TrOnco: A bioinformatic approach to predict tumorigenesis on de novo genomic fusions

Puchal-Batriu M[1], García de la Torre A [2], Cicala CM, Mas A, Elliott A, Serrano C, Muñoz-Torres PM and Nonell L

[1]*Master of Omics Data Analysis, Universitat de Vic - Universitat Central de Catalunya, Vic, Spain*

[2]*Vall d'Hebron Institute of Oncology, Barcelona, Spain*

marti.puchal@gmail.com, paumunoz@vhio.net

Abstract: Chromosomal translocations are structural rearrangements that can act as oncogenic events, triggering cancer –or its progression- by generating fusion proteins, altering gene expression, or generating genomic variability. These alterations often disrupt key cellular processes such as cell cycle control. Identifying oncogenic translocations -those with functional consequences that promote tumorgenesis- is critical for understanding cancer biology and developing targeted therapies. Efforts have been made to address this topic; methods such as DEEPrior and Oncofuse use deep learning models to analyze fusion protein sequences or their resulting domain composition. Here, we present TrOnco, a novel machine learning tool based on random forest, xgboost and a CNN. TrOnco is designed to classify translocations as oncogenic or nononcogenic by integrating multiple omics data, including genomics, transcriptomics, or proteomics. The method was trained not only on oncogenic translocations from COSMIC, but also on a dataset of fusions identified in healthy control samples (Babiceanu et al., 2016). Overall, TrOnco provides a predictive framework to prioritize high-impact translocations for further experimental validation and clinical investigation. TrOnco is an open-source Python program. Source code, documentation, and installation instructions can be downloaded from https://github.com/martipuchal/TrOnco/

1 INTRODUCTION

Cancer is a leading cause of death, with an increase in prevalence in recent years. However, when we talk about cancer, we refer to more than 277 diseases with different causes, effects, and progressions. This variability makes early diagnosis or treatment difficult (Hassanpour and Dehghani, 2017). To understand the complexity of cancer, it is important to explore its causes. The most common cause is genomic mutation, which leads to uncontrolled cell proliferation. Such mutations can affect a single nucleotide or a larger chromosomal region (chromosomal aberrations), altering the structure and function of the resulting protein. At a single nucleotide level, usually only one protein is altered, and in some cases only partially. The consequences, however, vary greatly depending on the protein involved. With chromosomal aberrations, broader genomic changes occur, and several proteins may be altered. In this article, we focus on translocations, a chromosomal aberration consisting in the shift of a fragment of a chromosome to another part of the same chromosome or to a different one (Rabbitts, 1994). In general, multiple mutations are required for a tumor to develop. Some mutations play a more active role in tumorigenesis, and this must be considered in cancer analysis.

The correlation between fusions and cancer began with the discovery in the 1980s of the BCR-ABL fusion and the MYC-immunoglobulin heavy chain fusion in tumoral tissue. These findings marked the beginning of the study of chromosomal aberrations, and in the following years knowledge of fusions increased significantly (Mitelman et al., 2007). This progress was mainly due to the greater availability of data over the last 20 years, particularly with the creation of COSMIC (Catalogue of Somatic Mutations in Cancer) by the Cancer Programme at the Wellcome Sanger Institute, which compiled and curate all fusion reports, resulting in a complete database of cancer fusions. However, the role of the resulting chimeric protein from these fusions is still unknown for most cases (Kloosterman et al., 2017).

Despite these cases, most fusions might not play an active role in tumor formation or have a significant biological impact. However, when focusing on

cancer, it is important to understand the oncogenic capacity of gene fusions, which can also have significant implications for patient care and the drug sensibility of the tumor. (Annala et al., 2013).

Numerous tools have been developed to analyze gene fusions, each using different types of input data and offering varying levels of reliability in their results. Some methods, such as FusionInspector, analyze fusions directly from RNA-seq FASTQ files, using a target file that contains the fusions of interest (Haas et al., 2023). Others, such as Oncofuse, DEEPrior, and Pegasus, rely on genomic positions to analyze fusions.

For analyzing genomic positions, two main machine learning approaches were used: CART and deep learning. Machine learning, a subfield of artificial intelligence, develops algorithms capable of identifying patterns and making predictions. With training data from known fusions, supervised learning methods can predict the nature of future cases (Singh et al., 2016). CART is a tree-based method that analyzes fusions through decision rules, while deep learning employs multi-layered neural networks to capture complex patterns. The architecture of these networks, inspired by the neuron association in the brain, allows interconnected neurons to generate outputs from layered interactions. Each model has distinct strengths: CART methods are simpler to implement, whereas deep learning—when properly tuned—can uncover more intricate relationships in the data.

