Predicting treatment of rheumatoid arthritis with LIVI

Prediction with a classifier on top of the LIVI model

Esther Wit [1]
Supervisor(s): Marcel Reinders [1], **Kirti Biharie** [1], **Inez den Hond** [1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Esther Wit
Final project course: CSE3000 Research Project
Thesis committee: Marcel Reinders, Kirti Biharie, Inez den Hond, Christoph Lofi

Abstract

Large datasets can today be created with single cell RNA-sequencing (scRNA-seq), allowing to measure the RNA expression per cell. In 2026, a new model Latent Interaction Variational Interference (LIVI), was proposed to analyze this data. LIVI is novel in capturing both cell and donor-specific variation into the latent space with a Variational auto-encoder (VAE). The research of Vagiaki et al. was primarily focused on discovering expression quantitative trait loci (eQTLs). It is interesting to see how well the cell- and donor latent space captures other characteristics, such as treatment succes/failure. This research investigates whether the latent spaces of the LIVI model capture major cell types, sub-cell types, and treatment response. A simple classifier (MLP/SVM/Random Forest) was added on top of the latent spaces to evaluate this. Major cell types are clearly distinguishable from the cell latent space C, sub-cell types with a relatively larger class size can be well distinguished in the cell latent space C, and treatment response is partially captured in the DxC space, but not fully separable by a simple classifier. SVM and MLP outperform the random forest in classifying the treatment response. These findings indicate that biologically and clinically relevant information is preserved within the LIVI latent representations.

1 Introduction

Over the last decade, it has become easier to analyze large datasets of DNA/RNA due to advancements in biochemistry. This opens up the possibility to get a better understanding of how genomic variations, such as single-nucleotide polymorphisms (SNP), affect disease phenotype. SNPs are reported to be associated with many diseases, including autoimmune diseases, but their specific role in the pathogenesis cannot be fully explained yet [1]. Research in this field can lead to new insights of how the genome affects diseases such as inflammatory bowel disease, cancer and Rheumatoid Arthritis (RA) [2]. RA is a chronic auto-immune disease causing inflammation in the body, often presenting around the joints. The disease is heterogeneous with different RA phenotypes. It is characterized by joint swelling, joint tenderness and destruction of synovial joints, resulting often in severe disability and premature mortality [3].

Large datasets can today be created with single cell RNA-sequencing (scRNA-seq), allowing RNA expression to be measured at the single-cell level, rather than being averaged over across many cells as in bulk sequencing. This allows us to investigate how DNA variants affect gene expression in a cell-specific manner and allows for classifying sub-cell types. Genetic variants that regulate gene expression are known as expression quantitative trait loci (eQTL), many of which remain undiscovered. Various types of models exist to analyze gene expression to discover eQTLs, disease-gene and disease-cell type associations, which can be used for gene mapping and precision medicine [4]. The basis of these models varies from linear regression models [5] [6], to zero-inflated negative binomial (ZINB) model [7], to linear mixed models and to probabilistic models. However these models have limited scalability to these large datasets and/or a lot of association testing needed, limiting the statistical power.

In 2026, a new model Latent Interaction Variational Interference (LIVI), was proposed by Vagiaki et al. [8]. LIVI is an variational auto-encoder (VAE) which models gene expression. A VAE consists of an encoder, latent space and a decoder. The encoder encodes the information into the latent space, reducing the dimensionality of the data. The decoder is able to reconstruct the information from the latent space. LIVI is novel in capturing both

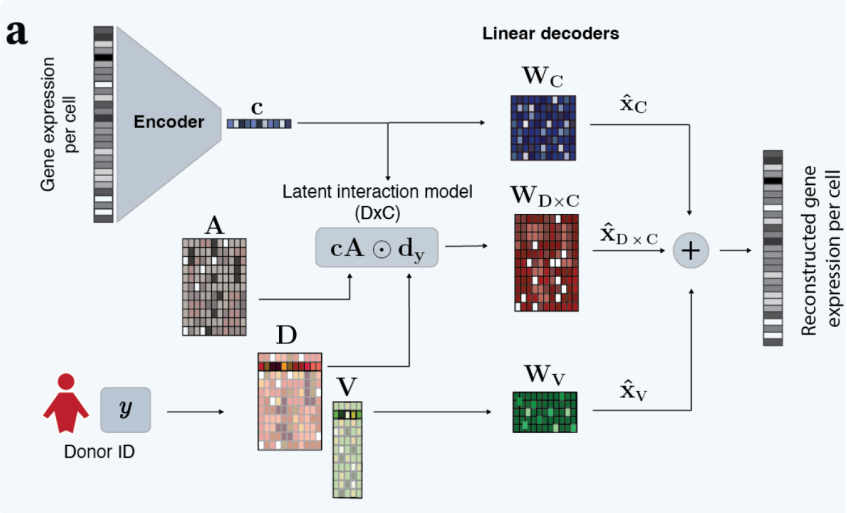


Figure 1: LIVI uses single-cell gene-expression measurements as input. This is encoded in C,D and V latent space and can be decoded using decoders W_C, W_{DXC} W_V. Figure from Vagiaki et al. [8]

cell and donor-specific variation into the latent space with a VAE, addressing limitations mentioned above. LIVI uses single-cell gene-expression measurements as input. This is encoded into latent spaces C, D and V (Figure 1); The first latent space C captures shared cell-state variation. 15 cell-factors are used in this latent space. Secondly, the donor embedding D captures cell-state-specific donor effects on gene expression. 700 donor factors are used in this latent space and are computed for each donor id. Third, the latent space V accounts for global sources of inter-individual variation. 5 global donor factors are used in this latent space. The $D \times C$ latent space (latent interaction model) maps the donor level of the respective patient to the cell level using assignment matrix A, resulting in a row for each cell with donor factor levels based on the donor id and cell expression. Using the decoders, reconstruction of the scRNA-seq data from the latent space is possible.

Originally, the LIVI model has been trained on the OneK1K cohort, which consists of scRNA-seq data from 1.27 million peripheral blood mononuclear cells (PMBCs) collected from 982 donors [9]. In our research, a patient cohort of 82 RA patients is analyzed using the LIVI model to discover more about the underlying biology of this disease heterogeneity [10]. The dataset consists of 85 synovial tissue samples. Samples were collected mainly from the knee or wrist joint and consisted of thousands of cells, which were analyzed using scRNA-seq. Patients exhibited moderate to high disease activity of RA or osteoarthritis (OA).

The research of Vagiaki et al. [8] was primarily focused on discovering eQTLs. It is therefore relevant to investigate whether the cell- and donor latent space captures other characteristics, such as treatment failure. Treatment failure is a clinically relevant indicator of therapeutic response and is categorized for this dataset into 4 categories based on clinical records: naive, methotrexate failure, TNF failure and OA control. The first three categories are sub-categories of RA, OA is used as control group. Treatment-naive patients are in

the early stage of their disease course. Methotrexate failure refers to patients who do not adequately respond to methotrexate (MTX), while TNF failure describes patients who do not respond to anti-TNF therapy, which is typically administered after methotrexate failure.

Previous studies have shown that latent representations learned by autoencoders can support downstream classification tasks, often through joint training of the encoder and classifier [11] [12]. In contrast, this project uses the pre-trained LIVI latent representations as fixed feature spaces and evaluates whether simple supervised classifiers can recover biological and clinical labels from them. Rather than developing an optimized prediction model, classifier performance is used to assess which information is preserved within the LIVI latent spaces. Based on the LIVI architecture, major cell-type and sub-cell-type information is expected to be primarily represented in the C latent space, whereas treatment-response categories are expected to be more strongly associated with the $D \times C$ latent space, which captures interactions between donor-specific and cell-specific factors.

