A Bayesian Approach to Modeling Tamoxifen Resistance in Breast Cancer Cells Through Adaptive Hamiltonian Monte Carlo Posterior Sampling

A Thesis

by

Miguel Fernandez-de-Retana

Supervisor: Aitor Almeida Escondrillas

Collaboration Acknowledgment

This project was conducted as a collaborative effort between

their combined support and resources significantly contributed to the success of this project.

Submitted in partial fulfillment of the academic requirements established by the

University of Deusto – Faculty of Engineering

Deusto

Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Abstract

Breast cancer is a major health concern globally, with diagnoses and projected cases rising significantly, becoming the most prevalent cancer among women and a leading cause of cancer-related deaths in the population. Hormone therapies, particularly tamoxifen, are vital for treating estrogen receptor-positive (ER^+) breast cancer, drastically improving survival rates by reducing recurrence and mortality. However, a significant challenge remains, as a substantial portion of patients develop resistance to the drug (estimated to be between 30-50%) within a critically long 5-year treatment window. To address this, our project proposes to analyze the complex mechanisms driving tamoxifen resistance. Employing a Bayesian modeling framework, and leveraging RNA sequencing data from cell-lines (in collaboration with the CIC bioGUNE research center) and publicly available patient data, the study aims to unravel the intricacies of this resistance phenomenon. To this end, we have further developed pyHaiCS, a Python library for Computational Statistics featuring a wide range of Hamiltonian sampling algorithms, including single-chain and multi-chain variants; a variety of numerical schemes for the integration of the simulated Hamiltonian dynamics, or a novel adaptive algorithm for the automatic tuning of the parameters. Finally, the project aims to identify key genetic biomarkers linked to tamoxifen resistance and develop robust, clinically applicable predictive models for patient prognosis under endocrine therapy. Ultimately, identifying therapeutic targets, paving the way for personalized and more effective treatments to improve patient outcomes. In practice, the methodologies developed here are intended to be broadly generalizable to other cancer types and drug resistance mechanisms.

Keywords

Bayesian Computational Statistics · Breast Cancer Research · Gene Expression Analysis · Hamiltonian Monte Carlo (HMC) · Hormone Therapy · Tamoxifen Resistance

Contents

1 Introduction 1

2 Objectives & Scope 4
- 2.1 General Objectives . . . 4
- 2.2 Specific Objectives . . . 5
- 2.3 Scope of the Project & Limitations . . . 6

3 Fundamentals & Theoretical Background: An Elemental Initiation to Bayesian Clinical Analysis 9
- 3.1 A *Brief* Introduction to Breast Cancer Research . . . 9
- 3.2 An Introduction to Bayesian Modeling & Clinical Studies . . . 12
- 3.3 An Introduction to Hamiltonian-based Monte Carlo Methods . . . 15
- 3.4 Related Work . . . 38

4 Bayesian Modeling of Tamoxifen Resistance in MCF7 Cells 41
- 4.1 Available Multi-Source Genetic Data . . . 41
- 4.2 Pre-Processing & Statistical Analysis of Genetic Sequencing Data . . . 43
- 4.3 Integration of Cell-Line & Patient RNA-seq Data for Genetic Biomarker Identification . . . 49
- 4.4 Python in Hamiltonian for Computational Statistics . . . 50
- 4.5 Bayesian Logistic Regression & Other Resistance Models . . . 61

5 Results & Discussion 72
- 5.1 Evaluation Metrics . . . 72
- 5.2 Results of Tamoxifen Prognosis Models . . . 74
- 5.3 Identification of Potential Genetic Biomarkers . . . 81

6 Planning, Budgeting & Ethical Considerations of the Project 87
- 6.1 Planning & Budgeting of the Project . . . 87
- 6.2 Ethical Reflection & Assessment of the Study . . . 92

7 Conclusions & Future Work 98
- 7.1 Conclusions & General Assessment . . . 98
- 7.2 Future Work & Promising Research Directions . . . 100

Bibliography 104

List of Figures

1.1 Overview of the Thesis Structure . . . 3

2.1 Visual Summary of the Project's Scope and Limitations . . . 7

3.1 Molecular Classification of Breast Cancer Subtypes . . . 11
3.2 Tamoxifen Mechanism of Action in Breast Cancer Cells . . . 12
3.3 Hierarchy of Hamiltonian-based Monte Carlo Methods . . . 26

4.1 Distribution of Tamoxifen Responses in TCGA-BRCA Patients . . . 42
4.2 Pre-processing Workflow for RNA-seq Data . . . 43
4.3 Illustration of the Gene Mapping/Alignment Process . . . 44
4.4 Differential Expression Analysis (DEA) Workflow . . . 45
4.5 Volcano Plot of Statistically Significant Differentially Expressed Genes . . . 49
4.6 Distribution of Genes in the Joint Analysis of Cell-Line & Patient Data . . . 50
4.7 Logo of the pyHaiCS Library . . . 51
4.8 Summary of the Features in the pyHaiCS Ecosystem . . . 52
4.9 General Features of pyHaiCS . . . 53
4.10 Tree Visualization of Features in pyHaiCS . . . 55
4.11 Exploration of Space in Sampling from a Banana-Shaped Distribution . . . 56
4.12 Flow Diagram for the SE_MI_KR Compartmental Model w/ Transmission Rate $\beta(t)$ . . . 58
4.13 Daily COVID-19 Incidence in the Basque Country . . . 60
4.14 Optical Talbot Effect of a Wave . . . 61
4.15 Comparison Between the Three Main Paradigms of Ensembling . . . 64
4.16 Visualization of the Random Forest Ensemble Method . . . 65
4.17 Multi-Layer Perceptron (MLP) Architecture . . . 66
4.18 Autoencoder (w/ Classifier Module) Architecture . . . 67
4.19 Variational Autoencoder (VAE w/ Classifier Module) Architecture . . . 68
4.20 Bayesian Neural Network (BNN) Architecture . . . 69
4.21 Summary of the Different Approaches to Training Bayesian Models . . . 70

5.1 Relationship Between the MCC and F_1 Score . . . 73
5.2 Recall/Sensitivity by Model (Average Across 5-Folds of Stratified CV) . . . 77
5.3 MCC by Model (Average Across 5-Folds of Stratified CV) . . . 78
5.4 Global Gene Contribution . . . 83
5.5 Gene Contribution in Resistant Cohort . . . 83
5.6 Local Gene Contribution . . . 84

5.7 Kaplan-Meier Survival Analysis for the Identified Genes in ER+ Breast Cancer Patients . . . 85

6.1 Key Components of Project Management & Their Relationship to Project Success 88
6.2 Project Chronogram: Tasks & Milestones – Weekly Gantt Chart 91

7.1 Visual Summary of the Project 99
7.2 Visual Summary of Future Work Directions 101

List of Tables

2.1 Specific Objectives for Biomarker Identification and Prognostic Modeling 5
2.2 Specific Objectives for pyHaiCS Library Development 6

3.1 Jeffrey's Scale for Interpreting the Bayes Factor 14
3.2 Comparison of Relevant Properties of Hamiltonian-based Samplers 26
3.3 Evaluation Metrics for Estimating the Performance of Hamiltonian-based Sampling Methods 30
3.4 Special Cases of 2- & 3-Stage Splitting Integrators 33
3.5 Summary of Related Work in Bayesian Methods and Breast Cancer Research 40

4.1 Commonly Used RNA-seq Normalization Methods 47
4.2 Datasets Used for Benchmarking the BLR Model 57
4.3 Popular Kernels for Support-Vector Machines 63

5.1 Performance of Tamoxifen *Point-Estimate* Prognostic Models 75
5.2 Performance of Tamoxifen *Bayesian* Prognostic Models 76

6.1 Human Resource Plan: Roles, Responsibilities & Expertise 89
6.2 Project Tasks by Phase 90
6.3 Technical Equipment & Travel Expenses Budget 92
6.4 Human Resources Budget (by Institution & Role) 93

List of Algorithms

1 Random-Walk Metropolis-Hastings (RW-MH) 17
2 Hamiltonian Monte Carlo (HMC) 20
3 Verlet/Leapfrog Numerical Integrator 21
4 Generalized Hamiltonian Monte Carlo (GHMC) 22
5 Mix & Match Hamiltonian Monte Carlo (MMHMC) 23
6 Adaptive Integration Approach in Computational Statistics (s-AIA) 35
7 Tuning Stage of s-AIA (s-AIA-Tuning) 36
8 Burn-In Stage of s-AIA (s-AIA-Burn-In) – Part 1 37
8 Burn-In Stage of s-AIA (s-AIA-Burn-In) – Part 2 38

I Introduction

> *"'Begin at the beginning,' the King said, very gravely, 'and go on till you come to the end: then stop.'"*
>
> ~ Lewis Carroll, *Alice in Wonderland* (1865) [1]

In the field of medical research, breast cancer stands out as a significant global health challenge, especially for women, where it represents the most commonly diagnosed cancer and a leading cause of cancer-related mortality [2]. The complexity of breast cancer is accentuated by its *heterogeneity*, which manifests in various subtypes characterized by distinct biological behaviors with diverse responses to treatment (as presented in Section 3.1). Among these, estrogen receptor-positive (ER^+) breast cancer accounts for approximately 70-80% of all cases, making it the most prevalent subtype [3]. The estrogen receptor (ER) plays a pivotal role in the pathogenesis of ER^+ breast cancer, as its activation by estrogen promotes *cell proliferation* and *tumor growth*. Consequently, endocrine therapy targeting the ER signaling pathway has become a cornerstone of treatment for this subtype [4–7]. For several decades, tamoxifen, a selective estrogen receptor modulator (SERM), has become the standard therapy for ER^+ breast cancer. By binding to estrogen receptors, tamoxifen acts as an antagonist in breast tissue, effectively blocking estrogen's proliferative effects. This mechanism has led to significant improvements in recurrence-free and overall survival rates for patients with ER^+ breast cancer, reducing recurrence by ~50% and mortality by ~30% after a standard 5-year treatment [8].

Introduction & Project Motivation

Despite the remarkable success of tamoxifen, a significant clinical challenge persists: between 30% and 50% of patients with ER^+ breast cancer will eventually develop resistance to this therapy, leading to disease recurrence and metastasis [9]. This phenomenon of resistance can be particularly insidious, where tumors initially respond but eventually relapse and progress during or after treatment. Understanding the biological mechanisms that drive tamoxifen resistance is therefore of paramount importance for improving patient outcomes, more so when we consider that tamoxifen treatments may last as long as 5 years. It is thus crucial to identify patients at risk of developing resistance early in their treatment course, as this could inform clinical decisions and guide the selection of alternative therapeutic strategies, hopefully paving the way for more personalized treatment approaches in precision oncology [10].

Addressing this complex biological problem requires sophisticated analytical approaches capable of handling high-dimensional genomic data extracted from cellular samples. Moreover, the challenge lies in the need to identify relevant signatures of potential genetic biomarkers predictive of tamoxifen resistance. In this study, we propose to leverage the power of Bayesian statistical methods to tackle this challenge, by combining RNA sequencing data from both *patients* and lab-grown *cell-lines*. Unlike traditional frequentist methods, Bayesian

inference naturally incorporates prior expert knowledge and provides a formal way to quantify uncertainty in estimates and predictions. This ability to model uncertainty is crucial when dealing with noisy biological data and patient heterogeneity. Moreover, to efficiently explore the complex posterior distributions arising from high-dimensional genomic data, we employ Hamiltonian-inspired methods, such as Hamiltonian Monte Carlo (HMC) and its variants. These methods leverage the principles of classical mechanics to navigate the parameter space more effectively, enabling faster convergence and more accurate sampling compared to traditional Markov-Chain Monte Carlo (MCMC) approaches.

Thus, the overarching aim of this project is to develop and implement sophisticated Bayesian statistical models designed to integrate relevant sequencing data associated with tamoxifen response; to utilize these models to identify potential signatures that are predictive of resistance; to rigorously evaluate the predictive accuracy and robustness of the developed models using appropriate validation techniques; and finally, to explore the biological implications (and limitations) of our findings, seeking to highlight key pathways or genes critically involved in the resistance mechanism. Likewise, central to this work is the development of the *open-source* library pyHaiCS, which facilitates the implementation of Hamiltonian-inspired sampling methods for computational statistics. This library is designed to be user-friendly and accessible, allowing researchers to easily apply advanced sampling techniques to their own data. The library is built on top of the popular Python library JAX (by Google) [11, 12], which provides efficient numerical computations, native support for hardware accelerators (such as GPUs and TPUs), automatic differentiation, and ensures compatibility with existing Python data analysis workflows.

General Structure

This thesis is organized to guide the reader through the research process, from the initial background and motivation to the final discussion and implications of our findings (see Figure 1.1). Each chapter is designed to build upon the previous ones, providing a logical progression of ideas and concepts. First, Chapter 2 outlines the specific objectives, scope, and limitations of the study, and the research questions we aim to address. This is followed by Chapter 3, which seeks to provide a thorough introduction to the fundamentals of breast cancer biology, the role of endocrine therapy with tamoxifen, and an overview of the basics of Bayesian statistical modeling. This chapter also includes a section which hopefully serves as a standalone initiation to the theoretical foundations behind Hamiltonian-based Monte Carlo sampling methods, as well as a dedicated section (Section 3.4) that reviews related prior research in the field, providing a comprehensive literature overview.

Building on these foundations, Chapter 4 details the sequencing data used across this work, the specific preprocessing steps applied, and the development of the integrated Bayesian models. Then, the pyHaiCS library is introduced, including its design principles, key features, and examples of its usage. The chapter ends with a description of the Bayesian Logistic Regression (BLR) method used across this study, as well as the other resistance models developed. The results of these prognostic models are then presented in Chapter 5, along with the evaluation metrics used to assess their performance and a model explainability analysis using SHAP values. This chapter also includes a detailed analysis and external validation of the identified biomarkers, including their potential implications in underlying biological pathways and their relevance to tamoxifen resistance. Likewise, this chapter finds the discussion and interpretation of these findings in the context of existing knowledge, and critically evaluates the study's limitations.

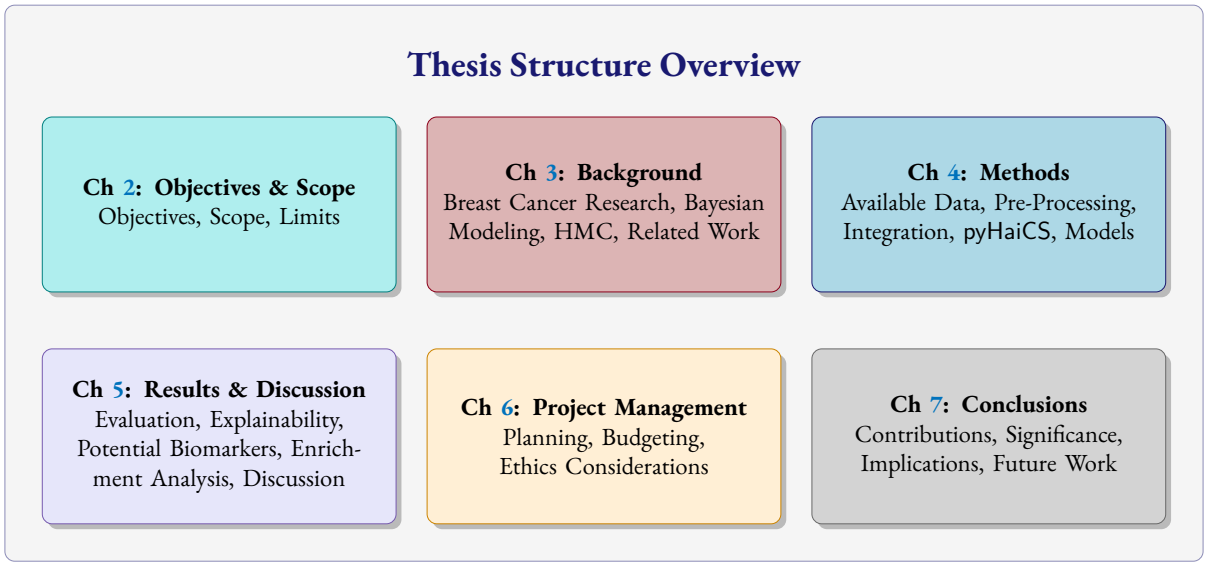

Figure 1.1: Overview of the Thesis Structure

From the perspective of project management, Chapter 6 outlines the planning, budgeting, and ethical considerations associated with the project. It provides a comprehensive overview of the project's timeline, including key milestones and tasks, as well as the human and technical resources and costs required to successfully complete our investigation. This chapter also addresses ethical considerations related to data collection, analysis, and interpretation, ensuring that our research adheres to established ethical guidelines and standards, as well as considerations for data privacy and a gender perspective in such a sensitive field. Finally, Chapter 7 summarizes the key contributions of this work, highlighting its significance in advancing our understanding of tamoxifen resistance in breast cancer. The chapter also discusses the potential implications of the findings for clinical practice and patient treatment, as well as future promising research directions in this area, including potential extensions of the models developed or the integration of additional data.

2 Objectives & Scope

> *"Not everything that counts can be counted, and not everything that can be counted counts."*
>
> ~ Albert Einstein

In this chapter, we outline the *objectives* of the project, as well as the *scope* and *limitations* within which the research is conducted. The objectives are divided into general and specific objectives, which will guide the development and evaluation of the project through explicitly measurable goals. The general objectives provide a broad overview of the intended outcomes, while the specific objectives break down these goals into more detailed and actionable tasks. Additionally, the scope defines the boundaries and areas of focus of the project, while the limitations acknowledge potential constraints set by the research design, methodology, or available resources.

2.1 General Objectives

In terms of the general objectives of this project, our aim is twofold. First, we seek to identify potential *genetic biomarkers* associated with the resistance of ER^+ breast cancer cells to tamoxifen therapy, hopefully leading to improved prognostic models for predicting treatment strategies and patient outcomes. In more practical terms, we aim to develop a set of models that can accurately predict the likelihood of tamoxifen resistance in breast cancer patients based on their genetic profiles. This will involve the integration of RNA sequencing data from both patients and lab-grown cell-lines, the development of advanced Bayesian models, and to rigorously evaluate their performance using appropriate validation techniques. The resulting models should be interpretable in their predictions, and the identified biomarkers should be biologically relevant, providing insights into the underlying mechanisms of tamoxifen resistance. The ultimate goal is to provide clinicians with a reliable tool for identifying patients at risk of developing resistance, enabling more personalized treatment strategies (e.g., through alternative therapies or treatments).

Second, due to the general scarcity of open-source libraries for Bayesian modeling in Python — and the lack of user-friendly tools for implementing advanced sampling techniques — we aim to develop an *open-source* library for computational Bayesian statistics based on Hamiltonian-inspired Monte Carlo sampling methods: pyHaiCS. This library will facilitate the implementation of advanced sampling techniques in Bayesian modeling, making them more accessible to researchers in the field. The library will be designed to be user-friendly and compatible with existing data analysis and Machine Learning workflows, allowing researchers to easily apply advanced sampling techniques to their own data and research inquiries.

Table 2.1: Specific Objectives for Biomarker Identification and Prognostic Modeling

ID	Specific Objective Description
SO 1.1	Acquire and pre-process RNA-seq data from MCF7 cell-lines (control vs. tamoxifen-resistant replicates).
SO 1.2	Acquire and pre-process RNA-seq for ER^+ tamoxifen-treated breast cancer patients from the TCGA-BRCA cohort.
SO 1.3	Perform Differential Expression Analysis (DEA) independently on cell-line data and patient data to identify genes significantly expressed in *resistant* phenotypes.
SO 1.4	Integrate cell-line and patient DEA results to establish a refined list of candidate biomarkers exhibiting concordant expression changes across both data sources.
SO 1.5	Formulate and implement Bayesian Logistic Regression (BLR) models incorporating cell-line derived differential expression information as informative priors for model parameters.
SO 1.6	Develop and evaluate a range of baseline Machine Learning models (e.g., standard Logistic Regression, Support Vector Machines, Random Forest, MLP) for comparative analysis against the Bayesian models.
SO 1.7	Implement and evaluate advanced Bayesian models, specifically Bayesian Neural Networks (BNNs), to capture complex relationships.
SO 1.8	Rigorously evaluate the predictive performance of all developed models using stratified cross-validation and metrics suitable for imbalanced data (Recall, MCC, F1-score), explicitly addressing class imbalance through data augmentation (SMOTE).
SO 1.9	Employ model explainability techniques (e.g., SHAP) on best-performing models to identify and rank the contribution of individual genes to the prediction of tamoxifen resistance.
SO 1.10	Validate the prognostic significance and potential clinical relevance of the identified key biomarkers using external datasets and survival analysis methodologies (e.g., Kaplan-Meier Survival Analysis, Enrichment Analysis).

2.2 Specific Objectives

Now that we have outlined the general objectives of the project, we can break them down into more specific and actionable tasks, that are measurable and achievable within the scope of this study. The primary general objective of this research is to identify potential genetic biomarkers predictive of tamoxifen resistance in breast cancer through the integration of cell-line and patient data. To achieve this overarching goal, a series of specific, measurable steps must be undertaken to ensure that the research is conducted systematically and effectively. These steps include the acquisition of the data, its pre-processing, the development of an integration pipeline to combine the data from the two different sources, the development and comparative evaluation of various predictive models (including our proposed Bayesian framework), and thorough validation and interpretation of the findings to ascertain their biological relevance and potential clinical utility. The detailed specific objectives related to biomarker discovery and prognostic modeling are outlined in Table 2.1.

Table 2.2: Specific Objectives for pyHaiCS Library Development

ID	Specific Objective Description
SO 2.1	Implement core Hamiltonian Monte Carlo (HMC) and Generalized HMC (GHMC) sampling algorithms (single-chain and multi-chain) utilizing JAX for performance optimization (automatic differentiation, JIT compilation, hardware acceleration).
SO 2.2	Implement a thorough suite of numerical integrators for simulating Hamiltonian dynamics, including the standard Verlet/Leapfrog integrator and parametrizable Multi-Stage Splitting Integrators (MSSIs, e.g., 2-stage and 3-stage).
SO 2.3	Develop and implement the s-AIA algorithm for automatic and adaptive tuning of HMC/GHMC sampler and integrator parameters.
SO 2.4	Design and implement a modular, intuitive, and user-friendly API.
SO 2.5	Integrate standard Markov-Chain Monte Carlo (MCMC) diagnostic tools (e.g., Potential Scale Reduction Factor - PSRF, Effective Sample Size - ESS, Monte Carlo Standard Error - MCSE) for assessing sampling convergence and efficiency.
SO 2.6	Create comprehensive documentation, including detailed API references, practical tutorials showcasing usage, and diverse benchmark examples (e.g., Bayesian Logistic Regression, multivariate Gaussian sampling, epidemiological models) for testing and demonstration.
SO 2.7	Release pyHaiCS as a publicly accessible open-source library to encourage community use and contribution.

Complementing the biological research, the second general objective addresses a methodological gap by focusing on the development of the open-source Python library, pyHaiCS. This library seeks to provide the scientific community with an accessible, efficient, and user-friendly tool for implementing advanced Hamiltonian-inspired Monte Carlo sampling techniques for Bayesian inference, exploiting the capabilities of the JAX framework. The development process involves implementing core and advanced algorithms, ensuring seamless integration with existing scientific Python workflows, and providing comprehensive support for users to facilitate broader adoption. The specific tasks required to realize this objective are enumerated in Table 2.2.

2.3 Scope of the Project & Limitations

To ensure clarity regarding the boundaries of this research endeavor, this section delineates the specific scope of the project and acknowledges its inherent limitations (as summarized in Figure 2.1). The scope of this thesis is centered on the investigation of tamoxifen resistance mechanisms within the specific context of estrogen receptor-positive (ER^+) breast cancer. Methodologically, the core focus lies in the application and development of advanced Bayesian statistical modeling techniques, particularly Bayesian Logistic Regression (BLR) utilizing Hamiltonian Monte Carlo (HMC) based inference and Bayesian Neural Networks (BNNs). A key aspect within the scope is the integration of multi-source genomic data, specifically leveraging RNA sequencing data from established MCF7 cell-lines (control versus tamoxifen-resistant) to inform prior distributions within models trained on patient data. The patient data considered is restricted to ER^+ cases documented as

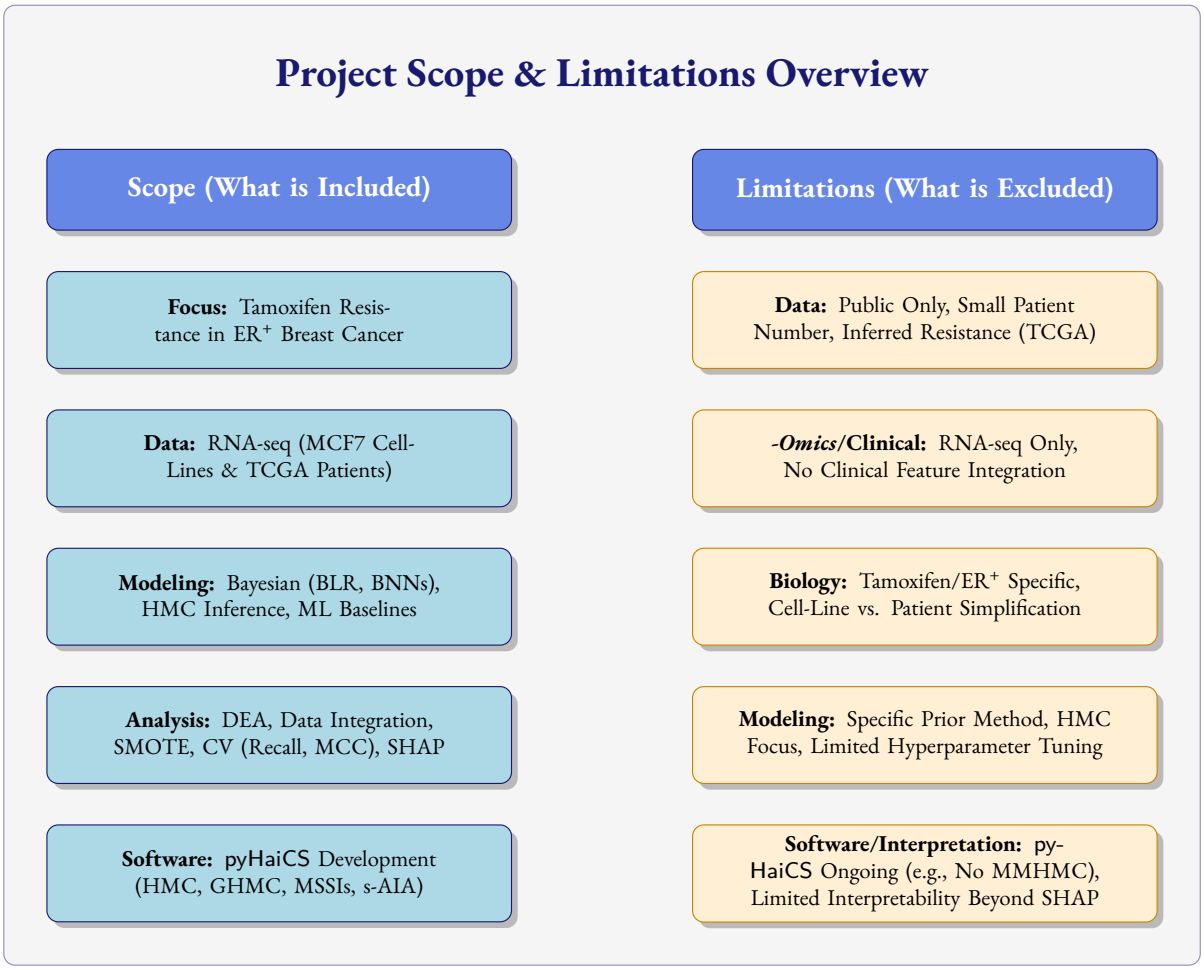

Figure 2.1: Visual Summary of the Project's Scope and Limitations

having received tamoxifen treatment, with resistance inferred from clinical outcome records. The analytical pipeline encompassed within this scope includes differential expression analysis (DEA) for initial biomarker screening, strategies for integrating cell-line and patient expression data based on concordance, the application of data augmentation techniques (specifically SMOTE) to address class imbalance in the patient cohort, rigorous model evaluation using stratified cross-validation and appropriate performance metrics (notably Recall and MCC), and model interpretation via SHAP value analysis. Furthermore, the development of the pyHaiCS Python library constitutes a significant component of this project's scope. This includes the implementation of core HMC and GHMC samplers, various numerical integrators (Verlet/Leapfrog, MSSIs), the adaptive s-AIA tuning algorithm, and essential MCMC diagnostic tools, all leveraging the JAX framework for computational efficiency and designed for integration within the broader scientific Python ecosystem. The provision of benchmark examples and documentation for pyHaiCS is also considered within the project's intentions.

Despite the *openness* of this study, several limitations necessarily constrain the reach and interpretation of this work. Firstly, the study relies exclusively on publicly available RNA-seq datasets (MCF7 lines and TCGA patients) and does not involve the generation of novel experimental data. The effective sample size of the patient cohort, particularly after filtering for specific clinical criteria, remains relatively modest, which may impact the statistical robustness and generalizability of our findings. The inference of tamoxifen resistance from clinical

outcome data in TCGA, while a common practice, introduces a layer of potential ambiguity compared to direct experimental measures of resistance. Furthermore, this research is confined to genomic data (RNA-seq); the integration of other *-omics* data types (e.g., proteomics, epigenomics), which could offer a more comprehensive biological picture, falls outside the current scope. Similarly, while basic clinical information informs patient selection and outcome definition, the sophisticated integration of diverse clinical variables (e.g., tumor stage, grade, patient comorbidities) into the predictive models is not undertaken here.

From a biological perspective, the focus on tamoxifen resistance in ER^+ breast cancer means that the findings, particularly the identified biomarkers, may not be directly applicable to other endocrine therapies or different breast cancer subtypes (e.g., HER2). Crucially, while the study identifies potential biomarkers through statistical association and model interpretation (SHAP), it does not perform experimental functional validation to confirm causal roles in resistance. Moreover, the exploration of biological pathways is preliminary and constrained by the signature size and available annotation databases. Furthermore, the assumption that resistance mechanisms observed in the MCF7 cell line directly mirror those in patient tumors represents a potential simplification.

Regarding the modeling and software aspects, while a range of models is evaluated, the primary emphasis remains on the specified Bayesian approaches, and an exhaustive comparison of all conceivable Machine Learning algorithms is not feasible, nor is a large-scale sweep of hyperparameter tuning for each model. Likewise, the method for incorporating cell-line information as priors in the BLR models is rooted in a biologically-motivated intuition provided by previous research on the project, yet may not be the most optimal approach. Furthermore, the focus on HMC and its variants (GHMC, s-AIA) is deliberate, but other advanced sampling methods (e.g., variational inference, sequential Monte Carlo) are not covered in detail. The pyHaiCS library, while functional for the core methods used (HMC, GHMC, s-AIA), is presented as an ongoing development; certain advanced HMC variants (like MMHMC) or exhaustive optimizations and comparisons against all existing MCMC platforms were not included in the scope of its development within this thesis timeframe. The interpretability analysis, while informative, was primarily demonstrated on a selected model due to practical constraints, and a comprehensive, comparative study across all complex models developed was not conducted. Likewise, SHAP values provide a tool for interpretation at the individual feature level and exclusively for the predictions of a given model, and while they can be informative, they do not provide a complete picture of the underlying biological mechanisms involved in tamoxifen resistance. The biological interpretation of the identified biomarkers and pathways is based on existing literature and databases, and while efforts were made to ensure accuracy, the complexity of biological systems means that definitive conclusions should be drawn cautiously.

3 Fundamentals & Theoretical Background: An Elemental Initiation to Bayesian Clinical Analysis

> *"Science is built up with facts, as a house is with stones. But a collection of facts is no more a science than a heap of stones is a house."*
>
> ~ Henri Poincaré, *Science and Hypothesis* (1905) Ch. 9 [13]

In this chapter, we introduce the reader to the theoretical foundations behind Bayesian statistical modeling and Hamiltonian-based Monte Carlo methods. We begin by providing a comprehensive introduction to the basics of breast cancer research and the pivotal role of Bayesian modeling in clinical studies. We then present the Bayesian Logistic Regression (BLR) model for binary classification, and its practical limitations which motivate the need for advanced sampling schemes. Therefore, we subsequently introduce the reader to the fundamentals of Monte Carlo (MC) methods, focusing on the Random-Walk Metropolis-Hastings (RW-MH) algorithm, and how its limitations prompt the need for more advanced sampling methods. Finally, we provide an in-depth introduction to Hamiltonian-based methods, focusing on the Hamiltonian Monte Carlo (HMC) algorithm (and its subsequent extensions), methods for numerical integration in the context of Hamiltonian dynamics, and the development of adaptive schemes for deriving the optimal parameters of these integration methods. Lastly, we conclude this chapter by providing a brief overview of the most relevant literature and related work on the topic.

3.1 A *Brief* Introduction to Breast Cancer Research

In this section, we provide readers with a quick and accessible introduction to the field of breast cancer research. In simple terms, a tumor is nothing more than an abnormal mass of tissue that forms when cells grow (and divide) under unrestricted proliferation [14]. Tumors may be benign (i.e., not representing a major health issue) or malignant: i.e., those that grow large and infiltrate into nearby tissues. The collection of diseases originated by these malignant tumors are known as *cancers* [15–17] and represent one of the major causes of death worldwide across all income levels [18, 19]. Despite there being over 100 different types of cancerous diseases — categorized based on the affected tissue or organ of the human body [17] — we hereby restrict ourselves to discussing breast cancer in this work; the second leading cause of cancerous death, closely following lung cancer [20].

With nearly 2.3 million new diagnosed cases in the year 2020 — and nearly 3.9 million projected new cases to be reached by 2030 [21] — breast cancer represents around 12% of all cancer diagnoses (~ 25% in the case of women exclusively), and accounts for almost 15% of all female cancer-related deaths, making it the most prevalent cancer among women [2]. Despite common beliefs, breast cancer is actually a spectrum of diseases that

present distinct biological characteristics, and as a consequence, require unique treatments [22]. This heterogeneity is thus a crucial aspect of breast cancer and a correct identification of a tumor within the landscape of breast cancer is essential to provide an accurate assessment of the disease. To put it simply, in the human breast, three major groups of cells coexist: **(1)** *luminal* cells, which can be found in the lobules and whose function is to produce milk; **(2)** *basal* cells, responsible for pushing the milk into the ductal tubes by muscular contraption, and **(3)** all the connective tissue holding everything in place, which is mainly comprised of fibrous and fatty tissue [23, 24].

Histological Classification: Depending on where the tumor develops, breast cancers are classified in the first place into either *lobular* or *ductal*, with the latter representing over 80% of the diagnosed cases. This *histological* classification can be subsequently divided into lobular or ductal carcinoma *in situ* (LCIS or DCIS), and the *invasive* lobular or ductal carcinomas (ILC and IDC). The *invasive* type constitutes a much higher risk due to its ability to infiltrate other tissues [25].

Molecular Classification: Aside from the original location of the tumors, another classification of breast cancer exists based on the presence (or absence) of specific *receptors* on the cancer cells such as the *estrogen receptor* (ER), *progesterone receptor* (PR) or the *human epidermal growth factor receptor 2* (HER2). An abundance of any of these indicators is usually pointed out by referring to it as positive (+), while its absence or low presence in the tumor is indicated by stating the receptor is negative (−). From this, the most widely accepted classification of breast cancer subtypes consists of the following four major molecular subtypes [9]:

- **Luminal A** (ER^+, PR^+, $HER2^-$) is the most common breast cancer subtype. It is characterized by an overexpression of hormone receptors (ER and PR) and an absence of HER2. It has the most favorable prognosis due to its small proliferation and aggressiveness compared to other subtypes. It can be treated with hormone-targeted therapies, such as Tamoxifen, which we will discuss throughout this work.

- **Luminal B** (either ER/PR^+, $HER2^+$) is a far less common luminal subtype that also shows expression of ER and PR, but in lower quantities than for Luminal A. This is paired with a high expression of HER2 incurring in higher proliferation rates. These tumors usually have an intermediate prognosis and accompany the hormone targeted therapies with HER2-targeted therapies.

- **HER2-enriched** (ER^-, PR^-, $HER2^+$) is less common than the two previous cancer types and is characterized by not having any hormone receptor expression. HER2-enriched tumors are treated with specific therapies like trastuzumab (Herceptin) and other HER2-targeted drugs (e.g., Lapatinib).

- **Basal** (ER^-, PR^-, $HER2^-$). Also known as **triple-negative** breast cancer (TNBC, or TN as in Figure 3.1) due to their lack of any of the three major receptors. With an incidence ranging from 10 to 20% of cases, it has the poorest prognosis as none of the treatments available for the other subtypes can be used. Therefore, it is usually treated by a combination of surgery and chemotherapy (often used in the rest of subtypes as well, alongside the specific therapy).

Among these molecular subtypes (summarized below in Figure 3.1), luminal (ER^+) subtypes are not only the most common subtype but, in the majority of cases, offer the best treatment path for a complete recovery. They are characterized by a high abundance of estrogen receptors present in their cancerous cells, which fuel their

Molecular Subtypes	Luminal A	Luminal B		HER2+	TN
		(HER2-)	(HER2+)		
Biomarkers	ER+ PR+ HER2- Ki67low	ER+ PR- HER2- Ki67high	ER+ PR-/+ HER2+ Ki67low/high	ER- PR- HER2+ Ki67high	ER- PR- HER2- Ki67high
Frequency of Cases (%)	40–50	20–30		15–20	10–20
Histological Grade	Well Differentiated (Grade I)	Moderately Differentiated (Grade II)		Little Differentiated (Grade III)	Little Differentiated (Grade III)
Prognosis	Good	Intermediate		Poor	Poor
Response to Therapies	Endocrine	Endocrine Chemotherapy	Endocrine Chemotherapy Target Therapy	Target Therapy Chemotherapy	Chemotherapy PARP Inhibitors

ER: estrogen receptor; PR: progesterone receptor; HER2: human epidermal growth factor receptor 2.

Figure 3.1: Molecular Classification of Breast Cancer Subtypes (Table Extracted from [30])

growth [26]. Therefore, understanding the intricate relationship between estrogen and these breast cancer cells is pivotal in designing effective treatment strategies targeted to these cells specifically [4, 27–29].

The development of drugs such as *tamoxifen* — the most extensively used treatment in ER+ breast cancer — has shown to reduce recurrence by ~50% and mortality by ~30% after a standard 5-year treatment following surgery [8]. The idea behind this, as exemplified in Figure 3.2, is that tamoxifen binds to the estrogen receptors in the cancer cells, blocking the estrogen from binding to the receptors, as estrogen promotes cell division and growth in breast tissue; hence, stimulating the proliferation of cancer cells [31]. However, despite the success of tamoxifen in treating ER+ breast cancer, a significant number of patients develop resistance to the drug (estimated to be between 30-50% [9]), leading to a recurrence of the disease. This resistance is a major challenge in the treatment of breast cancer, and understanding the mechanisms behind it is crucial to developing new treatment strategies that can overcome it.

Although the exact biological mechanisms behind tamoxifen resistance are not yet fully understood, it has been suggested that key genes modulating estrogen such as ESR1 [32] or SOX2 [33], as well as pathways related to the epidermal growth factor family [34], may play a role in the development of a resistance to the drug. In addition, clinical trials have led to the discovery of prognostic signatures in breast cancer, such as the 21-gene OncotypeDX [35] and the 70-gene Mammaprint [36], which predict metastasis or recurrence in certain breast cancer subtypes. More recently, 6-gene signatures have been identified as potential biomarkers responsible for tamoxifen resistance in ER+ breast cancer [37]. However, the complexity of the biological mechanisms involved in tamoxifen resistance makes it difficult to predict which patients will develop resistance to the drug, and more research is needed to identify reliable biomarkers that can be used to predict this resistance phenomenon *a priori*: as has been stated before, a standard tamoxifen treatment takes as long as 5 years, and if resistance is present or acquired during that period of time, the patient may potentially lose precious time that could be used to try alternative treatments.

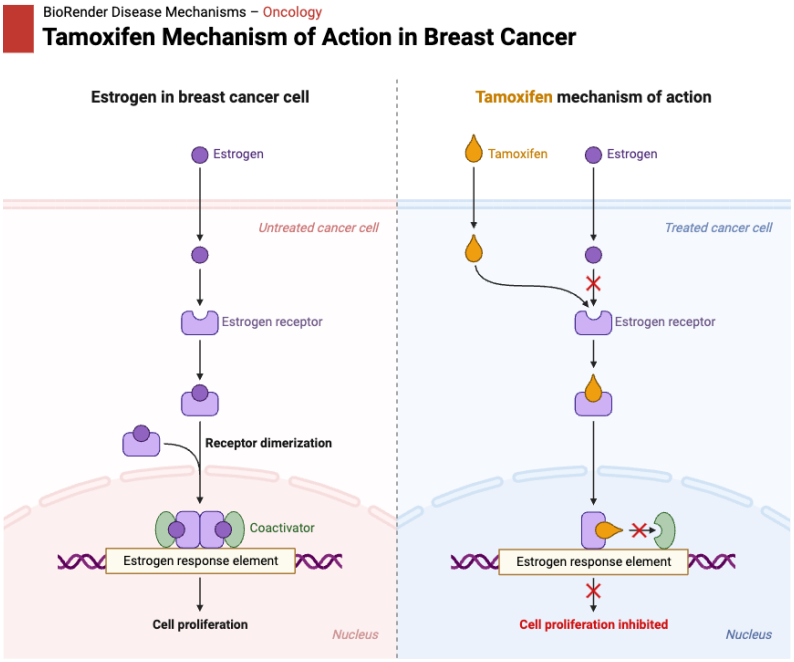

Figure 3.2: Tamoxifen Mechanism of Action in Breast Cancer Cells (Adapted from [38])

3.2 An Introduction to Bayesian Modeling & Clinical Studies

At the core of this work, the use of Bayesian models is quintessential to determining the resistance of MCF7 breast cancer cells under Tamoxifen therapy by expertly integrating informative prior knowledge from cell-lines into the patient's clinical data. This section seeks to provide the reader with a brief introduction to Bayesian Statistical Modeling (BSM) and its application in clinical studies.

In the traditional frequentist approach, the parameters θ of a statistical model $\mathcal{M}$ are considered *fixed* and unknown, whereas the data $\mathcal{D}$ is random and collected through clinical trials. Thus, inference on the model can be made by evaluating the probability $p(\mathcal{D}|\theta)$. In contrast, Bayesian statistics treats the parameters as *random variables* and the data as fixed, i.e., **data informs the model**. Thus, Bayesian inference is made by computing the posterior probability $p(\theta|\mathcal{D})$. This allows the quantification of uncertainty about the parameters of the model by specifying a prior distribution over them. The prior distribution encodes our beliefs about the parameters before observing the data, and the posterior distribution encodes our beliefs about the parameters after observing the data. The posterior distribution is obtained by updating the prior distribution using the fundamental **Bayes' Theorem** in Theorem 3.2.1 below.

