

Prioritizing TCR Repertoire Network Properties - A Novel Approach to Select Network Signatures

A Thesis submitted to the faculty of

San Francisco State University

In partial satisfaction of the

requirements for

the Degree

Masters of Science

in

Statistics Data Science

by

Shilpika Banerjee

San Francisco, California

December 2022

Copyright by
Shilpika Banerjee
2022

Certification of Approval

I certify that I have read Prioritizing TCR Repertoire Network Properties - A Novel Approach to Select Network Signatures by Shilpika Banerjee and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirement for the degree Masters of Science at San Francisco State University.

Dr. Tao He, Ph.D
Associate Professor
Thesis Committee Chair

Dr. Mohammad R. Kafai, Ph.D
Professor

Dr. Alexandra Piryatinska, Ph.D
Professor

,

Abstract

T-cells are one of the key components of the adaptive immune system. T-cell Receptors (TCR) are a group of protein complexes found on the surface of T-cells. TCRs are responsible for recognizing and binding to certain antigens found on abnormal cells or potentially harmful pathogens. Once the TCRs bind to the pathogens, the T-cells attack these cells and help the body fight infection, cancer, or other diseases. TCR repertoires, which are continually shaped throughout the lifetime of an individual in response to pathogenic exposure, can serve as a fingerprint of an individual's current immunological profile. The similarity among TCRs sequence directly influences the antigen recognition breadth. Network analysis, which allows interrogation of sequence similarity, thereby adds an important layer of information. Due to the heterogeneous nature of TCR network properties, it is extremely difficult to perform statistical inference or machine learning directly between subjects. In this work, a novel method is proposed to prioritize the network properties that are associated with the outcome of interest, based on features extracted from heterogeneous global/local network properties. Schemes to select the top features associated and to simulate the network properties using the real data are also presented. Extensive simulation studies and real data analysis were performed to demonstrate the proposed methods. Performance measures including F-1 score, false discovery rate, sensitivity, power, and stability were calculated for each model and are used for model comparison.

Acknowledgments

I would like to begin by thanking my supervisor Dr. Tao He, whose invaluable guidance and support helped me through each step of my thesis and taking it to completion. Thank you Dr. Tao He for being patient and always present for my questions. I would like to express gratitude to my committee members, Dr. Mohammad R. Kafai and Dr. Alexandra Piryatinska for their support and valuable feedback.

I can never be grateful enough to my friends and family for all that they have done for me. I would like to take this opportunity to thank my friends and family for their unwavering support and encouragement throughout my graduate program. Especially my mother, Mrs. Seema Banerjee, my husband, Dr. Aritra Sengupta, my sister, Nabanita Banerjee and my in-laws. Thank you for being there for me emotionally and intellectually as I have worked on my coursework. I would like to dedicate this work to my father, Late Pronab Kumar Banerjee, who taught me to dream big and to never give up on my pursuits.

Finally, I would like to thank the Mathematics department and the SF Build committee for the Bridge Award and the SF Build Agent of Change scholarships that allowed me to conduct my thesis.

This work was supported in part by National Science Foundation (NSF DMS-2137983) and National Institute of Health (NIH/NCI R21CA264381 and NIH/NLM R01LM013763-01A1).

Table of Contents

List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Background	2
1.2 Motivation and Objective	3
1.3 Challenges	4
1.4 Contribution	5
2 Methods	9
2.1 Aggregating Network Data	9
2.2 Models	12
2.3 Prioritizing Network Properties	14
2.4 Selecting top network features	19
3 Implementation	23
3.1 Insight to the Real Data	23
3.2 Real Data Analysis	25
4 Simulation Study	29
4.1 Data Simulation Scheme	29
4.2 Simulation Results	37
5 Discussion	41
5.1 Inferences from Real Data Analysis	41
5.2 Model Performance Comparison	42
5.3 Significant TCR Network Properties and Features	43
Bibliography	44
Appendices	46

TABLE OF CONTENTS

vii

.1	Appendix A: List of California State University Campuses	46
.2	Appendix B: Abbreviations of California State University Campuses	47

List of Tables

2.1	Aggregated Transitivity data for a Patient	10
2.2	TCR Network Properties, corresponding Feature Blocks, Feature Indexes and Group indexes.	11
2.3	Aggregated Network Data Layout for a Patient	11
4.1	Dependency of some TCR properties on the Node_count	35
4.2	Performance measures of Group Lasso_CV and Group Plasso models.	39
4.3	Performance measures of Lasso_CV and Plasso models.	39
4.4	Performance measures of Exclusive Lasso model.	40

List of Figures

1.1	TCR network properties used for analysis ([2]Miho <i>et al.</i> , 2019).	4
2.1	General Behaviour for Lasso, Group Lasso and Exclusive Lasso.	14
2.2	Permutation Assisted Tuning: The Augmented Model	17
3.1	Network figures for two representative patients	24
4.1	Dependencies and Hierarchy of the Network Properties	30
4.2	Sample TCR network data of a subject showing multiple clusters and nodes. . .	31
4.3	Histogram plot for cluster-count and $\log(\text{cluster-count})$ of the 65 patients' TCR repertoire data.	32
4.4	Histogram plots for the 'Cluster-count' from the original data versus the 'Cluster-count' from the simulated data.	32
4.5	Histogram plots for 'Node-count' from the original data versus the 'Node-count' from the simulated data.	34
4.6	Histogram plots for comparing the density of 'Assortativity' from the observed data and from the simulated data.	35

Chapter 1

Introduction

T cells are crucial components of the adaptive immune system, mediating anti-tumoral immunity and immune response to infections. They are necessary for effective host-response to a wide range of pathogens. T cells are defined by their T cell receptors (TCRs), which are protein complex on T-cell surface. TCRs mount a response to harmful foreign invaders by targeting specific antigens based on nucleotide sequence. TCRs act as the arms of the T cells with memory and can remember harmful pathogens they have seen before, thereby providing a life-long protection which enables a swift response in case of a similar future encounter. Thus, understanding the TCR repertoire could lead to insights regarding immune response pathology while also discovering indicative bio-markers and lead to therapeutic strategies.

As the immune repertoire ages, it is shaped based on the environmental exposure of an individual throughout their lifetime. Therefore, performing statistical inferences directly on TCR data between subjects is challenging due to its heterogeneous nature. In fact,

there is less than 20% overlap across repertoire, even for the same subject. However, it is observed that the similarity among TCRs sequence directly influences the antigen recognition breadth. Therefore, interrogation of TCR sequence similarity can add an important layer of information. This can be achieved through network property analysis of TCR repertoire. A clonal network is constructed where each clone is defined as a node, and then based on the sequence distance (Levenshtein distance), an edge is drawn based on a certain similarity condition (e.g., one letter difference in sequence).

1.1 Background

In this work we analysis the data of 65 patients from the Phase I trial (NCT01693562, 14 September 2012) of durvalumab, an immune checkpoint inhibitor (ICI) designed to activate exhausted tumor-reactive T cells. Durvalumab consolidation therapy is administered to patients with stage III, non-small cell lung cancer (NSCLC) and their immunophenotypic responses are observed. The patients exhibiting increased TCR repertoire diversity on day 15 attained significantly longer overall survival (OS) than those with decreased diversity. Patients with larger TCR clusters showed improved OS than patients with smaller TCR clusters ([1]Elliot Naidus *et al.*, 2021). It was inferred that early TCR repertoire diversification after durvalumab therapy for NSCLC may be predictive of increased survival. Therefore, drawing quantitative analysis of the TCR repertoire in ‘longer overall survival’ and ‘shorter overall survival’ cohorts may provide a better understanding of the immune landscape involving

T cell response. This information can then be used to develop tools to improve patient stratification, prediction of disease outcome, and patient response to treatments.

1.2 Motivation and Objective

The immunophenotypic response data captures the TCR repertoire details for the 65 patients. The TCR network is continually shaping over a patient’s lifetime and is also impacted as a response to the immunotherapy administered to the patient, thereby making the data heterogeneous in nature. The network data captured for our analysis is shown in the Figure 1.1. A total of fifteen network and non-network properties of the TCR repertoire data are used for this work which are referenced as the TCR network properties collectively. Some of these network properties are global and some are clonal (local) network properties ([2]Miho *et al.*, 2019). Another data set representing the overall survival stats for these 65 patients was also referenced. It was also made available that the patients with overall survival months (OS_mon) ≥ 20.3 have a higher survival chance than the other patients. For the analysis, the network properties are used as the explanatory variables and the overall survival month (OS_mon) as the response variable. Patients with $OS_mon \geq 20.3$ are categorized into ‘longer overall survival’ group and patients with $OS_mon < 20.3$ are categorized into ‘shorter overall survival’ group. The objective here is to investigate the TCR repertoire network properties and develop novel statistical method to prioritize the important network properties that are associated with the clinical outcome of increased overall survival.

