

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Mathematical modeling of Celiac Disease: Physics-Informed
Neural Networks and data augmentation with sparse patient
data

verfasst von | submitted by

Kailey Reese Walker BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 821

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Mathematik

Betreut von | Supervisor:

Assoz. Prof. Dr. Philipp Christian Petersen M.Sc.

Zusammenfassung

Zöliakie ist eine Autoimmunerkrankung, die durch Glutenexposition ausgelöst wird. Derzeit ist die einzige verfügbare Behandlung eine strenge glutenfreie Ernährung, die sowohl herausfordernd sein kann als auch in seltenen Fällen die Symptome nicht lindert. Darüber hinaus können einzelne Patienten wesentlich unterschiedliche Reaktionen sowohl auf Gluten als auch auf eine glutenfreie Ernährung zeigen, was die Diagnose, Prognose und Entwicklung neuer Behandlungsmethoden erschwert. Während es mathematische Modelle vieler Autoimmunerkrankungen gibt, bleiben Modelle der Zöliakie weitgehend unerforscht aufgrund von Datenmangel, hoher Variabilität in verfügbaren Daten und der Komplexität der Erkrankung. In dieser Arbeit untersuchen wir Methoden zur besseren Nutzung der verfügbaren Patientendaten und streben die Entwicklung patientenspezifischer Modelle an, um personalisierte Behandlungsstrategien zu ermöglichen. Zunächst verwenden wir multiple Imputation und rekurrente neuronale Netze, um fehlende Daten und unregelmäßige Stichproben zu adressieren. Anschließend entwickeln wir ein Modell gewöhnlicher Differentialgleichungen (ODE-Modell), das individuelle Patientenreaktionen auf Glutenexposition und die Umstellung auf eine glutenfreie Ernährung beschreibt. Schließlich bewerten wir die Fähigkeit von Sparse-Regression Physics-Informed Neural Networks (SR-PINNs), zugrunde liegende Gleichungen aus synthetischen biologischen Zeitreihendaten wiederherzustellen, die durch das ODE-Modell generiert wurden. Unsere Ergebnisse zeigen bedeutende Einschränkungen. Wir stellen fest, dass Imputations- und neuronale Netz-Techniken erhebliche Nachteile bei kleinen Längsschnittdaten aufweisen und dass SR-PINNs, obwohl effektiv bei Benchmark-PDGs getestet, Schwierigkeiten haben, wenn sie auf realistischere steife Gleichungen zur Modellierung biologischer Prozesse angewendet werden. Diese Erkenntnisse verdeutlichen Herausforderungen bei der Anwendung moderner Methoden auf datenarme biologische Probleme und weisen Richtungen für zukünftige Entwicklungen auf.

Abstract

Celiac Disease is an autoimmune disorder triggered by exposure to gluten. Currently, the only available treatment is a strict gluten free diet, which can be both challenging and, in rare cases, fails to alleviate symptoms. Additionally, individual patients can have substantially different responses to both gluten and a gluten free diet, complicating diagnosis, prognosis, and development of new treatment methods. While there are mathematical models of many autoimmune disorders, models of Celiac Disease remain largely unexplored due to a lack of data, high variation in available data, and complexity of the disease. In this paper we explore methods to better utilize the available patient data, and aim to develop patient-level models to enable personalized treatment strategies. We first use multiple imputation and recurrent neural networks to address missing data and irregular sampling. We then develop an ODE model describing individual patient responses to gluten exposure and adoption of a gluten free diet. Finally, we assess the ability of Sparse-Regression Physics-Informed Neural Networks (SR-PINNs) to recover governing equations from synthetic biological time-series data generated by the ODE model. Our results reveal significant limitations. We find that imputation and neural network techniques have significant drawbacks in small longitudinal datasets, and that SR-PINNs, while effective when tested on benchmark PDEs, face difficulties when applied to more realistic, stiff equations modeling biological processes. These findings highlight challenges in applying modern methods to data-limited biological problems and suggest directions for future development.

Contents

Contents	iii
1 Introduction	2
1.1 Mechanisms of Celiac Disease	2
1.2 Modeling of autoimmune diseases	3
1.2.1 ODE construction	4
1.3 Data collection	4
1.3.1 Problems with collected datasets	6
1.4 Neural Network discovery	7
1.5 Project approach	9
2 Data processing	10
2.1 Study types	10
2.2 Exploratory data analysis	10
2.2.1 Dataframe A	10
2.2.2 Dataframe B	13
2.3 Dataframe C	24
2.4 Imputation of missing data	27
2.4.1 Types of missing data	27
2.4.2 Multiple imputation	29
2.4.3 Evaluation of imputation techniques	32
2.4.4 Imputation of Dataset B	33
3 Data generation	53
3.1 Data representatives	53
3.1.1 Generation model	54
3.1.2 Generation results	56
3.2 ODE representatives	57
3.2.1 Assumptions	57
3.2.2 ODE model	58
3.2.3 Parameter estimates	59
3.2.4 Analysis	60
3.2.5 Numerical solution(s)	62
3.2.6 Problems	65
4 SR-PINNs	67
4.1 Problem formulation	67
4.2 ADO	68
4.2.1 STRidge algorithm	70
4.2.2 Optimization algorithms	70
4.3 Data preprocessing	71
4.4 Candidate function library	72
4.5 Hyperparameter selection	72
4.6 Results	73
4.6.1 Baseline implementation	73
4.6.2 Tolerance regularization	74
4.6.3 Data loss training	76
4.6.4 Alternative loss functions	77
4.6.5 Biased collocation sampling	78

4.6.6 Combined best practices	78
4.6.7 Single variable analysis: variable T	79
4.7 Discussion and future directions	83
5 Conclusion	85
5.1 Future directions	86
Bibliography	87
Appendix	92
A.1 Glossary	93
B.2 Code and data availability	94

Chapter 1

Introduction

Celiac Disease is an autoimmune disorder characterized by immunological activation in response to gluten. Most patients with Celiac Disease manifest the disease in different ways with different levels of severity [1, 2]. Despite this, the treatment scheme for all patients is the same: every patient is instructed to follow a strict gluten-free diet for the rest of their lives. Maintaining an entirely gluten free diet is not only extremely challenging (and in most cases impossible), but it can also be detrimental to overall well-being. In fact, those who are most vigilant about what they can and cannot eat tend to have lower quality of life indices [3]. Additionally, some patients with Celiac Disease continue to exhibit severe symptoms even after a gluten free diet has been adopted, and these patients often have no alternative treatment options available [4].

In general, the modeling of autoimmune diseases has been used to predict the efficacy of certain treatments, aid in drug development, and develop diagnostic programs. In the case of Celiac Disease, mathematical modeling of treatment schemes has proven very difficult, with predictions of efficacy of treatments producing varying results. This is due to a variety of reasons, including lack of data availability and the complexity of the disease.

In this paper we aim to improve how Celiac Disease is modeled. We focus on development of patient-specific models in order to explore individualized treatment schemes. Due to the extremely limited availability of patient data, we explore methods that focus on estimating and interpolating data, and methods that are designed to overcome difficulties related to sparse and noisy datasets. We explore the mechanisms of Celiac Disease (Section 1.1), the availability of data (Section 1.3), and different methods of modelling autoimmune disorders (Sections 1.2, 1.4). We present our project approach and thesis outline in Section 1.5.

1.1 Mechanisms of Celiac Disease

Celiac Disease involves an innate immune response and an adaptive immune response that is initiated after a patient eats gluten-containing food. Once consumed, the gluten is partially digested in the small intestine into gliadin fragments. The partially processed gliadin migrates and is absorbed into the intestinal wall in spots of higher permeability, where the adaptive immune response takes place [5]. First, the gliadin undergoes a process called deamidation by tissue transglutaminase 2 (TG2), which enhances the ability for the gliadin to provoke an immune response. After deamidation, the gliadin interacts with antigen-presenting cells (APCs), which treat the gliadin as an antigen and absorb and display gliadin peptides on their surfaces, alerting $CD4^+$ T-cells. In addition, due to the deamidation process, APCs can absorb TG2 complexes, enabling the production of anti-gliadin and anti-TG2 antibodies called Immunoglobulin G (IgG) and Immunoglobulin A (IgA) that attack both the deamidated gluten peptides (DGP) and TG2. The ensuing innate immune response results in the production of inflammatory and tissue-damaging cytokines and intraepithelial lymphocytes (IELs), damaging the walls and villi of the small intestine. The results of this innate immune response can further increase the permeability of the intestinal walls, creating a cascading loop of gliadin processing and immune reactions [2].

In order to be able to model Celiac Disease, it is useful to understand whether the concentrations of enzymes, antibodies, cells, and gluten are interdependent. We outline the biological connections between immune markers below.

IL-15: IL-15 antibodies are major drivers of the immune response in Celiac Disease and other autoimmune disorders [6]. IL-15 antibodies attach to APCs and activate them. Additionally, they attach to intestinal epithelial cells. IL-15 has been shown to be upregulated after ingestion of gluten [7].

IELs: The presence of IL-15 on IELs directly causes the increased expression of natural-killer receptors on IELs, which causes them to activate and target intestinal cells [7, 6]. Additionally, activation of IELs is dependent on the presence of IFN- γ [7], and the activation of CD8+ cells has been shown to be dependent on the presence of CD4+ cells and IL-15 in mice [8]. The development and survival of memory CD8+ cells is dependent on the presence of IL-15 [6]. Notably, change in CD3+ IELs has been shown to be uncorrelated with gluten intake [9].

IL-21 and IFN- γ : CD4+ T-cells produce IL-21 and IFN- γ , and CD8+ cells produce IFN- γ [7, 8]. In addition, IL-21 promotes the production of IFN- γ and IL-21 [6, 7, 8], and IL-15 promote both the production and expression of IL-21 and IFN- γ [10].

IL-2: IL-2 has been shown to be rapidly generated by T-cells after introduction of gluten. It has been shown to be highly correlated with digestive effects such as nausea and vomiting after gluten ingestion, and therefore may contribute to intestinal damage [11].

CD4+ T-cells: CD4+ T-cells are activated by presentation of gluten by APCs [12]. Notably, CD4+ cell production does not seem to be dependent on the presence of IL-15 [8]. In addition, there are CD4+ cells that retain gluten-specific memory, and mobilize faster after repeated exposure to gluten [7].

TG2: Change in TG2 concentrations in serum over time is correlated with the amount of gluten ingested and how long gluten is ingested for [9]. In addition, TG2 concentration is inversely proportional to IgA and IgG, which both attack TG2 [6].

IgG and IgA: Change in IgG and IgA is correlated with higher amounts of IL-15 [7]. Production of anti-DGP antibodies is dependent on the presence of IL-21 and IFN- γ . Additionally, overexpression of IL-15 results in production of anti-TG2 and anti-DGP IgG and IgA in mice [6]. It is unclear what other effects anti-TG2 and anti-DGP have during the immune reaction of Celiac Disease. Some studies have found that both have effects on the death rate of intestinal cells, and may have an impact on the degradation of the intestinal villi. Additionally, both have been found in extra-intestinal organs that are affected by Celiac Disease [13].

Villi size: IFN- γ is required to kill intestinal epithelial cells efficiently [7]. However, CD8+ lymphocytes and other IELs are directly responsible for the destruction of intestinal epithelial cells and the decrease in villi size [7]. The concentration of IL-15 correlates with the amount of intestinal damage [6]. It has been shown in mice that the presence of (activated) CD4+ T-cells is not enough to trigger villi and crypt damage, but that together activated CD4+ T-cells and IL-15 work together to induce villi damage [8].

1.2 Modeling of autoimmune diseases

Due to the complexity and unpredictable nature of autoimmune diseases, as well as the inherent risk associated with experiments conducted directly on people, mathematical modeling of autoimmune diseases is a common practice in treatment development schemes. However, modeling of Celiac Disease has been extremely limited, and very few such models exist [14]. Fortunately, autoimmune diseases share commonalities that allow general models to be adapted to a specific disease.

The most common practice is to model the immune response as a system of ODE or PDE equations [15]. Machine learning, usually in the form of support vector machines, random forests, and neural networks, have also been used to model the progression of some autoimmune diseases. However, machine learning has typically been used to aid in diagnosis rather than in prediction of disease progression, often in the form of image analysis [14].

Despite differences in purpose, we can utilize the models developed for diagnosis and treatment in order to construct our own. While many models utilize PDEs which are both spatially and temporally dependent, we will not include spatial dependence in our model due to a lack of information about it. Very little is known about how the geometry of the intestines impacts the immune reaction in Celiac Disease, and a model of such an impact would mostly be guess-work. Hence, we focus on ODE construction. Specifically, we aim to construct an ODE system which describes the immune response in Celiac Disease patients who are on a gluten or gluten-free diet.

1.2.1 ODE construction

There are several ways in which ODEs may be constructed. We focus on two methods: manual construction of ODEs using techniques from systems biology, and using neural networks to estimate governing equations directly from data.

Typically, growth-decay models are used to describe autoimmune reactions. A simple ODE model may consist of immune cells that attack targeted cells, resulting in damaged cells. This model is similar to the SIR model, and may be freely adapted to be more or less complex [16, 17].

Using machine learning methods to model autoimmune disorders has seen recent growth, but these models are typically used to predict outcomes based directly on data. Models which directly estimate ODEs from data are underutilized. We will therefore explore modeling Celiac Disease using neural networks, specifically by utilizing current advancements on neural network discovery of ODE systems in order to predict governing equations from patient data. We will explore whether physics-informed neural networks (PINNs) are adaptable to biological data, which is typically far more noisy and sparse than other data. Specifically, we will explore the Sparse-Regression PINNs (SR-PINNs) developed by [18], which has been curated specifically for sparse and noisy data.

1.3 Data collection

There are a multitude of studies conducted on Celiac Disease patients, including drug trials, gluten challenges, responses to gluten free diets, and more. However, much of the data collected in these studies is confidential, and access to it is highly restricted. There are several datasets that describe individual patient data, but oftentimes values are measured only once, which is not useful data if we want to develop a time dependent model of the reaction of Celiac Disease patients. We have found four studies that collect time dependent data, three of which we will use due to the fourth having extremely limited information (data found in [19]). We have labeled these datasets A-C. Here we present the first several (5-6) entries of each dataframe, and a brief overview of the study's methods. The full dataframes can be found in the cited papers.

In the studies shown, villi size is measured as villi height vs. crypt depth, called Vh:CrD. This measures the average value of several measurements taken of the length of the villi vs. the depth of the tissue between the villi. A higher ratio indicates greater villi height and less potential cellular damage, indicating healthier intestinal walls. A lower ratio indicates potential villi and/or tissue damage [9, 20, 21]. Notably, this measurement is controversial due to high variability and lack of specificity, as well as subjectivity in how heights and depths are measured. However, this is the standard way to document intestinal damage in Celiac Disease [22].

Dataframe A [9]

Patient ID	Diet	Time on diet (days)	Gluten ingested (g)	IgA-TG2 (U/ml)	CD3- IELs (cells/mm)	Vh:CrD ratio
1	0	–	–	10	33	2.8
1	1	29	5.0	319	88	0.8
2	0	–	–	9	43	3.5
2	1	84	4.9	13	84	3.4
3	0	–	–	5	64	2.9
3	1	38	4.9	10	128	1.3

Table 1.1: First six entries of Dataframe A, with data from [9]. Diet: 0 = gluten-containing, 1 = gluten free diet.

In this study, 25 patients with diagnosed Celiac Disease are put on a gluten challenge for up to 12 weeks (84 days), and are measured at weeks 4, 8, and 12. However, only data of a single measurement per patient is provided. Each patient had been on a gluten free diet for at least two years (730 days), with a median of 11 years on a strict gluten free diet and a range of 2-34 years, and were in clinical remission. The exact amount of time each patient has been on a gluten free diet is not specified. The average amount of gluten consumed per day is provided in ‘Gluten ingested (g)’.

It is important to note that because this study is conducted with volunteers who are known to have Celiac Disease, it is very likely that patients with very severe cases are not included in the study, as following a gluten diet for twelve weeks (or even for four weeks) could be debilitating. Hence, these patients are likely representative of less severe cases of Celiac Disease.

Dataframe B [20]

ID	Sex	Age	Diet	Days on diet	Coh.	IgA-TG2 (U/ml)	IgG-DGP (U/ml)	CD4+ (%)	TCR- $\gamma\delta$ CD3+ IELs (%)	Vh:CrD
1	F	22	0	–	1	13.7	38	0.3	2.7	–
1	F	22	1	28	1	8.2	33	0.3	–	–
1	F	22	1	70	1	5.5	29	0.2	–	–
1	F	22	1	182	1	2.2	15	0	–	–
1	F	22	1	364	1	–	–	–	–	–

Table 1.2: First five entries (first patient) from Dataframe B, with data from [20]. Diet: 0 = gluten-containing, 1 = gluten free; Coh. = Cohort.

In this study, 33 patients with diagnosed Celiac Disease are followed during their first year on a gluten free diet after diagnosis, and are measured at different intervals. Three Cohorts of patients are measured. Cohort 1 was measured on days 0, 28, 79, 182, and 364, with CD3+ IELs and Vh:CrD ratio only measured on days 0 and 364. Cohort 2 was measured on days 0, 3, 7, 21, and 365, with CD3+ IELs and Vh:CrD ratio only measured on days 0 and 365. Cohort 3 was measured on days 0, 7, 14, 21, 28, and 365. CD3+ IELs and Vh:CrD ratio are not measured in patients in Cohort 3.

Dataframe C [21]

ID	Sex	Age at dx	Age at study	Diet	Mo. on GFD	Hrs since gluten	Gluten (g)	IL-2 (ng/ml)	CD8+ (%)	TCR- $\gamma\delta$ CD3+ IELs (%)	CD3- IELs (%)
1	F	31	24	0	75	–	–	0.5540	0.01530	1.56	38.41
1	F	31	24	1	75	4	10	9.4020	–	–	–
1	F	31	24	1	75	36	10	–	1.74540	–	–
2	M	20	11	0	9	–	–	0.1300	0.07080	10.90	23.55
2	M	20	11	1	9	4	10	12.3420	–	–	–

Table 1.3: First five entries of Dataframe C, with data from [21]. Diet: 0 = gluten-containing, 1 = gluten free diet (GFD); dx = diagnosis; Mo. = months; Hrs = hours.

This study follows 17 patients with Celiac Disease who have been on a gluten free diet for at least one month. Baseline CD8+ T-cells, IL-2, CD3+ IELs, and CD3- IELs are measured before the trial. The patients then undergo a gluten trial: each is given 10 grams of gluten per day for 3 days. Blood samples are taken 4 hours after the first dose of gluten, measuring levels. Samples are additionally taken 6 days after the first dose of gluten measuring CD8+ T-cells.

1.3.1 Problems with collected datasets

Dataset timelines do not align

The discovery of ODEs on this data assumes that the patients' data all follows the same timeline. However, patients from the same study were not necessarily seen on the same days—e.g. Patient 1 may have data from days 1 and 50, whereas Patient 2 has data from days 1 and 27. One fix would be removing days that do not align, which introduces bias and removes our already limited data. A better approach would be interpolating between days in order to standardize the timeline used.

Collected data is patient-specific

Because the data is patient-specific (i.e. is longitudinal), discovery of ODEs on the data does not make sense directly, since computation of derivatives is not possible. There are several possible solutions.

First, we could use each patient as an independent dataset and train the model with each dataset, as described in [18]. Specifically, in [18], all datasets are stacked, and the model is trained simultaneously on the stacked data using a combined loss function. If we treat every patient data as an independent dataset, we would have over 100 datasets, which would result in high complexity and computational costs.

Another solution is to compress the data across patients, for instance by taking the mean concentrations. This way we could train the model on one dataset, and therefore avoid catastrophic interference. However, this would lead to extremely sparse datasets, and a lot of the information we gathered would be lost.

Finally, we could make a model that uses information from several patients, and predicts the disease progression of a single patient. We will pursue this method initially. If the model performs well, we then treat each patient as an independent dataset and train simultaneously on the stacked data, learning independent parameters for each patient.

Different studies measure different aspects of the immune response

As described in 3.2.1, the concentrations of the immune response components are dependent on one another. However, there does not seem to be a standardized set of immune markers to measure

within studies. Additionally, there is not even a standardized way to measure specific components. For example, samples of IgA and IgG are measured with units $\frac{U}{ml}$, where U stands for ‘units’. The ‘units’ are dependent on the type of test done to measure IgA and IgG, and differ depending on the study. This results in data between independent studies being non-comparable and difficult to interpret for quantitative analysis.

Missing immune markers

Not all relevant variables are measured in the datasets we are working with. Specifically, IL-15, IL-21, and IFN- γ are not measured. While IFN- γ is important for modelling Celiac Disease, its role largely corresponds to that of IL-21. Additionally, it has been shown that IL-15 and IL-21 share many redundant functions, including up-regulation of cytotoxicity in IELs as well as promotion of the production of IFN- γ [6]. However, it would be preferable to have measurements of IL-15, especially in comparison to IL-21, in order to be able to accurately model the immune response of Celiac Disease.

Due to the above problems, we explore several methods of cleaning and interpreting the data that we have, including imputation on the datasets to fill in missing entries (Chapter 2). We additionally use our datasets to generate synthetic patients with more measurements in Chapter 3.

1.4 Neural Network discovery of ODE and PDE systems

Deep neural networks are a common tool utilized in a variety of machine-learning applications due to their ability to discover complex patterns from data. We will briefly describe how Neural Networks are designed, and how Neural Network discovery of ODE systems works.

A neuron is a function $v : \mathbb{R}^d \rightarrow \mathbb{R}$, s.t.

$$v(x) = \sigma(w^\top x + b),$$

for all $x \in \mathbb{R}^d$. Here $w \in \mathbb{R}^d$ is a weight vector, $b \in \mathbb{R}$ is the bias, and $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is an activation function. A (feedforward) neural network is a function that utilizes layers of neurons, taking the neurons of the l^{th} layer as the inputs to the neurons in the $(l+1)^{\text{st}}$. The neural network is trained using gradient-based descent methods, optimizing over the internal parameters of the network in order to minimize the difference between the output of the neural network (called the prediction) and the data provided. Assuming the data is fixed, this difference is described by a loss function $\mathcal{L} : \Theta \rightarrow \mathbb{R}$. Typically, $\Theta \subset \mathbb{R}^p$, where $p \in \mathbb{N}$ denotes the total number of trainable parameters (weights and biases) [23].

Recent advances in neural networks have led to the introduction of physics-informed neural networks (PINNs), which are well known as alternatives to traditional PDE solvers in high-dimensional systems. PINNs incorporate the governing PDE into the loss function and penalize deviations from the PDE at collocation points where derivatives are computed using automatic differentiation. Therefore, PINNs can learn solutions of the PDE using limited observational data. However, this usage of PINNs relies on an explicit form of PDEs, which in our case we do not know. Instead we will use PINNs for data-driven discovery of an ODE system, which we explain below.

Consider a set of nonlinear ODEs in the form

$$\dot{u} + \mathcal{N}[u; \lambda] = 0,$$

where $u = u(t) \in \mathbb{R}^n$; $t \in [0, T]$, is the solution, $\dot{u}$ is the first time derivative, and $\mathcal{N} : \mathbb{R}^n \times \Lambda \rightarrow \mathbb{R}^n$ is a nonlinear operator parameterized by $\lambda \in \Lambda \subset \mathbb{R}^m$, $m \in \mathbb{N}$.

The data-driven discovery of an ODE system using PINNs results both in estimating the solution u and learning of the parameter(s) λ which describes the provided data. Let $u_\theta : [0, T] \rightarrow \mathbb{R}^n$ be a neural-network approximation of the solution u , where u_θ is parameterized by the network's weights and biases, denoted by $\theta \in \mathbb{R}^{N_\theta}$. We define

$$f_\theta(t; \lambda) := \dot{u}_\theta(t) + \mathcal{N}[u_\theta(t); \lambda],$$

where the derivative $\dot{u}_\theta$ is obtained through automatic differentiation. Given observations $\{(t_d^i, u^i)\}_{i=1}^{N_d} \subset [0, T] \times \mathbb{R}^n$ and a set of collocation points $\{t_f^i\}_{i=1}^{N_f} \subset [0, T]$ where the physical structure of the ODE is enforced, we learn the parameters θ and λ by minimizing the loss given by:

$$\begin{aligned}\mathcal{L}_d(\theta) &= \frac{1}{N_d} \sum_{i=1}^{N_d} \|u_\theta(t_d^i) - u^i\|_2^2, \\ \mathcal{L}_f(\theta, \lambda) &= \frac{1}{N_f} \sum_{i=1}^{N_f} \|f_\theta(t_f^i; \lambda)\|_2^2, \\ \mathcal{L}(\theta, \lambda) &= \mathcal{L}_d(\theta) + \mathcal{L}_f(\theta, \lambda).\end{aligned}$$

Here $\mathcal{L}_d$ enforces adherence to the data, while $\mathcal{L}_f$ enforces the ODE structure at the collocation points [24].

The method developed by [24] requires high amounts of observation data to achieve any reasonable degree of accuracy. Fortunately, advancements to neural network discovery has also led to a novel approach of PDE discovery using sparse regression on noisy datasets [18], which we will adapt for discovery of ODEs.

The method described by [18] is explained in detail in Chapter 4. The authors use the following approach to discover derivatives on sparse and noisy data:

1. **Construct a neural network approximation of the data:** A neural network approximation of the data u_θ is fitted by training using the loss function

$$\mathcal{L}_d(\theta) = \frac{1}{N_d} \sum_{i=1}^{N_d} \|u_\theta(t_d^i) - u^i\|_2^2.$$

Time derivatives are computed using automatic differentiation on the collocation points $\{t_c^i\}_{i=1}^{N_c} \subset [0, T]$ and stacked into $\dot{U}_\theta \in \mathbb{R}^{N_c \times n}$.

2. **Define a function library:** Define a library of candidate functions Φ (composed of e.g. polynomials and interaction terms) evaluated at the collocation points $\{t_c^i\}_{i=1}^{N_c}$. If $u \in \mathbb{R}^n$ and there are M candidate terms, then

$$\Phi_{i,:} = [\phi_1(u_\theta(t_c^i)), \dots, \phi_M(u_\theta(t_c^i))]; \quad \Phi \in \mathbb{R}^{N_c \times M}.$$

Initialize a coefficient matrix $\Lambda \in \mathbb{R}^{M \times n}$.

3. **Update coefficients via sparse regression:** Estimate the sparse coefficients that map library terms to the time derivatives by minimizing

$$\mathcal{L}_p(\Lambda) = \frac{1}{N_c} \|\dot{U}_\theta - \Phi \Lambda\|_F^2 + \alpha \|\Lambda\|_0,$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\Lambda\|_0 = |\{(i, j) : \Lambda_{i,j} \neq 0\}|$ promotes sparsity of Λ . Minimizing this loss yields a discovered ODE $\dot{u}(t) \approx \Phi \Lambda$. Note that $\|\cdot\|_0$ measures the number of nonzero elements of a matrix (or vector) and is therefore not a true norm, so minimization of this loss is an NP-hard problem. This is discussed in more detail in 4.1, and the issue is addressed by the algorithm described in 4.2.

In practice, these steps are alternated: u_θ is refined, Φ is recomputed, and Λ is re-estimated until convergence to the solution is achieved.

1.5 Project approach

Our goal is to develop a method to construct an ODE model of the Celiac Disease immune response in an individual patient which can be used for a variety of applications, including estimating the effects of treatments and long-term prognosis. Using the data we have collected, we will impute missing data points (Chapter 2) and generate synthetic patients that may be used to train our model (Chapter 3). We will manually formulate a system of ODEs that captures key components of the Celiac immune response (Section 3.2). Finally, we will use our data to train a Sparse-Regression PINNs model, which will result in a system of ODEs describing the immune reaction of Celiac Disease patients (Chapter 4).

Chapter 2

Data processing

Because we are handling longitudinal data, data processing is not straightforward. Additionally, because we have multiple datasets from different studies, each dataframe must be treated individually and undergoes a separate process of cleaning. For example, while it makes sense to standardize the timeline on a gluten diet for Dataframe A [9], standardization of the timeline during the gluten trial for [21] would not, since the patients are measured only twice for each concentration, and imputing these missing values would introduce a lot of bias.

2.1 Study types

The first several entries of each study are provided in Section 1.3. Notably, dataframes A and C only provide a maximum of two measurements per concentration for each patient, which is generally far less useful than three or more measurements of patients due to limitations in models that can be applied to such data [25]. Usage of these datasets will largely be dependent on whether it makes sense to estimate additional measurements for patients, thus providing more data that can be utilized by more advanced analysis models.

2.2 Exploratory data analysis

We will perform preliminary exploratory data analysis (EDA) on our dataframes. We primarily explore the trajectories of variables in time, and briefly examine the distributions of variables through histograms.

2.2.1 Dataframe A

Important to note is that Dataframe A has no missing values. Hence, imputation will not be necessary. We analyze the distributions of IgA-TG2, CD3+ IELs, and Vh:CrD over time. Additionally, we analyze the distribution of the baseline values of each.

IgA-TG2

The distributions of baseline and post-gluten-diet IgA-TG2 measurements are visualized in Figure 2.1. Both distributions are right-skewed. However, it is apparent that many patients have far greater values of IgA-TG2 after consuming gluten.

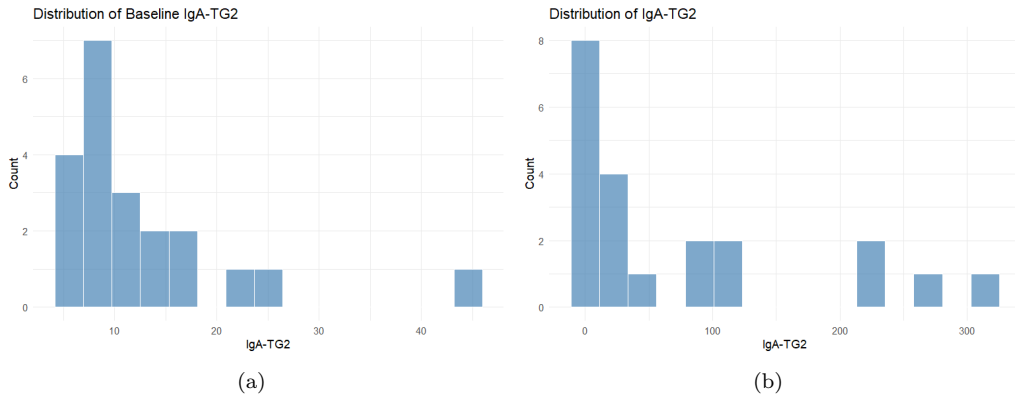


Figure 2.1: (a) Histogram of baseline IgA-TG2 measurements. (b) Histogram of IgA-TG2 measurements after following a gluten diet.

The trajectories of IgA-TG2 are visualized for each patient in Figure 2.2. Notice that some patients seem to have a very steep increase in IgA-TG2, but some patients actually see a decrease in IgA-TG2.

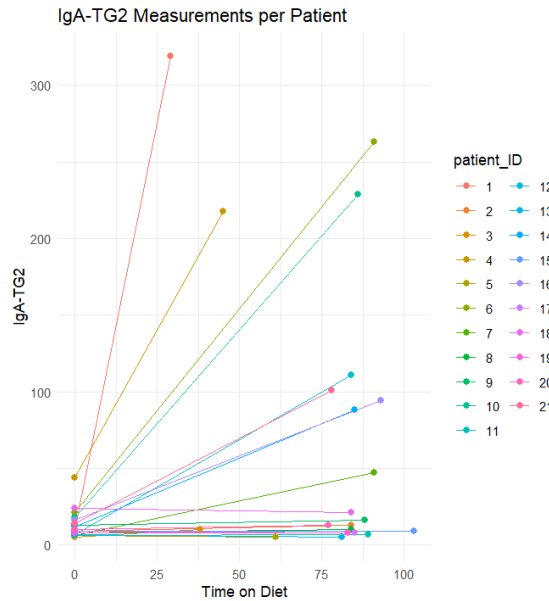


Figure 2.2: Trajectories of IgA-TG2 over time.

CD3+ IELs

Histograms of baseline CD3+ measurements and measurements of CD3+ after consumption of gluten are visualized in Figure 2.3. Interestingly, the distribution of CD3+ becomes centered around 80-100 after gluten consumption, while baseline values tended to be centered around 40-60.

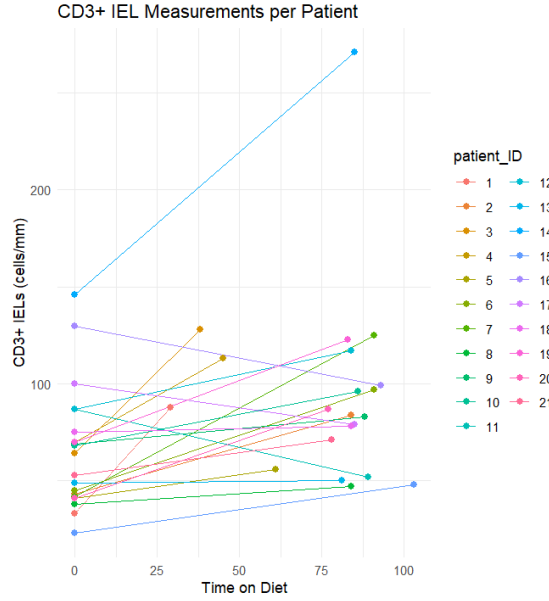


Figure 2.4: Trajectories of CD3+ over time.

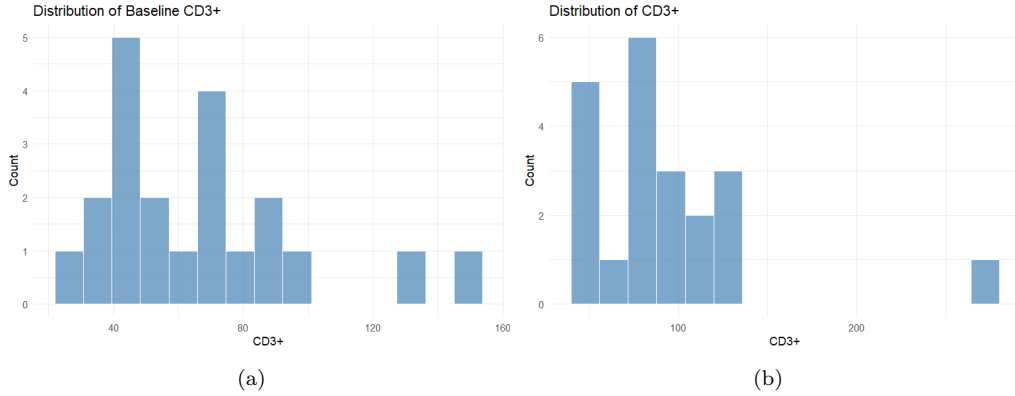


Figure 2.3: (a) Histogram of baseline CD3+ IEL measurements. (b) Histogram of CD3+ measurements after following a gluten diet.

The trajectories of CD3+ measurements for each patient are visualized in Figure 2.4. Several patients see a decrease in CD3+, but most patients have an increase in CD3+, especially those who were measured early in the study. It is possible that CD3+ drops off after long-term gluten consumption.

Vh:CrD ratio

The distributions of Vh:CrD at baseline and after gluten consumption are shown in Figure 2.5. Notice that after gluten consumption, the distribution appears to be bimodal and left-skewed, with many patients with a Vh:CrD ratio below 2.

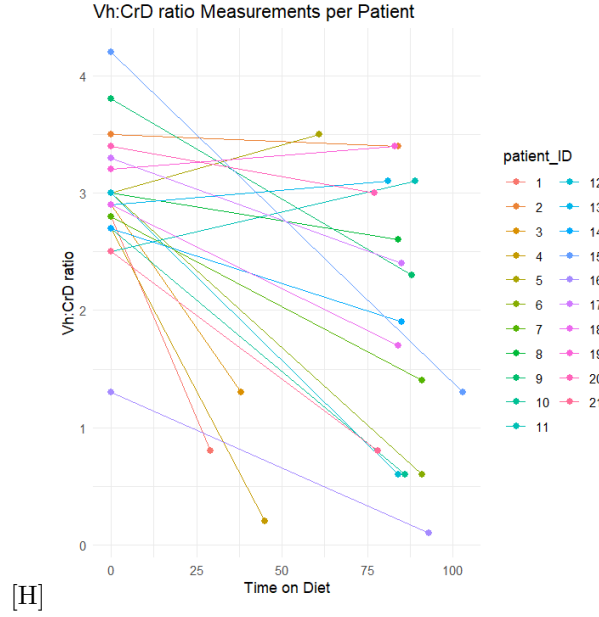


Figure 2.6: Trajectories of Vh:CrD over time.