Theorem 3.2.1 (Bayes' Theorem)**.** *Let θ be the parameters of a statistical model $\mathcal{M}$, and $\mathcal{D}$ be the observed data. Then, the posterior distribution of θ given $\mathcal{D}$ can be expressed as:*

$$\overbrace{p(\theta|\mathcal{D})}^{Posterior} = \frac{\overbrace{p(\theta)}^{Prior}\overbrace{p(\mathcal{D}|\theta)}^{Likelihood}}{\underbrace{p(\mathcal{D})}_{Marginal\ Likelihood}} \tag{3.1}$$

where $p(\theta)$ is the prior distribution of θ (i.e, our initial belief about the parameters), $p(\mathcal{D}|\theta)$ is the likelihood of the data given the parameters, and $p(\mathcal{D})$ is the marginal likelihood of the data.

Fundamentally,

- the *posterior* distribution $p(\theta|\mathcal{D})$ encapsulates our *updated knowledge*. It offers not only a description of the values of the model parameters, but also inherently accounts for the uncertainty associated with them.
- the *likelihood* $p(\mathcal{D}|\theta)$ represents the statistical model $\mathcal{M}$ used to describe the data $\mathcal{D}$.
- the *prior* $p(\theta)$ is used to pass onto the model our beliefs about the parameters θ and their uncertainty.
- the *marginal likelihood* $p(\mathcal{D})$ is a normalizing constant that ensures that the posterior distribution integrates to one.

In practice, given a set of N data points $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, the posterior under model $\mathcal{M}$ is given by Eq. (3.2).

$$\overbrace{p(\theta|\mathbf{x})}^{\text{Posterior}} = \frac{\overbrace{p(\theta)}^{\text{Prior}}\ \overbrace{p(\mathbf{x}|\theta)}^{\text{Likelihood}}}{\underbrace{p(\mathbf{x})}_{\text{Marginal Likelihood}}} \tag{3.2}$$

In this case, the marginal likelihood $p(\mathbf{x})$ — also known as the prior predictive distribution — guarantees that the posterior $p(\theta|\mathbf{x})$ integrates to one, and is obtained by *marginalizing* the likelihood $p(\mathbf{x}|\theta)$ over the parameters θ of the model as in Eq. (3.3).

$$p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)\, d\theta \tag{3.3}$$

Thus, the posterior is proportional to the product of the likelihood and the prior, as in Eq. (3.4).

$$p(\theta|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta) \tag{3.4}$$

Moreover, given a new observation $\tilde{\mathbf{x}}$, predictions can be made by using the *posterior predictive distribution* of the model, conditional on the posterior distribution as:

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \int p(\tilde{\mathbf{x}}, \theta|\mathbf{x})\, d\theta = \int p(\tilde{\mathbf{x}}|\theta)p(\theta|\mathbf{x})\, d\theta \tag{3.5}$$

Additionally, any expected value of a function of the parameters $f(\theta)$ over the posterior can be computed as in Eq. (3.6). As we shall see in the next section, this property — and more specifically the complexity of the integrals associated with it — is the foundational block of why we need sampling methods such as Hamiltonian Monte Carlo when dealing with Bayesian models.

$$\mathrm{E}_p\big[f(\theta)\big] = \int f(\theta)p(\theta|\mathbf{x})\, d\theta \tag{3.6}$$

Finally, the Bayesian modeling framework rests upon three key pillars:

Table 3.1: Jeffrey's Scale for Interpreting the Bayes Factor [40]

Bayes Factor $\mathscr{B}$	**Strength of Evidence (for $\mathscr{M}_1$)**
$\mathscr{B} < 1$	Negative (Supports $\mathscr{M}_2$)
$1 < \mathscr{B} < 3$	Barely Worth Mentioning
$3 < \mathscr{B} < 10$	Substantial
$10 < \mathscr{B} < 30$	Strong
$30 < \mathscr{B} < 100$	Very Strong
$\mathscr{B} > 100$	Decisive

1. **Calculation of the Parameters:** The parameters of the model can be calculated by *marginalizing* the posterior distribution. Moreover, given a subset of parameters ϑ that we are not interested in, we can marginalize over them as in Eq. (3.7).

$$p(\theta|\mathbf{x}) = \int p(\theta, \vartheta|\mathbf{x})\, d\vartheta \tag{3.7}$$

2. **Prediction of New Data:** The posterior predictive distribution can be used to make predictions about new observations by marginalizing the likelihood over the parameters θ as in Eq. (3.5).

3. **Model Selection:** The task of selecting a model $\mathscr{M}_1$ over another model $\mathscr{M}_2$ is performed by evaluating the *Bayes factor* [39, 40] in Eq. (3.8): a Bayesian alternative to *hypothesis testing* in frequentist statistics relying on the marginal likelihoods of the two models.

$$\mathscr{B} = \frac{p_{\mathscr{M}_1}(\mathbf{x})}{p_{\mathscr{M}_2}(\mathbf{x})} = \frac{\int p(\mathbf{x}|\theta_1)p(\theta_1)\, d\theta_1}{\int p(\mathbf{x}|\theta_2)p(\theta_2)\, d\theta_2} \tag{3.8}$$

   where $p_{\mathscr{M}_1}(\mathbf{x})$ and $p_{\mathscr{M}_2}(\mathbf{x})$ are the marginal likelihoods of the two models, and θ_1 and θ_2 are the parameters of the models. Based on this, [40] introduced a categorization for the values of $\mathscr{B}$ based on in terms of strength of evidence in favor of the model $\mathscr{M}_1$ over $\mathscr{M}_2$. are summarized in Table 3.1.

A common issue in these Bayesian inference tasks is the computation of high-dimensional and usually analytically intractable integrals. Thus, sampling methods for overcoming this limitation will be next introduced in Section 3.3.

Bayesian Logistic Regression – BLR

Bayesian Logistic Regression (BLR) is the probabilistic extension of the traditional *point-estimate* logistic regression model by incorporating a prior distribution over the parameters of the model. In the BLR model, given K data instances $\{\mathbf{x}_k, y_k\}_{k=1}^K$ where $\mathbf{x}_k = (1, x_1, \dots, x_D)$ are vectors of D covariates and $y_k \in \{0, 1\}$ are the binary responses, the probability of a particular outcome is linked to the linear predictor function through the *logit* function as in Eq (3.9).

$$p(y_k|\mathbf{x}_k, \theta) = \sigma(\theta^T\mathbf{x}_k) = \frac{1}{1 + \exp(-\theta^T\mathbf{x}_k)}, \quad \theta^T\mathbf{x}_k \equiv \text{logit}(p_k) = \log\left(\frac{p_k}{1 - p_k}\right) = \theta_0 + \theta_1 x_{1,k} + \dots \theta_D x_{D,k} \tag{3.9}$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1, \dots, \theta_D)^T$ are the parameters of the model, with the term θ_0 usually denoted as the *intercept*. The prior distribution over the parameters $\boldsymbol{\theta}$ is usually chosen to be a Multivariate Gaussian distribution as in Eq. (3.10).

$$\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \text{Usually } \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+1}) \tag{3.10}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{D+1}$ is the *mean* vector, $\boldsymbol{\Sigma} \in \mathbb{R}^{D+1}$ is the *covariance* matrix, $\mathbf{0}$ is the zero vector and $\mathbf{I}_{D+1}$ is the identity matrix of order $D+1$. In the upcoming sections we shall describe how information extracted from the MCF7 cell-lines was incorporated into the priors of the coefficients in order to inform our BLR model.

In order to simplify the notation, let us define the *vectorized* response variable $\mathbf{y} = (y_1, \dots, y_K)$, and the *design* matrix $X \in \mathbb{R}^{K,D}$ in Eq. (3.11) as the input to the model.

$$X = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,D} \\ 1 & x_{2,1} & \dots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{K,1} & \dots & x_{K,D} \end{pmatrix} \tag{3.11}$$

The likelihood of the data is given by the product of the Bernoulli distributions as in Eq. (3.12).

$$\mathscr{L}(\mathbf{y}|X, \boldsymbol{\theta}) \equiv p(\mathbf{y}|X, \boldsymbol{\theta}) = \prod_{k=1}^{K} p(y_k|X_k, \boldsymbol{\theta}) = \prod_{k=1}^{K} \left(\frac{\exp(X_k\boldsymbol{\theta})}{1 + \exp(X_k\boldsymbol{\theta})} \right)^{y_k} \left(\frac{1}{1 + \exp(X_k\boldsymbol{\theta})} \right)^{1-y_k} \tag{3.12}$$

where $X_k = (1, x_{k,1}, \dots, x_{k,D})$ is the k-th entry *row* vector of the design matrix X.

3.3 An Introduction to Hamiltonian-based Monte Carlo Methods

As devised in Section 3.2, the practical application of Bayesian statistical modeling techniques entails the computation of intractable integrals over marginalizations of complex probability distributions. For instance, at *inference* time, given a new observation, one can obtain predictions by computing the *posterior predictive distribution* conditioned on the posterior of $\boldsymbol{\theta}$ as in Eq. (3.13).

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \int p(\tilde{\mathbf{x}}, \boldsymbol{\theta}|\mathbf{x})\, d\boldsymbol{\theta} = \int p(\tilde{\mathbf{x}}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{x})\, d\boldsymbol{\theta} \tag{3.13}$$

Unfortunately, in most *real-life* scenarios, these complex integrals have no analytical solution, and thus numerical approximation methods are required. At the core of these numerical integration algorithms, we find Markov-Chain Monte Carlo (MCMC) methods, a specific subset of the broader Monte Carlo (MC) family.

For the uninitiated readers, the goal of Monte Carlo methods is to draw samples from a *target* distribution $\pi(\boldsymbol{\theta})$, which can in turn be used to estimate an integral I by using a *sample average estimator* $\hat{I}$ — also know as the Monte Carlo estimator — as in Eq. (3.14).

$$I = \mathrm{E}_{\pi}[f(\boldsymbol{\theta})] = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta} \xrightarrow{n\to\infty} \hat{I} = \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{\theta}_i), \quad \boldsymbol{\theta}_i \sim \pi(\boldsymbol{\theta}) \quad \text{(i.i.d samples)} \tag{3.14}$$

As such, the whole principle behind these numerical methods relies on the assumption that an integral of a function $f(\theta)$ can be expressed as an expected value over a probability distribution $\pi(\theta)$ (an example is provided in the box below). This methodology is followed by all MC methods and the differences among them emerge mainly from the approach taken to draw the samples.

***Example* – Monte Carlo Estimation of a Simple Integral**

Let us say that we want to compute the integral of $f(x) = x^2$ over the interval $[a, b]$ using the Monte Carlo estimator. From Eq. (3.14), the integral of $f(x)$ can be rewritten as an expected value over a uniform target distribution $\pi(x)$ as:

$$I' = \int f(x)\,dx = \int f(x)\frac{\pi(x)}{\pi(x)}\,dx \xrightarrow{\pi(x)\equiv U(a,b)} \widehat{I'} = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i), \quad x_i \sim \text{Uniform}(a,b)$$

as:

$$\pi(x) = \begin{cases} \dfrac{1}{b-a} & \text{for } x \in [a,b] \\ 0 & \text{elsewhere} \end{cases}$$

In this work, we focus on a subset of Markov-Chain Monte Carlo based on Hamiltonian dynamics [41–43]. However, in general terms, MCMC methods iteratively construct a Markov-Chain whose invariant distribution is the target distribution $\pi(\theta)$. Generating a large number of walks over the chain leads to its eventual convergence to the target distribution. Moreover, the *Markov property* in Eq. (3.15) guaranties that transitions to a new state in the chain depend exclusively on the current state [44, 45]. That is:

$$P(X_{n+1} = x_{n+1} | X_n = x_n, \ldots, X_0 = x_0) = P(X_{n+1} = x_{n+1} | X_n = x_n) \tag{3.15}$$

In Algorithm 1, we present the most basic form of Markov-Chain Monte Carlo sampling: a Random-Walk Metropolis-Hastings (RW-MH) sampler [46, 47]. Every iteration of the RW-MH algorithm consists of two major steps:

1. A move in the parameter space is generated from the *proposal* distribution $q(\theta' | \theta)$
2. An acceptance-rejection test — also know as the Metropolis-Hastings test (e.g., also used in the Simulated Annealing algorithm for combinatorial optimization [48, 49]) — is used to determine if the proposed move in the Markov-Chain should be accepted. For that purpose, an acceptance probability α — which depends on the current and previous states — is calculated.

However, because of the Random-Walk behavior of the RW-MH method, the parameter space is poorly explored, and thus the Markov-Chain *slowly* converges towards the desired target distribution. Additionally, as the number of dimensions in the problem increases (as is usually the case in *real-world* problems) this issue only becomes worse.

In order to circumvent these obstacles, Hamiltonian-based Markov-Chain Monte Carlo methods severely improve upon the performance of the chain convergence by applying deterministic proposals from *molecular dynamics* (MD). In essence, new states are proposed by computing trajectories according to Hamiltonian dynamics by using numerical integration methods (such as the Verlet/Leapfrog integrator from Algorithm 3).

Algorithm 1 Random-Walk Metropolis-Hastings (RW-MH)

Input:
  N: Number of Monte Carlo Samples
  $q(\theta'|\theta)$: Proposal Distribution
  $\pi(\theta)$: Target Distribution
Output:
  $\{\theta_n\}_{n=1}^{N}$: Samples from the Target Distribution
Initialize θ_0
for $n = 1$ to N **do**
  $\theta = \theta_{n-1}$
  Sample Proposal $\theta' \sim q(\theta'|\theta)$
  Compute Acceptance Probability: $\alpha = \min\left\{1, \frac{\pi(\theta')q(\theta|\theta')}{\pi(\theta)q(\theta'|\theta)}\right\}$
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < \alpha$ **then**
    $\theta_n = \theta'$ // *Update Accepted*
  else
    $\theta_n = \theta$ // *Update Rejected*
  end if
end for

This bypasses the slow exploration of the state space that occurs when Metropolis updates are done using a simple random-walk proposal distribution. As the simplest example of such methods, Hamiltonian Monte Carlo (HMC) [41, 42, 50–52], produces a chain whose invariant distribution is an augmented target distribution $\pi(\theta, \mathbf{p})$ related to the Hamiltonian function $H(\theta, \mathbf{p})$ as in Eq. (3.16).

$$\pi(\theta, \mathbf{p}) = \pi(\theta)p(\mathbf{p}) \propto \exp\big(-H(\theta, \mathbf{p})\big) \tag{3.16}$$

where $\mathbf{p}$ is an auxiliary momentum variable[1] that is usually drawn from a Gaussian distribution $\mathbf{p} \sim \mathcal{N}(0, M)$.

The Hamiltonian $H(\theta, \mathbf{p})$ is a mathematical tool stemming from physics, where a system of D particles is described by its energy and time evolution in t. The system is represented by two generalized canonical coordinates, one for the *position* $\theta = (\theta_1, \dots, \theta_D)$ and one for the *momentum* $\mathbf{p} = (\mathbf{p}_1, \dots, \mathbf{p}_D)$. In the HMC method, a separable Hamiltonian with two terms that are independent of each other is considered as in Eq. (3.17).

$$H(\theta, \mathbf{p}) = K(\mathbf{p}) + U(\theta) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\theta) \tag{3.17}$$

where $U(\theta)$ and $K(\mathbf{p})$ denote the potential and kinetic energy functions, respectively. The *kinetic* energy is defined using the auxiliary momentum variables $\mathbf{p}$ and a mass matrix M, which is a symmetric positive definite as in Eq. (3.18).

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} \tag{3.18}$$

[1]Note that, in the case of HMC, these momentum variables are discarded after every iteration. However, for GHMC and MMHMC, they are updated and refined across iterations.

Meanwhile, the *potential* energy term is related to the target distribution as in Eq. (3.19).

$$U(\boldsymbol{\theta}) = -\log \pi(\boldsymbol{\theta}) + \text{const.} \tag{3.19}$$

Finally, the dynamics of the particle system are described by the *Hamiltonian equations of motion* in Eq. (3.20): a set of ordinary differential equations for the generalized canonical coordinates. In practice, a proposal for the new state $(\boldsymbol{\theta}', \mathbf{p}')$ is generated by integrating these Hamiltonian dynamics using a numerical integrator $\Psi_{\varepsilon,L}$ — such as the Verlet/Leapfrog integrator in Algorithm 3 — for L steps (i.e., the trajectory length) and with a step-size ε. The complete HMC algorithm is summarized in Algorithm 2.

$$\dot{\boldsymbol{\theta}} = H_{\mathbf{p}}(\boldsymbol{\theta}, \mathbf{p}; t) = M^{-1}\mathbf{p} \qquad \dot{\mathbf{p}} = -H_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{p}; t) = -U_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \tag{3.20}$$

Moreover, by defining $\mathbf{z} = (\boldsymbol{\theta}, \mathbf{p})$ we can rewrite the Hamiltonian dynamics in Eq. (3.20) in matrix form as in Eq. (3.21).

$$\dot{\mathbf{z}} = \mathbf{J}H_{\mathbf{z}}(\mathbf{z}) \qquad \mathbf{J} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \tag{3.21}$$

where I is the $D \times D$ *identity* matrix. From this alternative formulation, several key properties of the Hamiltonian can be derived [50, 53–55]:

1. **Conservation of the Hamiltonian:** That is, any fluctuation in the potential $U(\boldsymbol{\theta})$ of the system, must be balanced by a change in the Kinetic energy $K(\mathbf{p})$.

$$\dot{H}(\boldsymbol{\theta}, \mathbf{p}) = \nabla H(\boldsymbol{\theta}, \mathbf{p})^T J^{-1} \nabla H(\boldsymbol{\theta}, \mathbf{p}) = 0 \tag{3.22}$$

Theorem 3.3.1 (Conservation of the Hamiltonian)**.** *Let $H(\boldsymbol{\theta}, \mathbf{p}, t)$ be the Hamiltonian of a system, where $\boldsymbol{\theta}$ and $\mathbf{p}$ are the generalized coordinates and momenta, respectively. If the Hamiltonian does not explicitly depend on time, i.e., $\dot{H}(\boldsymbol{\theta}, \mathbf{p}) = 0$, then the Hamiltonian is **conserved**. Formally, this can be stated as:*

$$\dot{H}(\boldsymbol{\theta}, \mathbf{p}, t) \equiv \frac{dH}{dt} = \frac{\partial H}{\partial \boldsymbol{\theta}}\dot{\boldsymbol{\theta}} + \frac{\partial H}{\partial \mathbf{p}}\dot{\mathbf{p}} + \frac{\partial H}{\partial t} = 0 \tag{3.23}$$

where $\dot{\boldsymbol{\theta}}$ and $\dot{\mathbf{p}}$ are the time derivatives of $\boldsymbol{\theta}$ and $\mathbf{p}$, respectively.

2. **Conservation of the Volume:** The differential volume element $d\mathbf{z}$ is conserved. That is, $\nabla \cdot \dot{\mathbf{z}} = 0$.

Theorem 3.3.2 (Conservation of Volume under the Hamiltonian, Liouville's Theorem)**.** *Let $H(\boldsymbol{\theta}, \mathbf{p}, t)$ be the Hamiltonian of a system with generalized coordinates $\boldsymbol{\theta}$ and momenta $\mathbf{p}$. Then, the phase-space distribution function $\rho(\boldsymbol{\theta}, \mathbf{p}, t)$, i.e., the density of states in the phase space, satisfies Liouville's equation:*

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \dot{\mathbf{z}}) = \frac{\partial \rho}{\partial t} + \sum_{i=1}^{D} \left(\frac{\partial \rho}{\partial \theta_i}\dot{\theta}_i + \frac{\partial \rho}{\partial p_i}\dot{p}_i \right) = 0 \tag{3.24}$$

where $\dot{\mathbf{z}} = (\dot{\boldsymbol{\theta}}, \dot{\mathbf{p}})$ *is the phase-space velocity vector. In Hamiltonian mechanics, this reduces to:*

$$\nabla \cdot \dot{\mathbf{z}} = \frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} + \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{p}} = 0 \tag{3.25}$$

Thus, the Hamiltonian $H(\boldsymbol{\theta}, \mathbf{p}, t)$ *is conserved along the trajectories of the system in the phase-space or, alternatively, the phase-space volume is conserved over time.*

Proof. Let $\dot{\mathbf{z}} = (\dot{\boldsymbol{\theta}}, \dot{\mathbf{p}})$ be the phase-space velocity vector of our Hamiltonian. Then, the divergence of $\dot{\mathbf{z}}$ can be expressed as:

$$\begin{aligned}
\nabla \cdot \dot{\mathbf{z}} = \nabla^T \dot{\mathbf{z}} &= \left(\frac{\partial}{\partial \boldsymbol{\theta}}, \frac{\partial}{\partial \mathbf{p}} \right) \cdot \left(\dot{\boldsymbol{\theta}}, \dot{\mathbf{p}} \right)^T \\
&= \frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} + \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{p}} \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} (M^{-1}\mathbf{p}) + \frac{\partial}{\partial \mathbf{p}} (-U_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \quad \text{(from Eq. (3.20))} \\
&= 0 \quad \text{(Separable Hamiltoninan as in Eq. (3.17))}
\end{aligned} \tag{3.26}$$

■

3. **Reversibility of the Hamiltonian Flow:** For a Hamiltonian flow Φ_t: $\Phi_t(\mathbf{z})^T \mathbf{J}^{-1} \Phi_t(\mathbf{z}) = \mathbf{J}^{-1}, \forall \mathbf{z} \in \Omega$ and a mapping $\mathcal{F}(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}, -\mathbf{p})$ in the phase-space Ω, then:

$$\Phi_{-t} = \mathcal{F} \circ \Phi_t \circ \mathcal{F} \tag{3.27}$$

where $\circ$ is the composition operator. Thus, a backwards evolution in the Hamiltonian chain is equivalent to flipping the initial momenta, evolving in time, and flipping the final momenta. This is an essential condition for the Markov-Chain to converge to an invariant target distribution [50, 53, 55].

In practice, in all Hamiltonian-based samplers, we devise a first burn-in/warm-up stage where the algorithm evolves the chain until it has converged to the intended target distribution. Thus, the samples taken during this phase are discarded. After the burn-in stage is finished and convergence has been reached, samples are collected at the production stage (i.e., the N samples mentioned throughout this section).

For reference, Algorithm 3 summarizes the *Verlet* or *Leapfrog* integrator [50, 56], the most popular and most *intuitive* integration scheme for Hamiltonian dynamics (see Eq. (3.28)). However, we believe that the study of these numerical integrators is well outside the scope of this thesis, as we wish to exclusively provide the most essential theoretical background to understand this work. Nevertheless, please note that there exists a vast area of research delving into more robust and efficient integration methods for improving HMC sampling [50, 51, 55, 57–63].

$$\begin{aligned}
\mathbf{p}_{\frac{\varepsilon}{2}} &= \mathbf{p}_0 - \frac{\varepsilon}{2} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}_0) \\
\boldsymbol{\theta}_\varepsilon &= \boldsymbol{\theta}_0 + \varepsilon M^{-1} \mathbf{p}_{\frac{\varepsilon}{2}} \\
\mathbf{p}_\varepsilon &= \mathbf{p}_{\frac{\varepsilon}{2}} - \frac{\varepsilon}{2} U_{\boldsymbol{\theta}}(\boldsymbol{\theta}_\varepsilon)
\end{aligned} \tag{3.28}$$

Algorithm 2 Hamiltonian Monte Carlo (HMC)

Input:
  N: Number of Monte Carlo Samples
  ε: Step-Size
  L: Number of Integration Steps (Trajectory Length)
  M: Mass Matrix
  $\Psi_{\varepsilon,L}$: Hamiltonian Numerical Integrator
Output:
  $\{\theta_n\}_{n=1}^{N}$: Samples from the Target Distribution
Initialize θ_0
for $n = 1$ to N **do**
  $\theta = \theta_{n-1}$
  Sample Momentum $\mathbf{p} \sim \mathcal{N}(0, M)$
  Integrate Hamiltonian Dynamics to Generate Update Proposal: $(\theta', \mathbf{p}') = \Psi_{\varepsilon,L}(\theta, \mathbf{p})$
  Compute $\Delta H = H(\theta', \mathbf{p}') - H(\theta, \mathbf{p})$
  // Metropolis-Hastings Test:
  Compute Acceptance Probability: $\alpha = \min\{1, \exp(-\Delta H)\}$
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < \alpha$ **then**
    $\theta_n = \theta'$ *// Update Accepted*
  else
    $\theta_n = \theta$ *// Update Rejected*
  end if
  Discard Momentum $\mathbf{p}'$
end for

Although at this point the reader should be more than familiar enough with the intricacies of Hamiltonian sampling, let us provide some additional interesting properties of these samplers. In HMC, the momentum variables $\mathbf{p}$ are discarded at the end of every iteration, i.e., potentially, valuable information between steps in the chain may be lost. In practice, one may lose track of a good exploration by resorting to this full momentum update at each iteration, slowing down convergence. One solution to this problem — adopted by the Generalized Hamiltonian Monte Carlo (GHMC) [50, 51, 64] method in Algorithm 4 — is to replace the momentum discards with a *partial momentum update* (PMU), i.e., in every iteration, a new momentum update $\mathbf{p}^*$ is proposed. Instead of drawing a new momentum $\mathbf{p}$ in every iteration, we draw an additional *noise vector* $\boldsymbol{\mu} \sim \mathcal{N}(0, M)$ as compute the proposed updates as in Eq. (3.29).

$$\begin{aligned} \mathbf{p}^* &= \sqrt{1-\phi}\mathbf{p} + \sqrt{\phi}\boldsymbol{\mu} \\ \boldsymbol{\mu}^* &= -\sqrt{\phi}\mathbf{p} + \sqrt{1-\phi}\boldsymbol{\mu} \end{aligned} \tag{3.29}$$

where the role of $\phi \in (0, 1]$ is to control the extent to which the momentum can deviate from its current direction. Conversely, as the momentum is not discarded completely from one iteration to the next, the Metropolis-Hastings test in GHMC includes a momentum flip upon rejection $\mathcal{F}(\theta, \mathbf{p}) = (\theta, -\mathbf{p})$. By applying these two modifications, the resulting Markov-Chain is said to be *irreversible*, which accelerates convergence and reduces

Algorithm 3 Verlet/Leapfrog Numerical Integrator

1: **Input:**
   $(\theta, \mathbf{p})$: Initial Position & Momentum
   ε: Step-Size
   L: Number of Integration Steps
2: **Output:**
   $(\theta', \mathbf{p}')$: Updated Position & Momentum
3: *Half-Step Update for Momentum:* $\mathbf{p} = \mathbf{p} - \frac{\varepsilon}{2} U_\theta(\theta)$
4: **for** $i = 1$ to $L - 1$ **do**
5: *Full-Step Update for Position:* $\theta = \theta + \varepsilon M^{-1}\mathbf{p}$
6: *Full-Step Update for Momentum:* $\mathbf{p} = \mathbf{p} - \varepsilon U_\theta(\theta)$
7: **end for**
8: *Full-Step Update for Position:* $\theta' = \theta + \varepsilon M^{-1}\mathbf{p}$
9: *Half-Step Update for Momentum:* $\mathbf{p}' = \mathbf{p} - \frac{\varepsilon}{2} U_\theta(\theta)$

variance in the samples [65]. If $\phi = 0$, then one long trajectory is produced without the partial momentum update. This is called the *Molecular Dynamics Monte Carlo* (MDMC) method.

Furthermore, by building upon the improvements of GHMC, we can extend the sampler by adding *Importance Sampling* (IS) [66]. As a quick overview, importance sampling is a subtype of traditional Monte Carlo (i.e., non-Markovian) methods where instead of sampling directly from the desired *target* distribution $\pi(\theta)$, samples are taken from an alternative distribution, often called the *importance* distribution $q(\theta)$. Thus, the expected value problem in Eq. (3.14) can be rewritten in terms of these distributions as in Eq. (3.30).

$$\begin{aligned} I &= \mathrm{E}_\pi\big[f(\theta)\big] = \int f(\theta)\pi(\theta)\, d\theta = \int f(\theta)\frac{\pi(\theta)}{q(\theta)} q(\theta)\, d\theta \\ I &\xrightarrow{n\to\infty} \hat{I} = \frac{\sum_{i=1}^{N} \omega_i f(\theta_i)}{\sum_{i=1}^{N} \omega_i}, \quad \theta_i \sim q(\theta), \quad \omega_i = \frac{\pi(\theta_i)}{q(\theta_i)} \end{aligned} \tag{3.30}$$

Following this approach, the Mix & Match Hamiltonian Monte Carlo (MMHMC) [50, 51] method in Algorithm 5 was devised. Without getting into excessive detail, a *k-th order truncated modified Hamiltonian* is introduced as in Eq. (3.31) following an asymptotic expansion in powers of the integration step-size. As a result, the usage of *modified* Hamiltonians entails a reduction in the expected energy error in the numerical integration [50, 67–69].

$$\tilde{H}^{[k]}(\theta, \mathbf{p}) = H(\theta, \mathbf{p}) + \varepsilon^p H_{p+1}(\theta, \mathbf{p}) + \cdots + \varepsilon^{k-1} H_k(\theta, \mathbf{p}) \tag{3.31}$$

Thus, in the case of MMHMC, sampling is done with respect to a modified distribution $\tilde{\pi}(\theta, \mathbf{p})$ as in Eq. (3.32).

$$\tilde{\pi}(\theta, \mathbf{p}) \propto \exp\Big(-\tilde{H}^{[k]}(\theta, \mathbf{p})\Big) \tag{3.32}$$

From this, the original distribution can be recovered by the importance weights in Eq. (3.33).

$$\omega_i = \exp\Big(-\big[H(\theta_i, \mathbf{p}_i) - \tilde{H}^{[k]}(\theta_i, \mathbf{p}_i)\big]\Big) \tag{3.33}$$

Algorithm 4 Generalized Hamiltonian Monte Carlo (GHMC)

Input:
- N: Number of Monte Carlo Samples
- ε: Step-Size
- L: Number of Integration Steps (Trajectory Length)
- M: Mass Matrix
- ϕ: Momentum Noise Parameter $\left(\phi \in (0, 1]\right)$
- $\Psi_{\varepsilon,L}$: Hamiltonian Numerical Integrator

Output:
- $\{\boldsymbol{\theta}_n\}_{n=1}^{N}$: Samples from the Target Distribution

Initialize $(\boldsymbol{\theta}_0, \mathbf{p}_0)$
for $n = 1$ to N **do**
  $(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}_{n-1}, \mathbf{p}_{n-1})$
  // Partial Momentum Update:
  Sample $\boldsymbol{\mu} \sim \mathcal{N}(0, M)$
  Proposed Updated Momentum: $\mathbf{p}^* = \sqrt{1-\phi}\mathbf{p} + \sqrt{\phi}\boldsymbol{\mu}$
  Proposed Noise Vector: $\boldsymbol{\mu}^* = -\sqrt{\phi}\mathbf{p} + \sqrt{1-\phi}\boldsymbol{\mu}$
  Integrate Hamiltonian Dynamics to Generate Update Proposal: $(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{\varepsilon,L}(\boldsymbol{\theta}, \mathbf{p}^*)$
  Compute $\Delta H = H(\boldsymbol{\theta}', \mathbf{p}') - H(\boldsymbol{\theta}, \mathbf{p}^*)$
  // Metropolis-Hastings Test:
  Compute Acceptance Probability: $\alpha = \min\{1, \exp(-\Delta H)\}$
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < \alpha$ **then**
    $(\boldsymbol{\theta}_n, \mathbf{p}_n) = (\boldsymbol{\theta}', \mathbf{p}')$ *// Update Accepted*
  else
    // Momentum Flip:
    $(\boldsymbol{\theta}_n, \mathbf{p}_n) = (\boldsymbol{\theta}, -\mathbf{p}^*)$ *// Update Rejected*
  end if
end for

Furthermore, due to the introduction of the modified Hamiltonian, the partial momentum update must be modified as well by using an *extended* Hamiltonian as in Eq. (3.34).

$$\hat{H}(\boldsymbol{\theta}, \mathbf{p}, \boldsymbol{\mu}) = \tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p}) + K(\boldsymbol{\mu}) = \tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p}) + \frac{1}{2}\boldsymbol{\mu}^T M^{-1}\boldsymbol{\mu} \tag{3.34}$$

Finally, for a much more extensive and in-detail discussion on MMHMC we refer the reader to [50, 51], and provide in Table 3.2 a comprehensive summary of the relevant properties of Hamiltonian-based samplers against those of *vanilla* Random-Walk Metropolis-Hastings. In general terms, the reader should keep in mind that the performance of these methods is essentially determined by the following factors:

- **Choice of Sampler:** As we have seen, Hamiltonian-based samplers are more efficient than Random-Walk Metropolis-Hastings due to their ability to explore the target distribution more effectively by simulating the Hamiltonian dynamics of a system to model the Markov-Chain. Within this family of methods, each algorithm builds upon the previous one by introducing additional modifications such as partial momentum updates, modified Hamiltonians, or importance sampling re-weighting. From HMC, where the auxiliary momenta are discarded at each iteration; GHMC, where a partial momentum update is pro-

Algorithm 5 Mix & Match Hamiltonian Monte Carlo (MMHMC)

Input:

N: Number of Monte Carlo Samples
ε: Step-Size
L: Number of Integration Steps (Trajectory Length)
M: Mass Matrix
ϕ: Momentum Noise Parameter $\left(\phi \in (0, 1]\right)$
$\Psi_{\varepsilon,L}$: Hamiltonian Numerical Integrator

Output:

$\{\boldsymbol{\theta}_n\}_{n=1}^{N}$: Samples from the Target Distribution

Initialize $(\boldsymbol{\theta}_0, \mathbf{p}_0)$
for $n = 1$ to N **do**
  $(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}_{n-1}, \mathbf{p}_{n-1})$
  // Partial Momentum Update:
  Sample $\boldsymbol{\mu} \sim \mathcal{N}(0, M)$
  Proposed Updated Momentum: $\mathbf{p}^* = \sqrt{1-\phi}\mathbf{p} + \sqrt{\phi}\boldsymbol{\mu}$
  Proposed Noise Vector: $\boldsymbol{\mu}^* = -\sqrt{\phi}\mathbf{p} + \sqrt{1-\phi}\boldsymbol{\mu}$
  Compute $\Delta\hat{H} = \hat{H}(\boldsymbol{\theta}, \mathbf{p}^*, \boldsymbol{\mu}^*) - \hat{H}(\boldsymbol{\theta}, \mathbf{p}, \boldsymbol{\mu})$
  // Metropolis-Hastings Test (for Momentum Acceptance):
  Compute Acceptance Probability: $P = \min\{1, \exp\left(-\Delta\hat{H}\right)\}$
  Sample $v \sim \text{Uniform}(0, 1)$
  if $v < P$ **then**
    $\mathbf{p}^\dagger = \mathbf{p}^*$ *// Update Accepted*
  else
    $\mathbf{p}^\dagger = \mathbf{p}$ *// Update Rejected*
  end if
  Integrate Hamiltonian Dynamics to Generate Update Proposal: $(\boldsymbol{\theta}', \mathbf{p}') = \Psi_{\varepsilon,L}(\boldsymbol{\theta}, \mathbf{p}^\dagger)$
  Compute $\Delta\tilde{H}^{[k]} = \tilde{H}^{[k]}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p}^\dagger)$
  // Metropolis-Hastings Test:
  Compute Acceptance Probability: $\alpha = \min\{1, \exp\left(-\Delta\tilde{H}^{[k]}\right)\}$
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < \alpha$ **then**
    $(\boldsymbol{\theta}_n, \mathbf{p}_n) = (\boldsymbol{\theta}', \mathbf{p}')$ *// Update Accepted*
  else
    $(\boldsymbol{\theta}_n, \mathbf{p}_n) = (\boldsymbol{\theta}, -\mathbf{p}^\dagger)$ *// Update Rejected, Momentum Flip*
  end if
  Compute $\Delta H = H(\boldsymbol{\theta}_n, \mathbf{p}_n) - \tilde{H}^{[k]}(\boldsymbol{\theta}_n, \mathbf{p}_n)$
  Compute Weight: $\omega_n = \exp(-\Delta H)$
end for
Final Monte Carlo Estimator (w/ Importance Sampling Weights): $\hat{I}_N = \dfrac{\sum_{i=1}^{N} \omega_i f(\boldsymbol{\theta}_i)}{\sum_{i=1}^{N} \omega_i}$

posed; and finally to MMHMC, where importance sampling and modified Hamiltonians are introduced, the performance of the sampler is improved by gradually adding these modifications.

Please note that, in this thesis, some additional methods such as the Targeted Shadow Hamiltonian Monte Carlo (TSHMC) [70] or the Generalized Shadow Hamiltonian Monte Carlo (GSHMC) [71] have been left out of this discussion for the sake of brevity and because they represent an intermediate step between GHMC and MMHMC, where GHMC is simply extended by using the k-th order modified Hamiltonians $\tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p})$.

- **Choice of Integrator:** Despite only having presented the Verlet or Leapfrog integrator in this work, there exists a vast area of research on the development of more efficient and robust numerical integrators for Hamiltonian dynamics. The choice of integrator is crucial for the performance of the sampler, as it directly affects the accuracy of the numerical approximation of the Hamiltonian dynamics. For example, the choice of *symplectic* integrators is essential for ensuring the conservation of the Hamiltonian and the volume in the phase-space as stated in Theorems 3.3.1 and 3.3.2. Likewise, extensive work has been carried in the design of *mulit-stage* integrators (which will be introduced in a latter section), or numerical integrators that can be used under modified (or *shadow*) Hamiltonians [50, 57, 58].

- **Choice of Parameters:** The choice of the step-size ε, the trajectory length L, and the momentum noise parameter ϕ are all crucial for the performance of the sampler. If ε is too small, the sampler will take too long to explore the target distribution, while if it is too large, it may lead to integration inaccuracies. Similarly, the choice of L is crucial for the accuracy and sampling efficiency of the numerical integration of the Hamiltonian dynamics. The momentum noise parameter ϕ controls the extent to which the momentum can deviate from its current direction, thus values that are too large increase momenta rejection rates, while values that are too small may slow down convergence. Additionally, the order of truncation k in the modified Hamiltonians is also crucial: higher-order truncations lead to more accurate numerical approximations of the Hamiltonian, but also increase the computational cost of the sampler.

  As a way to counteract the potential issues with the choice of these parameters, one can resort to *adaptive* methods (more on this later) such as the No-U-Turn Sampler (NUTS) [72–74] that automatically tune the step-size ε and the trajectory length L during the burn-in phase of the sampler. Likewise, alternative *ad-hoc* schemes can be devised by applying combinatorial optimization techniques (e.g., Simulated Annealing, Genetic Algorithms, Swarm Intelligence methods, etc.) to finding the optimal configuration of these parameters. However, the computational cost of these methods would be significant considering the complexity of the problem.

- **Choice of Initial Configuration:** Finally, the reader might have noticed that we have not yet commented on the choice of the initial configuration $(\boldsymbol{\theta}_0, \mathbf{p}_0)$ in the samplers. This is because, as of the time of this writing, not much work has been done on the optimization of the initial configuration for Hamiltonian-based samplers. However, it is known that the choice of the initial configuration can have a significant impact on the performance of the sampler, as it can affect the *convergence rate* and the quality of the samples. Thus, the usual practical approach to solving this potential issue is to sample several *chains* $(\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_m)$ in parallel, starting from different initial configurations, and then combine the samples to obtain the final estimate of the target distribution [51, 75].

In theory, provided that the m chains have *sufficiently* converged to the target distribution, they should reduce the error of the estimator $\hat{I}$ by a factor of $\sqrt{m}$, which would be equivalent to the error reduction we should get from sampling a single chain $\mathscr{C}$ with m-times more samples (i.e., $|\mathscr{C}| = m \cdot N$). However, when sampling from a Markov-Chain, there is a burn-in/warm-up overhead as the first b samples have to be discarded to ensure that the chain has converged, i.e., for single-chain MCMC sampling this overhead is of $b/(N+b)$, while for multi-chain sampling the resulting overhead is $(m \cdot b)/(N+b)$. In other words, single-chain sampling *should* be more performant in terms of reducing burn-in overhead, were we capable of starting the chain from an optimal configuration $(\boldsymbol{\theta}_0^*, \mathbf{p}_0^*)$. On the other hand, the shorter m chains are independent and thus can be run in parallel, which can lead to significant speedups in the sampling process.

Although this multi-chain approach is computationally expensive, it is the most common practice in the field. However, the development of more efficient methods for optimizing the initial configuration of the sampler is an open research topic that remains to be addressed.

Although the aforementioned aspects are outside the scope of this thesis, we provide below an interesting example of existing adaptive methods for multi-stage integrators [57].

***Example* – Discussion on Adaptive Tuning Methods for 2-Stage Integrators**

The use of multi-stage splitting integrators has shown improved conservation of the Hamiltonians over the traditional Verlet/Leapfrog method in Algorithm 3 [76]. However, they are typically more unstable [63] and thus greatly benefit from adaptive methods for parameter tuning and selection. For that reason, the *Adaptive Integration Approach* (AIA) [77] introduced an automatic selection process for the best Hamiltonian numerical integrator in practical scenarios. Given the two solution flows φ_t^A and φ_t^B in Eq. (3.35).

$$\varphi_t^A(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta}, \mathbf{p} - tU_{\boldsymbol{\theta}}(\boldsymbol{\theta})) \qquad \varphi_t^B(\boldsymbol{\theta}, \mathbf{p}) = (\boldsymbol{\theta} + tM^{-1}\mathbf{p}, \mathbf{p}) \tag{3.35}$$

We can define any *two-stage* splitting integrator $\Psi_\varepsilon^{(2)}$ as in Eq. (3.36) (we spare the reader the derivation of how these integrators come to be, as they serve here the sole purpose of illustrating the adaptive tuning schemes).

$$\Psi_\varepsilon^{(2)} = \varphi_{b\varepsilon}^B \circ \varphi_{\varepsilon/2}^A \circ \varphi_{(1-2b)\varepsilon}^B \circ \varphi_{\varepsilon/2}^A \circ \varphi_{b\varepsilon}^B \tag{3.36}$$

where $0 < b < 1/2$ is a parameter that specifies the individual integrator within the family, ε is the step-size, and $\circ$ is the composition operator. Then, the Modified AIA (MAIA) and Extended MAIA (e-MAIA) were proposed in [57] to tune the b, ε parameters and b, ε, ϕ respectively. These methods work under modified (truncated) Hamiltonians by minimizing the expectation of the energy error as in Eq. (3.37).

$$\min \mathrm{E}\left[\Delta\tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p})\right] \qquad \Delta\tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p}) = \tilde{H}^{[k]}(\boldsymbol{\theta}', \mathbf{p}') - \tilde{H}^{[k]}(\boldsymbol{\theta}, \mathbf{p}^\dagger) \tag{3.37}$$

Readers who wish to learn more about these adaptive methods and multi-stage integrators are encouraged to read [57, 78], although some notes on the topic will be provided by the end of the chapter.