The objective of this research is to discover if the latent spaces of the LIVI model capture the major cell type, sub-cell type and treatment (failure). To evaluate this, simple supervised classifiers are trained on the LIVI latent spaces and their performance is used as an indicator of the information encoded within each latent space.

2 Results

2.1 Literature review ML model for classifier

To be able to predict from the latent spaces created by the LIVI encoder, suitable classifiers were researched. Literature research in the ACM DL database [13] has been performed with the following search term: encoder AND ("classification" OR "prediction") AND "RNA" (n=383). Articles were scanned for title and abstract if they contained an auto-encoder with a ML model on top to classify. Eight articles have been analyzed, full analysis table can be found in Appendix A.

Multiple auto-encoder (AE) models with a softmax layer on top were found. Other variations were a CNN, MLP or k-means layer on top. In most of these models the auto-encoder and ML model were trained together. In our research the latent space has already been calculated and the classifier will be trained separately.

These studies show that latent representations from autoencoders can support downstream classification, but most models train the encoder and classifier jointly. In contrast, this project uses the already-trained LIVI representation as a fixed representation and evaluates whether simple classifiers can recover biological and clinical labels from its latent spaces. Therefore, the classifiers are not primarily used as optimized prediction tools, but as probes for the information encoded in LIVI.

The choice has been made to research the following ML models: Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Random Forest (Table 2 in Methods). The three classifiers were selected to probe different aspects of the latent space structure. An SVM is suitable for detecting if the signal is linearly separable signal in high-dimensional space [14]. An MLP was included to capture potential non-linear structure that a linear classifier

might miss [15]. Random Forest was included as a tree-based baseline to assess whether the signal is in a small number of latent factors, as it performs well when individual features carry strong discriminative information through axis-aligned splits [16]. Logistic regression, commonly used as a softmax output layer in related work, was not included here because it assumes linear separability and would largely overlap with the linear SVM in what it can capture, offering limited additional insight.

To verify these models against a baseline, a stratified random Dummy classifier was used. This classifier generates predictions according to the class distribution in the training data, meaning that classes with higher frequency are more likely to be predicted than rarer classes, which is suitable for imbalanced classes.

2.2 Patient characteristics

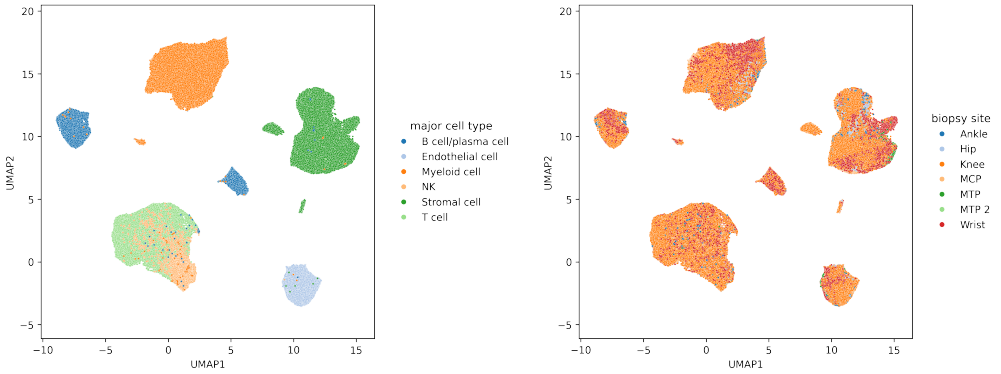
The dataset (table 1) consists of synovial tissue samples (n=85) of individuals with RA (n=72) or OA (n = 10). On 3 individuals repeat sampling has been performed. In total, scRNA-seq on these samples resulted in n = 314 011 cells. Age and sex are representative for disease population[3]. Treatment is classified into naive (n=28), methotrexate failure (n=28), TNF failure (n=16), OA control (n=10). The following classifiers were tested on top of the LIVI model: SVM, MLP and Random Forest. These were compared with a random stratified classifier as baseline.

Table 1: Population characteristics of the dataset of Zhang et al. [10]. Total samples is 85, of which 3 repeat samplings.

	***n* or Mean (SD)**
Diagnosis	
Rheumatoid arthritis (RA)	72
Osteoarthritis (OA)	10
Demographics	
Age (years)	57.9 (14.8)
Sex — men / women	22 / 60
Treatment group	
Naive	28
Methotrexate failure	28
TNF failure	16
Osteoarthritis control	10
Repeat	3
Clinical scores	
Krenn lining score	1.05 (0.50)
Krenn inflammation score	1.73 (0.84)
CDAI	33.58 (15.8)

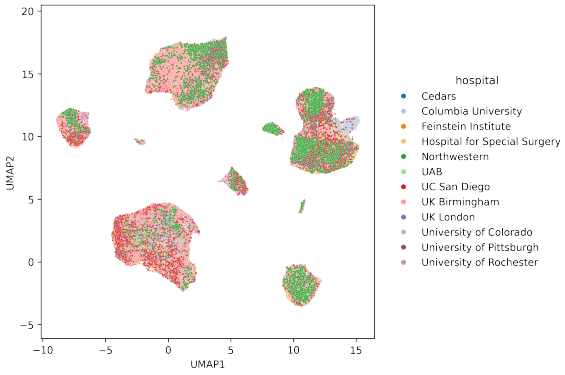
2.3 Biopsy site and hospital have no confounding effects on scRNA-seq

Biopsy site and hospital were identified as possible confounders as this might have influenced the cell sampling. Biopsy site and hospital show no effect when analyzed by cell-factors in a UMAP plot created with the scRNA-seq input data (Figure 2).



(a) UMAP colored by cell type.

(b) UMAP colored by biopsy site.

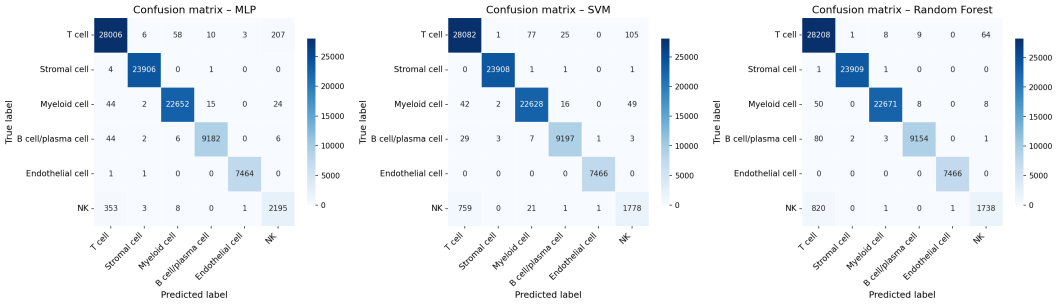


(c) UMAP colored by hospital.

Figure 2: UMAP projection based on scRNA-seq (input data). Cell-types (a) are clearly distinguishable in clusters. Biopsy (b) was mainly performed in the knee and wrist. Biopsy site (b) and hospital (c) show no confounding effect.