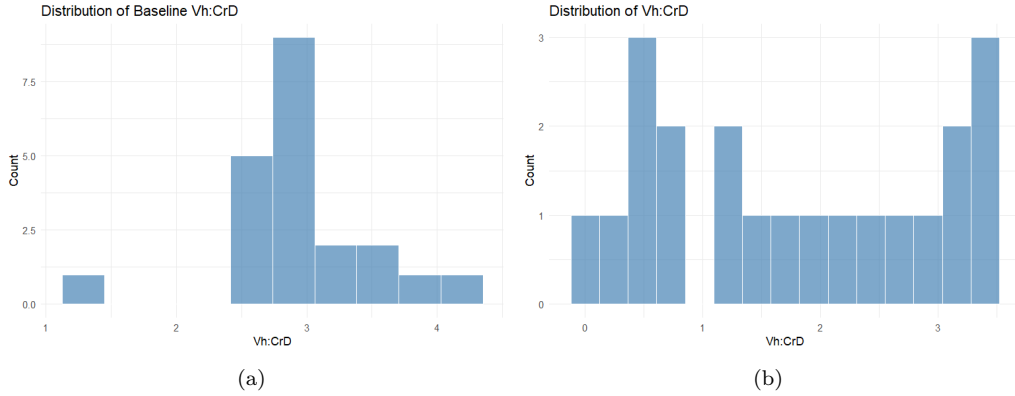


Figure 2.5: (a) Histogram of baseline Vh:CrD measurements. (b) Histogram of Vh:CrD measurements after following a gluten diet.

The trajectories of Vh:CrD ratios per patient are visualized in Figure 2.6. While a few patients had an increase in Vh:CrD, the large majority of patients saw a decrease in Vh:CrD. We expect to see a decrease in Vh:CrD, since the reaction response in Celiac Disease is often characterized by inflammatory cells attacking intestinal cells. However, some people with Celiac Disease do not exhibit typical symptoms, and instead the disease manifests extra-intestinally. The patients whose Vh:CrD ratio increased may have had other symptoms that are not described by this dataset. Due to the high variation in all variables, it does not make sense to impute values in order to standardize patient timelines, as this would result in highly variable and biased estimations.

2.2.2 Dataframe B

The distribution of each variable is pictured in Figure 2.7. Additionally, the percentage of all missing values is presented in Figure 2.7.a. (% NA).

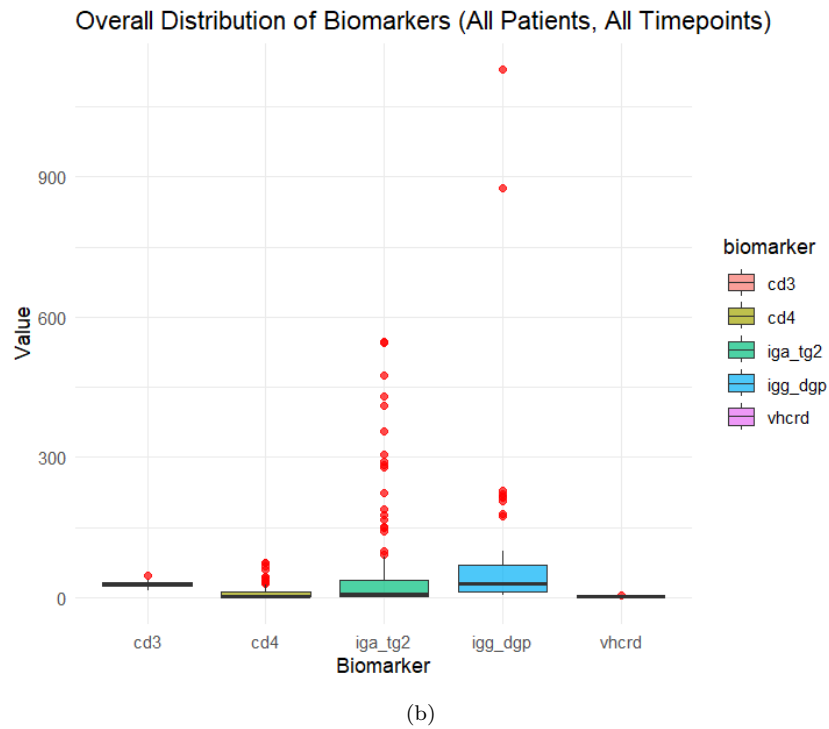
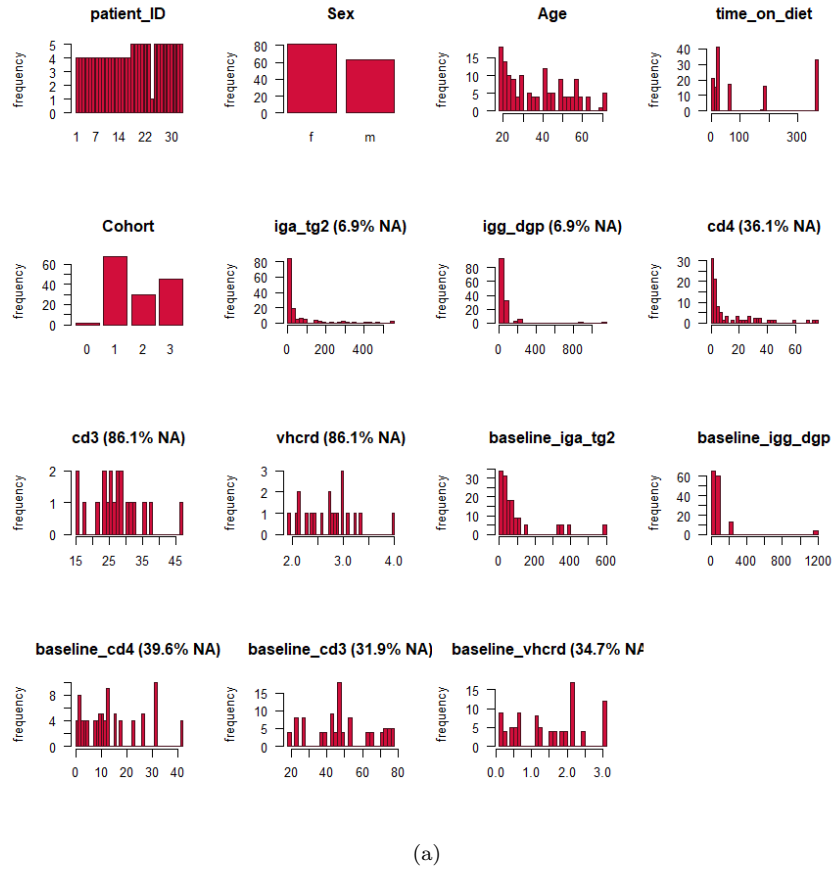


Figure 2.7: (a) Histogram of all variables in Dataframe B. (b) Boxplot of all biomarker values in Dataframe B.

Dataframe B has a lot of missing values. According to the study designers, most of these are due to patients not being able to attend a scheduled appointment for an unknown reason [20]. Some of them are due to measurements for certain variables only being taken twice—at baseline and post-trial. We are additionally missing measurements for CD4+ T-cells, CD3+ IELs, and Vh:CrD ratio in Cohort 3 of Dataframe B. This type of missing data is called systematically missing. There are specific ways of imputing this data, which we will experiment with in Section 2.4. The other missing data is sporadically missing—that which is not intentionally missing but is caused by factors unexplained by the study’s structure. Imputing this type of missing data typically introduces less bias.

The missing data patterns are pictured in Figure 2.8. Additionally, the percentage of missing data (vs. present data) is shown in Table 2.1. Several baseline values for patients are missing, specifically in Cohort 1. Important to note is that missing values of IgA-TG2 and IgG-DGP always appear together, and in Cohorts 1 and 2 also always appear with missing values of all three other measurements, indicating that these values are likely missing due to a missed appointment or other scheduling issue, which is confirmed by the study [20]. However, missing values of CD3+ IELs and Vh:CrD typically appear together intentionally—the designers of the study only included measurements of both quantities at baseline and post-trial. Hence we can assume that IgA-TG2, IgG-DGP, and CD4+ IELs values are likely missing at random, but we cannot assume that about CD3+ IELs or Vh:CrD values, which is likely to be a problem for imputation of these values. Additionally, while the proportion of missing data in Cohorts 1 and 2 is relatively high (at .125 and .157 respectively), the large majority of the missing data comes from the intentionally missing variables CD3+ IELs and Vh:CrD. Therefore, while we may use imputation on CD3+ IELs and Vh:CrD values, model development should probably be conducted only on the non-imputed values.

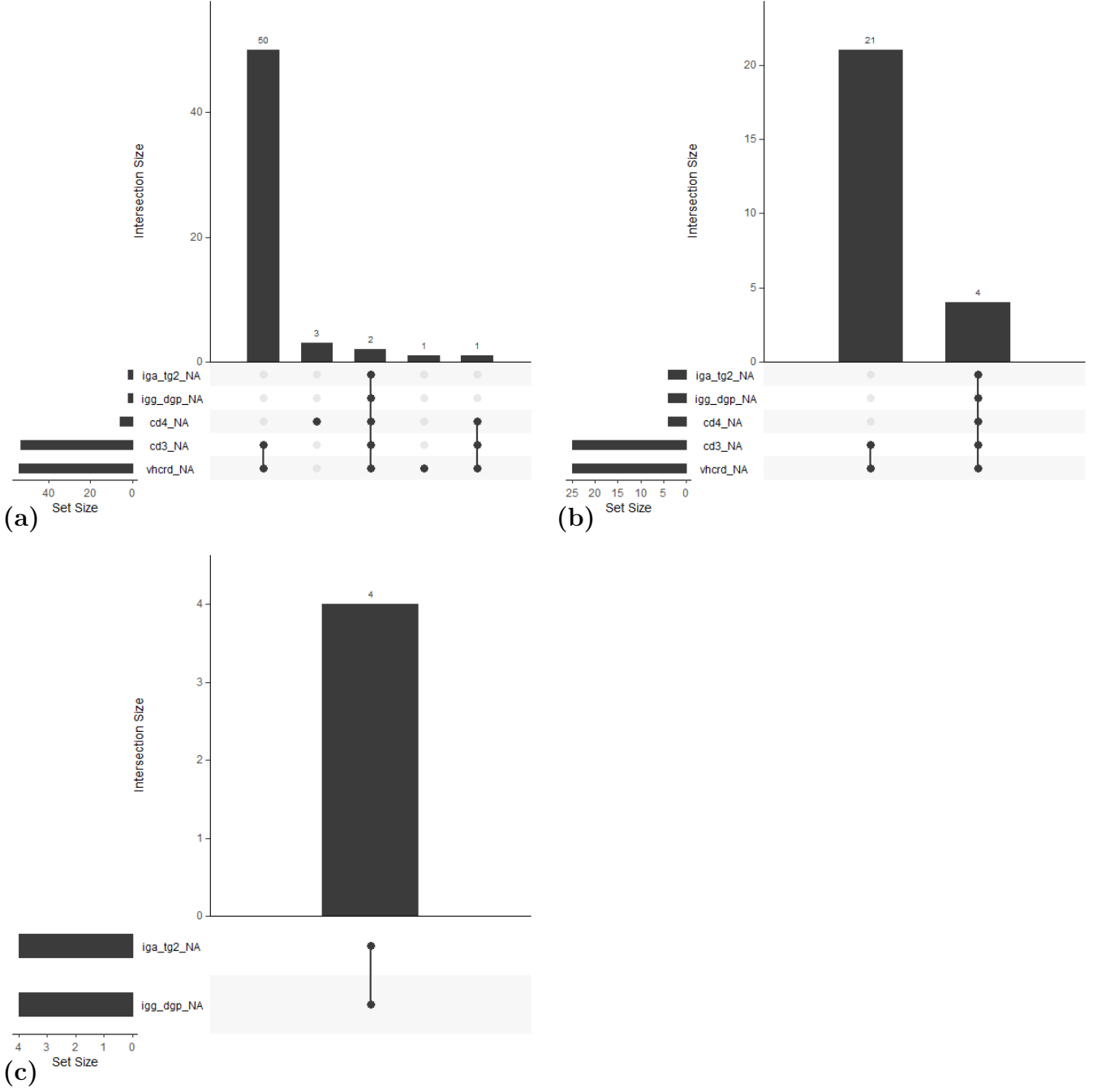


Figure 2.8: Missing data in Dataframe B for Cohorts 1 (a), 2 (b) and 3 (c).

	Present data (%)	Missing data (%)
Cohort 1	87.5%	12.5%
Cohort 2	84.3%	15.7%
Cohort 3	98.1%	1.9%

Table 2.1: Percentage of missing data in Cohorts 1, 2, and 3 of Dataframe B

We provide the trajectories of the measured variables over time before imputation. Additionally,

summary statistics and distributions are visualized.

IgA-TG2

A histogram of the IgA-TG2 values across all patients and time points is provided in Figure 2.9.a. Additionally, the distributions of IgA-TG2 in each timepoint in each Cohort are provided in Figure 2.9.b-d. Notably, the distributions are all right-skewed. Additionally, in later time-points, especially at the last measurement (on days 364 or 365), the distribution becomes far less varied and tends to have a distinct peak.

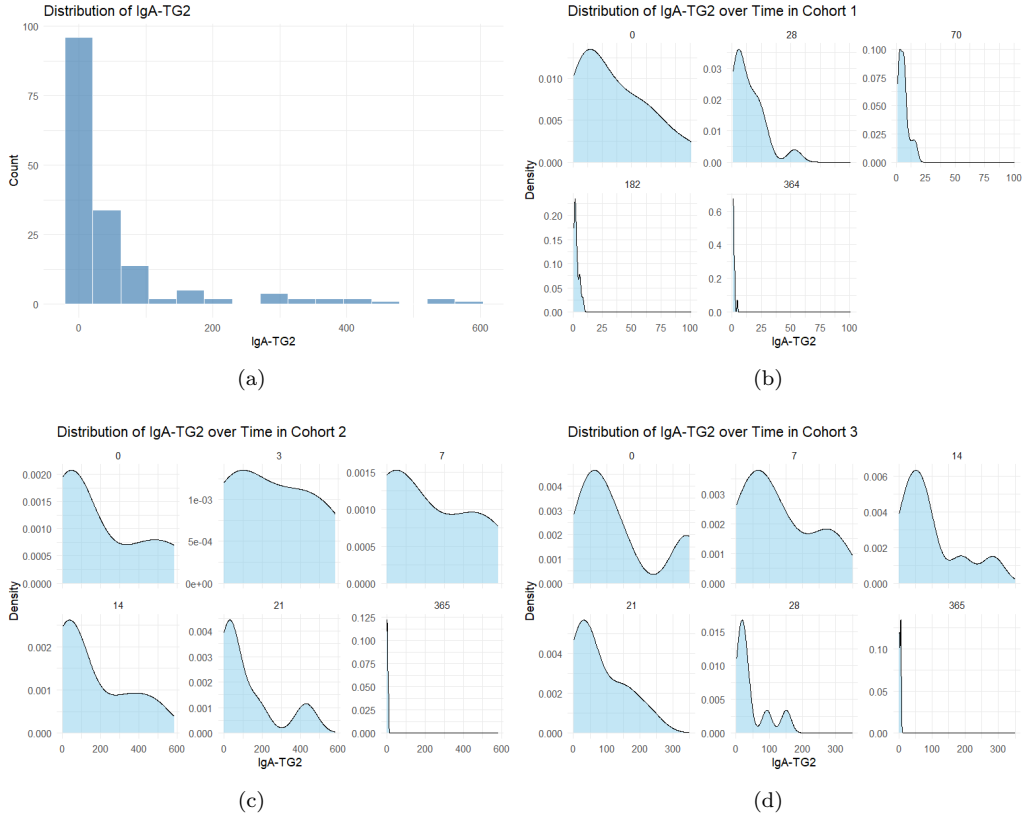


Figure 2.9: (a) Histogram of IgA-TG2 values. (b) Distribution of IgA-TG2 values in Cohort 1 over time. (c) Distribution of IgA-TG2 values in Cohort 2 over time. (d) Distribution of IgA-TG2 values in Cohort 3 over time.

The trajectories of IgA-TG2 over time are provided across all Cohorts (Figure 2.10). The median trajectory is also plotted. There tends to be a significant decrease in the last measurement, which makes some sense as the last measurement is done close to eleven months after the first several measurements in most patients, giving the patients time to recover from their gluten diet.

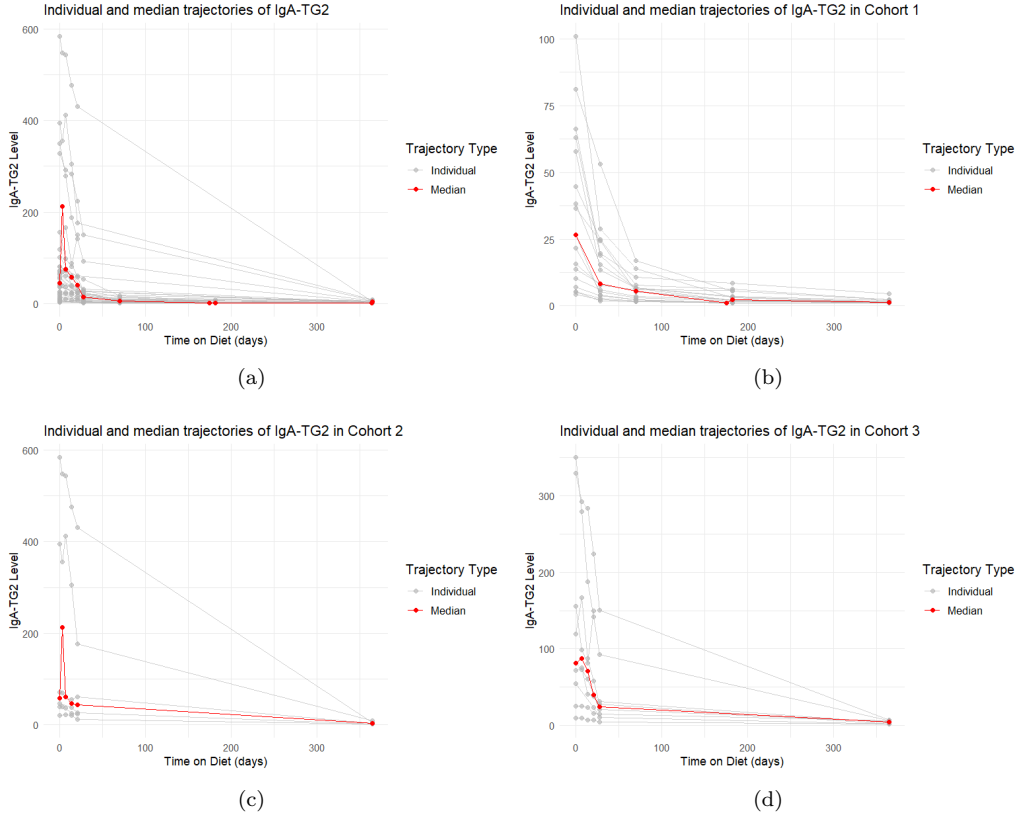


Figure 2.10: Trajectories of IgA-TG2 over time. (a) Trajectories across all cohorts. (b) Trajectories in Cohort 1. (c) Trajectories in Cohort 2. (d) Trajectories in Cohort 3.

The median, mean, and quantiles of IgA-TG2 are provided in Table 2.2. The statistics are taken over all patients over all timepoints. Notably, Cohort 1 has a much smaller range and skew than the other two Cohorts.

Cohort	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
All	1.00	3.60	13.40	59.99	56.90	583.90
1	1.00	1.95	5.20	12.30	13.55	101.00
2	1.30	21.55	43.15	153.67	318.25	583.90
3	1.60	10.25	40.30	81.51	114.08	350.00

Table 2.2: Summary statistics of IgA-TG2.

IgG-DGP

A histogram of the IgG-DGP values across all patients and time points is provided in Figure 2.11.a. Additionally, the distributions of IgG-DGP in each timepoint in each Cohort are provided in Figure 2.11. Note the major outlier with IgG-DGP measurements close to 1200. This is Patient 13, who is part of Cohort 1. The distributions in Cohort 1 look largely bimodal except for this patient, with a strong right skew. In Cohort 2, the measurements are more left-skewed until the final measurement, which is unimodal and has very little variance. The distributions in Cohort 3 are more right-skewed and bimodal. The differences in distributions between Cohorts is interesting, and the reasons for the differences are not clear from the study [20].

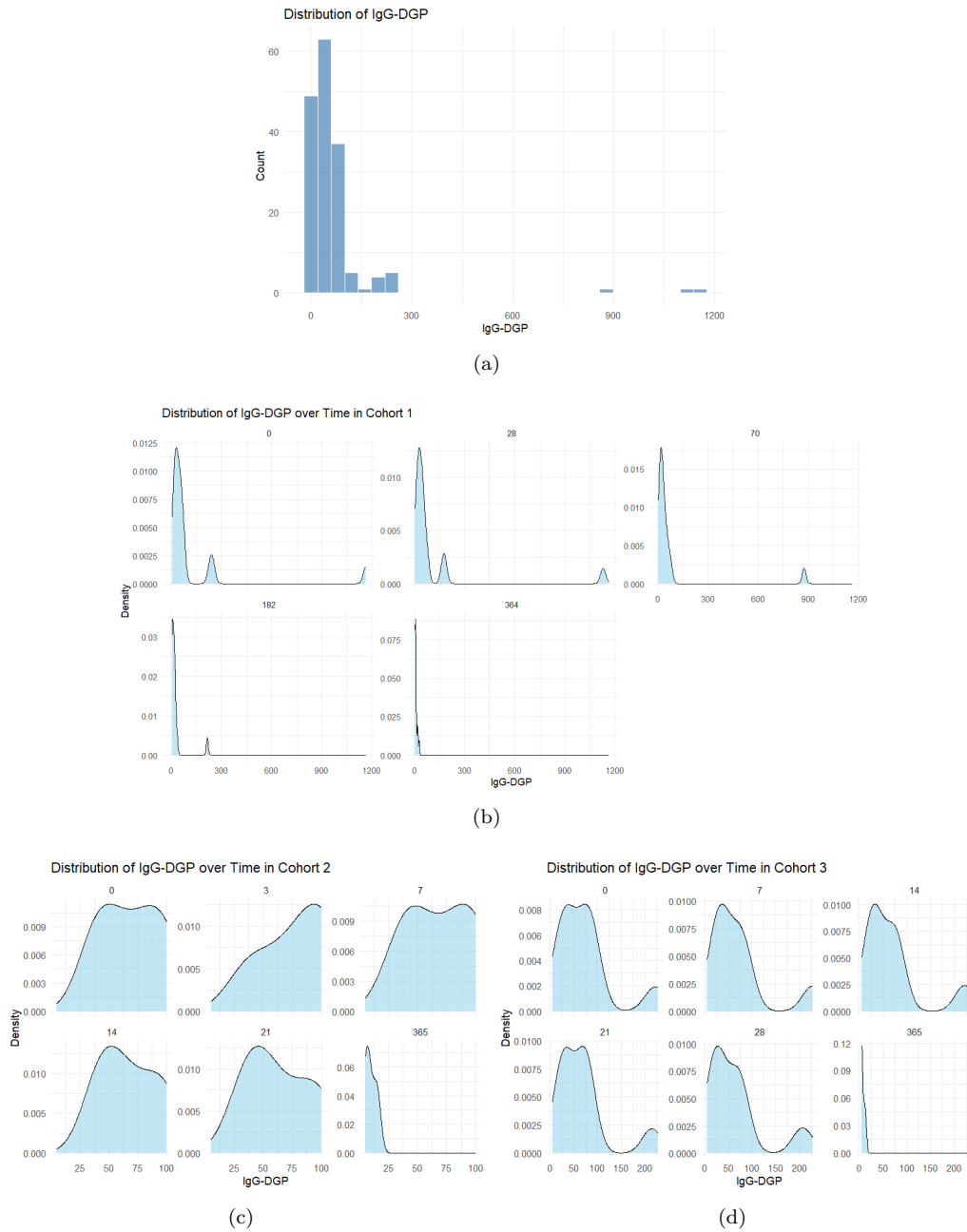


Figure 2.11: (a) Histogram of IgG-DGP values. (b) Distribution of IgG-DGP values in Cohort 1 over time. (c) Distribution of IgG-DGP values in Cohort 2 over time. (d) Distribution of IgG-DGP values in Cohort 3 over time.

The trajectories of IgG-DGP over time are provided across all Cohorts (Figure 2.12). Note that Figure 2.12 (b) and Figure 2.12 (d) do not include Patient 13, who is a major outlier. The median trajectory is also plotted. There seems to be an overall decrease in IgG-DGP over time.

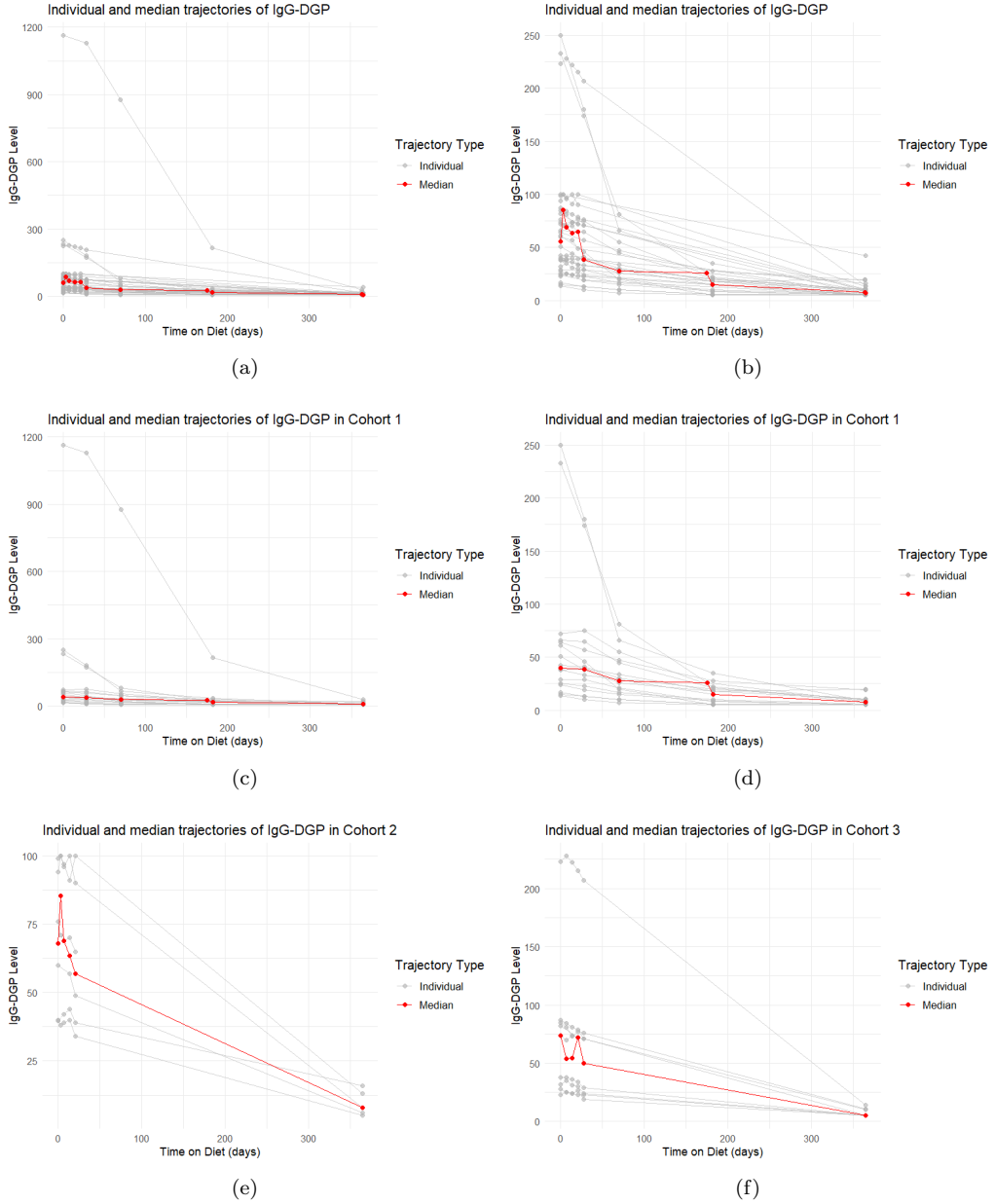


Figure 2.12: Trajectories of IgG-DGP over time. (a) Trajectories across all cohorts and (b) trajectories across all cohorts without major outlier Patient 13. (c) Trajectories in Cohort 1 and (d) trajectories in Cohort 1 without major outlier Patient 13. (e) Trajectories in Cohort 2. (f) Trajectories in Cohort 3.

The median, mean, and quantiles of IgG-DGP are provided in Table 2.3. The minimum value is 5 for all Cohorts, which is because any value less than 5 in [20] was marked as “<5”, so we changed these values to 5. The statistics are taken over all patients over all timepoints. Excluding Patient 13, Cohort 1 appears to have less right-skew than the other two Cohorts.

Cohort	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
All	5.00	16.50	34.00	67.43	71.50	1164.00
All without Patient 13	5.00	15.25	34.00	48.43	70.75	250.00
1	5.00	10.00	23.00	74.67	45.50	1164.00
1 without Patient 13	5.00	10.00	21.00	35.69	40.50	250.00
2	5.00	39.00	58.50	58.99	91.75	100.00
3	5.00	24.00	35.50	60.65	76.75	228.00

Table 2.3: Summary statistics of IgG-DGP

CD4+ T-cells

A histogram of the CD4+ T-cell values across all patients and time points is provided in Figure 2.13.a. Additionally, the distributions of CD4+ in each timepoint in each Cohort are provided in Figure 2.13. Cohort 1 appears to be more right-skewed, whereas Cohort 2 does not have a very standard distribution shape between timepoints.

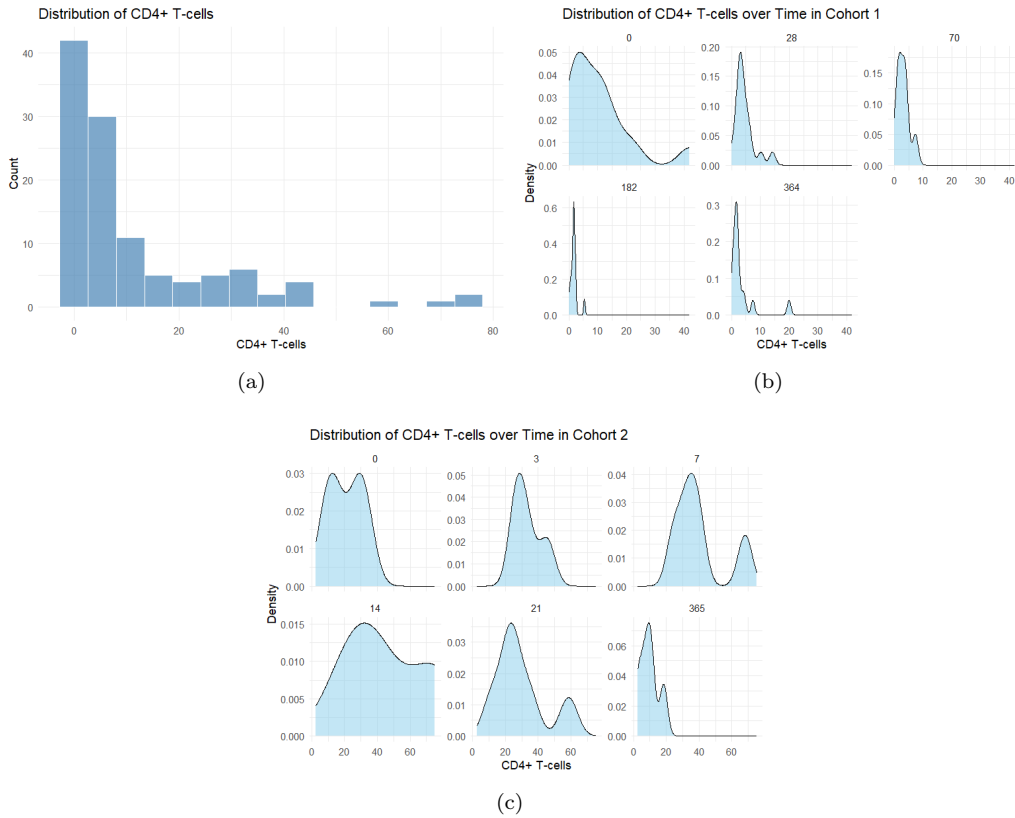


Figure 2.13: (a) Histogram of CD4+ T-cell values. (b) Distribution of CD4+ values in Cohort 1 over time. (c) Distribution of CD4+ values in Cohort 2 over time. (d) Distribution of CD4+ values in Cohort 3 over time.

The trajectories of CD4+ over time are provided across all Cohorts (Figure 2.14). The median trajectory is also plotted. In Cohort 1, there are seven patients who see an increase in CD4+ near the end of the trial. Additionally, in Cohort 2, CD4+ T-Cells increased for the first 14 days after the gluten free diet was started, but decreased after day 14. The reasons for this are described in [20], and seems to be associated with the specific markers that the T-cells carry. Specific data

about these markers are not provided, and therefore for the sake of this paper all CD4+ T-cells are assumed to have roughly the same properties. However, it is interesting that the gluten reaction seems to impact T-cell production and markers even weeks after gluten has been ingested.

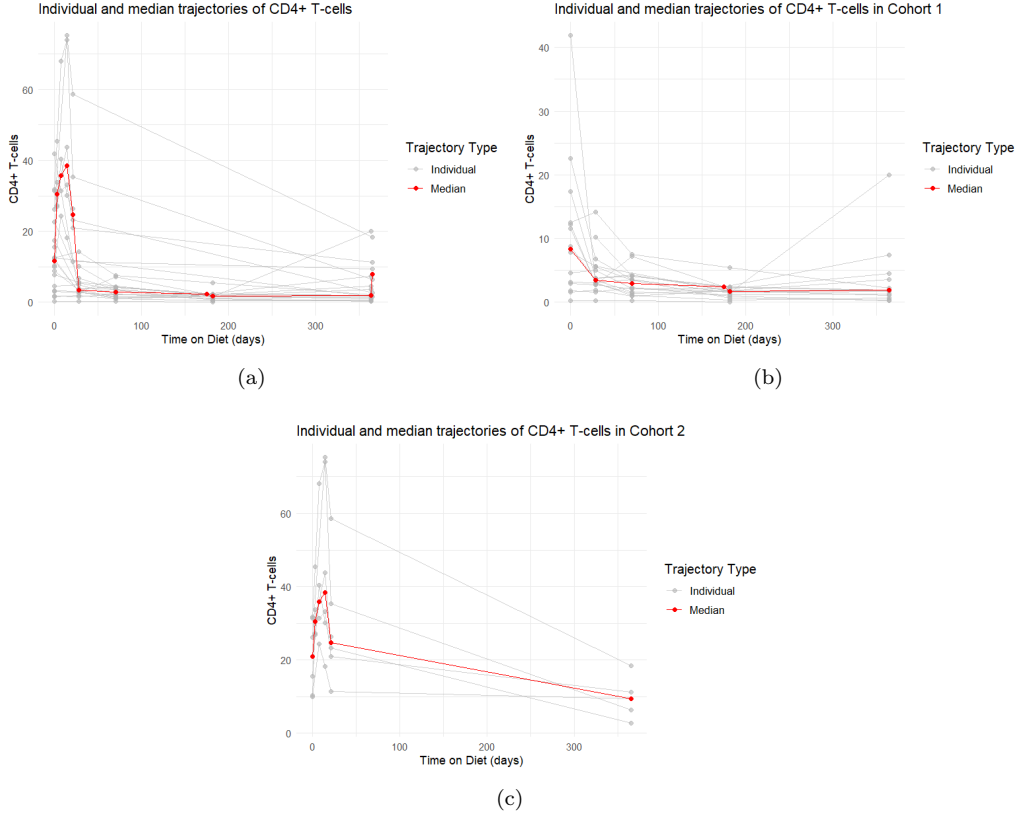


Figure 2.14: Trajectories of CD4+ over time. (a) Trajectories across all cohorts. (b) Trajectories in Cohort 1. (c) Trajectories in Cohort 2. (d) Trajectories in Cohort 3.

The median, mean, and quantiles of CD4+ are provided in Table 2.4. The statistics are taken over all patients over all timepoints.

Cohort	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
All	0	1.80	3.80	11.75	15.60	75.30
1	0	1.600	2.400	4.508	4.550	41.900
2	2.70	17.55	27.10	29.93	35.42	75.30

Table 2.4: Summary statistics of CD4+ T-cells.

CD3+ IELs

A histogram of the CD3+ IELs values across all patients and time points is provided in Figure 2.15.a. Additionally, the distributions of CD3+ in each timepoint in each Cohort are provided in Figure 2.15. Note that CD3+ is only measured twice for each patient. Similarly to CD4+, the distribution of CD3+ in Cohort 2 is bimodal, as compared to the distribution in Cohort 1 which is more unimodal. Additionally, the variance in Cohort 2 decreases significantly in the last measurement.

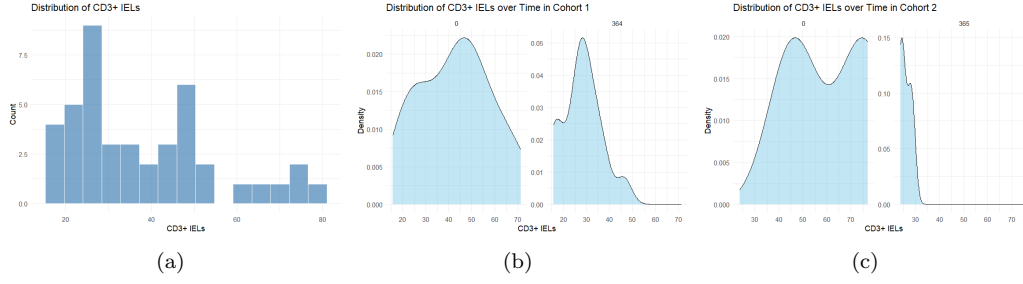


Figure 2.15: (a) Histogram of CD3+ IELs values. (b) Distribution of CD3+ values in Cohort 1 over time. (c) Distribution of CD3+ values in Cohort 1 over time. (d) Distribution of CD3+ values in Cohort 2 over time.

