asymptotic levels may also be affected by day-dependent experimental variability, similarly to the degradation rates.

   However, introducing additional hierarchical structure or batch effects for the parameters B_{jdr} would considerably increase the dimensionality of the model and may lead to identifiability issues due to the strong interaction between the asymptotic level and the degradation rate in the exponential trajectory model.

   For this reason, the modeling framework focuses primarily on the rate parameters k_{jdr}, while the parameters B_{jdr} are estimated independently for each trajectory without further hierarchical structure. As a consequence, part of the variability that might be associated with day-specific effects on the asymptotic fluorescence level is effectively absorbed by these trajectory-level parameters.

The statistical model introduced above provides a formal description of the degradation dynamics and of the variability observed across designs, experimental days, and replicate trajectories. In order to estimate the unknown parameters of this model and quantify the associated uncertainty, a Bayesian inferential framework is adopted. Posterior inference is performed using the probabilistic programming language JAGS (Just Another Gibbs Sampler), which allows the hierarchical structure of the model to be implemented and explored through Markov Chain Monte Carlo (MCMC) sampling.

4.6 Design-level kinetic label

Within the statistical model introduced above, the parameter of primary interest is the design-level effect μ_j, which represents the expected log-degradation rate associated with the experimental design $\mathbf{x}_j$, after accounting for day-dependent batch effects and trajectory-level variability.
Since the objective of the study is to identify experimental configurations that maximize the degradation rate, this quantity is used as the kinetic descriptor associated with each design. In practice, the label assigned to design j is defined as the posterior median of the corresponding log-rate parameter,

$$\hat{\mu}_j = \mathrm{median}\big(\mu_j \mid \mathrm{data}\big). \tag{4.48}$$

The estimation of these quantities requires performing posterior inference for the hierarchical model described above. In the following chapter we describe the computational methodologies adopted to estimate the model parameters and to exploit the resulting kinetic labels for the exploration of the experimental design space.

Transition to the methodological chapters The statistical model introduced in this chapter provides a formal representation of the degradation dynamics and of the variability observed across experimental designs, experimental days, and replicate trajectories. In order to estimate the unknown parameters of this model and quantify the associated uncertainty, a Bayesian inferential framework is adopted.
The methodological procedures used to perform this inference and to explore the experimental design space are presented in the following chapters. Chapter 5 describes the Bayesian inference procedure implemented using JAGS for estimating the parameters of the hierarchical degradation model, while Chapter 6 introduces the Active Learning framework used to iteratively guide the exploration of the design space.

Chapter 5

Bayesian Inference with JAGS

5.1 Introduction

JAGS (Just Another Gibbs Sampler) is a framework for performing Bayesian inference based on Markov Chain Monte Carlo (MCMC) methods.
We recall some theoretical concepts, included in G. Mastrantonio's Lecture notes [3].
In the Bayesian framework, the parameter vector θ is treated as a random variable whose distribution represents the uncertainty about the parameter values before observing the data. Therefore, it is necessary to specify the prior distribution of θ.
Once a sample y has been observed, we are interested in the conditional distribution $f(\theta \mid y)$.
We now recall the **Bayes Theorem**.

Theorem 5.1.1 (Bayes Theorem) *Given two random variables X and Y with density functions $f(x)$ and $f(y)$, the following relationship holds:*

$$f(x \mid y) = \frac{f(y, x)}{f(y)} = \frac{f(y \mid x)f(x)}{f(y)} \tag{5.1}$$

where

$$f(y) = \int f(y, x)\, dx$$

if X is continuous, and

$$f(y) = \sum_x f(y, x)$$

if X is discrete.

Applied to our context, where θ denotes the unknown parameter vector and y the observed data, Bayes' theorem allows us to write:

$$f(\theta \mid y) = \frac{f(y \mid \theta)f(\theta)}{f(y)}. \tag{5.2}$$

where:

- $f(\theta \mid y)$ is the posterior distribution of θ;
- $f(y \mid \theta)$ is the likelihood function of the observed data given the parameters;

- $f(\theta)$ is the prior distribution, representing prior knowledge about the parameters before observing y. It can be informative, weakly informative, or non-informative;
- $f(y)$ is a normalizing constant, also called the marginal likelihood or model evidence.

In the present work, both the observed data y and the parameter vector θ are modeled as continuous random variables. Consequently, all probability functions involved (prior, likelihood, and posterior) are probability density functions, and the posterior distribution is defined with respect to the Lebesgue measure on the parameter space.
It is useful to observe that the posterior distribution can be written as the ratio between an unnormalized density, $k(\theta|y)$ and a *normalizing constant* $C(y)$, as follows:

$$f(\theta \mid y) = \frac{f(y \mid \theta) f(\theta)}{\int f(y \mid \theta) f(\theta)\, d\theta} = \frac{k(\theta|y)}{C(y)} \tag{5.3}$$

The numerator $k(\theta|y) := f(y \mid \theta)f(\theta)$ is referred to as *kernel* and it is proportional to the posterior density. Since the kernel does not integrate to 1, the introduction of a normalizing constant is necessary to well-define $f(\theta|y)$ as density function.
That said, the posterior distribution can be expressed in proportional form as

$$f(\theta \mid y) \propto f(y \mid \theta) f(\theta) = k(\theta|y)$$

In many practical problems, the normalizing constant $C(y) = \int f(y \mid \theta)f(\theta)\, d\theta$ is analytically intractable. For this reason, Markov Chain Monte Carlo (MCMC) methods are employed to generate samples from the posterior distribution without explicitly computing the normalizing constant.

In particular, JAGS operates directly on the kernel and produces samples from the posterior distribution using Gibbs sampling and related MCMC algorithms. Moreover, it represents statistical models through directed acyclic graphs (DAGs), where nodes correspond to stochastic variables and edges represent conditional dependencies. This graphical representation allows complex hierarchical models to be specified in a modular way, while MCMC algorithms are automatically selected for the corresponding full conditional distributions.
Prima di continuare la narrazione con l'introduzione degli algoritmi utilizzati, è bene sottolineare l'importanza dei metodi bayesiani: essi permettono di fare inferenza anche se non si conosce in forma chiusa la distribuzione a posteriori, purché si sia in grado di campionare da essa.

Before introducing the specific MCMC algorithms implemented in JAGS, it is worth emphasizing the methodological implications of the proportional formulation of the posterior distribution. The key observation is that Bayesian inference does not require the posterior density to be available in closed form. It suffices to generate samples $\theta^{(s)}$ from the corresponding distribution.
Therefore, the inferential goal is shifted from obtaining a closed-form expression of $f(\theta \mid y)$ to constructing a sampling mechanism whose stationary distribution coincides with it.
This perspective differs substantially from classical least squares estimation (LSE).

Least Squared Estimation and its limitations

Consider a nonlinear regression model

$$y_i = f(t_i, \theta) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

The least squares estimator is defined as

$$\hat{\theta}_{\mathrm{LSE}} = \arg\min_{\theta} \sum_{i=1}^{n} \big(y_i - f(t_i, \theta)\big)^2,$$

that is, as the minimizer of the empirical squared error. This procedure provides a point estimate but does not directly yield a full uncertainty characterization of θ. Under suitable regularity conditions and for sufficiently large samples, one obtains the asymptotic approximation

$$\hat{\theta}_{\mathrm{LSE}} \approx \mathcal{N}\big(\theta_0, \sigma^2 (J^\top J)^{-1}\big),$$

where J denotes the Jacobian matrix of partial derivatives of f evaluated at $\hat{\theta}_{\mathrm{LSE}}$. However, this result relies on local linearization and asymptotic arguments. In nonlinear or multilevel models, such approximations may become unstable or technically involved.

In contrast to a purely frequentist approach, the Bayesian framework treats the parameter vector θ as a random quantity and provides the full posterior distribution $f(\theta \mid y)$, allowing uncertainty to be directly quantified and propagated across all levels of the model.
This feature is particularly useful in the present study, where hierarchical components such as batch effects, day-specific effects, and trajectory-level variability are explicitly modeled. Posterior inference is performed using JAGS, which enables sampling from the complex posterior distribution through Markov Chain Monte Carlo (MCMC) methods.
We now proceed to describe the MCMC algorithms that allow JAGS to generate samples from the posterior distribution.

MCMC algorithms

Bayesian methods allow us to approximate the posterior statistics of the parameters θ by sampling from their posterior distribution $f(\theta \mid y) \propto f(y \mid \theta) f(\theta) = k(\theta \mid y)$. It suffices to construct a Markov chain that converges to its stationary distribution, coinciding with the posterior distribution $f(\theta \mid y)$. Once convergence is achieved, posterior expectations and other summary statistics can be approximated using Monte Carlo methods of the sampled values.

Jags combines different strategies to simulate from the posterior probability density function.
The general approach used for sampling is **Gibbs Sampling**, but other techniques are involved to enforce it or, in some cases, substitute it. These are listed below:

- Metropolis-Hastings
- Adaptive Rejection sampling
- Slice sampling

We now describe each of these methods

5.1.1 Gibbs Sampling

The Gibbs sampler is an MCMC algorithm designed to generate samples from a multivariate distribution by iteratively sampling from its full conditional distributions.
The key idea is to exploit the factorization of the joint distribution through its *full conditional distributions.*
In particular, the i-th full conditional distribution is defined as

$$f(\Theta_i \mid \theta_{-i}, y) = f(\Theta_i \mid \theta_1, ...\theta_{i-1}, \theta_{i+1}, ...\theta_p, y) \tag{5.4}$$

This expression states that, for each $i = 1, \dots, p$, the full conditional of θ_i is the distribution of θ_i conditioned on all the remaining components of θ, denoted by θ_{-i}.

Algorithm. Given an initial value $\theta^{(0)} = (\theta_1^{(0)}, \dots, \theta_p^{(0)})$, for $b = 1, \dots, B$ repeat:

$$\begin{aligned}
\theta_1^{(b)} &\sim f(\theta_1 \mid \theta_2^{(b-1)}, \dots, \theta_p^{(b-1)}, y), \\
\theta_2^{(b)} &\sim f(\theta_2 \mid \theta_1^{(b)}, \theta_3^{(b-1)}, \dots, \theta_p^{(b-1)}, y), \\
&\vdots \\
\theta_p^{(b)} &\sim f(\theta_p \mid \theta_1^{(b)}, \dots, \theta_{p-1}^{(b)}, y).
\end{aligned}$$

The vector $\theta^{(b)} = (\theta_1^{(b)}, \dots, \theta_p^{(b)})$ constitutes the b-th draw from the Markov chain.

Each update is performed conditionally on the most recent values of the remaining components. This sequential updating defines a first-order discrete-time Markov chain $(\theta^b)_{b \geq 0}$, since the construction of $\theta^{(b)}$ is only based on $\theta^{(b-1)}$. Therefore, the following Markov property holds:

$$f(\theta^{(b)} | \theta^{(b-1)}, \theta^{(b-2)}, ..., \theta^{(1)}) = f(\theta^{(b)} | \theta^{(b-1)}) \tag{5.5}$$

The Markov chain is specified by an initial distribution $f_0^{(}\theta^{(0)})$ and a stochastic transition function $T(\theta^{(b)} | \theta^{(b-1)})$, defined by the algorithm through the product of full conditional distributions.

Properties

- Gibbs sampler requires the ability to derive the full conditional distributions and sample directly from them;
- The generated samples depend on the initialization, especially during the early iterations of the chain;
- The components of $\theta^{(b)}$ are generally correlated with each other, reflecting the dependencies among parameters in the posterior distribution;
- In Gibbs sampling the *detailed balance condition* is satisfied:

  $$f(\theta^{(b)}) T(\theta^{(b)} | \theta^{(b-1)}) = f(\theta^{(b-1)}) T(\theta^{(b-1)} | \theta^{(b)}) \tag{5.6}$$

  which guarantees that the posterior distribution $f(\theta \mid y)$ is a stationary distribution of the Markov chain.

Under standard regularity conditions (irreducibility and aperiodicity), the chain converges in distribution to the posterior:

$$\theta^{(b)} \xrightarrow{d} f(\theta \mid y).$$

After a suitable burn-in period, the simulated draws can, therefore, be regarded as samples from the posterior distribution.

When the model admits conjugate priors, the full conditionals belong to known parametric families and sampling is straightforward. In more complex models, such as the model considered in this work, however, full conditionals may not be available in closed form or may not be directly simulable; in such cases, Metropolis–Hastings steps can be embedded within the Gibbs scheme (**Metropolis-within-Gibbs**).

5.1.2 Metropolis–Hastings Algorithm

In many practical Bayesian models, the full conditional distributions are not available in closed form or cannot be directly sampled from. In such cases, the Gibbs sampler cannot be applied in its standard form.
The Metropolis–Hastings (MH) algorithm provides a general strategy to construct a Markov chain whose stationary distribution coincides with a given target density:

$$f(\theta \mid y) \propto k(\theta \mid y) \tag{5.7}$$

which is the posterior distribution of the parameters θ given the observed data y, proportional to the kernel, the only required factor for the implementation of the algorithm.

The general idea of the algorithm, consists of, at a given current state $\theta^{(b-1)}$, chosen a proposal distribution, proposing a candidate value θ^* from it, with density function $q(\theta^* \mid \theta^{(b-1)})$. The candidate is accepted with a certain probability α. If the proposal is rejected, the chain remains at the current state.

Algorithm. Given an initial value $\theta^{(0)}$ and a chosen distribution Q (Proposal Distribution), for $b = 1, \dots, B$ repeat:

$$\begin{aligned}
\text{Proposal:}\quad & \theta^* \sim q(\theta^* \mid \theta^{(b-1)}), \\
\text{Acceptance probability:}\quad & \alpha(\theta^*, \theta^{(b-1)}) = \alpha \min\left\{ \frac{f(\theta^* \mid y)\, q(\theta^{(b-1)} \mid \theta^*)}{f(\theta^{(b-1)} \mid y)\, q(\theta^* \mid \theta^{(b-1)})}, 1 \right\}, \\
\text{Accept–reject step:}\quad & u \sim \mathcal{U}(0,1), \\
& \theta^{(b)} = \begin{cases} \theta^* & \text{if } u \le \alpha(\theta^*, \theta^{(b-1)}), \\ \theta^{(b-1)} & \text{otherwise.} \end{cases}
\end{aligned}$$

The vector $\theta^{(b)}$ represents the b-th state of the Markov chain. The sequence $(\theta^{(b)})_{b \ge 0}$ defines a first-order discrete-time Markov chain.
Regarding the detailed balance condition and convergence properties, the same considerations as the Gibbs Sampler can be applied to Metropolis-Hastings.

Properties

- If the proposal density is the full conditional then $\alpha = 1$ and the proposal value is always accepted.
- The algorithm only requires knowledge of the kernel of the target distribution.
- Its efficiency strongly depends on the choice of the proposal distribution: bad choices lead to a constant rejection.
- Poorly tuned proposals lead to high rejection rates and strong autocorrelation.
- In high-dimensional problems, random-walk proposals may mix slowly.

- If the proposal distribution is symmetric, $q(\theta^* \mid \theta^{(b-1)}) = q(\theta^{(b-1)} \mid \theta^*)$, the acceptance probability simplifies to
$$\alpha = \min\left\{\frac{f(\theta^*|y)}{f(\theta^{(b-1)}|y)}, 1\right\}. \tag{5.8}$$

Metropolis–within–Gibbs. In many practical Bayesian models the full conditional distributions required by the Gibbs sampler are not available in closed form or cannot be sampled directly. In such situations, Metropolis–Hastings steps can be embedded within the Gibbs scheme, leading to the so–called *Metropolis–within–Gibbs* algorithm.
Let $\theta = (\theta_1, \ldots, \theta_p)$ denote the parameter vector and consider the posterior distribution $f(\theta \mid y)$. In a standard Gibbs sampler, each component θ_i is updated by sampling directly from its full conditional distribution
$$f(\theta_j \mid \theta_{-i}, y),$$
where θ_{-i} denotes the vector θ without its i-th component.
When this distribution cannot be directly simulated, a Metropolis–Hastings step can be used instead. In particular, given the current state $\theta_i^{(b-1)}$, a candidate value θ_i^* is proposed from a proposal distribution $\theta_i^* \sim q(\theta_j^* \mid \theta_i^{(b-1)})$
The proposed value is then accepted with probability
$$\alpha = \min\left\{\frac{f(\theta_i^* \mid \theta_{-i}^{(b)}, y)\, q(\theta_i^{(b-1)} \mid \theta_i^*)}{f(\theta_i^{(b-1)} \mid \theta_{-i}^{(b)}, y)\, q(\theta_i^* \mid \theta_i^{(b-1)})}, 1\right\}. \tag{5.9}$$
If the proposal is accepted, we set $\theta_i^{(b)} = \theta_i^*$; otherwise the chain remains at the previous value $\theta_i^{(b)} = \theta_i^{(b-1)}$.
This hybrid scheme preserves the posterior distribution $f(\theta \mid y)$ as the stationary distribution of the Markov chain, since each Metropolis step targets the corresponding full conditional distribution. The Metropolis–within–Gibbs algorithm therefore combines the efficiency of Gibbs sampling with the flexibility of Metropolis–Hastings updates, allowing inference in complex Bayesian models where some conditional distributions are not directly tractable.
In addition to Gibbs sampling and Metropolis–Hastings updates, JAGS implements other sampling strategies that are commonly used to sample from continuous distributions when the full conditionals possess specific regularity properties.

