Type 1 Diabetes project: Prediction of T1D status based on TCR repertoire and identification of T1D-associated TCR clonotypes

Kseniia Lupyr [1] **Artemiy Dakhnovets** [1] **Valeriia Vladyina** [1]

Abstract

Type 1 diabetes (T1D) is a chronic autoimmune disease in which the immune system destroys insulin-producing beta cells in the pancreas. The primary treatment is insulin injections, which do not prevent severe complications. A new approach aims to target autoreactive T cells, the specificity of which is defined by the T-cell receptor (TCR). In this project, we classify the TCR repertoires of patients with T1D and healthy individuals and search for T1D-associated TCR clonotypes (features that have the biggest impact on the model). The results can potentially be applied for diagnostics and new therapy development. The project implements ML methods for published and our own unpublished data of patients with T1D and healthy controls.

Github repo: T1D ML project
Presentation file: link

1. Introduction

In Type 1 diabetes (T1D) the immune system attacks and destroys the insulin-producing beta cells in the pancreas. The only generally accepted treatment for T1D is insulin therapy. However, this therapy addresses the consequences of the disease and does not prevent disease progression or the development of severe chronic complications. There are reasons to believe that T-cell targeted treatment that aims at the cause of the disease is possible for T1D.

T-cells are key players of adaptive immunity. On the one hand, they maintain cytotoxic activity; on the other hand, they orchestrate immune response by activating or inhibiting other immune cells. T-cells specifically recognize a peptide in a complex with an HLA molecule on the surface of a host cell.

T1D-associated HLA haplotypes DR3-DQ2 and DR4-DQ8 are present in up to 90% of individuals with T1D (Redondo MJ, 2018; Noble JA, 2011). Moreover, it is known that genetic variations that are associated with a high expression of proinsulin in the thymus cause a T1D protective effect by enhancing T cell tolerance (Pugliese A, 1997; Sabater L, 2005; Vafiadis P, 1997). Additionally, T1D-associated gene variants are particularly enriched in the open chromatin of stimulated CD4+ effector T cells (Robertson CC, 2021). Finally, a lot of drugs that affect T-cell function and antigen presentation have shown efficacy in T1D (Herold KC, 2024).

Each T cell acts with particular precision against a specific antigen through T cell receptor (TCR) which recognizes a peptide-MHC complex on the surface of host cells triggering T cell activation. The number of all functional TCRs in an organism forms an individual TCR repertoire, the remarkable diversity of which is assessed as 10^8 (Qi Q, 2014). TCR sequencing data – TCR repertoire – is a valuable source for TCR specificity prediction.

The complementarity-determining region 3 (CDR3) of the T-cell receptor (TCR) plays a pivotal role in the recognition and binding of antigenic epitopes in complex with HLA. TCRs are generated through a quasi-stochastic process—V(D)J recombination, involving random assembling of V, D, and J segments, trimming, and insertion of nucleotides in the junction regions. CDR3 regions exhibit remarkably high diversity through the V(D)J recombination mechanism, allowing them to adapt to a wide range of existing and potential antigens. In this project, we define TCR clonotype as a unique combination of CDR3 sequence, V, and J segment if it otherwise is not specified explicitly.

The aim of the project is to build a classifier of T1D status based on TCR repertoire and find T1D-associated TCR clonotypes.

For this purpose, TCR repertoires of healthy individuals and ones with T1D were obtained. **The dataset** consists of 408 TCR repertoires of patients with T1D and 165 TCR repertoires of healthy patients. All TCR libraries were made under the same protocol. The diversity of most of the sam-

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Kseniia Lupyr <kseniia.lupyr@skoltech.ru>.

Final Projects of the Machine Learning 2025 Course, Skoltech, Moscow, Russian Federation, 2025.

ples is between 50 000 and 300 000 clonotypes.

2. Literature review

High-throughput sequencing reveals repertoire skewing in autoimmune diseases and cancers, but analyzing millions of TCR sequences demands computational tools. ML methods demonstrated the ability to classify TCR repertoires by condition and to extract condition-associated TCRs from TCR repertoires in the case of CMV (Emerson RO, 2017; De Neuter N, 2019), cancer (Ostmeyer J, 2019), SARS-CoV-2 infection (Vlasova EK, 2023) and T1D (Rawat et al., 2024).

(Rawat et al., 2024) implemented a statistical approach, LogReg regression, and the DeepRC model from (Widrich et al., 2020) for the prediction of T1D status and identification of T1D-associated TCRs. To reduce the number of features, the authors used k-mers instead of the original clonotype sequences. However, this approach did not show efficacy.

