Machine Learning at Genome Scale

PhD Thesis

Gabriele Tazza
Supervisor: László Vidács, Dr. and Tibor Gyimóthy, Prof.

Doctoral School of Computer Science
Department of Software Engineering
Faculty of Science and Informatics
University of Szeged

Szeged
2026

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, high-throughput technologies such as Next-Generation Sequencing (NGS) and mass spectrometry have allowed the generation of enormous amounts of omics data quickly and at reduced costs. This changed how we address the study of biological systems. In the past, the approach was almost fully reductionist, with targeted experiments and hypothesis-based methods to investigate specific mechanisms. Now, thanks to the availability of this amount of data and computational resources, we moved to data-driven approaches, which allow us to explore complex systems more in detail. Instead of focusing on one mechanism at a time, we can analyze the data to learn insights on the system, especially in domains like system biology.
In this context, Machine Learning (ML) has become particularly useful when the underlying mechanisms of a system are partially unknown or too complex for classical approaches. However, in many applications, datasets can still be small in terms of the number of samples, and high in number of features, which makes it harder for ML models to generalize well. This creates a classic example of the so-called "curse of dimensionality", a well known challenge that can limit the efficacy of ML in the biological field. Another challenge is to integrate the different layers of omics data. Each one, transcriptomics, proteomics, or epigenomics, represents a different aspect of the biological system and combining them is not always straightforward, as it requires approaches that handle heterogeneity while keeping the biological relevance of each individual data type. For this reason, multi-omics integration is crucial to exploit the information contained in this data, but it still remains an open challenge.
This thesis fits into this context and explores the use of ML, in particular Deep Learning (DL) methods, which can work well with this kind of data, and that address the challenge of integration of multi-omics layers. The common theme of the work is the methodological development of ML approaches at genome-scale.

This thesis is structured as follows: after this introduction, Chapter 2 presents an overview of the background of this work, introducing the European project E-MUSE

and the main scientific topics covered in the thesis, with the aim of providing a common starting point for readers coming from different scientific backgrounds.

In Chapter 3, we presented the first research contribution of this thesis on metagenomics data analysis using a deep learning approach. It explores how data augmentation and 2D embedding improve the performance of classification models in a scenario of data scarcity, using SuperTML, a method inspired by computer vision techniques.

In Chapter 4, we focused on supervised multiple kernel learning (MKL) methods. Although MKL is often underused in the context of multi-omics data integration, we proposed and evaluated different approaches based on classical MKL algorithms and deep learning, showing that kernel-based models can be a valid and competitive alternative for supervised learning tasks in multi-omics settings. In addition, we presented a novel feature importance method for biomarker discovery.

In Chapter 5, we presented a hybrid framework that combines data-driven and mechanistic approaches to predict metabolic fluxes, an important problem in systems biology. The proposed MINN approach reduces the amount of data needed to train a neural network by incorporating prior knowledge from genome-scale metabolic modeling, allows the integration of multi-omics data, and improves the interpretability of the results.

1.1 Contributions

The figures, tables and results included in this thesis was published in scientific papers (listed at the end of the thesis) and the author is responsible for the following contributions:

Journal/Conference	Rank	**chapter 3**	**chapter 4**	**chapter 5**
Foodsim2024 [171]	NA			x
BMC BioData Mining [21]	Q1		x	
IEEE Access [172]	Q1	x		
CSBJ [170]	Q1			x

Table 1.1: *The connection between the thesis chapters and publications.*

Chapter 3.: In this chapter, we showed that SuperTML is an effective approach to deal with small tabular and high-dimensional data, such as metagenomics. Also, we tested several image augmentation techniques, including a custom one implemented by us, in order to regularize the learning process and improve the predictions. We performed

an experimental evaluation showing that SuperTML, used together with image augmentation, can compete with state-of-the-art methods for disease classification. The related literature survey, experimental design, implementation, visualization, software development, and analysis of the results were carried out by the author.

Chapter 4.: In this chapter, we presented different novel supervised multiple kernel learning methods for multi-omics data integration. Specifically, we extended known MKL methods for unsupervised learning to a supervised framework, and we introduced a novel approach based on deep learning called DeepMKL. We tested our methods against different state-of-the-art approaches on four different multi-omics datasets, and we showed that MKL-based approaches are a valid solution to multi-omics data analysis while they are still underused in bioinformatics research. In addition, we introduced a biomarker discovery method based on the DeepMKL architecture. The author is responsible for the literature survey, implementation, visualization, software development, and analysis of the results that regards the deep learning domain.

Chapter 5.: In this chapter, we presented MINN, a hybrid (data-driven/mechanistic) framework that integrates multi-omics data into genome-scale metabolic models to predict metabolic fluxes. We showed that MINN outperforms both pure mechanistic models (pFBA) and purely data-driven approaches (random forests) on a small multi-omics dataset from E. coli single-gene KO grown in minimal glucose medium. The author is responsible for the literature survey, implementation, visualization, software development and analysis of the results that regards the machine learning domain, while the hybrid optimization strategies, the genome-scale metabolic models preparation and the mechanistic modeling part were carried out by the co-authors.

Chapter 2

Background

This chapter provides an overview of the background relevant to this thesis. It begins with a brief description of E-MUSE, the European project within which this thesis is framed, and then summarizes the key concepts explored in the following chapters, namely: omics data, mechanistic models, machine learning, and their applications at genome scale. Given the multidisciplinary nature of the E-MUSE project, which is also reflected in this thesis, the aim of this chapter is to offer a common ground for readers who may come from different scientific backgrounds. The following sections are not intended to be a comprehensive literature review; instead, each chapter will include its own dedicated review of the related work. This chapter serves to give the reader a general understanding of the motivation and main contributions of the thesis.

2.1 E-MUSE project

E-MUSE

This thesis is framed into the European project E-MUSE: *Complex microbial Ecosystems MUltiScale modElling: mechanistic and data-driven approaches integration* (`https://www.itn-emuse.com`). E-MUSE is a Marie Skłodowska-Curie Action Innovative Training Network with the goal of developing new methodologies for modeling complex biological systems. The project seeks to improve our understanding of microbial ecosystems, with a specific focus on fermented food products such as cheese. E-MUSE combines different approaches such as genome-scale metabolic models, dynamic modeling, and machine learning to analyze multi-omics with the aim of identifying links between features in the data and macro scale properties related to cheese ripening and consumer preferences. The research building block of E-MUSE is organized in three work packages (WPs), each focusing on a different aspect of microbial ecosystem modeling.

WP1-Systems modeling focuses on mechanistic modeling by integrating omics data with genome-scale models to study microbial ecosystems and cellular functions. It includes the physiological characterization of microorganisms to refine dynamic models and understand metabolic contributions. Furthermore, WP1 explores model-based control strategies to regulate cellular behavior through environmental modulation.
WP2-Data-based modeling focuses on combining multi-omics integration with statistical, network-based, and machine learning/deep learning approaches to uncover key biological features. Identifying biomarkers improves our understanding of microbial functions and product properties like taste and texture.
WP3-Predictive modeling, combines agent-based and partial differential equation models to study microbial ecosystems. The main focus is cheese production, where experimental data are used to predict microbial behavior, flavor development, and quality under different conditions. The work also extends to consumer-oriented process models, integrating biochemical, microbiological, and sensory properties to improve both traditional and plant-based cheese manufacturing.

This thesis is part of WP2 and, specifically, it represents one of the outcomes of the Early Stage Researcher 8 (ESR8) project: "Machine learning at genome scale". The motivations and objectives of this project will be detailed in the next paraghraph.

ESR8: Machine learning at genome scale

The ESR8 project aims to develop machine learning models to analyze multi-omics data and support mechanistic modeling. Specifically, in the context of WP2 of the E-MUSE project, the main focus is on applications in the fermented food domain, which is characterized by datasets that are both small in sample size and high-dimensional. High dimensionality refers to the high number of features for each sample, such as gene expression levels, metabolite concentrations, and microbial abundances. This is a perfect example of the "curse of dimensionality". The number of samples required to train effectively predictive models grows exponentially with the increase of the dimensionality and this causes overfitting and poor predictive performance. In addition, the integration of different omics layers is a crucial step as it provides a more complete view over the different aspects of a biological mechanism. This represents another challenge due to differences in data types, scales, and noise levels, which makes it more difficult to extract meaningful relationships across different datasets. For all these reasons, applying ML, especially DL, to fermented food data is particularly challenging because these methods require large datasets to generalize well.
The ESR8 project has two main objectives to address these issues. The first one is to develop methods capable of analyzing small, high-dimensional datasets while maintaining robust predictive performance. The second objective is to integrate mechanistic knowledge into machine learning models to improve their predictive capabilities and interpretability. By using genome-scale models and biological knowledge to constrain

the learning process, the search space for machine learning algorithms is reduced. This not only decreases computational complexity but also mitigates the curse of dimensionality by limiting the number of non-biologically relevant patterns the model needs to consider.

As anticipated before, this work is one of the outcomes of the ESR8 project, and its motivations and objectives are framed within the context of this project. The next chapters will introduce and describe in detail all the methods developed during the PhD program, while in the next sections of this chapter the focus will be on the introduction of the concepts related to machine learning, omics data and genome-scale metabolic models, key topics crucial for understanding the context of this thesis, the developed methods, and their applications. The lack of available data within the consortium, forced us to focus mainly on publicly available biomedical datasets or, in general, datasets not strictly related to the food domain. However, the approaches developed remain highly relevant for the E-MUSE and can be readily transferred to food studies.

2.2 Machine learning

In this section, we want to introduce ML in a way that is accessible to any type of reader. According to Arthur Samuel [145], ML is a research field that allows the computers to learn without being explicitly programmed. In fact, in machine learning, the system can automatically learn rules from data without the necessity of explicitly defining each rule, which represents a crucial difference with traditional computer programming. For example, in a scenario where the task is to filter emails based on their size, traditional programming is sufficient because it is straightforward to define explicit conditions on a measurable quantity such as the size of an email. On the other hand, if the goal is to filter emails based on, for example, the probability of being spam or useful, it becomes much more difficult to define clear rules. In such case, machine learning is a more suitable approach, as it can learn patterns from historical data of previous spam and useful emails without relying on any definition.

We used this example because it is useful to introduce supervised learning, a machine learning class where models are trained using labeled datasets, in a way that is both intuitive and formally correct.

Given a dataset:

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{2.1}$$

$x_i \in \mathcal{X}$ are the features of the input and $y_i \in \mathcal{Y}$ are the labels, supervised learning

aims to find a function

$$f : \mathcal{X} \to \mathcal{Y} \quad \text{such that} \quad f(x_i) \approx y_i \tag{2.2}$$

In the filtering emails example, x_i are the features of the emails in the dataset, while y_i are the labels "spam" or "useful". The first ones are vectors where each element corresponds to the frequency of a word in the text of the email. This way of converting email into numerical features is called Bag-of-Words model, and it is one of the simplest ways to encode text data. The second ones are one-hot encoded vectors of two elements, one for each possible label: "spam" or "useful". In this case, this is defined as a "classification" problem because the goal is to assign a label to each email. In contrast, in a "regression" problem, you need to predict a continuous value, for example, the estimated reading time of an email.
Finally, $\mathcal{X}$ and $\mathcal{Y}$ are the sets of all possible configurations for x_i and y_i.

The function f is a family of models with some parameters θ. In supervised learning, we want to approximate the relationship between input features and labels using an optimal set of parameters θ^*. To achieve that, the optimization has to be done in a way that the predictions $f(x_i; \theta)$ are as close as possible to the true labels. This can be achieved by minimizing a loss function $\mathcal{L}$ which measures the difference between predicted values and true ones. We can find the optimal parameters by solving:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \mathcal{L}(y_i, f(x_i; \theta)) \tag{2.3}$$

Once trained, the model can be used to predict the label $\hat{y}$ of a new input x:

$$\hat{y} = f(x; \theta^*) \tag{2.4}$$

We specifically focused on supervised learning for two main reasons: it is easier to use as an illustrative example to introduce ML, and it is the category of all the methods discussed in the next chapters of this thesis. But there are other important categories of machine learning. First one is unsupervised learning, in this case the dataset does not contain the labels, which remain unknown. Here, one goal could be grouping similar emails without knowing which ones are spam or useful; a task called clustering. Another category is reinforcement learning, this type of approach is often used in fields as robotics or game playing where an agent learns how to act in an environment based on the maximization of a reward function.

In the introduction of supervised learning, we introduced the function f, which represents the actual ML model. This function can take different forms depending on the algorithm used. For example, f can be a simple rule-based model like k-Nearest Neighbors (kNN) [169], a probabilistic model such as Naive Bayes [185], or a linear

model like Logistic Regression [14].

In this thesis, we focus on neural networks. And more specifically on deep learning, a term that refers to neural networks composed of many layers and a large number of parameters. Although deep learning is ML, it is often considered as a separate subfield due to its capacity to solve effectively very complex problems, such as image processing, natural language processing and more. The type of neural network that could be used for the email classification example is the Multi-Layer Perceptron (MLP), which consists of layers of neurons connected by weights and followed by non-linear activation functions. To give an illustrative example of a MLP, we show the simple case with one hidden layer and ReLU activation function σ:

$$\hat{y} = \text{softmax}\left(\sigma(XW^h + b^h)W^{out} + b^{out}\right) \tag{2.5}$$

where:

- $X \in \mathbb{R}^{N \times d_{in}}$ is the input data and d_{in} the number of input features
- $W^h \in \mathbb{R}^{d_{in} \times d_h}$ are the weights of the hidden layer and d_h the number of neurons of the hidden layer
- $b^h \in \mathbb{R}^{1 \times d_h}$ are the biases of the hidden layer
- $W^{out} \in \mathbb{R}^{d_h \times d_{out}}$ are the weights of the output layer and d_{out} the number neurons of the output layer
- $b^{out} \in \mathbb{R}^{1 \times d_{out}}$ are the biases of the output layer

The output $\hat{y} \in \mathbb{R}^{N \times d_{out}}$ is the result of the softmax function. An appropriate loss function for classification task, such as email filtering, is the cross-entropy loss. Given the true labels $y \in \mathbb{R}^{N \times d_{out}}$, the loss is computed as the mean cross entropy loss over the data:

$$L_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{d_{out}} y_{ij} \log(\hat{y}_{ij}) \tag{2.6}$$

This loss function is expressed in terms of the neural network weights, which represent the parameters of the model, as in the example before. Also in this case, the parameters are optimized in order to minimize the loss functions. This is done during training through back-propagation and gradient descent.

The one described is a simple version of the more complex architectures used in this work and it serves as a useful starting point for the next chapters. Several other architectures exist for more advanced tasks such as Convolutional Neural Networks

(CNN) [206] for image processing, also used in this thesis (Chapter 3), Transformers [177] for natural language processing, and finally, Recurrent Neural Network (RNN) [198] for time series analysis.

In the following sections, we will discuss omics data and genome-scale metabolic models, focusing also on neural networks applied in this context and their characteristics.

2.3 Omics data

In this section, we want to introduce omics data. We want to show the main categories: genomics(DNA), transcriptomics(RNA), proteomics(proteins), and epigenomics (regulatory modifications to DNA) and the role of ML/DL in their analysis. Recent advances in high-throughput technologies, created a perfect context for collecting huge amounts of datasets and developing new data-driven approaches. The field of study that focuses on multiple omics layers is called multi-omics. This integrative approach provides a more comprehensive view of biological systems and has become essential in fields such as systems biology, personalized medicine, and disease research [139]. Here, we focus on the types of omics used in this work.

Genomics Genomics is the study of the DNA content of an organism, also known as the genome. Data in genomics provides much information on genes, such as structure, function, their variation within the genome, and how they vary between individuals or species. It is worth mentioning that the related field is metagenomics, which involves the study of the collective genomes of different organisms found in a specific sample. Next-generation sequencing is the most common technology to measure genomics data because it allows to sequence the entire genome or targeted regions relatively fast. The output of this process is normally a large set of DNA reads that are further aligned to a reference genome in order to detect differences. These differences are so-called "variants" that include changes such as Single Nucleotide Polymorphisms (SNPs), insertions, deletions, or larger structural changes. Commonly, variants are stored in standard formats, including the Variant Call Format (VCF), and widely used to study mutations or compare genomes between samples. In some studies, such as at the population level or in metagenomics, the data may include abundance profiles of estimations about how frequently a gene or sequence appears in a sample. This will be the case with the data used for our work in Chapter 3.
These data are used in a wide range of applications, starting from the identification of disease-causing mutations, comparing individual or population genomes, and the investigation of evolutionary patterns. In clinical settings, genomics data also play a significant role in precision medicine, where treatments can be tailored according to a patient's genotype.

Transcriptomics Transcriptomics is the study the RNA molecules, also known as the transcriptome. It shows which genes are being expressed and to what degree. mRNA (messenger RNA) are molecules produced during the transcription of the genes. They carry the recipe to produce proteins which are needed to perform a specific function. mRNA expression is the most common type of transcriptomics data, and it is represented as a matrix of gene expression values, where rows represent genes, columns represent samples, and the values are counts or normalized measures. Another important type of gene expression data comes from miRNA (microRNA). miRNAs are small, non-coding RNA molecules that themselves do not code for proteins but control gene expression by binding to target mRNAs, either inhibiting their translation or facilitating their degradation. miRNA expression data are useful to study how gene expression is controlled beyond the mRNA level. Both are used for our analysis in Chapter 4.

In general, transcriptomics data provide rich information on gene activities and their regulation. For this reason, they are widely used to study biological processes such as response to environmental changes and the mechanisms underlying several diseases.

Proteomics Proteomics is the study of all proteins expressed in a cell, tissue, or organism, also known as the proteome. Proteomics focuses on the final functional products of genes: proteins. And it is very important for determining what is going on inside the cell. Proteins are involved in most biological processes, and their expression levels can be linked with the state of the cell, of the environment, or with the presence of a disease. Proteomics data are collected using Mass spectrometry, which is a high-throughput technology to measure thousands of proteins in a sample. Similarly to transcriptomics, proteomics data are represented using a matrix, where rows represent genes, columns represent samples, and the values are expression levels. Proteomics, together with gene expression, are used for the work in Chapter 5.

Proteomics adds an extra layer to complement genomics and transcriptomics and helps to connect the activity of genes to actual cell behavior.

Epigenomics Epigenomics is the study of all the chemical modifications to the DNA that can affect gene expression without modifying the DNA sequence. The most important epigenomics data is DNA methylation, where a methyl group is added to the cytosine bases. This modification can regulate gene expression, and it is linked with cell differentiation and development of diseases. DNA methylation data are collected using bisulfite sequencing. Its output is a matrix of methylation levels at different locations of the genome for different samples. This type of data is used, in the context of a multi-omics analysis, for our work in Chapter 4.
Epigenomics represents a further layer that complements genomics, transcriptomics,

and proteomics, helping to explain how gene activity is controlled in different contexts.

Machine learning and deep learning are widely used both in the single-omics and multi-omics analyses. Their ability to handle high-dimensional and heterogeneous data makes them particularly suitable for prediction tasks, feature selection, and multi-omics integration [8, 98]. In recent years, many methods have been proposed to integrate genomics, transcriptomics, proteomics, and epigenomics data to improve prediction performance and gain deeper biological insights [2]. These approaches are especially useful in the context of precision medicine and biomarker discovery, where combining multiple omics levels can reveal complex regulatory mechanisms that are not visible from a single layer.

A peculiar aspect of omics datasets is their high dimensionality (large number of features) and a small sample size (few number of observations). The high dimensionality depends on the high-throughput technologies which can measures tens of thousand of expression levels of genes or proteins. In addition, combining multiple omics further increases the number of features, complicating the problem. On the other hand, the low sample size depends on the nature of biological experiments, which often require measurements from different individuals, conditions, or time points. In most cases, it is not feasible to perform huge numbers of experiments, making the number of samples much smaller than the number of features. It is worth mentioning that there are exceptions where data availability is not a major issue, such as the Human Cell Atlas project (`https://www.humancellatlas.org`), which provides millions of transcriptomic profiles across diverse human tissues.
Analyzing high dimensional and low sample size data presents a well known challenge called "curse of dimensionality" [48], where the increase of dimension in the data requires an exponential increase in the number of samples needed to train the predictive models. This usually causes overfitting and poor predictive performance.

The chapter 3 of this thesis presents a deep learning-based approach focused on image data augmentation to address the problem of small sample size. The idea is to generate modified additional training samples from the original genomics data in order to improve generalization and reduce overfitting. This is done after converting the tabular data into 2D representations suitable for CNNs, which allow the use of image augmentation techniques [195].

Finally, another important aspect of multi-omics data analysis is the biomarker discovery. Integrating multiple layers of biological information can help to identify more robust and meaningful biomarkers, for example, for a specific condition or disease. In chapter 5, together with different supervised learning methods for multi-omics data, we present an interpretability approach to find new biomarkers.
In the next section, we will introduce genome-scale metabolic models and fluxomics,

focusing on how neural networks can be applied in this field.

2.4 Genome-scale metabolic models and constraint-based modeling

This section introduces genome-scale metabolic models and constraint-based modeling approaches. The goal is to describe what these methods are, their strengths and limitations, and how they can be useful in the field of systems biology. In addition, we will also provide a connection with data-driven approaches and hybrid modeling, which represent the core topics in the Chapter 5 of this thesis.

Genome-scale metabolic models are mathematical representations of all known metabolic reactions in an organism. These models are built using information from the genome, which tells us which enzymes the organism can produce and which biochemical reactions they can catalyse. A GEM is usually represented as a large network, where nodes are metabolites and edges are reactions connecting them. Each reaction is linked to one or more genes through known gene-protein-reaction associations [62]. Figure 2.1 represents a graphical representation of the *E.coli* core GEM, a manually reduced GEM focused on central carbon metabolism of *Escherichia coli*.

GEMs are useful because they provide a global view of metabolism, allowing simulations of growth under different conditions, and help identify essential genes or reactions; however, using them alone is often prohibitive, as estimating reaction rates (fluxes) requires detailed kinetic parameters and enzyme concentrations, which means costly and time-consuming experiments. A flux v is typically described as:

$$v = k_{cat} \cdot e \cdot f(s, p) \tag{2.7}$$

where:

- k_{cat} is the catalytic constant, or turnover number, which represents the number of substrate molecules converted to product per enzyme molecule per unit time when the enzyme is fully saturated with substrate (molecule that is consumed by a reaction), i.e. the enzyme efficiency;
- e is the enzyme concentration;
- $f(s, p)$ is a (often nonlinear) function of the concentrations of substrates s and products(molecule that is produced by a reaction) p, and the corresponding affinity parameters.

For these reasons, GEMs are often used together with constraint-based modeling (CBM). This approach does not require precise kinetic information but instead relies on

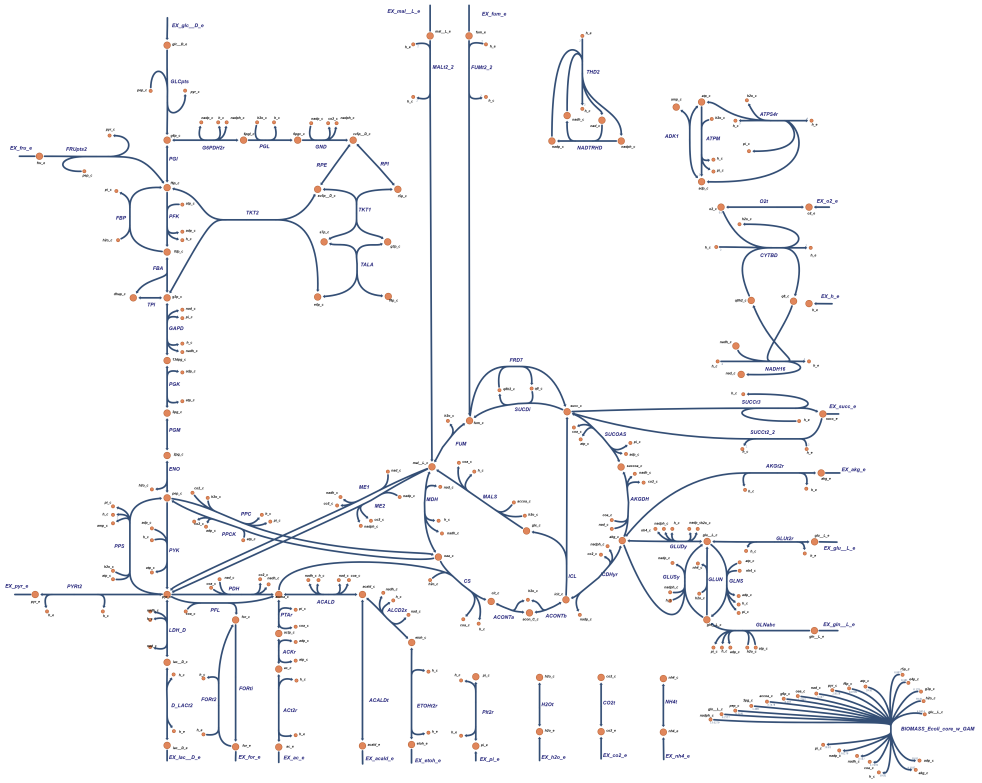

Figure 2.1: *Graphical representation of the E. coli core genome-scale metabolic model generated using Escher [83]. Metabolites are shown as nodes and biochemical reactions as edges, illustrating the structure of central carbon metabolism.*

general assumptions and constraints to study the system. The most common method in this framework is called Flux Balance Analysis (FBA). FBA assumes that the cell is in a steady state, meaning that the concentration of metabolites remains constant over time. It uses a stoichiometric matrix, usually noted as S, which represents how each metabolite is consumed or produced by each reaction. The system is then solved as a linear programming problem: we look for a set of reaction fluxes that satisfy the constraints and optimize an objective function, often the biomass production [124]. To formalize this, we have:

$$\begin{aligned} \text{Maximize:} \quad & c^T V \\ \text{Subject to:} \quad & SV = 0 \\ & V_{\min} \leq V \leq V_{\max} \end{aligned}$$

where:

- $V \in \mathbb{R}^n$ is the vector of reaction fluxes, i.e., $V = (v_1, v_2, \ldots, v_n)^T$
- $S \in \mathbb{R}^{m \times n}$ is the stoichiometric matrix, with entries s_{ij} representing the stoichiometric coefficient of metabolite m_i in reaction v_j
- $c \in \mathbb{R}^n$ is the objective coefficient vector, used to define which flux (or combination of fluxes) is to be maximized (e.g., biomass or metabolite production)
- $V_{\min}, V_{\max} \in \mathbb{R}^n$ are the vectors of lower and upper bounds on each reaction flux. These bounds capture known physiological or thermodynamic constraints (e.g., irreversible reactions have $v_j \geq 0$, reversible ones may have $v_j < 0$)

The constraint $SV = 0$ ensures mass balance at steady state: for each metabolite, the total production equals total consumption. The inequality $V_{\min} \leq V \leq V_{\max}$ defines feasible ranges for each reaction flux based on experimental data or biological knowledge.

Many software toolboxes are available to perform the linear programming optimization required for FBA, such as the COBRA Toolbox (MATLAB) [66], COBRApy (Python) [44],and CBMpy (Python) [122]. All of them support a wide range of constraint-based modeling tasks.

FBA is a powerful approach because it allows predictions of metabolic fluxes without requiring detailed kinetic parameters; however, its predictive accuracy is limited by the steady-state assumption, which cannot capture dynamic changes, and by the dependence on known uptake fluxes, as imprecise values for these can strongly affect the quality of the predictions [125]. In particular, one critical limitation is the lack of a straightforward way to convert medium composition, defined by extracellular metabolite concentrations, into quantitative bounds on uptake fluxes, which are essential for computing growth or other phenotypes.

In recent years, machine learning has become a valid alternative for this task, especially with the increasing availability of high-throughput omics data [6, 56, 190]. One of the main advantages of machine learning is that it can work with any kind of data, not just fluxes, and it doesn't rely on any mechanistic assumptions. However, this flexibility comes with some important drawbacks. First, machine learning models behave mostly as black-boxes, making it hard to extract any mechanistic understanding from their results. Second, they typically require large amounts of data to be trained properly, which is often not available in biology-related fields, as discussed in the previous sections. Because of these complementary strengths and limitations, it seems natural to try and combine mechanistic modeling with machine learning. This idea has gained a lot of attention recently, and several works have explored how to integrate these two approaches, as reviewed in [143] and [202]. According to [143], most of the existing approaches fall into two categories: using ML to provide inputs for FBA [39, 81, 115], or using the output of FBA as input for ML models [37, 106]. While this combination

can be effective, it is not a true integration, as the two models are applied one after the other rather than being merged into a single framework. To the best of our knowledge, only two works have proposed hybrid models that truly integrate ML and FBA:and [65]. The first introduces Artificial Metabolic Neural Networks (AMNs), a type of neural network that incorporates mechanistic constraints from FBA directly into the training process. This is done by including a mechanistic layer inside the network, and a custom loss function that reflects the structure of the underlying metabolic model and the biological costraints, in a way similar to other Knowledge-Informed Neural Networks, such as Physics-Informed Neural Networks [38]. The second work, FlowGAT [65], uses a Graph Attention Network (GAT) to predict gene essentiality by combining both the structure of the GEM and the solution provided by FBA. Unlike the previous approaches, these models represent a more complete integration between data-driven learning and mechanistic modeling.

Chapter 5 of this thesis fits into this context by presenting a hybrid modeling approach that follows the blueprint of AMNs proposed in [46], and extends it to the integration of multi-omics data. In this way, we combine the strengths of machine learning, flexibility in handling different input data and strong predictive performance, with those of mechanistic models, such as lower data requirements and interpretability of the results.

Chapter 3

Improving microbiome-based disease prediction with SuperTML and data augmentation

The use of neural networks in the analysis of microbiome-based datasets is limited by the small number of samples and the high dimensionality of the data. These limitations can lead to overfitting and poor generalization, making classical deep learning approaches less suitable for this context. In this chapter, we present our work "Improving microbiome-based disease prediction with SuperTML and data augmentation" published in "IEEE access" [172], where we explore the use of SuperTML, a novel deep learning method originally developed for small tabular datasets, applied for the first time to microbiome-based disease prediction. SuperTML converts microbiome abundance tabular datasets into 2D images and processes them with convolutional neural networks, transforming the task into an image classification problem. To further improve the performance and reduce overfitting, we also apply data augmentation techniques, using several image transformations commonly used in image processing to artificially increase the variability and robustness of the training set.
After the Introduction 3.1, the chapter is structured as follows. The Related Works section 3.2 reviews the use of SuperTML in the literature and explores image augmentation techniques commonly used in image processing as a regularization strategy. The Materials and Methods section 3.3 describes the SuperTML architecture, the visual transformation process, and the image augmentation techniques tested in this work. The Results and Discussion 3.4 reports the performance comparison between SuperTML, DeepMicro, and standard neural networks across six disease datasets, and discusses the role of data augmentation in improving performance. Finally, in the Concluding Remarks and conclusion section 3.5, we summarize the main findings of the chapter, discuss current limitations related to data size, model interpretability, and computational requirements, and suggest possible directions for future work.

3.1 Introduction

With the term ”human microbiome” we refer to all the microorganisms in our bodies, such as bacteria, viruses, fungi, etc. These microbes live in the gut, skin, mouth, and respiratory system. The gut microbiome, particularly, plays a central role in digestion, metabolism, immunity, and protection against harmful bacteria. The composition of each person's gut microbiome is unique and affected by a balance of causes such as genetics, diet, and lifestyle. A shift in this balance can cause obesity, diabetes, colorectal cancer, inflammatory bowel disease, and mental disorders. This specific relationship to diseases is a relatively new and developing field that may provide prophylactic or therapeutic tools to improve human health [31] [1]. In recent years, applying deep learning computational methods in biomedical research has led to revolutionary advances in the diagnosis and prediction of diseases [197]. High-dimensional tabular data, such as those derived from microbiome studies, represent a significant challenge due to data scarcity [18], making it difficult for traditional feed-forward neural networks (FNNs) models to generalize effectively. Typically, tabular data are analyzed using classical machine learning models, which are shown to be state-of-the-art for prediction tasks. The authors in [59] investigate the weaknesses of FNNs on this kind of data compared to tree-based models, finding that FNNs are not robust to uninformative features, which are common in domains characterized by high-dimensional data. In 2019, [163] presented SuperTML, a method inspired by Super Characters method [164], which showed encouraging performance, especially with small datasets. This framework is based on two steps: the first one is embedding the one-dimensional vectors into 2d images. The second one involves the use of a Convolutional Neural Network (CNN) to perform the downstream task, i.e., classification. The original work presents two main limitations: the lack of hypotheses on how the inner mechanism of SuperTML works and the use of trivial datasets, such as Iris [137], Wine [160], and Adult [12], to evaluate the performance of this approach.

In this chapter, we present our work which aims to test this framework in a challenging scenario: microbiome-disease prediction, where the datasets are small and high-dimensional. Compared to the simple benchmark datasets used in the original SuperTML paper, microbiome datasets are more complex mainly because of their high number of features. This makes the embedding step of SuperTML particularly challenging, since it's hard to fit and organize a large number of features into a limited pixel space. For this reason, we consider microbiome-disease prediction a proper challenging setup to test how robust and scalable SuperTML really is.

We aim to compare the performance of SuperTML with classical machine learning methods that reached the state-of-the-art in this domain. For this reason, we compare our results to DeepMicro [121]: a framework to predict the presence or absence of a particular disease based on data of strain level and species level abundance profiles.

DeepMicro is a two-stage approach: first, it applies an autoencoder to reduce the dimensionality of the original input data. Then, a machine learning model performs the classification using the learned latent space representation.
Furthermore, since SuperTML reformulates the problem as an image processing task, we apply image augmentation techniques to synthetically enlarge the datasets and regularize the model. This capability is an intrinsic advantage of SuperTML in the context of data scarcity, as augmentation is widely recognized as one of the most effective regularization techniques in image processing [195]. We test several augmentation transformations to investigate which ones are effective for the peculiar images created by the embedding step of SuperTML.
Lastly, in the discussion section of this chapter, we examine our results, highlighting the strengths and identifying the various limitations of the SuperTML framework.
To summarize, the main contributions of this work are:

- A comparative analysis between SuperTML and classic FNNs over six microbiome-disease datasets. SuperTML consistently outperformed FNNs across five out of six datasets, confirming its superiority.
- A comparative analysis between SuperTML and DeepMicro over six microbiome-disease datasets. SuperTML, when enhanced with augmentation, achieved the highest AUC scores in five out of six datasets, demonstrating its competitive performance.
- Experiments with various image augmentation techniques showed their effectiveness in improving model performance. However, no single transformation consistently outperformed the others across all datasets, leaving open questions about the inner workings of SuperTML.
- Qualitative analysis on the various limitations of this framework especially the dependence between the image dimension and the dimensionality of the dataset which makes it computationally challenging to use SuperTML with very high-dimensional datasets.

This evaluation shows the effectiveness of SuperTML in complex scenarios and provides insights into areas where further research and development are necessary to optimize its application.

3.2 Related Works

Before going into the SuperTML method and our analysis, this section reviews its use in the literature and summarizes the main concepts of image augmentation techniques,

providing the necessary background and context for our work.