Table 3.2: Comparison of Relevant Properties of Hamiltonian-based Samplers

	RW-MH	HMC	GHMC	MMHMC
Correlation	✓	✓	✓	✓
Irreversibility	✗	✗	✓	✓
Importance Sampling	✗	✗	✗	✓

***Summary* – Categorization of Hamiltonian-based MCMC Samplers**

The following three Hamiltonian-based sampling methods have been introduced in this work:

- **Hamiltonian Monte Carlo** (HMC)
- **Generalized Hamiltonian Monte Carlo** (GHMC) – Improves HMC performance by adding a partial momentum update across samples, i.e., the additional momenta are not discarded at the end of each integration process.
- **Mix & Match Hamiltonian Monte Carlo** (MMHMC) – Improves HMC performance by adding:
  - **Partial momentum updates** across samples (same as GHMC).
  - Usage of *truncated* **modified Hamiltonians** $\tilde{H}^{[k]}$ with better conservation under symplectic numerical integrators for improved sampling.
  - Importance sampling **re-weigthing**.

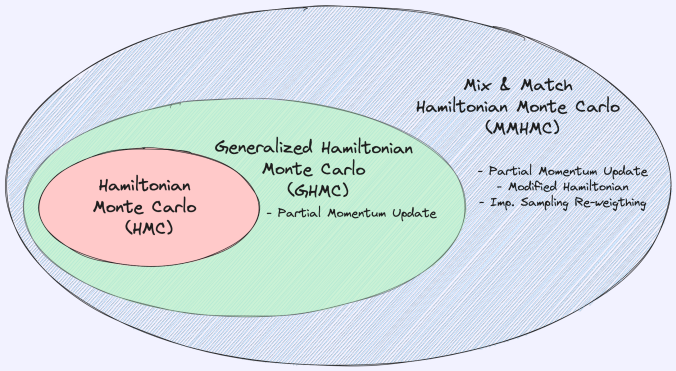

Figure 3.3: Hierarchy of Hamiltonian-based Monte Carlo Methods

Evaluating the Sampling Quality of Hamiltonian-based Methods

So far, we have introduced the Hamiltonian Monte Carlo (HMC) method for posterior sampling, as well as subsequent extensions constructed by adding complexity to enhance the quality of the sampling. However, we have not yet explored how these enhancements can be *quantified* appropriately. In this short section below, we

introduce several reference metrics [51, 57, 79] for assessing the performance of our sampling methodologies by focusing on the following criteria:

- State **space exploration** performed by the chain;
- **Sampling efficiency** — i.e., the ability of the method to produce more uncorrelated samples;
- **Convergence** to the target distribution $\pi(\boldsymbol{\theta})$.

The usual metrics for evaluating the sampling quality of our methods are presented below:

1. **Acceptance Rate (AR)**: Measures the ratio of accepted proposals in the Metropolis test. That is:

$$\mathrm{AR} = \frac{N_{\mathrm{acc}}}{N} \tag{3.38}$$

2. **Potential Scale Reduction Factor (PSRF)**: Used to assess convergence to the (stationary) target distribution of M-parallel Markov chains [80].

   First, we define the *within-chain* (W) and *between-chain* (B) variances respectively as:

$$W = \frac{1}{M}\sum_{m=1}^{M}\sigma_m^2, \qquad B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{\theta}_m - \bar{\theta}_G)^2 \tag{3.39}$$

   where M is the number of Markov-Chains drawn, N is the number of samples from the posterior distribution, σ_m^2 is the variance of chain m, and $\bar{\theta}_m, \bar{\theta}_G$ are the individual local chain mean and global mean across the chains, respectively.

   Then, the sample variance from all the chains combined is obtained as the weighted average of these two values:

$$\hat{\sigma} = \left(1 - \frac{1}{N}\right)W + \frac{B}{N} \tag{3.40}$$

   which yields:

$$\widehat{V} = \hat{\sigma} + \frac{B}{MN} \tag{3.41}$$

   Finally, the **potential scale reduction factor** $\widehat{R}$ is defined as follows:

$$\widehat{R} = \sqrt{\left(\frac{d+3}{d+1}\right)\frac{\widehat{V}}{W}} \tag{3.42}$$

   where d is the number of degrees of freedom of a t-distribution with mean $\bar{\theta}_G$ and variance $\widehat{V}$ estimated by the method of moments as:

$$d \approx \frac{2\widehat{V}^2}{\mathrm{Var}(\widehat{V})} \tag{3.43}$$

   A value of $\widehat{R} \approx 1$ indicates a good convergence, and the closer to 1 the values are, the better this convergence is. In practice, a value of at least $\widehat{R} < 1.01$ is recommended for achieving convergence to the target distribution.

3. **Effective Sample Size (ESS)**: The samples extracted using these Hamiltonian-based algorithms are not equivalent to i.i.d. samples extracted directly from the target distribution. Hence, we need to define a way to quantify the efficiency of a sampler in producing *independent* samples from $\pi(\boldsymbol{\theta})$.

   For instance, in order to measure the sampling efficiency of our Markov-Chain Monte Carlo we can use the **Effective Sample Size** (ESS): the number of equivalent i.i.d. samples from the target distribution our method can generate.

   Note: Due to the complexity of properly defining the Effective Sample Size metric — specially in the context of MMHMC, where we need to account for the importance reweighing in Eq. (3.33) — we will limit ourselves here to presenting the *simplest* formulation of the ESS for strictly MCMC methods (i.e., without any sort of importance sampling): the Autocorrelation-Based ESS with several stopping criterion.[2]

   Autocorrelation-Based ESS for MCMC Sampling: The simplest form of ESS estimation for *strictly* MCMC methods is based on the infinite sum of the autocorrelations of the samples as in Eq. (3.44).

$$\text{ESS} = \frac{N}{1 + 2\sum_{k=1}^{\infty} \gamma_k} \tag{3.44}$$

   where γ_k is the sample **autocorrelation** at lag k. The following stopping criteria are usually employed in order to approximate the infinite series above. These methods include:

   - **Geyer's Stopping Criteria** [81] proposes that the autocorrelation sequence is truncated where the pairwise sums become non-positive. Because pairwise sums of the elements of that sequence are positive, any deviations are only possible due to noise.

     Let the sums of adjacent pairs of autocovariances be

$$\Gamma_k = \gamma_{2k} + \gamma_{2k+1}, \qquad \forall k \in [0, N/2] \tag{3.45}$$

     Then, Γ_k is strictly positive and decreasing (i.e., for irreducible, reversible, stationary Markov Chains). Geyer's criterion yields three possible threshold estimators:

     a) **Initial Sequence Positive Estimator (ISPE)**. That is, our threshold m is the largest integer such that Γ_k remains strictly positive. This criterion for truncating the infinite sum works well most of the times, but beware of cases in which the estimated autocorrelations stays positive for many lags. That is:

$$\text{ESS} = \frac{N}{1 + 2\sum_{k=1}^{2m+1} \gamma_k} = \frac{N}{-1 + 2\sum_{k=0}^{m} \Gamma_k} \tag{3.46}$$

[2] As of the time of this writing, no comprehensive overview has been published on the extensive alternative definitions and formulations of the ESS. Therefore, we *might* plan to publish such a review, as well as our novel work towards defining a proper ESS metric for joint Markov-Chain Monte Carlo and Importance Sampling algorithms, in the near future.

b) **Initial Monotone Sequence Estimator (IMSE)**. Further reducing the "bumps" in the ACF curve by reducing the estimated Γ_i to the minimum of the preceding ones so that the estimated sequence is monotone. That is,

$$\Gamma_k = \min\{\Gamma_0, \Gamma_1, \dots, \Gamma_{k-1}\} \tag{3.47}$$

This is the implementation used in the software package Stan and the popular Python library Arviz.

c) **Initial Convex Sequence Estimator (ICSE)**. Even further reducing Γ_i to the **greatest convex minorant** (GCM).

- From the **variance** of the estimator:

$$\text{ESS} = \frac{N}{1 + 2\sum_{k=1}^{N-1} \frac{N-k}{N} \gamma_k} \tag{3.48}$$

- The infinite sum is often truncated at lag k when $\gamma_k < 0.05$.
- Until correlation switches sign. This method does **not** estimate the ESS of super-efficient chains (where ESS > N) correctly.

In practice, the ESS is usually reported as the ratio to the number of samples N. That is, super-efficient samplers yield values of the ESS > 1, while ESS = 1 means that the samples from the method are i.i.d. in practice.

4. **Monte Carlo Standard Error (MCSE)**: Moreover, an important performance metric derived from the Effective Sample Size is the **Monte Carlo Standard Error** (MCSE). It measures the variability in the estimated parameters, indicating the deviation of simulation results from N samples θ_n and true values taken to be the estimated mean value $\bar{\theta}$. A lower MCSE implies higher precision and reliability of the samples, thus more effective convergence to the target distribution.

$$\text{MCSE} = \sqrt{\frac{\sigma^2}{\text{ESS}}} = \frac{\frac{1}{N-1}\sum_{n=1}^{N}(\theta_n - \bar{\theta})^2}{\sqrt{\text{ESS}}} \tag{3.49}$$

5. **Integrated Autocorrelation Time (IACT)**: Likewise, the **Integrated Autocorrelation Time** (IACT) can also be defined as the number of Monte Carlo iterations needed, on average, for an independent sample to be drawn.

$$\text{IACT} = \frac{N}{\text{ESS}} \tag{3.50}$$

That is, on average, IACT correlated samples are required in order to reduce the variance of the estimator $\hat{I}$ by the same amount as a single uncorrelated sample.

Table 3.3: Evaluation Metrics for Estimating the Performance of Hamiltonian-based Sampling Methods

Metric	Assesses	Equation
PSRF	Convergence	(3.42)
AR	Space Exploration	(3.38)
MCSE	Variance	(3.49)
ESS	Sampling Efficiency	(3.44)
IACT	Sampling Efficiency	(3.50)
GRAD-e	Sampling Efficiency	(3.51)

6. **Number of Gradient Computations by Integrator**: Additionally, for a given Hamiltonian integrator, the ratio of the number of gradient computations and the ESS is also generally regarded as a reliable metric for assessing the quality of the estimator:

$$\text{GRAD-e} = \frac{\text{Number of gradient computations}}{\text{ESS}} = \frac{k \cdot \bar{L}}{\text{ESS}} \tag{3.51}$$

where k is the stage of the splitting integrator, and $\bar{L}$ can either be computed as the *average* number of integration steps, or as $\bar{L} = (L_{\text{upper}} + 1)/2$.

A comprehensive summary of all these presented metrics can be found in Table 3.3.

Adaptive Methods for Multi-Stage Splitting Integrators

So far, we have limited our discussion on Hamiltonian numerical integrators to the Verlet/Leapfrog method in Algorithm 3. However, in practice, the numerical integration of the Hamiltonian equations of motion is crucial for Hamiltonian-based Monte Carlo methods, since its accuracy and efficiency strongly affect the overall performance of the method. The Verlet/Leapfrog integrator is currently the method of choice due to its simplicity, optimal stability properties and computational efficiency. In this section, we introduce the readers to the recently proposed family of Multi-Stage Splitting Integrators (MSSIs) [50, 57, 78, 79] which have shown promising performance in statistical and molecular simulation applications; and present the s-AIA (Adaptive Integration Approach in Computational Statistics) [79] algorithm for adaptive tuning of the integration and sampling parameters.

As a quick remainder to the reader, Hamiltonian Monte Carlo (HMC) is a Markov-Chain Monte Carlo (MCMC) method for obtaining correlated samples $\boldsymbol{\theta}_i$ from a target distribution $\pi(\boldsymbol{\theta})$ as in Eq. (3.52).

$$\boldsymbol{\theta}_i \sim \pi(\boldsymbol{\theta}) \tag{3.52}$$

In practice, this is achieved by simulating the Hamiltonian dynamics of a system, where the Hamiltonian function is defined as follows:

$$H(\boldsymbol{\theta}, \mathbf{p}) = K(\mathbf{p}) + U(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p} + U(\boldsymbol{\theta}) \tag{3.53}$$

where $K(\mathbf{p})$ corresponds to the kinetic energy of the system, and $U(\theta)$ to the potential energy, respectively[3]. The Hamiltonian potential $U(\theta)$ is related to the target distribution by Eq. (3.54).

$$U(\theta) = -\log \pi(\theta) + \mathscr{C} \tag{3.54}$$

In HMC, both the position θ and momentum $\mathbf{p}$ are updated through the numerical integration of the Hamiltonian dynamics in Eq. (3.55).

$$\dot{\theta} = H_{\mathbf{p}}(\theta, \mathbf{p}) = M^{-1}\mathbf{p}, \qquad \dot{\mathbf{p}} = -H_{\theta}(\theta, \mathbf{p}) = -U_{\theta}(\theta) \tag{3.55}$$

In practice, the integration of the Hamiltonian dynamics in Eq. (3.55) is performed by resorting to the idea of *splitting*. First, we define the split systems below:

$$\text{(System } A\text{)} \qquad \dot{\theta} = K_{\mathbf{p}}(\mathbf{p}) = M^{-1}\mathbf{p} \qquad \dot{\mathbf{p}} = -K_{\theta}(\mathbf{p}) = 0 \tag{3.56}$$

$$\text{(System } B\text{)} \qquad \dot{\theta} = U_{\mathbf{p}}(\theta) = 0 \qquad \dot{\mathbf{p}} = -U_{\theta}(\theta) \tag{3.57}$$

From these systems, we define the solution flows as in Eq. (3.58).

$$\varphi_t^A(\theta, \mathbf{p}) = (\theta + tM^{-1}\mathbf{p}, \mathbf{p}), \qquad \varphi_t^B(\theta, \mathbf{p}) = (\theta, \mathbf{p} - tU_{\theta}(\theta)) \tag{3.58}$$

These flows are often called a *position drift* and a *momentum kick* respectively. The integration of the target dynamics in Eq. (3.55) is carried out by combining drifts and kicks.

Example 3.3.1 (1-Stage Velocity Verlet Integrator)**.** *For example, the popular 1-Stage Velocity Verlet integrator:*

$$\begin{aligned} \mathbf{p} &= \mathbf{p} - \frac{\varepsilon}{2} U_{\theta}(\theta) \\ \theta &= \theta + \varepsilon M^{-1}\mathbf{p} \\ \mathbf{p} &= \mathbf{p} - \frac{\varepsilon}{2} U_{\theta}(\theta) \end{aligned} \tag{3.59}$$

Can be expressed as a composition of the flows in Eq. (3.58)*:*

$$\Psi_{\varepsilon}^{(1-VV)} = \varphi_{\varepsilon/2}^B \circ \varphi_{\varepsilon}^A \circ \varphi_{\varepsilon/2}^B \tag{3.60}$$

As per the notation used throughout this text, ε is the length of an integration step, i.e., the step-size.

Avid readers may now realize that the "Leapfrog" integrator we have been discussing so far (see Algorithm 3) is nothing but the 1-Stage Velocity Verlet in Eq. (3.60) wrapped by two additional half-step momentum kicks at the beginning and at the end of the integration process.

[3]Note that the Hamiltonian is separable, as $K(\mathbf{p})$ and $U(\theta)$ depend exclusively on $\mathbf{p}$ and θ, respectively.

Definition 3.3.1 (Multi-Stage Splitting Integrators)**.** *Let k be a positive integer. We define the family of k-stage splitting integrators (with $k-1$ free parameters) as follows:*

$$\Psi_\varepsilon^{(k)} = \begin{cases} \varphi^B_{b_1\varepsilon} \circ \varphi^A_{a_1\varepsilon} \circ \cdots \circ \varphi^A_{a_{k'}\varepsilon} \circ \varphi^B_{b_{k'+1}\varepsilon} \circ \varphi^A_{a_{k'}\varepsilon} \circ \cdots \circ \varphi^A_{a_1\varepsilon} \circ \varphi^B_{b_1\varepsilon} & \text{if } k = 2k' \\ \varphi^B_{b_1\varepsilon} \circ \varphi^A_{a_1\varepsilon} \circ \cdots \circ \varphi^B_{b_{k'}\varepsilon} \circ \varphi^A_{a_{k'}\varepsilon} \circ \varphi^B_{b_{k'}\varepsilon} \circ \cdots \circ \varphi^A_{a_1\varepsilon} \circ \varphi^B_{b_1\varepsilon} & \text{if } k = 2k' - 1 \end{cases} \quad (b_i, a_j \in \mathbb{R}^+) \tag{3.61}$$

where $\circ$ is the composition of solution flows, and the coefficients b_i, a_j must satisfy the following:

$$\begin{cases} 2\sum_{i=1}^{k'} b_i + b_{k'+1} = 2\sum_{j=1}^{k'} a_j = 1 & \text{if } k = 2k' \\ 2\sum_{i=1}^{k'} b_i = 2\sum_{j=1}^{k'-1} a_j + a_{k'} = 1 & \text{if } k = 2k' - 1 \end{cases} \tag{3.62}$$

The integrators in Eq. (3.61) are symplectic as compositions of flows thus leading to a conservation of the Hamiltonian, and reversible due to their palindromic structure. Likewise, the number of stages k is the number of times the integration algorithm performs an evaluation of gradients $U_\theta(\theta)$ per step-size ε.

Remark. *Evidently, most of the computational cost in HMC arises from gradient evaluations. Since splitting integrators with different numbers of stages require different numbers of gradient evaluations per step, using a common value of L and ε does not yield fair comparisons. To address this, if $\hat{L}$ is the number of gradient evaluations used by the 1-stage Velocity Verlet method with step size ε, then a k-stage integrator is run for $L = \hat{L}/k$ steps of size $k\varepsilon$. Ensuring that all integrators simulate the dynamics over the same time interval $\hat{L}\varepsilon$ and incur the same number of gradient evaluations.*

Example 3.3.2 (2-Stage Splitting Integrators)**.** *From the general formulation of the multi-stage Hamiltonian integrators in Eq.* (3.61) *we can easily derive the 1-parameter family of 2-stage integrators as follows:*

$$\Psi_\varepsilon^{(2)} = \varphi^B_{b_1\varepsilon} \circ \varphi^A_{a_1\varepsilon} \circ \varphi^B_{b_2\varepsilon} \circ \varphi^A_{a_1\varepsilon} \circ \varphi^B_{b_1\varepsilon} \quad (\because k = 2, k' = 1) \tag{3.63}$$

Likewise, the conditions in Eq. (3.62) *must be satisfied:*

$$\begin{cases} 2\sum_{i=1}^{k'} b_i + b_{k'+1} = 1 \implies 2b_1 + b_2 = 1 \implies b_2 = 1 - 2b_1 \\ 2\sum_{j=1}^{k'} a_j = 1 \implies 2a_1 = 1 \implies a_1 = \dfrac{1}{2} \end{cases} \tag{3.64}$$

Moreover, because all coefficients must be in $\mathbb{R}^+$:

$$b_2 > 0 \implies 1 - 2b_1 > 0 \implies b_1 < \frac{1}{2} \tag{3.65}$$

That is: $a_1 = 1/2, b_1 \in (0, 1/2), b_2 = 1 - 2b_1$. Therefore, the integrators can be rewritten as (note that for simplicity in this case $b = b_1 \in (0, 1/2)$):

$$\Psi_\varepsilon^{(2)} = \varphi^B_{b\varepsilon} \circ \varphi^A_{\varepsilon/2} \circ \varphi^B_{(1-2b)\varepsilon} \circ \varphi^A_{\varepsilon/2} \circ \varphi^B_{b\varepsilon} \tag{3.66}$$

Table 3.4: Special Cases of 2- & 3-Stage Splitting Integrators

Integrator	**Nº of Stages (k)**	**Coefficients**
1-Stage Velocity Verlet (VV1)	1	-
2-Stage Velocity Verlet (VV2)	2	$b = 1/4$
2-Stage BCSS (BCSS2)	2	$b = 0.211781$
2-Stage Minimum-Error (ME2)	2	$b = 0.193183$
3-Stage Velocity Verlet (VV3)	3	$a = 1/3, b = 1/6$
3-Stage BCSS (BCSS3)	3	$a = 0.296195, b = 0.118880$
3-Stage Minimum-Error (ME3)	3	$a = 0.290486, b = 0.108991$

As the reader may realize, if we set $b = 0$, we get the 1-stage Velocity Verlet integrator in Eq. (3.60).

Example 3.3.3 (3-Stage Splitting Integrators)**.** *Likewise, the 2-parameter family of 3-stage integrators can be derived as well in the same fashion:*

$$\Psi_\varepsilon^{(3)} = \varphi^B_{b_1\varepsilon} \circ \varphi^A_{a_1\varepsilon} \circ \varphi^B_{b_2\varepsilon} \circ \varphi^A_{a_2\varepsilon} \circ \varphi^B_{b_2\varepsilon} \circ \varphi^A_{a_1\varepsilon} \circ \varphi^B_{b_1\varepsilon} \quad (\because k = 3, k' = 2) \tag{3.67}$$

Then, we take a look at the constraints in Eq. (3.62):

$$\begin{cases} 2\sum_{i=1}^{k'} b_i = 1 \Longrightarrow 2(b_1 + b_2) = 1 \Longrightarrow b_2 = \dfrac{1}{2} - b_1 \\ 2\sum_{j=1}^{k'-1} a_j + a_{k'} = 1 \Longrightarrow 2a_1 + a_2 = 1 \Longrightarrow a_2 = 1 - 2a_1 \end{cases} \tag{3.68}$$

Moreover, because all coefficients must be in $\mathbb{R}^+$, we can rewrite the coefficients as $a, b \in (0, 1/2)$, $a_2 = 1-2a$, $b_2 = 1/2 - b$. Therefore, the family of integrators can be expressed as:

$$\Psi_\varepsilon^{(3)} = \varphi^B_{b\varepsilon} \circ \varphi^A_{a\varepsilon} \circ \varphi^B_{(1/2-b)\varepsilon} \circ \varphi^A_{(1-2a)\varepsilon} \circ \varphi^B_{(1/2-b)\varepsilon} \circ \varphi^A_{a\varepsilon} \circ \varphi^B_{b\varepsilon} \tag{3.69}$$

Finally, Table 3.4 provides a comprehensive summary of some interesting specific cases of 2-stage and 3-stage splitting integrators.[4]

Although the use of these multi-stage splitting integrators has shown improved conservation of the Hamiltonians over the traditional Verlet/Leapfrog method, they are typically more unstable and thus generally benefit from *adaptive* methods for parameter tuning and selection. For this reason, the *Adaptive Integration Approach* (AIA) [77] introduced an automatic selection process for the best Hamiltonian numerical integrator in practical scenarios. Subsequent improvements such as Modified AIA (MAIA), and Extended-MAIA (e-MAIA) quickly garnered attention due to their boost in performance. Finally, [79] introduced a variation on these methods

[4]The derivation of the coefficients can be found in [79], but we limit ourselves here to presenting these values as they will be useful in the s-AIA method.

for its application in computational statistics: s-AIA. For an in-depth explanation of the method, we refer the readers to [79]. However, we provide the full pseudocode in Algorithm 6 as s-AIA, readily implemented in pyHaiCS, will be used in practice later to select the optimal integration parameters.

Derivation of $\rho_k(\varepsilon, b)$

Given a k-stage palindromic splitting integrator Ψ_ε (ε is the integration step-size), it acts on a configuration (θ_i, p_i) at the i-th iteration as:

$$\Psi_\varepsilon \begin{pmatrix} q_i \\ p_i \end{pmatrix} = \begin{pmatrix} q_{i+1} \\ p_{i+1} \end{pmatrix} = \begin{pmatrix} A_\varepsilon^{\mathbf{z}} & B_\varepsilon^{\mathbf{z}} \\ C_\varepsilon^{\mathbf{z}} & D_\varepsilon^{\mathbf{z}} \end{pmatrix} \begin{pmatrix} q_i \\ p_i \end{pmatrix} \tag{3.70}$$

for suitable method-dependent coefficients $A_\varepsilon^{\mathbf{z}}, B_\varepsilon^{\mathbf{z}}, C_\varepsilon^{\mathbf{z}}, D_\varepsilon^{\mathbf{z}}$ ($\mathbf{z} = \{b_i, a_j\}$ is the set of $k-1$ integration coefficients). Then:

$$\rho_k(\varepsilon, \mathbf{z}) = \frac{(B_\varepsilon^{\mathbf{z}} + C_\varepsilon^{\mathbf{z}})^2}{2\left(1 - A_\varepsilon^{\mathbf{z}^2}\right)} \tag{3.71}$$

Special Cases: In the particular cases where $k = 2, 3$, the expressions for $\rho_k(\varepsilon, b)$ are given by Eqs. (3.72) and (3.73), respectively.

$$\rho_2(\varepsilon, b) = \frac{\varepsilon^4 \left(2b^2 \left(\frac{1}{2} - b\right)\varepsilon^2 + 4b^2 - 6b + 1\right)^2}{8(2 - b\varepsilon^2)\left(2 - \left(\frac{1}{2} - b\right)\varepsilon^2\right)\left(1 - b\left(\frac{1}{2} - b\right)\varepsilon^2\right)} \tag{3.72}$$

$$\rho_3(\varepsilon, b) = \frac{\varepsilon^4 \left[-3b^4 + 8b^3 - 19/4b^2 + b + b^2\varepsilon^2\left(b^3 - 5/4b^2 + b/2 - 1/16\right) - 1/16\right]^2}{2(3b - b\varepsilon^2(b - 1/4) - 1)(1 - 3b - b\varepsilon^2(b - 1/2)^2)(-9b^2 + 6b - \varepsilon^2(b^3 - 5/4b^2 + b/2 - 1/16) - 1)} \tag{3.73}$$

For 3-stage integrators the pairs (b, a) are restricted, for stability reasons, to those that satisfy:

$$6ab - 2a - b + \frac{1}{2} = 0 \tag{3.74}$$

Algorithm 6 Adaptive Integration Approach in Computational Statistics (s-AIA)

Input:

N_{tune}: Number of Monte Carlo Samples (*Tuning Stage*)
N_{check}: Number of Monte Carlo Samples for AR Check (*Tuning Stage*)
$N_{\text{burn-in}}$: Number of Monte Carlo Samples (*Burn-In Stage*)
N_{prod}: Number of Monte Carlo Samples (*Production Stage*)
D: Dimensionality of the Data
k: k-Stage of the Integrator (Usually 2 or 3)
α_{target}: Targeted Acceptance Rate (Usually $\alpha_{\text{target}} = 0.92$ as in [79])
ξ: Sensibility (> 0)
δ_ε: Step-Size Increment (> 0)
I_ω: Frequencies Calculation (1 = yes, 0 = no)
M: Mass Matrix

Output:

$\{\theta_n\}_{n=1}^{N}$: Samples from the Target Distribution (i.e., the Trajectory)

1) Tuning Stage:

Initialize Tuning Parameters: $N = N_{\text{tune}}, \varepsilon = 1/D, L = 1, \Psi_\varepsilon = \Psi_\varepsilon^{(VV)}$

$\varepsilon_{\text{tuned}}$ = s-AIA-Tuning$(N, N_{\text{check}}, \alpha_{\text{target}}, \xi, \varepsilon, \delta_\varepsilon, L, M, \Psi_\varepsilon)$ – See Algorithm 7

2) Burn-In Stage:

Initialize Burn-In Parameters: $N = N_{\text{burn-in}}, \varepsilon = \varepsilon_{\text{tuned}}, L = 1, \Psi_\varepsilon = \Psi_\varepsilon^{(VV)}$

$\{\bar{\varepsilon}_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}, \{\varepsilon_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$ = s-AIA-Burn-In$(N, I_\omega, D, \varepsilon, L, M, \Psi_\varepsilon)$ – See Algorithm 8

Compute Optimal Integration Coefficients $\{b^k_{\text{opt}_i}\}_{i=1}^{N_{\text{prod}}}$ such that:

$$b^k_{\text{opt}_i} = \underset{b\in(b_{\text{MEk}},b_{\text{VVk}})}{\arg\min}\left(\max_{0<\varepsilon<\bar{\varepsilon}_{\text{prod}_i}} \rho_k(\varepsilon, b)\right)$$

See Table 3.4 for the values of $b_{\text{MEk}}, b_{\text{VVk}}$ in the cases where $k = 2, 3$, and Section 3.3 for the complete derivation of $\rho_k(\cdot)$.

3) Production Stage:

At each iteration, L_{prod_i} is drawn from:

$$L_{\text{prod}_i} \sim \mathcal{U}(1, 2\bar{L} - 1), \qquad \bar{L}\varepsilon = \bar{L}k = \tau D \text{ (usually, } \bar{L}k = D)$$

Initialize Prod. Parameters: $N = N_{\text{prod}}, \varepsilon \in \{\varepsilon_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}, L \in \{L_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}, \Psi_\varepsilon = \Psi_\varepsilon^{(\text{s-AIA}k)}$

Run HMC$(N_{\text{prod}}, \varepsilon_{\text{prod}_i}, L_{\text{prod}_i}, M, \Psi_\varepsilon)$

Algorithm 7 Tuning Stage of s-AIA (s-AIA-Tuning)

Input:

N_{tune}: Number of Tuning Iterations
N_{check}: Number of Iterations for AR Check
α_{target}: Targeted Acceptance Rate
ξ: Sensibility
ε: Step-Size
δ_{ε}: Step-Size Increment
L: Number of Integration Steps (i.e., Trajectory Length)
M: Mass Matrix
Ψ_{ε}: Hamiltonian Numerical Integrator

Output:

$\varepsilon_{\text{tuned}}$: Tuned Step-Size

Initialize $\varepsilon_{\text{tuned}} = \varepsilon, N = N_{\text{tot}} = N_{\text{acc}} = 0$
while $N_{\text{tot}} + N_{\text{check}} < N_{\text{tune}}$ **do**
    Run HMC($N_{\text{check}}, \varepsilon_{\text{tuned}}, L, M, \Psi_{\varepsilon}$)
    $N = N + N_{\text{check}}$
    N_{acc} = Number of Acceptances Over the Last N Iterations
    AR = N_{acc}/N
    if AR < $\alpha_{\text{target}} - \xi$ **then**
        $\varepsilon_{\text{tuned}} = \varepsilon_{\text{tuned}} - \delta_{\varepsilon}$
        $N = 0$
    else if AR > $\alpha_{\text{target}} + \xi$ **then**
        $\varepsilon_{\text{tuned}} = \varepsilon_{\text{tuned}} + \delta_{\varepsilon}$
        $N = 0$
    end if
    $N_{\text{tot}} = N_{\text{tot}} + N_{\text{check}}$
end while

Algorithm 8 Burn-In Stage of s-AIA (s-AIA-Burn-In) – Part 1

Input:

$N_{\text{burn-in}}$: Number of Burn-In Iterations
I_ω: Frequencies Calculation (1 = yes, 0 = no)
D: Dimensionality of the Data
ε: Step-Size
L: Number of Integration Steps (i.e., Trajectory Length)
M: Mass Matrix
Ψ_ε: Hamiltonian Numerical Integrator

Output:

$\{\bar{\varepsilon}_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$: Set of Randomized Dimensionless Step-Sizes
$\{\varepsilon_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$: Set of Randomized Production Step-Sizes

Run HMC($N_{\text{burn-in}}, \varepsilon, L, M, \Psi_\varepsilon$)
During the simulation, at each step ζ:
    Find eigenvalues $\lambda_j^{(\zeta)}$ of the Hessian matrix $H_{i,j}^{(\zeta)} = \dfrac{\partial^2 U(\boldsymbol{\theta}^{(\zeta)})}{\partial\theta_i^{(\zeta)}\partial\theta_j^{(\zeta)}}, i, j = 1, \ldots, D$
    Compute Frequencies: $\omega_j^{(\zeta)} = \sqrt{\lambda_j^{(\zeta)}}, j = 1, \ldots, D$
Average Frequencies: $\omega_j = \dfrac{1}{N_{\text{burn-in}}} \displaystyle\sum_{\zeta=1}^{N_{\text{burn-in}}} \omega_j^{(\zeta)}$
$\tilde{\omega} = \max \omega_j$
N_{acc} = Number of Acceptances Over the Last N Iterations
AR = $N_{\text{acc}}/N_{\text{burn-in}}$
if (I_ω == 0) **then**
    $S = \max\left(1, \dfrac{2}{\tilde{\omega}\varepsilon}\sqrt[6]{\dfrac{2\pi(1-\text{AR})^2}{D}}\right)$
    if $S \leq 2$ **then**
        Fitting Factor: $S_f = S$
        Stability Limit: $0 < \varepsilon < SL = \dfrac{2k}{S_f\tilde{\omega}}, k = 1, 2, 3, \ldots$
        Compute $\{\varepsilon_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$ such that $\mathcal{R}(\varepsilon_{\text{prod}}) \leq \mathcal{R}(SL)$, where $\mathcal{R}(\cdot)$ is a randomization scheme.
        Compute Dimensionless Step-Sizes: $\{\bar{\varepsilon}_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$, where:

$$\bar{\varepsilon}_{\text{prod}_i} = \begin{cases} \dfrac{2\varepsilon_i}{\varepsilon}\sqrt[6]{\dfrac{2\pi(1-\text{AR})^2}{D}} & \text{if } S > 1 \\ \tilde{\omega}\varepsilon_i & \text{otherwise} \end{cases}$$

    end if
end if
// Continues in Part 2...

Algorithm 8 Burn-In Stage of s-AIA (s-AIA-Burn-In) – Part 2

if $(I_\omega == 1)$ **or** $(S > 2)$ **then**

$S_\omega = \max\left(1, \frac{2}{\varepsilon}\sqrt[6]{\frac{2\pi(1-\mathrm{AR})^2}{\sum_{j=1}^{D}\omega_j^6}}\right)$

Compute Std. Deviation of the Frequencies: $\sigma_\omega = \sqrt{\frac{\sum_{j=1}^{D}(\omega_j - \tilde{\omega})^2}{D}}$

if $\sigma_\omega \leq 1$ **then**

Fitting Factor: $S_f = S_\omega$

Stability Limit: $0 < \varepsilon < SL = \frac{2k}{S_f\tilde{\omega}}, k = 1, 2, 3, \ldots$

Compute $\{\varepsilon_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$ such that $\mathscr{R}(\varepsilon_{\text{prod}}) \leq \mathscr{R}(SL)$, where $\mathscr{R}(\cdot)$ is a randomization scheme.

Compute Dimensionless Step-Sizes: $\{\bar{\varepsilon}_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$, where:

$$\bar{\varepsilon}_{\text{prod}_i} = \begin{cases} \frac{2\tilde{\omega}\varepsilon_i}{\varepsilon}\sqrt[6]{\frac{2\pi(1-\mathrm{AR})^2}{\sum_{j=1}^{D}\omega_j^6}} & \text{if } S_\omega > 1 \\ \tilde{\omega}\varepsilon_i & \text{otherwise} \end{cases}$$

else if $\sigma_\omega > 1$ **then**

Stability Limit: $0 < \varepsilon < SL = \frac{2k}{S_\omega(\tilde{\omega} - \sigma_\omega)}, k = 1, 2, 3, \ldots$

Compute $\{\varepsilon_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$ such that $\mathscr{R}(\varepsilon_{\text{prod}}) \leq \mathscr{R}(SL)$, where $\mathscr{R}(\cdot)$ is a randomization scheme.

Compute Dimensionless Step-Sizes: $\{\bar{\varepsilon}_{\text{prod}_i}\}_{i=1}^{N_{\text{prod}}}$, where:

$$\bar{\varepsilon}_{\text{prod}_i} = \begin{cases} \frac{2(\tilde{\omega} - \sigma_\omega)\varepsilon_i}{\varepsilon}\sqrt[6]{\frac{2\pi(1-\mathrm{AR})^2}{\sum_{j=1}^{D}\omega_j^6}} & \text{if } S_\omega > 1 \\ (\tilde{\omega} - \sigma_\omega)\varepsilon_i & \text{otherwise} \end{cases}$$

end if

end if

3.4 Related Work

The application of Bayesian methods in clinical studies has been extensively explored in the literature. For instance, [82] proposes a dynamic stochastic model for predicting breast cancer survival in large population cohorts, whereas [83] introduces a simple model for screening breast cancer patients, or [84] presents a novel Hybrid Bayesian Network to predict survival in HER2+ breast cancer patients. Similarly, [85, 86] present hierarchical Bayesian approaches for predicting prognostic survival outcomes in the context of pancreatic cancer. Most recently, Bayesian methods have been further used to study the limitations of hormone-based therapies as breast cancer treatments in order to develop more favorable solutions [87–89].

In the context of breast cancer therapy research — more specifically of cell-line genomic studies — several genes (or families of them) have been identified as potential biomarkers responsible for the resistance to these

endocrine treatments. For instance, the SOX2-SOX9 [33, 90, 91] and Interleukin [92] families, Notch [93, 94] or Wnt [33] pathways and cell proliferation [95–97] have often been validated through clinical *omics* samples demonstrating the potential of these methods for identifying prognostic biomarkers. Furthermore, as mentioned earlier, the development of prognostic signatures such as Mammaprint [36] and OncotypeDX [35] has been crucial for the identification of high-risk breast cancer patients. Likewise, genetic signatures specifically related to the problem of resistance to hormone therapy in breast cancer have been identified most recently [98–102]. More specifically, in the context of tamoxifen resistance, [103–105] identify key individual biomarkers in the resistance development process, [106] proposes a 5-gene signature, and finally [37] proposes a 6-gene signature by combining patient and cell-line RNA-seq data — using a Simulated Annealing (SA) algorithm coupled with a Bayesian logistic regression model with a single per-signature global score — to predict resistance to hormone-based therapies. It is from this last work that we take inspiration for the development of our Bayesian modeling framework.

Finally, we believe that it is worth commenting on the limitations of the current *state-of-the-art* methods in the field. As we have seen, the development of prognostic signatures for breast cancer resistance to hormone-based therapies has been a topic of interest for many years. However, the current methods are limited in their ability to provide a comprehensive understanding of the underlying biological mechanisms that lead to this phenomenon. In particular, the current methods are based on statistical models that do not take into account the underlying biological or molecular functions, and can therefore be limited in their ability to predict resistance in new patients. As a matter of fact, some studies suggest that the current existing methods may be yielding suboptimal signatures that seem significant at first glance but are not actually meaningful in order to determine effective, and biologically coherent, prognostic signatures [107–109]. This is mainly due to spurious correlations with proliferation genetic markers [108], poor validation schemes, or a high number of genes being considered in the signature [109]. Because of these reasons, we have decided to approach the problem from a different perspective — without relying on the identification of a specific set of genes that conform a signature —, by developing a Bayesian modeling framework that can provide a more comprehensive understanding of the underlying biological mechanisms that drive resistance to hormone-based therapies in breast cancer. Finally, a summary of the related work described above can be found in Table 3.5.[5]

[5] Please note that due to the broad nature of the topic, and because an extensive overview of Bayesian statistical modeling and Monte Carlo posterior sampling methods has already been provided throughout this whole chapter, we have decided to focus on the most relevant work that is directly related to the study of potential genetic biomarkers for resistance to hormone-based therapies in breast cancer. These references provide a comprehensive overview of the *state-of-the-art* in the field and are a good starting point for further research, as well as for comparing our results with the existing genetic signatures.

Table 3.5: Summary of Related Work in Bayesian Methods and Breast Cancer Research

Authors	**Contribution**	**Reference**
Teng et al. (2022)	Dynamic Bayesian model for breast cancer survival prediction in large population cohorts	[82]
Huang et al. (2018)	Bayesian simulation model for breast cancer screening, incidence, treatment, and mortality	[83]
Su et al. (2023)	Novel prognostic Hybrid Bayesian Network for modeling HER2+ breast cancer survival	[84]
Chu et al. (2022), Smith et al. (2014)	Hierarchical Bayesian approaches for predicting prognostic survival outcomes in pancreatic cancer	[85, 86]
Wang et al. (2022), He et al. (2024), Jiang et al. (2024)	Bayesian methods to study limitations of hormone-based therapies in breast cancer	[87–89]
Piva et al. (2014), Aurrekoetxea et al. (2024), Domenici et al. (2019)	Identification of SOX2-SOX9 family as potential biomarkers for resistance to endocrine treatments	[33, 90, 91]
Sarmiento et al. (2020)	Interleukin family as potential biomarkers for resistance to endocrine treatments	[92]
Magnani et al. (2013), Simoes et al. (2015)	Notch pathway as potential biomarker for resistance to endocrine treatments	[93, 94]
Piva et al. (2014)	Wnt pathway as potential biomarker for resistance to endocrine treatments	[33]
Gao et al. (2014), Huang et al. (2011), Palafox et al. (2022)	Cell proliferation as potential biomarker for resistance to endocrine treatments	[95–97]
Van't Veer et al. (2002)	Development of Mammaprint prognostic signature	[36]
Paik et al. (2004)	Development of OncotypeDX prognostic signature	[35]
Harrod et al. (2022), Xia et al. (2022), Miller et al. (2015), Kang et al. (2024), Jin et al. (2024)	Identification of genetic signatures related to resistance to hormone therapy in breast cancer	[98–102]
Hermawan et al. (2020), Mihaly et al. (2013), Wang et al. (2021)	Identification of key individual biomarkers in tamoxifen resistance development process	[103–105]
Rahem et al. (2020)	Proposal of a 5-gene signature for tamoxifen resistance	[106]
Parga-Pazos et al. (2024)	Proposal of a 6-gene signature for tamoxifen resistance using Bayesian regression models	[37]
Manjang et al. (2021), Venet et al. (2011), Goh et al. (2018)	Critique of current limitations of prognostic cancer genetic signatures	[107–109]

4 Bayesian Modeling of Tamoxifen Resistance in MCF7 Cells

"Today's posterior distribution is tomorrow's prior."

~ D. V. Lindley, *Bayesian Statistics: A Review* (1972) Ch. 1 [110]

In this chapter, we describe the methodologies deployed to model the genetic resistance of MCF7 cells to endocrine therapy, more specifically, tamoxifen treatment. We begin by discussing the available *multi-source* genetic data — including cell-line and patient RNA-seq data — for the study of tamoxifen resistance in MCF7 cells. We then discuss the pre-processing and statistical analysis of genetic sequencing data, more specifically, the process of Differential Expression Analysis (DEA) for identifying genes that are differentially expressed between tamoxifen-resistant and tamoxifen-sensitive cells. We then describe the integration of cell-line and patient RNA-seq data for genetic biomarker identification which is crucial for the development of predictive models for tamoxifen resistance in cells where data is scarce and high-dimensional. Likewise, we introduce pyHaiCS, a Python library for Hamiltonian-based Monte Carlo methods tailored towards practical applications in computational statistics. Finally, we discuss the different Bayesian models deployed for the study of tamoxifen resistance in MCF7 cells, including Bayesian Logistic Regression and other *point-estimate* resistance models.

The methodologies discussed in this chapter are aimed at providing a comprehensive framework for understanding and predicting tamoxifen resistance in MCF7 cells. By leveraging Bayesian modeling techniques, we aim to not only identify key genetic biomarkers associated with resistance but also to develop robust predictive models that can be used in clinical settings for predicting prognostic outcomes in breast cancer patients undergoing endocrine therapy. Likewise, the usage of explainability techniques, such as SHAP values, allows for a more nuanced understanding of the underlying genetic mechanisms driving tamoxifen resistance in MCF7 cells. Finally, we emphasize how the methodologies discussed in this chapter are generalizable to other cancer types and drug resistance mechanisms. By integrating cell-line and patient RNA-seq data, we can develop robust predictive models that are both interpretable and generalizable, making them valuable tools for researchers and practitioners in the field of cancer research and bioinformatics.

