1 Introduction

While a properly functioning immune system prevents illness by recognizing nonself antigens as foreign, a malfunctioning immune system can recognize self antigens as foreign causing autoimmune diseases such as Rheumatoid Arthritis (RA). In recent years immune therapies have been proposed that attempt to treat autoimmune diseases such as RA by shifting the relative balance between inflammatory and regulatory immune response in favor of the regulatory populations. For example, sustained delivery of chemokines [12, 20], cytokines [11, 19] and small molecule inhibitors [1, 19] can modulate immune cell function (e.g. dendritic cells, T cells) in inflamed tissues to **resolve** RA and other autoimmune disease outcome in pre-clinical animal models. However, the effect of the immunotherapy regimen is influenced by factors such as timing, dosage, and the current balance of inflammatory/regulatory response in the patient - thus making identification of effective treatment standards a challenging problem.

For this reason, there is a growing need for a observable measure of immune system health which can be used for the prediction and prevention of RA and other autoimmune diseases [5, 10, 13]. However, the question of identifying observables is complicated by our relative lack of understanding of how the immune system determines self vs non-self and the number of potential observables which have been identified as contributing to function of the immune system. To clarify the problem at hand, we therefore propose two relatively uncontroversial theses.

First, we presume that the question of identification of observables for prediction of autoimmune disease progression cannot be decoupled from the question of modeling, since in the absence of a predictive model, there is no way to verify that a certain set of observables can be used for prediction. That is, for any proposed set of observables, there must exist an associated predictive model with some associated accuracy in predicting autoimmune disease progression. Second, we presume that the immune system is deterministic in that the self-nonself decision (and hence autoimmune disease progression) is governed by a dynamical process wherein the relative populations of immunogenic and regulatory cells and molecules evolve over time and that the relative balance of these populations directly influences the establishment or elimination of autogenic response in autoimmune disease. That is, we presume that, given a method for modelling the immune system, there exists a set of observables capable of effectively predicting the process of self-nonself determination.

Given these assertions, we can propose three necessary components of any process for identification of observables with clinical predictive power. First, we require a method for modeling based on given set of observables. While such a model may be based on physical principles, such a model may also be derived from data-based methods such as machine learning. Second, we require a way to test suitability of the predictive model associated with any given set of observables. Specifically, this test of suitability may include predictive accuracy of the associated model, along with other metrics such as clinical feasibility and robustness to patient variation. Finally, we require a methodology for

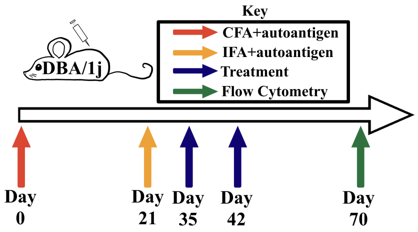


Figure 1: A graphical description of the experimental procedure of inducing and treating RA in mice. The first two steps induce RA, the next two steps is the application of the treatment and the final step is the data generation using flow cytometry. CFA = complete Freund's adjuvant, IFA = incomplete Freund's adjuvant.

selection and rejection of observables in order to obtain a set of observables with maximal suitability as defined previously. In this paper, we consider each of these requirements: using experimental data and a variety of machine learning algorithms to generate models; defining an appropriate metric for suitability; and using feature selection algorithms to find a set of observables with maximal suitability. Once we have addressed these requirements, we apply the proposed methodology - arriving at a set of maximally suitable observables, which we define as the "immune state". An outline of our approach to addressing these required subproblems is listed below.

For the first problem, in Section 2, we initially define our immunological dataset obtained from ongoing trials of RA immunotherapy. Then, in Section 3, we define our approach to modelling. Specifically, we define a set of machine learning algorithms which uses a given subset of data observables to identify both all other observables as well as RA outcome - as measured by severity of inflammation.

For the second problem, in Section 4, we propose a dual metric for suitability of a given set of observables based partially on predictive power of the associated model. The first part of this metric is based on minimality (not prediction), wherein we impose a penalty based on the number of observables in the set (cardinality) in order to reduce experimental and clinical complexity. Second, in order to ensure that relevant immunological data is not lost, we also add a penalty based on the error of the associated model to predict observables from the data not included in the given set. Third, to measure efficacy of the prediction, we impose a penalty based on the error in prediction of RA severity - a quantity we refer to as the "disease state".

For the third problem, in Section 5, we use a variety of feature selection algorithms to determine the set of observables which are optimally suited using the suitability metric described above. We then, in Section 6, apply the resulting algorithms to our dataset and propose a set of maximally suitable observables, which we define as the "immune state".

2 A mouse model of rheumatoid arthritis and associated observables

The goal of this paper is to propose a methodology for identifying observable measures for immune system health. To better illustrate this methodology, we consider the approach as applied to a particularly rich dataset obtained from an ongoing series of experiments involving the use of biomaterials-based particles [17] containing metabolites that promote self tolerance in intermediate/late stage RA in a DBA/1j mouse model which develops severe arthritis when immunized with bc2 autoantigen. In this experimental series, the particles were synthesized either with or without auto-antigen bc2 - a strategy designed to determine if the particles can generate AG-specific anti-inflammatory response. An overview of the experimental procedure is provided in Fig. 1. The chronology of the experiment is listed here in detail. The data collection used for model generation occurs exclusively on Day 70.