5.1.3 Adaptive Rejection Sampling

In some Bayesian models the full conditional distributions required by the Gibbs sampler are not associated with standard parametric families, but they still possess structural properties that allow efficient sampling. In particular, when a full conditional distribution is continuous, univariate, and log–concave, JAGS can employ the *Adaptive Rejection Sampling* (ARS) algorithm. This one was originally introduced by Gilks and Wild (1992) [4] and permits to generate samples directly from the full conditional distribution.
The main idea is exploiting the concavity of the logarithm of the full conditional to build, through tangent lines, an envelope density which can be sampled easily and whose candidate values are accepted with a certain probability.
The procedure is called *adaptive* because whenever a proposal is rejected, an additional tangent line is added and improves the approximation given by the envelope, which get progressively tighter and increases the acceptance rate during the simulation.

Even though ARS algorithm has several advantages, in this work it cannot be applied: in complex nonlinear models the full conditional distributions are generally not log–concave.
Consequently, JAGS typically relies on slice sampling or Metropolis–within–Gibbs updates rather than Adaptive Rejection Sampling.

5.1.4 Slice Sampling

Slice sampling is an MCMC technique designed to generate samples from a target density function, such as a full conditional distribution $f(\theta_i \mid \theta_{-i}, y)$., when it is not available in closed form and does not belong to a standard parametric family. In contrast with Adaptive Rejection Sampling, slice sampling does not require log–concavity of the density and can therefore be applied to a much broader class of posterior distributions.

The key idea of slice sampling is to introduce an auxiliary variable and transform the problem of sampling from the target density function into sampling uniformly from the region under its graph. Let $f(\theta_i \mid \theta_{-i}, y)$ be the target density.
Consider then the joint distribution

$$p(\theta_i, u) = \begin{cases} 1 & \text{if } 0 < u < f(\theta_i \mid \theta_{-i}, y), \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

. This distribution is uniform over the region $A = \{(\theta_i, u) : 0 < u < f(\theta_i \mid \theta_{-i}, y)\}$.
Marginalizing with respect to the auxiliary variable u recovers the original density $f(\theta_i \mid \theta_{-i}, y)$.
Therefore, if one can simulate from the joint distribution, the resulting samples of θ_i follow the desired conditional distribution.
The algorithm alternates between two steps. Starting from the current value $\theta_i^{(b-1)}$, a sample is taken from the uniform distribution

$$u \sim \mathcal{U}\big(0, f(\theta_i^{(b-1)} \mid \theta_{-i}, y)\big). \tag{5.11}$$

This defines a horizontal level that determines the *slice*

$$S = \{\theta_i : f(\theta_i \mid \theta_{-i}, y) > u\}. \tag{5.12}$$

The next state of the chain is then obtained by drawing

$$\theta_i^{(b)} \sim \mathcal{U}(S). \tag{5.13}$$

In practice the set S is not known explicitly, and numerical procedures such as the *stepping–out* and *shrinkage* strategies are used to construct an interval containing the slice and sample uniformly within it.
Slice sampling has several advantages compared with other MCMC methods. In particular, it does not require the specification of a proposal distribution, automatically adapts the scale of the sampling region to the local shape of the target density, and remains applicable even when the density is not log–concave. For these reasons it is frequently employed in Bayesian computation software, including JAGS, to update parameters whose full conditional distributions are complex and not directly simulable.
For further details on the theoretical properties and practical implementation of slice sampling see Neal (2003)[4], Gilks et al. (1996)[5], and Robert and Casella (2004)[6].
The following section is the direct application of JAGS framework to our problem

5.2 Two-stage Bayesian inference using JAGS

The statistical model introduced in the previous section provides a conceptual description of the degradation dynamics and of the main sources of variability affecting the fluorescence trajectories. However, a fully joint implementation of the model—including all designs, experimental days, and trajectory-level parameters—would lead to a high-dimensional nonlinear hierarchical inference problem, which may be computationally demanding and difficult to stabilize given the limited number of trajectories available for several designs.

For this reason, posterior inference is implemented through a two-stage Bayesian strategy in JAGS. The main idea is to separate the estimation of day-specific batch effects from the estimation of design-specific degradation rates.

In the first stage, a joint model is fitted to the reference trajectories (RefA and RefB) in order to estimate the day effect on the degradation rate. Since the reference conditions are repeatedly observed across multiple experimental days and are specifically intended to monitor the reproducibility of the experimental system, they provide the most stable source of information for the estimation of batch effects.

In the second stage, each design is modeled separately through a trajectory-level exponential model. Posterior samples of the corresponding degradation rates are then corrected a posteriori using the batch factors estimated in the first stage. In this way, the final design-level kinetic descriptor is constructed from rate samples that have been adjusted for day-specific variability.

This two-stage procedure should be interpreted as a computational approximation of the general statistical model introduced previously. It does not correspond to a fully joint posterior inference on all parameters simultaneously. Nevertheless, it preserves the main structural components of the problem, namely: the nonlinear trajectory-level model for fluorescence decay, the batch effect acting on degradation rates, and the construction of a design-level kinetic descriptor suitable for downstream Active Learning.

5.2.1 Joint model on reference trajectories

Let $c = 1, \dots, C$ index the normalized fluorescence trajectories obtained by pooling together all curves belonging to the two reference conditions, RefA and RefB, across the available experimental days. For each trajectory c, let $i = 1, \dots, T$ index the post-transient time points, and define

$$dt_i = t_i - t_0, \qquad t_0 = 50 \text{ minutes}. \tag{5.14}$$

Each trajectory c is associated with two categorical indices:

$$\text{ref}(c) \in \{1,2\}, \qquad \text{day}(c) \in \{1, \dots, D\}, \tag{5.15}$$

where $\text{ref}(c)$ identifies the reference condition (RefA or RefB) and $\text{day}(c)$ denotes the experimental day.

For the reference trajectories, the normalized fluorescence observations are modeled as

$$y_{c,i} \sim \mathcal{N}\big(b_c + (1 - b_c)\exp(-k_c\, dt_i),\ \sigma_y^2\big), \qquad i = 1, \dots, T. \tag{5.16}$$

Here, $k_c > 0$ denotes the degradation rate associated with trajectory c, while b_c represents its normalized asymptotic fluorescence level. In the joint model, the parameters b_c are not estimated through a prior distribution but are treated as plug-in quantities constructed directly from the observed trajectories. More precisely, for each trajectory c, the quantity b_c is set equal to a robust empirical summary of the tail of the corresponding normalized curve, computed as the median of the last K observed time points. In the implementation adopted here, $K = 5$. Therefore,

$$b_c := b_{0,c}, \tag{5.17}$$

where $b_{0,c}$ is a deterministic quantity derived from the data.
This choice represents a deliberate simplification of the model. By fixing b_c rather than estimating it jointly with the remaining parameters, the dimensionality of the posterior distribution is reduced and the identifiability of the rate and day-effect parameters is improved. In preliminary attempts, more flexible formulations allowing additional batch effects or random effects on the asymptotic level led to unstable chains and poor convergence diagnostics, due to the strong dependence between the parameters controlling the tail level and those governing the decay rate.
The trajectory-specific degradation rates are modeled on the logarithmic scale according to

$$\log k_c = \mu_{\mathrm{ref}(c)} + \alpha_{\mathrm{day}(c)} + u_c, \tag{5.18}$$

where

- μ_1 and μ_2 denote the reference-specific log-rates associated with RefA and RefB, respectively;
- α_d is the day-specific batch effect on the log-rate scale;
- u_c is a trajectory-specific random deviation.

The residual trajectory-level terms are assumed to satisfy

$$u_c \sim \mathcal{N}(0, \sigma_k^2), \qquad c = 1, \dots, C, \tag{5.19}$$

independently across trajectories.
To ensure identifiability of the additive decomposition in Eq. 5.18, one experimental day (Day 3 in our case, since it is the first day the first batch set is tested) is selected as baseline, say d_0, and the corresponding day effect is fixed to zero:

$$\alpha_{d_0} = 0. \tag{5.20}$$

On the original rate scale, the day effect is represented by the multiplicative factor

$$R_d := \exp(\alpha_d), \qquad d = 1, \dots, D. \tag{5.21}$$

Hence, the trajectory-specific rates can be written as

$$k_c = \exp\big(\mu_{\mathrm{ref}(c)}\big)\, R_{\mathrm{day}(c)}\, \exp(u_c). \tag{5.22}$$

Posterior inference in this first stage is performed via MCMC in JAGS. The parameters directly sampled from the posterior are

$$\{\mu_1, \mu_2, \alpha_d, u_c, \sigma_y, \sigma_k, \sigma_\alpha\}, \tag{5.23}$$

while the quantities

$$k_{\mathrm{ref},r} = \exp(\mu_r), \qquad R_d = \exp(\alpha_d) \tag{5.24}$$

are deterministic transformations monitored as derived posterior quantities. If $\theta^{(m)}_{\text{joint}}$ denotes the m-th posterior draw from the joint reference model, then posterior inference on the batch effect is based on the induced samples

$$R_d^{(m)} = \exp\big(\alpha_d^{(m)}\big). \tag{5.25}$$

Posterior medians, credible intervals, and other summaries of the batch effect are then obtained directly from the sample $\{R_d^{(m)}\}_{m=1}^M$.

5.2.2 Design-level model

In the second stage, posterior inference is performed separately for each sheet corresponding to a specific design or reference condition. Let $c = 1, \dots, C_{j,d}$ index the normalized trajectories associated with a given design j, at day d, and let $i = 1, \dots, T$ index the post-transient time points. For notational simplicity, within each design-level fit the curve index c is used instead of the full triplet (j, d, r), while the corresponding day label is retained externally and used later for batch correction.

For each trajectory c in design j and day d, the normalized fluorescence is modeled as

$$y_{c,i} \sim \mathcal{N}\big(b_c + (1 - b_c)\exp(-k_c\, dt_i),\ \sigma_y^2\big)\,, \qquad i = 1, \dots, T. \tag{5.26}$$

In contrast with the joint reference model, the asymptotic levels b_c are not fixed here but are treated as unknown trajectory-specific parameters and estimated a posteriori. This choice provides additional flexibility in the fit of individual design trajectories, at the price of increased parameter uncertainty. In the implementation, each b_c is assigned the prior

$$b_c \sim \text{Beta(2,2)}, \qquad c = 1, \dots, C_{j,d}, \tag{5.27}$$

which places mass on the interval (0,1) and mildly favors interior values.

The trajectory-specific rates are modeled on the log scale as

$$\log k_c = \mu_j + u_c, \tag{5.28}$$

where

- μ_j is the design-level log-rate associated with design j, estimated with curves on the same day d;
- u_c is a residual trajectory-specific deviation.

The residual terms satisfy

$$u_c \sim \mathcal{N}(0, \sigma_k^2), \qquad c = 1, \dots, C_{j,d}, \tag{5.29}$$

independently across curves. Hence, the curve-specific rates can be written as

$$k_c = \exp(\mu_j)\exp(u_c). \tag{5.30}$$

In order to avoid mixing variability originating from different experimental days, the above model is not fitted simultaneously to all trajectories of a given design. Instead, for each design j the trajectories are partitioned according to their experimental day, and the model is fitted separately to the subset of curves corresponding to each day.

More precisely, let $\mathcal{D}_j$ denote the set of experimental days on which design j was measured. For each $d \in \mathcal{D}_j$, the model in Eq. 5.26–5.28 is fitted using only the trajectories observed on that specific day. In this way, a separate posterior distribution for the parameter μ_j is obtained for each day on which design j was measured.

Posterior inference for each design–day subset is again performed via MCMC in JAGS. The parameters directly sampled from the posterior are

$$\{\mu_j, u_c, b_c, \sigma_y, \sigma_k\}, \tag{5.31}$$

while the quantities

$$k_c = \exp(\mu_j + u_c), \qquad k_j = \exp(\mu_j) \tag{5.32}$$

are monitored as deterministic transformations.
If

$$\theta_j^{(s)} = \big(\mu_j^{(s)}, u^{(s)}, b^{(s)}, \sigma_y^{(s)}, \sigma_k^{(s)}\big) \tag{5.33}$$

denotes the s-th posterior draw of the design-level model for a fixed design–day subset, then the induced posterior draw of the curve-specific rate is

$$k_c^{(s)} = \exp\big(\mu_j^{(s)} + u_c^{(s)}\big). \tag{5.34}$$

It is important to stress that, at this stage, the design-level model does not include the day-specific batch effect explicitly. Consequently, the posterior draws $\{k_c^{(s)}\}$ should be interpreted as uncorrected curve-specific degradation rates associated with a given design and experimental day. The corresponding day-specific correction is introduced only after the design-level fits have been completed, using the posterior distribution of the batch factors R_d estimated from the joint reference model.

Posterior sampling and interpretation of the two-stage fit

Let

$$\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(S)} \tag{5.35}$$

denote the posterior draws produced by MCMC from a generic JAGS model. In the present work, posterior inference is entirely sample-based: unknown parameters are not summarized only through point estimates, but through the empirical distribution of the sampled values. Posterior means, medians, credible intervals, and derived quantities are all computed from these samples.
In the joint reference model, the posterior sample $\{\theta_{\text{joint}}^{(m)}\}_{m=1}^{M}$ is used to infer the distribution of the day-specific batch factors $\{R_d^{(m)}\}_{m=1}^{M}$.
In the design-level analysis, separate posterior samples are obtained for each subset of trajectories corresponding to a given design j and experimental day d. For each such subset, the posterior draws $\{\theta_j^{(s)}\}_{s=1}^{S_j}$ induce posterior samples of the trajectory-specific degradation rates

$$\{k_c^{(s)}\}_{s=1}^{S_j}, \tag{5.36}$$

where c indexes the trajectories belonging to design j observed on day d. These samples represent the posterior distribution of the trajectory-specific degradation rates before batch correction.
The two-stage structure implies that the final inference is not based on a single global posterior distribution over all unknown quantities. Instead, posterior samples obtained from the first-stage reference model and from the second-stage design-level models are combined afterwards in order to construct batch-corrected degradation rates. This strategy does not correspond to full joint posterior inference, but it allows the uncertainty on the batch effect and the uncertainty on the trajectory-level rate estimates to be combined through a subsequent Monte Carlo correction step, described in the following section.
From a practical perspective, this two-stage formulation has several advantages. The joint estimation of day effects is stabilized by using only the reference trajectories, which are specifically intended to track inter-day variability and are repeatedly observed across multiple days. At the same time, the design-level fits remain relatively simple, because they do not require the explicit inclusion of batch effects inside every individual nonlinear model.
This simplification comes at the price of a partial decoupling between the estimation of batch effects and the estimation of design-specific rates. In particular, the day effects α_d are inferred only from the reference trajectories and are subsequently transferred to the design-level models through batch correction, rather than being jointly estimated with all design-specific parameters in a single

hierarchical fit. The resulting procedure should therefore be interpreted as a computationally stable approximation of the fully joint Bayesian model, rather than as its exact posterior implementation.

Batch-corrected degradation rates and label construction

After the two stages of posterior inference described above have been completed, the final goal of the analysis is to construct *batch-corrected degradation rates* and, subsequently, a single kinetic label associated with each experimental design.
As discussed in the previous sections, the joint reference model provides posterior samples of the day-specific batch factors

$$\{R_d^{(m)}\}_{m=1}^{M}. \tag{5.37}$$

In the second stage, design-level models are fitted separately for each design j and for each experimental day on which that design was measured. For a given design j and day d, the trajectories observed on that day are modeled through

$$\log k_c = \mu_j + u_c, \tag{5.38}$$

where c indexes the trajectories belonging to design j and day d.
From the corresponding posterior sample of the design-level fit, posterior draws of the trajectory-specific degradation rates are obtained as

$$k_c^{(s)} = \exp\bigl(\mu_j^{(s)} + u_c^{(s)}\bigr), \qquad s = 1, \dots, S_j. \tag{5.39}$$

These samples represent the posterior distribution of the degradation rates associated with trajectory c before correcting for the day-specific batch effect. The experimental day corresponding to trajectory c is denoted by $d(c)$.

Construction of corrected rates. Since the posterior samples of k_c and R_d originate from two separate MCMC simulations, the corrected degradation rates are not obtained through a one-to-one correspondence between posterior draws. Instead, they are constructed through a Monte Carlo combination of the two posterior distributions.
More precisely, for each trajectory c, a collection of batch-corrected samples is generated as

$$k_c^{\mathrm{corr},(\ell)} = \frac{k_c^{(s_\ell)}}{R_{d(c)}^{(m_\ell)}}, \qquad \ell = 1, \dots, L_c, \tag{5.40}$$

where

$$s_\ell \in \{1, \dots, S_j\}, \qquad m_\ell \in \{1, \dots, M\}.$$

In practice, the indices s_ℓ and m_ℓ are selected so as to generate a Monte Carlo sample from the distribution of the ratio $k_c/R_{d(c)}$. In the implementation adopted here, the number of corrected samples L_c is chosen equal to the number of available posterior draws of k_c, while the corresponding batch factors are sampled with replacement from the posterior sample of $R_{d(c)}$. This procedure produces a Monte Carlo approximation of the posterior distribution of the batch-corrected rate associated with trajectory c.

Aggregation across trajectories of the same design. The objective of the analysis is not to estimate a separate kinetic parameter for each trajectory, but rather to associate a single degradation descriptor with each experimental design.