2.4 Major cell types can be distinguished from C latent space

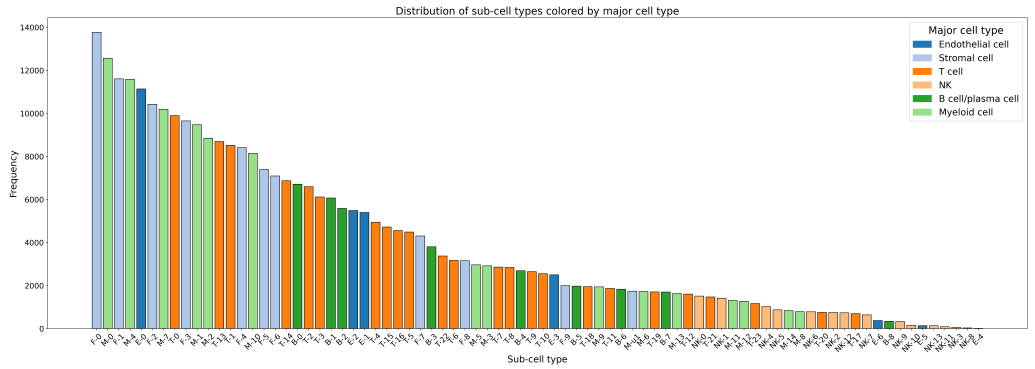
The 6 major cell types (B-cell, endothelial cell, myeloid cell, NK cell, stromal cell and T-cell) in the dataset are distinguishable based on the scRNAseq (input data) (Figure 2a). The major cell types are also distinguishable based on the C latent space (Figure 3), as classifiers trained on the C latent space are able to distinguish these cell-types; SVM, MLP and Random Forest (Table 2 in Methods) trained on the 15 cell factors of the C (cell) latent space, label almost all cells with their true label as can be seen in the confusion matrices (Figure 3). This confirms that the C latent space retains biologically meaningful cell-state information, supporting its use for more detailed sub-cell type classification.



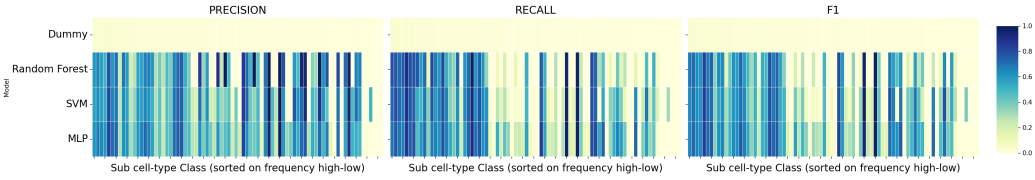
(a) CM of MLP (b) CM of SVM (c) CM of Random Forest

Figure 3: Confusion matrix (CM) with on y-axis true label and on x-axis predicted label. Diagonal shows correctly predicted samples. For each classifier, almost all samples are predicted correctly, on diagonal.

2.5 Prediction of sub-cell type is possible based on the C latent space



(a) Plot of cell type distribution, sorted on decreasing frequency



(b) Heatmap of sub-cell type classifier on the C latent space

Figure 4: Distribution of the cell types (a), colored by major cell type. On the x-axis the cell-type can be found and on the y-axis the frequency. Frequency varies from 13768 to 12. Heatmap of sub-cell type classifier on the C latent space with the metrics precision, recall and F1 score (b). Classes are sorted descending on frequency on the x-axis (order equal to Figure a). Classifiers (MLP, random forest and SVM) are on the y-axis. Larger classes (left) have better precision, recall and F1 scores. Smaller classes (right) have worse recall, indicating that these labels are predicted relatively less compared to larger classes.

The major cell types are easily distinguished, resulting in the question if the sub-cell types are also separable by a classifier on the C latent space. There is a large difference between sub-cell types in their frequency within the data (Figure 4a). It is easier to predict larger cell-types (Figure 4b) and smaller subtypes are predicted less (low recall), as expected by their frequency in the training set.

Confusion of cell-types is mainly seen within the overarching cell-type, for example B-0 and B-1 confusion. Major cell-types are thus still distinguished by the classifiers. Figure 5 gives the SVM classifier as example, other classifiers show a similar pattern (Appendix B).

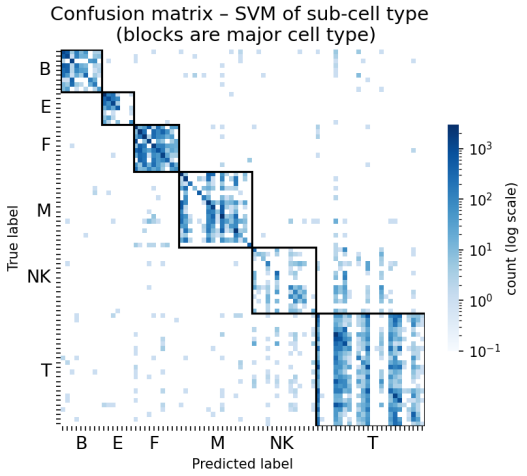


Figure 5: Confusion matrix of SVM. On y-axis true label and on x-axis predicted label of the sub-cell type, thus each small square is a sub-cell type prediction vs true. The log of the count is taken. Diagonal shows correctly predicted samples. It can be seen that confusion of sub-cell type is in the squares, within the major cell-type.

2.6 MLP and SVM classifier trained on DxC latent space outperform Random Forest in treatment prediction

Since the C latent space captures major cell types and partially sub-cell type structure, and treatment response is expected to depend on both donor- and cell-specific effects, the D×C latent space is researched if it captures the treatment success/failure. Classifiers were trained on the DxC latent space to predict the four categories of treatment (Figure 6) per cell. The D×C latent space (latent interaction model) maps the donor level of the respective patient to the cell level using assignment matrix A, resulting in a row for each cell with donor factor levels based on the donor id and cell expression.

Because treatment labels are donor-level labels, all cells from the same donor were kept in the same train (70%) or test (30%) fold. This prevents leakage of donor-specific information between train and test sets.

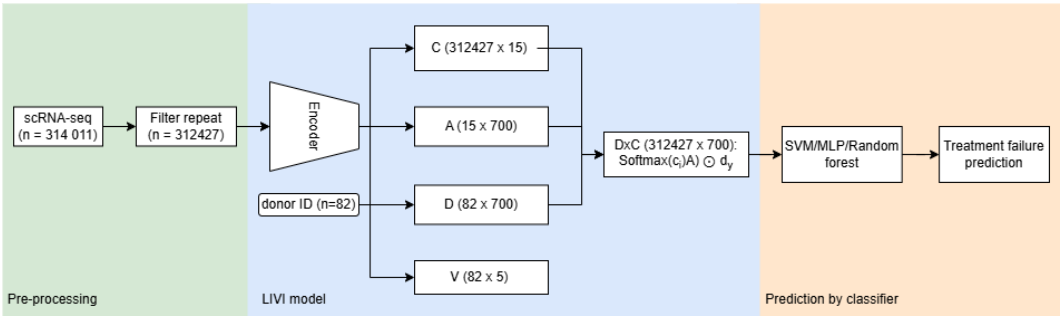


Figure 6: Outline of the study design, consisting of the following three compartments: preprocessing, LIVI model and prediction by classifier. The LIVI model has 3 latent spaces: the C cell space, the D donor space and the V global donor space. The cell and donor space are combined in the DxC space using the assignment matrix A (computation in Methods, 4.2).

To verify if the donor factors used to compute the DxC latent space show similar uniqueness [8] as in the ONEK1K dataset [9], pairwise Pearson correlation has been performed. The 700 donor factors from the D (donor) latent space are unique as they have a low Pearson correlation between factors (Figure 7), suggesting that the hyperparameter of 700 donor factors established by Vagiaki et al. [8] for the ONEK1K dataset is also suitable for this dataset.