Day 0 and 21: RA was induced in mice to generate an autoimmune response for the development of severe polyarthritis. On day 35, the mice were divided into 3 groups, each receiving a distinct therapeutic regimen.

Group 0 - Days 35/42: The control group consists of 5 control mice, each receiving two subcutaneous injections of phosphate buffered saline (PBS) near the hind legs on days 35 and 40.

Group 1 - Days 35/42: Treatment group 1 consists of 5 mice. Each mouse receives two injections of 0.5 mg of biomaterials-based particles without embedded auto-antigen bc2 near the hind legs on days 35 and 40.

Group 2 - Days 35/42: Treatment group 1 consists of 8 mice. Each mouse receives two injections of 0.5 mg of biomaterials-based particles with embedded auto-antigen bc2 near the hind legs on days 35 and 40.

Measurements Taken on Days 62/70: Paw thickness measurements are used to determine arthritic scores for all mice and were obtained either on day 62 or 70, and are defined on the interval $[0, 5]$. Furthermore, flow cytometry was preformed on cells collected from the popliteal lymph node, cervical lymph node and spleen of each mouse on day 62 or 70. The flow cytometry procedure stained for CD4, CD8, Ki67 (proliferation), CD25 (activation), Foxp3 (Treg transcription factor (TF)), Tbet (Th1/Tc1 TF), GATA3 (Th2/Tc2 TF), RORyT (Th17/Tc17 TF), CD44 (memory marker), CD62L (memory marker), and a tetramer that is specific to the autoantigen. Based on this staining, we identified 41 different combinations of markers which might be used to classify the phenotype of a T cell and determined the percentage of either CD4 or CD8 T cells presenting the associated combination of markers.

Summary of Associated Dataset: The data consists of 84 samples based on 18 mice, each sample is associated with a mouse and sample location, all samples are taken on day 62/70, and each sample consists of 43 features and one label. The first two features of each sample indicate group number (0-2) and sample location (1-3). The remaining 41 features defining the percentage (0-100) of the CD4/CD8 population exhibiting the associated combination of markers. The label for each sample is the arthritic score (0-5).

Based on this data, in the following section, we will propose several methods of machine learning to construct predictive models which use subsets of the features to predict both label and remaining features. For generating these models, all features are scaled to the interval $[0, 1]$.

3 Predictive Model Generation via Machine Learning Algorithms

In the previous section, we provided a dataset consisting of a large number of features (43) and a single label (disease state). As discussed in the introduction, to identify clinically significant observables, we will use a metric of suitability combined with a feature selection algorithm to determine which observables have the most predictive power. However, the use of such feature selection algorithms requires a procedure for using a subset of the features to predict both the remaining features and the label. In this section, therefore, we define several state-of-the-art algorithms capable of generating predictive models from given data. Specifically, we focus on ML algorithms for solving the problem of regression.

Suppose we are given a dataset of m samples, wherein each sample $\{x_i, y_i\}$ defines a set of features $\{x_i \in \mathbb{R}^n\}_{i=1}^m$ and an associated label $\{y_i \in \mathbb{R}\}_{i=1}^m$. The regression problem, then, is to find a predictive model, $f : \mathbb{R}^n \to \mathbb{R}$ which minimizes the predictive errors $f(x_i) - y_i$ in an appropriately defined metric. However, this metric and the resulting optimization problems vary significantly between algorithms. In the following subsections, we define several state-of-the-art machine learning algorithms which will be combined with feature selection algorithms in Sections 5 and 6 to determine features with the most predictive power. Finally, we note that in the context of feature selection algorithms, when only a subset of the available features are used, the remaining features become labels.

ML Algorithms for Regression:

In this section, we define five ML algorithms for potential use in combination with feature selection algorithms, including advantages and disadvantages of each.

Before beginning, we note that the choice and tuning of ML algorithms is something more of an art than a science. Specifically, we want to avoid overfitting the training data - thus allowing our predictive models to perform well on unlabelled data. To this end, each of the ML algorithms we define has an associated set of “regularization parameters” which should be selected through a some ad hoc process. These tuning parameters will then affect how well the resulting predictive model will generalize to unlabeled data. In each case, therefore, we specify these parameters. However, we do not define how these parameters are chosen until Section 6, as this process will vary depending on the dataset.

In each case below, we assume the data set contains m samples, $\{x_i, y_i\}_{i=1}^m$, each with n features, $x_i \in \mathbb{R}^n$ and a label $y_i \in \mathbb{R}$.