4.1 Available Multi-Source Genetic Data

We begin by discussing the available genetic data for our study of tamoxifen resistance in MCF7 cells [111]. The data used in this study derives from multiple sources. On the one hand, we have cell-line RNA-seq data for MCF7 CTRL and MCF7 TamR cells as controls for ER^+ breast cancer and resistance to tamoxifen, respectively. These cells were obtained from the American Type Culture Collection (ATCC) and were cultured in Dulbecco's Modified Eagle Medium (DMEM) supplemented with 8% fetal bovine serum (FBS) and 1% penicillin-

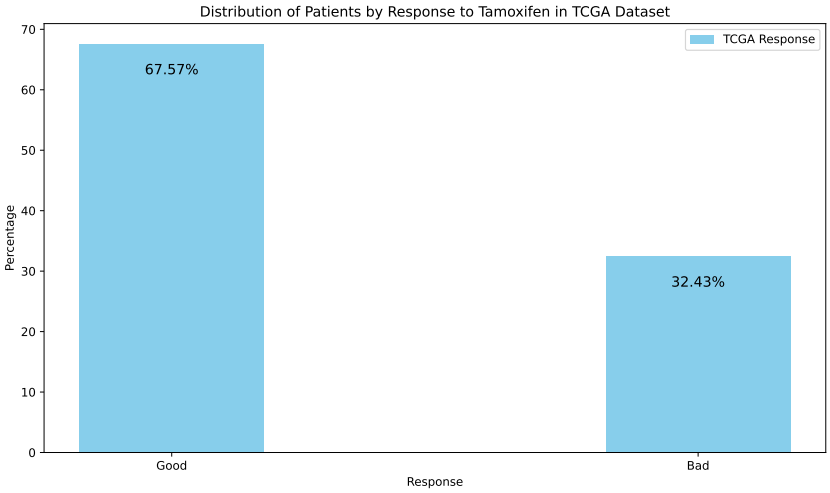

Figure 4.1: Distribution of Tamoxifen Responses in TCGA-BRCA Patients

streptomycin [33]. The MCF7 CTRL cells were treated with dimethyl sulfoxide (DMSO) as a control, while the MCF7 TamR cells were exposed to tamoxifen for 6 months to induce resistance. The cell-line RNA-seq data consists of gene expression profiles for MCF7 CTRL and MCF7 TamR cells, which were obtained using the Illumina HiScan-SQ platform [33, 91]. The sequencing was done entirely at CIC-bioGUNE, where the Cancer Heterogeneity Lab handled the library preparation, and the Genomics Platform performed the sequencing. Moreover, each cell line was sequenced in **triplicate** to ensure reproducibility and reliability of the results.

On the other hand, we have transcriptomic and clinical records for breast cancer patients from the publicly available **TCGA-BRCA** (The Cancer Genome Atlas Breast Invasive Carcinoma) database [112]. The TCGA database contains multi-omics data for over 30 cancer types, including breast cancer. Specifically, the patient RNA-seq data consists of gene expression profiles for breast cancer patients undergoing endocrine therapy, including tamoxifen treatment. As such, the clinical records include information on patient demographics, tumor characteristics, treatment regimens, and survival outcomes. In this case, the data contains detailed clinical and RNA-seq (IlluminaGA_RNASeqV2) data collected from more than 1000 breast cancer patients. However, only around 200 patients were ER+ and underwent tamoxifen endocrine therapy, making the data relatively scarce. Moreover, within these patients, only a small subset of 37 patients had both RNA-seq and survival data available, further limiting the sample size for our study. Regarding the distribution of outcomes in the patient data, around 30% of patients were classified as tamoxifen-resistant, while the remaining 70% were classified as tamoxifen-sensitive based on their survival outcomes (see Figure 4.1).

As a summary, the criteria followed for the selection of patients in the TCGA database were as follows:

1. Patients with available RNA-seq data.
2. Patients diagnosed with ER+ breast cancer.
3. Patients who underwent tamoxifen endocrine therapy for more than 2 years.
4. Patients with available survival data. That is, the persistence or disappearance of the tumor, the appearance of a recurrence or metastatic event or the eventual death of the patient.

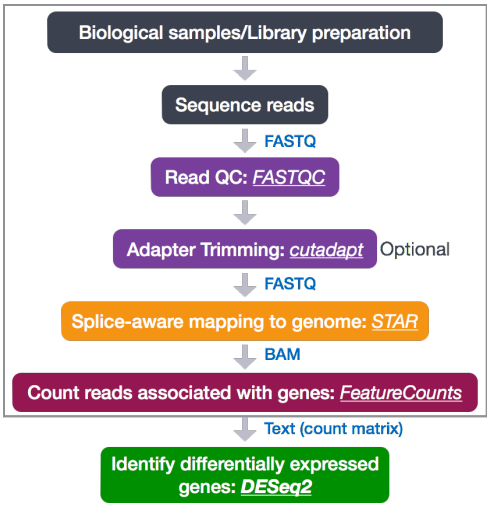

Figure 4.2: Pre-processing Workflow for RNA-seq Data (Diagram Extracted from this Post)

As a final note on the data, it is important to mention that, in the case of the TCGA dataset, the resistance to the treatment is not explicitly specified in most clinical records. However, it can be inferred in terms of the response to the treatment and a worsening of the patient's condition even after a treatment considered to be effective. In this case, we received the TCGA data used in [37], and the resistance classification was already done. For more information on the process for classifying patients as tamoxifen-resistant or tamoxifen-sensitive, we refer the reader to the original source [37] and his thesis [113].

4.2 Pre-Processing & Statistical Analysis of Genetic Sequencing Data

As stated above, RNA-sequencing data is at the absolute core of our study. This data is used to identify genes that are differentially expressed between tamoxifen-resistant and tamoxifen-sensitive MCF7 cells. The process of identifying these genes is known as **Differential Expression Analysis** (DEA). DEA is a statistical method used to identify genes that are differentially expressed between two or more conditions, such as treatment groups or disease states. In our case, we will apply this procedure first to each subset of data — i.e., cell-line and patient RNA-seq data — separately, and then we will integrate the results between the two sources of data. The goal is to identify genes that are differentially expressed between tamoxifen-resistant and tamoxifen-sensitive MCF7 cells, as well as between tamoxifen-resistant and tamoxifen-sensitive breast cancer patients; and extract those significant genes which are differentially expressed in the same direction in both cell-line and patient data. These genes will serve as potential biomarkers for tamoxifen resistance in MCF7 cells and breast cancer patients.

Although this text does not intend to delve into the technical details of DEA, it is important to mention that the process involves several steps, including data pre-processing, normalization, quality control, and statistical analysis. For the interested reader, we recommend the following resources for a more in-depth understanding of DEA [114–120] and also [121].

4.2.1 Pre-Processing Steps of RNA-seq Data

The pre-processing of RNA-seq data is a crucial step in order to convert raw sequencing reads into gene expression profiles that can be used for DEA. Generally, the workflow for DEA typically involves the following pre-processing steps (summarized in Figure 4.2):

1. **Reading of Raw Sequencing Data:** The first step is to read the *raw* sequencing data, which is typically stored in FASTQC format. This data contains the raw sequences of nucleotides obtained from the sequencing machine, as well as quality scores for each base call.

2. **Quality Control:** The next step is to perform quality control on the raw sequencing data. This involves checking the quality scores of the base calls, as well as identifying and removing low-quality reads.

3. **Trimming and Filtering:** After quality control, the next step is to trim and filter the raw sequencing data. This involves removing adapter sequences, low-quality bases, and other artifacts that may affect the accuracy of the gene expression profiles.

4. **Alignment:** Once the raw sequencing data has been pre-processed, the next step is to *align* the reads to a reference genome. This involves mapping the reads to the genome in order to determine the location of each read. The most common alignment tools used for RNA-seq data is the STAR aligner [122]. The alignment process is illustrated below in Figure 4.3.

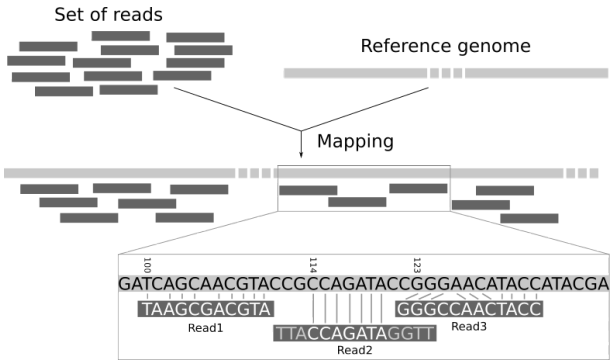

Figure 4.3: Illustration of the Gene Mapping/Alignment Process (Source)

5. **Quantification:** After alignment, the next step is to quantify the gene expression levels. This involves counting the number of reads that align to each gene in the reference genome. The most common tools used for quantification are HTSeq [123] and featureCounts [124]. These tools take as input the alignment file and a gene annotation file, and output a count matrix that contains the number of reads that align to each gene in each sample. This is commonly referred to as the *gene count matrix*, which is used as input for the Differential Expression Analysis (DEA) workflow in the following section.

After the pre-processing steps, the gene count matrix is used as input for the following steps in the DEA workflow, which involves statistical analysis to identify genes that are statistically differentially expressed between two or more cohorts. To put things into perspective, the gene count matrix usually consists of tens of thousands of genes (usually around 20,000-30,000) and hundreds of samples (sequenced cell-lines or patient samples). As

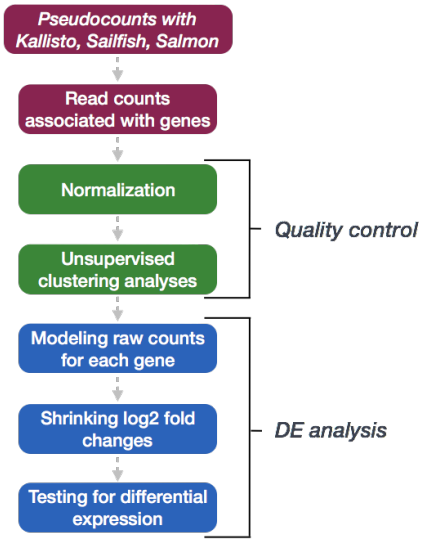

Figure 4.4: Differential Expression Analysis (DEA) Workflow (Diagram Extracted from this Post)

such, the DEA process is computationally intensive and highly-dimensional, thus methods for reducing the dimensionality of the data are often employed to improve the efficiency and accuracy of the analysis ranging from extracting the most significant genes to the use of machine learning techniques for feature selection. In our case, we will focus on the identification of genes that are *commonly* differentially expressed between tamoxifen-resistant and tamoxifen-sensitive MCF7 cells, as well as between tamoxifen-resistant and tamoxifen-sensitive breast cancer patients.

Likewise, some considerations must be taken into account when performing sequencing of RNA data. For instance, the *read length* refers to the length of the nucleotide sequences obtained from the sequencing machine. Longer read lengths are generally preferred as they provide more information about the gene expression levels. Conversely, the *sequencing depth* refers to the number of reads obtained for each sample. Higher sequencing depths are generally preferred as they provide more accurate estimates of gene expression levels. Finally, due to the high experimental noise in the process of sequencing RNA data, it is important to perform *replicates*. Replicates involve sequencing the same sample multiple times to ensure reproducibility and reliability of the results. In our case, the cell-line RNA-seq data was sequenced in triplicate for each sample. For a more extensive commentary on so-called *good practices* for dealing with RNA-seq data, we highly recommend readers check [116], where a marvelous step-by-step guide is provided for approaching the challenging processes related to experimental design and quality control.

4.2.2 Statistical Analysis of RNA-seq Data

Once the gene count matrix has been obtained, the next step is to perform **Differential Expression Analysis (DEA)** to identify genes that are differentially expressed in a statistically significant manner. In the context of our study, we are interested in identifying genes that are differentially expressed between **1)** tamoxifen-resistant and tamoxifen-sensitive MCF7 cells, and **2)** tamoxifen-resistant and tamoxifen-sensitive breast cancer patients. The DEA process involves several steps summarized in Figure 4.4 and detailed below:

1. **Count Matrix Pre-Processing:** The first step in the DEA process is to pre-process the gene count matrix. This mainly involves dealing with missing entries of data and filtering out genes with low expression levels. Missing records can be removed or imputed using the usual *roaster* of techniques, such as mean/median imputation, univariate methods, or by using more advanced multivariate approaches such as MICE [125] or missForest [126]. Internally, these methods infer the missing entries based on the observed data by iteratively estimating and filling missing values based on observed relationships within the dataset. MICE uses a sequential approach where each variable with missing values is modeled as a function of other variables, typically using regression models, in a chained iterative process until convergence. In the same fashion, MissForest employs random forest predictors to predict missing values by leveraging the interactions between variables. Luckily, in our case, the gene count matrix is complete, and no missing data is present in either the cell-line or patient RNA-seq data. Likewise, genes with low expression levels can be filtered out using a threshold value, such as 30 in our case. To contextualize how important this simple step is, in the case of the TCGA dataset, applying this sort of naive filtering lowers the number of genes from 36,791 to 24,344.

2. **Normalization:** The next step in the DEA process is to **normalize** the gene count matrix. Normalization is a crucial step in RNA-seq data analysis as it corrects for differences in sequencing depth and gene length between samples. Without normalization, the interpretation of gene expressions becomes unfathomable, as differences in counts — e.g., emerging from technical variability — might be erroneously attributed to nonexistent biological differences. In order to ensure that gene expression levels are comparable between samples, several normalization methods have been proposed in the literature [127]. These methods are summarized in Table 4.1[1].

   Generally, let $K_{G\times S}$ be the gene count matrix, where G is the number of genes and S is the number of samples. Each element K_{ij} represents the raw count of reads mapped to gene i in sample j. The goal of normalization is to transform K into a normalized count matrix N such that the gene expression levels are comparable across samples. In practice, this is accomplished by dividing each count by a scaling factor s_j , which is calculated based on the total number of reads in each sample:

$$N_{ij} = \frac{K_{ij}}{s_j} \tag{4.1}$$

   In our case, we will use the **Relative Log Expression (RLE)** method for normalization introduced in [132]. Originally proposed for the normalization of microarray data in the DEseq package, the RLE method has been adapted for RNA-seq data and is widely used in the field. The RLE method normalizes the gene count matrix by first computing the geometric mean for each gene across all samples, and then scaling each sample by the median of the values of the ratios of sample counts as in Eq. (4.2).

$$s_j = \underset{i\in G}{\text{median}}\left(\frac{K_{ij}}{\left(\prod_{k=1}^{S} K_{ik}\right)^{1/S}}\right) \tag{4.2}$$

[1]Please note that *normalization* here refers to the process of scaling the gene count matrix to ensure that gene expression levels are comparable across samples, not to be confused with the normalization of the data distribution in the context of Machine Learning. In practice, traditional ML-scaling techniques will be applied after the DEA process, once we begin training our models.

Table 4.1: Commonly Used RNA-seq Normalization Methods

	Sequencing Depth	Read Length	Within-Sample Comparisons	Between-Sample Comparisons	Reference
Counts Per Million (CPM)	✓	✗	✓	✗	
Transcripts Per Million (TPM)	✓	✓	✓	✗	[128]
Fragments Per Kilobase Million (FPKM)	✓	✓	✓	✗	[129]
Trimmed Mean of M-values (TMM)	✓	✗	✗	✓	[130]
Count Adjusted w/ TMM Factors (CTF)	✗	✗	✗	✓	[131]
Relative Log Expression (RLE)	✓	✗	✓	✓	[132]

As depicted in Table 4.1, the RLE method is particularly useful for normalizing RNA-seq data as it corrects for differences in sequencing depth, as well as ensures that gene expression levels are comparable both *within* and *between* samples, making it ideal for our study of tamoxifen resistance in MCF7 cells.

3. **Statistical Analysis:** After normalization, the next step is to perform a statistical analysis to identify those genes that are differentially expressed between tamoxifen-resistant and tamoxifen-sensitive MCF7 cells, as well as between tamoxifen-resistant and tamoxifen-sensitive breast cancer patients. At its core, **Differential Expression Analysis** (DEA) is about *quantifying* the differences in gene expression levels between two or more conditions — such as treatment groups or disease states — thus providing valuable insights into the underlying biological mechanisms governing cellular processes. In fact, it allows us to **(1)** identify *which genes* exhibit statistically significant changes in their expression levels, **(2)** *quantify* the magnitude of these changes; hopefully uncovering potential biomarkers implicated in some biological processes.

   Typically, we assume that the count-data obtained after the sequencing process follows a **negative binomial distribution** as in Eq. (4.3).

$$K_{ij} \sim \text{NB}(\mu_{i,j}, \alpha_i) \tag{4.3}$$

   where $\mu_{i,j}$ is the mean expression level of gene i in sample j, and α_i is the gene-specific dispersion parameter for gene i. The negative binomial distribution is a common distribution for modeling count data as it accounts for overdispersion, which is often observed in RNA-seq data due to biological variability and technical noise [133]. Of course, in practice, these parameters $\mu_{i,j}$ and α_i are not known and need to be estimated from the data. Likewise, a further statistical test needs to identify if the changes in expression are *statistically significant* or not.

First, the mean expression levels $\mu_{i,j}$ are estimated using the sample-specific scaling factors s_j obtained during the normalization step in Eq. (4.2). That is, the expected value of the observed counts for gene i in sample j is given by Eq. (4.4) as in:

$$\mu_{ij} = s_j q_{ij} \tag{4.4}$$

where q_{ij} is a quantity proportional to the concentration of RNA fragments from gene i in sample j [132]. For the dispersion parameter α_i, it is commonly estimated by using the **maximum likelihood estimation** (MLE) method in Eq. (4.5).

$$\alpha_i = \frac{\text{Var}(K_{ij}) - \mu_{ij}}{\mu_{ij}^2} \tag{4.5}$$

To test for differential expression between conditions, a **generalized linear model** (GLM) is employed with the link function in Eq. (4.6) below:

$$\log_2(q_{ij}) = \sum_k x_{jk}\beta_{ik} \tag{4.6}$$

where x_{jk} is the design matrix for the groups to compare. In the case of a two-group comparison, as in our case, the design matrix elements indicate whether a sample j belongs to the altered or control group. The GLM model returns a set of estimated coefficients $\hat{\beta}_{ik}$ indicating the overall expression strength of each gene i, from which we can derive the **Fold-Changes** (FCs) in gene expression levels between the two groups. The FC is calculated as the ratio of the mean expression levels against the control group. Subsequently, the obtained fold-changes need to be tested for statistical significance. This is typically done using a **Wald test** [134] between the two groups. The Wald test statistic is calculated as in Eq. (4.7).

$$W_{i,g_1-g_2} = \frac{\hat{\beta}_{g_1} - \hat{\beta}_{g_2}}{\sqrt{\text{Var}\left(\hat{\beta}_{g_1} - \hat{\beta}_{g_2}\right)}} \tag{4.7}$$

Finally, the p-values from the Wald test are adjusted for multiple testing [135] to obtain the **False Discovery Rate** (FDR) score for significance.

4. **Significance Filtering:** From the previous step, we have obtained the two reference metrics of DEA, i.e., the fold-changes (usually expressed in its logarithmic form), and the false-discovery rate. To further reduce the dimensionality of the problem, and to remove potentially non-significant genes, we apply some filtering based on these metrics. In our case, we will filter out genes with an absolute log-fold change under 0.5 $\left(|\log_2 \text{FC}| < 0.5\right)$ and an FDR score smaller than 0.1 (FDR < 0.1). This is a common practice in DEA to ensure that only the most significant genes are retained for further analysis. In practice, this filtering step reduces the number of genes from tens of thousands to a few hundred, making the analysis more manageable and interpretable. To contextualize things once again, in the case of the TCGA dataset, applying this filtering step reduces the number of genes from 24,344 to a mere 144.

As such, all the steps described above are applied to both the cell-line and patient RNA-seq data, separately. As a result, we obtain a subset of genes that are differentially expressed between tamoxifen-resistant

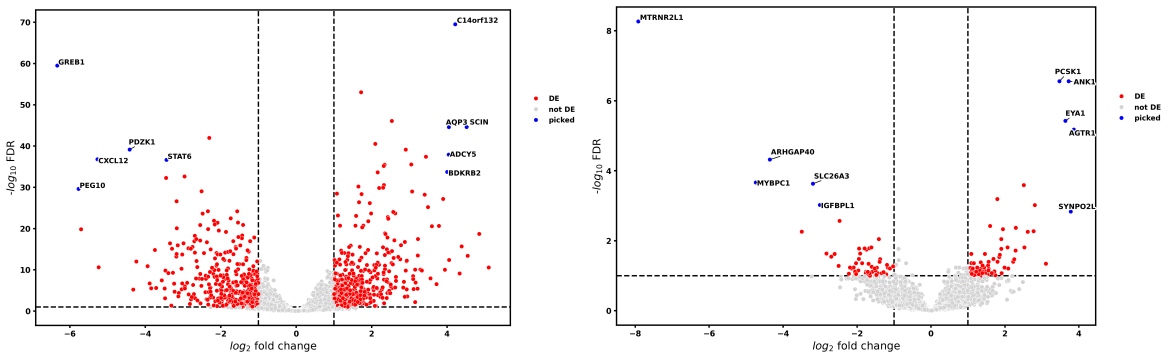

Figure 4.5: Volcano Plot of Statistically Significant Differentially Expressed Genes.
(*Left:* MCF7 Cell-Lines, *Right:* TCGA Patients)

and tamoxifen-sensitive MCF7 cells, as well as between tamoxifen-resistant and tamoxifen-sensitive breast cancer patients. Visually, these results are often represented using **Volcano Plots**, where the x-axis represents the log-fold change in gene expression levels, and the y-axis represents the $-\log_{10}$ of the FDR score. The genes that are significantly differentially expressed are highlighted in red, while the non-significant genes are shown in gray. Figure 4.5 shows the results of the DEA analysis for both cell-line and patient data (note that the blue points correspond to the top 10 most differentially expressed genes in each dataset, but this is just for illustrative purposes).

In any case, this process of applying DEA to each dataset separately serves as a preliminary step for identifying potential biomarkers for tamoxifen resistance in MCF7 cells and breast cancer patients. In the next section, we will discuss how to combine the results of this separate analysis and accomplish the integration of cell-line and patient RNA-seq data to hopefully uncover some potential genetic biomarkers responsible for this biological phenomenon.

4.3 Integration of Cell-Line & Patient RNA-seq Data for Genetic Biomarker Identification

Following the pipeline presented in the section above, we performed differential expression analysis on the cell lines and patients RNA-seq data, separately. The results of the DEA analysis were used to identify potential candidate genes related to resistance in each separate cohort of sequencing data. In order to combine both sources of data, patients with a positive clinical response in the TCGA dataset were considered comparable to control cells in the MCF7 cell-lines, whereas resistant patients in the TCGA group of patients were considered comparable with TamR cells in the MCF7 cell-lines.

In practice, the integration process of our *multi-source* sequencing data is quite simple. First, for each two-group comparison, genes with an expression level under the set threshold (i.e., 30 in our case) were removed, and RLE normalization as in Eq. (4.2) was applied to the resulting gene count matrix (Step 1 in Figure 4.6). Then, the results were filtered according to the selection criteria presented previously, i.e., $\left(|\log_2 \text{FC}| > 0.5\right)$ and (FDR < 0.1) (Step 2 in Figure 4.6). Finally, because of our assumption that patients with a positive clinical response can be compared to control MCF7 cells, we select only those genes that are expressed in the same direction in both cases: i.e., those that are either differentially *over-expressed* or *under-expressed* (Step 3 in Figure

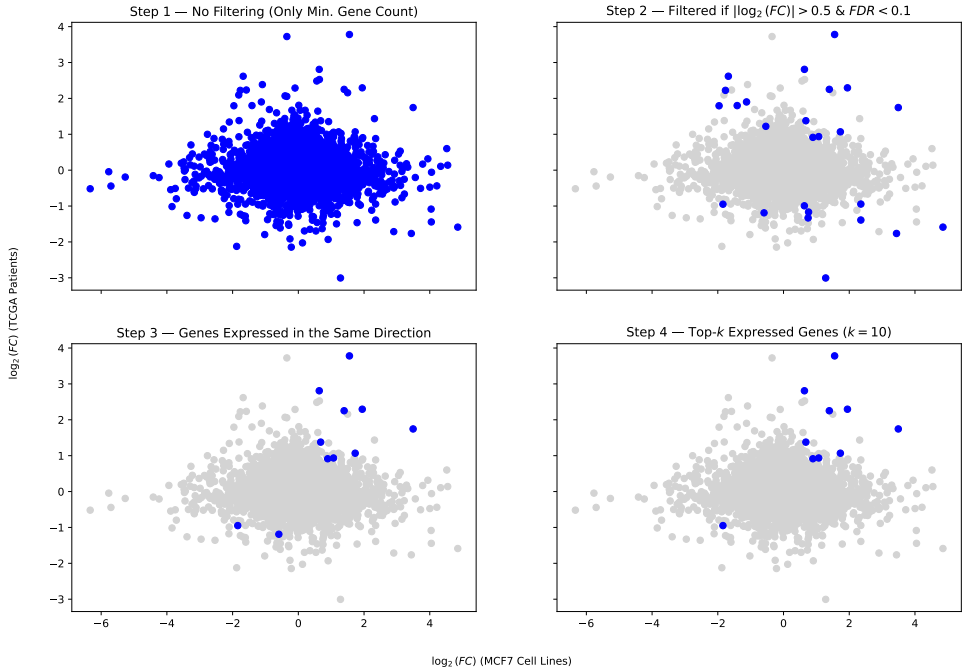

Figure 4.6: Distribution of Genes in the Joint Analysis of Cell-Line & Patient Data

4.6). As a final note, by applying this process, the number of relevant genes goes down to an astonishing 11, a far more *reasonable* number of features to handle than the original 36,791.[2]

4.4 Python in Hamiltonian for Computational Statistics

Introducing pyHaiCS, a Python library for Hamiltonian-based Monte Carlo methods tailored towards practical applications in *computational statistics*. From sampling complex probability distributions, to approximating complex integrals — such as in the context of Bayesian inference — pyHaiCS is designed to be fast, flexible, and easy to use, with a focus on providing a user-friendly interface for researchers and practitioners while also offering users a variety of *advanced features*.

Although currently in development, our library implements a wide range of *sampling algorithms* — including single-chain and multi-chain Hamiltoninan Monte Carlo (HMC) and Generalized HMC (GHMC); a variety of numerical schemes for the *integration* of the simulated Hamiltonian dynamics (including a generalized version of Multi-Stage Splitting integrators), or a novel *adaptive* algorithm — Adaptive Integration Approach in Computational Statistics (s-AIA) — for the automatic tuning of the parameters of both the numerical integrator and the sampler.

Likewise, several utilities for *diagnosing* the convergence and efficiency of the sampling process, as well as *multidisciplinary* benchmarks — ranging from simple toy problems such as sampling from specific distribu-

[2]Regarding the motivation behind selecting *only* the top-10 most expressive genes (Step 4 in Figure 4.6), this is actually both biologically motivated — some members of the Cancer Heterogeneity Lab at bioGUNE raised their concerns about that particular gene being included in our *prognostic signature* as it may be actually involved in other biochemical processes — as well as computationally motivated by some further tasks we performed in our study of anti-cancer therapy modeling (this however, is well outside the scope of this thesis).

Figure 4.7: Logo of the pyHaiCS Library

tions, to more complex real-world applications in the fields of computational biology, Bayesian modeling, or physics — are provided.

The main features of pyHaiCS[3] — as summarized in Figure 4.8 — include its:

- **Efficient Implementation:** pyHaiCS is built on top of the JAX library developed by Google [11, 12], which provides **automatic differentiation** for computing gradients and Hessians, and Just-In-Time (JIT) *compilation* for fast numerical computations. Additionally, the library is designed to take advantage of multi-core CPUs, GPUs, or even TPUs for *accelerated* sampling, and to be highly **parallelizable** (e.g., by running each chain of multi-chain HMC in a separate CPU core/thread in the GPU).
- **User-Friendly Interface:** The library is designed to be easy to use, with a simple and intuitive API that abstracts away the complexities of Hamiltonian Monte Carlo (HMC) and related algorithms. Users can define their own potential functions and priors, and run sampling algorithms with just a few lines of code.
- **Integration with Existing Tools:** The library is designed to be **easily integrated** with other Python libraries, such as NumPy [136], SciPy [137], and Scikit-Learn [138]. This allows users to capitalize on existing tools and workflows, and build on top of the rich ecosystem of scientific computing in Python. Hence, users can easily incorporate pyHaiCS into their existing Machine Learning workflows, and use it for tasks such as inference, model selection, or parameter estimation in the context of Bayesian modeling.
- **Advanced Features:** pyHaiCS supports a variety of Hamiltonian-inspired sampling algorithms, including single-chain and multi-chain HMC (and GHMC), generalized k-th stage Multi-Stage Splitting integrators, and adaptive integration schemes (such as s-AIA).

Additionally, at the time of this writing, and to the best of our knowledge, there are no available *open-source* projects for bayesian programming in Python aside from PyMC: a library for bayesian statistical modeling [139]. Although PyMC seeks to address a much broader scope than pyHaiCS — such as including Variational Inference models, tools for performing inference on ordinary differential equations (ODEs), or even Gaussian Processes — its current implementation of Markov-Chain Monte Carlo methods (such as HMC) is tediously slow, allows for very little tuning, lacks many *state-of-the-art* advancements in the field (such as in terms of more advanced samplers and integrators), and has an overwhelming API — mainly due to its symbolic approach to priors and likelihoods — which makes it hard to integrate within existing ML pipelines.

[3] Library Available at https://github.com/miguelfrndz/pyHaiCS. Official Documentation Available at https://pyhaics.github.io/

Figure 4.8: Summary of the Features in the pyHaiCS Ecosystem

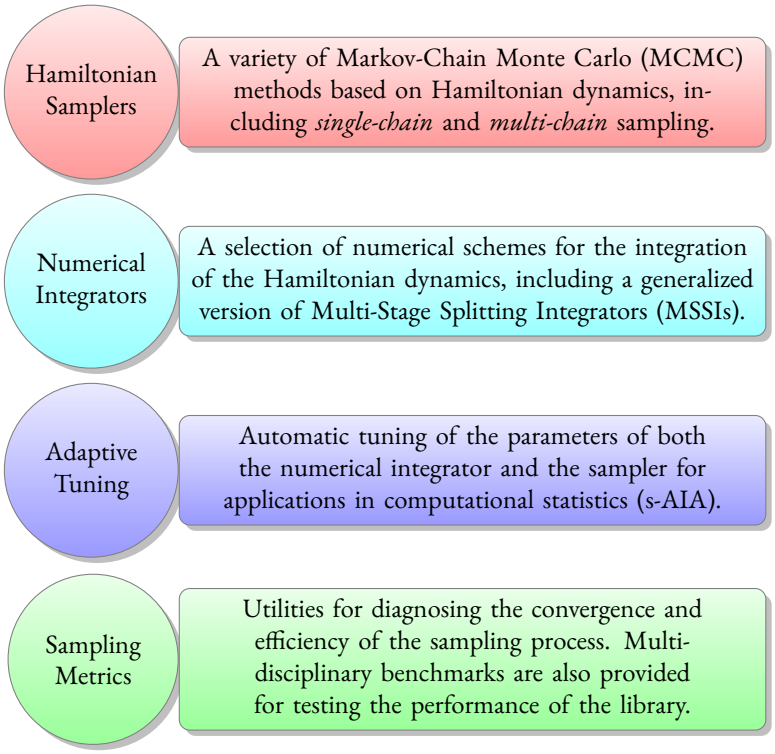

Figure 4.9: General Features of pyHaiCS

In order to provide a functional and easy-to-use library, and especially to ensure that our code can be easily integrated into existing workflows, we have designed pyHaiCS with a simple rule in mind: *Objects are specified by interface, not by inheritance*. That is, much alike Scikit-Learn, inheritance is not enforced; and instead, code conventions provide a consistent interface for all samplers, integrators, and utilities. This allows for a more flexible and modular design, and makes it easier for users to extend the library with their own custom implementations. As Scikit's design around making all estimators have a consistent `fit` and `predict` interface, pyHaiCS follows a similar approach, but with a focus on Hamiltonian Monte Carlo methods and its related algorithms. For instance, all integrators in pyHaiCS have a consistent `integrate` method, which takes as input the potential function, the initial state, and the parameters of the integrator, and returns the final state of the system after the integration process. This consistent interface makes it easy for users to switch between different integrators, or to implement their own custom ones, without having to worry about the underlying details of the implementation. Moreover, pyHaiCS is designed to be highly modular, with each component of the library being self-contained and independent of the others, as well as being easily extensible and customizable. As a further point of strength, our library handles all **auto-differentiation**, such as potential gradients and Hessians, through the JAX library, which provides a fast and efficient way to compute gradients as well as a higher level of abstraction for the user to focus on the actual problem at hand. By only defining the **potential** function of the Hamiltonian, the user can easily run the sampler and obtain the posterior distribution of the parameters of interest. As an example of the *ease-of-use* of pyHaiCS, Listing 1 shows a simple example of defining a Bayesian Logistic Regression (BLR) model.

Regarding the actual features implemented in pyHaiCS, and the general organization of its API, Figure 4.9 provides a high-level overview of the main components of the library, while Figure 4.10 shows a tree visual-

Listing 1 Example of How to Run a Bayesian Logistic Regression Model in pyHaiCS

```
# Step 1 - Define the BLR model
@jax.jit
def model_fn(x, params):
    return jax.nn.sigmoid(jnp.matmul(x, params))

# Step 2 - Define the log-prior and log-likelihood
@jax.jit
def log_prior_fn(params):
    return jnp.sum(jax.scipy.stats.norm.logpdf(params))

@jax.jit
def log_likelihood_fn(x, y, params):
    preds = model_fn(x, params)
    return jnp.sum(y * jnp.log(preds) + (1 - y) * jnp.log(1 - preds))

# Step 3 - Define the log-posterior (remember, the oppositve of the potential)
@jax.jit
def log_posterior_fn(x, y, params):
    return log_prior_fn(params) + log_likelihood_fn(x, y, params)

# Initialize the model parameters (including intercept term)
key = jax.random.PRNGKey(42)
mean_vector, cov_mat = jnp.zeros(X_train.shape[1]), jnp.eye(X_train.shape[1])
params = jax.random.multivariate_normal(key, mean_vector, cov_mat)

# HMC for posterior sampling
params_samples = haics.samplers.hamiltonian.HMC(params,
                            potential_args = (X_train, y_train),
                            n_samples = 1000, burn_in = 200,
                            step_size = 1e-3, n_steps = 100,
                            potential = neg_log_posterior_fn,
                            mass_matrix = jnp.eye(X_train.shape[1]),
                            integrator = haics.integrators.VerletIntegrator(),
                            RNG_key = key)

# Average across chains
params_samples = jnp.mean(params_samples, axis = 0)

# Make predictions using the samples
preds = jax.vmap(lambda params: model_fn(X_test, params))(params_samples)
mean_preds = jnp.mean(preds, axis = 0)
```

ization of the features implemented in pyHaiCS. As can be seen, the library is organized around four main components: *Hamiltonian Samplers*, *Numerical Integrators*, *Adaptive Tuning*, and *Sampling Metrics*. Each of these components is further divided into sub-components, such as the different samplers implemented in the library (e.g., HMC, GHMC, and the yet to be implemented, MMHMC), the numerical integrators (such as

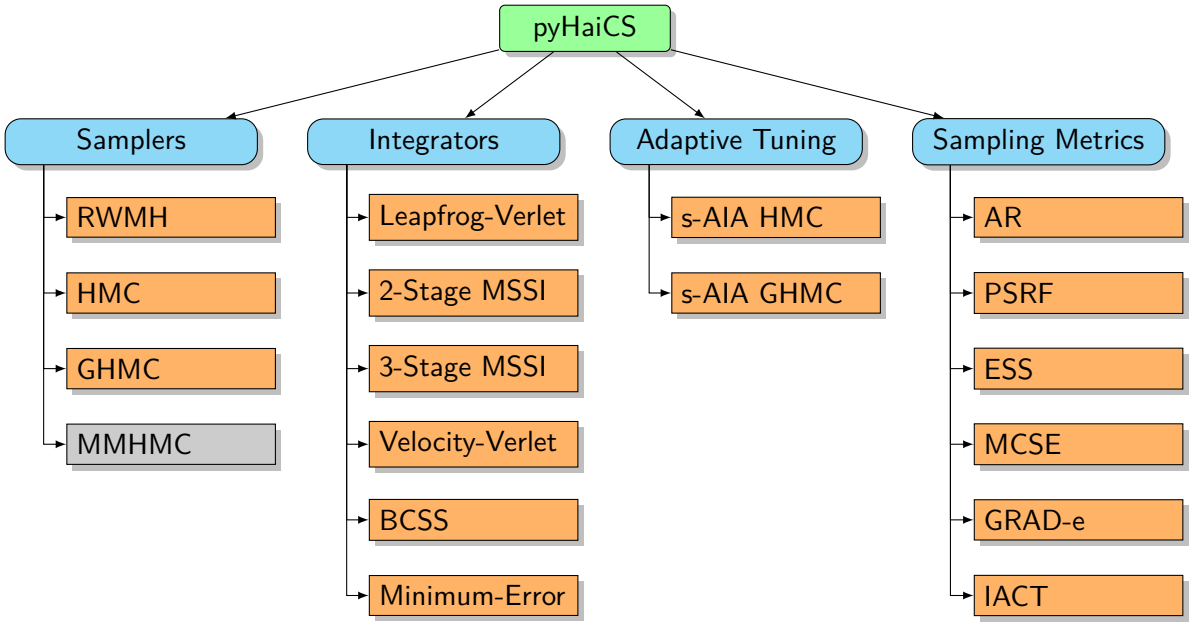

Figure 4.10: Tree Visualization of Features in pyHaiCS (In Gray, Features Yet to be Implemented)

variants of Velocity-Verlet, and 2-Stage and 3-Stage MSSIs), or the s-AIA adaptive tuning scheme. The library also includes a variety of sampling metrics for diagnosing the convergence and efficiency of the sampling process, as well as multidisciplinary benchmarks (and code examples) for testing the performance of the library. These experimental models are presented in the subsection below.

Experimental Models Included in pyHaiCS

In this section, we present a classical set of benchmarking scenarios for comparing the performance and behavior of the different Hamiltonian-based estimators based on their following intrinsic properties (remember Table 3.2):

1. **Correlation** in the Samples.
2. **Reversibility** of the Chain.
3. Influence of the **Importance Sampling** Re-Weighting (e.g., for MMHMC).

These benchmarks, provided in the pyHaiCS repository, include:

- **Banana-Shaped Distribution:** Given data $\{y_k\}_{k=1}^K$, we sample from the **banana-shaped posterior distribution** [50, 140] of the parameter $\boldsymbol{\theta} = (\theta_1, \theta_2)$ for which the likelihood and prior distributions are respectively given as:

$$y_k|\boldsymbol{\theta} \sim \mathcal{N}(\theta_1 + \theta_2^2, \sigma_y^2), \quad k = 1, 2, \dots, K \tag{4.8}$$

$$\theta_1, \theta_2 \sim \mathcal{N}(0, \sigma_\theta^2) \tag{4.9}$$

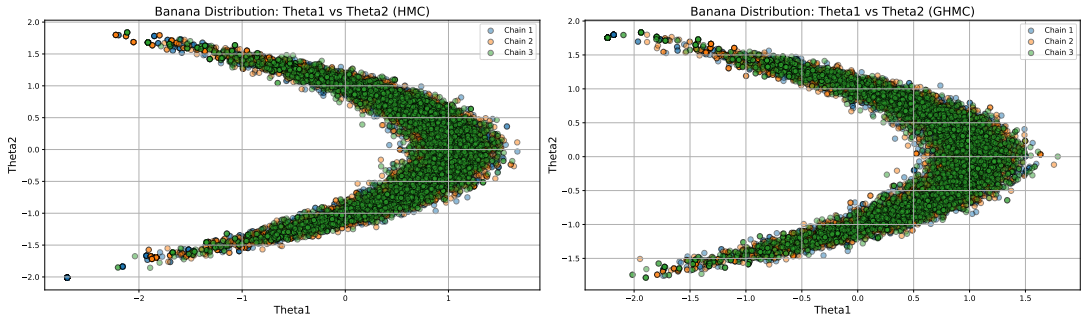

Figure 4.11: Exploration of Space in Sampling from a Banana-Shaped Distribution

The sample data are generated with $\theta_1 + \theta_2^2 = 1, \sigma_y = 2, \sigma_\theta = 1$. Then, the *potential* function is given by:

$$U(\boldsymbol{\theta}) = \frac{1}{2\sigma_y^2}\sum_{k=1}^{K}(y_k - \theta_1 - \theta_2^2)^2 + \log\left(\sigma_\theta^2 \sigma_y^{100}\right) + \frac{1}{2\sigma_\theta^2}(\theta_1^2 + \theta_2^2) \tag{4.10}$$

The resulting samples in Figure 4.11 were produced for 10 independent chains, each with 5000 burn-in iterations, 5000 samples, $L = 14$ integration steps, a step-size of $\varepsilon = 1/9$, and a momentum noise of $\phi = 0.5$.

- **Multivariate Gaussian Distribution:** Let's take a step up and look at a more *serious* example; sampling from a D-dimensional **Multivariate Gaussian Distribution** $\mathcal{N}(\mathbf{0}, \Sigma)$ [50, 72], where the precision matrix Σ^{-1} is generated from a Wishart distribution [141].

  In this case, we will take $D = 1000$ dimensions and, for strictly computational reasons, we take the covariance matrix to be diagonal with

  $$\Sigma_{ii} = \sigma_i^2 \tag{4.11}$$

  where σ_i^2 is the i-th smallest eigenvalue of the original covariance matrix. Thus, the potential function in this case is defined as:

  $$U(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T\Sigma^{-1}\boldsymbol{\theta} \tag{4.12}$$

- **Bayesian Logistic Regression (BLR):** As introduced in Section 3.2, Bayesian Logistic Regression (BLR) is the probabilistic extension of the traditional *point-estimate* logistic regression model by incorporating a prior distribution over the parameters of the model. In the BLR model, given K data instances $\{\mathbf{x}_k, y_k\}_{k=1}^K$ where $\mathbf{x}_k = (1, x_1, \ldots, x_D)$ are vectors of D covariates and $y_k \in \{0, 1\}$ are the binary responses. The probability of a particular outcome is linked to the linear predictor function through the *logit* function as in:

  $$p(y_k|\mathbf{x}_k, \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^T\mathbf{x}_k) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T\mathbf{x}_k)} \tag{4.13}$$

  $$\boldsymbol{\theta}^T\mathbf{x}_k \equiv \text{logit}(p_k) = \log\left(\frac{p_k}{1 - p_k}\right) = \theta_0 + \theta_1 x_{1,k} + \ldots \theta_D x_{D,k} \tag{4.14}$$

Table 4.2: Datasets Used for Benchmarking the BLR Model

Dataset	D	K	Reference
German	25	1000	German Credit Dataset
Sonar	61	208	Sonar Dataset
Musk	167	476	Musk Dataset (Version 1)
Secom	444	1567	SECOM Dataset

where $\boldsymbol{\theta} = (\theta_0, \theta_1, \dots, \theta_D)^T$ are the parameters of the model, with the term θ_0 usually denoted as the *intercept*. The prior distribution over the parameters $\boldsymbol{\theta}$ is usually chosen to be a Multivariate Gaussian distribution as:

$$\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \text{Usually } \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+1}) \tag{4.15}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{D+1}$ is the *mean* vector, $\boldsymbol{\Sigma} \in \mathbb{R}^{D+1}$ is the *covariance* matrix, $\mathbf{0}$ is the zero vector and $\mathbf{I}_{D+1}$ is the identity matrix of order $D + 1$.