Moreover, the different programs also differ in their procedures for analyzing fusions. Pegasus uses the protein domains retained and lost, along with the function of the chimeric protein, to obtain a prediction with a gradient boosting–based machine learning method (Abate et al., 2014). Oncofuse focuses on the chimeric protein, analyzing the domains retained and lost and the protein interactions, while also incorporating Gene Ontology terms and tissue-specific expression of the genes involved. To integrate this data, it uses a Bayesian neural network based on a deep learning model (Shugay et al., 2013). DEEPrior, although also focused on the chimeric protein, analyzes only the protein sequence. In contrast, DEEPrior focus on the analysis of the resulting protein from the gene fusion but only taking into account the sequence of the protein of interest. Using a deep learning model, it predicts the oncogenic potential of the fusion based solely on protein sequence (Lovino et al., 2020).

The main objectives of this project were threefold. First, we translated Oncofuse, the most widely cited program for fusion analysis, from Groovy to Python. Second, given that Oncofuse was originally developed in 2013 with outdated training data and reference databases, we updated these resources to enhance predictive accuracy. Third, we developed an easy-to-use retraining pipeline to allow seamless integration of new datasets as they become available. Building on these objectives, we introduce TrOnco, a Python-based tool designed to analyze gene fusions and chromosomal aberrations—whether arising within the same chromosome or across different chromosomes—and to predict their oncogenic potential.

2 IMPLEMENTATION

To build TrOnco, we used Oncofuse as our main reference. Among the previously mentioned tools, Oncofuse presented a particularly interesting approach to fusion analysis: by integrating domain and protein interaction data and Gene Ontology (GO) annotations, it was able to analyze for each fusion, not only the resulting chimeric protein but also the biological role of the genes involved in the fusion. The complete source code can be found on GitHub in addition with a detailed description of all the functions of TrOnco.

The initial objective of the project was to translate Oncofuse—originally written in Groovy, a now deprecated Java-based language—into Python, the most widely used programming language in bioinformatics. During this process, we also updated the databases and libraries, including the Gene Ontology resource, to ensure compatibility with current standards. The version of Oncofuse used as a reference for developing TrOnco was the latest available release (February 8th, 2016).

Moreover, we added complementary data and features to improve the analysis. As with Oncofuse, we consider the tissue of origin of the fusion to analyze the gene expression of the involved genes. We also expanded the number of tissues of origin by including endometrial tissue in the libraries. In addition, three different methods were added to predict the oncogenic potential of the fusions.

2.1 General schema of the program:

TrOnco is a Python-based program for analyzing fusions, integrating multiple omics data with machine learning and deep learning algorithms. TrOnco uses genomic positions and tissue information to analyze fusions. This tissue information can be provided via the command line or as a column in a TSV file.

The main process of TrOnco can be divided into three main steps, Figure 1:

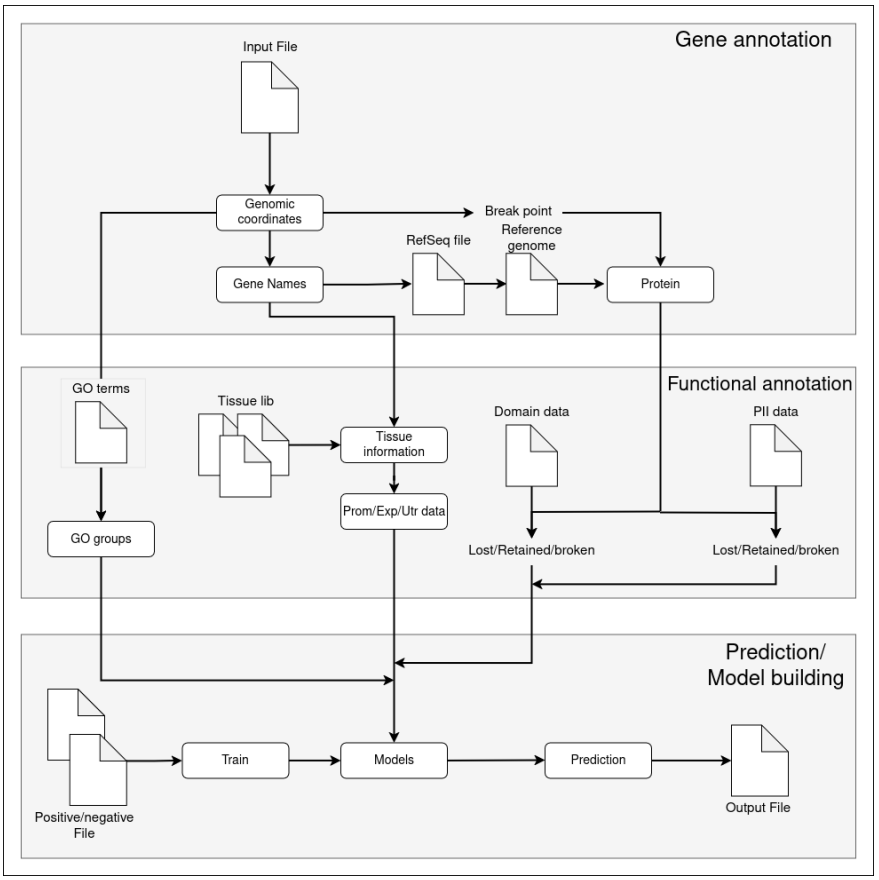

Figure 1: Global overview of the pipeline developed in this article. In the diagram, three distinct phases are identified: Gene annotation, where the genomic positions of the fusions are annotated as genes, and from these genes the protein sequence can be retrieved. Functional annotation, where the fusions are analyzed to obtain the features used for the models. And model building and prediction, where the analysis is applied to classify the fusions according to their role in tumorigenesis.

First the gene annotation step, where genes involved in the fusion are annotated using either the genomic position of the breakpoint or two additional inputs, one for the 5' gene and another for the 3' gene. Using the RefSeq database, we retrieve genomic information such as chromosomal positions of exon start and end and the coding DNA sequence (CDS). This information is used to build the chimeric protein and determine the length of the 5' and 3' transcripts.

Second the functional annotation step, TrOnco uses the gene names to retrieve gene expression information from tissue libraries. For each tissue, a folder contains gene expression in TPM, promoter activity from ChIP-seq data, and UTR data for each gene. This information helps characterize the fusion, capturing the role of the promoter and the expression levels of the 5' and 3' genes. In TrOnco, tissue-specific folders are available for four tissue types—Epithelial, Hematological, Mesenchymal, and Endometrial—as well as an average profile that can be used when the tissue of origin is unknown or not represented.

Additionally, functional information of the annotated genes is retrieved using Gene Ontology (GO) terms. As described in the section below, a table is built with genes and their associated GO terms. Following Oncofuse, groups of GO terms associated with cancer are identified, including transcription cofactors, GTPases, helicase/histone modifiers, kinase activity, protein binding, and transcription factors. TrOnco improves the classification method by counting the number of GO terms separately for each gene. This approach emphasizes genes with GO terms linked to cancer, under the assumption that not only the chimeric protein but also the individual genes can influence the fusion's function.

On the protein level, domain and protein interaction analyses are performed, following the Oncofuse methodology. Protein length from the 5' and 3' genes is used to determine domain retention, loss, or breaking: a domain is considered broken if the breakpoint is in its middle, lost if it is after the breakpoint in the 5' protein or before the breakpoint in the 3' protein,