Regularized Linear Regression (LR) The regularized linear regression algorithm returns a predictive model $y = f(x) = w^T x + b$, where w solves the following optimization problem.

$$\min_{w \in \mathbb{R}^n} \quad \sum_{i=1}^{m} (y_i - w^T x_i - b)^2 + \alpha_2 ||w||^2 + \alpha_1 ||w||.$$

In this case, $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$ are the regularization parameters. Linear regression has the advantage of low computational complexity. However, the resulting predictor is linear and if the underlying physical process is nonlinear, accuracy of the predictive model will be poor.

ϵ-loss Support Vector Regression (SVR) The support vector regression problem uses a predictive model has the form $f(x) = \sum_{i=1}^m \alpha_i k(x, x_i)$ where $\alpha \in \mathbb{R}^m$ is the decision variable and k is a user selected positive kernel function. The objective function being minimized includes $\sum_i |f(x_i) - y_i|$ for any i such that $|f(x_i) - y_i| \geq \epsilon$, where ϵ is a tuning parameter. In addition, there is a regularization parameter, C where regularization increases as C decreases. SVR can generate accuracte nonlinear predictive models for appropriate choice of k. However, the selection of the kernel heavily influences the resulting accuracy and this process of selection is difficult to automate.

Tessellated Kernel Learning (TKL) The TKL algorithm (and kernel learning algorithms in general) improves on the SVR problem by automating the search for a kernel function. Note we consider that the class of kernel learning algorithms to include Deep Learning (although the search problem in this case is non-convex). These approaches are limited, however, by the class of kernels over which they are able to search. The class of Tesselated Kernels has been shown in [7] to have the properties of universality, density, and tractability - meaning the resulting algorithms are rather accurate and generalize well to new data. Specifically, the TKL algorithm was shown in [8] to be more accurate and more robust than all other tested ML algorithms (including multi-layer neural networks) - at the cost of some additional computational complexity. The regularization parameters in this case are the ϵ and C as defined above for SVR.

Decision Tree Algorithms Decision trees are composed of a series of conditional statements that branch in a “tree” like manner. We say the “depth” of a decision tree is how many conditional statements appear in a branch before leading to a label denoted the “leaf”. Both the depth of the decision trees and the maximum number of leaves are regularization parameters that can be modified by the user. Decision trees are often weak predictors alone and in this paper we use ensemble (random forest) or boosting (boosted trees) methods to increase predictive performance. These algorithms are defined as follows.

- **Random Forest:** The random forest algorithm is an ensemble machine learning method based on a combination of decision trees. Ensemble methods use a combination of predictive models (trees) that individually have poor generalization but when used in combination can have significantly improved predictions. The number of decision trees combined in the random forest algorithm can be used as a regularization parameter.
- **Boosted Trees:** Gradient boosting is another machine learning method also based on a combination of decision trees. In the boosted algorithm trees are added to the predictive model sequentially, and each additional tree is fit to the current residuals of the model. A “learning rate” is a weight applied to the addition of each decision tree, and is often used as a regularization parameter. Small learning rates tend to improve the generalization of the predictive models.

Next we will focus on a metric we may use to identify the observables which are most suitable to the task of predicting self vs nonself determination in autoimmune disease.

4 Quantifying Suitability of a Given Set of Observables

In the previous section, we provided a procedure for using a subset of the features to predict both the remaining features and the label. To identify a set of observables for predicting self vs nonself determination we rigorously define a metric for suitability in order to select the observables which lead to superior predictive models.

First, for the sake of generality, we define the algorithm, OPT, which we use as a placeholder for the machine learning algorithms described in the previous section.

Definition of OPT : Given a dataset $\{x_i, y_i\}_{i=1}^m \subset \mathbb{R}^w \times \mathbb{R}^q$, $OPT(\{x_i, y_i\}_{i=1}^m)$, returns a predictive function, $f = \text{arg}OPT(\{x_i, y_i\}_{i=1}^m)$, where $f : \mathbb{R}^w \to \mathbb{R}^q$.

Next, given a possible set of feature indices $F := \{1, \cdots, n\}$, we define the set of partitions of F as $\mathcal{P}(F)$, and the set of all possible partitions of F of length $w \leq n$ as follows.

$$B_w := \{v \in \mathbb{N}^w \mid v \in \mathcal{P}(F)\}$$

For a given selection of features, $b \in B_w$, we denote the associated projection $P_b : \mathbb{R}^n \to \mathbb{R}^w$ so that $(P_b(x))_i = x_{b_i}$ for $x \in \mathbb{R}^n$ and $i = 1, \cdots, w$.

As discussed previously, our goal in this section is to define a metric of suitability for a given selection of features, $b \in B_w$. To this end, we consider three cost/penalty functions, M_1, M_2, L. The function L is simply a function of the cardinality of the number of features selected, $L(|b|_C)$. The costs M_1 and M_2, however, measure how well the selection of features can be used to predict the remaining features. However, for a given set of data, these metrics will vary depending on which data points are used for training OPT and which are used to evaluate its performance. To explicitly account for the effect of choice in partitioning of data samples, we now define the set of samples $S := \{1, \cdots, m\}$, and the set of partitions of S as $P(S)$. As for features, we denote the set of sample partitions of length r as

$$S_r := \{v \in \mathbb{N}^r \mid v \in \mathcal{P}(S)\}$$

and for a given selection of samples, $g \in S_r$, we denote the associated projected data set as $P_g(X) := \{x_i \in X,\ i \in g\}$.

