Final Degree Project

Optimization of AI algorithms for predicting immunogenicity.

Author: Carlos Celaya Iturralde

Supervisor: Víctor Javier Sánchez-Arévalo Lobo

Institution: Hospital 12 de Octubre

Tutor: Arturo Vera García Francisco

de Vitoria University

Table of Contents

Summary

The immunogenic potential of tumor-specific neoantigens represents a promising frontier for personalized cancer immunotherapy, especially in difficult-to-treat neoplasms such as pancreatic ductal adenocarcinoma (PDAC). This study focuses on optimizing computational workflows and machine learning (ML) algorithms to improve the prediction of neoantigen immunogenicity, using large-scale immunopeptidomic data and multi-cohort integration. Building on the NeoRanking model, the methodology incorporates advanced variable selection techniques, data preprocessing, metric correction, and evaluation of a wide range of ML and deep learning models, including Gradient Boosting, AdaBoost, XGBoost, TabPFN, and graph neural networks. Key innovations include the use of the NDCG metric to evaluate ranking performance, strategic subsampling of negative examples, and the integration of an immunogenicity quality parameter for therapeutic purposes. The results show an improvement in predictive performance across various datasets, although differences in distribution between cohorts limit generalization. Application to PDAC data allowed the identification of a subset of high-confidence neoantigens, some of which were recurrent across patients and associated with non-classical genes such as MUC6, suggesting a potential use in shared targeted immunotherapies. Despite challenges such as extreme class imbalance and computational limitations, the proposed approach provides a solid foundation for the in silico prioritization of therapeutically relevant neoantigens.

Introduction

Pancreatic Ductal Adenocarcinoma: Origin, Characteristics, Diagnosis, and Treatment

Pancreatic ductal adenocarcinoma (PDAC) originates from ductal cells in the pancreas. This type of cancer develops through a multi-stage carcinogenesis process, beginning with low-grade dysplastic lesions that progress to carcinoma in situ and eventually to metastatic disease. This process involves the gradual acquisition of mutations in oncogenes and tumor suppressor genes, as well as changes in the pancreatic microenvironment, which shifts from a proinflammatory state to a highly fibrous and immunosuppressive desmoplastic tumor environment ([1] [2]).

PDAC is the most common type of pancreatic cancer, accounting for between 85% and 95% of all solid pancreatic tumors. It is one of the deadliest forms of cancer, ranking as the seventh leading cause of cancer death worldwide, responsible for more than 300,000 deaths annually [3,4]. The five-year survival rate is extremely low, ranging from 3% to 7.2%, due to late diagnosis and resistance to current therapies ([4] [5]).

Early diagnosis of PDAC remains a clinical challenge, as many cases are detected at advanced stages. Imaging techniques such as computed tomography and magnetic resonance imaging are essential for tumor detection and staging, but they have limitations in the early stages due to the small size or location of the tumor [6,7]. In response to these limitations, complementary molecular approaches such as proteomics and multi-omic technologies are emerging, which allow the detection of alterations in protein expression and specific mutations associated with PDAC through biomarkers ([8]).

The treatment of PDAC remains challenging due to its resistance to chemotherapy and the complexity of its tumor microenvironment. Current treatments include surgery, chemotherapy, radiation therapy, and immunotherapy, although most are palliative and aim to relieve symptoms and prolong survival [9]. Precision therapy, which uses genetic information to guide treatment, is emerging as a promising new direction, although its efficacy has been limited due to tumor heterogeneity ([10]). In addition, combination therapies including immunotherapy are being investigated to enhance the immune response and overcome barriers in the tumor microenvironment ([11]).

The solution of neoantigens

Given the limitations of conventional cancer therapies, especially in tumors such as PDAC, there is a need for more specific and effective approaches. In this context, one of the most promising strategies is personalized immunotherapy based on neoantigens, peptides derived from somatic mutations unique to the tumor, absent in healthy tissues, and capable of being recognized as foreign by the immune system ([12]) ([13]) .

The therapeutic value of neoantigens lies precisely in their exclusively tumor origin [14]. These molecules arise as a result of acquired genetic alterations, which generate aberrant proteins that can be identified by the immune system as foreign [15]. Among the different sources of neoantigens, single nucleotide variants (SNVs) stand out due to their high prevalence and clinical relevance ([16]) ,constituting the most common type of somatic mutation in various types of cancer ([17]) .

These SNVs can induce changes in the amino acid sequence of tumor proteins, generating peptides with altered sequences. These peptides are processed in the cell cytosol by the proteasome, transported to the endoplasmic reticulum by TAP transporters, and finally loaded onto molecules of the major histocompatibility complex (MHC class I, known as HLA in humans) ([18]).Once on the cell surface, the peptide-MHC complex can be recognized by the T cell receptor (TCR) of CD8+ T lymphocytes, which activates the adaptive immune response ([19]). This process is shown in Figure 1.

In this study, a mutation is defined as a sequence of 25 amino acids with an SNV located in the central position. From these sequences, neoepitopes or neoantigens are generated, which are short peptides of between 8 and 12 amino acids that include the mutation [20].

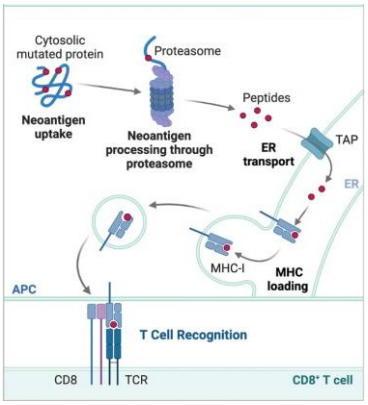

Figure 1. Processing of neoantigens to trigger the immune response.

The immunogenicity of these peptides, i.e., their ability to induce an effective immune response, can be validated by in vitro immunological assays. Among the most commonly used are

These include ELISpot and the use of HLA multimers, tools that enable functional evaluation of whether a peptide is recognized by the patient's T lymphocytes [2122] .

Biomedical applications for neoantigens

The therapeutic use of neoantigens has led to various innovative clinical strategies, particularly in the field of personalized immunotherapy:

-Personalized neoantigen vaccines: After computational identification of between 10 and 30 neoepitopes with high immunogenicity scores, these are synthesized as messenger RNA or long peptides. A notable example is the clinical trial conducted by MSK, where autogene cevumeran, an RNA-lipoplex vaccine containing up to 20 neoantigens selected per patient after surgical resection of PDAC, was administered. This study showed specific CD8+ T cell responses to several of the neoantigens in 50% of patients (8 out of 16) ([23]). In another phase I clinical trial using neoantigenic peptides with an adjuvant, a three-year disease-free survival rate of 56% was observed in resected patients, compared with an approximate historical rate of 20% ([24]) .

-Adoptive TCR cell therapy: Another avenue being explored involves isolating or modifying T lymphocytes so that they express TCRs specific to key tumor neoantigens. For example, a case has been reported of a patient with pancreatic metastases treated with T cells transduced with a TCR targeting the KRAS G12D mutation restricted to HLA-C*08:02. The therapy, combined with IL-2, achieved a 72% reduction in metastasis size ([25]). This case represents one of the first examples of success in TCR therapy against a driver neoantigen in PDAC. In scenarios where algorithms identify shared mutations, such as in KRAS or TP53, it is possible to develop preconfigured TCR receptors for groups of patients with those same mutations ([26]).

-Biomarkers: For a biomarker to have widespread clinical application, it must be present in a significant fraction of the affected population. Although there is no strict threshold, it is generally considered clinically useful if it appears in at least 10-20% of patients. In the case of PDAC, most of the identified neoantigens are highly patient-specific, limiting their usefulness as broad biomarkers. However, some recurrent mutations such as those in KRAS or TP53 could reach frequencies high enough to be considered shared biomarkers ([27]).

State of the art in AI

Given the difficulty of identifying neoantigens with true immunogenic potential [28], the combination of bioinformatics approaches and artificial intelligence has made it possible to overcome limitations.

inherent in traditional methods [29] , such as the high rate of false positives in binding affinity predictions [30] . By integrating structural, gene expression, and sequence data, these technologies offer a more holistic and accurate approach to neoantigen identification ([31] ,which is key to the development of personalized immunotherapies by facilitating the selection of candidates with a high probability of inducing effective immune responses and improving clinical outcomes in cancer patients ([32] .

The analysis starts from the same basic data, regardless of the approach used: a data frame that includes neopeptide sequences and their associated mutations, the HLA alleles of each patient (or those with the highest affinity for each peptide), and the immunogenicity annotation (positive or negative). Considering all possible combinations of amino acids in a neopeptide, the total number can reach $4.34 \cdot 10^{15}$ (see Appendix 1), which represents a significant computational challenge, especially when incorporating haplotype variability and the need for experimental validation.

To address this problem, it is common to process the data in tabular format, using characteristics generated by bioinformatics tools that predict different stages of the immune process (Figure 1). Each characteristic is assigned a numerical value representing the probability of that event occurring, thus facilitating the final prediction.

In this context, machine learning has been widely used [33]. A notable reference is the NeoRanking study ([20] ,which reprocessed whole exome sequencing (WES) and transcriptome (RNA-seq) data from 120 cancer patients from two large-scale immune screening trials (TESLA and HiTIDE) and an internal dataset of 11 patients. From 46,017 somatic SNV mutations, 1,781,445 neopeptides were generated, of which 212 mutations and 178 peptides were immunogenic, as shown in Table 1.

Table 1. Initial distribution of NeoRanking data. The datasets are the different cohorts available to the model, which will be developed further below.

Data set	Immunogenic	Non-immunogenic	Not evaluated
NCI mutations	146	11,651	24,899
NCI neo-peptides	103	418,872	953,486

Table 1. Initial distribution of NeoRanking data. The datasets are the different cohorts available to the model, which will be developed further below.

Data set	Immunogenic	Non-immunogenic	Not evaluated
TESLA mutations	36	461	6,231
TESLA neo-peptides	34	702	300,505
HiTIDE mutations	30	751	1,812
HiTIDE neo-peptides	41	1,511	106,191

The classifiers developed by Müller et al. (LR and XGBoost) accurately predicted the immunogenicity of mutations and peptides in different datasets, improving performance by up to 30% compared to previous methods. In addition, they evaluated the importance of features using Shapley values, identifying those derived from tools such as MixMHCpred, NetMHCpan, and PRIME as the most relevant, followed by stability range and gene expression from TCGA (Figure 2).

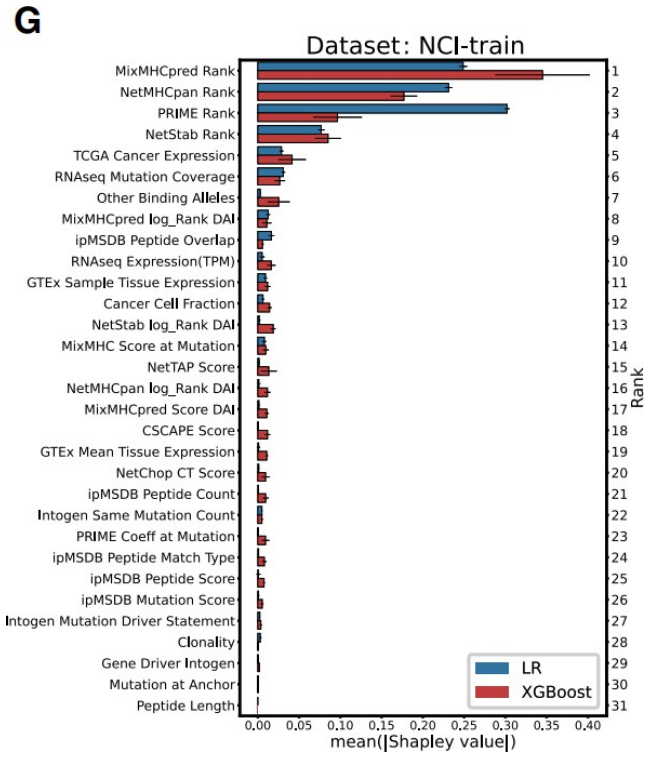

Figure 1. Shapley values extracted from NeoRanking with the NCI dataset of the model, showing the importance of each feature for predicting neopeptides in each LR and XGB model.

However, the study also revealed that model performance varies depending on the dataset used: classifiers trained on NCI showed good generalization but lower metrics when applied to TESLA and HiTIDE (Figure 3) [20].

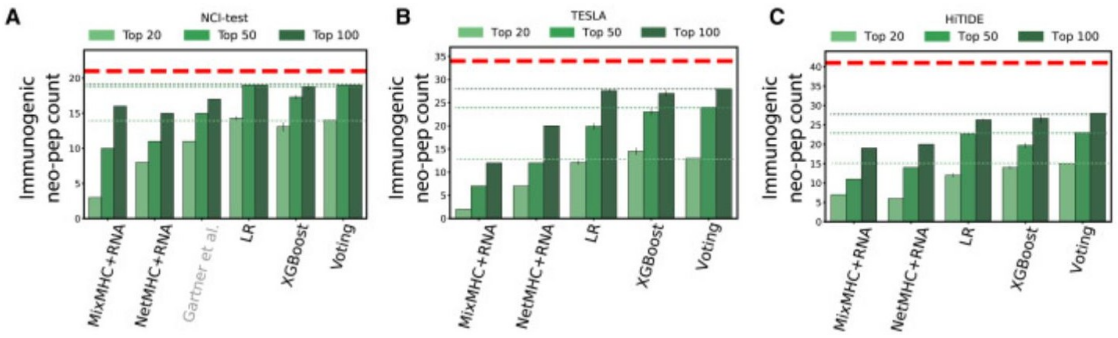

Figure 3. Initial performance of the NeoRanking model with top k metrics on different datasets and evaluating the top k metric.

An alternative approach was proposed by Neodb, who introduced the Immuno-GNN model [34], the first based on graph neural networks (GNNs) for this problem. Instead of treating sequences as linear, they represented each peptide and HLA as a fully connected graph, where each node corresponds to an amino acid. These were transformed into

vectors of 20 physicochemical features using an encoder. Immuno-GNN demonstrated superior performance in sensitivity, F1 score, and TopK metric compared to previous methods, demonstrating its greater discriminatory power [34].

Another approach is offered by NeoAPred [35], which explicitly incorporates the spatial dependence between neoantigens and HLAs, recognizing that both adopt three-dimensional structures. This deep neural network model was trained with experimental immunopeptide data, integrating physicochemical properties (hydrophobicity, surface accessibility), immunogenic motifs, and three-dimensional structural information of the peptide-MHC complex. NeoAPred demonstrated superior performance to previous methods, achieving an AUROC of 0.81 and an AUPRC of 0.54 on the evaluation set. This approach allows for more accurate modeling of the interaction with T receptors, and its effectiveness is due to the integration of structural features, beyond the sequence, which provide a more complete view of the immunogenic potential of each peptide ([35]).

Initial work

Javier Pérez Cardoso's Master's Thesis ("Development of a pipeline for the discovery of in silico biomarkers for PDAC") consisted of the automated generation of tumor neoantigens from VCF files of 180 patients from the PACA-CA and APIG-AU cohorts of the ICGC. The workflow, developed in Bash and Python, applied tools such as VEP to filter and annotate mutations, generating peptide sequences and calculating characteristics according to the NeoRanking study, with the exception of five due to computational limitations (described in Materials and Methods). Due to this computational cost (two months of processing), the analysis was limited to 62 pancreatic cancer patients. HLA typing was also performed from CRAM files, and a modified version of NeoRanking was trained with the available features to predict immunogenicity.