3.2.1 SuperTML

Deep Learning (DL) has become the standard approach in several fields, such as Computer Vision (CV) [182], Natural Language Processing (NLP) [95], and Speech Recognition (SR) [110]. DL's achievements in such domains rest in its ability to learn complex hierarchical representations of the data, especially in the case of data with an underlying structure such as grid-like data, sequences, graphs, etc. DL architectures, namely Convolutional Neural Networks (CNNs), transformer-based models, and Recurrent Neural Networks (RNNs), are built to exploit such properties [22], which explains their great success in the domains mentioned above. Applying DL to tabular data, especially high-dimensional ones, is still challenging. The authors in [18] concluded that ML methods are still state-of-the-art for small and medium-sized datasets (less than 1M samples); the only cases in which DL outperforms classical ML approaches are vast datasets. Another contribution of [18] is the introduction of a taxonomy that organizes DL methods for tabular data into three groups: data transformation methods, specialized architectures, and regularization models.
This section reviews previous works on SuperTML [163], a data transformation method that embeds one-dimensional vectors into images by printing each element in the image canvas and then applying a CNN to the generated image. SuperTML borrows the idea from the Super Characters method [164], a technique used to convert a sentiment analysis task to an image classification one. The same idea has also been applied to image captioning with SuperCaptioning [165]. The authors of SuperTML explicitly highlight the excellent performance on small datasets and the limited amount of overfitting due to the usage of transfer learning with pre-trained CNN.
After the publication of [163], other works used SuperTML to benchmark real-world datasets. The authors in [149] compared the performance of SuperTML with other ML methods on medical data, concluding that SuperTML competes with ML tree-based models, but they remain the best choice for those kinds of data. In contrast, the authors in [91] tested SuperTML for dengue prediction using weather data; they showed that SuperTML, in combination with Resnet18, significantly outperforms classical ML methods on a small dataset. The authors in [112] compared SuperTML with a 1D-CNN on an industrial dataset. Their conclusion highlighted the superiority of both CNN approaches w.r.t classical ML ones. However, due to the high computational time of SuperTML, they suggest the 1D-CNN for that kind of data. The authors also concluded that SuperTML works better with the lower-dimensional dataset because of the simplicity of the images created.
Recently, [74] proposed a new method, called Dynamic Weighted Tabular Method (DWTM), inspired by the Variable Font-SuperTML (VFTML) [163]: a version of SuperTML where the font size of the printed digits depends on the feature importance

of the variable. DWTM, instead, assigns an area of the image to each digit based on its weight. The weights represent the associativity between each feature and the class (e.g., Pearson correlation and chi-square test). The results show that DWTM outperforms other ML methods on six benchmark datasets.
Similar to SuperTML, several recent studies have explored using CNNs for tabular data analysis. One approach involves converting low-dimensional and mixed-type tabular data into 2D images, as seen in the LM-IGTD framework [61]. Another method, Tab2Visual [108], transforms tabular datasets into visual representations, where each feature is encoded as a bar of varying width. This transformation enables CNNs and Vision Transformers (ViTs) to process tabular data as images. Tab2Visual also incorporates image augmentation and transfer learning to improve classification accuracy. A different strategy, Fuzzy CNNs (FCNNs) [86], introduces fuzzy logic to convert tabular features into images by mapping features to fuzzy membership values, which are then represented as rectangular shapes, allowing CNNs to capture feature importance while maintaining interpretability. A significant limitation noted in the existing literature is that studies focus on datasets with relatively few features. This work, instead, focuses on SuperTML applied to microbiome data, a domain characterized by small and high-dimensional datasets. Furthermore, considering SuperTML a fully fledged image processing method, we test several augmentation techniques suitable for it. For this reason, we dedicate the following section to reviewing augmentation techniques for image data.

3.2.2 Data Augmentation

Due to the high complexity of DL models, overfitting is one of the central issues in this field. One method that can help reduce it is data augmentation. Data augmentation is any perturbation applied to the data aimed at enlarging the size of the training set. By doing this, data augmentation helps to increase the variability of the dataset and improve the generalization of the models trained on it.
In the image domain, data augmentation refers to those techniques that alter the image by flipping, mixing, erasing, etc. More precisely, using the taxonomy introduced by [195], these methods can be divided into basic and advanced approaches. The basic approaches are further grouped into Image Manipulation, Image Erasing, and Image Mix techniques.
Image Manipulation includes transformations such as rotation, flipping, and cropping. Image Erasing contains those methods that delete one or more sub-regions by replacing the pixel values with a constant or random value [28, 42, 155, 207]. Image Mix regards those techniques that mix two or more images or sub-regions of the images into one [64, 67, 73, 180, 199].
The advanced approaches are divided into Auto Augment, Feature Augmentation, and Deep Generative Models-based techniques. Auto Augment refers to those methods that

automatically search the optimal augmentation approach to improve performance [34, 69, 100]. Feature augmentation refers to those techniques that apply the transformation on the learned feature space conversely to the input space [41, 87, 94]. The Deep-Generative Models-based techniques exploit GAN [57] to keep the gap between the augmented image and the original one from being too large, ensuring that the two images belong to the same data distribution.
In our analysis, we employ techniques mainly from the Image Manipulation and Image Erasing groups, which we believe are particularly well-suited for handling the images generated through SuperTML.

Table 3.1: *Summary of disease datasets.*

Disease	**Dataset**	**# Samples**	**# Controls**	**# Patients**	**# Features**
Inflammatory Bowel Disease	IBD	110	85	25	443
Type 2 Diabetes	EW-T2D	96	43	53	381
Type 2 Diabetes	C-T2D	344	174	170	572
Obesity	Obesity	253	89	164	465
Liver Cirrhosis	Cirrhosis	232	114	118	542
Colorectal Cancer	Colorectal	121	73	48	503

3.3 Materials and Methods

3.3.1 Datasets

To evaluate the performance of SuperTML and the tested augmentation techniques, we considered six publicly available datasets analyzed in the DeepMicro work, featuring human gut metagenomic species-level relative abundance profiles linked with different diseases: inflammatory bowel disease (IBD) [135], type 2 diabetes in European women cohort (EW-T2D) [79], type 2 diabetes in Chinese (C-T2D) [134], cohortobesity (Obesity) [90], liver cirrhosis (Cirrhosis) [136], and colorectal cancer (Colorectal) [203].
We selected these datasets for two reasons: first, these are relatively small datasets (less than 500 observations), such as many datasets in the microbiome research domain, and second, their dimension (between ~ 400 and ~ 600 features) is difficult to represent in the SuperTML embedding. For these reasons, these datasets represent a challenge for SuperTML that has yet to be explored.
We did not select the six strain-level relative abundance profiles, also present in the DeepMicro article, because the number of features is prohibitively big, $\sim 10^5$, to fit in the SuperTML image embedding without choosing a font size excessively small or an image dimension too big for computational feasibility.
We also selected another dataset, the HIGGS dataset [3]. Unlike microbiome-disease datasets, which are characterized by high-dimensional data and small sample sizes, the HIGGS dataset consists of approximately 700k observations with only 30 features. This difference allows us to examine whether the improvements observed with SuperTML and its augmented versions hold when applied to a dataset with a completely different structure.

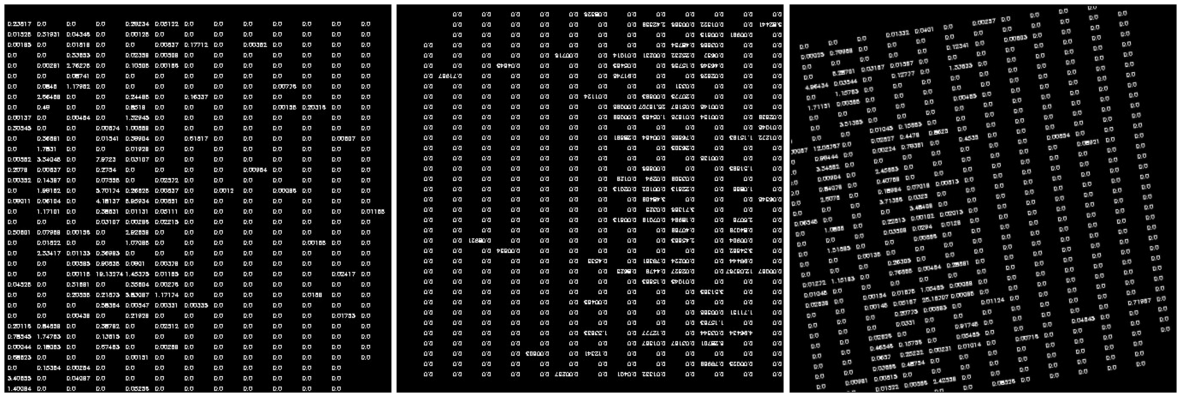

Figure 3.1: *SuperTML images with baseline augmentation types: the first image shows the original SuperTML without any augmentation, the second image demonstrates the effect of RandFlip, where the image is randomly flipped along a selected spatial axis, and the third image applies RandRotate, introducing a random rotation within a specified angle range.*

Figure 3.2: *SuperTML images with different Image Erasing augmentation types: the first image applies RandomErasing, where a randomly sized rectangular region is removed and replaced with a constant value; the second image represents RandCoarseDropout, which removes multiple randomly sized rectangular regions and replaces them with a fill value; the third image shows RandCoarseShuffle, where the selected regions are shuffled; and the fourth image applies CellDropout, specifically removing the exact areas where features are printed, simulating features dropping.*

3.3.2 Methods

As introduced in the previous sections, SuperTML is an approach to tabular data analysis that leverages two-dimensional embeddings. This idea has been successful in the NLP domain, specifically by transforming a sentiment analysis task into an image classification task [164].
In this framework, each data point is translated into an image format, representing each feature as a floating point number printed onto a black background. This image is then processed using a Convolutional Neural Network (CNN) to perform the downstream prediction task. The original work pre-trained the CNNs on more extensive datasets, such as ImageNet [40]. Here, we train the CNNs from scratch since we have not noticed any improvement related to pre-trained models.
Another important detail is the choice of the image dimension, which strictly depends on the font's dimension, the digits' precision, and the number of data features. For computational demand reasons, we chose not to have bigger images than $450x450$ in the DeepMicro dataset, and we adjusted the font size and the precision of the digits accordingly. In the case of HIGGS data, which is characterized by only 30 features, the image dimension is $224x224$.
By converting the problem into image processing, this work further explores image augmentation techniques to assess their effectiveness for the unique images generated by SuperTML. Our primary interest is to check which kind of augmentation is best suited to this framework; for this reason, we tried several augmentation techniques for each dataset. First, classical augmentation transformations, such as Random Rotation and Random Flipping, have been used as a baseline. Figure 3.1 shows them in comparison with an original non-augmented SuperTML image.

Then, we tried several Image Erasing transformations, such as Random Erasing, Random Coarse Dropout, and Random Coarse Shuffle, and we implemented a custom transformation named Random Cell Dropout. In the context of SuperTML images, the idea behind Image Erasing techniques is masking the features to force the model to learn with a randomly selected feature subset. The issue with these techniques is that they do not erase exactly the area of the image belonging to the represented feature but an area that potentially can partially overlap with a feature; to address this, we implemented the Cell Dropout method. Cell Dropout is specifically implemented for the SuperTML generated images: it randomly selects a subset of features and masks them completely. Figure 3.2 shows the differences between all the Image Erasing methods.

To complete the analysis, we also tried other techniques belonging to the Image Manipulation family, such as Random Zoom, which acts similarly to erasing transformation, forcing the model to focus on a subset of features, but it also modifies them by zooming on them. Random Elastic, which distorts the features as shown in Figure 3.3, and Random Gaussian Noise, which adds noise randomly sampled from a Gaussian distribution. The idea behind the Image Manipulation transformations is to force the model to learn using distorted/manipulated digits in order to improve its ability to distinguish them, which intuitively is crucial in the context of SuperTML. Fig 3.4 shows the details of the digits in the Random Gaussian Noise and Random Elastic transformations. For clarity, we also included more explicative figures and detailed descriptions of these transformations in the following section.

All the augmentation transformations were implemented using the MONAI library [25].

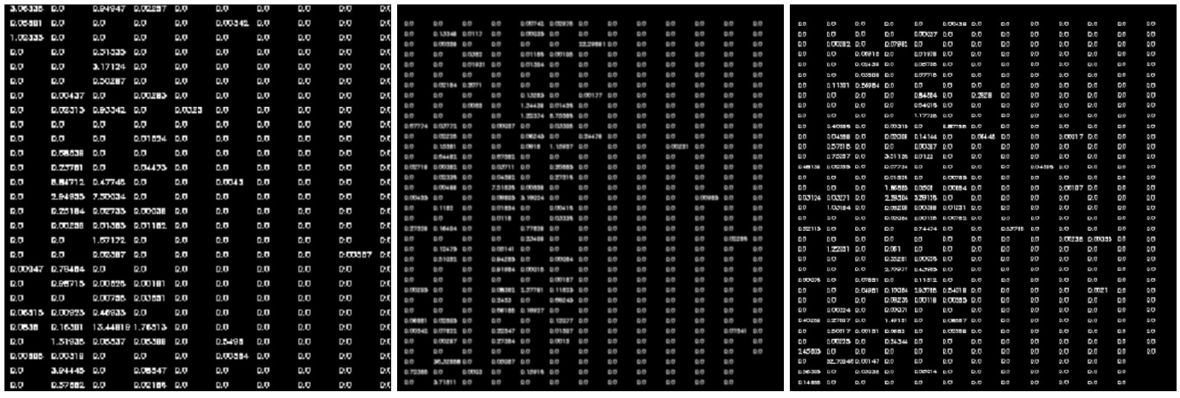

Figure 3.3: *SuperTML images with different Image Manipulation augmentation types: the first image applies RandZoom, which randomly scales the image by a factor within a specified range; the second image represents RandGaussianNoise, where random Gaussian noise is added to the image; and the third image applies Rand2DElastic, introducing smooth, localized deformations to mimic realistic elastic distortions in the image.*

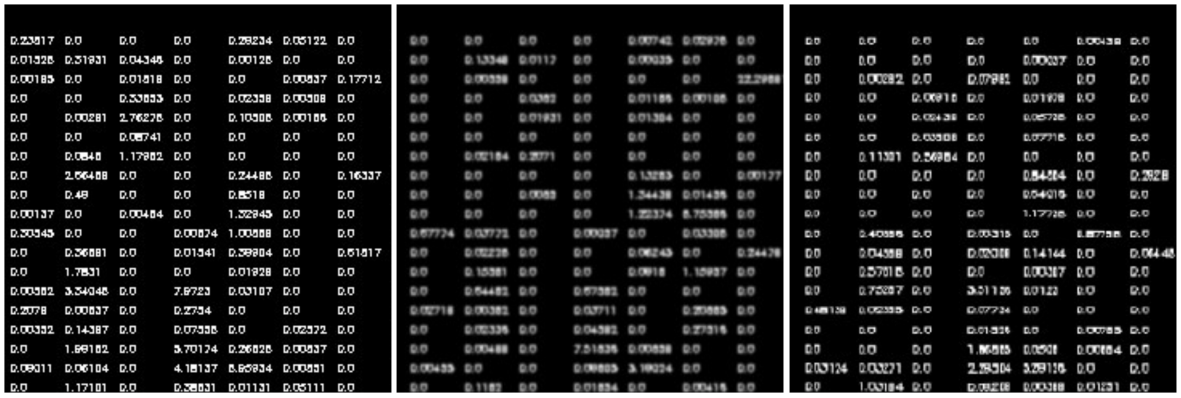

Figure 3.4: *Zoomed-in views of the SuperTML images, focusing on the original (non-augmented), RandGaussianNoise, and Rand2DElastic versions to better highlight the detailed effects of these transformations.*

3.3.3 SuperTML and Augmented Images: toy examples

This section presents a detailed description of each image augmentation technique used in our analysis. We created a toy example of a vector of 25 randomly generated values to have an easily readable image. We grouped the image augmentations as we did in the Methods section; in this way, they can represent a more explicative version of the previous figures.
Figure 3.5 shows the SuperTML embedding without any transformation applied.

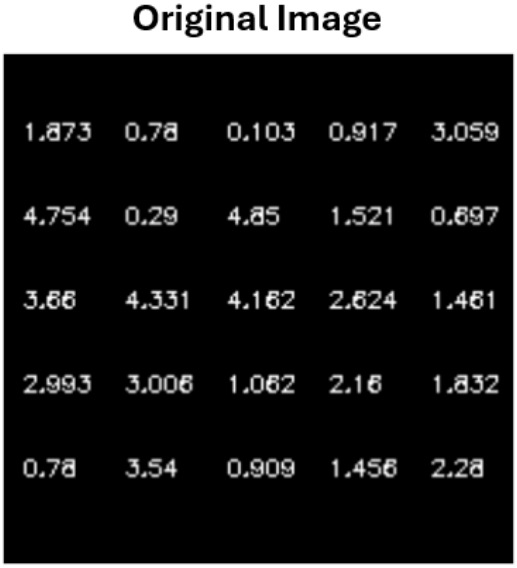

Figure 3.5: *Image created by SuperTML, it represents the 2D embedding of a data-point. In this case randomly generated vector of 25 values.*

Figure 3.6 shows the Random Flip and Random Rotation transformations. Random Flip randomly flips an input image along specified spatial axes with a given probability. In this case, the image is first flipped along the horizontal axis and then the vertical one. Random rotation, instead, randomly rotates images by a specified angle range along given spatial axes.

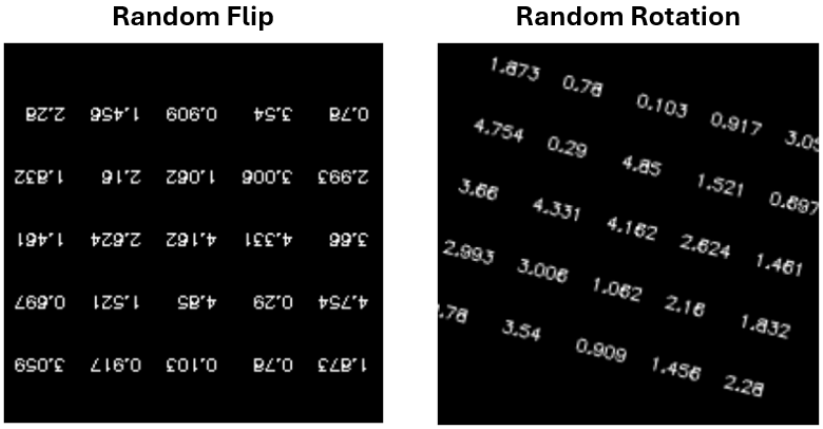

Figure 3.6: *On the left: the Random Flip augmented image. On the right: the Random Rotation augmented image.*

Figure 3.7 shows 4 different transformations belonging to the Image Erasing family. The Random Erasing transformation randomly selects a rectangular region with random dimensions and replaces it with a constant value, noise, or random pixels. The Random Coarse Dropout randomly selects multiple rectangular regions of random sizes and replaces them with a specified fill value. The Random Coarse Shuffle, instead, randomly selects multiple rectangular regions of random sizes and shuffles their contents within the image. The CellDropout transformation randomly selects the regions belonging to feature values and replaces them with zero values that correspond to black pixels; in this way, it randomly drops a specified number of features.

Figure 3.8 shows all three transformations belonging to the Image manipulation group. Random Zoom will randomly scale an image by a certain factor within given minimum and maximum, zooming it either in or out, while preserving the aspect ratio. Random Gaussian Noise adds random Gaussian noise to an image, with given mean and standard deviation. Random Elastic applies random elastic deformation by displacing image pixels locally, similarly to real life elastic distortions.

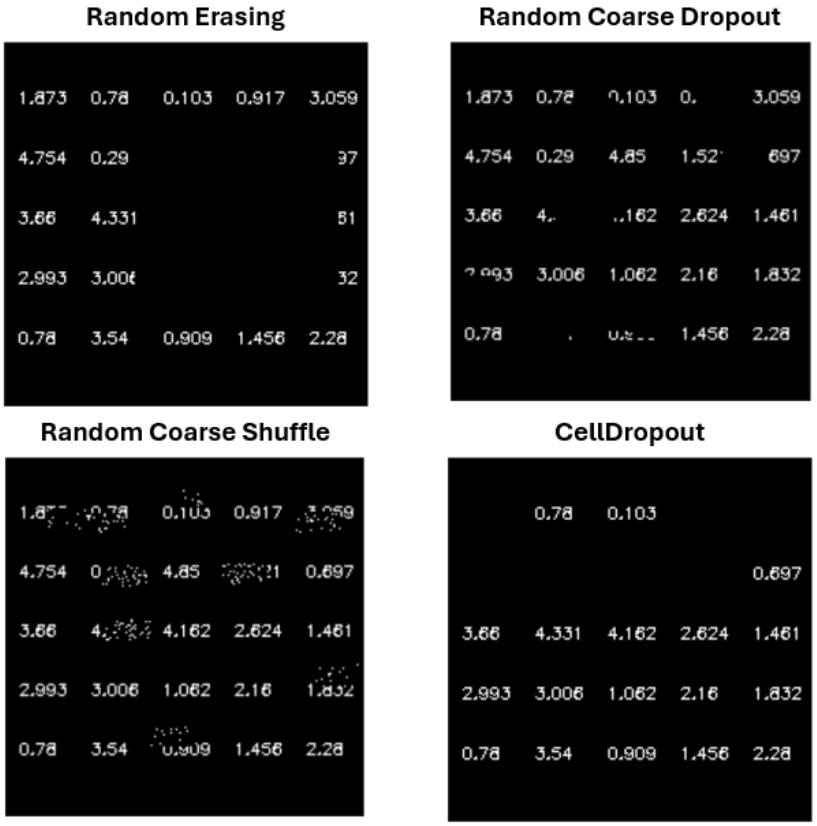

Figure 3.7: *On the top left: Random Erasing augmented image. On the top right: the Random Coarse Dropout augmented image. On the bottom left: the Random Coarse Shuffle augmented image. On the bottom right: the CellDropout augmented image.*

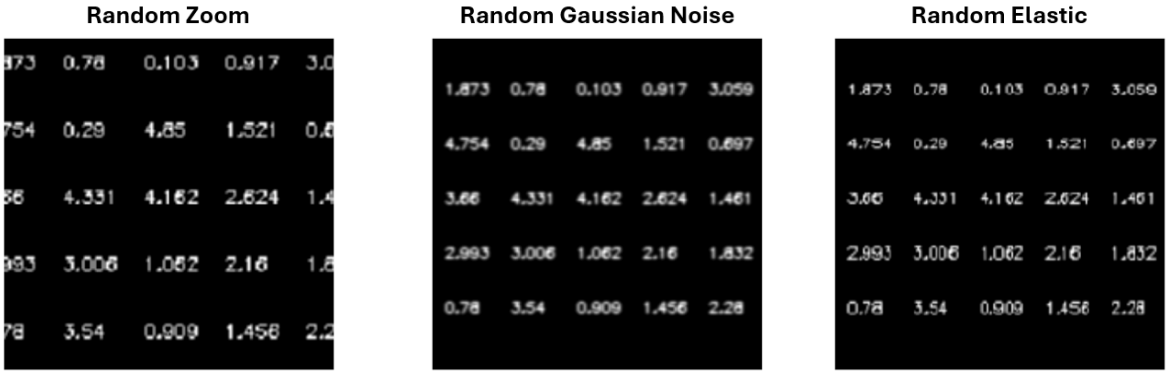

Figure 3.8: *On the left the Random zoom augmented image. In the center: the Random Gaussian Noise augmented image. On the right: the Random Elastic augmented image.*

3.3.4 Performance evaluation

Experimental Setup

In order to have a fair comparison with the DeepMicro results, we used the same evaluation pipeline. It consists of nested cross-validation: the outer loop splits the dataset into train and test, respectively 80% and 20%, using $0, 1, 2, 3, 4$ as random seed to assure reproducibility. Then, there is an inner k-fold cross-validation (k=5) for each split to optimize the hyperparameters over the train and a final evaluation on the test set with the best model found.
We calculate the metrics for each split and report the mean and the standard deviation. Figure 3.9 shows the described evaluation pipeline. We calculated the accuracy (ACC) and the area under the curve (AUC) but, following DeepMicro work, we used only the AUC to compare the performance among the models.
The HIGGS dataset contains around $700k$ samples and for this reason it does not need a cross-validation pipeline. We just split the dataset into train, validation, and test sets ($\sim 200k$, $\sim 50k$, $\sim 450k$). To quantify the performance of the models, we used the approximate median significance (AMS) metric:

$$AMS = \sqrt{2\left((s+b+b_r)\log\left(1+\frac{s}{b+b_r}\right)-s\right)}$$

where s, b are the unnormalised true positive and false positive rates, respectively. b_r is 10, the constant regularisation term and log is the natural logarithm.
More precisely:

$$s = \sum_{i=1}^{n} w_i \mathbf{1}\{y_i = s\}\mathbf{1}\{\hat{y}_i = s\}$$

and

$$b = \sum_{i=1}^{n} w_i \mathbf{1}\{y_i = b\}\mathbf{1}\{\hat{y}_i = s\}$$

where $(y_1, \ldots, y_n) \in \{b, s\}^n$ is the vector of true labels, $(\hat{y}_1, \ldots, \hat{y}_n) \in \{b, s\}^n$ the vector of predicted labels, $(w_1, \ldots, w_n) \in \mathbb{R}_+^n$ is the vector of weights and $1\{A\}$ the indicator function which is 1 if the argument is true and 0 if it is false.
As detailed in [3], the weights are an artifact of the way the ATLAS full-detector simulation works and for this reason they are not given as an input to the classifier, they are used only to calculate the AMS metric to evaluate and test the models. As extensively presented in [4], in High-Energy Physics (HEP), detecting rare events like Higgs boson decays involves defining a "search region," i.e., a region of the 30-dimensional space of input variables. If the events observed in this region significantly

exceed the expected number of background events, the background-only hypothesis can be rejected, providing statistical evidence for the signal. If the signal process is present, then the observed statistical significance with which one rejects the background-only hypothesis can be approximated by the already defined AMS [33]. AMS quantifies a classifier's ability to separate signals from the background by maximizing true positive detections while minimizing false positives. Unlike simple accuracy, which is not helpful in highly imbalanced datasets (signal events are extremely rare w.r.t. background ones), AMS focuses on statistical significance, ensuring that a classifier optimally detects rare signals, keeping false positive detections low.

Hyperparameters

As introduced in the previous section, the inner loop of the evaluation pipeline is a k-fold cross-validation (k=5) to find the best hyperparameters. Here, we briefly present the structure of the CNN architecture used for the second step of SuperTML and all the hyperparameters related to it, the learning process, and the augmentation stage.
For both the analyses on DeepMicro and HIGGS datasets, as anticipated in the Method section, we trained a CNN architecture built as follows from scratch: n blocks of convolutional layers, Leaky ReLU [191], Batch Normalization [191], and Max Pooling [200]. After the n blocks, a flattening operation is followed by a dropout [158] and a final linear layer for the final classification step.
The hyperparameters search space consists of the learning rate, number of convolutional layers, number of kernels, L2 regularization value, and dropout intensity. Furthermore, we fixed several hyperparameters: each parameter in the augmentation functions is fixed, each random augmentation's probability is 0.5, and the optimizer used is Adam.

3.4 Results and Discussion

In this section, we present all the results obtained from our analysis. As shown in Figure 3.10, SuperTML without augmentation outperforms a standard FNN in five out of six datasets, while with augmentation, it consistently achieves better performance across all six datasets. These results confirm the effectiveness of SuperTML compared to FNNs in an unexplored challenging scenario: microbiome-based high dimensional and small datasets. The fact that SuperTML performs better than simple FNNs can be explained by CNN's ability to learn in high dimensions. This means that it is easier to learn from an image using the inductive biases of the CNN compared to learning in high-dimensions with a feedforward neural network, which will face the curse of dimensionality. Additionally, the SuperTML framework can benefit from the use of image augmentation to further improve its performance. Regarding the comparison between the DeepMicro framework and SuperTML, the results show competitive performance

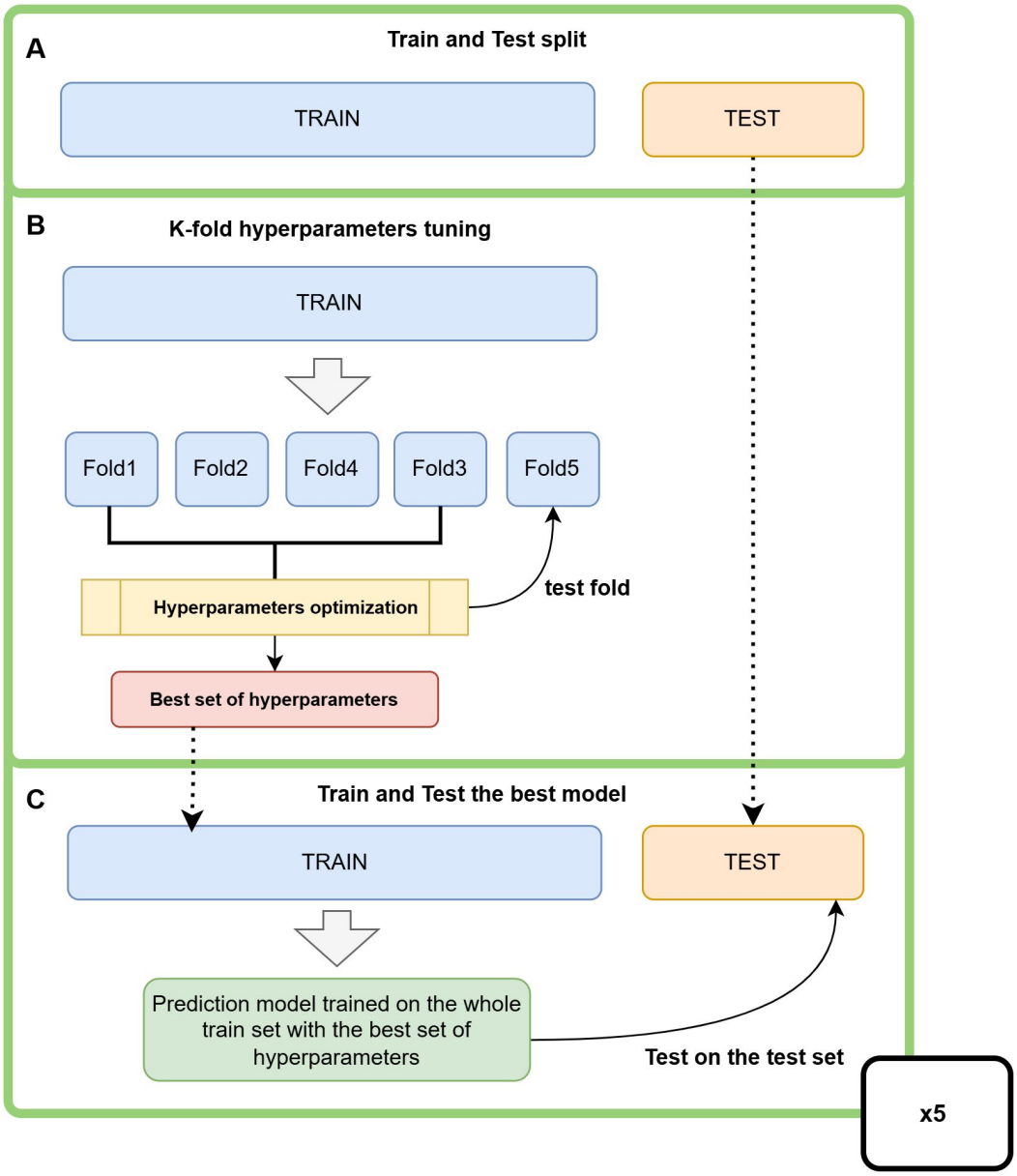

Figure 3.9: *Evaluation pipeline used for DeepMicro datasets. A) The first step is a split of the dataset in a train and test set. B) The second step is a k-fold cross-validation (k=5) loop for optimizing the hyperparameters: here, the training set is split into five folds, and the model is trained on four folds and validated on the remaining one. This process is repeated five times, each time using a different fold for validation. The final performance is then averaged across all iterations to select the best hyperparameters. C) The final step consists of training the model on the whole train set using the best hyperparameters and then testing the performance on the test set. These steps are repeated 5 times and the final test performances are averaged across the different test splits.*

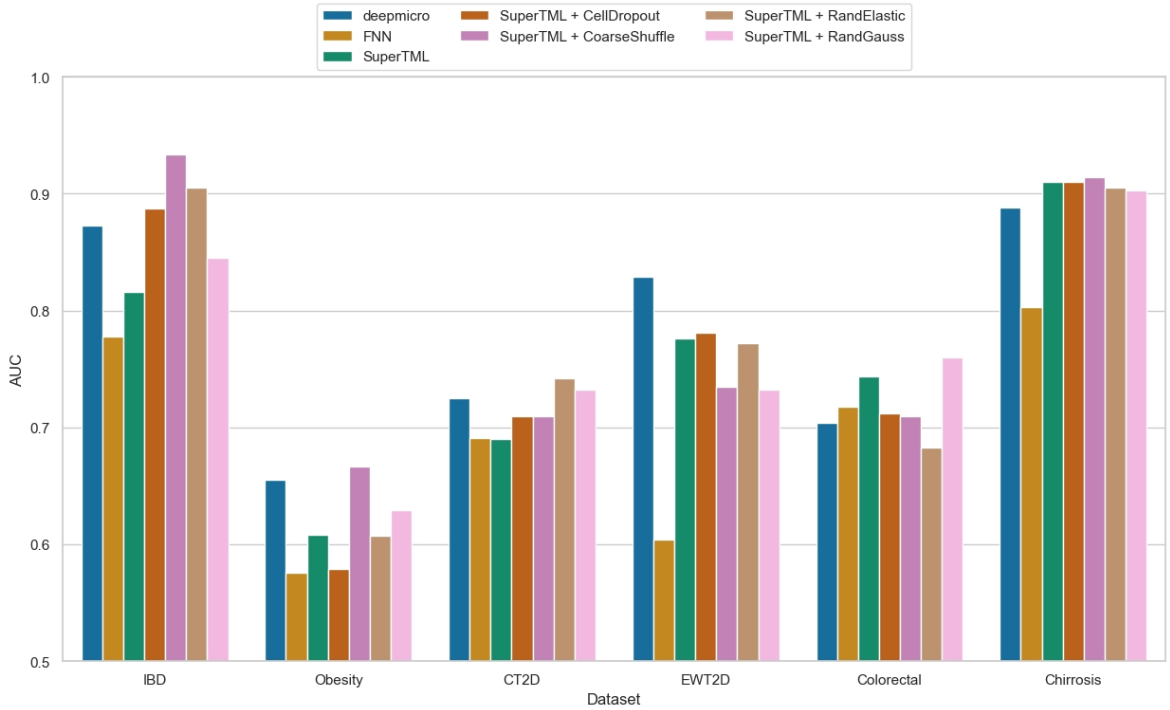

Figure 3.10: *Results on the 6 DeepMicro datasets: each bar represents the average models AUC performance over 5 different test splits*

for SuperTML used together with augmentation, obtaining higher AUC scores for five datasets out of six. This result shows that SuperTML with augmentation can be considered a valid alternative to classical machine learning models in the context of microbiome used as a predictor of disease. Another important observation from Figure 3.10, concerns the performance across different disease datasets. While SuperTML with augmentation shows competitive performance, some datasets: Obesity, CT2D, EW-T2D, and Colorectal; consistently show lower AUC scores across all models. Our hypothesis is that this lower performance is due to two key factors. First, these diseases likely have a weaker correlation with gut microbiome composition, making classification naturally more challenging. Second, the type of data used plays a crucial role. In DeepMicro's analysis [121], both strain-level marker profile and species-level relative abundance profile datasets show the same pattern of lower performance for these diseases. However, strain-level marker profile datasets achieve higher average performance across all diseases, suggesting that species-level relative abundance profiles may not be informative enough for this kind of task.
Another relevant result is shown in Figure 3.10: the augmentation improves the AUC score consistently for all six microbiome-disease datasets, w.r.t. SuperTML used without augmentation. The same result appears in Figure 3.11 , where the data augmentation improves the AMS metric for the HIGGS dataset. These results suggest that

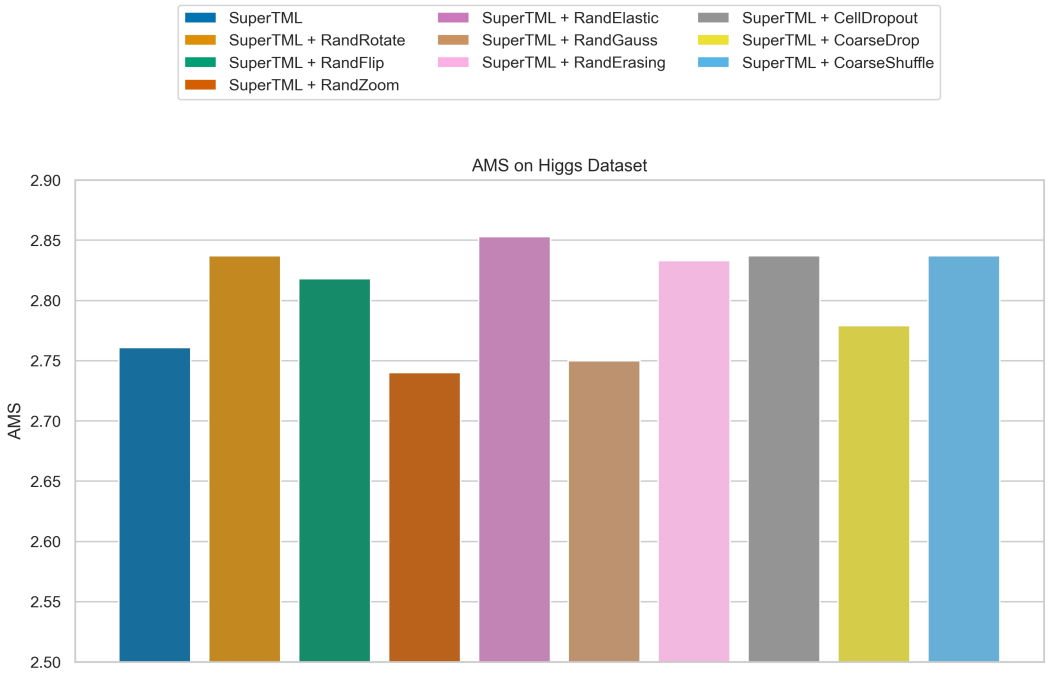

Figure 3.11: *Results on HIGGS dataset: each bar represents the models AMS metric over the test split*

SuperTML benefits from augmentation as a regularization technique, just as any other image processing framework based on CNN. Since the HIGGS dataset differs significantly from microbiome-disease datasets in both scale and dimensionality, these results support the effectiveness of augmentation in improving model performance across different types of data. This suggests that augmentation not only helps prevent overfitting in small, high-dimensional datasets but also enhances generalization in larger, low-dimensional ones.