Therefore, the costs M_1 and M_2 are a function of the feature partition, b, the training partition, $g \in S_r \in P(S)$ and the associated test partition, $h := S/g \in S_{m-r}$, so that we have $M_1(b, g)$ and $M_2(b, g)$. Specifically, we have

$$\begin{aligned} M_1(b, g) &= \sum_{i \in S/g} |f_{b,g}(P_b(x_i)) - y_i| \\ M_2(b, g) &= \sum_{i \in S/g} \left| d_{b,g}(P_b(x_i)) - P_{F/b}(x_i) \right| \\ f_{b,g} &= \arg OPT(\{P_b(x_{g_i}), y_{g_i}\}_{i=1}^r) \\ d_{b,g} &= \arg OPT(\{P_b(x_{g_i}), P_{F/b}(x_{g_i})\}_{i=1}^r)) \end{aligned}$$

In the ideal case, we would average these costs over all possible partitions of the data set to give an estimate of the predictive power of $b \in B_w$. However, such an approach would result in very large computational overhead. Therefore, we use the k-fold cross validation approach, wherein we divide the samples into k training partitions of size $\frac{m(k-1)}{k}$, which we label as $g(i) \in S_{\frac{m(k-1)}{k}}$ for $i = 1, \cdots, k$. Then the average cost of the feature partition b over the k sample partitions is

$$J(b) = \frac{1}{k} \sum_{i=1}^{k} J'(b, g(i), S/g(i)).$$

where

$$J'(b, g) := \beta_1 \sqrt{M_1(b, g)} + \beta_2 \sqrt{M_2(b, g)} + L(|b|_C) \tag{1}$$

and where $\beta_1, \beta_2 \geq 0$ are given weights, the values of which are discussed in Section 6.

In the following section, we now define the feature selection problems as minimization of this metric and present algorithmic approaches to solving this problem.

5 Feature Selection Algorithms

In the previous section, we defined the metric of suitability as a function of the partition, $b \in B_w$. Using this metric, the feature selection problem is defined as the following combinatoric optimization problem.

$$\min_{b \in B_w, w \in \mathbb{N}} \quad J(b) \tag{2}$$

Optimization problems of this form are a special case of feature selection (typically solved using wrapper methods) and, being combinatorial optimization, Problem (2) is NP-hard [6]. As a consequence, most Feature Selection (FS) algorithms as applied to this problem are either heuristic, in that they are not guaranteed to converge to a globally optimal solution, or solve unrelated problems which may or may not yield reasonable values for Problem (2).

Nonetheless, several techniques have been proposed that enjoy relative accuracy and computational efficiency. We focus first in Subsection 5.1 on FS methods designed specifically for problems of the same form as Optimization Problem (2), then in Subsection 5.2 consider two other FS approaches that do not directly try to solve the optimization problem of interest but provide a comparison to the direct method.

5.1 Proposed Wrapper Method and Implementations

We first define the algorithm (a wrapper method) which will be used and then provide additional details on the various ML algorithms which are combined with this wrapper to solve Problem (2).

The most common wrapper methods are Sequential Feature Selection (SFS) algorithms [6]. SFS algorithms begin with an empty (or full) set of features and sequentially add (or remove) the highest value (or cost) feature until the set of features is a certain size or meets a performance metric.

The SFS algorithm used in this paper is as described in [9]. This SFS algorithm begins with $b := \emptyset$, and iteratively selects a locally optimal feature (with respect to the objective function of Optimization Problem (2)) at each step.

Clearly, the effectiveness of Feature Selection depends on the ML algorithm (OPT) used to generate the predictive model. Therefore, in the numerical results generated in Section 6, we test all the machine learning algorithms proposed in Section 3. Unfortunately, the accuracy of the reliability and accuracy of the predictive model is influenced by user-selected parameters within the algorithm. For reproducibility, we list here the selections for these parameter values.

Linear Regression: We test all 16 combinations of $\alpha_1 \in [0, 0.1, 1, 5]$ and $\alpha_2 \in [0, 0.1, 1, 5]$ and the data from choice yielding highest suitability (J) is listed in Table 1.

TKL: We use the default TK kernel parameters and test $\epsilon = .1$, and $C \in [1, 5, 10]$ and the data from choice yielding highest suitability (J) is listed in

Table 1.
SVR: We test all combinations of $\epsilon = .1$, $C \in [1, 5, 10]$ and 3 kernel functions (linear, RBF, or 3rd degree polynomial) and the data from choice yielding highest suitability (J) is listed in Table 1. For the RBF kernel the features are normalized by their variance and a bandwidth of $\frac{1}{n}$ is selected.
Random Forest We test 9 combinations of number of trees ($\mathrm{n_{trees}} \in [50, 100, 150]$) and the maximum tree depth of ($\mathrm{max_{depth}} \in [5, 10, 20]$) and the data from choice yielding highest suitability (J) is listed in Table 1.
Boosted Trees We test 15 combinations of number of trees ($\mathrm{n_{trees}} \in [50, 100, 150, 250]$) and learning rate ($\mathrm{LR} \in [0.01, 0.1, 0.5]$) and the data from choice yielding highest suitability (J) is listed in Table 1.