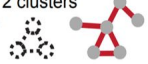









Variables	Network Properties	Definition	Illustration
membership	Cluster size, number	Connected component of a graph in which any two nodes are connected	Number = 2 clusters Size = 3,6 
node_count	Number of Nodes	The fundamental unit of which graphs are formed: v	
deg	Degree	The number of edges incident to a vertex v : $deg(v)$	
AA_length	Non-network properties (Pre-infusion and Dose 2nd are phases when biological samples and clinical data were collected from the subjects.)		
Count_PRE_INFUSION			
Count_DOSE_2			
deg_avg	Average Degree	The average number of degrees per node: $2e/v$	
diam_length	Diameter Length	The length of the "longest shortest path" between any two vertices: $\max_{(v,w)} d(v,w)$	
assortativity	Assortativity	Pearson correlation coefficient of degree between pairs of linked nodes $r \in [-1,1]$	
transitivity	Clustering coefficient (Transitivity)	The probability that the adjacent vertices of a vertex are connected	
edge_density	Density	The ratio of the number of edges and the number of possible edges	
centr_degree	Degree Centrality	Centrality score based on node-level centrality c : $\sum (\max(c(w), w) - c(v), v)$	
centr_clo	Closeness Centrality	Node centrality in a graph: $C(v) = \frac{1}{\sum_w d(v,w)}$	
eigen_centrality	Eigenvector Centrality	Returns the eigenvector centralities of positions v within a given graph	
centr_eigen		A centralization measure	

Figure 1.1: TCR network properties used for analysis ([2]Miho *et al.*, 2019).

1.3 Challenges

The response variable, *OS_mon*, has a definite value for each of the 65 patients. The TCR network data (the explanatory variables) consists of a mix of global and local variables. The global variables are described by a single set of values, while the local variables are vectors of varying lengths. Since the TCR repertoire is constantly adapting to the health and the

environmental factors of the patient, the network properties are continually shaping. Given any two patients the TCR repertoire is never the same. Less than 20% overlap is observed in the TCR repertoires for a same subject. This heterogeneous nature of the TCR repertoire and network properties makes it difficult to perform statistical inference or machine learning directly between subjects. The heterogeneity issue also complicates the data simulation process required to perform the simulation study. Therefore, we require to develop ingenious ways to handle the TCR network data throughout this work and derive meaningful inferences.

1.4 Contribution

In this paper we proposed a strategy to extract features from the heterogeneous global/local network properties. Reading through the distribution of the network properties using the real data, we collect some summary statistics. These derived summary statistics (referred to as the ‘network features’) largely consist of the Minimum, the 1st Quartile (Q_1), the Median, the Mean, the 3rd Quartile (Q_3) and the Maximum values. This technique is uniformly repeated for all the properties. The network properties are then represented by aggregating these summary statistics (referred to as the ‘feature blocks/groups’). This approach helps to sum up the TCR network properties for each patient and renders the data suitable for making statistical inferences.

We then apply variable selection techniques like Lasso ([5]Tibshirani, 1996), Group Lasso , ([3]Yuan and Lin, 2005), and Exclusive Lasso ([6]Zhou, Jin and Hoi, 2010) to prioritize

the T-cell Receptor (TCR) network properties and to select the top network features. The technique of Group Lasso is applied to the feature blocks. This helped with identifying the significant feature blocks. Next, the Lasso and Exclusive Lasso techniques were used to identify the top performing network features. In conjunction with variable selection methods, the cross-validation technique is generally used to render the optimal tuning parameter λ that aids in shrinking and selecting the significant variables. Through this work we presented a comparison of how the cross-validation technique performs against a fairly new technique called the permutation-assisted tuning. This new technique uses a permutation copy of the original data set, with the intention of disrupting the structure existing between the response variable and the explanatory variables. This creates a data set with pseudo-variables that has the same dimensions as the original data set. The final data is widened (predictor space is doubled) by augmenting both the original data set (using the true active variables) and the permutation copy (using the pseudo-variables). The modified data is then fed into the variable selection algorithms to identify a tuning parameter λ such that the pseudo-variables are never selected. The permutation tuning technique is known to have lower false positives than the cross-validation technique. So far, the permutation assisted tuning has been used only on Lasso models. We add novelty in expanding its application to Group Lasso models and present how this technique can be used to identify the significant feature blocks. During this work we also deduced that the permutation assisted tuning method cannot be applied to Exclusive Lasso models. The Exclusive Lasso model bounded by its design selects at least one feature from every existing feature block. Hence, Exclusive Lasso will never converge to

find a λ value which can differentiate between the true variables and the pseudo-variables when using permutation tuning.

We proposed a procedure to simulate the network properties using the real data such that the correlation structure between the predictors are preserved. The network property distributions gave us an insight to some of the correlations that exists among the explanatory variables. Using approximate correlation structures we first simulated a heterogeneous form of the TCR network data and then aggregated it using the summary statistics technique proposed earlier. The simulated data set gave us the flexibility to compare the model performances and ensured the soundness of the results using various performance measures like — Sensitivity, False Discovery Rate, F1 score, Power, and Stability.

We provide the results of our analysis from the original data set, where we are able to capture the significant network properties (feature blocks) and network features (summary statistics). In order to validate those results and compare between the different variable selection methods we perform the simulation study. In this section we present an alternative technique than GPUs to perform large-scale network data simulation. Analyzing the network properties, identifying the superficial dependencies and correlations guides us to simulate the network data. The simulated data is then fed into the previously used variable selection models and performance measures like - Sensitivity, False Discovery Rate, F1 score, Power and Stability are computed. We observe that the permutation assisted tuning method, used to derive the tuning parameter, outperforms the cross-validation technique when performing variable selection. We finally present the set of significant network properties and the top

network features that contribute in determining increased overall survival months in the patients.

Chapter 2

Methods

2.1 Aggregating Network Data

TCR network repertoire varies in structure and sizes across different subjects and are continually shaping even for the same subject. This heterogeneous nature of the TCR repertoire data renders the dataset unfit for direct statistical inferences. The technique used to handle this heterogeneity is accomplished by aggregating the TCR network repertoire data for each patient. Summary statistics of these network properties were deduced per subject which were then considered as explanatory variables. The derived summary statistics largely consisted of Minimum, 1st Quartile (Q_1), Median, Mean, 3rd Quartile (Q_3) and Maximum data points. These summary statistics helped with generating a holistic view of the TCR network properties for the patients belonging to the two cohorts (longer and shorter overall survival).

Certain network properties also contained a significant number of ‘NA’ values. For those properties along with computing the aforementioned set of summary statistics (Min, Q_1 , Median, Mean, Q_3 and Max) for the not null (or ‘NA’) rows, the probability of those value being ‘NA’ was also computed. Feature blocks were then created by collectively considering the summary statistics (including ‘prob(NA)’) from the network data for each patient. This technique allowed us to represent the heterogeneous network data between the two groups of patients efficiently. Using this aggregated form of network data we could then perform comparison of the network properties, run feature selection techniques, and also use this data to aid with simulation of the network property data discussed in the later sections.

Table 2.1: Aggregated Transitivity data for a Patient

prob(NA)	Min	Q1	Median	Mean	Q3	Max
0.698	0	0	0	0.141	0.201	1

Table 2.1 is a representation of the summary statistics of ‘Transitivity’, a network property, which was aggregated for a single patient. ‘Transitivity’ has a large percentage of values as ‘NA’ values, therefore, prob(NA) was also computed. The entire list of summary statistics is provided in Table 2.2. The derived features were indexed individually (as ‘Feature Index’ column) as well as collectively (column ‘Group’) to feed these aggregated features to various variable selection models.

Table 2.2: TCR Network Properties, corresponding Feature Blocks, Feature Indexes and Group indexes.

Properties	Feature Blocks	Feature Index	Groups
membership	# of clusters	1	1
node_count	Min, Q1, Median (Q2), Mean, Q3, Max	2-7	2
deg	Min, Q1, Median (Q2), Mean, Q3, Max	8-13	3
AA_length	Min, Q1, Median (Q2), Mean, Q3, Max	14-19	4
Count_PRE_INFUSION	Min, Q1, Median (Q2), Mean, Q3, Max	20-25	5
Count_DOSE_2	Min, Q1, Median (Q2), Mean, Q3, Max	26-31	6
deg_avg	Min, Q1, Median (Q2), Mean, Q3, Max	32-37	7
diam_length	Min, Q1, Median (Q2), Mean, Q3, Max	38-43	8
assortativity	prob(NA), Min, Q1, Median (Q2), Mean, Q3, Max	44-50	9
transitivity	prob(NA), Min, Q1, Median (Q2), Mean, Q3, Max	51-57	10
edge_density	Min, Q1, Median (Q2), Mean, Q3, Max	58-63	11
centr_degree	Min, Q1, Median (Q2), Mean, Q3, Max	64-69	12
centr_clo	prob(NA), Min, Q1, Median (Q2), Mean, Q3, Max	70-76	13
eigen_centrality	Min, Q1, Median (Q2), Mean, Q3, Max	77-82	14
centr_eigen	prob(NA), Min, Q1, Median (Q2), Mean, Q3, Max	83-89	15

Using the aggregation strategy the entire TCR network data was consolidated in a manner such that a single layer of information exists for each patient. Table 2.3 is a snapshot of the aggregated network data layout for a single patient. The resultant data set consists of 89 network features (individual summary statistics) and 15 network properties (aggregated summary statistics). Based on the variable selection models, either the network features or the network properties would be used.