The results about the most suitable transformations for this task reveal an interesting complexity. The only ones that achieve the best results for multiple datasets are the Random Coarse Shuffle on three datasets and Random Elastic on two. Others, such as Random Gauss and Cell Dropout, reach the highest AUC only on one dataset.

Cell Dropout, the transformation explicitly implemented for this framework, did not perform as expected. This suggests that there are better strategies than forcing the model to focus on a subset of features in a context such as SuperTML. However, it is still an open question about which transformation best suits this task. This could also be relevant to understanding the inner mechanism of SuperTML, meaning understanding which features it learns for the prediction. This could be relevant in the context of biomarker discovery.

Another important limitation to discuss is the link between the input data dimension and the 2d embedding dimension. Specifically, if the data dimensions are too

extensive, the resulting images become too large for effective computation, potentially restricting the applicability of SuperTML in broader scenarios, such as the strain-level profile datasets or multi-omics research which are characterized by dimensions the order $\sim 10^5$. One straightforward solution to this issue is the implementation of dimensionality reduction techniques similar to the DeepMicro framework. By reducing the dimensionality before applying SuperTML, it may be possible to maintain the integrity of the data while ensuring the computational feasibility of the model. Finally, another aspect to consider is interpretability, which is particularly challenging in SuperTML due to how the embedding works. In this step, the metagenomic features are turned into numbers that will be printed as pixels on a black canvas. This makes it very hard to trace back which features actually contribute to the prediction. This might be mitigated by using pixel-level attribution methods such as Integrated Gradients [166]. These algorithms could allow us to highlight the most important pixels for a given prediction and, if properly mapped back to the original features, they could help us understand which ones are most relevant for the disease. This could be a first step to interpret the model predictions which could be very useful to find potential biomarkers, and eventually open the way to apply these methods in the context of precision medicine.

3.4.1 Results details

This section contains the tables with the extended and detailed results for each dataset.

HIGGS		
Method	**AMS**	**ACC**
SuperTML	2.761	0.833
SuperTML + RandRotate	2.837	0.836
SuperTML + RandFlip	2.818	0.832
SuperTML + RandZoom	2.740	0.833
SuperTML + RandElastic	**2.853**	0.833
SuperTML + RandGauss	2.750	0.834
SuperTML + RandErasing	2.833	0.834
SuperTML + CellDropout	2.837	0.833
SuperTML + CoarseDrop	2.779	0.831
SuperTML + CoarseShuffle	2.837	0.836

Table 3.2: *AMS and ACC scores of SuperTML-based methods for HIGGS dataset.*

Table 3.2 reports the results for SuperTML-based models on the HIGGS dataset, with all metrics calculated on the test split. The AMS values are presented in the main manuscript as a bar plot (Figure 7), while accuracy (ACC) is reported here for completeness. As explained in the Performance Evaluation section of the main manuscript, accuracy is not an appropriate metric for rare signal events such as Higgs boson decays. As expected, ACC does not highlight any differences among the models, confirming its limited usefulness in this context.
Tables 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8 report the results for all methods on the microbiome-disease datasets. The metrics were calculated following the evaluation pipeline described in the main manuscript (Figure 5). The AUC scores for each dataset are also presented as bar plots in Figure 6.
Following the approach adopted by the authors of *DeepMicro: deep representation learning for disease prediction based on microbiome data*, we optimized all models using AUC as the target metric. For this reason, our analysis and conclusions are based primarily on the AUC results. Here, for completeness, we also report the accuracy scores, which show slightly different patterns compared to the AUC results. For example, only four out of six datasets show an improvement in accuracy when using SuperTML with augmentation compared to the basic SuperTML. Similarly, for DeepMicro, which achieved the highest accuracy in four out of six datasets.

IBD		
Method	**ACC**	**AUC**
DeepMicro	0.809 ± 0.017	0.873 ± 0.030
MLP	0.754 ± 0.046	0.778 ± 0.049
SuperTML	0.800 ± 0.036	0.816 ± 0.116
SuperTML + RandRotate	0.745 ± 0.068	0.682 ± 0.205
SuperTML + RandFlip	0.791 ± 0.022	0.865 ± 0.075
SuperTML + RandZoom	0.781 ± 0.044	0.823 ± 0.100
SuperTML + RandElastic	0.818 ± 0.121	0.905 ± 0.088
SuperTML + RandGauss	0.791 ± 0.036	0.845 ± 0.113
SuperTML + RandErasing	0.799 ± 0.054	0.837 ± 0.126
SuperTML + CellDropout	0.818 ± 0.057	0.887 ± 0.047
SuperTML + CoarseDrop	0.818 ± 0.040	0.821 ± 0.135
SuperTML + CoarseShuffle	0.781 ± 0.060	$\mathbf{0.934} \pm 0.015$

Table 3.3: *ACC and AUC scores for IBD dataset. The reported numbers are the average and the standard deviation metrics over the 5 test splits.*

C-T2D		
Method	**ACC**	**AUC**
DeepMicro	0.644 ± 0.025	0.725 ± 0.025
MLP	0.611 ± 0.052	0.691 ± 0.0451
SuperTML	0.574 ± 0.037	0.690 ± 0.054
SuperTML + RandRotate	0.605 ± 0.054	0.731 ± 0.050
SuperTML + RandFlip	0.617 ± 0.047	0.704 ± 0.040
SuperTML + RandZoom	0.599 ± 0.063	0.709 ± 0.023
SuperTML + RandElastic	0.597 ± 0.057	$\mathbf{0.742} \pm 0.054$
SuperTML + RandGauss	0.620 ± 0.045	0.732 ± 0.059
SuperTML + RandErasing	0.631 ± 0.075	0.719 ± 0.074
SuperTML + CellDropout	0.606 ± 0.044	0.710 ± 0.032
SuperTML + CoarseDrop	0.605 ± 0.044	0.711 ± 0.032
SuperTML + CoarseShuffle	0.605 ± 0.044	0.710 ± 0.032

Table 3.4: *ACC and AUC scores for C-T2D dataset. The reported numbers are the average and the standard deviation metrics over the 5 test splits.*

EW-T2D		
Method	**ACC**	**AUC**
DeepMicro	0.740 ± 0.037	$\mathbf{0.829} \pm 0.039$
MLP	0.580 ± 0.092	0.604 ± 0.185
SuperTML	0.720 ± 0.087	0.776 ± 0.151
SuperTML + RandRotate	0.590 ± 0.058	0.778 ± 0.127
SuperTML + RandFlip	0.600 ± 0.070	0.697 ± 0.118
SuperTML + RandZoom	0.650 ± 0.070	0.733 ± 0.129
SuperTML + RandElastic	0.600 ± 0.070	0.772 ± 0.132
SuperTML + RandGauss	0.610 ± 0.086	0.732 ± 0.059
SuperTML + RandErasing	0.631 ± 0.075	0.785 ± 0.124
SuperTML + CellDropout	0.620 ± 0.120	0.781 ± 0.109
SuperTML + CoarseDrop	0.659 ± 0.149	0.747 ± 0.142
SuperTML + CoarseShuffle	0.650 ± 0.070	0.735 ± 0.099

Table 3.5: *ACC and AUC scores for EW-T2D dataset. The reported numbers are the average and the standard deviation metrics over the 5 test splits.*

Obesity		
Method	**ACC**	**AUC**
DeepMicro	0.674 ± 0.034	0.655 ± 0.013
MLP	0.592 ± 0.069	0.576 ± 0.072
SuperTML	0.607 ± 0.044	0.608 ± 0.074
SuperTML + RandRotate	0.576 ± 0.056	0.624 ± 0.078
SuperTML + RandFlip	0.588 ± 0.109	0.619 ± 0.059
SuperTML + RandZoom	0.549 ± 0.096	0.625 ± 0.079
SuperTML + RandElastic	0.576 ± 0.059	0.607 ± 0.053
SuperTML + RandGauss	0.529 ± 0.118	0.629 ± 0.065
SuperTML + RandErasing	0.573 ± 0.092	0.592 ± 0.037
SuperTML + CellDropout	0.631 ± 0.057	0.579 ± 0.034
SuperTML + CoarseDrop	0.635 ± 0.053	0.612 ± 0.054
SuperTML + CoarseShuffle	0.619 ± 0.051	$\mathbf{0.667} \pm 0.079$

Table 3.6: *ACC and AUC scores for Obesity dataset. The reported numbers are the average and the standard deviation metrics over the 5 test splits.*

Colorectal		
Method	**ACC**	**AUC**
DeepMicro	0.809 ± 0.046	0.704 ± 0.020
MLP	0.608 ± 0.053	0.718 ± 0.084
SuperTML	0.679 ± 0.094	0.744 ± 0.154
SuperTML + RandRotate	0.648 ± 0.039	0.712 ± 0.114
SuperTML + RandFlip	0.617 ± 0.047	0.704 ± 0.040
SuperTML + RandZoom	0.600 ± 0.075	0.684 ± 0.124
SuperTML + RandElastic	0.624 ± 0.032	0.683 ± 0.057
SuperTML + RandGauss	0.656 ± 0.064	$\mathbf{0.760} \pm 0.093$
SuperTML + RandErasing	0.640 ± 0.056	0.706 ± 0.136
SuperTML + CellDropout	0.664 ± 0.047	0.712 ± 0.122
SuperTML + CoarseDrop	0.656 ± 0.096	0.744 ± 0.143
SuperTML + CoarseShuffle	0.632 ± 0.058	0.710 ± 0.133

Table 3.7: *ACC and AUC scores for Colorectal dataset. The reported numbers are the average and the standard deviation metrics over the 5 test splits.*

Cirrhosis		
Method	**ACC**	**AUC**
DeepMicro	0.830 ± 0.029	0.888 ± 0.011
MLP	0.702 ± 0.046	0.803 ± 0.039
SuperTML	0.812 ± 0.047	0.910 ± 0.020
SuperTML + RandRotate	0.782 ± 0.075	0.896 ± 0.019
SuperTML + RandFlip	0.846 ± 0.028	0.909 ± 0.012
SuperTML + RandZoom	0.808 ± 0.052	0.907 ± 0.014
SuperTML + RandElastic	0.821 ± 0.082	0.905 ± 0.028
SuperTML + RandGauss	0.812 ± 0.119	0.903 ± 0.046
SuperTML + RandErasing	0.834 ± 0.015	0.907 ± 0.013
SuperTML + CellDropout	0.821 ± 0.037	0.910 ± 0.016
SuperTML + CoarseDrop	0.829 ± 0.044	0.912 ± 0.014
SuperTML + CoarseShuffle	0.838 ± 0.039	$\mathbf{0.914} \pm 0.012$

Table 3.8: *ACC and AUC scores for Cirrhosis dataset. The reported numbers are the average and the standard deviation metrics over the 5 test splits.*

3.5 Concluding Remarks

In this chapter, we introduced and discussed the application of SuperTML to microbiome-based disease prediction, a task that is often limited by the small size and high dimensionality of biomedical datasets. Our analysis shows that SuperTML consistently outperformed standard feedforward neural networks in five out of six datasets, confirming its effectiveness in capturing complex patterns in microbiome profiles. When compared with DeepMicro, SuperTML with image augmentation achieved the highest AUC scores in five out of six datasets, showing competitive performance against state-of-the-art approaches. The results also confirmed that image augmentation acts as an effective regularization strategy for SuperTML, though no single transformation proved consistently superior across all datasets. This leaves open questions about which augmentations work best and how SuperTML internally handles them.

Finally, as directions for future work, one possibility is to explore ways to extend SuperTML to very high-dimensional data, such as strain-level or multi-omics profiles, by introducing a dimensionality reduction step before embedding. Another important direction is to work on interpretability, for example by adapting computer vision methods to trace back the most relevant pixels to the original microbiome features. This could make the predictions more understandable for researchers and practitioners in healthcare, and represent a step forward toward applications in precision medicine.

The datasets and the source code are available on GitHub at `https://github.com/gabrieletaz/microbiome_supertml`

The author of this PhD thesis is responsible for the following contributions presented in this chapter:

II/1. Contributed to conceptualization and design of the work: transforming the microbiome-disease classification tasks into image classification using SuperTML and comparison with Deepmicro.

II/2. Literature survey on SuperTML and image augmentation.

II/3. Conceptualization and implementation of all the experiments reported in the chapter.

II/4. Conceptualization and implementation of the novel CellDropout transformation from scratch.

Chapter 4

Supervised Multiple Kernel Learning approaches for multi-omics data integration

Advances in high-throughput technologies have originated an increasing availability of omics datasets. The integration of multiple heterogeneous data sources is currently an issue for biology and bioinformatics. Multiple kernel learning (MKL) has shown to be a flexible and valid approach to consider the diverse nature of multi-omics inputs, despite being an underused tool in genomic data mining. In this context, we present our work "Supervised multiple kernel learning approaches for multi-omics data integration" published in "BMC BioData Mining" [21] where we introduce novel MKL approaches based on different kernel fusion strategies. To learn from the meta-kernel of input kernels, we adapted unsupervised integration algorithms for supervised tasks with support vector machines. We also present deep learning architectures for kernel fusion and classification. The results show that MKL-based models can outperform more complex, state-of-the-art, supervised multi-omics integrative approaches. After the Introduction 5.1, this chapter is structured as follows. The Related Work section 4.2, provides a literature survey on the multiple kernel learning approaches, introducing the theoretical framework and explaining why MKL framework is a flexible tool for integrating heterogeneous data sources. In addition, it provides a background of the relevant deep learning approaches used for multi-omics integration. The Material and methods section 4.3, presents the different kernel-based models evaluated in this work, including both standard and DeepMKL architectures, along with the dataset and the preprocessing steps used for benchmarking. The Performance evaluation section 4.3.3, describes the evaluation pipeline and evaluation metrics. Results section 4.4 presents the performance and the biomarkers found using the proposed methods. Finally, in the Discussion and concluding remarks section 4.5 we give a brief discussion of the Chapter main findings and how they fit in the context of multi-omics integration literature. We

also discuss potential future work directions and list the detailed contributions of the author to this chapter.

4.1 Introduction

Data integration has recently attracted substantial attention in the research literature, both for the statistical challenges and promising potential applications in fields such as biology and medicine. Multi-omics data have become increasingly available following the significant growth of high-throughput technologies. The availability of such rich while complex data has expanded the number of available algorithms and methodologies to properly conduct analyses, with the possible need to create novel research profiles [52]. In this context, Kernel methods have proven to be a very promising technique for integrating and analyzing high-throughput technologies-generated data. Kernel methods benefit from the possibility of providing a nonlinear version of any linear algorithm that relies solely on dot products. For instance, unsupervised methods such as Kernel Principal Component Analysis [147], Kernel Canonical Correlation Analysis [11], Kernel Discriminant Analysis [142] and Kernel Clustering [51] are all examples of nonlinear algorithms enabled by the so-called kernel trick.
Kernel-based methods also include supervised classification algorithms. Support vector machine is the most popular one, along with Kernel partial least squared regression [141] or Kernel discriminant analysis [142].
Several methodologies are also available to integrate multiple high throughput data sources through the so-called Multiple kernel learning (MKL) approach. These methods combine modern optimization techniques' power with kernel methods' framework, providing a new multi-source genomic data learning tool.
In this work, we review classical MKL algorithms, while also exploring alternative MKL approaches. Specifically, we propose a novel approach that consists of adapting unsupervised algorithms for multiple kernel integration to a supervised context, i.e., fitting an SVM classification model on a fused kernel obtained through an unsupervised algorithm for the convex linear combination of input kernels. This approach mimics what more recent deep learning-based methods realize using Autoencoders [188]. First, the lower dimensional latent representation is learned in an unsupervised way by an Autoencoder, and then this embedding is used to perform a downstream task such as classification [192].
More recently, Deep learning has emerged as a valid alternative to dealing with data integration challenges. A key strength of deep learning lies in its ability to learn homogeneous representations from heterogeneous data sources (images, text, tabular data), making it a perfect candidate for multi-omics integration problems.
Different deep learning methods have already been applied in this domain with promising results. Architectures such as Autoencoders [188], [197], Graph Neural Networks

[80] [183] or Multi-head Attention [55] have been successfully adapted to different multi-omics integration tasks reaching the state-of-the-art. Deep learning has also been used as an alternative approach to multiple kernel fusion [157] to integrate different kernels from a single data source. This type of architecture can be easily adapted to integrate heterogeneous data sources, such as multi-omics datasets. With this intention, we introduce a novel deep learning framework tailored for Multiple kernel learning (MKL), namely DeepMKL, specifically within multi-omics integration. This method exploits both the advantages of kernel learning and deep learning by transforming the input omics using different kernel functions and guiding their integration in a supervised way, optimizing the neural network weights to minimize the classification error. To sum up, while Multiple kernel learning remains an under-utilized tool for genomic data mining [186], in this work, we propose MKL methods to integrate multi-omics data based both on unsupervised convex linear optimization and deep learning. We aim to show the advantages of this setting by comparing it with state-of-the-art methods. Our results align with recent findings in [23], where the authors compare traditional machine learning (ML) models with Graph Neural Networks (GNNs) in single omics analysis, concluding that the benefits of GNNs are overstated. We similarly demonstrate that classical ML approaches, such as MKL methods, show competitive results against GNNs in the context of multi-omics analysis.

4.2 Related Works

Many machine learning methods are available to unravel biological system mechanisms and find new biomarkers. The big challenges associated with multi-omics data mining and integration are the intrinsic high dimensionality, heterogeneity and nonlinearity of the sample space. For this reason, refined methods are needed to give practitioners new direction and solutions for analyzing such complex datasets. Numerous integration strategies are available in the literature, including early, mixed, hierarchical, intermediate and late integration. In this work, we focus on the mixed integration type, which has demonstrated to ensure great adaptability for omics data fusion as reviewed in [132].

Early stage integration, the easiest and fastest procedure available, nonetheless poses intrinsic drawbacks. More specifically, since early integration is based on the concatenation of the original data, it naturally increases the input dimensionality while giving more importance to omics with a bigger number of features. Moreover, while being extremely easy and fast to realize, this practice tends to mislead learning algorithms as it does not consider the specific data distribution of each input dataset.

On the contrary, mixed integration allows ML algorithms to conduct the learning phase on more refined and less dimensional datasets. As these methods produce new versions of the input datasets which are more homogeneous than original versions, it facilitates

ML algorithms to operate on a unified single input for learning.
Furthermore, another very popular strategy is late integration, which consists of applying each machine learning model separately on each input dataset and then of combining their respective predictions in a later stage. However, as claimed in [132], this approach may not be relevant for biological applications. Indeed, an integration based solely on the combination of different model predictions cannot be compared to a procedure that directly considers complementary information among different omics, as it can be seen as a multiple single-omics analyses.
In the present work, we will investigate mixed integration techniques for multi-omics data integration in comparison to the state-of-the-art method i.e. MOGONET in [183], a late integration methodology based on GNNs.

4.2.1 Mixed integration

It is generally accepted that a classification model trained with information obtained from different sources leads to a more comprehensive overview of the problem [72], [30].
In the field of omics sciences, when different data obtained on the same individuals are available, the integrated analysis can provide richer information about the biological system compared to the results achieved using a single layer of information. New achievements have been reached in a wide area of research, for instance allowing the identification of molecular signatures of human breast tumours [117] or for microbial communities profiling [60].
Each omic dataset contains a different aspect of the mechanisms regulating a biological phenotype. In addition, the technologies used to collect them differ. Consequently, the nature and structure of those data are usually very diverse, generating a remarkably heterogeneous framework. Mixed integration or transformation-based strategies undertake the flaws of concatenation-based approaches applying ML algorithm to a simpler representation of each input dataset. The original omics are transformed separately to obtain a clearer, richer and lower in dimensions version. Standard transformation methods that can be used are kernel-based, graph-based, and deep learning methods.
In this work we will focus our attention on kernel-based integration and on deep learning-based methods applied on kernel learning.

4.2.2 Multiple kernel learning

Kernel methods have been shown to offer an elegant and natural mathematical solution to address data integration from heterogeneous sources, as using kernels enables the representation of the datasets in terms of pairwise similarities between sample points [205], [184]. Given a dataset of n observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ with $\boldsymbol{x}_i \in \mathbb{R}^p$, a function k defined as k: $\mathbb{R}^p \times \mathbb{R}^p \longrightarrow \mathbb{R}$ is a valid kernel if it is symmetric and positive semi-

definite i.e. $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$ and $c^T \boldsymbol{K} \mathrm{c} \geqslant 0$, $\forall c \in \mathbb{R}^n$, where $\boldsymbol{K}$ is the $n \times n$ kernel matrix containing all the data pairwise similarities $\boldsymbol{K} = k(x_i, x_j)$.
Every kernel function is associated with an implicit function ϕ: $\mathbb{R}^p \longrightarrow \mathcal{H}$ which maps the input points into a generic feature space $\mathcal{H}$, with possibly an infinite dimensionality, with the expression $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$. This relation allows the implicitly computation of the dot products in the feature space by applying the kernel function to the input objects, without explicitly computing the mapping function ϕ [148].

It is generally accepted that the sample space of many research problems, such as omics data, is often nonlinear [140]. This nonlinearity is linked also to the incomplete understanding, for instance, of gene interactions and biological pathways, which suggests that genes are not connected in a simple linear way. In this context, kernel methods offer a natural and not computationally expensive approach to kernelized i.e. obtain nonlinear version of any algorithm purely based on dot-product calculations. Indeed, by replacing the linear dot product in the input space by the kernel pairwise values, it is possible to implicitly obtain the value of the dot product as it was computed directly in the feature space. This is the so-called *kernel trick*, which allows algorithms designed initially for linear data to be extended to nonlinear frameworks by implicitly mapping the input points into high-dimensional feature spaces induced by the kernel.

In the context of multi omics integration, given different datasets based on the same n observations, kernel methods provide another advantage, namely they allow to represent every original dataset with a $n \times n$ kernel matrix $\boldsymbol{K}$. So, even if the original data types are heterogeneous (counts, factors, continuous data, networks, images), after the kernel transformation, all the M input datasets will have the form of a $n \times n$ matrix with real numbers as entries, with M equal to the number of available omic datasets. Moreover, the meta-kernel obtained from the combination of the M input kernels is a global similarity matrix containing the sample's similarities based on the original datasets' variables. MKL assures great adaptability as many kernel functions are available, such as linear, Gaussian, polynomial, or sigmoid. In this way it is possible to choose and to apply a specific kernel function on a certain omic input, as each function may be more suitable for a specific omic.
The most common approach in Multiple kernel learning is to compute a convex linear combination of kernel Gram matrices. Analytically, given M different datasets, MKL consists of the linear combination of the M kernel matrices, as in

$$K^* = \sum_{m=1}^{M} \beta_m K^m, \tag{4.1}$$

with $\beta_m \neq 0$ and $\sum_{m=1}^{M} \beta_m = 1$.

It directly follows that the simplest solution is to fix all the weights to be equal, i.e. to $\frac{1}{M}$. Of course, this setting does not allow us to benefit from the adaptability of the multiple kernel framework. All kernels will contribute equally to the classifier, not

taking into account possible redundant or less informative sources of information. The experiment section will denote this setting as **MKL-naive**.

Contrarily, the β_m weights can be optimized more appropriately. Usually, in supervised learning, they are tuned, minimizing the prediction error. The literature offers many algorithms for supervised MKL optimization. For instance, in the work in [88], the weights are optimized with semidefinite programming techniques. The fused kernel is then used to train an SVM classifier, giving better performances than single omic analysis.
Another approach can be found in [138] where the convex linear combination is obtained through a weighted 2-norm regularization constrained formulation to promote a sparse kernel combination and using a subgradient descent for weights optimization. The so-called **SimpleMKL** method is available in the R package *RMKL* developed by [186].
The *RMKL* package proposes several other algorithms such as **SEMKL**, Simple and Efficient MKL by [193] where the weights computation is based on the equivalence between group-lasso and MKL. Both SimpleMKL and SEMKL belong to the class of algorithms known as wrapper methods for Multiple kernel learning, thus updating kernel weights after each iteration.
A more sophisticated version of these wrapper methods specialized in the reduction of the number of SVM computations is **SpicyMKL** in [168], which is a proximal minimization method that converges super-linearly. This algorithm is also implemented in the *RMKL* package under the name of DALMKL.
A different way to find the kernel coefficients in the convex linear combination of kernels can be found in [194] with **GA-fKPLS**, where the authors propose to compute the kernel parameters and weights using genetic algorithms.
A different approach to MKL is presented in [53] and [54], where the authors question the practice of assigning the same weight to a kernel over the whole input space. In this work, they propose a localized Multiple kernel learning **LMKL** based on the local selection of the appropriate kernel function, allowing to reduce the number of support vectors.
To be noted that these wrappers methods have been recently tested in [186], where it has been shown that all these algorithms seem to have similar performance in the case of an analysis with few kernels.
Multiple kernel learning can also be used in the unsupervised learning framework. In this context, selecting appropriate criteria for weight optimization is less straightforward, as it cannot be based on a target variable of interest. In other words, as it is natural to optimize the weights through the minimization of the prediction error for supervised learning, the same does not apply in an unsupervised context. Hence, the algorithms available to effectively determine a strategy to guide the fusion process of the input kernels in an unsupervised framework are less numerous than in the supervised

literature In [109], the authors proposed **STATIS-UMKL**, a methodology to provide an approach to reach a consensus kernel based on the resemblance of the different kernels. Specifically, the meta kernel is defined by maximizing the average similarity between kernels, measured using their cosines according to the Frobenius dot product. The similarity matrix between two kernels $C = (C_{mm'})_{m,m'=1,\dots,M}$ gives insight into how the different kernels relate to each other, revealing whether they complement or provide distinct information. This matrix can then be used to derive the meta kernel K^*, which maximizes the overall similarity with all other kernels in the set.

We have previously introduced how kernels enable us to map data into a higher-dimensional feature space without explicitly computing that space. In this new space, data that are not linearly separable in the original input space may become linearly separable, making it easier to apply linear classification techniques. While kernel methods offer this advantage of making previously nonlinearly separable data linearly separable, this benefit comes with a trade-off. The original features are no longer explicitly accessible after the kernel transformation, as the data is represented through similarities in a new feature space. Consequently, this makes interpreting the model more challenging, as it becomes difficult to directly trace back the role of individual features in the transformed space to the original input variables without referring to a label. In this context in [109], the authors proposed a method based on kernel PCA and random permutation to evaluate the importance of the original variables. Specifically the idea consists in recomputing the K^m kernels after the permutation of all the values of the samples for a given measure j, obtaining a new kernel $\tilde{K}^{m,j}$. The *Crone-Corsby* distances of kernel matrices are then computed to assess which variables lead to the most significant differences between the original kernel and the new kernels $\tilde{K}^{m,j}$. Also, in [19], the authors proposed *KPCA-IG*, an approach which provides a data-driven feature importance, where the influence of each original variable can be computed in the space of the kernel principal components as in the standard PCA. This method offers a computationally fast feature ranking methodology to identify the most relevant original variables, solely based on partial derivative of the kernel function.

4.2.3 Deep Learning approaches

Deep learning techniques are increasingly being employed in the context of multi-omics data analysis. One of the advantage of deep learning is its capacity to learn homogeneous representations from different input sources. In particular, multi-modal architectures allow the use of heterogeneous datasets, such as images, tabular data, time series, or graphs, to learn the underlying complex relationships among different aspects of a biological phenotype.
As reviewed in [159], this kind of architecture is gaining popularity in the biomedical field, where data are becoming increasingly multi-modal. Recently, in this context, different works introduced approaches based on multi-modal deep learning to deal with

different types of omics data, these multi-modal architectures are suited for both Mixed and Late integration strategies. As introduced, we will concentrate on Mixed integration approaches compared to the Late integration methods that can be regarded as the state-of-the-art for the datasets of interest in our analysis [183] [55] [63].
One of the most commonly used deep learning methods for Mixed integration strategies is Autoencoder. Autoencoder is an unsupervised deep learning method used to learn a latent representation of the data by minimizing the reconstruction error between the input and the reconstructed output. In the context of Mixed integration, they can be easily used to learn independent homogeneous latent representations to integrate them in a final shared layer [192]. Autoencoders can also be used to learn latent representations that depend on different omics inputs, as in [188]. In this case, the approach uses Autoencoders in two different steps, first as a pre-processing for the two different inputs and then as an integration step, part of the learning process. Other possible approaches for Mixed integration involve the use of feedforward neural networks. In particular, in [101], the authors built an architecture based on different encoding subnetworks to learn homogeneous representations from the different types of omics data, then a fusion step to create a concatenated representation of multi-omics, and finally, a classification sub-network is used to perform the cancer subtype classification. Alternatively, in [151], a similar architecture equipped with a triplet loss is used for drug response prediction. Despite this, several state-of-the-art methods belong to the Late integration family, such as MOGONET by [183], MOADLN by [55] and Dynamics in [63]. MOGONET transforms the input data into matrices of similarity among observations to build a graph structure and apply a Graph Convolutional Neural Network to each omic to obtain an initial prediction. After this first step, a View Correlation Discovery Network (VCDN) finally combines all the independent predictions to determine the correct label.
MOADLN, instead, uses the Self-attention mechanism to build a similarity network and exploit the correlation between intra-omic observations. In this case, each input instance is an element of a set i.e. a specific observation within a single omics type, and the Self Attention mechanism learns the weights for each of these elements, meaning that it determines the significance of each instance in relation with others within the same omics type. Also, for MOADLN, the first step is the initial independent prediction for each omics type, followed by a final combination through a Multi-Omics Correlation Discovery Network (MOCDN) to explore the cross-omics relations. Dynamics assesses feature-level and modality-level informativeness dynamically across different samples. It incorporates a sparse gating mechanism to capture variations in features information within each omics while using actual class probability to asses the classification confidence at the modality level [63].

4.3 Materials and methods

As considered in the previous section, [183] and [55] claim to be the state-of-the-art in terms of predictive performance.
In this section, we present all the experiments to test different MKL methods, architectures and combinations in order to compare possible solutions for multi-omics data integration.

4.3.1 Datasets

The datasets considered in this work are the publicly available ROSMAP for Alzheimer's Disease classification, BRCA for breast invasive carcinoma PAM50 subtype classification, LGG for grade classification in low-grade glioma and KIPAN for kidney cancer type classification. In order to be sure to conduct a fair comparison with MOGONET, we used the same datasets. [183] performed an initial feature selection obtained through the sequential calculation of an ANOVA F-value on the original data to evaluate whether a feature was significantly different across different classes. Moreover, the authors kept the number of features such that the first principal component after feature pre-selection explains at least 50% of the variance.
In the case of ROSMAP and BRCA, as also [55] proceeded, we conducted the analysis on the preprocessed datasets available in [183] GitHub repository. Instead, for LGG and KIPAN, we downloaded the datasets and performed the same pre-processing steps as in [183] since the author did not provide the preprocessed ones. For each of the 5 datasets three types of omics are considered for classification purposes: mRNA expression (mRNA), DNA methylation (meth), and miRNA expression data (miRNA). Table 4.1 contains all the details for the five datasets.

Dataset	Classes	Number of features mRNA, meth, miRNA	Features for training mRNA, meth, miRNA
ROSMAP	NC: 169, AD: 182	55,889; 23,788; 309	200; 200; 200
BRCA	Normal-like: 115, Basal-like: 131, HER2-enriched: 46, Luminal A: 436, Luminal B: 147	20,531; 20,106; 503	1000; 1000; 503
KIPAN	KICH: 65; KIRC: 345 ; KIRP: 297	60,484; 25,972 ; 1882	2000; 2000; 445
LGG	Grade 2: 257 ; Grade 3: 266	60,484; 25,972 ; 1882	2000; 2000; 548

Table 4.1: *The ROSMAP dataset contains two classes: Alzheimer's disease (AD) patients and normal control (NC). The breast invasive carcinoma dataset (BRCA) contains PAM50 subtype classes: normal-like, basal-like, human epidermal growth factor receptor 2 (HER2)-enriched, Luminal A, and Luminal B. The KIPAN dataset contains different kidney cancer type: chromophobe renal cell carcinoma (KICH), clear renal cell carcinoma (KIRC), and papillary renal cell carcinoma (KIRP). Finally, the LGG dataset is for grade classification in low-grade glioma (LGG).*

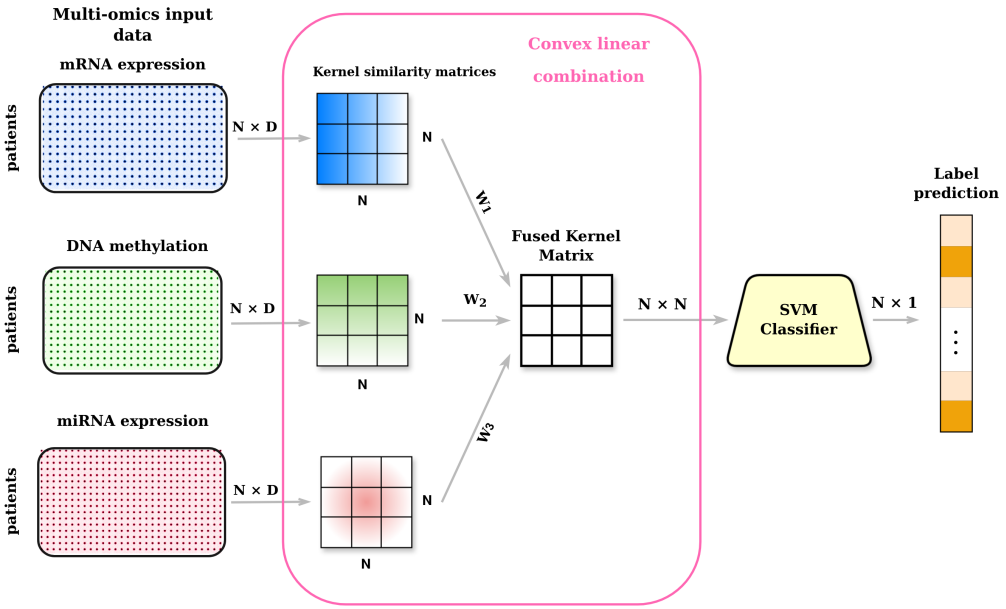

Figure 4.1: *A kernel function is applied on each dataset separately. In MKL, a convex linear combination provides a fused Meta-kernel that summarizes the information of input omics. Then an SVM classifier is used for classification.*

4.3.2 Methods

Both in [183] and [55], the authors compared the performance of their methods, MOGONET and MOADNL, respectively, with other typical classification algorithms such as K-nearest neighbours (KNN), Support vector machine (SVM), LASSO regression and block s(PLSDA) as in DIABLO [153].
Taking SVM as an example, the analysis is applied to the concatenation of the 3 multi-omics datasets, where its performance shows a significantly lower accuracy in both studies. However, as SVM can be viewed as a kernel-based classification algorithm, applying it to an early stage integration, i.e., to a combined dataset obtained by simple concatenation of the input datasets, as we have seen, it can be seen as an oversimplification. Moreover, a proper parameters tuning must be carried out along with the choice of a suitable kernel function. Thus, our analysis compares MOGONET's performance with more suitable and fair usage of Multiple kernel learning with support vector machines.
Moreover, new approaches of Multiple kernel learning in combination with deep learning classification models are presented in order to exploit at the same time the adaptability of kernel methods avoiding the optimization of the weights in the convex linear combination and the classification power of deep architectures.