The second approach used by (Rawat et al., 2024) is the implementation of the DeepRC model (Widrich et al., 2020). Transformer architectures excel at immune repertoire analysis by treating each TCR sequence as an instance in an MIL framework. (Widrich et al., 2020) demonstrates that attention layers in these models mathematically resemble updates in continuous-state Hopfield networks, enabling storage of exponentially more patterns than classical neural networks. This capability proves critical when detecting rare antigen-specific clones present at 0.01% frequency amidst background sequences.

Another multi-instance learning framework that can be used for TCR reprtoire classification is DeepTCR (Sidhom et al., 2021). Using data from HIV elite controllers, the model employs a multi-head attention mechanism with adaptive inverse square root unit (ISRU) activation to identify predictive TCRs within heterogeneous repertoires. This architecture recognizes that only a minority of clones drive antigen-specific responses—a finding corroborated by experimental validation showing high concordance between predicted and validated HIV-specific TCRs1. The attention weights provide interpretability, highlighting clonotypes that contribute most to repertoire-level predictions.

Furthermore, in a recent article (Zaslavsky et al., 2025), a multiclassifier trained on features from both B cell and T cell receptor data successfully classified the disease status of 542 individuals with COVID-19, HIV, lupus, type 1 diabetes, recent flu vaccination, and healthy controls, achieving a multiclass area under the receiver operating characteristic curve (AUROC) of 0.986 on testing data. The authors introduced three strategies for implementing a predictive model for disease classification. Specifically, they used three models per gene locus, BCR immunoglobulin heavy chain (IgH) and TCR beta chain (TRB), to identify immune states. Each model focused on different aspects of immune repertoires shared among individuals with the same diagnosis. The first model considered gene segment frequencies and IgH somatic hypermutation rates for each isotype. The second model examined highly similar clusters of CDR3 sequences. The third model addressed potential structural or binding similarities by training on embeddings of CDR3 sequences generated with the Evolutionary Scale Modeling-2 (ESM-2) protein language model. Finally, they implemented a meta-model based on a ridge logistic regression classifier trained on the output probabilities of the six base models (the aforementioned three models trained on BCR and TCR repertoire data separately) to map the combined predicted probability vectors to validation set sample disease labels. In the context of our final project, we plan to implement some of the described approaches for predicting type 1 diabetes, particularly the use of CDR3 sequence embeddings generated with protein language models, which appear promising for our data analysis.

The application of machine learning to the T cell repertoire classification has progressed from proof-of-concept studies to robust analytical pipelines that outperform traditional bioinformatics tools. Deep learning architectures now enable both exploratory analysis of repertoire diversity and precise prediction of TCR-repertoire interactions, with model interpretability methods providing unprecedented biological insights.

3. Experiments

We studied the structure of our data in detail and carried out normalization, feature selection, and division into training and test datasets. We implement classical ML models first on feature tables where each feature corresponds to the selected TCR clonotype and second on embeddings obtained with ESM-2. Additionally, we implemented the DeepRC model on our data.

3.1. Data pre-processing

The dataset consists of 414 TCR repertoires of patients with T1D and 165 TCR repertoires of healthy individuals. Moreover, the data are composed of independently obtained technical batches. See Table 1.

3.1.1. Data normalization

Each TCR repertoire is a set of TCR clonotypes that are defined as unique combinations of CDR3 sequence, TRBV gene, and TRBJ gene. These unique combinations serve as features for the construction of feature tables. TCR repertoires were obtained using unique molecular identi-

Status	Batch	TCR Repertoires
T1D	batch 1	158
	batch 2	250
Healthy	aging	65
	rosati	100

Table 1. Distribution of TCR repertoires across patient groups and data batches

fiers (UMIs) that allow us to estimate the initial number of RNA molecules in the sample. So, the abundance of each TCR clonotype is a number of corresponding UMIs.

The abundance of TCR clonotypes (features) varies depending mostly on sequencing depth. We investigated the diversity of TCR repertoires (Figure 3) and normalized each sample to 30,000 UMIs to ensure comparable diversity across all TCR repertoires. The normalization was made by the downsample command from Mixcr software (https://mixcr.com). Samples with a total number of UMIs less than 30,000 were not taken into analysis. As a result, we got 389 TCR repertoires of patients with T1D and 123 TCR repertoires of healthy individuals. The distribution by batches is in Table 2.

Status	Batch	TCR Repertoires
T1D	batch 1	153
	batch 2	230
Healthy	aging	57
	rosati	66

Table 2. Distribution of TCR repertoires across patient groups and data batches after normalization

The same number of UMIs in each sample allows us not to normalize count data after feature table construction from TCR repertoires. The chosen threshold for normalization saved most of the samples.