For his part, Luis Miguel Víboras Caballero's Master's Thesis ("Bioinformatic Evaluation of Neoantigen Quality and its Relationship with Immune Editing in Pancreatic Ductal Adenocarcinoma") developed and applied two workflows to evaluate the immunogenic quality of neoantigens and their relationship with the immune editing process, comparing patients with different survival times. Based on the data generated by Pérez Cardoso, he calculated two key metrics: amplitude (A), which measures the difference in presentation between mutated and wild-type peptides, and recognition as non-self (R), evaluated by similarity to known epitopes via BLASTp. These metrics were used to rank immunogenic quality, observing that patients with shorter survival times had a higher burden of high-quality immunogenic mutations, suggesting an active process of immune selection and escape.

tumor. In contrast, patients with long survival showed less recognizable mutations, which could indicate more effective elimination of immunogenic clones in early stages.

Objectives

The main objective of this thesis is to improve the predictive capacity of neoantigen immunogenicity, with a particular focus on therapeutic applicability. To achieve this, the following specific objectives are proposed:

- Improve the metrics in the NeoRanking model, addressing the inconsistencies observed in the evaluation of the Top-k metric.
- Explore and test other Machine Learning and Deep Learning algorithms beyond those used in NeoRanking, with the hypothesis that a broader set of algorithms and a more sophisticated prediction combination process can generate a more robust and accurate model.
- Design an initial mutation filter to process only the most promising neopeptides, reducing the computational load.
- Implement the quality parameter obtained by Luis Miguel to make predictions with better therapeutic application.
- Extract neoantigens with potential for immunotherapy and vaccines from pancreatic cancer data obtained by Javier.

Materials and methods

Biostatistical analysis

The work is based on the same initial data and algorithms used in the NeoRanking model [20]described above. All code was written in a Linux environment in Python 3.8.10 and bash 5.2.21 and can be found on GitHub: https://github.com/CarlosCelayaItu/TFG_CARLOS_CELAYA. Access is private, so to access it, you must send a request with the name of the account you want to request access to my personal email (c.celayaiturralde@gmail.com) or to the university email.

All code used for the various iterations of machine learning (ML) models is based on the same structure as the original NeoRanking model. First,

It performs preprocessing (preprocess_data.sh), then trains the models (train_classifier.sh), and finally evaluates them (test_voting_classifier.sh). The metrics can be viewed in a PDF file using CalculateMetricsAndGeneratePDF1.py. In each iteration, the models, some of their parameters, or the data are altered, or a program is added to this structure and the necessary parts are modified to adapt to the change. To learn more about the initial structure of the NeoRanking model, see Appendix 2.

The NCI cohort (112 patients, divided into 89 training and 23 testing) includes melanoma, colorectal, lung, and breast adenocarcinoma. Immunogenicity was assessed with IFN-γ ELISpot, and it stands out for having the lowest selection bias and the lowest density of positive neoantigens.

The TESLA cohort (8 patients) focused on melanoma and lung cancer. HLA-I multimers were used to assess TILs and PBMCs. Peptide selection prioritized affinity and expression by RNA-seq, resulting in a high density of positive neoantigens.

HiTIDE (11 patients) included metastatic melanoma, lung cancer, kidney cancer, and stomach cancer. Its evaluation was also performed with ELISpot, using TILs generated in vitro. Extensive pre-screening based on MixMHC, PRIME, gene expression, and gene coverage was applied, resulting in a higher number of immunogenic peptides per mutation. Both TESLA and HiTIDE were used as test sets.

During data balancing, an extreme imbalance was identified that could not be effectively resolved, so it was decided to develop two different algorithms: one for tumors with a high presence of positive neoantigens (HPPN) and another for low presence (LPPN). In the case of HPPN, the TESLA and HiTIDE cohorts were included, and intensive random subsampling was applied to NCI, reducing the number of negative neoantigens from 100,000 to just 1,000, with the aim of approximating an imbalance of 1:50.

In contrast, for the LPPN algorithm, the complete NCI_test dataset was retained, whose imbalance (1:5,000) is more representative of tumors such as pancreatic ductal adenocarcinoma (PDAC). In this type of cancer, which has not undergone prioritization processes for sequencing, the low mutational burden produces a low density of positive neoantigens, with an estimated average of 35 per tumor ([36]).Although this model yielded unfavorable metrics, it was decided to continue its development to verify whether its results were comparable to those obtained with HPPN.

On the other hand, to implement deep learning (DL) algorithms, it was necessary to significantly expand the initial database, as the NeoRanking data were not

sufficient. To this end, additional databases were used, mainly Neodb [34] , supplemented by ITSNdb [37] and DEPneo [38] . Unlike NeoRanking, where all haplotypes are annotated per patient, these databases only record the haplotype with the highest affinity for each neopeptide, which significantly alters the calculation of the characteristics, as it directly affects the final value of those dependent on the HLA context. The complete extraction and processing methodology is detailed in Appendix 3. The following programs are in the Biostatistics folder.

A UMAP analysis was also performed using the original NeoRanking data, excluding unlabeled instances, with the aim of studying the distribution of the cohorts. Only the five most relevant features according to Shapley values were selected, as the remaining ones added noise and made it difficult to form clear clusters. This analysis sought to identify whether there was any dataset whose distribution was similar to that of pancreatic cancer, which was particularly relevant given the difficulty subsequently observed in generalizing across cohorts. The UMAP_step_1.py and UMAP_step_2.py programs were used for this purpose.

Finally, after integrating the data from DL and ML and following the instructions in Annex 4, several comparative box plots were created with Wilcoxon statistical analysis focusing on the four main characteristics and distinguishing between positive and negative classes using NeopepBoxplot.py. Specific box plots were also generated for the available mutations in order to study their distributions in the new combined dataset using MutationsBoxplot.py.

Correction of metrics

NeoRanking used the Top-k metric to evaluate the performance of algorithms when classifying neoantigens with the highest probability of being immunogenic. However, it was observed that this metric was applied inconsistently, as it was calculated by summing the individual Top-k per patient rather than globally per cohort, which generated results higher than the analyzed k value and made interpretation difficult. In addition, this approach prevented the incorporation of neoantigens annotated in the literature, as in the case of those used for DL. For these reasons, the evaluation algorithm was redesigned to incorporate more global metrics that are adaptable to new data and easier to interpret.

In this context, more robust metrics for sorting tasks were explored, with particular emphasis on NDCG. Unlike Top-k, which only evaluates the first items on the list, NDCG measures the quality of the entire sort, giving greater weight to the most relevant items

relevant items in high positions. To do this, it calculates a cumulative gain (DCG), which decreases logarithmically with position, and normalizes it against the ideal order (IDCG) [39]. This metric allows models with different numbers of positives to be compared and is particularly useful when the position in the ranking is critical [40].

Despite the difficulties associated with establishing thresholds to separate positive and negative classes, the precision-recall curve continues to be evaluated. This curve provides information on the feasibility of setting an effective threshold and, once defined, allows for the rapid filtering of patients with a low probability of response, which helps optimize clinical resources and reduce unnecessary adverse effects [41].

The Top-k metric was maintained in the section on therapies, using it together with the quality factor as a tiebreaker, and the top 20, 50, and 100 values were evaluated. On the other hand, some sections of the appendix included the confusion matrix corresponding to different data sets, used to check whether the balancing strategies had been effective.

Data filtering

All of the following programs can be found in the initial_filtering folder. Data labeled as "not_tested" (unlabeled) was excluded, as it was initially classified as negative in the NeoRanking model because its inclusion increased the proportion of false negatives, did not provide any informative value, and there are too many instances of the negative class. The filter_neopep_data_org.py program in NeoRanking_initial was used for this purpose.

To improve training, filtering was applied to improve subsampling on the negative class. The 100,000, 10,000, and 1,000 neoantigens with the highest probability of being immunogenic within that class were selected, prioritizing those that were more difficult to classify in order to reduce the amount of data, thereby reducing the computational capacity required to run the program and achieving a more selective subsampling of the data. For this, the basic program was used, followed by filter_top_neopeptides.py, and the data was divided into different tops that were used in each folder with the corresponding reduction.

Unlike the filtering proposed by NeoRanking, which only considered patients with at least one immunogenic neoantigen, because the way the metrics were configured did not provide any extra value, in this study we chose to include all patients under the assumption that this approach allows for more representative and realistic learning about the problem by including more peptides that are difficult to classify and are not immunogenic. Therefore, the metrics of the top 100,000 were compared between two programs, one that included filtering for each

patient and another that is not NeoRanking_top_100000_with_filtering and NeoRanking_top_100000_without_filtering. The graphs shown only belong to HPPN.

For LPPN, the same filtering was performed to improve subsampling and unlabeled data, but since there was a different set of NCI_test, these changes were reevaluated. Removing the filtering by patient was not repeated, as its effectiveness had been previously demonstrated.

For mutation data, only unlabeled data and immunogenic peptide filtering were removed for each patient, so there is no need to establish a baseline model, as the effectiveness of the filters is already shown in HPPN.

Model testing

With the aim of improving the prediction of neopeptide immunogenicity, different machine learning models were explored, selected for their complementarity in terms of generalization capacity, sensitivity to minority classes, and ease of interpretation. Table 2 summarizes their main characteristics and advantages in the biomedical context. These models were applied to HPPN, LPPN, and mutations within the folders corresponding to the base model, mainly modifying the OptimizationParams.py and test_voting_classifier.sh files and adding plot_algorithms.py and analyze_combinations.py. The methodological proposals were first applied to HPPN, given its positive bias, since if they did not show improvement in this configuration, it was unlikely that they would do so in LPPN or mutations.

Table 2. Comparison of the advantages of ML models

Model	Key advantages	References
Decision Trees	- Simplicity and high interpretability—Generation of understandable rules, useful in biomedical contexts	[42]
Random Forest	- Reduction of overfitting through tree aggregation—Good tolerance to noise and high dimensionality, while maintaining partial interpretability	[44]
AdaBoost	- Iterative improvement of classification errors - Effective in unbalanced data sets	[46]

Model	Key advantages	References
Gradient Boosting	- High accuracy through iterative correction of model errors	[47]
K-Nearest Neighbors	- Intuitive and non-parametric model - Useful as a baseline or for comparing behaviors in simple data sets	[48]

At HPPN, the problem was framed as a difference in distributions between cohorts, attempting to make TESLA and HiTIDE resemble NCI_test. To do this, a redistribution was performed in one iteration, randomly sampling 50% of the TESLA data and 50% of the HiTIDE data to incorporate them into the training set. This procedure is documented in the ML_with_dataset_changed folder, in the redistribute_TESLA_and_HiTIDE.py program. On this redistribution, normalization was applied with COMBAT, a technique commonly used in gene expression studies to correct batch effects between experiments, while maintaining relevant biological variability ([50]). Its implementation can be found in the ComBat folder, with ComBat.py being the main file.

Deep learning approaches, such as NeoaPred and Neodb, were applied to both DL-specific databases and ML data, based on the hypothesis that they could generate models less dependent on the affinity features used in NeoRanking. These experiments are grouped in the DL_and_metamodel folder. In the case of NeoaPred, computational limitations and the high time cost of calculating the immunogenicity prediction restricted its use to 200 samples from NCI_train. The modified programs are run from the original program, through Foreigness_processing.py and metrics_estimator.py.

Neodb was used to evaluate whether the model was capable of generalizing with data annotated with all haplotypes or whether it depended on each neopeptide being associated with the haplotype with the highest affinity. In addition, the model was retrained with the NCI cohort to verify whether an improvement could be obtained. The procedure can be found in the GNN-Neodb folder, and HLA_correction.py must be run before GNN2.py.

TabPFN, a transformer model trained with millions of synthetic classification tasks on tabular data, was used as an advanced reference model due to its generalization ability and training speed once the model was loaded. Although it belongs to the field of deep learning, it works exclusively with tabular data, so it was only used

with HPPN to evaluate whether it could outperform traditional ML models. It is located in the TabPFN folder and runs on Google Colab due to GPU requirements. The input data is obtained from Neopep_data_org.txt.

Inspired by the approach described in Interpretable Machine Learning Models for PISA Results in Mathematics [33], a stacking-type metamodel was proposed to combine the predictions of individual ML models. This approach has been shown to improve predictive performance through the structured integration of optimized base models, without sacrificing interpretability. For its implementation, different model selection strategies were tested: by importance (top importance), by diversity, and by type balance. The most effective strategy was diversity. All versions are found in the MetaModel folder; the final version used was model_architecture_XGB8.py, executed with run_metamodel_XGB_v8.py.

In the case of LPPN, an attempt was made to improve the balance by applying SMOTE-Tomek, a technique that combines the generation of synthetic examples of the minority class using SMOTE [49] with the elimination of ambiguous examples through Tomek Links [51] . The objective was to improve the separation between classes and the quality of the dataset. The implementation can be found in ML_baseline_with_SMOTE, with modifications to the main training programs.

For the mutation approach, a different strategy was adopted from that of HPPN, aimed at reducing the difference between training and testing metrics. The hyperparameters were adjusted in OptimizationParams.py so that the models were more regulated, within the Mutations_regulated folder, using the NeoRanking base program for execution. From this point on, the unified methodology was applied to HPPN, LPPN, and mutations, since the previous iterations did not yield satisfactory results.

In order to select the best ensemble system, a program was used that evaluated all possible combinations of the most promising algorithms by voting, assigning the same weight to each one. This was developed in the Ensemble folder, adapting the test_voting_classifier.sh program to the new operation.

Once the best immunogenicity prediction model had been identified, it was necessary to retrain it with a reduced number of features, as many were not available for pancreatic cancer data. This modification did not affect performance, as the removed features were not used by the model, as can be seen in the Shapley values. The columns removed were: mutant_other_significant_alleles, gene_driver_Intogen, nb_same_mutation_Intogen, mutation_driver_statement_Intogen and

mut_is_binding_pos_DAI_MixMHC_mbp. The rest of the set was formatted appropriately using Quita_columnas.py, located in the Predict folder, and then the base program was run to obtain the metrics and the precision-recall curve. Finally, using predict_immunogenicity.sh, predictions were made on the pancreatic cancer data. For LPPN, the same case occurred as with HPPN. In the case of mutation, it also did not affect performance when retraining, and the following features were removed: nb_same_mutation_Intogen, nb_mutations_in_gene_Intogen, mutation_driver_statement_Intogen, and gene_driver_Intogen were removed.

Biomedical Application

The tasks corresponding to the Biomedical_applications_and_quality_analysis section begin with the calculation of the quality factor for the NeoRanking data, following the methodology described in Luis Miguel's TFG. This implementation can be found in the quality_calculator folder, in the Calculate_quality.py file. Subsequently, we analyzed how to combine this quality with the immunogenicity results, using different association metrics: pointwise biserial correlation coefficient, Mann-Whitney U test with its corresponding p-value, and normalized mutual information.

Due to computational limitations, quality results were only available for 169,000 neoantigens and 57 patients with pancreatic cancer. The impact of the quality factor on sorting metrics (NDCG and Top-k) was evaluated by applying a tiebreaker system, in which immunogenicity values were modified by rounding at different thresholds. This part can be found in the Neopeptide_reordering_v2.py program.

Based on the best rankings combining immunogenicity and quality, predictions were made on pancreatic cancer data. First, the Precision-Recall curve was analyzed to determine whether it was possible to establish a threshold without significantly compromising precision or recall. In this process, it was decided to prioritize precision, and the optimal threshold was selected using confidence intervals generated by resampling (10,000 samples with replacement). For each candidate threshold, the observed precision and its 95% confidence interval were calculated, selecting the one whose lower limit exceeded 95%, thus maximizing coverage without losing statistical confidence. This procedure is implemented in Neopep_threshold.py.

