TCRGP: Determining epitope specificity of T cell receptors

Emmi Jokinen [1,*], Markus Heinonen [1,2], Jani Huuhtanen [3,4], Satu Mustjoki [3,4] and Harri Lähdesmäki [1,*]

[1]Department of Computer Science, Aalto University, Espoo, 02150, Finland. [2]Helsinki Institute for Information Technology, Espoo, 02150, Finland. [3]Translational Immunology Research program and Department of Clinical Chemistry and Hematology, University of Helsinki, Helsinki, 00014, Finland. [4]Hematology Research Unit Helsinki, Helsinki University Hospital Comprehensive Cancer Center, Helsinki, 00014, Finland.

[*] To whom correspondence should be addressed.

Abstract

T cell receptors (TCRs) can recognize various pathogens and consequently start immune responses. TCRs can be sequenced from individuals and methods that can analyze the specificity of the TCRs can help us better understand the individual's immune status in different diseases. We have developed TCRGP, a novel Gaussian process (GP) method that can predict if TCRs recognize certain epitopes. This method can utilize different CDR sequences from both TCRα and TCRβ chains from single-cell data and learn which CDRs are important in recognizing the different epitopes. We have experimented with one previously presented and one new data set and show that TCRGP outperforms other state-of-the-art methods in predicting the epitope specificity of TCRs on both data sets. The software implementation and data sets are available at `https://github.com/emmijokinen/TCRGP`.

Contact: emmi.jokinen@aalto.fi, harri.lahdesmaki@aalto.fi

1 Introduction

The adaptive immune system implements various complex mechanisms for surveillance against both pathogens and pathological cells arising in our body. To initiate an adequate adaptive immune response, a pathogenic or aberrant peptide, called epitope (or antigen), must first be bound by the major histocompatibility complex (MHC) class I or II molecule expressed on surface of nucleated cells or professional antigen-precenting cells. The peptide-MHC (pMHC) complex is then presented to T cells which can recognize the complex via the T cell receptor (TCR) protein, consequently leading to T cell activation and proliferation via various downstream pathways (Davis and Bjorkman, 1989). Depending of co-stimulatory cues that may be present simultaneously with the TCR-pMHC binding, T cell can then react directly against the target by either cell-mediated killing of the pathological cell or by helping other immune cells to eliminate the target (Davis *et al.*, 2003).

T cells undergo non-homologous recombination during T cell maturation, which involves combining the variable regions of TCR loci from a large collection of variable (V), diversity (D) and joining (J) gene segments as well as random nucleotide insertions and/or deletions at the junctions (Bassing *et al.*, 2002; Cabaniols *et al.*, 2001). It is estimated that somatic recombination can result in the range of 10^{18} different TCR genes (Arstila *et al.*, 1999; Robins *et al.*, 2009). This enormous diversity enables T cells to recognize a wide range of pathogen-specific epitopes. Furthermore, it is estimated that a single TCR can interact with at least 1 million different pMHCs (cross-reactivity), and a given pMHC can elicit response to millions of TCRs (Wooldridge *et al.*, 2012; Sewell, 2012). Given these three levels of diversity, predicting TCR's epitope specificity is notably challenging (Miho *et al.*, 2018).

It is well known that for example the CDR3β sequence of a TCR is important in recognizing peptides presented to the T cell, but it is still unclear which specific biophysical or structural features of the CDR3 or of other parts of the TCR determine the antigen recognition specificity of that T cell. Although high-throughput DNA sequencing has enabled large-scale characterization of TCR sequences (Robins *et al.*, 2009), it still remains exhaustive to profile epitope-specific TCRs as they require sample-consuming experiments with distinct pMHC-multimers for each epitope of interest. Therefore, there is a great need for models that examine which epitopes a TCR can recognize, or what TCRs an epitope can bind. Curated databases of experimentally verified TCR-peptide interactions have recently been launched, such as the VDJdb and McPAS (Shugay *et al.*, 2017; Tickotsky *et al.*, 2017). Such data sources are enabling more comprehensive, data-driven analysis of TCR-peptide interactions, and make it possible to use techniques from statistical machine learning for the aforementioned tasks.

Only a few computational TCR specificity models have been proposed in the literature. Dash *et al.* (2017) devel-

oped a method called TCRdist, which uses a BLOSUM62 based distance measure between amino acids. They define the distance between two (aligned) TCRs as a weighted sum over the amino acid distances. Using this distance they then determine if the TCR is closer to some epitope-specific cluster of TCRs or to a cluster of some background TCRs. De Neuter *et al.* (2018) use a different strategy in that they first extract biophysical and other features from TCR amino acid sequences and then train a random forest classifier using these feature vectors to predict TCR-epitope recognition. We propose a method called `TCRGP` which builds on non-parametric modelling using Gaussian process (GP) classification. GPs are Bayesian non-parametric models that have become popular in many applications. Probabilistic formulation of GPs allows robust model inference also from small data sets, as is currently the case for TCR-peptide interaction information in curated databases. As the space of all TCRs that can recognize a certain epitope is potentially very large, it is important to avoid overfitting to the limited sample of TCRs that is available. Indeed, TCRGP clearly outperforms the current state-of-the art methods for predicting the epitope specificity of TCRs. At the same time, `TCRGP` also scales to extremely large data sets which we expect for the future epitope specific TCR-seq data sets.

2 Methods

In this section, we will describe the data we use (Section 2.1), how we divide it for training and testing of the models (Section 2.2), and how we represent the TCR sequences (Section 2.3). We will also describe GPs and how we utilize them (Sections 2.4 and 2.5) and how we use multiple kernel learning to exploit the different CDRs in our model (Section 2.6).

