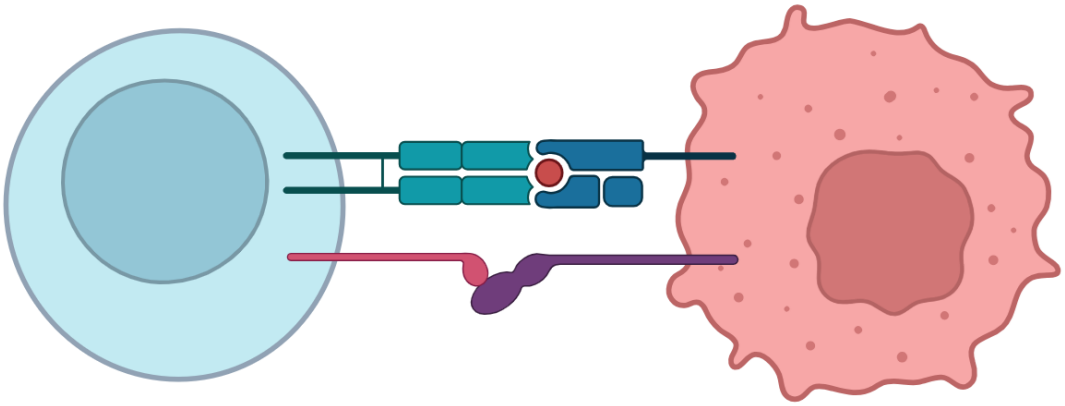


Tensor decomposition for time-resolved immune cell sequencing in cancer

From raw $\gamma\delta$ TCR sequences to interpretable patient-level immune patterns

Master's thesis in Mathematics

PABLO VADILLO BERGANZA
VIOLANT MORENO CREIXELL

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Tensor decomposition for time-resolved immune cell sequencing in cancer

From raw $\gamma\delta$ TCR sequences to interpretable patient-level immune patterns

PABLO VADILLO BERGANZA and
VIOLANT MORENO CREIXELL

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Computing Disease Evolution in Cancer (CODEc) group
Chalmers University of Technology
Gothenburg, Sweden 2025

Tensor decomposition for time-resolved immune cell sequencing in cancer
From raw $\gamma\delta$ TCR sequences to interpretable patient-level immune patterns
PABLO VADILLO BERGANZA and
VIOLANT MORENO CREIXELL

Supervisor: Eszter Lakatos, Department of Mathematical Sciences, Chalmers
Examiner: Marina Axelson-Fisk, Department of Mathematical Sciences, Chalmers

Master's Thesis 2025

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Computing Disease Evolution in Cancer (CODEc) group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A T cell (left) interacting with a cancer cell (right) by means of membrane receptors. Created in `https://BioRender.com`.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Tensor decomposition for time-resolved immune cell sequencing in cancer
From raw $\gamma\delta$ TCR sequences to interpretable patient-level immune patterns
PABLO VADILLO BERGANZA and
VIOLANT MORENO CREIXELL
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

The immune system plays a crucial role in the detection and elimination of cancer cells, with T cell receptors (TCRs) enabling antigen recognition. $\gamma\delta$ T cells are a less common and relatively understudied subset of T cells compared to their $\alpha\beta$ counterparts. Although they have been shown to play important roles in cancer immunity, especially due to their ability to act independently of MHC (Major Histocompatibility Complex) presentation, much less is known about their behavior over time. This makes them a promising but challenging target for immune repertoire analysis.

This thesis applies different tensor decomposition methods to time-resolved $\gamma\delta$ TCR sequencing data from sarcoma patients to uncover interpretable immune dynamics. After preprocessing, data from 13 of 16 patients were retained, and multiple tensors of varying temporal lengths (70, 100, 200, and 300 days) were constructed to balance patient availability with time resolution. Three decomposition models (CP, Tucker, and PARAFAC2) were applied to both simulated and real data to evaluate their ability to capture latent patterns across sequences, time, and patients. Furthermore, a clustering pipeline was applied to extract the patient outcomes and was compared to ground-truth data.

Our results with the simulated data validate that tensor decomposition can be an effective tool for finding relevant patterns and subgroups in the data. In particular, by clustering the patient factor matrices of the Tucker decomposition, we observed groupings that showed partial agreement with known clinical labels, suggesting that the model captures some meaningful variation in immune response. In addition to unsupervised pattern discovery, we evaluated classical immunological metrics, including richness, evenness, and clonality, over time to further characterize immune repertoire dynamics. Therefore, this work confirms that tensor decomposition can extract informative, low-dimensional representations from complex immune repertoire data and may support future efforts in stratifying patients or monitoring treatment responses.

Keywords: tensor decomposition, $\gamma\delta$ T cells, TCR sequencing, sarcoma, immune repertoire, CP decomposition, Tucker decomposition, PARAFAC2, K-medoids.

Acknowledgements

We would like to express our gratitude to our supervisor, Eszter Lakatos, for her invaluable guidance, encouragement, and constructive feedback throughout this thesis.

We also want to thank our colleagues in our Computing Disease Evolution in Cancer (CODEc) group meetings for their insight, which helped shape this project.

In addition, we extend our gratitude to our examiner, Marina Axelson-Fisk, for her time and her valuable contribution.

We would also like to thank Sahlgrenska University Hospital for providing the dataset and answering our questions when needed.

Lastly, we are deeply thankful to our friends and families for their continued support.

Pablo Vadillo Berganza and Violant Moreno Creixell, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

ALS	Alternating Least Squares
CDR3	Complementarity-Determining Region 3
CORCONDIA	Core Consistency Diagnostic
CP	CANDECOMP/PARAFAC
DE_{50}	Diversity Evenness score
HOOI	Higher-Order Orthogonal Iteration
HOSVD	Higher-Order SVD
MHC	Major Histocompatibility Complex
NKT	Natural Killer T Cells
NMI	Normalized Mutual Information
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
TCR	T Cell Receptor
WCSD	Within-Cluster Sum of Distances

Nomenclature

Below is the nomenclature of arrays, indices and elements that have been used throughout this thesis.

Arrays

$\mathcal{X}$	Third or higher-order tensor
$\mathbf{X}$	Second-order tensor or Matrix
$\mathbf{x}$	First-order tensor or Vector

Indices

i,j,k,i_1,...,i_n	Indices for tensor modes
I,J,K,I_1,...,I_N	Indices for dimension spaces

Elements

x_{ijk}	Element of array located in (i,j,k)

Contents

List of Acronyms **ix**

Nomenclature **xi**

List of Figures **xv**

List of Tables **xix**

1 Introduction **1**
1.1 Background and Related Work 1
1.2 Aim 2
1.3 Structure 2

2 Theory **3**
2.1 Sarcoma 3
2.1.1 Description, Epidemiology, and Risk Factors 3
2.1.2 Treatment Strategies 4
2.2 T Cells 6
2.2.1 Role of $\gamma\delta$ T Cells in Tumor Surveillance 6
2.2.2 T Cell Receptor Variability 7
2.3 DNA Immune Repertoire Sequencing 8
2.3.1 Sequencing Process 8
2.3.2 Common Immunology Metrics 8
2.4 Tensors and Preliminaries 11
2.4.1 Tensor Operations 12
2.5 Tensor Decomposition 14
2.5.1 Singular Value Decomposition (SVD) 14
2.5.2 Principal Component Analysis (PCA) 15
2.5.3 CANDECOMP/PARAFAC (CP) Decomposition 16
2.5.4 Tucker Decomposition 18
2.5.5 PARAFAC2 20
2.5.6 Number of Components 21

3 Methods **25**
3.1 Dataset Description 25
3.2 Preprocessing and Tensor Construction 26
3.3 Simulated data 28

3.4 Decomposition models . . . 31
3.5 Selection of number of components . . . 32
3.6 Clustering with Tucker Decomposition . . . 33
3.7 Immunology metrics . . . 34

4 Results **37**
4.1 Tensor construction . . . 37
4.2 CP Decomposition . . . 38
4.2.1 Selection of Number of Components . . . 38
4.2.2 Decompositions . . . 41
4.3 Tucker Decomposition . . . 44
4.3.1 Selection of Number of Components . . . 44
4.3.2 Decompositions . . . 50
4.4 PARAFAC2 . . . 58
4.4.1 Selection of Number of Components . . . 58
4.4.2 Decompositions . . . 59
4.5 Immunology Metrics . . . 61
4.6 Clustering and Ground-Truth Validation . . . 64

5 Discussion **69**
5.1 Validation of Methods . . . 69
5.2 Interpretation of Results from Real Data . . . 70
5.3 Limitations and Future Work . . . 72

6 Conclusion **75**

Bibliography **77**

A Overview of GitHub repository location of additional figures **I**

B Other data related plots **III**
B.1 Patient data kept for each tensor . . . III
B.2 Tensor data . . . VI
B.3 Ground-truth labels over time . . . XI

C Tensor Decomposition Results **XIII**
C.1 CP Decompositions . . . XIII
C.2 Tucker Decompositions . . . XIII

D Clustering **XV**

List of Figures

2.1 A vector, a matrix and a third-order tensor. 11
2.2 Mode-1, mode-2 and mode-3 fibers. 11
2.3 Horizontal, lateral and frontal slices. 12
2.4 The mode-1 matricization of a tensor $\mathcal{X}$. 12
2.5 Rank-one tensor. 13
2.6 The SVD of a matrix $\mathbf{A}$. 15
2.7 PCA: A matrix decomposed into the factors score and loading matrix. 15
2.8 CP Decomposition of a third-order tensor. 16
2.9 Tucker Decomposition of a third-order tensor. 18
2.10 PARAFAC2 Decomposition of a set of matrices. 20

3.1 Examples of TCR sequence temporal trends of patients from Groups A, B and C in *tensor_1*. 30
3.2 Examples of TCR sequence temporal trends of patients from Groups A, B and C in *tensor_2*. 30
3.3 Examples of TCR sequence temporal trends of patients from Groups A, B and C in *tensor_3* and *tensor_4*. 31

4.1 Sampling data-points for all patients. Those excluded are indicated but shown nonetheless for whole data comparison. 38
4.2 Sequence temporal trends retained in each tensor and in whole temporal data from patient *SARK012*, which has more temporal data than kept in the tensors, hence showing the information trade-off. On the rightmost plot, the time splits are marked in vertical dashed lines for days 70, 100, 200, and 300. 38
4.3 Plots of the reconstruction errors for $R = 1, 2, 3, 4, 5, 6, 7$ for the simulated tensors. The dotted red line represents the elbow. 39
4.4 Plots of the CONCORDIA values for $R = 1, 2, 3, 4, 5, 6, 7$ for the simulated tensors. 39
4.5 Plots of the reconstruction errors for $R = 1, 2, 3, 4, 5, 6, 7$ for all tensors. The dotted red line represents the elbow. 40
4.6 Plots of the CONCORDIA values for $R = 1, 2, 3, 4, 5, 6, 7$ for the real tensors. 40
4.7 CP decomposition for *tensor_1* with $R = 3$. 41
4.8 CP decomposition for *tensor_70 with* $R = 2$. 42
4.9 CP decomposition for *tensor_100 with* $R = 2$. 42
4.10 CP decomposition for *tensor_200 with* $R = 2$. 43

4.11 CP decomposition for *tensor_300 with* $R = 3$. 43
4.12 Number of components per threshold using the SVD method in the simulated data. 46
4.13 Singular values for *tensor_1* (top) and *tensor_2* (bottom) with a 0.30 threshold. Yellow bars indicate retained values, suggesting component counts of (3, 2, 3) and (2, 2, 3), respectively. 46
4.14 Number of components per threshold using the SVD method. 49
4.15 Reconstruction error (top) and singular values (bottom) for *tensor_70* with a 0.2 threshold. Yellow bars indicate retained components. . . . 49
4.16 Tucker decomposition results for *tensor_1* with components (3, 2, 3). 51
4.17 Tucker decomposition results for *tensor_2* with components (2, 2, 3). 52
4.18 Tucker decomposition results for *tensor_70* with components (2, 2, 3). 54
4.19 Tucker decomposition results for *tensor_100* with components (2, 2, 2). 55
4.20 Tucker decomposition results for *tensor_200* with components (2, 2, 2). 56
4.21 Tucker decomposition results for *tensor_300* with components (3, 3, 3). 57
4.22 Plots of the reconstruction errors for $R = 1, 2, 3, 4, 5, 6, 7$ for the simulated PARAFAC2 arrays. The dotted red line represents the elbow. 58
4.23 Plots of the CONCORDIA values for $R = 1, 2, 3, 4, 5, 6$ for the simulated PARAFAC2 arrays. 58
4.24 Plots of the reconstruction errors and CORCONDIA values for $R = 1, 2, 3, 4, 5, 6$ for the real PARAFAC2 array. 59
4.25 PARAFAC2 decomposition for *tensor_3* with $R = 3$. The dotted red line represents a threshold for which the projection of the time dimension is plotted. 59
4.26 PARAFAC2 decomposition for *tensor_4* with $R = 2$. The dotted red line represents a threshold for which the projection of the time dimension is plotted. 60
4.27 PARAFAC2 decomposition for the real data with $R = 3$. The dotted red line represents a threshold for which the projection of the time dimension is plotted. 61
4.28 Temporal evolution of the studied immunology metrics. 63
4.29 Change in the metrics between the first point (before treatment) and the last point (after treatment). For metrics bounded between 0 and 1 (right), the difference is shown in absolute values. For the rest (left), in percentage. 64
4.30 WCSD for increasing number of clusters k for both simulated tensors. 65
4.31 Pairplot visualizations of K-medoids clustering results on patient factor matrices for the two simulated tensors. The figures show the clustering structure for a pair of exemplary features for *tensor_1*, and *tensor_2*, with patient names overlaid and $k = 3$. 65
4.32 WCSD for increasing number of clusters k for all tensors. The cross represents the elbow point of each curve. 66

4.33 Pairplot visualizations of K-medoids clustering results on patient factor matrices for the four tensors. The figures show the clustering structure across a pair of exemplary components for *tensor_70*, *tensor_100*, *tensor_200*, and *tensor_300*, with patient names overlaid and $k = 4$. 67
4.34 Clustering results from using immunology metrics. 67

B.1 Patients with data under 100 but over 70 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor. III
B.2 Patients with data under 200 but over 100 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor. . . . IV
B.3 Patients with data under 300 but over 200 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor. . . . IV
B.4 Patients with data over 300 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor. V
B.5 *Tensor_70* time series VI
B.6 *Tensor_100* time series VII
B.7 *Tensor_200* time series VIII
B.8 *Tensor_300* time series IX
B.9 Clinical ground-truth labels evolution over time for each patient (except excluded ones). XI

C.1 CP decomposition for *tensor_2* with $R = 3$. XIII

D.1 NMI comparison between clustering methods and ground-truth with $k = 3$. XV
D.2 NMI score comparison between both patient clustering (Tucker vs. TCR metrics) across all tensors for increasing value of k. XVI

List of Tables

2.1 Most common sarcoma subtypes. Adapted from [15]. *Not included in [15] but present in our dataset; information from [16]. 5
2.2 Summary of commonly used immunology metrics, typical range and interpretation. R stands for richness, the unique TCR sequences of the sample. 11

3.1 Dataset description summary. For each patient, their sarcoma subtype, their total number of sequences throughout all the samples, the number of days passed from the first sequencing to the last and the total number of samples (or data points) are specified. The ordering of the patients corresponds with the ordering in the data. In bold, the two patients whose first sequencing was performed after starting treatment. 26
3.2 Summary of the created arrays. 28
3.3 Summary of the generated arrays. 29

4.1 DIFFIT results for the simulated *tensor_1*. In bold, the optimal rank with the largest $b_{t(m)}$ value. 44
4.2 DIFFIT results for the simulated *tensor_2*. In bold, the optimal rank with the largest $b_{t(m)}$ value. 45
4.3 DIFFIT results for *tensor_70*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%. 47
4.4 DIFFIT results for *tensor_100*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%. 47
4.5 DIFFIT results for *tensor_200*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%. 48
4.6 DIFFIT results for *tensor_300*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%. 48
4.7 NMI scores comparing TCR-based, Tucker-based clusterings, and ground-truth (GT) labels across tensor subsets. 66

1
Introduction

This chapter presents a brief background on sarcoma and the role of T cells in this pathology. In addition, related work on tensor decomposition and its significance is discussed in the context of this study. The objectives of the thesis are subsequently outlined. Finally, the chapter concludes with a description of the overall structure of the thesis.

1.1 Background and Related Work

Sarcoma is an unusual type of cancerous tumor, representing only 1% of adult and about 15% of child cancer diagnoses in Europe [1], and nearly 21% of the child solid cancer diagnoses in the United States of America [2]. It is a heterogeneous maladie with more than 70 known subtypes, which can mainly be categorized into bone sarcomas and soft tissue sarcomas [3]. The overall incidence rate is approximately 5.6 cases per 100,000 people annually, with about 85% classified as soft tissue sarcomas and 13–15% as bone sarcomas. The respective survival rates are 58% and 62% [1].

The immune system is a complex network of interactions with several cells and actors to face any non-self entity entering the body. In terms of cancer, the immune system is capable of detecting tumor cells, which show abnormal quantities of membrane proteins [4]. One of the immune cells which can detect these are T cells. These can do so as they express a membrane receptor, the so-called T-cell receptor (TCR), which is able to detect these non-self compounds.

The TCR is made from two different transmembrane glycoprotein chains. There are four types of these chains: α, β, γ and δ. Based on the TCR, T cells can be categorized in two classes: $\alpha\beta$ and $\gamma\delta$ T cells. The first is the most typical configuration, representing around 90% of the total T cell population, while the latter represents up to a 10%. However, it has been observed that in epithelial tissues, this fraction is fairly higher [5].

The number and composition of T cells (and the dynamical changes that alter these) have been found to be closely related to survival and response to treatment in cancer patients [6]. One way to monitor this is through DNA sequencing of blood samples with an ultrasensitive technique [7]. Blood samples offer a less invasive alternative to direct tumor biopsies, particularly when the tumor is located in a region that is difficult to access.

However, data resulting from this are hard to analyze in their raw form due to their high dimensionality and complex structure. Tensor decomposition methods, which are mathematical tools that reduce the dimensionality of data arrays or tensors, enable researchers to uncover latent patterns that may not be apparent through traditional analysis. These techniques are particularly useful in biological settings where data can span multiple modes, such as time, tissue types, or cell populations.

For instance, Hore et al. [8] applied tensor decomposition to multi-tissue gene expression data. By organizing expression levels across individuals, tissues, and genes into a tensor, they were able to extract sparse gene networks that had both statistical and biological relevance, while controlling for confounding factors. In addition, tensor decomposition has also been used for multi-way analysis of EEG data [9].

In the context of this thesis, tensor decomposition is employed to analyze time-resolved sequencing data from $\gamma\delta$ T cells, with the goal of identifying dynamic immune signatures associated with sarcoma progression. This approach allows for the discovery of latent biological structure in complex longitudinal datasets.

1.2 Aim

Given the demonstrated potential of tensor decomposition for analyzing data with three or more modes, this project aims to apply this technique to longitudinal $\gamma\delta$ T cell clone data from sarcoma patients at Sahlgrenska University Hospital. The goal is to uncover underlying biologically meaningful patterns in the dataset, such as patient subgroups or characteristic disease trajectories.

1.3 Structure

This thesis follows a structure as outlined below:

- **Chapter 1** introduces the background and related work, outlines the aim of the thesis, and provides an overview of its structure.
- **Chapter 2** delves into the theoretical background, including sarcoma classification and treatment, T cell biology with a focus on $\gamma\delta$ T cells, DNA-based immune repertoire sequencing, and the mathematical foundations of tensor decomposition.
- **Chapter 3** describes the methodology used throughout the project, including dataset construction, preprocessing, simulation of data, decomposition models, and the application of immunology-related metrics.
- **Chapter 4** presents the results, including the outcomes of simulation studies, decomposition analyses, clustering of temporal patterns, and immune metric evaluation.
- **Chapter 5** provides a discussion of the results, reflects on limitations, and suggests directions for future research.
- **Chapter 6** summarizes the key conclusions of the thesis.

2
Theory

This chapter provides a brief introduction to sarcoma and T cells, along with an overview of current DNA sequencing methods, with particular emphasis on the technique used to generate the dataset for this project. It also presents background on tensors and the mathematical principles underlying the tensor decomposition methods employed in this thesis. Much of the content on tensors in this chapter is based on the work of Kolda and Bader [10].

2.1 Sarcoma

This section presents a brief description of what sarcoma is and its subtypes, epidemiology, risk factors, and current treatments.

2.1.1 Description, Epidemiology, and Risk Factors

Sarcoma is a diverse group of cancers that arise in bone or soft tissue, encompassing over 70 subtypes of the disease [3]. These subtypes are typically classified based on their site of origin as either primary bone sarcomas or soft tissue sarcomas. An overview of the most common subtypes is provided in Table 2.1. This maladie is a fairly unusual disease, representing only 1% of adult and about 15% of child cancer diagnoses in Europe [1], and nearly 21% of the child solid cancer diagnoses in the United States of America [2]. Its overall incidence is estimated at 5.6 cases per 100,000 people annually, with around 85% classified as soft tissue sarcomas and 13–15% as bone sarcomas. The respective survival rates are 58% and 62% [1].

Despite being a heterogeneous group of tumors, there have been four familiar cancer syndromes identified to be associated with sarcomas [11]. For instance, mutations in the retinoblastoma gene have been linked to a higher frequency of suffering from osteosarcoma. Also, people with Li-Fraumeni syndrome (which translates to mutations in the p53 gene, responsible for regulating cell division) show a higher chance of developing cancer, including a variety of sarcomas. The loss of the NF1 gene has also been associated with increased risk of sarcoma, more specifically, malignant peripheral nerve sheath tumor. Lastly, mutations in c-kit (a receptor that binds to stem cell factor, which signals some specific cells to grow [12, 13]) has been shown to be present in gastrointestinal stromal tumors.

Beside these genetic factors, other risk factors that have been associated with in-

creased risk of suffering from sarcoma include exposure to ionizing radiation (including radiation therapy for other cancers), chronic lymphedema, exposure to chemicals (such as vinyl chloride), and human herpes-virus 8 [14].

2.1.2 Treatment Strategies

Currently, there are clear standard therapies that aim to treat sarcoma. These are usually used in combination with other therapies. Here, only the most common ones are detailed. For a more complete overview and treatments still under development, refer to [15].

Classical therapy is to surgically resect the primary tumor (or any resectable tumor) whenever it is possible. Surgeons remove all tissue detected to be cancerous as well as a safety margin to ensure that there are no cancerous cells left behind. When the tumor is found on a limb, it is considered whether the limb can be salvaged or needs to be amputated. Since both have demonstrated similar oncologic outcomes, limb salvaging is usually preferred as long as the limb reconstruction leads to a functional extremity with preserved neurovascular structures.

In addition to surgical resection, chemotherapy is often administered before and/or after surgery to reduce tumor size and eliminate any remaining cancerous cells. For osteosarcoma and Ewing's sarcoma, as they are normally sensitive to chemotherapy, a combination of doxorubicin (an anthracycline), cisplatin (platinum-based), methotrexate (a folate pathway inhibitor), and ifosfamide (an alkylating agent) is employed. For soft tissue sarcoma, only doxorubicin, or in combination with ifosfamide, are usually delivered as well as radiation therapy to control the tumor growth in localized areas. By contrast, for treating chondrosarcoma, a high reliance on wide margin resection is needed as it is resistant to current chemo and radiation therapies to a great degree. A more specialized chemotherapy, namely chemo-targeted therapy, targets specific mutations. For instance, most gastrointestinal stromal tumor patients present mutations in both proto-oncogene receptor kinase, referred to as c-kit in previous sections, and platelet-derived growth factor receptor alpha (involved in tumor progression [17]). Therefore, these are targeted by tyrosine kinase inhibitors, such as imatinib, which block the growth signals from these unregulated genes.

Additional mutations in cancer cells can lead to drug resistance caused by the selective pressure against the drug-sensitive tumor population. Therefore, efforts are being made to develop new treatment strategies. An example of this is the recent research on immunotherapy, which consists of modifying T cells to better detect patient-specific tumor cell antigens. The effectiveness of this therapy is still being studied in clinical trials, but so far, promising results have been obtained for synovial sarcoma.

Table 2.1: Most common sarcoma subtypes. Adapted from [15]. *Not included in [15] but present in our dataset; information from [16].

Sarcoma Subtype		**Description**
Bone Sarcoma	Chondrosarcoma	Arises in cartilage and bone (surface or center). Originates from mutant chondrocytes.
	Ewing Sarcoma	Arises in long and flat bones and extraskeletal sites (85% and 15%). Originates from small and round undifferentiated mesenchymal cells.
	Osteosarcoma	Arises on the surface or the center of bones. Originates from differentiated mesenchymal cells commited to bone that produce an extracellular osteoid matrix.
Soft Tissue Sarcoma	Fibrosarcoma	Arises in deep soft tissues. Originates from monomorphic fibroblastic cancer cells in collagenous matrix.
	Gastrointestinal Stromal Tumors	Arises in the gastrointestinal track (mainly stomach and small intestine). Originates from spindle and epitheloid cells or cells differentiating towards interstitial cells of Cajal.
	Leimyosarcoma	Arises in the peritoneum and uterus. Originates from mesenchimal cells with smooth muscle differentiation.
	Liposarcoma	Arises in variable locations (most common in retroperitoneal space). Originates from adipocyte-like cells with variable differentiation and heterogeneous morphology, growing within a vascularized stroma.
	Rhabdomyo-sarcoma	Arises in variable locations. Originates from mesenchymal cells with variable myogenic differentiation.
	Undifferentiated pleomorphic sarcoma	Arises most frequently in the limbs. Originates from undifferentiated cells that lack specialized features and abnormal cell structure and variation in cell shape and size.
	Synovial sarcoma	Arises mostly ini deep soft tissue of the limbs. Originates from spindle cells with variable differentiation (mesenchymal and/or epithelial).
	Angiosarcoma*	Arises most commonly in cutaneous lesions (60%) and in soft tissues, visceral organs, bone and retroperineum. Originates from spindled, polygonal, epitheloid and primitive round cells, showing vascular and endothelial differentiation.

2.2 T Cells

T cells derive from hematopoietic stem cells that mature in the thymus [18, 19]. One can classify them by TCR structure, function, maturation stage, or tissue localization. Nevertheless, the following discussion focuses on the first classification, as the rest are not as relevant for this project. For a more complete overview, refer to [4, 18, 20]

T cells can express two types of TCR. This receptor is made from two different transmembrane glycoprotein chains, of which four types exist, namely α, β, γ, and δ. However, only two combinations of these chains have been identified: $\alpha\beta$ or $\gamma\delta$. Approximately 90–95% of T cells express the $\alpha\beta$ TCR, while the remaining 5–10% express the $\gamma\delta$ TCR [5, 18]. Notably, most $CD4^+$ and $CD8^+$ T cells express $\alpha\beta$ TCRs, whereas $\gamma\delta$ T cells typically express neither CD4 nor CD8, with only a small subset co-expressing one or both markers [4]. Therefore, they do not recognize antigens presented by Major Histocompatibility Complex (MHC) class I nor II, but they interact with those presented by the non-classical CD1 MHC, like Natural Killer T Cells (NKT) [18]. However, $\gamma\delta$ can also recognize antigens in a MHC independent manner [4], allowing them to recognize phosphoantigens, lipids, small molecules, and stress-induced proteins.

2.2.1 Role of $\gamma\delta$ T Cells in Tumor Surveillance

A hallmark of tumor immune evasion is the downregulation of MHC molecules [21, 22, 23], which prevents their recognition by conventional T cells that rely on MHC-mediated antigen presentation, like $\alpha\beta$ T cells. Unlike them, $\gamma\delta$ T cells are not constrained by this mechanism: they can recognize tumor cells in an MHC-independent manner, oftentimes due to the presence of stress-associated molecules on their surface [24].

In cancer, $\gamma\delta$ T cells exert antitumor effects through two main mechanisms [24]. The first is direct cytotoxicity, in which, upon encountering transformed or stressed cells, $\gamma\delta$ T cells can rapidly respond by killing these targets. This response does not rely on prior sensitization or classical antigen presentation but is triggered by stress-induced cues on tumor cells. Depending on the tumor type and context, $\gamma\delta$ T cells can eliminate cancer cells through multiple killing strategies, such as the release of perforins and granzymes (to induce tumor cell death), expressing FasL and TRAIL (to trigger apoptosis in tumor cells), or engaging in antibody-dependent cytotoxicity through CD16.

Their second antitumor function lies in their ability to shape and coordinate the broader immune response, bridging innate and adaptive immunity. That is possible because they share rapid, nonspecific activation, like innate immune cells, but they can also affect the activation of other adaptive immune cells. So, $\gamma\delta$ T cells are early producers of IFNγ, which results in an amplification of $\alpha\beta$ T cells and an induction of the expression of MHC on tumor cells, avoiding the immune escape due to the

downregulation of it. Moreover, $\gamma\delta$ T cells can promote B cell functions, including antibody production of IgE, which has been shown to have a tumor-protective effect.

Unfortunately, despite these beneficial roles, $\gamma\delta$ T cells do not always act in favor of the host. In certain tumors, especially under the influence of specific cytokines in the microenvironment, some $\gamma\delta$ T cell subsets can promote tumor growth. A key example is the subset that produces IL-17. Rather than contributing to tumor elimination, these cells can support angiogenesis, recruit immunosuppressive cells, and facilitate metastasis. Other $\gamma\delta$ T cells can suppress immune activity more directly, either by inhibiting cytotoxic T cells or by mimicking regulatory T cell functions.

2.2.2 T Cell Receptor Variability

The TCR is the main receptor in charge of antigen recognition. Since there is an extremely large number of antigens from pathogens and malignant cells, the number of possible antigen combinations is huge. This added to the fact that pathogens and malignant cells can rapidly mutate and evolve, makes it nearly impossible to store all those antigen combinations in the germline DNA that codifies for antigen receptors. Therefore, the immune system has evolved to customize such antigen receptors [18].

In order to achieve these tailored receptors, TCRs undergo a recombination process of the gene segments that codify for this receptor [5, 6, 18]. This process occurs in the thymus during T cell development, where three processes occur. The first process is somatic recombination of gene segments. Enzymes cut and join one of each variable (V), diversity (D, which is only in β and δ chains), and joining (J) segments of the TCR gene, which are later recombined and spliced onto a constant (C) segment. The second process is the generation of junctional diversity, where additional random modifications at the joining locations between the segments are made, such as the insertion or deletion of nucleotides, providing an increased diversity. Lastly, combinatorial pairing takes place. This process brings the individual chains (α, β, γ, and δ) together. Since each TCR is formed by a pair of chains, random pairing at this stage introduces another layer of variability. It is estimated that the recombination process can theoretically generate up to 2×10^{19} different $\alpha\beta$ TCR [25]. This number only increases for $\gamma\delta$ TCR as the δ chain includes multiple D segments [5].

Despite the substantial number of different TCR clones that are generated, only a few progress to T cell maturation. There are two reasons as to why this happens. Firstly, some receptors could recognize self-antigens and trigger autoreactive responses. Therefore, a negative selection takes place to ensure that no T cell will react strongly to self antigens. However, T cells still need to be able to recognize self-cells. For that reason, the second selection process, called thymic selection, positively selects T cells whose TCR are able to recognize self-peptides in the context of self-recognition by giving them a survival signal [5].

2.3 DNA Immune Repertoire Sequencing

2.3.1 Sequencing Process

The immune receptor repertoire refers to the complete set and distribution of immune receptors present in an individual at a given point in time [26]. Sequencing, in this context, involves determining the precise order of nucleotides in DNA or RNA molecules. Consequently, immune repertoire sequencing entails identifying the nucleotide sequences that encode immune receptors, such as TCR or B cell receptors. These techniques enable high-throughput analysis of receptor diversity by reading the DNA molecules extracted from biological samples. In the case of immune repertoire sequencing, targeted sequencing is commonly employed, which focuses on specific regions of DNA, particularly those encoding the variable segments of TCR genes.

The data analyzed in this thesis was generated using a highly sensitive sequencing technique developed by Johansson et al. [7], based on the principle of sequencing-by-synthesis, as used in platforms such as Illumina and SiMSen-Seq. This method targets the Complementarity-Determining Region 3 (CDR3) region of the TCR gene, which includes the V and J segments, and enables detection of specific DNA sequences from blood samples, even when present at frequencies as low as 0.1%. By incorporating unique molecular identifiers (UMIs) and targeted amplification, this technique allows for accurate quantification of original $\gamma\delta$ TCR sequences while minimizing amplification bias. The full protocol, including DNA isolation from blood samples, is described in the original publication.

2.3.2 Common Immunology Metrics

In the context of T cells and their TCRs, immune sequencing has been employed to assess the number and distribution of different TCR clonotypes across various settings. By applying a range of diversity and clonality metrics, this approach has shown promise in predicting patient prognosis and response to therapy in cancer [6, 27, 28, 29]. Specifically, Porciello et al. [6] report that high baseline TCR diversity is often associated with better responses to immunotherapy. Conversely, increased clonality (indicating a repertoire dominated by a limited number of T cell clones) or reduced evenness following treatment may reflect tumor-specific T cell expansion, which is likewise correlated with improved clinical outcomes. In contrast, low TCR diversity observed in aged individuals and cancer patients is indicative of immune senescence and may serve as a marker of reduced immune fitness.