3.1.2. The logic behind train/test split

In our experimental design, we implemented a rigorous train-test split to ensure robust model evaluation. Unlike conventional random splitting, we deliberately constructed test and training datasets based on technically and biologically independent batches. Such methodology strengthens the validity of our findings and provides greater confidence that observed performance will translate to practical applications where new patient samples would inherently differ from those used during model development. The combinations of batches were chosen to maintain similar class balance in train and test sets. The distribution of batches and number of samples between train and test datasets is presented in Table 3.

dataset	status	batch	nb
train	T1D	T1D batch 2	230
	Healthy	rosati	66
test	T1D	T1D batch 1	153
	Healthy	aging	57

Table 3. Distribution of batches and number of samples between train and test datasets

3.2. Feature selection and feature table construction

The number of unique features (TCR clonotypes) for such data can be around 10^8. The real number of unique TCR clonotypes in our normalized training dataset is more than $5 \cdot 10^6$. This is clearly too many for any classical machine learning model, given that we have several hundred samples. For further work, we decided to carry out feature selection. We used two strategies: feature selection by TCRnet (Pogorelyy & Shugay, 2019) and feature selection by F-test. In both cases we select TCR clonotypes that are overrepresented in TCR repertoires of T1D patients.

Feature selection was performed on the train dataset by comparing T1D batch 2 data and healthy patients from the batch "rosati". Selected features were used for the construction of the feature tables. In the resulting tables, each column corresponds to a selected feature (TCR clonotype), and each row corresponds to the patient. The count values correspond to the number of UMIs for the given TCR clonotype in the TCR repertoire of the given patient. For each feature selection method, two feature tables were constructed: for training and test datasets.

3.2.1. Feature selection by TCRnet

TCRnet utilizes the idea that the immune system responds to the challenge with numerous T-cells with similar TCRs. Therefore, in TCR repertoires of patients with T1D, T1D-associated TCR clonotypes would have more similar sequences than the same sequences in TCR repertoires of healthy individuals.

To compare TCR repertoires under different conditions, we merged all normalized TCR repertoires of T1D patients and separately merged normalized TCR repertoires of healthy donors. TCRnet calculated the number of similar sequences for each unique TCR clonotype in the merged T1D dataset and in the merged healthy dataset. If the number of similar TCR clonotypes for a given TCR clonotype is significantly higher in the T1D dataset than in the "background" healthy dataset, we select such a TCR clonotype as an interesting feature for further analysis. We took p-value ¡ 10^{-10} as a threshold for selection. It gave us 646 selected features.

3.2.2. Feature selection by F-test

TCR clonotype occurrences in the "rosati" healthy dataset and T1D batch 2 dataset were calculated with a further one-sided F-test implementation using R functionality. P-value ¡ 0.05 was used as a threshold for feature selection. 533 TCR clonotypes were selected for feature table construction.

3.3. Embedding construction

Another strategy to numerically represent TCR repertoires is to implement protein language models that encode the context of amino acids within CDR3 sequences and then average these amino acid embeddings across all CDR3 sequences belonging to the same TCR repertoire. Consequently, per-TCR-repertoire embeddings are computed as a weighted average of aggregated CDR3 embeddings, with the weights being the abundances of CDR3 sequences in the repertoire. To build these embeddings, we chose the ESM-2 protein language model developed by Meta AI (Lin et al., 2022). This model uses a deep neural network with multiple transformer layers to generate context-aware, high-dimensional embeddings for each amino acid in a protein sequence, capturing structural, functional, and evolutionary information. ESM-2 has various implementations with varying numbers of parameters and layers, pretrained on a massive dataset of protein sequences by masking parts of the sequences and predicting the missing amino acids (similar to masked language modeling in NLP). To date, ESM-2 outperforms all tested single-sequence protein language models across a range of structure prediction tasks and enables atomic resolution structure prediction. In the context of our project, we generated embeddings for TCR repertoires using the ESM-2 T33 UR50D model, which has 640 million parameters and 33 layers. This particular implementation produces high-dimensional embeddings with 1280 dimensions. Additionally, TCR repertoire entropy was used as an extra feature to capture repertoire diversity. To build embeddings we used CDR3 sequence data only, excluding information regarding TRBV and TRBJ genes.

3.4. Classification models used

3.4.1. Models used with TCRnet and F-test selected features

As mentioned earlier for the feature selection we use two methods TCRnet and F-test. The first experiments with logistic regression showed that the set of features based on the F-test generally led to higher F1-scores and ROC-AUC values compared to those obtained with TCRnet. See Table 4.