For this reason, the corrected samples obtained for the individual trajectories are subsequently aggregated at the design level.
Let

$$\mathcal{C}_j \tag{5.41}$$

denote the set of trajectories associated with design j. For each trajectory $c \in \mathcal{C}_j$, the procedure described in Eq. 5.40 produces a collection of corrected samples

$$\left\{k_c^{\mathrm{corr},(\ell)}\right\}_{\ell=1}^{L_c}. \tag{5.42}$$

All corrected samples corresponding to trajectories of design j are then concatenated into a single Monte Carlo sample

$$K_j = \bigcup_{c \in \mathcal{C}_j} \left\{k_c^{\mathrm{corr},(\ell)}\right\}_{l=1}^{L_c}. \tag{5.43}$$

This construction yields a large empirical sample representing the distribution of batch-corrected degradation rates associated with design j, combining information from all trajectories and incorporating both trajectory-level uncertainty and batch-effect uncertainty.

Log-scale representation and label definition. Since the statistical modeling of degradation rates is performed on the logarithmic scale, the aggregated corrected sample is transformed as

$$\log K_j = \{\log k : k \in K_j\}. \tag{5.44}$$

The final kinetic label associated with design j is then defined as the posterior median of the log-corrected degradation rates:

$$\widehat{\mu}_j = \mathrm{median}(\log K_j). \tag{5.45}$$

This quantity represents a robust summary of the corrected degradation dynamics of design j. Working on the log scale ensures coherence with the statistical model used to describe the trajectory-specific rates and provides a symmetric representation of multiplicative uncertainty.

Interpretation of the aggregation procedure. It is important to stress that the label construction does not reduce each trajectory to a single point estimate before aggregation. Instead, each trajectory contributes its full posterior distribution of corrected rates to the final sample K_j. This strategy preserves the uncertainty structure produced by the MCMC inference and avoids the loss of information that would arise from summarizing each trajectory separately prior to aggregation.
As a consequence, the final label for design j incorporates several sources of uncertainty simultaneously:

- uncertainty in the estimation of the trajectory-specific rates k_c;
- uncertainty in the day-specific batch factors R_d;
- variability between trajectories belonging to the same design.

Through the Monte Carlo combination and aggregation steps described above, these sources of uncertainty are propagated to the final kinetic descriptor used for downstream analysis and optimization.

From Bayesian Inference to Active Learning–Based Regression The Bayesian inference procedure described above provides, for each experimental design j, a kinetic label defined as the posterior median of the batch-corrected log degradation rates. Denoting this quantity by $\hat{\mu}_j$, we obtain a dataset of input–output pairs

$$\mathcal{D} = \{(\mathbf{x}_j, \hat{\mu}_j)\}_{j=1}^{n}, \tag{5.46}$$

where $\mathbf{x}_j$ represents the experimental configuration associated with design j, and $\hat{\mu}_j$ summarizes the corresponding degradation kinetics.

In the next stage of the analysis, these data are used to learn a functional relationship between the design variables and the degradation rate. In particular, we consider a regression model of the form

$$\mu_j = g(\mathbf{x}_j), \tag{5.47}$$

where $g(\cdot)$ denotes an unknown function mapping the experimental design parameters to the corresponding degradation dynamics.

For the purpose of the Active Learning procedure introduced in the next section, the quantities $\hat{\mu}_j$ are treated as point estimates of the underlying kinetic parameter. Although the Bayesian inference framework provides posterior uncertainty for these quantities, this information is not explicitly incorporated in the regression model. This choice is motivated by the desire to keep the learning procedure computationally simple and to focus the optimization step on predicting the average degradation behavior across the design space.

Chapter 6

Active Learning Framework

In this section, the methodology used to efficiently explore the experimental design space and identify conditions that maximize the degradation rate is described in detail. The approach relies on an Active Learning strategy implemented through an ensemble of Random Forests composed of regression trees.

6.1 Decision Tree Regression

Regression trees are nonparametric predictive models belonging to the class of supervised learning methods based on the recursive partitioning of the input variable space. The objective of a regression tree is to approximate an unknown function

$$f : \mathbb{R}^p \to \mathbb{R}$$

that links the feature vector $\mathbf{x} = (x_1, ..., x_p)$ to a continuous response variable y.
Regression trees are typically formulated as scalar-output models, mapping a feature vector $\mathbf{x} \in \mathbb{R}^p$ to a single response variable $y \in \mathbb{R}$. While extensions to multi-output regression exist, the standard formulation predicts one response variable at a time.
In this work, following what has already been introduced in Sec. 4, the input vector takes the form given in Eq. 4.1, containing the initial concentrations of the components of the cell-free system; the output variable is instead given by the degradation log-rate μ associated with each initial setup.

A regression tree constructs such an approximation by iteratively subdividing the feature space into disjoint regions through binary splits of the form

$$x_k \leq \tau \tag{6.1}$$

where x_k is an explanatory variable, i.e. an element of $\mathbf{x}$, and τ is a threshold. This recursive partitioning generates a hierarchical structure composed of internal nodes (decision nodes) and terminal leaves (terminal nodes). Each leaf corresponds to a region of the explanatory variable space within which the model prediction is constant.

In the regression setting, the prediction associated with a region R_m is generally defined as the average of the responses corresponding to the vectors $\mathbf{x}_i$ belonging to that region:

$$\hat{f}(\mathbf{x}) = \frac{1}{|R_m|} \sum_{i:\mathbf{x}_i \in R_m} y_i, \qquad x \in R_m \tag{6.2}$$

The construction of the tree follows a recursive splitting procedure that selects, at each node, the feature and threshold generating the most informative partition of the data. The specific criterion used to evaluate candidate splits and the corresponding optimization procedure are described in detail in the following section.

Properties and limitations Regression trees exhibit several relevant properties. First, they are able to model nonlinear relationships and interactions between variables without requiring explicit functional specifications. Moreover, they are flexible and easily interpretable due to their hierarchical structure.
However, individual trees also present important limitations. In particular, their predictions may exhibit high variance, meaning that small variations in the training dataset may lead to significantly different tree structures and therefore unstable predictions. In addition, the piecewise-constant approximation induced by the tree may result in limited predictive performance on unseen data when some regions of the feature space are poorly represented by the available training observations.
For these reasons, decision trees are often used as base learners within ensemble methods, where the predictions of multiple trees are aggregated in order to obtain a more stable estimator. This idea forms the basis of methods such as Random Forests, which will be introduced in the following sections.

We now proceed to illustrate the learning procedure of a single tree.

6.1.1 Learning procedure of a regression tree

The construction of a regression tree follows a recursive partitioning procedure of the feature space.
Let D be the training dataset

$$D = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n}, \tag{6.3}$$

where $\mathbf{x}_j$ represents the vector of the p features of design j and y_j is the corresponding response variable. In the work presented here, it refers to the log-rate of the protein degradation process, obtained by applying the JAGS framework starting from the observed data (for further details see Sec. **??**).
The objective of the algorithm is to partition the feature space into disjoint regions $R_1, R_2, ..., R_M$ in such a way as to reduce the prediction error within each region.
The tree is constructed node by node.
The process starts from a root node, which receives the n pairs $(\mathbf{x}_j, y_j)$. For this node, according to the procedure reported below, the feature k and the threshold τ are selected in order to generate a split, which leads to the formation of a right node and a left node, to which the n data points are assigned.
All nodes are characterized by the *mean squared error*, MSE, which evaluates a form of "impurity", that is, how much the responses of the individual data points deviate from the mean response of the node.
Given a generic node, let N be the set of indices of the pairs $(\mathbf{x}_j, y_j)$ assigned to the node, with cardinality $n_N = |N|$ (for the root node, trivially $n_N = n$). Its impurity, before the split, is defined as:

$$I_N := MSE(N) = \frac{1}{n_N} \sum_{i \in N} (y_i - \bar{y}_N)^2 \tag{6.4}$$

where $\bar{y}_N$ is the mean of the response values within the node, therefore:

$$\bar{y}_N := \frac{1}{|N|}\sum_{i\in N} y_i \tag{6.5}$$

Among all admissible binary splits, the one that reduces as much as possible the impurity transmitted from the parent node to the child nodes is selected.
In particular, consider for example the root node, with impurity:

$$I_{N^{root}} = \frac{1}{n}\sum_{i\in N^{root}} (y_i - \bar{y}_N^{root})^2 \tag{6.6}$$

Let L be the set of indices of the pairs $(\mathbf{x}_j, y_j)$ assigned to the left node and R the set assigned to the right node, with cardinalities respectively $n_L = |L|$ and $n_R = |R|$.
The child nodes are characterized by the *mean squared error*, MSE, which evaluates a form of node "impurity". Let L be the set of indices of the pairs $(\mathbf{x}_j, y_j)$ assigned to the left node and R the set assigned to the right node. Their cardinalities are respectively indicated as $n_L = |L|$ and $n_R = |R|$.

$$I_L := MSE(L) = \frac{1}{n_L}\sum_{i\in L}(y_i - \bar{y}_L)^2 \qquad I_R := MSE(R) = \frac{1}{n_R}\sum_{i\in R}(y_i - \bar{y}_R)^2 \tag{6.7}$$

where, as before, $\bar{y}_L$ and $\bar{y}_R$ are the mean response values within the left and right nodes. It is now possible to introduce a measure of the impurity of the parent node (in this case, the root) after the split:

$$I_{N^{root}}^{split} := \frac{n_L}{n} I_L + \frac{n_R}{n} I_R \tag{6.8}$$

This is the weighted mean of the MSE values of the child nodes.
Finally, defining the **gain**(k, τ) as the variation of the parent node impurity after the split $x_k \leq \tau$

$$gain_{N^{root}}(k, \tau) := I_{N^{root}} - I_{N^{root}}^{split} \tag{6.9}$$

it is now possible to formalize the identification of the *optimal* binary split. The latter is defined by the pair (k^*, τ^*) that maximizes the reduction of the mean squared error of the output variable produced by the split:

$$(k^*, \tau^*) = arg \max_{k,\tau} gain_{N^{root}}(k, \tau) \tag{6.10}$$

with $\tau \in \mathbb{R}$ and $k \in K$, where K contains the indices associated with the features that are candidates to generate the split. Further details will be provided later.

Once the best split has been selected, the dataset is divided into the two resulting subsets ($L = \{i : x_{i,k^*} \leq \tau^*\}$ and $R = \{i : x_{i,k^*} > \tau^*\}$) and the procedure is applied recursively to the child nodes, updating their impurity value.
The process continues until certain stopping conditions are reached. A node can no longer branch, therefore becoming a leaf, when:

- The maximum depth of the tree is reached;
- It reaches the minimum number of data points it must contain;
- The reduction of the error after the split is not significant;
- The node is pure, meaning that it contains only data with the same output variable.

The algorithm terminates when no further splits can be performed.
The terminal nodes generated by the algorithm define the regions R_m of the feature space: the initial n data points are distributed among the leaves, which therefore contain a subset of the training data.
When the tree evaluates a point x not belonging to the training set, it follows the splits defined at the different nodes from the root until it reaches a certain leaf m, and the value associated with it corresponds to the prediction associated with the region R_m, which is the mean of the response variables of the data belonging to that region (whose indices belong to the set $M = \{i : \mathbf{x}_i \in R_m\}$) and therefore contained in that leaf:

$$\hat{f}(x) = \bar{y}_M = \frac{1}{|M|} \sum_{i \in M} y_i \tag{6.11}$$

The algorithm just described corresponds to CART (Breiman et al., 1984)[7, 8].

6.1.2 Advantages and Drawbacks

The main advantages and limitations of regression trees are summarized below.

- Capability to model highly nonlinear relationships between input variables and the response without requiring explicit functional assumptions; Interactions between variables are automatically captured through the hierarchical structure of the tree.

- Simple interpretation: The sequence of binary splits provides a transparent representation of the decision process, making it possible to understand how different features contribute to the final prediction.

- Low bias but high variance: The low bias of regression trees stems from the limited prior assumptions imposed on the underlying regression function. Unlike parametric models, which require specifying a functional form in advance, regression trees adaptively partition the feature space and therefore can approximate complex nonlinear relationships. However, this flexibility comes at the cost of high variance, since splits are based on the specific trainset used. Small variations in the training dataset may lead to significantly different predictions. In other words, the estimator is unstable with respect to the training data. This instability generally limits the predictive accuracy of a single tree. Although deep trees can fit the training data very well, they often suffer from *overfitting*, meaning that the model captures noise present in the training set rather than the underlying signal.

- Ability to learn from relatively small datasets: Since regression trees do not rely on strong parametric assumptions and do not require the estimation of a large number of parameters, they can often learn useful relationships even when the available dataset is limited.

- Poor predictive performance on unseen data: the prediction within each region R_m is constant (average response of the training observations in R_m). The model implicitly assumes that the regression function is approximately constant within each partition of the feature space. If the true response function varies significantly inside a region R_m, the tree cannot capture this variation, since all points in that region are assigned the same predicted value. This situation may occur, for example, when a region contains only a small number of training observations but corresponds to a relatively large portion of the feature space. In such cases, many unseen points may fall inside the same region while having response values that are substantially different from those observed in the training set, leading to poor predictive accuracy.In other words, when observations with similar feature values correspond to very different responses, the piecewise-constant approximation induced by the tree becomes inadequate.

6.1.3 From Decision Trees Regression to Random Forest

In the present work, tree-based models were adopted due to their ability to learn meaningful relationships even when the available dataset is relatively small. In contrast to highly parameterized models such as neural networks, regression trees do not require the estimation of a large number of parameters and therefore can often be trained effectively in low-data regimes. This property makes them particularly suitable for experimental settings where the number of available observations is limited.

Despite these advantages, individual regression trees present well-known limitations. In particular, their predictions may exhibit high variance, meaning that small variations in the training data can lead to significantly different tree structures and consequently unstable predictions. Moreover, the piecewise-constant approximation induced by the tree may result in limited predictive performance on unseen data when the regions of the feature space are poorly represented by the available training observations. As will be discussed later in the results section, this latter limitation cannot be completely mitigated in the present study due to the intrinsic structure of the predictive model and the limited size of the dataset.

To mitigate the instability associated with single trees, the model adopted in this work relies on an ensemble of decision trees combined within a Random Forest framework. Aggregating the predictions of multiple trees allows for a significant reduction in the variance of the estimator while preserving the flexibility of tree-based models. The construction of Random Forests and the ensemble learning mechanisms underlying them will be described in the following section.

6.2 Random Forests

As discussed in the previous section, individual regression trees represent a flexible and interpretable model, but their predictions may exhibit high variance. Small variations in the training dataset may indeed lead to the construction of significantly different tree structures, thus producing unstable predictions.

To address this limitation, it is possible to resort to **ensemble learning** methods, which combine the predictions of multiple models in order to obtain a more stable estimator. Among these approaches, **Random Forests** constitute one of the most widely used ensemble techniques based on decision trees.

A regression Random Forest is a model composed of a collection of regression trees. Each tree produces a prediction for a given input vector $\mathbf{x}$, and the final prediction of the forest is obtained by aggregating the predictions of the individual trees, typically through averaging.

Formally, let $\hat{f}^{(b)}(\mathbf{x})$ denote the prediction of the b-th regression tree in the ensemble, with $b = 1, \dots, B$. The Random Forest prediction is then defined as

$$\hat{f}_{RF}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(\mathbf{x}) \tag{6.12}$$

The key idea behind Random Forests is that aggregating the predictions of multiple trees allows for a significant reduction in the variance of the estimator, while preserving the ability of decision trees to approximate nonlinear relationships between the variables. In particular, when the predictions of the individual trees are sufficiently diverse, their average tends to provide a more stable and accurate estimate of the underlying regression function.

The mechanisms used to generate this diversity among trees, which constitute the core of the Random Forest algorithm, will be described in the following sections.

6.2.1 Bootstrap Sampling

Bootstrap sampling is a resampling technique used to generate multiple training datasets from a single observed dataset. Each bootstrap sample is obtained by drawing observations from the original dataset *with replacement*, while preserving the original sample size.
As a result, each resampled dataset contains a slightly different composition of observations, with some points appearing multiple times and others possibly omitted.
In ensemble learning methods such as Random Forests, bootstrap sampling is used to construct different training sets for the individual trees forming the ensemble. This mechanism introduces variability among the trees, which contributes to reducing model variance and improving generalization performance.
The usual training dataset is considered

$$D = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n}, \tag{6.13}$$

where the feature vector satisfies $\mathbf{x}_j \in \mathbb{R}^p$ and the response variable satisfies $y_j \in \mathbb{R}$. This dataset represents the only available information about the process: the values of y are unknown outside this set.
We now draw n i.i.d. samples from a uniform distribution:

$$i_1, i_2, ..., i_n \sim \mathcal{U}(S), \qquad S = \{1,2,...,n\}. \tag{6.14}$$

These correspond to the indices associated with the n original observations.
The bootstrap dataset (or bootstrap sample) is therefore defined as

$$D^{(b)} = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), ..., (x_{i_n}, y_{i_n})\}. \tag{6.15}$$

As a direct consequence, some observations may appear multiple times, while others may not appear at all. It can be shown that, on average, a bootstrap sample contains approximately 63.2% distinct observations from the original dataset (the remaining observations are repetitions), while about 36.8% of the original observations are not selected and therefore remain out-of-bag.
The bootstrap dataset therefore has the same size n as the original dataset, but a slightly different composition. In this way, a controlled perturbation of the original dataset is generated.
If B denotes the number of trees composing the Random Forest, repeating the resampling procedure B times produces a collection of bootstrap datasets $D^{(1)}, D^{(2)}, ..., D^{(B)}$ on which B distinct trees can be trained.
Denoting by $\hat{f}^{(b)}$ the model obtained by training a tree on the dataset $D^{(b)}$, bootstrap sampling yields a collection of models $\hat{f}^{(1)}, \hat{f}^{(2)}, ..., \hat{f}^{(B)}$, each of which is generally different from the others. This property is fundamental: when the predictions of the different trees are subsequently aggregated (for instance by averaging), the resulting estimator is more stable and exhibits lower variance than that of a single tree.