Based on the defined threshold, patients with immunogenic neoantigens and their quantity were identified using Analyze_patients.py. In addition, the diagnostic and therapeutic potential was evaluated by analyzing the recurrence of neoantigens in the

top 10, 20, and 30 positions, as well as the genes associated with these peptides, using common_neoantigens.py.

For LPPN, an attempt was made to replicate the same methodology, but it was not possible to calculate the quality factor due to the high computational cost. Nevertheless, the same procedure was applied to determine the optimal threshold and analyze possible therapeutic and diagnostic applications, using programs analogous to those for HPPN.

In the case of mutations, once the model was optimized, the ensemble was applied in the same way as in HPPN. Pancreatic cancer data were sorted by immunogenicity, but the quality factor was not calculated, as it is not computationally feasible for sequences of this length. The threshold was set using the same statistical criteria, but prioritizing 100% recall, since in this case it is essential not to lose any immunogenic neoantigens. This analysis was performed with mutations_threshold.py, and the number of filtered mutations was measured with filtro_de_mutaciones.py.

Finally, the 15 best peptides most frequently repeated by HPPN and LPPN are used and a Venn diagram is created to see which are the best common ones using Venn_diagram.py

Results

Biostatistics

Table 3 shows the class distribution in the different datasets used. The NCI_test_LPPN, TESLA, and HiTIDE datasets are part of the original NeoRanking dataset, after excluding unlabeled samples. NCI_test_HPPN was generated from a filtered subset, described in the Materials and Methods section. NCI_train contains negative neoantigens selected after specific negative class filtering. On the other hand, NEPdb, NeodbTrain, NeodbTest, and ITSNdb were integrated to expand the deep learning database. A significant imbalance between classes is observed, reflected in the "Percentage of positives" column, which was taken into account throughout the analysis.

Table 3. Number of positive and negative instances in the datasets

Data set	Negative	Positive	Totals	Percentage of positives
HiTIDE_test	1.517	41	1,558	2.63
ITSNdb.csv	69	8	77	10.39

NCI_test_HPPN	1,007	21	1,028	2.04
NCI_test_LPPN	123,464	21	123,485	0.02
NCI_train	10,144	81	10,225	0.79
NEPdb.csv	243	55	298	18.46
NeodbTest.xlsx	484	736	1,220	60.33
NeodbTrain.xlsx	4,797	3,058	7,855	38.93
TESLA_test	702	34	736	4.62

The UMAP analysis (Figure 4) reveals that the cohorts closest to the pancreatic cancer profile are NCI_train and NCI_test, with a significantly lower Euclidean distance from the centroid than TESLA and HiTIDE. Qualitatively, the NCI data surrounds the PDAC data without excessive overlap, indicating similarity without complete identity. This proximity justifies its preferred use as a reference in models applicable to PDAC.

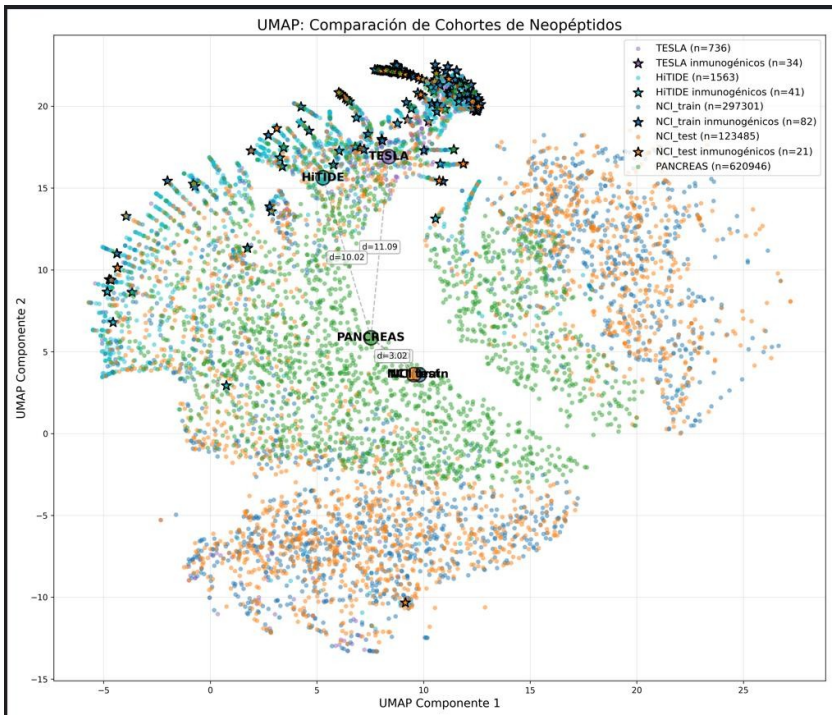

Figure 4. UMAP of the original data without unlabeled data.

Figure 5 shows box plots for the main features. There is a clear separation between classes in the NCI (both HPPN and LPPN) and TESLA sets. In

these cases, the distributions show minimal overlap, probably attributable to outliers. In contrast, in the rest of the datasets (HiTIDE, Neodb, etc.), the proximity of the medians and quartiles makes it difficult to differentiate between classes.

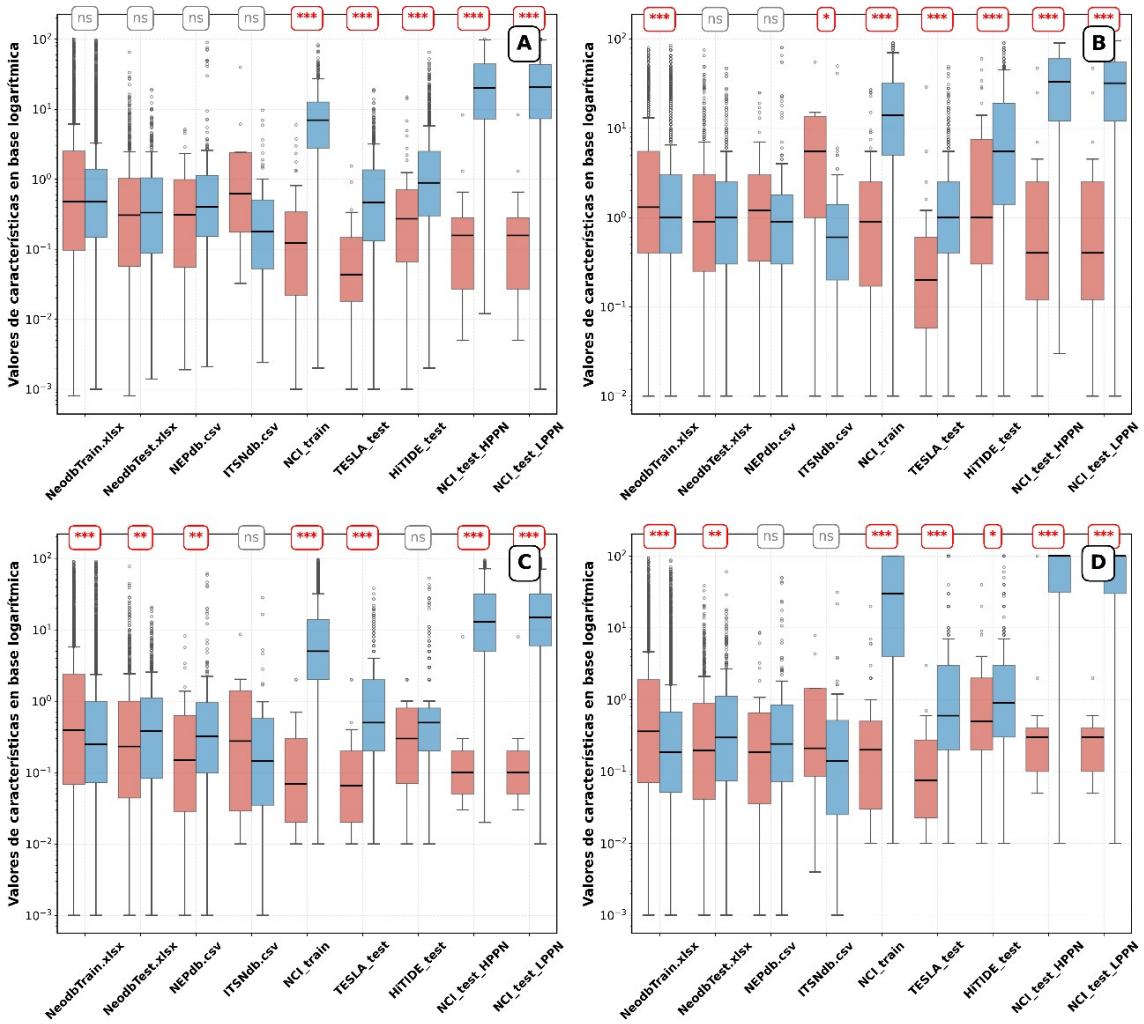

Figure 5. Box plots of the four main characteristics. The x-axis represents each data set and the y-axis represents the values of the characteristics on a logarithmic scale. A: EL_Rank_NetMHCpan, B: mut_Rank_Stab, C: mutant_rank, D: mutant_rank_PRIME. The Mann-Whitney U (Wilcoxon rank-sum) statistical test can be seen with the number of asterisks as follows:

****= p< 0.001 (highly significant) **= p< 0.01 (very significant)*

** = p < 0.05 (significant) ns= t significant (p≥ 0.05) Red is immunogenic and blue is non-immunogenic.*

Table 4 summarizes the class distribution in the mutation-based datasets. A much smaller imbalance can be seen than for neopeptide data.

Table 4. Differences between mutation distributions

Data set Data	Negative	Positive	Total	% Positive
HiTIDE	769	30	799	3.75
NCI	12,035	147	12,182	1.21
TESLA	466	36	502	7.17
TOTAL	13,270	213	13,483	1.5

The box plots (Figure 6) show partial overlap between classes in the most important characteristics, which makes it difficult to clearly separate immunogenic and non-immunogenic neoantigens.

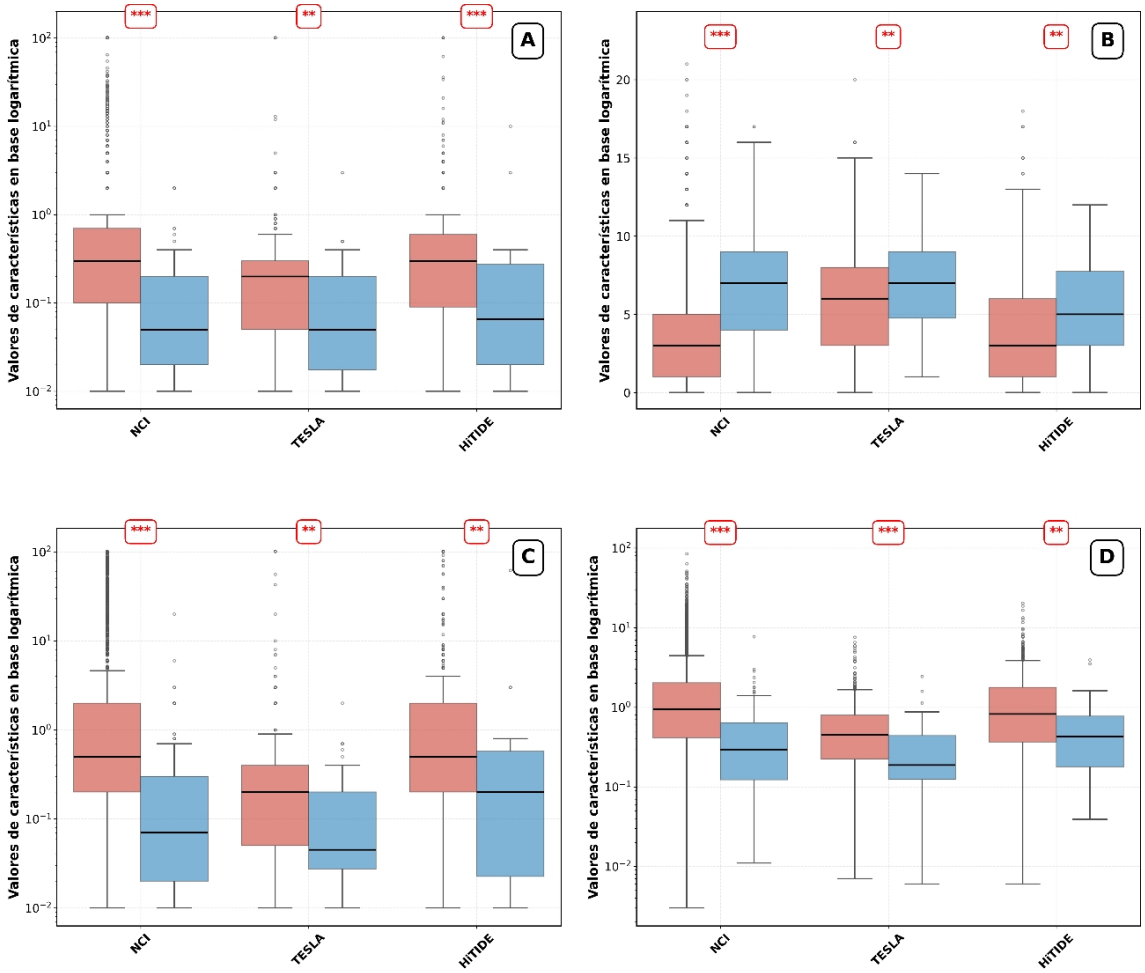

Figure 6 Box plots of the four main mutation characteristics in each cohort on the x-axis and the logarithmic value of each characteristic on the y-axis. A: MIN_MUT_RANK_CI_MIXMHC. B: COUNT_MUT_RANK_CI_netMHCpan, C: MIN_MUT_RANK_CI_PRIME, D: mut_Rank_EL_1. The Mann-Whitney U (Wilcoxon rank-sum) statistical test can be seen with the number of asterisks as follows:

****= p< 0.001 (highly significant) **= p< 0.01 (very significant)*

** = p < 0.05 (significant) ns= t significant (p≥ 0.05) Red is immunogenic and blue is non-immunogenic.*

Calculation of new metrics and filtering

Figure 7 shows the model performance after applying different subsampling strategies aimed at selecting only the neoantigens most likely to be immunogenic. In general, these filters improved the evaluation metrics, although they introduced overfitting. However, this improvement was smaller in the TESLA and HiTIDE cohorts, where the model did not generalize adequately.

Among the options evaluated, the worst-performing subsampling was the one that selected only 1,000 samples. In contrast, the top 100,000 and top 10,000 sets showed superior results, with slight variations depending on the cohort: the top 100,000 was more effective in NCI_test and HiTIDE, while TESLA obtained better metrics with the top 10,000. Given that processing the top 100,000 requires ten times more computational resources, the top 10,000 was selected as the optimal configuration. This decision is supported by the results of the confusion matrix (Appendix 5), where this subsampling obtained the best performance in the minority class and was more balanced.

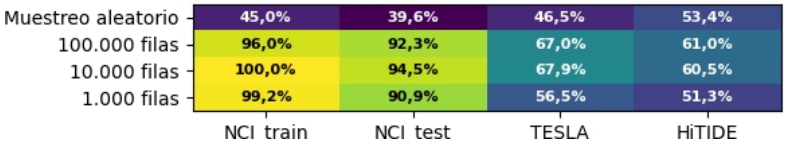

Figure 7. Heat map with NDCG on different improved subsampling filters in HPPN. The x-axis represents the datasets and the y-axis represents the filtering.

Figure 8 analyzes the impact of removing patient filtering, which excluded those without immunogenic peptides. Although the effect on the metrics was inconclusive, this change is considered beneficial as it allows for a more representative and realistic dataset, which is essential for improving the robustness of the model.