2.1 Data

Most of the TCRs (90-95%) are formed by an α and β-chain, but there are also TCRs which consist of γ and δ-chains (Allison *et al.*, 2001). Consequently there is also much more data available of α/β TCRs, and for that reason we only experiment with these TCRs.

The complementarity determining regions (CDR1, CDR2, and CDR3) of a TCR determine whether the TCR recognizes and binds to an antigen or not (Lefranc and Lefranc, 2001). Of these regions, the CDR3 is most variable and primarily interacts with the peptide, while the other regions primarily interact with the peptide binding groove of the MHC protein presenting the peptide (Rudolph *et al.*, 2006). Figure 1 shows these CDRs, in interaction with a peptide-MHC-complex (pMHC).

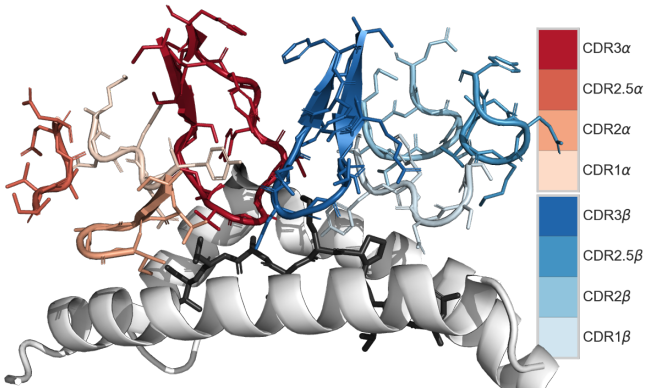

Figure 1: CDRs of a TCR binding to epitope pp65, NLVPMVATV (shown in black), presented by MHC-protein, whose binding groove is shown as a white cartoon. The original structure is from (Gras *et al.*, 2009).

Epitope gene	CDR3α	CDR1α	CDR2α	CDR2.5α	CDR3β	CDR1β	CDR2β	CDR2.5β
BMLF	CAASDGAGGTSYGKLTF	NSM------FDY	ISSI---KDK	NKSAKH	CASSLWT---GSHEQYF	SGH-------TS	YDE----GEE	F-PNYS
BMLF	CAESL-------DMLTF	DSS------STY	IFSN---MDM	NKKDKH	CASSVVG----GNEQFF	SGD-------LS	YYN----GEE	F-PDLH
BMLF	CAMREVMD--SNYQLIW	TSDP-----SYG	QGSY--DQQN	QKARKS	CASSVAQLAGGTDTQYF	SGD-------LS	YYN----GEE	F-PDLH
pp65	CAGQAS----QGNLIF	SIF-------NT	LYKA---GEL	GITRKD	CASSIQ--------ALLTF	SGH-------DY	FNN----NVP	P-NASF
pp65	CAVRDNSITGGFKTIF	TSG------FYG	NAL----DGL	SRSDSY	CASSYF-------DEKLFF	DFQ------ATT	SNEG---CKA	A-SLTL
pp65	CILSNN-----NDMRF	TISG-----TDY	GLT-----SN	AEDRKS	CSARDPSGLAGGLAETQYF	DFQ------ATT	SNEG---SKA	A-SLTL

Table 1: An example of aligned TCR sequences (from the data set of Dash *et al.*, 2017) for two peptides. CDRs are aligned according to IMGT definitions. Alignments of CDR1, CDR2 and CDR2.5 sequences for both the α- and β-chains are shared between epitopes, whereas alignments for CDR3 sequences are specific for each epitope and chain.

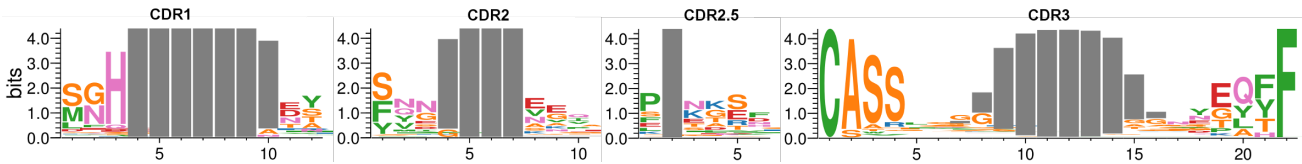

Figure 2: Sequence logos of the different CDRβs of TCRs recognizing the epitope NLVPMVATV (pp65). TCR sequence data from VDJdb. In each panel, x-axis corresponds to the amino acid position in the CDR-specific alignment.

Glanville *et al.* (2017) examined 52 crystal structure of TCR-pMHC interactions and concluded that the CDR3β is always in contact with the peptide, whereas CDR3α is in contact with the peptide on most times, but not always. In their examinations also CDR1 and CDR2 of both α and β-chains sometimes made contact with the peptide. Dash *et al.* (2017) noted that also a loop between CDR2 and CDR3 (IMGT positions 81-86) has been observed to make contacts with pMHC in solved structures, and they also utilized this region, which they called CDR2.5. CDR1, CDR2 and CDR2.5 can all be directly determined from the V-gene and -allele of the TCR. Dash *et al.* (2017) provide a table of all V-gene and allele combinations and the corresponding CDR1, CDR2, and CDR2.5 amino acid sequences aligned according to IMGT® definitions (Lefranc, 1999). Our method can utilize the aligned amino acid sequences of all these CDRs from either one or both of the α- and β-chains of the TCR. Table 1 shows a few examples of TCR sequences and their alignment. Figure 2 illustrates the diversity of TCRs recognizing epitope NLVPMVATV (pp65), demonstrating the challenge in predicting the epitope specificity of TCRs.