The trajectories of CD3+ over time are provided across all Cohorts in Figure 2.16. The median trajectory is also plotted. In Cohort 1, there are two patients who see an increase in CD3+. These two patients also had positive increases in CD4+ T-cells in the last two measurements. However, no patients have an increase in CD3+ in Cohort 1.

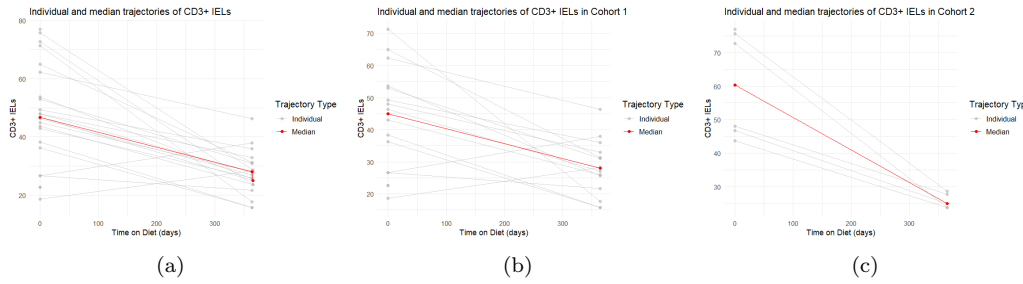


Figure 2.16: Trajectories of CD3+ IELs over time. (a) Trajectories across all cohorts. (b) Trajectories in Cohort 1. (c) Trajectories in Cohort 2. (c) Trajectories in Cohort 3.

The median, mean, and quantiles of CD3+ are provided in Table 2.5. The statistics are taken over all patients over all timepoints.

Cohort	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
All	15.70	25.85	33.00	38.21	47.35	77.00
1	15.70	25.93	32.15	35.95	46.30	71.30
2	23.70	26.35	43.70	44.78	60.35	77.00

Table 2.5: Summary statistics of CD3+ IELs

Vh:CrD ratio

A histogram of the Vh:CrD ratios across all patients and time points is provided in Figure 2.17.a. Additionally, the distributions of Vh:CrD in each timepoint in each Cohort are provided in Figure 2.17. Note that Vh:CrD is only measured twice for each patient. Interestingly, the Vh:CrD ratio becomes left-skewed in Cohort 2 in the last measurement.

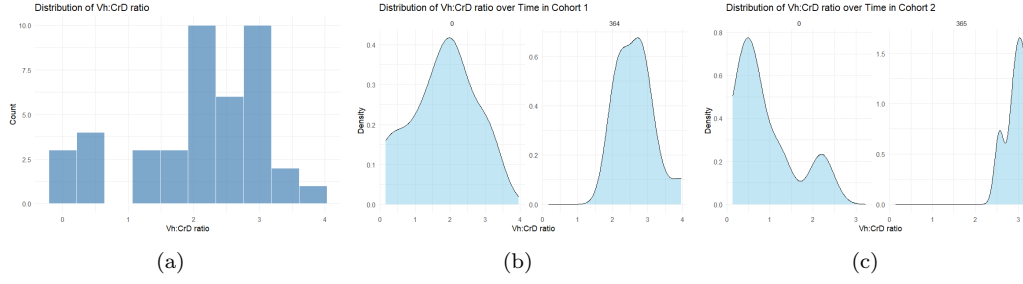


Figure 2.17: (a) Histogram of Vh:CrD ratio values. (b) Distribution of Vh:CrD values in Cohort 1 over time. (c) Distribution of Vh:CrD values in Cohort 1 over time. (d) Distribution of Vh:CrD values in Cohort 2 over time.

The trajectories of Vh:CrD ratios over time are provided across all Cohorts (Figure 2.18). The median trajectory is also plotted. In two patients in Cohort 1, the Vh:CrD ratio decreases over time. One of these patients (Patient 3) also had an increase in CD4+ T-cells and CD3+ IELs. However, the other patient (Patient 10) did not have an increase in either CD4+ or CD3+. It is not clear why this decrease in Vh:CrD ratio occurred; potentially, the patients ate gluten. However, this cannot be known for certain and was not discussed in [20].

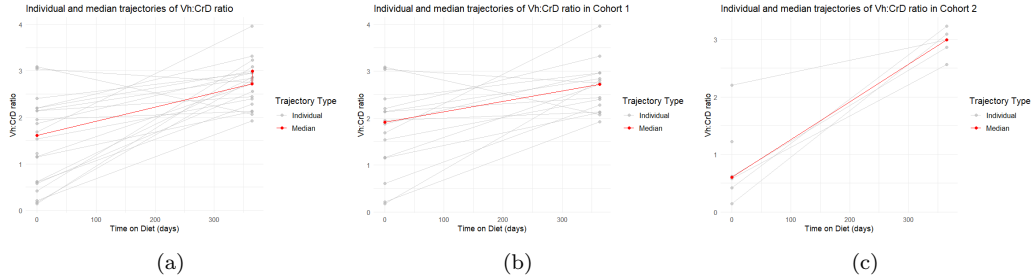


Figure 2.18: Trajectories of Vh:CrD ratios over time. (a) Trajectories across all cohorts. (b) Trajectories in Cohort 1. (c) Trajectories in Cohort 2. (c) Trajectories in Cohort 3.

The median, mean, and quantiles of Vh:CrD ratios are provided in Table 2.6. The statistics are taken over all patients over all timepoints.

Cohort	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
All	0.140	1.577	2.200	2.096	2.855	3.960
1	0.170	1.895	2.200	2.197	2.820	3.960
2	0.140	0.600	2.200	1.810	2.925	3.230

Table 2.6: Summary statistics of Vh:CrD ratios.

2.3 Dataframe C

While Dataframe C has inconsistent measurement points (IL-2 is measured at hour 4, while CD8+ T-cells are measured on day 6), there are no missing values. Because imputation would likely be highly biased, we will not impute CD8+ T-cell values at hour 4 or IL-2 at day 6. Additionally, only baseline values of CD3+ and CD3- IELs are taken. Using these values to model time dependent data would not make sense, but they may be used for other purposes. We will therefore analyze the distributions of IL-2 and CD8+ T-cells over time, and the distribution of baseline CD3+ and CD3-.

IL-2

The distributions of baseline and post-gluten-trial IL-2 measurements are visualized in Figure 2.19. Both distributions are right-skewed, but patients tend to have elevated levels of IL-2 after the gluten challenge. This is also represented in Figure 2.20, which visualizes the change in IL-2 after the gluten challenge. Most patients experience an increase in IL-2 levels.

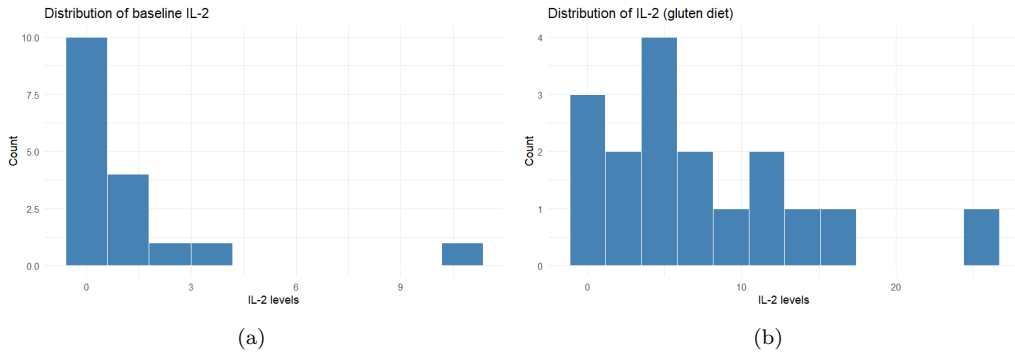


Figure 2.19: (a) Histogram of baseline IL-2 measurements. (b) Histogram of IL-2 measurements after following a gluten diet.

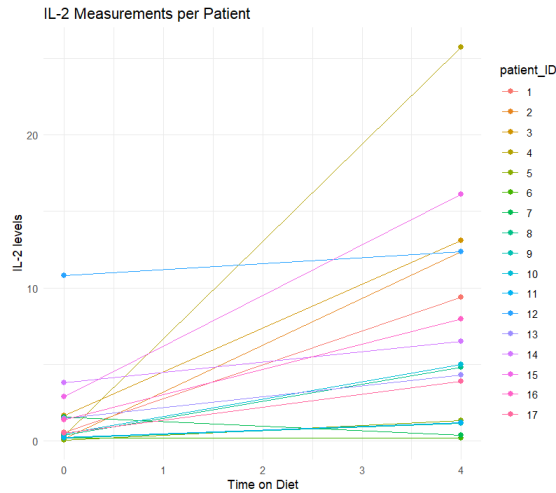


Figure 2.20: Trajectories of IL-2 over time.

CD8+ T-cells

The distributions of baseline and post-gluten-challenge CD8+ T-cell levels are visualized in 2.21. While CD8+ T-cell levels are significantly elevated for some patients after gluten exposure, some see only minor changes in levels.

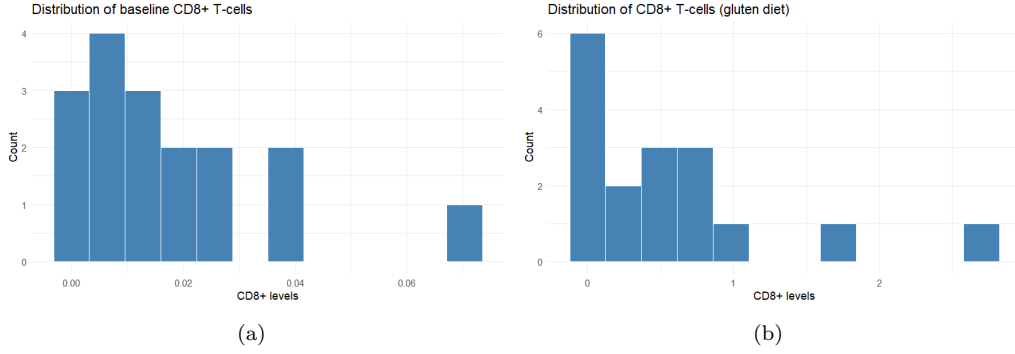


Figure 2.21: (a) Histogram of baseline CD8+ T-cell measurements. (b) Histogram of CD8+ measurements after following a gluten diet.

The trajectories of CD8+ T-cells are visualized in 2.22. Most patients experience elevated CD8+ T-cell levels after gluten exposure, but the degree of elevation is highly variable and patient-specific.

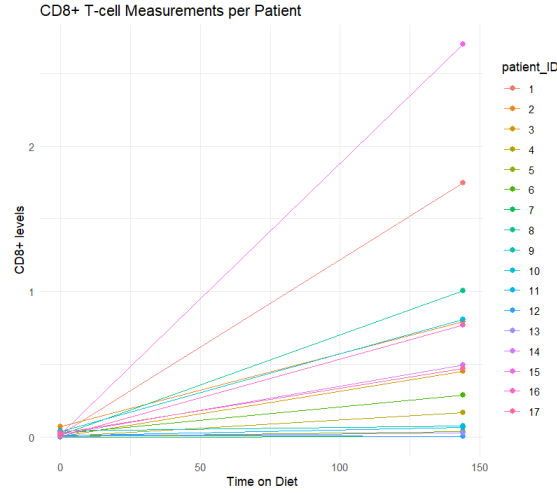


Figure 2.22: Trajectories of CD8+ T-cells over time.

CD3+ IELs

CD3+ IELs are only measured at baseline, pre-gluten-trial. The distribution of CD3+ levels is visualized in Figure 2.23. The distribution is very wide, and CD3+ level vary greatly between patients. One possible explanation for this deviation is that some patients have been on gluten free diets for several years, while others have only been on gluten free diets for a month. However, there are many possible explanations for the high variability in this data.

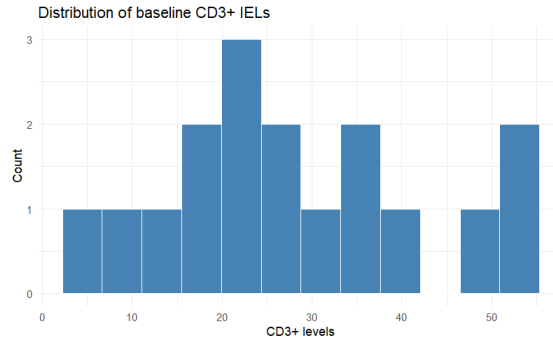


Figure 2.23: Histogram of baseline CD3+ IEL measurements.

CD3- IELs

The distribution of baseline CD3- IELs is visualized in Figure 2.24. Notice that all but one patient have $< 20\%$ CD3- IELs. While there is not much information about whether CD3- IELs decrease rapidly after adoption of a gluten free diet, this may be an explanation for why CD3- levels are relatively low for all patients despite many having high CD3+ levels. However, there are many possible explanations, and due to the lack of information from this study it is difficult to estimate the reason.

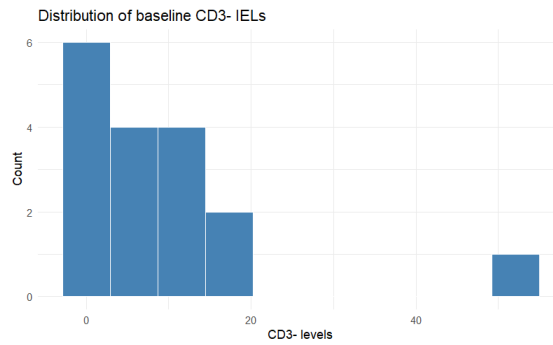


Figure 2.24: Histogram of baseline CD3- IEL measurements.

2.4 Imputation of missing data

Missing data is extremely common in studies, and the solution to missing data is a widely discussed issue. Common solutions include listwise and pairwise deletion, single imputation such as mean and regression imputation, the K-Nearest Neighbor (KNN) method, and multiple imputation. Unfortunately, deletion and single imputation tend to introduce bias or over-estimate the data covariance, leading to results that look better than they should [26, 27]. Additionally, the K-Nearest Neighbor method requires missing data to be missing completely at random (MCAR) [28], which, as discussed in Section 2.4.1, is a very strong assumption to make, and one that in general cannot be tested. Instead, we will explore multiple imputation techniques which can be used for data missing at random (MAR), namely: multivariate imputation by chained equations (MICE) [29] and Jomo [30].

2.4.1 Types of missing data

There are three main classifications of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). We adapt the definitions of the

three from [31] and [32].

Let N be the number of patients and p the number of measured variables. Let T be the number of measurements in each patient. Let Y be the $NT \times p$ matrix of all data, including entries that are missing, labelled as (-). For example, the matrix

$$Y = \begin{pmatrix} 1 & 1 & 22 & 1 & 1 & 28 & 8.2 & 33 & 0.3 & - & - \\ 1 & 1 & 22 & 1 & 1 & 28 & 5.5 & 29 & 0.2 & - & - \\ 1 & 1 & 22 & 1 & 1 & 28 & 2.2 & 15 & 0 & - & - \\ 1 & 1 & 22 & 1 & 1 & 28 & - & - & - & - & - \end{pmatrix},$$

which encodes Patient 1 of Dataframe B, includes entries with (-) corresponding to missing values. We assume each row follows a multivariate normal distribution; $Y_i \sim \mathcal{N}_p(\theta, \Sigma)$, where θ and Σ are unknown. Let O be an $NT \times p$ matrix s.t. $o_{ij} = 1$ if y_{ij} is observed, and 0 if y_{ij} is missing. Then we can write the matrix Y as:

$$\begin{aligned} Y_{\text{obs}} &= \{y_{ij} : o_{ij} = 1\} \\ Y_{\text{miss}} &= \{y_{ij} : o_{ij} = 0\}. \end{aligned}$$

The distribution of O may depend on Y . The relationship between O and Y is called the missing data model, with parameters contained in ψ . The missing data model is described by

$$p(O, Y_{\text{obs}}, Y_{\text{miss}} | \psi).$$

The data is said to be MCAR if

$$p(O | Y_{\text{obs}}, Y_{\text{miss}}, \psi) = p(O | \psi),$$

i.e. the probability of the data being missing is dependent only on ψ , and is independent of both the observed and missing data.

Data is MAR if the probability of missing is dependent on the observed data and the parameters ψ , but not on the missing data itself:

$$p(O | Y_{\text{obs}}, Y_{\text{miss}}, \psi) = p(O | Y_{\text{obs}}, \psi).$$

This type of missing data can usually be predicted when including the observed data in prediction methods.

Finally, data is MNAR if $p(O | Y_{\text{obs}}, Y_{\text{miss}}, \psi)$ does not simplify. In this case, the reason for missingness is dependent on the missing values. This happens e.g. when a patient with a very bad response to a treatment does not show up for measurements because of severe symptoms. This data cannot be predicted using typical techniques, which assume data is MCAR or MAR.

Unfortunately, it is not possible to test whether data is MAR as opposed to missing not at random (MNAR), because the data needed to test this is missing. We will assume all missing data is MAR and use some techniques to improve the probability of this being the case. We can therefore assume O and Y_{miss} are conditionally independent given Y_{obs} .

In addition to the assumption of MAR, the distinctness of the parameters ψ , which describes the missing data model and θ , which describes the full data Y , is important for modeling the missing data. Specifically, ψ is distinct from the parameter θ if

$$p(\theta, \psi) = p(\theta)p(\psi).$$

This is an important assumption of Bayesian inference and of imputation. It implies that the missing data are generated from the same model as the observed data. Hence, we can construct the model of the posterior distribution $p(Y | Y_{\text{obs}}, O)$ from the observed data alone, and use the constructed model to impute the missing data [33].

2.4.2 Multiple imputation

There are two main methods for multiple imputation when multiple variables contain missing values: joint modeling (JM), and fully conditional specification (FCS). Joint modeling assumes that the variables follow a joint multivariate distribution that is estimated and sampled from, directly imputing missing values by random draws. In contrast, FCS estimates a univariate model for each variable and imputes missing values sequentially [34, 35]. We will discuss the FCS technique MICE and the joint modeling technique Jomo below.

Many imputation methods use Gibbs sampling in order to draw random values to replace missing values in the data. Gibbs sampling is discussed below.

Gibbs Sampling

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm used to generate random values from a distribution, even when that distribution is unknown [36]. Consider a joint distribution $p(x, y_1, \dots, y_p)$ and suppose we wish to draw values from the marginal distribution

$$p(x) = \int \dots \int p(x, y_1, \dots, y_p) dy_1, \dots, dy_p.$$

Integration of this joint distribution is often impossible both analytically and numerically, and is typically not feasible. Instead, random values are sampled from $f(x|y)$ and $f(y|x)$, which are typically possible to compute [36]. By taking a large number of samples from the conditional distributions and averaging, we can estimate the marginal distribution since

$$\begin{aligned} p(x) &= \int p(x, y) dy \\ &= \int p(x|y)p(y) dy = \mathbb{E}[p(x|y)]. \end{aligned}$$

In the Bayesian setting, the primary goal is to estimate $p(\theta, \Sigma|Y) \propto p(\theta, \Sigma)p(Y|\theta, \Sigma)$. However, the likelihood $p(Y|\theta, \Sigma)$ cannot be directly computed if the data is missing. Specifically, in the case of missing data, the likelihood is dependent on the missing data model in the form of ψ (assuming the data is MAR):

$$\begin{aligned} p(Y|\theta, \Sigma) &= p(Y_{\text{obs}}, O|\theta, \Sigma, \psi) \\ &= \int p(O|Y_{\text{obs}}, Y_{\text{miss}}, \psi) \cdot p(Y_{\text{obs}}, Y_{\text{miss}}|\theta, \Sigma) dY_{\text{miss}} \\ &= \int p(O|Y_{\text{obs}}, \psi) \cdot p(Y_{\text{obs}}, Y_{\text{miss}}|\theta, \Sigma) dY_{\text{miss}} \\ &= p(O|Y_{\text{obs}}, \psi) \cdot \int p(Y_{\text{obs}}, Y_{\text{miss}}|\theta, \Sigma) dY_{\text{miss}} \\ &= p(O|Y_{\text{obs}}, \psi) \cdot p(Y_{\text{obs}}|\theta, \Sigma). \end{aligned}$$

We want to sample the missing values from the posterior distribution of the missing data and model parameters, given by

$$p(Y_{\text{miss}}, \theta, \Sigma|Y_{\text{obs}}).$$

We can use Gibbs sampling in order to infer θ and Σ and also predict the missing values. Given initial values $Y_{\text{miss}}^{(0)}$ and $\Sigma^{(0)}$, in one step of a Gibbs sampling scheme we do the following [32]:

1. Sample $\theta^{(s+1)}$ from $p(\theta|Y_{\text{obs}}, Y_{\text{miss}}^{(s)}, \Sigma^{(s)})$.
2. Sample $\Sigma^{(s+1)}$ from $p(\Sigma|Y_{\text{obs}}, Y_{\text{miss}}^{(s)}, \theta^{(s+1)})$.

3. Sample $Y_{\text{miss}}^{(s+1)}$ from $p(Y_{\text{miss}}|Y_{\text{obs}}, \theta^{(s+1)}, \Sigma^{(s+1)})$.

The key to the Gibbs sampler is that we are drawing from univariate conditional distributions, meaning all but one of the random variables are assigned fixed values. This should make sampling easier, as these conditional distributions are typically simpler than joint distributions. It has been shown that the Gibbs sequence converges to a stationary distribution independent of the initial values, and this stationary distribution is the target distribution $p(\theta, \Sigma, Y_{\text{miss}}|Y_{\text{obs}})$ [36].

The main difficulty in using Gibbs sampling is in computing the full conditional distributions. Additionally, under assumptions that the data is MAR, the imputation method MICE (explained below) has been shown to have better root mean square error (RMSE) than Gibbs sampling when the amount of missing data is between 0 – 10%, and Gibbs sampling has been suggested to introduce estimate bias in the imputed data at much greater rates than MICE [27]. Finally, the existence of a conditional distribution does not guarantee that the joint distribution (or marginal distribution) exists. The Gibbs sampling strategy is theoretically unsuitable in this case, and in practice may fail to converge.

Multiple Imputation by Chained Equations (MICE)

A common method to implement a pseudo-Gibbs method is Multiple Imputation by Chained Equations (MICE), where the conditionals for the missing data is fully specified and a joint distribution is assumed to exist. MICE is one of the most common FCS models used for imputation of longitudinal data. Important to note is that the existence of the joint distribution is uncertain, which may result in non-convergence of the method. However, this is generally treated as a non-concern, and in fact MICE may perform better than Gibbs in these scenarios [27].

When using MICE, it is important to include all variables that will be later used in the analysis, and it may be valuable to include variables that will not be included in order to improve the imputation results. Including more variables in the imputation process can increase the chances of the data being MAR, reducing bias [37]. Additionally, variables should be imputed in sequential order if the variables are sequential; for instance, repeated response measurements should be imputed in order.

Let Y_j be the j^{th} column of Y ($j \in \{1, \dots, p\}$) and Y_{-j} be the matrix Y without the column j . Let m be the total number of imputations. The MICE algorithm works as follows [37]:

1. A single imputation is performed, usually by replacing each missing value by the column mean. This initial imputation is a placeholder.
2. The placeholder imputations for one variable x_i are set as missing. The other imputations are maintained.
3. The observed values in variable x_i is fitted with a regression model with the other variables in the imputation model. The missing values in x_i are imputed with predictions from the regression model.
4. Steps 2 and 3 are repeated for each variable with missing values. After each variable has been imputed using a regression model, it is said that one cycle has been completed. The number of cycles to be completed is previously specified.
5. Once the specified number of cycles has been completed, steps 1-4 are repeated to create a specified number of new datasets.

Because this imputation method uses linear regression and draws the imputed values from a normal distribution, application of this method is problematic if the data is not normally distributed. There are a few solutions to this problem. One is to transform the variables to be imputed to normal distributions and, after imputation, re-transform the imputed dataset to the original scale.

Another method is to use predictive-mean matching (PMM), a technique which is integrated in the MICE method in R and is described below.

There is not a standardized recommendation for how many cycles should be completed; typically, convergence should be analyzed by observing successive cycles and determining whether significant changes have occurred between cycles. The number of recommended datasets to create is also not standardized; generally, recommendations are dependent on how much data is missing, as well as how correlated the missing values are with the observed values. A general guide is to create 20-40 datasets if feasible [38].

There is a variety of imputation methods that can be applied, and choosing the best method(s) can be both tedious and challenging due to the relative computational costs of each method and the unknown qualities of the data. There are a few standard methods which we briefly describe in Table 2.7. We also describe the methods in more detail below.

Name	R studio function name	Description
Predictive mean matching	<code>pmm</code>	Imputation by predictive mean matching; for single-level data
Bayesian linear regression	<code>norm</code>	Calculates imputations for univariate missing data by Bayesian linear regression
Bayesian linear regression (multi-level)	<code>2l.norm</code>	Imputes missing multi-level continuous data using a Gibbs sampler with a linear multi-level model. Assumes heterogeneous within-group variance.
Bayesian linear regression (multi-level)	<code>2l.pan</code>	The same as <code>2l.norm</code> , but assumes homogenous within-group variance.
Linear mixed effects model	<code>2l.lmer</code>	Imputes missing multi-level continuous values using a linear mixed effects model.
Bayesian linear regression (level 2 variables)	<code>2lonly.norm</code>	Imputes univariate missing data at level 2 using Bayesian linear regression analysis
Predictive mean matching (level 2 variables)	<code>2lonly.pmm</code>	Imputes univariate missing data at level 2 using predictive mean matching

Table 2.7: Mice methods

Predictive mean matching: For each missing value, a predicted value is calculated based on a given model. A group of candidates is chosen from a group that is closest to the predicted value of the missing entry. One candidate is randomly drawn, and their observed value is used to replace the missing value. This method may result in the same donor being used many times, which is particularly problematic if the proportion of missing data is high, or if the sample size is small [31]. This method can be used for single-level data as well as longitudinal data.

Bayesian linear regression: All variables are assumed to be continuous, and the model residuals are assumed to be close to normal. This method can be used for single-level data. The algorithm is adapted from that developed by [39]. It can also be used for multi-level data, in which case it uses a Gibbs sampler [29]. There are multiple methods that can be used to implement a Bayesian linear regression for imputation, as described above, the main differences being the assumption of homogeneity or heterogeneity of within-group variance, i.e. heteroscedasticity vs homoscedasticity or heteroscedasticity.

After datasets are formed, it is generally incorrect to combine the datasets, for instance by averaging the data. Instead, a model should be fitted to each imputed dataset separately and then

combined. Otherwise, results may be biased or result in smaller than expected variances [31].

Jomo (Joint Modeling)

Jomo is a joint modeling package in R studio which uses a multivariate linear mixed model (MLMM) to model the interactions between and within variables. The most common joint model is a multivariate normal model. The complete variables are used as predictors for the incomplete variables, which are treated as outcomes in a linear regression. Let the number of incomplete variables be c and let $Y_1, \dots, Y_r$ be the incomplete columns of our matrix of data Y . Then the joint model can be written as:

$$\begin{aligned} Y_{1,i} &= \beta_{0,1} + \beta_{1,1}Y_{c+1,i} + \beta_{2,1}Y_{c+2,i} + \dots + \beta_{p-c,1}Y_{p,i} + \epsilon_{1,i} \\ &\vdots \\ Y_{c,i} &= \beta_{0,c} + \beta_{1,c}Y_{c+1,i} + \beta_{2,c}Y_{c+2,i} + \dots + \beta_{p-c,c}Y_{p,i} + \epsilon_{c,i} \\ &\begin{pmatrix} \epsilon_{1,i} \\ \vdots \\ \epsilon_{c,i} \end{pmatrix} \sim \mathcal{N}(0, \Omega). \end{aligned}$$

Typically, joint modeling makes sense to use when the analysis focuses on differences between clusters [31]. An important aspect of Jomo as compared to other joint modeling techniques is that Jomo allows for heteroscedacity, i.e. the variance between groups may be different. However, Jomo does ignore non-linearities and may experience difficulties with small datasets. Jomo can be used in conjunction with MICE and implemented for specific variables, while MICE methods are implemented for others. Additionally, Jomo can handle imputation of both level 1 and level 2 variables, and can impute both at the same time. Jomo is generally recommended to be used to model level 2 variables (e.g. baseline variables) instead of MICE [34].

2.4.3 Evaluation of imputation techniques

In general, assessing imputation techniques is difficult because differences between observed and imputed values are not always problematic; in fact, sometimes the nature of the missingness leads directly to imputed results being significantly different than observed results [40]. For instance, in the case of MAR data, the differences in imputed vs. observed data distributions should be explained by the fully observed variables in the dataset, and differences do not necessarily indicate ‘bad’ or ‘problematic’ imputations [41]. Instead, variables that have large differences between imputed vs. observed values should be investigated and evaluated on a case-by-case basis, and decisions on whether the imputed data is reflective of the missing data should be made using domain knowledge.

There are many ways to assess imputed datasets, including error, tests of bias, and other metrics. Additionally, convergence of the imputation algorithm may be assessed using visualizations or criteria such as Gelman-Rubin criteria. Finally, the distribution of the resulting imputed data may be compared to the distribution of the incomplete data. We explain some of these techniques below.

Masked validation

The standard method of evaluating imputation techniques is to artificially mask values within a complete dataset and evaluate performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Deviation (MAD), Bias and Proportion Variance (PV), and use tests such as Repeated Measures of Analysis of Variance (RM ANOVA) to assess how well the imputed values reflect the true data [42, 29, 31]. This can only be done on complete cases. Depending on the data missingness and size of the dataset, it may be possible to use a complete subset of the data and use masked validation to estimate how well the imputation method performs.

```

Class: mids
Number of multiple imputations: 10
Imputation methods:
  patient_ID      Sex      Age      Cohort baseline_iga_tg2 baseline_igg_dgp
  "norm"          "norm"    "norm"    "norm"    "norm"    "norm"
PredictorMatrix:
  patient_ID Sex Age Cohort baseline_iga_tg2 baseline_igg_dgp baseline_cd4 baseline_cd3
patient_ID    0 0 1 1 1 1 1 1
Sex            1 0 1 1 1 1 1 1
Age            1 0 0 1 1 1 1 1
Cohort         1 0 1 0 1 1 1 1
baseline_iga_tg2 1 0 1 1 0 1 1 1
baseline_igg_dgp 1 0 1 1 1 0 1 1
  baseline_vhcrd
patient_ID      1
Sex             1
Age             1
Cohort          1
baseline_iga_tg2 1
baseline_igg_dgp 1
Number of logged events: 601
  it im      dep      meth      out
1 0 0      constant      Sex
2 1 1 baseline_cd4      norm Cohort3
3 1 1 baseline_cd3      norm Cohort1, Cohort3
4 1 1 baseline_vhcrd      norm Cohort1, Cohort3
5 1 2 baseline_cd4      norm Cohort3
6 1 2 baseline_cd3      norm Cohort2, Cohort3

```

Figure 2.25: Summary of MICE imputation of missing baseline values.

Comparing distributions

The distributions of the imputed vs. observed values may be evaluated using visualizations or tests. Tests can be especially difficult to perform on longitudinal data which is time dependent, and must be applied carefully. For instance, a Kolmogorov-Smirnov (KS) test could be applied to the first measurement of each patient, but should not be applied to test the time dependent measurements of patients. The trajectories of patients' measurements over time may be plotted before and after imputation to estimate whether imputed trajectories are different from observed trajectories.

2.4.4 Imputation of Dataset B

MICE

It is important to first separate missing variables based on whether they are “Level 1” or “Level 2” variables. Level 2 variables are those that are independent and constant within a patient, i.e. the baseline values of the measured quantities. Level 1 variables are those that vary within the patient, i.e. the measured variables after some amount of time. In our case, we have two cluster levels: patients and Cohorts. Cohort 3 is missing measured data that Cohorts 1 and 2 have. There are methods for imputing this data which we will use.

First, in order to implement MICE, it is important to have an idea of what variables are random and which are fixed effects. In the mixed-effects models, we use Patient ID as a random effect, and all other variables as fixed effects. When using MICE, we can specify Patient ID as a Level 2 clustering variable, and therefore need not specify it as a random effects. However, Cohort is technically a Level 3 variable, meaning it defines a cluster of Level 2 variables (patients). We cannot specify Level 3 variables in MICE, and we are not interested in examining the variation of measurements across Cohorts, and therefore specify it as a fixed effect. We will implement a two-step MICE procedure since we are missing both baseline (level 2) and time-varying (level 1) values. The imputation is pictured in Figure 2.25. Notice that only baseline CD4+, CD3+, and Vh:CrD are imputed; the baseline values of IgA-TG2 and IgG-DGP are not missing. The imputation method `norm` is used. After imputation, we construct ten imputed dataframes. The imputed values are pictured in Figure 2.28. Additionally, the distributions of the resulting data are visualized in Figure 2.29. Unfortunately, all three baselines are missing high amounts of data,

and the results of the imputations are highly variable.

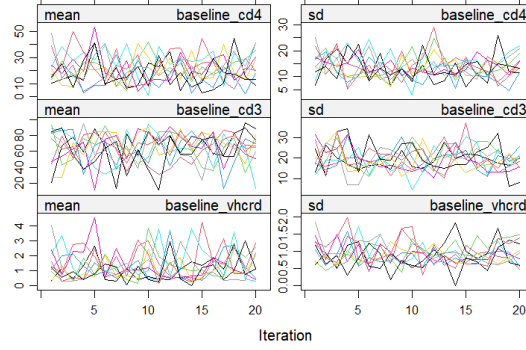


Figure 2.26: Traceplot of MICE imputations of baseline values of CD4+, CD3+, and Vh:CrD.

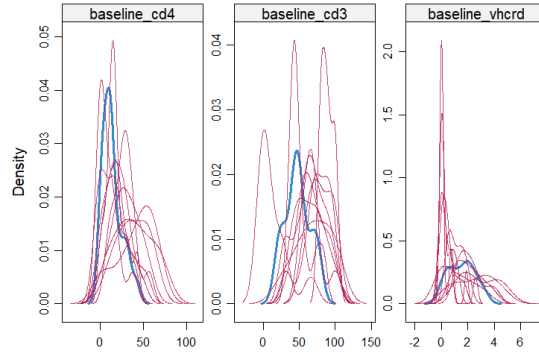


Figure 2.27: Density plot of MICE imputations of baseline values of CD4+, CD3+, and Vh:CrD.

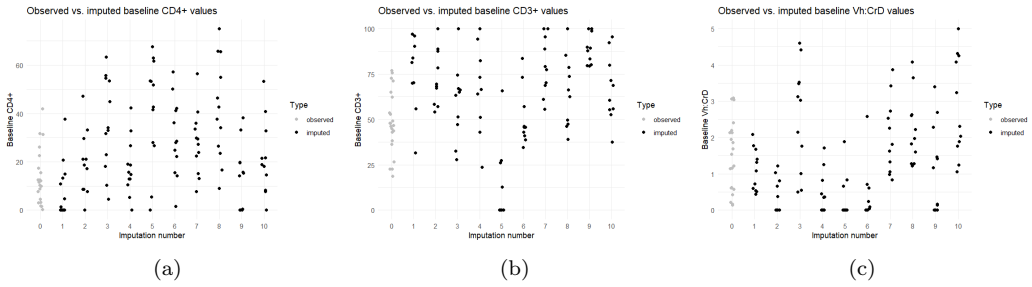


Figure 2.28: Imputed vs. observed values in ten imputed dataframes after MICE imputation of (a) CD4+ T-cells, (b) CD3+ IELs, and (c) Vh:CrD ratios. Imputed values are black.

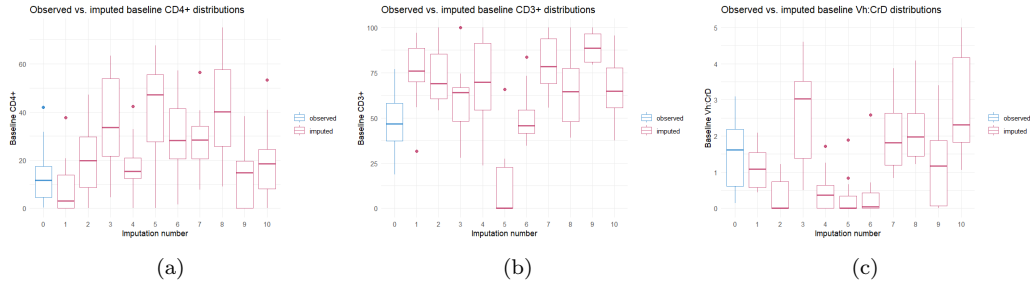


Figure 2.29: Distributions of baseline values of (a) CD4+ T-cells, (b) CD3+ IELs, and (c) Vh:CrD ratios after imputation by MICE.