Once multiple models have been trained on different bootstrap samples, their predictions must be combined to produce a single estimator. This is achieved through the bootstrap aggregating procedure, commonly referred to as **bagging**.

6.2.2 Bootstrap Aggregating (Bagging)

Bagging[9, 8] is an ensemble learning technique that consists of training multiple models on different bootstrap samples of the original dataset and subsequently aggregating their predictions.

Let $\hat{f}^{(1)}, \hat{f}^{(2)}, \ldots, \hat{f}^{(B)}$ denote the B models trained on the bootstrap datasets $D^{(1)}, D^{(2)}, \ldots, D^{(B)}$. Given a new input vector x, each model produces an individual prediction $\hat{f}^{(b)}(x)$. In the regression setting, the bagging estimator is obtained by averaging the predictions of all models:

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(x). \tag{6.16}$$

The primary motivation for bagging is the reduction of the variability of the estimator. Decision trees, which are commonly used as base learners, are known to be unstable models: small perturbations in the training dataset may lead to significantly different tree structures and, therefore, unstable predictions. By training multiple models on slightly different datasets and averaging their predictions, bagging stabilizes the estimator and mitigates the variability associated with a single model.

Bagging therefore provides a simple yet effective mechanism to improve the stability and predictive performance of models that are sensitive to small perturbations in the training data. In the context of tree-based methods, this approach constitutes the foundation upon which Random Forest models are built.

Another source of randomness, which enhances the diversity among trees is the **Feature Sampling**, which is briefly described below:

6.2.3 Random Feature Sampling

Although bagging reduces the variability of unstable predictors such as decision trees, the individual models trained on bootstrap samples may still exhibit similar structures. This occurs because, during the tree construction process, the same highly informative features may repeatedly be selected to generate splits.

Random Forests address this issue by introducing an additional source of randomness during the tree construction process. Instead of considering the full set of p input features when searching for the best split at each node, the algorithm randomly selects a subset of m features, with $m < p$. The optimal split is then chosen only among the variables belonging to this subset.

Formally, let $\mathbf{x} \in \mathbb{R}^p$ denote the input feature vector. At each node of the tree, a subset of m candidate features is sampled uniformly from the set $\{1, \ldots, p\}$, and the splitting rule is determined by evaluating only these features.

This random feature selection mechanism increases the diversity of the trees composing the ensemble, leading to models with different structures and decision boundaries. As a consequence, the aggregation of their predictions becomes more effective, improving the stability and predictive performance of the overall model.

In typical implementations of Random Forests, the number of candidate features m is chosen as $m = \sqrt{p}$ for classification problems and $m = p/3$ for regression problems, although different values may be adopted depending on the specific application.

In this work $m =$ was used during the Active Learning algorithm; instead, the final model, after hyperparameter tuning, $m =$ was set.

Combining bootstrap sampling with random feature selection leads to the Random Forest learning algorithm, which is described in the following section.

6.2.4 Random Forest Learning Algorithm

The Random Forest learning procedure described in the previous sections can be summarized through the following algorithm.

1. **Input:** training dataset
$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \tag{6.17}$$
number of trees B, and number of candidate features m.

2. **For** $b = 1, \ldots, B$:
    (a) Generate a bootstrap dataset $D^{(b)}$ by sampling n observations with replacement from D.
    (b) Train a regression tree $\hat{f}^{(b)}$ on $D^{(b)}$.
    (c) During tree construction, at each node select a random subset of m features from $\{1, \ldots, p\}$ and determine the optimal split using only these features.

3. **Output:** an ensemble of regression trees
$$\hat{f}^{(1)}, \hat{f}^{(2)}, \ldots, \hat{f}^{(B)}. \tag{6.18}$$

4. **Prediction:** for a new input vector $\mathbf{x}$, the Random Forest prediction is obtained by averaging the predictions of the individual trees:
$$\hat{f}_{RF}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{(b)}(\mathbf{x}). \tag{6.19}$$

6.2.5 Advantages and Limitations of Random Forests

Random Forest models inherit many properties of regression trees, such as the ability to model nonlinear relationships, the absence of strong parametric assumptions, and the fact that feature normalization is not required. These properties were discussed in Sec. 6.1.2 and naturally extend to tree-based ensemble methods.

In addition to these characteristics, Random Forests introduce several important advantages that arise from the ensemble learning framework.

Advantages The primary advantage of Random Forests is the reduction of the variance associated with individual decision trees. Since trees are known to be unstable predictors, small variations in the training dataset may lead to significantly different tree structures. By aggregating the predictions of multiple trees trained on different bootstrap samples, Random Forests produce a more stable estimator and improve predictive robustness.

Another important property is the increased robustness to irrelevant or weakly informative variables. Because each split considers only a randomly selected subset of candidate features, the algorithm reduces the risk that the same dominant variables are repeatedly selected across all trees, allowing the ensemble to explore different structures of the feature space.

Furthermore, the combination of bootstrap sampling and random feature selection enables Random Forests to operate effectively even when the number of input variables is relatively large, since different subsets of features are explored across the trees composing the ensemble.

Finally, Random Forests can often provide reasonable predictive performance even when the available dataset is relatively small. Since tree-based models do not require the estimation of a large number of parameters, they can learn useful relationships even in experimental contexts where collecting large amounts of data is costly or time-consuming.

Limitations Despite these advantages, Random Forests also present some limitations.
First, the interpretability of the model is reduced compared to a single decision tree. While the structure of an individual tree can be easily interpreted, the aggregation of many trees makes the overall prediction mechanism more difficult to analyze.
Second, Random Forest models generally require higher computational resources than simpler models. Training a large number of trees increases both computational time and memory requirements, particularly when the ensemble size or the dimensionality of the feature space is large.
Another important limitation concerns predictive performance on unseen data when the training dataset sparsely covers the feature space. Although ensemble averaging reduces the variance of individual trees, the model is still trained only on bootstrap resamples of the original dataset and therefore does not introduce new information about the underlying response function outside the observed data.
Consequently, when regions of the feature space contain few training observations, many unseen points may fall within those regions while having response values that differ significantly from the training samples. In such situations, the Random Forest estimator may produce inaccurate predictions, since the ensemble essentially averages predictions derived from the same limited information contained in the original dataset.

6.3 Predictive model choice

The objective of the predictive model in this work is to learn the relationship between the biochemical composition of the system and the degradation dynamics of the fluorescent protein. As described in Sec. **??**, the experimental fluorescence curves are modeled through a normalized exponential decay characterized by two parameters: the asymptotic level B (with amplitude $1 - B$) and the degradation rate k.
In principle, both parameters could be used as targets for the predictive model. However, in the present study the learning task is formulated as a regression problem with a single response variable, corresponding to the logarithm of the degradation rate, $\mu = \log k$. The goal of the predictive model is therefore to identify experimental conditions $\mathbf{x}$ that maximize this quantity.
This choice is motivated by both modeling and practical considerations. From a biological perspective, the degradation rate represents the most relevant quantity for characterizing the efficiency of the enzymatic process. From a machine learning perspective, focusing on a single scalar target simplifies the learning problem and allows the use of robust regression models that perform well in small-data regimes.
Alternative approaches could involve predicting multiple parameters simultaneously, such as both B and k, using multi-output models. However, given the discrete nature of the experimental design space and the relatively small size of the available dataset, priority was given to model stability and robustness rather than to increasing the dimensionality of the prediction task.

6.4 Active Learning

6.4.1 Introduction

Active Learning (AL) is a supervised learning paradigm designed for settings in which obtaining new observations is costly, either in terms of time, materials, or experimental resources. Instead of collecting data passively or randomly, Active Learning iteratively builds a predictive model and uses it to selectively identify the most informative points at which new measurements should be performed, with the goal of maximizing the information gained per unit of experimental cost. Active Learning is typically structured as a sequential cycle:

1. an initial dataset of observed experiments is collected;
2. a **surrogate model** $\hat{f}(\mathbf{x})$ is trained in order to approximate the relationship between the system composition $\mathbf{x}$ and the response variable y;
3. an **acquisition function** $a(\mathbf{x})$ is evaluated over a set of candidate points that have not yet been tested. The acquisition function balances two competing objectives:
   - **exploitation**: selecting points with high predicted response values;
   - **exploration**: selecting points in regions of the feature space where the model is more uncertain or where the available data provide limited support;
4. experiments are then performed on the selected points, the dataset is updated with the newly acquired observations, and the process is repeated until a stopping criterion is satisfied.

The main strength of Active Learning lies in its sample efficiency. When the number of experiments that can be performed is limited, AL aims to rapidly construct an informative dataset and guide the search toward promising configurations, avoiding a uniform exploration of the design space that would otherwise be impractical.

For this reason, the method is particularly suitable for the experimental context considered in this work. The objective of the present study is to identify initial configurations of the cell-free system that lead to maximal protein degradation. In this setting, the space of possible combinations (ATP, Mg, ClpX, ClpP, PEG) is large and discrete, and each additional observation requires a non-negligible experimental cost in terms of time, instrumentation, and biological components.

In the present work, the Active Learning strategy is inspired by the framework proposed in the literature for the optimization of cell-free systems, but it is adapted to the specific characteristics of the experimental platform considered here. In particular, the experiments are performed in a cell-free system based on an *E. coli* lysate, different from the one used in the reference study, and specifically configured to reproduce protein degradation dynamics. The system includes the exogenous proteolytic complex ClpXP, which enables the degradation of the fluorescent protein sfGFP carrying an ssrA degradation tag. As a result, the Active Learning procedure is employed to guide the exploration of experimental conditions that maximize the degradation rate rather than protein production.

After introducing the general principles of Active Learning, we now present the strategy adopted in the reference study by Borkowski et al. [2], *Large scale active-learning-guided exploration to maximize cell-free production*, which applies an Active Learning framework to guide the exploration of a large combinatorial space of cell-free reaction compositions and identify conditions that maximize protein production.

6.4.2 Active Learning in the Reference Study

The study by Borkowski et al., *Large scale active-learning-guided exploration to maximize cell-free production*[2] presents a relevant application of Active Learning to cell-free systems.

In this work, the authors aim to optimize protein production in a lysate-based cell-free expression system by efficiently exploring the large combinatorial space of reaction compositions. In particular, the experiments are performed using a reference lysate (Lysate_ORI) derived from *E. coli* strains, where the molecular components required for transcription and translation are assembled in vitro to reconstruct the cellular machinery responsible for the production of the reporter protein sfGFP.

Protein synthesis strongly depends on the concentration of multiple chemical components in the reaction buffer, and the resulting combinatorial space of possible compositions quickly becomes too large to explore exhaustively through experimental testing.

To address this challenge, Borkowski et al. propose an Active Learning framework that iteratively combines machine learning predictions with targeted experiments. The goal is to identify reaction compositions that maximize protein production while minimizing the number of experiments required to explore the design space.

In the following, an overview of the methodology adopted in the reference study is presented. For each of the main parts of the Active Learning procedure, the corresponding choices adopted in the present work will also be briefly outlined, in order to highlight the main methodological aspects of the proposed pipeline.

Design Space and Input Normalization In the study by Borkowski et al., the design space corresponds to a discrete domain whose elements are vectors containing the values of 11 biochemical components of the cell-free reaction mixture. The considered variables are: Mg-glutamate, K-glutamate, Amino Acids, tRNA, CoA, NAD, cAMP, Folinic Acid, Spermidine, 3-PGA, and NTPs.

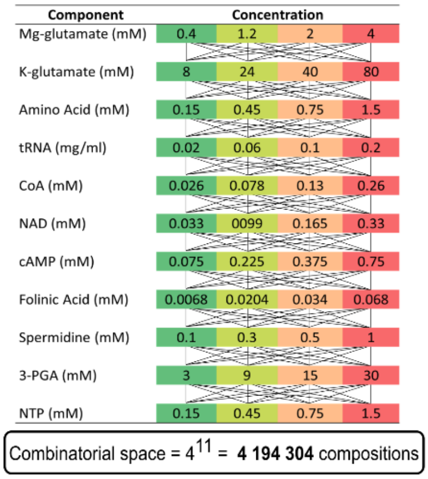
For each component, the concentration is not directly varied over an absolute range, but is instead expressed as a fraction of a predefined reference concentration corresponding to the maximum level allowed for that compound in the reaction mixture. Specifically, each component is evaluated at four discrete levels:

$$\{0.10,\ 0.30,\ 0.50,\ 1.00\}, \tag{6.20}$$

representing 10%, 30%, 50%, and 100% of the maximum concentration.

As a consequence, the resulting design space contains 4^{11} possible combinations of reaction compositions, corresponding to more than four million candidate experimental conditions. This extremely large combinatorial space makes a full experimental exploration impractical.

The specific concentration values associated with each component and their corresponding reference levels are reported in Fig. 6.1, reproduced from Borkowski et al. [2], where the normalized representation of the experimental design space is illustrated.



Component	Concentration			
Mg-glutamate (mM)	0.4	1.2	2	4
K-glutamate (mM)	8	24	40	80
Amino Acid (mM)	0.15	0.45	0.75	1.5
tRNA (mg/ml)	0.02	0.06	0.1	0.2
CoA (mM)	0.026	0.078	0.13	0.26
NAD (mM)	0.033	0099	0.165	0.33
cAMP (mM)	0.075	0.225	0.375	0.75
Folinic Acid (mM)	0.0068	0.0204	0.034	0.068
Spermidine (mM)	0.1	0.3	0.5	1
3-PGA (mM)	3	9	15	30
NTP (mM)	0.15	0.45	0.75	1.5

Combinatorial space = 4^{11} = **4 194 304** compositions

Figure 6.1: Chemicals in the cell-free system of the reference study and their corresponding concentration levels. The red value indicates the maximum concentration, while orange, light green, and dark green represent 50%, 30%, and 10% of the maximum concentration, respectively.

In the present study, the design space, described experimentally in Sec. 2.1, is approximately 1000 times smaller than the one considered in the reference work, with a total of 4704 possible combinations. Nevertheless, its size still makes an exhaustive experimental exploration impractical, and an Active Learning framework remains a fundamental tool to efficiently extract information from the domain.

In this case, the number of features is also smaller than in the reference study, with input vectors of dimension 5, corresponding to the concentrations of ATP, Mg^{2+}, ClpX, ClpP, and PEG8000. Unlike the reference work, no normalization of the input variables is required. This is due to the predictive model adopted in this study, which is based on Random Forests and is therefore inherently robust to feature scaling. The values assumed by the input variables thus correspond directly to experimentally meaningful absolute concentrations rather than normalized fractions of reference levels.

(Further details on the construction of the design space are provided in Sec. 2.1.)

Output definition and normalization In the reference study, the response variable used to train the predictive model is derived from the fluorescence associated with the expression of the reporter protein sfGFP. The fluorescence is measured using a plate reader and represents an endpoint measurement of protein production in each reaction composition.

However, the raw fluorescence signal contains contributions that are not related to protein production, such as background autofluorescence of the reaction mixture. For this reason, a background correction is first applied.

Let F_i denote the fluorescence measured for the i-th experimental composition and let F_{auto} denote the autofluorescence measured in a control reaction corresponding to the reference composition without DNA. Since protein synthesis in the cell-free system is initiated by transcription of the DNA template, removing the DNA prevents the production of the reporter protein and therefore provides an estimate of the background fluorescence of the reaction mixture. The corrected fluorescence is therefore defined as

$$F_i^{corr} = F_i - F_{\text{auto}}. \tag{6.21}$$

To enable comparison between measurements obtained across different plates and experimental runs, the corrected fluorescence is further normalized with respect to a reference reaction composition. Denoting by F_{ref} the fluorescence measured for the reference composition and applying the same background correction, the final response variable (referred to as *relative yield*) is defined as

$$yield_i = \frac{F_i - F_{\text{auto}}}{F_{\text{ref}} - F_{\text{auto}}}. \tag{6.22}$$

The resulting label therefore represents the relative protein production obtained for a given reaction composition compared to the reference mixture.
This normalization procedure produces a dimensionless response variable and reduces systematic variations across experimental plates, allowing the machine learning model to focus on relative differences in protein production across the explored compositions.

In the present work, the definition of the response variable differs substantially from that adopted in the reference study. While Borkowski et al. rely on a single endpoint fluorescence measurement to quantify protein production, the experimental observations considered here consist of time-series measurements of fluorescence describing the degradation dynamics of the reporter protein sfGFP. In particular, fluorescence measurements are collected over a time horizon of approximately 240 minutes, which corresponds to the time required for the degradation process to approach a steady-state regime under the considered experimental conditions. Rather than using the final fluorescence value directly as a label, the entire time series is used to infer the kinetic parameter governing the degradation dynamics.
In this experimental setting, the background signal corresponds to the fluorescence measured in a reference reaction in which the reporter protein is absent. Since the aim of the experiment is to study protein degradation rather than protein synthesis, removing the protein from the reaction mixture provides an estimate of the autofluorescence of the cell-free system. This background signal is therefore used to correct the fluorescence trajectories prior to the kinetic analysis.
More precisely, the degradation process is modeled through a parametric kinetic model fitted to the observed fluorescence trajectories. Parameter inference is performed within a Bayesian inference framework implemented through the JAGS software, as described in detail in Sec. 4. For each experimental design, this procedure yields posterior samples of the degradation rate parameter k, which are subsequently corrected for day-dependent batch effects estimated from the reference experiments.
The response variable used for training the predictive model is defined on the logarithmic scale. and it is derived through JAGS framework.
This choice allows the predictive model to be trained on a statistically inferred kinetic parameter summarizing the entire fluorescence trajectory rather than on a single endpoint measurement. As a consequence, the resulting labels incorporate information from the full time evolution of the degradation process, providing a more informative representation of the underlying biochemical dynamics.
(For further details see Chap. 4– 5)

Initial dataset and batch structure in the reference study In the study by Borkowski et al., the Active Learning workflow operates with a batch size of 102 experimental compositions per iteration: the new 102 distinct reaction compositions are tested and used to update the training set for the predictive model.