Following these initial findings with logistic regression, we decided to concentrate on the implementation of subsequent models to the features selected by the F-test. As a result, the entire pipeline—covering Random Forest, SVM, XGBoost — was used on features selected by the F-test. LogReg regression, first was used with elastic net penalty wit l1 ratio from 0 to 1. The analysis showed better performance with l1 ratio = 0 which corresponds to l2. Second, we used LogReg with l2. The search spaces for each algorithm are enlisted in the appendix.

Feature Selection Method	F1-Score	AUROC
TCRnet	0.8455	0.68
F-test	0.8903	0.87

Table 4. Quality metrics for logistic regression on different method feature selection.

3.4.2. Models used on embeddings

As described above, as training data we used embeddings from T1D batch 2 (only T1D repertoires) and "rosati" (health controls) batches. For testing we used T1D batch 1 (only diabetic repertoires) and "aging" (health controls) batches. Since embedding data is fairly high dimensional (1280 embedding dimensions + 1 entropy = 1281 features), it is crucial to apply dimensionality reduction technique on embeddings prior to classification. For this purpose, principal component analysis (PCA) was chosen as a linear method that projects the data onto principal components (PCs) that maximize the variance. The number of PCs was estimated in a supervised manner using 5-fold stratified cross-validation (CV). Alternatively, uniform manifold approximation and projection (UMAP) algorithm was utilized as a dimensionality reduction method that can capture complex non-linear manifolds where PCA fails. Likewise, UMAP's hyperparameters were tuned in a supervised manner through stratified 5-fold CV. For dimensionality reduction we utilized standardized embeddings. Next, for low-dimensional representations several classifiers were applied for predicting T1D status separately. We decided to use random forest and XGBoost. Additionally, we implemented a linear classifier, LogReg, with RBF sampler which is a scalable approximation of the RBF kernel that aids in learning non-linear decision boundaries without explicitly computing the kernel. RBFSampler+LogReg model was applied to raw embeddings. For parameter tuning we employed Bayesian optimization via `Optuna`. As the quality metric for optimization we used F1-score. The search spaces for each algorithm are enlisted in the appendix.

3.5. Deep learning models implementation

We decided to implement DeepRC model (Widrich et al., 2020) following (Rawat et al., 2024). When implementing the DeepRC model, we encountered significant challenges as the codebase had not been updated for several years. We

had to partially rewrite portions of the code to make it operational with current libraries and frameworks. Additionally, the data required preprocessing to conform to DeepRC's input requirements. More detailed descriptions of the input data specifications and storage structures are available in our GitHub repository.

4. Results

4.1. Models used on features selected by F-test

This section presents the results of several classification algorithms trained exclusively on the features selected by the F-test method. This choice is due to preliminary experiments (see previous chapters), where the F-test showed higher quality indicators compared to TCRnet. The performance quality metrics for the models are presented in Table 5.

Model	Test F1	AUROC
LogReg+ElasticNet	0.84	0.89
LogReg+L2	0.89	0.87
Random Forest	**0.91**	**0.91**
XGBoost	0.90	0.86
SVM	**0.90**	**0.92**

Table 5. Comparison of models (F-test features) on the test set: Test F1 and ROC AUC.

Figure 1 demonstrates ROC and PR curves to compare different implemented models.

Random Forest and SVM both show strong performance, but SVM achieved better results on the Precision-Recall curves as well, making it the overall best-performing model.

4.2. Models used on embeddings

Five different models were evaluated on testing data using various quality metrics. The performance quality metrics for the models are presented in Table 6. The best model, based on the aggregated quality score (mean of balanced accuracy, F1-score, and AUROC), was UMAP+XGBoost with 35 tuned UMAP components. However, all models trained on embedding data exhibited reduced performance compared to the classical approach.

Figure 2 demonstrates ROC and PR curves to compare different implemented models.

4.3. Deep learning models implementation

Unfortunately, the application of the DeepRC model to our dataset did not yield significant results. We observed a AUROC score of 0.5, which is equivalent to random prediction performance.