In our experiments, we use a data set collected by Dash *et al.* (2017), which contains epitope-specific paired TCRα and TCRβ chains for three epitopes from humans and for seven epitopes from mice, see Table 2 for details.

Species	Epitope species	Epitope gene	Epitope	Subjects	Samples
Human	EBV	BMLF	GLCTLVAML	6	76
	hCMV	pp65	NLVPMVATV	10	61
	IAV	M1	GILGFVFTL	15	275
Mouse	IAV	F2	LSLRNPILV	9	117
	IAV	NP	ASNENMETM	24	305
	IAV	PA	SSLENFRAYV	15	324
	IAV	PB1	SSYRRPVGI	34	642
	mCMV	m139	TVYGFCLL	8	87
	mCMV	M38	SSPPMRFV	14	158
	mCMV	M45	HGIRNASFI	13	291

Table 2: Data set from (Dash *et al.*, 2017) contains epitope-spcefic TCRs for Epstein-Barr virus (EBV), human Cytomegalovirus (hCMV), Influenza A virus (IAV) and mouse Cytomegalovirus (mCMV).

We also gather a new data set from VDJdb (`https://vdjdb.cdr3.net`), which is a database that contains TCR sequences with known antigen specificity (Shugay *et al.*, 2017). Every entry in VDJdb has been given a confidence score between 0 and 3 (0: critical information missing, 1: medium confidence, 2: high confidence, 3: very high confidence). We constructed our data set so that we selected all epitopes that have at least 50 TCRβ sequences with a confidence score at least 1 and found 22 such epitopes, see Table 3 for details. VDJdb also contains TCRα sequences, but since these are not in general paired with corresponding TCRβ sequences, we chose to only experiment with the TCRβ sequences.

For the training and testing of the models, we also required some background TCRs that we do not expect to recognize the epitopes in our data sets. For this purpose we used a set of background TCRs constructed in Dash *et al.* (2017).

The data sets we have used can be found from `github.com/emmijokinen/TCRGP`.

Epitope species	Epitope gene	Epitope	Subjects	Samples
CMV	p65	IPSINVHHY	17	65
CMV	p65	NLVPMVATV	103	413
CMV	p65	TPRVTGGGAM	29	184
EBV	BMLF1	GLCTLVAML	54	299
EBV	BZLF1	RAKFKQLL	17	225
EBV	BRLF1	YVLDHLIVV	6	66
Flu	M1	GILGFVFTL	50	239
Flu	HA	PKYVKQNTLKLAT	11	56
HCV	NS3	ATDALMTGY	7	152
HCV	NS3	CINGVCWTV	7	76
HCV	NS3	KLVALGINAV	4	65
HSV-2	VP22	RPRGEVRFL	5	68
YellowFeverVirus	NS4B	LLWNGPMAV	5	223
DENV1	NS3	GTSGSPIVNR	11	65
DENV3/4	NS3	GTSGSPIINR	8	51
HIV-1	p24	EIYKRWII	12	81
HIV-1	p24	FRDYVDRFYKTLRAEQASQE	17	141
HIV-1	p24	GPGHKARVL	1	62
HIV-1	p24	KAFSPEVIPMF	44	134
HIV-1	p24	KRWIILGLNK	27	212
HIV-1	p24	TPQDLNTML	21	52
HIV-1	Nef	FLKEKGGL	21	104

Table 3: Data set gathered from VDJdb contains epitope-specific TCRs for Cytomegalovirus (CMV), Epstein-Barr virus (EBV), Influenza A virus (Flu), Hepatitis C virus (HCV), Herpes Simplex virus type 2 (HSV-2), Yellow Fever virus, Dengue virus type 1 (DENV1), Dengue virus type 3 (DENV3/4), and Human immunodeficiency virus type 1 (HIV-1).

2.2 Leave-one-subject-out cross-validations

For the evaluation of the methods developed by us and others, we needed to divide our data sets for training and testing. Both of the data sets we use determine the subjects from whom each TCR in the data has been obtained from. We therefore chose to use leave-one-subject-out cross-validation, where we leave out all TCRs from one subject, train the model with all the other TCRs, test it with the TCRs left out, and repeat this for all subjects. This way the average number of TCRs per fold in the Dash data set varied between 6 (for NLVPMVATV) and 22 (for HGIRNASFI), and in the VDJdb dataset between 3 (for KAFSPEVIPMF) and 45 (for LLWNGPMAV). The number of subjects and samples for each epitope can be found from Tables 2 and 3.

We also randomly selected a set of background TCRs, so that there was always an equal number of epitope-specific and background TCRs in both training and test sets. We thought this would be the most realistic procedure for the evaluation, as this is likely how these kinds of models will be applied to new data: A model is trained with some set of TCRs and then predictions should be made for TCRs sequenced from an individual from who we have not seen any TCR sequences beforehand.

The data set we gathered from VDJdb contains TCR sequences from multiple studies, many of which have used same conventions for naming their subjects. Therefore

we used the combination of the PMID of the publication and the subject id as the subject identifier. For two epitopes, p24 (GPGHKARVL) and NS3 (KLVALGINAV) there were very few separate subjects, only one and four, respectively. With these epitopes we used 5-fold cross-validation instead of the leave-one-subject-out cross-validation.

2.3 Sequence representation

Computational methods require the data to have some presentation, that they can utilize. Character sequences with variable lengths often provide some challenges as many methods rely on numerical inputs of fixed sizes. One solution is to compare subsequences of same length instead of the complete sequences, which is what for example Generic String kernel (GSkernel) does (Giguere *et al.*, 2013).