The initial training dataset is also composed of 102 compositions. More precisely, 22 compositions are constructed deterministically in order to probe extreme regions of the design space: the influence of each individual component is systematically explored when the remaining components are fixed at extreme levels. The remaining 80 compositions are sampled randomly from the design space.

In order to monitor the uncertainty of measurements, each reaction composition is experimentally evaluated in triplicate. As a consequence, each batch of 102 designs corresponds to $102 \times 3 = 306$ experimental wells used to obtain replicate fluorescence measurements.
In addition to the candidate compositions selected by the Active Learning procedure, each 384-well plate also contains a set of control reactions used to monitor the consistency of the experimental measurements across plates.
In particular, 13 control compositions, measured in triplicate, include the reference composition, its corresponding version without DNA, and additional compositions that remain constant across plates. Furthermore, some of the control wells are filled with compositions that achieved high yields in previous iterations of the Active Learning workflow.
In the present work, the structure of the experimental batches differs from that adopted in the reference study, both in terms of batch size and in the organization of control measurements.

In our study, the batch size is considerably smaller: at each iteration, 12 distinct experimental designs are evaluated. Although the experiments are also performed using a 384-well plate, it is not possible to fully exploit its capacity. As explained in Sec. 2.1, the components of the reaction mixture (PEG, premix, and autolysate) are dispensed manually, while the reporter protein sfGFP is added to the wells sequentially using the I-DOT dispensing system.
To avoid introducing variability in the degradation dynamics due to delays in plate preparation, it is important that the time interval between the preparation of the first and the last well remains limited. For this reason, only two rows of the 384-well plate are used in each experimental run, corresponding to a total of 48 wells.
Despite this reduced experimental throughput, the available resources were used as efficiently as possible, partially following the strategy adopted in the reference study. In particular, each design is measured in triplicate and a set of control compositions is included in every plate.
Two reference compositions are systematically included: one containing the protein to be degraded and one without the protein. Considering both references and three experimental replicates, these controls occupy $2 \times 2 \times 3 = 12$ wells.
The inclusion of control compositions serves two main purposes:

Batch effect estimation The reference trajectories are used to estimate day-dependent batch effects, allowing degradation rates measured on different experimental days to be corrected and made comparable.

Experimental quality control The reference measurements provide a consistency check for the experimental procedure. In particular, the fluorescence trajectories are inspected to verify that the degradation process behaves as expected and that the measurements remain consistent across replicates. Significant deviations in the reference curves could indicate potential experimental issues, such as incorrect dispensing of reaction components or anomalies in the fluorescence acquisition performed by the Tecan plate reader. During the experimental sessions, no systematic issues affecting the control measurements were observed. Only in a few isolated cases did one of the three replicates fail to exhibit the expected degradation behavior.

Background monitoring

The remaining 36 wells are therefore used to evaluate 12 new candidate designs, each measured in triplicate.
Regarding the initial batch, it also consists of 12 experimental designs (excluding the control compositions), each measured in triplicate. These initial designs are selected deterministically following a strategy similar to that adopted in the reference study. In particular, they correspond to extreme conditions of the design space, where individual components are set either to their minimum or maximum concentration levels while the remaining variables are fixed. Further details on the construction of the initial dataset are provided in Sec. 3.1.3.
Infine, viene ora presentato il fulcro del paper, l'implementazione dell'Active Learning strategy

Active Learning loop: surrogate model and acquisition function In the study by Borkowski et al., the predictive model used within the Active Learning loop is a fully connected feed-forward neural network implemented as a Multilayer Perceptron (MLP).
Once trained, the network receives as input the vector containing the concentrations of the components of the cell-free system and returns as output an estimate of the relative protein production yield.

To build an Active Learning framework capable of proposing informative candidate compositions for the next experimental iteration, the authors employ an ensemble of predictive models. In particular, the ensemble is composed of 25 independent neural networks. Each ensemble member corresponds to the best model selected among 10 randomly initialized MLP networks that are trained independently.

The use of an ensemble is essential for defining the acquisition function $a(x)$. In particular, the ensemble allows the estimation of both the expected performance of a candidate composition and the uncertainty associated with the prediction.
The expected response is computed as the mean of the ensemble predictions:

$$\mu(x) = \frac{1}{C} \sum_{c=1}^{C} \hat{f}_c(\mathbf{x}) \tag{6.23}$$

where C denotes the number of models in the ensemble and $\hat{f}_c(\mathbf{x})$ represents the prediction produced by the c-th neural network for the input feature vector $\mathbf{x}$.
The predictive uncertainty is instead estimated through the dispersion of the ensemble predictions,

$$\sigma(x), \tag{6.24}$$

computed as the standard deviation of the predictions produced by the ensemble members. It is important to note that this uncertainty estimate does not originate from a Bayesian probabilistic model, but rather from the empirical variability among the predictions of the ensemble models.

At each iteration of the Active Learning loop, the acquisition function is not evaluated over the entire combinatorial design space, which would be computationally prohibitive given its size. Instead, the authors adopt a **random candidate screening** strategy: a large subset of 100,000 compositions is randomly sampled from the discrete design space, excluding those that have already been experimentally tested.
The acquisition function is then evaluated only on this sampled subset of candidates. The experimental batch is obtained by ranking the sampled compositions according to their acquisition value and selecting the top candidates. This Monte Carlo screening procedure provides a

computationally efficient approximation of the acquisition maximization problem while remaining feasible for very large combinatorial spaces.

For each sampled candidate $\mathbf{x}$, the ensemble provides an estimate of the **expected performance** and the **model uncertainty**. The selection of the 102 new experimental compositions is therefore performed by ranking candidates according to the value of the acquisition function

$$a(\mathbf{x}) = \mu(x) + \beta\sigma(x), \tag{6.25}$$

which corresponds to the classical Upper Confidence Bound (UCB) acquisition rule. In the implementation used in the study, the exploration parameter is fixed to $\beta = 1.41 \approx \sqrt{2}$ and remains constant across all iterations, balancing the trade-off between exploitation and exploration.

Stopping criterion In the reference study, the stopping criterion is mainly based on two conditions:

- The value of the yield no longer improves significantly across iterations;
- The prediction accuracy remains stable across iterations.

In particular, in the work of Borkowski et al. [2], the evolution of the best yield achieved is monitored across iterations, together with an accuracy measure computed at each iteration on the dataset accumulated up to that point.

Predictive accuracy is quantified through a **5-fold cross-validation** procedure, using the coefficient of determination R^2 as the scoring metric. Given the dataset available at iteration t, the samples are randomly partitioned into 5 folds; for each fold, an ensemble of neural networks is trained on the training folds and evaluated on the held-out validation fold. In the implementation released by the authors, the surrogate model consists of an ensemble of MLP regressors: for each fold, 25 MLP models are trained, and each ensemble member is itself obtained by repeating the training multiple times in order to account for the dependence on random initialization, finally selecting the instance with the highest training R^2. The final prediction on the validation fold is computed as the mean of the predictions produced by the 25 models of the ensemble, while the dispersion among these predictions provides an empirical estimate of predictive uncertainty.

The coefficient of determination R^2 measures the fraction of variability of the data in the validation fold that is explained by the model trained on the remaining data.

Denoting by y_i the observed values, by $\hat{y}_i$ the predicted values, and by $\bar{y}$ the empirical mean of the observations in the validation set, it is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{6.26}$$

This metric compares the residual sum of squares of the model predictions with the total variability of the observed data. Values close to 1 indicate that the model accurately explains the variability of the response variable, whereas values close to 0 indicate predictive performance comparable to that of a trivial model based solely on the sample mean.

Let $R^2_{t,1}, \dots, R^2_{t,5}$ denote the R^2 values obtained on the five validation folds at iteration t. The global cross-validated accuracy reported at iteration t is then computed as

$$R^2_t = \frac{1}{5}\sum_{k=1}^{5} R^2_{t,k}, \tag{6.27}$$

while its variability across folds is summarized through the empirical standard deviation

$$\sigma_{R^2,t} = \sqrt{\frac{1}{5}\sum_{k=1}^{5}\left(R^2_{t,k} - R^2_t\right)^2}. \tag{6.28}$$

In addition, the input features are normalized by rescaling each component with respect to its maximum value in the current dataset, thus ensuring that all variables lie on a comparable scale before training the neural network surrogate.
In the original study, both the best observed yield and the cross-validated value of R^2 reach a plateau after a small number of iterations (approximately 7), while the workflow is continued up to 10 iterations to confirm the absence of further improvements. At the end of the acquisition process, the authors obtain a final informative dataset composed of 1017 experimentally tested compositions.
In our work, the stopping criterion used does not differ substantially from the one described above.

We now proceed with a detailed presentation of the Active Learning pipeline adopted in this study.

6.5 Implementation of the Active Learning framework

The previous sections described the Active Learning strategy adopted in the reference study by Borkowski et al. We now turn to the Active Learning framework developed in the present work. Although the general structure of the iterative loop follows the same principles, several methodological choices differ substantially due to the characteristics of the available data and the experimental setup. In particular, differences arise mainly in the choice of the predictive model and in candidates generation. They will be elaborated in the following sections describing the Active Learning pipeline implemented in this work.

6.5.1 Predictive Model

The predictive model adopted in this work is an ensemble composed of $C = 10$ independent Random Forests, each consisting of $B = 200$ decision trees. Each tree is trained in a unique manner thanks to the use of **bootstrap sampling** and **feature sampling**.
The trees are grown without imposing a predefined maximum depth and follow the standard feature subsampling strategy used in Random Forests, where at each split only $\sqrt{p}$ variables are considered, with p denoting the number of input features.
Similarly to the approach adopted in the reference study, the ensemble is used to compute both the mean prediction and the predictive dispersion across models. For a given input vector $\mathbf{x}$, the ensemble mean and standard deviation are defined as

$$\mu(\mathbf{x}) = \frac{1}{C}\sum_{c=1}^{C}\hat{f}_c(\mathbf{x}), \qquad \sigma(\mathbf{x}) = \sqrt{\frac{1}{C-1}\sum_{c=1}^{C}\left(\hat{f}_c(\mathbf{x}) - \mu(\mathbf{x})\right)^2}, \tag{6.29}$$

where $\hat{f}_c(\mathbf{x})$ denotes the prediction produced for $\mathbf{x}$ by the c-th Random Forest. Each forest prediction $\hat{f}_c(\mathbf{x})$ corresponds then to the average of the predictions generated by its B decision trees.
The quantity $\sigma(\mathbf{x})$, analogously to the ensemble of neural networks used in the reference study, provides an empirical estimate of the model uncertainty in regions of the design space that have not yet been explored.

The choice of the predictive model used as surrogate in the Active Learning loop was guided by the specific characteristics of the problem and by the constraints imposed by the experimental setting. In particular, the number of available observations is limited and the response variable does not correspond to a direct experimental measurement, but rather to a kinetic parameter inferred from fluorescence time-series through a Bayesian modeling framework. These aspects require a predictive model that remains robust even in the presence of relatively small datasets and intrinsic variability in the observations.

In the reference study by Borkowski et al., the surrogate model used within the Active Learning loop consists of an ensemble of Multilayer Perceptron (MLP) neural networks. An MLP is a feed-forward artificial neural network composed of multiple fully connected layers of neurons. Each neuron computes a linear combination of the activations from the previous layer and applies a nonlinear activation function. Through the composition of several nonlinear layers, the network is able to approximate highly complex functions describing the relationship between input variables and the response variable. During training, the network weights are optimized through gradient-based algorithms such as Adam, by minimizing a loss function defined on the training dataset.

MLP-based models are particularly effective when relatively large datasets are available and when complex nonlinear relationships between variables must be captured. This is the case in the study by Borkowski et al., where the final dataset contains more than one thousand experimental observations and the number of input variables is relatively large.

In the present work, however, the use of neural networks would be less appropriate. The dataset available for training is significantly smaller and grows progressively during the Active Learning iterations, remaining, anyway, small. Moreover, the response variable corresponds to a kinetic parameter inferred from fluorescence trajectories rather than to a direct experimental measurement. Under these conditions, highly flexible models such as neural networks may become more sensitive to noise and more prone to overfitting.

Tree-based ensemble methods, such as Random Forests, provide a more robust alternative in small-data regimes. They naturally capture nonlinear relationships and interactions among variables without requiring a specific parametric formulation, and they perform particularly well when the input space is characterized by discrete experimental factors, as in the design space considered in this study.

For these reasons, in the Active Learning pipeline developed in this work the predictive model is implemented as a Random Forest ensemble rather than a neural network ensemble. This choice reflects the different scale of the available dataset, the nature of the response variable, and the need for a model that remains robust in small-data regimes.

6.5.2 Acquisition strategy

In the pipeline adopted in this work, the **Upper Confidence Bound** (UCB) is used as acquisition function, similarly to the approach proposed in the reference study. The UCB acquisition rule is defined by Eq. 6.25.

The parameter β, which in the reference study is kept fixed across all Active Learning iterations, is instead allowed to vary in our implementation. Since this parameter controls the balance between *exploration* and *exploitation*, represented respectively by $\sigma(\mathbf{x})$ and $\mu(\mathbf{x})$, it constitutes a powerful mechanism for guiding the selection of candidate samples.

More specifically, the role of β is to determine the relative importance of the predictive mean $\mu(\mathbf{x})$ and the predictive uncertainty $\sigma(\mathbf{x})$ when evaluating candidate designs. Very small values of β make the dispersion of the ensemble predictions almost negligible in the acquisition score. In this case, the new batch would be generated almost exclusively by selecting candidates whose labels, as predicted by the ensemble models, have the highest expected value.

On the other hand, values of β that bring $\sigma(\mathbf{x})$ to a scale comparable with $\mu(\mathbf{x})$ tend to favor candidates with intermediate values of the two statistics. This situation does not necessarily ensure strong exploitation of promising predicted log-rates, nor effective exploration of uncertain regions of the design space. For this reason, such intermediate values of β are generally less informative in guiding the search. Finally, very large values of β would prioritize candidates for which the models in the ensemble exhibit the highest predictive disagreement, therefore encouraging exploration of uncertain regions.

Returning to the UCB acquisition function, the candidate points are selected by solving the following optimization problem:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} a(\mathbf{x}), \tag{6.30}$$

where $\mathcal{X}$ denotes the design space excluding the points that have already been experimentally evaluated and labeled.

Since the resulting optimization problem is discrete and unconstrained, and the number of candidate designs in $\mathcal{X}$ remains computationally manageable, the problem can be solved through exhaustive evaluation. In practice, the acquisition function is computed explicitly for every candidate point $\mathbf{x} \in \mathcal{X}$, and the K points with the highest acquisition values are selected to form the next experimental batch.

The definition of the acquisition strategy described above implicitly determines the optimization objective of the Active Learning loop. In the following subsection, we discuss the choice of the target quantity used in this optimization process and its implications for the experimental design problem considered in this work.

Optimization objective and limitations The Active Learning procedure aims to identify experimental conditions that maximize the degradation speed of the reporter protein. In this study, the optimization objective is defined as the logarithm of the degradation rate k, inferred from the fluorescence trajectories through the Bayesian model described in Sec. 4. The surrogate model is therefore trained to predict this quantity, and the acquisition function is constructed accordingly.

Under the kinetic model introduced previously, the normalized fluorescence trajectories are described by

$$y^{norm}_{j,d,r}(t) = B_j + (1 - B_j)\exp\big(-k_{j,d,r}(t - t_0)\big), \tag{6.31}$$

where k controls the speed of degradation and B represents the asymptotic fluorescence level. Since the trajectories are normalized so that $y^{norm}(t_0) = 1$, the quantity $1 - B$ corresponds to the maximum fraction of fluorescence that can be removed during the experiment.

From a biological perspective, efficient degradation would ideally combine a large degradation rate k with a small plateau level B. In principle, the optimization problem could therefore be formulated as a multi-objective task involving both parameters. In the present work, however, the objective is defined solely in terms of the degradation rate k.

This simplification offers several practical advantages. The degradation rate provides a robust and interpretable summary of the dynamics and is generally more reliably estimated from fluorescence trajectories than the plateau level B, which can be more sensitive to noise and baseline variations. Moreover, focusing on a single objective contributes to maintaining a stable Active Learning loop in the small-data regime typical of biological experiments.

Nevertheless, the Bayesian inference procedure provides posterior information for both parameters k and B. Consequently, the dataset generated through the Active Learning iterations contains information not only about degradation speed but also about the magnitude of the fluorescence plateau. For this reason, the candidate selection strategy does not rely solely on the acquisition function. Instead, an additional exploration mechanism is introduced to ensure that the selected designs cover different regions of the biochemical design space.

In practice, the exploration behavior is controlled through the exploration parameter β in the UCB acquisition rule and through a stratified sampling strategy applied to the candidate space. This combination balances exploitation of promising configurations with a broader coverage of the design space, allowing the resulting dataset to support both the optimization of the degradation rate and potential future analyses of other aspects of the degradation dynamics, including the plateau parameter B.