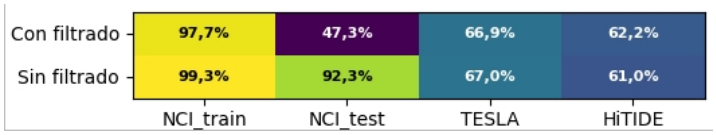

Figure 8. Heat map with NDCG for filtering patients who do not have immunogenic neoantigens. The x-axis represents the data sets and the y-axis represents the filtering.

Figure 9 shows the results of the improved subsampling in LPPN. Similar patterns to those described in HPPN were observed, although with lower overall performance. This deterioration is attributed to the extreme imbalance between classes. Although the top 100,000 presents better overall metrics (Appendix 6), the use of the top 10,000 was prioritized to favor computational efficiency and facilitate future iterations.

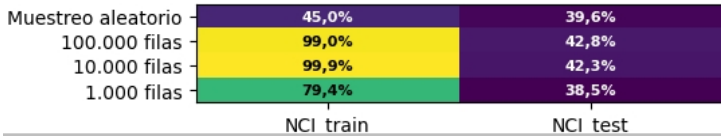

Figure 9. Heat map with NDCG on the different improved subsampling filters in LPPN. The x-axis represents the datasets and the y-axis represents the filtering.

Algorithm testing

Figure 10 shows the NDCG values for different algorithms evaluated in HPPN. As expected, the best performance was obtained in the NCI_test cohort, while TESLA and, especially, HiTIDE showed considerably lower predictive ability. The best-performing models were Logistic Regression, Gradient Boosting, AdaBoost, and Random Forest. Given its good performance on TESLA and HiTIDE, XGBoost was also considered promising. In contrast, Decision Trees and K-Nearest Neighbors showed poor performance and were discarded for further analysis.

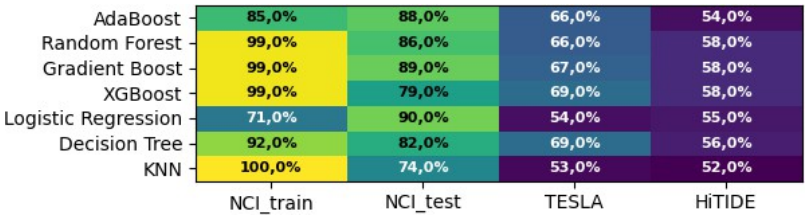

Figure 10. NDCG in HPPN when various ML algorithms were tested. The x-axis represents the datasets and the y-axis represents the models tested.

Figure 11 shows the effect of randomly redistributing 50% of the TESLA and HiTIDE data to create new training sets. This strategy slightly improved the metrics in TESLA but worsened them in HiTIDE, possibly due to the small number of

positive samples in each new set. In NCI_test, performance increased slightly in most algorithms.

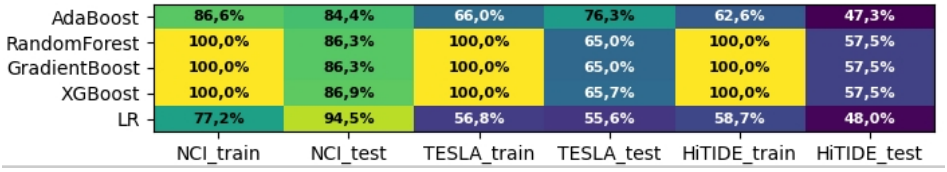

Figure 11. NDCG on HPPN when half of the HiTIDE and TESLA datasets were placed in the evaluation set. The x-axis represents the datasets and the y-axis represents the tested models.

Figure 12 shows the evaluation of normalization using the ComBat method. This technique improved the metrics in NCI_test and TESLA, but reduced performance in HiTIDE. Due to the loss of valuable samples and the lack of consistent improvements, it was decided not to apply this strategy in future iterations, restoring the initial configuration.

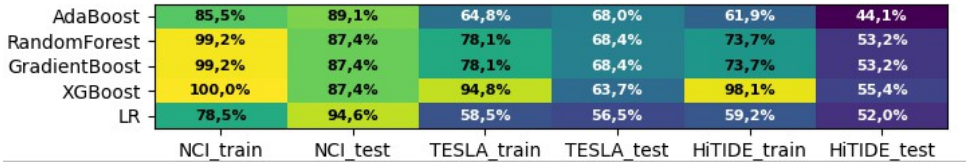

Figure 12. NDCG in HPPN when applying ComBat. The x-axis represents the data sets and the y-axis represents the models tested. The x-axis represents the data sets and the y-axis represents the models tested.

Figure 13 shows the results of the deep learning models. The performance of the GNN model was considerably lower than expected, despite being trained with data from NCI_train and NEPdb. Its poor performance on datasets such as ITSNdb and IEDB suggests that the cause is not solely related to haplotype annotation, but to limitations of the model itself.

The TabPFN model showed acceptable performance, but did not outperform traditional algorithms. NeoaPred, on the other hand, could only be applied to a small subset of 200 samples, which was sufficient to demonstrate its poor predictive power in this context.

As for the metamodel, an attempt was made to combine predictions from different classifiers with the aim of improving overall performance. However, no combination outperformed the

individual models, even in their best iterations. This suggests that the stacking strategy used did not add value in this case.

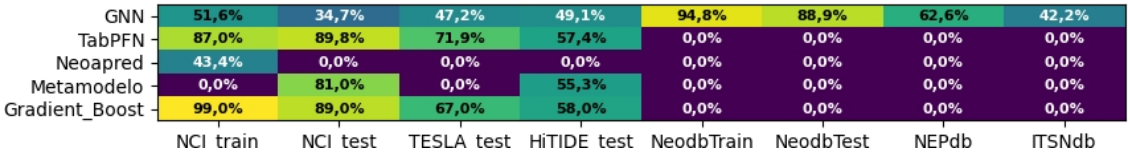

Figure 13. NDCG in HPPN after testing different Deep Learning models. The x-axis represents the datasets and the y-axis represents the models tested.

Figure 14 shows that the model applied to LPPN has significantly lower performance, with no effective predictive capacity. This shows that the SMOTE-Tomek balancing strategy was not suitable for this set. Despite this, the Random Forest, Gradient Boosting, and XGBoost algorithms stood out from the rest.

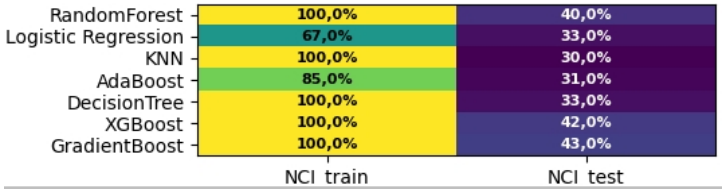

Figure 14. NDCG in LPPN when various ML algorithms were tested. The x-axis represents the data sets and the y-axis represents the models tested.

Figure 15 shows the results after testing different models with the mutations. There is clearly greater similarity in the performance of the TESLA and HiTIDE cohorts compared to NCI_test. Therefore, regularization techniques were applied to reduce model overfitting.

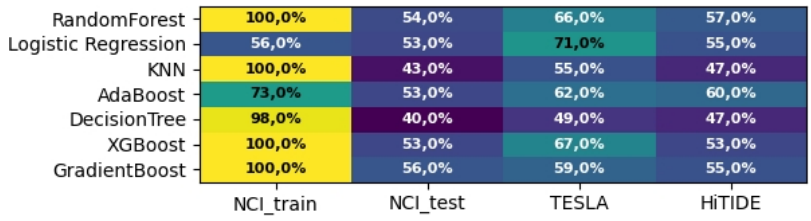

Figure 15. NDCG of mutations with different model testing. The x-axis represents the datasets and the y-axis represents the models tested.

Figure 16 shows that, although the effects of regularization were variable, there was an overall improvement in the metrics. For this reason, it was decided to incorporate this technique in subsequent iterations for the processing of mutation data.

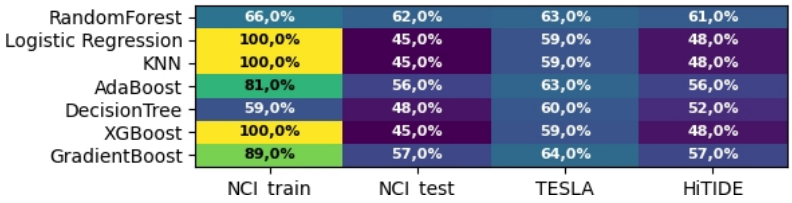

Figure 16. NDCG of mutations with regularization techniques. The x-axis represents the datasets and the y-axis represents the models tested.

Finally, it was decided to use the already trained models and evaluate all possible combinations of models in HPPN by voting among the different classifiers to see if any were better than the two used previously. As can be seen in Table 5, the combination of algorithms that works best overall is AdaBoost+ r GradientBoost, both in all data sets in general and in the TESLA and HiTIDE data sets.

Table 5. NDCG in the best combinations of HPPN algorithms

Combination of data sets	Combination of algorithms	NDCG
NCI_test	LR_XGBoost	0.9247
NCI_test	AdaBoost_LR_XGBoost	0.9202
NCI_test	GradientBoost_LR	0.9191
TESLA+ HiTIDE	AdaBoost_GradientBoost	0.6328
TESLA+HiTIDE	AdaBoost_GradientBoost_RandomForest_XGBoost	0.6312
TESLA+ HiTIDE	AdaBoost_GradientBoost_XGBoost	0.6306
All tests	AdaBoost_GradientBoost	0.718
All tests	AdaBoost_GradientBoost_XGBoost	0.716
All tests	GradientBoost_XGBoost	0.7121

In the case of the LPPN algorithm, the best-performing model combinations were consistent with those observed previously in the testing phase. As shown in Table 6, the Gradient Boost+ XGBoost combination offered the best results.

sustained manner, so it was selected as the final configuration for predictions in this scenario.

Table 6. NDCG in the best combinations of LPPN algorithms

Rank	Combination of Classifiers	NDCG
1	GradientBoost_XGBoost	0.45
2	LR_RandomForest_XGBoost	0.446
3	AdaBoost_LR_RandomForest_XGBoost	0.4451

With regard to mutations, a model ensemble strategy was also applied to identify the best combination. The most effective configuration was AdaBoost+ GradientBoost+ Logistic Regression+ Random Forest, which, although it was the second best in TESLA+HiTIDE, proved to be the most robust when evaluated on all test datasets and in NCI_test, as shown in Table 7.

Table 7. NDCG in the best combinations of LPPN algorithms

Combination of data sets Data	Combination of algorithms	NDCG
TESLA+ HiTIDE	GradientBoost_LR_RandomForest	0.6448
TESLA+ HiTIDE	AdaBoost_GradientBoost_LR_RandomForest	0.6427
TESLA+ HiTIDE	GradientBoost_LR_RandomForest_XGBoost	0.6422
NCI_test	AdaBoost_GradientBoost_LR_RandomForest	0.5871
NCI_test	GradientBoost_LR_RandomForest	0.5748
NCI_test	AdaBoost_LR_RandomForest	0.5737
All tests	AdaBoost_GradientBoost_LR_RandomForest	0.6242
All tests	GradientBoost_LR_RandomForest	0.6215
All tests	GradientBoost_LR_RandomForest_XGBoost	0.6182

Biomedical application

The analysis of the relationship between immunogenic quality and immunogenic annotation revealed a positive mean difference: 0.1968 in immunogenic samples versus 0.1229 in non-immunogenic samples (Δ = 0.074). Although the point-biserial correlation coefficient was very low (0.0219), it was statistically significant (p= 0.0104), suggesting a weak but not random association. However, the nonparametric Mann-Whitney test did not reach significance (p = 0.3313, effect = 0.0404) and the mutual information between the two variables was minimal (0.0006). Taken together, these results indicate that quality and immunogenicity are virtually independent.

Annex 7 presents a sensitivity analysis to decimal rounding of immunogenicity values for neoantigen reordering. It was observed that the rankings did not vary significantly between decimal places, except when rounded to the nearest integer, which deteriorated the predictive value. Therefore, rounding to one decimal place was chosen, which improved the NDCG metric and was retained as the criterion for future rankings.

Through resampling, Figure 17 estimated the optimal classification threshold for HPPN at 7.8891, achieving 100% accuracy with 27 neopeptides above the threshold. However, the Precision-Recall curve presented an AUC < 0.5, confirming the difficulty of finding a balance between precision and recall in this dataset.

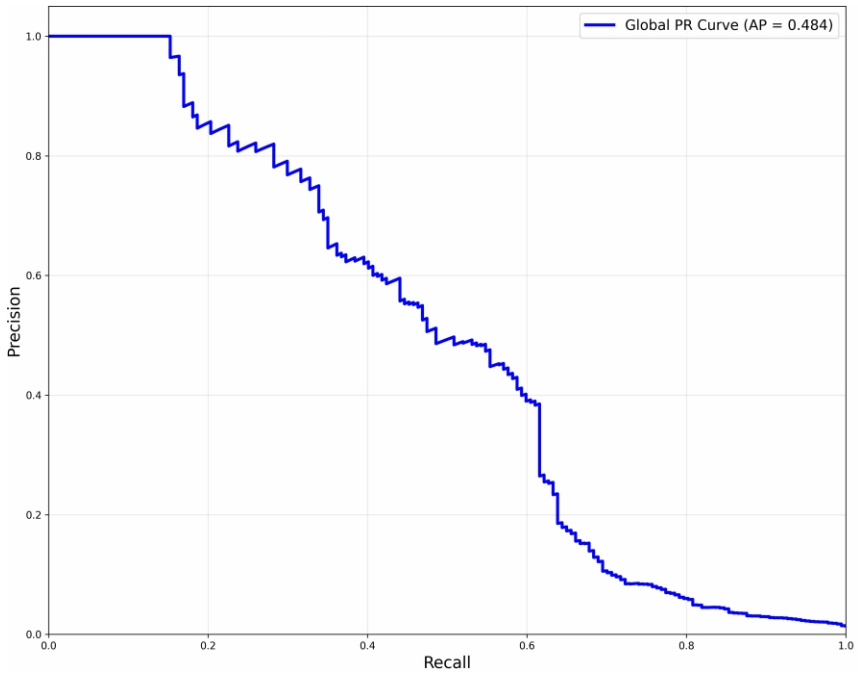

Figure 17. Precision-recall curve of HPPN in final predictions.

Despite the inherent positive bias of the model, its applicability for assessing immunogenicity in patients with pancreatic cancer was explored. The results, shown in Figure 18, show that a significant number of patients have at least one immunogenic neoantigen, although in varying proportions. This observation suggests a possible diagnostic value for the model, provided that accuracy is prioritized over sensitivity.

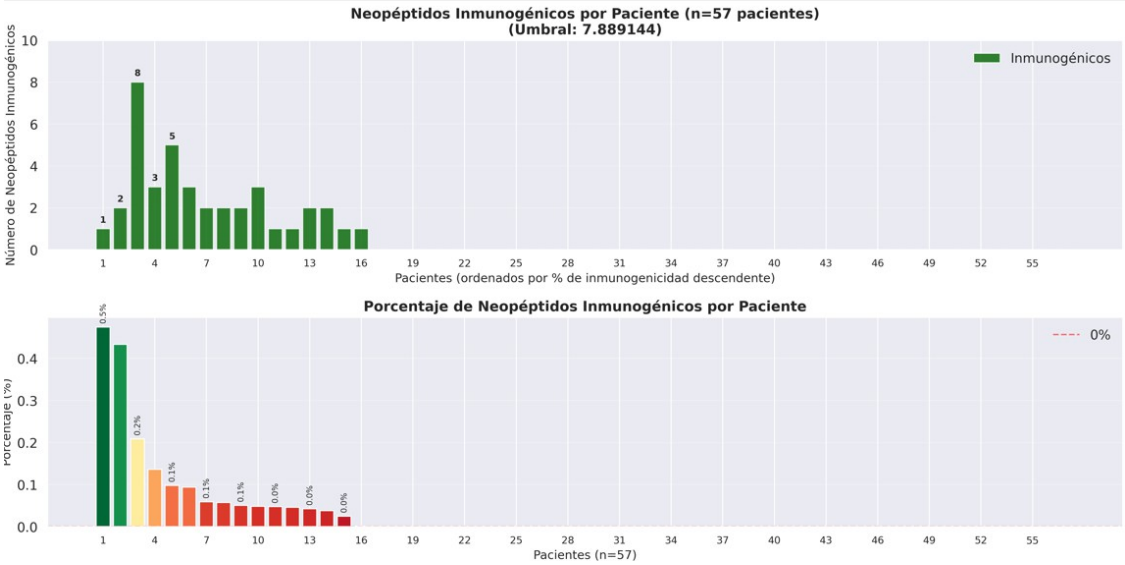

Figure 18. The first graph shows the patients on the x-axis and the number of immunogenic neoantigens per patient on the y-axis. The second graph shows the same data but as a percentage.