5.2 Suitability of Filter and Embedded Methods

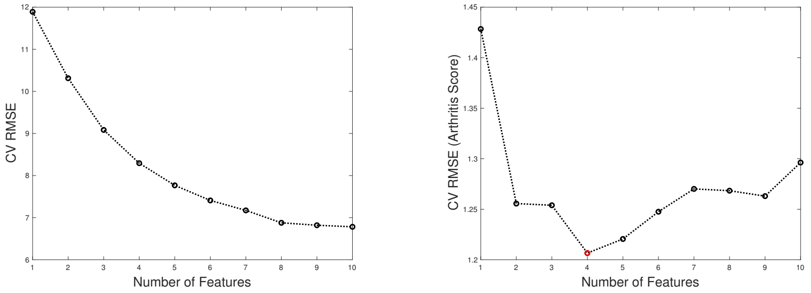
Alternative feature selection algorithms will be used as a baseline by which we may compare the wrapper method. We use three filter methods and one embedded method in the analysis.

Filter Methods Filter methods, given a set of data, use a rating function to rank each features relative "importance". After the features have been ranked, the user may select w features to be kept and the remaining features will be discarded. The rating functions used to generate the data in Table 1 are as follows.
Mutual Information (MI) The Mutual Information criteria [2] is a statistical function of two random variables that describes the amount of information contained in one random variable relative to the other.
Analysis of Variance (ANOVA) The ANOVA method [16] is a commonly used method for analyzing variable dependencies. The *F-test* is used to estimate the features importance.
Principle component analysis (PCA) This method approximates the data with linear manifolds [21]. The main methods used to perform PCA are based on the singular value decomposition and diagonalization of the correlation matrix. We calculate the importance based on the first 3 eigenvectors.

In all cases, once a set of features has been selected, suitability (J) is determined using each of the ML algorithms defined in Section 3 and the minimum of these values is listed in Table 1.

Embedded Methods Embedded FS methods attempt to embed the process of feature selection directly into the model generation process - typically adding a cost for inclusion of a particular feature in the model. These methods have been used in the gene expression domains as in [14] and have been successfully applied to mass spectrometry analysis in [22, 15, 18]. For this analysis, only a single embedded method was considered.
Mean Decrease in Impurity (RF) The Gini Importance or Mean Decrease in Impurity [4] is an embedded method for the Random forest algorithm. It

calculates the importance of features as the mean of the number of splits (over all trees) that include this feature, weighted by the probability of reaching this node.



(a) The objective function of Optimization Problem (2) as a function of the number of features for the MIS of the mouse using our TKL SFS method.

(b) The objective function of Optimization Problem (2) as a function of the number of features for the MDS of the mouse using our TKL SFS method.

Figure 2: The objective function of Optimization Problem (2) as a function of the number of features for the MIS and MDS of the mouse RA dataset using the TKL SFS method.

6 Feature selection analysis of rheumatoid arthritis data

We now apply the feature selection algorithms proposed in Section 5 to data generated from the mouse model of RA described in Section 2. We consider three variations of the feature selection problem as posed in (2). First, we let $\beta_1 = 1$ and $\beta_2 = \beta_3 = L(w) = 0$ - a case we denote as the Minimal Disease State (MDS). In this case, we are only concerned with predicting the progression of the disease and are not concerned with predicting non-selected features or with the number of features selected. Second, we let $\beta_1 = 0$ and $\beta_2 = \beta_3 = 1$ and $L(w) = \begin{cases} 0 & \text{for } w \leq 10 \\ \infty & \text{for } w > 10. \end{cases}$ In this case, ignore the disease state and are only concerned with reducing the number of features while retaining the ability to reconstruct discarded features - a case we denote as the Minimal Immune State (MIS). Finally, we let $\beta_1 = \beta_2 = \beta_3 = 1$ and $L(w)$ as defined for the MIS. We denote this final case as Minimal Overall State (MOS).

In Table 1, we see the objective value of Optimization Problem (2) (J) for each of the proposed feature selection algorithms as applied to MDS, MIS, and MOS. Preprocessing and regression were performed using Python 3.7 with scikit-learn 0.22.1 and MATLAB R2020a with TKL v1. To show that the results of Optimization Problem (2) as applied to MDS, MIS and MOS are consistent with other learning metrics [3], we also include data on these metrics for the chosen selection of features and associated predictor. These metrics are defined as follows. For given Let y to be the vector of labels associated with features x. Let $\hat{y}$ be the predicted labels as generated by the predictor when applied to features x (discarded features for MIS and disease state for MDS). Let $\bar{y}$ and $\bar{\hat{y}}$ be the average values of y and $\hat{y}$. Then we have the following.