The details of the stratified exploration strategy are described in the following section.

Stratified exploration The sole adoption of the UCB acquisition function, although effective in balancing exploitation and exploration with respect to the considered target, naturally tends to concentrate the selection of new experiments in regions of the design space that appear particularly promising in terms of predicted log-rate or that exhibit high predictive uncertainty. In the present work, however, an additional exploration mechanism is introduced in order to ensure a more balanced coverage of the experimental space and to construct a dataset that is informative not only with respect to the degradation rate parameter k, but also potentially useful for future analyses of other aspects of the degradation dynamics, such as the plateau parameter B.

To this end, the design space is partitioned into 6 regions defined on the basis of previous studies conducted in the same laboratory [1]. These studies have shown that the main factors influencing the degradation process are the concentration of ClpX and the availability of ATP. Consequently, the design space is partitioned according to the regimes of these two variables as follows:

×	ClpX_{off}	ClpX_{on}
ATP_0	R_1	R_4
ATP_{low}	R_2	R_5
ATP_{high}	R_3	R_6

Table 6.1: Partition of the design space into six regions obtained from the Cartesian product of ATP regimes and ClpX regimes. This partition reflects distinct biochemical conditions of the degradation system and is used to guide the stratified exploration strategy.

with

$$\text{ATP}_{\text{low}} = \{1,2,4\}, \qquad \text{ATP}_{\text{high}} = \{5.7,8\}, \qquad \text{ClpX}_{\text{off}} = \{0\}$$
$$\text{ClpX}_{\text{on}} = \{50,100,150,200,300,400\}.$$

This partition serves two purposes. First, it allows the implementation of a stratified exploration mechanism that promotes the selection of candidate designs from different biochemical regimes of the system. Second, it provides a convenient way to monitor how the exploration of the design space evolves throughout the Active Learning iterations. In particular, by tracking the number of sampled designs within each region, it is possible to detect potential imbalances in the exploration

process and to prevent situations in which some portions of the design space remain systematically underexplored or even completely unvisited.
The resulting regional cardinalities are therefore given by

$$|R_1| = 1 \cdot 1 \cdot 4 \cdot 7 \cdot 4 = 112,$$

$$|R_2| = 3 \cdot 1 \cdot 4 \cdot 7 \cdot 4 = 336,$$
$$|R_3| = 2 \cdot 1 \cdot 4 \cdot 7 \cdot 4 = 224,$$
$$|R_4| = 1 \cdot 6 \cdot 4 \cdot 7 \cdot 4 = 672,$$
$$|R_5| = 3 \cdot 6 \cdot 4 \cdot 7 \cdot 4 = 2016,$$
$$|R_6| = 2 \cdot 6 \cdot 4 \cdot 7 \cdot 4 = 1344.$$

In particular,

$$\sum_{r=1}^{6} |R_r| = 112 + 336 + 224 + 672 + 2016 + 1344 = 4704,$$

which is consistent with the total cardinality of the design space.
After the initial dataset of $12 + 2$ experiments, suppose that the number of already queried points in the six regions is

$$n_1 = 4, \qquad n_2 = 0, \qquad n_3 = 2, \qquad n_4 = 2, \qquad n_5 = 2, \qquad n_6 = 4.$$

If the same probability were assigned to each point in the design space, then the probability that a sampled point belongs to region R_r would simply be proportional to the cardinality of that region, namely

$$\mathbb{P}(R_r) = \frac{|R_r|}{|\mathcal{X}|}.$$

In the present case, this would yield

$$\mathbb{P}(R_1) = \frac{112}{4704}, \qquad \mathbb{P}(R_2) = \frac{336}{4704}, \qquad \mathbb{P}(R_3) = \frac{224}{4704},$$

$$\mathbb{P}(R_4) = \frac{672}{4704}, \qquad \mathbb{P}(R_5) = \frac{2016}{4704}, \qquad \mathbb{P}(R_6) = \frac{1344}{4704}.$$

As a consequence, the largest regions, in particular R_5 and R_6, would be naturally favored, whereas regions with smaller cardinality would risk being explored only marginally, or even remaining unobserved for many iterations. In particular, a region such as R_2, which contains no labeled point in the initial dataset, could continue to remain unexplored if the selection depended only on the cardinality of the domain or on the global ranking of candidate points.

Regional score. To mitigate this imbalance, the stratified component does not sample directly and uniformly from all remaining candidate points. Instead, a regional score is first introduced. Let

$$N_r = |R_r|$$

denote the original cardinality of region R_r, and let

$$n_r^{(t)}$$

be the number of points in region R_r that have already been queried up to iteration t. The regional score is then defined as

$$s_r^{(t)} = \left(1 - \frac{n_r^{(t)}}{N_r}\right) \sqrt{N_r}.$$

This score accounts for two aspects. First, it incorporates the fraction of the region that has already been explored through the term

$$1 - \frac{n_r^{(t)}}{N_r},$$

which penalizes regions that have already been heavily sampled. Second, it retains information on the size of the region through the factor

$$\sqrt{N_r},$$

which does not remove the influence of cardinality altogether, but attenuates it compared with a linear dependence.

In this way, the regional sampling probability is no longer directly proportional to N_r, but instead depends on an intermediate measure that reduces the advantage of larger regions while encouraging the exploration of regions that are still poorly observed.

From the score to the regional probability distribution. Let

$$\mathcal{R}_t \subseteq \{1, \dots, 6\}$$

denote the set of regions that, at iteration t, still contain at least one available candidate. The probability of selecting region R_r in the STRAT phase is defined as

$$p_r^{(t)} = \frac{s_r^{(t)}}{\sum_{h \in \mathcal{R}_t} s_h^{(t)}}, \qquad r \in \mathcal{R}_t.$$

For inactive regions, one naturally sets

$$p_r^{(t)} = 0.$$

Two-stage sampling mechanism. The selection of a point in the STRAT component is therefore performed in two stages.

First stage: selection of the region.

A region is sampled according to a multinomial distribution with one trial, that is, equivalently, a categorical distribution over six levels:

$$\big(Z_1^{(t)}, \dots, Z_6^{(t)}\big) \sim \text{Multinomial}\Big(1; p_1^{(t)}, \dots, p_6^{(t)}\Big).$$

Equivalently, if $R^{(t)}$ denotes the selected region, then

$$\mathbb{P}\big(R^{(t)} = R_r\big) = p_r^{(t)}.$$

Second stage: selection of the candidate within the chosen region.

Once the region $R^{(t)} = R_r$ has been selected, consider the set of candidates that are still available within that region, denoted by

$$\mathcal{X}_r^{(t)} \subseteq R_r.$$

The candidate is then sampled uniformly within this set:

$$X^{(t)} \mid \big(R^{(t)} = R_r\big) \sim \text{Unif}\big(\mathcal{X}_r^{(t)}\big).$$

It follows that, conditionally on the chosen region, all remaining points in that region have the same probability of being selected.

Candidate selection strategy. At each iteration of the Active Learning loop, a batch of 12 new experimental designs is selected from the candidate pool. The selection combines two complementary mechanisms: a model-driven acquisition rule based on the Upper Confidence Bound (UCB) and a stratified exploration procedure aimed at ensuring coverage of different regions of the design space.

Formally, the batch is partitioned as

$$B_t = B_t^{\mathrm{UCB}} \cup B_t^{\mathrm{STRAT}},$$

where B_t^{UCB} contains candidates selected according to the acquisition function and B_t^{STRAT} contains candidates drawn through stratified exploration. The relative size of these two subsets varies across iterations in order to progressively shift the exploration–exploitation balance.

The number of stratified samples decreases during the Active Learning process according to

$$N_{\mathrm{STRAT}}(t) = \begin{cases} 6, & t = 0, \\ 5, & t = 1, \\ 4, & t \in \{2{,}3\}, \\ 3, & t \in \{4{,}5\}, \\ 2, & t \geq 6. \end{cases}$$

This exploration component is particularly important in the early iterations, when the surrogate model is trained on a limited number of observations and its uncertainty estimates are still unreliable. Stratified sampling therefore ensures that candidate designs are drawn from different biochemical regimes of the design space, preventing the model from concentrating too early on a restricted region.

The exploration parameter β of the UCB acquisition function is also varied across iterations. The procedure starts with $\beta = 2$, followed by two iterations with $\beta = 3$. Once the model begins to capture more reliable patterns, the parameter is temporarily increased to $\beta = 10$ for two iterations in order to encourage exploration of regions with high predictive uncertainty. In the final stage the parameter is reduced again to $\beta = 3$, promoting exploitation around promising regions of the design space.

Starting from an initial dataset of 12 designs, the Active Learning loop proceeds until 72 experimental designs have been evaluated (excluding control measurements), producing a dataset that balances optimization of the degradation rate with a sufficiently broad coverage of the design space.

6.5.3 Stopping

As in the reference study, the strategy adopted in this work consists of monitoring the evolution of both the best design identified and the predictive accuracy of the surrogate model throughout the iterations of the Active Learning algorithm. The iterative process can therefore be stopped when additional acquisitions no longer produce significant improvements either in the quality of the identified designs or in the predictive performance of the surrogate model.

In our case, an additional practical constraint was introduced on the maximum number of iterations. Due to the high experimental costs and the time required to perform laboratory experiments, the total number of designs to be tested had to remain below 100. This constraint contributed to determining the maximum number of iterations of the algorithm.

To evaluate the evolution of the predictive capability of the model, a **cross-validation** procedure was employed. Cross-validation is a widely used statistical technique for estimating how well the performance of a model can be generalized to independent data that were not used during training.

The basic idea of cross-validation consists in partitioning the available dataset into distinct subsets, using part of the data to train the model and the remaining part to evaluate its predictive performance. By repeating this process multiple times and aggregating the performance obtained across the different data partitions, it is possible to obtain a more robust estimate of the model's generalization ability.

One of the most commonly used variants of this technique is **K-fold cross-validation**. In this scheme, the dataset is divided into K disjoint subsets, called *folds*. At each step of the procedure, one fold is used as the validation set, while the remaining $K-1$ folds are used to train the model. The process is repeated K times, so that each fold is used exactly once as the validation set.

In the original work by Borkowski et al., a 5-fold cross-validation procedure is employed. In the present work, however, due to the relatively small size of the dataset available during the early iterations of the Active Learning process, a different configuration based on a **repeated 3-fold cross-validation** was adopted, in order to obtain more stable performance statistics.

More precisely, at each iteration t and for each repetition $r = 1,2,3$, the dataset $\mathcal{D}_t$, containing n_t observations, is randomly partitioned into 3 folds. The model is then trained three times, each time leaving one fold out for validation and using the remaining two folds for training.

This procedure therefore produces a total of 9 independent evaluations of the model performance at each Active Learning iteration.

Model performance is quantified using the R^2 score, also known as the coefficient of determination, whose definition is provided below.

Definition 6.5.1 *The coefficient of determination, commonly denoted by R^2, is a statistical measure used to quantify the proportion of variability in a response variable that is explained by a predictive model.*

Let $y_1, \dots, y_n$ denote the observed values of the response variable and let $\hat{y}_1, \dots, \hat{y}_n$ denote the corresponding model predictions. Let $\bar{y}$ be the empirical mean of the observed responses,

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i. \tag{6.32}$$

The coefficient of determination is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}. \tag{6.33}$$

The numerator represents the residual sum of squares *(RSS), which measures the discrepancy between the observed values and the model predictions. The denominator corresponds to the variance of the response variable, which measures the overall variability of the observed data around their mean.*

Consequently, R^2 expresses the fraction of the total variance of the response variable that is explained by the model. Values of R^2 close to 1 indicate that the model explains most of the variability in the data, whereas values close to 0 indicate that the model performs similarly to a trivial predictor based on the sample mean. Negative values may occur when the model predictions are worse than the baseline prediction given by the mean of the observations.

In our case, during the Active Learning iterations, several performance indicators were monitored in order to assess both the predictive accuracy of the surrogate model and the stability of the

cross-validation procedure. These metrics were computed using the repeated cross-validation scheme described above.
The following metrics are computed.

Coefficient of determination on validation folds
For each repetition r and fold k, the coefficient of determination is computed as

$$R^2_{r,k} = 1 - \frac{\sum_{i \in F_{r,k}} (y_i - \hat{y}_i^{(r,k)})^2}{\sum_{i \in F_{r,k}} (y_i - \bar{y}_{r,k})^2}, \tag{6.34}$$

where

- y_i denotes the observed response value,
- $\hat{y}_i^{(r,k)}$ is the prediction produced by the model trained without fold k,
- $F_{r,k}$ is the validation fold,
- $\bar{y}_{r,k}$ is the mean of the observed responses within that fold.

The mean coefficient of determination across all folds and repetitions is

$$R^2_{\text{fold,mean}} = \frac{1}{9} \sum_{r=1}^{3} \sum_{k=1}^{3} R^2_{r,k}. \tag{6.35}$$

The variability of these values is summarized through the empirical standard deviation

$$R^2_{\text{fold,std}} = \sqrt{\frac{1}{9} \sum_{r=1}^{3} \sum_{k=1}^{3} \left(R^2_{r,k} - R^2_{\text{fold,mean}} \right)^2}. \tag{6.36}$$

The coefficient of determination measures the fraction of the variance of the response variable that is explained by the model predictions. Values close to 1 indicate high predictive accuracy, values close to 0 indicate performance comparable to predicting the mean response, and negative values indicate that the model performs worse than such a baseline.

Out-of-fold predictive accuracy
In addition to fold-wise scores, predictions are aggregated in an *out-of-fold* (OOF) manner. In this setting, each observation is predicted by a model that was trained without using that observation. For repetition r, let $\hat{y}_i^{\text{OOF},r}$ denote the out-of-fold prediction for observation i. The corresponding coefficient of determination is

$$R^2_{\text{OOF},r} = 1 - \frac{\sum_{i=1}^{n_t} (y_i - \hat{y}_i^{\text{OOF},r})^2}{\sum_{i=1}^{n_t} (y_i - \bar{y}_t)^2}, \tag{6.37}$$

where $\bar{y}_t$ is the empirical mean of the response values in the dataset at iteration t.
The mean and standard deviation across repetitions are

$$R^2_{\text{oof,mean}} = \frac{1}{3} \sum_{r=1}^{3} R^2_{\text{OOF},r}, \tag{6.38}$$

$$R^2_{\text{oof,std}} = \sqrt{\frac{1}{3} \sum_{r=1}^{3} \left(R^2_{\text{OOF},r} - R^2_{\text{oof,mean}} \right)^2}. \tag{6.39}$$

The OOF coefficient of determination provides a global estimate of the predictive accuracy of the model on unseen data across the entire dataset.

Root Mean Squared Error
The predictive error is also quantified through the root mean squared error (RMSE). For each fold it is computed as

$$RMSE_{r,k} = \sqrt{\frac{1}{|F_{r,k}|} \sum_{i \in F_{r,k}} (\hat{y}_i^{(r,k)} - y_i)^2}. \tag{6.40}$$

The mean and standard deviation across folds and repetitions are

$$RMSE_{\text{fold,mean}} = \frac{1}{9} \sum_{r=1}^{3} \sum_{k=1}^{3} RMSE_{r,k}, \tag{6.41}$$

$$RMSE_{\text{fold,std}} = \sqrt{\frac{1}{9} \sum_{r=1}^{3} \sum_{k=1}^{3} \left(RMSE_{r,k} - RMSE_{\text{fold,mean}}\right)^2}. \tag{6.42}$$

Similarly, using the out-of-fold predictions,

$$RMSE_{\text{OOF},r} = \sqrt{\frac{1}{n_t} \sum_{i=1}^{n_t} (\hat{y}_i^{\text{OOF},r} - y_i)^2}, \tag{6.43}$$

with mean and standard deviation

$$RMSE_{\text{oof,mean}} = \frac{1}{3} \sum_{r=1}^{3} RMSE_{\text{OOF},r}, \tag{6.44}$$

$$RMSE_{\text{oof,std}} = \sqrt{\frac{1}{3} \sum_{r=1}^{3} \left(RMSE_{\text{OOF},r} - RMSE_{\text{oof,mean}}\right)^2}. \tag{6.45}$$

RMSE measures the average magnitude of the prediction error and is expressed in the same units as the response variable.

Normalized RMSE
In order to make the error magnitude comparable across iterations, RMSE is also normalized by the empirical range of the response variable

$$\text{range}(y)_t = \max_i y_i - \min_i y_i. \tag{6.46}$$

The normalized errors are defined as

$$NRMSE_{\text{range,fold}} = \frac{RMSE_{\text{fold,mean}}}{\text{range}(y)_t}, \tag{6.47}$$

$$NRMSE_{\text{range,oof}} = \frac{RMSE_{\text{oof,mean}}}{\text{range}(y)_t}. \tag{6.48}$$

These quantities express the prediction error as a fraction of the variability of the response variable, thus providing a scale-independent indicator of predictive performance.