Multiple kernel learning - SVM

As presented in Section 4.2, there are many optimization algorithms to compute the coefficients of the convex linear combination of input kernel gram matrices in the literature.
For completeness, in this work, we will present the results obtained using **MKL-naive**, **SimpleMKL** and **SEMKL** in the case of binary classification problem and **STATIS-UMKL**.
On the contrary, STATIS-UMKL in [109] is an algorithm to obtain a consensus meta-kernel in an unsupervised framework. To the best of our knowledge, STATIS-UMKL has never been used with support vector machines for classification purposes. However, the peculiarity of this procedure, which aims to take the different specificities of each dataset into account by fusing them into a single meta-kernel, may also enhance classification performance. In Figure 4.1, it is possible to see the network structure for all the SVM algorithms that are used for the experiments. This architecture belongs to the Mixed integration type as the integration of the input omics is preceded by a data transformation, and the SVM algorithm is applied to the convex linear combination of the datasets performed at the feature space.
For completeness, we also trained a support vector machine on the direct concatenation of original datasets (**SVM-concat**) using the same tuning procedure for the hyperparameters used for the other algorithms.

Deep Multiple kernel learning

As introduced previously, employing neural network architectures is another way to combine the input kernel matrices by avoiding the task of convex linear optimization. More specifically, in [157], a deep learning architecture that includes a dense embedding of kernels and a multi-modal neural network is used for fusing multiple kernels.
In our case, we adapted this approach to a multi-omics analysis, meaning that the kernel matrices represent different data sources, i.e different omics, and not different representations of a single data source, as in a classic multiple kernel fusion problem. As shown in Figures 4.2 and 4.3, the structures of the proposed architectures are similar. They consist of a first dense embedding, realized by employing a Kernel PCA for each omic input. After this first step, a multi-modal neural network is used to learn in parallel three representations, one for each dense embedding, and then integrate them to perform the downstream task. In the case of Figure 4.2, we call this architecture **Deep Multiple kernel learning**, i.e. **Deep MKL**, to highlight that it is a Multiple kernel learning method that employs deep learning to combine the different kernels information. From a neural network perspective, the architecture is composed of three fully connected layers for each input, followed by an integration step that can be performed through a concatenation, sum, or weighted sum with learnable parameters

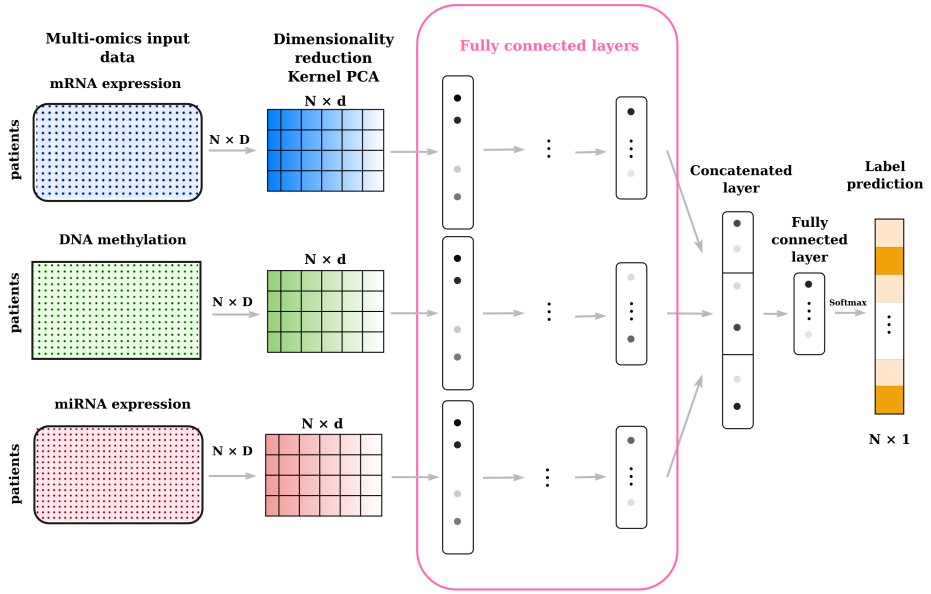

Figure 4.2: *Deep MKL (concat) takes in input the Kernel PCA dense embeddings of different omics datasets. It extracts the features using different feedforward sub-networks and then fuses the learnt representations by concatenating them for the final classification.*

of the three representations. Finally, another two fully connected layers are employed for the final classification step.
In the context of multi-modal architectures, cross-connections between modalities can improve the model's performance, allowing the flow of information between modalities at different learning process levels before the fusion step [16],[208]. In our context, this flow should inform each omic layer with each other, potentially improving the performances. We call the version of Deep MKL employing cross-connections **Cross-modal Deep MKL** in Figure 4.3. The architecture's structure is similar to the Deep MKL one, except that each cross-connection is, in practice, an additional layer followed by a concatenation step, which means that the Cross-modal Deep MKL architecture, w.r.t. Deep MKL's one, has an additional layer before the integration and classification steps. For both methods, each fully connected layer is followed by a Leaky Relu activation function, a Dropout, and batch normalization. Additional details on the architectures and their specific hyperparameters are discussed in Section 4.3.3.

Interpretability

Using a dense embedding such as Kernel PCA as a step of a neural network makes the Deep MKL models even more challenging to interpret than classical deep learning ones. In this framework, the principal components can be considered the input features of the neural network. Using an interpretability method such as SHAP in [104] or Integrated

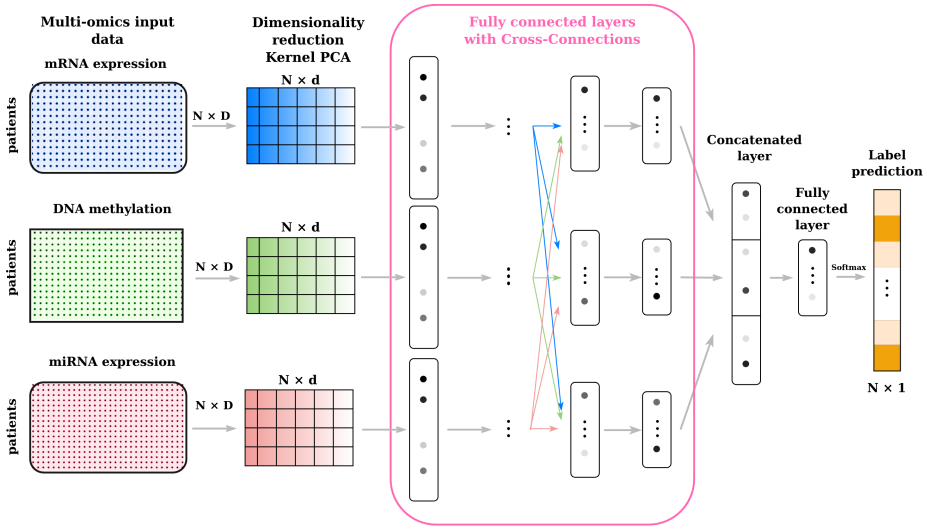

Figure 4.3: *Cross-modal Deep MKL (concat) takes in input the Kernel PCA dense embeddings of different omics datasets. It extracts the features using different feedforward sub-networks that are linked by cross-connections, then fuses the learnt representations by concatenating them for the final classification.*

Gradients [166] to rank the features would be insufficient because, as highlighted in Section 4.2.2, after a kernel transformation, the link between the original features, the genes, and the principal components is lost.
In this case, we propose a novel mitigation strategy for biomarkers discovery based on a two-step approach. First, we compute the rank of the input features, namely the kernel principal components, using Integrated Gradients [166] implemented in the library [116]. Then, we employ the recently published method proposed in [19] to recover the most relevant input variables for the selected principal components. As already introduced in section 4.2.2, KPCA-IG in [19] allows to obtain a data-driven feature ranking based on the selected kPCs, and it is available in the R package *kpcaIG* [20]. To the best of our knowledge KPCA-IG has never been used in combination with a supervised approach such as our proposed DeepMKL, used to select the most important kernel principal components in terms of prediction accuracy. Combining an unsupervised feature selection approach with a supervised learning method like DeepMKL offers a promising strategy for discovering novel biological and medical biomarkers. This hybrid pipeline may provide deeper insights than traditional methods focused solely on prediction performance, such as those that sequentially remove features to rank their importance based on the impact on prediction accuracy, as in [183] and [55]. We will demonstrate the application of this approach for biomarker identification in Section 4.4, highlighting its relevance from a biomedical point of view.

4.3.3 Performance evaluation

Experimental setup

To evaluate the classification performance of the MKL-SVM algorithms and Deep MKL, we implemented the same evaluation pipeline already used by MOGONET in [183] and by MOADLN in [55]. It consists of evaluating the model's performance on 5 random train/test partitions of the dataset. To maintain the balance of class distributions among the partitions, a stratified version of the split is adopted, keeping the ratio of 30/70 % for the train/test splits.
For final evaluation, we present the mean and standard deviation of different performance metrics among the 5 randomly generated training/test splits, with a seed set of $[0, 1, 2, 3, 4]$ for reproducibility purposes.
The seeds used in MOGONET and Dynamics are not publicly available, meaning that the results are not completely reproducible. For this reason, we recomputed all the metrics using their publicly available code and the same seeds of our experiments in order to have a fair comparison. On the contrary, we have not recomputed the metrics for MOADLN as the code is not publicly available.

Methods	Integration	Optimized Parameters	Description
block PLSDA	Mixed	ncomp	DIABLO
block sPLSDA	Mixed	ncomp, keepX	DIABLO
SVM concat	Early	C, σ	Direct concatenation
SVM naive	Mixed	C, σ	Sum of the kernel
SimpleMKL-SVM	Mixed	C, σ	Weighted sum of kernels
SEMKL-SVM	Mixed	C, σ	Weighted sum of kernels
STATIS-UMKL + SVM	Mixed	C, σ	Weighted sum of kernels
Deep MKL	Mixed	σ, epochs, principal components, dropout value	Deep Learning kernel fusion
Cross-Modal Deep MKL	Mixed	σ, epochs, principal components, dropout value	Deep Learning kernel fusion
NN_VCDN	Late	NA	Feedforward neural network
Dynamics	Late	NA	Dynamical Multimodal Fusion
MOGONET	Late	Optimized k	Graph convolutional network

Table 4.2: *Summary and description for all the tested methods with all the tuned hyperparameters*

Hyperparameters tuning

In the context of MKL-SVM, a Grid Search 5-folds cross-validation has been computed on the training sets employing a Gaussian radial basis kernel.

Cross-validation has been used to tune the following parameters:

- C parameter: the cost of constraints violation, the so-called C-constant of the regularization term in the Lagrange formulation of the support vector machine algorithm.
- The sigma parameter: the inverse kernel width for the radial basis kernel function.

For the experiments, the C parameter has been set in the range $[1, 25]$, while the sigma in the range of $[0.005, 0.00005]$ for both datasets.

In the context of our deep learning methods, we employed a Random Search 5-folds cross-validation for the hyperparameters tuning. Also, in this case, all the experiments were carried out using a Gaussian radial basis kernel for the Kernel PCA step. For all the DeepMKL models, we fixed the number of layers and the number of neurons as in MOGONET, i.e. [200, 200, 100] for ROSMAP and [400, 400, 200] for BRCA, LGG, and KIPAN. For all the Cross-modal Deep MKL architectures, as described in the 4.3.2, we implemented cross-connections between modalities, which, in practice, are additional layers. For this reason, we fixed the number of layers and neurons for each dataset as [200, 200, 100, 100] for ROSMAP and [400, 400, 200, 200] for BRCA, LGG, and KIPAN.
In order to have a training process as stable as possible, i.e., a smooth training loss curve, we added a dropout and a batch normalization after each feedforward layer. Additionally, we fixed small values for the learning rate, such as 5×10^{-5} for ROSMAP and KIPAN, 10^{-4} for BRCA, and 10^{-5} for LGG. Regarding the dropout, the intensity is 0.5 for ROSMAP and 0.3 for all the other datasets. Adam classifier [85] and a batch size of 32 are adopted for all the datasets. Regarding the choice of sigma value for the Kernel PCA and the number of principal components to keep, we defined different search spaces for each dataset since the choice of these hyperparameters depends on the topological structure of the data, which varies from dataset to dataset, similar to the k parameters used in MOGONET. In the case of ROSMAP, the sigma value for the Kernel PCA is chosen in the set of $\{0.0005, 0.0007, 0.001\}$. Meanwhile, for BRCA, the set is $\{0.00005, 0.0005, 0.005\}$. For LGG and KIPAN, the set is [0.0005,0.005]. Regarding the number of principal components in ROSMAP, we fixed it to 120. While in BRCA, we defined a search space in the $[2, 20]$ range to choose the optimal combination with the sigma parameter. We adapted the same strategy for LGG and KIPAN using a range of $[50, 200]$.
Since the variability among the different folds made the results unreliable for an early stopping strategy, we chose the number of epochs by defining a range from 100 to 200 with an interval of 10, letting the hyperparameter tuning optimization select the best value in combination with all the other parameters.
For reproducing MOGONET's results, we used the optimized parameter k, as suggested by the authors, namely equal to 2 for ROSMAP and 10 for all the other datasets. This

parameter controls the average number of edges per node of the Adjacency matrix used for training the graph convolutional neural networks.
Finally, for the DIABLO framework we used the 5-fold cross validation procedure to optimize the number of components (ncomp) for both block PLSDA and block sPLSDA, and the number of retained variables (keepX) for the sparse version. For the design matrix, the value of 0.1 has been used to prioritize the discriminative ability of the model, as suggested by the authors.

Metrics

We employed the same metrics used to evaluate state-of-the-art methods in order to have a fair comparison For binary classification, we used accuracy (ACC), F1 score (F1) and area under the curve (AUC).

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.2}$$

with TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

where Precision $= \frac{\text{TP}}{\text{TP+FP}}$ and Recall $= \frac{\text{TP}}{\text{TP+FN}}$.
The F1 score represents the harmonic mean between Precision and Recall and measures how balanced the two metrics are for a classifier. The Precision score measure how accurate the positive predictions are. While, the Recall metric measures how many True Positives are predicted out of the total number of positive samples.
The AUC score, or area under the ROC curve, measures the classifier's performance and its independence from the threshold.

In multi-class classification task, we used the accuracy (ACC), the macro-averaged F1 score (F1-macro) and the F1 score weighted by its support i.e. the number of instances in that class (F1-weighted).
In multi-class classification, the F1 score is calculated for each class in a one-vs-all manner. In the case of F1-macro, the F1 scores are then averaged, considering each class equally, regardless of the imbalance of the class distribution in the data.

$$\text{F1-macro} = \frac{1}{C} \sum_{i=1}^{C} F1_i \tag{4.4}$$

C is the number of classes and $F1_i$ is the F1 score for the class i.

The F1-weighted, instead, takes into account the imbalance of the class distribution in the data, and it is calculated by a weighted average where the weights are the percentage of the instances in one class.

$$\text{F1-weighted} = \frac{1}{C}\sum_{i=1}^{C}\left(\frac{\text{support}_i}{\text{total support}}\right)\cdot F1_i \tag{4.5}$$

where support_i is the number of instances of class i and total support is the total number of instances in the data.

4.4 Results and Discussion

We compared the classification performance of different MKL algorithms with different state-of-the-art methods such as MOGONET and Dynamics, as MOADLN's code is not publicly available. As anticipated, the MOGONET and Dynamics code seeds are unavailable; therefore, we could not replicate the results exactly. Thus, we proceeded with the computation of the metrics for these methods based on the publicly available code and using the same environment and seed selection of our experiments.

Regarding Deep MKL models, here we reported the results for only one integration mode, namely *weighted sum*. However, the detailed comparison between different integration modes is provided in the Additional Results section 4.4.1.
For BRCA in Table 4.8, all the MKL algorithms achieved the highest performances for all the metrics. Regarding KIPAN, as shown in Table 4.10, the MKL algorithms

	BRCA		
Algorithm	ACC	F1_weighted	F1_macro
block PLSDA	0.670 ± 0.016	0.726 ± 0.009	0.702 ± 0.011
block sPLSDA	0.668 ± 0.021	0.725 ± 0.012	0.708 ± 0.009
SVM concat	0.793 ± 0.018	0.800 ± 0.016	0.776 ± 0.017
SVM naive	0.838 ± 0.008	0.849 ± 0.008	0.828 ± 0.011
STATIS-UMKL + SVM	$\mathbf{0.846 \pm 0.011}$	$\mathbf{0.858 \pm 0.010}$	$\mathbf{0.837 \pm 0.018}$
Deep MKL (weighted sum)	0.827 ± 0.014	0.803 ± 0.015	0.831 ± 0.013
Cross-Modal Deep MKL (weighted sum)	0.829 ± 0.017	0.802 ± 0.022	0.834 ± 0.015
NN_VCDN	0.700 ± 0.018	0.692 ± 0.019	0.609 ± 0.014
Dynamics	0.826 ± 0.010	0.829 ± 0.010	0.793 ± 0.020
MOGONET	0.736 ± 0.038	0.726 ± 0.041	0.650 ± 0.053

Table 4.3: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on BRCA dataset.*

	ROSMAP		
Algorithm	ACC	AUC	F1
block PLSDA	0.666 ± 0.025	0.689 ± 0.034	0.658 ± 0.031
block sPLSDA	0.671 ± 0.027	0.705 ± 0.033	0.665 ± 0.017
SVM concat	0.765 ± 0.019	0.863 ± 0.044	0.763 ± 0.015
SVM naive	0.790 ± 0.006	$\mathbf{0.881 \pm 0.010}$	0.778 ± 0.018
SimpleMKL-SVM	0.758 ± 0.019	0.860 ± 0.021	0.748 ± 0.012
SEMKL-SVM	0.775 ± 0.039	0.869 ± 0.035	0.763 ± 0.037
STATIS-UMKL + SVM	0.784 ± 0.038	0.878 ± 0.019	0.772 ± 0.039
Deep MKL (weighted sum)	0.715 ± 0.028	0.800 ± 0.021	0.721 ± 0.027
Cross-Modal Deep MKL (weighted sum)	0.730 ± 0.025	0.802 ± 0.020	0.746 ± 0.039
NN_VCDN	$\mathbf{0.794 \pm 0.030}$	0.874 ± 0.024	0.807 ± 0.036
Dynamics	0.764 ± 0.026	0.870 ± 0.011	0.771 ± 0.031
MOGONET	0.787 ± 0.027	0.878 ± 0.021	$\mathbf{0.791 \pm 0.045}$

Table 4.4: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on ROSMAP dataset.*

obtained the best results comparable with Dynamics. Also for LGG, the MKL approaches show the best accuracy, where the optimized SVM-concat achieved the best results. On the other hand, for ROSMAP in Table 4.7, a similar trend can be seen for SVM-based approaches that show comparable accuracy with MOGONET, NN_VCDN and Dynamics, while Deep MKL algorithms perform worse than all the other methods. Thus, it can be seen that, kernel-based methods are consistently comparable and even outperformed state-of-the-art methods on all four datasets for all the computed performance metrics, Tables 4.8-4.7-4.9-4.10.

These results again show the kernel framework's advantages in genomics data mining, where even the results obtained with an SVM trained on the direct concatenation of the input datasets, SVM-concat, exhibits a relatively good performance, especially on ROSMAP and LGG, the smallest datasets. In [183], the performances obtained with SVM-concat are lower, suggesting that even a simple procedure such as early integration followed by proper parameter tuning and an appropriate kernel choice of the SVM may already give a good model alternative for certain datasets. Methods such as SEMKL and STATIS-UMKL, which aim to optimize the input kernel matrices' convex linear combination, showed high performances in most of the different metrics. It should be noted that the MKL with equal weights in SVM-naive showed the best performance in the ROSMAP dataset, indicating that the datasets are probably similarly informative in this context. For this dataset, the second best was STATIS-UMKL + SVM, where the mean over 5 runs of the 3 weights of the convex linear combination of kernel matrices of $0.361, 0.308, 0.331$ suggests that the 3 omics are equally important.

	LGG		
Algorithm	ACC	AUC	F1
block PLSDA	0.651 ± 0.024	0.713 ± 0.034	0.677 ± 0.029
block sPLSDA	0.637 ± 0.030	0.771 ± 0.039	0.692 ± 0.027
SVM concat	$\mathbf{0.723 \pm 0.030}$	$\mathbf{0.781 \pm 0.024}$	$\mathbf{0.741 \pm 0.032}$
SVM naive	0.709 ± 0.011	0.774 ± 0.024	0.724 ± 0.022
SimpleMKL-SVM	0.684 ± 0.011	0.759 ± 0.024	0.710 ± 0.020
SEMKL-SVM	0.691 ± 0.011	0.762 ± 0.028	0.719 ± 0.017
STATIS-UMKL + SVM	0.709 ± 0.009	0.774 ± 0.023	0.728 ± 0.015
Deep MKL (weighted sum)	0.687 ± 0.011	0.765 ± 0.025	0.684 ± 0.031
Cross-Modal Deep MKL (weighted sum)	0.700 ± 0.020	0.768 ± 0.026	0.695 ± 0.032
NN_VCDN	0.703 ± 0.036	0.754 ± 0.030	0.715 ± 0.028
Dynamics	0.707 ± 0.029	0.769 ± 0.027	0.714 ± 0.023
MOGONET	0.669 ± 0.026	0.711 ± 0.026	0.69 ± 0.032

Table 4.5: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on LGG dataset.*

	KIPAN		
Algorithm	ACC	F1_weighted	F1_macro
block PLSDA	0.882 ± 0.013	0.884 ± 0.013	0.871 ± 0.016
block sPLSDA	0.896 ± 0.012	0.898 ± 0.011	0.891 ± 0.017
SVM concat	0.953 ± 0.010	0.954 ± 0.009	0.949 ± 0.020
SVM naive	0.958 ± 0.010	0.959 ± 0.009	0.953 ± 0.018
STATIS-UMKL + SVM	0.959 ± 0.010	$\mathbf{0.960 \pm 0.010}$	0.955 ± 0.017
Deep MKL (weighted sum)	0.958 ± 0.011	0.954 ± 0.018	0.958 ± 0.011
Cross-Modal Deep MKL (weighted sum)	0.958 ± 0.009	0.952 ± 0.014	$\mathbf{0.958 \pm 0.009}$
NN_VCDN	0.957 ± 0.006	0.957 ± 0.006	0.952 ± 0.015
Dynamics	$\mathbf{0.960 \pm 0.011}$	0.960 ± 0.010	0.951 ± 0.022
MOGONET	0.940 ± 0.023	0.932 ± 0.032	0.941 ± 0.023

Table 4.6: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on KIPAN dataset.*

As expected, the two wrapper methods optimized for supervised multiple kernel learningn namely, SimpleMKL and SEMKL seem to have similar performance as already shown in [186]. On the other DIABLO linear approaches showed lower performances, proving the need of nonlinear based approaches in the context of complex omics datasets. The Deep MKL approach to integrating multiple kernels shows results comparable with the STATIS-UMKL + SVM method for the BRCA, LGG, and KIPAN datasets. In the case of the ROSMAP dataset, it performs worse than all the methods

based on SVM. The difference in performance can be largely attributed to the dataset sizes. This phenomenon is consistent with established understanding that deep learning models tend to underperform in scenarios involving smaller datasets [18].
Cross-connections, which were expected to improve the predictions as they ensure more layers of integration between different omics, show no consistent improvement w.r.t. the simpler Deep MKL architecture.

4.4.1 Additional Results

Integration modes

In this section we present the results of the comparison between the different integration modes of the deep learning architectures.

	ROSMAP		
Algorithm	ACC	AUC	F1
Deep MKL (concat)	$\mathbf{0.747 \pm 0.018}$	0.810 ± 0.021	0.762 ± 0.014
Deep MKL (sum)	0.745 ± 0.020	0.805 ± 0.020	0.762 ± 0.014
Deep MKL (weighted sum)	0.715 ± 0.028	0.800 ± 0.021	0.721 ± 0.027
Cross-Modal Deep MKL (concat)	$\mathbf{0.732 \pm 0.020}$	0.808 ± 0.018	0.751 ± 0.025
Cross-Modal Deep MKL (sum)	0.726 ± 0.021	0.809 ± 0.018	0.739 ± 0.043
Cross-Modal Deep MKL (weighted sum)	0.730 ± 0.025	0.802 ± 0.020	0.746 ± 0.039

Table 4.7: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on ROSMAP dataset.*

	BRCA		
Algorithm	ACC	F1_weighted	F1_macro
Deep MKL (concat)	0.835 ± 0.016	0.801 ± 0.021	0.840 ± 0.020
Deep MKL (sum)	$\mathbf{0.836 \pm 0.029}$	0.812 ± 0.036	0.842 ± 0.029
Deep MKL (weighted sum)	0.827 ± 0.014	0.803 ± 0.015	0.831 ± 0.013
Cross-Modal Deep MKL (concat)	$\mathbf{0.828 \pm 0.015}$	0.802 ± 0.018	0.832 ± 0.021
Cross-Modal Deep MKL (sum)	0.822 ± 0.027	0.786 ± 0.037	0.824 ± 0.030
Cross-Modal Deep MKL (weighted sum)	0.829 ± 0.017	0.802 ± 0.022	0.834 ± 0.015

Table 4.8: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on BRCA dataset.*

	LGG		
Algorithm	ACC	AUC	F1
Deep MKL (concat)	0.680 ± 0.028	0.763 ± 0.025	0.688 ± 0.019
Deep MKL (sum)	0.680 ± 0.018	0.770 ± 0.015	0.683 ± 0.024
Deep MKL (weighted sum)	$\mathbf{0.687 \pm 0.011}$	0.765 ± 0.025	0.684 ± 0.031
Cross-Modal Deep MKL (concat)	0.693 ± 0.012	0.758 ± 0.024	0.678 ± 0.023
Cross-Modal Deep MKL (sum)	0.695 ± 0.022	0.763 ± 0.023	0.678 ± 0.028
Cross-Modal Deep MKL (weighted sum)	$\mathbf{0.700 \pm 0.020}$	0.768 ± 0.026	0.695 ± 0.032

Table 4.9: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on LGG dataset.*

	KIPAN		
Algorithm	ACC	F1_weighted	F1_macro
Deep MKL (concat)	0.951 ± 0.010	0.945 ± 0.018	0.951 ± 0.0101
Deep MKL (sum)	0.956 ± 0.008	0.950 ± 0.019	0.956 ± 0.008
Deep MKL (weighted sum)	$\mathbf{0.958 \pm 0.011}$	0.954 ± 0.018	0.958 ± 0.011
Cross-Modal Deep MKL (concat)	0.957 ± 0.010	0.948 ± 0.021	0.957 ± 0.010
Cross-Modal Deep MKL (sum)	0.956 ± 0.009	0.950 ± 0.019	0.956 ± 0.009
Cross-Modal Deep MKL (weighted sum)	$\mathbf{0.958 \pm 0.009}$	0.952 ± 0.014	0.958 ± 0.009

Table 4.10: *Metrics average and standard deviation over 5 random test splits for the performance evaluation on KIPAN dataset.*

DeepMKL configurations

This section presents the results of a comparative analysis between different DeepMKL configurations. Specifically, we want to explore the effect of the neural network architecture's depth and width on the classification performance. We conducted these experiments using DeepMKL (weighted sum) on the BRCA and ROSMAP datasets. We started with the same configuration choices, i.e. [200,200,100] for ROSMAP and [400,400,200] for BRCA, used in MOGONET, which are also the ones reported in the Results and Discussion section of the Chapter. Then, we explored different configurations for depth and width.
As shown in Tables 4.11, 4.12, we tested three configurations for DeepMKL with two, three, and four layers. For each of these DeepMKL architectures, we tested three configurations with different numbers of layers, doubling and halving the number of neurons w.r.t our baseline. The results show that for the BRCA, DeepMKL is robust w.r.t differences in depth and width. For ROSMAP, the effect of varying the number of neurons is more clear. The three configurations with fewer neurons have worse performances in the case of DeepMKL with two, three, and four layers. While the ones with the larges number of neurons obtain the best performances. Similarly to the BRCA case, the DeepMKL for ROSMAP architecture seems robust w.r.t. the variation in the number of layers.

4.4.2 Biomarker discovery

We previously introduced the approach for biomarkers discovery employing a hybrid 2-step approach for the Deep MKL algorithm. First, the most relevant features, i.e.,

	BRCA		
DeepMKL Configuration	ACC	F1_weighted	F1_macro
[200, 100]	0.835 ± 0.020	0.841 ± 0.021	0.813 ± 0.024
[400, 200]	0.837 ± 0.016	0.843 ± 0.016	0.813 ± 0.022
[800, 400]	0.826 ± 0.028	0.831 ± 0.028	0.804 ± 0.030
[200, 200, 100]	0.833 ± 0.018	0.838 ± 0.019	0.808 ± 0.026
[400, 400, 200] (baseline)	0.827 ± 0.014	0.831 ± 0.013	0.803 ± 0.015
[800, 800, 400]	0.832 ± 0.027	0.838 ± 0.028	0.811 ± 0.028
[200, 200, 200, 100]	0.834 ± 0.016	0.838 ± 0.016	0.810 ± 0.021
[400, 400, 400, 200]	0.842 ± 0.019	0.849 ± 0.018	0.823 ± 0.014
[800, 800, 800, 400]	0.831 ± 0.025	0.837 ± 0.024	0.808 ± 0.024

Table 4.11: *Comparative study for different width and depth of the architecture - BRCA dataset.*

kernel principal components, are selected using Integrated Gradients [166] and subsequently KPCA-IG as in [19] is applied, obtaining a data-driven feature importance based on the kernel PCA representation of the data. The optimal tuned σ parameters adopted in the Deep MKL model are also used to run the KPCA-IG method. The most important biomarkers can be found in Tables 4.13 and 4.14.

For BRCA dataset the most important components are $[1, 2, 3]$, $[2, 1, 3]$ and $[2, 1, 4]$ for mRNA, meth and miRNA respectively. As the mRNA influence on the final prediction appeared to be more prominent, we included the first 15 most relevant genes, while we showed the first 10 for the DNA methylation and miRNA datasets. Same procedure is applied to the ROSMAP dataset where the most relevant components are $[1, 2, 21]$, $[1, 2, 3]$ and $[1, 4, 9]$ for the three datasets respectively. For the mRNA expression genes and those inferred from high-ranking DNA methylation features, we conducted gene set functional enrichment analysis using the ToppGene Suite [27] to assess the biological significance of genes identified by Deep MKL, highlighting biological annotations such as Gene Ontology (GO) terms that are significantly enriched in a specific set of genes. To correct for multiple comparisons and control the false discovery rate (FDR), the Benjamini–Hochberg procedure is employed, reporting the adjusted p-values.

For BRCA PAM50 subtype classification datasets, several of the 15 selected genes from the mRNA expression dataset were included in GO terms linked with breast cancer such as β-alanine transmembrane transporter activity (GO:0001761, $p = 2.324E - 2$),

	ROSMAP		
DeepMKL Configuration	ACC	AUC	F1
$[100, 50]$	0.689 ± 0.031	0.768 ± 0.028	0.697 ± 0.033
$[200, 100]$	0.717 ± 0.015	0.801 ± 0.021	0.719 ± 0.023
$[400, 200]$	0.736 ± 0.010	0.805 ± 0.015	0.746 ± 0.014
$[100, 100, 50]$	0.704 ± 0.022	0.774 ± 0.018	0.687 ± 0.038
$[200, 200, 100]$ (baseline)	0.715 ± 0.028	0.800 ± 0.021	0.721 ± 0.027
$[400, 400, 200]$	0.726 ± 0.013	0.806 ± 0.022	0.747 ± 0.022
$[100, 100, 100, 50]$	0.649 ± 0.070	0.739 ± 0.073	0.690 ± 0.045
$[200, 200, 200, 100]$	0.724 ± 0.016	0.808 ± 0.016	0.732 ± 0.015
$[400, 400, 400, 200]$	0.726 ± 0.025	0.802 ± 0.014	0.747 ± 0.030

Table 4.12: *Comparative study for different width and depth of the architecture - ROSMAP dataset.*

carnitine transmembrane transporter activity (GO:0015226, $p = 4.953E - 2$) and dystroglycan binding (GO:0002162, $p = 3.759E - 3$). For instance, β-alanine has been targeted for its several anti-tumor effects and as a co-therapeutic agent in the treatment of breast tumors [178]. Moreover, the gene SLC6A14 involved in the β-alanine and carnitine transmembrane transporter activities has already been addressed to have a pivotal role in the cancer stage [127], where its deletion has been linked to a reduction of cancer growth and metastatic spread [146], thus being selected as potential direct drug target for cancer therapy [15]. Also, the dystroglycan binding has been linked with breast cancer as the expression of this adhesion molecule is frequently reduced

Omics data type	**Biomarkers**
mRNA expression (15)	GABRP, SOX10, TFF1, KRT6B, AGR3, KLK7, SERPINB5, DSC3, KLK6, AGR2, MIA, TRIM29, SLC6A14, KRT16, KLK8
DNA methylation (10)	IGFBP4, RARA, NHLRC4, CA12, DNALI1, MIR26B, GPR37L1, RSAD1, RARG, NR2F6
miRNA expression (10)	hsa-mir-224, hsa-mir-452, hsa-mir-505, hsa-mir-675, hsa-mir-577, hsa-mir-375, hsa-mir-18a, hsa-mir-196b, hsa-mir-511-2, hsa-mir-145

Table 4.13: *Important biomarkers identified by DeepMKL + KPCA-IG in the BRCA dataset.*

Omics data type	Biomarkers
mRNA expression (15)	PREX1, CSRP1, MID1IP1, PLXNB1, MINDY1, SLC44A1, ANLN, CAVIN1, SLC6A9, DOCK5, ITPKB, SASH1, YES1, CLMN, CARHSP1
DNA methylation (10)	R3HDML, MYOD1, HYAL2, ALDH3B1, OTOP3, CHST14, GPR152, LAG3, ENG, MYO1C
miRNA expression (10)	hsa-miR-423-3p, hsa-mir-374b, hsa-miR-487b, hsa-miR-361-5p, hsa-miR-30b, hsa-miR-885-5p, hsa-miR-376a, hsa-miR-216a, hsa-miR-548b-3p, hsa-miR-26a

Table 4.14: *Important biomarkers identified by DeepMKL + KPCA-IG in the ROSMAP dataset.*

in human breast and colon cancers and is associated with tumor progression [150]. Within this GO, the two enriched genes that we found are AGR3 and AGR2. For instance, AGR3 had already been characterized as a novel potential biomarker both for breast cancer prognosis and early breast cancer detection [50], while AGR2 expression has been correlated with poor outcomes of patients with ER-positive breast cancer [70]. Among others, SERPINB5, DSC3, and GABRP have also been linked with malignant neoplasms of the breast. SERPINB5 has been indicated to inhibit tumor progression [152], DSC3 downregulation has been linked with several cancer types [36] and GABRP over-expression has been linked with poor prognosis, metastatic cancer, basal-like breast cancer [76, 97, 167]. For genes related to the identified DNA methylation features, several interesting GO were enriched, including prosaposin receptor activity (GO:0036505, $p = 1.607E-2$) and insulin-like growth factor II binding (IGF-2) (GO:0031995, $p = 4.011E-2$). Several studies have shown that prosaposin, a regulator of estrogen receptor alpha, promotes breast cancer growth [77, 189] and that IGFs play an important role in cancer development [92] and specifically and increased IGF-2 production has been linked with cancer development and progression in many conditions [32, 35, 120, 181]. Moreover, the highly-ranked miRNAs selected by our method have also exhibited an association with cancer. [204] found over-expression of hsa-miR-224 in breast cancer cell lines and in TNBC primary cancer samples. Another example is hsa-mir-675 as in [179] it has been shown that over-expression of this miRNA enhances the aggressive phenotype of breast cancer cells, including increased cell proliferation and migration in vitro and increased tumor growth and metastasis in vivo.