In order to simplify the notation, let us define the *vectorized* response variable $\mathbf{y} = (y_1, \dots, y_K)$, and the *design* matrix $X \in \mathbb{R}^{K,D}$ as the input to the model:

$$X = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,D} \\ 1 & x_{2,1} & \dots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{K,1} & \dots & x_{K,D} \end{pmatrix} \tag{4.16}$$

The likelihood of the data is given by the product of the Bernoulli distributions as:

$$\mathcal{L}(\mathbf{y}|X, \boldsymbol{\theta}) \equiv p(\mathbf{y}|X, \boldsymbol{\theta}) = \prod_{k=1}^{K} p(y_k|X_k, \boldsymbol{\theta}) = \prod_{k=1}^{K} \left(\frac{\exp(X_k\boldsymbol{\theta})}{1 + \exp(X_k\boldsymbol{\theta})} \right)^{y_k} \left(\frac{1}{1 + \exp(X_k\boldsymbol{\theta})} \right)^{1-y_k} \tag{4.17}$$

where $X_k = (1, x_{k,1}, \dots, x_{k,D})$ is the k-th entry *row* vector of the design matrix X.

Then, the potential function can be expressed as:

$$U(\boldsymbol{\theta}) = -\sum_{k=1}^{K} \left[y_k \cdot X_k\boldsymbol{\theta} - \log\left(1 + \exp(X_k\boldsymbol{\theta})\right) \right] + \frac{1}{2\alpha} \sum_{i=1}^{D} \theta_i^2 \tag{4.18}$$

For consistency with the benchmarks provided in [50], the pyHaiCS project includes the datasets in Table 4.2. All of them are publicly available *online* and their reference is also provided in the table.

- **Dynamic COVID-19 Epidemiological Models:** Another interesting application of Hamiltonian-based Monte Carlo is proposed in [142]. In their work, a SEIR (Susceptible-Exposed-Infectious-Remove) dynamic *compartmental* (i.e., by splitting the population into disjoint compartments and defining a transition flow between compartments) *mechanistic* (i.e., where the disease dynamics are purely governed by

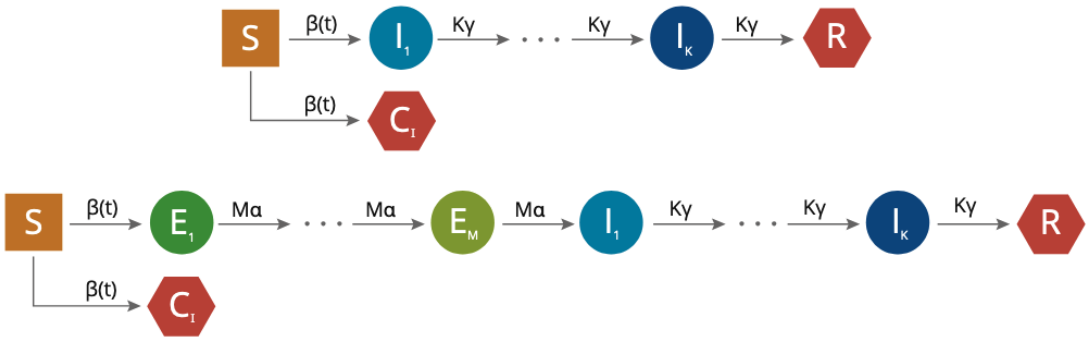

Figure 4.12: Flow Diagram for the SE_MI_KR Compartmental Model w/ Transmission Rate $\beta(t)$

differential equations) epidemiological model — with a time-dependent transmission rate parametrized using Bayesian P-splines — is applied to modeling the COVID-19 incidence data in the Basque Country (Spain).

More specifically, a SE_MI_KR model is defined with a time-depdendent transmission rate $\beta(t)$ (i.e., the average number of contacts per person, per time unit, multiplied by the probability of infection in a contact) parametrized by Bayesian P-Splines.

The SE_MI_KR model consists of the following compartments:

- S corresponds to the number of individuals that are **susceptible** to be infected.
- $E_1, \ldots, E_M$ reflect the number of individuals at different stages of **exposure** (i.e., infected but not yet infectious). The average time spent by an individual being exposed is $\frac{1}{\alpha}$.
- $I_1, \ldots, I_K$ reflect the number of **infectious** individuals. The average time spent by an individual being infectious is $\frac{1}{\gamma}$.
- R represents the number of individuals **removed** from the pool of susceptible individuals (either by death or by recovery).
- C_I is a **counter** of the total number of individuals that have been infected.
- $\beta(t)$ is the time-dependent transmission rate.

To model the transmission rate $\beta(t)$ a set of B-splines is used such that:

$$\log \beta(t) = \sum_{i=1}^{m} \beta_i B_i(t) \tag{4.19}$$

where $\{B_i(t)\}_{i=1}^m$ form a B-spline basis over the time interval $[t_0, t_1]$, with $m = q + d - 1$ (q is the number of knots, d is the degree of the polynomials of the B-splines); and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ is a vector of coefficients.

Likewise, the SE_MI_KR model is governed by the following system of ODEs:

$$\dot{S}(t) = -\beta(t)S(t)\frac{I(t)}{N}, \tag{4.20}$$

$$\dot{E}_1(t) = \exp\left(\sum_{i=1}^{m}\beta_i B_i(t)\right)S(t)\frac{I(t)}{N} - M\alpha E_1(t), \qquad \dot{E}_M(t) = M\alpha E_{M-1}(t) - M\alpha E_M(t), \tag{4.21}$$

$$\dot{I}_1(t) = M\alpha E_M(t) - K\gamma I_1(t), \qquad \dot{I}_K(t) = K\gamma I_{K-1}(t) - K\gamma I_K(t), \tag{4.22}$$

$$\dot{R}(t) = K\gamma I_K(t), \tag{4.23}$$

$$\dot{C}_I(t) = \exp\left(\sum_{i=1}^{m}\beta_i B_i(t)\right)S(t)\frac{I(t)}{N} \tag{4.24}$$

with the following constraints:

$$\begin{cases} S(t_0) = N - E_0, E_1(t_0) = C_I(t_0) = E_0 \\ E_2(t_0) = \cdots = E_M(t_0) = I_1(t_0) = \cdots = I_K(t_0) = R(t_0) = 0 \\ E(t) = \sum_{i=1}^{M} E_i(t) \\ I(t) = \sum_{j=1}^{K} I_j(t) \\ N = S(t) + E(t) + I(t) + R(t) \end{cases} \tag{4.25}$$

Note that the number of newly infected individuals at time t is given by $\beta(t)S(t)I(t)/N$.

Additionally, the total number of new individuals infected at day t is given by:

$$C(t) = C_I(t) - C_I(t-1) \tag{4.26}$$

Where our *predicted* (corrected) daily incidence as in Figure 4.13, is sampled from the following distribution:

$$\frac{\tilde{C}(t)}{\eta(t)} \sim \text{Neg. Binom.}(C(t), \phi) \tag{4.27}$$

To simplify the notation, we will use the following notations to represent the *state* $\mathbf{y}(t)$ and *parameters* $\mathbf{p}$ of our model respectively:

$$\mathbf{y}(t) = [S(t), E_1(t), \ldots, E_M(t), I_1, \ldots, I_K(t), R(t), C_I(t)]^T \tag{4.28}$$

$$\mathbf{p} = \left[\alpha, \gamma, E_0, \phi^{-1}, \tau, \boldsymbol{\beta}\right]^T \tag{4.29}$$

- **Talbot Physical Effect:** Lastly, in collaboration with the *Linear and Non-Linear Waves* group[4] at BCAM, we have included a final benchmark related to the analysis of Partial Differential Equations (PDEs) in the context of the phenomenon occurring when a plane light wave is diffracted by an infi-

[4]Thank you to Gabriel Ybarra (BCAM) and Luis Vega (BCAM, University of the Basque Country). The results of this work are currently under submission but have not been officially published yet. A preprint is available in [143]

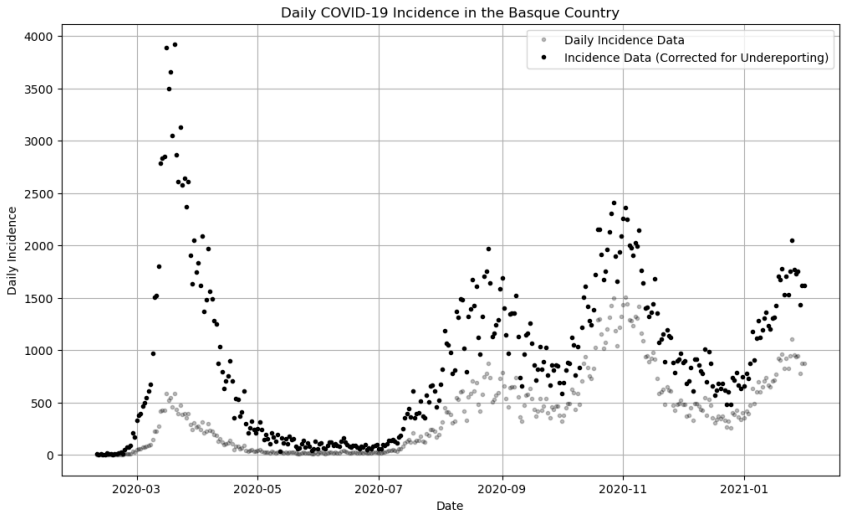

Figure 4.13: Daily COVID-19 Incidence in the Basque Country (Before/After Correction)

nite set of equally spaced slits (the *grating*, with distance d between the slits). That is, we wish to find solutions to the following differential equation:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \tag{4.30}$$

in the domain $0 \leq x \leq \frac{d}{2}, z \geq 0, t \geq 0$ under the *border* conditions in x:

$$\left.\frac{\partial u}{\partial x}\right|_{x=0} = \left.\frac{\partial u}{\partial x}\right|_{x=d/2} = 0 \tag{4.31}$$

the boundary conditions in z:

$$u(t, x, z = 0) = f(t, x) = \sin(\omega t)\theta(t)\,\chi\left(\frac{x}{w}\right) \tag{4.32}$$

and the initial conditions:

$$u(t = 0, x, z) = 0, \qquad \left.\frac{\partial u}{\partial t}\right|_{t=0} = 0 \tag{4.33}$$

Without getting into details of the (*long*) derivation process, the solution can be expressed in closed-form as in Eq. (4.34) below:

$$u(t, x, z) = \sum_n g_n \left(\sin \omega(t - z) - k_n z \int_z^t \frac{J_1(k_n\sqrt{\tau^2 - z^2})}{\sqrt{\tau^2 - z^2}} \sin \omega(t - \tau)\, d\tau \right) \theta(t - z) \cos k_n x \tag{4.34}$$

As can be seen, solving the problem entails numerically approximating the complex integral in magenta, which involves **(1)** a Bessel function of the first kind, **(2)** an avoidable singularity as $\tau \to z$, **(3)** a composition of two highly *oscillatory* functions. In order to circumvent the limitations of traditional solvers,

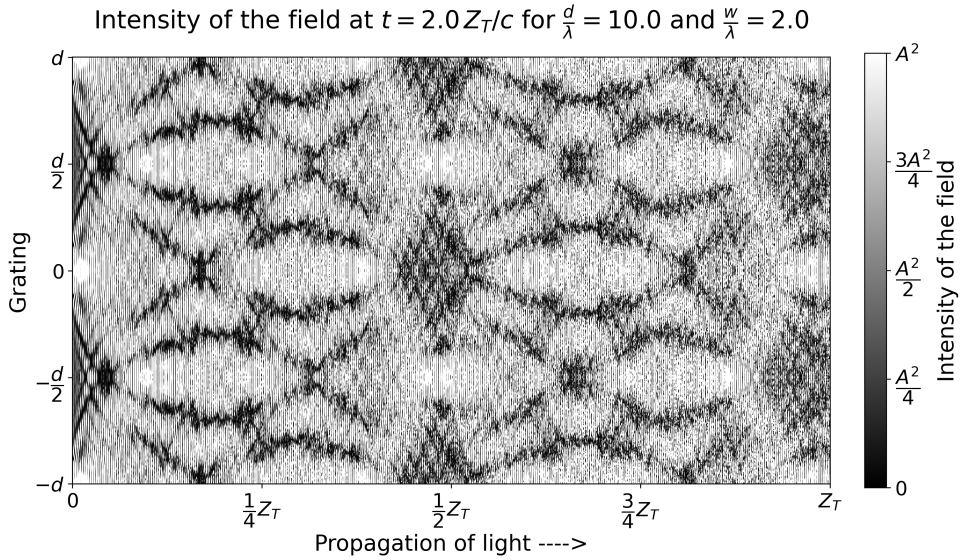

Figure 4.14: Optical Talbot Effect of a Wave (Field Intensity Approximated w/ Monte Carlo)

pyHaiCS was employed to numerically approximate the integral. An example of the resulting fields is provided in Figure 4.14.

4.5 Bayesian Logistic Regression & Other Resistance Models

Once we have obtained the *refined* list of potential biomarkers from the joint DEA-based analysis of patients and cell lines in Section 4.3, we can proceed to the next step: the development of our tamoxifen resistance models. More specifically, our goal is to train a model that can predict the resistance of a patient to tamoxifen treatment based on the expression levels of the selected genes. More generally, given the expression levels of the selected genes $\mathbf{x} = (x_1, \dots, x_D)$, we want to predict the probability of resistance to tamoxifen treatment $p(y = 1|\mathbf{x})$. This can be achieved by training a *binary classifier* on the data, but first, we need to *normalize* the expression levels of the genes to ensure that they are on the same scale. This is done by applying the *z-score* transformation to the expression levels of the genes, which ensures that the expression levels are centered around zero and have a standard deviation of one. The z-score transformation is given by:

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \tag{4.35}$$

where x_i is the expression level of gene i, μ_i is the mean expression level of gene i, and σ_i is the standard deviation of the expression level of gene i. Once the expression levels of the genes have been normalized, we can proceed to train the binary classifier on the data.

In the following subsections, we will present the different resistance models that we have developed, starting with the Bayesian Logistic Regression (BLR) model, and then moving on to other approaches such as *shallow* classifiers, ensemble methods, and deep learning architectures. Finally, we will provide a brief overview of our *weight-perturbatory* approach for training a Bayesian Neural Network (BNN) to predict the resistance of patients to tamoxifen treatment.

4.5.1 Bayesian Logistic Regression

Our main resistance model in this work is the Bayesian Logistic Regression (BLR), more specifically, combined with Monte Carlo sampling methods based on Hamiltonian dynamics. Despite Sections 3.2 & 4.4 having already covered the theoretical aspects of the BLR model and its implementation in the pyHaiCS library — mainly the definition of the associated Hamiltonian potential —, and Listing 1 having shown how to run a BLR model in pyHaiCS, we will now delve into the details of the particular BLR implementation for predicting tamoxifen resistance in breast cancer patients.

By considering a Bayesian framework, we can incorporate prior information about the parameters of the model extracted from the lab-grown MCF7 cell-lines. From the results of the DEA analysis in Section 4.2, we set a normal prior for each gene i centered in the mean $\mu_i = \log_2 \mathrm{FC}_i$ value of each gene as in Eq. (4.36).

$$\theta_i \sim \mathcal{N}(\log_2 \mathrm{FC}_i, 2.5^2), \quad i = 1, \dots, D \tag{4.36}$$

where the *standard deviation* was chosen following the recommendations in [144].

4.5.2 Traditional Shallow Models

In addition to the BLR model above, we also trained a variety of traditional *shallow* classifiers on the data as a baseline for comparison. These models were trained using the Scikit-Learn library in Python, and the hyperparameters of the models were tuned using a *grid-search* and *stratified cross-validation*. Of course, these *point-estimate* models do not include any expert prior information on the parameters of the model, nor do they provide any measure of uncertainty in the predictions. However, they can still provide valuable insights into the data and serve as a benchmark for the performance of the BLR model. Among these models we find:

- **(Point-Estimate) Logistic Regression:** A traditional *vanilla* logistic regression model trained using the *maximum likelihood* method, such as with the L-BFGS second-order optimizer [145].

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\theta^T \mathbf{x})} \tag{4.37}$$

- **Support-Vector Classifier (SVC):** A support-vector classifier trained using both a *linear* and a *radial basis function* (RBF) kernel. The *linear* SVC model is trained by maximizing the margin between the two classes using a hyperplane defined by:

$$\mathbf{w}^T \mathbf{x} + b = 0 \tag{4.38}$$

which minimizes $\frac{1}{2}\|\mathbf{w}\|^2$ subject to $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i$. A model prediction is given by:

$$\hat{y} = \mathrm{sign}(\mathbf{w}^T \mathbf{x} + b) \tag{4.39}$$

The *RBF* kernel [146] however, is used to handle non-linear data by mapping it to a higher-dimensional space using the following kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \quad \gamma > 0 \tag{4.40}$$

Table 4.3: Popular Kernels for Support-Vector Machines

Kernel	Expression
Linear	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d$
RBF	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \lVert \mathbf{x}_i - \mathbf{x}_j \rVert^2\right)$
Sigmoid	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\kappa \mathbf{x}_i^T \mathbf{x}_j + c\right)$

where the output prediction of the model is now given by:

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{4.41}$$

where α_i are the Lagrange multipliers (obtained by solving the dual optimization problem), y_i are the labels, and b is the bias term. Of course, many other kernels $K(\mathbf{x}_i, \mathbf{x}_j)$ exist in the literature for different non-linear mappings. Some of the most popular ones have been summarized in Table 4.3.

- **Naive Bayes:** A simple probabilistic classifier based on the Bayes theorem with strong (naive) independence assumptions between the features [147]. The model is trained by estimating the likelihood of the features given the class and the prior probability of the class. The prediction is then made by selecting the class with the highest posterior probability. That is:

$$\hat{y} = \underset{y}{\text{argmax}}\, p(y) \prod_{i=1}^{D} p(x_i|y) \tag{4.42}$$

In our case, we considered a Gaussian Naive Bayes model, i.e., the likelihood is assumed to be Gaussian:

$$p(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_{y,i})^2}{2\sigma_y^2}\right) \tag{4.43}$$

where $\mu_{y,i}$ and σ_y^2 are the mean and variance of the feature x_i in class y respectively. These parameters are estimated from the training data using the maximum likelihood method.

4.5.3 Ensemble Methods

In addition to the traditional shallow models above, we also trained a variety of **ensemble methods** on the data. Ensemble methods combine multiple models to improve the predictive performance of the model. The idea behind ensemble methods is that, by combining multiple models, each with different strengths and weaknesses, we can create a more robust and accurate estimator. There are many different ensemble methods in the literature, but they can be broadly classified into three categories (or *paradigms*) as summarized in Figure 4.15:

1. **Bagging:** *Bagging* (bootstrap aggregation) is an ensemble method that works by training multiple models on different subsets of the data (with *replacement*) and then combining the predictions of the models.

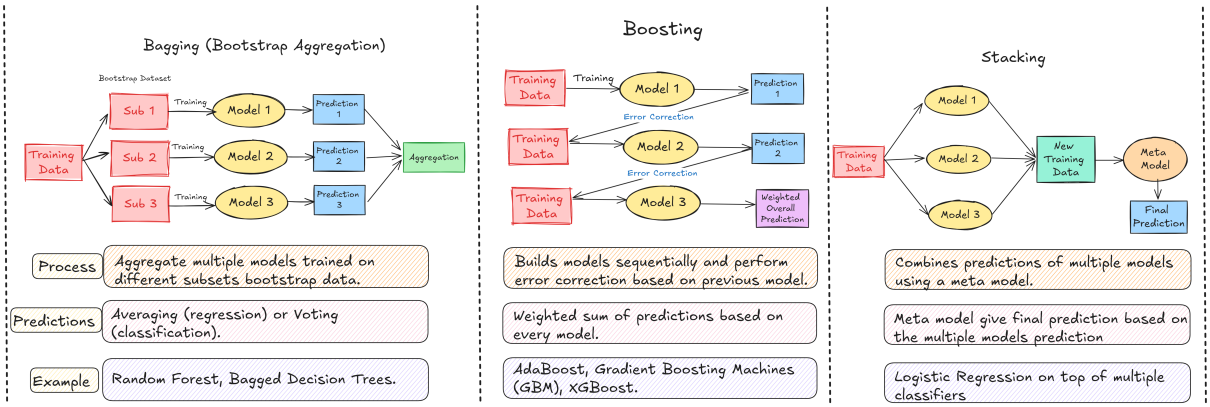

Figure 4.15: Comparison Between the Three Main Paradigms of Ensembling (Source)

The idea behind bagging is that by training multiple models on different subsets of the data, we can reduce the variance of the model and improve its predictive performance. The most popular bagging algorithm is the Random Forest algorithm in Figure 4.16.

2. **Boosting:** *Boosting* is an ensemble method that works by training multiple models sequentially, with each model learning from the mistakes of the previous model. The idea behind boosting is that by training multiple models sequentially, we can reduce the bias of the model and improve its predictive performance. The most popular boosting algorithm is the Gradient Boosting algorithm, which trains multiple weak learners (e.g., decision trees) sequentially, with each learner learning from the mislabeled instances of the previous learner.

3. **Stacking:** Lastly, and for the sake of completeness, *stacking* is an ensemble method that works by training multiple models and then combining the predictions of the models using another model (the *meta-learner*). For instance, training a set of estimators (not necessarily of the same type) and then combining the predictions of the models using a linear regression meta-learner.

In our case, we compared the performance of the following ensemble methods:

- **Random Forest:** A popular ensemble method that works by training multiple decision trees on different subsets of the data (with replacement) and then combining the predictions of the trees to make the final prediction (see Figure 4.16).

- **AdaBoost:** A boosting algorithm that works by training multiple weak learners sequentially, with each learner learning from the mistakes of the previous learner [148]. AdaBoost works by assigning a weight to each training example, with the weight of misclassified examples being increased in each iteration. The final prediction is then made by combining the predictions of the weak learners, with the weight of each learner being determined by its accuracy.

- **XGBoost:** A popular implementation of the Gradient Boosting algorithm that is optimized for speed and performance. Again, as the other gradient boosting methods, XGBoost [149] works by training multiple weak learners (e.g., decision trees) sequentially, with each learner learning from the mistakes of the previous learner. However, XGBoost is well-known for its speed and performance — even supporting fully distributed GPU training — and is widely used in practice for a variety of machine learning tasks.

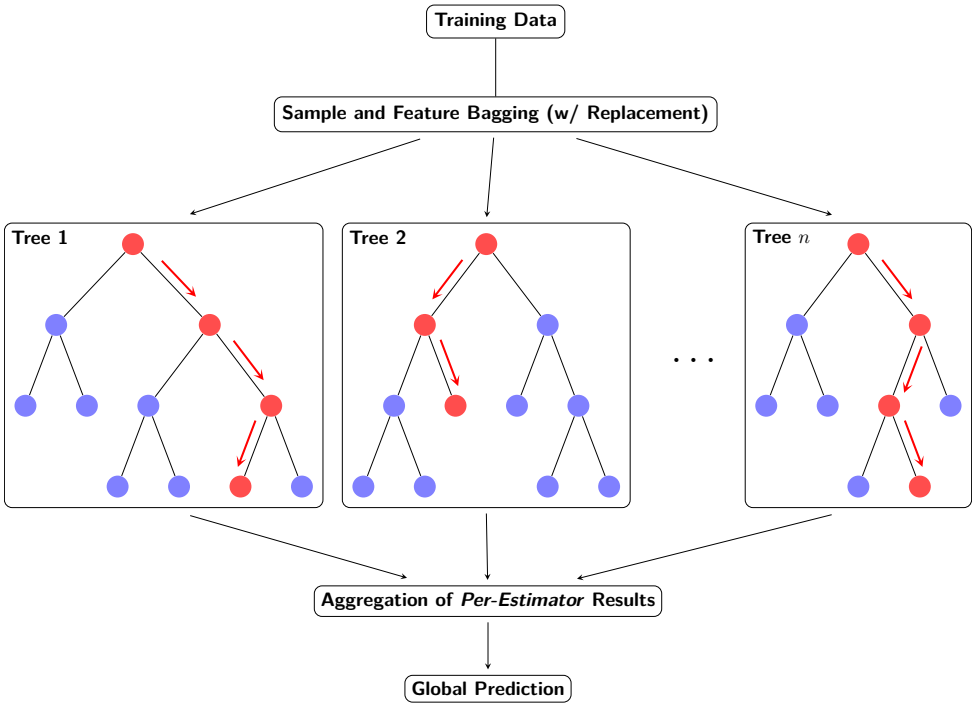

Figure 4.16: Visualization of the Random Forest Ensemble Method (TikZ Code Source)

- **Hist-Gradient Boosting:** Inspired by LightGBM [150], the Hist-Gradient Boosting algorithm is a variant of the Gradient Boosting algorithm that is implemented in Scikit-Learn and is optimized for dealing with large datasets ($N \geq 10,000$). In practice, this drastic speedup in performance is due to the estimator *binning* the input samples into integer-valued bins, thus greatly reducing the number of splitting points to be considered.

4.5.4 Neural Network Architectures

Next, we trained a variety of **deep learning** architectures on the data. Deep learning is a subfield of machine learning that focuses on training neural networks with multiple layers (i.e., *deep* neural networks) to learn complex patterns in the data. Deep learning has been shown to be highly effective for a wide range of machine learning tasks, including image recognition, speech recognition, and natural language processing. Although we do not wish to unnecessarily extend the length of this document with a detailed explanation of neural networks and their training algorithms (especially because it is not the focus of this work but rather a baseline for comparing our Bayesian models), we will provide a brief overview of the deep learning architectures that we trained on the data:

- **Multi-Layer Perceptron (MLP):** A traditional feedforward neural network with multiple layers of neurons. The MLP model is trained using the backpropagation algorithm, which works by iteratively updating the weights of the network θ by gradient descent to minimize the loss function $\mathscr{L}$:

$$\theta^* = \arg\min_{\theta} \mathscr{L}(\theta) \tag{4.44}$$

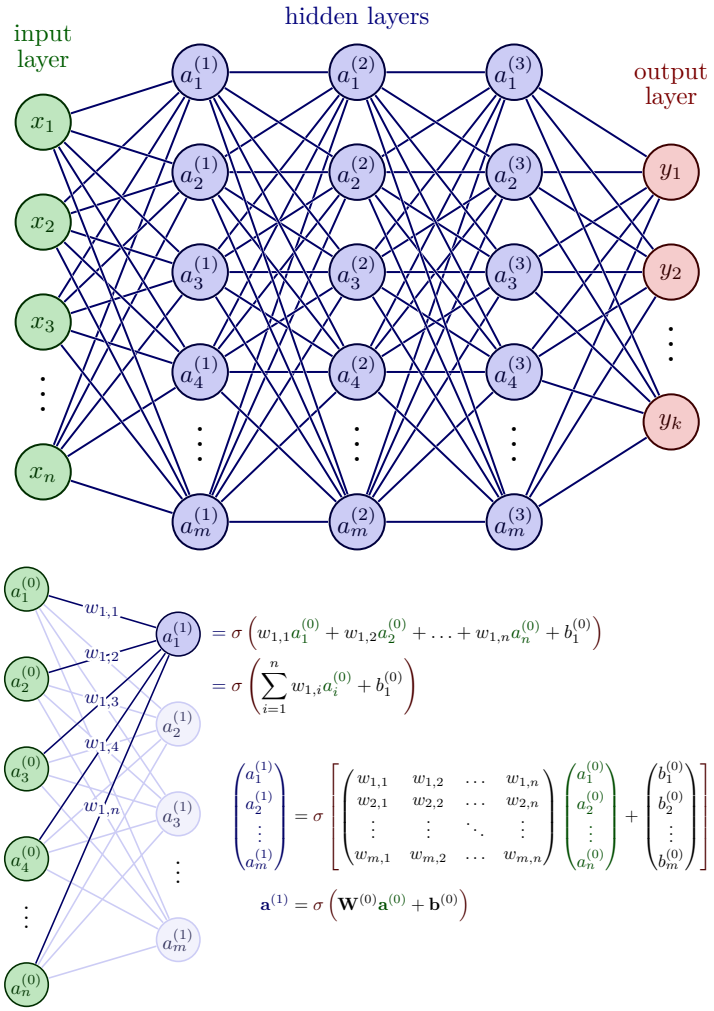

Figure 4.17: Multi-Layer Perceptron (MLP) Architecture (TikZ Source Code)

At its core, traditional *gradient descent* updates the model parameters at each step t, based on a loss function $\mathscr{L}$, according to the rule in (4.45):

$$\theta_t \leftarrow \theta_{t-1} - \alpha \vec{\nabla}_{\theta} \mathscr{L}(\theta_{t-1}) \tag{4.45}$$

where the scalar α (which can be either fixed or change during training with a *scheduler*) is commonly referred to as the *learning-rate*.

The MLP architecture, as summarized in Figure 4.17, consists of an input layer, one or more hidden layers, and an output layer. The hidden layers are typically fully connected, meaning that each neuron in a layer is connected to every neuron in the previous layer. The output layer is usually a softmax layer for classification tasks, which outputs a probability distribution over the classes. Additionally, in order to improve the generalization of the model, we also included **1)** *dropout* in the network, which randomly drop out a fraction of the neurons during training to prevent overfitting; and **2)** *early-stopping*, which stops the training process when the validation loss stops decreasing.

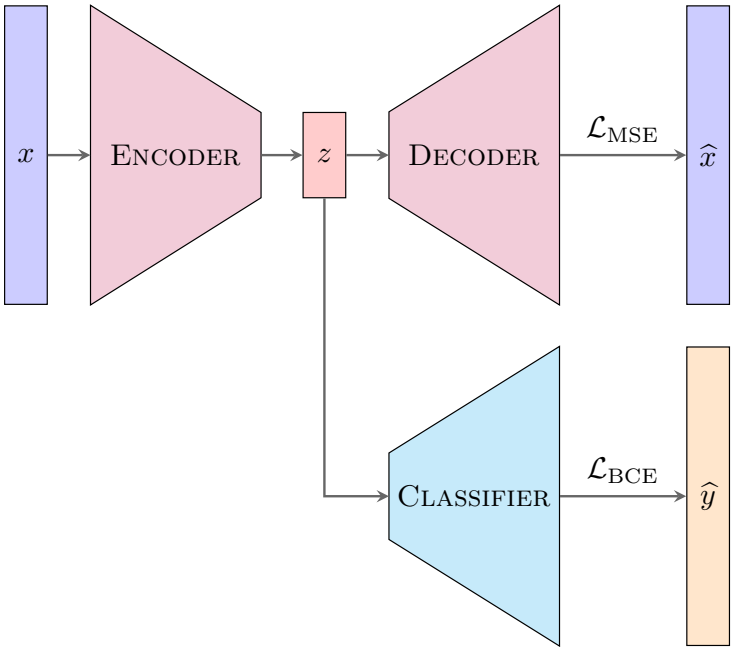

Figure 4.18: Autoencoder (w/ Classifier Module) Architecture

- **Autoencoder:** An autoencoder is a type of neural network that is trained to learn a compressed representation of the input data. The autoencoder, as in Figure 4.18, consists of an *encoder* network that maps the input data to a lower-dimensional representation (the *latent space*), and a *decoder* network that maps the lower-dimensional representation back to the original input data. The autoencoder is trained by minimizing the reconstruction error between the input data and the output data (such as the mean squared error). Once the autoencoder has been trained, the encoder network can be used to extract the compressed representation of the input data, which can then be used as input to another model (e.g., a classifier). In our case, we trained an autoencoder on the gene expression data to learn a compressed representation of the data that could be used as input to the classification module. The loss function of the autoencoder is given by:

$$\mathscr{L} = \mathscr{L}_{\text{MSE}} + \mathscr{L}_{\text{BCE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 - \frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{4.46}$$

  where $\mathbf{x}_i$ is the input genetic data, $\hat{\mathbf{x}}_i$ is the output *reconstructed* data, y_i is the ground-truth patient outcome, and $\hat{y}_i$ is the predicted label.

- **Variational Autoencoder (VAE):** Lastly, we also employed a variational autoencoder (VAE) [151] to learn a probabilistic representation of the input data. The VAE, as in Figure 4.19, consists of an encoder network that maps the input data to a distribution over the latent space, and a decoder network that maps the latent space back to the input data. The VAE uses what is commonly known as the *reparameterization trick* ($z = \mu + \sigma \odot \varepsilon$) to sample from the latent space, which allows the model to be trained using backpropagation and traditional gradient descent (μ and σ are learned as regular model parameters). The VAE is trained by maximizing a composite loss function that consists of two terms: the *reconstruction loss* and the *KL divergence* loss. The reconstruction loss measures the difference between the input data and

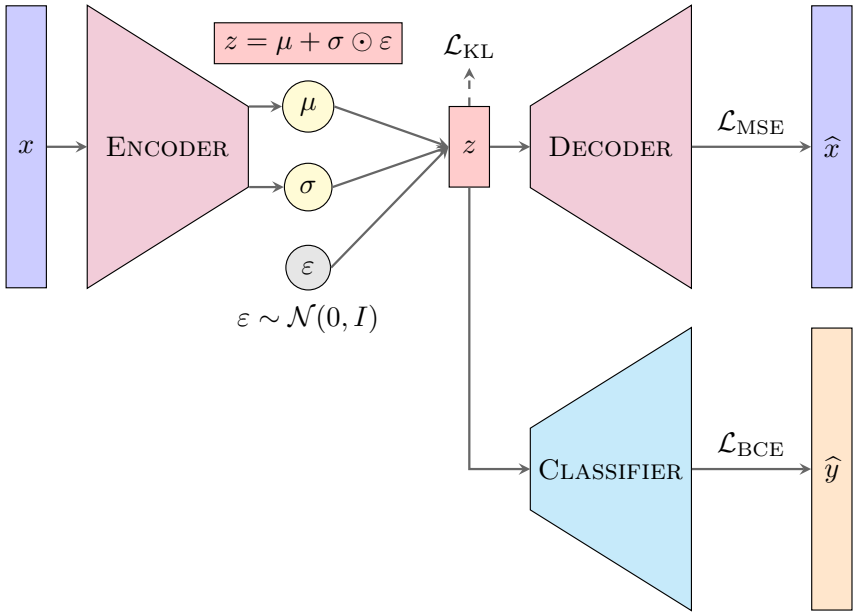

Figure 4.19: Variational Autoencoder (VAE w/ Classifier Module) Architecture

the output data (in our case $\mathscr{L}_{\text{MSE}}$), while the KL divergence loss measures the difference between the learned distribution over the latent space and a prior distribution (e.g., a standard normal distribution). Additionally, due to the classification module in the architecture, we also included a binary cross-entropy loss term $\mathscr{L}_{\text{BCE}}$ in the loss function. Thus, the loss function of our VAE is given by:

$$\begin{aligned} \mathscr{L} &= \mathscr{L}_{\text{MSE}} + \mathscr{L}_{\text{KL}} + \mathscr{L}_{\text{BCE}} \\ &= \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \frac{1}{N}\sum_{i=1}^{N} \text{KL}(q(\mathbf{z}_i|\mathbf{x}_i)\|p(\mathbf{z})) \\ &\quad -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \end{aligned} \tag{4.47}$$

where $\mathbf{z}_i$ is the latent representation of the input data, $q(\mathbf{z}_i|\mathbf{x}_i)$ is the learned distribution over the latent space, and $p(\mathbf{z})$ is the prior distribution over the latent space.

4.5.5 Bayesian Neural Networks

Finally, following upon our general interest in Bayesian models, we introduce Bayesian Neural Networks (BNNs) [152–154] as yet another example of stochastic statistical model. Without getting into too much detail, in a BNN, the weights of the neural network are treated as random variables of a probability distribution, rather than as fixed scalar parameters[5] as depicted in Figure 4.20. Despite the inherent advantages of BNNs against traditional *point-estimate* neural networks — such as their uncertainty quantification, robustness to overfitting, or the ability to introduce prior knowledge on the weight distributions — they pose a significant computational bottleneck both at training time and during inference. In fact, as you may remember from Section 3.2, dealing

[5]There are also some instances in which the probability distributions are defined for the activations instead, but let us only focus in the more traditional case.

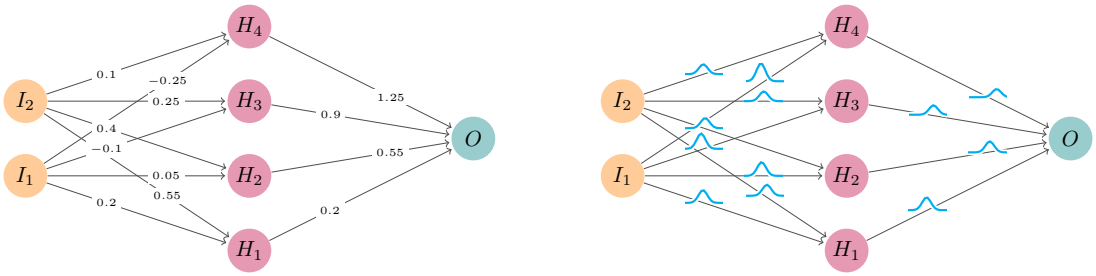

Figure 4.20: Bayesian Neural Network (BNN) Architecture (TikZ Source Code)
(*Left:* A *point-estimate* traditional network, *Right:* A BNN w/ probabilistic weights)

with Bayesian models entails the computation of intractable integrals and thus why we resort to using sampling methods.

Although we will not get here into the specific technical details of BNNs[6] — mainly because they are quite easily understood by extrapolating the concepts applied here to BLR to neural network weights — we will take the liberty of touching on two very interesting concepts on which we stumbled upon this work. First, the extrapolation to BNNs of a sensible *early-stopping* criterion. Indeed, through what we have decided to name as **Monte Carlo early-stopping**, BNNs offer the possibility of *independently* — i.e., not in terms of the loss being used for learning, or based on the evaluation metrics of the problem — quantifying the generalization capabilities of the model by sampling from the posterior distribution of the weights and evaluating the model against the actual *ground-truths* through some sort of similarity measures (e.g., cosine similarity, euclidean similarity, etc.). Second, the inherent problem of training and inferencing BNNs can be conceived from several perspectives. Although we will not make an exhaustive list of the methods available in the literature, we believe it is quite interesting to explain the general ideas behind these approaches. In general terms, we can divide the methods into the four categories below, all of which are summarized in Figure 4.21.

- **(*Properly*) Bayesian Methods:** First, we find those methods that are *purely* Bayesian in the sense that they *learn* a posterior distribution by adhering strictly to Bayesian principles: they treat parameters as random variables, update beliefs based on Bayes' theorem, and aim to approximate, or explore, the full posterior distribution rather than defaulting to point estimates. Broadly, this category comprises Markov-Chain Monte Carlo (MCMC) methods — which rely on *sampling-based* approximate methods to generate samples directly from the posterior distribution — and Variational Inference (VI) — an *optimization-based* approximate method seeking a simpler parametric distribution $q_\phi(\boldsymbol{\theta})$ that is closest (by a divergence measure such as the KL-divergence) to the true posterior $p(\boldsymbol{\theta}|\mathcal{D})$.

- ***Quasi*-Bayesian Methods:** The main problem with the approaches above is that the inherent stochasticity in the weights of the network impedes backpropagation as a *learning* method. To counter this, and exploit the training benefits of traditional gradient descent (GD), several approaches have been proposed in the literature to combine purely bayesian updates and GD. Among these, we find Bayesian Stochastic

[6] For the interested reader, [152] is a wonderful survey on the topic.

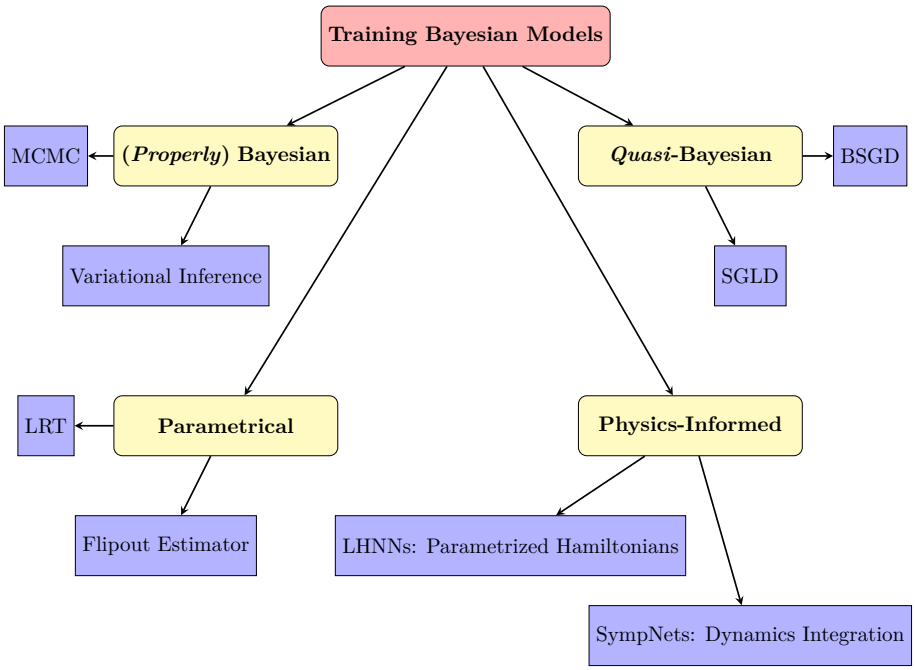

Figure 4.21: Summary of the Different Approaches to Training Bayesian Models

Gradient Descent (BSGD, also know as *probabilistic backpropagation*, or *Bayes-by-backprop*) [152, 155] which applies gradient descent over the log-prior and log-likelihood as in Eq. 4.48.

$$\theta_{t+1} \leftarrow \theta_t - \alpha_t \left(\lambda_1 \vec{\nabla}_\theta p(\theta_t) + \lambda_2 \frac{N}{n} \sum_{i=1}^{N} \vec{\nabla}_\theta p(x_i|\theta_t) \right), \qquad \lambda_1 + \lambda_2 = 1 \tag{4.48}$$

where λ_1 and λ_2 are the weights assigned to the gradients on the prior and likelihood, respectively. Likewise, from the same family of methods, Stochastic Gradient Langevin Dynamics (SGLD) [155] in Eq. (4.49) follows a similar approach with a slightly modified update rule.

$$\theta_{t+1} \leftarrow \theta_t - \alpha_t \left(\lambda_1 \vec{\nabla}_\theta p(\theta_t) + \lambda_2 \frac{N}{n} \sum_{i=1}^{N} \vec{\nabla}_\theta p(x_i|\theta_t) \right) + \eta_t \,, \qquad \eta_t \sim \mathcal{N}(0, \alpha_t) \tag{4.49}$$

However, in practice, it has been shown that although this method converges to a Markov-Chain that samples the true posterior as $\alpha_t \to 0, t \to \infty$, the samples become increasingly autocorrelated [155, 156].