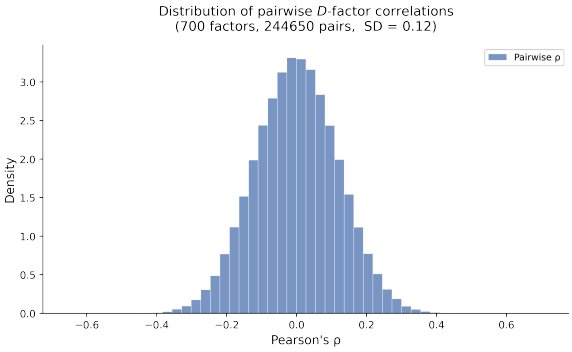


Figure 7: Distribution of pairwise D-factor correlation, showing little correlation between factors, indicating uniqueness of factors.

The classifier performance on the DxC space (Figure 8) shows high precision for OA. Osteoarthritis differs the most in its disease pattern and can therefore be clearly distinguished by the classifiers. The Naive class, which contains the largest number of training samples, also shows relatively high precision. Random Forest can clearly separate OA, but struggles to achieve high precision for the other classes. The MLP performs slightly worse than the SVM on TNF failure in terms of precision, although both classifiers perform close to the random baseline. As can be seen in the F1 heatmap, SVM and MLP achieve better overall performance across all classes compared to random forest. Confusion matrices can be found in Appendix D.

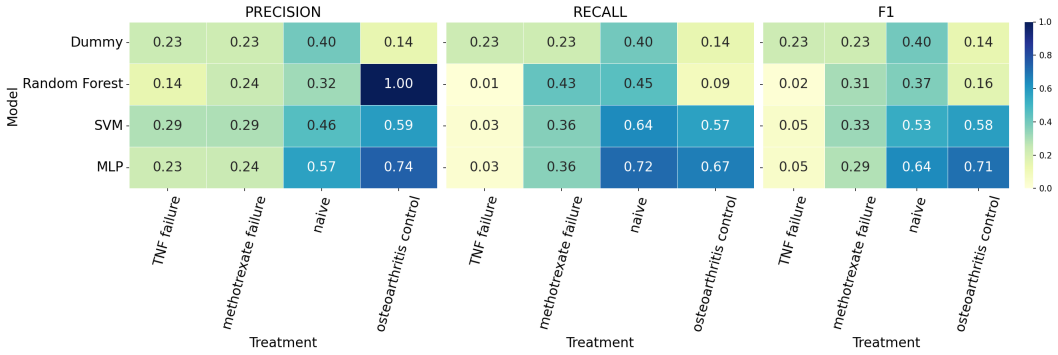
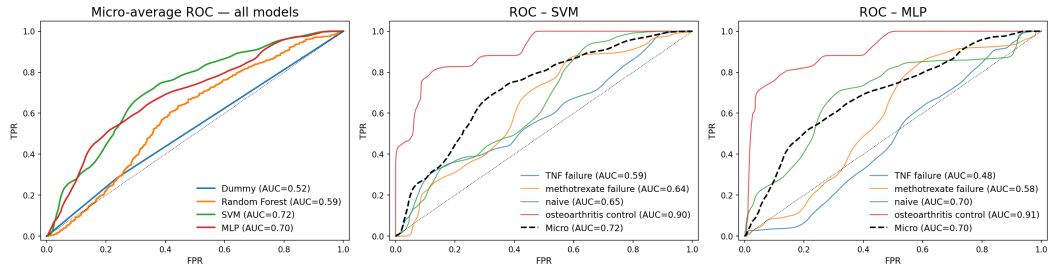


Figure 8: Heatmap of the precision, recall and F1 scores per classifier for each class on the test data. Precision is high for OA for all classifiers. Random Forest is outperformed by SVM and MLP on the other categories. Difference between SVM and MLP can be seen in methotrexate failure.



(a) ROC-AUC all classes (b) ROC-AUC for SVM (c) ROC-AUC for MLP

Figure 9: ROC-AUC curve of SVM, random forest, MLP and dummy classifier classifying treatment response based on the $D \times C$ latent space micro-averaged (a). ROC-AUC curve for SVM for all treatment categories (b) and MLP for all treatment categories (c). Random Forest (AUC=0.59) slightly outperforms the dummy classifier. SVM (AUC=0.72), MLP (AUC=0.70) difference can be explained by difference in performance on methotrexate failure.

Figure 9a shows the micro-average ROC-AUC curves for the classifiers and a 0.5 reference diagonal (black line) indicating chance-level performance. ROC-AUC quantifies class separability independent of the decision threshold, with 1 indicating perfect discrimination. The dummy classifier achieves an AUC of 0.52 and serves as a stratified random baseline. Random Forest (AUC = 0.59) slightly outperforms this baseline. The SVM achieves the highest performance (AUC = 0.72), which can be explained by its better discrimination of methotrexate failure compared to the MLP (AUC = 0.70), as shown in Figures 9b and 9c.

2.7 Training on single cell-type shows similar performance for MLP and worse performance for SVM compared to training on all cell-types

To see if training on a single cell-type from the DxC latent space would reduce the noise in the data and improve performance of the classifiers, classifiers were trained only on the T-cells (n=93864) from the DxC latent space (study design in Appendix E. In Figure 10 can be seen that SVM and MLP score high on precision of OA, on the other classes MLP outperforms SVM. In terms of recall, SVM performs well on naive, but underperforms the baseline on the other classes. MLP outperforms the baseline on methotrexate failure, naive and OA. Random Forest, does not classify TNF failure and OA control, as indicated by 0.0 scores. These two classes are the smallest in size. Confusion matrices can be found in Appendix F.

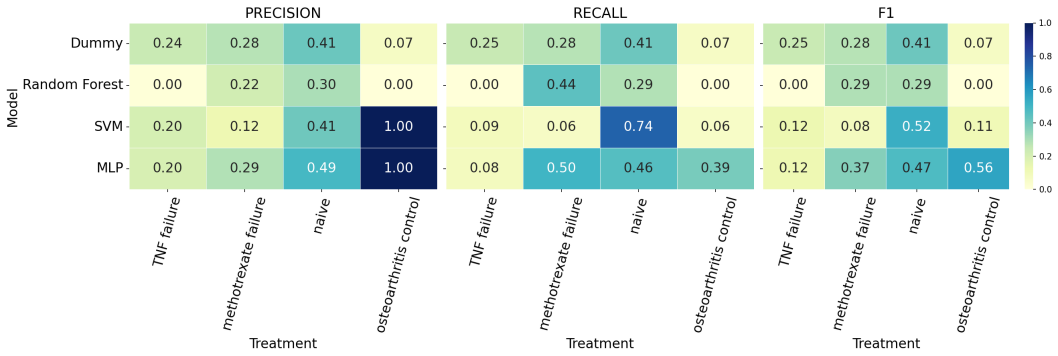
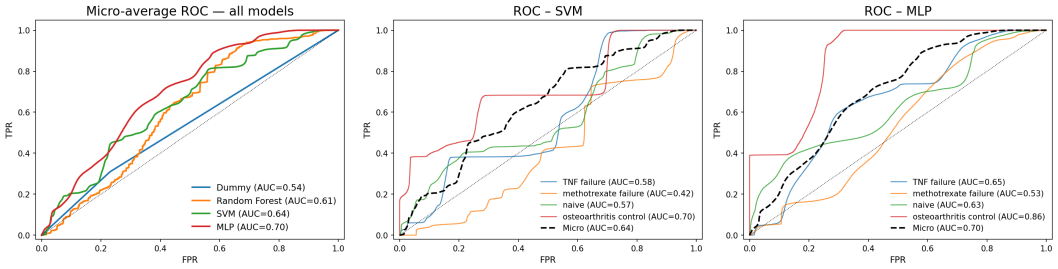


Figure 10: Heatmap of the precision, recall and F1 scores per classifier for each class based on only T-cells from the DxC space. Precision is high for OA for SVM and MLP. Random Forest fails to classify OA and TNF failure and is outperformed overall by SVM and MLP. MLP has a higher F1 score then SVM for all categories except naive.