However, by aligning the sequences more approaches become applicable. According to the IMGT definitions (Lefranc, 1999) CDR3s can be aligned by introducing a gap in the middle of the sequence (i.e. top of the loop). Alignments for CDR1s, CDR2s, and CDR2.5s can be found from `www.imgt.org`. When the sequences are aligned, all the sequences within a CDR class (1, 2, 2.5 or 3) have the same length (see Table 1).

We observe sequences $a_1 a_2 \cdots a_L$ of amino acids $a_j \in \mathcal{A} = \{A, R, N, \ldots, -\}$ at aligned positions $j = 1, \ldots, L$. The alignment guarantees that all sequences have the same possibly padded length L. We encode the amino acids a with global feature vectors $\boldsymbol{\phi}(a) \in \mathbb{R}^D$ that associate a D-length real-valued code with each of the 21 amino acids including the gap symbol. The sequences are then encoded as data vectors $\mathbf{x}$ by concatenating the L feature vectors into a $D \cdot L$ length column vectors $\mathbf{x} = (\boldsymbol{\phi}(a_1)^T, \ldots, \boldsymbol{\phi}(a_L)^T)^T \in \mathcal{X}$. We collect a dataset of N sequences into a matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times DL}$ with rows as sequences and columns as individual amino acid features in aligned order. Each sequence is associated with a class label $y_i \in \{0, 1\}$ that indicates whether the sequence was epitope-specific or not. We collect the class labels into an output vector $\mathbf{y} = (y_1, \ldots, y_N)^T \in \{0, 1\}^N$.

We can observe amino acid sequences for both the α- and β-chains and the four complementarity determining regions (CDR) $\{1, 2, 2.5, 3\}$ from a single TCR. Sequence data for each chain and CDR combination has individual alignments and sequence lengths. We denote the data as $(\mathbf{X}_{\alpha,1}, \mathbf{X}_{\alpha,2}, \mathbf{X}_{\alpha,2.5}, \mathbf{X}_{\alpha,3}, \mathbf{X}_{\beta,1}, \mathbf{X}_{\beta,2}, \mathbf{X}_{\beta,2.5}, \mathbf{X}_{\beta,3}, \mathbf{y})$.

Substitution matrices such as BLOSUM62 (Henikoff and Henikoff, 1992) describe the similarity of each amino acid. We modified the BLOSUM62 to include also the gap used in alignments and scaled the matrix values between 0 and 1. The resulting matrix $\mathbf{B} \in \mathbb{R}^{21 \times 21}$ is then positive semidefinite. We apply eigendecomposition $\mathbf{B} = \mathbf{V}\mathbf{S}\mathbf{V}^T$, where the column vectors of $\mathbf{V}$ encode orthogonal projections of the amino acids on the rows. We use the row vectors of $\mathbf{V}$, indexed by the amino acids a from the modified BLOSUM62, as our descriptions $\boldsymbol{\phi}(a) = \mathbf{V}_{a,:}$ with a feature representation $\boldsymbol{\phi}(a)^T \mathbf{S} \boldsymbol{\phi}(b) = [\mathbf{B}]_{ab}$ for any two amino acids $a, b \in \mathcal{A}$. It is possible to use also different substitution models and feature vectors obtained from different sources, or even use the so-called one-hot-encoding, but here we relied only on the eigenvectors of the (gap-extended) BLOSUM62. For our model, we utilized all the 21 components, but in Figure 3 we show how the amino acids locate on the first two components.

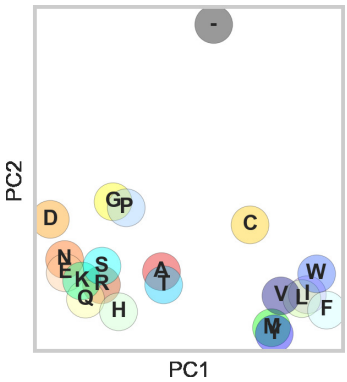

Figure 3: All 20 amino acids and the gap located on the first two orthogonal components of the modified BLOSUM62 substitution matrix. The amino acids are coloured according to Ruiz and Lefranc, 2000.

2.4 Gaussian process classification

We use Gaussian process (GP) classification (Rasmussen and Williams, 2006) to predict if a TCR recognizes a certain epitope or not. Gaussian processes model Gaussian distributions of non-parametric and non-linear functions. We apply a link function to squash the function values to a range $[0, 1]$ suitable for classification. GPs have a clear advantage of characterizing the prediction uncertainty with class probabilities instead of point predictions. GPs naturally model sequences through kernel functions focusing on sequence similarity as the explaining factor for class predictions.

We use a GP function f to predict the latent epitope-specificity *score* $f(\mathbf{x}) \in \mathbb{R}$ of a sequence $\mathbf{x}$. A zero-mean GP prior

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')),$$

defines a distribution over functions $f(\mathbf{x})$ whose mean and covariance are

$$\mathbb{E}[f(\mathbf{x})] = 0$$
$$\mathbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}'),$$

where $k(\cdot, \cdot)$ is the kernel function. We use the standard squared exponential kernel on the vectorized feature representation,

$$k(\mathbf{x}, \mathbf{x}'|\theta) = \sigma^2 \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')}{2\ell^2}\right), \qquad (1)$$

where ℓ is the length-scale parameter, σ^2 is the magnitude parameter and $\theta = (\ell, \sigma^2)$. For any collection of TCR sequences $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, the function values follow a multivariate normal distribution

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}}), \tag{2}$$

where $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^T \in \mathbb{R}^N$ collects all function predictions of the sequences, and $\mathbf{K_{XX}} \in \mathbb{R}^{N\times N}$ is the sequence similarity matrix with $[\mathbf{K_{XX}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The key property of Gaussian processes is that they couple all predictions to be dependent. The Gaussian process predicts similar epitope values $f(\mathbf{x}), f(\mathbf{x}')$ for sequences $\mathbf{x}, \mathbf{x}'$ if they are similar according to the kernel $k(\mathbf{x}, \mathbf{x}')$.