Table 2.3: Aggregated Network Data Layout for a Patient

	Membership	Node_Count	...	Centr_Eigen
Patient-1	# of clusters	(Min, Q1, Median, Mean, Q3, Max)	(..),...,(..)	(prob(NA), Min, Q1, Median, Mean, Q3, Max)

2.2 Models

Prioritizing the TCR network properties and selecting the most significant network signatures using the the TCR repertoire data is the primary contribution of this work. In the previous section we were able to aggregate the network property data in a systematically manner so that statistical inferences can now be drawn on them. In literature there are three most popular Variable Selection techniques — Filter method, Wrapper method and Embedded method. The Filter method picks up the intrinsic properties of the explanatory variables (i.e., their correlation with the response variable) measured via univariate statistics instead of cross-validation performance. This method ignores feature dependencies and has no interactions with the classification model for variable selection. The Wrapper method attempts to find the optimal variable subset by iteratively selecting the variables based on the classifier performance. This technique even though considers variable dependencies becomes computationally heavy (even impossible) in case of high dimensional feature space and is easily susceptible to overfitting. For our work we consider the Embedded method which combines the qualities of the Filter and the Wrapper methods while overcoming their respective limitations. Embedded method is the most preferred variable selection technique when handling high-dimensional genetic data.

We implement the following embedded techniques and compare their performances — Lasso ([5]Tibshirani, 1996), Group Lasso ([3]Yuan and Lin, 2005), and Exclusive Lasso ([6]Zhou, Jin and Hoi, 2010). The Figure 2.1 gives us a general idea of how these three

variable Selection methods work.

In Lasso the objective function penalizes the absolute size of the regression coefficients, based on the value of a tuning parameter λ . In doing so, Lasso can drive the coefficients of irrelevant variables to zero, thus performing automatic variable selection. We will deploy Lasso for selecting top network features in subsequent sections. Prior to that we implement the Group Lasso model, an extension of the Lasso, that performs variable selection on grouped variables (feature blocks). Group Lasso is known to perform better when predictors are not distinct but arise from common underlying factors. In our case, a set of aggregated features (a feature block) is derived from the same parent network feature which may have some ‘relevance’ among themselves. Using Group Lasso we try to prioritize the prominent network properties as a whole. Finally, we will use Exclusive Lasso to dissect each feature block and find their most significant summary statistics. Figure 2.1 shows how Lasso, Group Lasso and Exclusive Lasso perform variable selection. The Lasso is indifferent to the aggregated group structure (feature blocks) created for the network properties. In the Group Lasso, feature blocks compete among themselves (L_1 norm between feature blocks) and the most significant feature blocks (all sub features inclusive) are selected. The Exclusive Lasso enforces L_2 norm among different features blocks and L_1 norm between the features in a single block. As a result at least one feature is selected from each of the feature blocks.



Figure 2.1: General Behaviour for Lasso, Group Lasso and Exclusive Lasso. Block_(i) represents the i^{th} feature block (TCR network property) and F_{ij} represents the summary statistics derived for the i^{th} feature block.

2.3 Prioritizing Network Properties

We use the Group Lasso variable selection technique to identify the significant TCR network properties (feature blocks) that help to differentiate between patients belonging to the longer and the shorter overall survival cohorts.

2.3.1 Logistic Group Lasso Model set-up

Assume that we have independent and identically distributed observations (\mathbf{x}_i, y_i) where $i = 1, \dots, n$ of a p -dimensional vector $\mathbf{x}_i \in \mathbb{R}^p$ and a binary $y = (y_1, \dots, y_n)^T$ as the binary outcomes ($y_i \in 0, 1$) for all the n observations.. For this work, $n = 65$ (# of patients) and $p = 15$ (# of explanatory variables). The response variable $y_i = 1$ denotes $OS_{mon} \geq 20.3$

- patient belongs to the ‘longer overall survival’ group, and $y_i = 0$ denotes $OS_{mon} < 20.3$

- patient belongs to the ‘shorter overall survival’ group. The predictors can be written as $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ which are then grouped (based on some known common factors between the features) to form a total of G groups of predictors. In this study, the summary statistics derived from a single network property are grouped together to form a feature block. The degrees of freedom is denoted as df_g of the g^{th} predictor group. The predictor variables can be rewritten as $\mathbf{x}_i = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{iG})^T$ using the grouped variables $\mathbf{x}_{ig} \in \mathbb{R}^{df_g}$ where $g = 1, \dots, G$.

Applying the latter representation of the predictor variables, the log of odds (logit) for the logistic regression model can be written as:

$$\log\left\{\frac{p_{\boldsymbol{\beta}}(\mathbf{x}_i)}{1 - p_{\boldsymbol{\beta}}(\mathbf{x}_i)}\right\} = \eta_{\boldsymbol{\beta}}(\mathbf{x}_i) \quad (2.1)$$

where $p_{\boldsymbol{\beta}}(\mathbf{x}_i) = \mathbb{P}_{\boldsymbol{\beta}}(y_i = 1|\mathbf{x}_i)$ and $\eta_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \sum_{g=1}^G \mathbf{x}_{ig}^T \beta_g$. Here $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_G)^T$ are the coefficients for the G grouped predictors.

The logistic group lasso estimator $\hat{\boldsymbol{\beta}}_{\lambda}$ is derived by minimizing the objective function:

$$S_{\lambda}(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + \lambda \sum_{g=1}^G s(df_g) \|\beta_g\|_2^1 \quad (2.2)$$

where $l(\cdot)$ is the log-likelihood function and is given as

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n (p_{\boldsymbol{\beta}}(\mathbf{x}_i))^{y_i} (1 - p_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

$$l(\boldsymbol{\beta}) = \log[L(\boldsymbol{\beta})] = \sum_{i=1}^n [y_i \eta_{\boldsymbol{\beta}}(\mathbf{x}_i) - \log(1 + \exp(\eta_{\boldsymbol{\beta}}(\mathbf{x}_i)))] \quad (2.3)$$

and the penalty function is given as

$$\lambda \sum_{g=1}^G s(df_g) \|\beta_g\|_2^1 \quad (2.4)$$

In the equation (2.2), the tuning parameter $\lambda \geq 0$ controls the amount of penalization and is subject to $\sum_{g=1}^G \|\beta_g\|_2^1 \leq \theta(\lambda)$ where $\theta(\cdot)$ is some function of λ . In the penalty function equation (2.4), $\|\beta_g\|_2^1$ implies L_1 -norm inter-group and L_2 -norm intra-group. The function $s(\cdot)$ is used to rescale the penalty with respect to the dimensionality of the parameter vector β_g ([3]Yuan and Lin, 2006).

An optimal value for the tuning parameter λ can be derived using the Cross-Validation approach. Cross-validation is a statistical method for estimating machine learning model performance (or accuracy) through resampling. It is used to prevent overfitting in a predictive model, especially when the amount of data available is limited. The k-fold cross-validation technique involves dividing the whole data into k sets of almost equal sizes. The first set is selected as the test set and the model is trained on the remaining (k-1) sets. The test error rate is then calculated after fitting the model to the test data. This is referred to as the first ‘sub-problem’. The process is continued for all the k sets selecting a different test set each time and then averaging the overall error estimate. For hyperparameter tuning the best value of λ is not known therefore, the optimal value of λ is determined using the cross-validation technique. Different values of λ are used with each ‘sub-problems’ and the λ value which gives the lowest test error is chosen. λ .

In this thesis another approach called the permutation assisted tuning is presented alongside cross-validation for hyperparameter tuning. Until now permutation assisted tuning has been used only on Lasso regression model ([4]Yang *et al.*, 2020) to perform large scale genome-wide association studies-GWAS ([2] Miho *et al.*, 2019). This work presents the extension of

permutation assisted tuning to Group Lasso model.

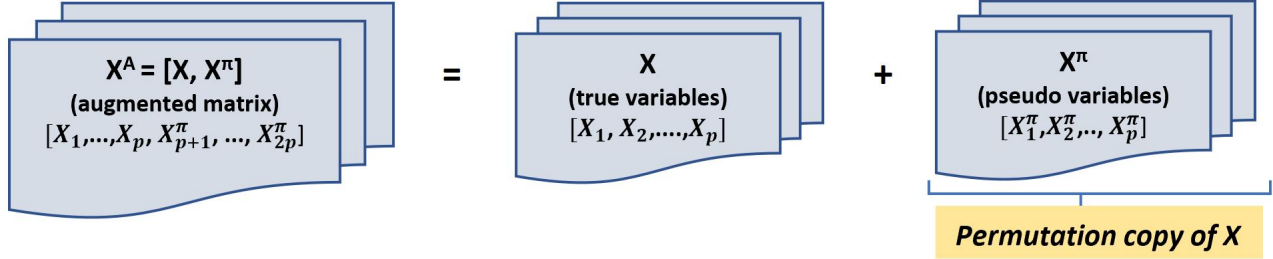


Figure 2.2: Permutation Assisted Tuning: The Augmented Model

True set of predictor variables (dim: $\mathbf{x}_i \in \mathbb{R}^p$) is augmented with its permutation copy (dim: $\mathbf{x}_i^\pi \in \mathbb{R}^p$) to form an augmented design matrix \mathbf{X}^A (dim: $\mathbf{x}_i^A \in \mathbb{R}^{2p}$).