After imputation of the baseline variables, we use the resulting dataframes to impute the level 1 variables. The imputation is summarized in Figure 2.30. We use imputation method `21.pan` to impute the level 1 variables. We construct ten complete imputed datasets.

```

Class: mids
Number of multiple imputations: 10
Imputation methods:
  patient_ID Sex Age time_on_diet Cohort iga_tg2
  "21.pan"    "21.pan" "21.pan"    "21.pan" baseline_iga_tg2 baseline_iga_dgp
  baseline_cd4 baseline_cd3 baseline_vhcrd
PredictorMatrix:
  patient_ID Sex Age time_on_diet Cohort iga_tg2 iga_dgp cd4 cd3 vhcrd baseline_iga_tg2
patient_ID 0 0 1 1 0 1 1 1 0 0 1
Sex -2 0 1 1 0 1 1 1 0 0 1
Age -2 0 0 1 0 1 1 1 0 0 1
time_on_diet -2 0 1 0 0 1 1 1 0 0 1
Cohort -2 0 1 1 0 1 1 1 0 0 1
iga_tg2 -2 0 1 1 0 0 1 1 0 0 1
  baseline_iga_dgp baseline_cd4 baseline_cd3 baseline_vhcrd
patient_ID 0 0 0 0
Sex 0 0 0 0
Age 0 0 0 0
time_on_diet 0 0 0 0
Cohort 0 0 0 0
iga_tg2 0 0 0 0

```

Figure 2.30: Summary of MICE imputation of missing level 1 variables.

The traceplots are pictured in Figure 2.31. Unfortunately, based on the oscillating pattern of the means of CD3+ and Vh:CrD, there is evidence that the method did not converge. Additionally, when examining the density plots (Figure 2.32), we see that the densities of the imputed values do not align with the densities of the observed values. It is not very surprising that the method does not converge, as there are many missing values in CD3+ and Vh:CrD. Based on the traceplots and density plots, it appears that the method converged for the rest of the variables.

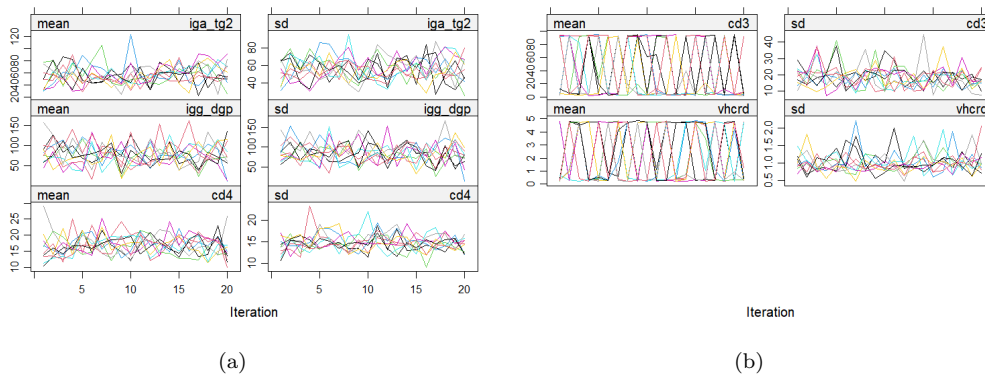


Figure 2.31

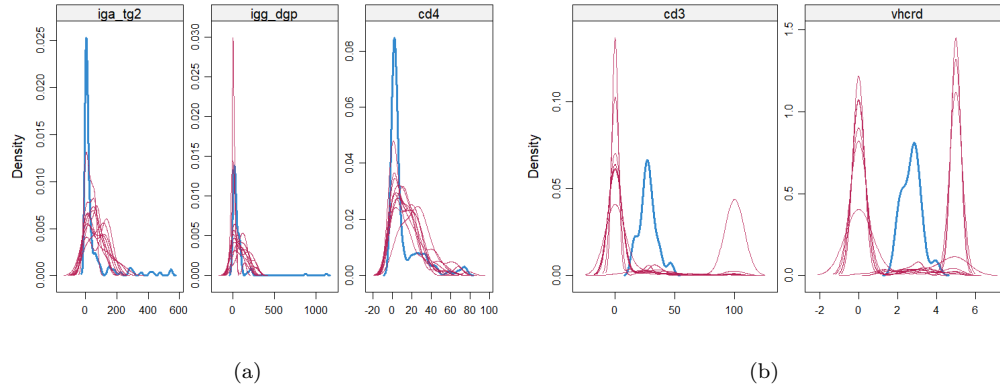


Figure 2.32

The values of the ten imputations performed on the first dataset (where baseline values are given by the first imputed dataframe) are visualized in Figure 2.33. Additionally, the distributions of the resulting values are pictured in Figure 2.34. The distributions of IgA-TG2 and IgG-DGP do not appear to be extremely variable, but considering the wide range of the observed values, this may not be a good thing. It is interesting that all of the imputed values of both variables are relatively small, and we should examine why this is—it may be because all of the missing values are at later timepoints. Clearly the imputed values of CD3+ and Vh:CrD are extremely random, which further suggests a lack of convergence. It may be preferable to use a different method for these variables.

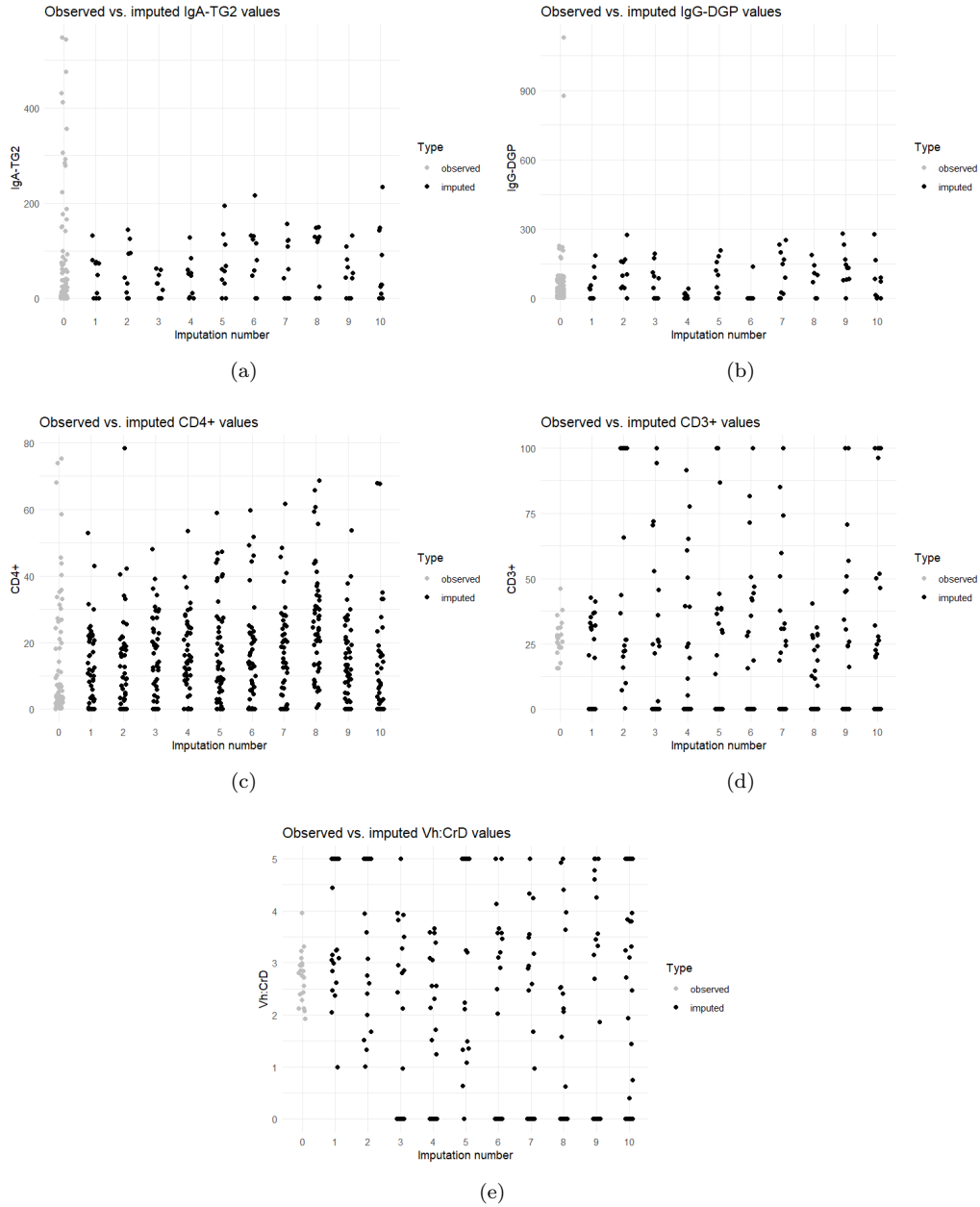


Figure 2.33: Imputed vs. observed values in ten imputed dataframes after MICE imputation of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios. Imputed values are black.

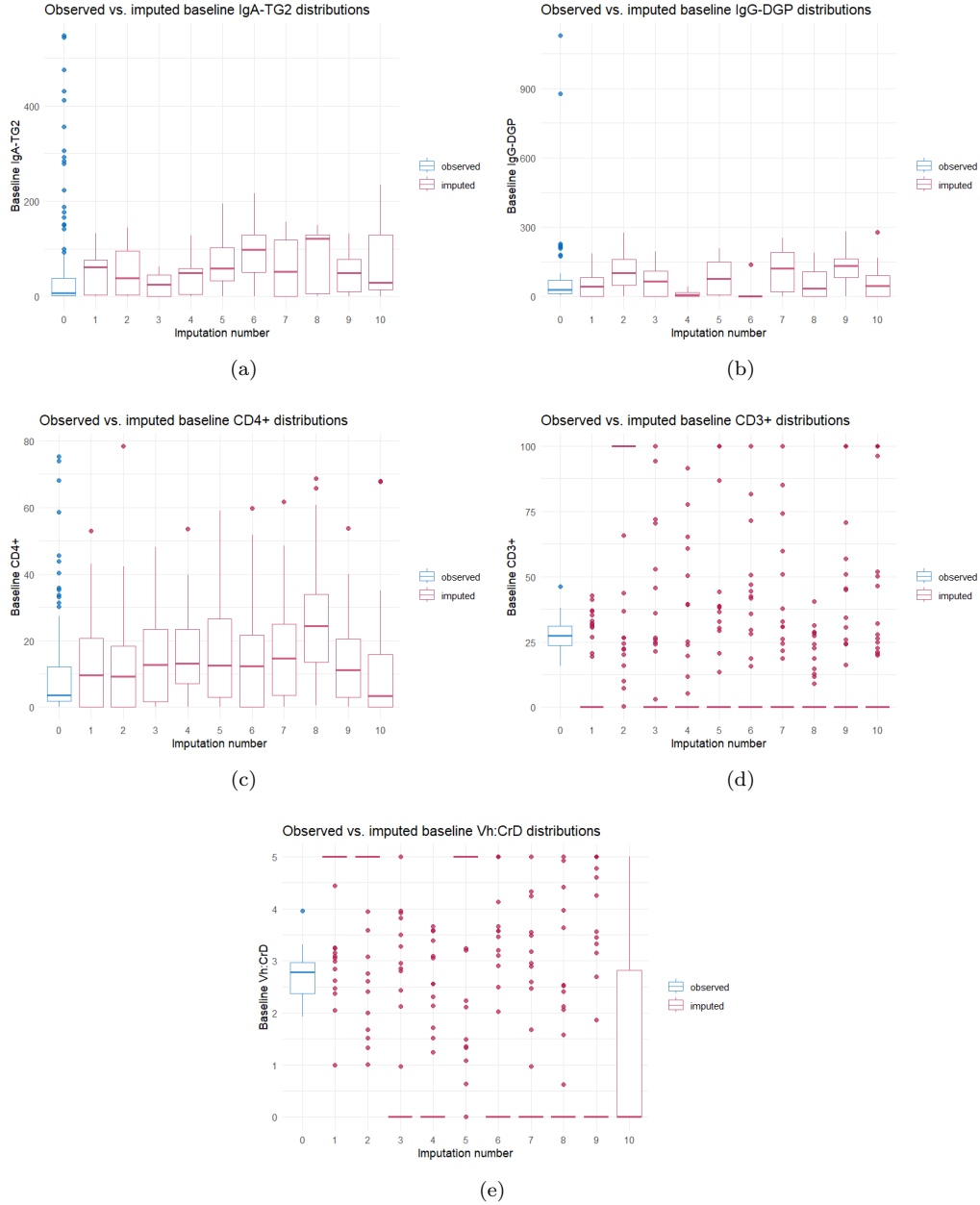


Figure 2.34: Distributions of values of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios after imputation by MICE.

We plot the trajectories of IgA-TG2 and IgG-DGP in patients who had missing values (Figure 2.35). The imputed values are shown. Trajectories are plotted by Cohort for improved readability. We see that the trajectories of IgA-TG2 seem to make sense, in general monotonically decreasing over time except for in patients 18 (Cohort 2), 29, and 33 (Cohort 3). It should be noted that not all observed patients have monotonically decreasing values of IgA-TG2, as shown in Figure 2.10. Many of the imputed values are the final measured values, which gives some explanation of why the imputed values have a smaller range than the observed values. Additionally, despite the wide range of the observed IgA-TG2 values, the mean, median, and third quadrant values are all significantly smaller than the maximum values, which additionally supports the smaller range in imputed values. We see that the trajectories of IgG-DGP are slightly more variable in

Cohorts 2 and 3. While the IgG-DGP measurements of patients in Cohorts 2 and 3 are not always monotonically decreasing, the initial variation in measurements is not as great as in the imputed patients'. Notice the difference in skew between days 0, 3, 7, and 14 in Figure 2.11. This may be source of the variance in the IgG-DGP values; when sampling from all observed values, the distribution varies across timepoints.

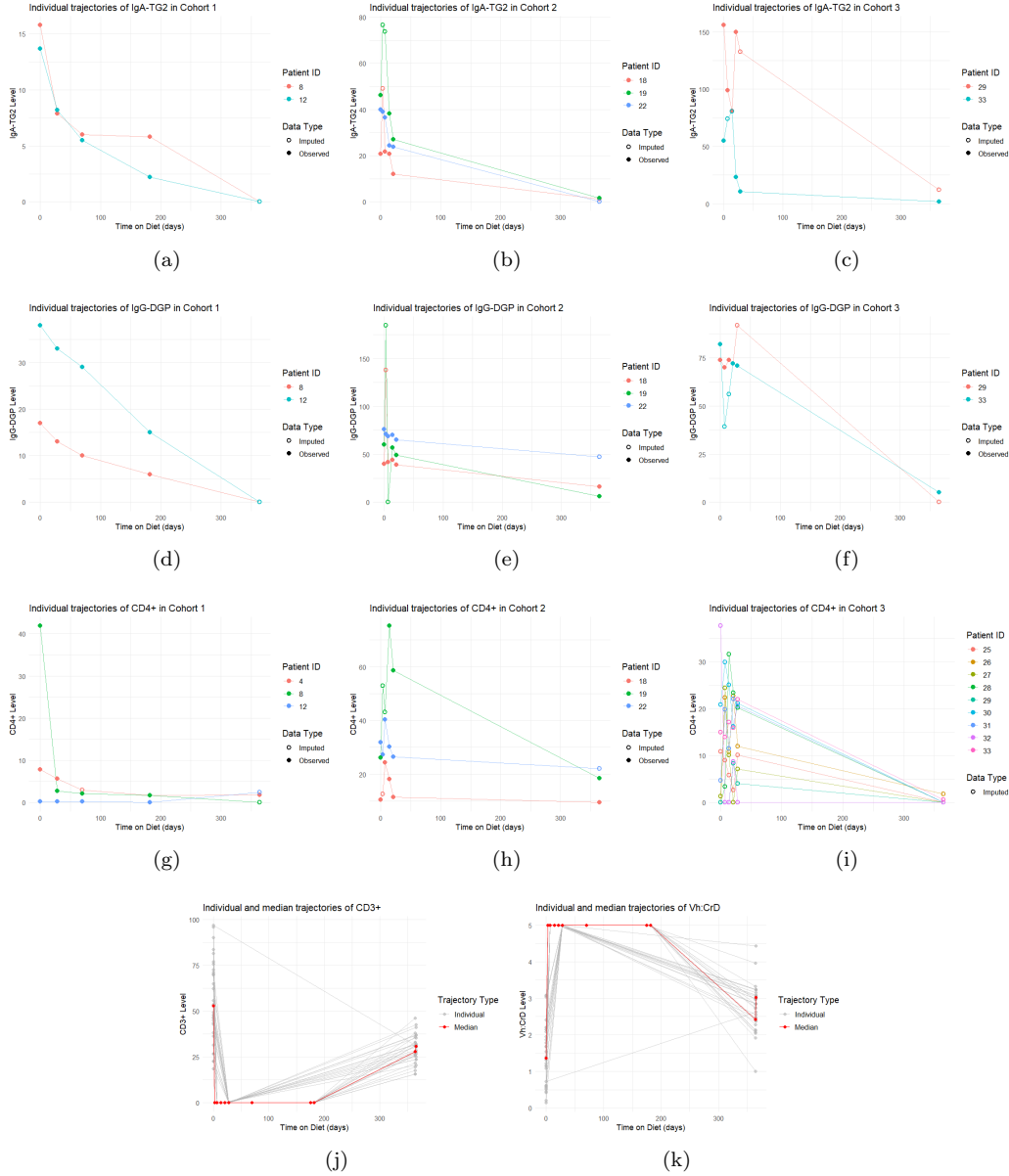


Figure 2.35: Imputed trajectories of values of (a-c) IgA-TG2, (d-f) IgG-DGP, (g-i) CD4+, (j) CD3+, and (k) Vh:CrD.

Jomo

We impute the missing level 1 and level 2 variables simultaneously using Jomo. The predictor matrix is the same as pictured in Figure 2.36.

	patient_id	sex	Age	time_on_ohc	cohort	iga_tg2	igg_dgp	cd4	cd3	vhcrd	baseline_iga_tg2	baseline_igg_dgp	baseline_cd4	baseline_cd3	baseline_vhcrd
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	11	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	13	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	14	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	15	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	16	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	17	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	18	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	19	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	20	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	21	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	22	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	23	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	24	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	25	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	26	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	27	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	28	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29	29	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	30	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	31	1	1	1	1	1	1	1	1	1	1	1	1	1	1
32	32	1	1	1	1	1	1	1	1	1	1	1	1	1	1
33	33	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	34	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	35	1	1	1	1	1	1	1	1	1	1	1	1	1	1
36	36	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	37	1	1	1	1	1	1	1	1	1	1	1	1	1	1
38	38	1	1	1	1	1	1	1	1	1	1	1	1	1	1
39	39	1	1	1	1	1	1	1	1	1	1	1	1	1	1
40	40	1	1	1	1	1	1	1	1	1	1	1	1	1	1
41	41	1	1	1	1	1	1	1	1	1	1	1	1	1	1
42	42	1	1	1	1	1	1	1	1	1	1	1	1	1	1
43	43	1	1	1	1	1	1	1	1	1	1	1	1	1	1
44	44	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	45	1	1	1	1	1	1	1	1	1	1	1	1	1	1
46	46	1	1	1	1	1	1	1	1	1	1	1	1	1	1
47	47	1	1	1	1	1	1	1	1	1	1	1	1	1	1
48	48	1	1	1	1	1	1	1	1	1	1	1	1	1	1
49	49	1	1	1	1	1	1	1	1	1	1	1	1	1	1
50	50	1	1	1	1	1	1	1	1	1	1	1	1	1	1
51	51	1	1	1	1	1	1	1	1	1	1	1	1	1	1
52	52	1	1	1	1	1	1	1	1	1	1	1	1	1	1
53	53	1	1	1	1	1	1	1	1	1	1	1	1	1	1
54	54	1	1	1	1	1	1	1	1	1	1	1	1	1	1
55	55	1	1	1	1	1	1	1	1	1	1	1	1	1	1
56	56	1	1	1	1	1	1	1	1	1	1	1	1	1	1
57	57	1	1	1	1	1	1	1	1	1	1	1	1	1	1
58	58	1	1	1	1	1	1	1	1	1	1	1	1	1	1
59	59	1	1	1	1	1	1	1	1	1	1	1	1	1	1
60	60	1	1	1	1	1	1	1	1	1	1	1	1	1	1
61	61	1	1	1	1	1	1	1	1	1	1	1	1	1	1
62	62	1	1	1	1	1	1	1	1	1	1	1	1	1	1
63	63	1	1	1	1	1	1	1	1	1	1	1	1	1	1
64	64	1	1	1	1	1	1	1	1	1	1	1	1	1	1
65	65	1	1	1	1	1	1	1	1	1	1	1	1	1	1
66	66	1	1	1	1	1	1	1	1	1	1	1	1	1	1
67	67	1	1	1	1	1	1	1	1	1	1	1	1	1	1
68	68	1	1	1	1	1	1	1	1	1	1	1	1	1	1
69	69	1	1	1	1	1	1	1	1	1	1	1	1	1	1
70	70	1	1	1	1	1	1	1	1	1	1	1	1	1	1
71	71	1	1	1	1	1	1	1	1	1	1	1	1	1	1
72	72	1	1	1	1	1	1	1	1	1	1	1	1	1	1
73	73	1	1	1	1	1	1	1	1	1	1	1	1	1	1
74	74	1	1	1	1	1	1	1	1	1	1	1	1	1	1
75	75	1	1	1	1	1	1	1	1	1	1	1	1	1	1
76	76	1	1	1	1	1	1	1	1	1	1	1	1	1	1
77	77	1	1	1	1	1	1	1	1	1	1	1	1	1	1
78	78	1	1	1	1	1	1	1	1	1	1	1	1	1	1
79	79	1	1	1	1	1	1	1	1	1	1	1	1	1	1
80	80	1	1	1	1	1	1	1	1	1	1	1	1	1	1
81	81	1	1	1	1	1	1	1	1	1	1	1	1	1	1
82	82	1	1	1	1	1	1	1	1	1	1	1	1	1	1
83	83	1	1	1	1	1	1	1	1	1	1	1	1	1	1
84	84	1	1	1	1	1	1	1	1	1	1	1	1	1	1
85	85	1	1	1	1	1	1	1	1	1	1	1	1	1	1
86	86	1	1	1	1	1	1	1	1	1	1	1	1	1	1
87	87	1	1	1	1	1	1	1	1	1	1	1	1	1	1
88	88	1	1	1	1	1	1	1	1	1	1	1	1	1	1
89	89	1	1	1	1	1	1	1	1	1	1	1	1	1	1
90	90	1	1	1	1	1	1	1	1	1	1	1	1	1	1
91	91	1	1	1	1	1	1	1	1	1	1	1	1	1	1
92	92	1	1	1	1	1	1	1	1	1	1	1	1	1	1
93	93	1	1	1	1	1	1	1	1	1	1	1	1	1	1
94	94	1	1	1	1	1	1	1	1	1	1	1	1	1	1
95	95	1	1	1	1	1	1	1	1	1	1	1	1	1	1
96	96	1	1	1	1	1	1	1	1	1	1	1	1	1	1
97	97	1	1	1	1	1	1	1	1	1	1	1	1	1	1
98	98	1	1	1	1	1	1	1	1	1	1	1	1	1	1
99	99	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100	100	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 2.36: Predictor matrix for simultaneous imputation of level 1 and level 2 variables using Jomo.

We construct five complete imputed datasets. The imputed values are plotted against the observed values in Figure 2.37. Additionally, the distributions of values are plotted in Figure 2.38. Notice that the distributions do not vary as much as in imputation by MICE. As previously stated, Jomo is generally recommended for imputations of level 2 variables over MICE, and can be used in conjunctions with MICE. Hence we will try this below.

Before imputation, we must transform our variables. Jomo assumes variables follow a multivariate normal distribution, and therefore if used on our raw data which has only non-negative values, will result in negative imputed values which are unrealistic. Some variables also have maximal values (e.g. CD4+ is measured out of 100 epithelial cells, and therefore has a maximum value of 100). The bounds and transformations are described in Table 2.8. Note that while all variables could take values at 0, only CD4+ does. Hence 1 is added only to CD4+ to ensure $\log(0)$ is not taken.

Variable	Bounds	Transformation	Back-transform
IgA-TG2	$[0, \infty]$	$\log(x)$	$\exp(y)$
IgG-DGP	$[0, \infty]$	$\log(x)$	$\exp(y)$
CD4+	$[0, 100]$	$\log\left(\frac{x+1}{1-\frac{x+1}{100}}\right)$	$100 \frac{\exp(y)}{1+\exp(y)} - 1$
CD3+	$[0, \infty]$	$\log(x)$	$\exp(y)$
Vh:CrD	$[0, 4.5]$	$\log\left(\frac{x}{1-\frac{x}{4.5}}\right)$	$4.5 \frac{\exp(y)}{1+\exp(y)}$

Table 2.8: Transforms and back-transforms of variables pre- and post-imputation by Jomo.

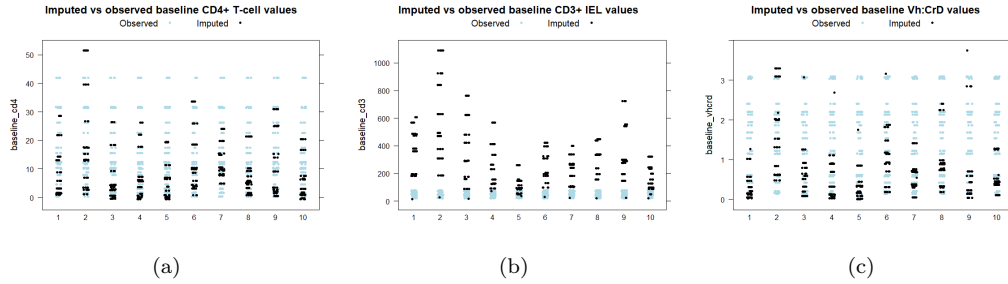


Figure 2.37: Imputed vs. observed values in ten imputed dataframes after Jomo imputation of baseline (a) CD4+ T-cells, (b) CD3+ IELs, and (c) Vh:CrD ratio values. Imputed values are black.

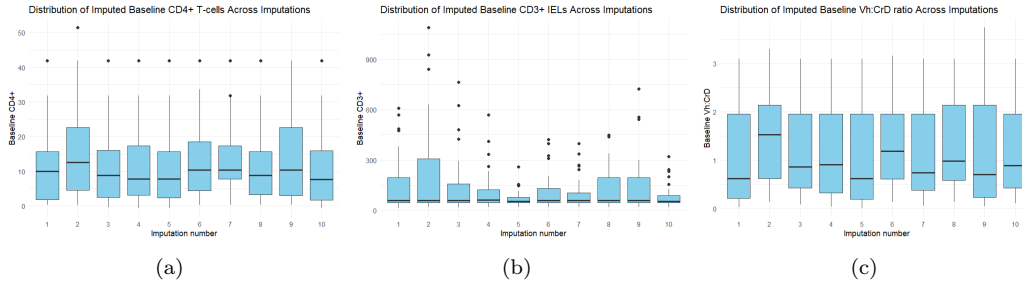


Figure 2.38: Distributions of baseline values of (a) CD4+ T-cells, (b) CD3+ IELs, and (c) Vh:CrD ratios after imputation by Jomo.

The imputations of the level 1 variables are plotted against observed values in Figure 2.39, and the distributions of the resulting values are plotted in Figure 2.40. Unfortunately, the imputation model for CD3+ And Vh:CrD appear to have not converged, and the resultant values of both variables are highly heterogeneous. Additionally, the ranges of imputed values of both variables are much greater than the range of the observed values. While we could limit the range of the imputations, the variation suggests issues with the imputation models themselves, and limiting the ranges may just mask the problems. However, the imputations of IgA-TG2 and IgG-DGP have much smaller ranges than the observed values, and the imputations for CD4+ have smaller ranges than those of the MICE imputations.

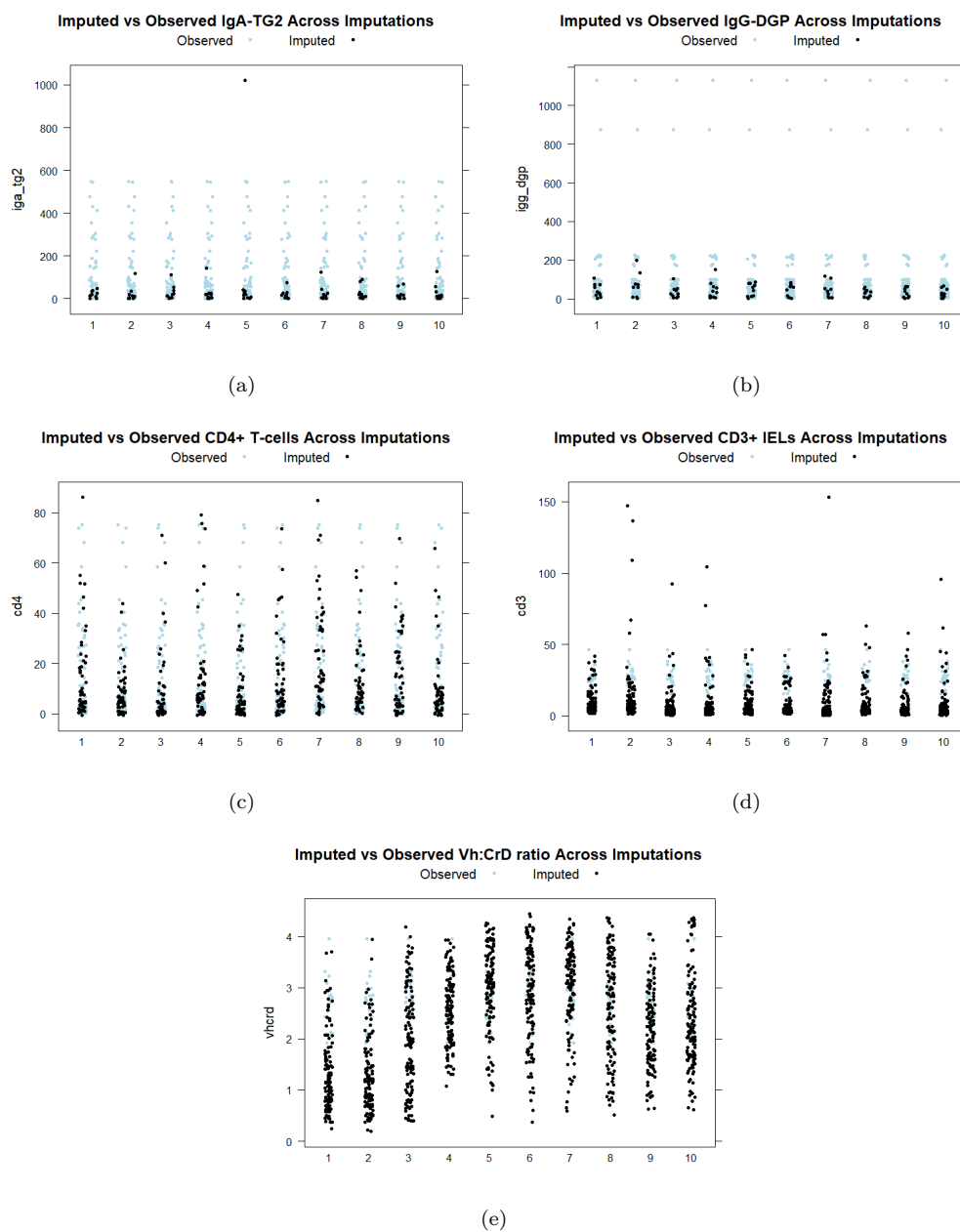


Figure 2.39: Imputed vs. observed values in ten imputed dataframes after MICE imputation of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios. Imputed values are black.

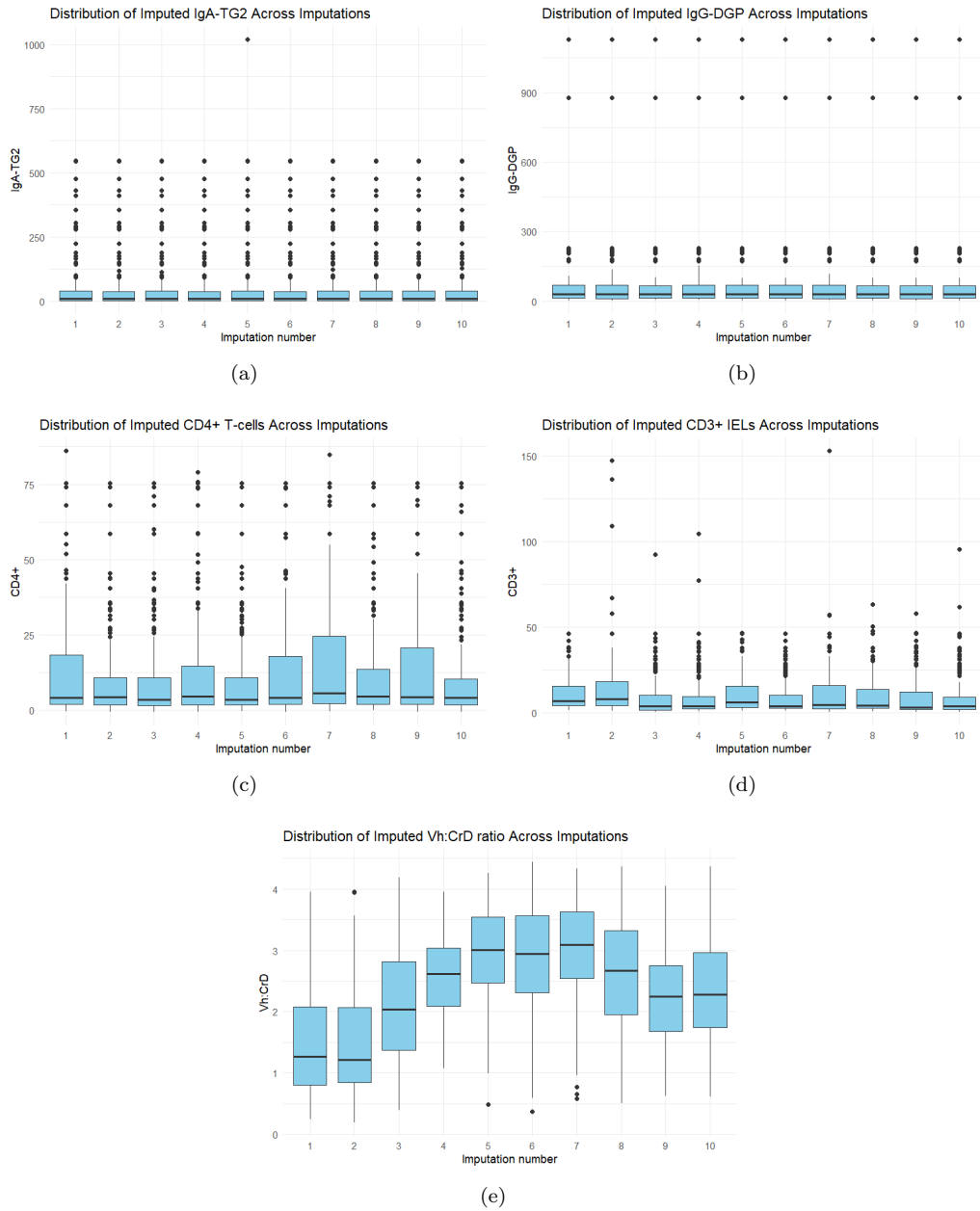


Figure 2.40: Distributions of values of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios after imputation by MICE.

We plot the trajectories of the measured variables after imputation in Figure 2.41 from the first imputed dataset. We can see that the imputed trajectories of IgA-TG2 and IgG-DGP are not very dissimilar from the imputed trajectories given by MICE. However, there is less range, and even when values are estimated to be large, they are not as great as those imputed by MICE. The consistent patterns in MICE and Jomo demonstrate the convergence of both models.

The main problem with imputing CD4+, CD3+, and Vh:CrD is the small sample size due to Cohort 3 not being measured at all, which cannot be remedied.