- **Parametrical (aka Weight Perturbation) Methods:** Following upon the previous approaches, the next logical step is to move forward towards *purely* parametrical methods. Similarly to the *reparametrization trick* in Variational Autoencoders, here the probabilistic weights in the network are represented in terms of a set of deterministic parameters that define the underlying distribution (as was the case with μ and σ for the Gaussian VAE in Figure 4.19), and are learned through traditional gradient descent optimization. Likewise, as was the case in the VAEs, stochasticity is introduced through the *random* perturbation of the weights (or the activations) of the model.

Within these types of methods we find the **Local Reparametrization Trick (LRT)** [157] which implements exactly the same approach used in VAEs. For instance, if we set a gaussian distribution for the weights, i.e., $q_\phi(\mathbf{w}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \mathbf{w} \in \mathbf{W}$, then the weights can be *locally* reparametrized as $\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In fact, [157] shows that the stochasticity in the weights (i.e., actually in $\boldsymbol{\varepsilon}$) only influences the expected log-likelihood through the activations of the neurons. As these have a much lower dimension, it is generally preferable to directly sample the activations instead. Moreover, the authors also introduce *variational dropout*, an extension of the traditional dropout generalization method for neural networks to the context of Bayesian models.

Another very famous approach to these weight perturbatory models, is the **Flipout Estimator** [158]. Similar to LRT, Flipout introduces stochasticity in the weights of the network by sampling from a distribution of perturbations. However, in this case, the sampling is much more computationally efficient as it uses a common perturbation $\widehat{\Delta \mathbf{W}}$ shared by all examples in the mini-batch. In fact, this method efficiently decorrelates the weight gradients between different samples in a mini-batch, and also reduces the variance compared to the shared perturbations in LRT. For the interested, both LRT and Flipout Estimation are the implement approaches in the wonderful Bayesian Torch [159] library by Intel: a PyTorch wrapper for Bayesian Neural Networks.

- **Physics-Informed Methods:** Finally, we find a very interesting approach to the problem of training BNNs by approximating the solving of the intractable integrals in the posterior distribution of the *properly* Bayesian methods above by employing Physics-Informed Neural Networks (PINNs). Although much work has been done in this particular field, we will limit ourselves here to three simple examples that are related to approximating Hamiltonian Monte Carlo (HMC).

  For instance, **Latent Hamiltonian Neural Networks (LHNNs)** [160], solve the computationally expensive limitations of HMC when dealing with large datasets by approximating the true Hamiltonian $H(\boldsymbol{\theta}, \mathbf{p})$ with a neural network parametrization $H_{\mathbf{W}}(\boldsymbol{\theta}, \mathbf{p})$ trained to minimize the loss in terms of the gradients with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$. Although at the time of its publication this method was praised due to its time-reversibility and conservation of the parametrized Hamiltonian, training the network still required computing analytical solutions to the gradients.

  On the other hand, **Symplectic Networks** such as SympNets [161] and its NeuralODE approximation using the Taylor expansion of the gradients [162] have proven to be a significant step up. In fact, symplectic networks directly estimate the whole process of integrating Hamiltonian dynamics. That is, given a pair of initial state $(\boldsymbol{\theta}^0, \mathbf{p}^0)$, the model estimates the final state $(\boldsymbol{\theta}^L, \mathbf{p}^L)$ after L supposed integrations steps. Likewise, not only is it much more computationally efficient to train, but also allows for non-separable Hamiltonians, which are well outside the scope of this work[7].

[7] Non-separable Hamiltonians are rarely used in applications related to computational statistics. In fact, they usually arise when using HMC for molecular dynamics written in the internal coordinates of the system.

5 Results & Discussion

"I think I did pretty well, considering I started out with nothing but a bunch of blank paper."

~ Steve Martin

In this chapter, we present the results of our study on tamoxifen resistance in breast cancer patients. After introducing the evaluation metrics used to assess the performance of our prognostic models, we will analyze and discuss the results of the models developed for predicting the likelihood of treatment resistance in patients. We will then discuss the potential genetic biomarkers identified by the models with the help of SHAP explainability and interpretability methods. Finally, we will validate our prognostic signature through a survival analysis on an external set of patients, as well as search for potential associations between our genes and well-known biological mechanisms.

5.1 Evaluation Metrics

Before we delve into the results of our study, it is important to understand the evaluation metrics used to assess the performance of our prognostic models. The choice of evaluation metrics is crucial in the context of imbalanced datasets, such as the one we are dealing with in this study, where only around 30% of the patients exhibit resistance to tamoxifen. In such cases, accuracy is not a reliable metric for evaluating the performance of the models. For instance, a model that predicts all patients as non-resistant would still achieve an accuracy of ~70%, which is not desirable. Therefore, we use a combination of evaluation metrics that are more suitable and informative for imbalanced datasets, such as precision, recall, specificity, F_1 score, and the quite robust Matthews Correlation Coefficient (MCC). These metrics are defined as follows:

- **Accuracy:** The ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5.1}$$

- **Precision:** The ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.2}$$

- **Recall/Sensitivity:** The ratio of correctly predicted positive observations to all observations in the actual class.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.3}$$

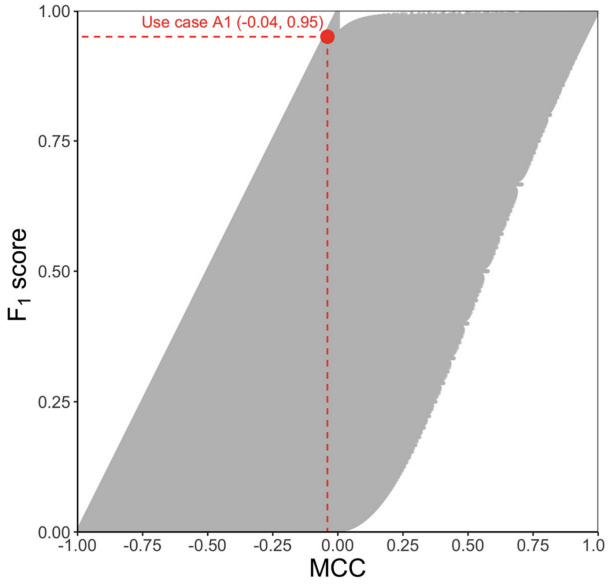

Figure 5.1: Relationship Between the MCC and F_1 Score for All the 21,084,251 Possible Confusion Matrices for a Dataset with 500 Samples. (Extracted from [164])

- **Specificity:** The ratio of correctly predicted negative observations to all observations in the actual class.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5.4}$$

- F_1 **Score:** The weighted average (harmonic mean) of Precision and Recall.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.5}$$

- **Matthews Correlation Coefficient (MCC):** Provides a robust measure — especially for imbalanced datasets — ranging from −1 (perfect inverse prediction) to +1 (perfect prediction), with 0 indicating a random prediction. In essence, the binary-class MCC in Eq. (5.6), is closely related to a χ^2-statistic over the confusion matrix $\mathscr{M}$, and measures the *correlation* between the observed and predicted classifications [163].

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{5.6}$$

  Likewise, the metric incorporates class imbalance and, unlike the F_1 score in Eq. (5.5) and other related metrics above, is invariant to *class swapping*. Moreover, it can be shown that the F_1 score is *independent* of TN, and in cases where the model performs well with only one of the classes, the two measures can be highly discordant [164]. To put this into context, Figure 5.1 summarizes this *peculiar* relationship between the two metrics for all the possible confusion matrices of a dataset with 500 samples.

When there are more than two classes, the MCC is defined as in Eq. (5.7).

$$\mathrm{MCC} = \frac{c \cdot s - \sum_{k}^{K} p_k \cdot t_k}{\sqrt{\left(s^2 - \sum_{k}^{K} p_k^2\right)\left(s^2 - \sum_{k}^{K} t_k^2\right)}} \tag{5.7}$$

where c is the number of correctly predicted samples, s is the total number of samples, t_k is the number of *actual* occurrences of class k, and p_k is the number of times class k was predicted by the model [165].

Most importantly, in the context of our work, we will mainly focus on the MCC and the recall (or sensitivity) metric for evaluating our prognostic models. First, as we have mentioned, the MCC is an extremely robust metric when dealing with imbalanced datasets, and is quite sturdy — as well as informative of the quality of the predictions in ambiguous cases — as a classification metric. Second, the recall metric is particularly important in the context of our study, as we are more interested in identifying patients that are resistant to tamoxifen, rather than those that are not. In other words, we are more concerned with the *true positives* (TP) and *false negatives* (FN) than with the *true negatives* (TN) and *false positives* (FP). This is because, in the context of tamoxifen resistance, it is more critical to prioritize identifying patients that are resistant to the drug, rather than those that are not. In this case, as usual in the critical context of health-related models, we want to be as sure as possible that our model is capable of discerning the most **unfavorable scenario** [166], i.e., we would rather wrongfully predict a *bad* prognostic outcome, than miss an actual one, potentially putting a patient's life at risk and wasting 5 critical years with a *useless* treatment.

5.2 Results of Tamoxifen Prognosis Models

To evaluate the performance of our prognostic models, we employed a Stratified 5-Fold Cross-Validation (CV) approach to ensure that each fold maintained the original class distribution, i.e., our ~30% resistant, ~70% non-resistant ratio. Stratification is critical for imbalanced datasets, as it prevents scenarios where certain folds lack representation of the minority class, which generally bias performance estimates. Likewise, all models were trained as well on the *augmented* dataset, in which we applied Synthetic Minority Oversampling Technique (SMOTE) [167] to balance the classes, generating synthetic samples for the resistant class while retaining the original instances. This allowed us to assess whether addressing class imbalance — which had not been done in previous studies [37] — improved model robustness, particularly in identifying true positives. Each model was trained and evaluated on both the original and augmented datasets to isolate the impact of augmentation.

The results of the models are summarized in Tables 5.1 — which includes the traditional *shallow* models, ensemble methods, and neural network architectures — and 5.2, encompassing all the Bayesian models. Likewise, Figure 5.2 shows the recall/sensitivity metric, while Figure 5.3 summarizes the Matthew's Correlation Coefficient (MCC) for each model, in both cases differentiating between the original and augmented datasets.

Table 5.1: Performance of Tamoxifen *Point-Estimate* Prognostic Models (Average Across 5-Folds of Stratified CV)

	Precision	**Recall/Sensitivity**	**Specificity**	F_1	**MCC**	**Accuracy**
Traditional *Shallow* Models						
Logistic Regression	0.551	0.367	0.842	0.351	0.193	0.700
► *Augmented Dataset*	0.778	0.763	0.720	0.728	0.533	0.740
SVC (Linear Kernel)	0.517	0.367	0.803	0.338	0.150	0.675
► *Augmented Dataset*	0.760	0.720	0.681	0.668	0.468	0.700
SVC (RBF Kernel)	0.800	0.167	0.922	0.180	0.090	0.679
► *Augmented Dataset*	0.853	0.681	0.840	0.724	0.564	0.760
Naive Bayes	0.633	0.533	0.880	0.567	0.431	0.779
► *Augmented Dataset*	0.860	0.682	0.844	0.698	0.573	0.760
Ensemble Methods (Bagging[1], Boosting[2])						
Random Forest[1]	0.607	0.367	0.840	0.374	0.209	0.700
► *Augmented Dataset*	0.828	0.840	0.760	0.808	0.645	0.800
Hist-Gradient[2]	1.000	0.000	1.000	0.000	0.000	0.679
► *Augmented Dataset*	0.769	0.680	0.762	0.681	0.487	0.720
AdaBoost[2]	0.587	0.600	0.800	0.563	0.416	0.750
► *Augmented Dataset*	0.839	0.761	0.760	0.765	0.569	0.760
XGBoost[2]	0.426	0.217	0.840	0.208	0.010	0.643
► *Augmented Dataset*	0.903	0.722	0.880	0.769	0.645	0.800
Neural Network (NN) Architectures						
MLP-1Layer	0.617	0.367	0.842	0.351	0.219	0.696
► *Augmented Dataset*	0.748	0.761	0.687	0.715	0.491	0.720
MLP-2Layer	0.900	0.633	0.920	0.648	0.581	0.839
► *Augmented Dataset*	0.914	0.962	0.880	0.933	0.844	0.920
Autoencoder	0.800	0.367	0.920	0.382	0.290	0.754
► *Augmented Dataset*	0.927	0.960	0.921	0.942	0.883	0.940
Variational Autoencoder	0.520	0.333	0.840	0.393	0.200	0.675
► *Augmented Dataset*	0.870	0.920	0.840	0.886	0.778	0.880

Table 5.2: Performance of Tamoxifen *Bayesian* Prognostic Models (Average Across 5-Folds of Stratified CV)

	Precision	**Recall/Sensitivity**	**Specificity**	F_1	**MCC**	**Accuracy**
Bayesian Models w/ Hamiltonian Monte Carlo (HMC) Posterior Sampling						
BLR-HMC	0.520	0.433	0.768	0.364	0.180	0.675
► *Augmented Dataset*	0.748	0.768	0.682	0.715	0.491	0.720
BLR-GHMC	0.550	0.433	0.800	0.386	0.215	0.700
► *Augmented Dataset*	0.802	0.760	0.760	0.743	0.565	0.760
BLR-(G)HMC (s-AIA$_2$)	0.550	0.433	0.800	0.386	0.215	0.700
► *Augmented Dataset*	0.802	0.760	0.760	0.743	0.565	0.760
BLR-(G)HMC (s-AIA$_3$)	0.550	0.433	0.800	0.386	0.215	0.700
► *Augmented Dataset*	0.802	0.760	0.760	0.743	0.565	0.760
Bayesian Neural Networks (BNNs) w/ Reparametrization Trick for Gradient Estimation						
BNN-1Layer	0.550	0.367	0.842	0.351	0.193	0.700
► *Augmented Dataset*	0.734	0.761	0.640	0.704	0.458	0.720
BNN-2Layer	1.000	0.833	1.000	0.893	0.878	0.946
► *Augmented Dataset*	0.967	**0.964**	0.960	0.956	**0.927**	0.960

Synthetic Minority Oversampling Technique (SMOTE)

SMOTE [167] is an oversampling technique used to address class imbalance in datasets. It works by generating synthetic samples for the minority class. This is achieved by selecting a sample from the minority class and finding its k nearest neighbors. Synthetic samples are then created by interpolating between the selected sample and its neighbors. This helps to balance the class distribution and improve the performance of Machine Learning models on imbalanced datasets. Formally, given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ is the class label, SMOTE works as follows:

1. For each sample $\mathbf{x}_i$ in the minority class, find its k nearest neighbors (also in the minority class).
2. Randomly select one of the neighbors, $\mathbf{x}_{nn}$, and compute the difference vector $\mathbf{d} = \mathbf{x}_{nn} - \mathbf{x}_i$.
3. Generate a synthetic sample $\mathbf{x}_{synth} = \mathbf{x}_i + \lambda \cdot \mathbf{d}$, where $\lambda \in [0, 1]$ is an *interpolation* coefficient.
4. Repeat the process for all samples in the minority class until the desired balance is achieved.

SMOTE is a powerful technique for addressing class imbalance, but it is important to be cautious when using it: generating too many synthetic samples can lead to overfitting (and loss of generalization), while generating too few may not effectively balance the classes.

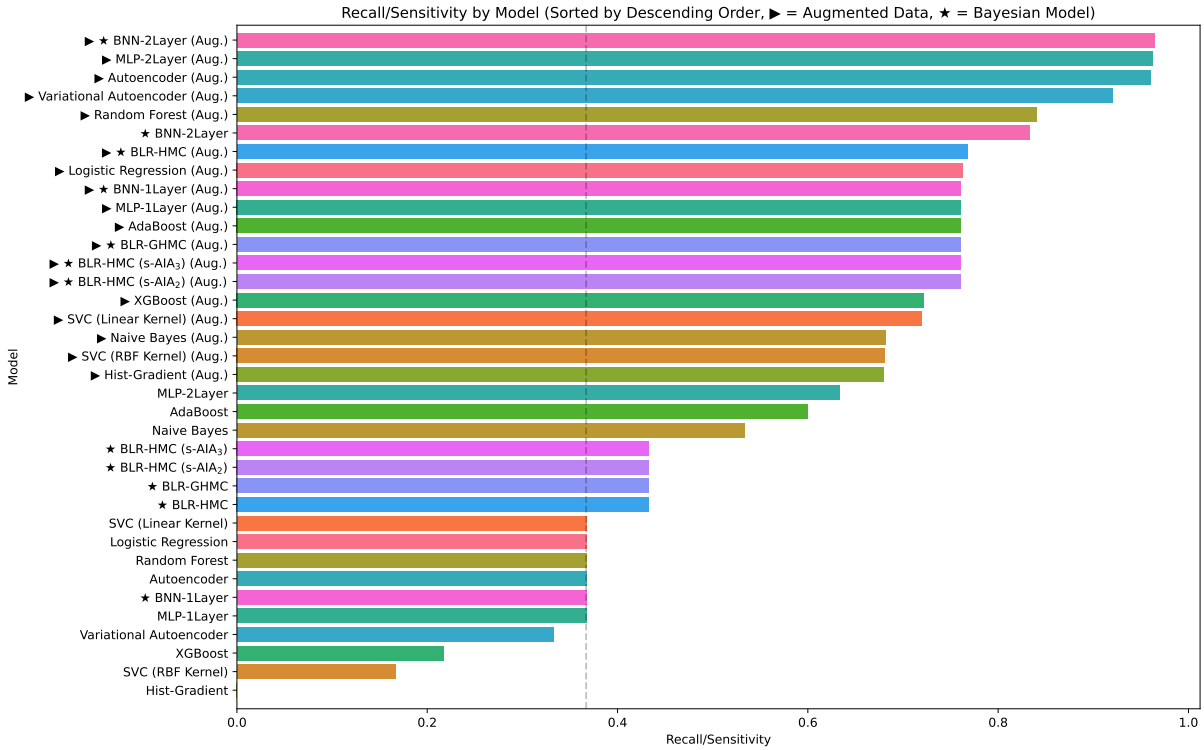

Figure 5.2: Recall/Sensitivity by Model (Average Across 5-Folds of Stratified CV) (In Gray, the Baseline from [37])

Across all models, **data augmentation** consistently enhanced performance, with the most dramatic improvements observed in neural networks and Bayesian architectures. For instance, the 2-layer MLP achieved an MCC of 0.844 and recall of 0.962 on augmented data, compared to 0.581 and 0.633 on the original imbalanced dataset. Similarly, the Autoencoder model exhibited a remarkable MCC of 0.883 and recall of 0.960 after augmentation, or the XGBoost model, which improved from an MCC of 0.010 to 0.645. This widespread tendency across all instances suggest that addressing class imbalance is *crucial* for improving the robustness of models in identifying tamoxifen resistance. For instance, just by looking at the recall metric, as in Figure 5.2, we can see that the majority of models exhibit a significant improvement in identifying resistant patients after data augmentation. This is particularly important in the context of tamoxifen resistance, where correctly identifying resistant patients is crucial for their treatment and care. More interestingly, the prior work upon which this study is based [37] did not address class imbalance, and the model developed in that study exhibited poor performance in identifying resistant patients (see the gray line in Figure 5.2, with a recall of 0.367). Our results suggest that, by addressing this simple class imbalance, we can significantly improve the performance of prognostic models for tamoxifen resistance, even doubling the recall rate in some cases.

Regarding the different model architectures, the superiority of neural networks (both Bayesian and *point-estimate*) is attributed to their capacity to learn hierarchical representations from high-dimensional (genetic) data. The 2-layer MLP, for example, exploited augmented data to achieve a recall of 0.962 and an MCC of 0.844, suggesting robust generalization. Similarly, the Autoencoder's reconstruction-based training likely enhanced its discriminative power for rare resistant cases, yielding an MCC of 0.883. Bayesian models further distinguished themselves by coupling high recall with calibrated uncertainty estimates. The BNN-2Layer not only achieved the highest recall but also provided posterior distributions for predictions, a critical feature for

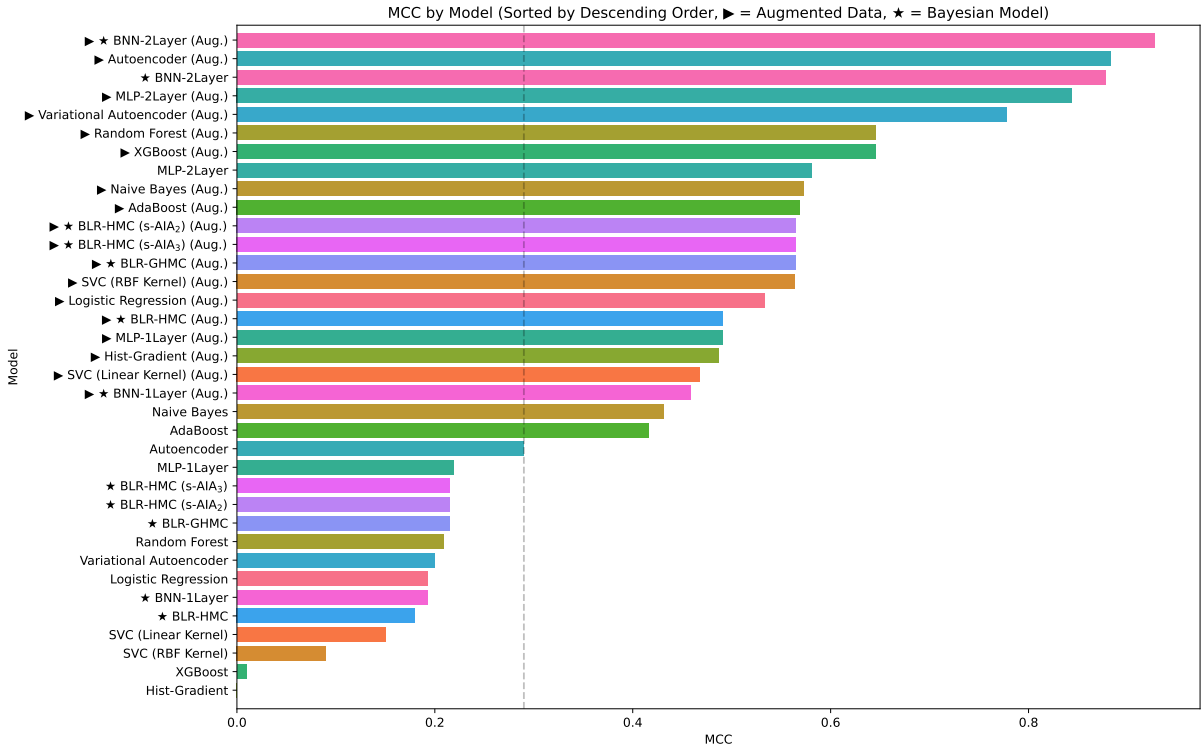

Figure 5.3: MCC by Model (Average Across 5-Folds of Stratified CV) (In Gray, the Baseline from [37])

clinical scenarios requiring risk stratification. Traditional models, however, exhibited more modest gains. While augmented Random Forest improved recall to 0.840, kernel-based methods like SVC (RBF) plateaued at 0.681, underscoring their limitations in imbalanced, high-dimensional settings. Notably, XGBoost prioritized precision (0.903) over recall (0.722), reflecting a conservative strategy ill-suited to clinical contexts where missing resistant cases carries significant risk.

Furthermore, our experiments further elucidate the benefits of **probabilistic modeling**, particularly Bayesian Neural Networks (BNNs), demonstrated significant improvements in performance metrics compared to traditional models. For instance, the 2-layer BNN achieved an MCC of 0.927 and a near-perfect recall of 0.964 on the augmented dataset, suggesting robust uncertainty quantification alongside high predictive power. Interestingly, the Bayesian Logistic Regressions with Hamiltonian Monte Carlo (BLR-HMC) methods — which integrate patient RNA-seq data with priors for each gene extracted from the expression analysis on the lab-grown cell-lines — reached a performance plateau of 0.565 MCC and 0.760 recall on the augmented data. This suggests that, contrary to prior beliefs, the added information from RNA-seq data might not be as impactful as initially anticipated.

However, the superior performance of augmented models must be interpreted with caution. While SMOTE mitigates class imbalance, synthetic samples may not fully represent **biological variability**. The high specificity (>0.84) and recall (>0.92) of top-performing models like the Autoencoder suggest minimal overfitting, but external validation on independent cohorts is essential to confirm generalizability. Nevertheless, our results establish a compelling precedent for integrating data augmentation and probabilistic modeling in prognostic tasks involving imbalanced biomedical data.

Discussion on the Tamoxifen Prognosis Models

Although we could fill countless pages based on meaningless pairwise comparisons between the different models above, let us try instead to dissect all of these results, and hopefully, shed some light into what we have been able to achieve, and of course, our limitations...

1. **On the Wonders of Data Augmentation...** As we have already discussed, data augmentation has had quite a remarkable impact in our study. In all cases, the models trained on the augmented dataset outperformed those trained on the original data. This is particularly evident in some of the neural network architectures, where *near-perfect* recall rates were achieved after data augmentation. Although quite trivial, this simple step had not been taken in the prior work upon which this study is based [37], and the results here suggest that it is crucial for improving the robustness of models in identifying tamoxifen resistance.

   Nevertheless, it is important to note that, while data augmentation has significantly improved the performance of our models, it might be limiting in terms of biological variability. Synthetic samples generated by SMOTE may not fully represent the underlying biological variability in the data. Therefore, it is essential to validate the models on independent cohorts (such as additional patient samples from other cancer repositories) to ensure their generalizability.

2. **Different Architectures Perform Very Differently...** It is not surprising that different model architectures perform very differently in our study. As we would expect, neural networks usually outperform ensemble methods, which in turn outperform traditional *shallow* methods. However, even a simple logistic regression or random forest, with proper data pre-processing, can outperform many of the much more powerful and robust methods. This is a classic case of **Occam's razor**, where simpler models are preferred to complex models, either due to bias-variance trade-offs, computational efficiency, or interpretability [168, 169]. In this case, it seems that some of the simpler models are more than enough to achieve the desired performance, and more complex models are much more demanding to train and validate; for instance, requiring a GPU and more sophisticated training techniques.

3. **On the Sensitivity of Models to Class Imbalance...** On the same line, deeper architectures, such as MLP-2 and BNN-2, seem to be less sensitive to class imbalance, whereas shallow one-layer neural networks (e.g., Autoencoder, VAE, MLP-1, BNN-1Layer) appear to be quite sensitive to class imbalance and severely improve when using the augmented dataset. Ensemble methods, such as Random Forest or XGBoost, perform very robustly when dealing with the augmented dataset yet are extremely sensitive to this imbalance and perform incredibly poorly when dealing with the original data. Nonetheless, we cannot be completely certain that the improvements observed in the augmented dataset are solely due to the correction in the class imbalance, as the models might have also learned more complex patterns in the data due to the augmentation in training samples. As you might remember from Section 4.1, we are dealing with as few as 37 patients! *Are we actually correcting for class imbalance or just for the lack of available data?*

4. **Regarding Bayesian Modeling...** As previously stated, Bayesian models have shown to be quite robust in our study. Consistently outperforming *point-estimate* models, BNNs, VAEs, and BLRs with HMC sampling, have shown to be particularly effective in identifying tamoxifen resistance. The BNN-2Layer, in particular, achieved the best results acorss all methods, achieving a near-perfect recall of 0.964 and an MCC of 0.927 on the augmented dataset. This suggests that Bayesian models are not only more robust in dealing with imbalanced datasets but also provide calibrated uncertainty estimates, which are crucial for clinical scenarios requiring risk stratification: i.e., where correctly identifying resistant patients is crucial for their treatment and care. However, it is important to note that Bayesian models are computationally expensive and require more sophisticated training techniques, such as Hamiltonian Monte Carlo (HMC) sampling or parametrical approaches, as you may recall from Section 4.5.5.

5. **On the Integration of Multi-Source Data...** However, this study raises a critical question: *Is the added information from RNA-seq data as impactful as initially anticipated?* The BLR-HMC models, which integrate patient RNA-seq data with priors for each gene extracted from the expression analysis on the lab-grown cell-lines, reached a performance plateau of 0.565 MCC and 0.760 recall on the augmented data, across its variants. This suggests that the added information from RNA-seq data might not be as impactful as initially anticipated. This is quite interesting, as it suggests that the genetic information from the cell-lines might not be as relevant as initially thought. This raises the question of whether the genetic information from the cell-lines is truly representative of the genetic information from the patients, and whether it is truly relevant in predicting tamoxifen resistance. This is a critical question that warrants further investigation...

   Aside from wondering whether our initial supposition for integrating *multi-source* sequencing data is correct — i.e., that patients with a positive clinical response can be considered comparable to control cells in the MCF7 cell-lines, whereas resistant patients in the TCGA group of patients can be considered comparable with TamR cells in the MCF7 cell-lines — we can still consider our study as a success. We have been able to develop a series of models that significantly outperform the prior work upon which this study is based, and have shown that by addressing class imbalance, we can significantly improve the performance of prognostic models for tamoxifen resistance. More so, we have shown the powerful benefits of using Bayesian models, particularly Bayesian Neural Networks, which, although not originally at the core of our study, have shown to be quite robust in this context, yielding almost perfect recall.

6. **Regarding Previous Work...** Finally, a quick word on the prior work upon which this study is based [37]. The model developed in that study exhibited poor performance in identifying resistant patients, with a recall of 0.367 and an MCC of 0.290. This is quite remarkable, as it shows that simple steps, such as addressing class imbalance, can have a significant impact on the performance of these prognostic models. This is a critical lesson and highlights the importance of considering class imbalance in the development of models, particularly in such critical contexts. However, despite the improvements observed in our study, mostly in terms of pre-processing

and model selection, we can observe that the prior work had a higher MCC than our BLR-HMC model (without the augmentation), suggesting that the signature distillation process might have been more effective in identifying tamoxifen resistance. *Which criterion should we use then? The MCC as a robust metric or the recall due to the critical context of our work?* Luckily, we don't have to choose, as we have many models that considerably outperform the previous approach. Nonetheless, this is a critical question that warrants further investigation... *Are these signature model-specific? How do our models perform when using the same signature distillation process?*

5.3 Identification of Potential Genetic Biomarkers

Once we have trained and evaluated our prognostic models to identify patients who develop a resistance to tamoxifen therapy, we can now proceed to identify key *potential* **genetic biomarkers** that are behind this resistance phenomenon. To do so, we can exploit **SHAP (SHapley Additive exPlanations) values** [170]: a *post-hoc* explainability method that provides a game-theoretic approach to explain individual predictions of Machine Learning models by quantifying the contribution of each feature to the prediction of a model. SHAP values are rooted in cooperative game theory, and fairly distribute the difference between a model's prediction and a baseline expectation (i.e., the average) across all input features. Applying this methodology allows us to identify the most important genes that potentially contribute to the development of resistance to tamoxifen in breast cancer cells.

Importantly, we need to clarify something; the SHAP values are not *causal* in nature, but rather *correlational*. They provide a measure of the importance of each feature in the prediction of the model, but do not imply a causal relationship between the feature and the outcome. In this context, the SHAP values will help us identify the most important genes that contribute to the development of resistance according to the predictions of our models, but further experimental validation should be required to establish a causal relationship between these genes and tamoxifen resistance.

As for the use of SHAP explainability in this work, we intend this to be a *proof-of-concept* study, where we will identify the most important genes that contribute to the development of resistance to tamoxifen in breast cancer cells using one of our best-performing models, the Random Forest classifier (using the augmented dataset)[1]. We will then compare these results with the genes identified in prior work and in related biological studies, in order to assess the consistency of the genes identified by our model, and potentially identify new genetic biomarkers that were not previously considered.

Figure 5.4 presents the *global* gene contribution, showing the mean absolute SHAP values across all patients (both resistant and non-resistant), highlighting the overall importance of each gene in predicting tamoxifen resistance. Figure 5.5 focuses on the resistant cohort — more specifically, although all samples are considered, the SHAP values are computed according to the probabilities of belonging to the resistant class — with a beeswarm plot illustrating the impact of each gene on the model's prediction, and a heatmap displaying gene contributions across individual instances. Lastly, Figure 5.6 provides the local gene contributions for two sample patients (one

[1]We have decided to use this model instead of a more complex one, such as a Bayesian Neural Network, due to the limitations of the SHAP library in handling Bayesian models or neural networks when dealing with probabilistic predictions. In fact, the SHAP library has trouble handling most classification models in Scikit-Learn when we ask it to work with classification probabilities.

resistant and one with a favorable treatment outcome), with waterfall plots detailing the per-gene contribution and decision plots showing the cumulative effect of genes on the model's prediction.

SHapley **A**dditive ex**P**lanations (SHAP) Values

SHAP values [170] provide a game-theoretic approach to explain individual predictions of machine learning models by quantifying the contribution of each feature. Rooted in cooperative game theory, SHAP values fairly distribute the difference between a model's prediction and a baseline expectation across all input features. Formally, for a model f and input instance $\mathbf{x}$, the SHAP value ϕ_i for feature i is computed as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \big(f(S \cup \{i\}) - f(S)\big) \tag{5.8}$$

where F is the set of all features ($|F| = M$), S is the subset of all features excluding i, and $f(S)$ is the expected model output conditioned on the feature subset S.
SHAP values satisfy three key properties:

1. **Local Accuracy**: $f(\mathbf{x}) = \phi_0 + \sum_{i=1}^{M} \phi_i$, where $\phi_0 = \mathrm{E}[f(\mathbf{x})]$
2. **Missingness**: $\phi_i = 0$ for missing features
3. **Consistency**: If a feature's marginal contribution increases, its SHAP value never decreases

Common implementations include:

- **KernelSHAP**: Model-agnostic approximation using weighted linear regression
- **TreeSHAP**: Polynomial-time exact computation for tree-based models, exploiting the structure of decision trees
- **DeepSHAP**: Deep learning-specific approximation using backpropagation

While SHAP provides theoretically grounded explanations, its computational complexity $\mathcal{O}(2^M)$ motivates approximate methods. The choice of background distribution for computing ϕ_0 significantly impacts interpretation quality.

First, let us take a look at the *global* feature importance plot in Figure 5.4. The variables displayed in the vertical axis correspond to the features of the model, i.e., our genes, whose global importance is greatest in the predictions: i.e., the absolute value of the SHAP coefficients, disregarding whether it impacts positively or negatively the predictions. As it stands, we can observe that, overall, the genetic biomarker with the greatest impact on assessing whether a patient is resistant to tamoxifen is the **CISH** gene, followed by **BCAS1**, **FRAS1**, **FIRRE**, **MGAT5B**, **HERC1**, **SYNPO2L**, **TMC7**, **VTN**, and **INSIG2**. These genes are well-known in the context of breast cancer prognostic modeling. For instance, [171] showed that CISH overexpression was associated with better metastasis-free outcomes, [172] demonstrated that BCAS1 is significantly relevant in the proliferation and metastasis of breast cancer, as did [173] showing the role of this gene in the proliferation of MCF7 cancer-

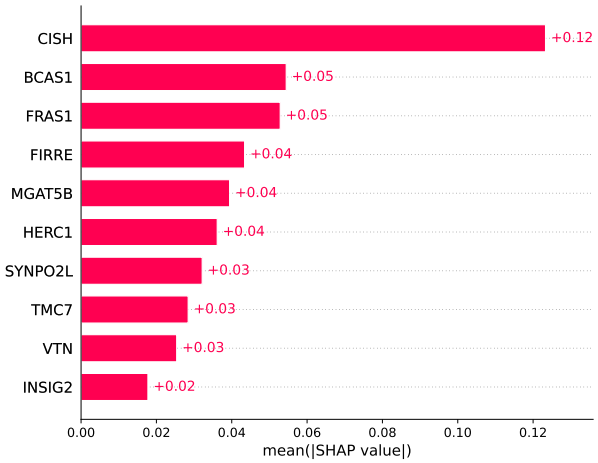

Figure 5.4: Global Gene Contribution – Mean Absolute SHAP Values Across All Patients

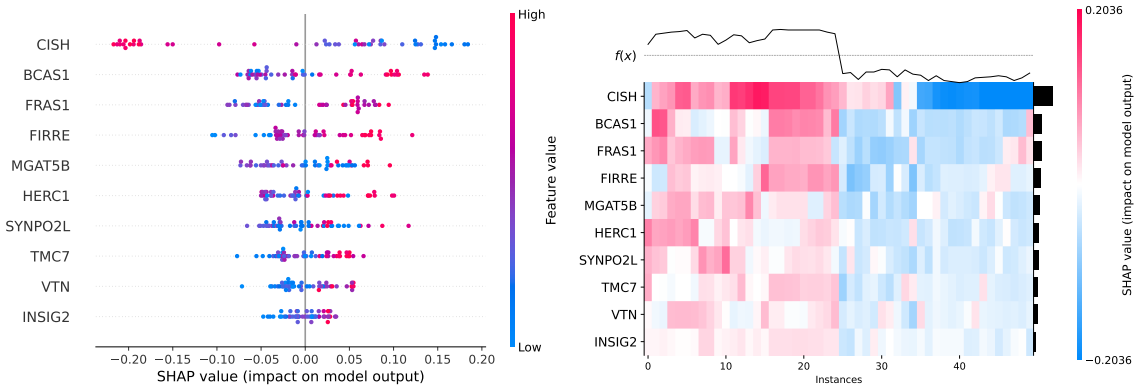

Figure 5.5: Gene Contribution in Resistant Cohort: Per-Gene SHAP Values in the Resistant Group (*Left:* Beeswarm Plot of Impact in Model Prediction, *Right:* Gene Contributions Across Instances)

ous cells. Moreover, [174] found the BCAS1 gene among a 4-gene signature of differentially expressed genes related to tamoxifen resistance in mammary carcinoma cells. Most recently, [37] also found the VTN, TMC7, INSIG2, HERC1, and FRAS1 genes among their succinct 6-gene signature, overperforming previous existing signatures in the analysis of the TCGA dataset.

Regarding the specific predictions made by the model as to whether a patient is resistant to tamoxifen, Figure 5.5 shows the SHAP contributions in terms of the expression value of each gene[2]. The beeswarm plot on the left-hand side shows the impact of each gene on the model's prediction, with the genes ordered by their mean SHAP value across all patients. The heatmap on the right-hand side displays the gene contributions across individual instances. As we can see, the CISH gene has the greatest discerning power in the model's predictions, lower expressions of the gene being associated with a higher probability of resistance to tamoxifen, while higher expressions are associated with a lower probability of resistance. For the other genes in our signature, the opposite is true: higher expressions are associated with a higher probability of resistance, while lower expressions are associated with a lower probability of resistance. From the figure on the right, we can see that an over-expression of the CISH gene carries most of the weight in the model's predictions of a low resistance probability.

[2]Note that the expressions were normalized and thus do not correspond directly to the values in the gene count matrix.

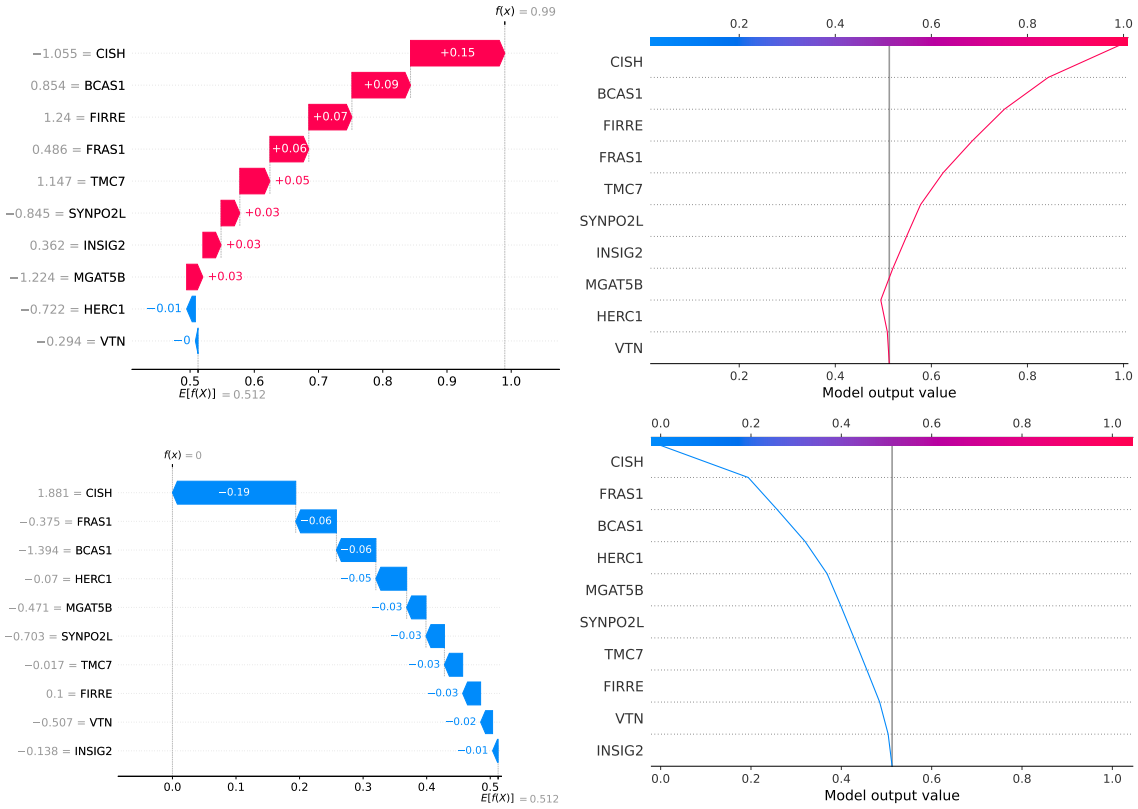

Figure 5.6: Local Gene Contribution – Per-Gene Contribution for Two Sample Patients (*Above:* Resistant, *Below:* Favorable Treatment) (*Left:* Waterfall Contribution Plots, *Right:* Decision Contribution Plots)

Finally, we can take a closer look at the local gene contributions for two sample patients in Figure 5.6. The top row corresponds to a patient predicted to be resistant to tamoxifen ($p(\text{Resistant}) = 0.99$), while the bottom row corresponds to a patient with a favorable treatment outcome ($p(\text{Resistant}) = 0$). The left-hand side shows the per-gene contribution for each patient, with the genes ordered by their SHAP value in the resistant cohort. The right-hand side shows the cumulative effect of genes on the model's predicted probabilities. This allows us to see how, for each patient, each gene contributes to the model's prediction and how the cumulative effect of genes influences the final prediction. In both cases, the average predicted probability across all patient instances (in the augmented dataset) is drawn (~ 0.512), with each gene contributing to the deviation towards the probability of that particular instance (see the gray line in the left plots). For example, for the first patient, the under-expression of the CISH gene increases the probability of resistance by 15%, followed by the over-expression of the BCAS1 gene increasing it by 9%. On the other hand, in the case of the non-resistant patient, the over-expression of the CISH gene decreases the probability of resistance by almost 20%. As we can see, however, the genes that have the greatest impact on the model's prediction for resistant patients are not necessarily consistent across both patients.