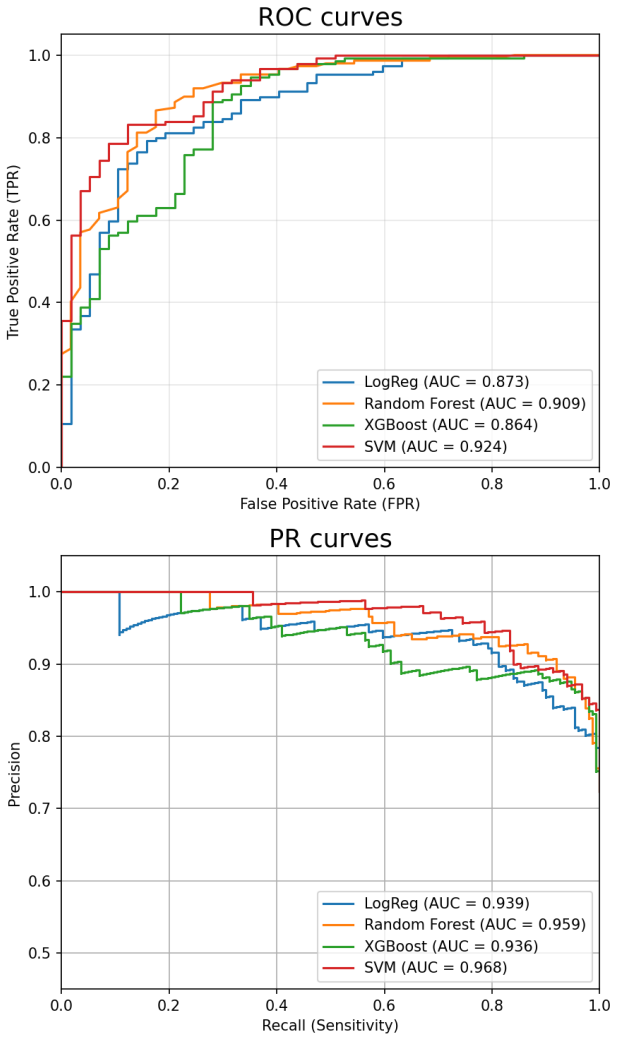

Figure 1. ROC and PR curves to compare performance of models trained and tested on features selected by F-test.

5. Discussions

In our study, we identified TCR sequences that appear enriched in patients with T1D compared to the healthy cohort. To strengthen the validity of our findings, we implemented a rigorous machine learning approach with a carefully designed train-test split strategy. This methodology provides greater confidence that observed performance will translate to practical applications where new patient samples would inherently differ from those used during model development. Importantly, our classifiers demonstrated high accuracy on the independent test dataset suggesting that our identified TCR signatures contain disease-associated signals that generalize across different patient cohorts.

Classical machine learning approaches outperformed more

Model	Bal Acc	F1-Score	AUROC
RBFSampler+LogReg	0.624	0.862	0.542
PCA+Random Forest	0.562	0.845	0.836
UMAP+Random Forest	0.516	0.837	0.809
PCA+XGBoost	0.588	0.852	0.782
UMAP+XGBoost	0.588	0.852	0.841

Table 6. Quality metrics for different models trained and tested on ESM-derived embeddings.

sophisticated methods in TCR repertoire-based classification of type one diabetes. Notably, traditional feature engineering combined with classical classifiers achieved superior results compared to the embedding-based approach. This performance discrepancy may be attributed to the unique characteristics of TCR sequences. The embedding algorithms we employed were primarily trained on conventional protein datasets, potentially limiting their ability to capture the distinctive structural and functional properties of TCR CDR3 regions. Probably, further studies on embeddings trained explicitly on immune receptor repertoires can increase the performance of TCR-repertoires-based classifiers.

Unfortunately, applying the DeepRC model to our dataset yielded disappointing results. This underperformance may stem from the model's ability to capture relevant immunological patterns in our specific context or insufficient training data for the deep learning architecture to identify meaningful signals. Additionally, the outdated codebase complicated the implementation for our data. Further investigation would be necessary to determine which of these factors most significantly impacted performance and whether modifications to the model architecture or data representation could improve results.

Moreover, our findings warrant careful interpretation. The observed TCR enrichment patterns may not necessarily reflect disease-specific immune responses, but could instead represent TCR repertoire differences stemming from genetic variation between the cohorts, particularly in HLA genes. HLA alleles significantly influence TCR selection during thymic development and subsequent peripheral expansion, potentially creating repertoire biases independent of disease status. Future studies would benefit from HLA-matched control groups or computational approaches that account for HLA-driven repertoire biases to more accurately identify TCRs genuinely associated with type one diabetes pathogenesis.