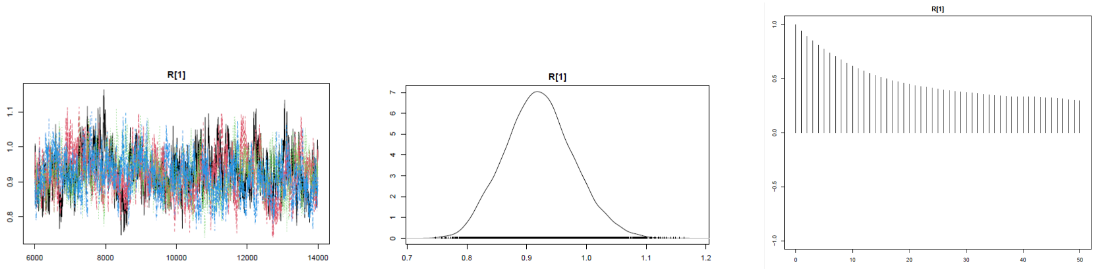
Chapter 7

Results and discussion

7.1 Bayesian Inference results

Regarding the Bayesian inference procedure, the results obtained with the implementation in R using JAGS can be considered overall satisfactory. Posterior samples were generated through Markov Chain Monte Carlo (MCMC) simulation. In order to assess the quality of the sampling and the degree of identifiability of the model parameters, standard MCMC diagnostics were analyzed, including traceplots of the chains, posterior distributions, and autocorrelation functions. Since the behaviour of the sampler varies across different experimental conditions, the following analysis has a general scope but focuses on a few representative cases. For the first-stage model, the batch parameter R_d corresponding to day 1 is considered (Figure 7.1), while for the second-stage model the diagnostics of the parameter μ_j are examined for three representative conditions: $Design23$, $Design38$, and the reference $RefB$ (Figure 7.2).

For the parameter R_1 (Figure 7.1), the chains exhibit a relatively stable and well-mixed behaviour, oscillating around a mean value approximately between 0.9 and 1.0. The posterior distribution appears unimodal and relatively concentrated, suggesting a reasonably well-identified estimate of the batch factor for the first experimental day. However, the autocorrelation function decreases slowly with the lag and remains positive even for relatively large lags. This behaviour indicates that consecutive samples are not independent and that the chain explores the parameter space relatively slowly, a phenomenon that is common in hierarchical models with exponential structure such as the one considered here.



(a) Traceplot **(b)** Density **(c)** Autocorrelation

Figure 7.1: MCMC diagnostics for the parameter $R[1]$, representing the multiplicative batch effect for day 1 relative to the baseline day (day 3). The columns show the traceplot of the chains, the posterior density estimate, and the autocorrelation function. Posterior samples were obtained using four parallel MCMC chains in JAGS with 1000 adaptation iterations, 5000 burn-in iterations and 8000 sampling iterations (thinning = 1).

The diagnostics associated with the parameter μ_j show more marked differences among the designs considered (Figure 7.2). In the case of $Design23$, the chains appear reasonably stationary and oscillate around a stable level. The posterior distribution is substantially unimodal and relatively concentrated, suggesting good identifiability of the parameter. In this case as well, the autocorrelation decreases slowly but in a regular manner, indicating mixing that is not perfect but still acceptable.

A markedly different situation is observed for $Design38$. In this case, the traceplot reveals strong instability of the chains and problematic mixing. The posterior distribution of μ_j exhibits a clearly multimodal structure, indicating that the model struggles to determine a single plausible value for the parameter. When posterior distributions are multimodal, the chains may move between different regions of the parameter space, increasing the variability of the estimates and reducing their reliability. For this design, the inference on μ_j must therefore be interpreted with particular caution, since the parameter appears to be poorly identified by the model.

The behaviour of the parameter μ_j for $RefB$ lies between the two previous cases. The chains do not show perfect convergence but remain concentrated within the same region of the parameter space, with no evidence of exploration of distinct areas. The posterior distribution also appears essentially unimodal, suggesting that the estimate of the parameter for the reference condition is overall stable, despite the presence of non-optimal mixing.

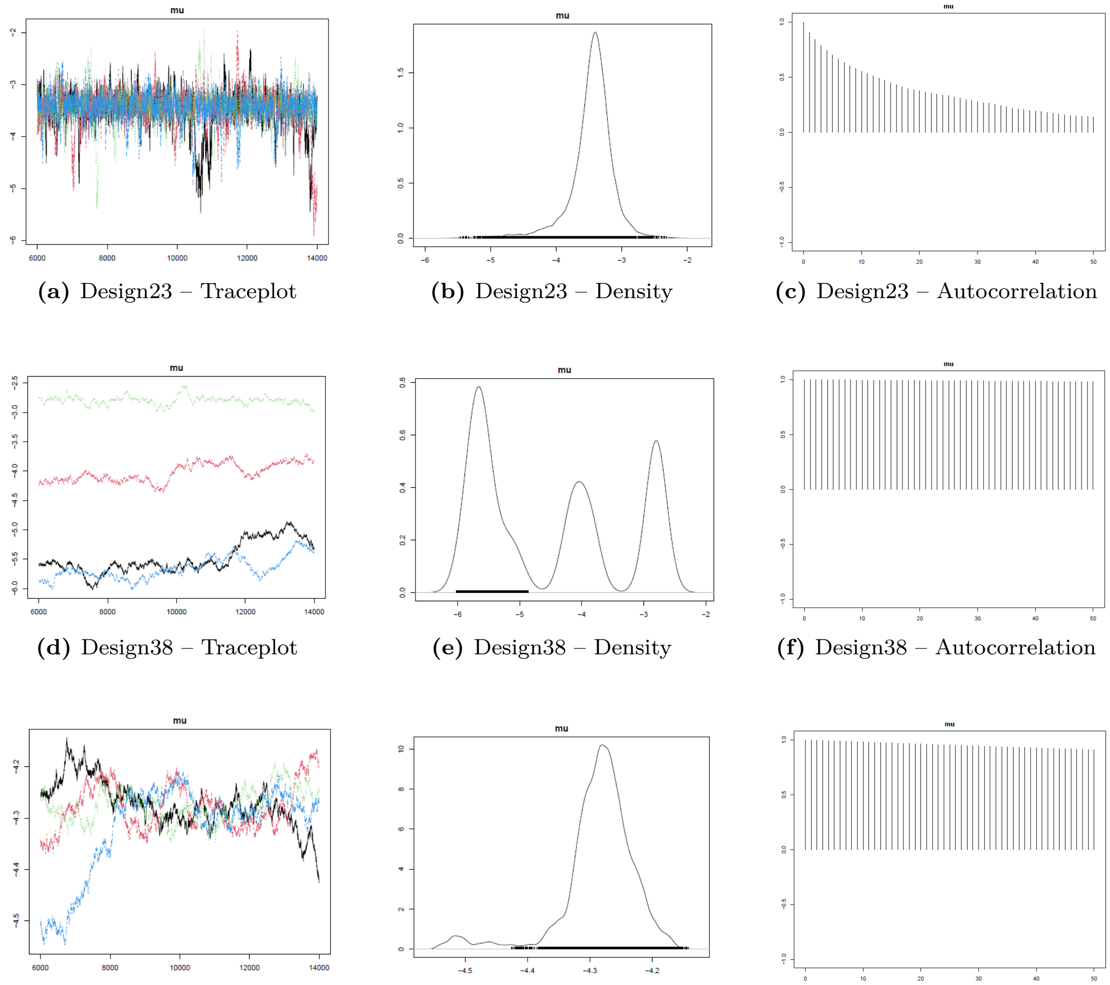
More generally, a common feature emerging from the diagnostic analysis is the presence of non-negligible autocorrelation in many of the inferred parameters. This behaviour is observed both in the joint model and in the design-specific models, and is consistent with the hierarchical structure of the problem and with the dependence between the parameters governing the exponential dynamics. Consequently, even in cases where the chains appear stationary and the posterior distributions are unimodal, the exploration of the posterior distribution remains relatively slow.

Regarding the first stage, the estimates of the parameters $R[d]$ appear overall more stable, as illustrated in Figure 7.1 and summarized in Table **??**. This is plausibly due both to the more regularized structure of the joint model, which is based exclusively on the fluorescence curves of the reference designs, and to the fact that the baseline parameter b is not sampled here but fixed from the tail of the trajectories. Moreover, the estimated values of $R[d]$ generally remain relatively close to 1: the median values reported in Table **??** range approximately between 0.80 and 1.18. This suggests the presence of a non-zero but moderate batch effect, interpretable as a multiplicative correction that is present but not dominant with respect to the overall variability observed in the data.

Greater inferential difficulties instead emerge in the second stage, where the parameter b is estimated together with μ_j and the curve-specific parameters. In particular, the designs exhibiting the poorest diagnostics, as shown in Figure 7.2, typically correspond to combinations characterized by high degradation activity, especially for large values of ATP and ClpX. Under these conditions the observed dynamics are faster and the fluorescence curve tends to reach lower final values, making smaller estimates of b plausible and, more generally, increasing the overall amplitude of the trajectory.

In these situations the final portion of the curve, from which the identification of the plateau largely depends, also becomes more sensitive to experimental noise. Since the final fluorescence levels may approach the residual background and the experimental variability of the tail, the model may struggle to distinguish between variations in the plateau parameter b and variations in the decay rate k. This dependence between the parameters amplifies the uncertainty in the estimate of k and, consequently, also in μ_j.

Overall, the identifiability of the parameters appears heterogeneous across the different designs. Some conditions, such as $Design23$, produce posterior distributions that are reasonably stable and interpretable, whereas others, such as $Design38$, lead to significantly more fragile inference. Nevertheless, considering the nature of the model adopted — based on a two-stage procedure, several approximations, and intrinsically noisy experimental data — the overall quality of the inferred quantities can still be regarded as adequate for the purposes of the analysis.



(a) Design23 – Traceplot **(b)** Design23 – Density **(c)** Design23 – Autocorrelation

(d) Design38 – Traceplot **(e)** Design38 – Density **(f)** Design38 – Autocorrelation

(g) RefB – Traceplot **(h)** RefB – Density **(i)** RefB – Autocorrelation

Figure 7.2: MCMC diagnostics for the parameter μ for Design 23, Design 38 and RefB. Columns show respectively the traceplot of the chains, the posterior density estimate, and the autocorrelation function. Posterior samples were obtained using JAGS with four parallel chains, 1000 adaptation iterations, 5000 burn-in iterations and 8000 sampling iterations (no thinning). The traceplots display the sampled values of μ across iterations, while the autocorrelation plots report the correlation between samples at increasing lags (up to lag 50).

7.2 Active Learning and general results

In this section, the experimental and computational results obtained in this work are presented. In particular, the performance of the pipeline developed in this thesis is compared with that reported in the reference study *"Large-scale active-learning-guided exploration to maximize cell-free production"* [2].

As regards the Active Learning strategy, the overall outcome can be considered satisfactory. After only 5 iterations, corresponding to the exploration of just 74 out of the 4704 points constituting the design space, it was possible to identify a combination of initial concentrations of the cell-free system components associated with protein degradation characterized by a log-rate μ of approximately -2.39, corresponding to a degradation rate k of about $9.10 \times 10^{-2}\,\mathrm{min}^{-1}$.

This means that, under these biochemical conditions, the fluorescence component associated with GFP-ssrA decreases at a characteristic rate of approximately 9% per minute relative to the remaining amount. Since fluorescence intensity is approximately proportional to the amount of fluorescent protein present in the system, the estimated parameter provides a quantitative measure of the effective efficiency of the degradation process mediated by the ClpXP complex in the *cell-free* system considered here.

This value corresponds to a relatively fast degradation regime for the ClpXP system and can therefore be regarded as a satisfactory result, especially in light of the limited number of experiments required to obtain it.

Considering more specifically the covariate values associated with the best configuration found, namely (8.0, 6.0, 400, 300, 5), one observes that the optimal configuration lies on the boundary of the explored design space, with ATP and ClpX reaching their highest tested concentrations. This observation is consistent with the biochemical role of the ClpXP system, where ClpX acts as the ATP-driven motor responsible for substrate unfolding and translocation into ClpP. Higher concentrations of both ATP and ClpX can therefore increase the effective degradation capacity of the system. The fact that the optimum occurs at the edge of the explored region suggests that further improvements might be achieved by extending the admissible concentration ranges in future experiments.

We now turn to the results obtained across the individual Active Learning iterations. The predictive performance metrics of the surrogate model throughout the iterations are summarized in Table 7.1. Table 7.1.

Table 7.1: Evolution of the predictive performance of the surrogate model across Active Learning iterations. Metrics are computed using repeated 3×3 cross-validation. Subscripts f and o denote fold-level and out-of-fold estimates respectively, while *sd* indicates the standard deviation across repetitions.

Iter	n	R^2_f	*sd*	R^2_o	*sd*	RMSE$_f$	*sd*	RMSE$_o$	*sd*	NRMSE$_f$	NRMSE$_o$
0	14	0.469	0.367	0.579	0.057	1.892	0.748	2.001	0.132	0.238	0.252
1	26	-1.209	5.353	0.584	0.225	1.700	0.565	1.752	0.461	0.189	0.194
2	38	0.680	0.155	0.683	0.025	1.441	0.465	1.511	0.059	0.157	0.165
3	50	0.695	0.098	0.708	0.031	1.329	0.279	1.355	0.071	0.142	0.145
4	62	0.603	0.165	0.621	0.104	1.384	0.489	1.446	0.199	0.148	0.154
5	74	0.748	0.147	0.720	0.041	1.090	0.510	1.185	0.085	0.116	0.126

Starting from the first iteration, the model — consisting of an ensemble of 10 random forests, each composed of 200 trees — is trained on a training set containing only 14 labeled designs. This is a very small number of observations from which to obtain a reliable prediction $\hat{f}(x)$ and an accurate estimate of the model performance.

The values of R^2 computed on the validation folds ($R^2_{\text{fold}} = 0.4694$) and out-of-sample ($R^2_{\text{oof}} = 0.5787$) appear initially encouraging, as they suggest the presence of a first predictive signal. However, these scores are highly unstable, particularly the fold-level estimate, which is strongly affected by data partitions generating very small training and validation sets. This instability becomes evident in the following iteration, where the fold-level coefficient of determination reaches a strongly negative value, suggesting performance worse than the baseline model. In practice, this result likely reflects the fact that the model was evaluated on a validation set containing particularly difficult points.

Evidence supporting this interpretation can be observed in the behavior of the other metrics: the RMSE, which measures the standard deviation of the residuals, decreases, while R^2_{oof} increases.

The latter metric is generally more robust than R^2_{fold} (the metric used in the reference paper), since the prediction for each data point is obtained using a model trained on all the remaining observations. For this reason, it is particularly suitable for small datasets. It can also appear slightly more optimistic, as it is less penalized by points for which regression is especially difficult.
Continuing through the Active Learning iterations, the predictive accuracy increases again in the next step, even when evaluated using the metric adopted in the study by Borkowski et al. Both R^2 scores become more aligned and their corresponding standard deviations decrease. At this stage, it becomes possible for the first time to state that the model is not only improving, but also becoming more reliable. The normalized error confirms this improvement, as $NRMSE$ decreases to approximately 0.16.

At iteration 3, a further consolidation of the model performance is observed: R^2 increases again while both $RMSE$ and $NRMSE$ decrease further. This represents the most stable phase of the entire process, with average accuracy, stability, and consistency across different metrics all improving simultaneously. At this stage the ensemble is trained on a dataset containing 50 labeled designs, mainly located in the most promising region of the design space, namely where the concentration of ClpX is non-zero and ATP is relatively high.

In particular, since the exploration parameter β of the acquisition function was set to relatively small values (2 and later 3), the new batch of data was selected primarily on the basis of the highest predicted labels. In addition, the stratified acquisition strategy, introduced to prevent the model from concentrating on only a few regions of the domain, did not prove particularly effective. The difference in scores between regions containing many data points and regions containing few observations remained substantial. As a result, regions characterized by non-zero ClpX and medium–high ATP levels were sampled much more frequently.

Consequently, even randomly selected samples tended to fall within these same regions of the domain. These areas are in fact the richest in data and therefore receive higher acquisition scores than the other regions. Exploring and testing primarily these parts of the design space inevitably leads to improved predictive performance, although the improvement remains limited, especially at the global level.

To partially address this issue and increase the exploration capability of the model, the exploration parameter β was set to 10 when generating the next experimental batch. This change encouraged the selection of points for which the ensemble of random forests exhibited higher predictive uncertainty, while still maintaining a preference for exploitation.

Before proceeding further, it is worth noting that among the 12 experimental designs evaluated in this iteration there is also the one that largely determines the maximum log-rate identified in the study, corresponding to Design 42. From this iteration onward, the best observed log-rate does not change substantially, unlike what happens in the first iteration where the best value increases from approximately -3.84 (estimated through JAGS) to about -2.77.

At iteration 4, a collective deterioration of the metrics is observed, both in terms of average values and standard deviations. This behavior likely reflects the effect of the increased exploration parameter β, which encourages the selection of more informative but also more heterogeneous points. A new acquisition step can therefore expand the explored region of the domain, temporarily worsening some performance metrics while improving the overall informativeness of the dataset. The fact that the model remains substantially better than in the early iterations, and that the uncertainty estimates remain reasonably calibrated, suggests a phase of adjustment rather than a reversal of the overall trend.

By the fifth iteration, the model performance improves again, not only relative to the previous step but also with respect to the entire history of the process. The two coefficients of determination ($R^2_{\text{fold}} = 0.7478$ and $R^2_{\text{oof}} = 0.7197$) are now well aligned, and the standard deviation across out-of-fold repetitions is relatively small. This indicates that the estimated performance is credible and not driven by a particularly favorable data split. Moreover, the normalized prediction errors reach their lowest values, approximately 12%, which is a satisfactory result given the multiple sources of noise and uncertainty affecting this biological system.