The correlation coefficient (CC):

$$\mathrm{CC} = \frac{\sum_{i=1}^{N}(y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2 \sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}}_i)^2}}$$

Mean Absolute Error and relative Mean Absolute Error (MAE and rMAE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|; \quad \mathrm{rMAE} = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\sum_{i=1}^{N}|y_i - \bar{y}_i|}$$

Root Mean Squared Error and relative Root Mean Squared Error (RMSE and rRMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}; \quad \mathrm{rRMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}}$$

To illustrate convergence of the FS algorithm, in Fig. 2 we see the objective value of the TKL FS algorithm as applied t Optimization Problem (2) as each feature is sequentially added to the list of selected features.

Table 1: Results for finding MIS and MDS using the RA data in Section 2 using each of the algorithms described in Section 5. The bolded values are the best metrics for each column.

		MIS						MDS	
model	J	MAE	rRMSE	rMAE	cc	model	J	MAE	rRMSE
Random Forest	**6.47**	**4.46**	**0.43**	**0.32**	**0.85**	TKL	**1.22**	**0.96**	**0.79**
Boosted Trees	6.66	4.47	**0.44**	0.36	0.83	Linear Regression	1.26	1.07	0.90
TKL	6.74	4.70	**0.44**	0.42	0.84	Boosted Trees	1.34	1.19	0.99
Linear Regression	6.89	5.21	0.49	0.36	0.82	MI	1.34	1.22	1.01
SVR	7.37	5.37	0.47	0.35	0.82	SVR	1.38	1.09	0.88
PCA	12.84	9.80	0.69	0.66	0.69	ANOVA	1.38	1.23	1.03
RF	13.91	10.26	0.79	0.84	0.58	Random Forest	1.39	1.20	0.98
ANOVA	14.25	10.95	0.76	0.78	0.63	RF	1.39	1.27	1.07
MI	14.52	10.90	0.73	0.71	0.66	PCA	1.40	1.33	1.10

In the following analysis, we first list the features selected by the algorithm which returned the minimal value of the objective (J). We then discuss the biological significance of these features. Next, we compare with features selected by other algorithms to find features selected by all or a majority of the algorithms tested.

6.1 Case 1: Features for Predicting Disease Progression (MDS)

High predictive accuracy is important for tracking the disease progression and predicting the effectiveness of treatments based on measurements of the observables - which is important for autoimmune diseases where the disease state may be difficult to measure.

Most Important Features Using Best SFS Algorithm For MDS, the SFS TKL algorithm performed best. The corresponding 4 features were

(1) *CD8+Ki67+*

(2) *CD4+Foxp3+CD25+Ki67+*

(3) *CD4+GATA3+Ki67+*

(4) *CD4+RORyT+Autoantigen*

This group of cells consists of markers for cytotoxic (1), regulatory (2), helper (3), and helper (4) subpopulations, respectively, where the second helper population (4) is specific for the RA autoantigen. We conclude that the selected features correspond to what would be expected in a measure of relevant T cell sub-populations.

Agreement with other algorithms In Fig. 3 we show the observables that were selected by each of the proposed algorithms. The three features selected most often by the FS methods are

- *CD4+Foxp3+CD25+* (regulatory)
- *CD4+GATA3+Ki67+* (helper)
- *CD8+Ki67+* (cytotoxic)

As indicated above, two of these features, (1) and (3), were also selected by the TKL algorithm. Note that as expected filter methods did not perform as well as the wrapper or embedded methods.

Overall predictive accuracy when using selected features

6.2 Case 2: Features for Predicting Remaining Features (MIS)

This case studied the features most important for determining the overall state of the immune system (and not the progressive state of any particular disease).

In Table 1, we list the performance of the various algorithms and in Fig. 3(a) we indicate the T cell markers selected by each algorithm.

The proposed FS algorithms can be clearly divided into two groups by the achieved objective value. Specifically, Five of the methods performed poorly - achieving minimal objective values greater than 7 (ANOVA,PCA,SVR,MI,RF). The filter methods and embedded method performed particularly poorly. SVR, which uses a standard RBF kernel, performed the worst of the SFS based methods. Furthermore, the selected features for these poorly performing methods were inconsistent. For these reasons, we will discount results from the ANOVA, PCA, SVR, and MI and limit our analysis to features selected by Random Forest, Boosted Trees, TKL, and Linear Regression.

Most Important Features Using Best SFS Algorithms Unlike in the previous subsection, there was broad agreement among all 4 high-performing algorithms as to the most significant features for optimizing MIS. First, if we consider markers specific to helper and regulatory T cells, and counting the number of times a feature was selected by these four methods (each method selected 10 features), the following features were each chosen by at least 3 algorithms.

(1) *CD4+Foxp3+CD25+Ki67+Autoantigen* (3 times)

(2) *CD4+GATA3+Ki67+Autoantigen* (3 times)

(3) *CD4+Roryt+CD25+Autoantigen* (3 times)

(4) *CD4+Tbet+Autoantigen* (3 times)

We note that every selected features was autoantigen specific - indicating this additional information is particularly useful for creating predictive models.