(a) ROC micro-averaged. (b) ROC for SVM (c) ROC for MLP

Figure 11: ROC-AUC curve micro-averaged per classifier based on only T-cells from the DxC space (a). MLP is red, random forest orange, SVM green and Dummy classifier blue. MLP has the highest AUC (0.70). ROC-AUC curve for SVM (b) and MLP (c). Black link indicates micro-average. OA is red, methotrexate failure orange, naive green and TNF failure blue. The dotted line represents random.

In Figure 11a the micro-average ROC-AUC curve can be seen for the different classifiers. Random Forest (AUC=0.59) slightly outperforms the dummy classifier. MLP (AUC=0.70) performs better than SVM (AUC=0.64) this can be explained by better performance over all classes, see 11b & 11c. SVM underperforms when compared to trained on all classes (Figure 11b and 9b), MLP performs similair (Figure 11c and 9c). Random Forest also performs slightly worse (Figure 9a and 11a).

3 Discussion

This research investigated whether the latent spaces of the LIVI model capture major cell types, sub-cell types, and treatment response. To evaluate this, simple supervised classifiers were trained on the LIVI latent spaces.

The main findings are that major cell types are clearly distinguishable from the C latent space, sub-cell types are partially captured in the C space, and treatment response is partially captured in the DxC space, but not fully separable by a simple classifier. These findings indicate that biologically and clinically relevant information is preserved within the LIVI latent representations.

The six major cell types are clearly distinguishable from the C latent space. Sub-cell type classification performs somewhat worse, though higher-frequency sub-cell types are well distinguished. This is likely explained by greater similarity between sub-cell types and the small size of some classes. Importantly, misclassifications remain within the correct major cell type, confirming that broad cell class distinctions are well preserved.

For treatment prediction, all three classifiers outperform the stratified dummy baseline, indicating that the DxC space carries relevant information. SVM slightly outperforms MLP when trained on all cells, while MLP performs better on T-cells alone, suggesting that including all cell types adds noise for MLP whereas SVM benefits from the larger number of data points.

Random forest underperformance relative to SVM and MLP suggests that treatment signal is distributed across many latent factors rather than concentrated in a few. This makes it harder for tree-based models, which rely on sequential threshold splits of individual features, to capture the full signal effectively. In contrast, models such as SVM and MLP can combine information across many dimensions more smoothly, which may explain their better performance.

OA is most separable by the classifiers, as reflected by its highest AUC. This aligns with OA presenting distinctly from RA both clinically and in gene expression [17]. The other treatment categories are all subgroups of RA. It would therefore be interesting to investigate whether treatment classification within RA improves when OA samples are excluded from training.

Several limitations should be considered when interpreting these results. First, the treatment groups are imbalanced in terms of donors, which may contribute to lower performance for minority classes. Second, the number of cells per donor has a large variation, impacting both the latent space as well as the classifier. Third, treatment labels are assigned at donor level, while classification is performed on individual cells. Finally, although more than

300,000 cells were available, these originated from only 82 individuals, limiting the diversity of donor-level information available to the classifiers.

The number of individuals (n=82) in this research was limited. However, by assigning donor treatment labels to individual cells and training on the DxC latent space, over 300,000 samples were available for model training. It would be interesting to investigate whether a larger cohort improves treatment prediction performance. With more donors available, classification from the D (donor) latent space may also become feasible. This was not explored in the current study because 82 donors on approximately 700 latent factors would likely result in substantial overfitting. Furthermore, the C latent space does not appear to contain treatment-related information (Appendix C), suggesting that classifiers trained solely on C would not outperform the baseline.

In this research, cell data was grouped per donor for train/test split. Without having a single donor in either training or test set, the model was able to classify all samples correctly, indicating information leakage. It is important to take this into account for future research.

In future work, the classifiers could be further optimized. Gradient boosting would be an interesting alternative to random forest. Random forests combine independently trained trees, gradient boosting however iteratively optimizes new trees based on the errors of previous trees.

Additionally, it could be that 2 layers in the MLP fail to fully capture the complexity of the data. Further research could evaluate number of layers as a hyperparameter. This might result in MLP outperforming the SVM, however more layers make the classifier more prone to overfitting. Other options that could be researched are combining MLP and SVM into a single classifier, training the encoder with the loss of the classifier, or usage of a different encoder.

Although treatment-response information is present in the DxC latent space, the observed performance is not yet sufficient for clinical application. Rather, these results demonstrate that LIVI latent representations contain biologically meaningful information that can be recovered by relatively simple classifiers and may provide a useful basis for future research into treatment-response prediction.

4 Methods

4.1 Background on variational auto-encoder (VAE)

VAEs are a type of neural network and a generative model that extends traditional autoencoders by learning a probabilistic representation of data. This gives a smooth and continuous latent space, allowing for sampling in the latent space.

The VAE consists of three parts:

- **Encoder** maps with a probabilistic function the input data to the latent space using a Gaussian distribution.

- **Latent space** captures the important features of the input into a lower dimensionality.
- **Decoder** takes the latent space and reconstructs it into the original data space.

The marginal likelihood includes both the reconstruction error and Kullback-Leibler (KL) divergence. The reconstruction error ensures data consistency, how well the model can reconstruct the data. The Kullback-Leibler (KL) divergence ensures regularization, such that the latent space is a normally distributed space (Gaussian distribution).

4.2 Background on LIVI model

The $D \times C$ latent space (latent interaction model) maps the donor level to the cell level and is computed with the following formula:

$$\mathbf{z}_i^{D \times C} = (\text{Softmax}(\mathbf{c}_i)\mathbf{A}) \odot \mathbf{d}_y \tag{1}$$

In the equation above $\mathbf{z}_i^{D \times C}$, cell-state specific donor effects of single cell i, are computed by the Hadamard (element-wise) product $\odot$ of cell state specific donor factors $\mathbf{c}_i$ and cell-state-specific donor factors $\mathbf{d}_y$. Assignment matrix $\mathbf{A}$ is used for this mapping from $\mathbf{d}_y$ to $\mathbf{c}_i$.

Inference was performed using the code from the LIVI paper. LIVI was trained for this research with 15 C factors, 700 D factors and 5 V factors on 17,049 genes across 85 individuals. The encoder used hidden dimensions of 5000, 2000, 500 and 100. Training was performed with a learning rate of 8×10^{-4}. Covariates (pool, sex and cis-eQTLs) were not incorporated during training, in contrary to the original model.

4.3 Dataset & Preprocessing

Gene expression data were obtained from Zhang et al. [10]. For the treatment analysis, samples labeled "repeat" were excluded, as treatment conditions may have changed at the time of repeat sampling.

The dimensions of the latent spaces were C space 314 011 x 15 for cell major type and subtype. Dimension of DxC was 312427 x 700 for treatment categories after excluding repeat, and 93864 x 700 after excluding repeat and including T-cells.

Features were standardized using zero-mean unit-variance scaling (sklearn StandardScaler). For treatment categories, one-hot encoding was used.