Next, the frequency of occurrence of the most recurrent neoantigens among different patients was evaluated, with the aim of identifying candidates for biomarkers or preconfigured TCR therapies. Several peptides were found to be present in more than 10% of patients, exceeding the generally accepted minimum threshold for biomarkers. However, there is some uncertainty about the biological validity of these findings, as the genes involved (such as MUC6) are not classic oncogenes such as KRAS or TP53, as documented in Appendix 8.

Table 8. Most common neoantigens in the top immunogenic neoantigens in pancreatic cancer data with HPPN. The number of times a neoantigen is repeated in the top 10, 20, and 30 among several patients (N) and the frequency in the total number of patients with which it is repeated (F) are shown.

Rank	Neoantigen	Gene	N10	N20	N30	%F10	%F20	%F30
1	TPSLPQTTL	MUC6	8	9	11	14	15.8	19.3

2	YPTPSLPQTTL	MUC6	2	7	9	3.5	12.3	15.8
3	TPNTHTVII	MUC6	5	6	9	8.8	10.5	15.8
4	LVTPNTHTV	MUC6	6	8	8	10.5	14	14
5	STGPITATSF	MUC6	3	8	8	5.3	14	14
6	SPFSSTGPITA	MUC6	6	7	8	10.5	12.3	14
7	SPFSSTGPI	MUC6	3	7	8	5.3	12.3	14
8	KALQTVISF	TAS2R46	6	6	8	10.5	10.5	14
9	TTTYPTTSH	MUC6	3	4	7	5.3	7.0	12.3
10	HIKALQTVISF	TAS2R46	0	6	6	0	10.5	10
11	FSSTGPITA	MUC6	4	5	6	7.0	8.8	10.5
12	SPFSSTGPVTA	MUC6	3	5	6	5.3	8.8	10.5
13	TTTYPTPSL	MUC6	3	5	6	5.3	8.8	10.5
14	TTYPTPSL	MUC6	2	5	6	3.5	8.8	10.5
15	HIKALQTVI	TAS2R46	0	5	6	0.0	8.8	10.5

In the case of LPPN, a threshold of 4.999999 was defined, also with 100% observed precision, but this only allowed three neopeptides above the threshold to be identified. The Precision-Recall curve in Figure 19 in this case showed an extremely low AUC of 0.272,

made it impossible to apply the model with confidence to the pancreatic cancer data and, as a result, no immunogenic neoantigens were detected using this criterion.

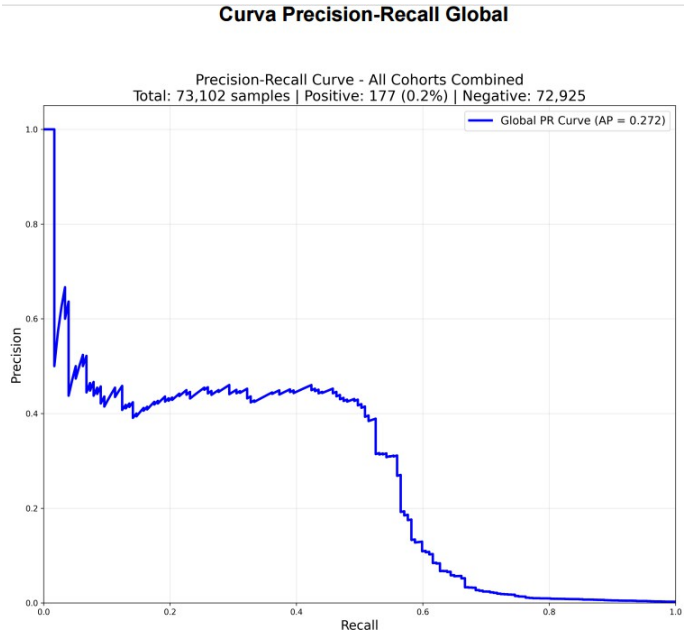

Figure 19. Precision-recall curve for LPPN

Despite this, when LPPN neopeptides were reordered according to immunogenicity, several were identified with high recurrence rates among patients, suggesting potential for use as biomarkers or in immunotherapy. Observations regarding the genes involved are consistent with those observed in HPPN and can be reviewed in Appendix 8.

Table 9. Most common neoantigens in the top immunogenic neoantigens in pancreatic cancer data with LPPN. The number of times a neoantigen is repeated in the top 10, 20, and 30 among several patients (N) and the frequency in the total number of patients with which it is repeated (F) are shown.

Rank	Neoantigen	Gene	N	N20	N30	%F10	%F20	%F30
1	TPSLPQTTL	MUC6	6	8	11	10.5	14.0	19.3
2	LVTPNTHTV	MUC6	9	10	10	15.8	17.5	17
3	STGPITATSF	MUC6	7	9	9	12.3	15.8	15.8
4	TPNTHTVII	MUC6	5	8	9	8.8	14	15.8
5	TTTYPTTSH	MUC6	4	7	7	7.0	12.3	12

Rank	Neoantigen	Gene	N10	N20	N30	%F10	%F20	%F30
6	SPFSSTGPVTA	MUC6	5	5	7	8	8.8	12.3
7	FSSTGPITA	MUC6	3	5	7	5.3	8.8	12.3
8	TPNTHTVI	MUC6	2	5	7	3.5	8.8	12.3
9	YPTPSLPQTTL	MUC6	4	4	7	7	7	12.3
10	SPFSSTGPITA	MUC6	4	5	6	7.0	8.8	10.5
11	KALQTVISF	TAS2R46	4	5	6	7.0	8.8	10.5
12	TTTTYPTTSH	MUC6	0	3	6	0.0	5.3	10.5
13	SEVDKSKEEL	CTAGE6	4	5	5	7.0	8.8	8.8
14	TTYPTPSL	MUC6	2	5	5	3.5	8.8	8.8
15	IVLGIFVRY	SLC9B1	4	4	5	7	7	8

Figure 20 shows a Venn diagram with the immunogenic peptides common to both models, supporting the existence of shared neoantigens with potential therapeutic value.

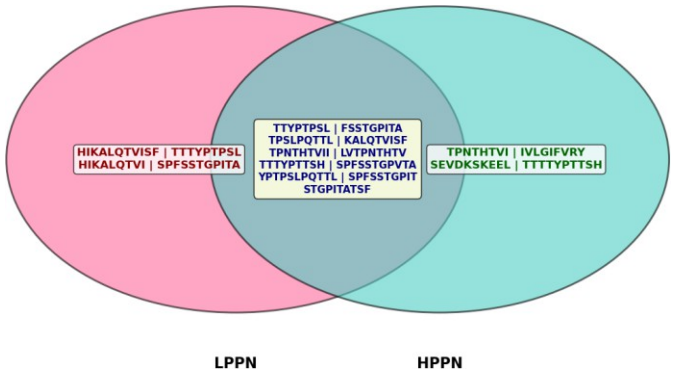

Figure 20. Venn diagram showing the most immunogenic neoantigens in HPPN and LPPN.

Finally, for mutations, the same threshold methodology was used as in HPPN, but prioritizing recall. A conservative threshold of 0.7551 was established, which allowed 100% sensitivity to be achieved without false negatives. This filter retained 15.54% of

the 12,811 mutations, significantly reducing the computational load without compromising relevant cases.

Discussion

The change to NDCG metrics is considered a success, as it allowed more patient data to be incorporated and performance to be compared objectively. Unlike Top-k, NDCG evaluates the entire ranking, not just the top positions, which gives a more accurate view of the model. Being normalized as a percentage, it facilitates comparison between different data sets, even when introducing DL models. In addition, it allowed for a clearer assessment of how filtering affects the order of results, revealing improvements over the original metric used in NeoRanking.

UMAP analysis confirmed that the NCI cohort is most similar to pancreatic cancer in terms of feature distribution, which is consistent, as both were not pre-filtered and tend to have low positive neoantigen presence (LPPN). Although the algorithm performed poorly, this confirms that if it can be generalized for NCI, good generalization for pancreatic cancer data will be achieved.

As a recommendation for the future, it is suggested that UMAP be applied before using the model with new cancer data. Evaluating the proximity of the centroid to the reference cohorts could anticipate the level of performance to be expected and avoid predictions on data with very different distributions.

Following biostatistical analysis, it was concluded that the characteristics are not very discriminative in the TESLA and HiTIDE cohorts, with HiTIDE being the most difficult according to box plots and whisker plots. This suggests limited predictive power in both, which was also confirmed when applying techniques such as data redistribution or normalization by ComBat, with no relevant improvements. This is probably due to the prioritization system used in these trials, where only peptides with high affinity for MHC were sequenced. Since immunogenicity also depends on other factors, such as affinity with TCR, many of the selected peptides have similar characteristics, making their classification difficult.

Pre-filtering to remove unlabeled cases was not accompanied by specific metrics, as its usefulness was previously supported by NeoRanking. This strategy helps reduce false negatives and limit the negative class, which was also confirmed during the subsampling process.

Filtering by patient made sense under the original metrics, but with the switch to NDCG, it is no longer useful. As shown in Figure 8, its application reduces performance by eliminating peptides that are difficult to classify. Therefore, it is considered appropriate to dispense with it under the current metrics.

Regarding subsampling, a clear improvement is observed when selecting the 10,000 most difficult negative neoantigens, rather than applying random reduction. In HPPN, this strategy improves metrics and reduces computational cost. In LPPN, although keeping all data could be beneficial for learning, applying SMOTE on such a large set would be inefficient. Therefore, filtering is also maintained at 10,000 samples.

The results with Deep Learning models, such as NeoAPred and Neodb's GNN, support the hypothesis that exploring features other than those in NeoRanking is a promising avenue, especially for data where the latter do not perform well, as discussed above. However, significant challenges remain: it is necessary to correctly adapt the representation of multiple haplotypes in ML datasets to the more restricted format of DL models, which typically use a single HLA per peptide or vice versa. It is recommended to continue along these lines and, instead of choosing the one that shows the highest affinity in NetMHCpan, evaluate whether it would be possible to develop a better methodology.

TabPFN showed promising performance, even outperforming classic models on TESLA. Its advanced architecture allows for greater abstraction capacity. However, it was not integrated into the voting system due to its high computational cost and average performance on other datasets.

The meta-model yielded low metrics in the evaluation set, probably because it was trained with the same data as the base models. This limits its ability to learn effective combinations and suggests that it should be trained more independently.

After evaluating multiple models, no single model was identified as clearly superior for classifying neoantigens in the three algorithms used. All exhibited overfitting, which was addressed differently in each workflow. Although some models obtained better individual metrics, it was decided to keep several in the model voting or stacking process, as they may specialize in different types of neoantigens. For example, in HPPN, Logistic Regression and XGBoost stand out in NCI_test, possibly because they capture more linear patterns. In contrast, KNN and Decision Trees were discarded due to their poor performance and limited contribution.

In LPPN, the use of SMOTE-Tomek did not improve the metrics, probably due to extreme imbalance. It is recommended to explore models more suitable for highly imbalanced contexts, such as those aimed at anomaly detection.

In the case of mutations, the original dataset had fewer negative examples, which prevented replicating the initial NCI filtering. This was treated as a case of overfitting, and regularization techniques were applied, which improved performance on the evaluation set. Ideally, this regularization should have been extended to the other models, although this was not possible due to time constraints.

The best model combination on HPPN was AdaBoost + GradientBoost, with good performance on all three cohorts. On NCI_test, Logistic Regression and XGBoost also performed well, possibly due to the data structure.

With regard to the quality factor, it may not be well adjusted for ML datasets, as they are not specific to pancreatic cancer. However, the study on which it is based supports its usefulness. The tiebreaker methodology applied does not negatively affect overall performance and should improve on PDAC-specific data, where it was originally validated. However, it is considered that in order to demonstrate its usefulness, it should be tested on labeled pancreatic cancer data.

The Precision-Recall curve showed an AUC (<) of 0.5, indicating that neoantigens are difficult to separate in binary classification. In addition, the decline in precision after the 100% threshold is very abrupt. Therefore, we chose to work with 100% precision, prioritizing diagnostic safety. This makes sense in therapies, where only a fraction of neoantigens usually induce an immune response. This approach was useful in HPPN, but could not be applied in LPPN, as the threshold was found to be too restrictive, preventing reliable predictions.

Threshold analysis shows that 34% of patients have at least one immunogenic neoantigen. Based on the criterion of prioritizing precision over recall, it can only be stated with certainty that the detected antigens are very likely to be immunogenic. Given the number of patients affected, it makes sense to continue investigating this line of research. Although it cannot be said that the number of patients is insufficient, since safety is being prioritized over quantity, the algorithm cannot be used for vaccines and immunotherapy.

Threshold analysis in HPPN revealed that 34% of patients had at least one immunogenic neoantigen. Although the approach prioritizes precision over recall, it ensures that the antigens detected are highly likely to be real. Although the number of patients with

positive predictions is low, it does not invalidate the approach, which is aimed at minimizing false positives.

As for their application as biomarkers or preconfigured TCR therapies, the results are promising. A sufficient frequency is observed in the top 10, 20, and 30 to consider them viable candidates. In addition, there is recurrence of certain neoantigens among patients, both in LPPN and HPPN, as shown in the Venn diagram. This justifies conducting in vitro trials to validate their clinical utility.

One of the most recurrent genes was MUC6, which encodes a mucin. Although it is not a classic oncogene, its size and structure could explain its high mutation frequency and appearance in the rankings. For the genes obtained as the most immunogenic, it is recommended to investigate why these genes have been the most frequent.

In the case of mutations, the threshold was established by prioritizing recall using a statistical approach similar to that of HPPN. This ensures that no immunogenic neoantigens are lost in the filtering process. Thanks to this strategy, the computational load was reduced by 20% without compromising the sensitivity of the analysis.

Conclusion

The main objective of this thesis was to improve the predictive capacity of neoantigen immunogenicity in the context of pancreatic ductal adenocarcinoma (PDAC), with a focus on its therapeutic potential. Based on the results obtained, it can be stated that the objectives set have been largely achieved:

Improvement of NeoRanking model metrics: An improvement in evaluation was achieved with NDCG, as this modification facilitated the incorporation of new datasets and improved the overall interpretation of the immunogenicity ranking.

Initial filtering to reduce computational load: An efficient filtering system was designed to select the most promising neoantigens within the negative class. This strategy reduced computational capacity without sacrificing model performance, thereby improving learning.

Exploration of new Machine Learning and Deep Learning algorithms: Multiple models were tested, including Gradient Boosting, XGBoost, AdaBoost, and deep neural networks. Although some DL models such as GNN or NeoAPred did not achieve the expected performance,

the combination of algorithms using ensemble techniques demonstrated improvements, especially in high-density positive neoantigen environments.

