Deltect: Genomic Deletion Pathogenicity Classifier

CMPT 310: Introduction to Artificial Intelligence

Brandon Tang, Tanvir Samra, Kaira Martinez, Jenny Wei

Fall 2025

1 AI System

1.1 Introduction

The goal of our project is to develop a machine learning model, Deltect, that can predict the pathogenicity of genomic deletions. Genomic deletions are a type of structural variant where a segment of DNA is missing. Depending on where they occur, deletions can be associated with benign, to pathogenic variants associated with diseases.

In clinical genetics, manually classifying the pathogenicity of a deletion is challenging and time consuming. Fortunately, there are databases such as ClinVar that provide curated labels of pathogenicity, but for most, deletions remain unclassified, especially when extracted from sequencing data such as BAM/SAM files that are commonly used in practice. Deltect aims to leverage the curated deletions to train a classifier than can estimate a patient's genomic data through BAM/CRAM files, and classify if a deletion is pathogenic.

1.2 Methodology

Our methodology involves supervised learning trained on ClinVar deletion variants and HG002/1000 Genomes to predict the pathogenicity. We use Sci-kit learn's Random Forest Classifier, with Gradient Boosting and optional XGBoost

1.2.1 Data Aquisition

We extract deletion variants to train our model from 2 main sources:

1. **ClinVar** to extract deletions that are pathogenic

2. **HG002/1000 Genomes Project** that are benign or uncertain for class balancing

Other specifications:
- **Reference Genome:** hs37d5 (GRCH37)
- **Gene Annotations:** GENCODE V19 GTF for gene-level context

To gather ClinVar data, we use the NCBI Entrez API via Biopython, restricitng our query to deletions on selected chromosomes at a time **TODO: chosen chromosome explain later which one**.

Rationale for Using Only Pathogenic ClinVar Variants

Although ClinVar contains both pathogenic and benign classifications, we intentionally restricted it to pathogenic deletions. This decision was driven by the composition of the HG002 dataset, where there consists a large volume of benign/likely benign/ uncertain deletions. When combined with ClinVar's full datasets, benign samples dominated having 22% of the training samples that are pathogenic.

While the model's benign predictions were high, this imbalance led to a lower recall performance. In a medical context, recall is especially critical because false negatives correspond to pathogenic deletions being incorrectly classified as benign, which is what we would want the least to occur.

Mitigating Class Imbalance

To address this issue of a lower recall performance, we balanced the training set of 10,000 deletions with:
- 5,000 pathogenic deletions (from ClinVar)
- 5,000 benign or uncertain deletions (from HG002)

This balance ensures that the model avoids majority class bias, and improves sensitivity to pathogenic variants.

1.2.2 Feature Engineering

For each deletion variant, we construct a set of features based on its position and annotations:

Genomic features:
- Deletion size
- Chromosome position (start & end), normalized coordinates

Sequenced-based features:
- GC content
- Sequence complexity

Gene features:

- One-hot encoded gene symbols
- Review satus confidence scores (one-hot)

Categorical features:

- Variant type
- Review Status

1.2.3 ML Model Choice and Training

We implemented a Random Forest Classifier using Skikit-Learn to predict deletion pathogenicity. Random Forests are well-suited for this task because:

- they handle mixed feature types such as numerical + categorical
- they model non-linear relationships between features
- they are robust to noise and variability in biological labels
- they provide feature importance scores and **probability??? to polish**

We performed iterative hyperparameter tuning to identify the configuration that yielded the best performance. The key hyperparameters tuned include:

- class_weight
- **TODO: plug in final hyperparameters of model**

The final selected configuration produced the strongest performance on the hold-out test set.

1.2.4 Training and Testing

We utilize a train-test split with 80% of the dataset for training and 20% for testing. We performed a 5-10 fold cross-validation on the training set to esitmate the performance of our model.

We evaluate the model on the test set, and accumlated the following metrics:

- Mean Precision: **96.4%**
- Mean Recall: **98.7%**
- Mean Specificity: **90.1%**
- MSE: **0.0188**

1.3 AI Pipeline

TODO: insert training pipeline image here. add validation pipeline

Our system can be viewed as an end-to-end AI pipeline with two main stages:

1. **Training Pipeline**
   - Fetch deletions from clinVar
   - Data preprocessing and normalization
   - Feature encoding
   - Analyze clinical significance distribution
   - Model Training and Evaluation
   - Save trained model

2. **Validating Pipeline TODO**

3. **Inferring Pipeline**
   - Extract deletions from BAM file using CIGAR strings
   - Convert deletion data to model format
   - Predict Pathogenicity for deletion variants

1.4 Limitations

TODO

2 Features Table (1-2 pages)

TODO

Description	Platform	Completeness	Code	Authors(s)	Notes
Random Forest Classifier	Python				
Inference	Python				
BAM extraction					

3 External Tools & Libraries (1/2 page)

TODO

3.1 Datasets

- ClinVar
- HG002/1000 Genomes Project

3.2 Frameworks

- Scikit-Learn Random Forest Classifier

3.3 External Open Source

Double check requirements.