Before introducing the metrics, a few definitions need to be introduced. As [6] describes, *richness* (R) refers to the number of unique TCR sequences in a sample, *evenness* refers to the distribution of such unique TCR sequences (for instance, a sample is considered even if the different sequences have the same distribution and there are no dominant sequences), and *diversity* is a term that accounts for both richness and evenness, therefore, a high diversity indicates a broad and poly-

clonal immune repertoire, capable of recognizing a wide range of antigens, while a low diversity suggests a more clonal or skewed response, often due to expansion of antigen-specific T cells.

The most commonly used diversity metrics include the Shannon entropy, Inverse Simpson index, Gini coefficient, and the Diversity Evenness score (DE_{50}) score. Additional metrics such as clonality and Pielou's evenness index are derived from the Shannon entropy and provide complementary insights into repertoire structure. Next, these are defined and summarized in Table 2.2.

The Shannon entropy index (H) incorporates both richness and evenness, providing a measure of overall diversity. However, it is sensitive to small fluctuations in the frequencies of rare clonotypes and assumes a relatively uniform clone distribution. It is defined as [30]:

$$H = -\sum_{i=1}^{R} p_i \log p_i \tag{2.1}$$

where p_i is the proportion of the i-th clonotype. This index ranges from 0, indicating low diversity with dominance by a few clones, to $\log(R)$, where R is the richness, corresponding to a highly diverse and evenly distributed repertoire with no dominant clonotypes.

Based on this, one can define clonality (C), a metric used to estimate the degree of clonal expansion within a repertoire. Conceptually, it reflects the probability that two randomly selected sequences originate from the same clone and is therefore inversely related to diversity. It is defined as [31]:

$$C = 1 - \frac{H}{\log(R)} \tag{2.2}$$

where H is the Shannon entropy and R is the richness. Clonality ranges from 0, indicating a highly diverse and evenly distributed repertoire, to 1, indicating a monoclonal population dominated by a single clone.

Pielou's evenness index is a diversity metric derived from Shannon entropy that quantifies how evenly clone frequencies are distributed within a repertoire. It ranges from 0, indicating dominance by a few clones, to 1, where all clones are present in equal frequencies. The index is calculated by dividing the Shannon entropy by its maximum possible value for a given richness R, and is defined as [32]:

$$J = \frac{H}{\log(R)} \tag{2.3}$$

The Inverse Simpson index is particularly useful for datasets with high-frequency reads, as it places greater emphasis on the most abundant clonotypes. In other words, it is less sensitive to rare clones, unlike Shannon, and instead reflects the dominance of a few highly represented sequences. It is defined as [30]:

$$D = \frac{1}{\sum_{i=1}^{S} p_i^2} \tag{2.4}$$

where p_i is the relative frequency of the i-th clonotype. The index ranges from 1, indicating a monoclonal population, to R, the total number of unique clonotypes, which corresponds to a perfectly even distribution.

The Adapted Gini coefficient (AG) is a diversity measure proposed by Rousseau et al. [33] to quantify inequality in ecological or immunological contexts. It measures how far the actual distribution of TCRs deviates from a perfectly even one. The bigger the gap (area), the more uneven the TCR distribution is. It is defined as:

$$AG = 2\left[\sum_{i=1}^{R} ip_i\right] - 1 \tag{2.5}$$

where p_i are the normalized abundances (i.e., clone frequencies) sorted in descending order, and R is the number of unique clones. In some versions of the formula, a factor of $\frac{1}{n}$ is included for normalization when frequencies are not already scaled to sum to 1. However, in the context of immune repertoire analysis, where $\sum p_i = 1$, the simplified form above is appropriate and commonly used. AG does not necessarily range between 0 and 1, but it can be normalized using an arctangent transformation as follows:

$$\frac{2}{\pi}\arctan(AG(X)) \tag{2.6}$$

Therefore, values close to 0 mean perfect equality while values close to 1 represent perfect inequality, meaning that a few clones dominate the population.

Another widely used diversity measure is the Gini–Simpson index (G'), which reflects the probability that two randomly chosen elements belong to different clones. It is defined as [34]:

$$G' = 1 - \sum_{i=1}^{R} p_i^2 \tag{2.7}$$

where p_i is the normalized frequency of the i-th clone. The Gini–Simpson index ranges from 0 (no diversity) to nearly 1 (maximum diversity), and is conceptually distinct from the original Gini inequality index.

Lastly, the DE_{50} index is a clonality metric that quantifies how many unique clonotypes contribute to the first 50% of total reads when the repertoire is ordered by decreasing clone frequency. In other words, it reflects the number of dominant clones required to account for half of the repertoire, providing insight into the degree of clonal expansion. The index ranges from 1, indicating that only one clone dominates the sample, to R, the total number of unique clones, which corresponds to a perfectly even distribution. It is defined as [35]:

$$DE_{50} = \min_k \left\{ \sum_{i=1}^{k} p_i \geq 0.5 \right\} \tag{2.8}$$

where p_i denotes the normalized frequency of the i-th most abundant clonotype.

Table 2.2: Summary of commonly used immunology metrics, typical range and interpretation. R stands for richness, the unique TCR sequences of the sample.

Metric	Measure of	Typical range	Interpretation
Shannon entropy	Richness and evenness	[0, log(R)]	[monoclonal, diverse]
Clonality	Diversity	[0, 1]	[diverse, monoclonal]
Pielou	Evenness	[0, 1]	[monoclonal, diverse]
Inverse Simpson	Diversity	[1, R]	[monoclonal, diverse]
AG (Norm.)	Inequality	[0, 1]	[equal, unequal]
Gini-Simpson	Diversity	[0, 1]	[monoclonal, diverse]
DE_{50}	Clonality	[1, R]	[monoclonal, diverse]

2.4 Tensors and Preliminaries

A *tensor* is a multidimensional array. The number of dimensions of a tensor is known as *modes*, *ways* or *order*, that is, a N-mode, N-way or Nth-order tensor is an element of the product of N vector spaces. In this sense, a matrix would be a 2-way tensor and a vector a 1-way tensor. Tensors of order three or higher are usually called higher-order tensors. In Figure 2.1 a vector $\mathbf{a} \in \mathbb{R}^I$, a matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ and a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ are illustrated.

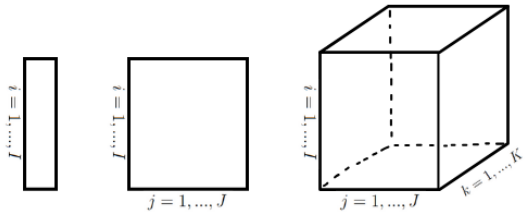


Figure 2.1: A vector, a matrix and a third-order tensor.

Fibers are the higher-order equivalent for matrix rows and columns, and they are defined by fixing every index but one. In matrices, a row would be a mode-1 fiber and a column a mode-2 fiber. In 3-way tensors, mode-1, mode-2, and mode-3 fibers represent row, column, and tube fibers, respectively, as shown in Figure 2.2. Similarly, *slices* are two-dimensional sections of a tensor and are defined by fixing every index but two. In 3-way tensors, horizontal, lateral, and frontal slices are obtained by fixing modes one, two, and three, respectively, as shown in Figure 2.3.

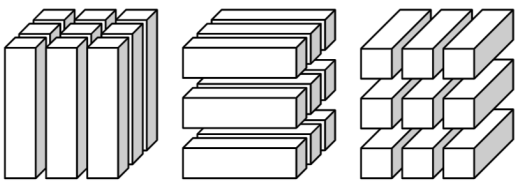


Figure 2.2: Mode-1, mode-2 and mode-3 fibers.

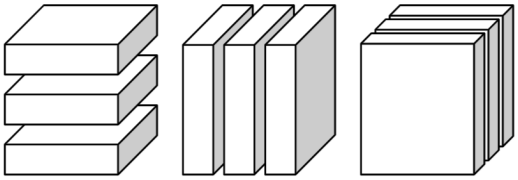


Figure 2.3: Horizontal, lateral and frontal slices.

A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is said to be *diagonal* if $x_{i_1 i_2 \ldots i_N} = 0$ only if $i_1 = i_2 = \ldots = i_N$. The elements $x_{i_1 i_2 \ldots i_N}$ such that $i_1 = i_2 = \ldots = i_N$ form the *superdiagonal* of the tensor.

2.4.1 Tensor Operations

Some relevant operations and related concepts are presented in this section. Namely, the matricization of a tensor and some relevant product operations: the inner product and the norm of a tensor, the outer product of vectors together with rank-one tensors, the matricization of a tensor and the n-Mode product.

The *Matricization* of a tensor, also known as unfolding or flattening, consists in the transformation of a tensor into a matrix by reordering its elements. In this report, the special case of mode-n matricization will be reviewed, as it is the only relevant form of matricization for this project; for a more general discussion of matricization, see Kolda and Balder [36]. Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, its mode-$n$ matricization, denoted by $\mathbf{X}_{(n)}$, is computed by arranging the n-mode fibers of the tensor to be the columns of $\mathbf{X}_{(n)}$: the tensor element $x_{i_1 i_2 \ldots i_N}$ in $\mathcal{X}$ is mapped to the matrix element $x_{i_n j}$ in $\mathbf{X}_{(n)}$ where

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \quad \text{with} \quad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m. \tag{2.9}$$

In Figure 2.4 an example of a mode-1 matricization is shown. The mode-1 fibers $\mathbf{x}_{:jk}$ are rearranged into the columns of the new matrix.

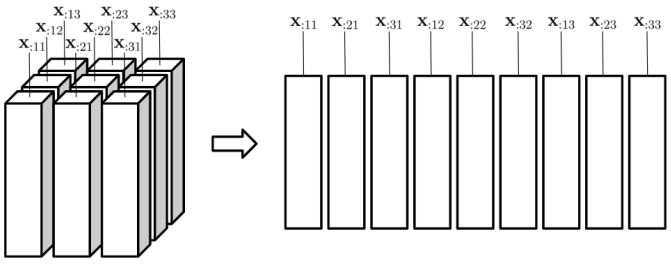


Figure 2.4: The mode-1 matricization of a tensor $\mathcal{X}$.

The *inner product* of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the sum of the products of

their elements,

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} x_{i_1 i_2 ... i_N} y_{i_1 i_2 ... i_N}. \tag{2.10}$$

Note that $\mathcal{X}$ and $\mathcal{Y}$ must have the same size.

Analogous to the Frobenius norm, the *norm* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is the square root of the sum of the squares of all its elements, and it is equivalent to computing the square root of the inner product of the tensor with itself,

$$||\mathcal{X}|| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} x_{i_1 i_2 ... i_N}^2}. \tag{2.11}$$

Given two vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^J$, their *outer product*, denoted by $\mathbf{a} \circ \mathbf{b}$, is the matrix $\mathbf{M} \in \mathbb{R}^{I \times J}$ resulting from the multiplication of each element of both vectors, such that $m_{ij} = a_i b_j$.

Analogously, given N vectors $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, ..., \mathbf{a}^{(N)}$, their outer product defines a tensor $\mathcal{X}$,

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ ... \circ \mathbf{a}^{(N)}. \tag{2.12}$$

The tensor $\mathcal{X}$ is said to be *rank one* as it can be expressed as the outer product of vectors. Every element of the tensor is defined as

$$x_{i_1 i_2 ... i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} ... a_{i_N}^{(N)} \tag{2.13}$$

A three-mode rank-one tensor is illustrated in Figure 2.5.

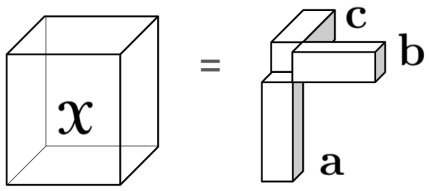


Figure 2.5: Rank-one tensor.

The *n-Mode product* consists in multiplying a tensor by a matrix or a vector in mode n. The definitions and properties of the n-Mode product differ if the tensor is multiplied by a matrix or a vector. For that reason, both operations will be presented separately.

Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$, the n-Mode matrix product of the former with the latter is denoted by $\mathcal{X} \times_n \mathbf{A}$ and results in a tensor of size $I_1 \times I_2 \times ... \times I_{n-1} \times J \times I_{n+1} \times ... \times I_N$. The product is computed as follows: each mode-n fiber of the tensor is multiplied by the matrix $\mathbf{A}$, which leads to the following result element-wise:

$$(\mathcal{X} \times_n \mathbf{A})_{i_1 ... i_{n-1} j i_{n+1} ... i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 ... i_N} a_{j i_n}. \tag{2.14}$$

When more than one product is performed in different modes then the multiplication is commutative, i.e.,

$$\mathcal{X} \times_n \mathbf{A} \times_m \mathbf{B} = \mathcal{X} \times_m \mathbf{B} \times_n \mathbf{A} \tag{2.15}$$

If more than one product is performed in the same mode, then we have the following property,

$$\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n (\mathbf{BA}) \tag{2.16}$$

The n-Mode vector product can be defined in a similar way. Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ and a vector $\mathbf{a} \in \mathbb{R}^{I_n}$, the n-Mode vector product is denoted by $\mathcal{X} \bar{\times}_n \mathbf{a}$ and results in a tensor of size $I_1 \times I_2 \times ... \times I_{n-1} \times I_{n+1} \times ... \times I_N$. Analogous to its matrix counterpart, the product is computed by multiplying each mode-n fiber of the tensor by the vector a, which leads to the following result element-wise:

$$(\mathcal{X} \bar{\times}_n \mathbf{a})_{i_1 ... i_{n-1} i_{n+1} ... i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 ... i_N} a_{i_n}. \tag{2.17}$$

Unlike the matrix product, the multiplication performed in different modes is not commutative, as the order of the intermediate results changes. In particular, we have the following property, for $n < m$:

$$\mathcal{X} \bar{\times}_n \mathbf{a} \bar{\times}_m \mathbf{b} = (\mathcal{X} \bar{\times}_n \mathbf{a}) \bar{\times}_{m-1} \mathbf{b} = (\mathcal{X} \bar{\times}_m \mathbf{b}) \bar{\times}_n \mathbf{a}. \tag{2.18}$$

2.5 Tensor Decomposition

Tensor decomposition is a powerful tool for tensor analysis and dimensionality reduction, and it has applications in several fields, from numerical linear algebra to neuroscience. In this section, the well-known methods Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) will be reviewed. Next, the two main tensor decompositions that can be thought of as higher-order extensions of both SVD and PCA will be presented: CANDECOMP/PARAFAC (CP) decomposition and Tucker decomposition. A variation of CP, PARAFAC2 will also be presented. Finally, some metrics that will be used to evaluate the fit of the decompositions will be discussed.

2.5.1 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) method decomposes a matrix $\mathbf{A}$ into the product of three matrices such that

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{2.19}$$

where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n)$ and $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)$ are orthogonal matrices and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_n)$ has nonnegative diagonal elements arranged in descending order of magnitude [37]. $\sigma_1, \sigma_2, ..., \sigma_n$ are called the *singular values* of $\mathbf{A}$ whereas $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n$ and $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$ are the *left* and *right singular vectors* of $\mathbf{A}$ respectively.

The number of singular values of $\mathbf{A}$ is said to be the rank of the matrix [38]. In this sense, the SVD of $\mathbf{A}$ can be written as a sum of rank-one matrices [39],

$$\mathbf{A} = \sum_{i=1}^{\text{rank}(\mathbf{A})} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r. \tag{2.20}$$

The SVD of a matrix is unique given the orthogonality constraints and assuming the singular values are distinct and ordered in the diagonal matrix [10]. In Figure 2.6, the SVD of a matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ is illustrated, with $\mathbf{U} \in \mathbb{R}^{I \times N}$, $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$ and $\mathbf{V} \in \mathbb{R}^{J \times N}$.

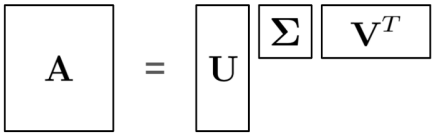


Figure 2.6: The SVD of a matrix $\mathbf{A}$.

Let $\mathbf{A}$ be a matrix whose SVD is described as in Equation 2.20. Then, a *rank-k* approximation that minimizes $||\mathbf{A} - \mathbf{B}||$ is given by

$$\mathbf{B} = \sum_{i=1}^{k} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r. \tag{2.21}$$

This is, a matrix $\mathbf{A}$ can be approximated by the leading k factors of its SVD [10]. The rank-k approximation is widely used for compression and noise reduction [40].

2.5.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) provides an approximation for a matrix $\mathbf{A}$ by decomposing it into the product of two matrices as follows,

$$\mathbf{A} = \mathbf{F}\mathbf{Q}^T, \tag{2.22}$$

where $\mathbf{F}$ is the factors score matrix and $\mathbf{Q}$ the loading matrix. These matrices capture the essential data patterns of $\mathbf{A}$, i.e. the principal components. These are computed as linear combinations of the original variables; each principal component is computed with the largest possible variance, and with the constraint of being orthogonal to each other [41].

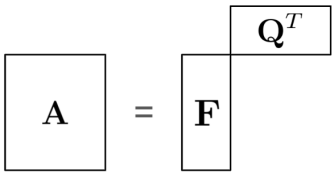


Figure 2.7: PCA: A matrix decomposed into the factors score and loading matrix.

In Figure 2.7, the decomposition of a matrix $\mathbf{A}$ into $\mathbf{F}$ and $\mathbf{Q}^T$ is illustrated. The columns in $\mathbf{F}$, known as *score vectors*, represent the values of the original data in the new variables or factor scores, while the rows of $\mathbf{Q}^T$, known as *loading vectors*, explain how every variable contributes to the factor scores [41, 42].

Assuming that the matrix $\mathbf{A}$ is centered column-wise, there exists a direct relation between PCA and SVD: the factor scores can be obtained directly from the SVD [40, 41]. Particularly, the following equality holds,

$$\mathbf{F} = \mathbf{U}\mathbf{\Sigma}. \tag{2.23}$$

2.5.3 CANDECOMP/PARAFAC (CP) Decomposition

The concept of a polyadic form of a tensor, that is, expressing a tensor as a finite sum of rank-one tensors, was first introduced by Hitchcock in 1927 [43]. In 1970, it started to gain popularity among the psychometrics community with the names CANDECOMP (canonical decomposition) [44] and PARAFAC (parallel factors) [45]. Today, it is commonly referred to as CANDECOMP/PARAFAC (CP) decomposition, as proposed by Kiers [46]. In Figure 2.8 the CP decomposition of a third-order tensor is illustrated. The tensor $\mathcal{X}$ is decomposed into a sum of R rank-one tensors, which can be expressed as $\mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$ for $1 \leq i \leq R$.

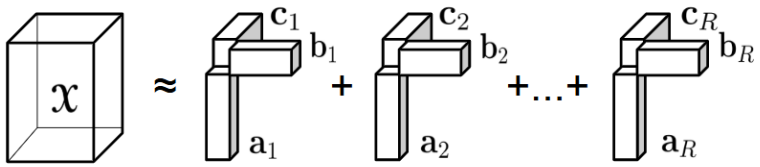


Figure 2.8: CP Decomposition of a third-order tensor.

The task of computing the CP Decomposition for a given third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is often formulated as an optimization problem,

$$\min_{\hat{\mathcal{X}}} ||\mathcal{X} - \hat{\mathcal{X}}||, \quad \text{where} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{2.24}$$

The goal is to minimize the difference $||\mathcal{X} - \hat{\mathcal{X}}||$, having

$$\mathcal{X} \approx \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{2.25}$$

Here, R, a positive integer, is the number of components of the decomposition, $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$ for $r = 1, 2, ..., R$ are the vectors from the rank-one tensors and, if normalized, $\lambda \in \mathbb{R}^R$ is a vector containing the weights. The $\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r$ vectors can be combined into three *factor matrices* $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$,

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}_1 \;\; \mathbf{a}_2 \;\; ... \;\; \mathbf{a}_R], \\ \mathbf{B} &= [\mathbf{b}_1 \;\; \mathbf{b}_2 \;\; ... \;\; \mathbf{b}_R], \\ \mathbf{C} &= [\mathbf{c}_1 \;\; \mathbf{c}_2 \;\; ... \;\; \mathbf{c}_R]. \end{aligned} \tag{2.26}$$

The focus of this report will be on third-order tensors, but it is also possible to express the CP decomposition for a Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ as

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ ... \circ \mathbf{a}_r^{(N)}. \tag{2.27}$$

The rank of a tensor $\mathcal{X}$, $rank(\mathcal{X})$, is defined as the smallest number of rank-one tensors required to generate $\mathcal{X}$ as their sum [43, 47]. In this sense, the rank of a third-order tensor $\mathcal{X}$ would be R such that

$$\mathcal{X} = \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{2.28}$$

A CP decomposition with $R = rank(\mathcal{X})$ is known as *rank decomposition*.

Although the definition of the rank of a tensor is analogous to the matrix case, as seen in subsection 2.5.1, their properties are somewhat different. One difference is that the rank of a tensor with real values may not be the same depending on the domain, which does not apply to matrix ranks. Another important difference is that, in general, there is no direct way to compute the rank of a tensor, as opposed to the matrix case, which can be easily obtained with the SVD. Moreover, Håstad showed that the problem of computing the rank of a tensor is NP-Hard [48].

As described in subsection 2.5.1, the decomposition of a matrix is unique only under certain constraints and assumptions. However, the CP decomposition is unique under much weaker conditions. $\mathcal{X} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ is only possible for a unique combination of rank-one tensors, with the exception of the permutation of said rank-one tensors and the scaling of the vectors $\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r$ for $r = 1, ..., R$. A sufficient condition for uniqueness for CP decompositions was presented by Kruskal [47] for third-order tensors,

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2, \tag{2.29}$$

where R is the number of components and $k_{\mathbf{X}}$ represents the k-rank of a matrix X, i.e., the maximum value k such that k columns of the matrix are linearly independent. In Equation 2.29, $k_{\mathbf{A}}, k_{\mathbf{B}}, k_{\mathbf{C}}$ represent the k-ranks of the factor matrices.

In subsection 2.5.1, the concept of rank-k approximation was introduced, stating that a matrix can be approximated by the leading k factors of its SVD. However, this does not hold true for tensors and CP decompositions, and a rank-k approximation might not even exist. In cases where a good rank-k approximation for a tensor does exist, the tensor is said to be *degenerate*.

As mentioned earlier, finding the rank of a tensor is NP-Hard, thus, the first issue that arises when computing a CP decomposition for a tensor is the number of components R to select. This is not a trivial task and will be discussed in later sections.

There exist many algorithms to compute the CP decomposition of a tensor given

the number of components. The Alternating Least Squares (ALS) method is the most established and was presented in 1970 [44, 45]. In Algorithm 1, the ALS for third-order tensors is described, adapted from Kolda and Balder [10]. Note that the symbols $\odot$, $*$ and † represent the *Khatri-Rao product*, the *Hadamard product*, and the *Moore–Penrose pseudoinverse* of a matrix, respectively. For a more detailed explanation of these, see Kolda and Balder [10].

Algorithm 1 ALS algorithm to compute the CP decomposition with R components for a third-order tensor $\mathcal{X}$ of size $I \times J \times K$.

Input: $\mathcal{X}$, R
Initialize: $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$
  repeat
    $\hat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$
    for $r = 1...R$ **do**
      $\lambda_r \leftarrow ||\hat{\mathbf{a}}_r||$
      $\mathbf{a}_r \leftarrow \hat{\mathbf{a}}_r / \lambda_r$
    end for
    $\hat{\mathbf{B}} \leftarrow \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})(\mathbf{A}^T\mathbf{A} * \mathbf{C}^T\mathbf{C})^\dagger$
    for $r = 1...R$ **do**
      $\lambda_r \leftarrow ||\hat{\mathbf{b}}_r||$
      $\mathbf{b}_r \leftarrow \hat{\mathbf{b}}_r / \lambda_r$
    end for
    $\hat{\mathbf{C}} \leftarrow \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})(\mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B})^\dagger$
    for $r = 1...R$ **do**
      $\lambda_r \leftarrow ||\hat{\mathbf{c}}_r||$
      $\mathbf{c}_r \leftarrow \hat{\mathbf{c}}_r / \lambda_r$
    end for
  until fit ceases to improve or maximum iterations are reached
  return λ, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$

2.5.4 Tucker Decomposition

The Tucker Decomposition was introduced by Tucker [49, 50]. It can be thought of as a form of higher-order PCA; it decomposes a tensor into a *core tensor* multiplied by factor matrices along each mode, which act as the principal components in that particular mode. In Figure 2.9, the Tucker decomposition of a third-order tensor is illustrated; a tensor $\mathcal{X}$ is decomposed into a core tensor $\mathcal{G}$ and the factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$.

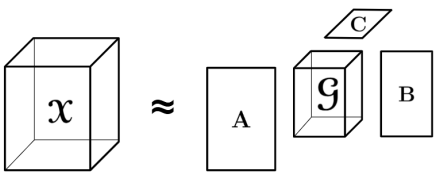


Figure 2.9: Tucker Decomposition of a third-order tensor.

Similarly to CP, the Tucker decomposition for a given third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ can also be formulated as an optimization problem;

$$\min_{\hat{\mathbf{x}}} ||\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}||, \quad \text{where} \quad \hat{\boldsymbol{\mathcal{X}}} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}\ \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r, \tag{2.30}$$

where $\mathbf{a}_p$, $\mathbf{b}_q$ and $\mathbf{c}_r$ are the columns of the factor matrices $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$, and g_{pqr} is an element of the core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{P \times Q \times R}$, for $p = 1, ..., P$, $q = 1, ..., Q$ and $r = 1, ..., R$, which shows the level of interaction between the different components. P, Q and R are the number of components along modes 1, 2 and 3, respectively. Thus, as opposed to CP, where all modes shared the same number of components, in Tucker every mode has its own number of components. Moreover, CP can be thought of as a special version of Tucker, where $P = Q = R$ and $\boldsymbol{\mathcal{G}}$ is superdiagonal, so the components do not interact with each other. If P, Q and R are smaller than I, J, and K (the dimensions of $\boldsymbol{\mathcal{X}}$), the core tensor $\boldsymbol{\mathcal{G}}$ can be thought of as a compressed version of $\boldsymbol{\mathcal{X}}$. Thus, as in PCA, the Tucker decomposition can be used for dimensionality reduction.

As it was highlighted in subsection 2.5.3, the focus of this report is on third-order tensors, but as it is the case with CP, it is also possible to express the Tucker decomposition for a Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$:

$$\boldsymbol{\mathcal{X}} \approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} ... \sum_{r_N=1}^{R_N} g_{r_1 r_2 ... r_N}\ \mathbf{a}^{(1)}_{r_1} \circ \mathbf{a}^{(2)}_{r_2} \circ ... \circ \mathbf{a}^{(N)}_{r_N}. \tag{2.31}$$

The *n-rank* of a tensor $\boldsymbol{\mathcal{X}}$, $\text{rank}_n(\boldsymbol{\mathcal{X}})$, is the column rank of $\mathbf{X}_{(n)}$, i.e., the dimension of the vector space spanned by the mode-n fibers [51]. Similarly to the rank decomposition with the rank of a tensor described in subsection 2.5.3, given a tensor $\boldsymbol{\mathcal{X}}$, an exact Tucker decomposition of rank $(R_1, R_2, ..., R_N)$ where $R_n = \text{rank}_n(\boldsymbol{\mathcal{X}})$ can be found.

Unlike CP, Tucker decomposition is not unique, making it possible to change the core $\boldsymbol{\mathcal{G}}$ without affecting the fit, as long as the inverse modification is applied to the factor matrices.

As in CP, finding the right number of components for the decomposition is not trivial and will be discussed later. There exist many algorithms to compute the Tucker decomposition. The first known algorithm was introduced by Tucker in 1966 [50], and is commonly referred to as Higher-Order SVD (HOSVD). See Kolda and Balder [10] for the detailed procedure of HOSVD. As the name suggests, HOSVD can be thought of as a generalization of SVD to tensors. In 2000, De Lathauwer, De Moor, and Vandewalle came up with an ALS algorithm for computing the Tucker decomposition and called it Higher-Order Orthogonal Iteration (HOOI) [52]. In Algorithm 2, the HOOI for third-order tensors is described, adapted from Kolda and Balder [10].

Algorithm 2 HOOI algorithm to compute the Tucker decomposition with P, Q, R components for a third-order tensor $\mathcal{X}$ of size $I \times J \times K$.

Input: $\mathcal{X}$, P, Q, R
Initialize: $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$ using HOSVD
  repeat
    $\mathcal{Y}^{\mathbf{A}} \leftarrow \mathcal{X} \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T$
    $\mathbf{A} \leftarrow P$ leading left singular vectors of $\mathbf{Y}^{\mathbf{A}}_{(1)}$
    $\mathcal{Y}^{\mathbf{B}} \leftarrow \mathcal{X} \times_1 \mathbf{A}^T \times_3 \mathbf{C}^T$
    $\mathbf{B} \leftarrow Q$ leading left singular vectors of $\mathbf{Y}^{\mathbf{B}}_{(2)}$
    $\mathcal{Y}^{\mathbf{C}} \leftarrow \mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T$
    $\mathbf{C} \leftarrow R$ leading left singular vectors of $\mathbf{Y}^{\mathbf{C}}_{(3)}$
  until fit ceases to improve or maximum iterations are reached
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T$
  return λ, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$

2.5.5 PARAFAC2

PARAFAC2 is a variant of CP proposed by Harshman in 1972 [53]. While CP is applied to a tensor, PARAFAC2 can be applied to a list of matrices with the same number of columns but different numbers of rows. Thus, it relaxes some of the CP constraints in the sense that it allows one of the modes to have varying dimensions; PARAFAC2 applies the same factor matrix along one mode and allows the other factor matrix to vary, as opposed to CP, where both factor matrices are the same across the two modes of a set of parallel matrices.

Let $\mathbf{X}_k$ for $k = 1, ..., K$ be a set of matrices such that each $\mathbf{X}_k$ is of size $I_k \times J$, then the PARAFAC2 model can be expressed mathematically as,

$$\mathbf{X}_k \approx \mathbf{U}_k \mathbf{S}_k \mathbf{V}^T, \qquad \text{for } k = 1, ..., K, \tag{2.32}$$

where $\mathbf{U}_k$ is the varying factor matrix of size $I_k \times R$, $\mathbf{S}_k$ is a diagonal matrix of size $R \times R$, $\mathbf{V}$ is the factor matrix that does not vary of size $J \times R$, and R is the number of components of the decomposition. In Figure 2.10, an example of a PARAFAC2 decomposition is illustrated.

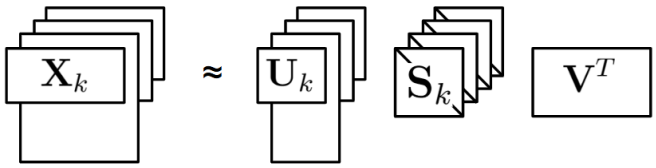


Figure 2.10: PARAFAC2 Decomposition of a set of matrices.

Without imposing additional constraints, PARAFAC2 is not unique. Thus, Harshman [53] forced the cross product $\mathbf{U}_k^T \mathbf{U}_k$ to be constant over k; $\mathbf{U}_k^T \mathbf{U}_k = \boldsymbol{\phi}$ for $k = 1, ..., K$. This is obtained by defining $\mathbf{U}_k$ as,

$$\mathbf{U}_k = \mathbf{Q}_k \mathbf{H}, \qquad \text{for } k = 1, ..., K, \tag{2.33}$$

where $\mathbf{H}$ is a $R \times R$ matrix that does not vary over the slices and $\mathbf{Q}_k$ is an orthonormal matrix of size $I_k \times R$, i.e., $\mathbf{Q}_k^T \mathbf{Q}_k = \mathbf{I}$. This definition makes the cross product to be constant over k as,

$$\mathbf{U}_k^T \mathbf{U}_k = \mathbf{H}^T \mathbf{Q}_k^T \mathbf{Q}_k \mathbf{H} = \mathbf{H}^T \mathbf{H} = \boldsymbol{\phi}. \tag{2.34}$$

Algorithms for computing PARAFAC2 can be divided into two categories: *indirect fitting* [53] and *direct fitting* [54] algorithms. The former finds $\mathbf{V}$, $\mathbf{S}_k$, and $\boldsymbol{\phi}$ by fitting the cross products

$$\mathbf{X}_k^T \mathbf{X}_k \approx \mathbf{V} \mathbf{S}_k^T \mathbf{U}_k^T \mathbf{U}_k \mathbf{S}_k \mathbf{V}^T = \mathbf{V} \mathbf{S}_k^T \boldsymbol{\phi} \mathbf{S}_k \mathbf{V}^T, \qquad \text{for } k = 1, ..., K, \tag{2.35}$$

while the latter first finds $\mathbf{Q}_k$ from a SVD and then updates $\mathbf{H}$, $\mathbf{S}_k$, and $\mathbf{V}$ using one step of the ALS algorithm described in subsection 2.5.3.