Implementation of the immunogenic quality parameter: Although the quality parameter has been successfully integrated to ensure performance improvements, its effectiveness in labeled pancreatic cancer data still needs to be demonstrated.

Extraction of neoantigens with therapeutic and diagnostic potential: Based on the data applied to PDAC, several candidate neoantigens were identified with sufficient population frequency to be considered biomarkers or candidates for preconfigured TCR therapy. Although the algorithm applied has a positive bias and certain limitations in PDAC data, the results suggest that there is a subset of recurrent neoantigens with clinical potential. The algorithms cannot be applied to immunotherapy due to poor performance for binary classification.

Overall, this work has achieved its objectives, developing a more robust, generalizable, and clinically relevant predictive model for the evaluation of neoantigens. Despite some limitations, such as the inability to differentiate TESLA and HiTIDE based on characteristics, extreme imbalance, or differences between sets, the improvements introduced and the results obtained support the usefulness of this approach for future applications in personalized immunotherapy, especially in HPPN approaches and with high prioritization.

Appendix

Appendix

Estimation of the search space for peptides of length 8 to 12:

The total number of possible fixed-length peptide sequences can be calculated by considering the combinations of the 20 canonical amino acids. For a length n, the total number of combinations is 20^n . When considering peptides of length between 8 and 12 residues, the total search space is defined as:

$$\sum_{i=8}^{12} 20^n = 20(^8) + 20(^9) + 20(^{10}) + 20(^{11}) + 20(^{12})$$

This is approximately equivalent to $4.34 \cdot 10^{15}$ unique sequences possible s.

Appendix 2

Python libraries used throughout the project:

pandas v2.0.3: Data manipulation and analysis

numpy v1.24.4: Numerical operations and arrays

scikit-learn v1.3.2: Transformers for (QuantileTransformer, StandardScaler, PowerTransformer, Min (QuantileTransformer, StandardScaler, PowerTransformer, MinMaxScaler, FunctionTransformer)

collections.Counter: Frequency counting for imputation (standard Python module) Hyperopt v0.2.7:

Bayesian hyperparameter optimization

SHAP v0.41.0: Calculation of Shapley values for interpretability XGBoost

v1.5.1: Gradient boosting algorithm

CatBoost v1.0.4: Gradient boosting algorithm for categorical features Genomic

Analysis Tools

HLA-HD v1.4.0: HLA typing from WES and RNA-seq data STAR v2.7.8a:

RNA-seq read alignment

GATK v4.2.0.0: Genomic analysis workflow (HaplotypeCaller, Mutect2) Sequenza v3.0.0:

Tumor content and copy number estimation

Mutect1 v1.1.5: Somatic variant detection VarScan2 v2.4.4:

Somatic variant detection Neoantigen prediction tools

MixMHCpred v2.1: HLA class I binding affinity prediction NetMHCpan

v4.1: HLA class I binding affinity prediction

PRIME v1.0.1: Antigen presentation and TCR recognition prediction NetMHCstabpan v1.0a: HLA

class I binding stability prediction

NetChop v3.1: Proteasomal processing prediction NetCTLpan

v1.1: TAP transport prediction

CScape (July 2017): Prediction of oncogenicity of mutations

Databases and References

GENCODE Release 38: Genomic annotation

GRCh37 v37: Human reference genome GTEx

v8: Tissue-specific gene expression

TCGA (September 2021): Cancer gene expression data IntOGen:

Database of oncogenic driver genes

ipMSDB: In-house database of HLA-bound peptides (547,476 unique HLA-I peptides)

Data Preprocessing for Neoantigen Prediction:

The preprocess_data.sh program implements a complete data preprocessing workflow for neoantigen prediction that combines intelligent filtering, categorical encoding, and advanced normalization. It first applies the MLRowSelection class to filter data using three biologically justified criteria: it retains only SNV mutations, excludes

peptides with NetMHCpan MHC prediction ranks greater than 20 (indicating low binding affinity), and removes sequences derived from 22 immune system genes (HLA, T cell receptors, etc.). It then uses a custom encoder (CatEncoder) to transform categorical variables. The process includes specific imputation of missing values according to strategies defined by characteristic (maximum, minimum, mode, or constant values), followed by differentiated normalization in NormalizeData.py, which performs quantile transformation for the data used in machine learning and converts the data to uniform distributions between 0-1. Finally, it implements data balancing by subsampling the majority class (limiting non-immunogenic peptides to 100,000 samples) and structures the features into optimized sets for neopeptides (24 features) and mutations (27 features), generating datasets ready for training predictive models with all transformations stored for reproducibility.

For the processing of neopeptides, 24 main features (ml_features_neopep) were used, including:

Patient information and peptide sequences

MHC prediction ranks (MixMHCpred, NetMHCpan, PRIME)

Stability and processing values

Gene expression data (GTEx, TCGA) Oncogenic

driver information (Intogen) Differential affinity

indices (DAI)

And for mutations, 27 main features (ml_features_mutation) were used, with an emphasis on:

Minimum ranks and counts of MHC-bound peptides

Multiple binding affinity measures (EL ranks) Stability

values for the three best predictions Expression

information and oncogenic drivers

Model training:

The model training code implements a robust machine learning workflow for neoantigen immunogenicity prediction with train_classifier.sh, which combines Bayesian hyperparameter optimization, advanced class balancing strategies, and systematic evaluation of multiple algorithms. The system uses the ClassifierManager class to manage five main algorithms (Logistic Regression, Linear and RBF SVM, XGBoost, and CatBoost) with hyperparameter spaces specifically designed for each, including regularization parameters, class weights, and boosting configurations from OptimizationParams.py. Optimization is performed using Hyperopt v0.2.7 with 200 Bayesian search iterations and 10 independent replicates per algorithm, using a custom objective function (sum_exp_rank) that exponentially weights the rankings of true immunogenic neoantigens (α=0.005 for neopeptides, α=0.05 for mutations). The workflow implements 5-fold stratified cross-validation during optimization and handles class imbalance by automatically calculating class weights and specific strategies such as scale_pos_weight in XGBoost. The code includes advanced features such as automatic filtering of irrelevant columns (patient, mutant_seq, wt_seq), intelligent conversion of categorical data types, saving/loading of models with specific compatibility for different frameworks (pickle for sklearn, native methods for XGBoost/CatBoost), and a voting system that combines multiple LR and XGBoost replicas for greater predictive robustness, all executed in parallel with performance and runtime monitoring.

Model Evaluation

The programs test_voting_classifier.sh, test_voting_classifier1.py and CalculateMetricsAndGeneratePDF1.py implement a complete model evaluation workflow that combines prediction with voting classifiers and comprehensive performance analysis with specialized metrics for imbalanced data. test_voting_classifier.py manages the loading and evaluation of multiple classifiers using the load_classifiers() function, which searches for model files using glob patterns, loads two batches of classifiers (typically LR and XGBoost) with different weights, and creates voting classifiers that combine predictions using weighted average probabilities. Processing is executed in parallel per patient using joblib.Parallel, where each patient is evaluated independently by filtering their specific data, excluding non-numeric columns (patient, mutant_seq, wt_seq), and applying the voting classifier to generate immunogenicity probabilities that are sorted in descending order for each sorting. The results are stored in CSV format with detailed information per neoantigen including set

data, patient, predicted probability, actual immunogenicity, and sequences. analysis_and_charts.py processes these results to generate a comprehensive analysis that includes top-k metrics (evaluating how many immunogenic neoantigens are found in the first 20, 50, 100, etc. rankings), traditional classification metrics (ROC-AUC, precision-recall, F1-score, NDCG, MAP, MRR), specific class imbalance analysis with metrics such as balanced accuracy and Cohen's kappa, normalized and absolute confusion matrices, and comparative visualizations between datasets. The system automatically generates a comprehensive PDF report with more than 20 different graphs, metric tables per dataset, class distribution analysis, ROC and precision-recall curves, comparative heat maps, and confusion matrices, providing a comprehensive evaluation of the neoantigen immunogenicity prediction system.

Appendix 3

To develop the Neodb database, researchers used the IEDB (Immune Epitope Database) dated June 19, 2022, as their primary source, focusing specifically on molecular immunogenicity assays with experimental validation. The data collection process was characterized by a rigorous multi-stage filtering protocol to ensure the highest quality of the training set. Initially, filters were applied to select only linear sequences with both positive and negative assay results, excluding B cell assays and MHC assays, restricting the MHC type to Class I, limiting the host to Homo sapiens, and considering only T cell assays based on cytokine release or cytotoxicity. Subsequently, additional cleaning was performed by removing entries without 4-digit HLA alleles or without sufficiently explicit experimental information, establishing strict criteria for samples (more than four negative experiments to classify a sample as negative, and at least one responder subject for positive samples), and discarding redundant instances of peptide-HLA complexes. The final training set consisted of 8,412 examples, with 3,467 positive samples (41.2%) and 4,945 negative samples (58.8%)([34]).

The NEPdb database represents a meticulously curated collection of experimentally validated neoantigens in human cancer, employing a semi-automated curation process to ensure reliability. Initially, a programmatic literature search was performed using natural language processing techniques with specific keywords such as "cancer immunotherapy," "neoepitope," "neoantigen," and "immunogenicity." Subsequently, manual filtering of the 848 resulting articles (published between January 2000 and

March 2022) to retain only those with neoantigen data explicitly validated by experimentation. Each entry in the database includes comprehensive information: tumor type, gene symbol, peptide sequences (mutant and wild-type), corresponding HLA allele, assay details (T cell source, APC source, antigen type), clinical trial information, and bibliographic details, thus constituting a valuable data source with 173 immunogenic neoepitopes and 17,376 experimentally validated ineffective neopeptides ([38]) .

The ITSNdb (Immunogenic Tumor Specific Neoantigen database) contains exclusively 199 neoantigens of 9 and 10 amino acids derived from SNV mutations, with complete information on mutant and wild-type sequences, and most importantly, with rigorous experimental validation at three critical levels: (1) confirmed positive cell processing, (2) experimentally validated binding to MHC-I molecules (by mass spectrometry or binding competition assays), and (3) verified positive or negative immunogenic response (by tetramer titration or IFN-γ or TNF ELISPOT assays). This unique approach allows immunogenicity to be assessed without interference from factors such as binding affinity and processing (already verified), thus constituting a valuable tool for the development and objective evaluation of computational predictors. The database contains 129 immunogenic and 70 non-immunogenic neoantigens, with a predominance of the HLA-A*02:01 allele (40.2% of peptides) ([37]).

Appendix 4

The previous DL databases were extracted from the documentation provided by each contributor in Excel and CSV files in order to create a larger and more consistent database, combining different types of immunological validation and sources. A three-stage workflow was executed in the DL_dataset_obtention folder. This workflow processes peptide data from multiple sources to create a unified dataset with immunogenicity characteristics calculated using computational prediction tools. The process must be run sequentially starting with merge_new_dataset.py, which merges and standardizes four different databases (NeodbTrain.xlsx, NeodbTest.xlsx, NEPdb.csv, ITSNdb.csv) into a unified file called merged.csv, normalizing column names to Seq, Immunogenicity, and HLA, and converting immunogenicity values to binary format. Next, Filtrado.py is run, which takes the merged.csv file and processes it to filter sequences by length between 8-12 amino acids, identifies and reports duplicate sequences with conflicting values immunogenicity eliminates duplicates maintaining only

unique sequence-immunogenicity combinations, and generates the filtered_merged.csv file along with count statistics per source file. Finally, feature_calculator_v3.py processes each peptide in the dataset individually by running four computational prediction tools: NetMHCpan for HLA-peptide binding prediction (EL_Rank), NetMHCstabpan for complex stability prediction (mut_Rank_Stab), MixMHCpred for presentation prediction (mutant_rank), and PRIME for immunogenicity prediction (mutant_rank_PRIME), saving progress every 10 processed sequences and generating detailed processing logs in the final merged_with_features.csv file.

It is important to have the prediction tools installed, which could not be attached because they were too large, but each tool can be downloaded from the corresponding GitHub repository. It is also important to configure the paths in the code correctly, and bear in mind that the last step may take several hours depending on the size of the dataset due to the computational resources required by the prediction tools.

The DL+ML_dataset_obtention.py file is a complete data processing workflow for creating a clean dataset of immunogenic neopeptides. The program automates five sequential steps: first, it filters and cleans the initial data by removing duplicates and sequences outside the range of 8-12 amino acids, then processes and maps the Neopep data (including mutant sequences, immune response types, and computational features), merges both datasets while maintaining standard columns, updates missing HLA values using patient information from reference files, and finally cleans the CSV format by correcting formatting issues caused by commas in the HLA fields. The result is a final merged_neopep_data_clean.csv file containing structured data ready for machine learning analysis, with complete information on peptide sequences, immunogenicity, HLA alleles, and predictive computational features.

Appendix 5

As can be seen in Figures S1, S2, and S3, the best metrics for the minority class are those obtained by filtering the top 10,000 most immunogenic neoantigens, which are superior to those obtained from the top 1,000 and 100,000 in all cohorts.