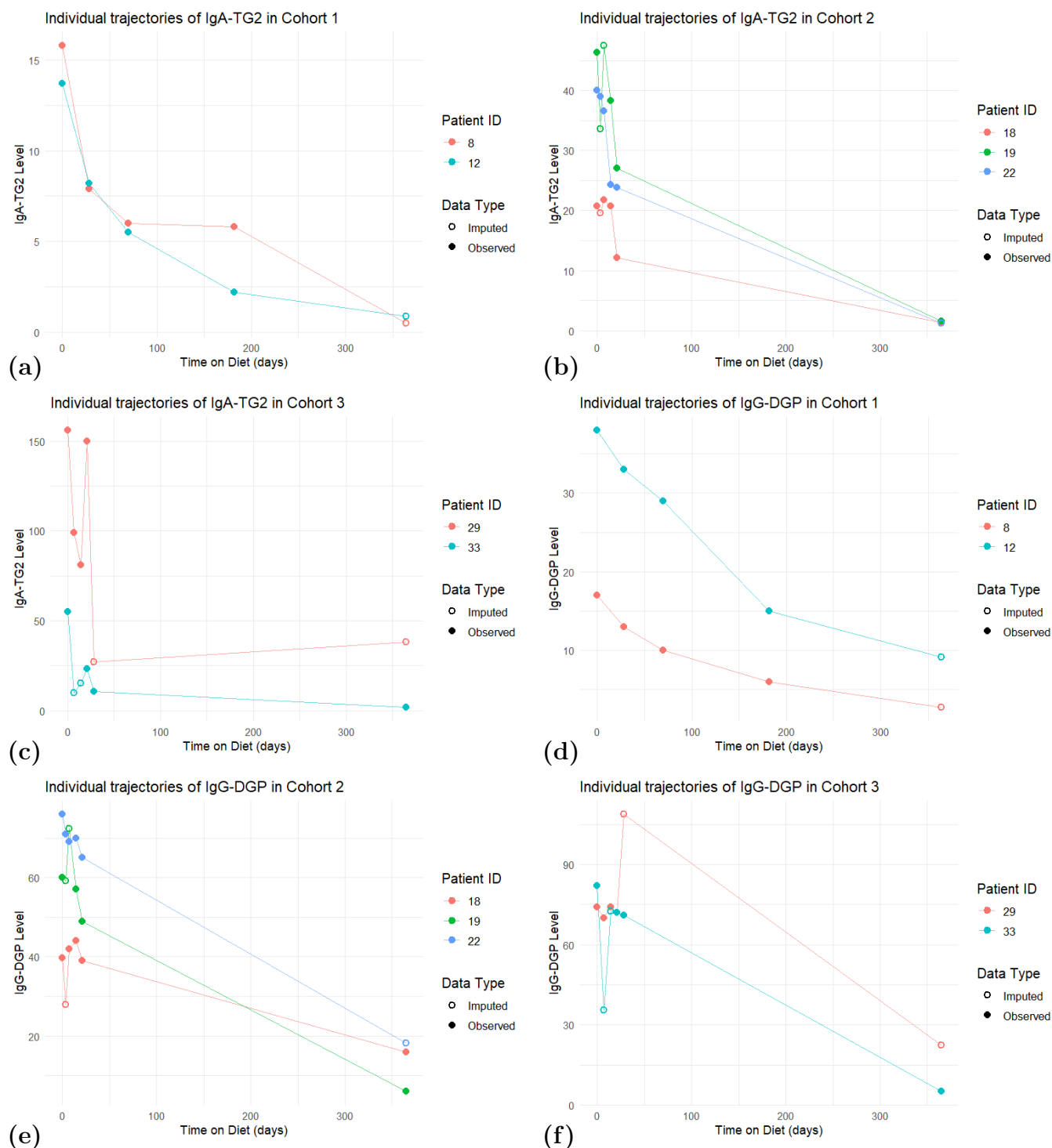


Figure 2.41: Trajectories of (a–c) IgA-TG2, (d–f) IgG-DGP, (g–i) CD4+ T-cells, (j–l) CD3+ IELs, and (m–o) Vh:CrD ratios over time after imputation by Jomo.

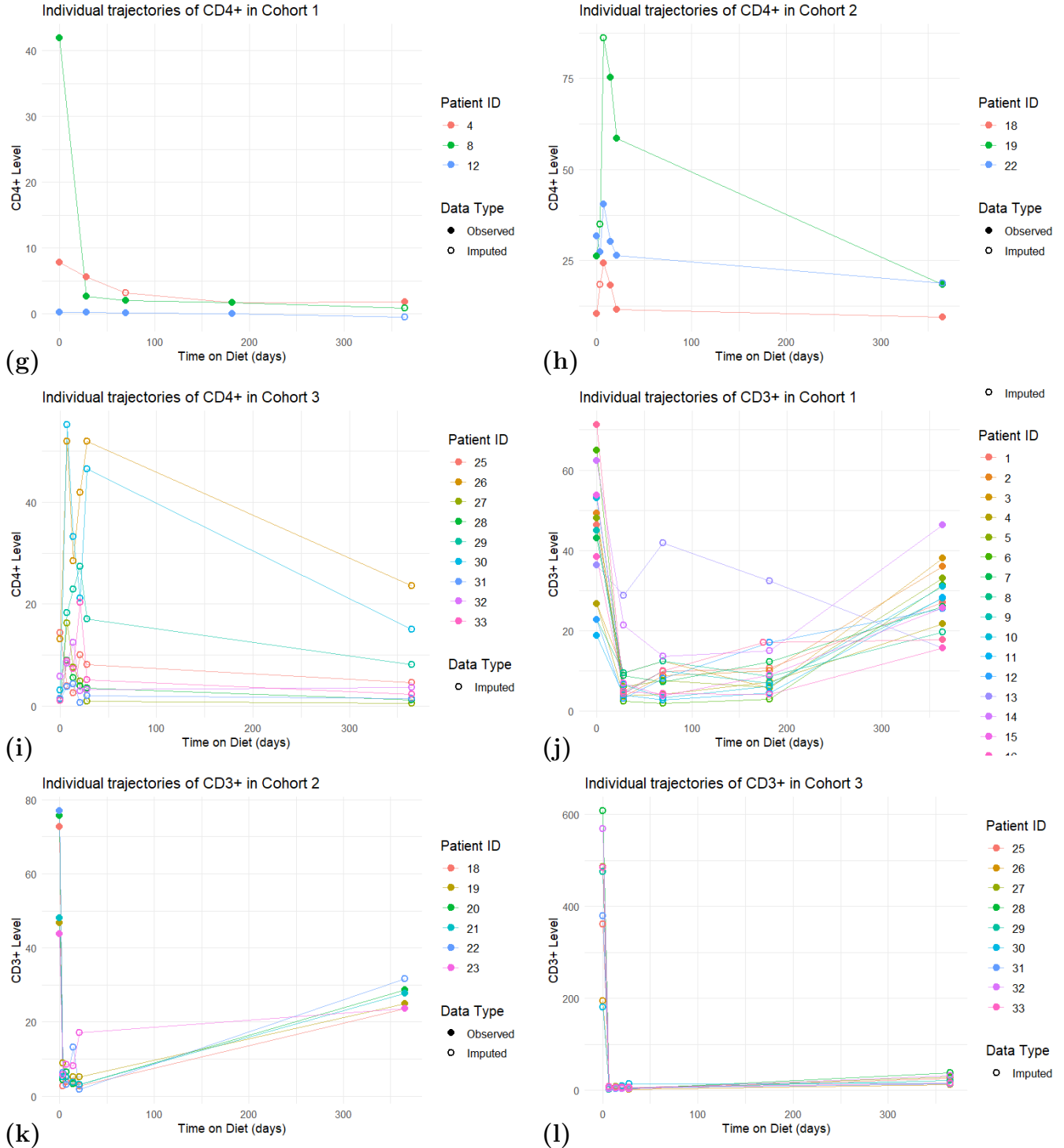


Figure 2.41: Trajectories of (a–c) IgA-TG2, (d–f) IgG-DGP, (g–i) CD4+ T-cells, (j–l) CD3+ IELs, and (m–o) Vh:CrD ratios over time after imputation by Jomo (continued).

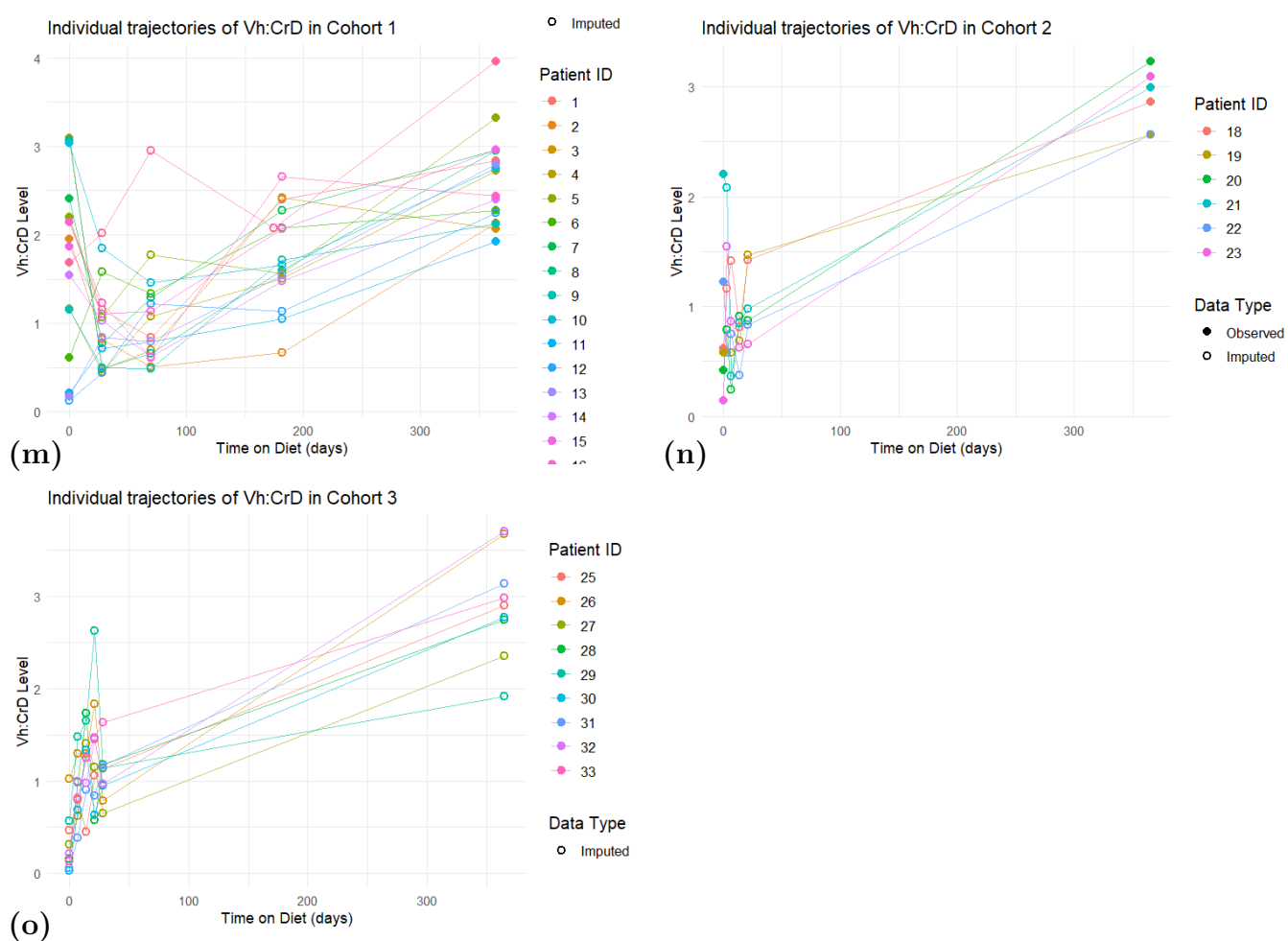


Figure 2.41: Trajectories of (a–c) IgA-TG2, (d–f) IgG-DGP, (g–i) CD4+ T-cells, (j–l) CD3+ IELs, and (m–o) Vh:CrD ratios over time after imputation by Jomo (continued).

Jomo and MICE

We use Jomo to impute the missing baseline values of CD4+, CD3+, and Vh:CrD. The results are shown in Figure 2.42. We see high variability in baseline CD4+ and CD3+ values, but the baseline imputations of Vh:CrD seem mostly reasonable. This is additionally demonstrated in the distribution plots of the imputation model (Figure 2.43), which show limited changes in distributions of Vh:CrD imputations. Therefore, Jomo may be appropriate for modeling Vh:CrD but may be worse at imputing CD3+ values.

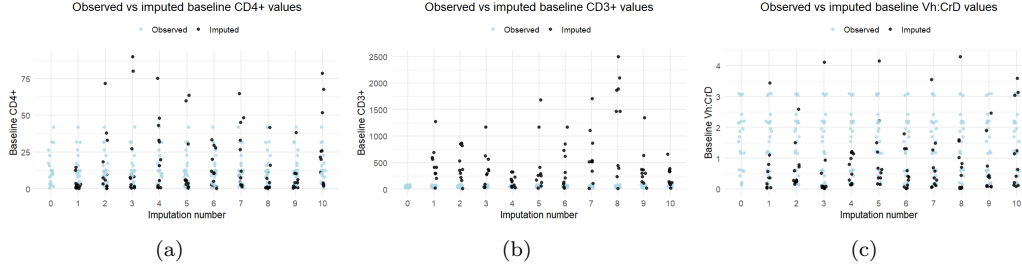


Figure 2.42: Imputed vs. observed values in ten imputed dataframes after Jomo imputation of baseline values of (a) CD4+, (b) CD3+, and (c) Vh:CrD. Imputed values are black.

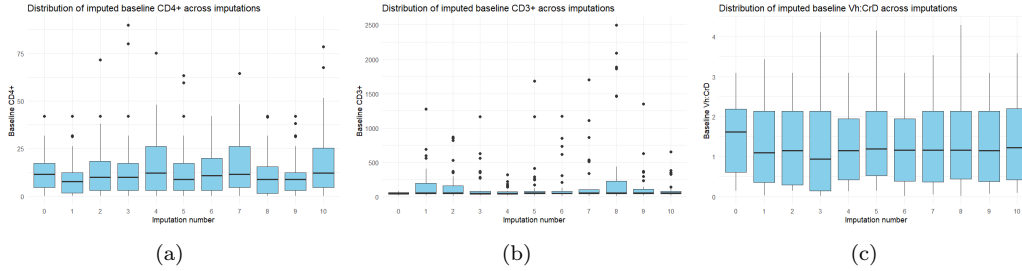


Figure 2.43: Distributions of baseline values of (a) CD4+, (b) CD3+, and (c) Vh:CrD after imputation by Jomo.

After imputation of baseline values, we use MICE to impute the level 1 variables IgA-TG2, IgG-DGP, CD4+, CD3+, and Vh:CrD. We use all ten imputed dataframes from Jomo as inputs for the model, and construct ten dataframes for each. We show the ten dataframes constructed for the first Jomo dataframe. The density plots of the imputation models (Figure 2.44) demonstrate good convergence, with little observable pattern in the model densities.

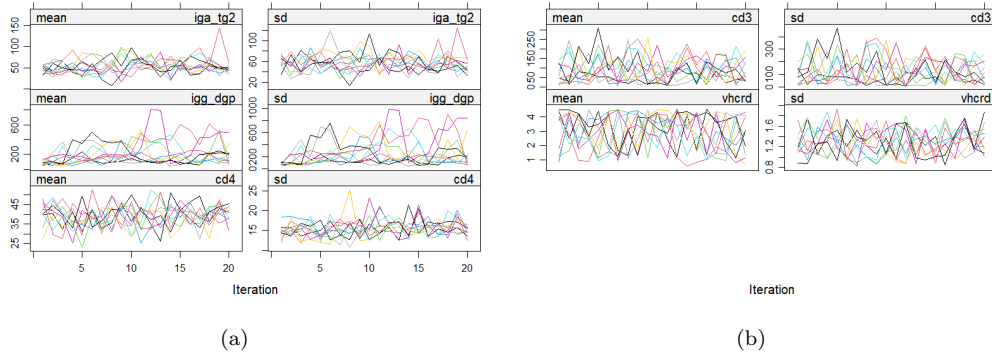


Figure 2.44: Density plots of models of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios after imputation by MICE.

The imputed values are visualized in Figure 2.45. We see high variation in all of the datasets, especially in CD3+ and Vh:CrD. The imputations of both IgA-TG2 IgG-DGP have lower ranges than expected. However, CD4+ imputations appear to be reasonable, if left-skewed.

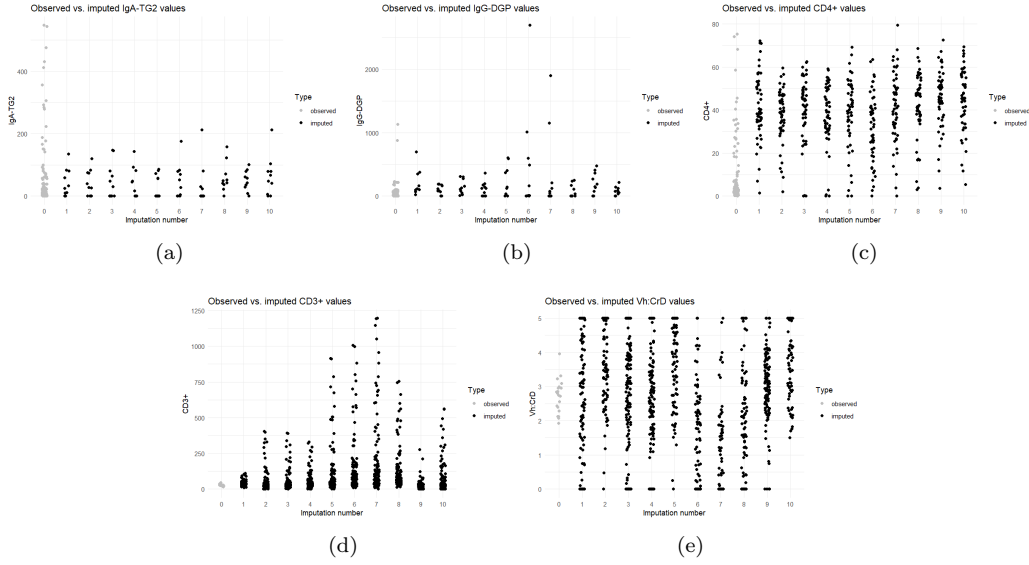


Figure 2.45: Imputed values of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios after imputation by Jomo.

The distributions of the imputed values are shown in Figure 2.46. We see that that distributions of IgA-TG2 and CD4+ do not reflect the distributions of the observed values. However, while the distributions of IgG-DGP are not exactly reflective of the original distribution, they do not deviate that dramatically.

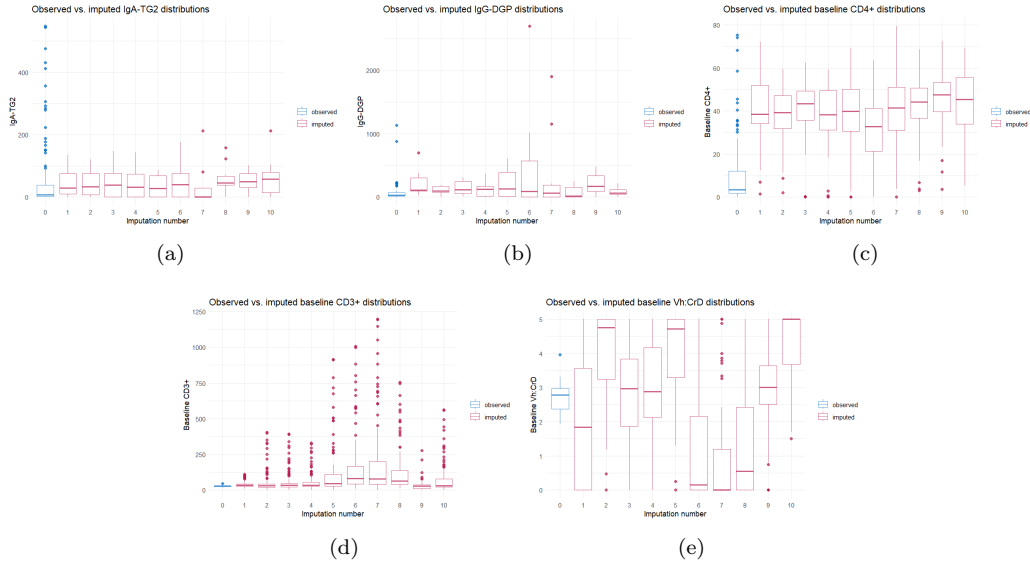


Figure 2.46: Distribution plots of models of (a) IgA-TG2, (b) IgG-DGP, (c) CD4+ T-cells, (d) CD3+ IELs, and (e) Vh:CrD ratios after imputation by Jomo.

Final imputation

We use what we've learned from the previous imputations to make our final model. First, imputations of IgA-TG2, IgG-DGP, and CD4+ look reasonable for the most part using both MICE and Jomo. However, Jomo results in extremely low variance between imputations, and the range of the imputed values is generally very low, which is unexpected. We expect some variance between imputations, and because the MICE model converges for these three variables it is not troubling that the imputed datasets have some variance. Hence we prefer MICE for imputation of these variables.

The MICE model seems to converge for all three baseline variables (CD4+, CD3+, and Vh:CrD). However, the variance of the imputed values of the baseline CD3+ and Vh:CrD is extremely high. Additionally, both the MICE and Jomo imputations of Vh:CrD and CD3+ are highly variable. This is not surprising considering we have very high levels of missing data and no data from Cohort 3. Additionally, because this missingness is due to study design, it is not necessarily true that the data is MAR and therefore may not meet the requirements of multiple imputation. It is therefore best not to impute these variables.

Our final imputation technique is therefore as follows:

1. Impute baseline CD4+ using `21only.pmm` from MICE.
2. Impute IgA-TG2, IgG-DGP, and CD4+ using `21.pan` from MICE.
3. Impute missing baseline Vh:CrD and CD3+ in Cohorts 1 and 2 using `pmm` from MICE, and impute missing Vh:CrD and CD3+ only at timepoints 364 or 365 days using `21.pan` from MICE.

The imputed trajectories are pictured in Figure 2.47. Notice that only the trajectories of the first imputed dataset are shown. Although we do not present the results of the other datasets, we will use all ten imputed datasets when applying any model to this data.

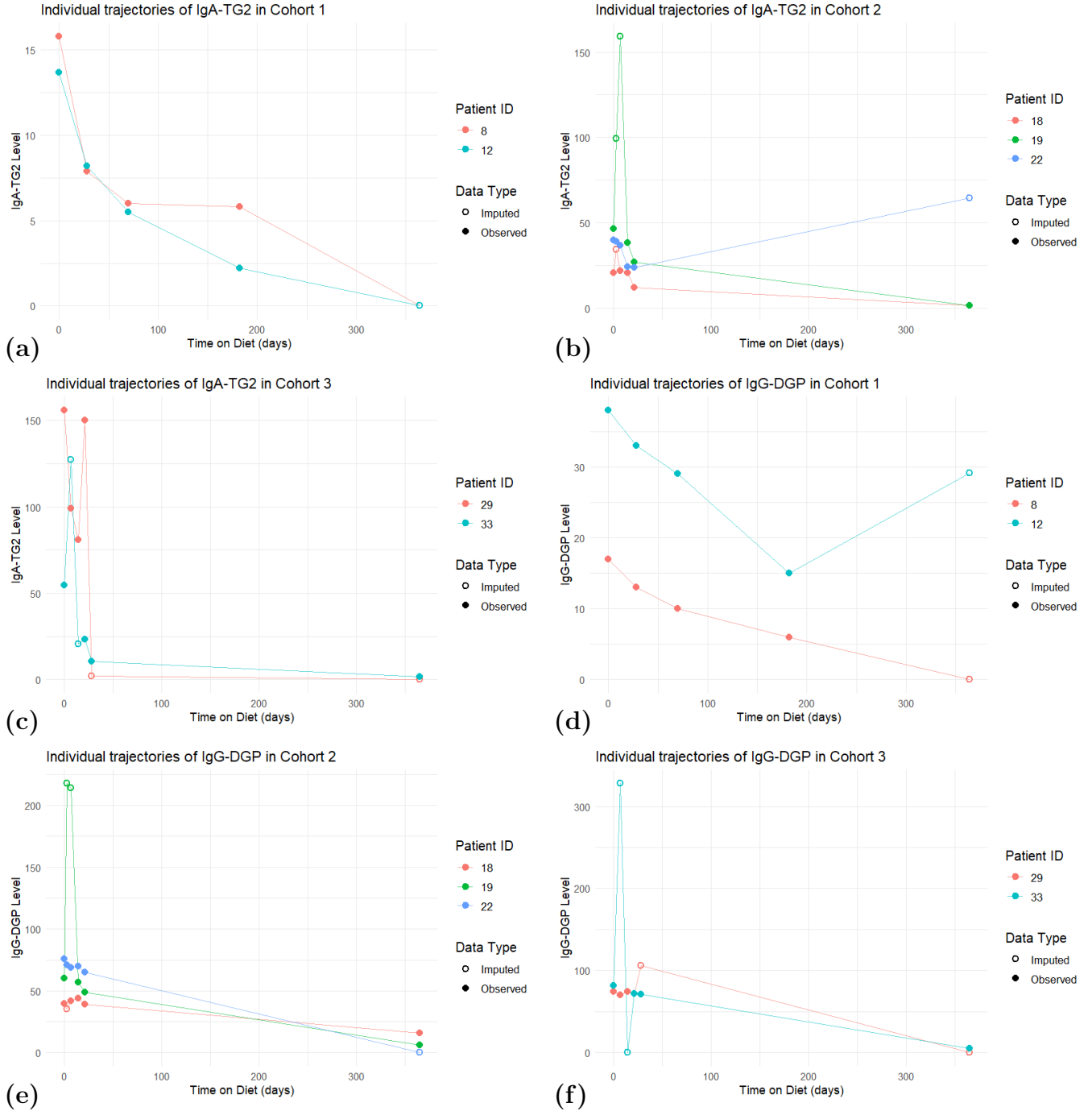


Figure 2.47: Trajectories of (a–c) IgA-TG2, (d–f) IgG-DGP, (g–i) CD4+ T-cells, (j, k) CD3+ IELs, and (l, m) Vh:CrD ratios over time after final imputation by MICE.

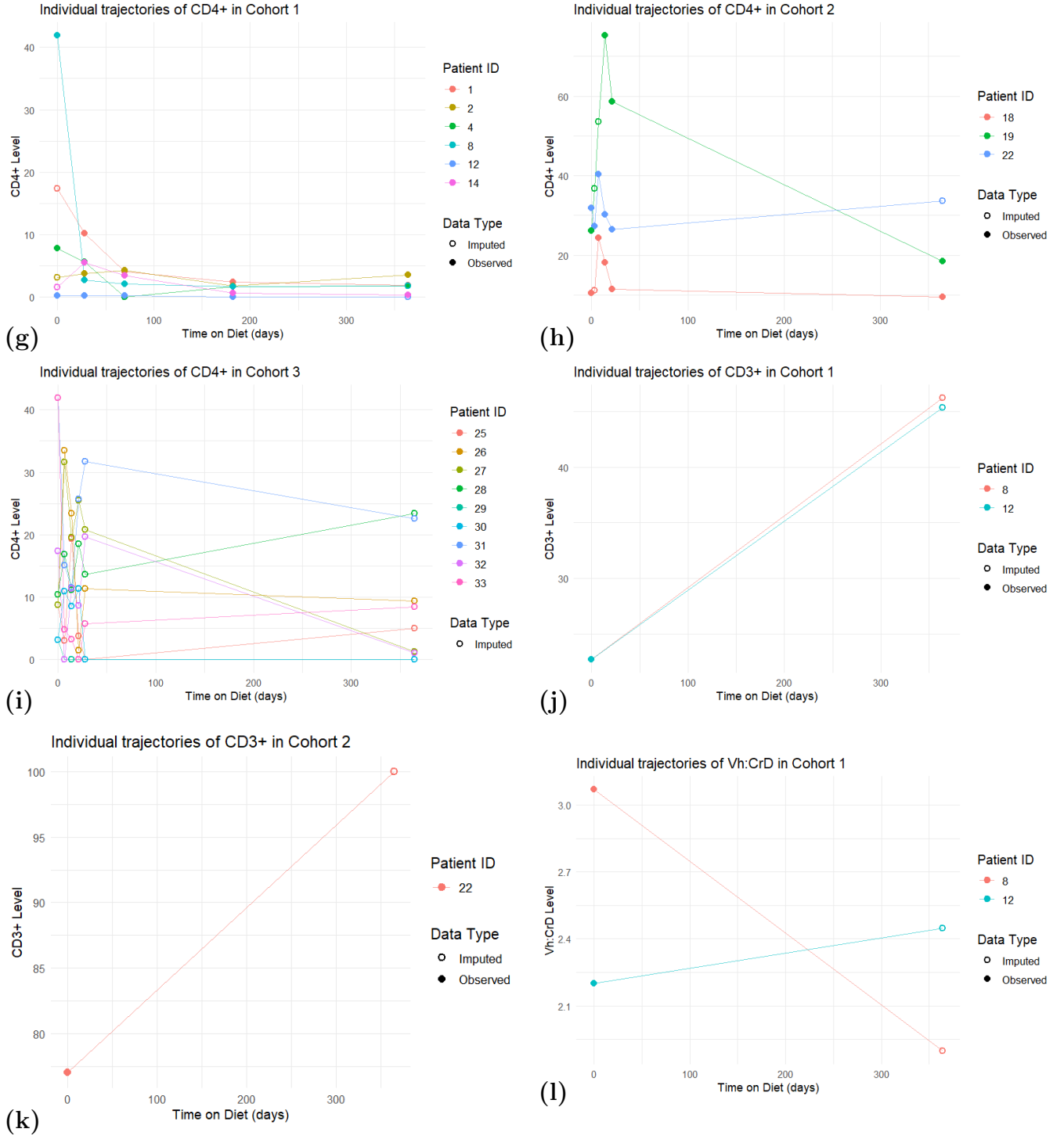


Figure 2.47: Trajectories of (a–c) IgA-TG2, (d–f) IgG-DGP, (g–i) CD4+ T-cells, (j, k) CD3+ IELs, and (l, m) Vh:CrD ratios over time after final imputation by MICE (continued).

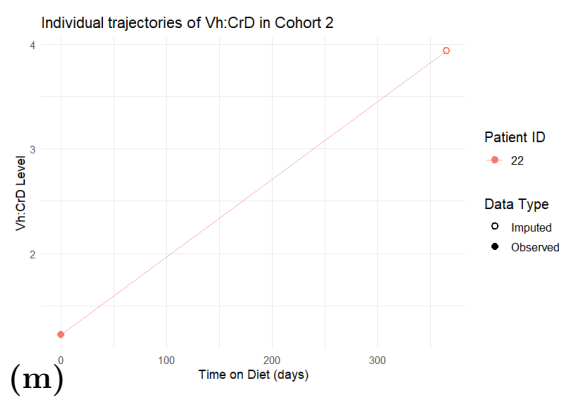


Figure 2.47: Trajectories of (a–c) IgA-TG2, (d–f) IgG-DGP, (g–i) CD4+ T-cells, (j, k) CD3+ IELs, and (l, m) Vh:CrD ratios over time after final imputation by MICE (continued).

Chapter 3

Data generation

Our largest dataset has thirty-three patients with four to five measurements each. Additionally, it is very common for measurement points to span a considerable range of time. Consequently, using these datasets for ODE discovery with PINNs will result in uninformative models. Instead, we will attempt to generate synthetic patients with measurements in many timepoints for use in our PINNs model.

We will use synthetic data generation techniques that utilize the patient datasets and create data representatives (Section 3.1). Additionally, we will develop an ODE model and generate an ODE representative of the Celiac immune response during gluten exposure and after adoption of a gluten free diet (Section 3.2).

3.1 Data representatives

There are a few well-known methods of generating synthetic patient data, including generative adversarial networks (GANs), recurrent neural networks (RNNs), and Bayesian networks. As a form of deep learning, GANs generally require large amounts of data, and are typically used to synthesize large medical databases in order to anonymize the data. However, GANs have been adapted to model and generate longitudinal patient data, typically using larger sample sizes than we have [43]. RNNs are deep neural networks used to model time-series data and longitudinal data, often in the form of an imputation model, and also require large amounts of observed data. In contrast, Bayesian networks allow for flexibility in sample sizes, but are typically used as an alternative to regression models for data analysis. However, because they can be used to estimate the distribution of the data, they may be used to generate new data by simply sampling from the estimated distribution.

Although Bayesian networks are meant to be flexible, applying them to complicated time dependent systems quickly becomes extremely tedious. A major problem with Bayesian networks is the need to program each node and link manually. This becomes difficult when nodes must be generated for each timestep, and each dataset has different timesteps and different variables. When using three datasets, three separate models must be created, quickly resulting in thousands of commands simply describing the relationships between variables. There are libraries which make this easier, such as Bayes Server and pyAgrum, but these still require manual inputs which become very tedious, and do not seem to be viable options when datasets are large and/or time dependent.

Because we have pre-existing longitudinal datasets, it is possible to simply expand that dataset to include the desired number of timepoints and impute the missing data. Both GANs and RNNs can be adopted to do this, but GANs require far larger amounts of data in general, and likely would not work for our problem at all. We will therefore use Recurrent Neural Networks in order to impute an expanded dataset. This is because it is applicable to large amounts of data, and has a chance of working on our small datasets. In order to make a model to generate data for one synthetic patient unifying the data from all three sets, we will make a list of all desired variables. We will make two models: one to model patients who are on a gluten free diet who consume gluten, and one for patients who begin a gluten free diet. Hence, we will model Datasets A and C together, and model B separately. This way, we can also train a model on each imputed version of Dataset B, and pool the results. This will mean that the model of Dataset B will not

include CD3-, IL-2, or CD8+. However, it will still include information about IELs (in the form of CD3+). Hence, although not ideal, the model will still describe the variables we are most interested in while creating a dataset that PINNs can be used on.

3.1.1 Generation model

Instead of using a Generative Adversarial Network or a Bayesian Network, we use Recurrent Neural Networks (RNNs) in order to interpolate our missing values. While RNNs are typically not the best method for handling extremely small amounts of data, they are the most likely method to perform decently.

Multiple imputation methods such as MICE and Jomo have been shown to work on both large and small datasets, and also on relatively sparse datasets, and have been shown to out-perform other imputation methods due to their ability to deal with irregular sampling time between variables. A method called Multi-directional Recurrent Neural Networks (MRNN) utilizes the advantages of both multiple imputation and RNN methods, leading to even further improved imputations [44]. We adapt the methods used in [44, 45] to develop our model, which we explain here.

We assume we have N patients and P measured variables (which may be categorical or continuous measurements), each measured at T timepoints. We define Y to be the $(N \cdot T) \times P$ matrix of all data, with missing entries labelled (-). We define X to be the $P \times N \times T$ array of patient data. We take M to be the $P \times N \times T$ matrix which has entries 0 if the entry in X is missing (-), and 1 if it is observed (equivalent to the matrix O defined in Section 2.4.1). We additionally define Δ to be the $P \times N \times T$ matrix which describes how much time has elapsed since the previous measurement for each variable in the time-series. For instance, consider a matrix for Patient n :

$$X_n = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 22 & 22 & 22 & 22 \\ 0.3 & 0.2 & 0 & - \\ - & - & - & - \end{pmatrix}.$$

Then we define M_n to be:

$$M_n = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Assuming the first two rows of X_n describe categorical variables (e.g. sex or age) which are constant, we define Δ_n to be:

$$\Delta_n = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \delta_1^2 & \delta_2^2 & \delta_3^2 \\ 0 & \delta_1^3 & \delta_2^3 & \delta_3^3 \end{pmatrix},$$

where $\delta_t^p \geq 0$ is the time that has elapsed since variable p was measured at timestep $t - 1$.

The objective is to find a function f which replaces missing values while minimizing the estimation loss. Hence, the problem is:

$$\min_f \mathbb{E} \left[\sum_{t=1}^T \sum_{p=1}^P (1 - m_t^p) \mathcal{L}(f_t^p(Y, M, \Delta), x_t^p) \right],$$

where $\mathcal{L}(f_t^p(Y, M, \Delta), x_t^p) = (f_t^p(Y, M, \Delta) - x_t^p)^2$. Because we do not know the exact values of missing data, we estimate the loss empirically. We evaluate the model by computing the loss

between an observed measurement x_t^p and its estimate $\hat{x}_t^p$ formed using the data without x_t^p . The total loss for the dataset using mean squared error is given by:

$$\mathcal{L}(\{\hat{x}_t^p, x_t^p\}) = \sum_{n=1}^N \left[\frac{\sum_{t=1}^T \sum_{p=1}^P m_t^p(n) \times (\hat{x}_t^p - x_t^p)^2}{\sum_{t=1}^T \sum_{p=1}^P m_t^p(n)} \right].$$

The model uses an interpolation block and an imputation block in order to produce estimates depending on between- and within-variable information. We describe both below.

Interpolation model

The interpolation model constructs a function Φ which estimates the point x_t^p given x_{t-1}^p and x_{t+1}^p . Hence, the value of x_t^p is not used in the construction of its estimate $\tilde{x}_t^p$. The interpolation function is constructed using a bi-directional RNN with activation functions g and h (e.g. ReLU or tanh), weight matrices $W_f, U_f, V_f, W_b, U_b, V_b$, and biases c_f, c_b . If $z_t^p = [x_t^p, m_t^p, \delta_t^p]$, then the estimate $\tilde{x}_t^p$ is given by:

$$\begin{aligned} f_t^p &= g(W_f^p f_{t-1}^p + V_f^p z_{t-1}^p + c_f^p) \\ b_t^p &= g(W_b^p b_{t+1}^p + V_b^p z_{t+1}^p + c_b^p) \\ \tilde{x}_t^p &= h(U_f^p f_t^p + U_b^p b_t^p + c_0^p). \end{aligned}$$

The model is implemented using PyTorch. The weight matrices are initialized as block diagonal matrices. The weights $W_f, W_b \in \mathbb{R}^{P \times d \times d}$, $V_f, V_b \in \mathbb{R}^{P \times d \times 3}$, and $U \in \mathbb{R}^{P \times d \times 2}$, where d is the specified number of hidden dimensions in the neural network, are initialized as block-diagonal matrices with values drawn from a Xavier Uniform or a He Uniform distribution (described below). The biases $c_f, c_b \in \mathbb{R}^{P \times d}$ and $c_0 \in \mathbb{R}^P$ are also initialized as block-diagonal matrices with Xavier or He Uniform values.