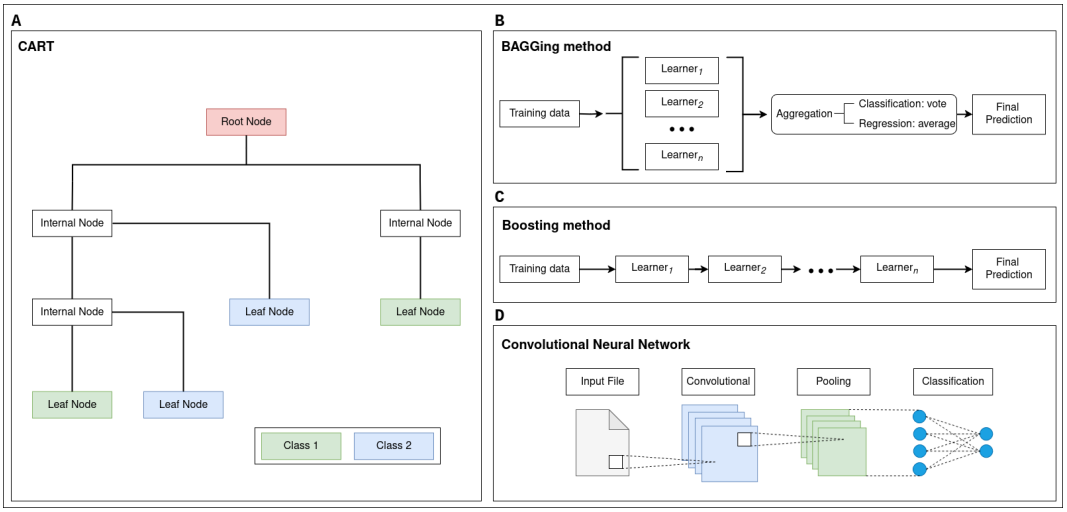

Figure 2: Schema of the machine learning methods and techniques. **A:** CART diagram: each node represents a decision rule, and terminal nodes (leaves) show the final predicted class or value. **B:** Bagging ensemble learning diagram: the estimators—each being a CART—are built independently and then merged to generate the final predictive model. **C:** Boosting ensemble learning diagram: the estimators are built by taking into account the previously generated ones, and through successive iterations a final predictive model is obtained. **D:** Schema of the main steps in a Convolutional Neural Network: by combining a convolutional layer, a pooling layer, and a classification layer, a CNN is able to extract patterns from the data.

and retained if it is before the breakpoint in the 5' protein or after the breakpoint in the 3' protein. Protein interactions are treated similarly: if the region associated with an interaction is preserved in the chimeric protein, the interaction is considered maintained.

The characterization of fusions uses previously obtained information: gene expression, promoter expression, and UTRs of both genes, combined with protein domains and interactions of the encoded proteins. The count of Gene Ontology groups for each gene is also considered. This generates multiple variables with complex correlations and hidden patterns, making machine learning methods suitable for prediction.

The final step is prediction, where the compiled data serve as input for Random Forest, XGBoost, and a The Convolutional Neural Network (CNN). In contrast to Oncofuse, TrOnco implements three approaches, extending its capabilities. Each method has strengths and limitations, and combining them improves adaptability across datasets.

Random Forest (scikit-learn version 0.24.2) was chosen as the easiest to implement with good performance. Random Forest is based on Classification and Regression Trees (CART) and uses a bagging method to optimize predictions. CART predicts the value of a variable using multiple input variables, working as a decision tree where each split is based on a predictor variable, generating a node with a prediction for each branch. Nodes are split into sub-nodes according to a variable and a threshold, repeated a predefined number of times (Figure 2). Bagging involves constructing multiple independent estimators (CART-bsed) and aggregating them to develop the predictive model. Estimators are created using bootstrapped datasets, retaining only those with predictive value (Asselman et al., 2023). Random Forest can extract hidden patterns, handle high-dimensional data, and accommodate highly correlated predictors, making it a suitable first approach for new datasets (Boulesteix et al., 2012). For this analysis, default parameters were used.

XGBoost (version 2.1.1) is an extreme gradient boosting method that extends the classical Random Forest approach with additional parameters to improve performance, with this method the tree depth, the learning rate or the child weight can be customized among other parameters being able to adapt to different types of data. While implementation is more complex, XGBoost reduces overfitting and increases predictive accuracy in many cases. Like Random Forest, XGBoost is CART-based, but uses a boosting method, building weak models sequentially where each new tree corrects the residuals of previous trees. XGBoost also supports multi-core computing, allowing scalability to large datasets and reducing computational time. Key parameters used in this study included a maximum depth of 6, a maximum number of nodes from the root, a minimum child weight of 12 (the minimum sum of instance weights in a child), and a learning rate (eta) of 0.6. These settings balance data variability capture and overfitting

prevention.

CNN was implemented to test the approach used by Oncofuse and DEEPrior, which apply CNNs or Bayesian neural networks. Deep learning methods can identify complex patterns in data but require careful implementation. CNNs consist of layers of interconnected neurons inspired by the brain. We used TensorFlow (v2.4.1) with Keras (v2.4.3) to implement a CNN in TrOnco. Although CNNs are typically used for 2D image analysis, we adapted the model to process one fusion at a time, since fusions in a dataset are independent.

Our CNN contains multiple layers: two convolutional layers, two pooling layers, a regularization layer, a reshaped data layer, and dense layers including the output layer. Convolutional layers apply filters across the spatial dimensions of the data to generate 1D activation maps, and pooling layers reduce their dimensionality (O'Shea and Nash, 2015). While the function of each layer can be complex, this architecture captures the variability of the data and improves prediction accuracy (Yu et al., 2019).