Figure 2.2 illustrates the idea of creating a permutation copy (set of pseudo-variables) from the original set of predictor variables and then augmenting them to increase the overall dimension of the predictor set while keeping the sample size constant. The new set of predictors for the augmented matrix can be written as $\mathbf{x}_i^A = (x_{i1}, \dots, x_{ip}, x_{i(p+1)}^\pi, \dots, x_{i(2p)}^\pi)^T$ which are then grouped (based on the same common factors between the features used for the original predictor set) to now form a total of $2G$ groups of predictors. The predictor variables for the augmented matrix can be rewritten as $\mathbf{x}_i^A = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{iG}, \mathbf{x}_{i1}^\pi, \dots, \mathbf{x}_{iG}^\pi)^T$ using the grouped variables $\mathbf{x}_{ig} \in \mathbb{R}^{\text{df}_g}$ where $g = 1, \dots, G$. Note that the grouping of the original predictor variables and that of the pseudo-variables remain the same, i.e. $[(1, 2, \dots, G) = (G + 1, G + 2, \dots, 2G)]$

The logistic group lasso estimator $\hat{\beta}_\lambda^A$ for this augmented design matrix is derived by minimizing the new objective function which is created by leveraging the equations (2.2)

and (2.3):

$$S_\lambda(\boldsymbol{\beta}^A) = - \sum_{i=1}^n [y_i \eta_{\boldsymbol{\beta}^A}(\mathbf{x}_i) - \log(1 + \exp(\eta_{\boldsymbol{\beta}^A}(\mathbf{x}_i)))] + \lambda \sum_{g=1}^{2G} s(\text{df}_g) \|\beta_g^A\|_2^1 \quad (2.5)$$

In equation (2.5), the new tuning parameter $\lambda \geq 0$ controls the amount of penalization. Because of the way the pseudo-variables were constructed we know that the contribution of these variables should be absent from the model. Therefore, it is important that a preferred tuning parameter λ should be able to rule out the coefficients that are assigned to the pseudo-variables making them active predictors ([4]Yang *et al.*, 2020). We consider the point λ on the group lasso path at which the variable group \mathbf{x}_g where $g \in 1, \dots, G$ remains in (or last enters in terms of decreasing λ 's) the model but does not allow \mathbf{x}_{g+1} where $g+1 \in G+1, \dots, 2G$ to entire the model.

$$W_g = \sup\{\lambda : \hat{\beta}_g^A(\lambda) \neq 0\}; g = 1, \dots, 2G \quad (2.6)$$

We set $W_g = 0$ if a variable group does not enter the model even without the group lasso penalty (i.e., $\hat{\beta}_g^A(0) = 0$). This new statistic W_g can be viewed as an importance metric for the g^{th} variable group, as an active variable tends to remain longer in the model as the penalty λ increases compared to an inactive feature block.

Since these pseudo-variables are known to be inactive, a preferred tuning procedure should be able to rule out those λ parameters that identify a pseudo-variable as active. Motivated by this, in terms of increasing λ 's, C_π is defined as $C_\pi = \max_{(G+1) \leq g \leq 2G} (W_g)$ as a benchmark to separate active variables from inactive ones. That leads to the following

variable selection procedure

$$\hat{S}_\pi = \{g : W_g > C_\pi, g = 1, \dots, G\} \quad (2.7)$$

In other words, only those original variables will be selected which have importance metric W_j greater than C_π , the maximum of importance metrics of all p pseudo-variables. In terms of decreasing λ 's, we iterate through all the λ values until the point where pseudo-variables are assigned coefficients. That is $W_g \neq 0$ where $g \in G + 1, \dots, 2G$. We then use the λ value prior to the λ identified in the previous step. This technique guarantees that only the original active variables make into the model.

In equation (2.7), \hat{S}_π denotes the estimator of true active variables under a particular permutations π . Since the permutation copies will render different selection results, each time this variable selection model is executed, the \hat{S}_π will be affected by the different permutations. The variable selection is stabilized by using different permutations iteratively and evaluating the frequency of the selected variables across the iterations. This technique of using Group Lasso with permutation assisted tuning henceforth would be referred to as the 'Group Plasso' model.

2.4 Selecting top network features

After prioritizing the network properties (feature blocks) we explore the top network features across the entire set of predictors and dive deeper to identify the significant aggregated feature from each of the feature blocks.

2.4.1 Logistic Lasso Model set-up

The Lasso shrinkage and variable selection technique is used on the observed data to identify the top network features. The various features compete using the L_1 -norm to produce a significant set of variables. Lasso is indifferent to any group structures created for the network properties. The objective function $S_\lambda(\boldsymbol{\beta})$ defined using the negative log likelihood penalizes the absolute size of the $\boldsymbol{\beta}$ coefficients based on the value of the tuning parameter λ . The logistic lasso estimator $\hat{\boldsymbol{\beta}}_\lambda$ is derived by minimizing the objective function:

$$S_\lambda(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + \lambda \sum_{j=1}^p |\beta_j| \quad (2.8)$$

where $l(\cdot)$ is the log-likelihood function given as

$$l(\boldsymbol{\beta}) = \sum_{i=1}^n [y_i \eta_{\boldsymbol{\beta}}(\mathbf{x}_i) - \log(1 + \exp(\eta_{\boldsymbol{\beta}}(\mathbf{x}_i)))] \quad (2.9)$$

and the penalty function is given as

$$\lambda \sum_{j=1}^p |\beta_j| \quad (2.10)$$

The tuning parameter $\lambda \geq 0$ controls the amount of penalization and is subject to $\sum_{j=1}^p |\beta_j| \leq \theta(\lambda)$. The term $|\beta_j|$ in the penalty function implies L_1 -norm between the variables. For a given value of λ , a certain number of variables with non-zero coefficients can be selected. For a typical lasso solution path $\hat{\boldsymbol{\beta}}_\lambda$, more variables can enter the model when λ decreases. As the value of λ increases, the irrelevant coefficients cease to exist leading to automatic variable selection.

Optimal solution for the tuning parameter λ can be deduced using different techniques. The most preferred approach is by using Cross-Validation. In our work we use Cross-Validation and the Permutation Assisted Tuning to derive the optimal λ . The latter approach has lower false positives ([4]Yang *et al.*, 2020) in comparison to the Cross-Validation approach.

Permutation assisted tuning for Lasso, also known as Plasso ([4]Yang *et al.*, 2020), works in the same manner as it does for Group Lasso. The only difference lies in the way that Group Lasso groups the original and the pseudo-variables, whereas, for Lasso the original and the pseudo-variables remain un-grouped. After augmentation of the permutation copy (pseudo-variables), the corresponding lasso problem is

$$\boldsymbol{\beta}^A(\lambda) = \operatorname{argmin}_{\boldsymbol{\beta}^A} \left[- \sum_{i=1}^n [y_i \eta_{\boldsymbol{\beta}^A}(\mathbf{x}_i) - \log(1 + \exp(\eta_{\boldsymbol{\beta}^A}(\mathbf{x}_i)))] + \lambda \sum_{j=1}^{2p} |\beta_j^A| \right] \quad (2.11)$$

where $\boldsymbol{\beta}^A = (\beta_1^A, \dots, \beta_p^A, \beta_{p+1}^A, \dots, \beta_{2p}^A)^T$ are the coefficients for both the p original variables and p pseudo-variables.

2.4.2 Exclusive Lasso Model set-up

Exclusive Lasso ([6]Zhou, Jin and Hoi, 2010) is used to make inferences on the aggregated features from each feature block. It makes variables in the same group compete with every other variable within the same group and thus generates sparse solutions (within a feature block). The objective function $S_\lambda(\boldsymbol{\beta})$ for exclusive lasso is defined similar to that of Group

Lasso.

$$S_\lambda(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + \lambda \sum_{g=1}^G s(\text{df}_g) \|\beta_g\|_1^2 \quad (2.12)$$

where the penalty term is given as

$$\lambda \sum_{g=1}^G s(\text{df}_g) \|\beta_g\|_1^2 \quad (2.13)$$

The term $\|\beta_g\|_1^2$ implies L_2 -norm among the groups and L_1 -norm within each group. The $l(\cdot)$ is the log likelihood function for logistic regression and the function $s(\cdot)$ is used to rescale the penalty with respect to the dimensionality of the parameter vector β_g ([3]Yuan and Lin, 2006). The logistic exclusive lasso estimator $\hat{\boldsymbol{\beta}}_\lambda$ is derived by minimizing the objective function from the equation (2.12). The tuning parameter $\lambda \geq 0$ controls the amount of penalization of the model and is subject to $\sum_{g=1}^G \|\beta_g\|_1^2 \leq \theta(\lambda)$, where $\theta(\cdot)$ is some function of λ .