2.5.6 Number of Components

As already mentioned, finding the right number of components or *rank* for a tensor decomposition remains a challenge. For CP, most procedures compute multiple CP decompositions with different numbers of components until a good enough fit is found. In this sense, the *reconstruction error* can be used to measure the fit of the decomposition; it can be calculated as the sum of the squared differences between the original and reconstructed tensor, normalized by the sum of squares of the original tensor [55]. Mathematically, the reconstruction error, ε_r, can be defined as

$$\varepsilon_r = \frac{||\mathcal{X} - \hat{\mathcal{X}}||^2}{||\mathcal{X}||^2}, \tag{2.36}$$

where $\mathcal{X}$ is the original tensor and $\hat{\mathcal{X}}$ the reconstructed tensor.

In reality, the data is usually noisy and, as Kolda and Balder [10] suggest, the fit alone should never determine the number of components in the decomposition. Bro and Kiers [56] proposed a new diagnostic, called *core consistency diagnostic* (Core Consistency Diagnostic (CORCONDIA)) to compare different numbers of components in CP. As mentioned in subsection 2.5.4, CP can be thought of as a special case of Tucker, where the core tensor is diagonal. Following this, the idea behind CORCONDIA is to evaluate whether CP with a specific number of components is an appropriate decomposition for a given tensor; if adding off-superdiagonal interactions to the core tensor can improve the fit considerably, then the number of components might be inappropriate or Tucker might be a better model for the given tensor [55]. For a third-order tensor, CORCONDIA can be computed as

$$\text{CORCONDIA} = 100 \left(1 - \frac{\sum_{i=1}^{R} \sum_{j=1}^{R} \sum_{k=1}^{R} (\mathbf{g}_{ijk} - \mathbf{t}_{ijk})^2}{\sum_{i=1}^{R} \sum_{j=1}^{R} \sum_{k=1}^{R} \mathbf{t}_{ijk}^2} \right), \tag{2.37}$$

where $\mathbf{t}_{ijk}$ for $i, j, k = 1, ..., R$ represent the elements of the CP core tensor, this is, $\mathbf{t}_{ijk} = 1$ if $i = j = k$, $\mathbf{t}_{ijk} = 0$ elsewhere, and $\mathbf{g}_{ijk}$ for $i, j, k = 1, ..., R$ represent the elements of the Tucker core given the factor matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$. A large CORCONDIA value suggests that adding interactions between the components from the

different modes does not improve the CP model, and thus CP is an appropriate decomposition [56].

The task of finding the appropriate number of components for Tucker has been addressed by Timmerman and Kiers [57], who proposed a method called DIFFIT, that aimed to find an optimal balance between the fit and the number of components. Given a third-order tensor of size $I \times J \times K$, the DIFFIT method can be summarized in six steps:

1. Determine the fit of all Tucker models with (P, Q, R) components for which $P \leq QR$, $Q \leq PR$ and $R \leq PQ$, up to $P = P_{\max}$, $Q = Q_{\max}$, $R = R_{\max}$. Here, $P_{\max}$, $Q_{\max}$ and $R_{\max}$ represent maximal values that can be chosen *a priori*. The $P \leq QR$ constraints are introduced as the fit for a model with $P > QR$ gives the same fit as a model with $P = QR$ [58].
2. For each value of s, determine the best fit among models for which $P+Q+R = s$. Note that $s = 3, 5, 6, ..., P_{\max} + Q_{\max} + R_{\max}$. 4 is not included as there is no possible combination of P, Q, R such that $P \leq QR$, $Q \leq PR$ or $R \leq PQ$.
3. For every s, compute the difference in fit of s and its predecessor $s-1$, dif_s. Note that for $s = 3$, dif_3 is equal to the fit. Once the dif_s are calculated, compute $dif_{t(m)}$, containing the dif_s values that are as high as their successors. Thus, $t(m)$ indicates the elements of the subset of elements of s for which $dif_s > dif_{s+n}$ for $n = 1, ..., S - s$.
4. Compute $b_{t(m)} = dif_{t(m)}/dif_{t(m+1)}$. This is called the *salience value* of the solution with $t(m)$ components. A large value of $b_{t(m)}$ indicates that the inclusion of $t(m)$ components, instead of $t(m-1)$, increases the fit of the model considerably, while the inclusion of any component beyond the $t(m)$th component hardly improves it.
5. For the $b_{t(m)}$ values, consider only those that satisfy $dif_{t(m)} > ||\mathcal{X}||^2/(s_{\max}-3)$, which is used as a threshold, and select the maximal $b_{t(m)}$.
6. Choose the numbers of components P, Q and R associated with the maximal $b_{t(m)}$.

Timmerman and Kiers [57] advise not to use DIFFIT too rigidly, as there might be more than one interesting solution. Thus, the suboptimal solution(s) should also be considered.

Another approach to determine the numbers of components in the Tucker decomposition is to compute the SVD along each mode [59]. However, since this is a matrix decomposition method, in order to apply it to the tensors, they need to be matricized first along each mode. According to [59], the number of components can be determined from the number of singular values of each mode. However, this number can be as large as the number of elements for said mode, complicating the interpretability of the results. Therefore, the reconstruction error (Equation 2.36) measured between the original and reconstructed matricised tensors obtained using only the first k singular values for each mode can be used.

The task of finding the right number of components for PARAFAC2 has been discussed by Kamstrup-Nielsen, Johnsen and Bro [60], which showed that a PARAFAC2

model can be transformed into a CP model, and by doing so the CORCONDIA values can be computed to evaluate the optimal rank for the decomposition.

As in PARAFAC2, CP can also be formulated in terms of the frontal slices of a tensor. Let $\mathbf{X}_k$ for $k = 1, ..., K$ be the frontal slices of a tensor $\mathcal{X}$ of size $I \times J \times K$, its CP decomposition with R components can be formulated as follows:

$$\mathbf{X}_k \approx \mathbf{A}\mathbf{D}_k\mathbf{B}^T, \qquad \text{for } k = 1, ..., K, \tag{2.38}$$

where $\mathbf{A}$, $\mathbf{B}$ are the factor matrices and $\mathbf{D}_k \equiv diag(c_{k:})$ for $k = 1, ..., K$, this is, $\mathbf{D}_k$ is a diagonal matrix of size $R \times R$ holding the kth row of the factor matrix $\mathbf{C}$ on its diagonal. When comparing this formulation of CP with the PARAFAC2 model in Equation 2.32, the main difference is that, while $\mathbf{A}$ is constant throughout all slices, $\mathbf{U}_k$ depends on k. Substituting $\mathbf{U}_k$ in Equation 2.32 with Equation 2.33, we can formulate the PARAFAC2 model as

$$\mathbf{X}_k \approx \mathbf{Q}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^T, \qquad \text{for } k = 1, ..., K. \tag{2.39}$$

We can then multiply $\mathbf{Q}_k^T$ on the left side of both terms,

$$\mathbf{Q}_k^T\mathbf{X}_k = \mathbf{Y}_k \approx \mathbf{Q}_k^T\mathbf{Q}_k\mathbf{H}\mathbf{S}_k\mathbf{V}^T = \mathbf{H}\mathbf{S}_k\mathbf{V}^T, \qquad \text{for } k = 1, ..., K, \tag{2.40}$$

as $\mathbf{Q}_k$ is orthonormal for $k = 1, ..., K$. This transformation allows us to approximate the PARAFAC2 model to a CP model and use CORCONDIA to find the optimal number of components.

3
Methods

This chapter describes the methods used in this project; it begins with the data description and its transformation into a tensor, followed by the generation of simulated data. Then, the different functions used for the decomposition and the methods used for the selection of the number of components are presented, as well as the immunology metrics analysis. Each procedure is explained in the following sections. The code was fully implemented in Python, which can be found in `https://github.com/pablovadillo/Tensordecomposition/tree/main`.

3.1 Dataset Description

The data used in this project was collected by Sahlgrenska University Hospital. It consists of TCR clone counts (both in frequency and absolute values) from blood samples of 16 patients with sarcoma over time obtained by the ultrasensitive technique described in section 2.3 [7]. In particular, of the 16 patients, 3 suffered from Ewing sarcoma, another 3 from Myxoid liposarcoma, 2 from Leiomyosarcoma, another 2 from Myxofibrosarcoma, 2 others from high-grade undifferentiated sarcoma, and the remaining 4 suffered from: Angiosarcoma, Osteogenic fibrosarcoma, Osteosarcoma, and Pleomorphic sarcoma.

Each row of the dataset consists of the following fields: patient ID, sarcoma subtype, TCR sequences (identified by their nucleotide and amino acid sequences, as well as the V, D, and J segments), and the counts of the TCR sequences over time. The number of samples over time differs from patient to patient, and the samples were not taken in constant time intervals. Thus, some patients inherently suffer from a higher time resolution than others due to a higher number of time points for a similar number of days. For 14 of the 16 patients, the first sequencing was performed before treatment. A brief summary of this data can be found in Table 3.1.

Another dataset with patient outcomes was also used. However, this clinical data was hidden from us until the analysis was finished to avoid biases. This new dataset contained information about each studied time point, more specifically, the type of systemic treatment administered, what clinical response was observed, whether they had surgery and/or radiotherapy on that day, and whether the patient passed away. The systemic treatments are reduced to: Off-treatment, Adjuvant, Neoadjuvant, Palliative targeted, and Palliative chemotherapy. Note that individual patients may

Table 3.1: Dataset description summary. For each patient, their sarcoma subtype, their total number of sequences throughout all the samples, the number of days passed from the first sequencing to the last and the total number of samples (or data points) are specified. The ordering of the patients corresponds with the ordering in the data. In bold, the two patients whose first sequencing was performed after starting treatment.

Patient ID	Sarcoma subtype	Sequences	Days	Data points
SARK012	Leiomyosarcoma	622	843	15
SARK018	Angiosarcoma	101	186	10
SARK019	Leiomyosarcoma	2539	202	10
SARK034	Osteogenic fibrosarcoma	22433	951	34
SARK017	Ewing sarcoma	605	9	3
SARK021	High grade undifferentiated sarcoma	924	1187	7
SARK025	Osteosarcoma	14721	462	19
SARK041	Myxofibrosarcoma	699	77	5
SARK011	**Ewing sarcoma**	**3121**	**178**	**5**
SARK027	High grade undifferentiated sarcoma	301	232	5
SARK029	Myxofibrosarcoma	357	355	4
SARK032	Myxoid liposarcoma	1224	223	5
SARK033	Myxoid liposarcoma	258	93	4
SARK037	Ewing sarcoma	683	112	5
SARK052	**Myxoid liposarcoma**	**333**	**57**	**3**
SARK059	Pleomorphic sarcoma	825	317	11

have received multiple types of systemic treatment over the course of their therapy. The labeled clinical responses are: progressive disease (PD), meaning that cancer has grown since before; stable disease (SD), meaning that cancer stayed roughly the same; partial response (PR), meaning that cancer reduced somewhat; and complete response (CR), meaning that cancer seems mostly eliminated at the moment. The time evolution of the clinical response for each included patient can be found in Appendix B.3.

3.2 Preprocessing and Tensor Construction

The first step in the preprocessing was to look for missing values in the data. In the case of the absolute counts, only the data for a patient presented missing values, specifically in the first columns (first sequence). We assumed that the missing values represented zero values for two reasons; the counts of the missing TCR sequences in the remaining samples were not that significant, and the frequency values associated with the missing absolute values were also zero. Analogously for the frequencies, only one value was missing, also corresponding to the first sequence of the same patient. Again, the count value corresponding to the missing frequency was zero, so a zero-imputation approach was taken.

For the construction of the tensors, we used the frequency values instead of the absolute counts to prevent patients with the highest sequence counts from disproportionately influencing the decomposition. Therefore, this approach emphasized the relative changes in TCR sequences over time. The two patients who had their first sequencing performed after treatment were ultimately discarded due to the difficulty in temporally aligning them with the other patients. In addition, since the patients had a different number of samples and were taken at varying intervals of time, the initial challenge of the project was to determine how to unify the time dimension across patients for the construction of the tensor.

To handle these differences in sampling times and durations across patients, the time dimension was standardized through linear interpolation, ensuring that each patient had data available at consistent 10-day intervals. Based on this unified time axis, multiple tensors were constructed, each representing a different trade-off between time span and patient availability. Specifically, tensors were built for 70, 100, 200, and 300 days, containing 13, 11, 9, and 6 patients respectively. All subsequent analyses were performed independently on each of these tensors. Hence, if a patient had data up to Day 170, they would be included in the 70-day and 100-day groups, using only the data available up to those respective time points. However, they would not be included in the 200-day or 300-day groups, as their follow-up data did not extend that far.

To construct the four tensors, patient data were stacked along the third mode of each new tensor, corresponding to the different day groups: Mode 1 represents the top 100 TCR sequences, ranked by their pre-treatment counts; Mode 2 corresponds to the time dimension, with samples interpolated at 10-day intervals; and Mode 3 captures the individual patients. The tensors were named after the number of days they had data for, that is, *tensor_70*, *tensor_100*, *tensor_200*, and *tensor_300*. The dimensions of the final tensors were $100 \times 8 \times 13$ for *tensor_70*, $100 \times 11 \times 11$ for *tensor_100*, $100 \times 21 \times 9$ for *tensor_200*, and $100 \times 31 \times 6$ for *tensor_300*.

A fifth array with varying lengths in time was also constructed for PARAFAC2; it consisted of a list of 13 matrices, one for each patient, with the first mode corresponding to the time dimension and the second mode to the top 100 TCR sequences. Note that the matrices were transposed to meet the definition of the PARAFAC2 model, where each matrix is of size $I_k \times J$, with I_k varying in size for $k = 1, ..., K$. In this case, each matrix was of size $I_k \times 100$, with I_k being the time steps for each patient $k = 1, ..., 13$.

Finally, it was decided to center the tensors across the time mode (Mode 2 for the tensors, and Mode 1 for the PARAFAC2 array). The centering across the second mode for a third-order tensor can be computed as follows,

$$x_{ijk}^{cent} = x_{ijk} - \frac{\sum_{j=1}^{J} x_{ijk}}{J} \tag{3.1}$$

where J stands for the dimension of the second mode of said tensor. This computation is often referred to as *single centering* [61, 62]. In the case of the PARAFAC2

array, every matrix was centered across the first mode as follows,

$$x_{ijk}^{cent} = x_{ijk} - \frac{\sum_{i=1}^{I_k} x_{ijk}}{I_k}, \tag{3.2}$$

where I_k represents the time dimension for each matrix, with varying length for $k = 1, ..., 13$. The choice of centering across the time mode was to remove any constant level to allow the decomposition to focus solely on relative changes over time for all sequences and patients.

To prevent those matrices with the longest time-steps in the PARAFAC2 array from dominating the decomposition, the data was scaled across every matrix as follows,

$$x_{ijk}^{scal} = \frac{x_{ijk}}{\sqrt{(\sum_{i=1}^{I_k} \sum_{j=1}^{J} x_{ijk}^2)}}, \tag{3.3}$$

which is known as *scaling within* the third mode in third-order tensors [61]. In our case, the scaling occurs within every matrix, with the first mode I_k varying in length.

The combination of centering and scaling in the PARAFAC2 data is not problematic, as they were performed within different modes and, as Bro and Smilde [62] suggest, the centering was performed before scaling. In Table 3.2, a summary of the created arrays, with their dimensions and preprocessing information can be found.

Table 3.2: Summary of the created arrays.

Name	Array type	Dimension	Centered	Scaled
tensor_70	tensor	$100 \times 8 \times 13$	Yes	No
tensor_100	tensor	$100 \times 11 \times 11$	Yes	No
tensor_200	tensor	$100 \times 21 \times 9$	Yes	No
tensor_300	tensor	$100 \times 31 \times 6$	Yes	No
parafac2_data	list of matrices	$I_k \times 100 \times 13$	Yes	Yes

3.3 Simulated data

We generated synthetic data consisting of two tensors and two PARAFAC2 arrays. This served two purposes: to validate that the methods are capable of capturing meaningful patterns and to understand the appropriateness of each method. In Table 3.3, a summary of the generated arrays can be found.

The two simulated tensors, named *tensor_1* and *tensor_2*, were made of size $100 \times 10 \times 10$, with dimensions corresponding to the TCR sequences, time, and patients. *Tensor_1* was created to be more suitable for a CP decomposition, that is, with components not interacting much with each other, i.e., with little off-superdiagonal interaction. In contrast, *tensor_2* was created to be more suitable for Tucker, with clear interactions between its components. Both tensors were

Table 3.3: Summary of the generated arrays.

Name	Array type	Dimension	Purpose
tensor_1	tensor	$100 \times 10 \times 10$	Good for CP
tensor_2	tensor	$100 \times 10 \times 10$	Bad for CP, good for Tucker
tensor_3	list of matrices	$I_k \times 100 \times 10$	Good for PARAFAC2
tensor_4	list of matrices	$I_k \times 100 \times 10$	Bad for PARAFAC2

initialized with random noise, sampled from a normal distribution of mean 0 and standard deviation 0.1. To ensure non-negativity, the absolute value of each sampled number was taken.

The patients were labeled with numbers from 1 to 10 according to their index in the tensor. In both tensors, patients were divided into three groups; Group A consisted of patients 1, 4, 6, Group B of patients 2, 8, 10, and Group C of patients 3, 5, 7, 9. Each group was defined slightly differently in *tensor_1* and *tensor_2*.

Starting with *tensor_1*, patients in Group A had their first two TCR sequences increase over time linearly, following

$$x_{ijk} = 0.2j + |\mathcal{N}(0, 0.1)|, \tag{3.4}$$

with $i = 1, 2$, $j = 1, 2, ..., 10$, $k = 1, 4, 6$, and $|\mathcal{N}(0, 0.1)|$ being the absolute value of the noise. Patients in Group B had their third and fourth TCR sequences increase over time with polynomial growth following

$$x_{ijk} = 0.003(j-1)^3 + |\mathcal{N}(0, 0.1)|, \tag{3.5}$$

with $i = 3, 4$, $j = 1, 2, ..., 10$ and $k = 2, 8, 10$. Last but not least, patients in Group C had their fifth to eighth sequences increase during the first 5 time points and subsequently decrease during the last 5 time points, following a parabola like pattern,

$$x_{ijk} = \begin{cases} 0.1 \cdot 5^2 - 0.1 \cdot (6-j)^2 + |\mathcal{N}(0, 0.1)|, & \text{for } j = 1, 2, 3, 4, 5. \\ 0.1 \cdot 5^2 - 0.1 \cdot (j-5)^2 + |\mathcal{N}(0, 0.1)|, & \text{for } j = 6, 7, 8, 9, 10, \end{cases} \tag{3.6}$$

with $i = 5, 6, 7, 8$ and $k = 3, 5, 7, 9$. Note that the selection of the coefficients in the equations is not arbitrary, as they were chosen to have all groups within similar ranges; Equation 3.4 reaches a maximum of 2 at $j = 10$, Equation 3.5 reaches a maximum of 2.19 at $j = 10$, and Equation 3.6 reaches a maximum of 2.4 at $j = 5, 6$. In Figure 3.1 the evolution of the TCR sequences over time of patients 6, 10 and 9 in *tensor_1* is illustrated. As we can observe, the groups in *tensor_1* do not interact much with each other; each group involves different TCR sequences and patients, and presents different time patterns.

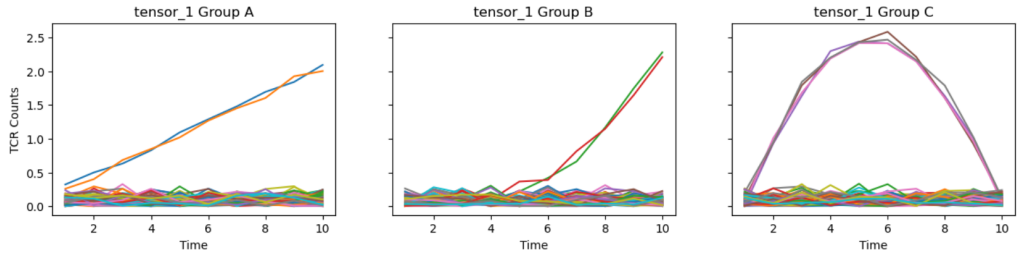


Figure 3.1: Examples of TCR sequence temporal trends of patients from Groups A, B and C in *tensor_1*.

For *tensor_2*, patients in both Group A and Group B were defined using the same pattern as Group A in *tensor_1* (see Equation 3.4), applied to sequences $i = 1, 2$ in Group A, and sequences $i = 3, 4$ in Group B. Group C was defined analogously to Group C in *tensor_1*, in Equation 3.6, but for the first four sequences, $i = 1, 2, 3, 4$. In Figure 3.2 the evolution of the TCR sequences over time of patients 6, 10 and 9 in *tensor_2* is illustrated. In contrast to *tensor_1*, *tensor_2* shows clear interactions between the groups; Groups A and B share the same time pattern, while Group C involves the TCR sequences in both Groups A and B.

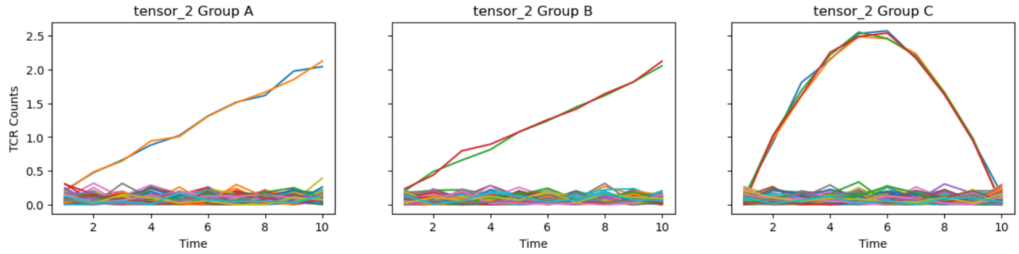


Figure 3.2: Examples of TCR sequence temporal trends of patients from Groups A, B and C in *tensor_2*.

The two PARAFAC2 arrays, although technically not tensors, were labeled *tensor_3* and *tensor_4*. They consisted of a list of 10 matrices, one for each patient, of sizes $I_k \times 100$, where I_k denoted the time dimension, varying for $k = 1, ..., 10$, and the second mode corresponded to the TCR sequences. *tensor_3* was created without much interaction between the two invariant modes, i.e. TCR sequences and patients, while *tensor_4* was created with interactions between the invariant modes, to see the differences in their PARAFAC2 decomposition. As mentioned in section 3.2, modes 1 and 2 were swapped to meet the PARAFAC2 requirements. Patients were labeled analogously to the tensors *tensor_1* and *tensor_2*, and distributed to the same groups A, B and C. The number of time steps for each of the matrices was randomly generated between 3 and 20 and then multiplied by 2, to ensure an even number of time steps, for the definition of Group C, as we will see subsequently.

For both *tensor_3* and *tensor_4*, Group A was defined analogously to Group A in *tensor_1* and *tensor_2*, described in Equation 3.4, for $i = 1, ..., I_k$, $j = 1, 2$, and $k = 1, 4, 6$. Note that i and j are swapped here. Group B was also defined

analogously for both PARAFAC2 arrays to Group B in *tensor_1*, described in section 3.3, for $i = 1, ..., I_k$, $j = 3, 4$, and $k = 2, 8, 10$. Group C, similarly to Group C in *tensor_1* and *tensor_2*, was defined as follows:

$$x_{ijk} = \begin{cases} 0.1 \cdot (\frac{I_k}{2})^2 - 0.1 \cdot (\frac{I_k}{2} + 1 - j)^2 + |\mathcal{N}(0, 0.1)|, & \text{for } j = 1, ..., \frac{I_k}{2}. \\ 0.1 \cdot (\frac{I_k}{2})^2 - 0.1 \cdot (j - \frac{I_k}{2})^2 + |\mathcal{N}(0, 0.1)|, & \text{for } i = \frac{I_k}{2} + 1, ..., I_k, \end{cases} \tag{3.7}$$

with $k = 3, 5, 7, 9$, $j = 5, 6, 7, 8$ for *tensor_3* and $j = 1, 2$ for *tensor_4*. That is, the pattern defined in Group C was applied to the fifth to eight sequences in *tensor_3* and to the first two sequences in *tensor_4*. In Figure 3.3 the evolution of the TCR sequences over time of patients 6, 10 and 9 in *tensor_3* and patients 6, 8 and 5 in *tensor_4* are illustrated.

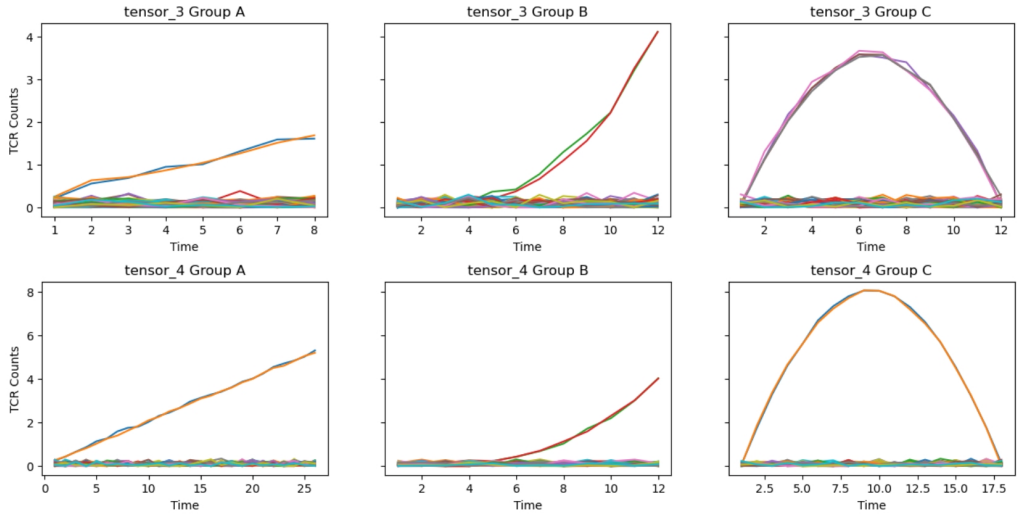


Figure 3.3: Examples of TCR sequence temporal trends of patients from Groups A, B and C in *tensor_3* and *tensor_4*.

In *tensor_3* each group involves different TCR sequences and patients, while in *tensor_4* groups A and C both involve the first two TCR sequences.

Because the frequency values were used in the real data, the simulated data was transformed into frequencies by dividing every element by the sum of the elements on its row. The simulated data was also centered across the time mode (Mode 2 for the tensors, and Mode 1 for the PARAFAC2 arrays). Moreover, the PARAFAC2 arrays were scaled across every matrix, to be consistent with the preprocessing of the real data.

3.4 Decomposition models

To analyze both the simulated and real datasets, three tensor decomposition models were applied: CP, Tucker, and PARAFAC2. All decompositions were implemented using the `TensorLy` library [63], specifically the `parafac`, `tucker`, and `parafac2` functions from the `tensorly.decomposition` module. These functions compute

the respective decompositions using established algorithms: ALS for CP, HOOI for Tucker, and the direct fitting approach for PARAFAC2. Detailed descriptions of the theoretical foundations for these models are provided in subsection 2.5.3, subsection 2.5.4, and subsection 2.5.5. The factor matrices were initialized using the SVD in all methods.

The primary goal of applying these decomposition models was to uncover latent structure in the data, including temporal patterns and relationships between patients and TCR sequences. For both simulated and real tensors, multiple decompositions were computed using different numbers of components. The final model selection was based on how well the decompositions captured relevant patterns, assessed primarily through visual comparison with the original (interpolated) data.

Although non-negative constraints can enhance interpretability (especially for Tucker decomposition, as suggested by Gillet et al. [59]) such constraints were not compatible with our data due to preprocessing steps involving centering, which introduced negative values. As a result, standard decomposition was applied throughout.

All decomposition models were evaluated across tensors of varying lengths and patient counts (as described in section 3.2), and all downstream analyses were performed independently for each tensor.

3.5 Selection of number of components

The first step in computing the decomposition of a tensor is to decide on the number of components to use. As mentioned above, this is not a trivial task and, therefore, several methods should be considered to make a final decision. In general, the nature of the data and the interpretability of the results should also be taken into account when deciding the number of components.

For CP, the reconstruction error was plotted for different numbers of components and the elbow method was used to select the optimal value. The function `KneeLocator` from the `kneed` Python package [64] was used to compute the elbow of the curves. However, as mentioned in subsection 2.5.6, the reconstruction error should not be used as the only measure to find the optimal number of components due to the noisy nature of the data, so the CORCONDIA value was also calculated and plotted for all tensors and different numbers of components. CORCONDIA was automatically calculated using the library `TensorLy-Viz` [65], which enables to compute different metrics for CP models, including CORCONDIA.

For Tucker, two approaches were used. First, the DIFFIT method was implemented for all tensors to find the numbers of components that considerably increased the fit. Secondly, the SVD with the reconstruction error was also applied, but this time to the tensors matricised along each mode. Therefore, the reconstruction error was measured between the original and reconstructed matricised tensors obtained using

only the first k singular values for each mode. This metric quantifies the relative difference between the original and reconstructed tensors, with lower values indicating better approximations. Thresholds from 0.10 to 0.40 in 0.05 increments were applied to find the number of components.

For PARAFAC2, as with CP, both the reconstruction error with the elbow method and CORCONDIA were used to find the optimal number of components. For CORCONDIA, the transformation described in subsection 2.5.6 was applied.

3.6 Clustering with Tucker Decomposition

Tucker decomposition can also be used for clustering by applying a clustering algorithm to one of the resulting factor matrices, which typically would be the one corresponding to the dimension of interest [59]. In this work, clustering was performed on the patient factor matrix from each tensor decomposition to identify groups of patients with similar temporal or immunological profiles.

Following the approach in [59], the clustering algorithm chosen was K-medoids [66], which has several advantages over K-means. Unlike K-means, which computes centroids as the mean of cluster points, K-medoids selects actual data points (medoids) as cluster centers. This makes it more robust to noise and outliers [67, 68]. Furthermore, K-medoids minimizes the sum of distances between each point and the medoid of its cluster (Within-Cluster Sum of Distances (WCSD)), whereas K-means minimizes the within-cluster sum of squared distances.

This algorithm was implemented using the built-in function `kmedoids` from the `pyclustering` library [69]. More specifically, firstly, a set of initial medoids was selected randomly from the data points. Then, a pairwise distance matrix was computed using `calculate_distance_matrix` function from the same library, which computes the Euclidean distance, which was passed to the `kmedoids` function with the argument `data_type='distance_matrix'` to indicate that a precomputed distance matrix was being used. With this, performance was increased [70]. After running the `process()` method to perform the clustering, the final cluster assignments and medoid indices were extracted. Labels were assigned to each data point based on the cluster it belonged to, enabling further downstream analysis. The function returned both the cluster labels and the indices of the final medoids.

Since the goal is to cluster the patients without using the ground truth, the number of clusters k was defined in an unsupervised manner. To this end, a range of different k from 2 to 9 (from 2 to 6 in the *tensor_300*, as the number of clusters cannot be greater than the number of patients) were tested. Using the elbow method to the WCSD for changing k, the optimal number of clusters was found. This was done automatically using the `KneeLocator` from the `kneed` library.

The WCSD was calculated as follows:

$$WCSD = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i) \tag{3.8}$$

where k is the number of clusters, C_i is the set of points in cluster i, m_i is the medoid (representative point) of cluster i, and $d(x, m_i)$ is the distance between point x and medoid i.

To evaluate the resulting clustering, the found patient groups were compared to the ground-truth patient outcomes. Since these are defined for each time point for each patient, only the label for the last data point was considered for simplicity. As cluster labels are arbitrary (i.e., the numeric label assigned to each group has no inherent meaning), a label-invariant comparison metric was needed. For this purpose, the Normalized Mutual Information (NMI) was used, computed with the `normalized_mutual_info_score` function from the `sklearn.metrics` Python library [71].

NMI measures the agreement between two clusterings of the same dataset by quantifying the amount of shared information between the label assignments. It normalizes this value to fall between 0 and 1, where 0 indicates completely independent groupings and 1 reflects perfect alignment. Since NMI is invariant to label permutations and symmetric, it is ideal for comparing clustering strategies. It is defined as:

$$NMI(U, V) = \frac{MI(U, V)}{mean(H(U), H(V))} = \frac{\sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \log\left(\frac{P(i,j)}{P(i)P'(j)}\right)}{mean(H(U), H(V))} \tag{3.9}$$

where $MI(U, V)$ is the mutual information between two label assignments (U and V) of the same N objects. $H(U)$ and $H(V)$ represent the uncertainty (entropy) of each partition set, defined as $H(U) = -\sum_{i=1}^{|U|} P(i) \log(P(i))$, and analogously for $H(V)$. Here, $P(i) = |U_i|/N$ is the probability that a randomly selected object from U belongs to class i (similarly for V), and $P(i, j) = |U_i \cap V_j|/N$ is the probability that an object belongs to both class i in U and class j in V [71].

3.7 Immunology metrics

Besides tensor decomposition, common immunology metrics in the field were also explored. Namely, Shannon entropy (Equation 2.1), Inverse Simpson index (Equation 2.4), adaptive Gini coefficient (Equation 2.5), Gini-Simpson index (Equation 2.7), DE_{50} score (Equation 2.8), clonality (Equation 2.2), and Pielou's evenness index (Equation 2.3), all described previously. All the metrics' computation was done with our own code, with the exception of Shannon entropy, which was obtained from the Python function `entropy` from the `scipy.stats` library [72]. These were computed using the original frequencies for all TCR sequences for each original time point (i.e., without interpolation). Then, the difference between before treatment

(first data point labeled *Day 0*) and after treatment (the last data point known to us) was also computed.