Among the cytotoxic T cells, the algorithms were remarkably consistent.

(5) *CD8+GATA3+CD44+CD62L(LO)* (4 times)

(6) *CD8+Tbet+CD44+CD62L(LO)* (4 times)

(7) *CD8+Ki67+* (3 times)

(8) *CD8+Tbet+Ki67+* (3 times)

Overall, we note that the memory T cells (CD62) seem to be particularly significant. In addition, the algorithms tend to choose features which have been sorted by the most markers - indicating that perhaps this filtering provides additional useful information to the algorithm. Supporting this hypothesis, we also note that in our analysis of the results, that if certain data-rich biomarkers are left out, such as antigen-specific *CD8+GATA3+Ki67+*, antigen-specific *CD4+Foxp3+CD25+Autoantigen* and antigen-specific *CD4+GATA3+CD44+CD62(Lo)+*, then those features are poorly predicted using all methods. For these cell populations the prediction error is approximately 12, significantly exceeding the average error for predicting other features. If these T cells in particular are required with high accuracy it is best that they be measured directly. By contrast, the biomarkers that most easily predicted are *CD4+Foxp3+CD25+*, *CD4+GATA3+*, *CD4+Tbet+CD44+CD62(Lo)*, and *CD8+Tbet+*. In these cases the prediction error values are 5 or

Finally, the fact that the *location* feature (origin of the tested cells) was not chosen by any of the top 4 methods implies there is significant uniformity in immune state among lymph nodes and spleen.

less.

6.3 Case 3: Features for Minimizing Weighted Objective (MOS)

Next we consider the problem of selecting features that are optimal for predicting a combination of the MIS and MDS objectives.

In Table 1, we list the performance of the various algorithms and in Fig. 3(c) we indicate the T cell markers selected by each algorithm.

The proposed FS algorithms can be clearly divided into two groups by the achieved objective value. As before, the SFS methods significantly outperformed the filter and embedded methods, and hence we again discount the ANOVA, PCA, RF, and MI results and limit our analysis to the features selected by the top four SFS methods - the Boosted Trees, TKL, SVR, and Linear Regression methods.

Most Important Features Using Best SFS Algorithms Like the previous subsection, there is broad agreement among all high-performing algorithms as to the most significant features for optimizing MOS.

A few T cell populations were selected quite often by the different methods. All four methods selected the following antigen-specific population

(1) *CD4+Tbet+Autoantigen*

In addition, three of the four methods all chose the non-antigen specific cells,

(2) *CD8+GATA3+CD25*

(3) *CD4+Tbet+Ki67+*

7 Conclusion

In this paper, we have considered the problem of identification of three different subsets of Tcells related to the overall response of the process of self-nonself determination as well as the effectiveness of a recently developed approach to immunotherapy for RA. Specifically, we have used a set of mouse-model experiments to obtain a robust dataset of T cell markers and populations at the end stage of a proposed immunotherapy treatment. We then used feature selection algorithms to determine the minimal number of markers and populations needed to effectively predict both the rest of the dataset and the current state of the disease. Our results show that while a minimal number of T cell markers may be used to predict the remaining T cell subsets with relatively low error, prediction of immunotherapy outcome is less reliable, implying that a full measure of the immune state would require additional data beyond the T cell populations collected in this analysis.

Funding

This work was supported by the National Science Foundation under grant No. 1931270, No. 1935453 and No. 026257-001.

References

[1] A. Acharya, M. Sinha, M. Ratay, X. Ding, S. Balmert, C. Workman, Y. Wang, D. Vignali, and S. Little. Localized multi-component delivery platform generates local and systemic anti-tumor immunity. *Advanced Functional Materials*, 2017.

[2] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE transactions on neural networks*, 1994.

[3] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga. A survey on multioutput regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5, 2015.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. 1984.

[5] P. Brodin and M. Davis. Human immune system variation. *Nature reviews immunology*, 2017.

[6] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40, 2014.

[7] B. Colbert and M. Peet. A convex parametrization of a new class of universal kernel functions. *Journal of Machine Learning Research*, 21(45), 2020.

[8] B. Colbert and M. Peet. A new algorithm for tessellated kernel learning, 2020.

[9] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1997.

[10] M. Davis. A prescription for human immunology. *Immunity*, 2008.

[11] J. Fisher, S. Balmert, W. Zhang, R. Schweizer, J. Schnider, C. Komatsu, L. Dong, V. Erbas, J. Unadkat, A. Aral, et al. Treg-inducing microparticles promote donor-specific tolerance in experimental vascularized composite allotransplantation. *Proceedings of the National Academy of Sciences*, 2019.

[12] J. Fisher, W. Zhang, S. Balmert, A. Aral, A. Acharya, Y. Kulahci, J. Li, H. Turnquist, A. Thomson, M. Solari, et al. In situ recruitment of regulatory T cells promotes donor-specific tolerance in vascularized composite allotransplantation. *Science Advances*, 2020.