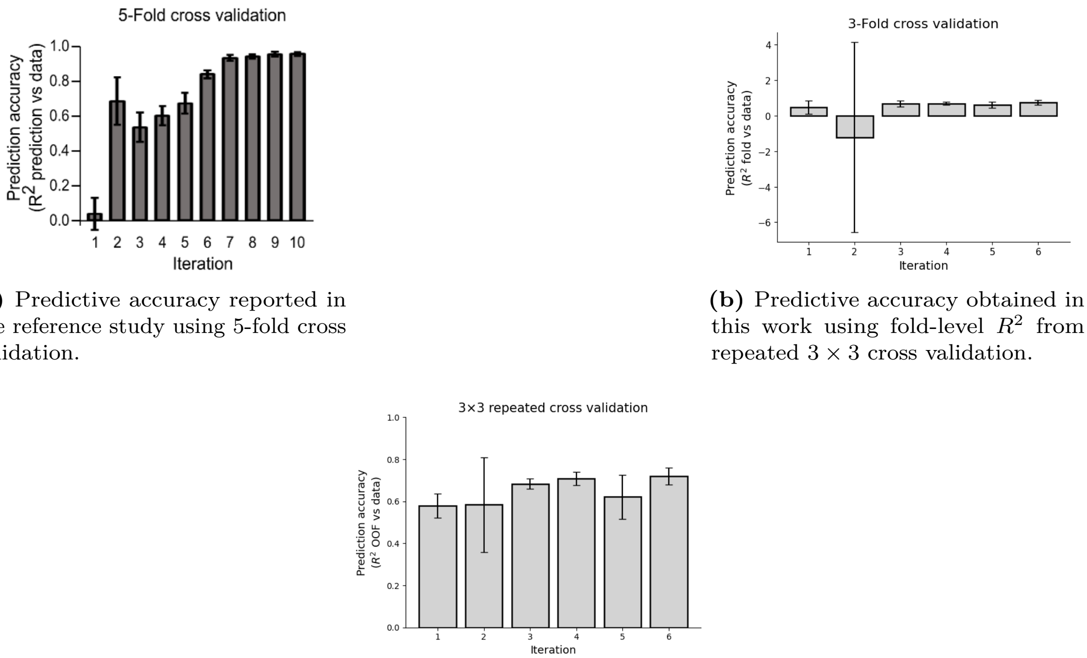
The complexity of the problem also becomes evident when comparing these results with those reported in the reference study. As shown in Figure 7.3, in the original work — whose goal was to identify the optimal combination maximizing protein synthesis yield — the surrogate model exhibits a rapid increase in predictive accuracy as the Active Learning iterations progress, reaching R^2 values close to 1 in the final stages of the algorithm. This behavior reflects both the progressive enrichment of the training dataset and the ability of the Active Learning strategy to efficiently explore the most informative regions of the design space.

In the present work, however, the model must deal with data originating from a biological system in which a different cellular process is being reproduced and optimized. The problem is characterized by higher intrinsic variability and by a limited possibility of generating experimental data, both of which contribute to reducing the maximum predictive accuracy achievable by the surrogate model. Nevertheless, a progressive improvement in model performance is still observed as the number of experimentally evaluated designs increases.

Importantly, the magnitude and stability of the performance depend strongly on the validation metric considered. When examining the fold-level coefficient of determination (R^2_{fold}), the score appears highly variable across iterations, especially during the early stages of the Active Learning cycle. This instability is mainly due to the still limited dataset size, which makes cross-validation estimates particularly sensitive to the specific partition of the data into folds.

In contrast, the coefficient of determination computed using out-of-fold predictions (R^2_{OOF}) provides a more stable estimate of predictive performance. Since each observation is evaluated using a model trained on almost the entire dataset, this metric is less affected by the variability introduced by small validation sets. As shown in Figure 7.3, the values of R^2_{OOF} display a more regular increasing trend and remain positive across all iterations considered.

Overall, although the predictive accuracy achieved in this study does not reach the very high values reported in the reference work, the observed trend confirms that the Active Learning framework still allows a progressive improvement in the quality of the surrogate model. The differences in absolute performance can plausibly be attributed to the smaller size of the experimental dataset, the higher intrinsic variability of the biological system under investigation, and the limited exploration of certain regions of the design space.



(a) Predictive accuracy reported in the reference study using 5-fold cross validation.

(b) Predictive accuracy obtained in this work using fold-level R^2 from repeated 3 × 3 cross validation.

(c) Predictive accuracy obtained in this work using out-of-fold R^2.

Figure 7.3: Comparison between the predictive accuracy reported in the reference Active Learning study and the performance obtained in the present work. The first panel reproduces the evolution of the cross-validation score reported in the original paper, while the second and third panels show the corresponding metrics computed for the surrogate model developed in this study using fold-level and out-of-fold validation schemes.

7.3 Final Considerations

Table 7.2: Best experimental design identified as the dataset grows during the Active Learning procedure. For each stage, the table reports the best observed log-rate y_μ and the corresponding experimental conditions.

N design	Best design	y_μ	μ_{var}	ATP	Mg	ClpX	ClpP	PEG
12	**Design6**	-3.837378	0.04519	8.0	6.0	400	400	6
24	**Design13**	-2.766299	0.003899	8.0	10.0	400	400	6
36	**Design27**	-2.599642	0.012804	8.0	10.0	400	300	6
48	**Design42**	-2.412609	0.004319	8.0	6.0	400	300	6
60	**Design42**	-2.412609	0.004319	8.0	6.0	400	300	6
72	**Design68**	**-2.397245**	0.211629	8.0	6.0	400	300	5

As can be observed from Table 7.2, which reports the evolution of the best design identified during the Active Learning process, the last iterations did not produce a substantial improvement in the maximum observed log-rate. Between iterations 3 and 4, the best value changes only slightly, from −2.4058 to −2.4120. The corresponding design remains the same; however, since the label is estimated through posterior inference with JAGS, small fluctuations may occur due to the sampling procedure inherent to the Bayesian framework.

In the final iteration, a design was identified that appears slightly better than the previous one. However, this improvement is most likely the result of fluctuations arising from the propagation of errors between the inferred estimates and the experimental measurements.

Despite the fact that the predictive accuracy had not yet reached a clear stability, the decision was made not to proceed with further iterations due to practical experimental constraints. In particular, ClpX represents a relatively costly resource to produce. The protein must be expressed and purified using standard protein expression and purification protocols, which typically require one or two days of experimental work. Moreover, as is common for many ATPases belonging to the AAA+ family, the enzymatic activity of ClpX can decrease over time or after repeated freeze–thaw cycles, making it preferable to use fresh and homogeneous preparations across comparable experimental runs.

For this reason, the experiments were designed to use a single batch of ClpX produced within the same expression and purification cycle. The use of different batches could introduce systematic variations in the enzymatic activity of the protein, for instance due to differences in oligomerization state, effective concentration of the active form, or the presence of residual impurities. Such batch effects would make it more difficult to directly compare results obtained across different iterations of the Active Learning algorithm, introducing an additional source of uncontrolled biological variability.

Considering these practical aspects, and given that by iteration 5 the algorithm already exhibited a reasonably good predictive capability of the surrogate model together with a substantial stabilization of the best observed design, the experimental cycle was terminated at this stage. This decision allowed the entire dataset to remain within the same biological and experimental context, avoiding potential batch effects that could have compromised the interpretability of the results.

Overall, although the accuracy metrics show a progressive improvement up to the final iteration, the magnitude of these improvements gradually decreases, suggesting that the model may already be approaching a performance plateau. This behavior can be interpreted in light of some structural limitations of the adopted framework. The main ones include:

- the metrics used to evaluate the model performance;
- the generation and selection of candidates for the new experimental batches.

The performance of the model is monitored through prediction accuracy and residual errors computed on the dataset progressively constructed during the Active Learning iterations. This dataset is partitioned into a training subset used to fit the model and a validation subset used to evaluate predictive performance.

By considering the complete dataset, consisting of feature vectors and their corresponding labels, and computing the quantities defining the acquisition function (namely μ and σ), it is possible to observe in some designs a significant discrepancy between the true label and the model prediction. In these cases, large residual values — for instance for Design 9, where the residual is approximately 5 — are not associated with correspondingly large predictive uncertainty σ. The designs characterized by the largest prediction residuals are reported in Table 7.3.

This means not only that the model makes substantial prediction errors, but also that it does so with a relatively high degree of confidence. This behavior highlights an important limitation of the predictive model adopted in this work. Tree-based models partition the feature space through recursive splits that are constructed exclusively on the basis of the input variables rather than the labels. As a consequence, once a data point is assigned to a given region of the space — represented by the observations located in the corresponding leaf node — the predicted label is assumed to be close to the average value of the labels associated with that region.

Table 7.3: Designs with the largest absolute prediction residuals, aggregated across cross-validation repetitions. Regions are defined according to the experimental design space partition based on ClpX presence and ATP concentration.

Design	Region	y_{true}	$\hat{y}$	σ_{pred}	Residual	|Residual|	ATP	Mg	ClpX	ClpP	PEG
Design9	R3	-11.398557	-6.409471	0.142539	-4.989086	4.989086	8.0	14.0	0.0	400.0	6.0
Design1	R1	-11.803096	-8.093816	0.159382	-3.709281	3.709281	0.0	14.0	0.0	0.0	3.0
Design48	R2	-4.101769	-7.062664	0.149769	2.960895	2.960895	4.0	6.0	0.0	200.0	3.0
Design2	R1	-10.626623	-8.159385	0.116942	-2.467238	2.467238	0.0	6.0	0.0	0.0	6.0
Design3	R3	-10.296856	-8.031635	0.182302	-2.265220	2.265220	8.0	6.0	0.0	0.0	3.0
Design59	R5	-4.695607	-6.891521	0.146830	2.195914	2.195914	2.0	7.5	150.0	0.0	5.0
Design47	R4	-4.020924	-6.184493	0.145093	2.163569	2.163569	0.0	10.0	400.0	0.0	4.0
Design35	R5	-4.228206	-5.960612	0.205640	1.732406	1.732406	1.0	14.0	100.0	200.0	4.0
Design36	R5	-9.829644	-8.129832	0.142136	-1.699813	1.699813	1.0	6.0	50.0	0.0	5.0
Design23	R4	-3.554050	-5.249411	0.112781	1.695361	1.695361	0.0	6.0	400.0	200.0	3.0
Design58	R4	-7.322930	-5.703637	0.170012	-1.619293	1.619293	0.0	7.5	200.0	400.0	4.0
Design33	R4	-8.013377	-6.680433	0.168774	-1.332944	1.332944	0.0	7.5	200.0	0.0	4.0
Design34	R2	-8.859397	-7.595853	0.164930	-1.263543	1.263543	2.0	14.0	0.0	300.0	6.0
Design68	R6	-2.397245	-3.581941	0.073591	1.184697	1.184697	8.0	6.0	400.0	300.0	5.0
Design72	R2	-7.431591	-8.593027	0.150613	1.161435	1.161435	1.0	7.5	0.0	0.0	6.0

With a limited number of observations and with an underlying true function that varies substantially even within the same region of the feature space, the model is unable to identify a function that accurately approximates the true relationship between inputs and outputs. Consequently, the predictive performance cannot reach very high levels.

Moreover, the largest residuals are associated with designs characterized by zero ClpX concentration. During the Active Learning cycle, only 10 points in total were explored in these regions. This number is too small to adequately learn the variability within those regions, yet large enough to negatively affect the validation metrics used to evaluate predictive accuracy.

A more substantial improvement in accuracy would likely have required additional Active Learning iterations. However, if the same pipeline had been maintained, it is plausible that new experimental batches would have continued to be selected mainly within the same regions of the design space already identified as promising and already relatively well explored. In such a scenario, the expected gains would probably have remained modest, particularly in terms of global predictive accuracy, since the model would have continued refining its knowledge within a restricted portion of the experimental domain.

Achieving a more substantial improvement in predictive performance would instead have required either a drastic increase in the sampling density within the most relevant regions or a forced exploration of currently under-sampled areas of the design space, including those of lower experimental interest. However, such a strategy would have required a non-trivial modification of the candidate selection pipeline, partially departing from the structure induced by the surrogate model and the acquisition function used in this study.

Since the primary goal of this work is to characterize and optimize the protein degradation process mediated by the ClpXP complex, experimental configurations with zero ClpX concentration are of limited biological relevance, as they do not correspond to conditions in which significant degradation can occur.

For this reason, it was not considered a priority to design an Active Learning pipeline aimed at systematically exploring regions of the design space characterized by zero ClpX concentration. Instead, the focus was placed on the regions that are most relevant for studying the phenomenon of interest.

Considering these aspects, and given that by iteration 5 the surrogate model already exhibited a reasonable predictive capability and the best observed design appeared to have largely stabilized, it was considered appropriate to stop the Active Learning cycle at this point. This choice inevitably involves some compromises in terms of global prediction accuracy and in the precise estimation of the optimum, but it allows the analysis to remain focused on the experimental configurations that are most relevant from a biological perspective.

Overall, the results obtained show that the Active Learning approach made it possible to rapidly identify promising regions of the design space, progressively improving the quality of the surrogate model and leading to the identification of experimental conditions associated with higher protein degradation rates. Despite some intrinsic limitations of the predictive model, which prevented a complete identifiability of certain parameters — including the log-rate — the estimates proved sufficiently accurate for the framework to effectively guide the exploration of the experimental space using a relatively limited number of experiments.

Chapter 8

Conclusions

In this work, an integrated framework for the study and optimization of protein degradation in a cell-free system was developed and tested, combining Bayesian inference implemented through `JAGS` with an Active Learning strategy based on surrogate models. Overall, the results obtained can be considered satisfactory, especially in light of the complexity of the biological system under investigation, the limited availability of experimental data, and the multiple sources of noise affecting the process.

From an inferential perspective, the Bayesian model exhibited heterogeneous performance across the different experimental designs. In particular, the first stage of the model, dedicated to estimating the batch effect, produced relatively stable results and revealed a non-negligible but moderate impact on the estimated degradation rates. Greater difficulties emerged in the second stage, where the simultaneous estimation of the final plateau, the decay rate, and the curve-specific variability made some parameters only partially identifiable. This limitation was particularly evident for designs characterized by stronger degradation dynamics, where the final portion of the fluorescence trajectories appeared more sensitive to experimental noise. Nevertheless, the inferred estimates proved sufficiently robust to provide a consistent and informative description of the system.

From the optimization perspective, the Active Learning procedure allowed an efficient exploration of the design space, identifying in only a few iterations promising experimental regions associated with high protein degradation rates. In particular, the framework was able to identify an experimental configuration corresponding to a log-rate of approximately $\mu \approx -2.39$, which corresponds to a degradation rate $k \approx 9 \times 10^{-2}\,\mathrm{min}^{-1}$, i.e. roughly a 9% decrease per minute in the remaining fluorescence signal. Although the predictive accuracy of the surrogate model did not reach extremely high values, the progressive improvement of the validation metrics and the rapid stabilization of the best observed designs indicate that the framework was able to extract useful information even from labels affected by inferential uncertainty. This represents one of the most interesting outcomes of the study: despite relying on a two-stage model based on simplifying assumptions, the overall pipeline proved robust enough to support the optimization process.

Naturally, the present work also presents some limitations. These include the truncation of fluorescence trajectories after 50 minutes, the preliminary normalization of the data, the assumption that batch effects primarily act on the degradation rate rather than explicitly on the plateau parameter, and the dependence of the inference quality on the noise affecting the tail of the trajectories. Furthermore, the acquisition strategy adopted within the Active Learning framework naturally favored certain regions of the design space, leaving others relatively underexplored.

Overall, however, the results obtained show that the proposed approach provides a solid methodological basis for the analysis of the problem. On the one hand, Bayesian inference allowed the extraction of interpretable dynamic information from noisy data affected by batch effects; on the other hand, the Active Learning framework enabled these inferred quantities to be used effectively to guide the search for experimental conditions promoting protein degradation. Looking forward, possible extensions of this work could include a more complete hierarchical model capable of explicitly accounting for variability of the plateau parameter b across experimental days and replicates, as well as more refined acquisition strategies aimed at better balancing exploration and exploitation in the less sampled regions of the design space. In this sense, the proposed framework proved to be effective for the experimental optimization of protein degradation processes in cell-free systems, suggesting potential further applications in the field of biorobotics.

Bibliography

[1] Yusei Hattori. «Investigation of the effects of composition in a cell-free protein synthesis–degradation system on protein synthesis and degradation». Bachelor's thesis. Kyoto Institute of Technology, 2024 (cit. on pp. 1, 11, 65).

[2] Olivier Borkowski, Mathilde Koch, Agnès Zettor, Amir Pandi, Angelo Cardoso Batista, Paul Soudier, and Jean-Loup Faulon. «Large scale active-learning-guided exploration to maximize cell-free production». In: *bioRxiv* (2019). DOI: `10.1101/751669` (cit. on pp. 9, 12, 55, 56, 61, 75).

[3] Gianluca Mastrantonio. *Statistica Bayesiana.* Lecture slides, Politecnico di Torino. 2023 (cit. on p. 32).

[4] Radford M. Neal. «Slice sampling». In: *Annals of Statistics* 31.3 (2003), pp. 705–767 (cit. on pp. 37, 38).

[5] Walter Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in Practice.* Chapman and Hall, 1996 (cit. on p. 38).

[6] Christian Robert and George Casella. *Monte Carlo Statistical Methods.* Springer, 2004 (cit. on p. 38).

[7] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984 (cit. on p. 49).

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd ed. Springer, 2009 (cit. on pp. 49, 51).

[9] Leo Breiman. «Bagging Predictors». In: *Machine Learning* 24.2 (1996), pp. 123–140. DOI: `10.1007/BF00058655` (cit. on p. 51).