To present the output of the algorithm, results were standardized across the three algorithms—Random Forest, XGBoost, and CNN—despite each algorithm using its own data structure. For each fusion in the input file, a probability score is assigned by each algorithm. Using a threshold optimized for maximum true positive rate and minimum false positive rate, fusions are classified as passenger if they are not related to tumorigenesis, or as driver if they are likely to contribute to tumorigenesis.

The output file consists of 18 columns containing relevant information for each fusion after analysis. The first two columns report the right and left breakpoints, followed by columns for the 5' and 3' genes. The next columns provide predictions from all three models, the associated scores, and the classification based on the selected threshold. Additional columns include the chimeric protein sequence and the Gene Ontology (GO) terms for the 5' and 3' genes. The remaining columns detail the domains retained, lost, or broken for both genes.

2.2 Annotation:

We have updated the databases used in the analysis of fusions. Gene Ontology (GO) terms play an active role in fusion characterization. The terms were downloaded from the Ensembl API in July 2025 http://www.ensembl.org. Moreover, a script to download and prepare the data in a suitable format for use with TRONCO was added to GitHub, enabling continuous updates of the program's datasets. For the genome version, we used hg19, consistent with the available data, as clinical studies commonly rely on hg19 to annotate genomic information.

The files affected by the genome version are the RefSeq file from NCBI, which is used to obtain the exon start and end positions to build the corresponding protein. Moreover, to build the corresponding protein from the gene, the whole-genome FASTA file from UCSC was used. Both files can be updated to a hg38 genome version by downloading the updated files and selecting the new version in TrOnco, thanks to its adaptability to multiple genome versions.

To train the model, we used fusion data from normal tissue and tumoral tissue. The normal tissue fusions were extracted from (Babiceanu et al., 2016), as done previously by DEEPrior and other programs, and included 1,399 different fusions. Fusions from tumoral tissue were obtained from the COSMIC database, downloaded from version 102 on May 21, 2025, and included 873 different fusions.

To test the different algorithms, we used an independent dataset. (Kou et al., 2024), analyzed fusions in a cohort of 5,534 colorectal cancer patients. Tumor samples were preserved in formalin-fixed paraffin-embedded tissue, and DNA was extracted for sequencing analysis. Fusions were detected using FusionMap, and only clinically actionable gene fusions were reported. The final testing cohort consisted of 51 fusions from different patients.

As a case study, in addition to the colorectal dataset, an endometrial stromal sarcoma (ESS) dataset from a VHIO research team was used to validate the predictions obtained with TrOnco. This dataset included 21,983 fusions from cancer tissue, among which a curated subset of fusions was available for validation.

2.3 Tissue library:

Another highlight of this project is the addition of endometrial tissue to the library of usable tissues. The preexisting tissues on the tissue library were epithelial, hematological, mesenchymal and an average library. To improve the prediction of endometrial fusions, we added this tissue to the TrOnco database. To build this library, we used normal tissue gene expression and ChIP-seq data. This process can be replicated with any tissue of interest, as we did for endometrial tissue. The steps are as follows: generate a folder containing the expression data and ChIP-seq data, and add the UTR expression file from the other libraries. This UTR file, which is gene-dependent only, is duplicated across all tissue libraries for or-

ganizational purposes.

2.4 Tools/packages used:

The main script of TrOnco is based on Polars, a Python library specialized in manipulating large datasets. The main reason for using this package instead of Pandas—the most popular library for data management—is its fast processing capability. Polars is built in Rust and designed for parallelism; its vectored and columnar processing improves the performance of most operations. This approach enhances the overall performance of TrOnco compared with other fusion analysis programs. However, for specific steps of the analysis or model training, Pandas is used for its ease of implementation and compatibility with other Python packages.

NumPy and Matplotlib were also employed for general processing tasks, such as numeric array operations in NumPy, and plotting or graphically representing data with Matplotlib. These packages also ensure compatibility and correct operation of other software components.

3 PERFORMANCE

The databases used for training and testing were as follows: for normal tissue, data were obtained from (Babiceanu et al., 2016), while fusions reported in tumoral tissue were taken from the COSMIC database. These two datasets were split into 90% for training and 10% for testing the algorithms. Moreover, 51 fusions from the Chinese colorectal cohort (Kou et al., 2024) were added to the test dataset. Two ROC curves were generated for each model using the training and test datasets. Additionally, ROC curves were generated for Oncofuse and DEEPrior with the test dataset (Figure 3).