From these metrics, the same clustering pipeline as in section 3.6 was applied. Since there was no patient factor matrix, the specific data used for this clustering was an array containing the metrics' values for the last time point of every patient. By considering only this last point, our objective was to capture the immune state of each patient after treatment, which is most relevant when trying to relate it to clinical outcomes. Including baseline values or using the difference between the first and last time points were considered. However, these options could introduce noise, especially when some metrics start very low, making small changes appear overly dramatic and were not implemented. Also, adding both time points (i.e., pre- and post-treatment) would increase the number of features without necessarily improving how well the clustering reflects biologically meaningful differences. The final time point alone offered a simpler and more direct way to group patients based on how their TCR repertoires looked after treatment. The resulting patient groups were also compared with those from the Tucker decomposition-based clustering and the ground-truth with NMI as well.

4

Results

This section presents the results obtained in this research. In particular, the results from the different methods to select the number of components will be shown, as well as the decomposition results. The results of the simulated and real data will be presented simultaneously, and the results of the immunology metrics will also be shown. To avoid confusion, the number of components for the decompositions will be written as (Mode 1, Mode 2, Mode 3), and the specific elements in the core tensor as [Mode 1, Mode 2, Mode 3].

4.1 Tensor construction

Figure 4.1 shows the distribution of the time samples for each patient. With this, an idea of the level of interpolation used to obtain the tensors is provided. For instance, patient *SARK025* has a considerable number of data points, making the interpolation more trustworthy than in the case of patient *SARK029*, who has too few points, forcing the interpolation to follow a more linear trajectory and potentially miss relevant variation not captured by the dataset.

To handle differences in sampling times and durations across patients, the time dimension was standardized through linear interpolation, ensuring each patient had data available at consistent 10-day intervals. Based on this unified time axis, multiple tensors were constructed, each representing a different trade-off between time span and patient availability. Specifically, tensors were built for 70, 100, 200, and 300 days, containing 13, 11, 9, and 6 patients, respectively. All subsequent analyses were performed independently on each of these tensors.

An important issue to note here is that this time division comes as a trade-off between capturing meaningful trend patterns and retaining as many patients as possible. The longer the time span included, the fewer patients are eligible due to limited sampling duration. An example of this is shown in Figure 4.2, while a broader overview of the effect across all patients can be found in section B.1.

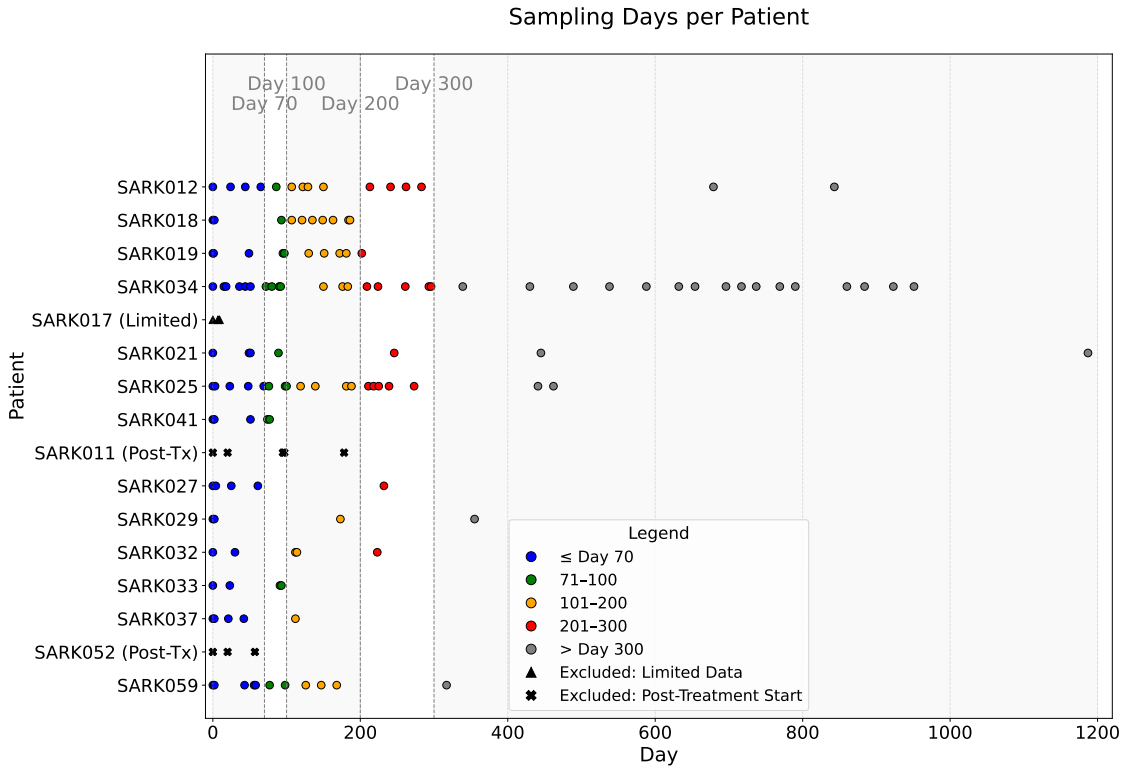


Figure 4.1: Sampling data-points for all patients. Those excluded are indicated but shown nonetheless for whole data comparison.

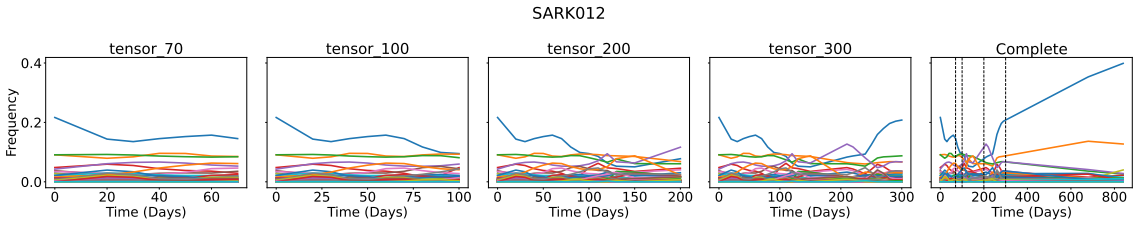


Figure 4.2: Sequence temporal trends retained in each tensor and in whole temporal data from patient *SARK012*, which has more temporal data than kept in the tensors, hence showing the information trade-off. On the rightmost plot, the time splits are marked in vertical dashed lines for days 70, 100, 200, and 300.

4.2 CP Decomposition

4.2.1 Selection of Number of Components

Starting with the simulated tensors, the first step to select the optimal number of components for CP was carried out by plotting the reconstruction error for $R = 1, 2, 3, 4, 5, 6, 7$. Note that R stands for the number of components. In Figure 4.3, the reconstruction errors for the different values of R are shown for the two simulated tensors.

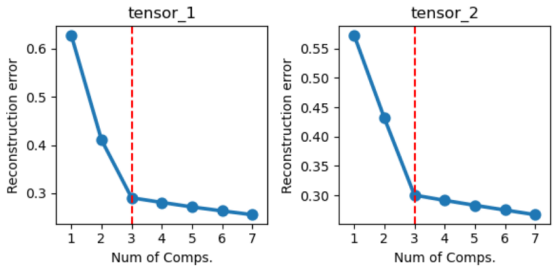


Figure 4.3: Plots of the reconstruction errors for $R = 1, 2, 3, 4, 5, 6, 7$ for the simulated tensors. The dotted red line represents the elbow.

As expected, the reconstruction error decreases substantially until $R = 3$, after which the improvement becomes marginal. This aligns with the simulated tensors being generated from 3 groups. At $R = 3$, the reconstruction errors are 29.07% for *tensor_1* and 30.05% for *tensor_2*. Although it may seem like significant errors, as we can observe from the graphs, an increase in R beyond 3 is just noise fitting.

Meanwhile, the CORCONDIA values for both tensors and $R = 1, 2, 3, 4, 5, 6, 7$ were also calculated. The CORCONDIA values can help determine whether CP is a suitable model for each of the tensors and the different values of R. In Figure 4.4, the CORCONDIA values for the different numbers of components are shown for the simulated tensors.

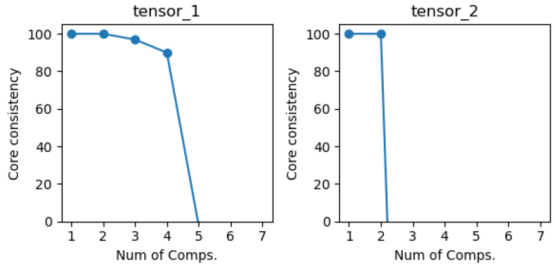


Figure 4.4: Plots of the CONCORDIA values for $R = 1, 2, 3, 4, 5, 6, 7$ for the simulated tensors.

As expected, the CORCONDIA value is 100 for $R = 1$ for all tensors. This is always the case for $R = 1$, as a core tensor of shape $1 \times 1 \times 1$ does not have off-superdiagonal elements. A CORCONDIA value above 90 can be interpreted as very *trilinear*, meaning little off-superdiagonal variation, while a CORCONDIA value around 50 means equal variation in both the superdiagonal and off-superdiagonal elements, which could be problematic. A negative CORCONDIA value implies an invalid model [56]. In this sense, for *tensor_1*, a CP decomposition up to $R = 4$ would be appropriate, while for *tensor_2* a CP decomposition with $R > 2$ would be invalid. This is not surprising, as *tensor_2* was generated to have interactions between the components.

In the end, $R = 3$ was chosen for the CP decomposition of *tensor*_1, and it was decided not to perform a CP decomposition on *tensor*_2, as the CORCONDIA

values suggested that a CP decomposition with $R = 3$ would not be valid.

The same process was performed for the real tensors, with the results of the reconstruction error illustrated in Figure 4.5. The elbow method selects $R = 3$ as the optimal number of components for *tensor_70*, *tensor_100* and *tensor_300*, with reconstruction errors of 33.26%, 39.57% and 35.50% respectively, and $R = 2$ with a reconstruction error of 31.22% for *tensor_200*. In all cases, the errors are comparable to those in the simulated data.

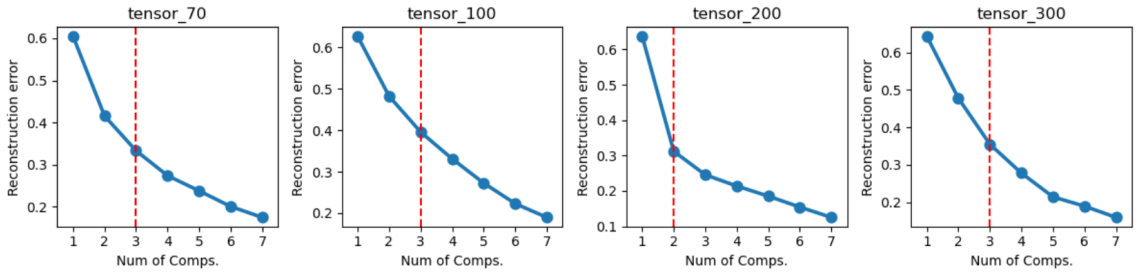


Figure 4.5: Plots of the reconstruction errors for $R = 1, 2, 3, 4, 5, 6, 7$ for all tensors. The dotted red line represents the elbow.

The CORCONDIA values of the real tensors for $R = 1, ..., 7$ are shown in Figure 4.6. $R = 2$ would give valid CP decompositions for *tensor_70*, *tensor_100* and *tensor_200*. In the case of *tensor_300*, $R = 3$ has a CORCONDIA value of 65.93, suggesting that $R = 3$ still provides an acceptable CP model for this tensor.

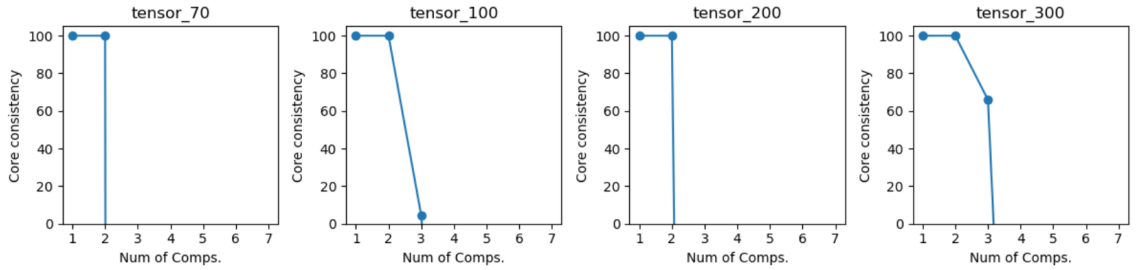


Figure 4.6: Plots of the CONCORDIA values for $R = 1, 2, 3, 4, 5, 6, 7$ for the real tensors.

In summary, for *tensor_200* and *tensor_300*, both the elbow method with the reconstruction errors and the CORCONDIA values suggest $R = 2$ and $R = 3$ to be the optimal number of components, respectively. For *tensor_70* and *tensor_100*, however, the results differ; the elbow method suggests $R = 3$ for both tensors, but the CORCONDIA values indicate that $R = 3$ produces invalid models. Thus, $R = 2$ was chosen as the optimal number of components for these tensors. However, it should be noted that the reconstruction error for $R = 2$ is quite large for both *tensor_70* and *tensor_100*, with values of 41.54% and 48.20% respectively. The discrepancy between CORCONDIA and the elbow method could indicate that CP is not a suitable model for *tensor_70* and *tensor_100*.

4.2.2 Decompositions

The CP decomposition for the simulated tensor *tensor_1* is shown in Figure 4.7. As we can observe, groups A, B, and C are clearly described by components 3, 2, and 1, respectively. The CP decomposition for *tensor_2* with $R = 3$ also correctly identifies the three groups. As mentioned earlier, it is not a valid decomposition and thus it is not included in the report. However, it can be found in Appendix C.1.

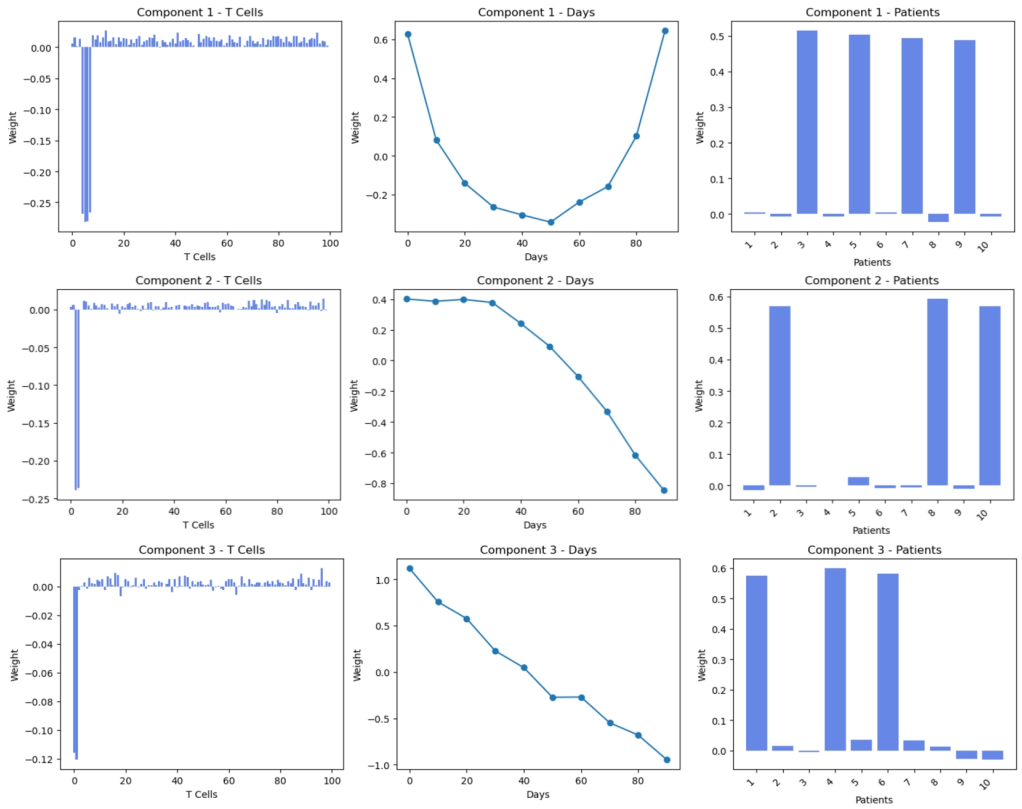


Figure 4.7: CP decomposition for *tensor_1* with $R = 3$.

The real tensors were also decomposed by CP, with the number of components selected in subsection 4.2.1. The decompositions are shown in Figure 4.8 for *tensor_70*, Figure 4.9 for *tensor_100*, Figure 4.10 for *tensor_200*, and Figure 4.11 for *tensor_300*.

The first component in the decomposition of *tensor_70* is mainly described by the first TCR sequence of patient *SARK027*, while the second component is explained by the first two sequences of patient *SARK041* and the second sequence of patient *SARK018*.

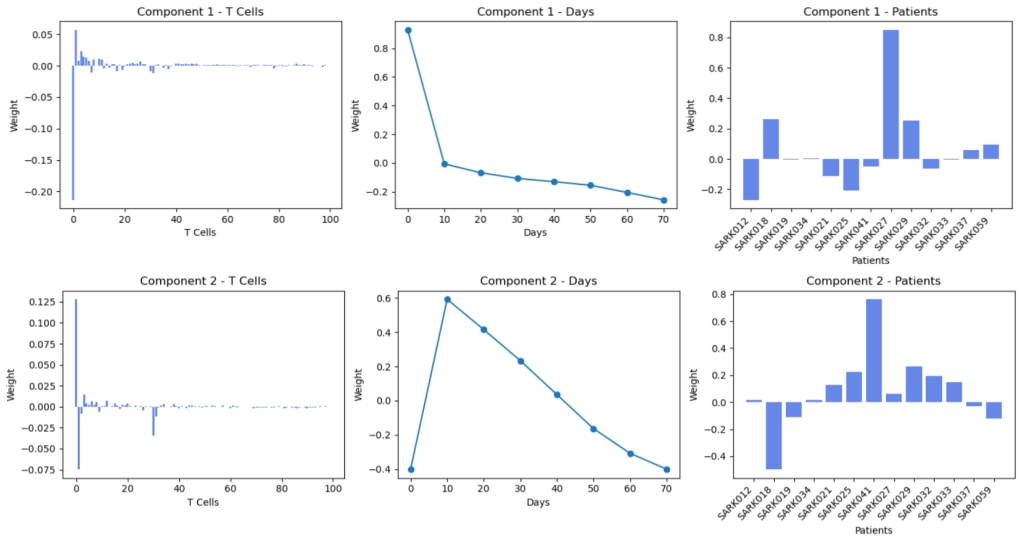


Figure 4.8: CP decomposition for *tensor_70 with* $R = 2$.

Analogously to *tensor_70*, the first component in the decomposition of *tensor_100* is mainly explained by the first sequence of patient *SARK027*, while the second component is explained by the second sequence of patients *SARK018* and *SARK037*.

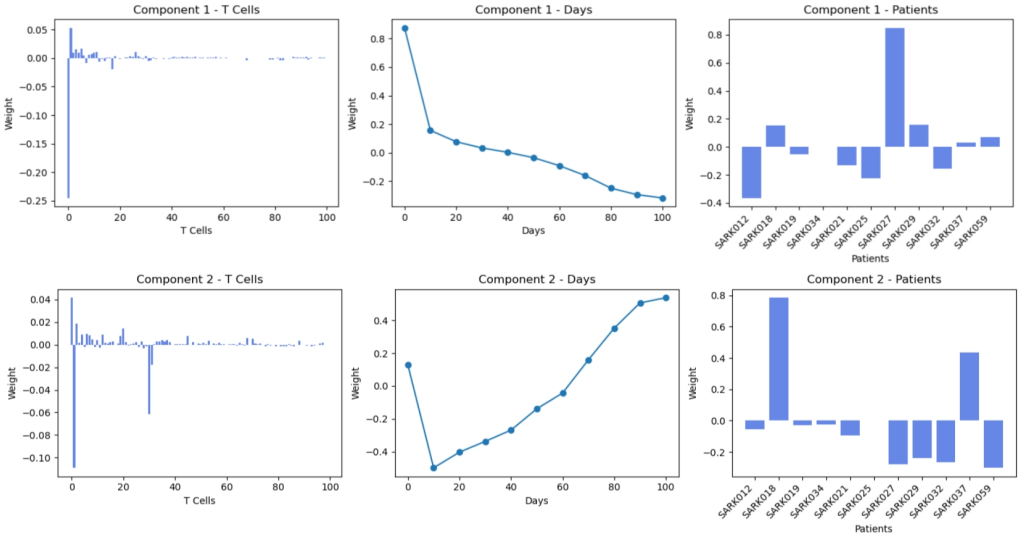


Figure 4.9: CP decomposition for *tensor_100 with* $R = 2$.

For *tensor_200*, its first component is mainly explained by the first TCR sequence, with a significant weight in patients *SARK027* and *SARK012*. Meanwhile, the second component is explained mainly by the patient *SARK019*, with its third sequence showing the described time pattern.

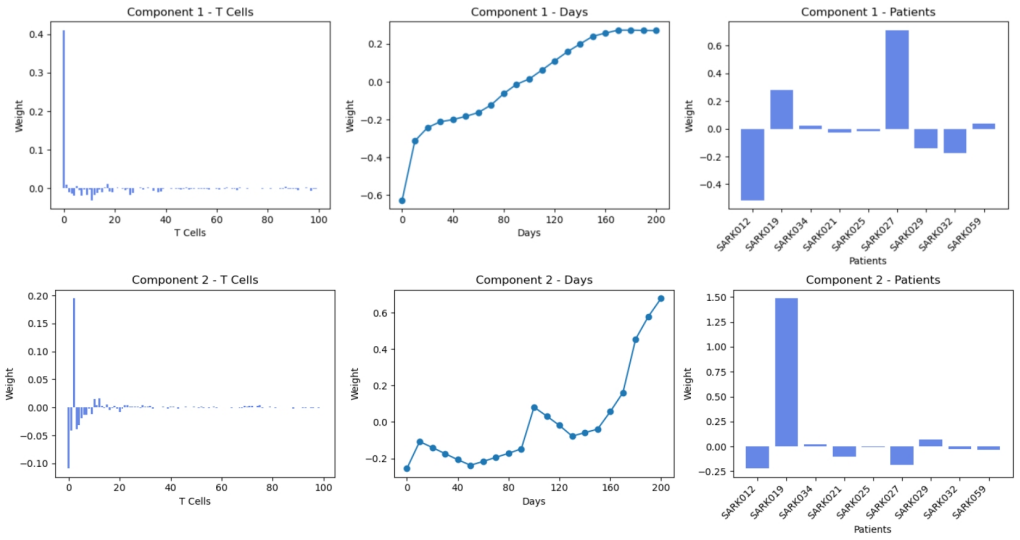


Figure 4.10: CP decomposition for *tensor_200 with* $R = 2$.

In the case of *tensor_300*, the first and third components are dominated by patient *SARK012*. In particular, its first sequence is explained by combining both time patterns. The second component is mainly explained by the second TCR sequence, with patients *SARK059* and *SARK029* presenting the highest weights.

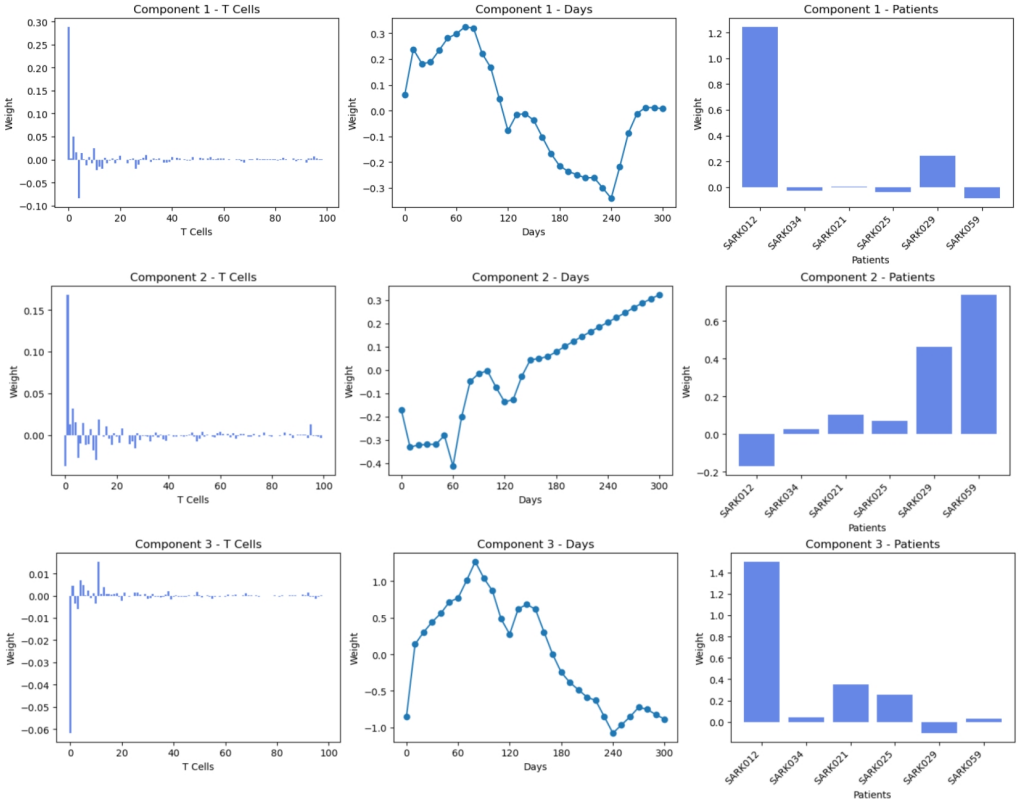


Figure 4.11: CP decomposition for *tensor_300 with* $R = 3$.

In summary, *SARK027* and *SARK018* are highlighted by all the tensors that include them (*tensor_70* to *tensor_200* for *SARK027* and *tensor_70* to *tensor_100* for

SARK018), while the weight of *SARK012*, included in all tensors, increases with the time-steps in the tensor.

4.3 Tucker Decomposition

4.3.1 Selection of Number of Components

Starting with the results for the simulated data, the first method used to select the number of components was DIFFIT, with the obtained results shown in Table 4.1 for *tensor_1* and Table 4.2 for *tensor_2*.

Table 4.1: DIFFIT results for the simulated *tensor_1*. In bold, the optimal rank with the largest $b_{t(m)}$ value.

s	P	Q	R	fit (%)	dif_s	$b_{t(m)}$
3	1	1	1	37.26	37.26	2.22
5	2	1	2	42.18	4.92	-
6	2	2	2	58.93	16.75	1.49
7	3	2	2	59.42	0.49	-
8	**3**	**2**	**3**	**70.69**	**11.27**	**18.37**
9	4	2	3	71.20	0.50	-
10	4	3	3	71.62	0.42	-
11	5	3	3	71.89	0.27	-
12	5	4	3	72.43	0.54	-
13	6	3	4	72.89	0.46	-
14	6	4	4	73.39	0.50	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮
120	100	10	10	100	0	∞

Table 4.2: DIFFIT results for the simulated *tensor_2*. In bold, the optimal rank with the largest $b_{t(m)}$ value.

s	P	Q	R	fit (%)	dif_s	$b_{t(m)}$
3	1	1	1	42.72	42.72	3.24
5	1	2	2	55.91	13.19	1.05
6	2	2	2	56.80	0.90	-
7	**2**	**2**	**3**	**69.34**	**12.54**	**19.12**
8	3	2	3	69.84	0.50	-
9	4	2	3	70.26	0.42	-
10	4	3	3	70.85	0.59	-
11	5	3	3	71.35	0.50	-
12	6	3	3	71.81	0.46	-
13	6	3	4	72.11	0.30	-
14	7	3	4	72.59	0.49	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮
120	100	10	10	100	0	∞

The method clearly outlines (3, 2, 3) as the optimal rank for *tensor_1* and (2, 2, 3) for *tensor_2*, with a significantly high $b_{t(m)}$ value in both cases. This indicates that beyond these ranks, as we increase the number of components, the increase in fit is marginal.

The second method used was the SVD with thresholds, the results of which are summarized in Figure 4.12. As expected, as the threshold increases, the number of components decreases, meaning that less information is kept. It is no surprise that Mode 1 (sequence dimension) starts with a large number of components, as it is the largest mode. However, it is at a threshold equal to 0.3 of the reconstruction error that its number of components is comparable to those of the other modes.

We can further pinpoint the most optimal number of components by examining the singular values of the tensor. Figure 4.13 shows the retained singular values corresponding to the same threshold, and it is evident that the selected singular values capture the majority of the variance in each mode, as the magnitude drops sharply after them. This suggests that suitable component numbers are (3, 2, 3) for *tensor_1* and (2, 2, 3) for *tensor_2*. These selections align well with the known ground truth used to simulate the data. Specifically, *tensor_1* was constructed to contain three distinct patient groups, two temporal patterns (a growing trend and a parabolic shape), and three TCR sequence clusters, matching the 3 (patients), 2 (time), and 3 (sequences) components identified. In contrast, *tensor_2* was designed with only two clearly separable patient groups, as two of the three original groups followed similar temporal dynamics and interacted, effectively behaving as a single group. This explains why the optimal decomposition for *tensor_2* favors just 2 components along the patient mode, while retaining 2 and 3 components for the temporal and sequence modes, respectively. These findings are also consistent with those obtained using the DIFFIT criterion.

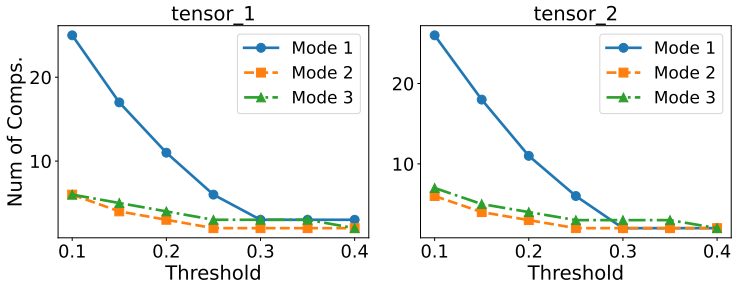


Figure 4.12: Number of components per threshold using the SVD method in the simulated data.

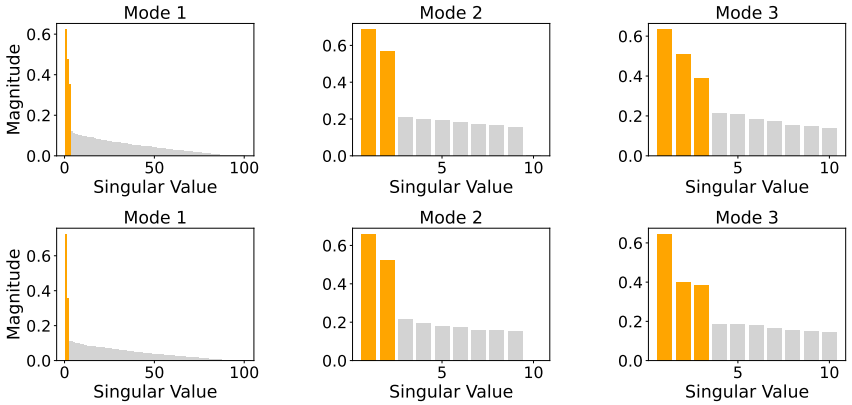


Figure 4.13: Singular values for *tensor_1* (top) and *tensor_2* (bottom) with a 0.30 threshold. Yellow bars indicate retained values, suggesting component counts of (3, 2, 3) and (2, 2, 3), respectively.

Regarding the real data tensors, the same approach was taken. Starting with DIFFIT, the results obtained are shown in Table 4.3, Table 4.4, Table 4.5, and Table 4.6 for *tensor_70*, *tensor_100*, *tensor_200*, and *tensor_300* respectively.

As opposed to the simulated data, the results for DIFFIT are not that clear towards a single combination of numbers of components. In the case of *tensor_100* and *tensor_300*, the rank (1, 1, 1) has the highest $b_{t(m)}$ value. However, in both cases, the fit is below 40%, which seems inappropriate. Thus, we decided to only consider those ranks with a fit of at least 50%. In this sense, the candidates for optimal ranks for *tensor_70* are (1, 2, 2) and (2, 2, 3). For *tensor_100*, they would be (5, 3, 6), (2, 2, 3), or (2, 2, 2), although the high fit of (5, 3, 6) could lead to noise fitting. In the case of *tensor_200*, (2, 2, 2) would be the clear choice while for *tensor_300* (4, 4, 4) and (3, 3, 3) would be the best options, although (4, 4, 4) could be overfitting the data, with a fit of 75%.