Kaplan-Meier Survival Analysis & Biological Pathways

Lastly, we can take the analysis of our potential biomarkers one step further by incorporating survival analysis on other available breast cancer datasets — in order to assess the prognostic value of these genes — or by exploring

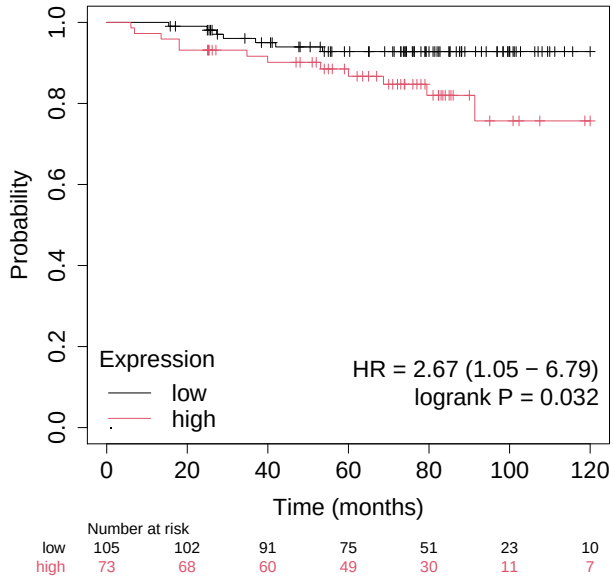

Figure 5.7: Kaplan-Meier Survival Analysis for the Identified Genes in ER+ Breast Cancer Patients – Relapse-Free Survival (RFS) Probability Over Time

the biological pathways in which these genes are involved, using tools such as Enrichr [175–177], to gain insights into the underlying mechanisms in which these genes participate.

In terms of genetic survival analysis, the Kaplan-Meier method (available here) is a non-parametric statistic used to estimate the prognosis of patients in terms of survival rates. It is a useful tool for comparing the survival of two or more groups, and can be used to assess the prognostic value of genetic biomarkers in breast cancer. By analyzing the survival curves of patients with high and low expression of the identified genes, we can determine whether these genes are associated with differences in patient survival. In our case, this serves as an independent validation tool to assess the prognostic relevance of our genetic biomarkers in an independent set of patients. Of course, in order to maintain the context of our study, the selected patients are also ER+ breast cancer patients who have undergone tamoxifen therapy. By doing so, we can run a Kaplan-Meier survival analysis on the selected genes over 178 new external validation patients.

Figure 5.7 shows the Kaplan-Meier survival curve for the identified genes in tamoxifen-treated ER+ breast cancer patients. The survival curve is split into two groups based on the mean expression of each gene: high expression (red) and low expression (black). The y-axis represents the probability of (relapse-free) survival, while the x-axis represents the time in months. The log-rank test is used to determine whether there is a significant difference in survival between these two groups. A p-value less than a given significance threshold α, e.g., 0.05, indicates a statistically significant difference in survival between the high and low expression groups; whereas the HR (Hazard Ratio) provides a measure of the risk in the high expression group compared to the low expression group. Specifically, the survival curve reveals a clear separation between the high and low expression groups of the gene signature, with the high expression group exhibiting a lower probability of relapse-free survival over time (i.e., they are more likely to relapse compared to the low expression group). The log-rank test indicates a statistically significant difference in survival between the two groups ($p = 0.032 < \alpha$). The hazard ratio (HR) is 2.67 (95% CI: 1.05 – 6.79), suggesting that patients with high expression of the gene signature are associated with a 2.67 times greater risk of relapse compared to those with low expression. This result indicates that the

combined expression of the identified genes — including the CISH gene which was found to be *underexpressed* in patients developing resistance to tamoxifen in our SHAP value analysis — is significantly associated with ER+ patients developing resistance to tamoxifen therapy and experiencing relapse. Nonetheless, this kind of analysis should be taken as more *qualitative* than *quantitative*. First, the sample size remains quite small, and the results should be interpreted with caution. Second, we do not know if the patients underwent the full treatment regime, if they also received chemotherapy, or if they had other comorbidities that could have influenced the results. Third, this analysis does not take into account the individual expression of each gene, but rather the mean expression of all the genes in the signature, disregarding differences in terms of over/under-expression. Lastly, the resulting confidence interval for the HR is quite wide, indicating some uncertainty in the magnitude of the risk increase.

Finally, we can explore the biological pathways in which these genes are involved using Enrichr, a web-based tool for gene set enrichment analysis (available here). Enrichr provides a large collection of gene set libraries and tools for pathway analysis, allowing us to identify the underlying biological pathways and processes in which our genes might be involved. By analyzing the enriched pathways, we can hopefully gain insights into the underlying mechanisms in which these genes participate. For instance, these genes appear to be involved in the Human ECM-receptor interaction pathway (remember Figure 3.2), which has been shown to play a critical role in breast cancer progression and survival [178–180]; and in the Interleukin-7 (IL-7) signaling pathway, also known to be involved in promoting breast cancer cell proliferation [181–184]. Unfortunately, our signature is quite small and thus it is difficult to extract any meaningful pathways from the analysis. Likewise, there are no reported pathways related to tamoxifen resistance in the Enrichr database, which makes it difficult to interpret the results. Furthermore, understanding and interpreting the biological mechanisms in which these genes are involved is a complex task that requires a deep understanding in the field of genetics and cellular biology, which is far beyond the scope of this study.

6 Planning, Budgeting & Ethical Considerations of the Project

"It's the job that's never started that takes longest to finish."

~ J.R.R. Tolkien, *The Lord of the Rings: The Fellowship of the Ring* (1954) [185]

In this chapter, we present the planning, budgeting, and ethical considerations of the project. First, we present the human resource plan, where we define the roles and responsibilities of the members involved in the project. Then, we present the working plan, where we define the tasks and milestones of the project, as well as the relevant dates and chronogram of the project. After that, we will focus on the budget of the project, where we present the estimated costs of the project broken down by category. Finally, we provide a reflection on the ethical considerations of the project, including the responsible use of clinical data and open-source software, patient privacy protection in breast cancer research, and the potential impact of our project on the patients, society, and the healthcare system.

6.1 Planning & Budgeting of the Project

As illustrated in Figure 6.1, the successful execution of our project requires careful planning and a thorough understanding of the resources and human capital needed to achieve the project objectives. This section outlines the key aspects of project management, including the human resources involved, the detailed working plan, and the financial considerations. The planning approach follows standard project management methodologies while being specifically tailored to the unique requirements of our tamoxifen modeling task. We hereby present a comprehensive overview of how the project will be structured and executed, ensuring that all technical, human, and financial resources are optimally utilized to achieve the project objectives.

6.1.1 Human Resource Plan

Before we can start developing the working plan and the budget, we need to carefully outline the human resources involved in the project. Despite the complexity of the project in terms of its wide scope — the work presented here as our thesis is only a small *subset* of what has been done in the whole project, and thus some of the tasks, such as those involving the RNA sequencing, were not explicitly performed for this work — and multi-institutional nature, we summarize in Table 6.1 below, to the best of our knowledge, the critical roles and responsibilities of the members involved in the project as well as the relevant skills and expertise of each member.

The project brings together a diverse team of experts from three key institutions, each contributing their unique expertise to ensure the project's success. At the University of Deusto, the Thesis Supervisor provides aca-

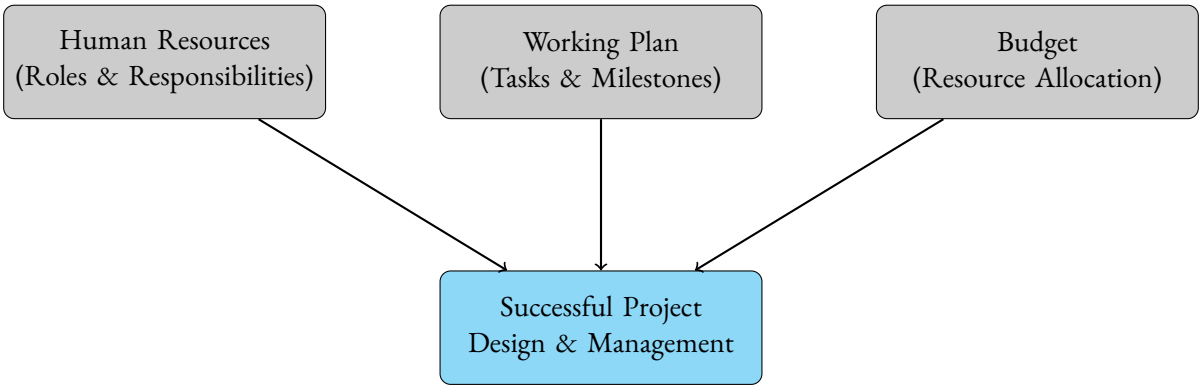

Figure 6.1: Key Components of Project Management & Their Relationship to Project Success

demic oversight and validates the research methodology, while the Student (who also serves as the Researcher at BCAM) is responsible for the core technical implementation, including mathematical modeling, data analysis, and exhaustive documentation of the developed methods. Moreover, he is also responsible for the development of the computational methods, including the implementation of the Hamiltonian Monte Carlo algorithms, the development of the pyHaiCS library, and the integration of biological knowledge with mathematical models.

Conversely, at BCAM, the Research Supervisor offers technical guidance and expertise in applied mathematics and Bayesian modeling in the context of computational biology. The Research Collaborators contribute specialized knowledge in Bayesian computation, particularly in Hamiltonian Monte Carlo methods, and provide crucial support for benchmarking as well as insight on the theoretical properties of the proposed methods. Last but not least, the CIC bioGUNE Research Center adds essential biological and clinical expertise to the project. The Clinical Advisor ensures the medical relevance and validity of our findings, while the Lab Technician handles the technical aspects of RNA sequencing and sample processing[1]. The Researcher in Bioinformatics, with previous experience in developing this type of prognostic models in the context of breast cancer, plays a crucial bridging role, integrating biological knowledge with mathematical models and providing expertise in RNA-seq analysis and cellular biology.

This multidisciplinary team structure ensures that our project benefits from expertise across mathematical modeling, clinical knowledge, and biological understanding, creating a strong and robust foundation for developing and validating our Bayesian approaches and identifying relevant potential genetic biomarkers responsible for tamoxifen resistance in the context of ER^+ breast cancer patients.

6.1.2 Working Plan

In this section, we present the working plan of the project, including the tasks and milestones of the project, which outline the systematic approach taken to develop our Bayesian framework for predicting tamoxifen resistance. The plan is structured into three main phases, each with specific objectives and milestones that align with both the academic requirements of this thesis and the research goals of the project. This structured approach ensures the methodical development of our mathematical models, the implementation of computational methods, and the validation of our findings through biological interpretation. The timeline and milestones are de-

[1] As stated before, all the sequencing was in practice performed for other tasks we developed in the context of our work at BCAM. In any case, we have decided to include it here as it is a crucial part of the project.

Table 6.1: Human Resource Plan: Roles, Responsibilities & Expertise

Role	Responsibilities	Key Skills & Expertise
University of Deusto — Academic Institution		
Thesis Supervisor	Project supervision and guidance; Academic oversight; Research methodology validation	Academic research; Mathematical modeling; Project management; University procedures
Student	Implementation of mathematical models; Data analysis; Documentation; Thesis development	Programming; Scientific writing; Data analysis; Machine Learning
BCAM (Basque Center for Applied Mathematics) — Research & Funding		
Research Supervisor *(Project Manager)*	Technical guidance; Mathematical model development; Research methodology	Applied mathematics; Computational biology; Research supervision
Researcher *(Same as Student Above)*	Development of Bayesian models; Implementation of statistical methods; Data analysis and visualization; Collaboration with clinical partners; Scientific documentation; Research methodology development	Mathematical modeling; Scientific programming; Statistical analysis; Computational statistics, Machine Learning
Research Collaborators	Technical guidance on HMC methods; Benchmarking support for pyHaiCS library; Methodological insights for deployed algorithms	Bayesian computation; MCMC methods; Performance optimization; Scientific software development
CIC bioGUNE Research Center — Research Partner		
Clinical Advisor	Clinical data interpretation; Biological validation; Medical insights	Clinical oncology; Breast cancer research; Medical expertise
Lab Technician	RNA sequencing; Sample processing; Experimental data collection	Laboratory techniques; Molecular biology; Data collection
Researcher in Bioinformatics	Integration of biological knowledge with mathematical models; RNA-seq data analysis; Biological pathway interpretation	Computational biology; Cellular biology; RNA-seq analysis; Mathematical modeling

signed to accommodate the complexity of the mathematical modeling process while maintaining academic rigor and research quality. The three phases are defined as follows:

1. **Definition & Planning**: This phase includes the definition of the project scope, the development of the work plan, and the identification of the resources needed to achieve the project objectives.

Table 6.2: Project Tasks by Phase

Code	Task Name	Description
	Phase 1: Definition & Planning	
T1.1	Project Scope Definition	Detailed definition of project objectives, milestones, and success criteria. That is, defining the *scope* of the project
T1.2	Resource Planning	Identification and allocation of human, technical, and financial resources (e.g., computational resources, collaborators, travel expenses to visit CIC bioGUNE, etc.)
T1.3	Kick-off Meeting with CIC bioGUNE	Collaboration with CIC bioGUNE to discuss the available clinical data and the biological motivation behind the project
T1.4	*State-of-the-art* Review	Review of the current literature on tamoxifen resistance prediction and Bayesian modeling in the context of breast cancer diagnosis
T1.5	Domain Knowledge Acquisition	Learning and preparation in key areas: e.g., cellular biology, cancer research, Hamiltonian Monte Carlo methods, and Bayesian modeling
T1.6	RNA Data Collection	Gathering and organization of multi-source RNA-seq data
	Phase 2: Development	
T2.1	RNA-seq Pre-Processing	Pre-processing of the RNA-seq data to be used for the development of the models
T2.2	pyHaiCS Library Development	Implementation of computational methods and algorithms in the pyHaiCS library
T2.3	Mathematical Model Development	Design and implementation of (Bayesian) models for tamoxifen resistance prediction
T2.4	Biological Validation	Integration of biological validation knowledge on model predictions (e.g., biological pathway enrichment analysis, survival analysis, etc.)
T2.5	Documentation	Development of technical documentation for pyHaiCS library and methods
	Phase 3: Project Closure & Control	
T3.1	Results Presentation	Presentation of project results to *stakeholders* (i.e., project supervisors, research partners, potential future collaborators, etc.)
T3.2	Thesis Development	Writing and preparation of the final thesis document
T3.3	Prepare Defense	Preparation of the thesis defense
T3.4	Project Defense	Presentation and defense of the thesis project
T3.5	Project Dissemination	Publication of results and documentation of project outcomes

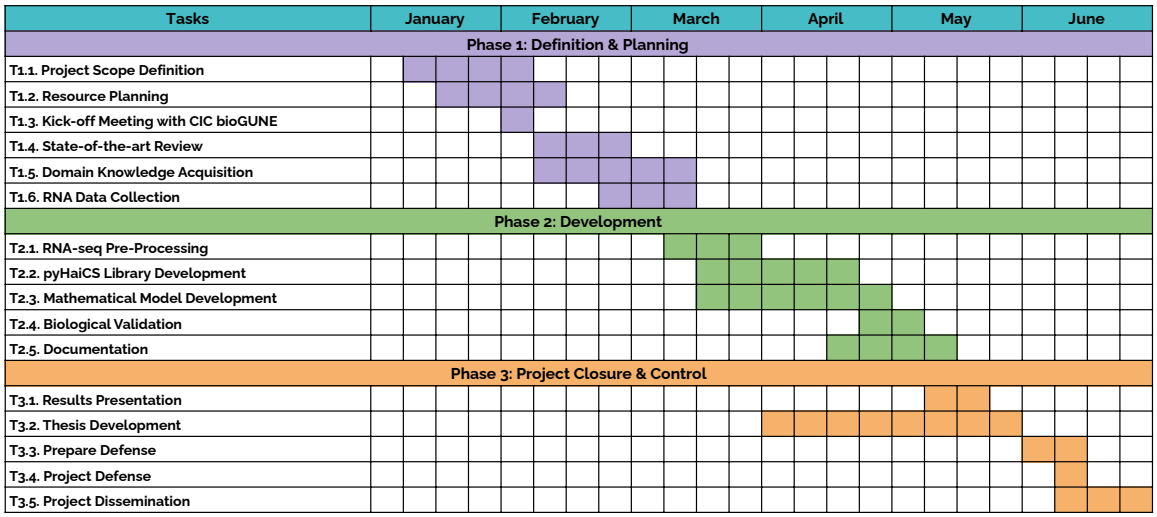

Figure 6.2: Project Chronogram: Tasks & Milestones – Weekly Gantt Chart

2. **Development**: This phase includes the development of the project deliverables, including the development of the mathematical models, the implementation of the computational methods in the pyHaiCS library, and the analysis of the data. Also, this phase includes the development of the pyHaiCS documentation, and the biological validation of the proposed models.

3. **Project Closure & Control**: This phase includes the closure of the project, including the presentation of the results and the dissemination of the project. Likewise, from the academic perspective, this phase includes the preparation of the thesis and its defense.

The tasks and phases of the project are presented in Table 6.2, whereas the *weekly* chronogram for the development of the project is presented in Figure 6.2 as a Gantt chart.

6.1.3 Budget of the Project

Once the working plan of our project has been defined, and the human resources involved in the project have been identified, we can proceed to the development of the budget of the project. In this section, we present the estimated costs of the project broken down by category. In order to present the budget in a comprehensive manner, we split the budget into two main groups: technical equipment & travel expenses (including RNA sequencing), and human resources.

Table 6.3 outlines the projected costs for essential technical equipment (e.g., computing hardware, storage, accessories), estimated expenses related to RNA sequencing[2], and anticipated travel costs for necessary coordination meetings (e.g., visits to CIC bioGUNE). All listed costs within this table are presented inclusive of the applicable Value Added Tax (IVA, in Spain). Likewise, complementing the material and operational costs, Table 6.4 provides a detailed breakdown of the human resources budget. These costs — reflecting the collaborative nature of the project by encompassing contributions from all three key institutions: the University of Deusto, BCAM (Basque Center for Applied Mathematics), and the CIC bioGUNE Research Center — are calculated

[2]The costs associated with RNA sequencing, including sample preparation, library preparation, and sequencing itself, are estimations derived from publicly available vendor information and provided approximations by technical experts. They serve to provide a sense of scale for these significant experimental costs, rather than representing a precise quote.

Table 6.3: Technical Equipment & Travel Expenses Budget

Item	Quantity	Unit Price (€)	IVA (21%)	Total (€)
Technical Equipment				
MacBook Pro Laptop	1	2,199.00	461.79	2,660.79
External SSD 1TB	1	80.00	16.80	96.80
USB-C Hub	1	25.00	5.25	30.25
Monitor	1	170.00	35.70	205.70
Other Accessories	1	70.00	14.70	84.70
RNA Sequencing				
MCF7 Sample Preparation (Culture, Extraction, QC)	1	1,500.00	315.00	1,815.00
Library Preparation (TruSeq Stranded Total RNA)	1	2,400.00	504.00	2,904.00
Cell Sequencing (Illumina HiScan-SQ, SE50)	1	1,500.00	315.00	1,815.00
Travel Expenses				
Taxis to CIC bioGUNE (round trip)	15	30.00	6.30	544.50
		Total Technical & Travel Expenses:		**10,156.74€**

based on the planned hours of dedication for each role and their varying levels of seniority, the corresponding institutional hourly rates, and an estimated overhead of 30% to account for social security contributions and other associated employer costs.

As a summary, the total projected cost for technical equipment, RNA sequencing, and travel expenses, as presented in Table 6.3, amounts to around 10,156.74€. Concurrently, the total estimated cost attributed to the dedicated human capital, covering personnel from all participating institutions, and detailed in Table 6.4, is calculated at 16,770.00€. Therefore, summing these figures, the overall estimated budget for the successful completion of this project should be around the **26,926.74€** mark.

6.2 Ethical Reflection & Assessment of the Study

Engineering has always had a long-standing tradition of service to society — as well as an established ethical obligation to health, safety, and public welfare — and carries inherent ethical obligations and responsibilities towards the public. We, as engineers, scientists, or researchers, are expected to make decisions and design so-

Table 6.4: Human Resources Budget (by Institution & Role)

Role	Hours	Hourly Rate (€)	Social Security (30%)	Total (€)
University of Deusto — Academic Institution				
Thesis Supervisor	40	30.00	360.00	1,560.00
Student/Researcher	400	20.00	2,400.00	10,400.00
BCAM (Basque Center for Applied Mathematics)				
Research Supervisor	30	30.00	270.00	1,170.00
Research Collaborators	10	20.00	60.00	260.00
CIC bioGUNE Research Center				
Clinical Advisor	20	40.00	240.00	1,040.00
Lab Technician	30	20.00	180.00	780.00
Bioinformatics Researcher	40	30.00	360.00	1,560.00
			Total Human Resources:	**16,770.00€**

lutions that are consistent and compliant with these basic ethical grounds, and "*to disclose factors that might endanger the public or the environment*" [186]. This ethical framework is particularly relevant in the field of biomedical research, where our work has direct implications for patient care and clinical decision-making. In this section, we reflect on the ethical dimensions of our project and its potential societal implications.

This ethical responsibility is not limited to the technical aspects of our work, but also extends to the *social* and *environmental* dimensions of our research. As engineers, we are responsible for ensuring that our work is conducted in a manner that respects the rights and dignity of individuals, promotes social justice, and minimizes harm to the environment. This includes considering the potential impact of our research on vulnerable populations, ensuring equitable access to healthcare technologies, and promoting sustainability in our work. Besides, we are not only responsible for the implications of our own work, but also for the work of our organizations and the broader engineering/scientific community towards clients, employers, and society in general. This responsibility encompasses a commitment to ethical conduct, transparency, and accountability in all aspects of our work.

While these spheres of responsibility establish an overall ethical framework for our project, we can further ground it in terms of the specific principles of professional ethics that *should* guide our decision-making and implementation. By applying these principles, we ensure our technical work remains aligned with our ethical obligations to patients, the healthcare system, and society at large: the stakes not only include scientific advancement but also direct impact on human lives and well-being.

Three Spheres of Responsibility

Our ethical approach encompasses three interconnected spheres of responsibility:

1. **Social Responsibility:** Our primary social obligation is to develop tools that genuinely seek to improve patient diagnosis and understand the underlying mechanisms of tamoxifen resistance. We aim to create models that can be used in clinical practice, ultimately benefiting patients and healthcare providers. We recognize that this sort of prediction models can impact treatment decisions with life-altering consequences. Therefore, we prioritize model interpretability alongside accuracy, enabling clinicians to understand the basis for predictions.

2. **Environmental Responsibility:** While computational research has a lower direct environmental impact than laboratory work, we acknowledge the energy consumption of computational resources, specially when training large Deep Learning models. In any case, our study deals with a very reduced set of data (i.e., 6 sequenced lab cells and less than 40 patients), and the developed models are designed to be efficient, minimizing the computational burden. We also promote the use of open-source software, which can reduce the need for expensive proprietary tools and encourage collaboration.

3. **Economic Responsibility:** In this line, by developing open-source software and transparent methodologies, we promote cost-effective research that reduces duplication of efforts and democratizes access to advanced analytical techniques across institutions with varying resource levels.

Ethical Framework & Principles

Our ethical assessment is guided by four fundamental principles of professional ethics:

- **Beneficence:** This project aims to benefit breast cancer patients through improved prediction of tamoxifen resistance, potentially sparing them long ineffective treatments. We uphold professional competence and integrity by ensuring our models are rigorously tested and validated.

- **Justice:** We consider the fair distribution of benefits resulting from this research. Access to improved predictive tools should not create or exacerbate healthcare disparities. The open-source nature of our pyHaiCS library promotes equity by making advanced computational methods freely available to the research community.

- **Autonomy:** We respect patient autonomy by ensuring our approach supports informed decision-making. By providing transparent explanations of our models, we enable clinicians to properly inform patients, avoiding paternalistic medicine and promoting shared decision-making.

- **Responsibility:** We acknowledge both individual and social responsibilities inherent to our work. As engineers and researchers, we are responsible for the technical validity of our models and their appropriate application in healthcare contexts.

In addition to these theoretical ethical and moral foundations, there are also actual practical implications and design choices that ensure ethical compliance and responsibility throughout the project. These include:

- **Data Ethics and Privacy Considerations:** Working with genomic and clinical data requires rigorous ethical safeguards. All patient data used in this project is fully anonymized in compliance with GDPR (General Data Protection Regulation) and relevant European bioethical frameworks, maintaining no connection between genetic profiles and patient identities. We exclusively utilize patient data collected with proper informed consent protocols, where patients understood the potential research applications of their genetic information. In practice, all patient data used for this study is publicly available and has undergone every anonymization step required for publication.
- **Open Science and Democratization of Technology:** Our commitment to open-source development of the pyHaiCS library aligns with ethical principles of scientific transparency and equitable access. By making our computational methods freely available, we promote scientific reproducibility, reduce barriers to advanced computational methods for resource-limited institutions, facilitate collaborative improvement of methodologies, and accelerate the pace of discovery by enabling others to build upon our work. This project supports the democratization of technology in healthcare research, potentially reducing global disparities in cancer research capabilities.
- **Potential Societal Impact:** The broader societal implications of our work extend beyond individual patients. More precise prediction of treatment response could reduce healthcare costs associated with ineffective treatments. Our methodological contributions may benefit researchers working on other predictive problems in medicine. By adhering to ethical principles and maintaining transparency, we help foster public trust in computational approaches to personalized medicine.

Practical Ethical Implementation

Our project translates ethical principles into practical implementation through:

- **Privacy Protection:** Rigorous anonymization protocols and secure data handling that comply with GDPR and bioethical frameworks
- **Open Science:** Development of pyHaiCS as open-source software to democratize access to advanced analytical techniques
- **Transparency:** Maintaining interpretable models and documentation to support informed clinical decision-making
- **Regulatory Alignment:** Ensuring compliance with European frameworks for genetic data handling and biomedical research

Through these measures, we ensure our technical work not only advances scientific knowledge but does so in a manner that respects patient dignity and upholds the highest standards of research ethics.

Data Biases & Gender Perspective

As with any study involving clinical data, particularly in the context of breast cancer research where the patient population is predominantly female and has historically had a severe *under-representation* of ethnic minorities [187], it is crucial to acknowledge the potential biases and limitations inherent in the datasets used. While this project sought to analyze and process information in an impartial manner, several forms of bias may still persist in the *anonymized* data, potentially affecting the generalizability of the findings. For instance, ethnic and socio-economic biases could arise depending on the demographic composition of the source data, particularly if certain groups were underrepresented or overrepresented in the sample. Survival bias is also a potential concern, especially in retrospective datasets where only entities that endured a specific period are available for analysis, possibly skewing conclusions. Additionally, gender bias must be acknowledged, particularly if the data or design reflects historically ingrained gender asymmetries in participation or representation. For example, the TCGA-BRCA dataset used throughout this work has been criticized for its lack of diversity, particularly in terms of ethnical representation [188], gender diversity [189] — with a 1 to 100 ratio of male to female patients —, and socio-economic status [190].

Data Bias & Representation Considerations

It is critical to acknowledge potential biases within our research datasets that could impact the generalizability and clinical applicability of our findings:

- **Demographic Representation:** The TCGA-BRCA dataset, though valuable, exhibits notable limitations in ethnic diversity, with predominant representation of Caucasian populations [188]. However, it has been shown that survival outcomes and treatment responses can vary significantly across different ethnic groups [191].
- **Survival Bias:** Our dataset inherently contains a form of survival bias, as data collection necessitates patients surviving long enough to provide follow-up information. This may lead to underrepresentation of the most aggressive resistance phenotypes, potentially skewing our understanding toward less lethal resistance mechanisms.
- **Socio-economic Factors:** Access to specialized cancer care and enrollment in research studies correlates with socioeconomic status, potentially resulting in datasets that incompletely represent disadvantaged populations. The generalizability of our models may therefore be limited in resource-constrained healthcare settings.
- **Age Distribution:** The age distribution within our patient cohort may not proportionally represent the full spectrum of breast cancer patients, particularly very young or elderly patients who are often underrepresented in clinical trials and databases.

Gender Perspective in Breast Cancer Research

While breast cancer predominantly affects women, it is essential to adopt a *gender-inclusive* perspective in research in the field. This ensures that the unique experiences and needs of all individuals affected by breast cancer, including men and transgender individuals, are meaningfully considered.

- **Beyond Binary Classifications:** Though breast cancer is often framed as a "*women's disease*", it affects individuals across the gender spectrum. Male breast cancer accounts for approximately 1% of cases [189], and transgender individuals — particularly those undergoing hormone therapy — may have unique risk profiles that remain understudied.
- **Sex as a Biological Variable:** We explicitly acknowledge the distinction between sex as a biological variable and gender as a social construct. Treatments like tamoxifen target biological pathways such as estrogen receptors; however, responses to endocrine therapy can vary based on a range of factors beyond binary sex assignment, including genetic variation, epigenetic influences, and hormone use in transgender individuals.
- **Inclusive Research Design:** We advocate for research methodologies that intentionally include diverse gender identities. This enhances the external validity of findings and ensures that clinical insights are relevant and responsive to the needs of all individuals affected by breast cancer.

7 Conclusions & Future Work

"Prince John: 'Are you finished?'
Sir Robin of Locksley: 'I'm only just beginning.'"

~ The Adventures of Robin Hood (1938) [192]

In this final chapter, we synthesize the principal findings related to our deep investigation, through Bayesian methodologies, into the field of breast cancer treatment, specifically on the challenges presented by tamoxifen resistance in estrogen receptor-positive (ER^+) breast cancer cells. This *multi-disciplinary* endeavor has spanned a wide range of topics from the fields of cellular biology, bioinformatics, and computational statistics — from differential expression analysis and biomarker identification, to sophisticated Hamiltonian Monte Carlo implementations and deep probabilistic modeling; all with the overarching goal of enhancing predictive capabilities while preserving *interpretability*. Finally, culminating in the development of a novel Python library, pyHaiCS, designed to facilitate the implementation of Hamiltonian Monte Carlo (HMC) methods for Bayesian inference. We summarize below the key findings and contributions of this work, critically assess its strengths and limitations, and outline promising avenues for future research, delineating both *methodological* and *biological* refinements that could potentially further advance the work on precision oncology for breast cancer treatment.

7.1 Conclusions & General Assessment

Throughout this thesis, we embarked upon the challenging task of uncovering potential *genetic biomarkers* indicative of *tamoxifen resistance* in ER+ breast cancer patients through a comprehensive analysis of both cell-line and patient data combined with advanced Bayesian modeling techniques grounded in Hamiltonian Monte Carlo (HMC) sampling methods. The primary purpose was to move beyond traditional *point-estimate* approaches by incorporating *prior* biological knowledge derived from cell-line experiments into models trained on patient genomic data, thereby aiming for more robust and interpretable predictions in a context plagued by data scarcity, high dimensionality, and class imbalance. This work is particularly relevant given the increasing prevalence of breast cancer and the pressing need for personalized treatment strategies that can effectively address the challenges posed by drug resistance and heterogeneity in patient responses, specifically, considering the long scale of the treatment process of these endocrine therapies. Moreover, a significant component of this study involved the development of pyHaiCS, our novel Python library specifically designed to facilitate the use of sophisticated HMC-based sampling techniques within the computational statistics domain.

At the core of our research was the analysis of RNA sequencing data from both MCF7 cell-line experiments and The Cancer Genome Atlas (TCGA) patient cohort. This dual approach allowed us to leverage both *in-vitro* and *in-vivo* data, providing a more complete view of the genetic mechanisms underlying this biological

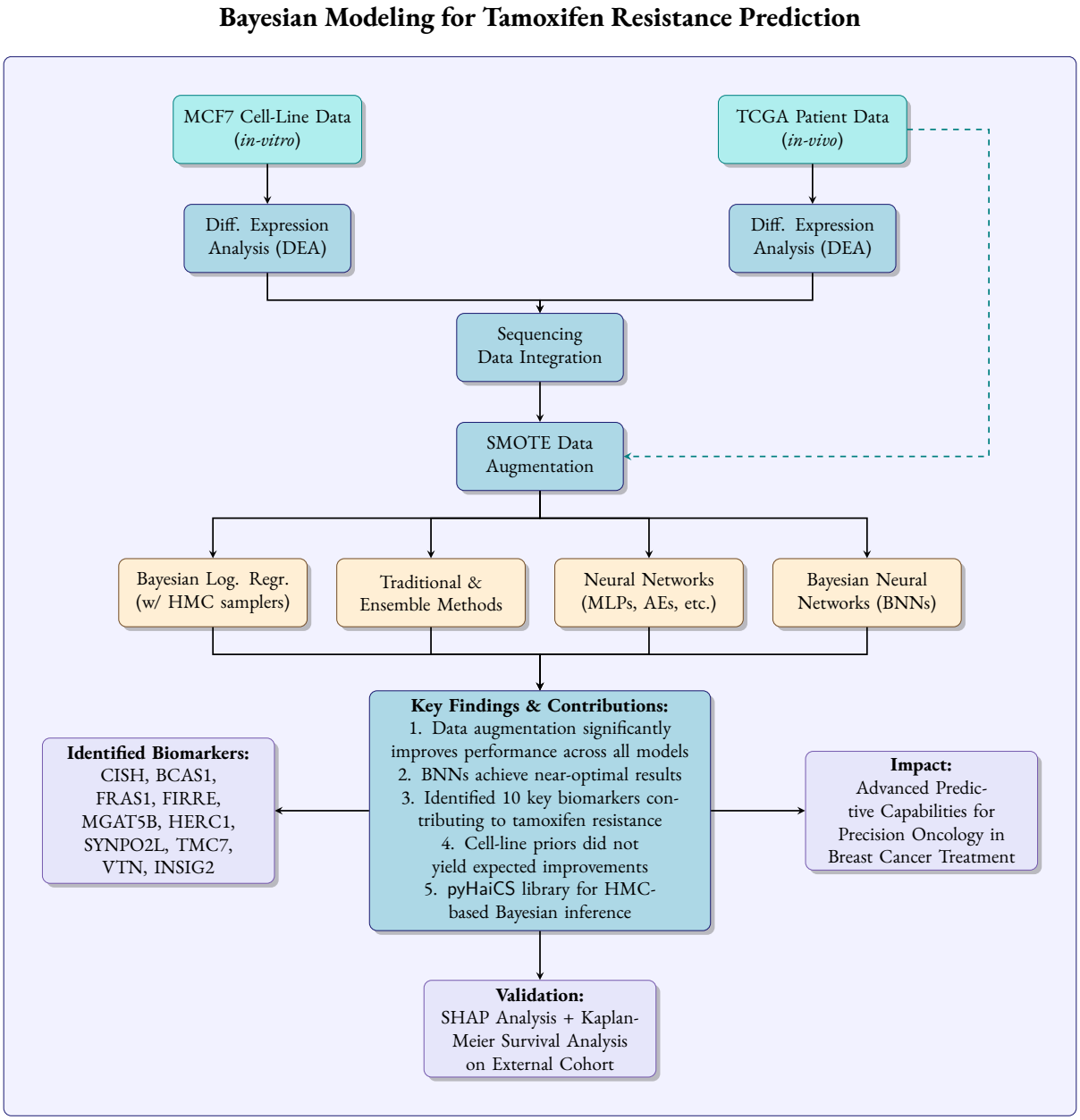

Figure 7.1: Visual Summary of the Project

phenomenon. The integration of these datasets was not without its challenges, particularly in terms of addressing the inherent class imbalance and high dimensionality associated with genomic data. Nonetheless, through careful and rigorous pre-processing, and a combined differential expression analysis, we were able to identify a refined set of *candidate biomarkers* (i.e., **CISH, BCAS1, FRAS1, FIRRE, MGAT5B, HERC1, SYNPO2L, TMC7, VTN, INSIG2**) that exhibited concordant expression changes across both datasets. Subsequently, a diverse array of predictive models was developed and evaluated, ranging from Bayesian Logistic Regression (BLR) implemented using various HMC samplers (HMC, GHMC, s-AIA tuned) within pyHaiCS, to traditional shallow classifiers, ensemble methods, and deep learning architectures including standard Multi-Layer Perceptrons (MLPs), Autoencoders, Variational Autoencoders (VAEs), and Bayesian Neural Networks (BNNs). The performance of these models was rigorously assessed using a variety of metrics with a particular emphasis on re-

call, due to its critical importance in clinical settings where false negatives can have severe consequences, and Matthew's Correlation Coefficient (MCC), which provides a robust measure of performance in classification tasks with imbalanced datasets.

Our key findings highlight the fundamental importance of addressing the **class imbalance** intrinsic to the patient dataset (few recorded resistant patients against a majority of sensitive patients) and the potential of **data augmentation** techniques to enhance model performance. The application of Synthetic Minority Oversampling Technique (SMOTE) data augmentation techniques consistently, and significantly, improved the performance across all tested models. Notably, probabilistic models, especially BNNs, demonstrated superior performance on augmented data, achieving near-optimal recall and MCC scores. Interpretability analysis, using SHAP values on a high-performing Random Forest model, identified key genes as major contributors to resistance prediction, aligning with existing literature and providing potential biological insights. Furthermore, Kaplan-Meier survival analysis on an external cohort of patients provided *independent validation* of the prognostic relevance of the derived gene signature. However, a crucial observation was the *performance plateau* of the BLR-HMC models incorporating cell-line priors, suggesting that the direct transfer of this specific prior information did not yield the expected improvement over models trained solely on augmented patient data, hinting towards a potential pitfall in integrating heterogeneous biological data sources and translating cell-line findings directly to patient populations, as had been done in previous work [37].

As a summary, this work successfully demonstrates the feasibility, and potential benefits, of employing advanced Bayesian modeling techniques, particularly BNNs, combined with appropriate data augmentation for predicting tamoxifen resistance. We provide a robust framework for biomarker discovery, model evaluation, and interpretation, contributing both methodological insights and a practical computational tool (pyHaiCS), not only for this specific application but also for broader applications in the field of computational statistics. While the direct integration of cell-line priors proved less impactful than anticipated, the overall results significantly advance predictive capabilities compared to previous work and, hopefully, lay a strong foundation for future research. A visual summary of the project is presented in Figure 7.1, illustrating the key components and findings of our work.

7.2 Future Work & Promising Research Directions

Building upon the findings and limitations of this thesis, several avenues for future research emerge, spanning both the biological and mathematical modeling domains. Below, we outline some *potential* directions for **future work** that could be pursued to further enhance our understanding of this resistance phenomenon, and improve predictive modeling in breast cancer as a whole. All proposed future research directions are summarized in Figure 7.2.

Patient-Centric Signature Discovery Given the observed limitations in directly translating cell-line priors, future efforts should focus on discovering prognostic signatures derived *exclusively* from patient data. This involves exploring the high-dimensional gene space within patient cohorts (e.g., TCGA [188], SCAN-B [193], or METABRIC [194]) using techniques such as Gaussian Mixture Models (GMMs) — in order to cluster similarly expressed genes — combined with an optimization algorithm to identify the most relevant gene combination according to some *fitness* metric within the gene pool (after previous pruning within each cluster).

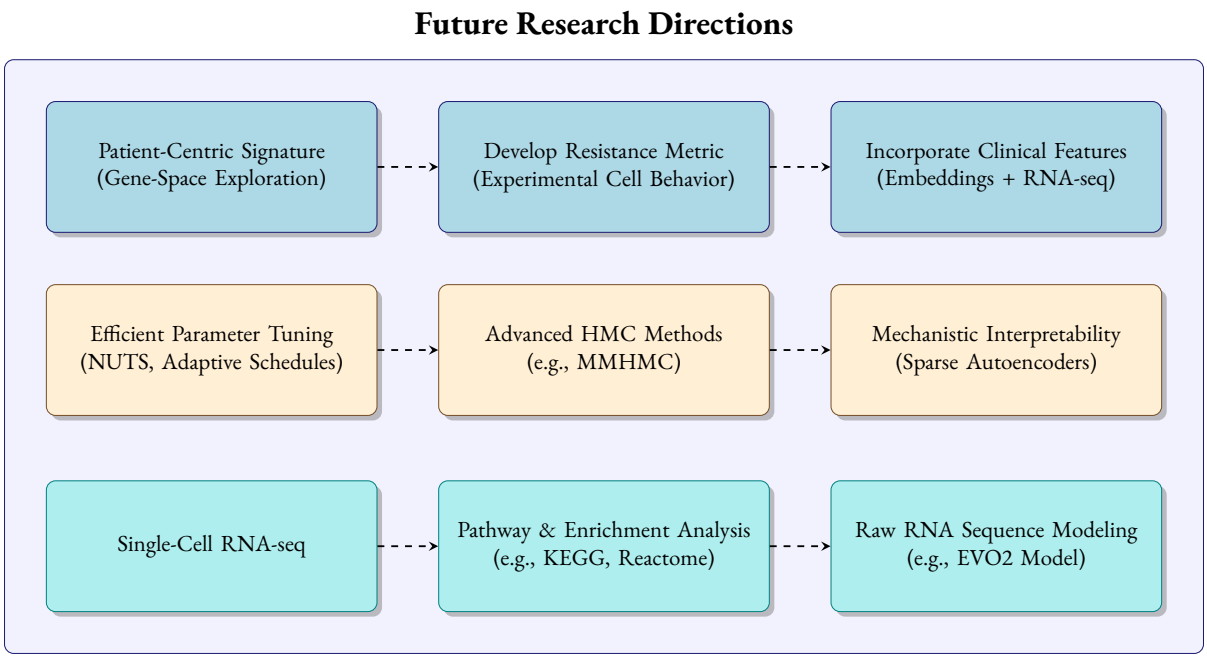

Figure 7.2: Visual Summary of Future Work Directions

This would allow, in practice, for the identification of robust, patient-derived, meaningful signatures that might better capture clinical reality. Such approaches can be readily extended to investigate resistance mechanisms for other endocrine therapies or different cancer types.

Refining Tamoxifen Resistance Quantification The binary classification of resistance, while practical, oversimplifies the biological reality. Future work could involve developing a *continuous* **resistance metric** by integrating quantitative experimental data from cell lines (e.g., mammosphere formation assays, proliferation, invasiveness, or migration). This continuous granular score could then serve as the target variable for regression models to find richer, more biologically grounded results. For instance, Double Machine Learning (DML) [195] could be employed to estimate the causal effect of our genes on the resistance score, allowing for a more nuanced understanding of the underlying biology.

Incorporating Richer Clinical Features Our current models rely exclusively on RNA-seq data. However, publicly available clinical datasets, such as TCGA [188] or METABRIC [194], contain an abundance of additional information including patient records (e.g., age, stage, treatment history), pathological reports, among others. Integrating this information could significantly enhance predictive accuracy. For instance, Natural Language Processing (NLP) models could be employed to generate embeddings in a latent space from these clinical features, which can then be fused with the genomic data to improve predictive performance.

Improving Parameter Tuning Methods The current implementation of the s-AIA sampler in pyHaiCS is computationally expensive and demanding — each iteration requires a full HMC simulation with a particular set of parameters, and thus a full recompilation of the JAX components. While it demonstrates superior exploration capabilities, its efficiency could be improved. Future work should focus on optimizing these tuning methods to reduce computational overhead while maintaining, or even improving, sampling efficiency.