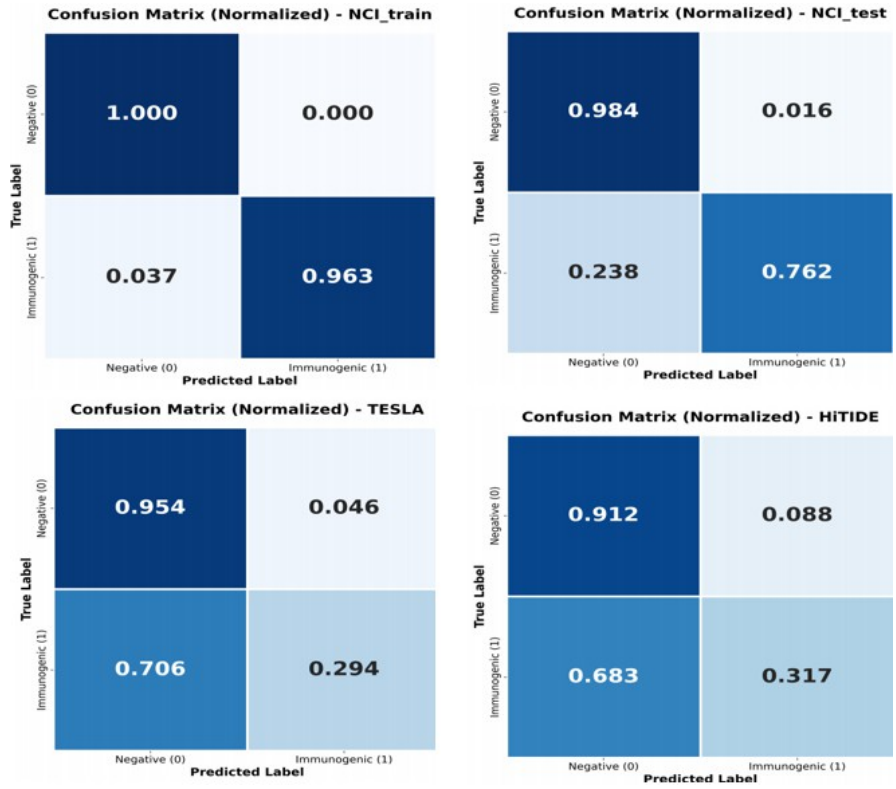

Figure S1. Confusion matrices in the top 100,000 subsampling in HPPN

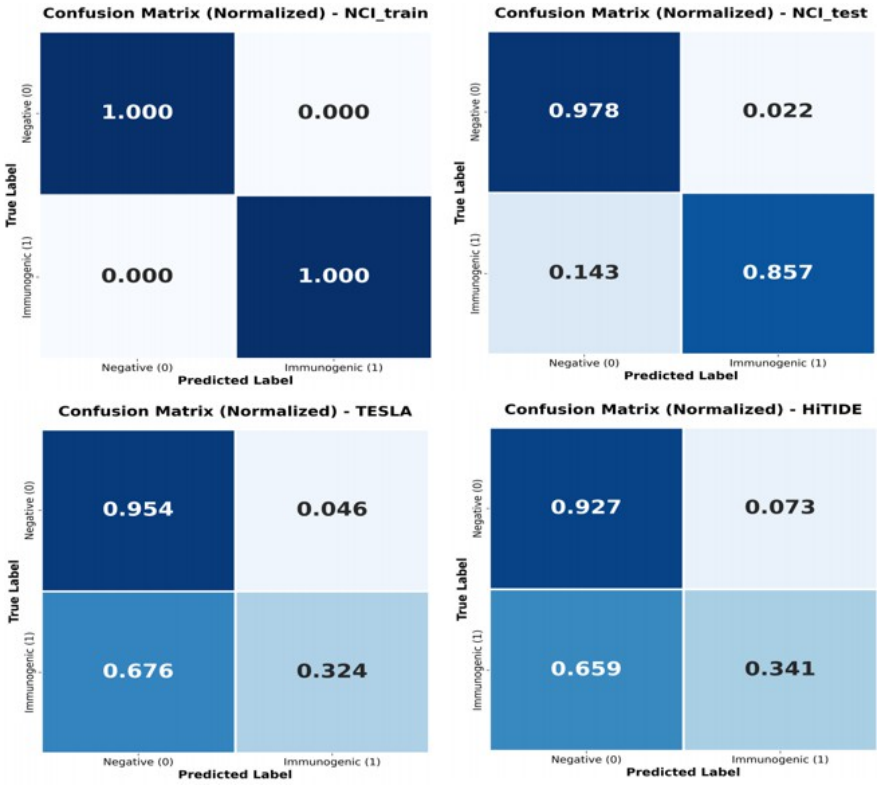

Figure S2. Confusion matrices in the top 10,000 subsampling in HPPN

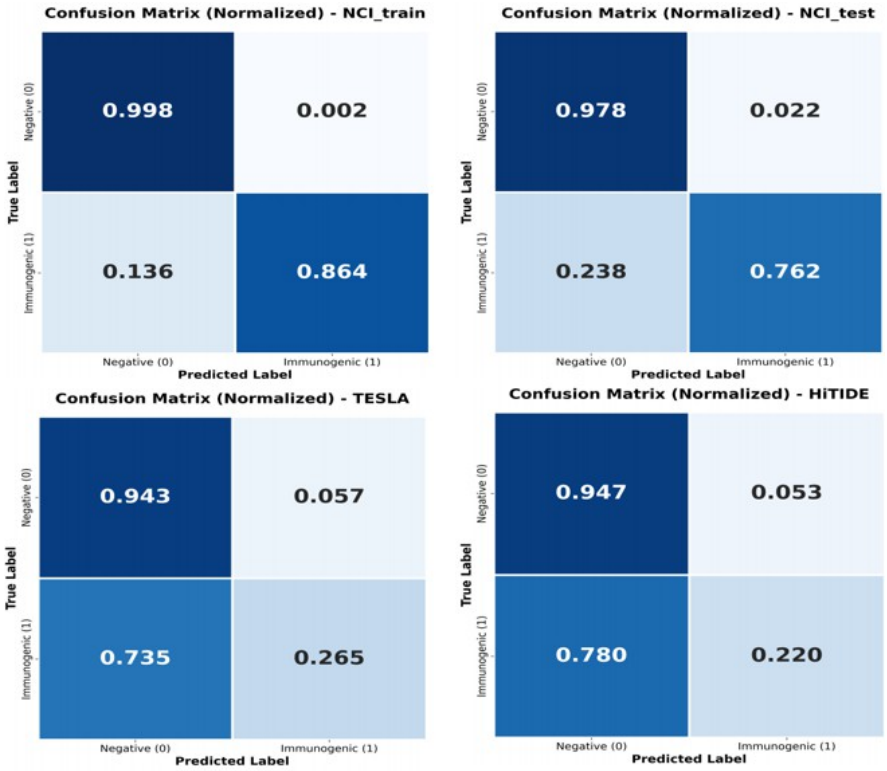

Figure S3. Confusion matrices at the top of subsampling 1,000 in HPPN

Appendix 6

As can be seen in Figures S4 and S5, the confusion matrices in the top 100,000 show better metrics than in the top 10,000.

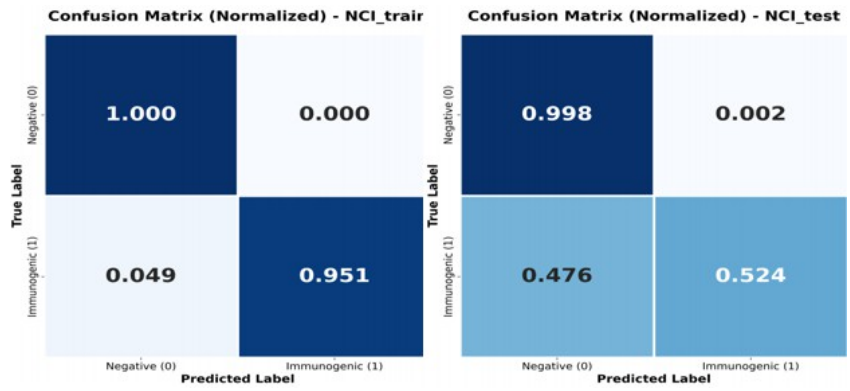

Figure S4. Confusion matrices in the top 100,000 subsample in LPPN

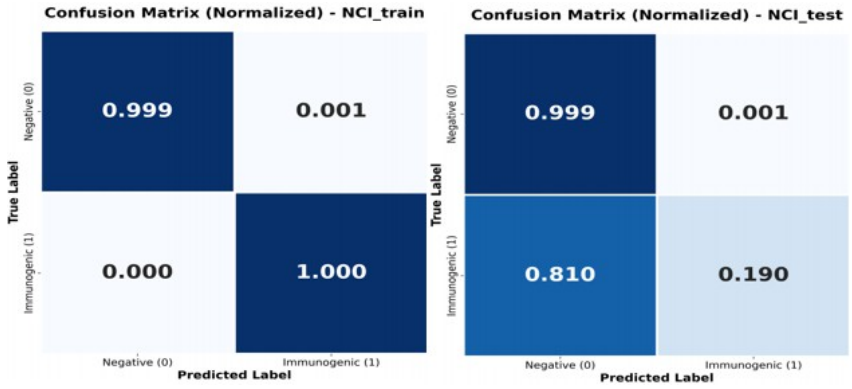

Figure S5. Confusion matrices in the top 10,000 subsamples in LPPN

Appendix 7

Figure S6 shows an improvement in NDCG quality when rounding to the first decimal place. Figure S8 shows an increase in top performance when rounding to the first decimal place, while Figures S7 and S9 show no change with rounding to the first decimal place.

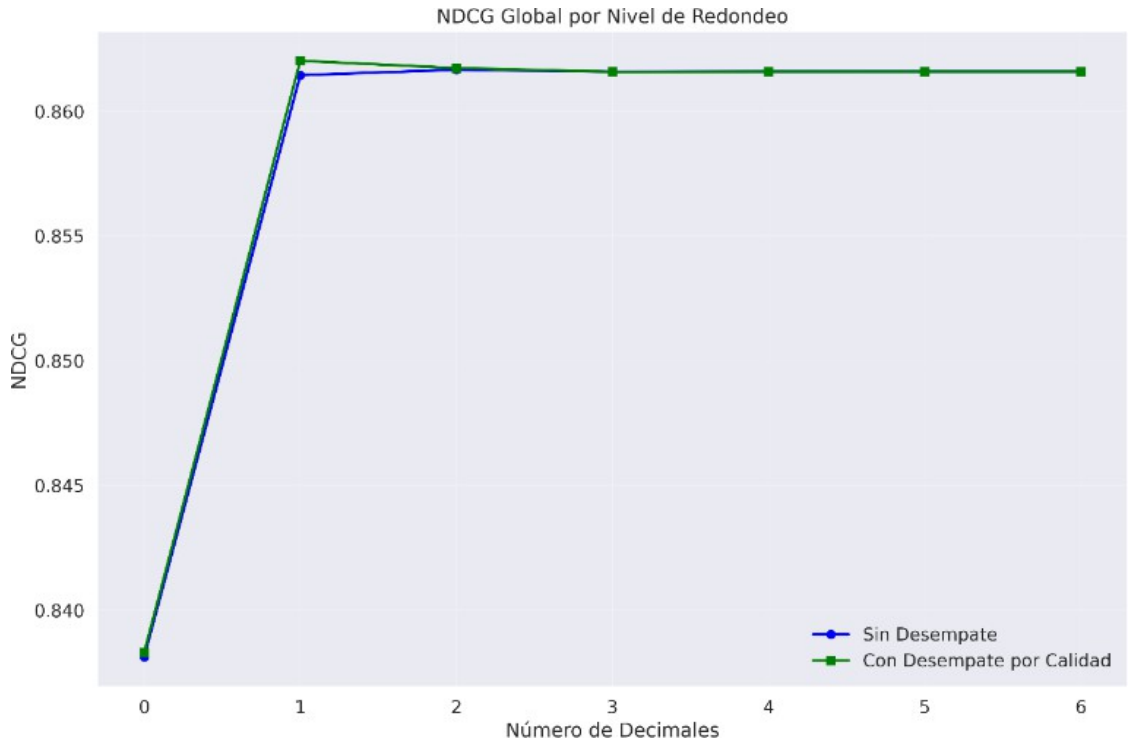

Figure S6. NDCG of the model rounding to different decimal places and tie-breaking by quality.

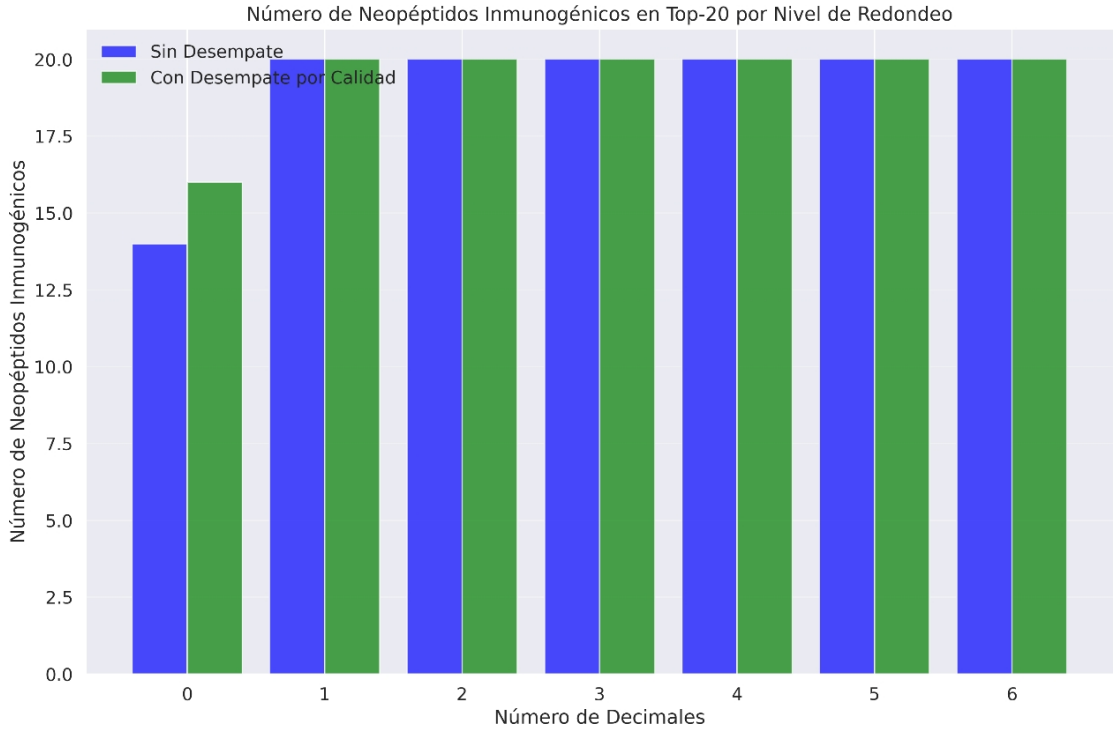

Figure S7. Top 20 HPPN at different roundings with different numbers of decimal places

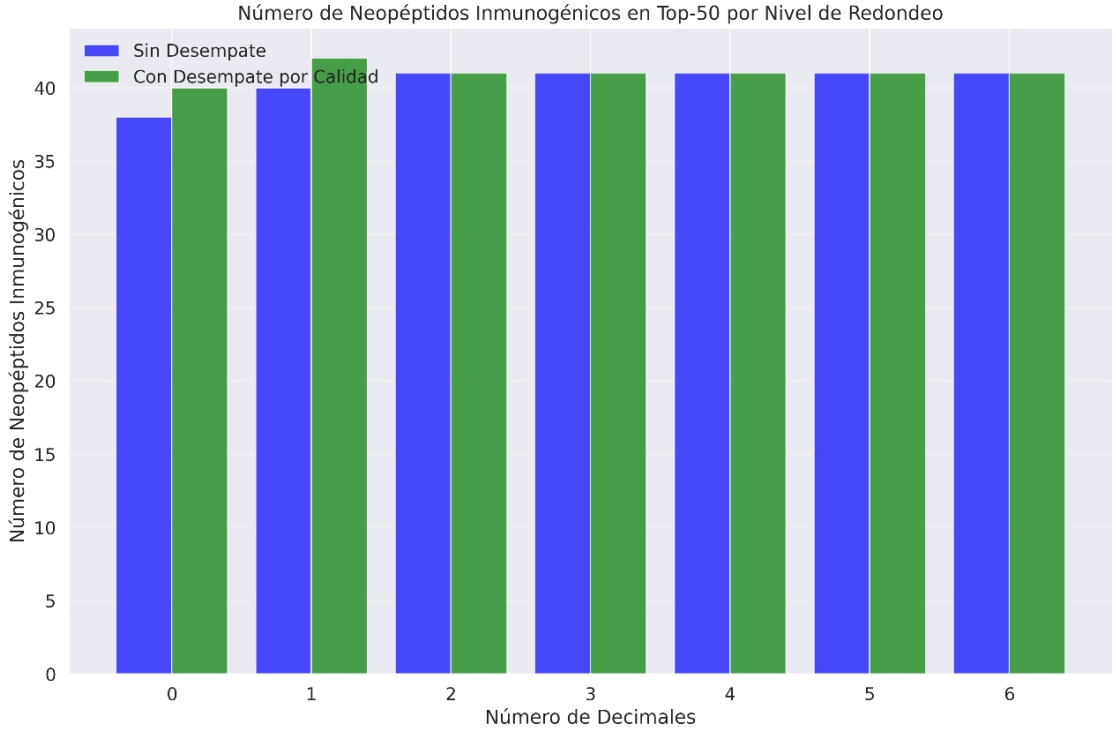

Figure S8. Top 100 HPPN at different roundings with different numbers of decimal places