The optimal value of the tuning parameter λ can be derived by using Cross-Validation. However, unlike Lasso and Group Lasso, permutation assisted tuning cannot be used to deduce lambda for Exclusive Lasso. Since exclusive lasso applies L_2 -norm across all groups (generates dense solution across groups and sparse solution within a group), it is bound to assign coefficients to all groups. As a result, when pseudo-variables are added to the model, exclusive lasso in all situations will generate coefficients for the groups containing the pseudo-variables. Hence, we can never achieve the desired model by excluding the groups containing the pseudo-variable in this case. Therefore, we will limit our usage to Cross-Validation approach for finding λ in the Exclusive Lasso model.

Chapter 3

Implementation

3.1 Insight to the Real Data

The original TCR network repertoire data consisted of two parts. The first data set comprises list of all the patients, their respective socio-demographic information (example: age, gender, race, ethnicity, smokers etc.), and their treatment responses (example: treatment status, overall survival months, treatment phase, etc.) The first data set is homogeneous in nature. We extract the ‘Patient Id’ and the ‘Overall Survival Months’ from this data set. The second data set expands the TCR network properties for each patient. This data set captures the changes in the network properties of each of the 65 patients during the Phase I trial of the consolidation therapy of durvalumab. As a result, this data is heterogeneous in nature. We apply the previously proposed technique (Chapter 2 section 2.1, Aggregating Network Data) of aggregating the network properties by extracting their significant summary

statistics. We then consolidate the TCR network data for each patient and their respective ‘Overall Survival Months’ (OS_mon), where the latter is considered as the response variable. Patients with $OS_mon \geq 20.3$ are known to show a higher chances of survival in comparison to those with $OS_mon < 20.3$. The Figure 3.1 shows the difference in the TCR network structures of two patients — one with $OS_mon = 2.73 (< 20.3)$ and another with $OS_mon = 31.28 (\geq 20.3)$. This gives the impression that difference in the network structures is indicative of different overall survival. Therefore, based on the response variable (OS_mon) we run the variable selection techniques to identify the most significant network signatures that distinguish between the two cohorts.

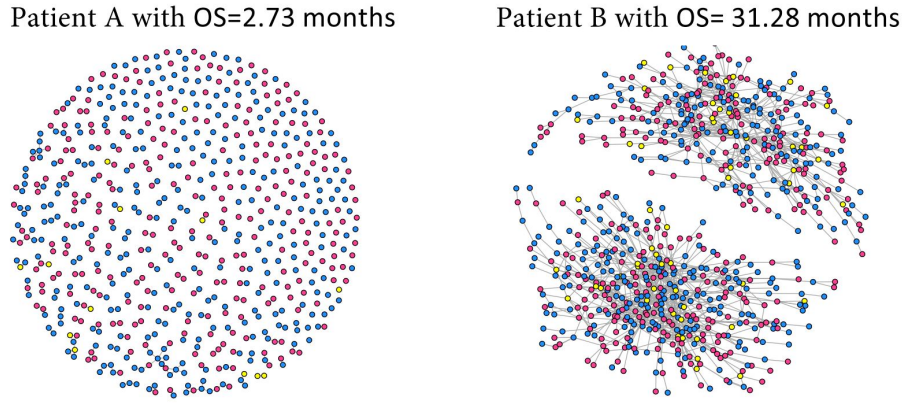


Figure 3.1: Network figures for two representative patients

The above figure represents the TCR network diversity of two patients who participated in the drug trial. The network structures of the two patients are indicative of the differences in network complexity with respect to the OS_mon values.

3.2 Real Data Analysis

3.2.1 Group Lasso_CV and Group Plasso

The technique of Group Lasso is applied to the aggregated data with the objective to prioritize the network properties. This requires the network features to be grouped based on some underlying common factors. In our case, we aggregated those network features together that are derived from a single network property. Referring to Table 2.2, a total of 15 groups are created for the 15 TCR properties. The summary statistics (network features) are aggregated based on the network properties from which they were derived and then are consolidated to form feature blocks. The set of 15 feature blocks are used to build the Group Lasso models.

Prior to fitting the real data to the Group Lasso model, we use Cross-Validation (CV) technique to find the optimal value for the tuning parameter λ . The cross-validated technique generates two λ values — λ_{min} : λ value corresponding to the minimum mean cross-validated error; λ_{1SE} : largest value of λ such that error is within one standard error of the cross-validated errors of λ_{min} . Since $\lambda_{1SE} \geq \lambda_{min}$, the model using λ_{1SE} has higher test error but smaller model than the model built using λ_{min} . In this study $\lambda_{1SE} = \lambda_{min}$. Using λ_{min} value as the tuning parameter for the Group Lasso model, the prominent network properties are identified. Based on the observed data for the 65 subjects, the group indexes - 1, 5, 6 are selected as the significant properties. Note that each of these group indexes represent a unique feature block. Refer Table 2.2. The groups - 1, 5, 6 correspond to the ‘Membership’,

‘Count_PRE_INFUSION’ and ‘Count_DOSE_2’ TCR network properties. The Group Lasso model with tuning parameter derived using the cross-validation technique is referred to as the ‘Group Lasso_CV’ model. Similarly, the Group Lasso model which uses the permutation assisted tuning for finding the optimal tuning parameter is referred to as the ‘Group Plasso’ model.

The real data is then used to feed the Group Plasso model. The original predictor variables compete with pseudo-variables (permutation copy) to determine the significant active predictors. The results from this technique is stabilized by using 10 different permutations iteratively, and evaluating the frequencies of the selected variables across those iterations. The output gives us Group - 1, 5 as the significant properties. These groups correspond to ‘Membership’, ‘Count_PRE_INFUSION’ TCR network properties.

We observe overlap in the TCR properties selected using Group Lasso_CV and Group Plasso. The common groups indexes are - 1 and 5.

3.2.2 Lasso_CV and Plasso

To identify the top network features the Lasso variable selection model is used on the ungrouped set of data that consists of 89 summary statistics. Lasso using cross-validation technique will be referred to as the ‘Lasso_CV’ model and the one using the permutation assisted tuning will be referred to as the ‘Plasso’ model.

Similar to the technique discussed in the previous Group Lasso models, first, Cross-

Validation (CV) approach is used to find the optimal value for the tuning parameter λ . Using λ_{min} value as the tuning parameter, the Lasso_CV model extracts the top network features significant in distinguishing between the two cohorts ('longer overall survival' and 'shorter overall survival'). Feature indexes - 25, 43, 82, 89 (refer Table 2.2) are rendered as the top features. These feature indexes correspond to the summary statistic of the Maximum values of the TCR properties 'Count_PRE-INFUSION', 'diam_length', 'eigen_centrality', 'centr_eigen'. The real data is then fed into the Plasso model which identifies the top features across 10 iterations using different permutation copies.

Both Lasso_CV and Plasso models generated the exact same set of features as the most significant ones.

3.2.3 Exclusive Lasso_CV

Subsequently, Exclusive Lasso model is used to identify the top features from each of the network feature blocks. When using this model, the aggregated summary statistics, 15 feature blocks are used as the explanatory variables. The results from the Exclusive Lasso model should be able to reaffirm the network features selected using Lasso_CV and Plasso models from their respective feature blocks. That is the output from the Lasso_CV and Plasso models should be a subset of the output from the Exclusive Lasso model.

An optimal value of the tuning parameter $\lambda = \lambda_{1SE}$ for exclusive lasso is derived using the cross-validation approach. Similar to the group lasso model, the real data is grouped

(refer Table 2.2). The exclusive lasso model selects the most influential features from each of the true feature blocks. It is observed that the network features selected using the Lasso_CV and Plasso models are also selected using the Exclusive Lasso.

Note that permutation assisted tuning method does not work for exclusive lasso, since the model bounded by its nature will always choose at least a single feature from every feature block. As a result, exclusive lasso is unable to differentiate between the true and the pseudo-variable groups therefore, fails to perform feature selection using permutation copies.

Chapter 4

Simulation Study

4.1 Data Simulation Scheme

Simulating network data to assess the different variable selection methods and to reproduce the experiments is vital. Data can be synthetically generated using various network simulation software tools which may require GPUs to perform large-scale simulations faster and to gain deeper insights. We present an alternative technique for simulating the TCR network data and then compute performance measures to assess the robustness, accuracy, and other parameters of the variable selection models implemented earlier. We analyze each of the network properties from the observed data, try to find any implicit relations, approximate the property distributions, and recreate the data correlation among the explanatory variables in the simulated data set. The original TCR repertoire data was heterogeneous in nature which led us to extract the summary statistics for each feature and then create

an aggregated data set. Since any correlation that exists among the network features is observed in the heterogeneous (non-aggregated) form of the data, it therefore vital for us to simulate the data in the non-aggregated form and then derive the summary statistics.