Deep MKL with KPCA-IG also identifies important biomarkers related to Alzheimer's disease. For AD patient classification, for genes identified by mRNA expression features, several enriched GO has been linked with the Alzheimer patalogy. For instance inositol-1,4,5-trisphosphate 3-kinase activity (GO:0008440 , $p = 3.912E-2$) linked with the gene ITPKB, it has been found to increase in human Alzheimer brain and to exacerbates mouse Alzheimer pathology [162]. Also the choline transmembrane trans-

porter activity (GO:0015220, $p = 4.110E-2$) as been showed to be linked with the disease, as the choline transporter was marked to be incremented in cortical brain regions from AD patients compared to non-AD control [17], as also the gene involved in the signature, namely SLC44A1 has been found to be up-regulated in Alzheimer patients [131]. Moreover, the first gene in the list, namely PREX1 has been reported to be linked with brain-related conditions, such as aberrant neuronal polarity and psychosis-related behaviors, in case of over-expression [96].
Additionally, the GO transforming growth factor beta binding (GO:0050431, $p = 2.69E-2$) was enriched for genes linked to the selected DNA methylation features by our procedure. Dysfunction in TGFβ signaling has been linked to exacerbated neuroinflammation promoting microglia's cytotoxic activation, which may contribute to neurodegeneration in AD [89]. Moreover, several genes are significantly annotated in aldehyde dehydrogenase (NADP+) activity (GO:0033721, $p = 2.911E-2$) where aldehyde dehydrogenase two activity and aldehydic load has been associated to a contribution in neuroinflammation and Alzheimer's disease-related pathology [78]. Another molecular function is the protein tyrosine kinase inhibitor activity (GO:0030292, $p = 3.271E-2$). It has been shown that tyrosine kinase inhibition can be viewed as a potential target for therapeutic intervention for treating Alzheimer's disease as it represents a valid mechanism for improving autophagic clearance of neurotoxic protein and mitigating mast cell and microglial-mediated inflammation [161]. Other GO potentially related to AD are hexosaminidase activity (GO:0015929, $p = 4.290E-2$) and galactose binding (GO:0005534, $p = 3.271E-2$), where abnormal cortical lysosomal β-hexosaminidase and β-galactosidase activity has been linked both to early and the advanced stage of Alzheimer's disease [107]. Regarding the miRNA biomarkers, our methods selected, among others, hsa-miR-361-5p, which was found to be abnormally expressed in AD patients [111]. Another highly-ranked miRNA, hsa-miR-885-5p, is substantially expressed in brain tissues and has been associated with AD [175].

4.5 Concluding remarks

In this chapter, we introduced and discussed the problem of multi-omics integration, providing a rich literature background on the multiple kernel learning framework and state-of-the-art deep learning approaches. In particular, we focused on a common issue observed in many of the current methods: their reliance on early or late integration strategies, which can be potentially limiting when dealing with different biological layers [132]. Multiple kernel learning is a well-established algorithm in the machine learning community, but its use remains limited among practitioners in bio-data mining. Unlike early or late integration methods, MKL naturally supports a mixed integration framework, making it a flexible and promising solution for combining heterogeneous biological data sources. This chapter presents two novel different approaches for Mul-

tiple kernel learning in the context of multi-omics data integration. One employs unsupervised learning techniques along with Support Vector Machines (SVM). The other utilizes deep learning as a substitute for convex linear optimization to integrate kernels. The proposed methodologies are tested and compared with state-of-the-art methods performances. The experimental results on four publicly available biomedical datasets show that approaches based on kernel mixed integration exhibit comparable or even improved performance w.r.t [183] [55] [63] while being considerably simpler. Also the novel deep learning-based procedures used to integrate input kernels and for classification demonstrate to be a valid alternative to the more classical Multiple kernel learning optimizations in the case of datasets with large enough sample size. In addition, we proposed a novel method for biomarkers discovery based on our newly proposed Deep MKL method, which proved effective for predicting the disease of interest, potentially showing disease mechanisms and helping in the development of personalized treatment protocols. In this case, our method offers deeper insights than traditional methods focused solely on prediction performance, such as those that sequentially remove features to rank their importance based on the impact on prediction accuracy, as in [183] and [55]. Future work could investigate other types of data kernel embedding and different deep architectures to exploit the kernel framework in the context of Deep multiple kernel learning. For classical multiple kernel learning, different types of kernel functions can be tested, as each omic dataset could benefit from ad-hoc kernel function choices. MKL showed that despite being under-utilized in multi-omics data analysis, it provides a fast and reliable solution that can compete with and outperform more complex architectures.

The code of the work presented in this chapter can be found at `https://github.com/gabrieletaz/MKL_MO`.

The author of this PhD thesis is responsible for the following contributions presented in this chapter:

III/1. Contributed to conceptualization and design of the work: comparing deep learning state-of-the-art approaches to multiple kernel learning ones, based on SVM and deep learning, on biomedical multi-omics datasets.

III/2. Literature survey regarding the deep learning approaches in the Related Work section.

III/3. Implementation of the data preprocessing steps.

III/4. Conceptualization and implementation of all DeepMKL architectures and relative evaluation pipeline and experiments.

III/5. Conceptualization and implementation of the novel two-steps intepretability method used for biomarker discovery.

III/6. Design of figures related to the DeepMKL architectures.

Chapter 5

MINN: A Metabolic-Informed Neural Network for Integrating Omics Data into Genome-Scale Metabolic Modeling

The understanding of cellular behavior relies on the integration of metabolism and its regulation. Multi-omics data provide a detailed snapshot of the molecular processes underpinning cellular functions and their regulation, describing the current state of the cell. While Machine Learning (ML) models can uncover complex patterns and relationships within these data, they require large datasets for training and often lack interpretability. On the other hand, mathematical models, such as Genome-Scale Metabolic Models (GEMs), offer a structured framework for analyzing the organization and dynamics of specific cellular mechanisms. At the same time, they don't allow for seamless integration of omics information. Recently, a new framework to embed GEMs in a neural network has been introduced: these hybrid models combine the strengths of mechanistic and data-driven approaches, offering a promising platform for integrating different data sources with mechanistic knowledge. In this chapter, we present our works "Metabolic-informed Neural Network for Multi-omics Data Integration" published in the proceedings of FOODSIM 2024 and "MINN: A metabolic-informed neural network for integrating omics data into genome-scale metabolic modeling" published in "Computational and Structural Biotechnology Journal". After the Introduction 5.1, this chapter is structured as follows. The Related Works section 5.2 provides a literature background on Genome-Scale Metabolic Models and Flux Balance Analysis, the integration of omics data into GEMs, and the development of hybrid approaches combining mechanistic and machine learning models. This section explains why combining data-driven and mechanistic frameworks is a promising direction for metabolic flux prediction. The Materials and Methods section 4.3 describes the MINN architectures proposed in our work, the datasets used, the preparation of GEMs, and the implementation of different optimization strategies to balance data-driven and mechanistic

objectives. It also details the evaluation pipeline, metrics, and computational settings used for the evaluations of our methods. The Results and Discussion section 5.4 presents the predictive performance of MINN compared with classical machine learning methods and pFBA, as well as results from different optimization strategies, GEM configurations, and the MINN-reservoir approach. It also includes statistical significance tests and additional analyses to highlight the robustness of the method. Finally, in the Concluding Remarks section 5.5, we summarize the main findings of our work, emphasizing the advantages of hybrid models for flux prediction, the role of mechanistic constraints in regularization, and the trade-offs between predictive accuracy and mechanistic fidelity.

5.1 Introduction

The phenotype of a cell is a complex interplay between its metabolic network, consisting of thousands of biochemical reactions, and the regulatory mechanisms controlling diverse cellular functions. Mechanistic models, such as GEMs [173], provide a structured framework to integrate and connect the available knowledge to find emergent properties in cellular systems. GEMs mathematically represent cellular metabolism, summarizing our information about the biochemical processes present in an organism [113, 125, 156]. One common approach to simulate cellular behavior using GEMs is constraint-based modeling (CBM). Among these methods, Flux Balance Analysis (FBA) [24, 124, 128] is particularly notable. FBA applies linear programming to optimize the distribution of metabolic fluxes, aiming to maximize specific objectives like biomass production while considering nutrient availability constraints. However, the predictive power of a mechanistic model like FBA is limited by the completeness of our understanding of cellular processes. Moreover, FBA typically has multiple feasible solutions. In such cases, the solution with the lowest sum of fluxes is usually selected, based on the assumption that cells try to minimize their enzyme production [93]. However, this assumption is often an oversimplification, which does not account for the complex regulatory mechanisms within cells.

Omics data can be integrated in GEMs to enhance their predictive power and tailor models to specific cellular contexts [105, 196]. Transcriptomics and proteomics offer indirect and direct proxies for metabolic activity, respectively, and have been incorporated into GEMs using tools such as GIMME [13] and CoCo [201] for gene expression data, and GECKO [29] and sEnz [174] for enzyme abundances. Metabolomics and fluxomics provide additional constraints through extracellular metabolite levels or isotope-labeling experiments [10, 68]. While multi-omics integration offers a more comprehensive view of the cellular state [103], these efforts remain limited by standardization challenges and the difficulty of mechanistically linking non-metabolic features to model reactions [99].

On the other hand, data-driven machine learning (ML) models can effectively extract patterns in high-dimensional datasets, such as multi-omics data, without prior knowledge of underlying molecular mechanisms. These models often demonstrate strong predictive capabilities, but are limited by the scarcity of biological datasets, frequently constrained by experimental costs and time. In recent years, ML models have also been explored for predicting metabolic fluxes using multi-omics data. Although FBA remains the preferred mechanistic framework for this task, integrating omics data into CBM frameworks remains a significant challenge [105]. Interestingly, recent work [56] demonstrated that purely ML-based approaches trained on omics data can outperform FBA-based methods in metabolic flux prediction. A more detailed overview of the literature is provided in the next section.

Given the complementary strengths and limitations of ML and GEMs, there has been increasing interest in trying to combine these two approaches [9, 82, 202]. Hybrid models that merge mechanistic knowledge with the predictive capabilities of ML offer a promising direction but so far, as highlighted in [143], the existing applications do not truly integrate ML and FBA. Instead, they mostly combine them in two separate steps: using ML as input for FBA [39, 81, 115], or using FBA as input for ML [37, 106].

Recently, [46] developed a framework that truly combines CBM with ML in a neural network architecture called an Artificial Metabolic Network (AMN). Their approach leveraged GEM structures and FBA constraints within neural networks to predict growth rates from media compositions.

In [46], the authors suggested three different possible configurations for incorporating the GEM structure and the FBA constraints in a neural network (NN). In this work, we selected one of these configurations, inspired by Physics-Informed Neural Networks (PINNs) [38], and we re-implemented and expanded it to integrate multi-omics data as inputs. We will refer to this new architecture as Metabolic-Informed Neural Network (MINN) with multi-omics integration (Figure 5.1). We applied this hybrid model to the dataset analyzed by [56], which examines how the metabolism of *Escherichia coli* adapts to varying growth rates and single-gene knockouts [75]. As discussed in [171], the combination of fluxes measured experimentally lies outside the solution space of FBA, causing a conflict between the optimization of the data-driven and the mechanistic objectives. To address this, we provide different mitigation strategies.
To summarize, in this work:

i. We describe the implementation of a MINN with multi-omics integration, using an early concatenation approach.

ii. We benchmark its predictive performances on the ISHII dataset, compared to pure ML methods [56].

iii. We recalculated the data to be in the FBA solution space and compared the predictive performances with those based on the original measurements.

iv. We explore different hybrid optimization strategies to address the conflict between the objectives, while using the original data.

v. Finally, we adapt the MINN to a "reservoir" configuration [46], which uses the MINN predictions to directly constrain pFBA, and compare its predictions with those of pFBA alone.

With these analyses, we provide a detailed overview of methods and strategies to adapt and use hybrid ML-FBA methods for multi-omics integration. Our findings highlight the potential of hybrid models to enhance the predictive accuracy and robustness of metabolic flux predictions. This can be intended as a first step in the direction of more precise and comprehensive metabolic network analyses, particularly for phenotypes where metabolism is significantly influenced by other layers of cellular organization, which are challenging to incorporate into FBA. Furthermore, with this work we aim to provide a guide to the use of the MINN framework, helping researchers choose the most suitable configuration based on the specific objective of their study.

5.2 Related Works

Given the multidisciplinary nature of this work, we considered it important to provide a common starting point for the main topics covered, namely: Genome-Scale Metabolic Models and Flux Balance Analysis, Omics Data Integration, and Hybrid mechanistic and data-driven modeling.

5.2.1 Genome-Scale Metabolic Models and Flux Balance Analysis

A genome-scale metabolic model (GEM) is a comprehensive reconstruction of an organism's metabolic network, representing the full metabolic capacity encoded by its genome. It serves as a structured knowledge base, integrating information on genes, proteins, enzymes, and metabolic pathways [113**?**]. GEMs have been primarily reconstructed for microorganisms, but models also exist for multicellular organisms, including humans. A typical microbial GEM contains hundreds or even thousands of reactions and metabolites, increasing in complexity for multicompartment systems like yeast [156]. To analyze such large models, a commonly used method is Flux Balance Analysis (FBA), which relies on the assumption of steady state or balanced growth [24**?**]. Under these conditions, the concentrations of metabolites remain constant over time, and the rates of production and consumption are balanced across all reactions. GEMs can therefore be formulated only in terms of reaction rates and treated as linear programming problems. FBA uses this framework to predict the metabolic behavior of the organism by optimizing, given the stoichiometric constraints, a specific objective function: commonly biomass production or, in biotechnological contexts, the yield of

a desired product. However, stoichiometric constraints alone are often insufficient to determine a realistic flux distribution. To improve predictive accuracy, FBA requires context-specific inputs—such as growth medium composition—typically incorporated by constraining exchange fluxes. These constraints may be derived from experimental measurements, assumptions about nutrient uptake kinetics, or a combination of both [128].

5.2.2 Omics Data Integration in Genome-Scale Metabolic Models and Flux Balance Analysis

GEMs are reconstructed primarily from genomic information, encoding the metabolic network of an organism. A GEM can be tailored to represent different cell strains by including or excluding reactions based on the presence or absence of genes encoding the relevant metabolic enzymes [173]. Beyond genomics, a broad range of omics data (e.g., transcriptomics, proteomics, metabolomics, and fluxomics) can significantly enhance the accuracy and predictive capacity of GEMs [105, 196]. The challenge lies in translating these data into metabolic fluxes. For enzyme-catalyzed reactions, the reaction rate is typically described by:

$$v = k_{cat} \cdot e \cdot f(s, p) \tag{5.1}$$

where:

- v is the reaction's flux;
- k_{cat} is the catalytic constant, or turnover number, which represents the number of substrate molecules converted to product per enzyme molecule per unit time when the enzyme is fully saturated with substrate, i.e., the enzyme efficiency;
- e is the enzyme concentration;
- $f(s, p)$ is a (often nonlinear) function of the concentrations of substrates s and products p, and the corresponding affinity parameters.

It is worth also noting that, while genomic, transcriptomic, and proteomic data provide rich layers of information, only features that can be explicitly linked to metabolic reactions can be directly integrated into GEMs [99]. As a result, much of the broader cellular context captured by these datasets, including regulatory, structural, or signaling components, is typically excluded from the model.

Transcriptomics data, while often weakly correlated with actual protein levels or enzymatic activity [133], remain useful to identify active metabolic genes and infer condition-specific pathway activation. When comparing multiple conditions, transcriptome profiles may suggest shifts in metabolic strategy. Several frameworks have

been developed to incorporate transcriptomics and gene co-expression data into GEMs [105, 126, 201]. Transcriptomic data have also enabled the development of metabolism and gene expression models (ME-models), which explicitly couple metabolic reactions with the expression of the genes encoding the corresponding enzymes [102, 119]. These models account for the transcriptional and translational cost of enzyme production and provide a mechanistic link between gene expression and flux capacity.

To address the challenge of nonlinear and context-dependent relationships between transcript levels and metabolic fluxes, some methods avoid imposing direct constraints and instead seek to maximize the consistency or correlation between gene expression and flux predictions [7, 129, 130, 209].

Proteomics data provide a closer proxy for metabolic capability. Presence or absence of specific enzymes can directly constrain which reactions are allowed under given conditions. Assuming enzyme saturation ($f(s,p) = 1$), enzyme levels scaled by k_{cat} values provide upper bounds for fluxes (v_{max}). These catalytic parameters can be obtained from databases such as BRENDA [26] and SABIO-RK [187], or estimated using statistical approaches or enzyme-kinetic models [43, 154]. Although these resources are growing rapidly, retrieving the relevant kinetic parameters remains a semi-automated process that often requires manual curation to ensure accuracy and model compatibility. Building on these concepts, more complex model formulations have been developed, such as enzyme-constrained GEMs (ecGEMs), which treat metabolism as a problem of protein budgeting under limited cellular capacity [29, 144]. Proteome-constrained models (pcGEMs) further account for the resource cost of enzyme production [45, 58]. While building these models often requires custom pipelines, tools such as GECKO [29] and sEnz [174] are making these tasks increasingly standardized and accessible.

Metabolomics data offer insights into both the structure and dynamics of metabolism. Although metabolite concentrations cannot be directly used in GEMs due to the steady-state assumption and lack of proportionality between concentrations and fluxes, their presence or absence can indicate pathway activity. Time-series measurements of extracellular metabolite levels can be converted into flux constraints for exchange reactions, allowing us to tune the GEMs to match observed uptake or secretion patterns [68]. Moreover, when quantitative metabolomics data are available for the reagents of a reaction, thermodynamic constraints can be imposed on its direction, further narrowing the feasible flux space and improving biological realism [118, 129, 130].

Fluxomics data: isotope-labeling experiments (e.g., growth in ^{13}C-glucose medium) allow direct estimation of intracellular flux distributions via Metabolic Flux Analysis (MFA) [10]. MFA uses ^{13}C metabolomics data to fit simplified metabolic models, enabling the estimation of intracellular fluxes based on the observed labeling patterns. These experimentally derived fluxes can be used to constrain specific reactions or to find the flux profile that best fits the measured data.

Multi-omics data : The simultaneous integration of diverse omics layers yields a

more holistic view of the cellular state. This systems-level approach is especially valuable for understanding dynamic or context-dependent responses, as it captures the interactions between different biological processes [103]. Tools like the IOMA (Integrative Omics-Metabolic Analysis) framework facilitate the incorporation of diverse omics layers into GEMs to enhance predictive fidelity [196].

In summary, omics data integration represents a powerful way to contextualize and refine GEMs, improving their ability to simulate real-world biological behavior. However, current methods remain limited by a lack of standardization and automation. Moreover, mechanistic integration is restricted to metabolic features explicitly represented in GEMs, meaning that valuable context from a broader view of cellular processes is still usually excluded.

5.2.3 Integrating FBA and Machine Learning for Enhanced Metabolic Predictions

In the previous sections, we discussed how FBA is a powerful approach to exploit the information stored in GEMs to predict the metabolic behavior of cells. However, FBA has at least four main limitations. First, its predictive power heavily depends on the amount of experimental measurements of exchange fluxes. Second, incorporating multi-omics data is challenging, because all measurements must be converted into fluxes, a process that usually requires iterative steps of time-consuming manual curation. Third, FBA and GEMs focus solely on metabolism and typically do not link it to the general status of the cell. Finally, FBA predicts flux distributions that tend to maximize the yield on the limiting substrates [176], often missing to capture "high rate-low yield" solutions [45].

In recent years, with the increasing availability of high-throughput technologies and data, ML has gained popularity as a valid alternative to mechanistic-based approaches [6, 56, 190]. The success of ML lies in its ability to find patterns in the data without making any mechanistic assumptions. However, a main drawback is that ML requires a high volume of data to train models successfully, and in many biology-related domains, datasets of suitable size are rare. In particular, experiments in microbial physiology tend to be one, if not two, orders of magnitude smaller than what ML requires. Moreover, ML behaves mostly as a black-box model, making it difficult to extract mechanistic understanding from its results. On the other hand, this black-box nature makes ML more amenable than mechanistic models for integrating diverse data sources, even those for which there is no clear understanding of their connections.

Therefore, it seems natural to integrate these two approaches to overcome each other's limitations and exploit their strengths. In recent years, as reviewed in [143] and [202], there have been many attempts to integrate these methods. [143] categorize these works into two groups: ML as input of FBA [39, 81, 115] and FBA as input of

ML [37, 106]. This division highlights that these methods do not truly integrate ML and FBA but rather concatenate them, using them in two distinct steps. To the best of our knowledge, only three works presented hybrid models that genuinely integrate FBA and ML: [46], [65], and [7]. The first introduces Artificial Metabolic Neural Networks (AMNs), which are Neural Networks that use FBA constraints to refine their solution and to regularize the network. This is achieved through a Mechanistic Layer, representing the structure of the mechanistic model inside the NN, and a custom loss function, similar to those of other Knowledge Informed Neural Networks (e.g. Physics-Informed Neural Network [38]). When using FBA alone for growth rate prediction, nutrient uptake fluxes often need manual adjustment to match experimental growth rates. This process can involve labor-intensive experiments or unsystematic "trial-and-error" adjustments, which may introduce arbitrary assumptions to align the model with observed data. The hybrid AMN framework proposed by [46] addresses these challenges by embedding mechanistic information into neural networks, providing a more systematic approach. The second presents FlowGAT, which integrates the structure of the GEM and the solution of FBA in a Graph Attention Network (GAT) to predict the gene essentiality. The third method, scFEA, combines single-cell transcriptomics with FBA-inspired constraints using a Graph Neural Network. Like AMNs, scFEA treats flux balance as a soft constraint in the loss function, but it also includes a term that explicitly maximizes the agreement between gene expression and predicted fluxes.

The MINN models follow the blueprint of AMNs and, in line with the approach presented in [46], represent a true hybrid model, integrating FBA constraints and multi-omics data to improve predictions of fluxes. One key feature of MINN is that it incorporates omics data not only for elements (genes, proteins, etc.) directly linked to metabolic activity, but also those representative of the broader cellular context, and it leaves it to the neural network component to learn the complex relationships between all omics layers and metabolic fluxes. However, as a possible future development, GPRs could be embedded directly into the network architecture or in the loss function of a MINN, strengthening the mechanistic link between omics data and flux predictions and further leveraging the structure encoded in the GEM.

5.3 Materials and Methods

5.3.1 Dataset

The dataset analyzed in this work was originally published by [75] and consists of 29 chemostat experiments, in which *E. coli* was grown in glucose minimal medium. Wild-type strain K-12 was grown at 5 different dilution rates (D = 0.1, 0.2, 0.4, 0.5, and 0.7 h^{-1}), while 24 different single-knockout mutant strains were cultivated at fixed dilution

rate (D = 0.2 h^{-1}). The same dataset was already used by [56] to test traditional ML for the prediction of metabolic fluxes from multi-omics data.
The dataset consists of transcriptomic, proteomic, and fluxomic measurements. For each sample, microarrays were used to assess the expression profiles of 79 genes and LC-MS/MS quantitative proteomics to measure the abundances of 60 proteins. ^{13}C-labeled metabolomics experiments were analyzed with MFA to estimate 47 metabolic fluxes: 37 reactions of the central carbon metabolism, 9 exchange fluxes (production or consumption of external metabolites) and biomass growth.

The metabolic model used by [75] to perform MFA is a core model that mainly represents the central carbon metabolism of *E. coli* and how it connects to the measured external metabolites. This model is much smaller and less complete than the GEM [47] integrated in the MINN. For the GEM to grow, many more different biomass components must be synthesized, diverting some metabolic precursors outside the pathways represented in the MFA model. For this reason, the fluxomics data from [75] lie outside the solution space [171]. In most of our analyses we used the original fluxomics data, to highlight the ability of the MINN to reconcile MFA fluxomics data with the structure of the full-size GEMs. However, to investigate the impact of this discrepancy, we repeated some of the analyses with a second set of fluxes, now residing in the FBA solution space. This second set of fluxomics data is composed of the fluxes with the minimum Euclidean distance from the original ones, following an approach detailed in the Supplementary Material of [105] and we refer to it as *FBA fit* data.

	GEMs	
GEM name	original splitted reactions	reduced and splitted reactions
iAF1260	2957	NA
iAF1260 *FVA-reduced*	2957	1873
iAF1260 *FBA-reduced*	2957	587
e_coli_core	115	NA
iNF517 *FVA-reduced*	1022	704

Table 5.1: *Dimensions of all the GEM used in this analysis.*

5.3.2 GEM preparation

In this section we describe all the genome-scale metabolic reconstructions utilized to build the MINNs. The most recent GEM available for E. coli K-12 is iML1515 [114], but we opted for iAF1260 [47]. The two differ mainly for the more comprehensive coverage of accessory pathways of iML1515, which are relevant in complex environments like the human gut, but not for growth on minimal medium. On the other hand, the size of

the GEM can heavily affect the complexity of the MINN: using a smaller GEM would improve the efficiency of our hybrid model by reducing the computational resources required for training. Possibly, it would also reduce the noise in the model, enhancing the prediction accuracy. iAF1260 is reasonably smaller than iML1515 (2382 reactions vs. 2712) and is also the same model used by [56] in their analyses.

We further reduced the size of the model by excluding all the reactions which cannot carry flux during growth in glucose minimal medium. This was achieved performing Flux Variability Analysis (FVA) and retaining only the reactions with a non-zero span. We refer to this model as *FVA-reduced* GEM. We also tested a second strategy, inspired from [46], to further reduce the model. We generated a dataset of 2000 FBA solutions by randomly selecting single-gene knockouts and varying the maximum glucose uptake rate within the experimentally observed range. Reactions that consistently carried zero flux across all the simulations were removed from the model. We refer to this model as *FBA-reduced* GEM. To investigate the impact of an extreme decrease in the genome-scale reconstruction size, we also built a MINN using the e_coli_core model [123], a manually reduced GEM focused on central carbon metabolism, which is the smallest model available in the BiGG database.

Finally, to further test the role of the GEM and the underlining metabolic network, we also tested our baseline configuration including the GEM of a different organism. We used the iNF517 model [49] for *Lactococcus lactis subsp. cremoris MG1363*. This microorganism is a lactic acid bacterium, with an incomplete TCA cycle, which makes it an interesting comparison for *E.coli*, both in terms of structure of the network in the central carbon metabolism and of general metabolic behavior. The model was reduced using the FVA-guided reduction approach. The results of this comparison are available in the Section 5.4.4.

In Table 5.1 we summarize the dimensions of each GEM. The GEMs were downloaded from the BiGG database [84] and handled/modified using CBMPy 0.8.4 [122]. In each model, reversible reactions were split into a forward and a reverse reaction using the built-in CBMPy function `cbmpy.CBTools.splitReversibleReactions`.

5.3.3 MINN architecture

This work presents a MINN architecture designed to predict multiple fluxes using multi-omics data, which provide key insights for metabolic predictions but are challenging to integrate with FBA [105]. Figure 5.1 illustrates the structure of our MINN architecture, built to predict fluxes measured in the ISHII dataset using proteomics, transcriptomics, and the measurements of two exchange fluxes, namely *R_EX_glc__D_e*, *R_EX_o2_e*. The data are integrated using an early concatenation strategy [5], where the three omics datasets are combined into a single matrix that is fed into the MINN.

The omics data are used in the first part of the model (shown in red in Figure 5.1), which is a pure feed-forward neural network. Here, the network is trained to

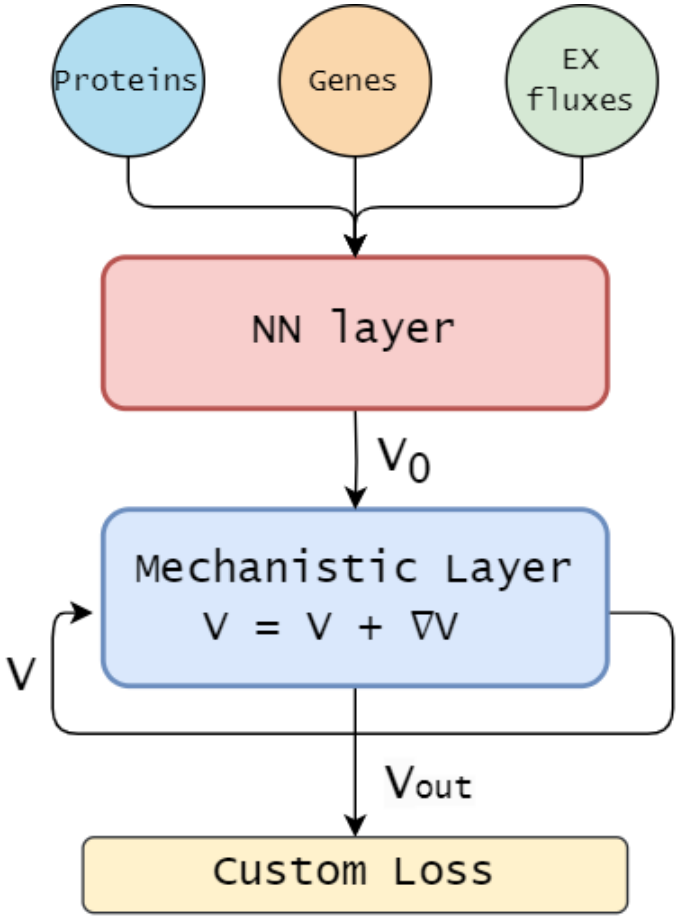

Figure 5.1: *Schematic representation of the MINN architecture. Protein and gene expression levels, and exchange flux data are used as input to a feed-forward neural network, which produces an initial estimate for the flux distribution V_0. This estimate is refined in a mechanistic layer via a gradient descent step to better align with flux balance constraints, resulting in the final flux distribution V_{out}. The custom loss function combines the discrepancy between the model predictions and the target fluxomics data with the violation of FBA constraints, and is used to train the network via backpropagation.*

learn a mapping from the input omics profiles to an initial estimate of the flux distribution, denoted as V_0. The input dimension d_{in} corresponds to the total number of features after concatenating the transcriptomics, proteomics, and exchange flux data, while the output dimension d_{out} corresponds to the total number of reactions in the GEM. This part of the model is purely data-driven, meaning that it does not rely on any mechanistic assumption or require prior knowledge such as gene-protein-reaction (GPR) associations. Instead, it learns directly from the data how to associate the omics features with a plausible flux configuration. This makes the method flexible and compatible with a wide range of input omics data, including those not directly related

to metabolism but still informative of the broader cellular context.

The second part, the mechanistic layer, blue in Figure 5.1, consists of a gradient descent optimization loop. This loop refines the output of the first neural network step by adjusting the final predicted flux distribution V to better comply with the FBA constraints, minimizing the FBA loss function L_{FBA}. The neural network weights are trained using a standard back-propagation algorithm and a custom loss function L_{MINN} that considers both the data error and the FBA constraints.

To formalize this, let the input data be $X \in \mathbb{R}^{N \times d_{in}}$ where N is the mini-batch dimension in the back-propagation algorithm and d_{in} is the number of features of X. The first NN part can be expressed as:

$$V_0 = \sigma(XW^h + b^h)W^{out} + b^{out} \tag{5.2}$$

with σ the ReLU activation function, $W^h \in \mathbb{R}^{d_{in} \times d_h}$, $b^h \in \mathbb{R}^{1 \times d_h}$, $W^{out} \in \mathbb{R}^{d_h \times d_{out}}$, $b^{out} \in \mathbb{R}^{1 \times d_{out}}$, the weight matrices and the biases of the input and hidden layer respectively, where d_h is the hidden layer dimension and d_{out} is the output dimension, which coincides with the dimension of the flux distribution. Then, V_0 is refined in the mechanistic layer through a gradient descent optimization, using only the mechanistic constraints. For simplicity, the notation refers to the simple case when the loop has one iteration:

$$V_{out} = V - lr\frac{\partial L_{\text{FBA}}}{\partial V} \tag{5.3}$$

$$L_{\text{FBA}} = \frac{1}{m}\left|SV\right|^2 + \frac{1}{n_{\text{in}}}\left|\text{ReLU}(P_{\text{in}}V - V_{\text{in}})\right|^2 + \frac{1}{n}\left|\text{ReLU}(-V)\right|^2 \tag{5.4}$$

The first element represents the steady-state constraint of FBA, with S as the stoichiometric matrix of the GEM. The term m denotes the number of metabolites and serves as the normalization term. The second element represents the upper-bound constraint on the vector of fluxes V_{in}. Here, P_{in} represents the projection matrix that projects the flux distribution vector V into the dimension of V_{in}, while the normalization term n_{in} stands for the number of bounded fluxes. Lastly, the last element symbolizes the lower bound constraint, which required since the GEM is built to ensure that all the fluxes are positive.

The custom loss used to train the weights of the MINN is:

$$L_{\text{MINN}} = L_1 + L_2 + L_3 + L_4 \tag{5.5}$$

$$= \frac{|P_{\text{ref}}V - V_{\text{ref}}|}{V_{\text{ref}}} + \frac{1}{m}|SV|^2 + \frac{1}{n_{\text{in}}}|\text{ReLU}(P_{\text{in}}V - V_{\text{in}})|^2 + \frac{1}{n}|\text{ReLU}(-V)|^2$$

where V_{ref} is the vector with the measured fluxes and P_{ref} a projection matrix that projects V to the dimension of V_{ref}; while the other elements represent the FBA constraints and are the same as in L_{FBA}.
In order to guide the reader through the understanding of the MINN architecture, we provide an illustrative toy example in Section 5.3.3

In [46], the authors used Mean Squared Error (MSE) as L_1 because they wanted to predict a single flux, specifically the growth rate. In our work, we use the Normalized Error (NE) [56] to have a scale-invariant L_1 when predicting multiple fluxes in order to avoid favoring reactions with higher flux values.

In addition, during our analysis a conflict between the data-driven and mechanistic losses emerged. In order to mitigate this issue, we multiply $L1$ with a constant c, which allow us to adjust the balance between the two losses:

$$L_{\text{MINN-balanced}} = c \cdot L_1 + L_2 + L_3 + L_4 \tag{5.6}$$

The c constant becomes a hyperparameter of the model, tuned using k-fold cross-validation and the optimized value determines the best balance between L_1 and $(L_2 + L_3 + L_4)$.