4.4 Experimental Setup

Five-fold cross-validation was performed using donor-level splits to prevent data leakage across individuals, given the limited number of donors (n = 82). Each fold preserves all cells from a given donor in either the training or test set exclusively. For the cell-level, no folding was performed. The overall train/test ratio was approximately 70/30.
A fixed random seed of 42 was used for all train/test splitting to ensure reproducibility.

4.5 Classifiers

Classifiers (Table 2) were implemented in Python using PyTorch and Sklearn. For the random forest, maximum tree depth was set to 10. Deeper trees were also evaluated but yielded no performance improvement, indicating that depth 10 does not constitute underfitting. MLP was also tested with one layer, this yielded worse performance, thus was excluded.

Table 2: Classifier overview with implementation details and hyperparameters

Classifier / Model	**Key Characteristics**	**Hyperparameters Used**
Dummy Classifier	Random baseline; samples predictions according to training class distribution; used as baseline.	strategy = `"stratified"`
SVM	One-vs-Rest strategy; linear decision boundary; probability estimates via calibration (`cv=3`).	`LinearSVC(max_iter=2000)`
Random Forest	Ensemble of decision trees using bootstrap aggregation.	`max_depth=10`; `random_state=42`; `number_of_estimtors=100`
MLP	PyTorch nn.Module; 2-layer network: Linear $\rightarrow$ ReLU $\rightarrow$ Linear; trained with Adam optimizer and CrossEntropyLoss.	hidden size = 128; epochs = 50; lr = 0.01.

4.6 Software

All code was implemented in Python (3.11.9). Key libraries: Sklearn (1.8.0), PyTorch (2.5.1), Matplotlib (3.10.8), Seaborn (0.13.2). Experiments were run on the DAIC HPC cluster of TU Delft [18]. The full codebase is available at request.

4.7 Analysis classifier

The classifiers are analyzed by the following metrics:

- **Precision, Recall and F1 score:** Precision and recall are computed per class for multi-class ML. Precision looks into how many "positive" predictions were correct, Recall at looks at how many of the "positive" labels were predicted positive. F1 is a combination of these scores.
- **Confusion Matrix:** Each cell in the matrix shows how often a specific actual class was predicted as another class. Off-diagonal values represent misclassifications.
- **ROC-AUC curve:** To be able to represent multiple classes, micro-averaging was used. Micro-averaging takes the proportion of the classes into account. The ROC-AUC curve shows how well the classifier is able to separate the classes.

5 Responsible Research

5.1 Reproducibility

In terms of reproducibility, the LIVI model is open-source available. Code used in this project is also available at request. Furthermore, standardized classifiers from Python packages have been used to increase reproducibility. Test/train split and folding procedure are described.

5.2 Ethical concerns

An ethical concern is the storage and usage of patient data. Patient data has only been stored within the DAIC environment and not downloaded locally. Data has been used in line with the AMP declaration. The results of this research can not be used to identify single patients.

During training of the classifiers, time and resources of the DAIC were requested. These classifiers took about 2 hours to train on the DAIC environment. For further research, I would do even more debugging locally before running in the DAIC environment to prevent requesting space that is in the end not needed due to code failure. When training on even larger datasets, this is even more important to take into account. Besides space and resources on the DAIC, energy consumption of the DAIC servers also plays a role.

AI also impacts energy consumption by usage of data centers. It is important to take into account for each request whether this contributes to the research, or is unnecessary and therefore only a drain of resources/energy. During this research AI was used to assist plotting, assist debugging, polish text writing and to format Latex. In usage of AI, I have made sure that no patient data was leaked into the AI.

Considering a (hypothetical) future scenario where classifiers will be used to predict the best treatment options, bias of the classifiers should be taken into account. For example, in this research could be seen for sub-cell types that smaller types were predicted worse. If this would also be the case for predicting a larger broader range of treatment options, classifiers might ignore rare treatment types.

References

[1] M. Akhtar et al., "Characterization of rheumatoid arthritis risk-associated SNPs and identification of novel therapeutic sites using an in-silico approach", *Biology*, vol. 10, no. 6, p. 501, Jun. 4, 2021, ISSN: 2079-7737. DOI: 10.3390/biology10060501. Accessed: Jun. 15, 2026. [Online]. Available: https://www.mdpi.com/2079-7737/10/6/501.

[2] J. Huang et al., "Single-cell rna sequencing in autoimmune diseases: New insights and challenges", *Pharmacology Therapeutics*, vol. 267, p. 108 807, 2025, ISSN: 0163-7258. DOI: https://doi.org/10.1016/j.pharmthera.2025.108807. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0163725825000191.

[3] D. M. Mitchell, P. W. Spitz, D. Y. Young, D. A. Bloch, D. J. McShane, and J. F. Fries, "Survival, prognosis, and causes of death in rheumatoid arthritis", *Arthritis & Rheumatism*, vol. 29, no. 6, pp. 706–714, Jun. 1986, ISSN: 0004-3591, 1529-0131. DOI: 10.1002/art.1780290602. Accessed: Apr. 24, 2026. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/art.1780290602.

[4] Z. Jia et al., "eQTL analysis: A bridge from genome to mechanism", *Genes & Diseases*, vol. 13, no. 3, p. 101 850, May 2026, ISSN: 23523042. DOI: 10.1016/j.gendis.2025.101850. Accessed: Apr. 24, 2026. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352304225003393.

[5] A. A. Shabalin, "Matrix eQTL: Ultra fast eQTL analysis via large matrix operations", *Bioinformatics*, vol. 28, no. 10, pp. 1353–1358, May 15, 2012, ISSN: 1367-4811, 1367-4803. DOI: 10.1093/bioinformatics/bts163. Accessed: Apr. 24, 2026. [Online]. Available: https://academic.oup.com/bioinformatics/article/28/10/1353/213326.

[6] A. Fort et al., "*MBV* : A method to solve sample mislabeling and detect technical bias in large combined genotype and sequencing assay datasets", *Bioinformatics*, vol. 33, no. 12, O. Stegle, Ed., pp. 1895–1897, Jun. 15, 2017, ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btx074. Accessed: Apr. 24, 2026. [Online]. Available: https://academic.oup.com/bioinformatics/article/33/12/1895/2982050.

[7] T. Ma, H. Li, and X. Zhang, "Discovering single-cell eQTLs from scRNA-seq data only", *Gene*, vol. 829, p. 146 520, Jun. 2022, ISSN: 03781119. DOI: 10.1016/j.gene.2022.146520. Accessed: Apr. 24, 2026. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378111922003390.

[8] D. Vagiaki, T. Heinen, M. Saraswat, B. Clarke, and O. Stegle, *Mapping trans -eQTLs at single-cell resolution using latent interaction variational inference*, Feb. 6, 2026. DOI: 10.64898/2026.02.04.703363. Accessed: Apr. 21, 2026. [Online]. Available: http://biorxiv.org/lookup/doi/10.64898/2026.02.04.703363.

[9] S. Yazar et al., "Single-cell eQTL mapping identifies cell type–specific genetic control of autoimmune disease", *Science*, vol. 376, no. 6589, eabf3041, Apr. 8, 2022, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.abf3041. Accessed: Apr. 24, 2026. [Online]. Available: https://www.science.org/doi/10.1126/science.abf3041.

[10] F. Zhang et al., "Deconstruction of rheumatoid arthritis synovium defines inflammatory subtypes", *Nature*, vol. 623, no. 7987, pp. 616–624, Nov. 16, 2023, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06708-y. Accessed: Apr. 21, 2026. [Online]. Available: https://www.nature.com/articles/s41586-023-06708-y.