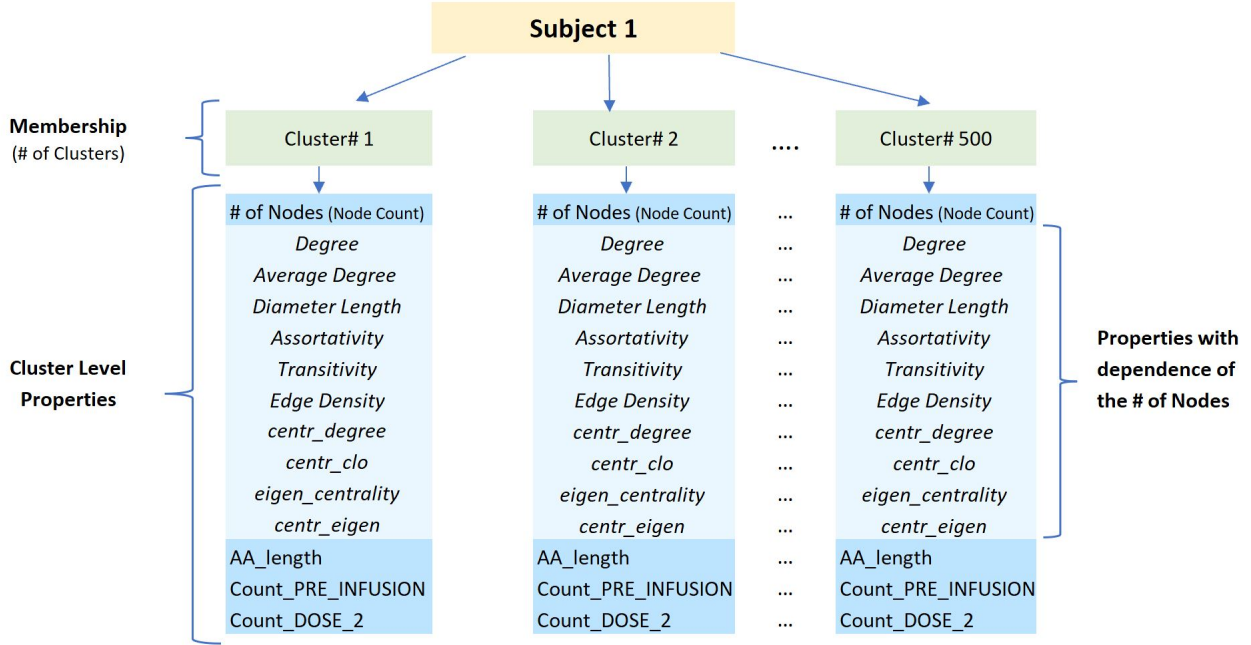


Figure 4.1: Dependencies and Hierarchy of the Network Properties

The above figure give a general idea of how the TCR network properties are dependent on each other and a rough hierarchical structure. This information was derived by analysing the real data.

A high-level hierarchy of the network properties as observed in the real data is shown in the Figure 4.1. The TCR repertoire network data of each subject has multiple clusters (membership) of varied levels of complexity. Some clusters could be dense (have more nodes) and some could be sparse (have fewer nodes). Refer Figure 4.3. Each cluster then has a varied number of nodes and associated nodal properties. As a result, simulating this network

data first requires generating the clusters for a subject. Then for each cluster we simulate the # of nodes and the associated network features.

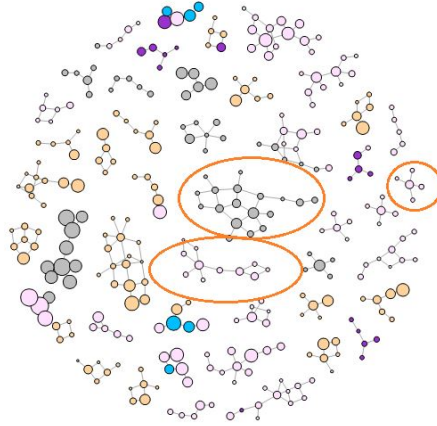


Figure 4.2: Sample TCR network data of a subject showing multiple clusters and nodes.

4.1.1 Simulating Cluster-count (Membership) data

The number of clusters for each of the 65 patients were observed from the original TCR repertoire data. Figure 4.3 represents the distribution of the cluster-count and the $\log(\text{cluster-count})$ of the original TCR repertoire data. The $\log(\text{cluster-count})$ has an approximate normal form. We then compute the mean and the standard deviation of this distribution. Using inbuilt R function, `rlnorm()`, we draw random samples from a log normal distribution, with $\text{meanlog} = \text{mean}(\log(\text{cluster-count}))$ and $\text{sdlog} = \text{sd}(\log(\text{cluster-count}))$, we generate samples which are then used as the cluster-count for 1000 dummy patients.

TheFigure 4.4 represents the histogram of the cluster-count from the original observed

data versus the cluster-count from the simulated data. The plots shows the semblance between the original data and the simulated data. The simulated cluster-count data serves as the basis for simulating the remaining properties.

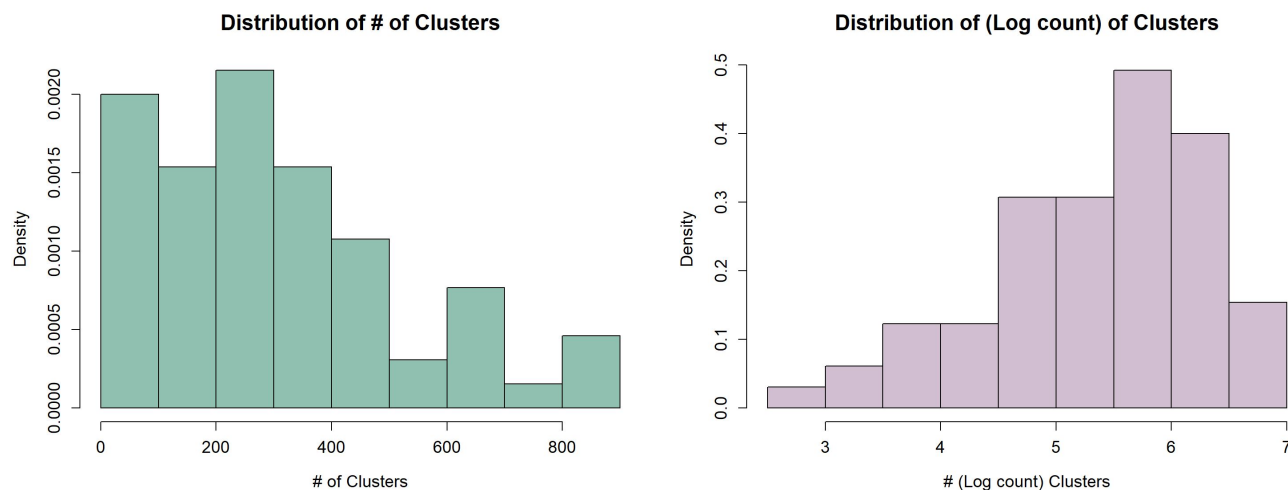


Figure 4.3: Histogram plot for cluster-count and $\log(\text{cluster-count})$ of the 65 patients' TCR repertoire data.

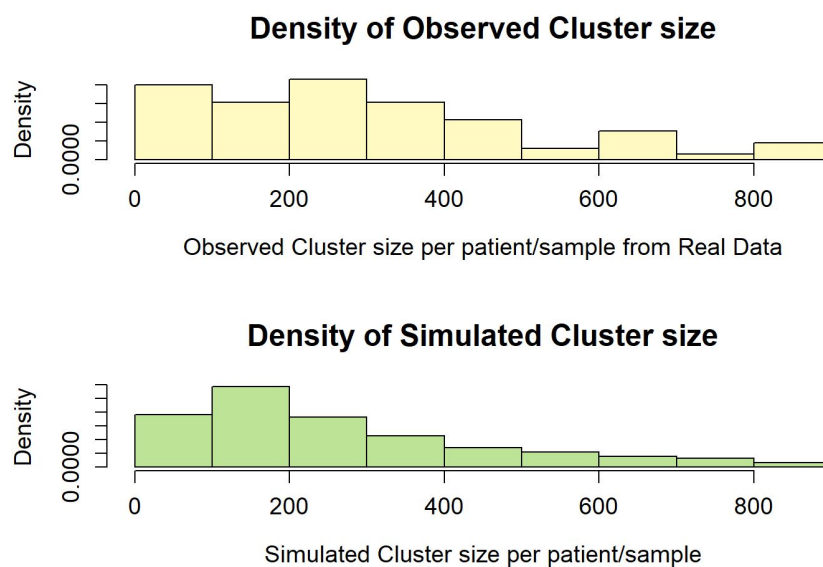


Figure 4.4: Histogram plots for the 'Cluster-count' from the original data versus the 'Cluster-count' from the simulated data.