6. Conclusion

Our study successfully accomplished all planned objectives while extending beyond the initial scope to implement

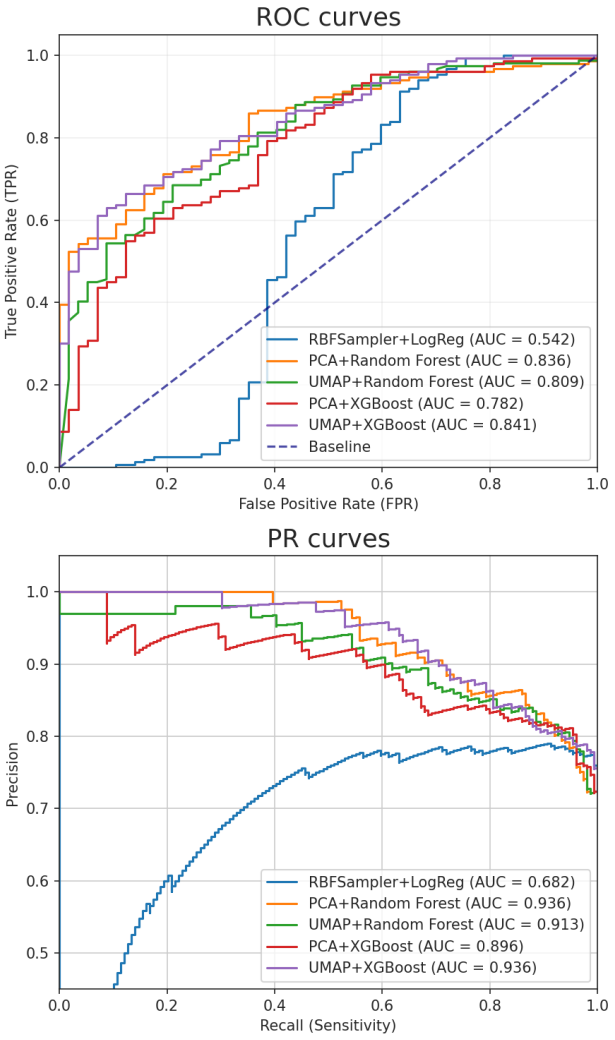

Figure 2. ROC and PR curves to compare performance of five different models trained and tested on ESM-derived embedding data.

embedding-based approach. We developed and validated a classification framework for distinguishing T1D patients from healthy individuals based on TCR repertoire signatures. Most of implemented classifiers demonstrated high accuracy on independent test datasets, with Random Forest and SVM models implemented on features selected by F-test exhibiting the strongest performance overall.

In summary, this work establishes a foundation for TCR-based biomarker development in T1D while providing methodological insights applicable to broader immune repertoire analysis. The identified TCR signatures and classification framework offer promising avenues for further investigation into disease mechanisms and potential diagnostic and target treatment applications.

7. Team member's contributions

Kseniia Lupyr: reviewing literate on the topic, data preprocessing, feature selection and feature table construction, implementing DeepRC, preparing the GitHub Repo, writing report.

Artemiy Dakhnovets: reviewing literature on the topic, building ESM-based embeddings, implementing ML models on embeddings, writing report.

Valeriia Vladyina: implementing ML models on feature tables constructed after feature selection with F-test and TCRnet, writing report.

References

De Neuter N, Bartholomeus E, E. G. K. N. S. A. J. H. e. a. Memory cd4 t cell receptor repertoire data mining as a tool for identifying cytomegalovirus serostatus. *Genes Immun.*, 2019.

Emerson RO, DeWitt WS, V. M. G. J. H. J. O. E. e. a. Immunosequencing identifies signatures of cytomegalovirus exposure history and hla-mediated effects on the t cell repertoire. *Nature Genetics*, 2017.

Herold KC, Delong T, P. A. B. N. B. T. W. L. The immunology of type 1 diabetes. *Nature Reviews Immunology*, 2024.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., and Shmueli, Y. e. a. Language models of protein sequences at the scale of evolution. *BioRxiv*, 2022.

Noble JA, V. A. Genetics of the hla region in the prediction of type 1 diabetes. *Curr Diab Rep.*, 2011.

Ostmeyer J, Christley S, T. I. C. L. Biophysicochemical motifs in t-cell receptor sequences distinguish repertoires from tumor-infiltrating lymphocyte and adjacent healthy tissue. *Cancer Res.*, 2019.

Pogorelyy, M. V. and Shugay, M. A framework for annotation of antigen specificities in high-throughput t-cell repertoire sequencing studies. *Frontiers in immunology*, 10:2159, 2019.

Pugliese A, Zeller M, F. A. J. Z. L. B. R. R. C. e. a. The insulin gene is transcribed in the human thymus and transcription levels correlated with allelic variation at the ins vntr-iddm2 susceptibility locus for type 1 diabetes. *Nat Genet.*, 1997.

Qi Q, Liu Y, C. Y. G. J. Z. D. L. J.-Y. e. a. Diversity and clonal selection in the human t-cell repertoire. *Proc Natl Acad Sci U S A.*, 2014.