Using the test dataset, we generated a confusion matrix for each of the three algorithms: Random Forest, XGBoost, and CNN Figure 4.

Method	AUC	Threshold
ONCOFUSE	0.76	0.8
DEEPrior	0.5	0.8
Random Forest	0.98	0.31
XGBoost	0.87	0.37440118
CNN	0.77	0.12119576

Table 1: Predictive algorithms with the AUC and the threshold used to classify the samples.

With the independent dataset we obtain a confusion matrix of the three different algorithms, Random Forest, XGBoost and CNN (Figure 4).

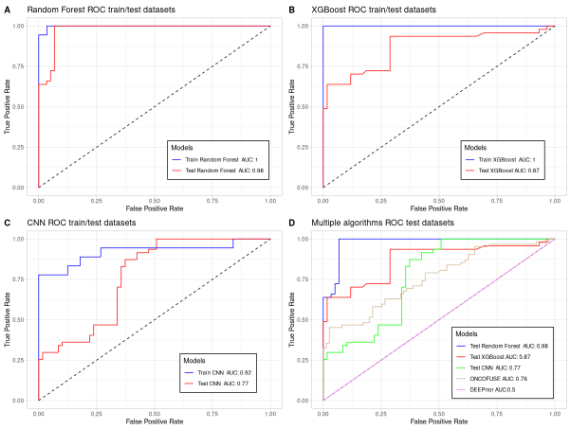

Figure 3: Multiple ROC curve analysis. **A:** ROC curves generated with the Random Forest model with the train and test dataset. **B:** ROC curves generated with the XGBoost model with the train and test dataset. **C:** ROC curves generated with the CNN model with the train and test dataset. **D:** Comparison of the approaches analyzed in this article with the ROC curve generated with the test dataset.

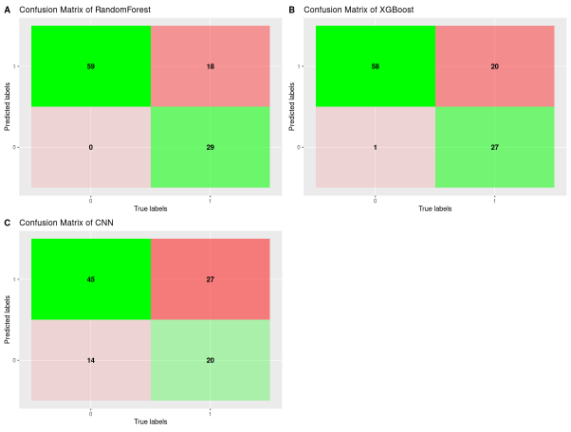

Figure 4: Confusion matrix of the different machine learning algorithms included in TrOnco, used the test dataset to obtain the confusion matrix. **A:** With an accuracy of 0.83 on the Random Forest model. **B:** With XGBoost the accuracy achieved was 0.80 **C:** The accuracy of the CNN model was 0.61

4 CASE STUDIES

To test the performance of our algorithms on real datasets and compare predictions with Oncofuse and DEEPrior, we used a colorectal carcinoma dataset and an Endometrial Stromal Sarcoma (ESS) dataset from a VHIO research group. As the ESS dataset fusions cannot be shared, we plotted only the proportions of passengers and drivers in the predictions. Thresholds

for the different methods were set to the default values described in Table 1. For Oncofuse and DEEPrior, thresholds were obtained from the respective publications.

4.1 Colorectal carcinoma:

The colorectal carcinoma dataset consisted of 51 fusions from different patients. Oncofuse predicted no driver fusions, and DEEPrior classified less than 25% as drivers. In contrast, TrOnco predicted more than 25% driver fusions using Random Forest and XGBoost, and 100% of driver fusions were correctly classified by the CNN model. (Figure 5).