4.1.2 Simulating Node-count data

The simulated cluster-count data dictates the number of TCR network property rows that each of the 1000 dummy patients would have. We then set to simulate the node-count associated with each of the cluster. It is observed that the node-count data from the original TCR repertoire has a right-skewed distribution with maximum frequency occurring for the values 2 and 3, and the remaining subsection has an approximate log-normal form. Therefore, to simulate the node-count, the entire distribution is considered in segments. We calculated the probabilities of the node-count = 2 and 3 and simulate data for these two categories. For the remaining segment of the distribution, we draw samples from the *rlnorm()* function using a similar technique as we did for simulating the cluster-count data. The Figure 4.5 depicts the histogram of the node-count from the original observed data versus the simulated data where we can observe the similarity in the two density functions. We then derive the summary statistics for the simulated node-count data and will use the aggregated data for assessing the various models in subsequent sections.

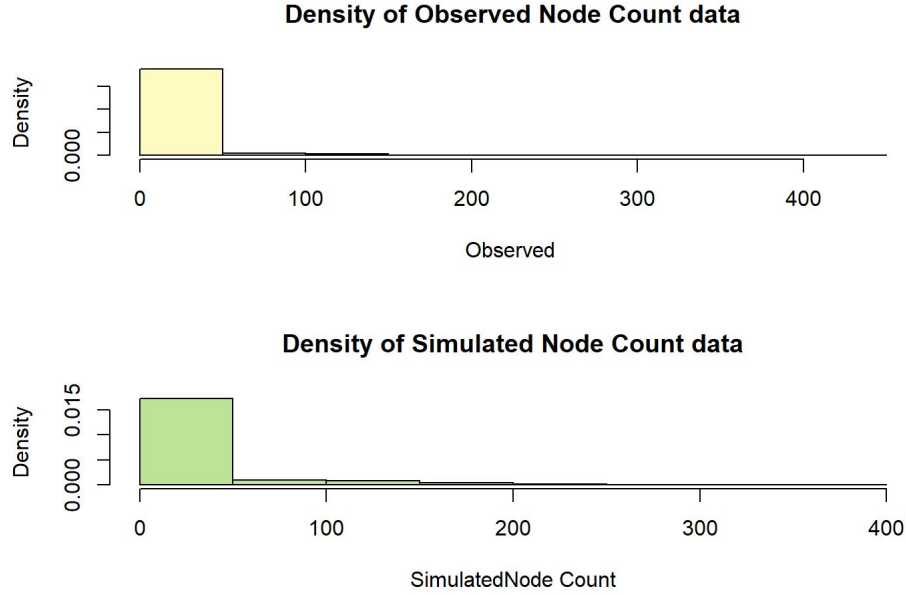


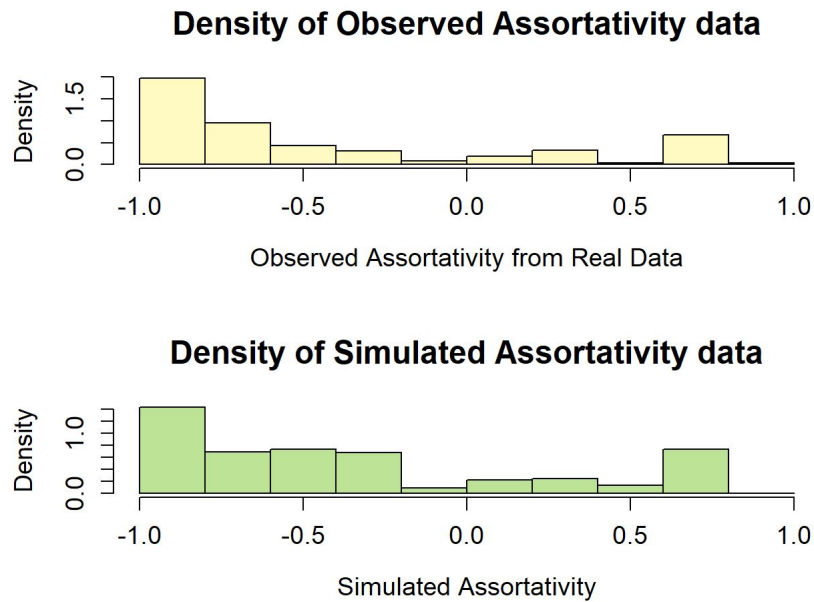
Figure 4.5: Histogram plots for ‘Node-count’ from the original data versus the ‘Node-count’ from the simulated data.

4.1.3 Simulating Remaining Network Properties

After the node-count data is simulated, the remaining set of network properties can be simulated. It is observed that some of the properties have a distinct value for certain values of Node-count. The Table 4.1 outlines some of these relations. Probabilities of the node-count values = 2 and 3 evaluated in the Simulating Node-count data sub-section are used to simulate the data for the remaining network properties. Following this, the simulated row-wise data is aggregated for each of the network properties, resulting into a single row of data per dummy patient. Figure 4.6 represents the histogram comparison of the observed versus simulated data of the network property called ‘Assortativity’.

Table 4.1: Dependency of some TCR properties on the Node_count

TCR properties	Values	Values
Node_count	2	3
Assortativity	NA	-1 (99.7% times)
Transitivity	NA	-
Degree	1	-
Diam_length	2	-
Deg_avg	1	-
Edge_density	1	-
Centr_degree	0	-
Centr_clo	NA	-
Eigen_centrality	1	-
Centr_eigen	NA	-

**Figure 4.6:** Histogram plots for comparing the density of ‘Assortativity’ from the observed data and from the simulated data.

4.1.4 Simulating Response Variable

In order to simulate the response variable, while keeping the correlation between the explanatory variables and the response intact, we use the model from the sub-section Lasso-CV and Plasso. In a binary classification setting with response variable $y_i \in 0, 1$ assume that we have independent and identically distributed observations (\mathbf{x}_i, y_i) where $i = 1, \dots, n$ of a p -dimensional vector $\mathbf{x}_i \in \mathbb{R}^p$. The log of odds is given as:

$$\log\left\{\frac{p_{\boldsymbol{\beta}}(\mathbf{x}_i)}{1 - p_{\boldsymbol{\beta}}(\mathbf{x}_i)}\right\} = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p = \eta_{\boldsymbol{\beta}}(\mathbf{x}_i)$$

where $p_{\boldsymbol{\beta}} = \mathbb{P}_{\boldsymbol{\beta}}(y_i = 1|\mathbf{x}_i)$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^T$ are the coefficients for the p predictors.

Using the Lasso model from subsection 3.2.2 where we obtained feature indexes [25, 43, 82, 89] as the significant variables. We use these as the true causal variables X_1, \dots, X_4 for simulating the response variable y_i 's where $i \in 1, \dots, 1000$. The log of odds $\eta_{\boldsymbol{\beta}}(\mathbf{x}_i)$ and $p_{\boldsymbol{\beta}} = \mathbb{P}_{\boldsymbol{\beta}}(y_i = 1|\mathbf{x}_i)$ can be written as:

$$\eta_{\boldsymbol{\beta}}(\mathbf{x}_i) = \beta_0 + \beta_1 X_1 + \dots + \beta_4 X_4 \quad (4.1)$$

$$p_{\boldsymbol{\beta}} = \mathbb{P}_{\boldsymbol{\beta}}(y_i = 1|\mathbf{x}_i) = \frac{\exp(\eta_{\boldsymbol{\beta}}(\mathbf{x}_i))}{1 + \exp(\eta_{\boldsymbol{\beta}}(\mathbf{x}_i))} \quad (4.2)$$

where the $\boldsymbol{\beta}$ coefficients are artificially created. We then sample randomly from a Bernoulli distribution using the function `rbern()` such that 1000 random samples are generated with probability $= p_{\boldsymbol{\beta}}$. These random samples are then used as simulated response values for our study.

4.2 Simulation Results

The simulated network property data is aggregated to produce summary statistics for each subject and is indexed similar to what was done for the original data. The simulated data also consists of 89 network features (individual summary statistics) and 15 network properties (feature blocks deduced by aggregating the summary statistics). The data set is then down-sampled to accommodate for the data imbalance. We assess the Lasso_CV, Plasso, Group Lasso_CV, Group Plasso, and Exclusive Lasso models using the simulated data. Performance measures are computed by iteratively running the variable selection models on the simulated data.

4.2.1 Evaluation Criteria

We used the following measures - Sensitivity, False Discovery Rate (FDR), F1 score, Power, and Stability to evaluate the performances of the different variable selection models. A false-positive finding means the selection of any predictor variable except the causal variables by the feature selection method. Sensitivity is defined as the proportion of correctly identifying causal variables among all causal variables in a single iteration. FDR is defined as the frequency of false-positive findings among all variables selected per method and iteration. F-1 score is the harmonic mean of the sensitivity and (1-FDR). It is a measure of performance accuracy of the model. The power of each causal variable is calculated as the frequency of correct selections among all iterations. To estimate stability of a variable

selection model, all pair-wise combinations of the selected variables list from each iteration are considered. For each pair, the stability of the two lists of selected variables is determined using the Jaccard's index given as: $J(A_i, A_j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$, where A_i is the list of variables selected in the i^{th} iteration and A_j is the list of variables selected in the j^{th} iteration such that $i \neq j$ and $i, j \in 1, 2, \dots, I$ (total number of iterations). The Jaccard's index, $J(A_i, A_j) = 0$ if the two lists do not overlap, and $J(A_i, A_j) = 1$ if the two lists contain the same variables. The average of all pairs is used as stability value for the particular method.