Table 4.3: DIFFIT results for *tensor_70*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%.

s	P	Q	R	fit (%)	dif_s	$b_{t(m)}$
3	1	1	1	39.48	39.48	2.41
5	**1**	**2**	**2**	**55.89**	**16.41**	**2.63**
6	2	2	2	58.46	2.58	-
7	**2**	**2**	**3**	**64.70**	**6.24**	**2.44**
8	3	2	3	66.77	2.08	-
9	3	2	4	69.32	2.55	1.07
10	4	2	4	71.71	2.39	1.19
11	5	2	4	73.23	1.51	-
12	5	3	4	74.96	1.74	-
13	5	3	5	76.98	2.01	1.11
14	6	3	5	78.63	1.65	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮
121	100	8	13	100	0	∞

Table 4.4: DIFFIT results for *tensor_100*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%.

s	P	Q	R	fit (%)	dif_s	$b_{t(m)}$
3	1	1	1	37.30	37.30	4.97
5	1	2	2	44.81	7.51	1.07
6	**2**	**2**	**2**	**51.80**	**6.99**	**1.33**
7	**2**	**2**	**3**	**57.07**	**5.27**	**1.41**
8	3	2	3	59.33	2.26	-
9	3	2	4	62.67	3.34	-
10	3	3	4	65.53	2.86	-
11	3	3	5	69.27	3.74	1.21
12	4	3	5	72.35	3.08	1.12
13	5	3	5	74.06	1.70	-
14	**5**	**3**	**6**	**76.80**	**2.74**	**1.46**
⋮	⋮	⋮	⋮	⋮	⋮	⋮
122	100	11	11	100	0	∞

Table 4.5: DIFFIT results for *tensor_200*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%.

s	P	Q	R	fit (%)	dif_s	$b_{t(m)}$
3	1	1	1	36.22	36.22	1.81
5	2	1	2	56.23	20.00	1.59
6	**2**	**2**	**2**	**68.78**	**12.55**	**4.29**
7	2	3	2	71.07	2.29	-
8	2	3	3	73.99	2.93	1.03
9	3	3	3	76.83	2.84	1.62
10	3	3	4	78.38	1.55	-
11	4	3	4	80.12	1.73	-
12	5	3	4	81.79	1.67	-
13	4	4	5	83.44	1.65	-
14	5	4	5	85.19	1.75	1.09
⋮	⋮	⋮	⋮	⋮	⋮	⋮
130	100	21	9	100	0	∞

Table 4.6: DIFFIT results for *tensor_300*. In bold, the optimal ranks with the largest $b_{t(m)}$ values and a fit above 50%.

s	P	Q	R	fit (%)	dif_s	$b_{t(m)}$
3	1	1	1	35.62	35.62	3.40
5	1	2	2	46.08	10.46	1.72
6	2	2	2	52.17	6.09	1.03
7	2	2	3	58.08	5.91	1.28
8	3	2	3	61.43	3.35	-
9	**3**	**3**	**3**	**66.06**	**4.63**	**1.39**
10	4	3	3	68.88	2.82	-
11	3	4	4	71.76	2.88	-
12	**4**	**4**	**4**	**75.09**	**3.33**	**1.65**
13	4	5	4	77.12	2.02	1.00
14	5	5	4	79.14	2.02	1.19
⋮	⋮	⋮	⋮	⋮	⋮	⋮
137	100	31	6	100	0	∞

Parallelly, the SVD method, together with a threshold for the error reconstruction, was also used to estimate the best number of components for Tucker decomposition. The results, summarized in Figure 4.14, suggest that as we increase the threshold, the smaller the number of components is, with Mode 1 consistently yielding the highest counts and Mode 2 the fewest. This is expected as a smaller reconstruction error entails keeping as many singular values as possible to not lose information.

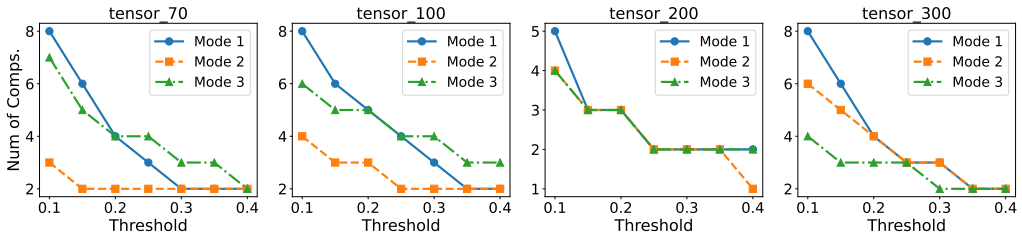


Figure 4.14: Number of components per threshold using the SVD method.

To decide which threshold suits our case best, we plotted the error curves and the singular values' magnitudes. The magnitude of a singular value reflects the amount of variance or information captured by the corresponding component, with larger values representing dominant patterns in the data and smaller values typically associated with noise. So, by looking at them, we can identify an effective rank by detecting where additional components provide diminishing returns. Figure 4.15 exemplifies this for *tensor_70* and for a threshold equal to 0.2. All reconstruction error curves show rapid initial drops followed by plateaus (some with a higher slope than others). In the given example, it is extremely clear that for Mode 2, the best number of components is 2, as there is a steep decrease in the singular values' magnitudes.

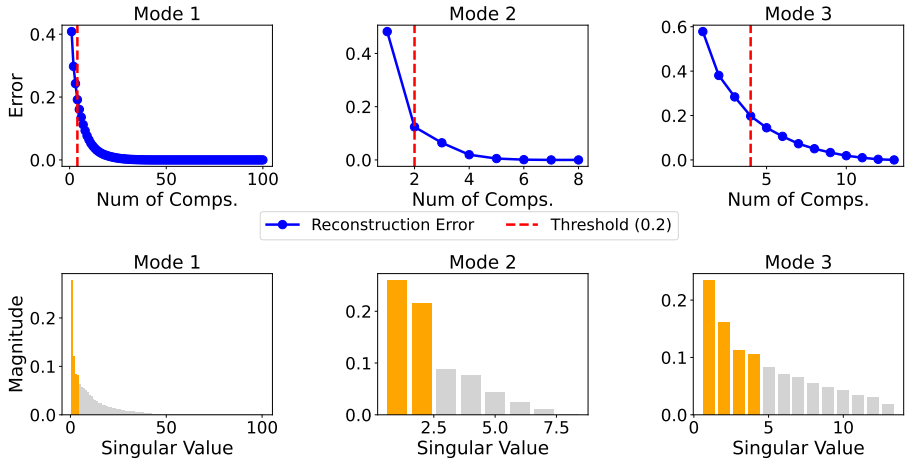


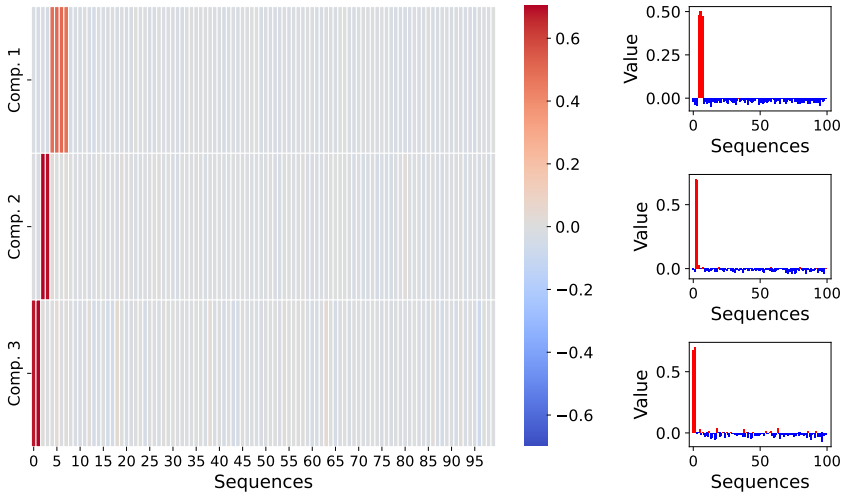
Figure 4.15: Reconstruction error (top) and singular values (bottom) for *tensor_70* with a 0.2 threshold. Yellow bars indicate retained components.

Given all this, we identified the thresholds around 0.2–0.3 to be the most optimal in the case of the real data, as they strike a balance between the combination of reconstruction error and retained singular values' magnitudes. In this way, only the values that explain dominant patterns are kept to avoid overfitting to the noise that the smallest values represent. Also, these fall closer to the DIFFIT results than lower thresholds, confirming that such a range of number of components is truly acceptable to decompose our tensors. However, the results for a threshold equal to 0.3 are the closest to the DIFFIT results.

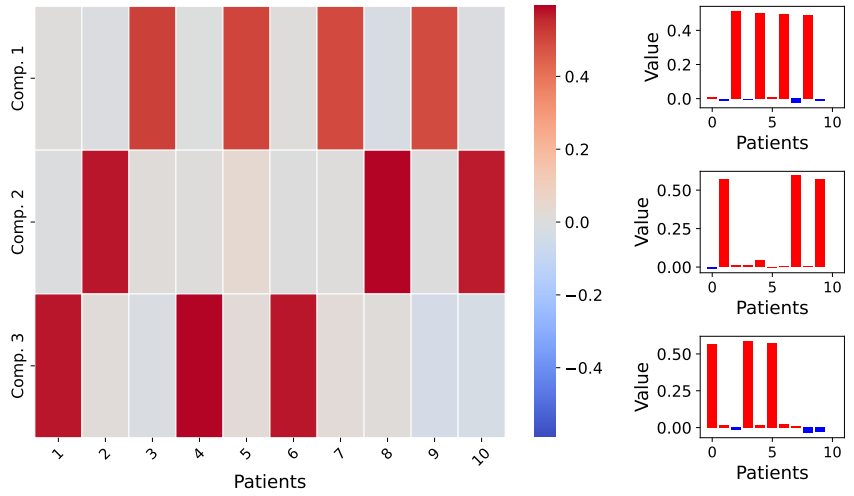
4.3.2 Decompositions

We applied Tucker decomposition to the simulated data tensors (*tensor_1* and *tensor_2*) with (3, 2, 3) and (2, 2, 3) as ranks, as found optimal according to the DIFFIT and SVD methods. Both decompositions show that the element [1, 1, 1] in the core tensor is the one that explains most of the data. Note that [1, 1, 1] stands for components 1, 1, 1 in the first, second, and third mode, respectively. Specifically, around 40% and 60% of the data is explained by these components. In both cases, the first component corresponds to Group C, with 4 patients, instead of 3 as in A and B, which could explain the higher weight. Moreover, element [1, 1, 1] fully describes Group C (patients 3, 5, 7, and 9), while Groups A (patients 1, 4, and 6) and B (patients 2, 8, and 10) need more than one element to be fully explained.

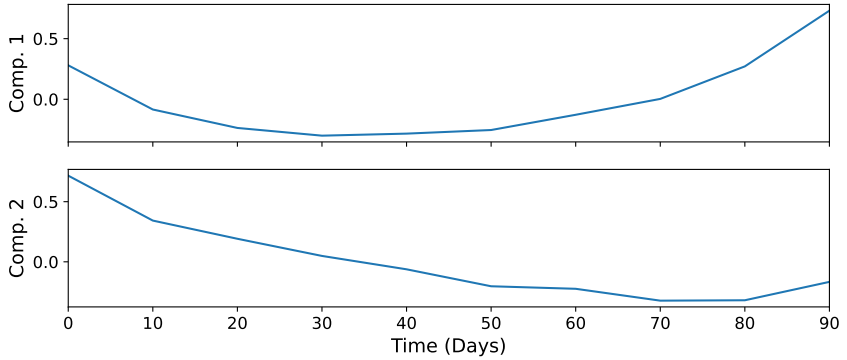
Focusing on the factor matrices, the three patient groups as well as the three distinct sequence groups are properly identified in *tensor_1*. However, the time patterns are not as straightforward to interpret, as multiple elements in the core tensor have to be combined to get the actual time patterns. This is not the case for *tensor_2*, where the two time patterns are clearly illustrated, and the three different groups can be easily interpreted.



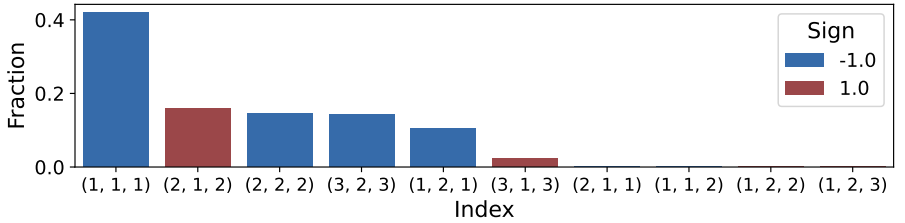
(a) Factor matrix corresponding to the TCR sequences.



(b) Factor matrix corresponding to the patients.

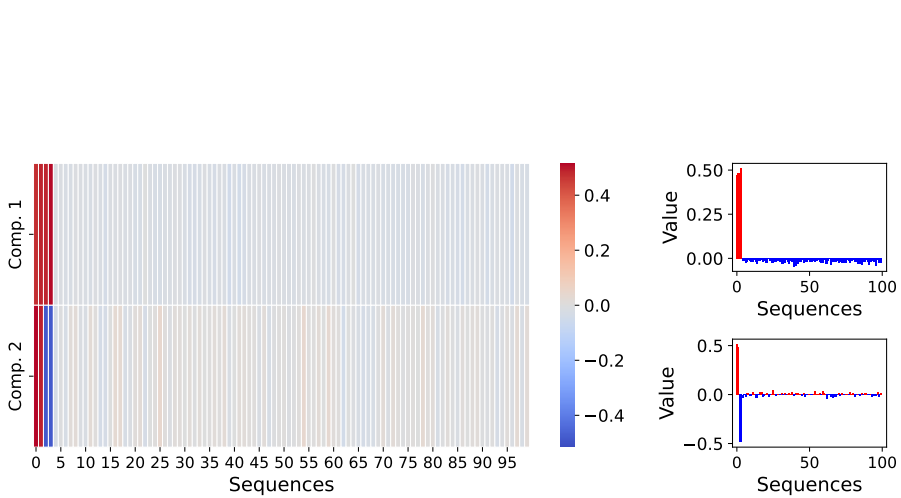


(c) Factor matrix corresponding to the time dimension.

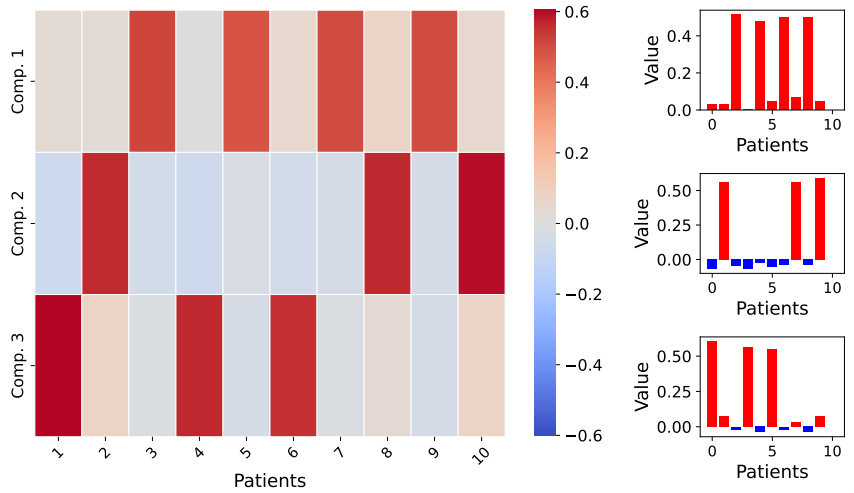


(d) Core matrix's most relevant indices.

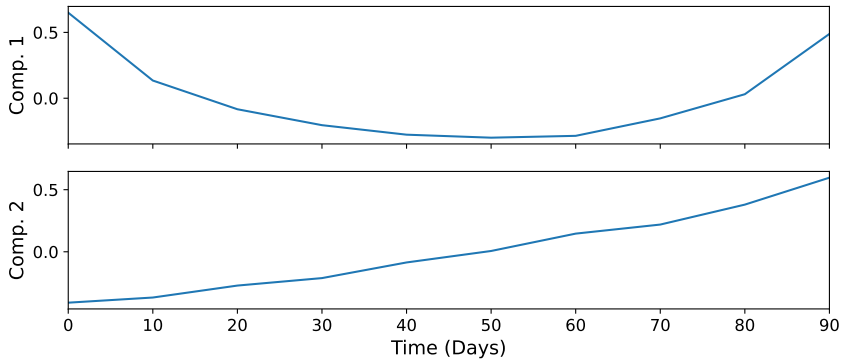
Figure 4.16: Tucker decomposition results for *tensor_1* with components (3, 2, 3).



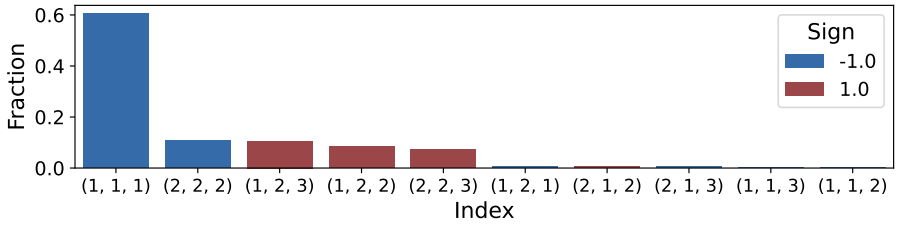
(a) Factor matrix corresponding to the TCR sequences.



(b) Factor matrix corresponding to the patients.



(c) Factor matrix corresponding to the time dimension.



(d) Core matrix's most relevant indices.

Figure 4.17: Tucker decomposition results for *tensor_2* with components (2, 2, 3).

Proceeding with the real data, Tucker decomposition was applied separately to the four tensors for 70, 100, 200, and 300 days. Several combinations of components were tested for each tensor, and the one that captured the most meaningful patterns without adding unnecessary complexity was chosen for each tensor. The complete set of results for all tested combinations can be found in section C.2.

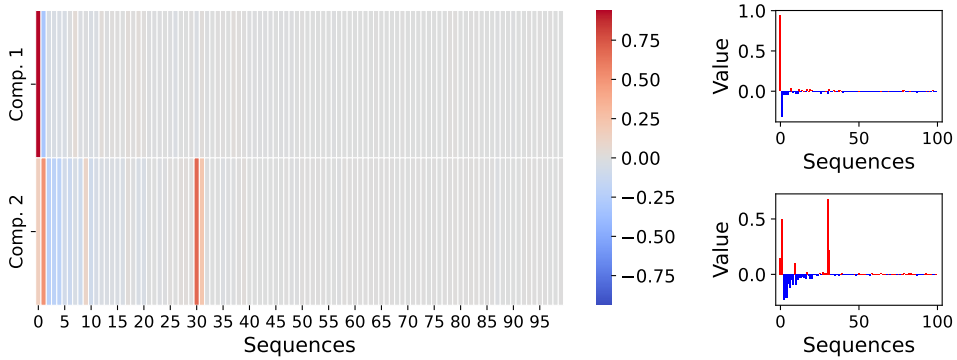
For *tensor_70*, the configuration (2, 2, 3) gave the most useful results (Figure 4.18). In this case, just two core indices, namely [1, 1, 1] and [1, 2, 2], explain about 80% of the data. The first shows a decreasing trend during the first 10 days (Figure 4.18c), mainly linked to the first sequence (Figure 4.18a) and to patient *SARK027*, who has the strongest presence in this pattern (Figure 4.18b). Other patients like *SARK018* and *SARK029* also show some presence. The second main pattern has the opposite time behavior and involves other patients, such as *SARK041* and *SARK018*, with strong positive or negative weights. Overall, the results suggest that only a few patterns are needed to explain most of the behavior in this tensor, which fits with the limited time window of 70 days.

For *tensor_100*, the best result came from using (2, 2, 2) as the rank (Figure 4.19). As in the 70-day tensor, the [1, 1, 1] component explains more than 60% of the data. The other components add little and mostly capture noise. This explains why, despite the low fit, the DIFFIT results suggested (1, 1, 1) as the optimal rank. Although some patients were dropped when building this tensor, the main temporal pattern is still similar to that in *tensor_70*, just extended over a longer time.

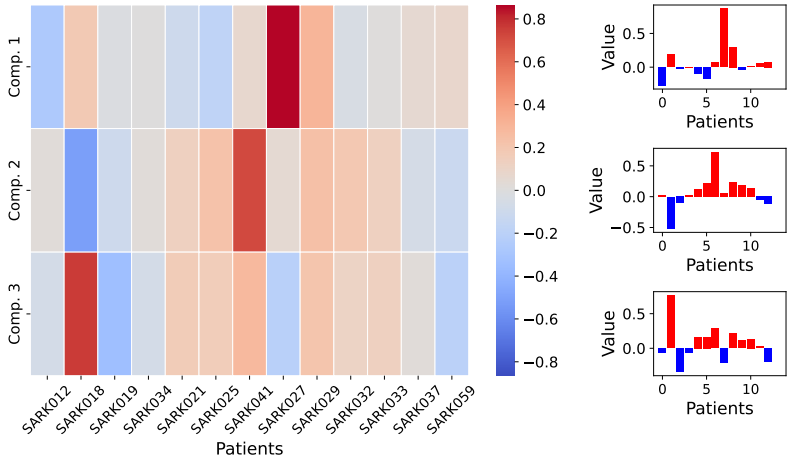
The results for *tensor_200* are different. Using (2, 2, 2) as the rank again (Figure 4.20), the time component shows two new patterns: one that steadily increases, and another that first decreases and then increases again. This is likely because fewer patients are included, but each one has more time points, allowing new trends to appear that were impossible to detect before.

For *tensor_300*, the most useful decomposition used (3, 3, 3) as the rank (Figure 4.21). As expected from having more temporal data, the time dimension shows more complex trends. One component starts low and increases at the end, another dips in the middle before rising again, and the third fluctuates with multiple peaks and dips.

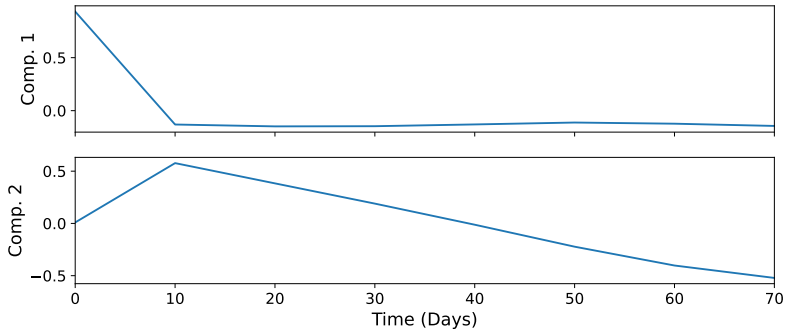
In summary, the number of patients and the amount of temporal data clearly affect the results. The 70- and 100-day tensors show strong, simple patterns shared across many patients, while the 200- and 300-day tensors reveal different and more detailed trends, likely because they have longer time coverage. Still, in all cases, only a few components are needed to explain most of the data, suggesting that while clear patterns are present, the rest of the variation may be due to noise or individual differences between patients.



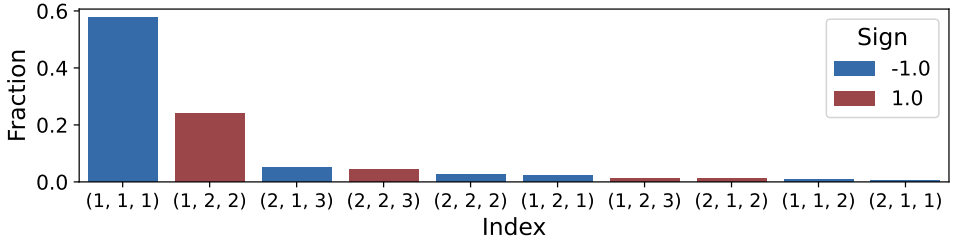
(a) Factor matrix corresponding to the TCR sequences.



(b) Factor matrix corresponding to the patients.

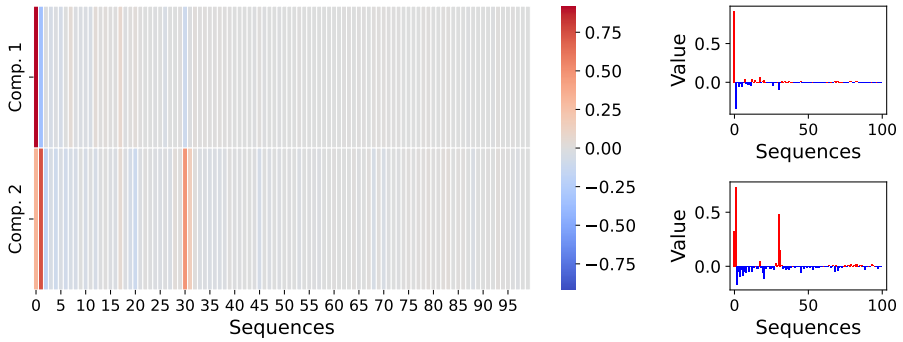


(c) Factor matrix corresponding to time.

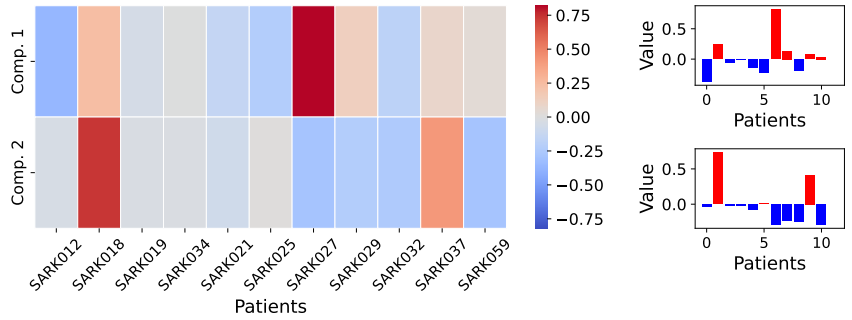


(d) Core matrix's most relevant indices.

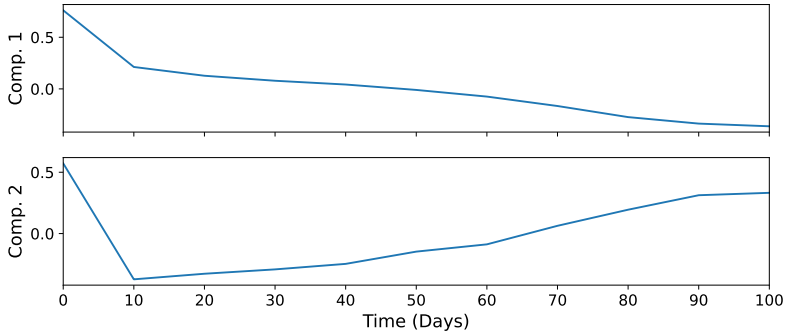
Figure 4.18: Tucker decomposition results for *tensor_70* with components (2, 2, 3).



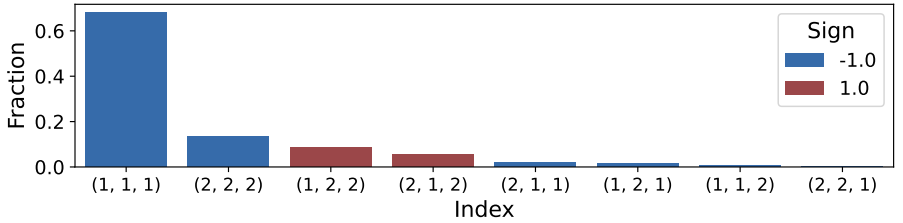
(a) Factor matrix corresponding to the TCR sequences.



(b) Factor matrix corresponding to the patients.

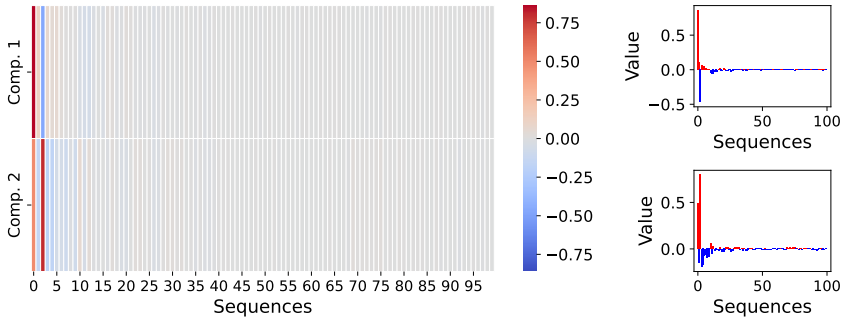


(c) Factor matrix corresponding to the time dimension.

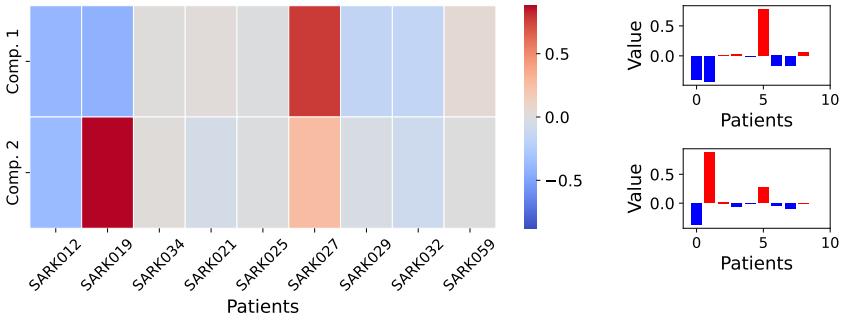


(d) Core matrix's most relevant indices.

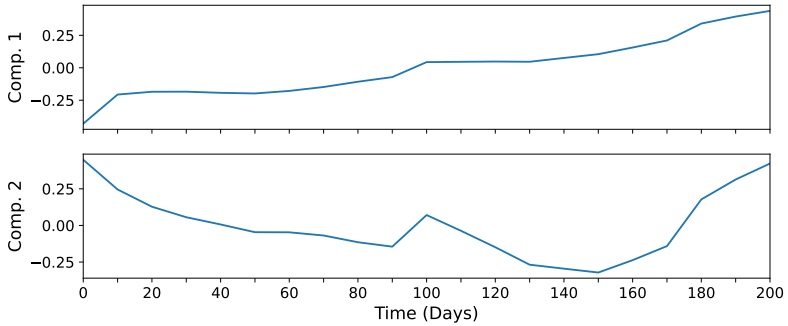
Figure 4.19: Tucker decomposition results for *tensor_100* with components (2, 2, 2).



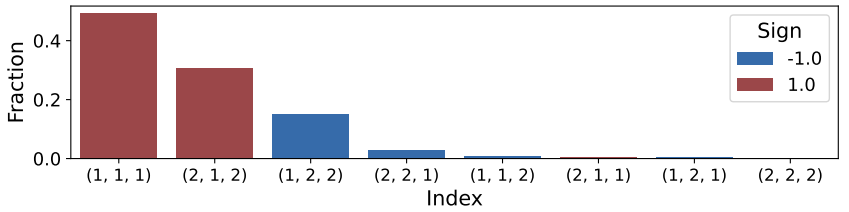
(a) Factor matrix corresponding to the TCR sequences.



(b) Factor matrix corresponding to the patients.

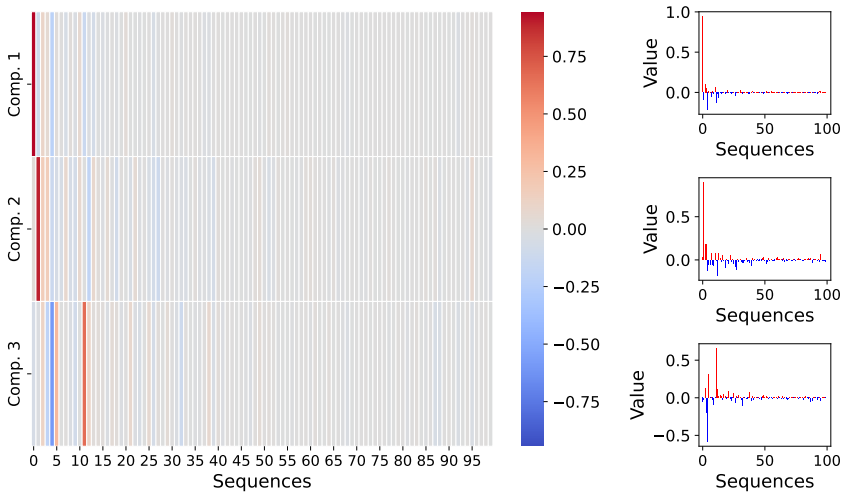


(c) Factor matrix corresponding to the time dimension.

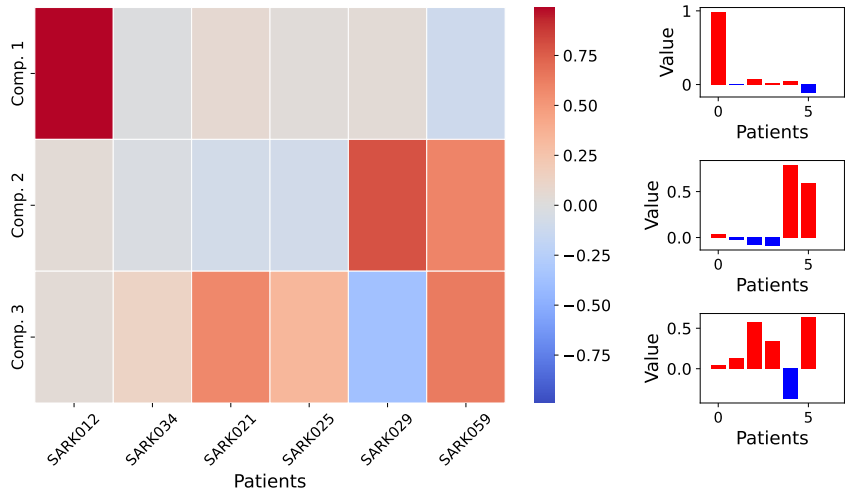


(d) Core matrix's most relevant indices.