It is important to note that the mechanistic part of the loss function includes only terms enforcing FBA constraints, to reduce the solution space, but does not include any term related to the FBA objective (e.g., biomass maximization). As a result, the optimization is guided solely by L_1, a data-driven objective, without imposing any predefined metabolic goal. This is particularly advantageous in cases where no clear cellular objective exists, such as in gene knockout mutants.

MINN-reservoir

Similarly to [46], we tested an additional configuration of the MINN, named MINN-reservoir. The training of the MINN-reservoir requires two steps, as shown in Figure 5.2. In the first step (5.2a), a MINN (with no omics data in input) is trained only to reproduce FBA using a dataset of FBA solutions. This dataset contains the results of 2000 FBA simulations, in which the reactions belonging to V_{in} *(R_EX_glc__D_e, R_EX_o2_e, R_EX_co2_e, R_EX_etoh_e and R_EX_ac_e)* were assigned random values, within the ranges of variability observed in the ISHII dataset. This procedure creates a model that acts as a pure approximator of an FBA solver (Pretrained block in Figure 5.2), capable of predicting the optimal flux distributions from measurements of

external metabolite fluxes, similar to an FBA solver. In the second step (Figure 5.2b), this Pretrained block is embedded into a new MINN architecture, where it replaces the Mechanistic Layer. The resulting architecture consists of a neural network layer that predicts V_{in} from multi-omics data and medium exchange fluxes (*R_EX_glc__D_e, R_EX_o2_e*), followed by the *Pretrained block*, which computes the flux distribution V_{out} from the predicted V_{in}. The two-step approach ensures that the predicted V_{in} values are compatible with FBA and can be reliably used by the solver to produce a true, linear programming solution. For the test sample in each split, the predicted V_{in} are extracted and used as additional constraints for pFBA: increasing the input information in a data-driven way while preserving the mechanistic structure of the model. Unlike the default configuration of the MINN, this approach produces as final output not only the full flux distribution, but a complete solution from a Linear Programming solver, which can be analyzed with all the tools developed for this purpose.

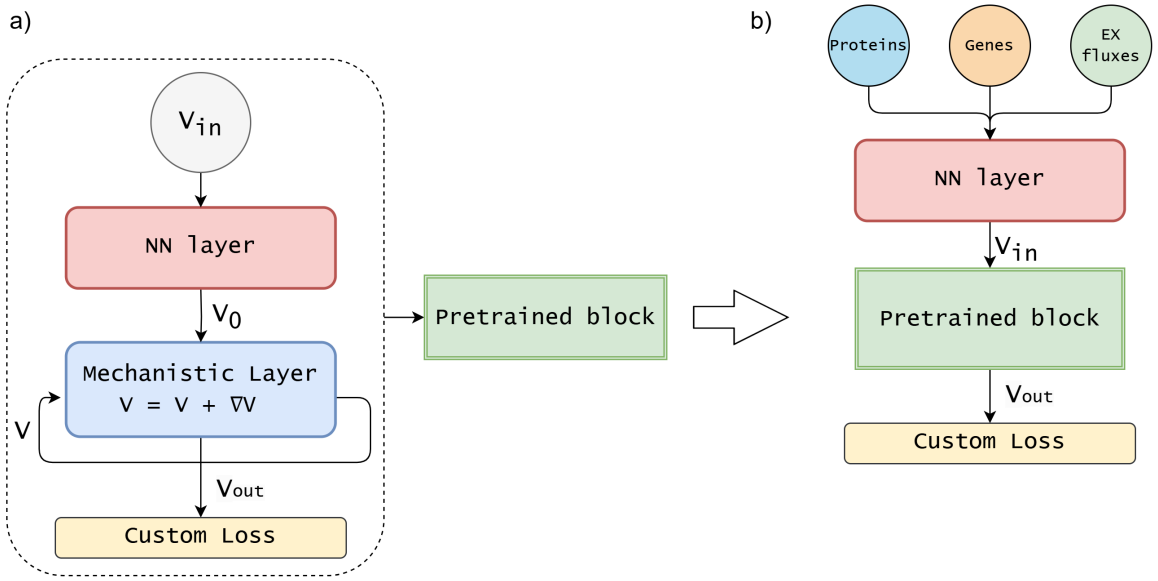

Figure 5.2: *Two-step training strategy of the MINN-reservoir architecture: a) In the first step, a MINN (with no omics data in input) is trained to approximate an FBA solver, using a dataset of simulated FBA solutions. The network learns to predict the flux distribution V_{out} from randomly sampled external fluxes V_{in} (R_EX_glc__D_e, R_EX_o2_e, R_EX_co2_e, R_EX_etoh_e and R_EX_ac_e). Once trained, its weights are frozen, and the resulting model is reused as a fixed Pretrained block. b) In the second step, this Pretrained block is embedded within a new architecture that takes omics data and medium exchange fluxes (R_EX_glc__D_e, R_EX_o2_e) as input. A neural network predicts V_{in}, which is then passed to the Pretrained block to compute V_{out}.*

MINN architecture: toy example

Here we present a toy example (Figure 5.3) that shows step by step how the MINN architecture works. Starting from a single input sample, we walk through the key components of the model: from omics feature concatenation, to the neural network prediction, and finally to the mechanistic refinement using FBA constraints.

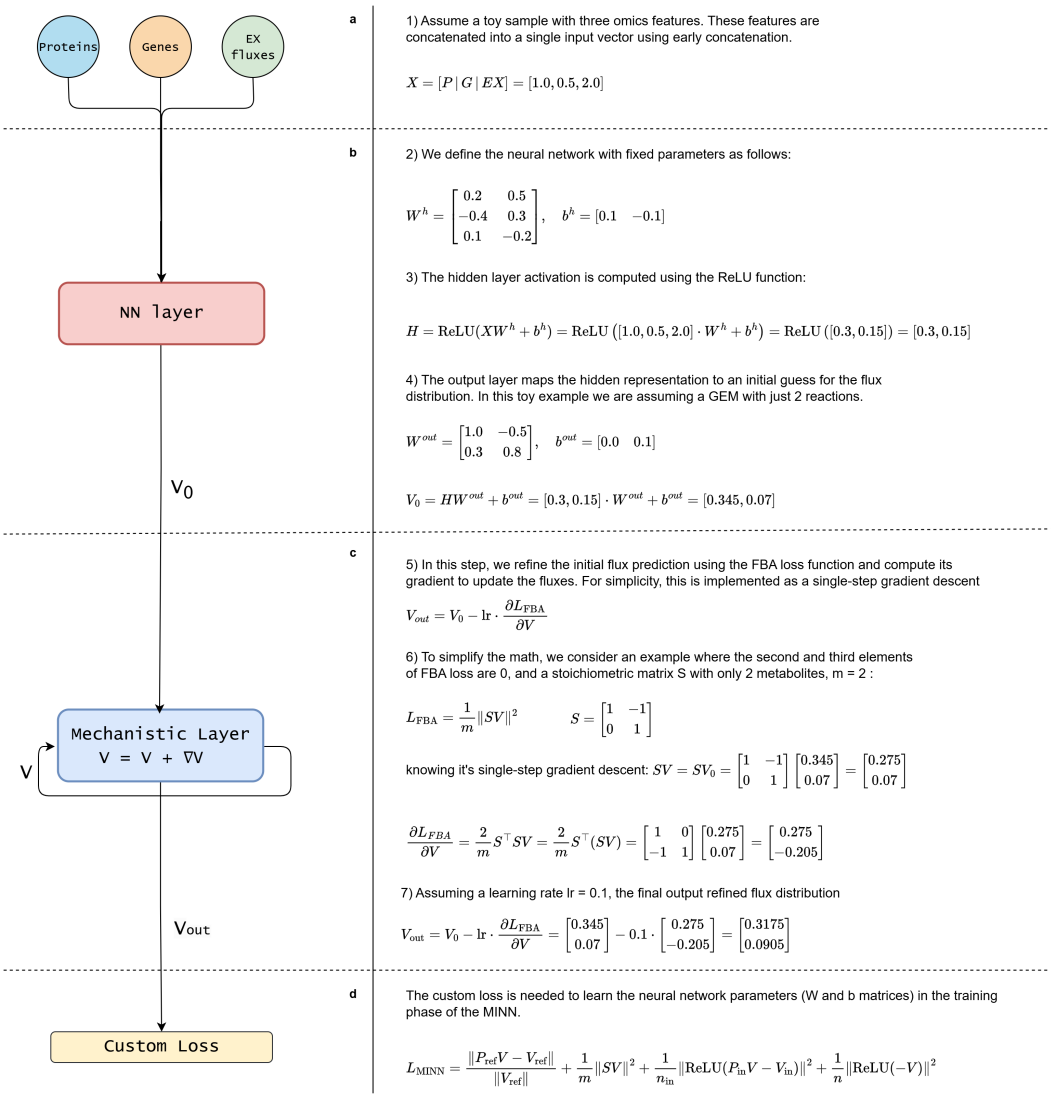

Figure 5.3: *Toy example illustrating the workflow of the MINN architecture.* ***(a)*** *Omics features (proteomics, transcriptomics, exchange fluxes) are concatenated into a single input vector X.* ***(b)*** *A feedforward neural network maps X to an initial flux prediction V_0 using learned weights.* ***(c)*** *A mechanistic layer refines V_0 via one step of gradient descent, enforcing FBA constraints, and outputs the predicted flux distribution V_{out}.* ***(d)*** *A custom loss combining prediction error and FBA constraints is used to update the neural network during training.*

5.3.4 Hybrid Optimization Strategies for Data-Driven and Mechanistic Integration

Equation 5.5 highlights how the loss of the MINN is composed of two components: L_1, which drives the optimization on the data, and $L_{FBA} = L_2 + L_3 + L_4$, which minimizes the divergence from the mechanistic constraints. In Equation 5.6, we already introduced the coefficient c, which allows us to tweak the balance between the two components, either manually or through hyperparameter optimization. In this section, we introduce three other methods to tune this balance while minimizing the trade-off between the two components.

In developing hybrid models that integrate mechanistic constraints with data-driven approaches, we propose different strategies to balance the objectives of maintaining adherence to the FBA constraints without substantially compromising predictive performance. These methods address the challenge posed by different scales of mechanistic and data-driven losses, ensuring that neither dominates the optimization process and that the model generalizes well to unseen data.

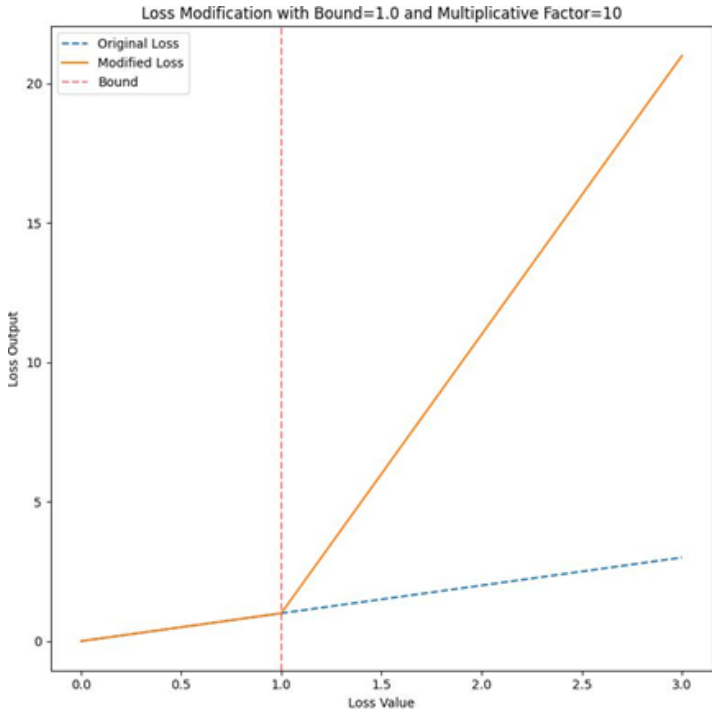

Figure 5.4: *Illustration of the mechanistic loss bound application. The original loss (blue dashed line) remains linear, while the modified loss (orange line) increases steeply after surpassing the bound (red vertical line). This demonstrates how the bound prevents the mechanistic loss from exceeding a set threshold by applying a multiplicative factor beyond this limit.*

Bound on Mechanistic Loss

The first method introduced is a bound on the mechanistic loss. This method ensures that the model's predictions do not deviate too far from the mechanistic solutions. A fixed threshold is set, and when the mechanistic loss exceeds this threshold, a multiplicative factor is applied to penalize further deviations. This approach softly constrains the model within a feasible solution space derived from mechanistic constraints, preventing the model from paying an excessive cost in terms of mechanistic loss to improve the data-driven one. To provide a clear view of the described bound on the mechanistic loss, Figure 5.4 shows the bound on mechanistic loss which penalizes solutions that stray too far from the mechanistic loss threshold, encouraging the model to respect mechanistic constraints during training.

Loss Balancing

A loss balancing mechanism was employed to handle the different scales of mechanistic and data-driven losses. This method normalizes each loss dividing it by the exponential average of its previous values, as detailed in [71]. Through this approach, both losses are considered equally during gradient updates, preventing one from outweighing the other during the training process. The loss balancing ensures that the mechanistic loss, which would typically be underrepresented due to its smaller scale, contributes adequately to model optimization alongside the data-driven loss.

Loss Weight Scheduler

Lastly, a dynamic loss weight scheduler was implemented to gradually shift the model's focus between the mechanistic and data-driven tasks over the course of training. For the first phase, the scheduler prioritizes the mechanistic loss, ensuring it starts from a solution closer to the mechanistic model's feasible space. As training progresses, there is a transition phase where the scheduler gradually increases the importance of the data-driven loss until it reaches the final phase, where the data-driven loss has a higher weight, guiding the model toward better predictive performance for the data-driven task. The transitions between the three training phases were defined based on the learning curves of the validation data from the inner K-Fold cross-validation loop (also used for hyperparameter optimization). The transition phase was triggered once the mechanistic loss had converged, which occurred at epoch 30. This phase lasted for 40 epochs, followed by a final phase of 80 epochs, during which the data-driven objective was given higher priority. The loss balance in the initial phase was fixed at 90% mechanistic and 10% data-driven. In contrast, the balance parameter for the final phase was subject to hyperparameter tuning, with a search space ranging from 80% to 100% data-driven loss (and the remainder assigned to the mechanistic loss). For the

sake of clarity, Figure 5.5 illustrates how the scheduler dynamically adjusts the weight of the losses over the course of training, allowing the model to optimize both objectives.

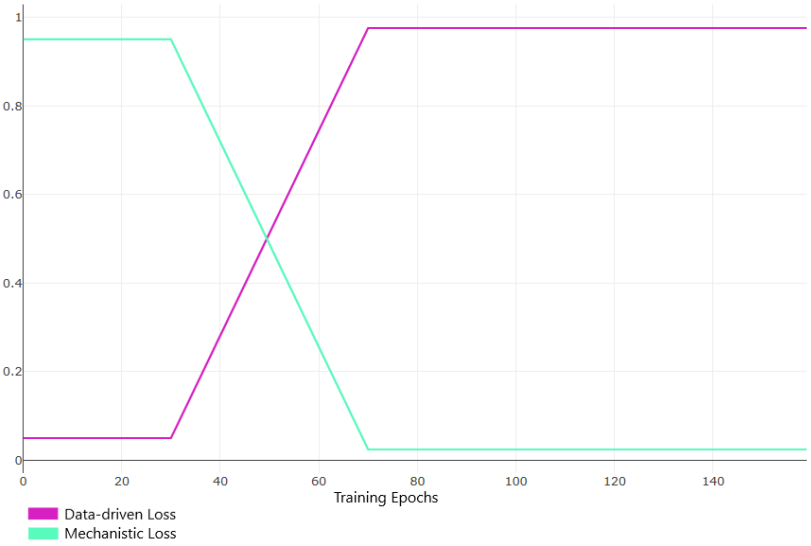

Figure 5.5: *Visualization of the dynamic loss scheduler. The scheduler adjusts the weight of the mechanistic and data-driven losses throughout training, starting with the mechanistic objective and gradually transitioning to prioritize the data-driven objective. This ensures the model initially aligns with mechanistic constraints before focusing on data-driven optimization.*

These methods provide a comprehensive framework for balancing the trade-offs between data-driven accuracy and mechanistic integrity.

5.3.5 Performance evaluation

We adopted the same evaluation pipeline for all our MINN configurations to evaluate the MINN performance and have a fair comparison with the results obtained by [56]. It consists of a dual-loop cross-validation process. The outer loop is a leave-one-out, and in each train loop, there is an inner loop of a k-fold with $k = 5$ to tune the hyper-parameters. The tuning concerns the dimension of the first hidden layer, the learning rate of the NN, the intensity of dropout and $L2$ regularization, and the c constant for the L_1 loss in the case of the MINN-c-balanced. Instead, for the *MINN-scheduler* model, only the hyperparameter controlling the balance between data-driven and mechanistic losses in the final training phase was tuned, where the data-driven component becomes predominant. The initial balance and the epoch marking the transition between phases were kept fixed. For a consistent comparison with the work of [56], we employed identical metrics to evaluate all our experiments: the regression coefficient R^2, the mean absolute error (MAE), the root mean squared error (RMSE)

and the normalized error (NE). As described later, in some of our results, we also report the L_2 as a metric to measure the quality of the predicted flux distribution.

Computational Settings

In this section, we report the runtime of the main experiments conducted in our analysis. The goal is to provide practical insights into the computational cost of each method, guiding practical adoption. All experiments were managed using the ClearML framework. While ClearML introduces a slight overhead due to logging and management features, this does not significantly affect the runtimes provided. For methods presented in Gonçalves et al., quantitative runtime data are not available in the original publication. Therefore, we provide qualitative intervals. For MINN-based approaches and other models we developed, we report quantitative runtime measurements obtained directly from our experimental pipeline. All experiments were executed on a machine equipped with an NVIDIA GeForce RTX 2080 Ti GPU (11GB memory), 20 CPU cores, and 126 GB of RAM. Full software configurations, including the Docker image and package dependencies, are available in the associated code repository to ensure full reproducibility. A summary of all runtime estimates is provided in Table 5.2.

Method	GEM	Evaluation pipeline	Runtime
pFBA	iAF1260	Test only	¡ 1min
NN	NA	Train + Val + Test	¡ 5h
RF	NA	Train + Val + Test	¡ 2h
MINN-c-balanced	iAF1260	Train + Val + Test	44h
MINN-c-balanced	iAF1260 FVA-reduced	Train + Val + Test	24h
MINN-c-balanced	iAF1260 FBA-reduced	Train + Val + Test	12h
MINN-c-balanced	e_coli_core	Train + Val + Test	9.45h
MINN-c-balanced	iNF517 FVA-reduced	Train + Val + Test	14.30h
reservoir + pFBA	iAF1260 FBA-reduced	(Pretrain) + Train + Val + Test	(6.40h) + 6.16h

Table 5.2: *Runtime details for baselines and MINN-based methods*

Hyperparameters details

Table 5.3 shows a detailed list of the hyperparameter search spaces used during the tuning process for each of the methods presented in this work. As described in the main manuscript, hyperparameter optimization is performed in the inner loop of the evaluation pipeline, which follows a 5-fold cross-validation scheme. We used random search to explore the search space. Some hyperparameters were optimized, while others were fixed based on prior knowledge from [56]. Specifically, we fixed the number of epochs at 100, the batch size at 5, and we always used the Adam optimizer. As shown in Table 5.3, we defined different search spaces for the value of the constant c, the parameter balancing the data-driven and mechanistic losses (for the last phase of the

schedulers models), the hidden layer size of the neural network, the dropout rate, the learning rate, and the $L2$ regularization term.

Method	c	Final Loss Balance Weight	Hidden Size	Learning Rate	Dropout Rate	L2
MINN-MSE-base	NA	NA	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}
MINN-unbalanced	NA	NA	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}
MINN-c-balanced	{10, 20, 30, 40, 50}	NA	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}
MINN-bound	{10, 20, 30, 40, 50}	NA	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}
MINN-scheduler	NA	[0.8, 1]	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}
MINN-scheduler-bound	NA	[0.8, 1]	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}
MINN-reservoir + pFBA	NA	NA	{200, 250, 300}	{0.0002, 0.0005, 0.0007}	{0.1, 0.25, 0.5}	{0, 0.0001, 0.001}

Table 5.3: *Hyperparameter search spaces used during tuning for each method. Each row continues across the two subtables. Square brackets denote discrete values, curly brackets indicate intervals.*

5.4 Results and discussion

This work aims to compare the predictive performance of the MINN w.r.t pure ML approaches and mechanistic models such as pFBA. For clarity, we divide all the results and discussions into three groups. The first one (Table 5.4) contains the results of the performance comparison between our approach and the pure ML ones presented in [56]. Here, we evaluate the predictive performance of different methods on the 45 reference fluxes measured in the ISHII dataset. The second one (Table 5.5) includes the results of the comparison between different approaches employed to mitigate the issue of conflicting losses. Here, we compare the performance of the different methods on the measured fluxes and the quality of the predicted flux distribution, which we measure using, as a proxy, L_2. The third group (Table 5.6), instead, compares the results obtained using the MINN-reservoir configuration with those of pFBA.

Lastly, we tested the impact of using different GEMs in the mechanistic layer of the MINN. In particular, when we included the GEM of *Lactococcus lactis* subsp. *cremoris*, a bacterium with an incomplete TCA cycle compared to *E. coli*, we observed a decrease in the quality of the predicted flux distribution. Although the difference was moderate, probably due to the conservation of central carbon metabolism, these results suggest

that the biological relevance of the GEM has an important role in the regularizing effect of the mechanistic layer. A more detailed analysis is available in Section 5.4.4.

5.4.1 MINN to predict measured fluxes

In the first group, as a baseline, we used the MINN architecture with an MSE as L_1 (MINN-MSE-base), as in [46]. As shown in Table 5.4, the results are already comparable with a Random Forest (RF), the best machine learning method in [56], and better than the NN approach. To avoid potential bias from the large discrepancies (up to two orders in magnitude) between the values of the fluxes we are predicting, we replaced the MSE in L_1 with a NE. As detailed in the Section 5.3, we multiply L_1 with a constant c, optimized during the cross-validation. This method, incorporating the c parameter, is referred to as MINN-c-balanced, while the one without this adjustment as MINN-unbalanced. Although the change from MSE to NE ensures a scale-invariant L_1, it also reduces its magnitude, amplifying the conflict between losses as the mechanistic constraints gain more influence. For this reason, the MINN-unbalanced obtained worse results w.r.t. the MINN-MSE-base, but the MINN-c-balanced shows the best results, achieving comparable or better performance than the RF in three out of four metric averages.
Moreover, the MINN-c-balanced shows a reduction in standard deviation. This suggests that the inclusion of biological constraints stabilizes the learning process, reduces overfitting, and leads to more robust and consistent predictions across the 29 leave-one-out splits. In addition, MINN not only learn flux values from data but also derive the entire flux distributions, as done by FBA simulations. These results suggest that MINN is a promising hybrid approach for flux prediction, showing consistent improvements in both average performance and standard deviation across all baselines, including pure mechanistic and ML methods. However, the statistical significance tests detailed in the sections 5.4.4 indicates that the difference between MINN-c-balanced and Random Forest is not statistically significant. This, together with the fact that our analysis was limited to a single microorganism, highlights the need for further investigations to better understand the effectiveness of hybrid approaches in the context of flux prediction. The challenge of predicting multiple fluxes, with significant variability in their values, was effectively addressed by substituting the MSE with an NE for the L_1 and adding the c constant that handled the emerging imbalances. The MINN's flexibility also makes it a powerful tool for integrating multi-omics and potentially other kinds of data with GEMs, enabling its application in diverse systems biology contexts.

5.4.2 MINN to predict a qualitative flux distribution

Regarding the second group of results, we aim to compare different methods to mitigate the issue of the conflicting losses already introduced in Section 5.3. We address this

	ISHII			
Model	R^2	MAE	RMSE	NE
pFBA*	0.823 ± 0.156	0.692 ± 0.733	1.058 ± 1.029	0.381 ± 0.185
NN*	0.967 ± 0.036	0.652 ± 0.945	0.936 ± 1.314	0.338 ± 0.338
RF*	0.970 ± 0.037	0.507 ± 0.804	0.729 ± 1.105	0.271 ± 0.347
MINN-MSE-base	0.950 ± 0.060	0.525 ± 0.525	0.736 ± 0.703	0.287 ± 0.282
MINN-unbalanced	0.951 ± 0.051	0.563 ± 0.739	0.814 ± 1.067	0.325 ± 0.442
MINN-c-balanced	0.950 ± 0.055	0.473 ± 0.480	0.678 ± 0.653	0.272 ± 0.280

Table 5.4: *Comparison of predictive performance between our proposed MINN-based approaches and purely mechanistic and machine learning methods from [56]. Metrics average and standard deviation over 29 leave-one-out splits. All the MINN models were generated using the iAF1260-FVA reduced GEM.*

*results from [56]

problem from two different points of view. First, we act on the mechanistic aspects of the MINN: the reference data. In contrast, the second perspective addresses the optimization process of the MINN, where we apply various hybrid optimization strategies discussed in Section 5.3.4 As a baseline, we employ our best method in terms of prediction performance, hence MINN-c-balanced.

In the first case, we compare it with a configuration of the same MINN-c-balanced which uses different fluxes data, recalculated to be in the solution space of FBA. We described this process in Section 5.3.1. As shown in the first section of Table 5.5, this approach, named MINN-c-balanced *FBA fit*, does not reduce the quality of the fluxes prediction, represented by the four metrics, but it improves theL_2 by reducing its value by a third. The results show that MINN maintains strong predictive performance even when the flux data are not in the FBA solution space. This suggests that its data-driven component can compensate for deviations from mechanistic constraints.

However, using flux data that aligns with the FBA solution space improves the quality of the predicted flux distribution. This correction makes the optimization process easier by alleviating the issue of conflicting losses.

Regarding the hybrid optimization strategies, we compare the MINN-c-balanced with three other methods previously introduced in Section 5.3.4: *MINN-bound*, which penalizes violations of the mechanistic loss that exceed a threshold; *MINN-scheduler*, which gradually shifts the focus from mechanistic to data-driven loss during training; and *MINN-scheduler-bound*, which combines both approaches. From the second section of the table, we observe that the *MINN-c-balanced* model achieves the lowest RMSE, indicating the best performance on the data-driven task. However, this comes at the cost of a higher L_2, suggesting a trade-off where improved performance is achieved at the expense of mechanistic fidelity. On the other hand, the models incorporating a mecha-

nistic bound, such as *MINN-bound* and *MINN-scheduler-bound*, show a small increase in RMSE and a modest reduction in L_2, suggesting a limited effect on improving the trade-off between mechanistic accuracy and data-driven performance. Interestingly, the *MINN-scheduler* model finds a better compromise between these objectives: at the cost of a moderate RMSE worsening, it achieves the lowest L_2 by a consistent margin. This demonstrates the strength of dynamic scheduling in keeping the solution close to the FBA feasible space, without heavily impacting data-driven performance. Interestingly, the balance parameter between the data-driven and mechanistic losses of the last phase, optimized during hyperparameter tuning, converged to values around 90–95% in favor of the data-driven loss, rather than the maximum of 100%, suggesting that the mechanistic component remained beneficial even during the data-driven–oriented phase.

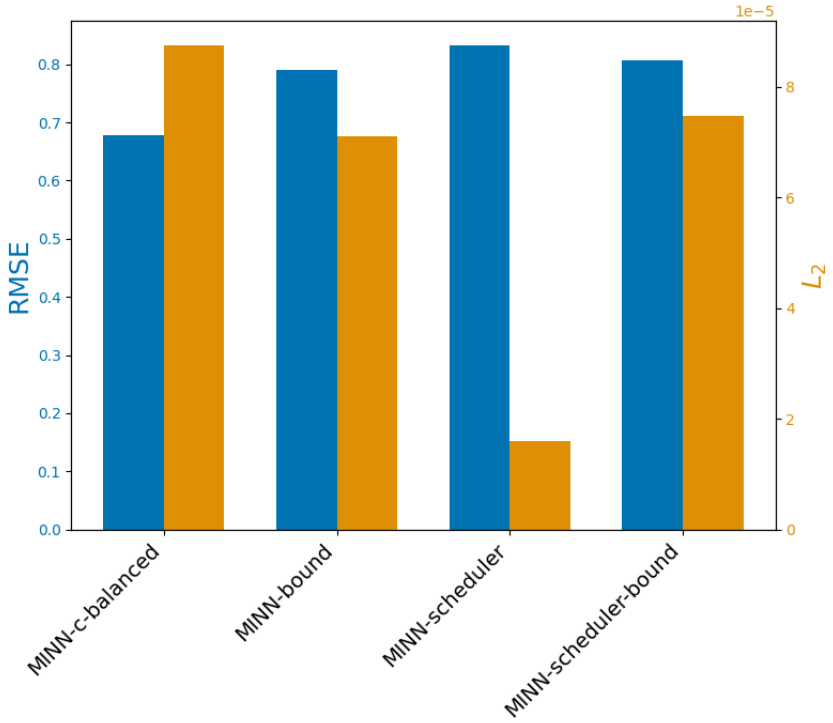

Figure 5.6: *Comparison of different methods based on data-driven task performance (RMSE) and mechanistic fit (L_2 loss), highlighting the trade-off between the two objectives.*

Figure 5.6 provides a visual comparison of the performance of these methods, focusing on the trade-offs between mechanistic fit and data-driven task performance. The *MINN-scheduler* model demonstrates a balanced performance across both objectives, with a moderate decline in data-driven accuracy but a substantial reduction in mechanistic loss, positioning it much closer to the FBA feasible solution. On the other hand,

the models incorporating a mechanistic bound (*MINN-bound* and *MINN-scheduler-bound*) show improvements in mechanistic fit but at a comparable cost in terms of data-driven performance.
These results highlight a clear trade-off between optimizing for mechanistic fidelity and predictive accuracy. While the *MINN-c-balanced* method achieves the lowest RMSE, indicating better performance on the data-driven task, its high mechanistic loss shows that the model prioritizes predictive accuracy over adherence to mechanistic constraints. In contrast, the *MINN-scheduler* method effectively reduces the mechanistic loss, with only a marginal increase in RMSE.

Overall, these results emphasize the need to carefully select hybrid optimization methods based on the specific priorities of the task. In cases where mechanistic accuracy is crucial, dynamic scheduling methods like *MINN-scheduler* provide a balanced solution, allowing the model to gradually adjust the emphasis between mechanistic fidelity and data-driven optimization.
Additionally, as shown in the third section of Table 5.5, we tested a configuration of the MINN, called MINN-divided loss, where the loss is purely data-driven ($L_{MINN} = L_1$), to check if we could further improve the prediction performance at the cost of L_2. The results show that the improvement in terms of prediction performance is marginal w.r.t. the substantial (four orders of magnitude) increase in the L_2 value. This confirms that the regularization effect happens exclusively in the Mechanistic Layer, while the complete L_{MINN} is needed to obtain a qualitative flux distribution as output of the MINN.

	ISHII				
Model	R^2	MAE	RMSE	NE	L_2
MINN-c-balanced	0.950 ± 0.055	0.473 ± 0.480	0.678 ± 0.653	0.272 ± 0.280	$8.75 \cdot 10^{-5} \pm 2.95 \cdot 10^{-4}$
MINN-c-balanced *FBA fit*	0.957 ± 0.061	0.489 ± 0.497	0.706 ± 0.720	0.295 ± 0.403	$2.98 \cdot 10^{-5} \pm 8.75 \cdot 10^{-5}$
MINN-bound	0.949 ± 0.050	0.548 ± 0.556	0.790 ± 0.779	0.308 ± 0.314	$7.1 \cdot 10^{-5} \pm 2.1 \cdot 10^{-4}$
MINN-scheduler	0.949 ± 0.058	0.581 ± 0.823	0.833 ± 1.146	0.299 ± 0.320	$1.60 \cdot 10^{-5} \pm 7.63 \cdot 10^{-5}$
MINN-scheduler-bound	0.946 ± 0.062	0.560 ± 0.551	0.806 ± 0.761	0.304 ± 0.287	$7.47 \cdot 10^{-5} \pm 3.78 \cdot 10^{-4}$
MINN-divided loss	0.951 ± 0.050	0.489 ± 0.471	0.703 ± 0.648	0.281 ± 0.289	0.22 ± 0.29

Table 5.5: *Performance comparison of different methods addressing the issue of conflicting losses. Metrics average and standard deviation over 29 leave-one-out splits. All the models were generated using the iAF1260-FVA reduced GEM.*

5.4.3 MINN-reservoir to improve pFBA predictions

In this third group, we present results for the MINN-reservoir method, which extends the use of MINN beyond applications focused solely on prediction. As introduced by

[46], the MINN-reservoir can be used to generate constraints for a mechanistic model in a data-driven manner. In our case, we trained the MINN-reservoir to predict three arbitrarily selected exchange fluxes *(R_EX_co2_e, R_EX_etoh_e and R_EX_ac_e)* which were then used as additional inputs for pFBA. FBA relies on optimization and is generally more accurate in predicting metabolic shifts caused by gene knockouts only after the microbial population has undergone an adaptation period. It is therefore of interest to explore whether the data-driven insights provided by the MINN-reservoir can help overcome this limitation in the case of the newly generated knockout strains from the ISHII dataset. Our baseline consists of a standard pFBA model that uses only the uptake rates of glucose (*R_EX_glc__D_e*) and oxygen (*R_EX_o2_e*) as inputs. We compare it with pFBA when it is provided with the same inputs plus the three extra constraints generated by the MINN-reservoir. This setup reflects a realistic use case, in which multi-omics data are available for all samples, while fluxomics data are only partially available. In this context, the MINN-reservoir allows us to estimate missing input fluxes from omics measurements, avoiding the need to experimentally quantify them for every sample.

	ISHII			
Model	R^2	MAE	RMSE	NE
pFBA	0.892 ± 0.127	0.496 ± 0.353	0.836 ± 0.625	0.306 ± 0.367
MINN-reservoir + pFBA	0.910 ± 0.091	0.445 ± 0.261	0.740 ± 0.421	0.253 ± 0.159

Table 5.6: *Performance comparison between the standard pFBA and MINN-reservour + pFBA. The results evaluate the effectiveness of the MINN-reservoir approach to enrich the input for pFBA in comparison to standard pFBA. Metrics average and standard deviation over 29 leave-one-out splits. The MINN-reservoir model was generated using the iAF1260-FBA reduced GEM.*

The advantage of this neural network-based approach is that, once trained, it can generate additional inputs for pFBA based on initial conditions alone, enabling a more informative and automated modeling pipeline. As shown in Table 5.6, the enhanced version (MINN-reservoir + pFBA) slightly improves the performance in terms of average metrics across the 47 fluxes. However, the most evident benefit is the reduction in the standard deviation, which indicates that the model produces more stable and consistent predictions across the 29 leave-one-out splits.

5.4.4 Additional Results

Other GEMs comparison

Here we present the results of the analysis to explore the role of the GEM in the models' performance. As shown in Table 5.7, we divided the results in two parts. The first one contains different GEMs in terms of dimension. Here the GEM is always iAF1260, but reduced in different ways described in details in the Section "GEM preparation". We also built a MINN with the *E. coli* core model (e_coli_core), to test the smallest version available on BiGG `http://bigg.ucsd.edu/`. The GEM's dimension can affect the complexity of the NN block in the MINN. Having a layer with many neurons can cause both overfitting and a higher computational time. At the same time, excessively reducing the GEM can decrease the flexibility in the optimization of FBA constraints and make the GEM less representative of the experimental context considered. The first section of Table 5.7 shows how the FVA reduction has better performances, in terms of metrics and L_2, and lower computational time (24h vs 44h) than the full GEM. FBA reduction, instead, performs slightly worse than FVA, but it halves the computational time. On the other hand, the e_coli_core GEM performs drastically worse than all the others, making this GEM unfit for this task.