Rawat, P., Shapiro, M. R., Peters, L. D., Widrich, M., Mayer-Blackwell, K., Motwani, K., Pavlović, M., al Hajj, G., Posgai, A. L., Kanduri, C., Isacchini, G., Chernigovskaya, M., Scheffer, L., Motwani, K., Balzano-Nogueira, L. O., Pettenger-Willey, C. M., Valkiers, S., Jacobsen, L., Haller, M. J., Schatz, D. A., Wasserfall, C. H., Emerson, R. O., Fiore-Gartland, A. J., Atkinson, M. A., Klambauer, G., Sandve, G. K., Greiff, V., and Brusko, T. M. Identification of a type 1 diabetes-associated t cell receptor repertoire signature from the human peripheral blood. *medRxiv*, 2024. doi: 10.1101/2024.12.10.24318751. URL https://www.medrxiv.org/content/early/2024/12/12/2024.12.10.24318751.

Redondo MJ, Steck AK, P. A. Genetics of type 1 diabetes. *Pediatr Diabetes*, 2018.

Robertson CC, Inshaw JRJ, O.-G. S. C. W.-M. S. C. D. Y. H. e. a. Fine-mapping, trans-ancestral and genomic analyses identify causal variants, cells, genes and drug targets for type 1 diabetes. *Nat Genet.*, 2021.

Sabater L, Ferrer-Francesch X, S. M. C. P.-J. M. P.-B. R. Insulin alleles and autoimmune regulator (aire) gene expression both influence insulin expression in the thymus. *J Autoimmun.*, 2005.

Sidhom, J.-W., Larman, H. B., Pardoll, D. M., and Baras, A. S. Deeptcr is a deep learning framework for revealing sequence concepts within t-cell repertoires. *Nature communications*, 12(1):1605, 2021.

Vafiadis P, Bennett ST, T. J. N. J. G. R. G. C.-e. a. Insulin expression in human thymus is modulated by ins vntr alleles at the iddm2 locus. *Nat Genet.*, 1997.

Vlasova EK, Nekrasova AI, K. A. I. M. S. E. M. S.-e. a. Robust detection of sars-cov-2 exposure in the population using t-cell repertoire profiling. *bioRxiv*, 2023. doi: 10.1101/2023.11.08.566227.

Widrich, M., Schäfl, B., Ramsauer, H., Pavlović, M., Gruber, L., Holzleitner, M., Brandstetter, J., Sandve, G. K., Greiff, V., Hochreiter, S., and Klambauer, G. Modern hopfield networks and attention for immune repertoire classification, 2020. URL https://arxiv.org/abs/2007.13505.

Zaslavsky, M. E., Craig, E., Michuda, J. K., Sehgal, N., Ram-Mohan, N., Lee, J.-Y., Nguyen, K. D., Hoh, R. A., Pham, T. D., Röltgen, K., et al. Disease diagnostics using machine learning of b cell and t cell receptor sequences. *Science*, 387(6736):eadp2407, 2025.

Appendix

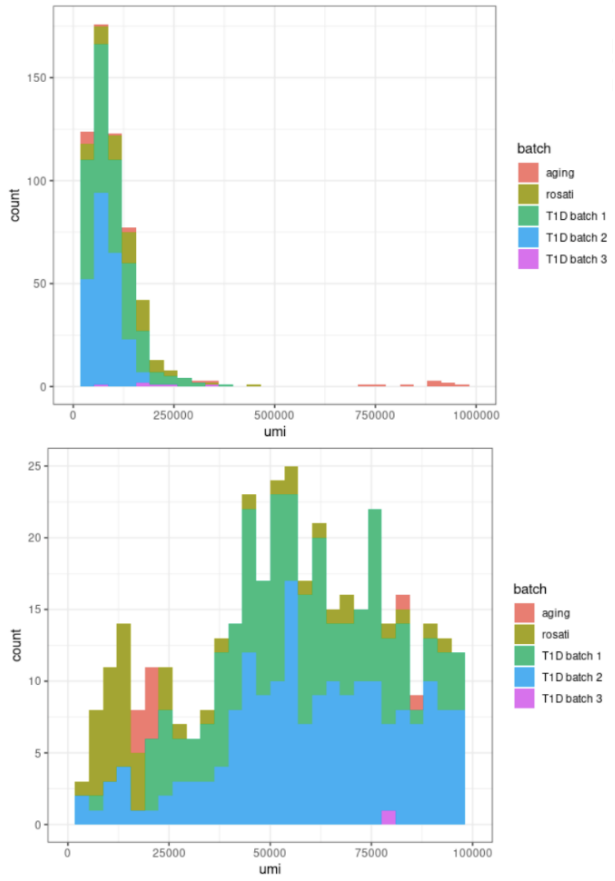

Figure 3. Number of UMIs distribution across batches

Search space for hyperparameters of models implemented on the feature tables

Logistic Regression (LogReg)

- `penalty`: Penalty type selected from {`l1`, `l2`, `elasticnet`, `none`}.
- `C`: Inverse regularization strength sampled logarithmically from 10^{-1} to 10^{1}.
- `solver`: Algorithm choice from {`newton-cg`, `lbfgs`, `liblinear`, `sag`, `saga`}.
- `max_iter`: Maximum number of iterations chosen from {100, 500, 1000, 5000}.