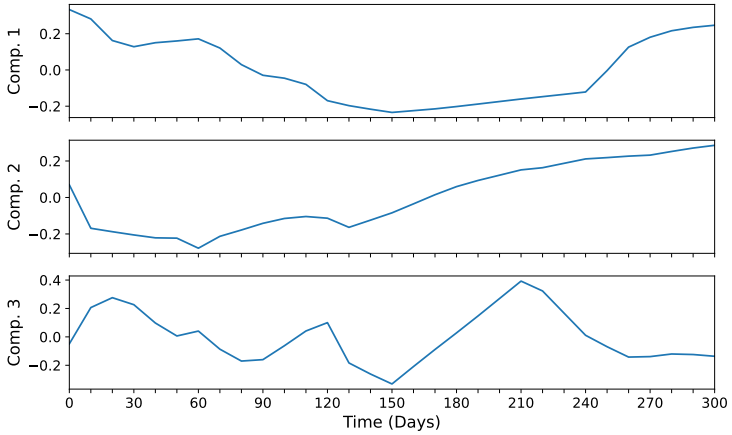
Figure 4.20: Tucker decomposition results for *tensor_200* with components (2, 2, 2).



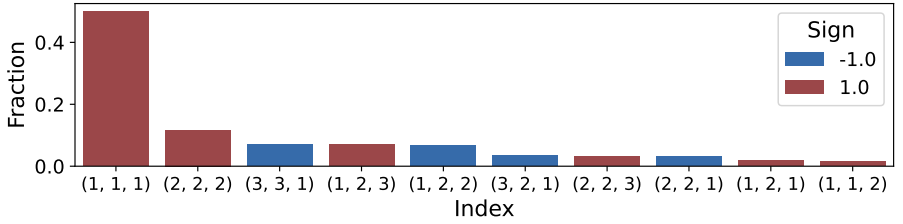
(a) Factor matrix corresponding to the TCR sequences.



(b) Factor matrix corresponding to the patients.



(c) Factor matrix corresponding to the time dimension.



(d) Core matrix's most relevant indices.

Figure 4.21: Tucker decomposition results for *tensor_300* with components (3, 3, 3).

4.4 PARAFAC2

4.4.1 Selection of Number of Components

Similarly to CP, the reconstruction error was also used to select the optimal number of components for PARAFAC2. In Figure 4.22, the results for the two simulated PARAFAC2 arrays *tensor_3* and *tensor_4* are illustrated, for $R = 1, ..., 6$.

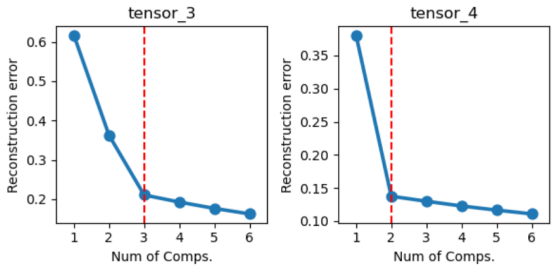


Figure 4.22: Plots of the reconstruction errors for $R = 1, 2, 3, 4, 5, 6, 7$ for the simulated PARAFAC2 arrays. The dotted red line represents the elbow.

The method clearly identifies $R = 3$ and $R = 2$ as the optimal ranks for *tensor_3* and *tensor_4*, respectively. The CORCONDIA values were also calculated for the simulated arrays, and the results are shown in Figure 4.23.

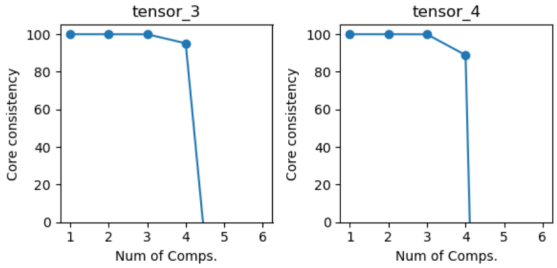


Figure 4.23: Plots of the CONCORDIA values for $R = 1, 2, 3, 4, 5, 6$ for the simulated PARAFAC2 arrays.

The CORCONDIA values suggest that a PARAFAC2 decomposition would be valid for both arrays up to $R = 4$. Thus, taking both methods into account and knowing the nature of the data, we chose $R = 3$ for *tensor_3* and $R = 2$ for *tensor_4*. Although we could go up to $R = 4$ with both arrays according to their CORCONDIA values, the reconstruction error plots suggest that that would just be noise fitting. The same approach was used for the real data, computing both the reconstruction error and the CORCONDIA values, which is shown in Figure 4.24. The results suggest $R = 3$ to be the optimal number of components for a PARAFAC2 decomposition of the real data.

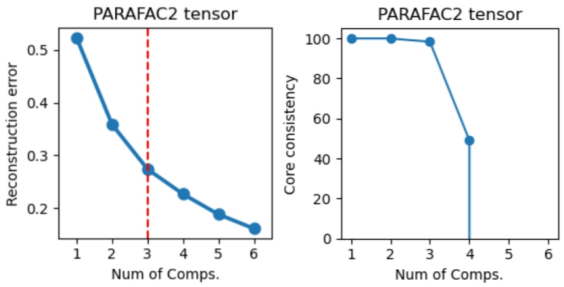


Figure 4.24: Plots of the reconstruction errors and CORCONDIA values for $R = 1, 2, 3, 4, 5, 6$ for the real PARAFAC2 array.

4.4.2 Decompositions

The PARAFAC2 decompositions for the simulated arrays are shown in Figure 4.25 for *tensor_3* and Figure 4.26 for *tensor_4*. The dotted red line in the patient mode represents a threshold for which the projection of the time dimension is plotted. It is half the absolute value of the maximum weight in the patient mode and it was included only to show those time patterns associated with the matrices with the largest weights, to help with the interpretation of the results.

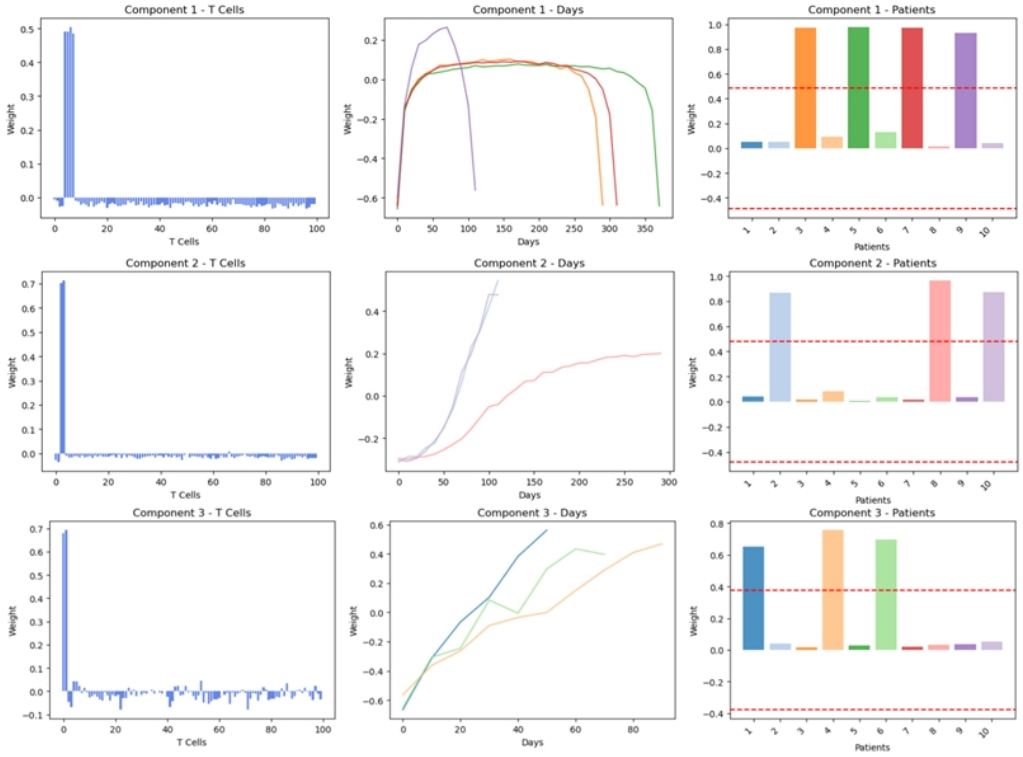


Figure 4.25: PARAFAC2 decomposition for *tensor_3* with $R = 3$. The dotted red line represents a threshold for which the projection of the time dimension is plotted.

As we can see, the decomposition of *tensor_3* clearly identifies groups A (patients 1, 4, 6), B (patients 2, 8, 10), and C (patients 3, 5, 7, 9), while for *tensor_4*, groups A and C are grouped into the first component. Recall from section 3.3 that

the only difference between *tensor_3* and *tensor_4* was the definition of Group C; in particular, the same time pattern was defined, but in *tensor_3* it was applied to sequences 5 to 8, while in *tensor_4* it was applied to the first two sequences, coinciding with Group A. Thus, PARAFAC2 classifies groups A and C in *tensor_4* into the same component, even when the time patterns are significantly different; instead, they are projected separately for every component and every matrix in the array.

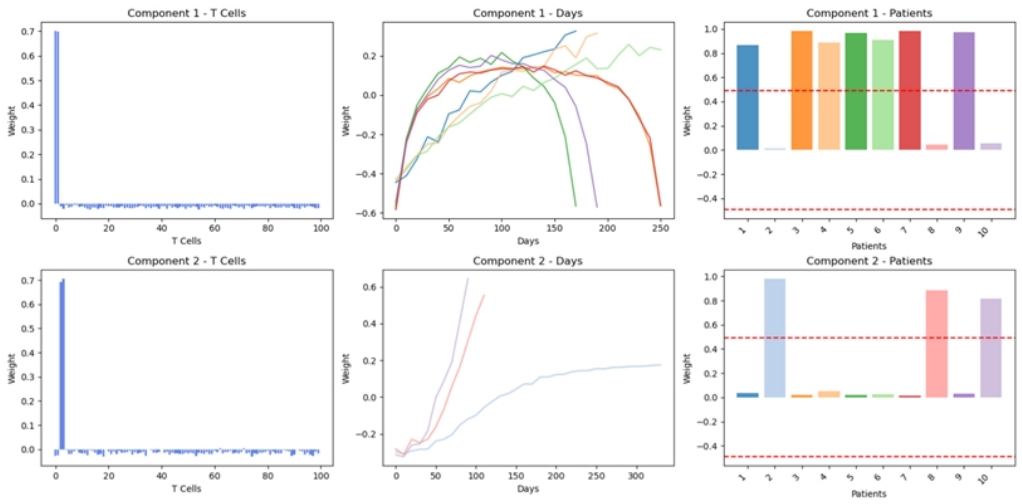


Figure 4.26: PARAFAC2 decomposition for *tensor_4* with $R = 2$. The dotted red line represents a threshold for which the projection of the time dimension is plotted.

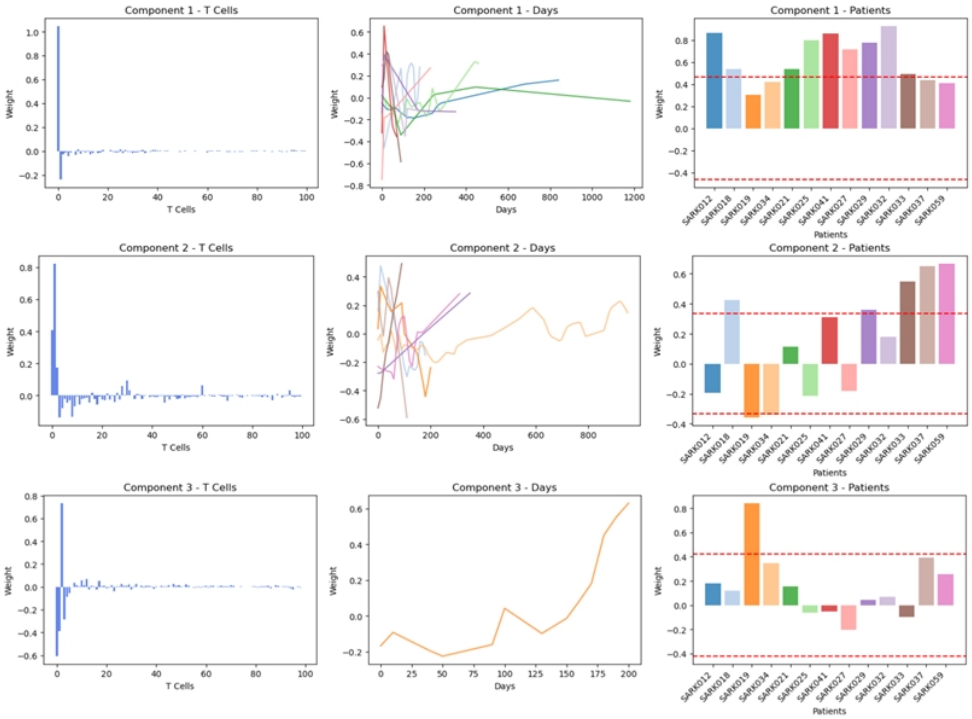


Figure 4.27: PARAFAC2 decomposition for the real data with $R = 3$. The dotted red line represents a threshold for which the projection of the time dimension is plotted.

The PARAFAC2 decomposition of the real data is shown in Figure 4.27. The same threshold was used to reduce the time projections to those with significant weight. The first component mainly describes the first sequence of the patients, while the second component mainly explains the second sequence. The third component is mainly explained by patient *SARK019*, with its first and third sequences. Again, as we saw with the simulated data, the time mode does not seem to describe similar patterns across the components.

4.5 Immunology Metrics

Since several immunology metrics are related to patient outcomes, these were computed and used for clustering the patients into different groups, to compare with the Tucker decomposition clustering results. The metrics' timeseries (Figure 4.28) show a stratification of patients in almost, if not all, metrics. These results show that patients *SARK034* and *SARK025* have an extremely diverse TCR repertoire. For instance, from the last value of the DE_{50} metric, there are two patients who have over and a little under 100 unique reads that make up 50% of the total population of sequences, respectively. Overall, these metrics stay quite stable over time for the majority of patients, with the exception of patients *SARK025* (especially for DE_{50} metric) and *SARK027* (which has a large increase in clonality).

Furthermore, focusing only on the percentage difference between the first time point

(pre-treatment) and the last time point (post-treatment) for each patient (Figure 4.29), a considerable increase in richness is observed in some patients (e.g., *SARK018*, *SARK032*), which may reflect TCR repertoire expansion. In contrast, patients such as *SARK025*, *SARK021*, *SARK012*, and *SARK019* showed substantial increases in clonality, suggesting a treatment-induced clonal expansion of certain TCR sequences, potentially specific to their tumor.

In accordance with this, metrics associated with evenness, such as Pielou, Gini, and Gini-Simpson, generally decrease over time, pointing to a shift toward fewer, more dominant clones in the repertoire. Shannon entropy and inverse Simpson index also tend to decline or stabilize in most patients, reinforcing the trend of reduced diversity post-treatment. Meanwhile, metrics like DE_{50} and Shannon show mixed responses, underscoring the heterogeneity of immune dynamics and potential differences in treatment responsiveness among patients, clinical response, disease severity, or the extent of immune system recovery.

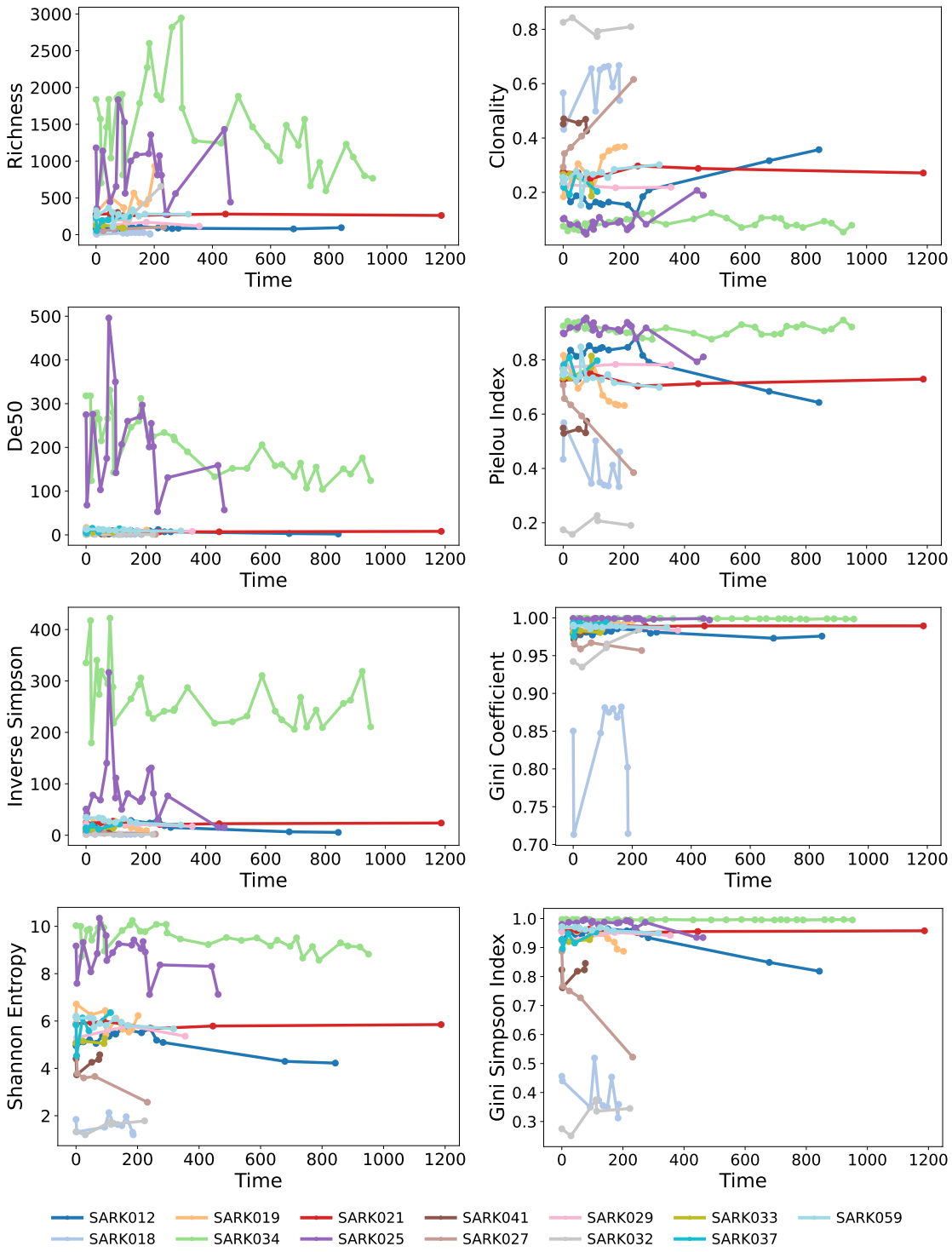


Figure 4.28: Temporal evolution of the studied immunology metrics.

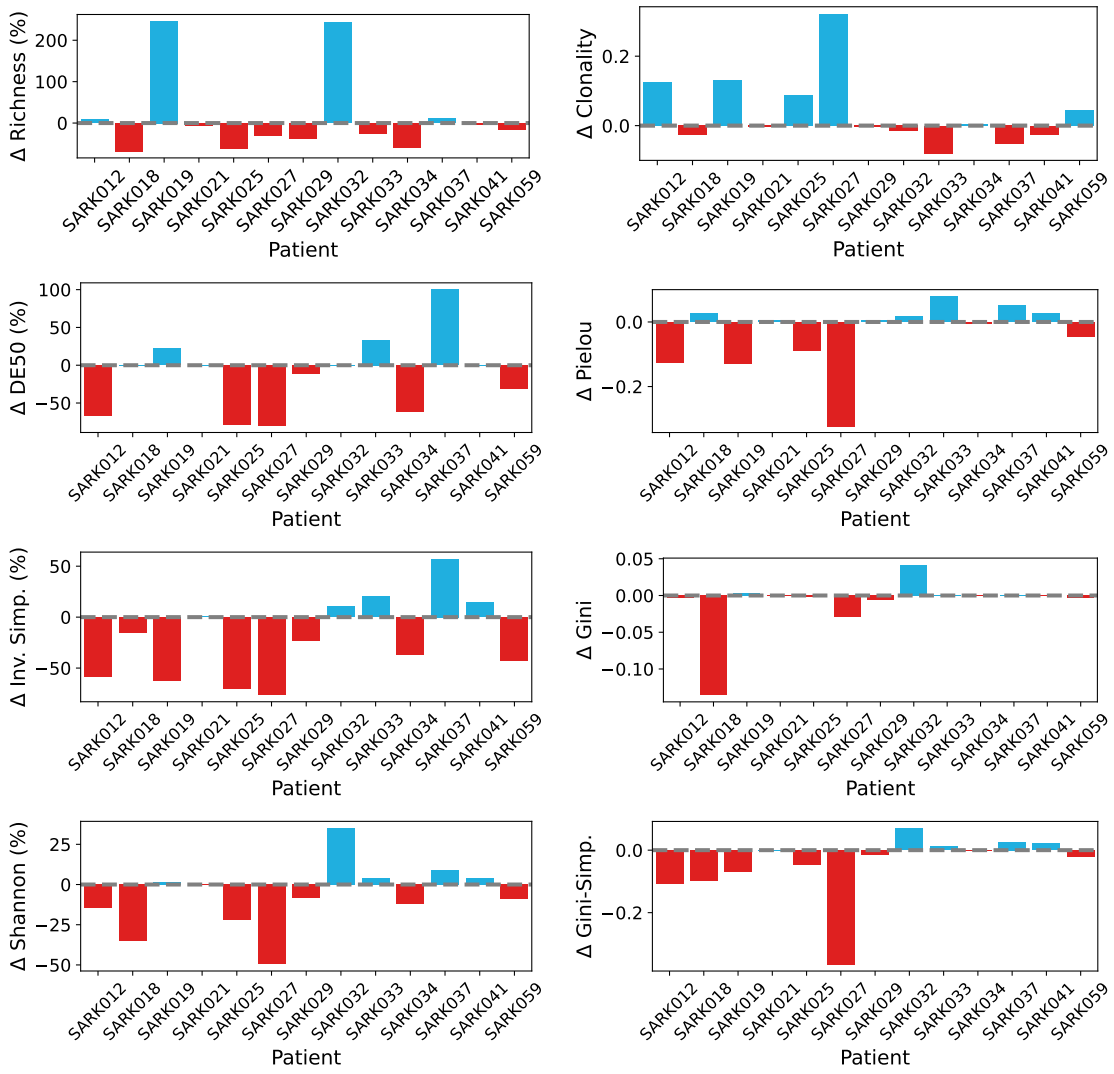
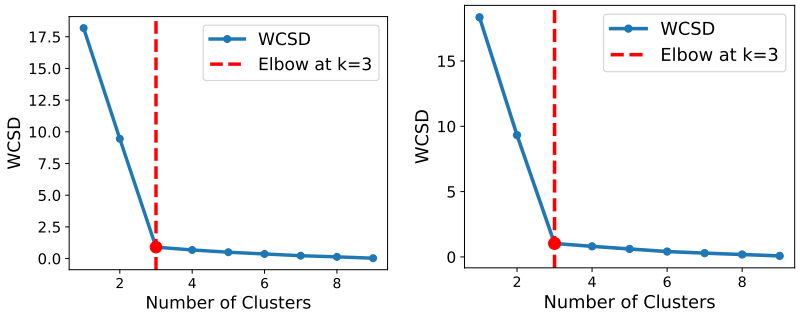


Figure 4.29: Change in the metrics between the first point (before treatment) and the last point (after treatment). For metrics bounded between 0 and 1 (right), the difference is shown in absolute values. For the rest (left), in percentage.

4.6 Clustering and Ground-Truth Validation

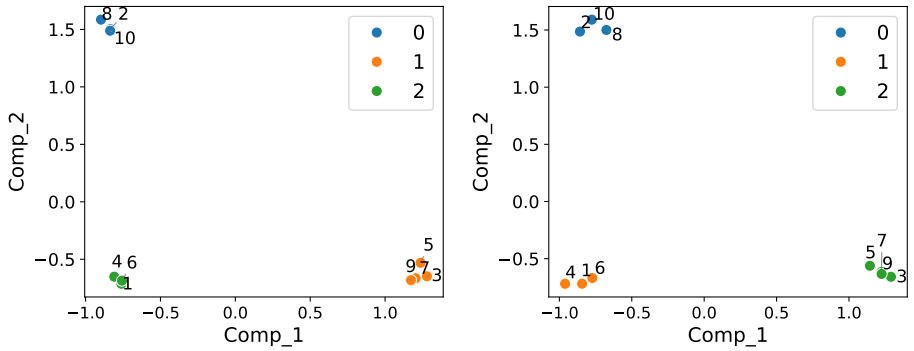
The results for clustering using Tucker decomposition on simulated data are as expected. Figure 4.30 correctly shows that the optimal number of clusters for these simulated tensors is $k = 3$, as the data was made to have three distinct patient groups. This validates the usefulness of Tucker decomposition when it comes to finding patterns to cluster. Figure 4.31 shows the distribution of the patients using the components of the patient's factor matrix. Even without a clustering pipeline, three clusters are evident: one group is made of patients 2, 8, and 10; another of 1, 4, and 5; and the last one of patients 3, 5, 7, and 9, which is consistent with the generated data.

Before describing any clustering results on the real data, we identify five possible



(a) *tensor_1* **(b)** *tensor_2*

Figure 4.30: WCSD for increasing number of clusters k for both simulated tensors.



(a) *tensor_1* **(b)** *tensor_2*

Figure 4.31: Pairplot visualizations of K-medoids clustering results on patient factor matrices for the two simulated tensors. The figures show the clustering structure for a pair of exemplary features for *tensor_1*, and *tensor_2*, with patient names overlaid and $k = 3$.

patient outcomes into which the data can be clustered. These are: complete remission, partial remission or stable disease, local recurrence, metastatic disease, and palliative care. Therefore, a maximum of 5 clusters are expected.

The WCSD curves show that $k = 4$ is the most optimal cluster separation across all tensors, as automatically detected by the `KneeLocator`, which identifies elbows based on diminishing returns in within-cluster variance. *Tensor_300* does not exhibit a clear elbow at (more specifically, it was not automatically detected). It is important to note that this tensor contains only 6 patients, limiting the range of cluster numbers that can be meaningfully tested. Moreover, selecting $k = 4$ in such a small cohort leads to highly granular clusters, having nearly one patient per cluster, which is not truly informative about patient clusters.

Focusing on the clustering groups, there is no agreement on which patients are clustered together. For instance, for *tensor_70*, patients *SARK012* and *SARK019* are grouped in the same cluster, but that is not the case for the other tensors, which are included in different clusters. Looking closely at Figure 4.33, it is not hard to

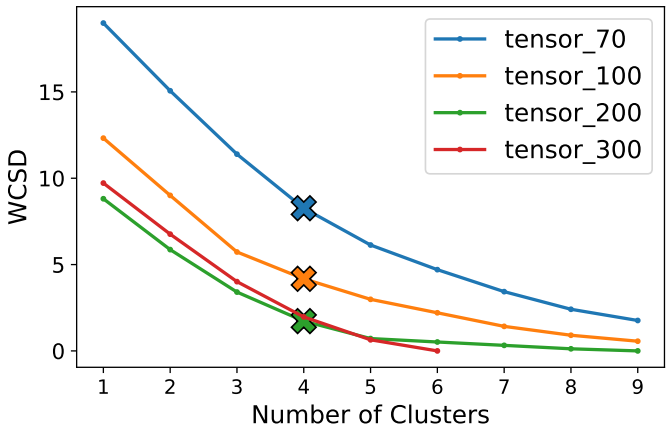


Figure 4.32: WCSD for increasing number of clusters k for all tensors. The cross represents the elbow point of each curve.

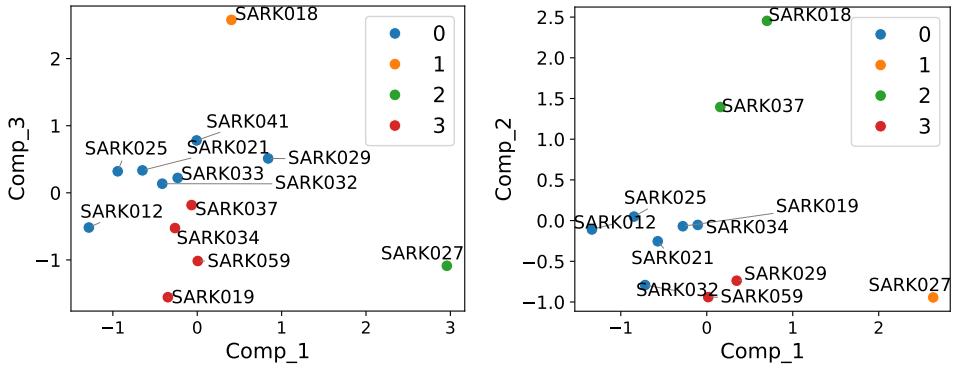
notice that the patients are never in the same spot nor have the same neighbors across tensors.

We applied the same clustering pipeline to the immunology metrics as for Tucker decomposition to find comparable patient groups. For this data, the method also finds $k = 4$ to be the most appropriate number of clusters. Figure 4.34 shows how the patients are grouped within four groups. The NMI score shows that the patients are grouped somewhat in the same clusters, especially for *tensor_300*. This observation is reasonable due to the number of patients for this tensor. The less patients available to group, the most likely is that one cluster is only constituted by one patient, and therefore, the clusters will be more similar between methods.

Comparing these patient groupings with the ground-truth labels, NMI scores result quite low, with the exception of *tensor_300*, which is the tensor with higher scores overall Table 4.7. However, this is not related to a better clustering performance, but it is an effect of a limited patient number. One of the reasons for such low scores could be the fact that one of the four ground-truth labels is not set to any patient on their last data point. Hence, this NMI score is comparing clustering with 4 patient groups to 3 ground-truth patient groups.

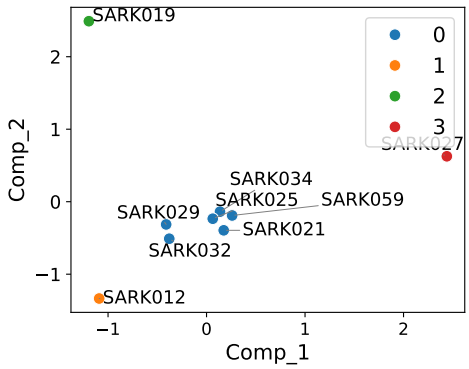
Table 4.7: NMI scores comparing TCR-based, Tucker-based clusterings, and ground-truth (GT) labels across tensor subsets.

Tensor	TCR vs GT	Tucker vs GT	TCR vs Tucker
tensor_70	0.3017	0.2268	0.5359
tensor_100	0.2918	0.3252	0.3945
tensor_200	0.3697	0.5285	0.4232
tensor_300	0.8975	0.6151	0.5579

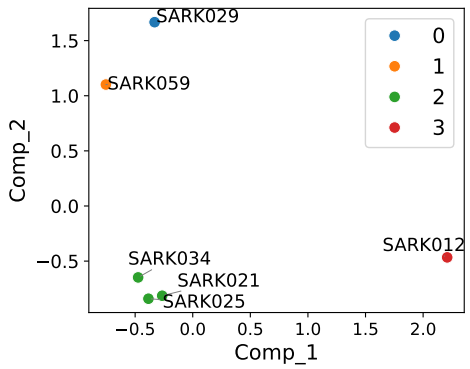


(a) *tensor_70*

(b) *tensor_100*

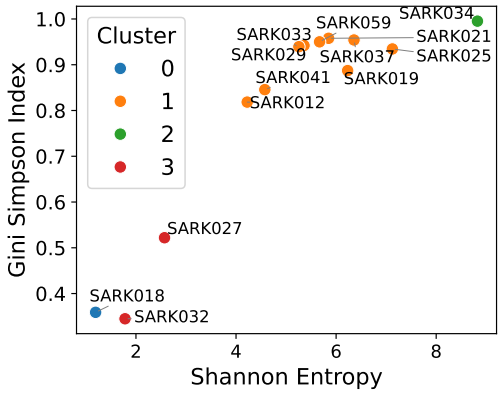


(c) *tensor_200*

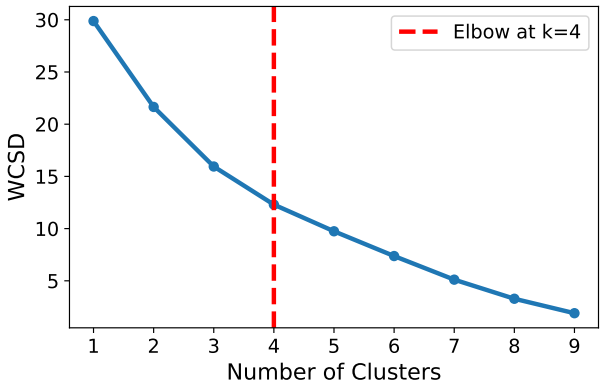


(d) *tensor_300*

Figure 4.33: Pairplot visualizations of K-medoids clustering results on patient factor matrices for the four tensors. The figures show the clustering structure across a pair of exemplary components for *tensor_70*, *tensor_100*, *tensor_200*, and *tensor_300*, with patient names overlaid and $k = 4$.