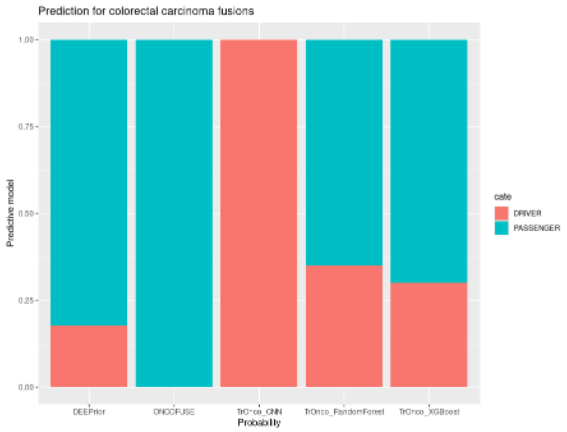

Figure 5: Barplot of the proportion of hits, fusion classified as driver, for all the TrOnco algorithms and ONCOFUSE and DEEPrior. The data used was the colorectal dataset, containing only driver fusions.

4.2 ESS:

The ESS dataset contained 21,984 fusions. Oncofuse predicted 19% as drivers, and DEEPrior predicted 37%. TrOnco models showed higher proportions: 67% for Random Forest, 55% for XGBoost, and 75% for CNN, indicating improved detection of driver fusions compared with the other tools. (Figure 6).

5 DISCUSSION

This project demonstrates the capabilities of TrOnco, improving it from other available fusion analysis programs. TrOnco offers three different algorithms to analyze fusions, along with the option to retrain models and add new features. Each algorithm has its strengths and limitations, providing different ap-

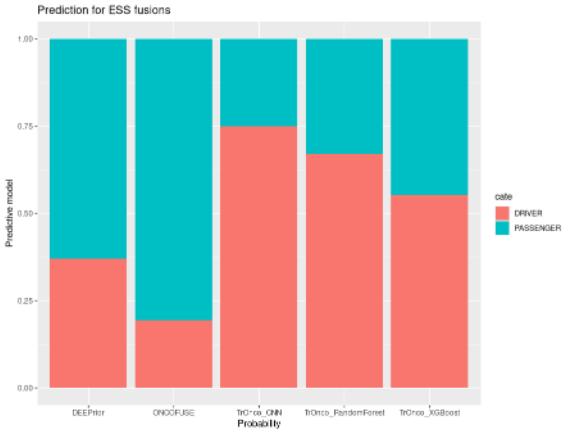

Figure 6: Barplot of the proportion of hits, fusion classified as driver, for all the TrOnco algorithms and ONCOFUSE and DEEPrior. The ESS dataset from the VHIO research group was used to build this plot; this dataset only contains driver fusions.

proaches to prediction. Moreover, updating the data and adding endometrial tissue improved the performance of our models, as TrOnco detected more fusions in both case studies. All algorithms are Python-based, a choice motivated by Python's optimization for machine learning and deep learning as well as its widespread use and familiarity among researchers.

Although XGBoost is known for achieving better performance than Random Forest, the increased parameter tuning it requires can limit its practical effectiveness. In our study, Random Forest produced the best results in both examples, while CNN and XGBoost were implemented and remain open for further optimization. Although CNN appeared to yield more hits in the case studies, this outcome can be attributed to the higher false positive rate of the model.

The main limitation of this project is the availability of data. With only one normal tissue dataset containing 51 fusions, training the algorithms was restricted. Similarly, the tumoral tissue fusions from COSMIC are limited. A crucial limitation is the definition of tumorigenic fusions: in this project, we assumed all fusions detected in tumoral tissue contribute to tumorigenesis. However, not all fusions are active drivers; tumors are mutagenic environments that induce genomic rearrangements, generating both driver and passenger fusions, which our current approach does not differentiate. Another limitation is the choice of the split proportion in the test and train datasets. In order to maximize the learning step and due to the limited data, we decided to use 90% of the data as training and 10% as test, which may reduce the reliability of the test measurements.

In the case study, no confusion matrix was generated due to the absence of normal fusions. Both

datasets were analyzed using only tumor samples, and the resulting graphs were based solely on tumor fusions.

An inherent limitation of machine learning methods is that, while they are very effective at uncovering hidden patterns in data, they are constrained by the data used to train them. We cannot rely on the outcomes of these models if the training data is inaccurate or significantly different from the data on which the model is applied. When using TrOnco as a tool to characterize de novo fusions, this limitation must be taken into account, and its results should be considered a preliminary step in understanding gene fusions rather than a substitute for wet-lab experiments(de Crécy-Lagard et al., 2025).

Finally, TrOnco has enhanced the prediction of oncogenicity compared with preexisting models. All code will be made available in a GitHub repository for free use, provided with the corresponding citation.

ACKNOWLEDGEMENTS

We would like to thank the developers of **ONCOFUSE** for sharing their code with the scientific community and for serving as a model in the development of TrOnco.