The initialization using a Xavier Uniform distribution (also called Glorot initialization) is standard when using symmetric activation functions with a gradient at origin of 1 ($f'(0) = 1$) such as tanh so as to keep variances of the inputs and outputs the same between layers during forward propagation, and to keep the variance of the gradients the same during backpropagation [46]. Of course, the ReLU activation function is not symmetric, nor does it have unit gradient at the origin, and in fact a different initialization (He or Kaiming initialization) is recommended when using ReLU. He initialization is almost identical to Glorot initialization, except that the variance of the weights is multiplied by 2. When using extremely deep neural networks (e.g. 30+ layer models), using Glorot initialization and ReLU activation function may result in a non-converging model, while He initialization results in a converging model. However, for shallower networks, both result in converging models; Glorot initialization may just take longer [47].

Imputation model

The imputation model constructs a function Ψ which estimates the point x_t^p with information only from the measurements of other features at time t . The imputation function is constructed using a neural network with fully connected layers independent of t , with activation functions σ, ϕ , weight matrices W, U, V , and bias vectors α, β . If $z_t = [\tilde{x}_t, m_t]$, then the estimate $\hat{x}_t^p$ is given by:

$$\begin{aligned} h_t &= \phi(Ux_t + Vz_t + \beta) \\ \hat{x}_t &= \sigma(Wh_t + \alpha). \end{aligned}$$

Implementation of the imputation model is done using PyTorch. The weights $W \in \mathbb{R}^{P \times d}$, $V \in \mathbb{R}^{d \times P}$, and $U \in \mathbb{R}^{d \times P}$ are initialized with Xavier or He Uniform values, and the biases $\beta \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}^P$ are initialized from a Xavier or He Uniform distribution.

M-RNN model

The final model is implemented in PyTorch and works as follows:

1. First, X is interpolated using simple linear interpolation so that no missing values exist in the dataset.
2. Next, X is interpolated using the interpolation model, with inputs X , O , and Δ . The estimate $\hat{X}$ is output.
3. Finally, X is imputed using the imputation model, with inputs X , $\hat{X}$, and O .
4. Loss is computed using only observed measurements.

The interpolation and imputation functions Φ and Ψ are learned jointly by stacking the Bi-directional RNN and fully-connected layers, resulting in:

$$\Phi^*, \Psi^* = \arg \min_{\Phi, \Psi} \mathcal{L} \left(\Psi \left(\Phi \left(\{x_t^p, m_t^p, \delta_t^p\}_{t=1}^T \right), m_t^p \right)_{p=1}^P, x_t^p \right).$$

The weights for both models are updated using Adam from PyTorch.

3.1.2 Generation results

Direct application to our dataset of 365 timepoints per patient results in mostly constant time-series data. This makes sense, as we have extremely sparse observed data, resulting in most estimated values to sample constant values around the value that should be estimated. In other words, interpolating x_t^p results in using information from x_{t-1}^p , which is usually the previous value sampled (the baseline value) and x_{t+1}^p , which is usually 0 after the first linear interpolation step, and imputation uses information from x_t not including x_t^p which is also usually constant (or 0), resulting in an estimation of $\hat{x}_t^p = \hat{x}_{t-1}^p$ except around observed measurements (usually near $t = 1$ or $t = 365$). We can only remedy this by making up more artificial data using a different model, so for the purpose of this project we do not continue with this method of generation.

Imputation could also be done using simpler methods such as MICE. Unfortunately, the resulting datapoints are highly variable even when sampling only a few extra timepoints. We attempt to impute just 7-8 missing points of IgA-TG2 in Dataset B, which has the highest number of observed points (around 4-5 per patient). However, even this results in high variation and points which do not make much sense biologically. The trajectories of randomly some sampled patients (1, 14, 27, 15, and 30) are shown in Figure 3.1.

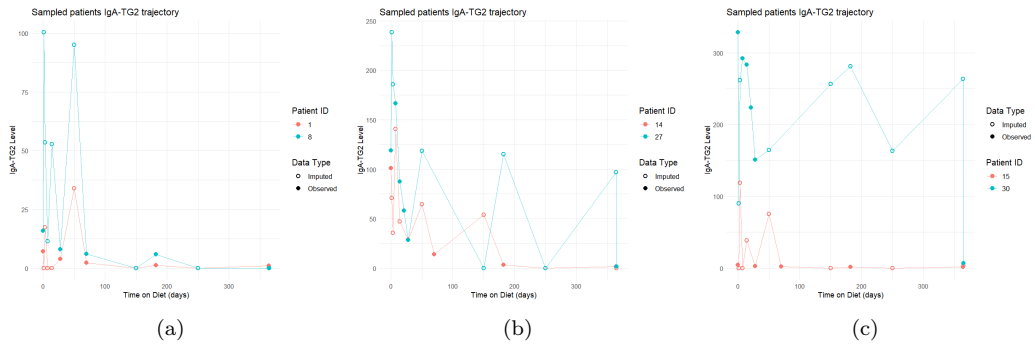


Figure 3.1: Trajectories of IgA-TG2 in patients (a) 1, (b) 14, 27, and (c) 15, 30.

Because generation of synthetic data does not seem to work with our very limited amount of data, we will continue with only generation of data using ODEs, and test the PINNs model using the ODE representatives. This will hopefully give us some idea of how well a PINNs model may work with more data.

3.2 ODE representatives

We develop ODEs in order to test our PINNs model. We model the concentrations of IELs, T-Cells, IgA-TG2, IL-15, IL-21, and Villi:Crypt ratio in Celiac Disease patients on and off a gluten free diet. Our aim in the construction of these ODEs is to estimate the minimal amount of data a study should provide in order to make meaningful statements about the dynamics of Celiac Disease using our PINNs model. We aim to estimate what variables are necessary to model the result of ingesting gluten in patients, preferably through Villi damage, which we model using Vh:CrD ratio. We do not develop a spatial model of Celiac Disease because we do not think this is necessary for estimating the effects of gluten. Instead, we focus on the variables which are typically measured in Celiac Disease trials, with the addition of the cytokines IL-15 and IL-21/IFN- γ .

3.2.1 Assumptions

Note that these ODEs are naive in that they do not encompass the entire immune response, and make a variety of assumptions on the important variables in the immune response. For instance, we model villi damage as a function of IELs, T-cells, cytokines and antibodies, although antibodies may not have a direct role in damaging the intestines, and there may be other variables which do. Additionally, we consider TCR- $\gamma\delta$ + IELs and CD4+ T-cells separately. However, in reality the classification of IELs and T-cells is not as clear as this; CD4+ and TCR- $\gamma\delta$ + are markers on T-cells and may be present on the same cell.

The average cell-shedding rate of villi cells (in one villi) is 1.3 per hour. The cells in the villi are produced by the crypt, which shed their cells at a comparable rate. Hence, the villi height:crypt depth ratio in control patients is maintained [48, 49]. However, in patients with active Celiac Disease, the crypt over-produces cells, while the villi is damaged and loses cells, which means that the Vh:CrD ratio decreases (since the crypt will grow larger). Additionally, the cell life of produced cells is reduced [50]. In treated Celiac Disease patients the crypt production rate changes throughout their recovery; initially, the crypt production rate is similar to that in untreated patients. Gradually, the production rate returns to that of control patients.

In general, there is not a maximal value of Vh:CrD ratio. However, in practice the crypt and villi reach a homeostasis, with an approximate maximal value of 4.5. Hence, we model Vh:CrD growth as logistic, with a maximal capacity 4.5.

The immune response to gluten is assumed to work as follows:

1. The digestion of gluten triggers the production of activated CD4+ T-cells.
2. Activated CD4+ T-cells produce cytokines IL-21 and IFN- γ .
3. The innate immune response leads to the production of IL-15 by intestinal cells.
4. Activated CD4+ T-cells trigger the production of anti-gliadin and anti-TG2 antibodies IgA/IgG-DGP and IgA/IgG-TG2. The presence of IL-15 results in increased production of antibodies.
5. The cytokines IL-15, IL-21 and IFN- γ bind to and activate IELs.
6. Activated IELs produce more IL-21 and IFN- γ .

We use the concentration of gluten to emulate the effects of gluten in the body, including the activation of APCs and the deamidation of gluten by TG2. This is because we do not have measurements for TG2, as TG2 is not measured in papers on Celiac Disease, and estimating the concentration of APCs is very difficult given the large number of different types of APCs. Hence, gluten is a fixed parameter we estimate in order to create a realistic-looking model. The value of gluten does not necessarily represent precisely how much gluten is being ingested.

3.2.2 ODE model

We construct ODEs based on growth-decay models with saturation rates, with dependencies based on the information from Section 1.1 and the assumptions outlined in Section 3.2.1. The ODEs constructed are based on the model developed in [51], and are consistent with the models of general autoimmune disorders described in [17, 16].

We describe the following variables:

$$\begin{aligned} I &= \text{CD3}^+ \text{ IELs, [1/100 cells]} & T &= \text{activated CD4}^+, [1/100 \text{ cells}], & A &= \text{IgA-/IgG-TG2}, \\ C_1 &= \text{IL-15}, & C_2 &= \text{IL-21/IFN-}\gamma, \\ V &= \text{Vh:CrD}, & G &= \text{Gluten effects.} \end{aligned}$$

The total number of cytokines is given by

$$C = C_1 + C_2$$

The ODEs describing the reaction in active Celiac Disease patients are given by:

T-Cells:

$$\frac{dT}{dt} = p_T T \left(1 - \frac{T}{a_T}\right) + d_T G - k_T T,$$

IELs:

$$\frac{dI}{dt} = p_I T \left(1 - \frac{I}{a_I}\right) + s_I C_2 - k_I I,$$

Antibodies:

$$\frac{dA}{dt} = p_A T + k_A C_1 + d_A G - h_A A,$$

Cytokines:

$$\begin{aligned} \frac{dC_1}{dt} &= (p_{C_1} - h_{C_1}) C_1 + d_{C_1} G \\ \frac{dC_2}{dt} &= (p_{C_2} - h_{C_2}) C_2 + b_{C_2} I + c_{C_2} T, \end{aligned}$$

Intestinal damage:

$$\frac{dV}{dt} = p_V V \left(1 - \frac{V}{a_V}\right) - k_V V \left(\frac{(g_V(I+T) + j_V C + l_V A)^2}{(g_V(I+T) + j_V C + l_V A)^2 + 3^2} \right).$$

We do not want to model G as dependent on t in the active Celiac model, because it is very difficult to predict the effects of time on G when we only sample once daily, and gluten would likely spike three times daily with meals. Instead, we simply set $G = 15$.

We model the ODEs of the reaction in recovering Celiac Disease patients by a decrease over time in all reaction concentrations. We assume the production rate of all concentrations to decrease over time, and the death or degradation rates to stay the same. These changes are reflected in the parameters. We only modify the ODE for intestinal damage:

$$\frac{dV}{dt} = (p_V - v_1 \exp(-v_2 t)) V \left(1 - \frac{V}{a_V}\right) - k_V V \left(\frac{(g_V(I+T) + j_V C + l_V A)^2}{(g_V(I+T) + j_V C + l_V A)^2 + 3^2} \right).$$

We model the intestinal damage as initially similar to that in active Celiac Disease patients who are still consuming gluten, with a gradual return to regular villi and crypt production as seen in healthy individuals.

We assume the effects from gluten decrease exponentially in time. We model

$$G \equiv G(t) = g_1 \exp(-g_2 t),$$

with $g_1 = 15$ and $g_2 = 0.3$.

3.2.3 Parameter estimates

We will assume the initial values of patients with active Celiac Disease to be given by averages of baseline values in treated Celiac Disease patients before gluten challenges. This means that in many cases, the patients are not comparable to control patients, but instead have existing intestinal damage and memory T-cells from previous gluten exposure. The concentration ranges are given in Table 3.1.

Variable	Control (no CD)	Active CD	Treated CD	Units
I	0-25 [10, 52]	14-32 [10, 52, 20]	15.7-36.2 [52, 20]	[1] (%)
T	0	4.60-22.60 [20]	1.500-6.400 [20]	[1] (%)
A	0.5-1.1 [53]	12.0-91.0 [53]	0-3.9 [20]	[U/ml]
C_1	0-150 [54, 55]	0-589.71 [55]	Unknown	[ng/ml]
C_2	0-175 [54, 55]	0-900 [54, 55]	0-175 [54]	[ng/ml]
V	2.6-4 [56, 57]	0-3 [9, 20, 57, 58]	2-4 [20, 56]	[1]

Table 3.1: Ranges of reasonable concentrations of variables.

In general, cytokine levels in treated Celiac Disease patients are similar to those of healthy individuals without Celiac Disease [54]. In order to model I , T , A_1 , and A_2 , and V in untreated and treated Celiac Disease patients we use the data from Dataset B [20].

The parameter estimates are given in Table 3.2. Note that some are estimated using a least-squares regression model, while some are gathered from studies.

The parameter p_T is estimated from [59] who found that on average, CD4+ T-cells divided 2-3 times over the course of 4 days in both active Celiac Disease patients and controls. We use this to estimate the proliferation rate of T-cells: we model the proliferation of T-cells to be exponential, with

$$\hat{T}(t) = T_0 e^{p_T t},$$

so

$$p_T = \frac{\ln\left(\frac{T(t)}{T_0}\right)}{t}.$$

Assuming that the T-cells divide 2-3 times in 4 days, then we have that $\frac{\hat{T}(t)}{T_0}$ is between $2^2 = 4$ and $2^3 = 8$, and

$$\begin{aligned} p_t &\in \left[\frac{\ln(4)}{4}, \frac{\ln(8)}{4} \right] \\ \implies p_t &\in [0.347, 0.520]. \end{aligned}$$

In order to estimate h_{A_1} , h_{A_2} , h_{C_1} , and h_{C_2} , we use their half-lives. We use the half-life formulas

$$N(t) = N_0 \left(\frac{1}{2} \right)^{\frac{t}{\tau}}, \quad N(t) = N_0 e^{-ht}$$

where τ is the half-life, N_0 is the initial concentration, t is the time, and h is the decay rate. We have that $\tau_A = 7$ days [60], $\tau_{C_1} = 2.5$ hours [61], and $\tau_{C_2} = 2$ hours [62]. Since $h = \frac{\ln(2)}{\tau}$, we get

$$\begin{aligned} h_{A_1} &= 0.0990, \\ h_{C_1} &= 6.654, \\ h_{C_2} &= 8.318. \end{aligned}$$

Parameter	Untreated CD	Recovering CD	Units
p_T [59]	0.433	0.433	[1/day]
a_T	40	40	[1]
<i>d_T</i>	0.907920357	7.14527834	[1/ng*h]
k_T [63]	.166	.061	[1/day]
<i>p_I</i>	0.00422992617	0.00201647536	[1/h]
a_I	100	100	[1]
<i>s_I</i>	3.554846×10^{-10}	0.00728824990	[ml/(ng*day)]
k_I [64]	0.002	.145	[1/day]
<i>p_A</i>	0.0274437341	$2.20428382 \times 10^{-4}$	[U/(ml*day)]
h_A [60]	0.0990	0.0990	[1/day]
<i>k_A</i>	$7.67565603 \times 10^{-4}$	0.00112417010	[U/(ng*day)]
<i>d_A</i>	0.0217662453	0.00264506271	[1/day]
<i>p_{C₁}</i>	6.28057306	6.56024353	[1/day]
h_{C₁} [61]	6.654	6.654	[1/day]
<i>d_{C₁}</i>	0.124593419	10.04228770	[1/(ml*day)]
<i>p_{C₂}</i>	5.74034856	6.90740402	[1/day]
h_{C₂} [62]	8.318	8.318	[1/day]
<i>b_{C₂}</i>	9.42365873	10.05044004	[ng/(ml*day)]
<i>c_{C₂}</i>	7.80929598	4.41713301	[ng/(ml*day)]
p_V [48]	0.6	1	[1/day]
a_V	4.5	4.5	[1]
k_V	1	1	[1/day]
<i>g_V</i>	0.00211499916	0.0141292456	[1]
<i>j_V</i>	0.00231880261	0.00215772575	[ml/ng]
<i>l_V</i>	0.00719681824	0.00729990238	[ml/U]
v₁ [50]	-	0.5	[1/day]
v₂	-	0.04	[1/day]
G	15	-	[ng]

Table 3.2: Estimated ODE parameters. Fixed parameters are bolded.

3.2.4 Analysis

This system is constructed from a combination of multiple ODE systems developed for autoimmune conditions and therefore is not a well-studied system. We tailored the ODEs to reflect our domain knowledge, and what we expect the solutions to look like, rather than ensuring the existence of analytical solutions or identifiability of parameters. This is because development of

such a system of ODEs is not the aim of our investigation; rather, we aim to explore whether the SR-PINNs model will be able to discover a system of coupled stiff ODEs. Nonetheless, we are interested in the theoretical properties of the ODE model as analysis may improve our ability to find solutions and can provide insight into how to improve the PINNs model if it struggles.

First we will explore whether closed-form analytical solutions to our two systems of ODEs exist (and are unique). Next we will explore whether the systems satisfy parameter identifiability.

Existence and uniqueness of solutions

We first try to solve the ODE system for active Celiac Disease patients. The ODE $\frac{dT}{dt}$ is a Riccati ODE of the form

$$\frac{dx}{dt} = f(t) + g(t)x + h(t)x^2,$$

where in the case of T , the functions f , g , and h are constants:

$$\frac{dT}{dt} = 13.618805355 + 0.267T - 0.010825T^2.$$

Let $a = -0.010825$, $b = 0.267$, and $c = 13.618805355$. We outline the solution steps below:

$$\begin{aligned} \frac{dT}{dt} &= aT^2 + bT + c, \quad T(0) = 4, \\ dt &= \frac{dT}{aT^2 + bT + c}, \\ t &= \int \frac{dT}{aT^2 + bT + c} \\ &= \frac{2}{\sqrt{4ac - b^2}} \tan^{-1} \left(\frac{2aT + b}{\sqrt{4ac - b^2}} \right) + C, \\ T &= \frac{-b + \sqrt{4ac - b^2} \tan\left(\frac{1}{2}\sqrt{4ac - b^2}(t - C)\right)}{2a}. \end{aligned}$$

Using the following identities

$$\begin{aligned} \tan^{-1}(z) &= \frac{1}{2i} \ln \left(\frac{1 + iz}{1 - iz} \right), \quad \tanh^{-1} = \frac{1}{2} \ln \left(\frac{1 + z}{1 - z} \right), \quad \text{for arbitrary } z \in \mathbb{C}, \\ \implies \tan^{-1}(ai) &= \frac{1}{2i} \ln \left(\frac{1 - a}{a + 1} \right) = -i \tanh^{-1}(a), \quad \text{for arbitrary } a \in \mathbb{R}, \end{aligned}$$

and the initial condition $T(0) = 4$ results in:

$$\begin{aligned} 4 &= \frac{-b + \sqrt{4ac - b^2} \tan\left(\frac{1}{2}\sqrt{4ac - b^2}(-C)\right)}{2a}, \\ \implies C &= -\frac{2}{\sqrt{4ac - b^2}} \tan^{-1} \left(\frac{8a + b}{\sqrt{4ac - b^2}} \right) \\ &= -2.46i \tan^{-1}(-0.181219i) \\ &= -(2.46i)(-i \tanh^{-1}(-0.181219)) = -0.450778. \end{aligned}$$

Then we can solve for T :

$$T = \frac{-0.267 + 0.813008777241i \tan\left(\frac{1}{2}0.813008777241i(t + 0.450778)\right)}{-0.02165}.$$

Using the identity

$$\tan(ix) = i \tanh(x), \quad \text{for arbitrary } x \in \mathbb{R},$$

this simplifies to

$$T = 12.336 - 37.552 \tanh(0.18324 + 0.4065t).$$

The equations $\frac{dI}{dt}$ and $\frac{dC_2}{dt}$ are coupled. Both derivatives are linear in I and C_2 , and therefore the system can be rewritten as a system of linear ODEs:

$$\begin{aligned} \frac{dX}{dt} &= B(t)X + F(t), \quad X(t) = \begin{pmatrix} I(t) \\ C_2(t) \end{pmatrix}, \\ A(t) &= \begin{pmatrix} -(k_I + \frac{p_I}{a_I}T(t)) & s_I \\ b_{C_2} & p_{C_2} - h_{C_2} \end{pmatrix}, \\ F(t) &= \begin{pmatrix} p_I T(t) \\ c_{C_2} T(t) \end{pmatrix}. \end{aligned}$$

Since $\frac{dI}{dt}$ is dependent on $T(t)$, this is a non-autonomous coupled system, which typically only have closed-form solutions for specific functions T . Additionally, $\frac{dV}{dt}$ is both nonlinear and non-autonomous, containing the term

$$V \frac{H(t)^2}{H(t)^2 + 9}, \quad \text{where } H(t) = g_V(I + T) + j_V C + l_V A.$$

This structure makes analytic solution methods intractable in most cases. Even if an explicit closed-form solution were to exist, it would likely be too complicated to yield meaningful biological insight. For this reason, we do not attempt to derive closed-form solutions and instead proceed using numerical methods.

Similarly, the ODE system describing recovering Celiac Disease has the same structure and therefore difficulties in finding analytical solutions would be the same. Therefore, the recovering model is also not expected to have an insightful closed-form solution in general. Consequently, we focus on qualitative and numerical analysis for both models rather than seeking explicit analytic formulas.

Identifiability

We use $p - h$ and $p - k$ terms in our ODE model which leads to non-identifiability of p , h , and k . Since we fix these terms, this is not a problem for the ODE model, but means that the PINNs model will only discover $p - h$ and $p - k$. We additionally use the quotients $\frac{p}{a}$ in some of the ODEs. This will result in the PINNs model not discovering a directly. Another obvious problem is dG . In the case of active Celiac Disease, G is treated as a constant. It is possible that the parameter d is not identifiable, and d may not be discovered by the PINNs model.

Another possible problem is with the parameters g_V , j_V , and l_V . It is possible that the parameters are correlated, but they are most likely identifiable. However, the ratio in $\frac{dV}{dt}$ will not be able to be discovered by the PINNs model, so the identifiability of these parameters is not a big issue. See Section 4.4 for more information.

3.2.5 Numerical solution(s)

Popular methods for solving linear ODEs include Runge-Kutta methods, linear multistep methods, and more recently machine learning methods such as feedforward Neural Networks. We will use SciPy to estimate the parameters and solutions to our ODEs, which utilizes linear multistep methods Adams-Bashforth-Moulton and the Backward Differentiation Formula (BDF) depending on the problem stiffness.

Data for fitting the ODEs and estimating the parameters is difficult to come by. Oftentimes,

patients are measured at baseline and post-trial, and typically very small sample sizes (around 20 patients) are used. Hence, data is extremely variable between patients. Instead of using exact values at specific timepoints, we will use ranges of values. The reasonable ranges are given in Table 3.1. Exact ranges used are given below in Tables 3.3, 3.5.

We use the SciPy `solve_ivp` function with the method `LSODA`, because integration fails using other methods (explicit Runge-Kutta method `RK45` of order 5 used for non-stiff problems and implicit methods `Radau` and `BDF`). The method `LSODA` works for smooth, linear and nonlinear, coupled, stiff and non-stiff problems (so should work for our problem). The method `LSODA` is an Adams/backward differentiation formula (BDF) method which utilizes switching, i.e. automatic detection of the stiffness of the problem after each integration step, allowing the method to change between Adams methods for non-stiff problems and BDF methods for stiff problems.

A BDF method used to solve the IVP

$$y' = f(t, y), \quad y(t_0) = y_0,$$

has the general formula

$$\sum_{k=0}^s \alpha_k y_{n+k} = h\beta f(t_{n+s}, y_{n+s}),$$

where h is the step size s.t. $t_n = t_0 + nh$, s is the order of the method used (up to order 5), the coefficients α_k and β are chosen so that the method achieves order s . Since f is evaluated at y_{n+s} , the method is implicit. The first order BDF is equivalent to the Backwards-Euler method:

$$y_{n+1} - y_n = hf(t_{n+1}, y_{n+1}).$$

When the problem is detected to be stiff, the order is chosen based on stability; the highest possible stable order is chosen (higher order improves accuracy but decreases stability).

In the case of a non-stiff problem, the Adams-Moulton formulas (implicit Adams) are used:

$$y_n = y_{n-1} + h \sum_{i=0}^{q-1} \beta_i y'_{n-i},$$

where q is the order, β_i depend only on q , and h is the step size. Again, order and step size are determined for maximum accuracy and stability.

Scipy `least_squares` function: Given the residuals $f(x)$ (an m -dimensional real function of n real variables) and the loss function $\rho(s)$ (a scalar function), `least_squares` finds a local minimum of the cost function $F(x)$:

$$\min_{\text{s.t. } lb \leq x \leq ub} F(x) = \frac{1}{2} \sum_i^{m-1} \rho(f_i(x)^2).$$

In our case, we aim to minimize the residuals of the estimated values of the concentrations $[I, T, A_1, C_1, C_2, V]$ given the parameters of the ODE system defined above. To do so, we penalize the residuals whenever they stray outside of the bounds.

Active Celiac Disease

Using the reasonable ranges of the concentrations, we impose ranges that the ODE solutions must follow. These are given in Table 3.3. Although not every concentration has been measured in

literature on day 15, we simply impose reasonable boundaries in order to help the ODE solver estimate the solutions.

Variable	Day 0	Day 15	Day 365
T	1.5-6.4	8-22.6	10-40.6
I	15.7-36.2	20-50	20-80
A	4-14	30-60	30-90
C_1	0-150	0-500	300-600
C_2	0-175	0-500	300-900
V	2-4	2-4	1-2

Table 3.3: Imposed ranges of concentrations for active Celiac Disease.

The initial values are estimated from the reasonable ranges of concentrations measured in recovered Celiac Disease patients. They are given in Table 3.4.

Variable	Initial value
T	4
I	22
A	10
C_1	77
C_2	77
V	4

Table 3.4: Initial values of concentrations in active Celiac Disease.

The estimated values of the reaction concentrations are visualized in Figure 3.2.

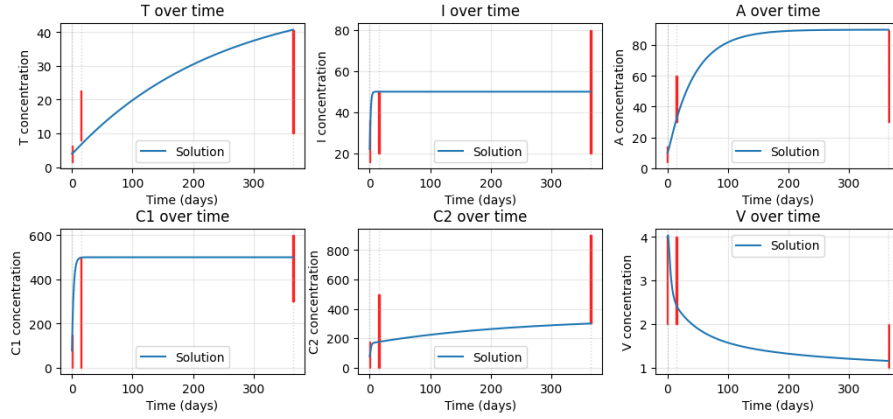


Figure 3.2: Estimated trajectories of the reaction variables. Desired ranges of the variables are given on days 0, 15, and 365 in red. Ranges are given in Table 3.3.

Recovering Celiac Disease

The imposed ranges on the variables are given in 3.5.

Variable	Day 0	Day 15	Day 365
T	10-40.6	8-30	1.500-6.400
I	20-80	20-60	15.7-36.2
A	30-90	30-70	0.5-1.1
C_1	0-150	300-500	365, 0-150
C_2	300-900	300-500	0-175
V	1-2	1-3	2-4

Table 3.5: Imposed ranges of concentrations for recovering Celiac Disease.

The initial values are taken from the final values measured on day 365 from the active Celiac Disease solutions. They are given in Table 3.6.

Variable	Initial value
T	40.7
I	50
A	90
C_1	500.47
C_2	300
V	1.15

Table 3.6: Initial values of concentrations in recovering Celiac Disease (given by final measurement of solution to ODEs describing active Celiac Disease).

The estimated values of the reaction concentrations are visualized in Figure 3.3.

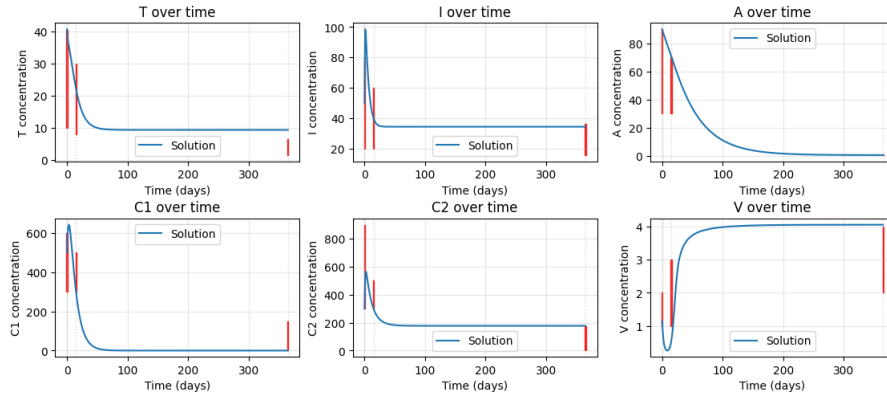


Figure 3.3: Estimated trajectories of the reaction variables. Desire ranges of the variables are given on days 0, 15, and 365 in red. Ranges are given in Table 3.5.

3.2.6 Problems

We now have data points for a single estimated patient with Celiac Disease for 365 days. We will use this data to test whether the PINNs model can discover such a coupled system of ODEs. If the PINNs model can discover this system within reasonable error bounds, we will explore the

necessary amount of data required for realistic estimation of the system. However, it is likely that the PINNs model will not be able to discover the ODEs that were used in creating this data due to several issues explained below.

Firstly, the fraction used to model V is far more complicated than any example equation used in [18]. In fact, using a standard candidate library and the PINN-SR model developed by [18], it will not be possible to discover the fraction. Implicit methods such as SINDY-PI [65] are necessary to estimate rational terms in ODEs, which we will not be exploring. The SR-PINN model will have to use terms from the candidate library, which we describe in Section 4.4, to fit the data we generate from the ODE.

Secondly, we have 365 measurement points for 6 variables, which is an unreasonably large number of measurement points if a real study were to be conducted with human subjects. However, while we severely over-estimate the number of measurements we might have access to were this data taken from a real study, the method described in [18] requires an even larger number of points. The smallest number of datapoints used is 1,000 for discovery of the Burgers' equation, which is one of the simplest PDEs to discover. The discovery of the coupled $\lambda - \omega$ reaction-diffusion PDEs used ~ 37.5 k datapoints. They also discover one PDE on an experimental dataset describing cell densities over time over 190 datapoints, but this was relatively simple, and we are discovering 6 coupled ODEs.

The final and likely most impactful issue is the scale of our parameters. The orders of magnitude of the parameters are very different, with some of order 1×10^2 and many of orders 1×10^{-2} to 1×10^{-3} , whereas the PDEs discovered in [18] have parameters of orders 1 to 1×10^{-2} . The main solution to this issue is non-dimensionalization of the ODEs, which is unrealistic when using real data.

In practical settings it is likely that the data is not described by closed-form ODEs at all, and if they are described this way, the ODEs are likely to be as complicated if not more than those we have constructed. Nonetheless, if the PINNs model from [18] discovers different ODEs than we expect, we will explore why, and may test the PINNs model with simpler ODEs.

Chapter 4

Sparse Regression PINNs

Physics-informed neural networks (PINNs) offer a promising approach to discovering governing equations from data by combining neural network function approximation with sparse regression techniques. This chapter investigates the application of the Sparse Regression PINN (SR-PINN) framework developed by [18] to our biological system data. While this method has demonstrated success on well-conditioned partial differential equations with abundant synthetic data, its application to real-world biological systems presents unique challenges.

Our investigation reveals a fundamental incompatibility between the smooth approximations produced by neural networks and the sharp transient dynamics characteristic of biological systems. The neural network’s tendency to produce overly smooth predictions—particularly at early timepoints where our system exhibits rapid changes—prevents accurate derivative estimation, which is essential for identifying the correct governing equations.

We begin by formulating the problem and describing the ADO (Alternating Direction Optimization) algorithm that alternates between sparse coefficient identification and neural network training (Sections 4.1–4.2). We then detail our experimental design, including data preprocessing, candidate library construction, and hyperparameter selection (Sections 4.3–4.5). Our experimental results (Section 4.6) systematically explore modifications to address the smoothness problem, including tolerance regularization, alternative loss functions, and variable subset analysis. Finally, we discuss fundamental limitations and potential improvements for future work (Section 4.7).

4.1 Problem formulation

We adapt the problem formulation from [24]. Consider a system of nonlinear ordinary differential equations:

$$\dot{u}(t) + \mathcal{F}[u(t); \lambda] = p(t)$$

with initial condition $u(0) = u_0$, where $u(t) \in \mathbb{R}^n$ denotes the solution vector, $\dot{u}(t)$ denotes the first time derivative, $\mathcal{F}$ is a nonlinear functional parameterized by λ , and $p(t)$ is a source term. Our objective is to discover the closed form of $\mathcal{F}[\cdot]$.

Physical systems are typically governed by equations sparse in a high-dimensional nonlinear function space—only a few terms from many possible candidates are active [66]. This sparsity assumption allows us to reformulate the problem as:

$$\dot{u}(t) = \phi(u)\Lambda$$

where $\phi = \phi(u) \in \mathbb{R}^s$ is a library of candidate functions with s terms, and $\Lambda \in \mathbb{R}^{s \times n}$ is a sparse coefficient matrix. This library is chosen such that only a few terms in ϕ describe $\mathcal{F}$, therefore establishing a sparse regression problem to determine which terms in ϕ are active. Therefore, $\Lambda \in \mathbb{R}^{s \times n}$ is the sparse coefficient matrix, so our problem becomes simplified to finding Λ such that $\dot{u}(t) = \phi\Lambda$ holds [18].

Since we observe data only at discrete times, we approximate both $u(t)$ and $\dot{u}(t)$ for $t \in \mathbb{R}$ using a deep neural network: $u_\theta = u(t; \theta)$, where θ denotes the trainable parameters of the DNN. We

denote the library of candidate functions as ϕ^θ , which is manually constructed before training.

The residual function

$$\mathcal{R}^\theta := \dot{u}_\theta(t) - \phi^\theta \Lambda$$

quantifies how well the discovered ODE fits the DNN approximation of the data. The aim of the method is to learn both the DNN parameters θ and the ODE coefficients Λ such that the DNN can satisfy the measurement data as well as the constraints of the ODE. The residual function will be sampled randomly on a large number of collocation points $D_c = \{t_i\}_{i=1}^{N_c}$, which will be used to describe the residual physical loss function $\mathcal{L}_p(\theta, \Lambda; D_c)$. In our case, $D_c \subset [0, 365]$.

Let $D_m = \{(t_i, u(t_i))\}_{i=1}^{N_m}$ describe the measurement data. The loss function can be written as the summation of the data loss $\mathcal{L}_d$, the physical residual loss $\mathcal{L}_p$, and a regularization term:

$$\mathcal{L}(\theta, \Lambda; D_m, D_c) := \mathcal{L}_d(\theta; D_m) + \alpha \mathcal{L}_p(\theta, \Lambda; D_c) + \beta \|\Lambda\|_0,$$

where $\|\cdot\|_0$ is the ℓ_0 norm s.t. $\|\Lambda\|_0$ promotes the sparsity of Λ when the loss is minimized. The data loss function is given by

$$\mathcal{L}_d(\theta, D_m) = \frac{1}{N_m} \|u^\theta - u^m\|_2^2,$$

where u^m is the measurement data and N_m is the total number of datapoints. As before, u^θ is the DNN-approximation of the solution. The physical loss function is given by

$$\mathcal{L}_p(\theta, \Lambda, D_c) = \frac{1}{N_c} \|\dot{U}(\theta) - \Phi(\theta)\Lambda\|_2^2,$$

where $\dot{U}$ is the discretized first-order derivative of $u(t)$ and Φ is the library of candidate functions evaluated on the collocation points. The number of collocation points is given by N_c .

The hyperparameters α and β balance these competing objectives. Important to note is that optimizing over the total loss is an NP-hard problem due to the regularization by the ℓ_0 norm, which is non-differentiable and non-convex (and notably not a norm) [67]. Hence, rather than directly solving the optimization problem, an alternating direction optimization (ADO) algorithm is used in order to divide the problem into solvable subproblems.