(a) Shannon entropy vs. Gini-Simpson index (k = 4)



(b) WCSD plot for determining optimal number of clusters for the TCR metrics data.

Figure 4.34: Clustering results from using immunology metrics.

5
Discussion

This chapter discusses the findings and limitations of this thesis and suggests possible future work.

5.1 Validation of Methods

The simulated data was generated with the objective of testing whether the decomposition methods worked as intended. In particular, the two simulated tensors, *tensor_1* and *tensor_2*, were created to be better suited for CP and Tucker, respectively. In that sense, by applying both decomposition methods to both simulated tensors, we aimed to show the correctness and appropriateness of the methods.

The plots of the reconstruction errors for CP in Figure 4.3 for both tensors show that $R = 3$ is the right number of components, aligned with the number of groups of patients generated. In both cases, the fit of the decomposition is around 70%, which means that the reconstruction error of 30% is due to noise. Meanwhile, the CORCONDIA values plotted in Figure 4.4 show that a CP decomposition with $R = 3$ is only valid for *tensor_1*, and not *tensor_2* due to the off-superdiagonal interaction between the factor matrices. This proves our hypothesis, as *tensor_2* was created to be more suited for a Tucker decomposition, and shows that the implementation of CORCONDIA is a useful tool to find out whether CP is a suitable model for our data. The CP decomposition of *tensor_1*, illustrated in Figure 4.7, clearly depicts the three generated groups, showing that CP can be used to find patterns and subgroups in the data.

For Tucker, both the DIFFIT results in Table 4.1 and Table 4.2, and the SVD method with a threshold of 0.30 suggest (3, 2, 3) and (2, 2, 3) to be the optimal ranks for *tensor_1* and *tensor_2*, respectively. The rank for *tensor_2* is as expected, as only two time patterns and two sets of TCR sequences are needed to describe the three groups. Thus, we can conclude that DIFFIT and the SVD method are reliable techniques to find the optimal ranks for Tucker. In the case of *tensor_2*, the 3 groups are clearly described in the decomposition, illustrated in Figure 4.17, while for *tensor_1*, illustrated in Figure 4.16, the two time patterns are not as straightforward to interpret.

All in all, *tensor_2* can only be correctly decomposed with Tucker, while for *tensor_1*, in principle, both approaches would give valid decompositions. As discussed

in subsection 2.5.3 and subsection 2.5.4, the Tucker decomposition is not unique, while CP is unique under some condition. According to De Lathauwer [73], given a tensor of size $I \times J \times K$, its CP decomposition with R number of components is deterministically unique if

$$R \leq K \quad \text{and} \quad R(R-1) \leq I(I-1)J(J-1)/2. \tag{5.1}$$

In the case of *tensor_1* this holds as we have

$$3 \leq 10 \quad \text{and} \quad 6 \leq 445500. \tag{5.2}$$

Thus, we can conclude that due to the better interpretability and uniqueness property, CP is the better model for *tensor_1*. This idea could be generalized to any model as long as the CORCONDIA value is good enough and the uniqueness property is met.

We also generated two arrays to test the PARAFAC2 model, *tensor_3* and *tensor_4*. *Tensor_3* was created to be a good array for PARAFAC2 and show its efficacy, without interaction between the non-varying modes in the different groups, while *tensor_4*, with interactions between the non-varying modes, was generated to show the limitations of the model.

The results of the reconstruction errors in Figure 4.22 show that $R = 3$ and $R = 2$ are the optimal number of components for *tensor_3* and *tensor_4*, respectively. These results are valid, as they both present a CORCONDIA value of 100, as shown in Figure 4.23. The decomposition of *tensor_3*, illustrated in Figure 4.25, shows each group in different components, while the decomposition of *tensor_4*, in Figure 4.26, shows groups A and C contributing to the same component. This results from relaxing the assumption of trilinearity in the varying mode, allowing components with shared sequences but differing time profiles to be modeled together. All in all, PARAFAC2 can be a powerful decomposition tool for arrays with varying dimensions in one mode, but has limitations in capturing patterns specific to said mode.

Lastly, the clustering method (k-medoids) and number of clusters selection (WCSD) were also verified with the simulated data. The pipeline correctly finds the expected number of patient groups, being 3. In this case, there is no need to compute the NMI as in Figure 4.31 is shown that the same patients are in the same groups in both cases.

5.2 Interpretation of Results from Real Data

The application of tensor decomposition to the real TCR repertoire aimed to explore whether meaningful patterns (across time, patients, and TCR sequences) could be captured in a lower dimension, i.e., using tensor decomposition. In this way, underlying biological processes, disease trajectories, or treatment effects could be detected.

Out of all tensor configurations, *tensor_200* appears to be a good option for analysis, balancing temporal resolution with the number of patients retained. While no strong global patterns are observed across all patients, some consistent signals emerge across multiple tensor decompositions. Notably, patients *SARK027* and *SARK018*, when included in the tensor (i.e., in all tensors except *tensor_300*), repeatedly show strong values in the decomposition results, suggesting distinctive temporal dynamics in their TCR repertoires. A common temporal pattern seen in many patients is the increasing dominance of a single TCR sequence over time, typically the most abundant sequence at Day 0. Interestingly, although the corresponding temporal factor often shows a decreasing trend, the associated core tensor index tends to be negative, which indicates an increasing pattern in the reconstructed signal. This aligns with repertoire metrics: *SARK027* exhibits a pronounced decrease in Pielou's evenness and Gini-Simpson index, along with a marked increase in clonality, reflecting a shift toward reduced diversity and clonal expansion. In contrast, *SARK018* shows a large decrease only in the Adapted Gini coefficient, being the only patient with such a strong drop in this metric, while changes in other diversity metrics remain moderate to small. These observations suggest that, despite the dominance of a few sequences, the decompositions do capture meaningful biological dynamics in selected patients.

The goal of analyzing TCR immunology metrics was to further investigate how patients might be grouped based on different clinical outcomes. These metrics revealed a degree of patient stratification, supporting the hypothesis that distinct patient subgroups may exist. For example, patient *SARK027* showed sharp changes across several diversity metrics, indicating strong clonal expansion, while patient *SARK018* showed a marked change only in the adapted Gini coefficient. These contrasting patterns suggest that patients may differ not only in the magnitude of TCR dynamics but also in the type of immunological shift they experience. This observation aligns with findings discussed in Porciello et al.'s review [6], which highlights the relevance of TCR-related features in differentiating immune responses across patient cohorts.

The parallel analysis with immunology metrics serves the purpose of verifying whether direct biological insight can be related to the information captured by tensor decomposition methods. However, the NMI metric suggests that these are not overlapping in the sense that the patients are not grouped in the same manner. One justification for this could be that, as mentioned, the defined ground-truth labels ended up reflecting 3 patient groups (PD, SD, and PR) instead of 4, decreasing the overlap between the methods and the ground-truth. As a result, we also checked how the $k = 3$ clustering results matched with the true labels. However, the NMI values were not better than for $k = 4$. Nevertheless, these are shown in Figure D.1.

Beyond the mismatch with ground-truth labels, inconsistencies in patient clustering across tensors raise further considerations. For example, in *tensor_70*, patients *SARK012* and *SARK019* are clustered together, while in other tensor configurations they are assigned to different clusters. The relative positions of patients in the component space also change depending on the tensor, as seen in Figure 4.33.

These variations could be influenced by several factors. One is the number of patients included in each tensor, which alters the structure of the component space and may affect how the K-medoids algorithm forms clusters. Another factor is the number of components used in the Tucker decomposition; changing this parameter can cause the decomposition to assign different relative weights to patients as it fits the model. These aspects highlight how sensitive clustering results can be to preprocessing and decomposition choices, potentially limiting the stability of patient groupings inferred purely from unsupervised tensor methods.

5.3 Limitations and Future Work

This thesis is not without its limitations. The first and most evident limitation falls on the dataset itself. The dataset we used contained data from 16 patients, three of whom had to be excluded (due to limited data or pre-treatment values missing). Hence, 13 patients can result in too few to draw significant patterns from the data and the results can be skewed to this specific population, lacking generability. Related to this, there are some aspects about the data gathering process that are unknown to us, for instance, if there was some periodicity in the data recollection process, what specific treatment the patients underwent (and if all patients had the same treatment), the age of the patients or their ethnicity. Not knowing these things may introduce some bias in the interpretation of our results, especially those using immunology metrics, as it is known that TCR repertoire decreases with age [74].

In the preprocessing, we interpolated the data due to the irregular time intervals between sequencing measurements. These interpolated values might not reflect the actual biological variability, which could have hindered the analysis. In addition, the 13 patients had their first measurement before treatment, but they could have been at different stages of the disease, which could explain the difficulties in finding common patterns across patients. Another limitation is the assumption that the TCR changes and evolution are only due to the progression of the disease. We assumed this for simplicity, but in reality, other factors can influence the T cell population, especially $\gamma\delta$ T cells, which take part in both adaptive and innate immune systems [18].

The clinical response labels used as ground truth were derived from the last available assessment, which may not fully capture the outcome evolution of each patient. Alternative labeling strategies, such as best overall response or trajectory-based response scoring, could provide a more robust basis for comparison.

In future work, we believe that the introduction of constraints in the decompositions could be interesting to explore. Constraints can enhance the models with domain knowledge and even improve the interpretability of the results. For instance, CANDELINC [75] is a variation of CP that introduces linear constraints in one or more factor matrices. These constraints can be used to introduce user knowledge and make the decomposition more meaningful. In the context of PARAFAC2, the direct

fitting approach makes imposing constraints on the varying mode quite challenging. However, in recent years, several studies have been developed to impose regularization in the varying mode and demonstrate its benefits [76, 77, 78, 79, 80].

An additional aspect which could be interesting to explore is to use Tucker decomposition results for a classification of new patients, given that ground-truth labels are available, as Gillet et. al. show to be promising [59]. This could provide clinicians with additional and more precise information about the predicted future outcome of a patient given their early evolution.

In the context of clustering, it could be worth exploring the use of other techniques, such as DBSCAN, hierarchical clustering, or agglomerative clustering. Even defining the ground-truth labels in another manner, such as labeling each patient according to the most repeated clinical outcome or using only two labels: 'Response' and 'No Response'.

In conclusion, we believe that future work could improve the tensor decomposition approach to find patient patterns in this specific setting. Applying the methods to a more extensive, even, and complete dataset, clearer patterns (and patient groups) could be identified.

6
Conclusion

In this project, we used different tensor decomposition algorithms to model time-resolved sequencing data of patients with cancer with the goal of finding relevant patterns. Namely PARAFAC (CP), Tucker, and PARAFAC2 were used. We also generated data to test and validate the models. CP was found to be the best model in decomposing trilinear data, due to its easier interpretability and uniqueness property. Meanwhile, Tucker was found to be best when interactions between components are necessary. Last but not least, PARAFAC2 allowed us to decompose data with varying lengths in the time dimension, but showed limitations in capturing variations in said mode, due to the relaxation of the trilinearity assumption.

The decompositions of the real data were generally dominated by the patients with the highest variations in TCR counts, and no relevant patterns or subgroups were found. However, the results of the simulated data validated the hypothesis that tensor decomposition can uncover relevant patterns and subgroups in the data.

Clustering applied to the simulated data confirmed the ability of the Tucker decomposition to reveal latent patient groups. In contrast, clustering results on the real data lacked consistency across tensors, reflecting the limited number of patients and variability in data quality.

Overall, we believe that this work shows that tensor decomposition should be considered as a powerful tool in cancer research, particularly in the analysis of higher-order tensors, as it can identify the most relevant and common patterns in the data. Further research on constrained models, clustering techniques, with a more robust dataset could be the key to finding new insights into the role of $\gamma\delta$ T cells on cancer.

Bibliography

[1] C. A. Stiller et al. "Descriptive epidemiology of sarcomas in Europe: Report from the RARECARE project". In: *European Journal of Cancer* 49.3 (2013), pp. 684–695. DOI: 10.1016/j.ejca.2012.09.011. URL: https://www.ejcancer.com/article/S0959-8049(12)00725-3/fulltext.

[2] Zachary Burningham et al. "The Epidemiology of Sarcoma". In: *Clinical Sarcoma Research* 2.1 (2012), p. 14. DOI: 10.1186/2045-3329-2-14. URL: https://clinicalsarcomaresearch.biomedcentral.com/articles/10.1186/2045-3329-2-14.

[3] Cleveland Clinic. *Sarcoma: Symptoms, Types, Causes, Diagnosis & Treatment*. Accessed: 18-Feb-2025. 2025. URL: https://my.clevelandclinic.org/health/diseases/17934-sarcoma.

[4] Yi Hu et al. "$\gamma\delta$ T cells: origin and fate, subsets, diseases and immunotherapy". In: *Signal Transduction and Targeted Therapy* 8.1 (2023), p. 434. DOI: 10.1038/s41392-023-01653-8. URL: https://www.nature.com/articles/s41392-023-01653-8.

[5] M. Attaf et al. "The T cell antigen receptor: the Swiss army knife of the immune system". In: *Clinical and Experimental Immunology* 181.1 (2015), pp. 1–18. DOI: 10.1111/cei.12622. URL: https://academic.oup.com/cei/article/181/1/1/6422102.

[6] Nicola Porciello et al. "T-cell repertoire diversity: friend or foe for protective antitumor response?" In: *Journal of Experimental & Clinical Cancer Research* 41.1 (2022), p. 356. DOI: 10.1186/s13046-022-02566-0. URL: https://jeccr.biomedcentral.com/articles/10.1186/s13046-022-02566-0.

[7] Gustav Johansson et al. "Ultrasensitive DNA Immune Repertoire Sequencing Using Unique Molecular Identifiers". In: *Clinical Chemistry* 66.9 (Aug. 2020), pp. 1228–1237. ISSN: 0009-9147. DOI: 10.1093/clinchem/hvaa159. eprint: https://academic.oup.com/clinchem/article-pdf/66/9/1228/33705299/hvaa159.pdf. URL: https://doi.org/10.1093/clinchem/hvaa159.

[8] Victoria Hore et al. "Tensor decomposition for multiple-tissue gene expression experiments". In: *Nature Genetics* 48.9 (2016), pp. 1094–1100. DOI: 10.1038/ng.3624. URL: https://www.nature.com/articles/ng.3624.

[9] Fengyu Cong et al. "Tensor decomposition of EEG signals: a brief review". In: *Journal of neuroscience methods* 248 (2015), pp. 59–69.

[10] Tamara G Kolda and Brett W Bader. "Tensor decompositions and applications". In: *SIAM review* 51.3 (2009), pp. 455–500.

[11] Crystal L Mackall, Paul S Meltzer, and Lee J Helman. "Focus on sarcomas". In: *Cancer cell* 2.3 (2002), pp. 175–178. DOI: 10.1016/S1535-6108(02)00132-0.

[12] National Cancer Institute. *NCI Dictionary of Cancer Terms: c-kit*. Accessed: 2025-02-27. 2025. URL: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/c-kit.

[13] National Center for Biotechnology Information (NCBI). *KIT gene - NCBI*. Accessed: 2024-03-11. 2024. URL: https://www.ncbi.nlm.nih.gov/gene/3815.

[14] Keith M. Skubitz and David R. D'Adamo. "Sarcoma". In: *Mayo Clinic Proceedings* 82.11 (2007), pp. 1409–1432. ISSN: 0025-6196. DOI: https://doi.org/10.4065/82.11.1409. URL: https://www.sciencedirect.com/science/article/pii/S0025619611614213.

[15] Thomas G. P. Grünewald et al. "Sarcoma treatment in the era of molecular medicine". In: *EMBO Molecular Medicine* 12.11 (2020), e11131. DOI: 10.15252/emmm.201911131. URL: https://pubmed.ncbi.nlm.nih.gov/33047515/.

[16] Jianlin Cao et al. "Angiosarcoma: a review of diagnosis and current treatment". In: *American Journal of Cancer Research* 9.11 (2019), pp. 2303–2313. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6895451/.

[17] National Center for Biotechnology Information (NCBI). *PDGFRA gene - NCBI*. Accessed: 2024-03-11. 2024. URL: https://www.ncbi.nlm.nih.gov/gene/5156.

[18] Francisco A. Bonilla and Hans C. Oettgen. "Adaptive immunity". In: *Journal of Allergy and Clinical Immunology* 125.2, Supplement 2 (2010). 2010 Primer on Allergic and Immunologic Diseases, S33–S40. ISSN: 0091-6749. DOI: https://doi.org/10.1016/j.jaci.2009.09.017. URL: https://www.sciencedirect.com/science/article/pii/S0091674909014055.

[19] Qi Yang, J. Jeremiah Bell, and Avinash Bhandoola. "T-cell lineage determination". In: *Immunological Reviews* 238.1 (2010), pp. 12–22. DOI: 10.1111/j.1600-065X.2010.00956.x. URL: https://doi.org/10.1111/j.1600-065X.2010.00956.x.

[20] Femke Broere and Willem van Eden. "T cell subsets and T cell-mediated immunity". In: *Nijkamp and Parnham's principles of immunopharmacology*. Springer, 2019, pp. 23–35.

[21] Douglas Hanahan and Robert A Weinberg. "The hallmarks of cancer". In: *cell* 100.1 (2000), pp. 57–70. DOI: 10.1016/S0092-8674(00)81683-9.

[22] Douglas Hanahan and Robert A. Weinberg. "Hallmarks of Cancer: The Next Generation". In: *Cell* 144.5 (2011), pp. 646–674. DOI: 10.1016/j.cell.2011.02.013.

[23] Douglas Hanahan. "Hallmarks of Cancer: New Dimensions". In: *Cancer Discovery* 12.1 (Jan. 2022), pp. 31–46. ISSN: 2159-8274. DOI: 10.1158/2159-8290.CD-21-1059. eprint: https://aacrjournals.org/cancerdiscovery/article-pdf/12/1/31/3052722/31.pdf. URL: https://doi.org/10.1158/2159-8290.CD-21-1059.

[24] Jang Hyun Park and Heung Kyu Lee. "Function of $\gamma\delta$ T cells in tumor immunology and their application to cancer therapy". In: *Experimental & molecular medicine* 53.3 (2021), pp. 318–327.

[25] Grant Lythe et al. "How many TCR clonotypes does a body maintain?" In: *Journal of Theoretical Biology* 389 (2016), pp. 214–224. ISSN: 0022-5193. DOI: https://doi.org/10.1016/j.jtbi.2015.10.016. URL: https://www.sciencedirect.com/science/article/pii/S0022519315005159.

[26] Ning Jiang, Alexandra A. Schonnesen, and Ke-Yue Ma. "Ushering in Integrated T Cell Repertoire Profiling in Cancer". In: *Trends in Cancer* 5.2 (2019), pp. 85–94. ISSN: 2405-8033. DOI: https://doi.org/10.1016/j.trecan.2018.11.005. URL: https://www.sciencedirect.com/science/article/pii/S2405803318302619.

[27] Donjete Simnica et al. "T cell receptor next-generation sequencing reveals cancer-associated repertoire metrics and reconstitution after chemotherapy in patients with hematological and solid tumors". In: *Oncoimmunology* 8.11 (2019), e1644110. DOI: 10.1080/2162402X.2019.1644110. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6791461/.

[28] Yi-Tung Chen et al. "Longitudinal High-Throughput Sequencing of the T-Cell Receptor Repertoire Reveals Dynamic Change and Prognostic Significance of Peripheral Blood TCR Diversity in Metastatic Colorectal Cancer During Chemotherapy". In: *Frontiers in Immunology* 12 (2022). ISSN: 1664-3224. DOI: 10.3389/fimmu.2021.743448. URL: https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2021.743448.

[29] Mahima Arunkumar and Christina E Zielinski. "T-Cell Receptor Repertoire Analysis with Computational Tools—An Immunologist's Perspective". In: *Cells* 10.12 (2021), p. 3582.

[30] Nick Borcherding, Qile Yang, and Ksenia Safina. *scRepertoire: A toolkit for single-cell immune receptor profiling*. 2025. URL: https://rdrr.io/github/ncborcherding/scRepertoire/man/clonalDiversity.html.

[31] Adaptive Biotechnologies. *immunoSEQ Analyzer Technical Note: Clonality*. https://www.adaptivebiotech.com/wp-content/uploads/2020/06/immunoSEQ_Analyzer-Tech-Note_Clonality_WEB_MRK-00355.pdf. Accessed 2025-04-07. 2020.

[32] Evelyn C Pielou. "The measurement of diversity in different types of biological collections". In: *Journal of theoretical biology* 13 (1966), pp. 131–144.

[33] Ronald Rousseau et al. "The relationship between diversity profiles, evenness and species richness based on partial ordering". In: *Environmental and Ecological Statistics* 6.2 (1999), pp. 211–223.

[34] Justyna Mika et al. "A comprehensive evaluation of diversity measures for TCR repertoire profiling". In: *BMC biology* 23.1 (2025), pp. 1–18.

[35] Akihiro Hosoi et al. "Increased diversity with reduced "diversity evenness" of tumor infiltrating T-cells for the successful cancer immunotherapy". In: *Scientific reports* 8.1 (2018), p. 1058.

[36] Tamara Gibson Kolda. *Multilinear operators for higher-order decompositions*. Tech. rep. Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA ..., 2006.

[37] Gilbert W Stewart. “On the early history of the singular value decomposition”. In: *SIAM review* 35.4 (1993), pp. 551–566.

[38] Virginia Klema and Alan Laub. “The singular value decomposition: Its computation and some applications”. In: *IEEE Transactions on automatic control* 25.2 (1980), pp. 164–176.

[39] Carla D Martin and Mason A Porter. “The extraordinary SVD”. In: *The American Mathematical Monthly* 119.10 (2012), pp. 838–851.

[40] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. “Singular value decomposition and principal component analysis”. In: *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[41] Hervé Abdi and Lynne J Williams. “Principal component analysis”. In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459.

[42] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[43] Frank L Hitchcock. “The expression of a tensor or a polyadic as a sum of products”. In: *Journal of Mathematics and Physics* 6.1-4 (1927), pp. 164–189.

[44] J Douglas Carroll and Jih-Jie Chang. “Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition”. In: *Psychometrika* 35.3 (1970), pp. 283–319.

[45] Richard A Harshman et al. “Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis”. In: *UCLA working papers in phonetics* 16.1 (1970), p. 84.

[46] Henk AL Kiers. “Towards a standardized notation and terminology in multiway analysis”. In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 14.3 (2000), pp. 105–122.

[47] Joseph B Kruskal. “Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics”. In: *Linear algebra and its applications* 18.2 (1977), pp. 95–138.

[48] Johan Håstad. “Tensor rank is NP-complete”. In: *Journal of algorithms* 11.4 (1990), pp. 644–654.

[49] Ledyard R Tucker. “Implications of factor analysis of three-way matrices for measurement of change”. In: *Problems in measuring change* 15.122-137 (1963), p. 3.

[50] Ledyard R Tucker. “Some mathematical notes on three-mode factor analysis”. In: *Psychometrika* 31.3 (1966), pp. 279–311.

[51] Joseph B Kruskal. “Rank, decomposition, and uniqueness for 3-way and N-way arrays”. In: *Multiway data analysis*. 1989, pp. 7–18.

[52] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. “On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors”. In: *SIAM journal on Matrix Analysis and Applications* 21.4 (2000), pp. 1324–1342.

[53] Richard A Harshman et al. “PARAFAC2: Mathematical and technical notes”. In: *UCLA working papers in phonetics* 22.3044 (1972), p. 122215.

[54] Henk AL Kiers, Jos MF Ten Berge, and Rasmus Bro. “PARAFAC2—Part I. A direct fitting algorithm for the PARAFAC2 model”. In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 13.3-4 (1999), pp. 275–294.

[55] Zhixin Cyrillus Tan and Aaron S Meyer. “The structure is the message: Preserving experimental context through tensor decomposition”. In: *Cell Systems* 15.8 (2024), pp. 679–693.

[56] Rasmus Bro and Henk AL Kiers. “A new efficient method for determining the number of components in PARAFAC models”. In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 17.5 (2003), pp. 274–286.

[57] Marieke E Timmerman and Henk AL Kiers. “Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima”. In: *British journal of mathematical and statistical psychology* 53.1 (2000), pp. 1–16.

[58] T Wansbeek and J Verhees. “Models for multidimensional matrices in econometrics and psychometrics”. In: *Multiway data analysis* (1989), pp. 543–552.

[59] Annabelle Gillet, Éric Leclercq, and Lucile Sautot. “A Guide to the Tucker Tensor Decomposition for Data Mining: Exploratory Analysis, Clustering and Classification”. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems LIV: Special Issue on Data Management - Principles, Technologies, and Applications*. Ed. by Abdelkader Hameurlain et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023, pp. 56–88. ISBN: 978-3-662-68014-8. DOI: 10.1007/978-3-662-68014-8_3. URL: https://doi.org/10.1007/978-3-662-68014-8_3.

[60] Maja H Kamstrup-Nielsen, Lea G Johnsen, and Rasmus Bro. “Core consistency diagnostic in PARAFAC2”. In: *Journal of Chemometrics* 27.5 (2013), pp. 99–105.

[61] Rasmus Bro. “PARAFAC. Tutorial and applications”. In: *Chemometrics and intelligent laboratory systems* 38.2 (1997), pp. 149–171.

[62] Rasmus Bro and Age K Smilde. “Centering and scaling in component analysis”. In: *Journal of chemometrics* 17.1 (2003), pp. 16–33.

[63] Jean Kossaifi et al. “TensorLy: Tensor Learning in Python”. In: *Journal of Machine Learning Research (JMLR)* 20.26 (2019).

[64] Ville Satopaa et al. “Finding a" kneedle" in a haystack: Detecting knee points in system behavior”. In: *2011 31st international conference on distributed computing systems workshops*. IEEE. 2011, pp. 166–171.

[65] Marie Roald and Yngve Mardal Moe. “TLViz: Visualising and analysing tensor decomposition”. In: ().

[66] Hae-Sang Park and Chi-Hyuck Jun. “A simple and fast algorithm for K-medoids clustering”. In: *Expert systems with applications* 36.2 (2009), pp. 3336–3341.

[67] Preeti Arora, Shipra Varshney, et al. “Analysis of k-means and k-medoids algorithm for big data”. In: *Procedia Computer Science* 78 (2016), pp. 507–512.

[68] Noor Kamal Kaur, Usvir Kaur, and Dheerendra Singh. “K-Medoid clustering algorithm-a review”. In: *Int. J. Comput. Appl. Technol* 1.1 (2014), pp. 42–45.

[69] Andrei Novikov. “PyClustering: Data Mining Library”. In: *Journal of Open Source Software* 4.36 (Apr. 2019), p. 1230. DOI: 10.21105/joss.01230. URL: https://doi.org/10.21105/joss.01230.

[70] Andrei Novikov. *PyClustering: Data Mining Library*. https://pyclustering.github.io/docs/0.10.1/html/d0/dd3/classpyclustering_1_1cluster_1_1kmedoids_1_1kmedoids.html. Accessed: 2025-05-09. 2019.

[71] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[72] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[73] Lieven De Lathauwer. "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization". In: *SIAM journal on Matrix Analysis and Applications* 28.3 (2006), pp. 642–666.

[74] Xiaoping Sun et al. "Longitudinal analysis reveals age-related changes in the T cell receptor repertoire of human T cell subsets". In: *The Journal of clinical investigation* 132.17 (2022).

[75] J Douglas Carroll, Sandra Pruzansky, and Joseph B Kruskal. "CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters". In: *Psychometrika* 45.1 (1980), pp. 3–24.

[76] Marie Roald et al. "An AO-ADMM approach to constraining PARAFAC2 on all modes". In: *SIAM Journal on Mathematics of Data Science* 4.3 (2022), pp. 1191–1222.

[77] Ardavan Afshar et al. "COPA: Constrained PARAFAC2 for sparse & large datasets". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 793–802.

[78] Jeremy E Cohen and Rasmus Bro. "Nonnegative PARAFAC2: A flexible coupling approach". In: *International conference on latent variable analysis and signal separation*. Springer. 2018, pp. 89–98.

[79] Nathaniel E Helwig. "Estimating latent trends in multivariate longitudinal data via Parafac2 with functional and structural constraints". In: *Biometrical Journal* 59.4 (2017), pp. 783–803.

[80] Mark H Van Benthem et al. "Getting to the core of PARAFAC2, a nonnegative approach". In: *Chemometrics and Intelligent Laboratory Systems* 206 (2020), p. 104127.

A

Overview of GitHub repository location of additional figures

For a complete collection of figures—including all tensor decompositions, component visualizations, clustering results, and time-resolved immunological metrics—please refer to the public GitHub repository:
`https://github.com/pablovadillo/Tensordecomposition/tree/main/Results`
Below is a guide to help locate the relevant materials for each section in this appendix:

- **Ground-truth:** Additional plots showing raw TCR count per tensor and per patient overtime as well as TCR counts and TCR metrics evolution related to the ground truth-labels over time for each patient can be found in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/Data%20plots`
- **Tensor components:** All extracted components from the Tucker decomposition for each tensor (e.g., tensor_70, tensor_100) are available in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/Tucker%20Components`
- **Tucker Decompositions:** Full Tucker decomposition reconstructions with different number of components can be found under `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/Tucker%20Decompositions`
- **Clustering results (real and simulated):** Visualizations of clustering applied to Tucker factor matrices are available in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/Tucker%20Clustering/All%20with%20labels`
- **TCR repertoire metrics:** Time-evolving plots of clonality, richness, and other immunological metrics for each patient are provided in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/TCR%20metrics/Time%20evolution`
- **TCR metric clustering:** Clustering results based on TCR metrics can be found in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/TCR%20metrics/Clustering`
- **TCR metric comparison by cluster:** Comparative plots of TCR metric time evolution across clusters are located at `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/TCR%20metrics/Metric%20by%20cluster`
- **Change in TCR metrics:** Plots showing the change in TCR metric values over time or across groups are in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/TCR%20metrics/Change%20in%20metrics`

- **Ground truth data:** Ground truth labels and supporting visuals are found in `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/GT`
- **Cluster comparison:** Comparative clustering plots and evaluations across tensors and clustering methods are located at `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/Clust%20Comparison`

Please consult these folders for figures not included here due to space constraints.

B

Other data related plots

This appendix chapter refers to some extra plots about the dataset used in this project.

B.1 Patient data kept for each tensor

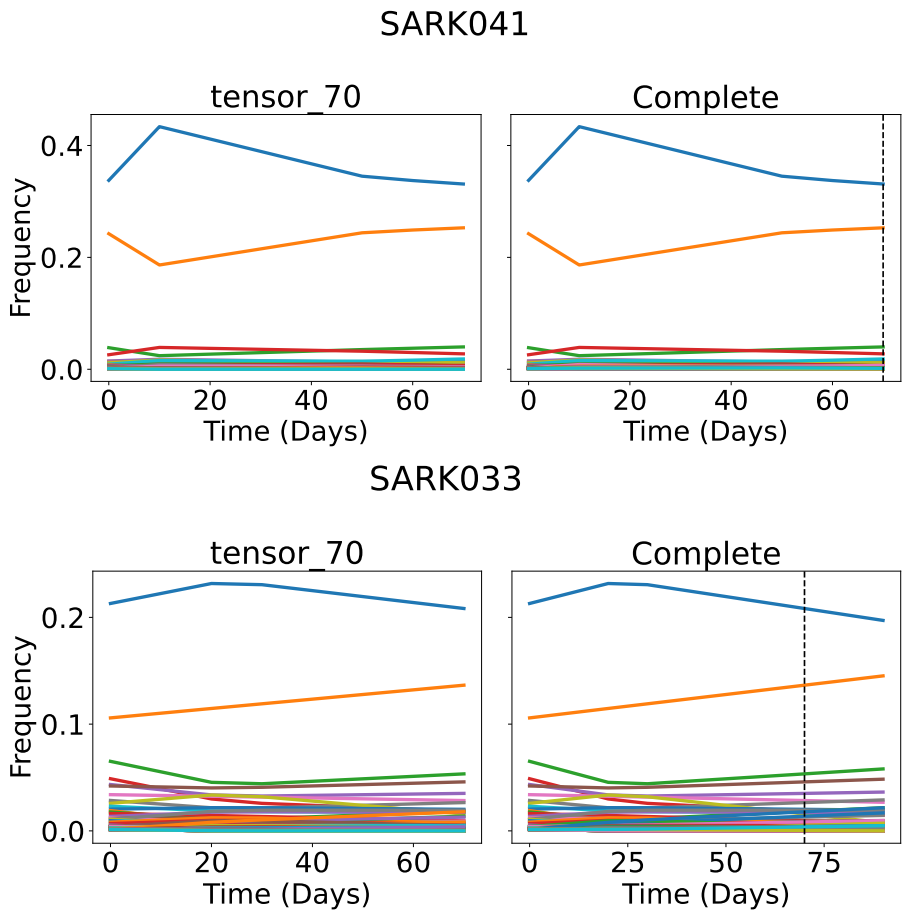


Figure B.1: Patients with data under 100 but over 70 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor.