The results show a trade-off between GEM size and computational efficiency. A stricter reduction, such as FBA reduction, shortens the computational time, but at the cost of slightly worse metrics and less reliable flux distribution. While this compromise may not be ideal for small models, it can be helpful for large GEMs, such as yeast or microbial communities, where computational feasibility is critical. In such cases, sacrificing some predictive power in exchange for reasonable runtimes can be a proper trade-off.
Additionally, the poor performance of the e_coli_core model highlights the need for an adequate dimension of the GEM, reinforcing the idea that excessively small models may not represent the experimental context of interest.

	ISHII				
GEM	R^2	MAE	RMSE	NE	L_2
iAF1260	0.818 ± 0.670	0.602 ± 0.653	1.084 ± 0.813	0.417 ± 0.268	$4.05 \cdot 10^{-5} \pm 1.57 \cdot 10^{-4}$
iAF1260 *FVA-reduced*	0.950 ± 0.055	0.473 ± 0.480	0.678 ± 0.653	0.272 ± 0.280	$8.75 \cdot 10^{-5} \pm 2.95 \cdot 10^{-4}$
iAF1260 *FBA-reduced*	0.950 ± 0.048	0.509 ± 0.518	0.730 ± 0.719	0.289 ± 0.295	$1.26 \cdot 10^{-4} \pm 2.84 \cdot 10^{-4}$
e_coli_core	0.061 ± 0.099	4.647 ± 9.959	19.46 ± 65.30	7.584 ± 26.70	$6.04 \cdot 10^{5} \pm 2.55 \cdot 10^{6}$
iAF1260 *FVA-reduced*	0.956 ± 0.056	0.512 ± 0.596	0.759 ± 0.815	0.285 ± 0.337	$4.27 \cdot 10^{-5} \pm 1.3 \cdot 10^{-4}$
iNF517 *FVA-reduced*	0.954 ± 0.057	0.546 ± 0.612	0.801 ± 0.855	0.304 ± 0.358	$5.24 \cdot 10^{-5} \pm 1.02 \cdot 10^{-4}$

Table 5.7: *Performance comparison between different GEMs. Metrics average and standard deviation over 29 leave-one-out splits.*

The second part includes a comparison between two GEMs representing two dif-

ferent bacteria, namely *E.coli* and *Lactococcus lactis subsp. cremoris*. The aim of this analysis is to explore how relevant the nature of the GEM is in the MINN architecture. We want to investigate if the regularization that improves the predictive power of the MINN w.r.t. a classical ML approaches is based on a relevant biological information injected in the model through the mechanistic layer, or it's simply a random type of regularization such as Dropout [158]. Since the *L.cremoris* GEM (iNF517) does not have some of the reactions present in the ISHII dataset, in order to have a fair comparison with the *E.coli* GEM (iAF1260), we reduced the number of fluxes to only those in common between *L.cremoris* and the ISHII dataset. As expected, using a GEM which belongs to another bacterium worsen the prediction performance and also the quality of the predicted flux distribution.

However, the difference in performance between the iAF1260 *FVA-reduced* and iNF517 *FVA-reduced* GEMs is not particularly large. One possible explanation is that the measured fluxes in ISHII dataset belong to the central carbon metabolism, which is highly conserved between both bacteria, reducing the impact of GEM differences. Another factor could be the neural network data-driven component, which may help compensate for discrepancies between GEMs, reducing their effect on predictive performance.

While further investigation is needed, these results suggest that the nature of the GEM plays an important role in the MINN framework. The mechanistic layer likely contributes with biologically relevant information beyond acting as a generic regularization mechanism.

Tests of Significance

To assess whether the performance differences observed in our experiments are statistically significant or potentially due to chance, we conducted a series of Wilcoxon signed-rank tests. For each of the main models comparison, we computed four separate p-values, one for each evaluation metrics used. To combine these into a single measure of significance, we employed Fisher's method, as implemented in the scipy library. We first compared MINN-c-balanced to pFBA (Table 2 of the main manuscript). The final combined p-value resulting from Fisher's method is $6.09 \cdot 10^{-18}$, allowing us to confidently state that the performance difference is statistically significant. We then compared MINN-c-balanced to the pure neural network (also Table 2), obtaining a combined p-value of 0.0009, which also confirms statistical significance. In contrast, the comparison between MINN-c-balanced and the Random Forest model showed a final combined p-value of 0.97, meaning we cannot reject the null hypothesis and conclude that there is a significant difference in performance between the two models. Lastly, for the comparison between the reservoir model and pFBA reported in Table 4 of the main manuscript, the final combined p-value is $2.34 \cdot 10^{-19}$, strongly indicating a statistically significant improvement in performance. These results highlight the

potential of hybrid approaches such as MINN-based methods, but also indicate that their advantage over traditional ML models like Random Forests may vary depending on the context or the dataset and remains an open question for future work.

5.5 Concluding remarks

Faure et al. [46] introduced a new hybrid architecture, which incorporates a Genome-Scale Metabolic Model in a neural network structure, and used it to predict *E. coli* growth rates in different growth media. In our work, we adapted this framework into a Metabolic-Informed Neural Network, which also uses multi-omics data as input, and tested it with a more challenging task: predicting metabolic fluxes for different E. coli single-gene KO strains grown in minimal glucose medium. The MINN showed improved performance compared to traditional machine learning, and the mechanistic component showed a regularizing effect on the predictions. We then explored the effect of different components of the architecture on the predictions and their accuracy. Finally, we tested the ability of the MINN to reconcile data and models in a flexible way, even in scenarios where data-driven and mechanistic optimization show a trade-off. To achieve this, we suggested different hybrid optimization strategies.
We chose a naive multi-omics integration approach, such as early concatenation. While the predictive performances are encouraging, as discussed in Chapter 3, mixed integration strategies are often to be preferred and they could be tested to further improve the prediction power of a MINN-based method. Moreover, GPR rules could be leveraged to more directly link omics data to metabolic fluxes, for example by incorporating into future versions of MINN a loss term that maximizes the correlation between expression levels and predicted fluxes [7].

Additionally, more work is needed for assessing the interpretability of the flux distribution. As a first step in this direction, in our simulation we kept track of L_2, as a proxy for how much the predicted metabolic profile complies with the theoretical assumptions of FBA. Moreover , this novel framework has been tested only for *E. coli*, the classical "work-horse" of microbial physiology. We hope the promising result of these works will prompt the creation of suitable datasets to apply these techniques to other microorganisms and to more diverse scenarios, in which secondary metabolism plays a bigger role and the information provided by the GEM is potentially even more effective in complementing the data-driven learning. In addition, the mechanistic component of MINN holds strong potential to improve predictions in more complex systems, such as eukaryotic cells or microbial communities, and the architecture of MINN is designed to scale seamlessly to these settings.
Although this was not the most favorable scenario for FBA, the results of the MINN-reservoir highlight its potential as a promising strategy to enable the full integration of FBA into a machine learning framework, effectively combining the advantages of both

mechanistic and data-driven approaches.
Finally, with this work, we provide a practical guide for choosing the most suitable MINN configuration based on the modeling objective. If the goal is only to achieve high predictive performance, the standard MINN configuration is sufficient. When the aim is to improve the quality of the predicted flux distribution, the optimization strategies presented here can reduce the mechanistic constraints violation. Lastly, if the objective is to enrich mechanistic models with additional inputs, the MINN-reservoir offers a viable solution to generate constraints in a data-driven way, while keeping the structure and interpretability of classical FBA.

The code used for the analyses presented in this chapter is available at `https://github.com/gabrieletaz/MINN`.

The author of this PhD thesis is responsible for the following contributions presented in this chapter:

IV/1. Contributed to conceptualization and design of the work: MINN architecture for integrating multi omics into Genome Scale metabolic modeling.

IV/2. Literature survey regarding the hybrid modeling methods in the Related Work section.

IV/3. Implementation of the MINN and MINN-reservoir relative experiments.

IV/4. Implementation of the code used in our analysis that regards the MINN and the MINN-reservoir.

Bibliography

[1] Nasrullah Abbasi, Nizamullah FNU, Shah Zeb, Muhammad Fahad, and Muhammad Umer Qayyum. Machine learning models for predicting susceptibility to infectious diseases based on microbiome profiles. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, 3(4):35–47, December 2024.

[2] Debabrata Acharya and Anirban Mukhopadhyay. A comprehensive review of machine learning techniques for multi-omics data integration: challenges and applications in precision oncology. *Briefings in Functional Genomics*, 23(5):549–560, April 2024.

[3] Claire Adam-Bourdarios, Glen Cowan, Cecile Germain, Isabelle Guyon, Balazs Kegl, and David Rousseau. Learning to discover: the higgs boson machine learning challenge - documentation. *c*, 2014.

[4] Claire Adam-Bourdarios, Glen Cowan, Cécile Germain, Isabelle Guyon, Balàzs Kégl, and David Rousseau. The Higgs boson machine learning challenge. In Glen Cowan, Cécile Germain, Isabelle Guyon, Balázs Kégl, and David Rousseau, editors, *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, volume 42 of *Proceedings of Machine Learning Research*, pages 19–55, Montreal, Canada, 13 Dec 2015. PMLR.

[5] Nigatu Adossa, Sofia Khan, Kalle T. Rytkönen, and Laura L. Elo. Computational strategies for single-cell multi-omics integration. *Computational and Structural Biotechnology Journal*, 19:2588–2596, 2021.

[6] Serhat Al, Fatma Uysal Ciloglu, Aytac Akcay, and Ahmet Koluman. Machine learning models for prediction of escherichia coli o157:h7 growth in raw ground beef at different storage temperatures. *Meat Science*, 210:109421, April 2024.

[7] Norah Alghamdi, Wennan Chang, Pengtao Dang, Xiaoyu Lu, Changlin Wan, Silpa Gampala, Zhi Huang, Jiashi Wang, Qin Ma, Yong Zang, Melissa Fishel, Sha Cao, and Chi Zhang. A graph neural network model to estimate cell-wise metabolic flux using single-cell RNA-seq data. *Genome Research*, 31(10):1867–1884, October 2021.

[8] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7), July 2016.

[9] Athanasios Antonakoudis, Rodrigo Barbosa, Pavlos Kotidis, and Cleo Kontoravdi. The era of big data: Genome-scale modelling meets machine learning. *Computational and Structural Biotechnology Journal*, 18:3287–3300, 2020.

[10] Maciek R. Antoniewicz. A guide to metabolic flux analysis in metabolic engineering: Methods, tools and applications. *Metabolic Engineering*, 63:2–12, January 2021.

[11] Francis Bach and Michael Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 03 2003.

[12] Ronny Kohavi Barry Becker. Adult, 1996.

[13] Scott A. Becker and Bernhard O. Palsson. Context-Specific Metabolic Networks Are Consistent with Experiments. *PLOS Computational Biology*, 4(5):e1000082, May 2008. Publisher: Public Library of Science.

[14] Viv Bewick, Liz Cheek, and Jonathan Ball. *Critical Care*, 9(1):112, 2005.

[15] Yangzom D. Bhutia, Ellappan Babu, Puttur D. Prasad, and Vadivel Ganapathy. The amino acid transporter slc6a14 in cancer and its potential use in chemotherapy. *Asian Journal of Pharmaceutical Sciences*, 9(6):293–303, December 2014.

[16] Ioana Bica, Petar Velickovic, and Hui Xiao. Multi-omics data integration using cross-modal neural networks. In *The European Symposium on Artificial Neural Networks*, pages 385,390, 2018.

[17] G. BISSETTE, F. J. SEIDLER, C. B. NEMEROFF, and T. A. SLOTKIN. High affinity choline transporter status in alzheimer's disease tissue from rapid autopsya. *Annals of the New York Academy of Sciences*, 777(1):197–204, January 1996.

[18] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–21, 2024.

[19] Mitja Briscik, Marie-Agnès Dillies, and Sébastien Déjean. Improvement of variables interpretability in kernel PCA. *BMC Bioinformatics*, 24(1), July 2023.

[20] Mitja Briscik, Mohamed Heimida, and Sébastien Déjean. kpcaig: Variables interpretability with kernel pca, 2024.

[21] Mitja Briscik, Gabriele Tazza, László Vidács, Marie-Agnès Dillies, and Sébastien Déjean. Supervised multiple kernel learning approaches for multi-omics data integration. *BioData Mining*, 17(1):1–25, December 2024. Number: 1 Publisher: BioMed Central.

[22] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[23] Céline Brouard, Raphaël Mourad, and Nathalie Vialaneix. Should we really use graph neural networks for transcriptomic prediction? *Briefings in Bioinformatics*, 25(2), January 2024.

[24] Frank J Bruggeman, Robert Planqué, Douwe Molenaar, and Bas Teusink. Searching for principles of microbial physiology. *FEMS Microbiology Reviews*, 44(6):821–844, November 2020.

[25] M. Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, Yufan He, Ziyue Xu, Ali Hatamizadeh, Andriy Myronenko, Wentao Zhu, Yun Liu, Mingxin Zheng, Yucheng Tang, Isaac Yang, Michael Zephyr, Behrooz Hashemian, Sachidanand Alle, Mohammad Zalbagi Darestani, Charlie Budd, Marc Modat, Tom Vercauteren, Guotai Wang, Yiwen Li, Yipeng Hu, Yunguan Fu, Benjamin Gorman, Hans Johnson, Brad Genereaux, Barbaros S. Erdal, Vikash Gupta, Andres Diaz-Pinto, Andre Dourson, Lena Maier-Hein, Paul F. Jaeger, Michael Baumgartner, Jayashree Kalpathy-Cramer, Mona Flores, Justin Kirby, Lee A. D. Cooper, Holger R. Roth, Daguang Xu, David Bericat, Ralf Floca, S. Kevin Zhou, Haris Shuaib, Keyvan Farahani, Klaus H. Maier-Hein, Stephen Aylward, Prerna Dogra, Sebastien Ourselin, and Andrew Feng. Monai: An open-source framework for deep learning in healthcare, 2022.

[26] Antje Chang, Lisa Jeske, Sandra Ulbrich, Julia Hofmann, Julia Koblitz, Ida Schomburg, Meina Neumann-Schaal, Dieter Jahn, and Dietmar Schomburg. BRENDA, the ELIXIR core data resource in 2021: new developments and updates. *Nucleic Acids Research*, 49(D1):D498–D508, January 2021.

[27] J. Chen, E. E. Bardes, B. J. Aronow, and A. G. Jegga. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic Acids Research*, 37(Web Server):W305–W311, May 2009.

[28] Pengguang Chen, Shu Liu, Hengshuang Zhao, Xingquan Wang, and Jiaya Jia. Gridmask data augmentation, 2020.

[29] Yu Chen, Johan Gustafsson, Albert Tafur Rangel, Mihail Anton, Iván Domenzain, Cheewin Kittikunapong, Feiran Li, Le Yuan, Jens Nielsen, and Eduard J. Kerkhoven. Reconstruction, simulation and analysis of enzyme-constrained metabolic models using GECKO Toolbox 3.0. *Nature Protocols*, 19(3):629–667, March 2024. Publisher: Nature Publishing Group.

[30] Davide Chicco, Fabio Cumbo, and Claudio Angione. Ten quick tips for avoiding pitfalls in multi-omics data integration analyses. *PLOS Computational Biology*, 19(7):e1011224, July 2023.

[31] Ilseung Cho and Martin J. Blaser. The human microbiome: at the interface of health and disease. *Nature Reviews Genetics*, 13(4):260–270, March 2012.

[32] Cláudia M. Coutinho-Camillo, M. Mitzi Brentani, Ossamu Butugan, Humberto Torloni, and Maria A. Nagai. Relaxation of imprinting of igfii gene in juvenile nasopharyngeal angiofibromas. *Diagnostic Molecular Pathology*, 12(1):57–62, March 2003.

[33] Glen Cowan, Kyle Cranmer, Eilam Gross, and Ofer Vitells. Asymptotic formulae for likelihood-based tests of new physics. *The European Physical Journal C*, 71(2), February 2011.

[34] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019.

[35] Hengmi Cui, Marcia Cruz-Correa, Francis M. Giardiello, David F. Hutcheon, David R. Kafonek, Sheri Brandenburg, Yiqian Wu, Xiaobing He, Neil R. Powe, and Andrew P. Feinberg. Loss of igf2 imprinting: A potential marker of colorectal cancer risk. *Science*, 299(5613):1753–1755, March 2003.

[36] Tiantain Cui, Linlin Yang, Yunxia Ma, Iver Petersen, and Yuan Chen. Desmocollin 3 has a tumor suppressive activity through inhibition of akt pathway in colorectal cancer. *Experimental Cell Research*, 378(2):124–130, May 2019.

[37] Christopher Culley, Supreeta Vijayakumar, Guido Zampieri, and Claudio Angione. A mechanism-aware and multiomic machine-learning pipeline characterizes yeast cell growth. *Proceedings of the National Academy of Sciences*, 117(31):18869–18879, July 2020.

[38] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3), July 2022.

[39] David Dai, Nicholas Horvath, and Jeffrey Varner. Dynamic sequence specific constraint-based modeling of cell-free protein synthesis. *Processes*, 6(8):132, August 2018.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[41] Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space, 2017.

[42] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

[43] Ioannis G. Diamataris, Loukas D. Peristeras, Konstantinos D. Papavasileiou, Vasilios S. Melissas, and Georgios C. Boulougouris. Statistical Inference of Rate Constants in Chemical and Biochemical Reaction Networks Using an "Inverse" Event-Driven Kinetic Monte Carlo Method. *The Journal of Physical Chemistry B*, 127(42):9132–9143, October 2023. Publisher: American Chemical Society.

[44] Ali Ebrahim, Joshua A Lerman, Bernhard O Palsson, and Daniel R Hyduke. Cobrapy: Constraints-based reconstruction and analysis for python. *BMC Systems Biology*, 7(1), August 2013.

[45] Ibrahim E. Elsemman, Angelica Rodriguez Prado, Pranas Grigaitis, Manuel Garcia Albornoz, Victoria Harman, Stephen W. Holman, Johan Van Heerden, Frank J. Bruggeman, Mark M. M. Bisschops, Nikolaus Sonnenschein, Simon Hubbard, Rob Beynon, Pascale Daran-Lapujade, Jens Nielsen, and Bas Teusink. Whole-cell modeling in yeast predicts compartment-specific proteome constraints that drive metabolic strategies. *Nature Communications*, 13(1):801, February 2022.

[46] Léon Faure, Bastien Mollet, Wolfram Liebermeister, and Jean-Loup Faulon. A neural-mechanistic hybrid approach improving the predictive power of genome-scale metabolic models. *Nature Communications*, August 2023.

[47] Adam M Feist, Christopher S Henry, Jennifer L Reed, Markus Krummenacker, Andrew R Joyce, Peter D Karp, Linda J Broadbelt, Vassily Hatzimanikatis, and Bernhard Ø Palsson. A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular Systems Biology*, (1):121, January 2007.

[48] Dylan Feldner-Busztin, Panos Firbas Nisantzis, Shelley Jane Edmunds, Gergely Boza, Fernando Racimo, Shyam Gopalakrishnan, Morten Tønsberg Limborg, Leo

Lahti, and Gonzalo G de Polavieja. Dealing with dimensionality: the application of machine learning to multi-omics data. *Bioinformatics*, 39(2), January 2023.

[49] Nicolas A. L. Flahaut, Anne Wiersma, Bert van de Bunt, Dirk E. Martens, Peter J. Schaap, Lolke Sijtsma, Vitor A. Martins dos Santos, and Willem M. de Vos. Genome-scale metabolic model for Lactococcus lactis MG1363 and its application to the analysis of flavor formation. *Applied Microbiology and Biotechnology*, 97(19):8729–8739, October 2013. Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 19 Publisher: Springer Berlin Heidelberg.

[50] Stefan Garczyk, Saskia von Stillfried, Wiebke Antonopoulos, Arndt Hartmann, Michael G. Schrauder, Peter A. Fasching, Tobias Anzeneder, Andrea Tannapfel, Yavuz Ergonenc, Ruth Knuchel, Michael Rose, and Edgar Dahl. Agr3 in breast cancer: Prognostic impact and suitable serum-based biomarker for early cancer detection. *PLOS ONE*, 10(4):e0122106, April 2015.

[51] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, May 2002.

[52] David Gomez-Cabrero, Imad Abugessaisa, Dieter Maier, Andrew Teschendorff, Matthias Merkenschlager, Andreas Gisel, Esteban Ballestar, Erik Bongcam-Rudloff, Ana Conesa, and Jesper Tegnér. Data integration in the era of omics: current and future challenges. *BMC Systems Biology*, 8(Suppl 2):I1, 2014.

[53] Mehmet Gönen and Ethem Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, page 352–359. ACM Press, 2008.

[54] Mehmet Gönen and Ethem Alpaydın. Localized algorithms for multiple kernel learning. *Pattern Recognition*, 46(3):795–807, March 2013.

[55] Ping Gong, Lei Cheng, Zhiyuan Zhang, Ao Meng, Enshuo Li, Jie Chen, and Longzhen Zhang. Multi-omics integration method based on attention deep learning network for biomedical data classification. *Computer Methods and Programs in Biomedicine*, 231:107377, April 2023.

[56] Daniel M. Gonçalves, Rui Henriques, and Rafael S. Costa. Predicting metabolic fluxes from omics data via machine learning: Moving from knowledge-driven towards data-driven approaches. *Computational and Structural Biotechnology Journal*, pages 4960–4973, January 2023.

[57] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial

nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[58] Pranas Grigaitis, Brett G. Olivier, Tomas Fiedler, Bas Teusink, Ursula Kummer, and Nadine Veith. Protein cost allocation explains metabolic strategies in Escherichia coli. *Journal of Biotechnology*, 327:54–63, February 2021.

[59] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 507–520. Curran Associates, Inc., 2022.

[60] Lionel Guidi, Samuel Chaffron, Lucie Bittner, Damien Eveillard, Abdelhalim Larhlimi, Simon Roux, Youssef Darzi, Stephane Audic, Léo Berline, Jennifer R. Brum, Luis Pedro Coelho, Julio Cesar Ignacio Espinoza, Shruti Malviya, Shinichi Sunagawa, Céline Dimier, Stefanie Kandels-Lewis, Marc Picheral, Julie Poulain, Sarah Searson, Lars Stemmann, Fabrice Not, Pascal Hingamp, Sabrina Speich, Mick Follows, Lee Karp-Boss, Emmanuel Boss, Hiroyuki Ogata, Stephane Pesant, Jean Weissenbach, Patrick Wincker, Silvia G. Acinas, Peer Bork, Colomban de Vargas, Daniele Iudicone, Matthew B. Sullivan, Jeroen Raes, Eric Karsenti, Chris Bowler, and Gabriel Gorsky. Plankton networks driving carbon export in the oligotrophic ocean. *Nature*, 532(7600):465–470, February 2016.

[61] Vanesa Gómez-Martínez, Francisco J. Lara-Abelenda, Pablo Peiro-Corbacho, David Chushig-Muzo, Conceicao Granja, and Cristina Soguero-Ruiz. Lm-igtd: a 2d image generator for low-dimensional and mixed-type tabular data to leverage the potential of convolutional neural networks, 2024.

[62] Charles R. Haggart, Jennifer A. Bartell, Jeffrey J. Saucerman, and Jason A. Papin. *Whole-Genome Metabolic Network Reconstruction and Constraint-Based Modeling*, page 411–433. Elsevier, 2011.

[63] Zongbo Han, Fan Yang, Junzhou Huang, Changqing Zhang, and Jianhua Yao. Multimodal dynamics: Dynamical fusion for trustworthy multimodal classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20675–20685, 2022.

[64] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon Hare. Fmix: Enhancing mixed sample data augmentation, 2020.

[65] Ramin Hasibi, Tom Michoel, and Diego A. Oyarzún. Integration of graph neural networks and genome-scale metabolic models for predicting gene essentiality. *npj Systems Biology and Applications*, 10(1), March 2024.

[66] Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastián N. Mendoza, Anne Richelle, Almut Heinken, Hulda S. Haraldsdóttir, Jacek Wachowiak, Sarah M. Keating, Vanja Vlasov, Stefania Magnusdóttir, Chiam Yu Ng, German Preciat, Alise Žagare, Siu H. J. Chan, Maike K. Aurich, Catherine M. Clancy, Jennifer Modamio, John T. Sauls, Alberto Noronha, Aarash Bordbar, Benjamin Cousins, Diana C. El Assal, Luis V. Valcarcel, Iñigo Apaolaza, Susan Ghaderi, Masoud Ahookhosh, Marouen Ben Guebila, Andrejs Kostromins, Nicolas Sompairac, Hoai M. Le, Ding Ma, Yuekai Sun, Lin Wang, James T. Yurkovich, Miguel A. P. Oliveira, Phan T. Vuong, Lemmer P. El Assal, Inna Kuperstein, Andrei Zinovyev, H. Scott Hinton, William A. Bryant, Francisco J. Aragón Artacho, Francisco J. Planes, Egils Stalidzans, Alejandro Maass, Santosh Vempala, Michael Hucka, Michael A. Saunders, Costas D. Maranas, Nathan E. Lewis, Thomas Sauter, Bernhard Ø. Palsson, Ines Thiele, and Ronan M. T. Fleming. Creation and analysis of biochemical constraint-based models using the cobra toolbox v.3.0. *Nature Protocols*, 14(3):639–702, February 2019.

[67] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty, 2019.

[68] David Henriques, Romain Minebois, Sebastián N. Mendoza, Laura G. Macías, Roberto Pérez-Torrado, Eladio Barrio, Bas Teusink, Amparo Querol, and Eva Balsa-Canto. A Multiphase Multiobjective Dynamic Genome-Scale Model Shows Different Redox Balancing among Yeast Species of the Saccharomyces Genus in Fermentation. *mSystems*, 6(4):e00260–21, August 2021. Publisher: American Society for Microbiology.

[69] Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules, 2019.

[70] R Hrstka, R Nenutil, A Fourtouna, M M Maslon, C Naughton, S Langdon, E Murray, A Larionov, K Petrakova, P Muller, M J Dixon, T R Hupp, and B Vojtesek. The pro-metastatic protein anterior gradient-2 predicts poor prognosis in tamoxifen-treated breast cancers. *Oncogene*, 29(34):4838–4847, June 2010.

[71] Hanzhang Hu, Debadeepta Dey, Martial Hebert, and J. Andrew Bagnell. Learning Anytime Predictions in Neural Networks via Adaptive Loss Balancing, May 2018.

[72] Sijia Huang, Kumardeep Chaudhary, and Lana X. Garmire. More is better: Recent progress in multi-omics data integration methods. *Frontiers in Genetics*, 8, June 2017.

[73] Hiroshi Inoue. Data augmentation by pairing samples for images classification, 2018.

[74] Md. Ifraham Iqbal, Md. Saddam Hossain Mukta, Ahmed Rafi Hasan, and Salekul Islam. A dynamic weighted tabular method for convolutional neural networks. *IEEE Access*, 10:134183–134198, 2022.

[75] Nobuyoshi Ishii, Kenji Nakahigashi, Tomoya Baba, Martin Robert, Tomoyoshi Soga, Akio Kanai, Takashi Hirasawa, Miki Naba, Kenta Hirai, Aminul Hoque, Pei Yee Ho, Yuji Kakazu, Kaori Sugawara, Saori Igarashi, Satoshi Harada, Takeshi Masuda, Naoyuki Sugiyama, Takashi Togashi, Miki Hasegawa, Yuki Takai, Katsuyuki Yugi, Kazuharu Arakawa, Nayuta Iwata, Yoshihiro Toya, Yoichi Nakayama, Takaaki Nishioka, Kazuyuki Shimizu, Hirotada Mori, and Masaru Tomita. Multiple high-throughput analyses monitor the response of e. coli to perturbations. *Science*, 316(5824):593–597, April 2007.

[76] Shu-Heng Jiang, Li-Li Zhu, Man Zhang, Rong-Kun Li, Qin Yang, Jiang-Yu Yan, Ce Zhang, Jian-Yu Yang, Fang-Yuan Dong, Miao Dai, Li-Peng Hu, Jun Li, Qing Li, Ya-Hui Wang, Xiao-Mei Yang, Yan-Li Zhang, Hui-Zhen Nie, Lei Zhu, Xue-Li Zhang, Guang-Ang Tian, Xiao-Xin Zhang, Xiao-Yan Cao, Ling-Ye Tao, Shan Huang, Yong-Sheng Jiang, Rong Hua, Kathy Qian Luo, Jian-Ren Gu, Yong-Wei Sun, Shangwei Hou, and Zhi-Gang Zhang. Gabrp regulates chemokine signalling, macrophage recruitment and tumour progression in pancreatic cancer through tuning kcnn4-mediated ca2+ signalling in a gaba-independent manner. *Gut*, 68(11):1994–2006, March 2019.

[77] Yang Jiang, Jinpeng Zhou, Peng Luo, Huiling Gao, Yanju Ma, Yin-Sheng Chen, Long Li, Dan Zou, Ye Zhang, and Zhitao Jing. Prosaposin promotes the proliferation and tumorigenesis of glioma through toll-like receptor 4 (tlr4)-mediated nf-kb signaling pathway. *EBioMedicine*, 37:78–90, November 2018.

[78] Amit U. Joshi, Lauren D. Van Wassenhove, Kelsey R. Logas, Paras S. Minhas, Katrin I. Andreasson, Kenneth I. Weinberg, Che-Hong Chen, and Daria Mochly-Rosen. Aldehyde dehydrogenase 2 activity and aldehydic load contribute to neuroinflammation and alzheimer's disease related pathology. *Acta Neuropathologica Communications*, 7(1), December 2019.

[79] Fredrik H Karlsson, Valentina Tremaroli, Intawat Nookaew, Göran Bergström, Carl Johan Behre, Björn Fagerberg, Jens Nielsen, and Fredrik Bäckhed. Gut

metagenome in european women with normal, impaired and diabetic glucose control. *Nature*, 498(7452):99–103, June 2013.

[80] Ziynet Nesibe Kesimoglu and Serdar Bozdag. Supreme: multiomics data integration using graph convolutional networks. *NAR Genomics and Bioinformatics*, 5(2), March 2023.

[81] Minseung Kim, Navneet Rai, Violeta Zorraquino, and Ilias Tagkopoulos. Multiomics integration accurately predicts cellular state in unexplored conditions for escherichia coli. *Nature Communications*, 7(1), October 2016.

[82] Yeji Kim, Gi Bae Kim, and Sang Yup Lee. Machine learning applications in genome-scale metabolic modeling. *Current Opinion in Systems Biology*, 25:42–49, March 2021.

[83] Zachary A. King, Andreas Dräger, Ali Ebrahim, Nikolaus Sonnenschein, Nathan E. Lewis, and Bernhard O. Palsson. Escher: A web application for building, sharing, and embedding data-rich visualizations of biological pathways. *PLOS Computational Biology*, 11(8):e1004321, August 2015.

[84] Zachary A. King, Justin Lu, Andreas Dräger, Philip Miller, Stephen Federowicz, Joshua A. Lerman, Ali Ebrahim, Bernhard O. Palsson, and Nathan E. Lewis. BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, 44(D1):D515–D522, January 2016.

[85] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[86] Arun D. Kulkarni. Fuzzy convolution neural networks for tabular data classification. *IEEE Access*, 12:151846–151855, 2024.

[87] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. *FeatMatch: Feature-Based Augmentation for Semi-supervised Learning*, page 479–495. Springer International Publishing, 2020.

[88] Gert R. G. Lanckriet, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. *Kernel-Based Integration of Genomic Data Using Semidefinite Programming*, chapter 1, page 231–260. The MIT Press, July 2004.

[89] Rosamaria Lappano and Marcello Maggiolini. Role of the g protein-coupled receptors in cancer and stromal cells: From functions to novel therapeutic perspectives. *Cells*, 12(4):626, February 2023.

[90] Emmanuelle Le Chatelier, Trine Nielsen, Junjie Qin, Edi Prifti, Falk Hildebrand, Gwen Falony, Mathieu Almeida, Manimozhiyan Arumugam, Jean-Michel Batto, Sean Kennedy, Pierre Leonard, Junhua Li, Kristoffer Burgdorf, Niels Grarup, Torben Jørgensen, Ivan Brandslund, Henrik Bjørn Nielsen, Agnieszka S Juncker, Marcelo Bertalan, Florence Levenez, Nicolas Pons, Simon Rasmussen, Shinichi Sunagawa, Julien Tap, Sebastian Tims, Erwin G Zoetendal, Søren Brunak, Karine Clément, Joël Doré, Michiel Kleerebezem, Karsten Kristiansen, Pierre Renault, Thomas Sicheritz-Ponten, Willem M de Vos, Jean-Daniel Zucker, Jeroen Raes, Torben Hansen, MetaHIT consortium, Peer Bork, Jun Wang, S Dusko Ehrlich, and Oluf Pedersen. Richness of human gut microbiome correlates with metabolic markers. *Nature*, 500(7464):541–546, August 2013.

[91] Mazharul Islam Leon, Md Ifraham Iqbal, Sadaf Meem, Furkan Alahi, Morshed Ahmed, Swakkhar Shatabda, and Md Saddam Hossain Mukta. *Dengue Outbreak Prediction from Weather Aware Data*, page 1–11. Springer International Publishing, 2022.

[92] Derek LeRoith and Charles T. Roberts. The insulin-like growth factor system and cancer. *Cancer Letters*, 195(2):127–137, June 2003.

[93] Nathan E Lewis, Kim K Hixson, Tom M Conrad, Joshua A Lerman, Pep Charusanti, Ashoka D Polpitiya, Joshua N Adkins, Gunnar Schramm, Samuel O Purvine, Daniel Lopez-Ferrer, Karl K Weitz, Roland Eils, Rainer König, Richard D Smith, and Bernhard Ø Palsson. Omic data from evolved *E. coli* are consistent with computed optimal growth from genome-scale models. *Molecular Systems Biology*, 6(1):390, January 2010.

[94] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q. Weinberger. On feature normalization and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12383–12392, June 2021.

[95] Hang Li. Deep learning for natural language processing: advantages and challenges. *National Science Review*, 5(1):24–26, September 2017.

[96] Qiongwei Li, Lifang Wang, Yuanlin Ma, Weihua Yue, Dai Zhang, and Jun Li. P-rex1 overexpression results in aberrant neuronal polarity and psychosis-related behaviors. *Neuroscience Bulletin*, 35(6):1011–1023, July 2019.

[97] Xiyin Li, Hairui Wang, Xing Yang, Xiaoqi Wang, Lina Zhao, Li Zou, Qin Yang, Zongliu Hou, Jing Tan, Honglei Zhang, Jianyun Nie, and Baowei Jiao. Gabrp sustains the stemness of triple-negative breast cancer cells through egfr signaling. *Cancer Letters*, 514:90–102, August 2021.

[98] Maxwell W. Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, May 2015.

[99] Wolfram Liebermeister, Elad Noor, Avi Flamholz, Dan Davidi, Jörg Bernhardt, and Ron Milo. Visual account of protein investment in cellular functions. *Proceedings of the National Academy of Sciences*, 111(23):8488–8493, June 2014. Publisher: Proceedings of the National Academy of Sciences.

[100] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[101] Yuqi Lin, Wen Zhang, Huanshen Cao, Gaoyang Li, and Wei Du. Classifying breast cancer subtypes using deep neural networks based on multi-omics data. *Genes*, 11(8):888, August 2020.

[102] Colton J. Lloyd, Ali Ebrahim, Laurence Yang, Zachary A. King, Edward Catoiu, Edward J. O'Brien, Joanne K. Liu, and Bernhard O. Palsson. COBRAme: A computational framework for genome-scale models of metabolism and gene expression. *PLOS Computational Biology*, 14(7):e1006302, July 2018. Publisher: Public Library of Science.

[103] Hongzhong Lu, Weiqiang Cao, Xiaoyun Liu, Yufei Sui, Liming Ouyang, Jianye Xia, Mingzhi Huang, Yingping Zhuang, Siliang Zhang, Henk Noorman, and Ju Chu. Multi-omics integrative analysis with genome-scale metabolic model simulation reveals global cellular adaptation of Aspergillus niger under industrial enzyme production condition. *Scientific Reports*, 8(1):14404, September 2018. Publisher: Nature Publishing Group.