Additionally, I would like to express my deepest appreciation to the VHIO bioinformatics team for their guidance and support during the development of this project.

REFERENCES

Abate, F., Zairis, S., Ficarra, E., Acquaviva, A., Wiggins, C. H., Frattini, V., Lasorella, A., Iavarone, A., Inghirami, G., and Rabadan, R. (2014). Pegasus: a comprehensive annotation and prediction tool for detection of driver gene fusions in cancer. *BMC Systems Biology*, 8(1):97.

Annala, M., Parker, B., Zhang, W., and Nykter, M. (2013). Fusion genes and their discovery using high throughput sequencing. *Cancer Letters*, 340(2):192–200. Next Generation Sequencing Applications in Cancer Research.

Asselman, A., Khaldi, M., and Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning xgboost algorithm. *Interactive Learning Environments*, 31(6):3360–3379.

Babiceanu, M., Qin, F., Xie, Z., Jia, Y., Lopez, K., Janus, N., Facemire, L., Kumar, S., Pang, Y., Qi, Y., Lazar, I. M., and Li, H. (2016). Recurrent chimeric fusion rnas in non-cancer tissues and cells. *Nucleic Acids Research*, 44(6):2859–2872.

Boulesteix, A.-L., Janitza, S., Kruppa, J., and König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 2(6):493–507.

de Crécy-Lagard, V., Dias, R., Sexson, N., Friedberg, I., Yuan, Y., and Swairjo, M. A. (2025). Limitations of current machine learning models in predicting enzymatic functions for uncharacterized proteins. *G3 Genes—Genomes—Genetics*, page jkaf169.

Haas, B. J., Dobin, A., Ghandi, M., Van Arsdale, A., Tickle, T., Robinson, J. T., Gillani, R., Kasif, S., and Regev, A. (2023). Targeted in silico characterization of fusion transcripts in tumor and normal tissues via fusioninspector. *Cell Reports Methods*, 3(5):100467.

Hassanpour, S. H. and Dehghani, M. (2017). Review of cancer from perspective of molecular. *Journal of Cancer Research and Practice*, 4(4):127–129.

Kloosterman, W. P., Coebergh van den Braak, R. R., Pieterse, M., van Roosmalen, M. J., Sieuwerts, A. M., Stangl, C., Brunekreef, R., Lalmahomed, Z. S., Ooft, S., van Galen, A., Smid, M., Lefebvre, A., Zwartkruis, F., Martens, J. W., Foekens, J. A., Biermann, K., Koudijs, M. J., Ijzermans, J. N., and Voest, E. E. (2017). A systematic analysis of oncogenic gene fusions in primary colon cancer. *Cancer Research*, 77(14):3814–3822.

Kou, F.-R., Li, J., Wang, Z.-H., Xu, T., Qian, J.-J., Zhang, E.-L., Zhang, L.-J., Shen, L., and Wang, X.-C. (2024). Analysis of actionable gene fusions in a large cohort of chinese patients with colorectal cancer. *Gastroenterology Report*, 12:goae092.

Lovino, M., Ciaburri, M. S., Urgese, G., Di Cataldo, S., and Ficarra, E. (2020). Deeprior: a deep learning tool for the prioritization of gene fusions. *Bioinformatics*, 36(10):3248–3250.

Mitelman, F., Johansson, B., and Mertens, F. (2007). The impact of translocations and gene fusions on cancer causation. *Nature Reviews Cancer*, 7(4):233–245.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks.

Rabbitts, T. H. (1994). Chromosomal translocations in human cancer. *Nature*, 372(6502):143–149.

Shugay, M., Ortiz de Mendíbil, I., Vizmanos, J. L., and Novo, F. J. (2013). Oncofuse: a computational framework for the prediction of the oncogenic potential of gene fusions. *Bioinformatics*, 29(20):2539–2546.

Singh, A., Thakur, N., and Sharma, A. (2016). A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1310–1315.

Yu, L., Li, B., and Jiao, B. (2019). Research and implementation of cnn based on tensorflow. *IOP Conference Series: Materials Science and Engineering*, 490(4):042022.

LINKS TO DATA

UCSC Genome Browser:

Link to download the whole genome fasta file for the desired genome version https://hgdownload.soe.ucsc.edu/downloads.html

RefSeq NCBI:

Link to dowload the RefSeq file for the desired genome version https://ftp.ncbi.nih.gov/refseq/H_sapiens