[11] B. Li, T. Wang, and S. Nabavi, "Cancer molecular subtype classification by graph convolutional networks on multi-omics data", in *Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, Gainesville Florida: ACM, Aug. 2021, pp. 1–9, ISBN: 978-1-4503-8450-6. DOI: 10.1145/3459930.3469542. Accessed: May 15, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3459930.3469542.

[12] T. Zhang, S. Zhao, and Z. Zhang, "Classification of liver cancer subtypes based on hierarchical integrated stacked autoencoder", in *2020 6th International Conference on Robotics and Artificial Intelligence*, Singapore Singapore: ACM, Nov. 20, 2020, pp. 79–83, ISBN: 978-1-4503-8859-7. DOI: 10.1145/3449301.3449316. Accessed: May 15, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3449301.3449316.

[13] A. for Computing Machinery. "Acml dl database", Accessed: Jun. 11, 2026. [Online]. Available: https://dl.acm.org/.

[14] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning", *Decision Analytics Journal*, vol. 3, p. 100 071, 2022, ISSN: 2772-6622. DOI: https://doi.org/10.1016/j.dajour.2022.100071. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772662222000261.

[15] A. Meyer-Baese and V. Schmid, "Chapter 7 - foundations of neural networks", in *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, Second Edition, Oxford: Academic Press, 2014, pp. 197–243, ISBN: 978-0-12-409545-8. DOI: https://doi.org/10.1016/B978-0-12-409545-8.00007-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124095458000078.

[16] L. Breiman, "Random forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 1, 2001, ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324.

[17] N. Zhu et al., "Identification of key genes in rheumatoid arthritis and osteoarthritis based on bioinformatics analysis", *Medicine*, vol. 97, no. 22, e10997, Jun. 2018, ISSN: 0025-7974. DOI: 10.1097/MD.0000000000010997. Accessed: Jun. 16, 2026. [Online]. Available: https://journals.lww.com/00005792-201806010-00086.

[18] T. DELFT. "Daic cluster", Accessed: Jun. 1, 2026. [Online]. Available: https://daic.tudelft.nl/docs/about/.

[19] C. Ji, N. Yu, Y. Wang, R. Qi, and C. Zheng, "An end-to-end deep hybrid autoencoder based method for single-cell RNA-seq data analysis", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 6, pp. 3889–3900, Nov. 2023, ISSN: 1545-5963, 1557-9964, 2374-0043. DOI: 10.1109/TCBB.2023.3328029. Accessed: May 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10299581/.

[20] N. N. Soylu and E. Sefer, "BERT2ome: Prediction of 2-o-methylation modifications from RNA sequence by transformer architecture based on BERT", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 3, pp. 2177–2189, May 1, 2023, ISSN: 1545-5963, 1557-9964, 2374-0043. DOI: 10.1109/TCBB.2023.3237769. Accessed: May 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10018863/.

[21] Z. Wang, "Enhancing breast cancer subtype classification via gene expression analysis and lightweight attention-based autoencoder", in *Proceedings of the 2025 5th International Conference on Computational Modeling, Simulation and Data Analysis*, Qingdao China: ACM, Dec. 14, 2025, pp. 1209–1213, ISBN: 979-8-4007-2000-0. DOI: 10.1145/3796731.3796913. Accessed: May 15, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3796731.3796913.

[22] C. Li, H. Wang, Y. Wen, R. Yin, X. Zeng, and K. Li, "GenoM7gnet: An efficient n^7 -methylguanosine site prediction approach based on a nucleotide language model", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 21, no. 6, pp. 2258–2268, Nov. 2024, ISSN: 1545-5963, 1557-9964, 2374-0043. DOI: 10.1109/TCBB.2024.3459870. Accessed: May 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10684781/.

[23] S. Guan, Q. Zou, H. Wu, and Y. Ding, "Protein-DNA binding residues prediction using a deep learning model with hierarchical feature extraction", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 5, pp. 2619–2628, Sep. 1, 2023, ISSN: 1545-5963, 1557-9964, 2374-0043. DOI: 10.1109/TCBB.2022.3190933. Accessed: May 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9829835/.

[24] G. Viaud, P. Mayilvahanan, and P.-H. Cournede, "Representation learning for the clustering of multi-omics data", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 1, pp. 135–145, Jan. 1, 2022, ISSN: 1545-5963, 1557-9964, 2374-0043. DOI: 10.1109/TCBB.2021.3060340. Accessed: May 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9357965/.

A ML model literature review

Table 3: Overview of literature review on combining VAE and ML model

Title	Authors	Input	Output	Model
Deep Hybrid Autoencoder for Single-Cell RNA-Seq [19]	Ji et al.	RNA-seq	Cell type	Autoencoder + MLP
BERT2OME: 2′-O-Methylation Prediction [20]	Soylu et al.	RNA sequence	Methylation label	BERT + 2D CNN
Cancer Subtype Classification on Multi-Omics Data [11]	Li et al.	9,759 genomic samples	33 cancer types	GCN + Softmax
Liver Cancer Subtype Classification via Stacked Autoencoder [12]	Zhang et al.	DNA/RNA/miRNA	Cancer subtypes	Hi-SAE + Softmax
Breast Cancer Subtype Classification via Attention Autoencoder [21]	Wang et al.	DNA/RNA/miRNA	Cancer subtypes	SAE + AE + Softmax
GenoM7GNet: N7-Methylguanosine Site Prediction[22]	Li et al.	RNA samples	m^7G site label	BERT + 1D CNN
Protein-DNA Binding Residues Prediction [23]	Guan et al.	DNA-binding proteins	Binding residues	Encoder + ConvGLU + Decoder
Representation Learning for Multi-Omics Clustering [24]	Viaud et al.	Multi-omics data	Clusters	SDAE/DDAE + k-means

B Confusion matrix sub-cell type

Confusion of cell-types is mainly seen within the overarching cell-type, for example B-0 and B-1 confusion. Major cell-types are thus still distinguished by the classifiers. All classifiers show a similar pattern (Figure 12, 13 and 14).

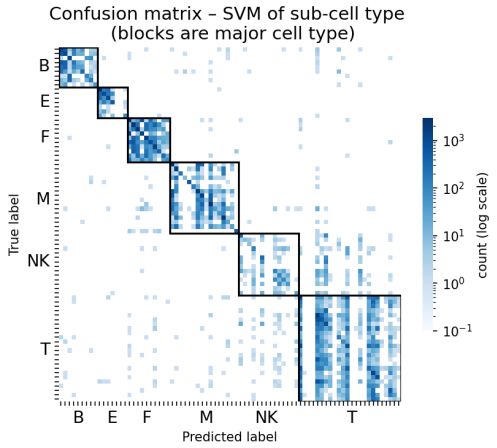


Figure 12: Confusion matrix of SVM. On y-axis true label and on x-axis predicted label. Color is based on the number of samples. Diagonal shows correctly predicted samples.It can be seen that confusion of sub-cell type is in squares, within the major cell-type.