[13] R. Germain and P. Schwartzberg. The human condition: an immunological perspective. *Nature immunology*, 2011.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 2004.

[15] K. Jong, E. Marchiori, M. Sebag, and A. Vaart. Feature selection in proteomic pattern data with support vector machines. *2004 Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2004.

[16] M. Kumar, N. Kumar Rath, A. Swain, and S.K. Rath. Feature selection and classification of microarray data using MapReduce based ANOVA and K-nearest neighbor. *Procedia Computer Science*, 54, 2015.

[17] J. Mangal, S. Inamdar, Y. Yang, S. Dutta, M. Wankhede, X. Shi, H. Gu, M. Green, K. Rege, M. Curtis, et al. Metabolite releasing polymers control dendritic cell function by modulating their energy metabolism. *Journal of Materials Chemistry B*, 2020.

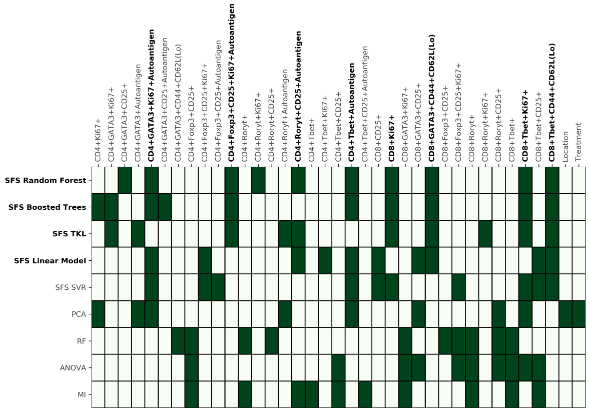
[18] J. Prados, A. Kalousis, J. Sanchez, L. Allard, O. Carrette, and M. Hilario. Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents. *Proteomics*, 4, 2004.

[19] M. Ratay, S. Balmert, A. Acharya, A. Greene, T. Meyyappan, and S. Little. TRI microspheres prevent key signs of dry eye disease in a murine, inflammatory model. *Scientific reports*, 2017.

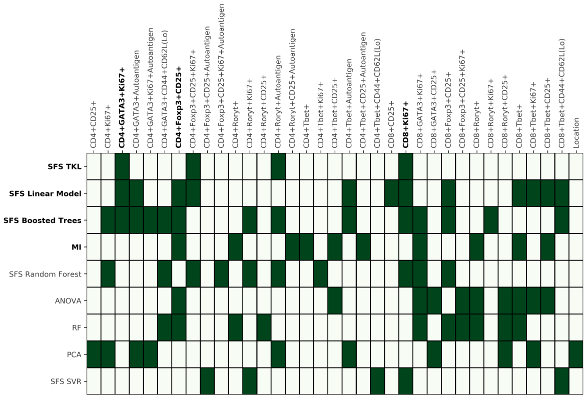
[20] M. Ratay, A. Glowacki, S. Balmert, A. Acharya, J. Polat, L. Andrews, M. Fedorchak, J. Schuman, D. Vignali, and S. Little. Treg-recruiting microspheres prevent inflammation in a murine model of dry eye disease. *Journal of Controlled Release*, 2017.

[21] F. Song, Z. Guo, and D. Mei. Feature selection using principal component analysis. *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, 1, 2010.

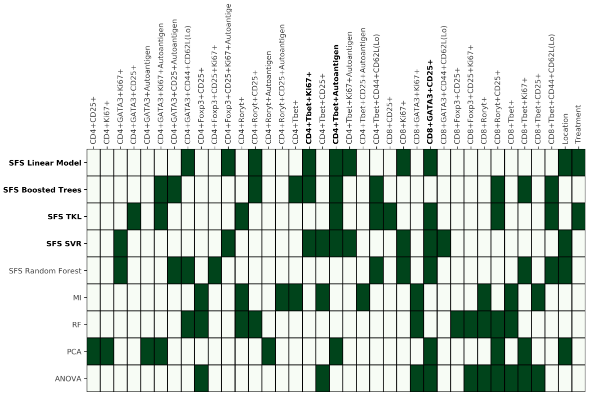
[22] X. Zhang, X. Lu, Q. Shi, X. Xu, H. Leung, L. Harris, J. Iglehart, A. Miron, J. Liu, and W. Wong. Recursive svm feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, 7, 2006.



(a) The features selected by various methods for predicting the Minimal Immune State (MIS).



(b) The features selected by various methods for predicting the Minimal Disease State (MDS).



(c) The features selected by various methods for predicting the Minimal Overall State (MOS).

Figure 3: The green squares indicate that the feature selection method (left) selected the feature (top). We show the observables selected by the nine different FS algorithms from Section 5 that compose the MIS (a) and MDS (b). The methods are ordered from highest objective function (RMSE) at the top to lowest objective at the bottom. The best four methods and the most commonly selected features by those methods are bolded.