4.2 Alternating Direction Optimization (ADO)

The Alternating Direction Optimization (ADO) algorithm addresses the NP-hard problem of minimizing the loss function $\mathcal{L}$ by decomposing it into two tractable subproblems, solved iteratively:

$$\begin{aligned} \Lambda_{k+1}^* &:= \arg \min_{\Lambda} \{\|\dot{U}(\theta_k^*) - \Phi(\theta_k^*)\Lambda\|_2^2 + \beta \|\Lambda\|_0\}, \\ \theta_{k+1}^* &:= \arg \min_{\theta} \{\mathcal{L}_d(\theta; D_m) + \alpha \mathcal{L}_p(\theta, \Lambda_{k+1}^*; D_c)\}. \end{aligned}$$

ADO alternates between:

1. **STRidge step:** Update sparse coefficients Λ given current network parameters θ (Section 4.2.1)
2. **Adam step:** Update network parameters θ given current coefficients Λ using gradient descent

Before ADO, pretraining initializes θ and Λ by optimizing with ℓ_1 regularization instead of ℓ_0 :

$$\{\theta^*, \Lambda^*\} = \arg \min_{\{\theta, \Lambda\}} \{\mathcal{L}_d(\theta; D_m) + \alpha \mathcal{L}_p(\theta, \Lambda; D_c) + \gamma \|\Lambda\|_1\}.$$

The ℓ_1 norm provides a convex approximation to sparsity, yielding reasonable initial estimates though with limited sparsity. Pretraining employs both Adam (for global search) and L-BFGS (for local refinement).

The complete ADO procedure is detailed in Algorithm 1.

Algorithm 1 Alternating Direction Optimization (ADO)

```

1: procedure ADO( $D_m, D_c, \alpha, \beta, \Delta\delta, \hat{\lambda}_0, \hat{\theta}_0, n_{\max}, n_{\text{str}}, n_{\text{it}}$ )    #  $\hat{\lambda}_0, \hat{\theta}_0$  are from pretraining via
   combined Adam and L-BFGS.
2:   Split measured data  $D_m$  into training and validation sets  $D_m^{\text{tr}} \in \mathbb{R}^{n_{\text{tr}} \times d}$ ,  $D_m^{\text{val}} \in \mathbb{R}^{n_{\text{val}} \times d}$ 
   with  $n_{\text{tr}}/n_{\text{val}} = 80/20$ ,  $N_m = n_{\text{tr}} + n_{\text{val}}$ .
3:   Split collocation points  $D_c$  into training and validation sets  $D_c^{\text{tr}} \in \mathbb{R}^{m_{\text{tr}} \times d}$ ,  $D_c^{\text{val}} \in \mathbb{R}^{m_{\text{val}} \times d}$ 
   with  $m_{\text{tr}}/m_{\text{val}} = 80/20$ ,  $N_c = m_{\text{tr}} + m_{\text{val}}$ .
4:   for  $k = 1 : n_{\max}$  do
5:     for  $i = 1 : d$  do                                # We consider a system of dimension  $d$ ; in our case,  $d = 6$ 
6:        $\dot{U}_i^{\text{tr}} \leftarrow \dot{u}_i^{\hat{\theta}_{k-1}}(t^{\text{tr}})$                 # where  $t^{\text{tr}}$  are the collocation points in  $D_c^{\text{tr}}$ 
7:        $\dot{U}_i^{\text{val}} \leftarrow \dot{u}_i^{\hat{\theta}_{k-1}}(t^{\text{val}})$             # where  $t^{\text{val}}$  are the collocation points in  $D_c^{\text{val}}$ 
8:        $\Phi^{\text{tr}} \leftarrow \phi^{\hat{\theta}_{k-1}}(t^{\text{tr}})$ ,  $\Phi^{\text{val}} \leftarrow \phi^{\hat{\theta}_{k-1}}(t^{\text{val}})$     # Assemble candidate function libraries
9:        $\beta^i \leftarrow \kappa \mathcal{L}_p^i(\hat{\theta}_0, \hat{\lambda}_0^i; D_c^{\text{val}})$                 #  $\ell_0$  regularization parameters
10:       $\epsilon^i \leftarrow \mathcal{L}_p^i(\hat{\theta}_{k-1}, \hat{\lambda}_{k-1}^i; D_c^{\text{val}}) + \beta^i \|\hat{\lambda}_{k-1}^i\|_0$ 
11:       $\delta_1^i \leftarrow \Delta\delta$                                     # initial threshold tolerance
12:    end for
13:    for  $j = 1 : n_{\text{str}}$  do
14:      for  $i = 1 : d$  do
15:         $\tilde{\lambda}_i \leftarrow \text{STRidge}(\dot{U}_i^{\text{tr}}, \Phi^{\text{tr}}, \delta_j^i, \hat{\lambda}_{k-1}, n_{\text{it}})$     # STRidge as in Algorithm 2
16:         $\tilde{\epsilon}^i \leftarrow \mathcal{L}_p^i(\hat{\theta}_{k-1}, \tilde{\lambda}_{k-1}^i; D_c^{\text{val}}) + \beta^i \|\tilde{\lambda}_{k-1}^i\|_0$ 
17:        if  $\tilde{\epsilon}^i \leq \epsilon^i$  then
18:          better_error  $\leftarrow$  TRUE
19:           $\delta_{j+1}^i \leftarrow \text{DeltaUpdate}(\delta_j^i, \text{better\_error})$ 
20:        else
21:          better_error  $\leftarrow$  FALSE
22:           $\delta_{j+1}^i \leftarrow \text{DeltaUpdate}(\delta_j^i, \text{better\_error})$ 
23:        end if
24:      end for
25:    end for
26:     $\hat{\Lambda}_k \leftarrow \{\tilde{\lambda}^i\}_{i=1}^d$                                 # Current best solution from STRidge cycles
27:     $\hat{\theta}_k \leftarrow \arg \min_{\theta} \{\mathcal{L}_d(\theta; D_m) + \alpha \mathcal{L}_p(\theta, \hat{\Lambda}_k; D_c)\}$     # Train DNN given  $\hat{\Lambda}_k$  via Adam with
    $\{D_m^{\text{tr}}, D_c^{\text{tr}}\}$  and validate with  $\{D_m^{\text{val}}, D_c^{\text{val}}\}$ 
28:  end for
29:  output: The best solution  $\theta = \hat{\theta}_{n_{\max}}$  and  $\Lambda = \hat{\Lambda}_{n_{\max}}$ 
30: end procedure

```

In [18], DeltaUpdate is given either by

$$\delta_{j+1} = \begin{cases} 1.2 \delta_j & \text{if better_error} = \text{TRUE}, \\ 0.8 \delta_j & \text{if better_error} = \text{FALSE} \end{cases} \quad (4.1)$$

or

$$\begin{aligned} \Delta\delta &= \Delta\delta/1.618 \quad \text{if better_error} = \text{FALSE} \\ \delta_{j+1} &= \begin{cases} \delta_j + \Delta\delta & \text{if better_error} = \text{TRUE}, \\ \max\{\delta_j - 2\Delta\delta, 0\} + \Delta\delta & \text{if better_error} = \text{FALSE}, \end{cases} \end{aligned} \quad (4.2)$$

probably depending on performance (although they do not specify this). In our case, we try the update (5.1) and the smoothed update given by

$$\delta_{j+1} = \begin{cases} \rho\delta_j + (1-\rho)1.2\delta_j & \text{if better_error} = \text{TRUE}, \\ \rho\delta_j + (1-\rho)0.8\delta_j & \text{if better_error} = \text{FALSE}. \end{cases} \quad (4.3)$$

The results of both are given in Section 4.6.

4.2.1 STRidge algorithm

Sequential Thresholded Ridge Regression (STRidge) used in [18] approximates ℓ_0 regularization of Λ and determines the sparsity of the coefficients of the candidate function library. STRidge addresses the limitations posed by Ridge regression, Lasso, and Sequential Thresholding Least Squares (STLS).

Lasso is convex optimization using ℓ_1 regularization, and reduces coefficients to 0. However, Lasso often has problems when data is correlated, which is certainly the case in our dataset which is composed of several coupled variables. In STLS, coefficients are iteratively thresholded. STLS is a common alternative to Lasso, but still fails when data is highly correlated. Ridge is a variation of least squares estimation using ℓ_2 regularization meant to mitigate problems that arise due to correlations in the data. However, because Ridge regression uses ℓ_2 regularization, the coefficients never become 0. In order to enforce sparsity, [68] propose hard thresholding through the form of Sequential Thresholding combined with Ridge regression, resulting in the algorithm STRidge.

Algorithm 2 Sequential Threshold Ridge Regression (STRidge)

```

1: procedure STRIDGE( $\dot{U}, \Phi, \delta, \hat{\lambda}, n_{it}$ )
2:   for  $k = 1 : n_{it}$  do
3:      $I_k \leftarrow \{i : \hat{\lambda}_i < \delta\}, \quad J_k \leftarrow \{j : \hat{\lambda}_j > \delta\}$       # Determine indices of coefficients  $\hat{\lambda}$  to
       change.
4:     if  $k > 1$  and  $J_k == J_{k-1}$  then
5:       break                                          # If the sparsity pattern is unchanged, break
6:     else
7:        $\hat{\lambda}_I \leftarrow 0$                                # Set small coefficients to 0 (sequential thresholding)
8:     end if
9:      $\hat{\lambda}_J \leftarrow \arg \min_{\lambda_J} \{\|\Phi_J \lambda_J - \dot{U}\|_2^2 + (1 \times 10^{-5})\|\lambda_J\|_2^2\}$       # Update remaining non-zero
       values with Ridge regression
10:  end for
11:  output: Sparse coefficient vector  $\hat{\lambda}$ 
12: end procedure

```

4.2.2 Optimization algorithms

Adam optimizer: We use Nadam (Nesterov-accelerated Adaptive Moment Estimation) from **TensorFlow** version 1, which combines the Nesterov accelerated gradient and Adam optimizer.

The current version (v.2) of the `Nadam` can be found in [69]. We use this optimizer with a learning rate of 1×10^{-2} during pretraining and 1×10^{-3} during ADO as in [18].

The Adam algorithm [70] is a stochastic gradient descent algorithm which uses momentum and adaptive learning rates with Root Mean Square Propagation (RMSProp) to accelerate the gradient descent process. Updates using momentum are given by:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \frac{df}{d\theta_t},$$

$$\theta_{t+1} = \theta_t - \alpha m_t,$$

where f is the objective function θ are the parameters to be estimated, and $\beta_1 \in [0, 1)$ is the momentum parameter. RMSProp uses the second moment vector to update parameters:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left(\frac{df}{d\theta_t} \right)^2,$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t} + \epsilon} \frac{df}{d\theta_t},$$

where $\beta_2 \in [0, 1)$, $\epsilon = 10^{-8}$. Adam combines both updates and corrects for the bias of the first and second moments which arises due to initialization at 0. The weight update becomes:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t},$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t},$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}.$$

The Nadam algorithm modifies the momentum to Nesterov momentum by updating the gradient by a small timestep, correcting the momentum term in case it overshoots [71]. Only the bias correction of the momentum and weight update changes:

$$\hat{m}_t = \frac{\beta_1}{1 - \beta_1^t} m_t + \frac{1 - \beta_1}{1 - \beta_1^t} \frac{df}{d\theta_t}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}.$$

Nesterov-accelerated Adam generally converges faster than Adam and improves loss.

L-BFGS optimizer: We use the `L-BFGS-B` (Limited-memory BFGS algorithm) optimizer from `SciPy`. The current version can be found in [72]. We use this optimizer during pretraining only.

The L-BFGS is a quasi-Newton method which minimizes a given function f w.r.t x , where x is a real-valued vector and f is a scalar function. The L-BFGS estimates the inverse of the Hessian and stores only a limited number of vectors which implicitly represent the approximation, thus using less memory than the BFGS which stores the approximation as a dense matrix, and is especially useful for large problems optimizing over many variables. The algorithm is described in [73].

4.3 Data preprocessing

Non-dimensionalization can have a major impact on the performance of a PINN model. While we could non-dimenesionalize the ODEs themselves, this is not realistic in practice as with real-world

data the ODEs would not be explicitly known. Instead, we aim to improve the results of the PINN model through scaling our data directly. This way, our method may be applied to other real-world datasets.

We use the min/max normalization method

$$\hat{x} = \epsilon + \frac{(1 - 2\epsilon)(x - x_{\min})}{x_{\max} - x_{\min}},$$

scaling each variable to $[\epsilon, 1 - \epsilon]$ where ϵ is a small positive constant. This method was proposed by [74] to improve NN estimation of stiff ODEs. The resulting ODEs are:

$$\frac{dx}{dt} = \frac{x_{\max} - x_{\min}}{t_{\max} - t_{\min}} \frac{d\hat{x}}{d\hat{t}}$$

We apply this transformation to all six variables (I, T, A, C_1, C_2, V) with $\epsilon = 0.01$.

4.4 Candidate function library

The candidate function library must reflect the ODEs that are being estimated. In [18], the candidate libraries contained partial derivative terms. Instead, we construct libraries with polynomial and interaction terms appropriate for our ODE system.

Our system includes a rational term in $\frac{dV}{dt}$ that cannot be directly represented in the standard SR-PINN framework. Implicit SINDy methods could identify such terms [75, 65] but require substantial model reformulation. Instead, we allow the algorithm to approximate $\frac{dV}{dt}$ using available polynomial and interaction terms, accepting that perfect recovery may not be possible.

When discovering all six equations simultaneously, we use the candidate library:

$$\phi = [\mathbf{1}, I, T, A, C_1, C_2, V, T^2, V^2, IT, IV, TV, AV, C_1V, C_2V]. \quad (4.4)$$

When excluding V (which we test since I, T, A, C_1, C_2 do not depend on V):

$$\phi = [\mathbf{1}, I, T, A, C_1, C_2, T^2, IT]. \quad (4.5)$$

These libraries contain only terms appearing in the true equations, providing the algorithm the best possible starting point. If this minimal library produces good results, larger libraries with additional candidate terms could test the method's ability to identify correct terms amid distractors.

4.5 Hyperparameter selection

Physics loss weight (α) We compute α based on the ratio of data scale to derivative scale:

$$\alpha = r_\sigma = \left(\frac{\sigma(u^m)}{\sigma(\dot{u}^m)} \right)^2$$

where derivatives are estimated via central differences. Following [18], we use $\alpha = r_\sigma/5$ during pretraining (relaxing physics constraints to prioritize data fitting) and $\alpha = r_\sigma$ during ADO. Values for our normalized data are shown in Table 4.1.

Configuration	α (Pretraining)	α (ADO)
All variables	5.87×10^{-4}	2.93×10^{-3}
Excluding V	2.29×10^{-4}	1.15×10^{-3}
Only T	5.00×10^{-6}	2.50×10^{-5}

Table 4.1: Physics loss weights for different variable configurations.

Sparsity weight (β) Following [18], we set $\beta = \kappa \mathcal{L}_p(\hat{\theta}_0, \hat{\Lambda}_0; D_c^{\text{val}})$ where κ is determined via Pareto analysis (Figure 4.1). While optimal κ differs by variable, $\kappa = 100$ provides the best overall balance.

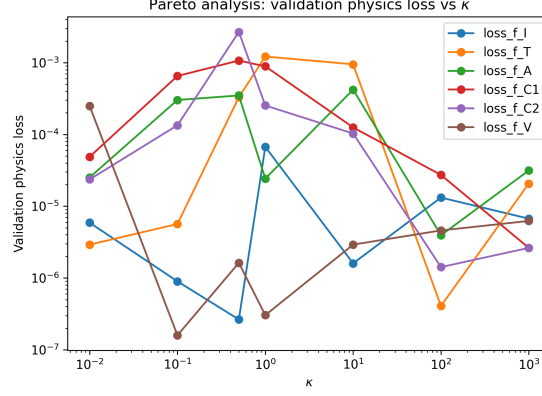


Figure 4.1: Pareto analysis of sparsity parameter κ . Each line represents a different variable. While individual optima vary, $\kappa = 10^2$ minimizes average error across all variables.

Other hyperparameters The rest of the hyperparameter choices are shown in Table 4.2. The collocation points N_c are sampled on $[\epsilon, 1 - \epsilon]$, the normalized time range.

Parameter	Value
γ (ℓ_1 weight in pretraining)	10^{-2}
β (sparsity weight)	$100 \times \mathcal{L}_p(\hat{\theta}_0, \hat{\Lambda}_0)$
$n_{\max}$ (ADO iterations)	6
n_{str} (STRidge cycles per iteration)	1000
n_{it} (Ridge iterations in STRidge)	100
$\Delta\delta$ (initial threshold)	1
N_c (collocation points)	10^3

Table 4.2: Hyperparameter configuration. Values follow [18] recommendations with adjustments based on preliminary experiments.

Network architecture (number of layers and neurons per layer) is determined empirically. The authors [18] typically use 2–8 hidden layers with 20–60 neurons per layer; we explore this range based on validation error.

4.6 Results

This section documents our systematic investigation of the SR-PINN framework’s performance on biological system data. We organize our experiments by the modifications attempted.

4.6.1 Baseline implementation

Applying the SR-PINN framework directly from [18] with our normalized data and candidate library produces smooth neural network predictions that fail to capture the system dynamics (Figure 4.2).

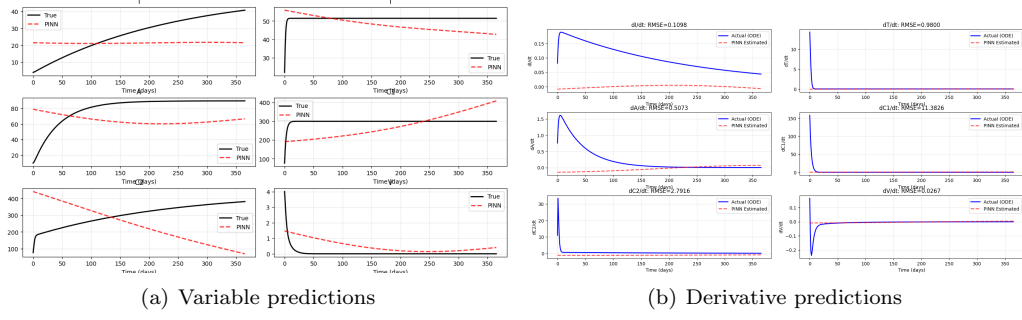


Figure 4.2: Baseline SR-PINN predictions. The network produces overly smooth approximations that miss sharp transients in the data, particularly at early timepoints. Since coefficient identification relies on derivative accuracy, this smoothness propagates to incorrect equation discovery.

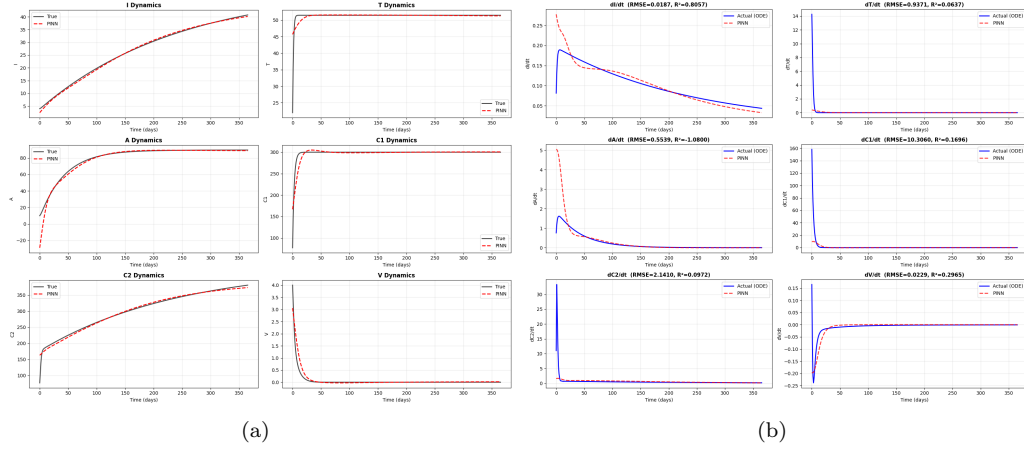
The fundamental issue is likely not coefficient identification, but rather the neural network’s inability to accurately approximate the data. The derivatives computed from these smooth approximations bear little resemblance to the true derivatives, making accurate equation discovery impossible.

4.6.2 Tolerance regularization

When training Adam with the full loss $\mathcal{L}$, STRidge has an impact on the NN approximations of the data, as Adam aims to minimize both the data and the physical loss. Because of this, STRidge may have a significant, although indirect, impact on Adam’s training. Because the model was developed for large, artificial datasets generated by non-stiff, well-conditioned PDEs, it is not applicable to smaller, more complex datasets such as ours. The STRidge Algorithm (Algorithm 2) is highly variable on small datasets, and the model outputs change drastically for different choices of hyperparameters. The ℓ_0 regularization term is very strong, and promotes sparsity whereas ℓ_1 or ℓ_2 regularization would output far more terms to explain the equations. However, this results in less stability.

We add tolerance regularization in two forms: we modify the tolerance update to be smoother, and we add hysteresis.

Smoothed threshold updates We first use the tolerance update given in equation (5.3), with $\rho = 0.1, 0.2, 0.5, 0.8, 0.9$, and find the best results with $\rho = 0.8$. However, this produces lengthy equations (e.g. 9 terms for $\frac{dI}{dt}$ versus 3 in the true equation) and only marginally improves derivative predictions (Figure 4.3, Table 4.3).

Figure 4.3: Predicted variables with PINN model with smoothed tolerance update; $\rho = 0.8$.

Variable	Variable Error		Derivative Error		
	Rel. L2	MSE	MSE	RMSE	R^2
I	0.011	5.68×10^{-5}	0.010	0.019	0.806
T	0.028	7.39×10^{-4}	0.878	0.937	0.064
A	0.024	4.72×10^{-4}	0.307	0.554	-1.080
C_1	0.024	5.36×10^{-4}	106.2	10.31	0.170
C_2	0.022	2.93×10^{-4}	4.58	2.141	0.097
V	0.292	4.96×10^{-4}	5.252×10^{-4}	0.023	0.297
Average	0.068	4.321×10^{-4}	18.66	2.330	0.059

Table 4.3: Estimation error between NN predictions and true values when using tolerance update given by Equation (5.2); $\rho = 0.8$.

Hysteresis To prevent coefficients near the threshold from rapidly toggling on/off, we implement a hysteresis band $[0.95\delta, 1.05\delta]$ where coefficients maintain their previous state (Algorithm 3). This modification does not improve results.

Algorithm 3 STRidge with Hysteresis

```

1: procedure STRIDGE( $\dot{U}, \Phi, \delta, \hat{\lambda}, n_{it}$ )
2:    $S_0 \leftarrow \mathbb{1}_{|\hat{\lambda}_i| > \delta}$  # Initial active set
3:   for  $k = 1 : n_{it}$  do
4:      $S_i^k \leftarrow \begin{cases} 1 & \text{if } |\hat{\lambda}_i| > 1.05\delta \\ 0 & \text{if } |\hat{\lambda}_i| < 0.95\delta \\ S_i^{k-1} & \text{otherwise} \end{cases}$  # Hysteresis band
5:      $I_k \leftarrow \{i : S_i^k == 0\}, \quad J_k \leftarrow \{j : S_j^k == 1\}$ 
6:     if  $k > 1$  and  $J_k == J_{k-1}$  then
7:       break
8:     end if
9:      $\hat{\lambda}_I \leftarrow 0$ 
10:     $\hat{\lambda}_J \leftarrow \arg \min_{\lambda_J} \{\|\Phi_J \lambda_J - \dot{U}\|_2^2 + (10^{-5})\|\lambda_J\|_2^2\}$ 
11:  end for
12:  return  $\hat{\lambda}$ 
13: end procedure

```

Overall, tolerance regularization addresses STRidge instability but does not resolve the underlying problem of poor derivative estimation from smooth neural network approximations.

4.6.3 Data loss training

Since the physics loss term may be forcing excessive smoothness, we train the neural network using only data loss:

$$\hat{\mathcal{L}}(\theta) = \mathcal{L}_d(\theta; D_m)$$

This improves variable predictions but maintains smooth derivatives (Figure 4.4). As expected, the discovered equations are entirely incorrect, with all true coefficients zeroed out.

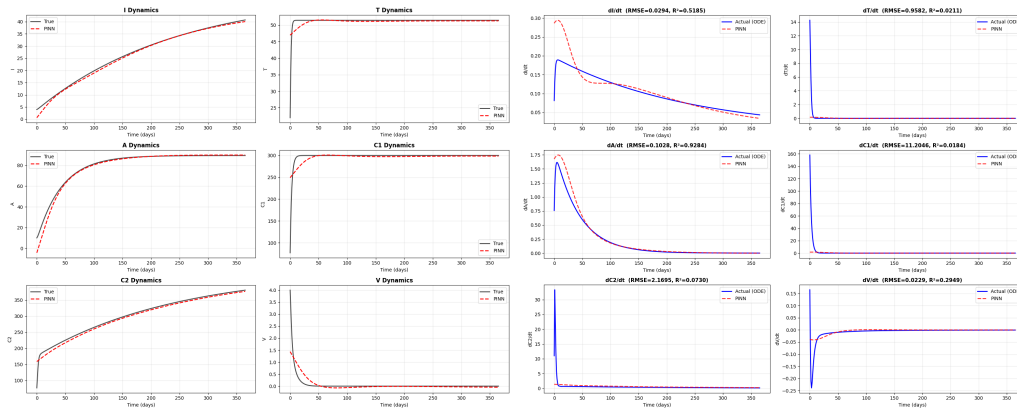


Figure 4.4: Predictions training with data loss only. Sharp early-time features remain elusive, and derivative predictions are inaccurate despite improved variable fitting.

Removing physics constraints improves data fitting but eliminates the mechanism for correct coefficient identification. The network has no incentive to produce an accurate Λ . Despite this, we will continue training with data loss only in order to try to improve the predictions of the data.

Without first improving the NN model of the data, the discovery of coefficients will never work well.

4.6.4 Alternative loss functions

The data loss is computed using L_2 loss, heavily penalizing large residuals and potentially encouraging smoothness at the expense of capturing sharp features. We test two alternatives:

Mean Absolute Error (MAE) MAE is less sensitive to outliers than L_2 . Results (Figure 4.5) are slightly worse than L_2 loss.

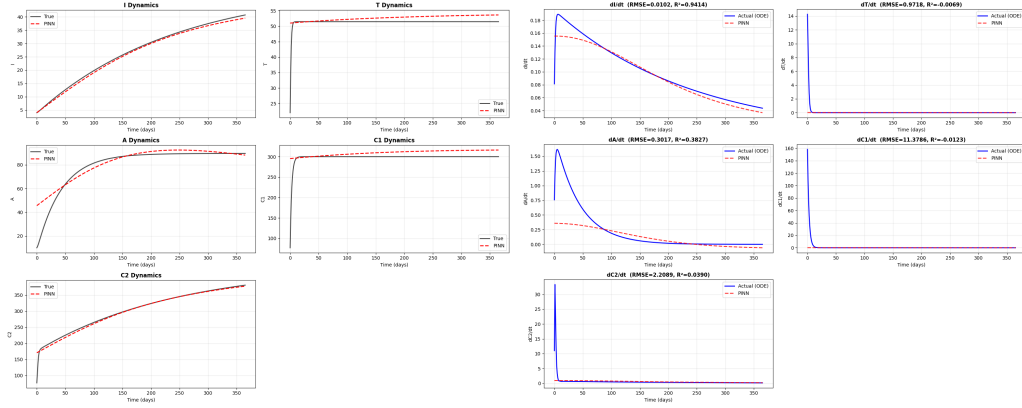


Figure 4.5: Predictions using MAE loss. Performance degrades compared to L_2 loss.

Huber Loss Transitions between quadratic and linear regimes:

$$L_\delta(x, \hat{x}) = \begin{cases} \frac{1}{2}(x - \hat{x})^2 & \text{if } |x - \hat{x}| < \delta \\ \delta(|x - \hat{x}| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Testing $\delta \in \{0.01, 0.1, 1, 5\}$, we find $\delta = 1$ performs best (Figure 4.6). While the fit of the NN to the data is worse, the fit of the derivatives shows marginal improvement over L_2 loss.

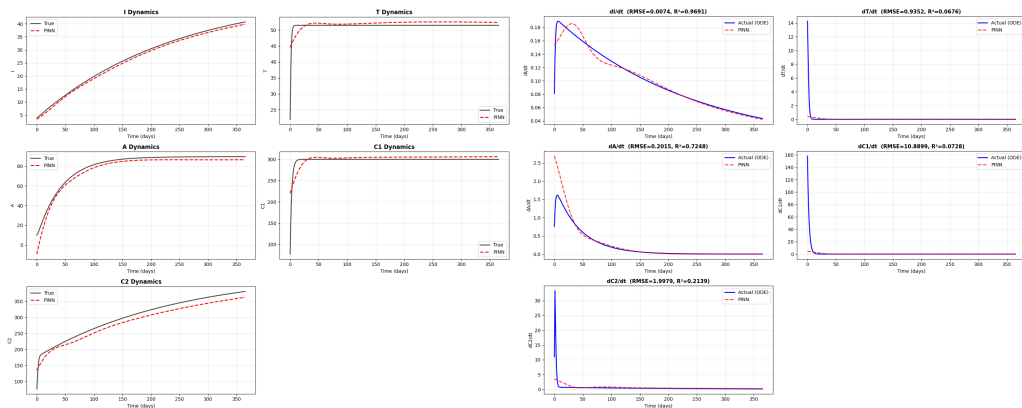


Figure 4.6: Predictions using Huber loss ($\delta = 1$). Slight improvement in capturing sharp features, but fundamental smoothness problem persists.

4.6.5 Biased collocation sampling

Since derivatives exhibit sharp changes primarily at early times ($t < 50$), we bias collocation point sampling to oversample this region. Using 100,000 total points with 60% in $t \in [0, 50]$ and 40% in $t \in [51, 365]$, combined with MAE loss, produces results shown in Figure 4.7. Greater bias ($>60\%$ early sampling) degrades performance.

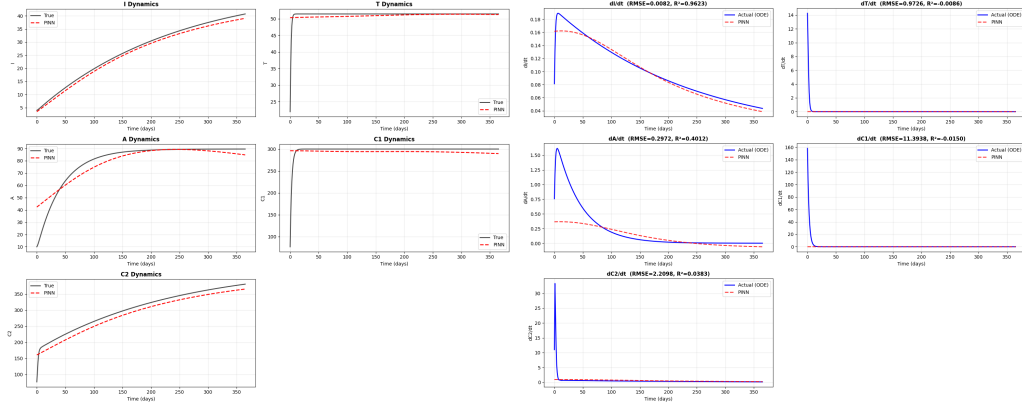


Figure 4.7: Predictions using MAE loss with biased collocation sampling (60% at $t < 50$). Modest improvement in early-time dynamics.

4.6.6 Combined best practices

Combining our most promising modifications—Huber loss ($\delta = 1$), biased collocation sampling (60% at $t < 50$), data loss only training, and excluding variable V —yields our best results (Figure 4.8, Table 4.4).

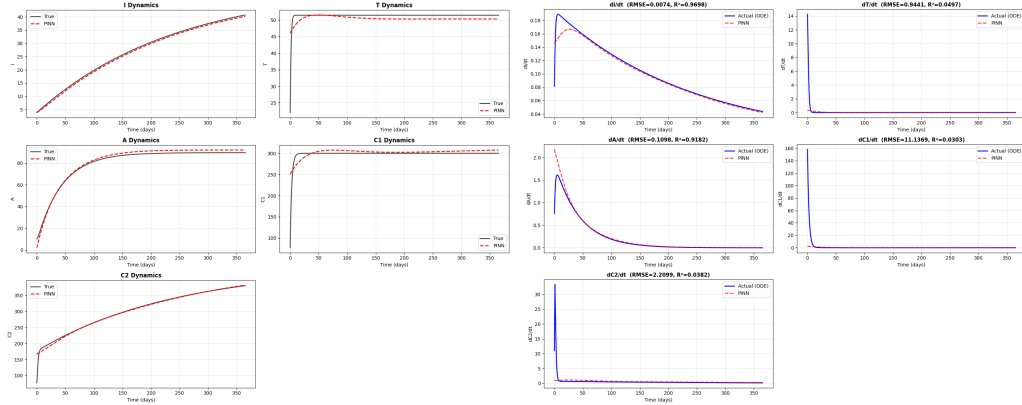


Figure 4.8: Best predictions combining Huber loss, biased sampling, and variable exclusion. While representing our most accurate attempt, sharp early-time features remain inadequately captured.

Variable	Variable Error		Derivative Error		
	Rel. L2	MSE	MSE	RMSE	R^2
I	0.019	1.68×10^{-4}	5.41×10^{-5}	7.36×10^{-3}	0.970
T	0.043	1.79×10^{-3}	0.891	0.944	0.050
A	0.032	8.28×10^{-4}	0.012	0.110	0.918
C_1	0.043	1.83×10^{-3}	124.0	11.14	0.030
C_2	0.020	2.24×10^{-4}	4.88	2.210	0.038
Average	0.031	9.68×10^{-4}	25.96	2.882	0.401

Table 4.4: Error metrics for best configuration. Variable I shows excellent derivative recovery ($R^2 = 0.970$), but other variables, particularly T and C_1 , remain problematic.

Even our best configuration fails to achieve the derivative accuracy ($R^2 \approx 0.9$ for all variables) necessary for reliable equation discovery. Variables I and A show promise, but the model struggles with variables exhibiting sharper dynamics.

4.6.7 Single variable analysis: variable T

To isolate difficulties, we train on uncoupled variable T alone using its minimal candidate library $[1, T, T^2]$, Huber loss, and biased collocation sampling. Despite this simplified setting, the model fails to capture T 's sharp initial dynamics (Figure 4.9, Table 4.5).

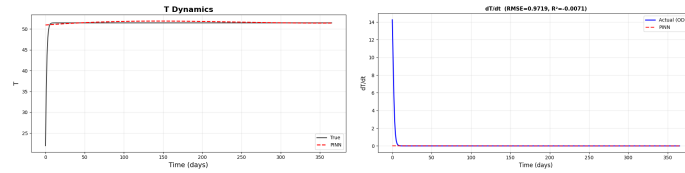


Figure 4.9: Single-variable analysis for T . Even with all optimizations and the correct candidate library, the model cannot capture the sharp initial transient.

Rel. L2	Variable Error		Derivative Error		
	MSE		MSE	RMSE	R^2
0.036	1.24×10^{-3}		0.943	0.971	-0.005

Table 4.5: Error metrics for single-variable T prediction. Negative R^2 indicates predictions worse than a constant model.

Discovered coefficients are completely incorrect (Table 4.6).

Term	Predicted	True
$\mathbf{1}$	-0.586	13.62
T	1.075	0.433
T^2	-0.262	-0.0108
MSE = 67.16, Cosine similarity = -0.42		

Table 4.6: Predicted versus true coefficients for $\frac{dT}{dt}$. Large errors and negative cosine similarity indicate the model recovers neither coefficient magnitudes nor their relative relationships.

We try a few methods for improving the estimation of T :

Increasing collocation points We try $N_c \in \{10^4, 10^5, 10^6\}$. This produces no improvement, confirming that insufficient sampling is not the primary issue.

Using ReLU activation function In [74], ReLU and GELU activation functions are used as an alternative to tanh activation functions in order to improve the prediction of stiff ODEs. Tensorflow 1.15 (which is what we use) does not have `gelu`. We therefore only try ReLU activation function. The results (Figure 4.10, Table 4.7) show significant improvement in the prediction of T , but still fail to capture the initial curve.

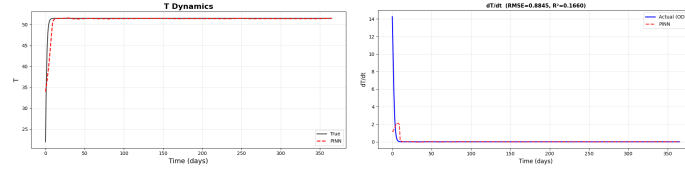


Figure 4.10: Single-variable analysis for T using ReLU activation function. Results are significantly improved versus tanh activation function, but are still not sufficient.

Variable Error		Derivative Error		
Rel. L2	MSE	MSE	RMSE	R^2
0.055	2.98×10^{-3}	0.782	0.884	0.166

Table 4.7: Error metrics for single-variable T prediction using ReLU activation function.