4.2.2 Simulation Results from Group Lasso_CV and Group Plasso

The assessment of the Group Lasso_CV and the Group Plasso models on the simulated data is done using the true causal variables derived using the Lasso_CV model (refer subsection 3.2.2) on the original data, as the 'gold-standard'. The causal network properties (feature blocks) that correspond to the top network features are derived from subsection 3.2.2. The performance measures for the simulation study on Group Lasso_CV and Group Plasso models are shown in the Table 4.2. It is observed that both Group Lasso_CV and Group Plasso models have high sensitivity which is desirable. The FDR value for the Group Plasso model is lower than that of the Group Lasso_CV model which is as expected. The Group Plasso model however has a much higher stability than the Group Lasso_CV model.

Table 4.2: Performance measures of Group Lasso_CV and Group Plasso models.

Models	True Group Indexes	Sensitivity	FDR	F1	Power	Stability
Group Lasso_CV	Group - 5, 8, 14, 15	0.775	0.5324	0.5597	0.8, 1.0, 0.8, 0.5	0.5911
Group Plasso	Group - 5, 8, 14, 15	0.7	0.45297	0.6094	1.0, 1.0, 0.7, 0.1	0.8123

4.2.3 Simulation Results from Lasso_CV and Plasso

Consider the top network features derived from the Lasso_CV model (refer subsection 3.2.2) as the ‘gold-standard’ for the causal variables. The Lasso_CV and the Plasso models are assessed using these true causal variables. It is observed that the Plasso model out performs the Lasso_CV model in terms of the FDR and F1 score. The remaining performance measures are at par. Refer Table 4.3.

Table 4.3: Performance measures of Lasso_CV and Plasso models.

Models	True Feature Indexes	Sensitivity	FDR	F1	Power	Stability
Lasso_CV	25, 43, 82, 89	1	0.5065	0.6594	1.0, 1.0, 1.0, 1.0	0.9212
Plasso	25, 43, 82, 89	1	0.2533	0.8533	1.0, 1.0, 1.0, 1.0	0.9111

4.2.4 Simulation Results from Exclusive Lasso_CV

For assessing the Exclusive Lasso model on the simulated data we use the top network features, derived using the Lasso_CV model (refer subsection 3.2.2), as the ‘gold-standard’ for the causal variables. It is observed that Exclusive Lasso consistently identifies the true causal variables in every iteration and therefore has a perfect sensitivity value. This result asserts the findings from the Lasso models. Note that Exclusive Lasso shows a very high

FDR and low F1 values. This observation is due to the compulsion of the Exclusive Lasso model to select at least one feature from each of the feature blocks.

Table 4.4: Performance measures of Exclusive Lasso model.

Models	True Feature Indexes	Sensitivity	FDR	F1	Power	Stability
Exclusive Lasso.CV	25, 43, 82, 89	1	0.8	0.3333	1, 1, 1, 1	1

Chapter 5

Discussion

All analyses and implementations were done using the R programming language with base packages and the subsequent analysis-specific packages. The code is available on **GitHub** for reference.

5.1 Inferences from Real Data Analysis

This thesis focuses on the application of variable selection techniques like Lasso, Group Lasso and Exclusive Lasso using different hyperparameter tuning techniques — cross-validation and permutation assisted tuning. Using the original sample data, the Group Lasso-CV model was able to prioritize three of the network properties (feature blocks) — ‘Membership’ (# of Clusters), ‘Count_PRE_INFUSION’, and ‘Count_DOSE_2’. The Group Plasso model selected only two of the aforementioned network properties — ‘Membership’ (# of

Clusters) and ‘Count_PRE-INFUSION’. The difference in the output of the two models can be attributed to the characteristics of the permutation tuning of having lower false positives ([4]Yang *et al.*, 2020) in comparison to the cross-validation technique. Therefore, the Group Plasso model appears to be more stringent than Group Lasso_CV in performing variable selection on the network feature blocks. When using Lasso_CV and Plasso both models selected the same set of network features as the top performing variables — the ‘maximum’ summary statistics for the TCR network properties ‘Count_PRE-INFUSION’, ‘diam.length’, ‘eigen_centrality’, ‘centr_eigen’. Exclusive Lasso (using only cross-validation) model reaffirmed the findings from Lasso_CV and Plasso by selecting the same set of network features from their corresponding feature blocks.

5.2 Model Performance Comparison

From the simulation study on the Group Lasso_CV and Group Plasso models it is observed that the Group Plasso model has a higher F1 score values. Given that the F1 score aggregates the results from the ‘Sensitivity’ and the ‘FDR’ values, higher F1 score is desirable. The Group Plasso model has a stability of $\sim 81\%$ while Group Lasso_CV has only $\sim 59\%$ stability. This shows that permutation assisted tuning improved the model stability significantly over cross-validation technique.

Similarly, when comparing the Lasso_CV and the Plasso models, the latter has a much lower FDR value and hence a higher F1 score. This aligns with the advantage of permutation

assisted tuning having lower false positives than cross-validation technique. Being able to lower the false positives while having the other performance measures almost similar to that of Lasso_CV, makes the Plasso model a preferable choice for variable selection than Lasso_CV.

Formally verification is required to prove whether the performance of permutation assisted tuning is superior than that of cross-validation for all variable selection problems.

5.3 Significant TCR Network Properties and Features

Referring to the output from the Table 4.2, the aggregated list of group indexes selected by both Group Lasso_CV and Group Plasso models are - ‘Membership’ (# of Clusters), ‘Count_PRE_INFUSION’, and ‘Count_DOSE_2’. These feature blocks correspond to the most significant TCR network properties. Similarly, using Lasso_CV and Plasso models on the original data identify the ‘maximum’ summary statistics for ‘Count_PRE_INFUSION’, ‘diam_length’, ‘eigen_centrality’, ‘centr_eigen’ emerge as the top performing TCR network features.

Bibliography

- [1] Elliot Naidus, Jerome Bouquet, David Y. Oh, Timothy J. Looney, Hai Yang, Lawrence Fong, Nathan E. Standifer, Li Zhang. “Early changes in the circulating T cells are associated with clinical outcomes after PD-L1 blockade by durvalumab in advanced NSCLC patients”. In: *Cancer Immunology, Immunotherapy* 70:2095–2102 (2021).
- [2] Enkelejda Miho, Rok Roskar, Victor Greiff and Sai T.Reddy. “Large-scale network analysis reveals the sequence space architecture of antibody repertoires”. In: *Nature Communications* 10:1321 (2019).
- [3] Ming Yuan and Yi Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society. Series B* 68.Part 1 (2006), pp. 49–67.
- [4] Songshan Yang, Jiawei Wen, Scott T. Eckert, Yaqu Wang, Dajiang J. Liu, Rongling Wu, Runze Li1 and Xiang Zhan. “Prioritizing genetic variants in GWAS with lasso using permutation-assisted tuning”. In: *Bioinformatics* 36:3811-7 (2020).

- [5] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1996), pp. 267–288.
- [6] Yang Zhou, Rong Jin, Steven C. H. Hoi. “Exclusive Lasso for Multi-task Feature Selection”. In: *JMLR Workshop and Conference Proceedings: 13th International Conference on Artificial Intelligence and Statistics (AISTATS)* 9 (2010), pp. 988–995.

.1 Appendix A: List of California State University

Campuses

- California State University, Bakersfield
- California State University Channel Islands
- California State University, Chico
- California State University, Dominguez Hills
- California State University, East Bay
- California State University, Fresno
- California State University, Fullerton
- Humboldt State University
- California State University, Long Beach
- California State University, Los Angeles
- California State University Maritime Academy
- California State University, Monterey Bay
- California State University, Northridge
- California State Polytechnic University, Pomona

- California State University, Sacramento
- California State University, San Bernardino
- San Diego State University
- San Francisco State University
- San José State University
- California Polytechnic State University, San Luis Obispo
- California State University San Marcos
- Sonoma State University
- California State University, Stanislaus

.2 Appendix B: Abbreviations of California State University Campuses

- CSU Bakersfield
- CSU Channel Islands
- Chico State
- CSU Dominguez Hills

- Cal State East Bay
- Fresno State
- Cal State Fullerton
- Humboldt State
- Cal State Long Beach
- Cal State LA
- Cal Maritime
- CSU Monterey Bay
- CSUN
- Cal Poly Pomona
- Sacramento State
- Cal State San Bernardino
- San Diego State
- San Francisco State
- San José State
- Cal Poly San Luis Obispo

- CSU San Marcos
- Sonoma State
- Stanislaus State