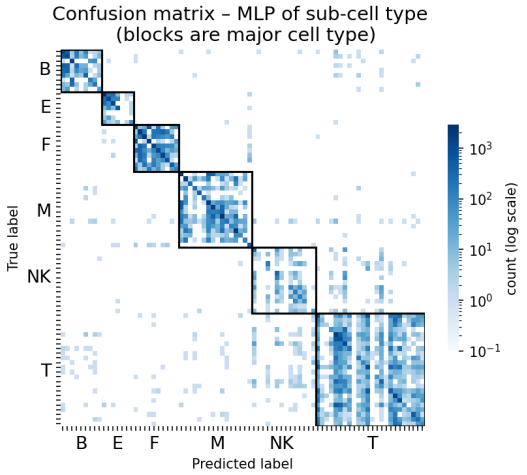


Figure 13: Confusion matrix of MLP. On y-axis true label and on x-axis predicted label. Color is based on the number of samples. Diagonal shows correctly predicted samples.It can be seen that confusion of sub-cell type is in squares, within the major cell-type.

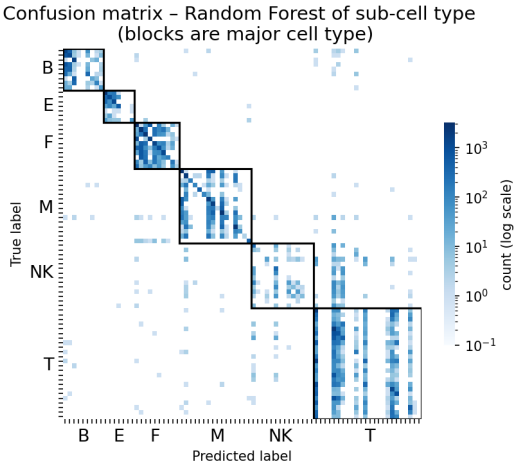
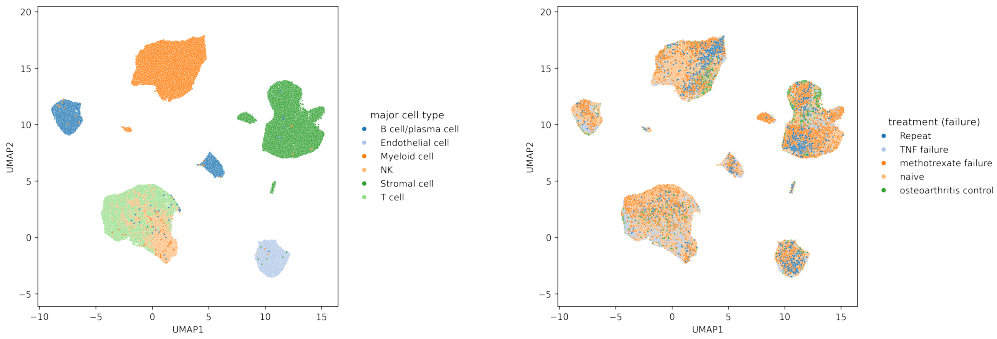


Figure 14: Confusion matrix of Random Forest. On y-axis true label and on x-axis predicted label. Color is based on the number of samples. Diagonal shows correctly predicted samples.It can be seen that confusion of sub-cell type is in squares, within the major cell-type.

C Treatment labels are not visibly separable in the UMAP of the C space

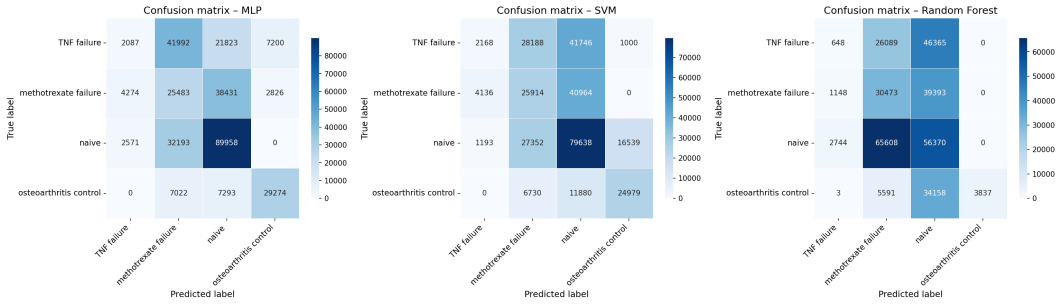


(a) UMAP colored by cell-type

(b) UMAP colored by treatment.

Figure 15: UMAP projection based on scRNA-seq factors. Cell-types (a) are clearly distinguishable and treatment (b) shows no effect.

D Confusion matrices per classifier for each class based on the DxC space



(a) CM of MLP

(b) CM of SVM

(c) CM of Random Forest

Figure 16: Confusion matrix (CM) with on y-axis true label and on x-axis predicted label. Diagonal shows correctly predicted samples. For MLP (a) and SVM (b) the majority of the samples within the naive and OA class is predicted correctly. As for methotrexate failure and TNF failure, confusion is mainly seen with naive. For Random forest (c) the same pattern applies, however OA is mainly predicted incorrectly, often confused with naive.

E Study design of prediction of treatment on only T-cells from the DxC space

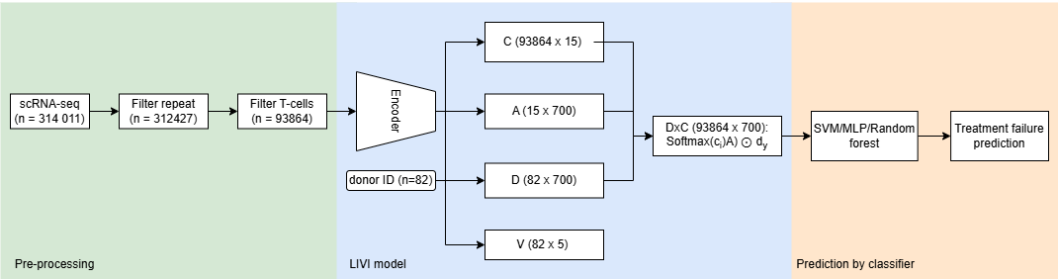
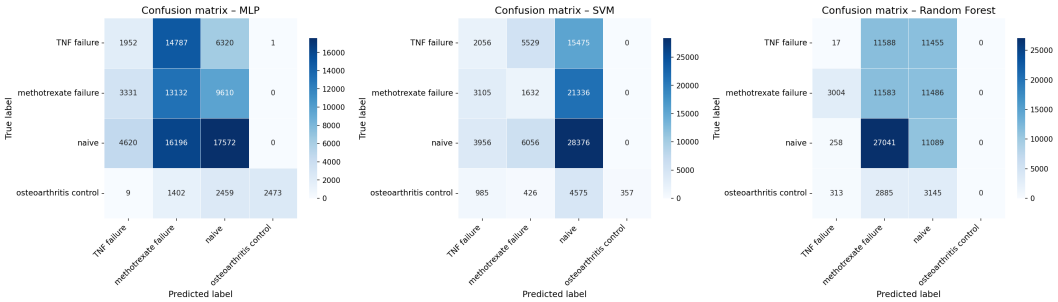


Figure 17: Outline of the study design of prediction of treatment on only T-cells from the DxC space, consisting of the following three compartments: pre-processing, LIVI model and prediction by classifier. The LIVI model has 3 latent spaces: the C cell space, the D donor space and the V global donor space. The cell and donor space are combined in the DxC space using the assignment matrix A (computation in Methods, 4.2).

F Confusion matrices per classifier for each class based on only T-cells from the DxC space



(a) CM of MLP (b) CM of SVM (c) CM of Random Forest

Figure 18: Confusion matrix (CM) per classifier for each class based on only T-cells from the DxC space with on y-axis true label and on x-axis predicted label. Diagonal shows correctly predicted samples. MLP (a) shows a similar pattern as for all cells (Figure 16a). SVM (b) performs worse compared to all cells, especially for OA (Figure 16b). Random forest(c) does not predict OA anymore (compared to Figure 16c).