The latent function $f(\mathbf{x})$ represents an unbounded real-valued classification score, which we turn into a classification likelihood by the probit link function $\Phi : \mathbb{R} \mapsto [0, 1]$,

$$\Phi(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{f} \exp\left(-\frac{1}{2}\tau^2\right) d\tau. \tag{3}$$

The joint model then decomposes into a factorized Bernoulli likelihood and Gaussian prior,

$$p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}) \tag{4}$$

$$= \left[\prod_{i=1}^{N} \mathrm{Ber}(y_i|\Phi(f_i))\right] \cdot \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}}), \tag{5}$$

where f_i is a shorthand for $f(\mathbf{x}_i)$. The objective of Gaussian process modelling is to infer the posterior distribution $p(\mathbf{f}|\mathbf{y})$, which is intractable for many non-Gaussian likelihoods. Additionally inferring the kernel hyper-parameters θ entails computing the marginalized *evidence*

$$p(\mathbf{y}; \theta) = \mathbb{E}_{p(\mathbf{f};\theta)}[p(\mathbf{y}|\mathbf{f})], \tag{6}$$

which is also intractable in general and has a limiting cubic complexity $\mathcal{O}(N^3)$ (Rasmussen and Williams, 2006). We tackle the scalability with sparse Gaussian processes (Snelson and Ghahramani, 2006) and the intractability with stochastic variational inference (Hensman *et al.*, 2015).

2.5 Variational inference for low-rank GP approximation

We consider low-rank sparse Gaussian processes by augmenting the system with M inducing *landmark* pseudo-sequences $\mathbf{z}_j \in \mathcal{X}$ with associated (label) function values $u_j = f(\mathbf{z}_j) \in \mathbb{R}$. We collect all inducing points into structures $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)^T$ and $\mathbf{u} = (u_1, \ldots, u_M)^T$. By conditioning the GP with these values we obtain the augmented Gaussian process joint model

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u}) \tag{7}$$

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{A}\mathbf{u}, \mathbf{K_{XX}} - \mathbf{A}\mathbf{K_{ZZ}}\mathbf{A}^T) \tag{8}$$

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{ZZ}}) \tag{9}$$

$$\mathbf{A} = \mathbf{K_{XZ}}\mathbf{K_{ZZ}}^{-1}, \tag{10}$$

where $\mathbf{K_{XX}} \in \mathbb{R}^{N\times N}$ is the kernel between observed sequences, $\mathbf{K_{XZ}}$ is between observed and induced sequences and $\mathbf{K_{ZZ}}$ is between induced sequences. The matrix $\mathbf{A}$ projects the M inducing points to the full observation space of N sequences.

Next, we define a variational approximation for the inducing points,

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{S}) \tag{11}$$

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u})d\mathbf{u} \tag{12}$$

$$= \mathcal{N}(\mathbf{f}|\mathbf{A}\mathbf{m}, \mathbf{K_{XX}} + \mathbf{A}(\mathbf{S} - \mathbf{K_{ZZ}})\mathbf{A}^T), \tag{13}$$

where $\mathbf{m} \in \mathbb{R}^M$ and $\mathbf{S} \succeq \mathbf{0} \in \mathbb{R}^{M\times M}$ are free variational parameters to be optimized. It can be shown that minimizing the Kullback-Leibler divergence $\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u}|\mathbf{y})]$ between the approximative posterior $q(\mathbf{u})$ and the true low-rank posterior $p(\mathbf{u}|\mathbf{y})$ is equivalent to maximizing the evidence lower bound (ELBO) (Blei *et al.*, 2017)

$$p(\mathbf{y}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)}[\log p(y_i|f_i)] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]. \tag{14}$$

The log expectation is tractable for Probit likelihoods (Hegde *et al.*, 2018), while the KL term similarly has a closed form for two Gaussian densities.

Due to the small data regime we choose the optimal assignment of selecting $\mathbf{Z} = \mathbf{X}$ and $\mathbf{u} = \mathbf{y}$, which corresponds to the full Gaussian variational approximation of Nickisch and Rasmussen (2008), while for larger datasets the inducing landmark points can also be optimised (Hensman *et al.*, 2015). We then optimize the ELBO (14) with respect to the variational parameters $\mathbf{m}$ and $\mathbf{S}$ as well as the kernel hyperparameters θ, that is, the lengthscales ℓ_{cr} and weights w_{cr}.

Finally, predictions $\mathbf{f}_*$ of new test sequences $\mathbf{X}_* \subset \mathcal{X}$ follow a variational predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) = \int p(\mathbf{f}_*|\mathbf{u})p(\mathbf{u}|\mathbf{y})d\mathbf{u} \tag{15}$$

$$\approx \int p(\mathbf{f}_*|\mathbf{u})q(\mathbf{u})d\mathbf{u} \tag{16}$$

$$= \mathcal{N}(\mathbf{f}_*|\mathbf{A}_*\mathbf{m}, \mathbf{K_{X_*X_*}} + \mathbf{A}_*(\mathbf{S} - \mathbf{K_{ZZ}})\mathbf{A}_*^T), \tag{17}$$

where $\mathbf{A}_*$ indicates projection from the landmark points $\mathbf{Z}$ to the new sequences $\mathbf{X}_*$. The predictive distribution is a Gaussian distribution for the latent test values $\mathbf{f}_*$, from which the distributions of the test labels can be retrieved through the link function.