Support Vector Machine (SVM)

- `C`: Inverse regularization strength sampled logarithmically from 10^{-1} to 10^{1}.
- `kernel`: Kernel function choice from {`linear`, `rbf`, `poly`}.
- `gamma`: Kernel coefficient set to either {`scale`, `auto`}.
- `degree`: Polynomial degree for `kernel = poly`, typically {2, 3, 4}.

Random Forest

- `n_estimators`: Number of trees in the forest, typically {10, 50, 100, 200, 500, ...}.
- `max_depth`: Maximum depth of each tree from {`None`, 3, 5, 10, ...}.
- `min_samples_split`: Minimum samples per split, e.g., {2, 5, 10, ...}.
- `min_samples_leaf`: Minimum samples at a leaf, e.g., {1, 2, 4, ...}.
- `bootstrap`: Whether bootstrap samples are used, in {`True`, `False`}.
- `max_features`: Feature subset size from {`auto`, `sqrt`, `log2`}.

XGBoost

- `n_estimators`: Number of boosting rounds, typically {100, 200, 500, ...}.
- `max_depth`: Maximum depth of each tree from {3, 5, 7, 10, ...}.
- `learning_rate`: Shrinkage parameter chosen from {0.01, 0.05, 0.1, 0.2, ...}.
- `subsample`: Subsample ratio of the training instances {0.6, 0.8, 1.0}.
- `colsample_bytree`: Subsample ratio of columns {0.6, 0.8, 1.0}.
- `gamma`: Minimum loss reduction for further partition, e.g., {0, 1, 5}.
- `reg_alpha, reg_lambda`: L1 and L2 regularization parameters from {0, 0.1, 1, 10}.

Search space for hyperparameters of models implemented on the embedding data

UMAP:

- `n_components`: Integer range $\{5, 10, \ldots, 40\}$ (step size 5).
- `n_neighbors`: Integer range $\{5, 10, \ldots, 50\}$ (step size 5).
- `min_dist`: Continuous range $[0.01, 0.99]$ controlling cluster tightness.
- `metric`: Categorical choice between `euclidean` and `cosine` distance metrics.

RBFSampler:

- `gamma`: Kernel coefficient sampled logarithmically from 10^{-6} to 10^{3}.
- `n_components`: Number of components selected from $\{50, 100, \ldots, 500\}$ in steps of 50.

LogReg:

- `C`: Inverse regularization strength sampled logarithmically from 10^{-6} to 10^{3}.
- `solver`: Algorithm choice from {`lbfgs`, `sag`, `saga`}.
- `fit_intercept`: Boolean flag for intercept term {`True`, `False`}.
- `max_iter`: Maximum iterations selected from $\{1000, 1500, 2000\}$.
- `class_weight`: Class weighting option {`None`, `'balanced'`}.

Random forest:

- `max_depth`: Maximum tree depth explored from 3 to 15.
- `min_samples_split`: Minimum samples required to split a node, range 2–20.
- `min_samples_leaf`: Minimum samples required at leaf nodes, range 3–10.
- `criterion`: Splitting criterion selected from {`gini`, `entropy`}.
- `max_features`: Features considered at each split: {`sqrt`, `log2`, `None`}.
- `class_weight`: Custom class weights:
  - Class 0 (negative): Sampled from $[1.0, 10.0]$.
  - Class 1 (positive): Sampled from $[0.1, 1.0]$.

XGBoost:

- `scale_pos_weight`: Sampled uniformly from $[0.2, 0.7]$ to address class imbalance.
- `learning_rate`: Log-uniform distribution in $[0.01, 0.2]$.
- `max_depth`: Integer values from 3 to 10 (inclusive).
- `min_child_weight`: Integer values from 1 to 10 (inclusive).
- `subsample`: Fraction of training data sampled per tree in $[0.6, 1.0]$.
- `colsample_bytree`: Fraction of features sampled per tree in $[0.6, 1.0]$.
- `lambda` (L2 regularization): Log-uniform distribution in $[10^{-8}, 10.0]$.
- `gamma` (minimum loss reduction): Log-uniform distribution in $[10^{-8}, 1.0]$.