[104] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[105] Daniel Machado and Markus Herrgård. Systematic Evaluation of Methods for Integration of Transcriptomic Data into Constraint-Based Models of Metabolism. *PLOS Computational Biology*, 10(4):e1003580, April 2014.

[106] Giuseppe Magazzù, Guido Zampieri, and Claudio Angione. Multimodal regularized linear models with flux balance analysis for mechanistic integration of omics data. *Bioinformatics*, 37(20):3546–3552, May 2021.

[107] Alessandro Magini, Alice Polchi, Alessandro Tozzi, Brunella Tancini, Michela Tantucci, Lorena Urbanelli, Tiziana Borsello, Paolo Calabresi, and Carla Emiliani. Abnormal cortical lysosomal beta-hexosaminidase and beta-galactosidase

activity at post-synaptic sites during alzheimer's disease progression. *The International Journal of Biochemistry and amp; Cell Biology*, 58:62–70, January 2015.

[108] Ahmed Mamdouh, Moumen El-Melegy, Samia Ali, and Ron Kikinis. Tab2visual: Overcoming limited data in tabular data classification using deep learning with visual representations, 2025.

[109] Jérôme Mariette and Nathalie Villa-Vialaneix. Unsupervised multiple kernel learning for heterogeneous data integration. *Bioinformatics*, 34(6):1009–1015, October 2017.

[110] Ambuj Mehrish, Navonil Majumder, Rishabh Bharadwaj, Rada Mihalcea, and Soujanya Poria. A review of deep learning techniques for speech processing. *Information Fusion*, 99:101869, November 2023.

[111] Ana Paula Mendes-Silva, Kelly Silva Pereira, Gesiane Thamire Tolentino-Araujo, Eduardo de Souza Nicolau, Camila Moreira Silva-Ferreira, Antonio Lucio Teixeira, and Breno S. Diniz. Shared biologic pathways between alzheimer disease and major depression: A systematic review of microrna expression studies. *The American Journal of Geriatric Psychiatry*, 24(10):903–912, October 2016.

[112] Luis Moles, Fernando Boto, Goretti Echegaray, and Iván G. Torre. *Convolutional Neural Networks for Structured Industrial Data*, page 361–370. Springer Nature Switzerland, October 2022.

[113] Jonathan Monk, Juan Nogales, and Bernhard O. Palsson. Optimizing genome-scale network reconstructions. *Nature Biotechnology*, 32(5):447–452, May 2014. Publisher: Nature Publishing Group.

[114] Jonathan M. Monk, Colton J. Lloyd, Elizabeth Brunk, Nathan Mih, Anand Sastry, Zachary King, Rikiya Takeuchi, Wataru Nomura, Zhen Zhang, Hirotada Mori, Adam M. Feist, and Bernhard O. Palsson. iML1515, a knowledgebase that computes Escherichia coli traits. *Nature Biotechnology*, 35(10):904–908, October 2017. Number: 10 Publisher: Nature Publishing Group.

[115] James Morrissey, Gianmarco Barberi, Benjamin Strain, Pierantonio Facco, and Cleo Kontoravdi. Next-fba: A hybrid stoichiometric/data-driven approach to improve intracellular flux predictions. *Metabolic Engineering*, March 2025.

[116] Kokhlikyan Narine, Miglani Vivek, Martin Miguel, Wang Edward, Alsallakh Bilal, Reynolds Jonathan, Melnikov Alexander, Kliushkina Natalia, Araya Carlos, Yan Siqi, and Reblitz-Richardson Orion. Captum: A unified and generic model interpretability library for pytorch, 2020.

[117] The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, September 2012.

[118] Bastian Niebel, Simeon Leupold, and Matthias Heinemann. An upper limit on Gibbs energy dissipation governs cellular metabolism. *Nature Metabolism*, 1(1):125–132, January 2019.

[119] Edward J O'Brien, Joshua A Lerman, Roger L Chang, Daniel R Hyduke, and Bernhard Ø Palsson. Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction. *Molecular Systems Biology*, 9(1):693, January 2013. Publisher: John Wiley & Sons, Ltd.

[120] Osamu Ogawa, David M. Becroft, Ian M. Morison, Michael R. Eccles, Jane E. Skeen, David C. Mauger, and Anthony E. Reeve. Constitutional relaxation of insulin–like growth factor ii gene imprinting associated with wilms' tumour and gigantism. *Nature Genetics*, 5(4):408–412, December 1993.

[121] Min Oh and Liqing Zhang. Deepmicro: deep representation learning for disease prediction based on microbiome data. *Scientific Reports*, 10(1), April 2020.

[122] Brett Olivier, Willi Gottstein, Douwe Molenaar, and Bas Teusink. CBMPy release 0.8.4, February 2023.

[123] Jeffrey D. Orth, R. M. T. Fleming, and Bernhard Ø. Palsson. Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide. *EcoSal Plus*, 4(1), February 2010.

[124] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248, March 2010.

[125] Edward J. O'Brien, Jonathan M. Monk, and Bernhard O. Palsson. Using genome-scale models to predict biological capabilities. *Cell*, 161(5):971–987, May 2015.

[126] Thummarat Paklao, Apichat Suratanee, and Kitiporn Plaimas. ICON-GEMs: integration of co-expression network in genome-scale metabolic models, shedding light through systems biology. *BMC Bioinformatics*, 24(1):492, December 2023.

[127] Luca Palazzolo, Chiara Paravicini, Tommaso Laurenzi, Sara Adobati, Simona Saporiti, Uliano Guerrini, Elisabetta Gianazza, Cesare Indiveri, Catriona M.H. Anderson, David T. Thwaites, and Ivano Eberini. Slc6a14, a pivotal actor on cancer stage: When function meets structure. *SLAS Discovery*, 24(9):928–938, October 2019.

[128] B. O. Palsson. Systems biology: Constraint-based reconstruction and analysis, 2015.

[129] Vikash Pandey, Daniel Hernandez Gardiol, Anush Chiappino-Pepe, and Vassily Hatzimanikatis. TEX-FBA: A constraint-based method for integrating gene expression, thermodynamics, and metabolomics data into genome-scale metabolic models, January 2019. Pages: 536235 Section: New Results.

[130] Vikash Pandey, Noushin Hadadi, and Vassily Hatzimanikatis. Enhanced flux prediction by integrating relative expression and relative metabolite abundance into thermodynamically consistent metabolic models. *PLOS Computational Biology*, 15(5):e1007036, May 2019. Publisher: Public Library of Science.

[131] Shouneng Peng, Lu Zeng, Jean-Vianney Haure-Mirande, Minghui Wang, Derek M. Huffman, Vahram Haroutunian, Michelle E. Ehrlich, Bin Zhang, and Zhidong Tu. Transcriptomic changes highly similar to alzheimer's disease are observed in a subpopulation of individuals during normal brain aging. *Frontiers in Aging Neuroscience*, 13, December 2021.

[132] Milan Picard, Marie-Pier Scott-Boyer, Antoine Bodein, Olivier Périn, and Arnaud Droit. Integration strategies of multi-omics data for machine learning analysis. *Computational and Structural Biotechnology Journal*, 19:3735–3746, 2021.

[133] Elena A. Ponomarenko, George S. Krasnov, Olga I. Kiseleva, Polina A. Kryukova, Viktoriia A. Arzumanian, Georgii V. Dolgalev, Ekaterina V. Ilgisonis, Andrey V. Lisitsa, and Ekaterina V. Poverennaya. Workability of mRNA Sequencing for Predicting Protein Abundance. *Genes*, 14(11):2065, November 2023. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[134] Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, Yangqing Peng, Dongya Zhang, Zhuye Jie, Wenxian Wu, Youwen Qin, Wenbin Xue, Junhua Li, Lingchuan Han, Donghui Lu, Peixian Wu, Yali Dai, Xiaojuan Sun, Zesong Li, Aifa Tang, Shilong Zhong, Xiaoping Li, Weineng Chen, Ran Xu, Mingbang Wang, Qiang Feng, Meihua Gong, Jing Yu, Yanyan Zhang, Ming Zhang, Torben Hansen, Gaston Sanchez, Jeroen Raes, Gwen Falony, Shujiro Okuda, Mathieu Almeida, Emmanuelle LeChatelier, Pierre Renault, Nicolas Pons, Jean-Michel Batto, Zhaoxi Zhang, Hua Chen, Ruifu Yang, Weimou Zheng, Songgang Li, Huanming Yang, Jian Wang, S Dusko Ehrlich, Rasmus Nielsen, Oluf Pedersen, Karsten Kristiansen, and Jun Wang. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, October 2012.

[135] Junjie Qin, MetaHIT Consortium, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, Nicolas Pons, Florence Levenez, Takuji Yamada, Daniel R Mende, Junhua Li, Junming Xu, Shaochuan Li, Dongfang Li, Jianjun Cao, Bo Wang, Huiqing Liang,

Huisong Zheng, Yinlong Xie, Julien Tap, Patricia Lepage, Marcelo Bertalan, Jean-Michel Batto, Torben Hansen, Denis Le Paslier, Allan Linneberg, H Bjørn Nielsen, Eric Pelletier, Pierre Renault, Thomas Sicheritz-Ponten, Keith Turner, Hongmei Zhu, Chang Yu, Shengting Li, Min Jian, Yan Zhou, Yingrui Li, Xiuqing Zhang, Songgang Li, Nan Qin, Huanming Yang, Jian Wang, Søren Brunak, Joel Doré, Francisco Guarner, Karsten Kristiansen, Oluf Pedersen, Julian Parkhill, Jean Weissenbach, Peer Bork, S Dusko Ehrlich, and Jun Wang. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65, March 2010.

[136] Nan Qin, Fengling Yang, Ang Li, Edi Prifti, Yanfei Chen, Li Shao, Jing Guo, Emmanuelle Le Chatelier, Jian Yao, Lingjiao Wu, Jiawei Zhou, Shujun Ni, Lin Liu, Nicolas Pons, Jean Michel Batto, Sean P Kennedy, Pierre Leonard, Chunhui Yuan, Wenchao Ding, Yuanting Chen, Xinjun Hu, Beiwen Zheng, Guirong Qian, Wei Xu, S Dusko Ehrlich, Shusen Zheng, and Lanjuan Li. Alterations of the human gut microbiome in liver cirrhosis. *Nature*, 513(7516):59–64, September 2014.

[137] R. A. Fisher. Iris, 1936.

[138] Alain Rakotomamonjy, Francis Bach, Stephane Canu, and Yves Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[139] Parminder S. Reel, Smarti Reel, Ewan Pearson, Emanuele Trucco, and Emily Jefferson. Using machine learning approaches for multi-omics data analysis: A review. *Biotechnology Advances*, 49:107739, July 2021.

[140] Ferran Reverter, Esteban Vegas, and Josep M Oller. Kernel-PCA data integration with enhanced interpretability. *BMC Systems Biology*, 8(S2), March 2014.

[141] Roman Rosipal and Leonard J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of machine learning research*, 2(Dec):97–123, 2001.

[142] Volker Roth and Volker Steinhage. Nonlinear discriminant analysis using kernel functions. *Advances in neural information processing systems*, 12, 1999.

[143] Ankur Sahu, Mary Ann Blatke, Jedrzej Jakub Szymański, and Nadine Topfer. Advances in flux balance analysis by integrating machine learning and mechanism-based models. *Computational and Structural Biotechnology Journal*, 19:4626–4640, 2021.

[144] Pierre Salvy and Vassily Hatzimanikatis. The ETFL formulation allows multi-omics integration in thermodynamics-compliant metabolism and expression models. *Nature Communications*, 11(1):30, January 2020. Publisher: Nature Publishing Group.

[145] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1.2):206–226, January 2000.

[146] Bradley K. Schniers, Mitchell S. Wachtel, Meenu Sharma, Ksenija Korac, Devaraja Rajasekaran, Shengping Yang, Tyler Sniegowski, Vadivel Ganapathy, and Yangzom D. Bhutia. Deletion of slc6a14 reduces cancer growth and metastatic spread and improves survival in kpc mouse model of spontaneous pancreatic cancer. *Biochemical Journal*, 479(5):719–730, March 2022.

[147] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[148] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, December 2001.

[149] Kanghyeon Seo, Bokjin Chung, Hamsa Priya Panchaseelan, Taewoo Kim, Hyejung Park, Byungmo Oh, Minho Chun, Sunjae Won, Donkyu Kim, Jaewon Beom, Doyoung Jeon, and Jihoon Yang. Forecasting the walking assistance rehabilitation level of stroke patients using artificial intelligence. *Diagnostics*, 11(6):1096, June 2021.

[150] Alessandro Sgambato, Mario Migaldi, Micaela Montanari, Andrea Camerini, Andrea Brancaccio, Giulio Rossi, Rodolfo Cangiano, Carmen Losasso, Giovanni Capelli, Gian Paolo Trentini, and Achille Cittadini. Dystroglycan expression is frequently reduced in human breast and colon cancers and is associated with tumor progression. *The American Journal of Pathology*, 162(3):849–860, March 2003.

[151] Hossein Sharifi-Noghabi, Olga Zolotareva, Colin C Collins, and Martin Ester. Moli: multi-omics late integration with deep neural networks for drug response prediction. *Bioinformatics*, 35(14):i501–i509, July 2019.

[152] Heidi Y. Shi, Rong Liang, Nancy S. Templeton, and Ming Zhang. Inhibition of breast tumor progression by systemic delivery of the maspin gene in a syngeneic tumor model. *Molecular Therapy*, 5(6):755–761, June 2002.

[153] Amrit Singh, Casey P Shannon, Benoît Gautier, Florian Rohart, Michaël Vacher, Scott J Tebbutt, and Kim-Anh Lê Cao. Diablo: an integrative approach for identifying key molecular drivers from multi-omics assays. *Bioinformatics*, 35(17):3055–3062, January 2019.

[154] Divya Singh, Tal Robin, Michael Urbakh, and Shlomi Reuveni. High-order Michaelis-Menten equations allow inference of hidden kinetic parameters in enzyme catalysis, June 2024. Pages: 2024.06.12.598609 Section: New Results.

[155] Krishna Kumar Singh, Hao Yu, Aron Sarmasi, Gautam Pradeep, and Yong Jae Lee. Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond, 2018.

[156] Vincent Somerville, Pranas Grigaitis, Julius Battjes, Francesco Moro, and Bas Teusink. Use and limitations of genome-scale metabolic models in food microbiology. *Current Opinion in Food Science*, 43:225–231, February 2022.

[157] Huan Song, Jayaraman J. Thiagarajan, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, and Andreas Spanias. A deep learning approach to multiple kernel fusion. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2292–2296. IEEE, 2017.

[158] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[159] Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics*, 23(2), January 2022.

[160] M. Forina Stefan Aeberhard. Wine, 1992.

[161] Max E Stevenson, Michaeline Hebron, Xiaoguang Liu, Balaraman Kavulu, Christian Wolf, and Charbel E-H Moussa. Inhibiting tyrosine kinase c-kit as a therapeutic strategy for alzheimer's disease. *Alzheimer's and amp; Dementia*, 18(S10), December 2022.

[162] Virginie Stygelbout, Karelle Leroy, Valérie Pouillon, Kunie Ando, Eva D'Amico, Yonghui Jia, H. Robert Luo, Charles Duyckaerts, Christophe Erneux, Stéphane Schurmans, and Jean-Pierre Brion. Inositol trisphosphate 3-kinase b is increased in human alzheimer brain and exacerbates mouse alzheimer pathology. *Brain*, 137(2):537–552, January 2014.

[163] B. Sun, L. Yang, W. Zhang, M. Lin, P. Dong, C. Young, and J. Dong. Supertml: Two-dimensional word embedding for the precognition on structured tabular data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2973–2981, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society.

[164] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. Super characters: A conversion from sentiment classification to image classification, 2018.

[165] Baohua Sun, Lin Yang, Michael Lin, Charles Young, Patrick Dong, Wenhan Zhang, and Jason Dong. Supercaptioning: Image captioning using two-dimensional word embedding, 2019.

[166] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

[167] Hye Youn Sung, San-Duk Yang, Woong Ju, and Jung-Hyuck Ahn. Aberrant epigenetic regulation of gabrp associates with aggressive phenotype of ovarian cancer. *Experimental and amp; Molecular Medicine*, 49(5):e335–e335, May 2017.

[168] Taiji Suzuki and Ryota Tomioka. Spicymkl: a fast algorithm for multiple kernel learning with thousands of kernels. *Machine Learning*, 85(1–2):77–108, June 2011.

[169] Panos K. Syriopoulos, Nektarios G. Kalampalikis, Sotiris B. Kotsiantis, and Michael N. Vrahatis. knn classification: a review. *Annals of Mathematics and Artificial Intelligence*, 93(1):43–75, September 2023.

[170] Gabriele Tazza, Francesco Moro, Dario Ruggeri, Bas Teusink, and László Vidács. Minn: A metabolic-informed neural network for integrating omics data into genome-scale metabolic modeling. *Computational and Structural Biotechnology Journal*, 27:3609–3617, 2025.

[171] Gabriele Tazza, Francesco Moro, Bas Teusink, and László Vidács. Metabolic-informed neural network for multi-omics data integration. In Jan F.M. Van Impe and Monika E. Polańska, editors, *13th International Conference on Simulation and Modelling in the Food and Bio-Industry (FOODSIM 2024)*, pages 193–197. Eurosis-ETI, 2024. Publisher Copyright: © 2024, EUROSIS-ETI. All rights reserved.; 13th International Conference on Simulation and Modelling in the Food and Bio-Industry, FOODSIM 2024 ; Conference date: 07-04-2024 Through 11-04-2024.

[172] Gabriele Tazza, Dario Ruggeri, and László Vidács. Improving microbiome-based disease prediction with supertml and data augmentation. *IEEE Access*, 13:144505–144515, 2025.

[173] Ines Thiele and Bernhard Ø Palsson. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature Protocols*, 5(1):93–121, January 2010. Publisher: Nature Publishing Group.

[174] Samira van den Bogaard, Pedro A Saa, and Tobias B Alter. Sensitivities in protein allocation models reveal distribution of metabolic capacity and flux control. *Bioinformatics*, 40(12):btae691, December 2024.

[175] Sandra Van der Auwera, Sabine Ameling, Katharina Wittfeld, Stefan Frenzel, Robin Bülow, Matthias Nauck, Henry Völzke, Uwe Völker, and Hans J. Grabe. Circulating microrna mir-425-5p associated with brain white matter lesions and inflammatory processes. *International Journal of Molecular Sciences*, 25(2):887, January 2024.

[176] Eunice van Pelt-KleinJan, Daan H. de Groot, and Bas Teusink. Understanding FBA Solutions under Multiple Nutrient Limitations. *Metabolites*, 11(5):257, May 2021. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[177] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[178] Roger A Vaughan, Nicholas P Gannon, Randi Garcia-Smith, Yamhilette Licon-Munoz, Miguel A Barberena, Marco Bisoffi, and Kristina A Trujillo. Beta-alanine suppresses malignant breast epithelial cell aggressiveness through alterations in metabolism and cellular acidity in vitro. *Molecular Cancer*, 13(1):14, 2014.

[179] Constance Vennin, Nathalie Spruyt, Fatima Dahmani, Sylvain Julien, François Bertucci, Pascal Finetti, Thierry Chassat, Roland P. Bourette, Xuefen Le Bourhis, and Eric Adriaenssens. H19non coding rna-derived mir-675 enhances tumorigenesis and metastasis of breast cancer cells by downregulating c-cbl and cbl-b. *Oncotarget*, 6(30):29209–29223, July 2015.

[180] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR, 09–15 Jun 2019.

[181] Peter Vorwerk, Heike Wex, Cornelia Bessert, Bianka Hohmann, Uwe Schmidt, and Uwe Mittler. Loss of imprinting of igf-ii gene in children with acute lymphoblastic leukemia. *Leukemia Research*, 27(9):807–812, September 2003.

[182] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:1–13, 2018.

[183] Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications*, 12(1), June 2021.

[184] X. Wang, E. P. Xing, and D. J. Schaid. Kernel methods for large-scale genomic data analysis. *Briefings in Bioinformatics*, 16(2):183–192, July 2014.

[185] Indika Wickramasinghe and Harsha Kalutarage. Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, 25(3):2277–2293, September 2020.

[186] Christopher M. Wilson, Kaiqiao Li, Xiaoqing Yu, Pei-Fen Kuan, and Xuefeng Wang. Multiple-kernel learning for genomic data mining and prediction. *BMC Bioinformatics*, 20(1), August 2019.

[187] Ulrike Wittig, Maja Rey, Andreas Weidemann, Renate Kania, and Wolfgang Müller. SABIO-RK: an updated resource for manually curated biochemical reaction kinetics. *Nucleic Acids Research*, 46(D1):D656–D660, January 2018.

[188] Xing Wu and Qiulian Fang. Stacked autoencoder based multi-omics data integration for cancer survival prediction, 2022.

[189] Yihong Wu, Linlin Sun, Weiying Zou, Jiejie Xu, Haiou Liu, Wenzhong Wang, Xiaojing Yun, and Jianxin Gu. Prosaposin, a regulator of estrogen receptor alpha, promotes breast cancer growth. *Cancer Science*, 103(10):1820–1825, August 2012.

[190] Thomas P. Wytock and Adilson E. Motter. Predicting growth rate from gene expression. *Proceedings of the National Academy of Sciences*, 116(2):367–372, December 2018.

[191] Bing Xu, Naiyan Wang, and Tianqi Chen. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015.

[192] Jing Xu, Peng Wu, Yuehui Chen, Qingfang Meng, Hussain Dawood, and Hassan Dawood. A hierarchical integration deep flexible neural forest framework for cancer subtype classification by integrating multi-omics data. *BMC Bioinformatics*, 20(1), October 2019.

[193] Zenglin Xu, Rong Jin, Haiqin Yang, Irwin King, and Michael R. Lyu. Simple and efficient multiple kernel learning by group lasso. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1175–1182, Madison, WI, USA, 2010. Omnipress.

[194] Haitao Yang, Hongyan Cao, Tao He, Tong Wang, and Yuehua Cui. Multilevel heterogeneous omics data integration with kernel fusion. *Briefings in Bioinformatics*, November 2018.

[195] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey, 2022.

[196] Keren Yizhak, Tomer Benyamini, Wolfram Liebermeister, Eytan Ruppin, and Tomer Shlomi. Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model. *Bioinformatics*, 26(12):i255–i260, June 2010.

[197] Tianwei Yu. Aime: Autoencoder-based integrative multi-omics data embedding that allows for confounder adjustments. *PLOS Computational Biology*, 18(1):e1009826, January 2022.

[198] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, July 2019.

[199] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[200] Afia Zafar, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17):8643, August 2022.

[201] Guido Zampieri, Stefano Campanaro, Claudio Angione, and Laura Treu. Metatranscriptomics-guided genome-scale metabolic modeling of microbial communities. *Cell Reports Methods*, 3(1), January 2023. Publisher: Elsevier.

[202] Guido Zampieri, Supreeta Vijayakumar, Elisabeth Yaneske, and Claudio Angione. Machine and deep learning meet genome-scale metabolic modeling. *PLoS Computational Biology*, 15(7), July 2019.

[203] Georg Zeller, Julien Tap, Anita Y Voigt, Shinichi Sunagawa, Jens Roat Kultima, Paul I Costea, Aurélien Amiot, Jürgen Böhm, Francesco Brunetti, Nina Habermann, Rajna Hercog, Moritz Koch, Alain Luciani, Daniel R Mende, Martin A Schneider, Petra Schrotz-King, Christophe Tournigand, Jeanne Tran Van Nhieu, Takuji Yamada, Jürgen Zimmermann, Vladimir Benes, Matthias Kloor, Cornelia M Ulrich, Magnus von Knebel Doeberitz, Iradj Sobhani, and Peer Bork. Potential of fecal microbiota for early-stage detection of colorectal cancer. *Mol. Syst. Biol.*, 10(11):766, November 2014.

[204] Li Zhang, Xin Zhang, Xin Wang, Miao He, and Shixing Qiao. Microrna-224 promotes tumorigenesis through downregulation of caspase-9 in triple-negative breast cancer. *Disease Markers*, 2019:1–9, February 2019.

[205] Xinhua Zhang. *Kernel Methods*, chapter 1, page 566–570. Springer US, 2011.

[206] Xia Zhao, Limin Wang, Yufei Zhang, Xuming Han, Muhammet Deveci, and Milan Parmar. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57(4), March 2024.

[207] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation, 2017.

[208] Tongxue Zhou. Modality-level cross-connection and attentional feature fusion based deep neural network for multi-modal brain tumor segmentation. *Biomedical Signal Processing and Control*, 81:104524, March 2023.

[209] Hadas Zur, Eytan Ruppin, and Tomer Shlomi. iMAT: an integrative metabolic analysis tool. *Bioinformatics*, 26(24):3140–3142, December 2010.

Declaration on the use of AI

AI tools were used during the preparation of this thesis mainly to assist writing and software development. In particular, the ChatGPT Premium and the Grammarly Premium versions were used for small specific tasks related to writing and software development. For the writing part, these tools were used mainly for language assistance, such as improving sentence structure, clarity, readability and suggesting alternative phrasings for sentences or short portions of text. Any text generated or modified with the assistance of AI tools was carefully reviewed, edited where necessary, and approved by the author. For the software part, ChatGPT was used as a programming assistant to support debugging and to suggest small code snippets, mainly for data visualization and for auxiliary steps of data analysis pipelines, such as logging, saving and organizing outputs for reporting, and supporting experiment management tasks. The use of AI tools was limited and did not replace the original scientific contributions of the author. All research design, analysis, interpretation of results, and scientific conclusions presented in this thesis remain responsibility of the author.

Summary

The PhD thesis presents machine learning approaches at genome-scale. Specifically, it focuses on methods that can improve predictions in the context of data scarcity, a common challenge in bioinformatics.

After an Introduction and a Background chapters, which serve to give a common starting point to the readers from different backgrounds, the dissertation consists of three major parts. In Chapter 3, we present a framework to improve microbiome-based disease prediction transforming the problem into an image classification one and exploiting image data augmentation as a regularization technique. Chapter 4 explores several supervised MKL methods for multi-omics integration and introduces a novel framework called DeepMKL which use deep learning optimization as a kernel fusion technique. In Chapter 5, we present MINN, a hybrid data-driven/mechanistic framework for integrating multi-omics into genome scale metabolic modeling to improve flexibility and prediction power.

Improving microbiome-based disease prediction with SuperTML and data augmentation

In Chapter 3, we examined the challenges of predicting disease from microbiome data, where the small sample size and high dimensionality often limit the performance of traditional neural networks. To address this, we tested SuperTML, a deep learning framework originally designed for small tabular datasets, and applied it to this context. Our results show that SuperTML is a valid alternative to state-of-the-art methods, and that its performance further improves when combined with data augmentation techniques. In most of the datasets studied, this approach outperformed traditional models, highlighting the importance of augmentation as a regularization method. Overall, this chapter presents SuperTML as a promising tool for microbiome-based disease prediction.

Supervised Multiple Kernel Learning approaches for multi-omics data integration

In Chapter 4, we explored the challenge of multi-omics integration. The diversity and complexity of these data sources often limit the effectiveness of traditional bioinformatics methods. To address this, we introduced two new approaches based on Multiple Kernel Learning (MKL), a framework that, despite being relatively underused, offers strong potential for this task. We explored an approach that adapts unsupervised learning techniques for supervised prediction using Support Vector Machines, as well as DeepMKL, a deep learning–based framework that integrates kernels without relying on convex linear optimization. Experiments on four publicly available biomedical datasets showed that both approaches provide a reliable and competitive solution, achieving comparable or even better performance than more complex state-of-the-art methods. In addition, we proposed a two-step strategy for biomarker discovery that leverages DeepMKL together with a novel interpretability procedure. This method proved effective in identifying biomarkers associated with diseases such as breast cancer and Alzheimer's, highlighting its potential to generate insights beyond prediction accuracy. Overall, this chapter showed that MKL represents a fast and robust solution for multi-omics integration, with the flexibility to compete with and complement more advanced architectures.

MINN: A Metabolic-Informed Neural Network for Integrating Omics Data into Genome-Scale Metabolic Modeling

In Chapter 5, we present the Metabolic-Informed Neural Network (MINN), a hybrid framework designed to integrate multi-omics data with Genome-Scale Metabolic Models for flux prediction. Unlike purely data-driven or purely mechanistic approaches, MINN combines the flexibility of neural networks with the structured constraints of metabolic models. We tested different versions of the architecture to handle the trade-off between predictive accuracy and biological consistency, and we proposed a strategy to couple MINN with parsimonious Flux Balance Analysis (pFBA) to enhance interpretability. On a small E. coli multi-omics dataset of single-gene knockouts, MINN outperformed both classical machine learning methods and mechanistic models, showing that the inclusion of biological constraints stabilizes the learning process and reduces overfitting. Overall, our findings show that MINN is an effective and robust framework for metabolic flux prediction and it provides a flexible tool that can be extended to more complex systems and larger datasets, opening the way for more comprehensive and interpretable analyses in systems biology.

Összefoglalás

A doktori értekezés a gépi tanulás genom-szintű alkalmazásait mutatja be. Különös hangsúlyt kapnak azok a megközelítések, amelyek javíthatják az előrejelzések pontosságát adatszegény környezetben – ez a bioinformatika egyik leggyakoribb kihívása. Az bevezető fejezetek (*Introduction* és *Background*) közös kiindulópontot adnak a különböző tudományterületekről érkező olvasóknak. Ezt követően a dolgozat három fő részből áll. A 3. fejezet egy olyan keretrendszert mutat be, amely a mikrobiom alapú betegség-előrejelzést képfelismerési problémává alakítja, és az adat augmentáció módszerét használja regularizációs technikaként. A 4. fejezet az ún. Supervised Multiple Kernel Learning (MKL) megközelítéseit vizsgálja a multi-omikai adatok integrációjára, és bemutatja a DeepMKL nevű, mélytanulás alapú keretrendszert, amelyben egy újszerű kernel fúziós megoldás kerül bemutatásra. Végül, az 5. fejezetben ismertetett Metabolic-Informed Neural Network (MINN) egy hibrid, adatvezérelt és mechanisztikus elemeket ötvöző neurális modell, amely a multi-omikai adatokat a genomszintű anyagcsere-modellekbe integrálja azzal a céllal, hogy javítsa a predikciós teljesítményt és a biológiai értelmezhetőséget.

A mikrobiom-alapú betegség előrejelzés fejlesztése SuperTML és adat augmentáció segítségével

A 3. fejezet a mikrobiom-adatokból történő betegség előrejelzés kihívásait tárgyalja. A kis mintaszám és a magas dimenziószám gyakran korlátozza a hagyományos neurális hálózatok teljesítményét. Ennek kezelésére a SuperTML keretrendszert alkalmaztuk, amelyet eredetileg táblázatos adatok feldolgozására fejlesztettek ki. Eredményeink szerint a SuperTML versenyképes alternatívát nyújt a jelenlegi state-of-the-art módszerekhez képest, és teljesítménye tovább javul, ha adat augmentációs technikákkal kombináljuk. A legtöbb vizsgált adathalmazon ez a megközelítés jobb eredményt ért el, mint a hagyományos modellek, kiemelve az augmentáció szerepét, mint regularizációs módszert. Összességében a SuperTML ígéretes és rugalmas eszköznek bizonyult a mikrobiom-alapú betegség előrejelzésben.

Supervised Multiple Kernel Learning megközelítések a multi-omikai adatintegrációban

A 4. fejezet a multi-omikai adatok integrációjának problémáját vizsgálja, amelyet a források sokfélesége és komplexitása nehezít. Ennek megoldására két új, Multiple Kernel Learning (MKL) alapú megközelítést vezettünk be. Az egyik módszer a felügyelet nélküli tanulási technikákat alakítja át felügyelt feladattá Support Vector Machine módszer segítségével, míg a másik, a DeepMKL, mély tanulási megoldást használ a kernelfúzió megvalósításához, elkerülve a hagyományos konvex lineáris optimalizálási módszerek korlátait. Négy publikusan elérhető orvosbiológiai adathalmazon végzett kísérleteink azt mutatták, hogy mindkét megközelítés stabil és versenyképes teljesítményt nyújtott, gyakran felülmúlva a komplexebb state-of-the-art modelleket. Emellett bemutattunk egy kétlépcsős biomarker felfedező stratégiát is, amely a DeepMKL-t egy új interpretability eljárással kombinálja. A módszer hatékonyan azonosította a mellrákhoz és az Alzheimer kórhoz kapcsolódó biomarkereket, ezzel bizonyítva, hogy az MKL nemcsak prediktív pontosságban, hanem biológiai értelmezhetőségben is előnyt jelenthet.

MINN: Metabolic-Informed Neural Network – neuronhálós modell az omikai adatok genomszitnű anyagcsere-modellekbe való integrálására

Az 5. fejezetben a Metabolic-Informed Neural Network-öt (MINN) mutattuk be, egy hibrid keretrendszert, amelyet arra terveztünk, hogy a multi-omikai adatokat a genomszintű metabolikus modellekkel integrálva anyagcsere-flux folyamatokat jelezzen előre. A kizárólag adatvezérelt és a tisztán mechanisztikus megközelítésekkel szemben a MINN a neurális hálózatok rugalmasságát a metabolikus modellek strukturált megkötéseivel ötvözi. A módszer több architekturális változatát is teszteltük a prediktív pontosság és a biológiai konzisztencia közötti kompromisszum megteremtése érdekében. Emellett stratégiát javasoltunk a MINN parsimonious Flux Balance Analysis (pFBA) módszerrel történő összekapcsolására, ami javítja az eredmények értelmezhetőségét. Egy kisméretű egygénes kiütéseket tartalmazó, E. coli multi-omikai adathalmazon a MINN teljesítménye felülmúlta a klasszikus gépi tanulási módszerek és a mechanisztikus modellek teljesítményét is. Az eredmények azt mutatják, hogy a biológiai megkötések beépítése stabilizálja a tanulási folyamatot, és csökkenti a túlillesztés kockázatát. Összességében megállapítottuk, hogy a MINN hatékony és robusztus keretrendszer metabolikus fluxfolyamatok előrejelzésére, ami kiterjeszthető összetettebb rendszerek és nagyobb adatkészletek feldolgozására is. Mindez lehetőséget teremt átfogó, jobban értelmezhető rendszerbiológiai elemzések megvalósítására.

Publications

Journal publications

[1] **Gabriele Tazza**, Francesco Moro, Dario Ruggeri, Bas Teusink, and László Vidács MINN: A metabolic-informed neural network for integrating omics data into genome-scale metabolic modeling. In *Computational and Structural Biotechnology Journal*, 27, 3609–3617, 2025.

[2] **Gabriele Tazza**, Dario Ruggeri and László Vidács Improving Microbiome-Based Disease Prediction With SuperTML and Data Augmentation . In *IEEE Access*, 13, 144505-144515, 2025.

[3] Mitja Briscik, **Gabriele Tazza**, László Vidács, Marie-Agnès Dillies and Sébastien Déjean Supervised multiple kernel learning approaches for multi-omics data integration . In *BioData Mining* , 17, 53, 2024.

[4] Dario Ruggeri, **Gabriele Tazza** and László Vidács Introducing MLOps to Facilitate the Development of Machine Learning Models in Agronomy: A Case Study. In *IEEE Access*, 13, 122059-122070, 2025.

Full papers in conference proceedings

[5] **Gabriele Tazza**, Francesco Moro, Bas Teusink, and László Vidács Metabolic-Informed Neural Network for Multi-Omics Data Integration. In *Proceedings of the 13th International Conference on Simulation and Modelling in the Food and Bio-Industry, FOODSIM 2024*, Eurosis-ETI, 193-197, 2024.