We have implemented our model using GPflows VGP-model (Matthews *et al.*, 2017; Opper and Archambeau, 2009). Our code, data sets, and some examples can be found from `https://github.com/emmijokinen/TCRGP`.

2.6 Multiple kernel learning

As mentioned in Section 2.1, when a TCR binds to a pMHC, its CDR3β is presumably always in contact with the peptide while the other CDRs may contact the peptide, but mainly contact the peptide binding groove of the MHC presenting the peptide. Dash *et al.*, 2017 took this into account by giving fixed weights for the distances between amino acids within different CDRs, giving more weight to the CDR3. As it can vary which CDRs can be in contact with different peptides, we did not want to determine the importance of these different CDRs beforehand, but instead created separate kernels for each CDR and let our model decide which of them are important. We define the kernel as a convex combination of the four CDR regions r and the two chains c,

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\substack{r \in \\ \{1,2,2.5,3\}}} \sum_{\substack{c \in \\ \{\alpha,\beta\}}} w_{cr} k_{cr}(\mathbf{x}, \mathbf{x}'; \theta_{cr}), \tag{18}$$

where the weights $w_{cr} \geq 0$ are non-negative.

3 Results

In Section 3.1 we use data from (Dash *et al.*, 2017) and present how our model TCRGP performs when different CDRs are available and compare TCRGP with TCRdist. The TCR-classifier by De Neuter *et al.*, 2018 is intended to be used only with TCRβ and does not utilize other CDRs than CDR3, we thus include it to our comparisons only in Section 3.2, where we compare these three methods with the data obtained from VDJdb.

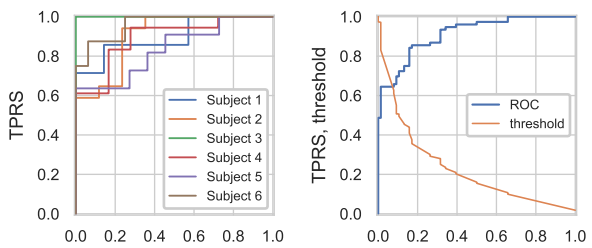

Figure 4: Left panel shows the AUROC curves for each subject in the Dash *et al.* (2017) data set for BMLF, when TCRGP has been trained using TCRα and TCRβ and all CDRs. The mean AUC is 0.905. Right panel shows the AUROC curves when the predictions have been combined and also the threshold values for classification are shown. From this figure we can determine which threshold values correspond to different true positive rates (TPRS) and false positive rates (FPRS).

3.1 Evaluating the significance of utilizing different CDRs

To evaluate the benefit of using different CDRs, we trained our model using either TCRα, TCRβ, or both with either only CDR3 or also with CDR1, CDR2, and CD2.5. We applied leave-one-subject-out cross-validation as described in Section 2.2. Figure 4 presents how the cross-validation was done for epitope BMLF and demonstrates how the classification results vary between different subjects likely due to the high variety of the TCRs.

For comparison, we also trained TCRdist in the same manner. From Figure 5, we can see that with both methods, on average using TCRβ is better than using TCRα, but using both sequences generally provides the best results. There are few exceptions, as can be seen from the Supplementary Figure S1 – for example with peptides pp65 and m139 it seems that both models perform better when using CDR3α instead of CDR3β. Also, on average utilizing all CDRs benefits the model when using TCRα, TCRβ, or both. Notably TCRGP outperforms TCRdist in all cases, except with BMLF when the models have been trained using only CDRβ, there the AUC-score for TCRGP is 0.818 and for TCRdist 0.821.

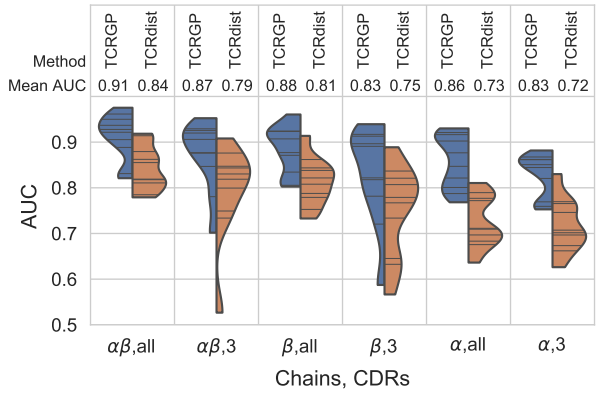

Figure 5: The blue parts of the violin plots illustrate the AUC-scores of predictions made by TCRGP for all the epitopes. The orange sides illustrate the AUC-scores obtained with TCRdist. A horizontal line within a violin plot presents the mean AUC-score obtained for one epitope. The used chains (α and/or β) and CDRs (three or all) are indicated below each panel.

Figures 5 and S1 also show that the AUC-scores can have notable differences between different epitopes even when same sequence information has been utilized. Some of these differences may be explained by the differences in the number of available training samples, for example for pp65 there were only 76 TCRs from 6 subjects in this data set. Also, the TCRs that recognize one epitope can be more diverse than the TCRs that recognize another epitope (Dash *et al.*, 2017), and if the TCRs are very diverse, it requires more sampling to get a representative sample of these TCRs for the training. It is also possible that some of the background TCRs that we do not expect to recognize any epitopes happen to recognize some of the epitopes in this data set and therefore create some error in the results.

We also examined more closely how TCRGP weighted the kernels created for the different CDRs, when all CDRs from both chains were utilized. Figure 6 illustrates which CDRs were found important for the different epitopes.

From this figure we can see that with most epitopes, CDR3β has been given most weight as one might expect, but using also other CDRs was found beneficial with all epitopes. However, with some epitopes CDR3β was not considered very important, for example with epitope m139 it seems that CDR3α is more important for the prediction, while with BMLF most of the weight was given to CDR2α and CDR2β.

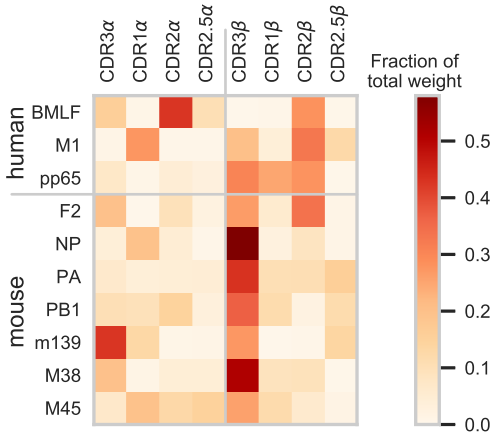

Figure 6: Fractions of weights given to kernels for different CDRs, when TCRGP has been trained for in epitopes in the data set from Dash *et al.* (2017) using both TCR chains and all CDRs.

3.2 Comparisons to other methods

We also experimented with the data we obtained from VDJdb and again used leave-one-subject-out cross-validation as described in Section 2.2. We trained TCRGP and TCRdist using only CDR3β and then also with the other CDRβs. Figure 7 shows the AUROC curves when TCRGP was trained with all CDRs for epitope ATDALMTGY.

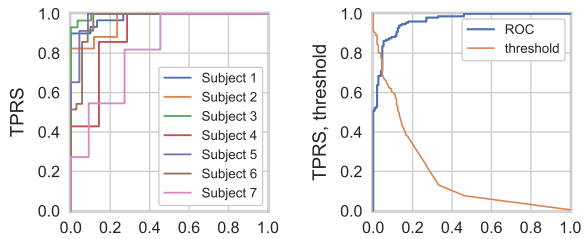

Figure 7: Left panel shows AUROC curves for each subject in the VDJdb data set for ATDALMTGY (HCV, NS3), when TCRGP has been trained using TCRα and TCRβ with all CDRs. The mean AUC is 0.944. Right panel shows the AUROC curves when all predictions have been combined and also the threshold values for classification are shown. From this figure we can determine which threshold values correspond to different true positive rates (TPRS) and false positive rates (FPRS).

We also trained a TCR-classifier by De Neuter *et al.*, 2018 using the same data. Unfortunately, the background TCR data set from Dash *et al.*, 2017 did not contain information of the J-gene, which is requested by this TCR-classifier. However, according De Neuter *et al.*, 2018 themselves, not much weight was given to the J-gene at least in their experiments. This TCR-classifier does not utilize other CDRβs in addition to CDR3β, but the Vβ-gene from which the CDRβs can be derived from. Thus all these three methods get the same sequence information, when TCRGP and TCRdist use all CDRs, although in a slightly different form.

Figure 8 shows the distributions of mean AUC scores for each model trained for the 22 different epitopes. Also with this data set from VDJdb, we can see that TCRGP and TCRdist both perform better, when all CDRβs have been utilized. Remarkably, TCRGP outperforms the other methods when using all CDRs, but also when only the CDR3β is utilized.

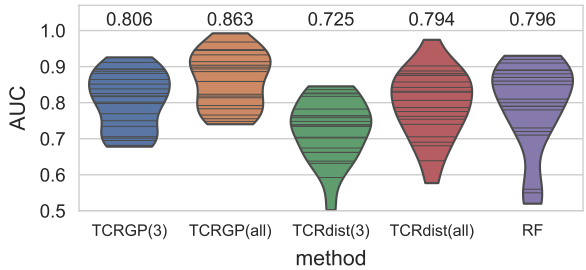

Figure 8: One violin plot presents the mean AUC-scores obtained with one method for all epitopes in our data set from VDJdb. Below each violin plot there is the name of the method used and in the brackets which CDRs have been used (3 for CDR3, all for CDR1, CDR2, CDR2.5, and CDR3). A horizontal line within a violin plot presents the mean AUC-score obtained for one epitope. RF refers to the Random Forest TCR-classifier of De Neuter *et al.*, 2018.

4 Discussion

In this paper we have demonstrated that we can accurately predict if a previously unseen TCR can recognize an epitope, for which we have had sufficient amount of training data. We have also shown that the other CDRs in addition to CDR3β can provide useful information for the classification task and that it can depend on the epitope in question which of the CDRs are important. The currently available sequence data of epitope-specific TCRs has allowed us to come this far, but it will be interesting to see what can be achieved when more data becomes available with modern high-throughput techniques presented recently (Bentzen *et al.*, 2018; Zhang *et al.*, 2018). Because of the limited data, we have not yet been able to consider the similarities of the epitopes and the significance of the different HLA-types of the MHC proteins presenting the

epitopes. Having a larger variety of epitopes and TCRs that recognize them, would allow us to also better model the cross-reactivity of the TCRs.

Eventually, when larger dataset become available, we may be able to model the similarity of the epitopes and consider the HLA-types in addition to the similarity of the TCRs and even predict if a TCR can recognize a previously unseen epitope. Furthermore, the proposed Gaussian process formalism has been shown to scale to very large datasets up to billion data points (Salimbeni and Deisenroth, 2017) by optimizing a small number of landmark sequences.

As the number of mixed repertoire data of TCRs of unkown-specifities in clinical settings is highly increasing (Emerson *et al.*, 2017; Savola *et al.*, 2017; Tumeh *et al.*, 2014), we expect that computational models like ours can be applied to a variety of research questions where exhaustive *ex vivo* pMHC-multimer assays are not feasible. Applications where our model can prove to be useful includes diagnosis of infectious diseases, linking antiviral immunity to initiation of autoimmune disorders and cancer immunotherapy.

Acknowledgements and Funding

This work has been supported by The European Research Council (M-IMM project), Academy of Finland (project numbers: 287224, 299915, 313271, 314442 and 314445), Finnish special governmental subsidy for health sciences, research and training, the Sigrid Juselius Foundation and the Finnish Cancer Societies

References

Allison, T. J. *et al.* (2001). Structure of a human $\gamma\delta$ T-cell antigen receptor. *Nature*, **411**(6839), 820.

Arstila, T. P. *et al.* (1999). A direct estimate of the human $\alpha\beta$ T cell receptor diversity. *Science*, **286**(5441), 958–961.

Bassing, C. H. *et al.* (2002). The mechanism and regulation of chromosomal V(D)J recombination. *Cell*, **109**(2), S45–S55.

Bentzen, A. K. *et al.* (2018). T cell receptor fingerprinting enables in-depth characterization of the interactions governing recognition of peptide–MHC complexes. *Nature biotechnology*, **36**(12), 1191.

Blei, D. M. *et al.* (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, **112**(518), 859–877.

Cabaniols, J.-P. *et al.* (2001). Most α/β T cell receptor diversity is due to terminal deoxynucleotidyl transferase. *Journal of Experimental Medicine*, **194**(9), 1385–1390.

Dash, P. *et al.* (2017). Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, **547**(7661), 89.

Davis, M. M. and Bjorkman, P. J. (1989). A model for t cell receptor and mhc/peptide interaction. In *Mechanisms of Lymphocyte Activation and Immune Regulation II*, pages 13–16. Springer.

Davis, M. M. *et al.* (2003). Dynamics of cell surface molecules during T cell recognition. *Annual review of biochemistry*, **72**(1), 717–742.

De Neuter, N. *et al.* (2018). On the feasibility of mining CD8+ T cell receptor patterns underlying immunogenic peptide recognition. *Immunogenetics*, **70**(3), 159–168.

Emerson, R. O. *et al.* (2017). Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire. *Nature genetics*, **49**(5), 659.

Giguere, S. *et al.* (2013). Learning a peptide-protein binding affinity predictor with kernel ridge regression. *BMC bioinformatics*, **14**(1), 82.

Glanville, J. *et al.* (2017). Identifying specificity groups in the T cell receptor repertoire. *Nature*, **547**(7661), 94.

Gras, S. *et al.* (2009). Structural bases for the affinity-driven selection of a public TCR against a dominant human cytomegalovirus epitope. *The Journal of Immunology*, **183**(1), 430–437.

Hegde, P. *et al.* (2018). Variational zero-inflated Gaussian processes with sparse kernels. In *Uncertainty in Artificial Intelligence*.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, **89**(22), 10915–10919.

Hensman, J. *et al.* (2015). Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*.

Lefranc, M. (1999). The IMGT unique numbering for immunoglobulins, T-cell receptors, and Ig-like domains. *Immunologist*, **7**(4), 132–136.

Lefranc, M.-P. and Lefranc, G. (2001). *The T cell receptor FactsBook*. Elsevier.

Matthews, A. G. d. G. *et al.* (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, **18**(40), 1–6.

Miho, E. *et al.* (2018). Computational strategies for dissecting the high-dimensional complexity of adaptive immune repertoires. *Frontiers in immunology*, **9**, 224.

Nickisch, H. and Rasmussen, C. (2008). Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, **9**, 2035–2078.

Opper, M. and Archambeau, C. (2009). The variational Gaussian approximation revisited. *Neural computation*, **21**(3), 786–792.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.

Robins, H. S. *et al.* (2009). Comprehensive assessment of T-cell receptor β-chain diversity in $\alpha\beta$ T cells. *Blood*, **114**(19), 4099–4107.

Rudolph, M. G. *et al.* (2006). How TCRs bind MHCs, peptides, and coreceptors. *Annu. Rev. Immunol.*, **24**, 419–466.

Ruiz, M. and Lefranc, M. (2000). Currents in computational molecular biology. *Frontiers Science Series*, (30), 126–127.

Salimbeni, H. and Deisenroth, M. (2017). Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599.

Savola, P. *et al.* (2017). Somatic mutations in clonally expanded cytotoxic T lymphocytes in patients with newly diagnosed rheumatoid arthritis. *Nature Communications*, **8**, 15869.

Sewell, A. K. (2012). Why must T cells be cross-reactive? *Nature Reviews Immunology*, **12**(9), 669.

Shugay, M. *et al.* (2017). VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic acids research*, **46**(D1), D419–D427.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

Tickotsky, N. *et al.* (2017). McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics*, **33**(18), 2924–2929.

Tumeh, P. C. *et al.* (2014). PD-1 blockade induces responses by inhibiting adaptive immune resistance. *Nature*, **515**(7528), 568.

Wooldridge, L. *et al.* (2012). A single autoimmune T cell receptor recognizes more than a million different peptides. *Journal of Biological Chemistry*, **287**(2), 1168–1177.

Zhang, S.-Q. *et al.* (2018). High-throughput determination of the antigen specificities of T cell receptors in single cells. *Nature biotechnology*, **36**(12), 1156.