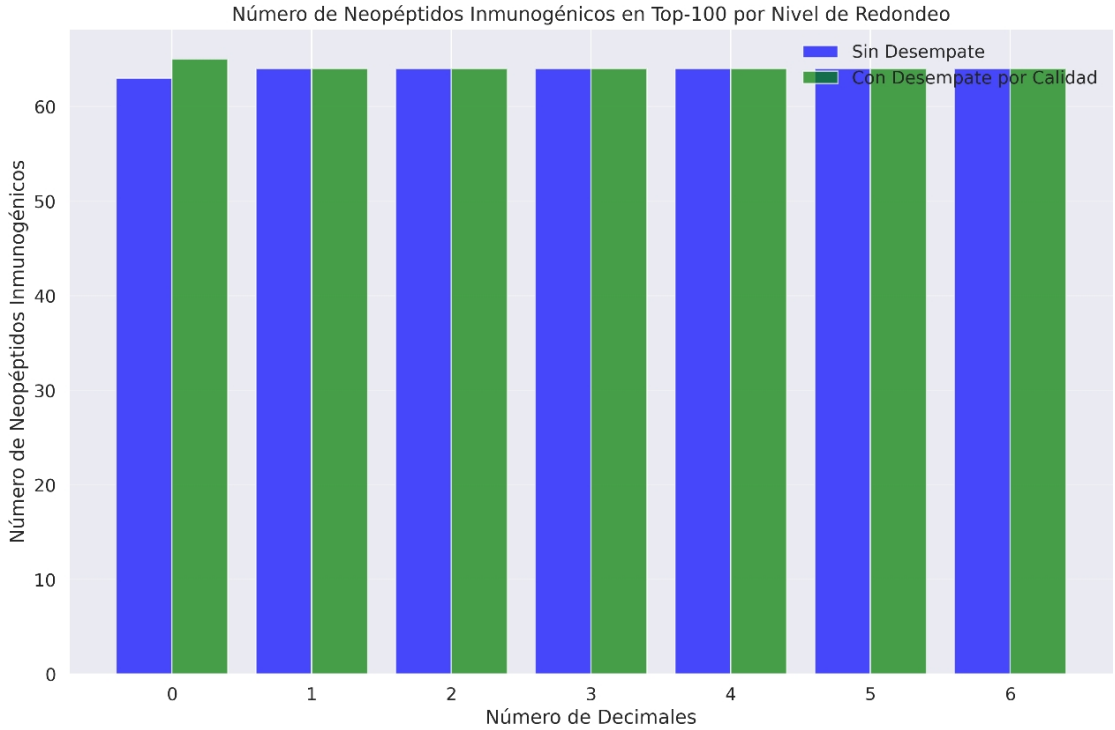

Figure S9. Top 100 HPPN at different roundings with different numbers of decimal places

Appendix 8

Tables S1 and S2 below show that in HPPN and LPPN, MUC6 is the most frequent gene in the final rankings, unlike KRAS, which would be expected.

Table S1. Most mutated genes in the top 10, 20, and 30 most immunogenic neoantigens per patient according to the final HPPN ranking. N corresponds to the number of times it appeared in each top of all patients and F to the same data in percentage.

Position	Gene	N	N20	N	F10	F20	F30	Total
1	MUC6	105	235	343	50.0	47	44.8	683
2	TAS2R31	16	41	53	7.6	8.2	6.9	110
3	TAS2R46	6	28	47	2.9	5.6	6.1	81
4	CTAGE6	14	26	40	6.7	5.2	5.2	80
5	MUC12	6	20	40	2.9	4.0	5.2	66
6	SLC9B1	8	19	34	3.8	3.8	4.4	61

7	KRT18	2	16	42	1.0	3.2	5.5	60
8	FRG1	4	12	24	2.9	2.4	3.1	40
9	LRRC37A3	7	15	18	3.3	3.0	2.3	40
10	PABPC1	7	13	18	3.3	2.6	2.3	38
11	MUC3A	4	13	17	1.9	2.6	2	34
12	KRAS	7	10	15	3.3	2	2	32
13	HRNR	3	8	12	1.4	1.6	1.6	23
14	LILRA1	4	7	9	1.9	1.4	1.2	20
15	MUC17	0	8	8	0	1.6	1	1

Table S2. Most mutated genes in the top 10, 20, and 30 most immunogenic neoantigens per patient according to the final LPPN ranking. N corresponds to the number of times it appeared in each top of all patients and F to the same data in percentage.

Position	Gene	N	N20	N	F10	F20	F30	Total
1	MUC6	99	195	278	50.8	46.3	42.4	572
2	KRT18	3	39	56	1.5	9.3	8.5	98
3	SLC9B1	7	20	49	3.6	4.8	7.5	76
4	TAS2R46	5	21	40	2.6	5.0	6.1	66
5	MUC12	6	22	29	3.1	5.2	4.4	57
6	CTAGE6	13	19	23	6.7	4.5	3.5	55
7	TAS2R31	10	15	27	5.1	3.6	4.1	52
8	TAS2R19	12	16	24	6.2	3.8	3.7	52
9	PABPC1	11	13	16	5.6	3.1	2.4	40
10	HRNR	3	7	23	1.5	1.7	3	33

Table S2. Most mutated genes in the top 10, 20, and 30 most immunogenic neoantigens per patient according to the final LPPN ranking. N corresponds to the number of times it appeared in each top list of all patients, and F corresponds to the same data in percentage.

Position	Gene	N	N	N	F10	F20	F30	Total
11	LRRC37A3	6	11	12	3.1	2.6	1.8	29
12	MUC3A	0	4	13	0.0	1.0	2	17
13	FRG1	2	4	8	1.0	1	1	14
14	LILRA1	2	2	8	1.0	0.5	1.2	12
15	PABPC3	2	4	6	1	1	0.9	12

References:

1. Storz, P. & Crawford, H. C. Carcinogenesis of Pancreatic Ductal Adenocarcinoma. *Gastroenterology* **158**, 2072–2081 (2020).
2. Ying, H. *et al.* Genetics and biology of pancreatic ductal adenocarcinoma. *Genes Dev.* **30**, 355–385 (2016).
3. Luchini, C., Capelli, P., & Scarpa, A. Pancreatic Ductal Adenocarcinoma and Its Variants. *Surg. Pathol. Clin.* **9**, 547–560 (2016).
4. Schawkat, K., Manning, M. A., Glickman, J. N. & Mortele, K. J. Pancreatic Ductal Adenocarcinoma and Its Variants: Pearls and Perils. *RadioGraphics* **40**, 1219–1239 (2020).

5. Halbrook, C. J., Lyssiotis, C. A., Pasca Di Magliano, M., & Maitra, A. Pancreatic cancer: Advances and challenges. *Cell* **186**, 1729–1754 (2023).
6. Elbanna, K. Y., Jang, H.-J., & Kim, T. K. Imaging diagnosis and staging of pancreatic ductal adenocarcinoma: a comprehensive review. *Insights Imaging* **11**, 58 (2020).
7. Bowman, A. W. & Bolan, C. W. MRI evaluation of pancreatic ductal adenocarcinoma: diagnosis, mimics, and staging. *Abdom. Radiol.* **44**, 936–949 (2019).
8. Vellan, C. J. *et al.* Application of Proteomics in Pancreatic Ductal Adenocarcinoma Biomarker Investigations: A Review. *Int. J. Mol. Sci.* **23**, 2093 (2022).
9. Adamska, A., Domenichini, A., & Falasca, M. Pancreatic Ductal Adenocarcinoma: Current and Evolving Therapies. *Int. J. Mol. Sci.* **18**, 1338 (2017).
10. Wei, H. & Ren, H. Precision treatment of pancreatic ductal adenocarcinoma. *Cancer Lett.* **585**, 216636 (2024).
11. Sarantis, P., Koustas, E., Papadimitropoulou, A., Papavassiliou, A. G., & Karamouzis, M. V. Pancreatic ductal adenocarcinoma: Treatment hurdles, tumor microenvironment and immunotherapy. *World J. Gastrointest. Oncol.* **12**, 173–181 (2020).
12. Chen, H. *et al.* Neoantigen-based immunotherapy in pancreatic ductal adenocarcinoma (PDAC). *Cancer Lett.* **490**, 12–19 (2020).
13. Peng, H. *et al.* Combination TIGIT/PD-1 blockade enhances the efficacy of neoantigen vaccines in a model of pancreatic cancer. *Front. Immunol.* **13**, 1039226 (2022).
14. Rashed, A., Grabocka, J., & Schmidt-Thieme, L. A Guided Learning Approach for Item Recommendation via Surrogate Loss Learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.* 605–613 (ACM, Virtual Event Canada, 2021). doi:10.1145/3404835.3462864.
15. Xie, N. *et al.* Neoantigens: promising targets for cancer therapy. *Signal Transduct. Target. Ther.* **8**, 9 (2023).

16. Peng, M. *et al.* Neoantigen vaccine: an emerging tumor immunotherapy. *Mol. Cancer* **18**, 128 (2019).

17. Zhou, S., Liu, S., Zhao, L., & Sun, H.-X. A Comprehensive Survey of Genomic Mutations in Breast Cancer Reveals Recurrent Neoantigens as Potential Therapeutic Targets. *Front. Oncol.* **12**, 786438 (2022).

18. Pishesha, N., Harmand, T. J., & Ploegh, H. L. A guide to antigen processing and presentation. *Nat. Rev. Immunol.* **22**, 751–764 (2022).

19. Jiang, N. *et al.* Two-Stage Cooperative T Cell Receptor-Peptide Major Histocompatibility Complex-CD8 Trimolecular Interactions Amplify Antigen Discrimination. *Immunity* **34**, 13–23 (2011).

20. Müller, M. *et al.* Machine learning methods and harmonized datasets improve immunogenic neoantigen prediction. *Immunity* **56**, 2650-2663.e6 (2023).

21. Zhang, W. *et al.* Personal Neoantigens From Patients With NSCLC Induce Efficient Antitumor Responses. *Front. Oncol.* **11**, 628456 (2021).

22. Chen, P. *et al.* Dominant neoantigen verification in hepatocellular carcinoma by a single-plasmid system coexpressing patient HLA and antigen. *J. Immunother. Cancer* **11**, e006334 (2023).

23. Rojas, L. A. *et al.* Personalized RNA neoantigen vaccines stimulate T cells in pancreatic cancer. *Nature* **618**, 144–150 (2023).

24. Liang, K.-L. & Azad, N. S. Immune-Based Strategies for Pancreatic Cancer in the Adjuvant Setting. *Cancers* **17**, 1246 (2025).

25. Leidner, R. *et al.* Neoantigen T-Cell Receptor Gene Therapy in Pancreatic Cancer. *N. Engl. J. Med.* **386**, 2112–2119 (2022).

26. Leidner, R. *et al.* Neoantigen T-Cell Receptor Gene Therapy in Pancreatic Cancer. *N. Engl. J. Med.* **386**, 2112–2119 (2022).

27. Frank, R. & Hargreaves, R. Clinical biomarkers in drug discovery and development. *Nat. Rev. Drug Discov.* **2**, 566–580 (2003).

28. Lang, F., Schrörs, B., Löwer, M., Türeci, Ö., & Sahin, U. Identification of neoantigens for individualized therapeutic cancer vaccines. *Nat. Rev. Drug Discov.* **21**, 261–282 (2022).

29. Cai, Y. *et al.* Artificial intelligence applied in neoantigen identification facilitates personalized cancer immunotherapy. *Front. Oncol.* **12**, 1054231 (2023).

30. De Mattos-Arruda, L. *et al.* Neoantigen prediction and computational perspectives towards clinical benefit: recommendations from the ESMO Precision Medicine Working Group. *Ann. Oncol.* **31**, 978–990 (2020).

31. Tang, Y. *et al.* TruNeo: an integrated pipeline improves personalized true tumor neoantigen identification. *BMC Bioinformatics* **21**, 532 (2020).

32. Bulashevska, A. *et al.* Artificial intelligence and neoantigens: paving the path for precision cancer immunotherapy. *Front. Immunol.* **15**, 1394003 (2024).

33. Hollmann, N. *et al.* Accurate predictions on small data with a tabular foundation model. *Nature* **637**, 319–326 (2025).

34. Wu, T. *et al.* Neodb: a comprehensive neoantigen database and discovery platform for cancer immunotherapy. *Database* **2023**, baad041 (2023).

35. Jiang, D. *et al.* NeoaPred: a deep-learning framework for predicting immunogenic neoantigen based on surface and structural features of peptide–human leukocyte antigen complexes. *Bioinformatics* **40**, btae547 (2024).

36. Łuksza, M. *et al.* Neoantigen quality predicts immunoediting in survivors of pancreatic cancer. *Nature* **606**, 389–395 (2022).

37. Nibeyro, G. *et al.* Unraveling tumor specific neoantigen immunogenicity prediction: a comprehensive analysis. *Front. Immunol.* **14**, 1094236 (2023).

38. Xia, J. *et al.* NEPdb: A Database of T-Cell Experimentally-Validated Neoantigens and Pan-Cancer Predicted Neoepitopes for Cancer Immunotherapy. *Front. Immunol.* **12**, 644637 (2021).

39. Rashed, A., Grabocka, J., & Schmidt-Thieme, L. A Guided Learning Approach for Item Recommendation via Surrogate Loss Learning. In *Proceedings of the 44th International*

ACM SIGIR Conference on Research and Development in Information Retrieval 605–613 (ACM, Virtual Event Canada, 2021). doi:10.1145/3404835.3462864.

40. Gaona-Cuevas, M., Bucheli Guerrero, V., & Vera-Rivera, F. H. The Smart Product Backlog: A Classification Model of User Stories. *IEEE Access* **12**, 150008–150019 (2024).

41. Chen, F. *et al.* Neoantigen identification strategies enable personalized immunotherapy in refractory solid tumors. *J. Clin. Invest.* **129**, 2056–2070 (2019).

42. Bhatt, C. M., Patel, P., Ghetia, T., & Mazzeo, P. L. Effective Heart Disease Prediction Using Machine Learning Techniques. *Algorithms* **16**, 88 (2023).

43. Soni, T., Gupta, G., & Dutta, M. A Comparative Analysis of Decision Trees, Random Forests, and XGBoost for Enhanced Crop Recommendation. In *2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS)* 626–630 (IEEE, Tashkent, Uzbekistan, 2024). doi:10.1109/ICTACS62700.2024.10841044.

44. Fadhlullah, A. F. & Widiyaningtyas, T. Comparative Analysis of Decision Tree and Random Forest Algorithms for Diabetes Prediction. *JTAM J. Teori Dan Apl. Mat.* **8**, 1121 (2024).

45. Zhang, D. *et al.* A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *IEEE Access* **6**, 21020–21031 (2018).

46. Akinola, S., Leelakrishna, R., & Varadarajan, V. Enhancing cardiovascular disease prediction: A hybrid machine learning approach integrating oversampling and adaptive boosting techniques. *AIMS Med. Sci.* **11**, 58–71 (2024).

47. Swain, S., Mohanty, M. N., & Pattanayak, B. K. Precision medicine in hepatology: harnessing IoT and machine learning for personalized liver disease stage prediction. *Int. J. Reconfigurable Embed. Syst. IJRES* **13**, 724 (2024).

48. Zhang, C. *et al.* Hybrid Metric K-Nearest Neighbor Algorithm and Applications. *Math. Probl. Eng.* **2022**, 1–15 (2022).

49. Gallego, A. J., Rico-Juan, J. R., & Valero-Mas, J. J. Efficient k-nearest neighbor search based on clustering and adaptive k values. *Pattern Recognit.* **122**, 108356 (2022).

50. Li, Y., Ammari, S., Balleyguier, C., Lassau, N., & Chouzenoux, E. Impact of Preprocessing and Harmonization Methods on the Removal of Scanner Effects in Brain MRI Radiomic Features. *Cancers* **13**, 3000 (2021).

51. Zeng, M., Zou, B., Wei, F., Liu, X., & Wang, L. Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. In *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)* 225–228 (IEEE, Chongqing, China, 2016). doi:10.1109/ICOACS.2016.7563084.