Weighted loss function We weigh the loss function so that it prioritizes fitting early times. Specifically, we define the weight

$$w(t) = \begin{cases} a \exp(-b \frac{t}{c}) & \text{if } t < c \\ 1 & \text{if } t \geq c, \end{cases}$$

and we define the empirical loss function (depending only on the data loss) as

$$\hat{\mathcal{L}} = \frac{1}{N} \sum_{i=0}^{365} w(t_i) \mathcal{L}_\delta(u(t_i), \hat{u}(t_i))$$

Clearly, choosing the scaling requires substantial fine-tuning. We try $a = 3, 5, 10$ and define b s.t. w is continuous. We maintain $c = 0.15$, so $b = \ln(\frac{1}{a})$ (note that t is normalized). We additionally increase the number of iterations from 1000 to 5000. The best results are when $a = 5$ (Figure 4.11). The results are significantly improved, and the sharp curve of T is well-estimated. However, clearly the derivative estimate is incorrect and exhibiting oscillations at early time points.

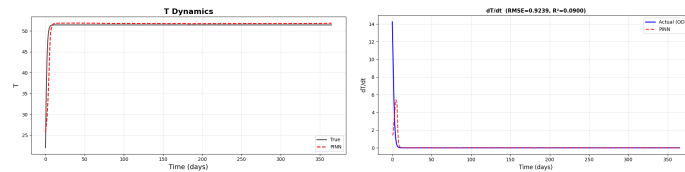


Figure 4.11: Estimate of T using weighted loss function.

Adaptive activation function We modify the activation function to be trainable. Specifically, we define the activation function function f as:

$$f(x) = \eta \text{ReLU}(x) + (1 - \eta) \tanh(x),$$

where η is a trainable parameter which is updated during Adam along with the weights and biases. Using this adaptive activation function (Figure 4.12) results in the best estimate of T , but oscillations in the derivative continue to be a problem.

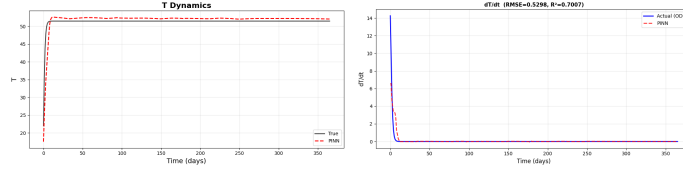


Figure 4.12: Estimate of T using adaptive activation function with ReLU activation.

Weighted loss and adaptive activation combined Finally, we combine both weighted loss and adaptive activation methods (Figure 4.13). This gives worse results than using the adaptive activation alone, which is interesting. It is possible that using different weights would result in an improved estimate, but selection of the ideal weight is highly unstable, with different weight values resulting in extremely different results.

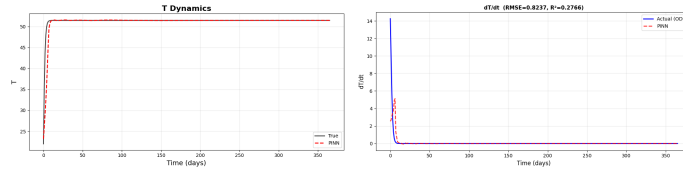


Figure 4.13: Estimate of T using weighted loss and adaptive activation function, resulting in the best estimate of T .

After thorough exploration of how to improve the PINNs model on a single variable, we investigate how our method performs on all variables excluding V . We first add only the adaptive estimation function with no weighted loss function, which performed the best on T alone (Figure 4.14). While the variable estimates all look better than previous methods, the derivative estimates exhibit oscillatory behavior in variables I and A , which were previously the variables that the model predicted the best.

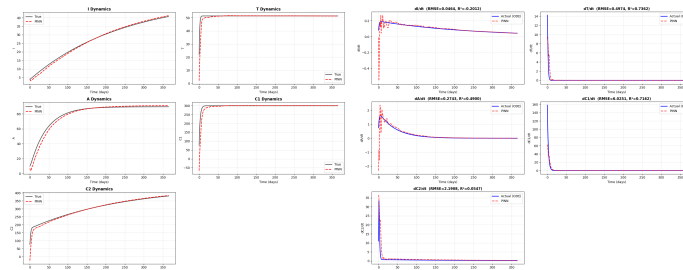


Figure 4.14: Estimate of variables excluding V using adaptive activation function.

Because the adaptive activation function alone did not perform well on all variables excluding V , we add the weighted loss function (Figure 4.15, Table 4.8). This appears to result in the

best estimates we have obtained so far, but the variables with the sharpest curves T, C_1, C_2 are predicted to have negative values, and the R^2 values are much worse than before.

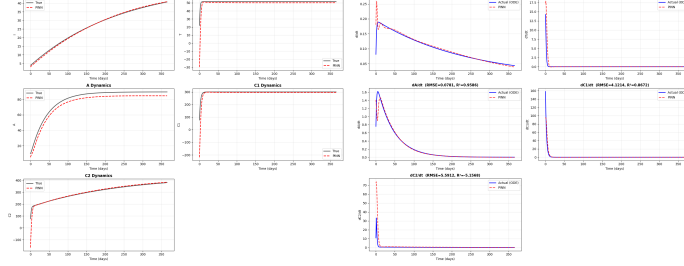


Figure 4.15: Estimate of variables excluding V using weighted loss with $a = 3$ and adaptive activation function.

Variable	Variable Error		Derivative Error		
	Rel. L2	MSE	MSE	RMSE	R^2
I	6.320×10^{-3}	1.844×10^{-5}	1.072×10^{-4}	0.0104	0.940
T	0.044	1.900×10^{-3}	1.085	1.042	-0.157
A	0.062	3.122×10^{-3}	6.100×10^{-3}	0.078	0.957
C_1	0.016	2.600×10^{-4}	16.99	4.121	0.867
C_2	0.016	1.542×10^{-4}	31.26	5.591	-5.157
Average	0.029	1.091×10^{-3}	9.868	2.169	-0.510

Table 4.8: Error metrics for

Using a loss function with $a = 2$ gives far better results (Figure 4.152). However, using $a = 1.5$ or $a = 2.5$ again results in negative predictions of T, C_1 , and C_2 . This exhibits the difficulty in fine-tuning the weighted loss, and the lack of stability in the model. As the model complexity increases, this stability will likely only worsen.

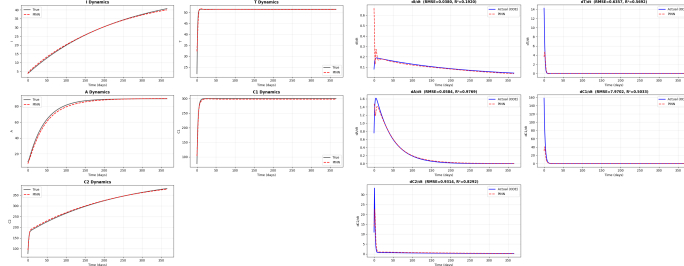


Figure 4.16: Estimate of variables excluding V using weighted loss with $a = 2$ and adaptive activation function.

Variable	Variable Error		Derivative Error		
	Rel. L2	MSE	MSE	RMSE	R^2
I	0.025	2.765×10^{-4}	1.447×10^{-3}	0.038	0.192
T	0.032	9.956×10^{-4}	0.404	0.636	0.569
A	0.015	1.878×10^{-4}	3.414×10^{-3}	0.058	0.977
C_1	0.037	1.358×10^{-3}	63.52	7.972	0.503
C_2	0.028	4.144×10^{-4}	0.867	0.931	0.829
Average	0.027	6.4646×10^{-4}	12.96	1.927	0.644

Table 4.9: Error metrics for combined improvements to method, including weighted loss function and adaptive activation function. Notice that although the results are significantly improved, the R^2 value is still relatively low for many variables, with an average of 0.644

Although these additions have improved the model, they have also increased the complexity of the model, resulting in far more difficulty estimating the ideal hyperparameter choices. Hence, this method is not necessarily useful in realistic settings. Additionally, despite the improvements to the model, the inaccuracy of the derivative predictions still leads STRidge to fail regardless of whether physical and regression loss are included in the empirical loss. So overall, despite significant tampering of the model, this PINNs method does not work for our biological data.

4.7 Discussion and future directions

Our experiments reveal that the SR-PINN framework faces several challenges when applied to biological system data:

Neural network smoothness bias Standard feedforward neural networks with smooth activation functions exhibit a bias toward smooth approximations. This is well-suited for the smooth synthetic PDE data used in [18] but incompatible with the sharp dynamics characteristic of many biological systems, and the SR-PINN fails to accurately model our data.

Coupled derivative-coefficient estimation The ADO algorithm’s success depends on accurate derivatives from the neural network to identify coefficients via STRidge. Simultaneously, good coefficients should guide the network toward physically consistent (and thus more accurate) solutions. Our results suggest this coupling breaks down when the neural network cannot approximate the data well: poor derivatives lead to incorrect coefficients, which provide poor guidance for network training, creating a negative feedback loop.

Small dataset amplification Real biological data is typically limited in quantity and contains measurement noise. Small datasets exacerbate both the neural network’s fitting challenges and STRidge’s threshold sensitivity. The 366 daily measurements in our dataset is inadequate for the SR-PINN framework’s dual optimization problem. Notably, 366 daily measurements are practically impossible to obtain in real studies, and true patient data would be even more sparse in practice.

Despite overall negative results, several modifications showed promise:

- **Huber loss + biased sampling:** This combination resulted in a significant improvement to the original PINN model, with better estimates of early dynamics.
- **Data-loss-only training:** Training only the data loss resulted in improvements to the predictions of the variables, indicating an indirect correlation between STRidge and Adam that was introduced by the loss function defined in [18].

- **Variable exclusion:** Reducing problem dimensionality improved stability and convergence.
- **Weighted loss function and adaptive activation function:** The best results were produced when using a loss function which prioritized early time points and an activation function which was learnable. This enabled the model to better estimate the sharp curves of both the variables and their derivatives.

While we focused on the SR-PINN framework, several alternative approaches may perform better on stiff problems such as ours. For example, stiff Neural ODEs have been used to model stiff dynamical systems [74, 76]. This is a black-box NN method and does not output estimated governing equations. However, stiff neural ODEs have been shown to approximate stiff problems well, and may be a better solution to estimating biological data. Alternatively, a simpler method such as using SINDy or other models that don't utilize neural networks may be a better approach, as this would require far less fine-tuning and may be more adaptable.

Our investigation highlights important considerations for machine learning in scientific discovery:

Method suitability A method's success on benchmark problems does not guarantee transferability to real-world applications. The smooth, well-conditioned PDEs used to demonstrate the viability of SR-PINNs in [18] differ fundamentally from our biological system data. Researchers should carefully assess whether benchmark problem characteristics match their target application.

Interpretability vs. accuracy trade-off SR-PINNs aim to provide interpretable equations (via sparse regression) rather than just accurate predictions. This interpretability comes at a cost: the framework makes strong assumptions about equation structure that may not hold for all systems. For some applications, predictive accuracy may be more valuable than equation interpretability. In those cases, using black box NNs or methods that don't rely on neural networks at all may be more suitable for the problem.

Chapter 5

Conclusion

This thesis investigated the application of sparse regression physics-informed neural networks (SR-PINNs) to model the immune response in patients with Celiac Disease after eating gluten and while on a gluten free diet. Through a multi-stage process involving data collection, processing, synthetic data generation, ODE modeling, and SR-PINN implementation, we aimed to address the challenge of developing predictive models from sparse, noisy biological data. Our findings demonstrate the limitations of machine learning approaches for biological system identification.

One of the most significant challenges we encountered was the extreme scarcity of longitudinal patient data for Celiac Disease. As detailed in Chapter 2, the available datasets consisted of measurements with substantial missing data and inconsistent measurement frequencies. This data limitation reflects challenges in autoimmune disease research. Scheduling issues and the complexity of repeated invasive procedures (such as intestinal biopsies) severely limit data collection capabilities. Future studies could be designed more effectively by standardizing data collection timelines and using available patient samples more effectively by using a more thorough array of tests.

We attempted to overcome the challenge of scarce data in Chapter 2, where we imputed missing data in Dataset B, and in Chapter 3, where we used all three available datasets to try to generate a single dataframe for a patient measured over the course of 365 days. While multiple imputation was possible over a relatively small number of missing entries, it became increasingly variable when introducing even a small number of additional measurement days. More complex methods such as Recurrent Neural Networks did not succeed at all when applied to our data. Due to these challenges, we were unable to use real data in our development of the SR-PINNs model in Chapter 4.

Instead of using real data, we developed the ODE model discussed in Chapter 3. Building on existing models of autoimmune disorders and incorporating known relationships between immune markers in the Celiac Disease immune response, we constructed a system of ODEs which modeled the immune response in Celiac Disease patients both during exposure to gluten and after the introduction of a gluten free diet. Our model captured the stiff dynamics which are common in biological systems, and the ODEs developed were a realistic stress-test for what an SR-PINNs model could achieve.

The central contribution of this thesis was the investigation of the applicability of SR-PINNs models to biological ODE discovery. Our results, detailed in Chapter 4, demonstrate that while SR-PINNs have shown success on benchmark PDE problems with smooth dynamics and known stability properties, they face significant setbacks when applied to more realistic, stiff biological systems. The neural network model designed to estimate the data failed to properly fit the stiff dynamics of our variables, and the fundamental framework of the SR-PINN, which relies on accurate approximation, failed. Although we explored a variety of potential improvements to the SR-PINN model developed by [18], ultimately none of the suggested remedies resulted in a model that was good enough to proceed with.

5.1 Future directions

Many biological systems are described by stiff ODEs, especially those describing autoimmune conditions which can see a severe ramp-up in intensity after initial inflammation. As such, the development of methods which can accurately model stiff dynamics is important for modeling realistic data. Fortunately, there are several alternative approaches which may perform better than SR-PINNs for modeling Celiac Disease immune dynamics, such as stiff neural ODE solvers. For systems where derivative estimation from data is possible, e.g. through finite differences or splines, SINDy may provide a more direct path to equation discovery, and does not rely on neural network approximations. However, many of these methods rely on non-noisy, non-sparse data.

Our work on the data processing and model development for Celiac Disease revealed key limitations in how data is collected, and presents important findings for how it should be collected in the future:

- Frequent measurements during the first 2 weeks of changes in diet, when immune dynamics are the most rapid, should be prioritized. Long-term dynamics tend to smooth out, and therefore measurements need not be as frequent
- Complete panels of immunological markers rather than measurements of a limited number of markers is far more valuable.
- Measurement protocols across studies should be more standardized.
- Data consisting of inconsistent measurement regimes within the same study (for example using Cohorts with different timelines or measuring different immune markers at different times) is almost useless for any modeling applications. More consistency within studies would be valuable.

Bibliography

- [1] V. Kumar, M. Jarzabek-Chorzelska, J. Sulej, K. Karnewska, T. Farrell, and S. Jablonska. Celiac disease and immunoglobulin a deficiency: How effective are the serological methods of diagnosis? *Clinical and Diagnostic Laboratory Immunology*, 9:1295–1300, November 2002. 2
- [2] J. A. Tye-Din, H. J. Galipeau, and D. Agardh. Celiac disease: A review of current concepts in pathogenesis, prevention, and novel therapies. *Frontiers in Pediatrics*, 6, November 2018. 2
- [3] R. L. Wolf, B. Lebwohl, A. R. Lee, P. Zybert, N. R. Reilly, J. Cadenhead, C. Amengual, and P. H. R. Green. Hypervigilance to a gluten-free diet and decreased quality of life in teenagers and adults with Celiac Disease. *Digestive Diseases and Sciences*, 63:1438–1448, June 2018. 2
- [4] National Organization for Rare Disorders. Refractory celiac disease - symptoms, causes, treatment. <https://rarediseases.org/rare-diseases/refractory-celiac-disease/>, 2025. [Online; accessed 03-March-2026]. 2
- [5] P. H. R. Green, B. Lebwohl, and R. Greywoode. Celiac disease. *Clinical reviews in allergy and immunology*, pages 1099–1106, May 2016. 2
- [6] V. Abadie and B. Jabri. IL-15: A central regulator of celiac disease immunopathology, 2014. 3, 7, 93, 94
- [7] J. Voisine and V. Abadie. Interplay Between gluten, HLA, innate and adaptive immunity orchestrates the development of Coeliac Disease. *Frontiers in Immunology*, 12, June 2021. 3, 93
- [8] N. Korneychuk, E. Ramiro-Puig, J. Ettersperger, J. Schulthess, N. Montcuquet, H. Kiyono, B. Meresse, and N. Cerf-Bensussan. Interleukin 15 and CD4+ T-cells cooperate to promote small intestinal enteropathy in response to dietary antigen. *Gastroenterology*, 146:1017–1027, 2014. 3
- [9] M. L. Lähdeaho, M. Mäki, K. Laurila, H. Huhtala, and K. Kaukinen. Small- bowel mucosal changes and antibody responses after low- and moderate-dose gluten challenge in celiac disease. *BMC Gastroenterology*, 11, November 2011. 3, 4, 5, 10, 59
- [10] M. Sarra, M. L. Cupi, I. Monteleone, E. Franzè, G. Ronchetti, A. Di Sabatino, P. Gentileschi, L. Franceschilli, P. Sileri, G. Sica, G. Del Vecchio Blanco, M. Cretella, O. A. Paoluzi, G. R. Corazza, F. Pallone, and G. Monteleone. IL-15 positively regulates IL-21 production in Celiac Disease mucosa. *Mucosal Immunology*, 6:244–255, July 2012. 3, 59, 93
- [11] O. G. Moscatelli, A. K. Russell, L. M. Henneken, L. Fothergill, A. Motyer, H. Reid, J. Rossjohn, V. Bryant, R. P. Anderson, M. Y. Hardy, and J. A. Tye-Din. Blood-based t-cell diagnosis of celiac disease. *Gastroenterology*, November 2025. 3, 93
- [12] A. Levescot, G. Malamut, and N. Cerf-Bensussan. Immunopathogenesis and environmental triggers in Coeliac Disease, July 2022. 3
- [13] I. Caputo, M. V. Barone, S. Martucciello, M. Lepretti, and C. Esposito. Tissue transglutaminase in celiac disease: Role of autoantibodies, April 2009. 3, 93, 94
- [14] I. S. Stafford, M. Kellermann, E. Mossotto, R. M. Beattie, B. D. MacArthur, and S. Ennis. A systematic review of the applications of artificial intelligence and machine learning in autoimmune diseases, December 2020. 3, 4

- [15] Y. Ugolkov, A. Nikitich, C. Leon, G. Helmlinger, K. Peskov, V. Sokolov, and A. Volkova. Mathematical modeling in autoimmune diseases: from theory to clinical application, 2024. 4
- [16] A. Arazi and A. U. Neumann. Modeling immune complex-mediated autoimmune inflammation. *Journal of Theoretical Biology*, 267:426–436, December 2010. 4, 58
- [17] S. Iwami, Y. Takeuchi, Y. Miura, T. Sasaki, and T. Kajiwara. Dynamical properties of autoimmune disease models: Tolerance, flare-up, dormancy. *Journal of Theoretical Biology*, 246:646–659, June 2007. 4, 58
- [18] Z. Chen, Y. Liu, and H. Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12, December 2021. 4, 6, 8, 66, 67, 69, 70, 71, 72, 73, 83, 84, 85
- [19] W. Dieterich, T. Ehnis, M. Bauer, P. Donner, U. Volta, E. O. Riecken, and D. Schuppan. Identification of tissue transglutaminase as the autoantigen of Celiac Disease. *Nat Med*, 1997. 4
- [20] L. F. Risnes, H. M. Reims, R. M. Doyle, S. W. Qiao, L. M. Sollid, K. E.A. Lundin, and A. Christophersen. Gluten-free diet induces rapid changes in phenotype and survival properties of gluten-specific t cells in celiac disease. *Gastroenterology*, 167:250–263, July 2024. 4, 5, 15, 18, 20, 21, 24, 59
- [21] S. Gómez-Aguillilla, S. Farrais, N. López-Palacios, B. Arau, C. Senosiain, M. Corzo, N. Fernandez-Jimenez, Á. Ruiz-Carnicer, F. Fernández-Bañares, B. P. González-García, E. Tristán, A. Montero-Calle, M. Garranzo-Asensio, I. Casado, M. Pujals, J. M. Hernández, J. Infante-Menéndez, G. Roy, C. Sousa, and C. Núñez. Diagnosis of celiac disease on a gluten-free diet: a multicenter prospective quasi-experimental clinical study. *BMC Medicine*, 23, December 2025. 4, 6, 10
- [22] A. Ensari and M. N. Marsh. Diagnosing celiac disease: A critical overview, 2019. 4
- [23] P. Petersen and J. Zech. Mathematical theory of deep learning. 2026. 7
- [24] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. 8, 67
- [25] S. W. Hawes, A. K. Littlefield, D. A. Lopez, K. J. Sher, E. L. Thompson, R. Gonzalez, L. Aguinaldo, A. R. Adams, M. Bayat, A. L. Byrd, L. F. S. Castro de Araujo, A. Dick, S. F. Heeringa, C. M. Kaiver, S. M. Lehman, L. Li, J. Linkersdörfer, T. J. Maullin-Sapey, M. C. Neale, T. E. Nichols, S. Perlstein, S. F. Tapert, C. E. Vize, M. Wagner, R. Waller, and W. K. Thompson. Longitudinal analysis of the ABCD® study. *Developmental Cognitive Neuroscience*, 72:101518, 2025. 10
- [26] A. N. Baraldi and C. K. Enders. An introduction to modern missing data analyses. *Journal of School Psychology*, 48:5–37, February 2010. 27
- [27] J. S. Granberg-Rademacker. A comparison of three approaches to handling incomplete state-level data. *State Politics & Policy Quarterly*, 7(3):325–338, 2007. 27, 30
- [28] A. M. Gad. Imputation methods for longitudinal data: A comparative study. *International Journal of Statistical Distributions and Applications*, 3:72, 2017. 27
- [29] S. van Buuren and K. Groothuis-Oudshoorn. MICE: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45, December 2011. 27, 31, 32
- [30] M. Quartagno, S. Grund, and J. Carpenter. jomo: A flexible package for two-level joint modelling multiple imputation. *The R Journal*, 9, January 2019. 27

- [31] S. van Buuren. *Flexible imputation of missing data*. Chapman & Hall/CRC, 2 edition, 2018. 28, 31, 32
- [32] P. D. Hoff. *A First Course in Bayesian Statistical Methods*. Springer, 2009. 28, 29
- [33] D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. 28
- [34] R. Wijesuriya, M. Moreno-Betancur, J. B. Carlin, I. R. White, M. Quartagno, and K. J. Lee. Multiple imputation for longitudinal data: A tutorial. *Statistics in Medicine*, 44, February 2025. 29, 32
- [35] P. C. Austin, I. R. White, D. S. Lee, and S. van Buuren. Missing data in clinical research: A tutorial on multiple imputation, September 2021. 29
- [36] G. Casella and E. I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992. 29, 30
- [37] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf. Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20:40–49, March 2011. 30
- [38] J. W. Graham, A. E. Olchowski, and T. D. Gilreath. How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention Science*, 8:206–213, September 2007. 31
- [39] D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons Inc., New York., 1987. 31
- [40] E. A. Stuart, M. Azur, C. Frangakis, and P. Leaf. Multiple imputation with large data sets: A case study of the children’s mental health initiative. *American Journal of Epidemiology*, 169:1133–1139, May 2009. 32
- [41] K. Abayomi, A. Gelman, and M. Levy. Diagnostics for multivariate imputations. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, 57:273–291, June 2008. 32
- [42] K. P. Junaid, T. Kiran, M. Gupta, K. Kishore, and S. Siwatch. How much missing data is too much to impute for longitudinal health indicators? A preliminary guideline for the choice of the extent of missing proportion to impute with multiple imputation by chained equations. *Population Health Metrics*, 23, December 2025. 32
- [43] J. Lin, D. Wu, and L. Huang. Self-supervised bi-directional mapping generative adversarial network for arbitrary-time longitudinal interpolation of missing data. *Biomedical Signal Processing and Control*, 105, July 2025. 53
- [44] J. Yoon, W. R. Zame, and M. van der Schaar. Estimating Missing Data in temporal data streams Using Multi-Directional Recurrent Neural Networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2019. 54
- [45] S. Boeder. M-RNN: Estimate missing values in time series. <https://www.kaggle.com/code/markwallbang/m-rnn-estimate-missing-values-in-time-series>, January 2021. [Online; accessed 25-November-2025]. 54
- [46] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 55
- [47] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, 2015. 55

- [48] Y. Kai. Intestinal villus structure contributes to even shedding of epithelial cells. *Biophysical Journal*, 120:699–710, February 2021. 57, 60
- [49] C. S. Potten and M. Loeffler. Stem cells: attributes, cycles, spirals, pitfalls and uncertainties lessons for and from the crypt. *Development*, 110:1001–1020, 1990. 57
- [50] R. Przemioslo, N. A. Wright, G. Elia, P. J. Ciclitira, St. Thomas’s Hospital, and London R. Przemioslo. Analysis of crypt cell proliferation in Coeliac Disease using MI-B1 antibody shows an increase in growth fraction. *Gut*, 36:22–27, 1995. 57, 60, 94
- [51] O. O. Demin, S. V. Smirnov, V. V. Sokolov, L. Cucurull-Sanchez, C. Pichardo-Almarza, M. V. Flores, N. Benson, and O. V. Demin. Modeling of Celiac Disease immune response and the therapeutic effect of potential drugs. *BMC Systems Biology*, 7, July 2013. 58
- [52] C. García-Hoz, L. Crespo, R. Pariente, A. De Andrés, R. Rodríguez-Ramos, and G. Roy. Intraepithelial Lymphogram in the diagnosis of Celiac Disease in adult patients: A validation cohort. *Nutrients*, 16, April 2024. 59
- [53] J. M. Kvamme, S. Sørbye, J. Florholmen, and T. S. Halstensen. Population-based screening for Celiac Disease reveals that the majority of patients are undiagnosed and improve on a gluten-free diet. *Scientific Reports*, 12, December 2022. 59
- [54] A. Di Sabatino, R. Ciccocioppo, F. Cupelli, B. Cinque, D. Millimaggi, M. M. Clarkson, M. Paulli, M. G. Cifone, and G. R. Corazza. Epithelium derived Interleukin 15 regulates intraepithelial lymphocyte Th1 cytokine production, cytotoxicity, and survival in Coeliac Disease. *Gut*, 55:469–477, April 2006. 59
- [55] F. Masaebi, M. A. Looha, M. Rostami-Nejad, M. A. Pourhoseingholi, N. Mohseni, G. Samasca, I. Lupan, M. Rezaei-Tavirani, and M. Reza Zali. The predictive value of serum cytokines for distinguishing Celiac Disease from non-Celiac gluten sensitivity and healthy subjects. *Iranian Biomedical Journal*, 24:335–341, 2020. 59
- [56] M. L. Lähdeaho, K. Kaukinen, P. Collin, T. Ruuska, J. Partanen, A. M. Haapala, and M. Mäki. Celiac disease: From inflammation to atrophy: A long-term follow-up study. *Journal of Pediatric Gastroenterology and Nutrition*, 41:44–48, July 2005. 59
- [57] M. M. Vargas, R. A. Neto, and V. L. Sdepanian. Quantitative histology as a diagnostic tool for Celiac Disease in children and adolescents. *Annals of Diagnostic Pathology*, 61, December 2022. 59
- [58] Baseline quantitative histology in therapeutics trials reveals villus atrophy in most patients with Coeliac Disease who appear well controlled on gluten-free diet. *GastroHep*, 2:22–30, January 2020. 59
- [59] A. J. Farchione, H. Cheon, D. Vremec, J. Neumann, G. M. Verstappen, M. Y. Hardy, M. B. Margetts, L. J. Howson, L. M. Henneken, M. Forde, J. A. Tye-Din, S. Heinzl, P. D. Hodgkin, and V. L. Bryant. Functional immune profiling reveals CD4+ T-cell dysregulation associated with Coeliac Disease, June 2025. 59, 60
- [60] R. A. Klaasen, D. J. Warren, R. Iversen, N. Bolstad, I. L. Andersen, P. Mjønes, E. Rønne, K. E. A. Lundin, L. M. Sollid, and E. Ness-Jensen. The development and validation of a high-capacity serological assay for Celiac Disease. *Clinical Biochemistry*, 107:13–18, September 2022. 60
- [61] K. C. Conlon, E. Lugli, H. C. Welles, S. A. Rosenberg, A. T. Fojo, J. C. Morris, T. A. Fleisher, S. P. Dubois, L. P. Perera, D. M. Stewart, C. K. Goldman, B. R. Bryant, J. M. Decker, J. Chen, T. A. Worthy, W. D. Figg, C. J. Peer, M. C. Sneller, H. C. Lane, J. L. Yovandich, S. P. Creekmore, M. Roederer, and T. A. Waldmann. Redistribution, hyperproliferation,

- activation of natural killer cells and CD8 T-cells, and cytokine production during first-in-human clinical trial of recombinant human Interleukin-15 in patients with cancer. *Journal of Clinical Oncology*, 33:74–82, January 2015. 60
- [62] S. Bhatia, B. Curti, M. S. Ernstoff, M. Gordon, E. I. Heath, W. H. Miller, I. Puzanov, D. I. Quinn, T. W. Flaig, P. VanVeldhuizen, K. Byrnes-Blake, J. A. Freeman, R. Bittner, N. Hunder, S. Souza, and J. A. Thompson. Recombinant interleukin-21 plus sorafenib for metastatic renal cell carcinoma: A phase 1/2 study. *Journal for ImmunoTherapy of Cancer*, 2, January 2014. 60
- [63] A. Di Sabatino, S. D’alo Á, D. Millimaggi, R. Ciccocioppo, R. Parroni, G. Sciarra, M. G. Cifone, and G. R. Corazza. Apoptosis and peripheral blood lymphocyte depletion in Coeliac Disease. *Immunology*, 103:435–440, March 2001. 60
- [64] A. Di Sabatino, R. Ciccocioppo, S. D’alò, R. Parroni, D. Millimaggi, M. G. Cifone, and G. R. Corazza. Intraepithelial and lamina propria lymphocytes show distinct patterns of apoptosis whereas both populations are active in fas based cytotoxicity in coeliac disease. *gut*, 49:380–386, 2001. 60
- [65] K. Kaheman, J. N. Kutz, and S. L. Brunton. SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics: SINDy-PI. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476, October 2020. 66, 72
- [66] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113:3932–3937, 2016. 67
- [67] T. T. Nguyen, C. Soussen, J. Idier, and E. H. Djermoune. NP-hardness of ℓ_0 minimization problems: revision and extension to the non-negative setting NP-hardness of 0 minimization problems: revision and extension to the non-negative setting. *13th International Conference on Sampling Theory and Applications, SampTA 2019, Bordeaux, France*, July 2019. 68
- [68] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017. 70
- [69] Nadam–TensorFlow v2.16.1. https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Nadam. [Online; accessed 26 Janary 2026]. 71
- [70] D. P. Kingma and J. Lei Ba. Adam: A method for stochastic optimization. January 2017. 71
- [71] T. Dozat. Incorporating nesterov momentum into adam. pages 1–4, 2016. 71
- [72] Minimize method L-BFGS-B. <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html>. [Online; accessed 26 Janary 2026]. 71
- [73] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. 71
- [74] S. Kim, W. Ji, S. Deng, Y. Ma, and C. Rackauckas. Stiff Neural Ordinary Differential Equations. September 2021. 72, 80, 84
- [75] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 2:52–63, June 2016. 72
- [76] C. Fronk and L. Petzold. Training stiff neural ordinary differential equations with implicit single-step methods. October 2024. 84

-
- [77] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 4 edition, 2002. 93, 94
- [78] Lumen Learning. Antigen-Presenting Cells. <https://courses.lumenlearning.com/wm-biology2/chapter/antigen-presenting-cells/>. [Online; accessed 28-April-2025]. 93
- [79] Cleveland Clinic. Cytokines. <https://my.clevelandclinic.org/health/body/24585-cytokines>. [Online; accessed 04-March-2026]. 93
- [80] H. Gehart and H. Clevers. Tales from the crypt: new insights into intestinal stem cells, January 2019. 94
- [81] Cleveland Clinic. T-Cells. <https://my.clevelandclinic.org/health/body/24630-t-cells>. [Online; accessed 28-April-2025]. 94
- [82] National Cancer Institute. Definition of intestinal villi. <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/intestinal-villi>. [Online; accessed 04-March-2026]. 94
- [83] M. K. Kowalski, D. Domżał-Magrowska, P. Szcześniak, M. Bulska, D. Orszulak-Michalak, and E. Małecka-Wojcieszko. Gluten-free diet adherence evaluation in adults with long-standing Celiac Disease. *Foods*, 14, January 2025. 94

Appendix

A.1 Glossary

adaptive immune response Immune response called upon the innate immune response to eliminate invading pathogens and toxic molecules produced by them. Adaptive immune responses are carried out by white blood cells called lymphocytes. Responses may involve the secretion of antibodies called immunoglobulins, such as IgA and IgG. They may also involve the activation of T-cells to attack the pathogen directly. Importantly, the adaptive immune response is able to distinguish between foreign and native bodies, and avoids attacking native bodies except for in the case of an autoimmune disorder such as Celiac Disease [77]. 2, 93, 94

antigen A molecule that can bind to an antibody such as IgA or IgG, or a T-cell receptor, and which may trigger an immune response. 2, 93

antigen-presenting cell (APC) A cell that detects and processes antigens into fragments that can be recognized by T-cells [78]. 2, 3, 57

cytokine Proteins which signal immune cells and regulate inflammation. May be produced by immune cells or epithelial cells [79]. Many cytokines are produced during the adaptive immune response of Celiac Disease, but major cytokines include IFN- γ , IL-15, and IL-21 [6]. 2, 57–59, 93

deamidation A chemical reaction which results in a change to the protein gliadin. The structure of the gliadin is changed and its ability to provoke an immune response is enhanced. 2, 57, 94

gliadin A protein primarily made up of amino acids glutamine and proline. Triggers immune response in Celiac Disease. 2, 57, 93, 94

IFN- γ Cytokine produced in conjunction with IL-21 by T-cells in Celiac Disease which promotes inflammation. The roles of IFN- γ and IL-21 are similar and often overlap [7]. 3, 7, 57, 58, 93

IL-15 Cytokine which promotes natural killer and T-cell activation. Additionally triggers lymphocytes, and is crucial for their proliferation and survival. Can be upregulated on the surface of any cell. Is typically upregulated during inflammation, and dysregulated increase in IL-15 expression is observed in many inflammatory autoimmune diseases including Celiac Disease [6]. 3, 7, 57, 58, 93

IL-2 Cytokine produced by lymphocytes (T-cells) at high rates during Celiac Disease reactions [11]. 3, 6

IL-21 Cytokine produced by lymphocytes (T-cells and IELs) at high rates during Celiac Disease reactions [10]. 3, 7, 57, 58, 93

Immunoglobulin A (IgA) Antibody which is produced during adaptive immune response to target foreign invaders. During Celiac Disease reaction, may be mobilized against gliadin, resulting in IgA-DGP, or against TG2, resulting in IgA-TG2. Presence of IgA-DGP and IgA-TG2 is a hallmark of Celiac Disease [13]. 2, 3, 5, 7, 10, 11, 15, 17, 18, 33, 36–52, 56–58, 93

- Immunoglobulin G (IgG)** Antibody which is produced during adaptive immune response to target foreign invaders. During Celiac Disease reaction, may be mobilized against gliadin, resulting in IgG-DGP, or against TG2, resulting in IgG-TG2. Presence of IgG-DGP and IgG-TG2 is a hallmark of Celiac Disease [13]. 2, 3, 5, 7, 15, 18–21, 33, 36–52, 57, 58, 93, 94
- innate immune response** Immune response which consists of pre-existing immune defenses which are not pathogen-specific. Examples include skin, mucosa, and other antimicrobial substances within the body. May call upon the adaptive immune response if it recognizes pathogens. Can distinguish between different pathogens and trigger the corresponding adaptive immune response to most effectively eliminate them [77]. 2, 57, 93
- intestinal crypt** Small glands located between villi in the small intestine in which stem cells are produced [80]. May be damaged during Celiac Disease immune response and recede deeper into intestinal walls [50]. 3, 4, 57, 59
- intraepithelial lymphocyte (IEL)** Specialized T-cell found in the intestinal walls. Disregulation of IELs is a hallmark of Celiac Disease, causing them to target intestinal cells, damaging the small intestine and villi [6]. 2, 3, 5–7, 10–12, 15, 22–24, 26, 27, 34, 35, 37, 38, 40–46, 48–52, 54, 57, 58, 93
- T-cell** Type of white blood cell (lymphocyte) that destroys pathogens and sends signals to immune system to help organize attacks against pathogens [81]. 2, 3, 6, 15, 21–26, 34, 35, 37, 38, 40–46, 48–52, 57, 59, 93, 94
- tissue transglutaminase (TG2)** Enzyme which causes gliadin to undergo deamidation in Celiac Disease. Is targeted in subsequent immune reaction. 2, 3, 5, 10, 11, 15, 17, 18, 33, 36–52, 56–58, 93, 94
- villi** Hair-like projections which line the walls of the small intestine, increasing surface area to improve absorption of nutrients [82]. May be damaged during Celiac Disease immune response and become shorter or, in severe cases, be removed entirely [83]. 2–4, 57, 59, 94

B.2 Code and data availability

The code and data used for the SR-PINNs model is available in <https://github.com/kreese8/SR-PINNs> upon request for access.