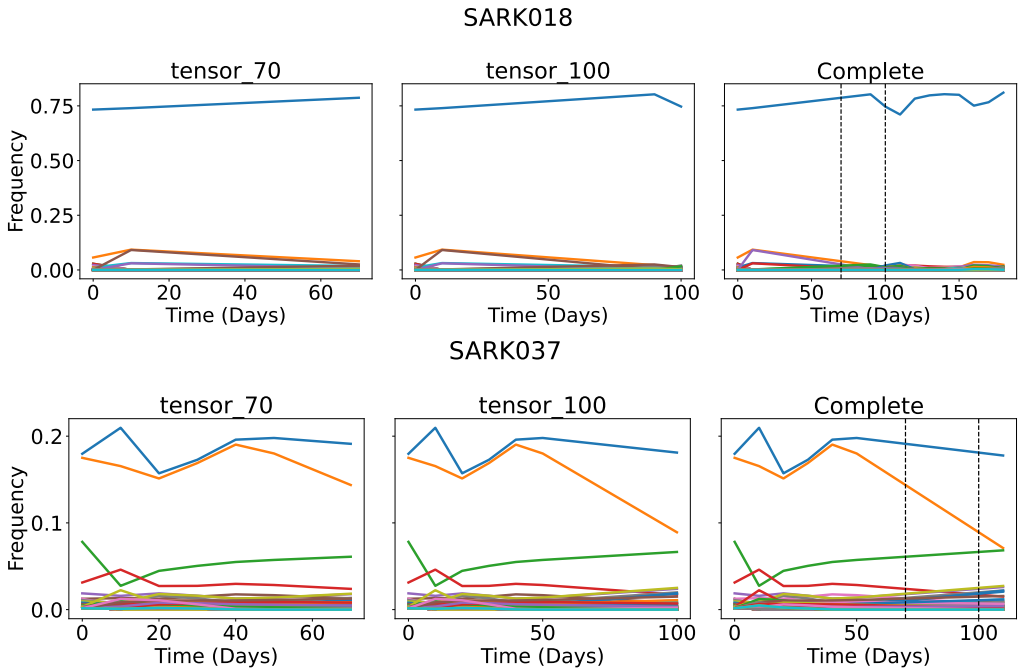


Figure B.2: Patients with data under 200 but over 100 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor.

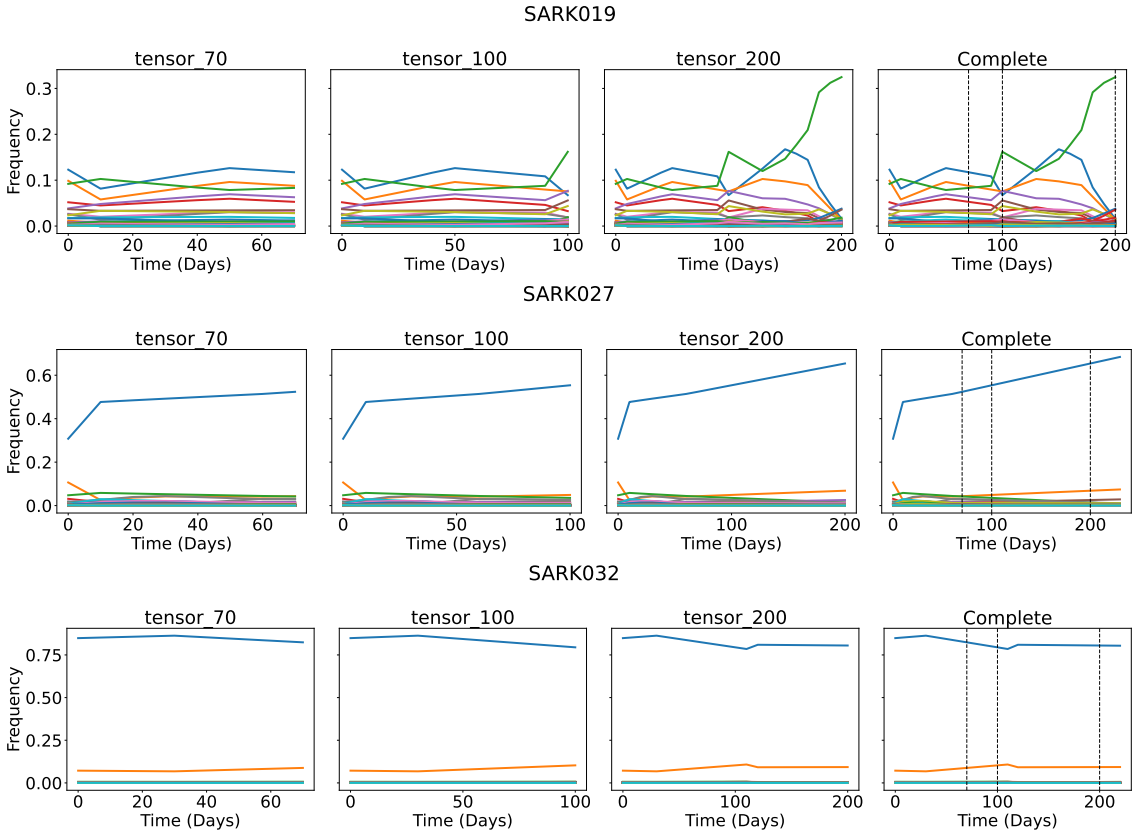


Figure B.3: Patients with data under 300 but over 200 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor.

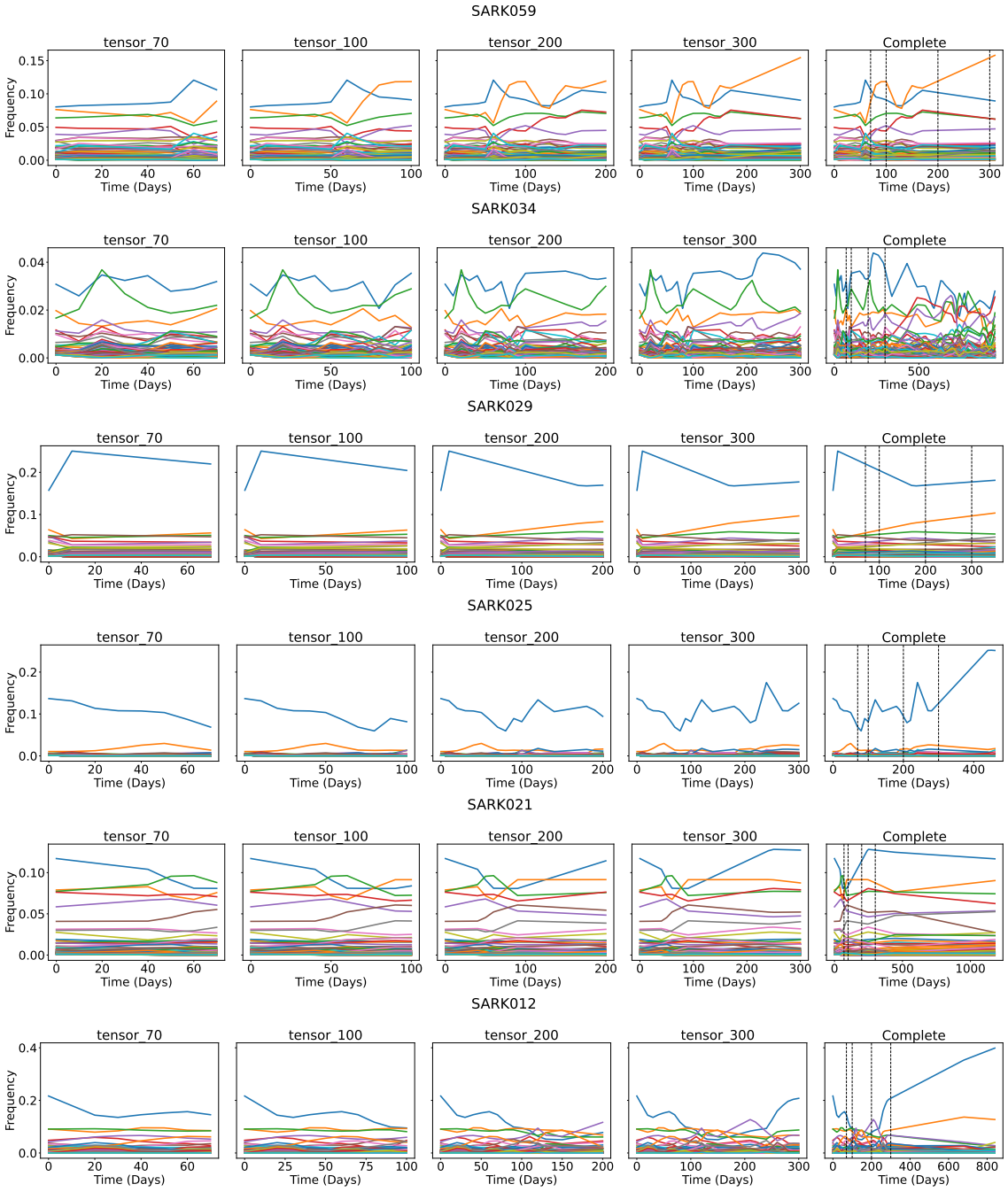


Figure B.4: Patients with data over 300 days. The vertical lines on the rightmost plot correspond to the cuttings for each tensor.

B.2 Tensor data

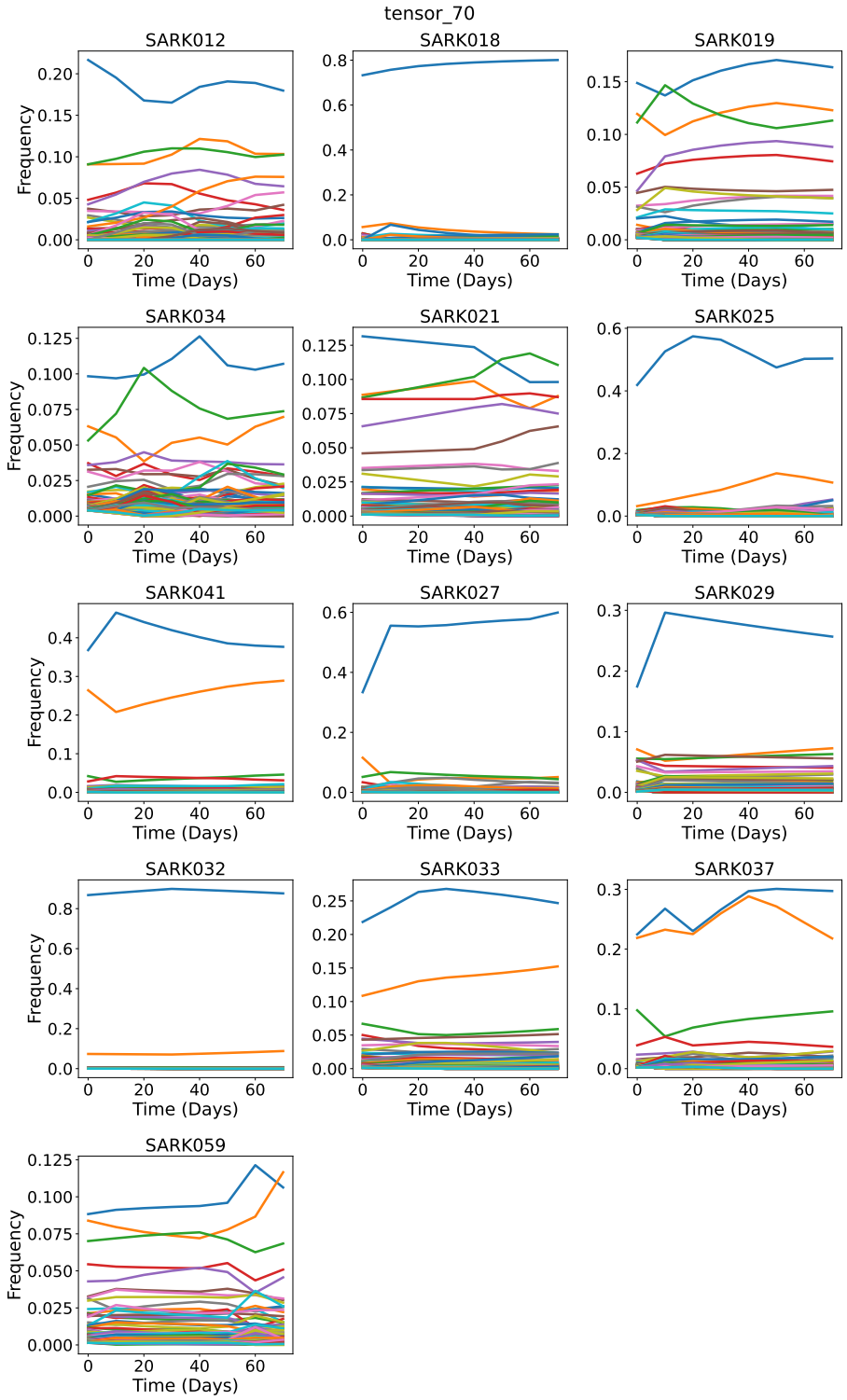


Figure B.5: *Tensor_70* time series

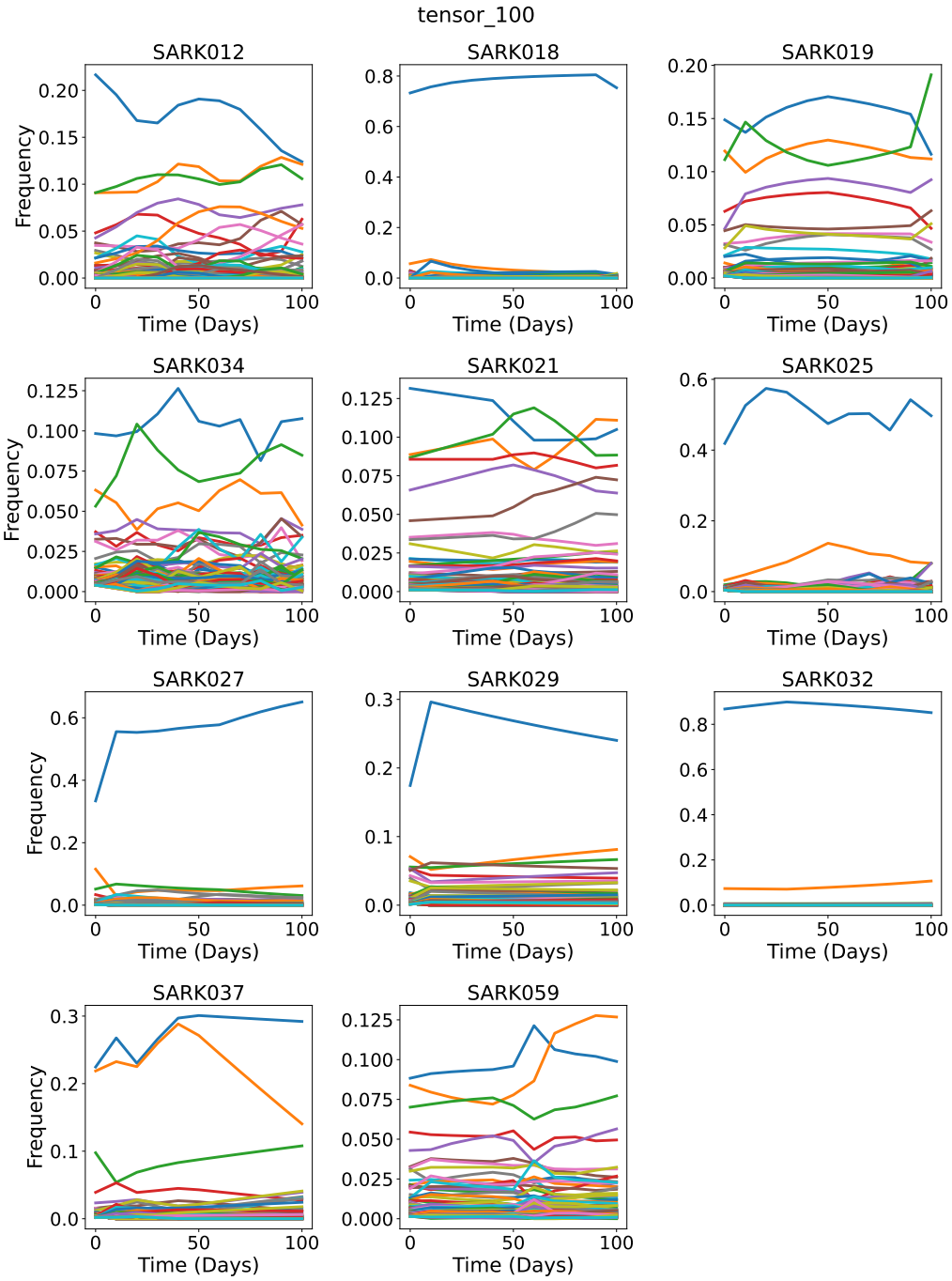


Figure B.6: *Tensor_100* time series

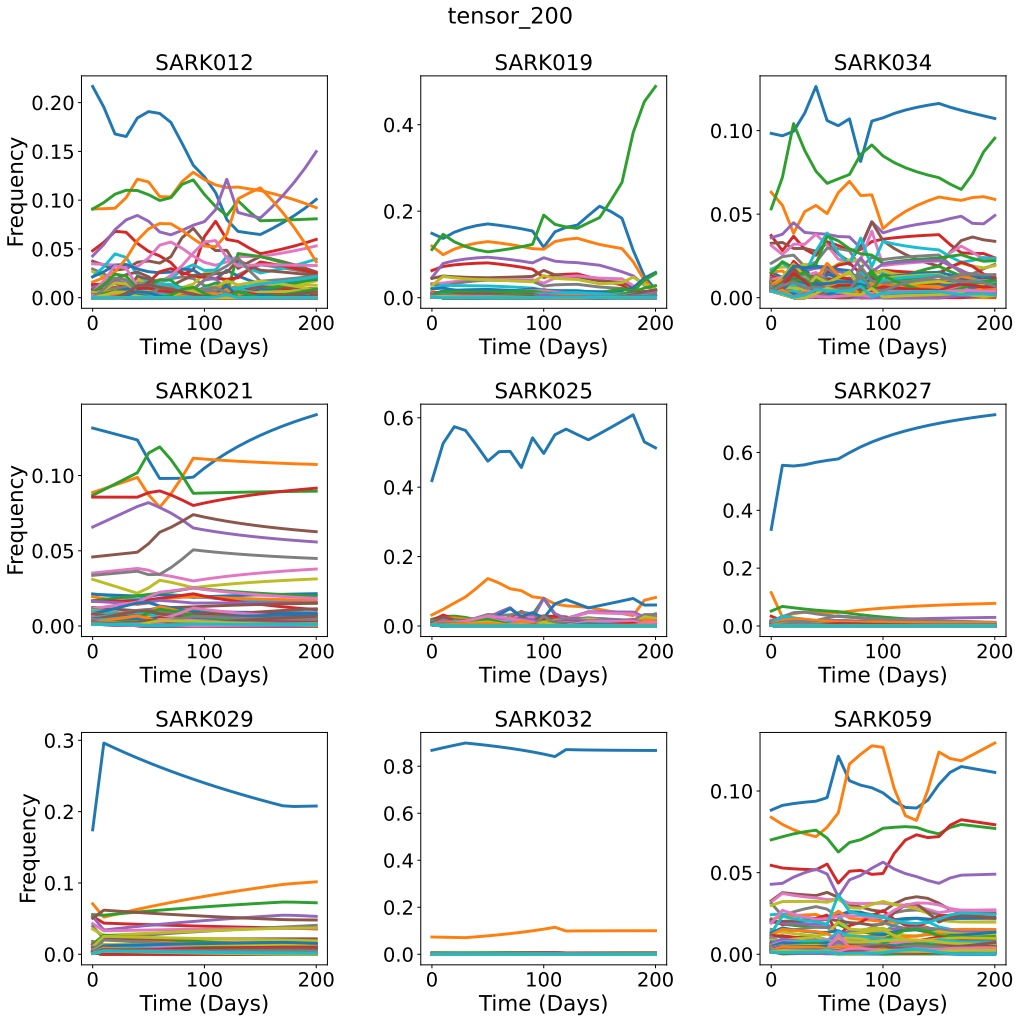


Figure B.7: *Tensor_200* time series

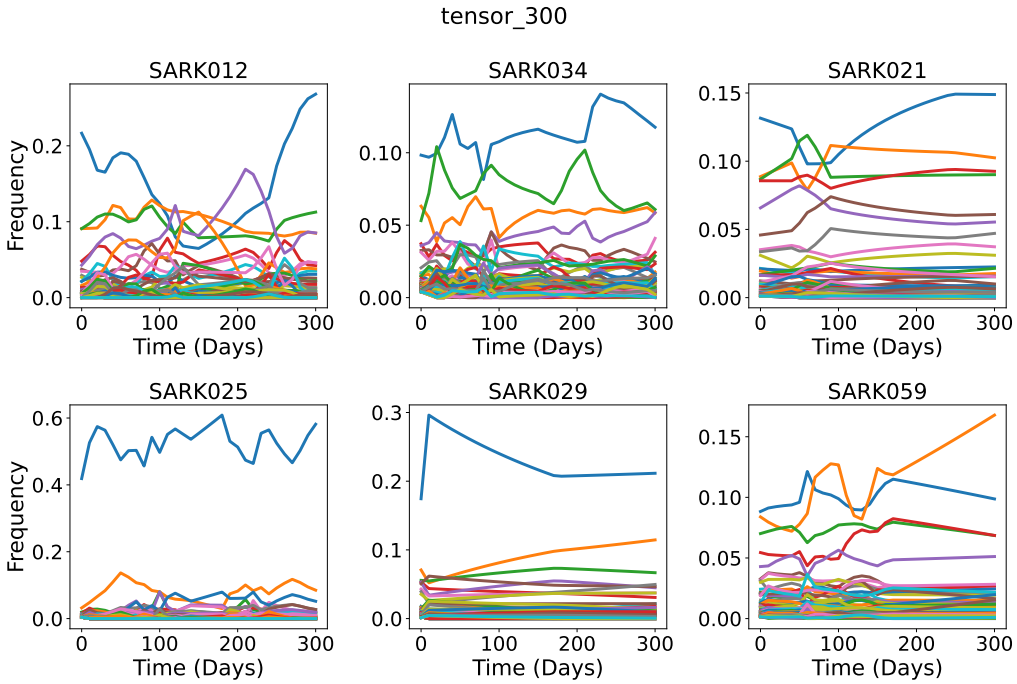


Figure B.8: *Tensor_300* time series

B.3 Ground-truth labels over time

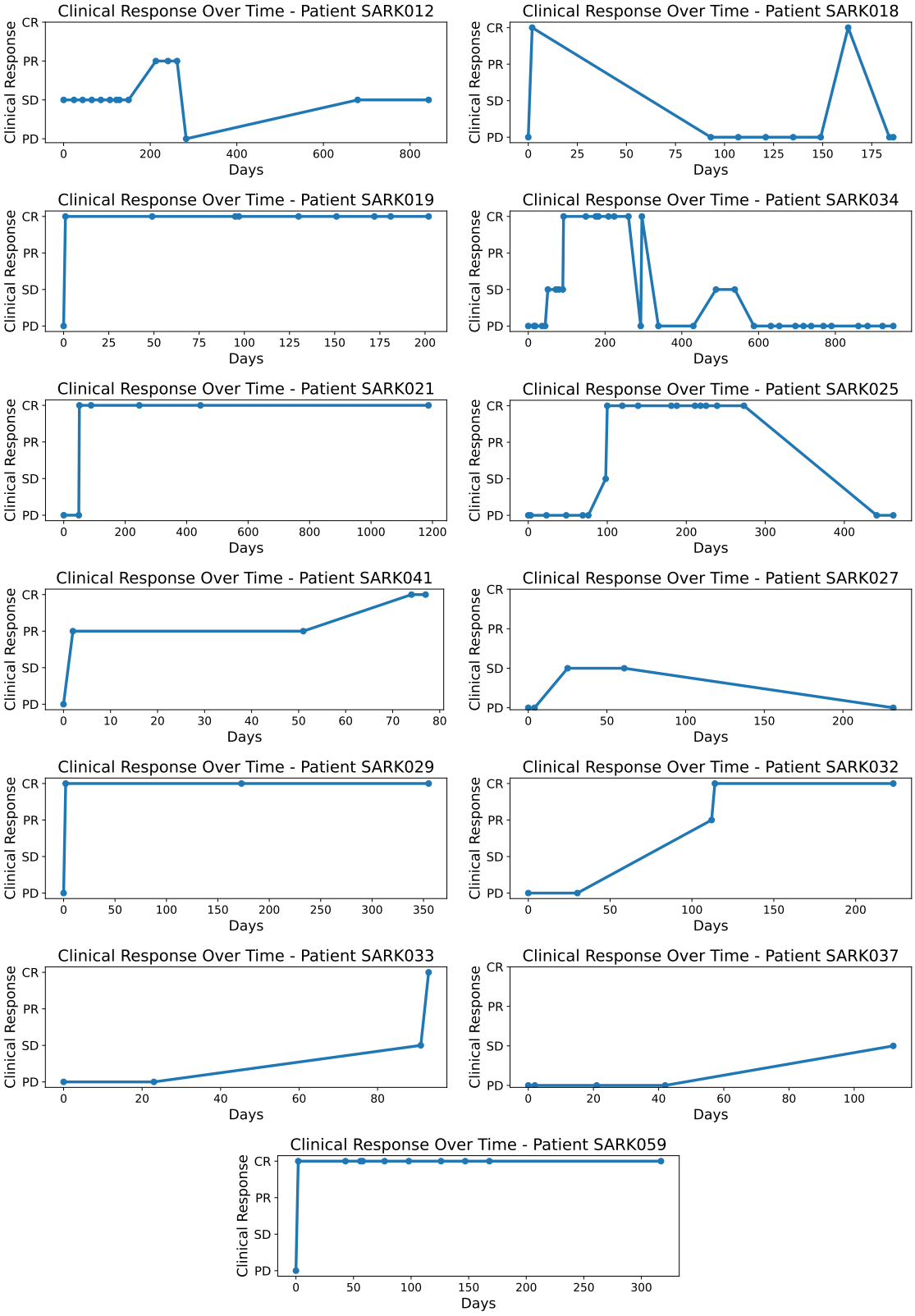


Figure B.9: Clinical ground-truth labels evolution over time for each patient (except excluded ones).

C

Tensor Decomposition Results

This chapter shows other results from the decompositions.

C.1 CP Decompositions

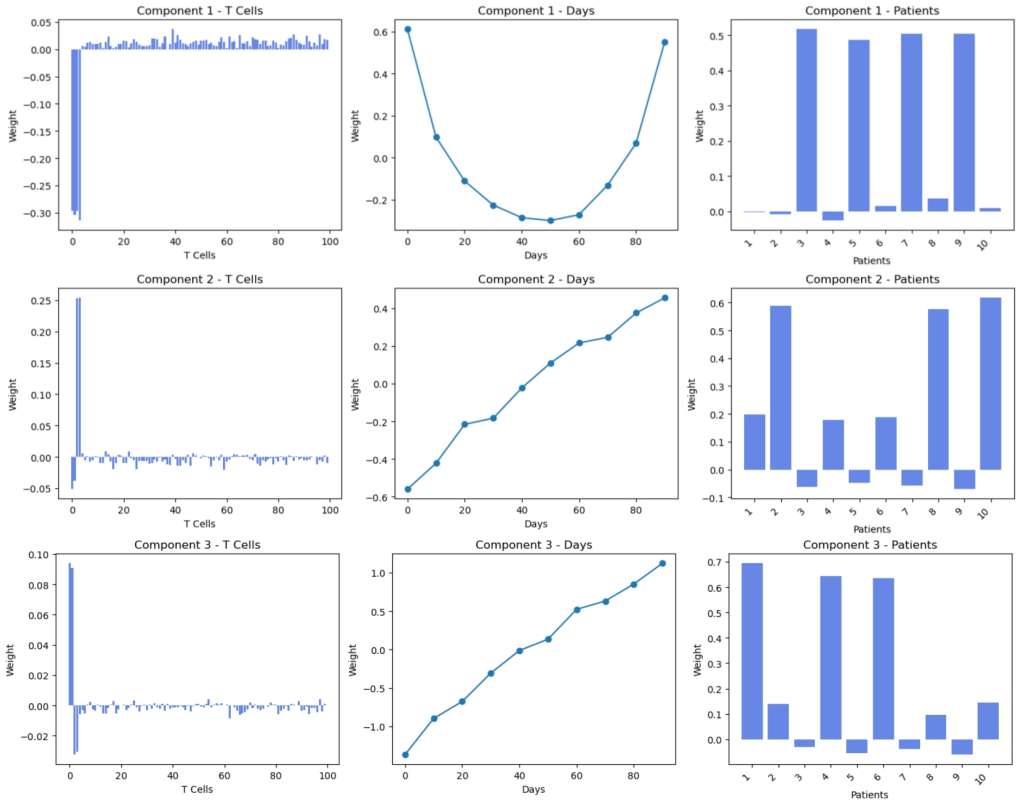


Figure C.1: CP decomposition for *tensor_2* with $R = 3$.

C.2 Tucker Decompositions

For figures about Tucker decompositions, refer to `https://github.com/pablovadillo/Tensordecomposition/tree/main/Results/Tucker%20Decompositions`.

D

Clustering

This chapter shows extra plots regarding the clustering results. For more extra figures, such as figures showing the metric evolution grouped by clusters or complete pairplots of all metrics/components, refer to the corresponding folder from the guide (Appendix A).

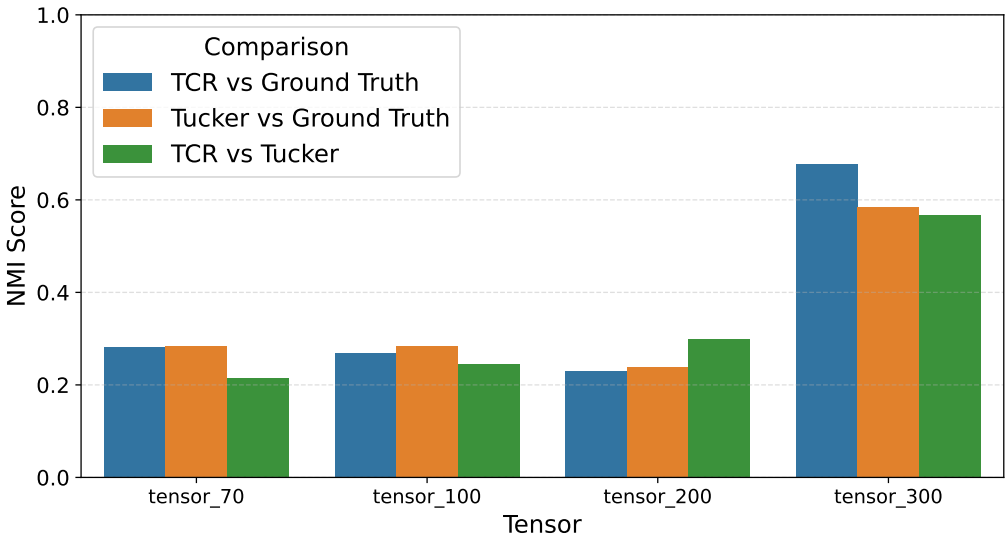


Figure D.1: NMI comparison between clustering methods and ground-truth with $k = 3$.

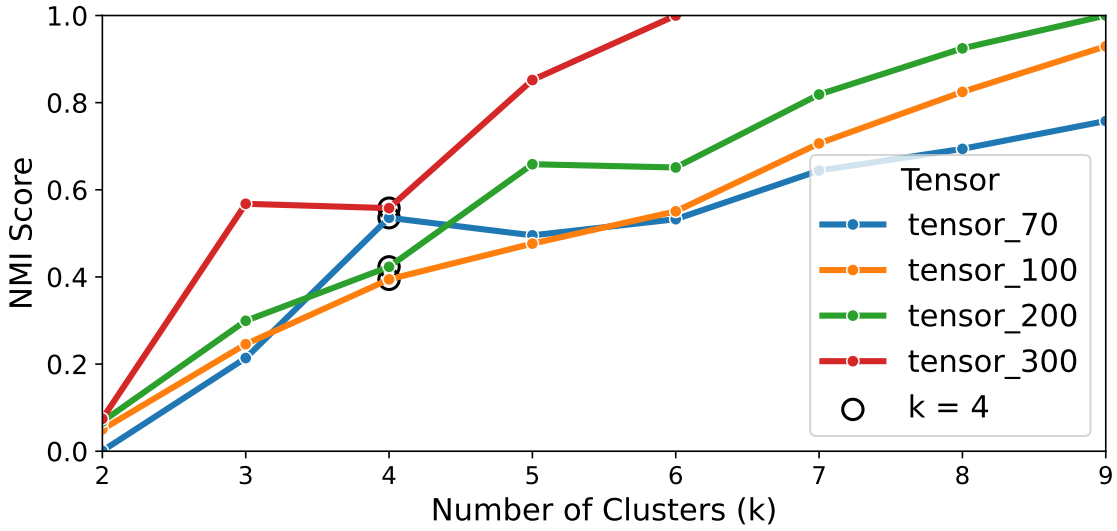


Figure D.2: NMI score comparison between both patient clustering (Tucker vs. TCR metrics) across all tensors for increasing value of k.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY