On generative models of T-cell receptor sequences

Giulio Isacchini,[1,2,*] Zachary Sethna,[3,*] Yuval Elhanati,[3] Armita Nourmohammad,[1,4,5] Aleksandra M. Walczak,[2] and Thierry Mora[2]

[1]*Max Planck Institute for Dynamics and Self-organization, Am Faßberg 17, 37077 Göttingen, Germany*
[2]*Laboratoire de physique de l'École normale supérieure (PSL University), CNRS, Sorbonne Université, and Université de Paris, 75005 Paris, France*
[3]*Memorial Sloan Kettering Cancer Center, New York, New York 10065, USA*
[4]*Department of Physics, University of Washington, 3910 15th Avenue Northeast, Seattle, WA 98195, USA*
[5]*Fred Hutchinson cancer Research Center, 1100 Fairview ave N, Seattle, WA 98109, USA*

T-cell receptors (TCR) are key proteins of the adaptive immune system, generated randomly in each individual, whose diversity underlies our ability to recognize infections and malignancies. Modeling the distribution of TCR sequences is of key importance for immunology and medical applications. Here, we compare two inference methods trained on high-throughput sequencing data: a knowledge-guided approach, which accounts for the details of sequence generation, supplemented by a physics-inspired model of selection; and a knowledge-free Variational Auto-Encoder based on deep artificial neural networks. We show that the knowledge-guided model outperforms the deep network approach at predicting TCR probabilities, while being more interpretable, at a lower computational cost.

I. INTRODUCTION

Deep learning methods are proving a very useful approach in many areas of physics and the natural sciences [1, 2]. These algorithms are successful in identifying hidden patterns in large amounts of data, often helping make progress in situations where traditional analyses reach their limits [3–5]. Despite the black box aspect of how the algorithm works and the lack of interpretability of the model features, machine learning is undoubtedly useful, especially in cases where the natural system of interest escapes our intuition or knowledge. However, as we show here on the example of immune repertoires, introducing physical or biological intuition into data-driven models can outperform basic uninformed machined learning approaches.

The adaptive immune system is made up of a large ensemble of diverse lymphocyte receptors that recognize different pathogens. The receptors expressed on the surface of T cells (T cell receptors - TCR) are generated by randomly assembling genomic templates for three genes (variable—V, diverse—D and junction—J) that make up of the so-called β chain, and two (V and J) for the α chain. Additionally to this combinatoric diversity, non-templated nucleotides are added at the junctions between these templates and nucleotides are deleted. Such recombined DNA forms the newly generated TCR that later undergoes thymic selection that tests for its ability to form a receptor protein and bind, albeit not too strongly, proteins that are natural to the host organism [6]. TCR that pass thymic selection are released into the periphery and form the naive repertoire (i.e. non-stimulated by foreign antigens). Due to the random addition and deletion of nucleotides, receptors sequences have different lengths and some are even out of frame or have stop codons, in which case they are called nonproductive. Conversely, sequences with no frameshift nor stop codon are conventionally called productive. High-throughput immune repertoire sequencing experiments sample blood from individual hosts, sort out TCRs, and sequence this subset [7–9]. Analysis of this kind of data makes it possible to characterize the statistics of both generated and naive repertoires.

TCR sequences differ from classical protein families, which are grouped by function and across species [10]. Those families are believed to have evolved over long time scales under a shared selective pressure that shapes their statistics. For such families, physics-inspired statistical inference methods have helped to predict contacts between amino-acids in the protein [11], define sectors of co-evolving residues [12], or find interaction partners [13]. Deep [14] and non-deep [15] machine learning approaches have also been successfully applied. By contrast, TCR generation is fairly well understood mechanistically. Previously we developed a statistical inference technique that uses biological knowledge of the underlying assembly processes to learn the statistics of generation and calculate the generation probability of each TCR sequence [16, 17]. Since thymic selection involves many specific interactions with antigen-presenting cells, modeling it from first principles is more difficult. Nevertheless, simple models of selection based on the assumption of an additive fitness [18] have been shown to well recapitulate some key statistics of these ensembles [19, 20]. However, a direct test of the performance of this method for the abundance of specific sequences in large cohorts is still lacking.

Recently, Davidsen et al. [21] described an elegant approach for learning the distribution of T-cell receptor beta sequences (TCRβ or simply TCR in the follow-

*These authors contributed equally.

ing), based on a Variational Auto-Encoder (VAE). The method makes it possible to generate new sequences with the same statistics as real repertoires, and to evaluate the frequency of individual sequences, which agree with the data with good accuracy. Its main strength is that it does not take any information about the origin of these sequences through VDJ recombination and thymic and peripheral selection. Yet it manages to extract statistical regularities imprinted by these processes.

Here we compare the VAE method [21] with the previously proposed model of generation and selection, called SONIA [19, 20]. We compare their performances for predicting the distribution of TCR sequences in controlled conditions, training and validating on the same datasets. Contrary to the claims of the original VAE paper [21], we show that that knowledge guided models perform as well as the variational auto-encoder or even better, at a lower computational cost.

II. MODEL DEFINITIONS

A. Knowledge-guided model

To predict the probability distribution of TCR sequences, we build a generative model that proceeds in two steps: initial generation, and selection.

First, a recombination model for the probability of generation of a sequence σ, denoted by $P_{\text{gen}}(\sigma)$, is learned from failed, nonproductive rearrangements, which are free of selection biases [16, 17]. This model describes in detail the probabilities of V, D, and J usages, and of deletion and insertion profiles. Calling E the collective variable describing the recombination scenario, the model predicts its probability $P_{\text{scenario}}(E)$. Its parameters are learned through Expectation-Maximization using the IGoR software [17].

Although the model is trained on non-productive sequences, it can be used to predict the probability of any sequence. Denoting $\hat{\sigma}(E)$ the amino-acid sequence produced by scenario E, we define the generation probability of a productive amino-acid sequence σ as:

$$P_{\text{gen}}(\sigma) = \frac{1}{F} \sum_E P_{\text{scenario}}(E) \mathbb{I}[\hat{\sigma}(E) = \sigma], \qquad (1)$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $F = \sum_E P_{\text{scenario}}(E)\mathbb{I}(\hat{\sigma}(E) \text{ is productive})$ is the probability that a random recombination scenario results in a productive sequence. More precisely, σ is defined by the choice of V and J genes (σ_V and σ_J), as well as the amino-acid sequence of the Complementarity Determining Region 3 (CDR3) that lies between V and J, $\sigma_1, \ldots, \sigma_L$. The sum in (1) involves a large number of terms due to the degeneracy of both the genetic code and the recombination process, but it can be done using a recursive technique akin to transfer matrices implemented in the OLGA software [22].

Second, a model of selection, called SONIA [20], is learned on top of the generation probability P_{gen} to describe the distribution of productive sequences,

$$P_{\text{SONIA}}(\sigma) = Q(\sigma) P_{\text{gen}}(\sigma), \qquad (2)$$

where

$$Q(\sigma) = \frac{1}{Z} \exp\left[h_{VJL}(\sigma_V, \sigma_J, L) + \sum_{i=1}^{L} h_{i,L}(\sigma_i) \right] \qquad (3)$$

is a selection factor calculated through additive "fields" h acting on the sequence elements, similarly to additive position-weight matrix models first introduced for DNA binding sites [18].

Within this framework, we can define three models according to the parametrization of h. In the first two models, the VJL field is decomposed as $h_{VJL}(\sigma_V, \sigma_J, L) = h_{VJ}(\sigma_V, \sigma_J) + h_L(L)$. A first model in which $h_{i,L}$ is left unconstrained is called the "Length-Position" (LP) model. This choice corresponds to the original model of [19], in which the selective pressure on each amino-acid may depend on the sequence length L. However, observations [19] suggest that these factors are to some extend independent of L. This invariance can be incorporated by assuming that the field can be decomposed into two contributions depending on the position of the amino acid from the right and left ends of the CDR3: $h_{i,L} = h_{i,\text{right}} + h_{L-i+1,\text{left}}$. The resulting "Left+Right" (LR) model has much fewer parameters and is less likely to overfit the data. For these two models, parameters are learned by maximizing the log-likelihood with an L^2 regularization using gradient ascent, as specified in Ref. [20].

In addition, because no software implementation of the selection model was provided with the original article [19], Davidsen et al. [21] compared their VAE approach to a reduced version of this selection model (not examined in [19]), which they call `OLGA.Q`. In that model, only VJ usage and CDR3 length were included: $h_{i,L} = 0$. Its parameters h_{VJL} were fitted by maximizing the likelihood analytically.

B. Variational Auto-Encoder

A VAE is an auto-encoder whose structure can be used as a generative probabilistic model. A good introduction can be found in Ref. [23]. In short, a VAE consists of a probabilistic encoder $q(z|\sigma)$ and a probabilistic decoder $p(\sigma|z)$, converting the sequence into a continuous multidimensional latent variable z and back. The goal of the encoder is to make the probabilistic mapping from σ to itself through q and p as faithful as possible, while at the same time making the distribution of the latent variable z as close as possible to a simple distribution, i.e. multivariate Gaussian with unit covariance.

Both p and q are parametrized by deep neural networks, whose parameters are optimized for these two objectives, using stochastic gradient descent. Once the

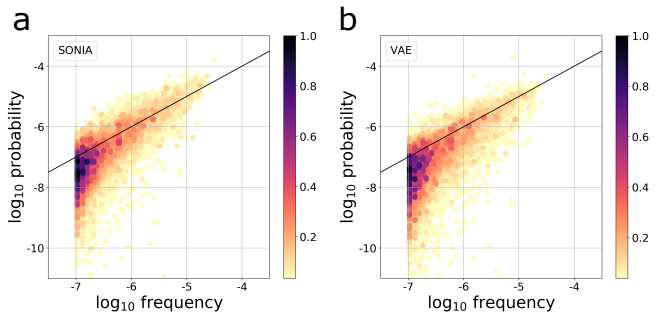

FIG. 1: Predicted TCR sequence probabilities (y-axis) versus empirical frequencies (y-axis), for (a) the SONIA Left+Right model ($\rho^2 = 0.53$) and (b) the VAE model ($\rho^2 = 0.47$). Models were trained on $2 \cdot 10^5$ sequences sampled from the training set assembled from the TCR β repertoires of 666 donors [25]. Frequencies refer to empirical frequencies in the same datasets. The SONIA model was built on top of a P_{gen} model trained on $2 \cdot 10^5$ non-productive sequences from the same donors.

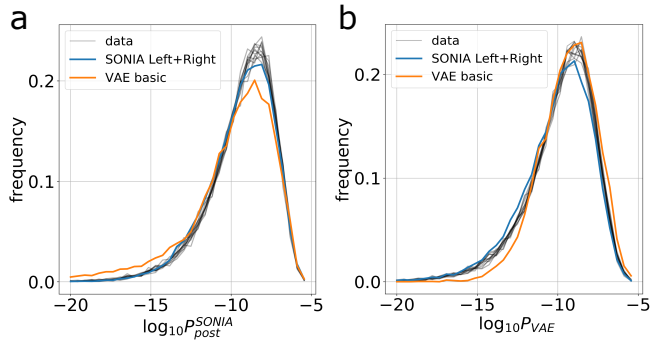

FIG. 2: (a) Distribution of P_{SONIA} of TCRs from 11 individuals from Ref. [26], as well as sequences generated by the SONIA Left+Right model, and the VAE. The SONIA model was trained on a set of 10^5 sequences, on top of the P_{gen} model trained for Fig. 1. (b) Distribution of P_{VAE} for the same sequences as in (a).

model is learned, new sequences can be generated by drawing z from $p_0(z)$, and σ from $p(\sigma|z)$, so that σ is distributed according to $P_{\text{VAE}}(\sigma) = \int dz\, p(\sigma|z) p_0(z)$. In practice, the predicted probability of a given sequence $P_{\text{VAE}}(\sigma)$ is evaluated using Monte-Carlo importance sampling. In Ref. [21], a variant of the traditional auto-encoder detailed in [24] was used. Here we focus on the version of the VAE called `basic` in that paper.

III. MODEL COMPARISON

A. Datasets and model training

The data consists of TCRβ sequence repertoires of 666 individuals [25]. We use the exact same procedure, dataset, and subsamples as in [21] for reproducibility.

For each individual, read counts are first discarded as they stem from clonal expansions. To train an initial P_{gen} model on which SONIA is built and trained, we used $2 \cdot 10^5$ nonproductive sequences drawn randomly from all donors. For all models, unique amino-acid sequences were first separated into a training and a testing dataset of equal sizes. All models were then trained on $2 \cdot 10^5$ or 10^6 TCRβ sequences randomly sampled from the training dataset with replacement, according to their frequency in the cohort, counting each unique nucleotide sequence in each patient. Their performance was assessed by their ability to predict the frequency of sequences from the testing set, $P_{\text{data}}(\sigma)$.

	10^6 ρ^2	10^6 D_{KL}	$2 \cdot 10^5$ ρ^2	$2 \cdot 10^5$ D_{KL}
VAE	0.48	1.7	0.47	2.0
P_{gen}	0.48	4.5	0.51	4.5
`OLGA.Q`	0.48	2.6	0.47	2.6
SONIA LP	0.52	1.8	0.52	1.7
SONIA LR	0.53	1.4	0.53	1.4

TABLE I: Pearson's correlation coefficients ρ^2 and Kullback-Leibler divergence D_{KL} (in bits) for the various models. Either 1 million or 200,000 sequences were used in the training dataset.

B. Predicting sequence frequencies

We used two measures of performance: Pearson's ρ^2 between the logarithms of the frequencies as in [21], and the Kullback-Leibler divergence: $D_{KL} = \langle \log_2[P_{\text{data}}(\sigma)/P_{\text{model}}(\sigma)] \rangle$ (model = VAE or SONIA), where the average $\langle \cdot \rangle$ is taken over 10^4 sequences from the testing set, sampled according to their relative frequencies within that set. We excluded $\sim 0.3\%$ of sequences for which $P_{\text{gen}} = 0$, probably due to sequencing errors. Note that, if not for the L^2 regularization, maximizing the log-likelihood would be equivalent to minimizing the D_{KL}. The scale of D_{KL} may be compared to the total entropy of the ensemble, $-\sum_\sigma P_{\text{SONIA}}(\sigma) \log_2 P_{\text{SONIA}}(\sigma) \approx 31$ bits [20].

Fig. 1 shows the predicted frequencies of the Left+Right SONIA model and the VAE model, both trained on the same $2 \cdot 10^5$ sequences, and compares them to data. The performances of all models and both datasets are reported in Table I. SONIA models perform generally better than the VAE, especially the Left+Right model which is the best model according to both measures of performance. Note that the Length-Position model of Ref. [19], also performs as well as the VAE. Davidsen et al. [21] did not compare their model to it owing to the absence of a readily available implementation.

Strikingly, even the basic model of generation with no selection ($h = 0$), P_{gen}, performs comparably to the

VAE, and sometimes better according to the ρ^2 measure, despite the model being trained on nonproductive sequences. Accordingly, the `OLGA.Q` model, which adds a minimal layer of selection on top of P_{gen}, also performs very well. These results differ substantially from the $\rho^2 = 0.26$ - 0.27 reported in [21] for `OLGA.Q`. In [21], the default model for P_{gen} was not actually trained on the dataset of interest, but rather used with its default parameters learned from a different dataset, which explains the poor reported performance.

We can also compare the two models by asking whether the distribution of frequencies are well reproduced by one another, using another TCR dataset from [26] to allow for a direct comparison to the results of Ref. [21] (Fig. 2). Both the VAE and SONIA agree with the data in their distribution of P_{model}. VAE-generated sequences have the same distribution of P_{SONIA} as SONIA-generated sequences, with a slight under-estimation of the distribution peak, and an excess of low-frequency sequences (Fig. 2a). The converse is true when looking at the distribution of P_{VAE} for SONIA- versus VAE-generated sequences (Fig. 2b). This suggests that the VAE and SONIA capture some features of the sequence statistics that are distinct from one another.

C. Computational times

SONIA is an order of magnitude faster than the VAE, which uses Monte-Carlo sampling to calculate predicted frequencies. The average computing time for $P_{\text{SONIA}}(\sigma)$ is 14 ms per sequence on a laptop computer and 3 ms on a 16-core computer, versus 0.18 s for $P_{\text{VAE}}(\sigma)$ (no parallelization possible).

SONIA was also faster to train. It took 33 minutes to train a SONIA model on 10^6 sequences using a 30-core computer, to which one should add 31 minutes to train an IGoR model on $2 \cdot 10^5$ nonproductive sequences. For the same amount of data and on the same machine, the VAE took 7 hours to train.

IV. CONCLUSION

In summary, both approaches, VAE and SONIA, perform equally well, with perhaps a slight advantage for the latter. SONIA is also much faster. These results suggest that, while knowledge-free approaches such as the VAE perform well, there is still value in preserving the structure implied by the VDJ recombination process as a baseline for learning complex distributions of immune repertoires. Extending the SONIA model considered here beyond a simple linear combination of features, and taking ideas from the modeling strategy of the VAE, offers interesting directions for future improvement in repertoire modeling.

In a more general context, while machine learning approaches are undoubtably a very useful tool, they can be made even more powerful when combined with models that describe the underlying physics or biology. This is the case when training data is limited, as has been reported in complex image processing of non-animate matter [27]. As we show, even if data is abundant, using models to guide learning can help.

Code availability. All code for reproducing the figures of this comment can be found at `https://github.com/statbiophys/compare_selection_models_2019/`. The SONIA package upon which that code builds is available at `https://github.com/statbiophys/SONIA/` [20].

Acknowledgements. The work of GI, TM and AMW was supported by grant European Research Council COG 724208.

[1] Ching T, et al. (2018) Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* 15.
[2] Carleo G, et al. (2019) Machine learning and the physical sciences. *Rev. Mod. Phys.* 91.
[3] Yamins DLK, Dicarlo JJ (2016) Using goal-driven deep learning models to understand sensory cortex.
[4] Senior AW, et al. (2020) Improved protein structure prediction using potentials from deep learning. *Nature* 577.
[5] Yang KK, Wu Z, Arnold FH (2019) Machine-learning-guided directed evolution for protein engineering.
[6] Yates AJ (2014) Theories and quantification of thymic selection. *Front. Immunol.* 5:13.
[7] Heather JM, Ismail M, Oakes T, Chain B (2017) High-throughput sequencing of the T-cell receptor repertoire: pitfalls and opportunities. *Brief. Bioinform.* 19:554–565.
[8] Minervina A, Pogorelyy M, Mamedov I (2019) TCR and BCR repertoire profiling in adaptive immunity. *Transpl. Int.* pp 0–2.
[9] Bradley P, Thomas PG (2019) Using T Cell Receptor Repertoires to Understand the Principles of Adaptive Immune Recognition. *Annu. Rev. Immunol.* 37:547–570.
[10] El-gebali S, et al. (2019) The Pfam protein families database in 2019. 47:427–432.
[11] Cocco S, Feinauer C, Figliuzzi M, Monasson R, Weigt M (2018) Inverse Statistical Physics of Protein Sequences: A Key Issues Review. *Rep. Prog. Phys. 81* 81:032601.
[12] Halabi N, Rivoire O, Leibler S, Ranganathan R (2009) Protein Sectors: Evolutionary Units of Three-Dimensional Structure. *Cell* 138:774–786.
[13] Bitbol AF, Dwyer RS, Colwell LJ, Wingreen NS (2016) Inferring interaction partners from protein sequences. *Proc. Natl. Acad. Sci. U. S. A.* 113:12180–12185.
[14] Riesselman AJ, Ingraham JB, Marks DS (2018) Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* 15:816–822.
[15] Tubiana J, Cocco S, Monasson R (2019) Learning protein constitutive motifs from sequence data. *Elife* 8.

[16] Murugan A, Mora T, Walczak AM, Callan CG (2012) Statistical inference of the generation probability of T-cell receptors from sequence repertoires. *Proc. Natl. Acad. Sci.* 109:16161–16166.

[17] Marcou Q, Mora T, Walczak AM (2018) High-throughput immune repertoire analysis with IGoR. *Nat. Commun.* 9:561.

[18] Berg OG, von Hippel PH (1987) Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. *J. Mol. Biol.* 193:723–750.

[19] Elhanati Y, Murugan A, Callan CG, Mora T, Walczak AM (2014) Quantifying selection in immune receptor repertoires. *Proc. Natl. Acad. Sci.* 111:9875–9880.

[20] Sethna Z, et al. (2020) Population variability in the generation and thymic selection of T-cell repertoires. pp 1–17.

[21] Davidsen K, et al. (2019) Deep generative models for T cell receptor protein sequences. *eLife* 8:e46935.

[22] Sethna Z, Elhanati Y, Callan CG, Walczak AM, Mora T (2019) OLGA: fast computation of generation probabilities of B- and T-cell receptor amino acid sequences and motifs. *Bioinformatics* 35:2974–2981.

[23] Kingma DP, Welling M (2019) An Introduction to Variational Autoencoders. *Found. Trends Mach. Learn.* 12:307–392.

[24] Higgins I, et al. (2017) beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *ICLR 2017* 1:1–22.

[25] Emerson RO, et al. (2017) Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire. *Nat. Genet.* 49:659–665.

[26] De Neuter N, et al. (2019) Memory CD4+ T cell receptor repertoire data mining as a tool for identifying cytomegalovirus serostatus. *Genes and Immunity* 20:255–260.

[27] Colas J, et al. (2019) Nonlinear denoising for characterization of solid friction under low confinement pressure. *Phys. Rev. E* 100:32803.