This could involve exploring alternative adaptive tuning schemes. For instance, the well-known No-U-Turn Sampler (NUTS) [72] could be integrated into the library to provide a more efficient alternative.

Exploring Advanced HMC Variants While pyHaiCS currently implements several HMC variants, there is room for further improvement. Future work could involve implementing more advanced samplers such as the Mix & Match HMC (MMHMC) [51] presented in Section 3.3, or Langevin-based methods (e.g., Stochastic Gradient Langevin Dynamics (SGLD) [155], Stochastic Gradient HMC (SGHMC) [196]). Additionally, incorporating methods for non-separable Hamiltonians would broaden the library's applicability to problems far beyond standard statistical modeling, such as **molecular dynamics**, where HMC was originally conceived. This could also include the implementation of methods that approximate the Hamiltonian using neural networks as briefly discussed in Section 4.5.5. For instance, those that estimate the Hamiltonian directly, or those methods that, given an initial position and momentum, will estimate the resulting momentum and position after the integration process.

Mechanistic Interpretability The current SHAP analysis provides valuable insights but represents only a starting point for understanding the model's internal decision-making process. Future work might delve deeper into the mechanistic interpretability of the deep learning models, for instance by employing *state-of-the-art* techniques such as Sparse Autoencoders (SAEs) [197–199] to analyze the internal representations and activations of the learned models.

From Bulk to Single-Cell RNA-seq The current analysis is based exclusively on bulk RNA-seq data, which *aggregates* gene expression across all cells in a sample. However, single-cell RNA-seq (scRNA-seq) provides a more granular view of gene expression at the *individual* cell level. Future work could involve adapting our models to analyze scRNA-seq data, allowing for a more detailed understanding of cellular heterogeneity and its implications for tamoxifen resistance.

Gene Set Enrichment & Pathway Analysis The current analysis focuses on individual genes. However, biological processes are often mediated by complex interactions among multiple genes. Future work could involve pathway analysis to identify gene sets associated with specific biological functions or processes, such as those available in KEGG [200] or Reactome [201]. Understanding the underlying biological and molecular functions of these genes could provide a broader scope for the analysis and potentially reveal novel insights into our resistance phenomenon.

Modeling Raw Sequencing Data Currently our methods rely on gene expression counts derived from aligning reads to a reference genome. An alternative approach involves analyzing the **raw RNA sequences** directly using sequence-based deep learning models such as the recently released Transformer-based EVO2 [202]. This would allow for the extraction of meaningful embeddings from the genetic sequencing data, potentially leading to improved predictive performance. Conversely, we might even use the model itself to identify mutations down at the **local nucleotide level**. This would be the ultimate goal of the work, as it would show the exact mutations that are causing the resistance, and not just the genes that are differentially expressed.

Dissemination of Results

Finally, a word on the future dissemination of this work. The results and methodologies developed in this thesis, and more accurately throughout the entire project, are currently being prepared for publication in two scientific journal articles. The first manuscript will focus on the pyHaiCS software, highlighting its *open-source* contributions to computational statistics and Bayesian inference with Hamiltonian-inspired sampling methods. The second manuscript will detail our work on tamoxifen resistance and the identification of potential genetic biomarkers. This publication will also incorporate additional research that extends beyond the scope of this thesis, including a novel *gene filtering* strategy and an optimization method designed to extract robust and meaningful *signatures* from high-dimensional genomic data.

Bibliography

[1] Lewis Carroll. *Alice's Adventures in Wonderland*. Macmillan & Co., 1865.

[2] Hyuna Sung, Jacques Ferlay, Rebecca L Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3):209–249, 2021.

[3] Tajda Tavčar Kunstič, Nataša Debeljak, and Klementina Fon Tacer. Heterogeneity in hormone-dependent breast cancer and therapy: Steroid hormones, her2, melanoma antigens, and cannabinoid receptors. *Advances in Cancer Biology-Metastasis*, 7:100086, 2023.

[4] Franco Lumachi, A Brunello, Marco Maruzzo, U Basso, and S Mm Basso. Treatment of estrogen receptor-positive breast cancer. *Current medicinal chemistry*, 20(5):596–604, 2013.

[5] Wenlin Shao and Myles Brown. Advances in estrogen receptor biology: prospects for improvements in targeted breast cancer therapy. *Breast Cancer Research*, 6:1–14, 2003.

[6] Jack-Michel Renoir, Véronique Marsaud, and Gwendal Lazennec. Estrogen receptor signaling as a target for novel breast cancer therapeutics. *Biochemical pharmacology*, 85(4):449–465, 2013.

[7] Eric A Ariazi, Jennifer L Ariazi, Fernando Cordera, and V Craig Jordan. Estrogen receptors as therapeutic targets in breast cancer. *Current topics in medicinal chemistry*, 6(3):181–202, 2006.

[8] Early Breast Cancer Trialists' Collaborative Group et al. Relevance of breast cancer hormone receptors and other factors to the efficacy of adjuvant tamoxifen: patient-level meta-analysis of randomised trials. *The lancet*, 378(9793):771–784, 2011.

[9] Therese Sørlie, Charles M Perou, Robert Tibshirani, Turid Aas, Stephanie Geisler, Hilde Johnsen, Trevor Hastie, Michael B Eisen, Matt Van De Rijn, Stefanie S Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19):10869–10874, 2001.

[10] Keith T Schmidt, Cindy H Chau, Douglas K Price, and William D Figg. Precision oncology medicine: the clinical relevance of patient-specific biomarkers used to optimize cancer treatment. *The Journal of Clinical Pharmacology*, 56(12):1484–1499, 2016.

[11] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[12] Roy Frostig, Matthew James Johnson, and Chris Leary. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 4(9), 2018.

[13] Henri Poincaré. Hypotheses in physics. *Scientific Work and Creativity: Advice from the Masters*, 1:163, 2012.

[14] Padmini Bisoyi. A brief tour guide to cancer disease. In *Understanding cancer*, pages 1–20. Elsevier, 2022.

[15] Joel S Brown, Sarah R Amend, Robert H Austin, Robert A Gatenby, Emma U Hammarlund, and Kenneth J Pienta. Updating the definition of cancer. *Molecular Cancer Research*, 21(11):1142–1147, 2023.

[16] Shende Pravin and Agrawal Sudhir. Integration of 3d printing with dosage forms: A new perspective for modern healthcare. *Biomedicine & Pharmacotherapy*, 107:146–154, 2018.

[17] Madihalli Somashekharaiah Chandraprasad, Abhijit Dey, and Mallappa Kumara Swamy. Introduction to cancer and treatment approaches. In *Paclitaxel*, pages 1–27. Elsevier, 2022.

[18] P Jean-Pierre and BC McDonald. Neuroepidemiology of cancer and treatment-related neurocognitive dysfunction in adult-onset cancer patients and survivors. *Handbook of clinical neurology*, 138:297–309, 2016.

[19] Maria Gabriela Cardoso Teles Monteiro and Guilherme Pertinni de Morais Gouveia. Physiotherapy in the management of gynecological cancer patient: A systematic review. *Journal of bodywork and movement therapies*, 28:354–361, 2021.

[20] Liora Colobatiu, Laura Gavrilas, and Andrei Mocan. Natural compounds as chemosensitizers: A lesson from plants. In *pH-Interfering Agents as Chemosensitizers in Cancer Therapy*, pages 147–165. Elsevier, 2021.

[21] Amirhossein Bahreyni, Yasir Mohamud, and Honglin Luo. Oncolytic virus-based combination therapy in breast cancer. *Cancer Letters*, page 216634, 2024.

[22] BC John Cho and David R McCready. Oncologic principles in breast reconstruction. *Clinics in Plastic Surgery*, 34(1):1–13, 2007.

[23] Lars-Arne Haldosén, Chunyan Zhao, and Karin Dahlman-Wright. Estrogen receptor beta in breast cancer. *Molecular and cellular endocrinology*, 382(1):665–672, 2014.

[24] José Adélaïde, Pascal Finetti, Ismahane Bekhouche, Laetitia Repellini, Jeannine Geneix, Fabrice Sircoulomb, Emmanuelle Charafe-Jauffret, Nathalie Cervera, Jérôme Desplans, Daniel Parzy, et al. Integrated profiling of basal and luminal breast cancers. *Cancer research*, 67(24):11565–11575, 2007.

[25] Jaafar Makki. Diversity of breast carcinoma: histological subtypes and clinical relevance. *Clinical medicine insights: Pathology*, 8:CPath–S31563, 2015.

[26] Emma Nolan, Geoffrey J Lindeman, and Jane E Visvader. Deciphering breast cancer: from biology to the clinic. *Cell*, 186(8):1708–1728, 2023.

[27] Franco Lumachi, Davide A Santeufemia, and Stefano MM Basso. Current medical treatment of estrogen receptor-positive breast cancer. *World journal of biological chemistry*, 6(3):231, 2015.

[28] Adrienne G Waks and Eric P Winer. Breast cancer treatment: a review. *Jama*, 321(3):288–300, 2019.

[29] V Craig Jordan and Angela MH Brodie. Development and evolution of therapies targeted to the estrogen receptor for the treatment and prevention of breast cancer. *Steroids*, 72(1):7–25, 2007.

[30] Renan Gomes do Nascimento and Kaléu Mormino Otoni. Histological and molecular classification of breast cancer: what do we know? *Mastology*, 30:1–8, 2020.

[31] Pelin Yaşar, Gamze Ayaz, Sırma Damla User, Gizem Güpür, and Mesut Muyan. Molecular mechanism of estrogen–estrogen receptor signaling. *Reproductive medicine and biology*, 16(1):4–20, 2017.

[32] Rinath Jeselsohn, Gilles Buchwalter, Carmine De Angelis, Myles Brown, and Rachel Schiff. Esr1 mutations—a mechanism for acquired endocrine resistance in breast cancer. *Nature reviews Clinical oncology*, 12(10):573–583, 2015.

[33] Marco Piva, Giacomo Domenici, Oihana Iriondo, Miriam Rábano, Bruno M Simões, Valentine Comaills, Inmaculada Barredo, Jose A López-Ruiz, Ignacio Zabalza, Robert Kypta, et al. Sox2 promotes tamoxifen resistance in breast cancer cells. *EMBO molecular medicine*, 6(1):66–79, 2014.

[34] C Kent Osborne, Jiang Shou, Suleiman Massarweh, and Rachel Schiff. Crosstalk between estrogen receptor and growth factor receptor pathways as a cause for endocrine therapy resistance in breast cancer. *Clinical cancer research*, 11(2):865s–870s, 2005.

[35] Soonmyung Paik, Steven Shak, Gong Tang, Chungyeul Kim, Joffre Baker, Maureen Cronin, Frederick L Baehner, Michael G Walker, Drew Watson, Taesung Park, et al. A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer. *New England Journal of Medicine*, 351(27):2817–2826, 2004.

[36] Laura J Van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin Van Der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415(6871):530–536, 2002.

[37] Martin Parga-Pazos, Nicole Cusimano, Miriam Rábano, Elena Akhmatskaya, et al. A novel mathematical approach for analysis of integrated cell–patient data uncovers a 6-gene signature linked to endocrine therapy resistance. *Laboratory Investigation*, 104(1):100286, 2024.

[38] TN Schuurman, PO Witteveen, E Van der Wall, JLM Passier, ADR Huitema, F Amant, and CAR Lok. Tamoxifen and pregnancy: an absolute contraindication? *Breast Cancer Research and Treatment*, 175:17–25, 2019.

[39] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[40] Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998.

[41] Radford M Neal. Mcmc using hamiltonian dynamics. *arXiv preprint arXiv:1206.1901*, 2012.

[42] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[43] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[44] Don Van Ravenzwaaij, Pete Cassey, and Scott D Brown. A simple introduction to markov chain monte–carlo sampling. *Psychonomic bulletin & review*, 25(1):143–154, 2018.

[45] Marco Taboga. Markov chain monte carlo (mcmc) diagnostics. *Lectures on probability theory and mathematical statistics*, 2017.

[46] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[47] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[48] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[49] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[50] Tijana Radivojevic et al. Enhancing sampling in computational statistics using modified hamiltonians. *PhD Thesis*, 2016.

[51] Tijana Radivojević and Elena Akhmatskaya. Modified hamiltonian monte carlo for bayesian inference. *Statistics and Computing*, 30(2):377–404, 2020.

[52] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

[53] Nawaf Bou-Rabee and Jesús Maria Sanz-Serna. Geometric integrators and the hamiltonian monte carlo method. *Acta Numerica*, 27:113–206, 2018.

[54] David Morin. *Introduction to classical mechanics: with problems and solutions*. Cambridge University Press, 2008.

[55] Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the hybrid monte carlo algorithm. 2013.

[56] Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.

[57] Elena Akhmatskaya, Mario Fernández-Pendás, Tijana Radivojević, and JM Sanz-Serna. Adaptive splitting integrators for enhancing sampling efficiency of modified hamiltonian monte carlo methods in molecular simulation. *Langmuir*, 33(42):11530–11542, 2017.

[58] Robert D Skeel, Guihua Zhang, and Tamar Schlick. A family of symplectic integrators: stability, accuracy, and molecular dynamics applications. *SIAM Journal on Scientific Computing*, 18(1):203–222, 1997.

[59] Alexandros Beskos, Frank J Pinski, Jesús Marıa Sanz-Serna, and Andrew M Stuart. Hybrid monte carlo on hilbert spaces. *Stochastic Processes and their Applications*, 121(10):2201–2230, 2011.

[60] Babak Shahbaba, Shiwei Lan, Wesley O Johnson, and Radford M Neal. Split hamiltonian monte carlo. *Statistics and Computing*, 24:339–349, 2014.

[61] Wei-Lun Chao, Justin Solomon, Dominik Michels, and Fei Sha. Exponential integration for hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1142–1151. PMLR, 2015.

[62] Robert I McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM Journal on Scientific Computing*, 16(1):151–168, 1995.

[63] Sergio Blanes, Fernando Casas, and Jesús Marıa Sanz-Serna. Numerical integrators for the hybrid monte carlo method. *SIAM Journal on Scientific Computing*, 36(4):A1556–A1580, 2014.

[64] AD Kennedy and Brian Pendleton. Cost of the generalised hybrid monte carlo algorithm for free field theory. *Nuclear Physics B*, 607(3):456–510, 2001.

[65] Andrew B Duncan, Tony Lelievre, and Grigorios A Pavliotis. Variance reduction using nonreversible langevin samplers. *Journal of statistical physics*, 163:457–491, 2016.

[66] Herman Kahn and Andy W Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

[67] Robert D Engle, Robert D Skeel, and Matthew Drees. Monitoring energy drift with shadow hamiltonians. *Journal of Computational Physics*, 206(2):432–452, 2005.

[68] Per Christian Moan and Jitse Niesen. On an asymptotic method for computing the modified energy for symplectic methods. *arXiv preprint arXiv:1304.0673*, 2013.

[69] Robert D Skeel and David J Hardy. Practical construction of modified hamiltonians. *SIAM Journal on Scientific Computing*, 23(4):1172–1188, 2001.

[70] Elena Akhmatskaya and Sebastian Reich. The targeted shadowing hybrid monte carlo (tshmc) method. In *New Algorithms for Macromolecular Simulation*, pages 141–153. Springer, 2006.

[71] Elena Akhmatskaya and Sebastian Reich. Gshmc: An efficient method for molecular simulation. *Journal of Computational Physics*, 227(10):4934–4954, 2008.

[72] Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[73] Alain Durmus, Samuel Gruffaz, Miika Kailas, Eero Saksman, and Matti Vihola. On the convergence of dynamic implementations of hamiltonian monte carlo and no u-turn samplers. *arXiv preprint arXiv:2307.03460*, 2023.

[74] Lee Devlin, Paul Horridge, Peter L Green, and Simon Maskell. The no-u-turn sampler as a proposal distribution in a sequential monte carlo sampler with a near-optimal l-kernel. *arXiv preprint arXiv:2108.02498*, 2021.

[75] Jeremy Heng and Pierre E Jacob. Unbiased hamiltonian monte carlo with couplings. *Biometrika*, 106(2):287–302, 2019.

[76] Robert I McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM Journal on Scientific Computing*, 16(1):151–168, 1995.

[77] Mario Fernández-Pendás, Elena Akhmatskaya, and Jesús María Sanz-Serna. Adaptive multi-stage integrators for optimal energy conservation in molecular simulations. *Journal of Computational Physics*, 327:434–449, 2016.

[78] Tijana Radivojević, Mario Fernández-Pendás, Jesús María Sanz-Serna, and Elena Akhmatskaya. Multistage splitting integrators for sampling with modified hamiltonian monte carlo methods. *Journal of Computational Physics*, 373:900–916, 2018.

[79] Lorenzo Nagar, Mario Fernandez-Pendas, Jesus Maria Sanz-Serna, and Elena Akhmatskaya. Adaptive multi-stage integration schemes for hamiltonian monte carlo. *Journal of Computational Physics*, 502:112800, 2024.

[80] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Burkner. Rank-normalization, folding, and localization: An improved r for assessing convergence of mcmc (with discussion). *Bayesian analysis*, 16(2):667–718, 2021.

[81] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992.

[82] Jing Teng, Honglei Zhang, Wuyi Liu, Xiao-Ou Shu, and Fei Ye. A dynamic bayesian model for breast cancer survival prediction. *IEEE Journal of Biomedical and Health Informatics*, 26(11):5716–5727, 2022.

[83] Xuelin Huang, Yisheng Li, Juhee Song, and Donald A Berry. A bayesian simulation model for breast cancer screening, incidence, treatment, and mortality. *Medical Decision Making*, 38(1_suppl):78S–88S, 2018.

[84] Fan Su, Jianqian Chao, Pei Liu, Bowen Zhang, Na Zhang, Zongyu Luo, and Jiaying Han. Prognostic models for breast cancer: based on logistics regression and hybrid bayesian network. *BMC Medical Informatics and Decision Making*, 23(1):120, 2023.

[85] Jiadong Chu, NA Sun, Wei Hu, Xuanli Chen, Nengjun Yi, and Yueping Shen. The application of bayesian methods in cancer prognosis and prediction. *Cancer Genomics & Proteomics*, 19(1):1–11, 2022.

[86] Brian J Smith and James J Mezhir. An interactive bayesian model for prediction of lymph node ratio and survival in pancreatic cancer patients. *Journal of the American Medical Informatics Association*, 21(e2):e203–e211, 2014.

[87] Fuzhang Wang, M Idrees, and Ayesha Sohail. "ai-mcmc" for the parametric analysis of the hormonal therapy of cancer. *Chaos, Solitons & Fractals*, 154:111618, 2022.

[88] Peng He, Jing Li, Minyan Chen, Meng Huang, Yibin Qiu, Qindong Cai, Yuxiang Lin, Chuan Wang, and Fangmeng Fu. Comparative efficacy and safety of extended adjuvant endocrine therapy for hormone receptor-positive early breast cancer: a bayesian network meta-analysis. *Breast Cancer Research and Treatment*, 203(1):13–28, 2024.

[89] Yi-Cheng Jiang, Jing-Jing Yang, Hai-Tian Zhang, Rui Zhuo, Sebastian De La Roche, Luz Angela Torres-De La Roche, Rudy Leon De Wilde, and Jie Dong. First-line endocrine therapy for hormone receptor positive and her-2 negative metastatic breast cancer: A bayesian network meta-analysis. *Oncology Letters*, 28(5):513, 2024.

[90] Iskander Aurrekoetxea-Rodriguez, So Young Lee, Miriam Rábano, Isabel Gris-Cárdenas, Virginia Gamboa-Aldecoa, Irantzu Gorroño, Isabella Ramella-Gal, Connor Parry, Robert M Kypta, Beñat Artetxe, et al. Polyoxometalate inhibition of sox2-mediated tamoxifen resistance in breast cancer. *Cell Communication and Signaling*, 22(1):425, 2024.

[91] Giacomo Domenici, Iskander Aurrekoetxea-Rodríguez, Bruno M Simões, Miriam Rábano, So Young Lee, Julia San Millán, Valentine Comaills, Erik Oliemuller, José A López-Ruiz, Ignacio Zabalza, et al. A sox2–sox9 signalling axis maintains human breast luminal progenitor and breast cancer stem cells. *Oncogene*, 38(17):3151–3169, 2019.

[92] Aida Sarmiento-Castro, Eva Caamaño-Gutiérrez, Andrew H Sims, Nathan J Hull, Mark I James, Angélica Santiago-Gómez, Rachel Eyre, Christopher Clark, Martha E Brown, Michael D Brooks, et al. Increased expression of interleukin-1 receptor characterizes anti-estrogen-resistant aldh+ breast cancer stem cells. *Stem Cell Reports*, 15(2):307–316, 2020.

[93] Luca Magnani, Alexander Stoeck, Xiaoyang Zhang, András Lánczky, Anne C Mirabella, Tian-Li Wang, Balázs Gyorffy, and Mathieu Lupien. Genome-wide reprogramming of the chromatin landscape underlies endocrine therapy resistance in breast cancer. *Proceedings of the National Academy of Sciences*, 110(16):E1490–E1499, 2013.

[94] Bruno M Simoes, Ciara S O'Brien, Rachel Eyre, Andreia Silva, Ling Yu, Aida Sarmiento-Castro, Denis G Alférez, Kath Spence, Angelica Santiago-Gomez, Francesca Chemi, et al. Anti-estrogen resistance in human breast tumors is driven by jag1-notch4-dependent cancer stem cell activity. *Cell reports*, 12(12):1968–1977, 2015.

[95] Tian Gao, Yong Han, Ling Yu, Sheng Ao, Ziyu Li, and Jiafu Ji. Ccna2 is a prognostic biomarker for er+ breast cancer and tamoxifen resistance. *PloS one*, 9(3):e91771, 2014.

[96] Lei Huang, Shuangping Zhao, Jonna M Frasor, and Yang Dai. An integrated bioinformatics approach identifies elevated cyclin e2 expression and e2f activity as distinct features of tamoxifen resistant breast tumors. *PloS one*, 6(7):e22274, 2011.

[97] Marta Palafox, Laia Monserrat, Meritxell Bellet, Guillermo Villacampa, Abel Gonzalez-Perez, Mafalda Oliveira, Fara Brasó-Maristany, Nusaibah Ibrahimi, Srinivasaraghavan Kannan, Leonardo Mina, et al. High p16 expression and heterozygous rb1 loss are biomarkers for cdk4/6 inhibitor resistance in er+ breast cancer. *Nature communications*, 13(1):5258, 2022.

[98] Alison Harrod, Chun-Fui Lai, Isabella Goldsbrough, Georgia M Simmons, Natasha Oppermans, Daniela B Santos, Balazs Győrffy, Rebecca C Allsopp, Bradley J Toghill, Kirsty Balachandran, et al. Genome engineering for estrogen receptor mutations reveals differential responses to anti-estrogens and new prognostic gene signatures for breast cancer. *Oncogene*, 41(44):4905–4915, 2022.

[99] Youli Xia, Xiaping He, Lorna Renshaw, Carlos Martinez-Perez, Charlene Kay, Mark Gray, James Meehan, Joel S Parker, Charles M Perou, Lisa A Carey, et al. Integrated dna and rna sequencing reveals drivers of endocrine resistance in estrogen receptor–positive breast cancer. *Clinical Cancer Research*, 28(16):3618–3629, 2022.

[100] Philip C Miller, Jennifer Clarke, Tulay Koru-Sengul, Joeli Brinkman, and Dorraya El-Ashry. A novel mapk–microrna signature is predictive of hormone-therapy resistance and poor outcome in er-positive breast cancer. *Clinical cancer research*, 21(2):373–385, 2015.

[101] Xiyu Kang, Jiaxiang Liu, and Xin Wang. Abstract po5-21-05: An endocrine resistant-related gene signature revealing the tumor microenvironment to predict the prognosis of hormone receptor-positive breast cancer patients. *Cancer Research*, 84(9_Supplement):PO5–21, 2024.

[102] Ming-Liang Jin, Xi Jin, and Zhi-Ming Shao. Novel gene signatures predictive of patient recurrence-free survival in hr+ her2-breast cancer. 2024.

[103] Adam Hermawan, Herwandhani Putri, and Muthi Ikawati. Bioinformatic analysis reveals the molecular targets of tangeretin in overcoming the resistance of breast cancer to tamoxifen. *Gene Reports*, 21:100884, 2020.

[104] Zsuzsanna Mihály, Máté Kormos, András Lánczky, Magdolna Dank, Jan Budczies, Marcell A Szász, and Balázs Győrffy. A meta-analysis of gene expression-based biomarkers predicting outcome after tamoxifen treatment in breast cancer. *Breast cancer research and treatment*, 140:219–232, 2013.

[105] Xiaopeng Wang and Shixia Wang. Identification of key genes involved in tamoxifen-resistant breast cancer using bioinformatics analysis. *Translational Cancer Research*, 10(12):5246, 2021.

[106] Sarra M Rahem, Nusrat J Epsi, Frederick D Coffman, and Antonina Mitrofanova. Genome-wide analysis of therapeutic response uncovers molecular pathways governing tamoxifen resistance in er+ breast cancer. *EBioMedicine*, 61, 2020.

[107] Kalifa Manjang, Shailesh Tripathi, Olli Yli-Harja, Matthias Dehmer, Galina Glazko, and Frank Emmert-Streib. Prognostic gene expression signatures of breast cancer are lacking a sensible biological meaning. *Scientific Reports*, 11(1):156, 2021.

[108] David Venet, Jacques E Dumont, and Vincent Detours. Most random gene expression signatures are significantly associated with breast cancer outcome. *PLoS computational biology*, 7(10):e1002240, 2011.

[109] Wilson Wen Bin Goh and Limsoon Wong. Why breast cancer signatures are no better than random signatures explained. *Drug Discovery Today*, 23(11):1818–1823, 2018.

[110] Dennis Victor Lindley. *Bayesian statistics: A review*. SIAM, 1972.

[111] Herbert D Soule, J Vazquez, A Long, S Albert, and MJJOTNCI Brennan. A human cell line from a pleural effusion derived from a breast carcinoma. *Journal of the national cancer institute*, 51(5):1409–1416, 1973.

[112] Brigham & Women's Hospital & Harvard Medical School Chin Lynda 9 11 Park Peter J. 12 Kucherlapati Raju 13, Genome data analysis: Baylor College of Medicine Creighton Chad J. 22 23 Donehower Lawrence A. 22 23 24 25, Institute for Systems Biology Reynolds Sheila 31 Kreisberg Richard B. 31 Bernard Brady 31 Bressler Ryan 31 Erkkila Timo 32 Lin Jake 31 Thorsson Vesteinn 31 Zhang Wei 33 Shmulevich Ilya 31, et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012.

[113] M Parga et al. Tackling the development of hormone therapy resistance in breast cancer through mathematical modelling. *PhD Thesis*, 2024.

[114] Radmila Hrdlickova, Masoud Toloue, and Bin Tian. Rna-seq methods for transcriptome analysis. *Wiley Interdisciplinary Reviews: RNA*, 8(1):e1364, 2017.

[115] Vahid Jalili, Enis Afgan, Qiang Gu, Dave Clements, Daniel Blankenberg, Jeremy Goecks, James Taylor, and Anton Nekrutenko. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update. *Nucleic acids research*, 48(W1):W395–W402, 2020.

[116] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17:1–19, 2016.

[117] Juliana Costa-Silva, Douglas Domingues, and Fabricio Martins Lopes. Rna-seq differential expression analysis: An extended review and a software tool. *PloS one*, 12(12):e0190152, 2017.

[118] Charlotte Soneson and Mauro Delorenzi. A comparison of methods for differential expression analysis of rna-seq data. *BMC bioinformatics*, 14:1–18, 2013.

[119] Shiyi Liu, Zitao Wang, Ronghui Zhu, Feiyan Wang, Yanxiang Cheng, and Yeqiang Liu. Three differential expression analysis methods for rna sequencing: limma, edger, deseq2. *JoVE (Journal of Visualized Experiments)*, (175):e62528, 2021.

[120] Franck Rapaport, Raya Khanin, Yupu Liang, Mono Pirun, Azra Krek, Paul Zumbo, Christopher E Mason, Nicholas D Socci, and Doron Betel. Comprehensive evaluation of differential gene expression analysis methods for rna-seq data. *Genome biology*, 14:1–13, 2013.

[121] Clarissa M Koch, Stephen F Chiu, Mahzad Akbarpour, Ankit Bharat, Karen M Ridge, Elizabeth T Bartom, and Deborah R Winter. A beginner's guide to analysis of rna sequencing data. *American journal of respiratory cell and molecular biology*, 59(2):145–157, 2018.

[122] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[123] Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. Htseq—a python framework to work with high-throughput sequencing data. *bioinformatics*, 31(2):166–169, 2015.

[124] Yang Liao, Gordon K Smyth, and Wei Shi. featurecounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30(7):923–930, 2014.

[125] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

[126] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[127] Jure Zmrzlikar, Matjaž Žganec, Luka Ausec, and Miha Štajdohar. Normalizing rna-seq data in python with rnanorm. 2023.

[128] Günter P Wagner, Koryu Kin, and Vincent J Lynch. Measurement of mrna abundance using rna-seq data: Rpkm measure is inconsistent among samples. *Theory in biosciences*, 131:281–285, 2012.

[129] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008.

[130] Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome biology*, 11:1–9, 2010.

[131] Kayla A Johnson and Arjun Krishnan. Robust normalization and transformation techniques for constructing gene coexpression networks from rna-seq data. *Genome biology*, 23:1–26, 2022.

[132] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Nature Precedings*, pages 1–1, 2010.

[133] Adrian Colin Cameron and Pravin K Trivedi. *Regression analysis of count data*. Number 53. Cambridge university press, 2013.

[134] Erika Cule, Paolo Vineis, and Maria De Iorio. Significance testing in ridge regression for genetic data. *BMC bioinformatics*, 12:1–15, 2011.

[135] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[136] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[137] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[138] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[139] Oriol Abril-Pla, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J Fonnesbeck, Maxim Kochurov, Ravin Kumar, Junpeng Lao, Christian C Luhmann, Osvaldo A Martin, et al. Pymc: a modern, and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, 9:e1516, 2023.

[140] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011.

[141] LR Haff. An identity for the wishart distribution with applications. *Journal of Multivariate Analysis*, 9(4):531–544, 1979.

[142] Hristo Inouzhe, Maria Xose Rodriguez-Alvarez, Lorenzo Nagar, and Elena Akhmatskaya. Dynamic sir/seir-like models comprising a time-dependent transmission rate: Hamiltonian monte carlo approach with applications to covid-19. *arXiv preprint arXiv:2301.06385*, 2023.

[143] Gabriel Ybarra Marcaida. Deducing the talbot effect from electrodynamics. *arXiv preprint arXiv:2507.00820*, 2025.

[144] Andrew Gelman, Aleks Jakulin, Maria Grazia Pittau, and Yu-Sung Su. A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics*, pages 1360–1383, 2008.

[145] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[146] Karl Thurnhofer-Hemsi, Ezequiel López-Rubio, Miguel A Molina-Cabello, and Kayvan Najarian. Radial basis function kernel optimization for support vector machine classifiers. *arXiv preprint arXiv:2007.08233*, 2020.

[147] Harry Zhang. The optimality of naive bayes. *Aa*, 1(2):3, 2004.

[148] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[149] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[150] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[151] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[152] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

[153] Christopher M Bishop. Bayesian neural networks. *Journal of the Brazilian Computer Society*, 4:61–68, 1997.

[154] Radford M Neal. Bayesian training of backpropagation networks by the hybrid monte carlo method. Technical report, Citeseer, 1992.

[155] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[156] Nicolas Brosse, Alain Durmus, and Eric Moulines. The promises and pitfalls of stochastic gradient langevin dynamics. *Advances in Neural Information Processing Systems*, 31, 2018.

[157] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

[158] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

[159] Ranganath Krishnan, Pi Esposito, and Mahesh Subedar. Bayesian-torch: Bayesian neural network layers for uncertainty estimation, January 2022.

[160] Somayajulu LN Dhulipala, Yifeng Che, and Michael D Shields. Efficient bayesian inference with latent hamiltonian neural networks in no-u-turn sampling. *Journal of Computational Physics*, 492:112425, 2023.

[161] Pengzhan Jin, Zhen Zhang, Aiqing Zhu, Yifa Tang, and George Em Karniadakis. Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

[162] Yunjin Tong, Shiying Xiong, Xingzhe He, Guanghan Pan, and Bo Zhu. Symplectic neural networks in taylor series form for hamiltonian systems. *Journal of Computational Physics*, 437:110325, 2021.

[163] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[164] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[165] Giuseppe Jurman, Samantha Riccadonna, and Cesare Furlanello. A comparison of mcc and cen error measures in multi-class prediction. 2012.

[166] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[167] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[168] Carl Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, 13, 2000.

[169] Botond B Antal, Anthony G Chesebro, Helmut H Strey, Lilianne R Mujica-Parodi, and Corey Weistuch. Achieving occam's razor: Deep learning for optimal model reduction. *PLOS Computational Biology*, 20(7):e1012283, 2024.

[170] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[171] Laurys Boudin, Alexandre De Nonneville, Pascal Finetti, Geoffrey Guittard, Jacques A Nunes, Daniel Birnbaum, Emilie Mamessier, and François Bertucci. Cish expression is associated with metastasis-free interval in triple-negative breast cancer and refines the prognostic value of pdl1 expression. *Cancers*, 14(14):3356, 2022.

[172] Gui-E Lai, Jian Zhou, Cui-Liu Huang, Cun-Jun Mai, Yi-Mei Lai, Zhi-Qin Lin, Tao Peng, Yuan Luo, and Feng-En Liu. A combination of transcriptome and methylation analyses reveals the role of lncrna hotairm1 in the proliferation and metastasis of breast cancer. *Gland Surgery*, 11(5):826, 2022.

[173] Ingvild S Fenne, Thomas Helland, Marianne H Flågeng, Simon N Dankel, Gunnar Mellgren, and Jørn V Sagen. Downregulation of steroid receptor coactivator-2 modulates estrogen-responsive genes and stimulates proliferation of mcf-7 breast cancer cells. *PloS one*, 8(7):e70096, 2013.

[174] Michael Becker, Anette Sommer, Jörn R Krätzschmar, Henrik Seidel, Hans-Dieter Pohlenz, and Iduna Fichtner. Distinct gene expression patterns in a tamoxifen-sensitive human mammary carcinoma xenograft and its tamoxifen-resistant subline maca 3366/tam. *Molecular cancer therapeutics*, 4(1):151–170, 2005.

[175] Edward Y Chen, Christopher M Tan, Yan Kou, Qiaonan Duan, Zichen Wang, Gabriela Vaz Meirelles, Neil R Clark, and Avi Ma'ayan. Enrichr: interactive and collaborative html5 gene list enrichment analysis tool. *BMC bioinformatics*, 14:1–14, 2013.

[176] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic acids research*, 44(W1):W90–W97, 2016.

[177] Zhuorui Xie, Allison Bailey, Maxim V Kuleshov, Daniel JB Clarke, John E Evangelista, Sherry L Jenkins, Alexander Lachmann, Megan L Wojciechowicz, Eryk Kropiwnicki, Kathleen M Jagodnik, et al. Gene set knowledge discovery with enrichr. *Current protocols*, 1(3):e90, 2021.

[178] Qiuwen Tan, Li Xu, Junhui Zhang, Liangju Ning, Yanling Jiang, Tao He, Jingcong Luo, Jie Chen, Qing Lv, Xiaoqin Yang, et al. Breast cancer cells interact with tumor-derived extracellular matrix in a molecular subtype-specific manner. *Biomaterials Advances*, 146:213301, 2023.

[179] Ming-Hsin Yeh, Yau-Jin Tzeng, Ting-Ying Fu, Jun-Jie You, Hong-Tai Chang, Luo-Ping Ger, and Kuo-Wang Tsai. Extracellular matrix–receptor interaction signaling genes associated with inferior breast cancer survival. *Anticancer research*, 38(8):4593–4605, 2018.

[180] Yulong Bao, Li Wang, Lin Shi, Fen Yun, Xia Liu, Yongxia Chen, Chen Chen, Yanni Ren, and Yongfeng Jia. Transcriptome profiling revealed multiple genes and ecm-receptor interaction pathways that may be associated with breast cancer. *Cellular & molecular biology letters*, 24:1–20, 2019.

[181] Maximilian Boesch, Lucas Onder, Hung-Wei Cheng, Mario Novkovic, Urs Mörbe, Sieghart Sopper, Guenther Gastl, Wolfram Jochum, Thomas Ruhstaller, Michael Knauer, et al. Interleukin 7-expressing fibroblasts promote breast cancer growth through sustenance of tumor cell stemness. *Oncoimmunology*, 7(4):e1414129, 2018.

[182] MAA Al-Rawi, K Rmali, G Watkins, RE Mansel, and WG Jiang. Aberrant expression of interleukin-7 (il-7) and its signalling complex in human breast cancer. *European Journal of Cancer*, 40(4):494–502, 2004.

[183] Faton Sermaxhaj, Natalija Dedić Plavetić, Ugur Gozalan, Ana Kulić, Ljubica Radmilović Varga, Marina Popović, Slavica Sović, Davor Mijatović, Besim Sermaxhaj, and Mentor Sopjani. The role of interleukin-7 serum level as biological marker in breast cancer: a cross-sectional, observational, and analytical study. *World journal of surgical oncology*, 20(1):225, 2022.

[184] MAA Al-Rawi, K Rmali, RE Mansel, and WG Jiang. Interleukin 7 induces the growth of breast cancer cells through a wortmannin-sensitive pathway. *Journal of British Surgery*, 91(1):61–68, 2004.

[185] J.R.R. Tolkien. *The Fellowship of the Ring (The Lord of the Rings, Book 1)*. The Lord of the Rings. Allen & Unwin, 1954.

[186] David Douglas, Greg Papadopoulos, and John Boutelle. *Citizen engineer: A handbook for socially responsible engineering*. Pearson Education, 2009.

[187] Grace Keegan, Angelena Crown, Charles DiMaggio, and Kathie-Ann Joseph. Insufficient reporting of race and ethnicity in breast cancer clinical trials. *Annals of Surgical Oncology*, 30(12):7008–7014, 2023.

[188] Wilma Lingle, Bradley J Erickson, Margarita L Zuley, Rose Jarosz, Ermelinda Bonaccio, Joe Filippini, Jose M Net, Len Levi, Elizabeth A Morris, Gloria G Figler, et al. The cancer genome atlas breast invasive carcinoma collection (tcga-brca). *(No Title)*, 2016.

[189] Ayca Gucalp, Tiffany A Traina, Joel R Eisner, Joel S Parker, Sara R Selitsky, Ben H Park, Anthony D Elias, Edwina S Baskin-Bey, and Fatima Cardoso. Male breast cancer: a disease distinct from female breast cancer. *Breast cancer research and treatment*, 173:37–48, 2019.

[190] Kelly A Hirko, Gabrielle Rocque, Erica Reasor, Ammanuel Taye, Alex Daly, Ramsey I Cutress, Ellen R Copson, Dae-Won Lee, Kyung-Hun Lee, Seock-Ah Im, et al. The impact of race and ethnicity in breast cancer—disparities and implications for precision oncology. *BMC medicine*, 20(1):72, 2022.

[191] Gertraud Maskarinec, Cherisse Sen, Karin Koga, and Shannon M Conroy. Ethnic differences in breast cancer survival: status and determinants. *Women's health*, 7(6):677–687, 2011.

[192] Michael Curtiz, William Keighley, Errol Flynn, Olivia De Havilland, Basil Rathbone, Claude Rains, and Rudy Behlmer. The adventures of robin hood, 1938.

[193] Lao H Saal, Johan Vallon-Christersson, Jari Häkkinen, Cecilia Hegardt, Dorthe Grabau, Christof Winter, Christian Brueffer, Man-Hung Eric Tang, Christel Reuterswärd, Ralph Schulz, et al. The sweden cancerome analysis network-breast (scan-b) initiative: a large-scale multicenter infrastructure towards implementation of breast cancer genomic analyses in the clinical routine. *Genome medicine*, 7:1–12, 2015.

[194] Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

[195] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

[196] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

[197] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[198] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[199] Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. *bioRxiv*, pages 2025–02, 2025.

[200] Minoru Kanehisa, Miho Furumichi, Yoko Sato, Masayuki Kawashima, and Mari Ishiguro-Watanabe. Kegg for taxonomy-based analysis of pathways and genomes. *Nucleic acids research*, 51(D1):D587–D592, 2023.

[201] Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, et al. The reactome pathway knowledgebase 2022. *Nucleic acids research*, 50(D1):D687–D692, 2022.

[202] Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, et al. Genome modeling and design across all domains of life with evo 2. *bioRxiv*, pages 2025–02, 2025.

Acronyms

AIA	Adaptive Integration Approach
AR	Acceptance Rate
BCAM	Basque Center for Applied Mathematics
BCE	Binary Cross-Entropy
BLR	Bayesian Logistic Regression
BNN	Bayesian Neural Network
BSM	Bayesian Statistical Modeling
CV	Cross-Validation
DEA	Differential Expression Analysis
DML	Double Machine Learning
e-MAIA	Extended Modified Adaptive Integration Approach
ER	Estrogen Receptor
ESS	Effective Sample Size
FC	Fold-Change
FDR	False Discovery Rate
GD	Gradient Descent
GDPR	General Data Protection Regulation
GHMC	Generalized Hamiltonian Monte Carlo
GLM	Generalized Linear Model
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
GSEA	Gene Set Enrichment Analysis
GSHMC	Generalized Shadow Hamiltonian Monte Carlo
HMC	Hamiltonian Monte Carlo
IACT	Integrated Autocorrelation Time
IS	Importance Sampling
JIT	Just-In-Time
KL	Kullback–Leibler
LHNN	Latent Hamiltonian Neural Network
LRT	Local Reparametrization Trick
MAIA	Modified Adaptive Integration Approach
MC	Monte Carlo
MCC	Matthews Correlation Coefficient

MCMC	Markov-Chain Monte Carlo
MCSE	Monte Carlo Standard Error
MDMC	Molecular Dynamics Monte Carlo
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MLP	Multi-Layer Perceptron
MMHMC	Mix & Match Hamiltonian Monte Carlo
MSE	Mean-Squared Error
MSSI	Multi-Stage Splitting Integrator
NLP	Natural Language Processing
NUTS	No-U-Turn Sampler
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
PMU	Partial Momentum Update
PSRF	Potential Scale Reduction Factor
pyHaiCS	Python in Hamiltonian for Computational Statistics
RBF	Radial Basis Function
RFS	Relapse-Free Survival
RLE	Relative Log Expression
RW-MH	Random-Walk Metropolis-Hastings
s-AIA	Statistical Adaptive Integration Approach
SA	Simulated Annealing
SAE	Sparse Autoencoder
SEIR	Susceptible-Exposed-Infectious-Remove
SERM	Selective Estrogen Receptor Modulator
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Oversampling Technique
SOTA	State-of-the-Art
SVC	Support-Vector Classifier
TCGA-BRCA	The Cancer Genome Atlas Breast Invasive Carcinoma
TP, TN, FP, FN	True Positives, True Negatives, False Positives, False Negatives
TPU	Tensor Processing Unit
VAE	Variational Autoencoder
VI	Variational Inference
VV	Velocity Verlet