boostDM

Release 1.0

BBGLab

May 25, 2022

CONTENTS

BoostDM is a method to score all possible point mutations (single base substitutions) in cancer genes for their potential to be involved in tumorigenesis.

The method has been described and used in this study:

> In silico saturation mutagenesis of cancer genes
> Ferran Muiños, Francisco Martinez-Jimenez, Oriol Pich, Abel Gonzalez-Perez, Nuria Lopez-Bigas
> DOI: 10.1038/s41586-021-03771-1
> URL: https://www.nature.com/articles/s41586-021-03771-1

These docs will guide you through the description, resources and practicalities of the method.

CHAPTER

ONE

CONTENTS

1.1 Resources

There are several public resources related to the **boostDM** framework, each taylored to a specific purpose.

1.1.1 paper

In the In silico saturation mutagenesis of cancer genes [Muiños *et al.*, 2021] paper the boostDM approach has been described, validated and used to analyze the role of background mutation probability in the generation of cancer driver mutations.

1.1.2 website

boostDM website is intended to explore the predictions and explanations resulting from the boostDM pipeline for a collection of models meeting high enough reliability requirements. The website is searchable by cancer gene, tumor type, and mutation coordinates.

1.1.3 pipeline repo

The boostDM pipeline repo is a public BitBucket repo containing the source code for reproducing the pipeline from a public, pre-processed input dataset. The input is supplied as part of the stable source dataset (doi: 10.5281/zenodo.4813082) hosted in Zenodo.

1.1.4 paper analyses repo

boostDM analyses is a public repo containing a collection of scripts and jupyter notebooks to generate the figures displayed in the [Muiños *et al.*, 2021] paper. These scripts feed on the stable source dataset (doi: 10.5281/zenodo.4813082) published in Zenodo.

1.1.5 IntOGen

IntOGen website is intended to explore the landscape of mutations in driver genes after analyzing more than 28,000 tumors. IntOGen is instrumental for boostDM as it provides important annotations used as features in the learning pipeline.

1.1.6 Cancer Genome Interpreter

Cancer Genome Interpreter is a variant interpretation framework for researchers and clinicians that feeds on boostDM outputs for the annotation os single nucleotide variants.

1.2 Motivation

To be able to identify individual cancer mutations a novel approach is required that annotates all possible mutations in a gene, independently of their probability of occurrence, as potential drivers or passengers. Instead of relying on functional impact metrics [Kircher *et al.*, 2014, Pollard *et al.*, 2010], this method should measure the ability of a mutation to drive tumorigenesis. Moreover, as the function of each cancer gene is different, as well as their role in tumorigenesis, we can expect that the features that define driver mutations will be different per gene. Thus, we aim to create an approach that learns the features that define driver mutations for each cancer gene independently. In addition, the same cancer gene may have different mechanisms of tumorigenesis in different tissues (e.g. compare EGFR in Lung adenocarcinoma and Glioblastoma). Thus, if enough data is available, we aim to create gene and cancer type specific models. Furthermore, it would be desirable that such classification yields human-readable results, which help researchers point at the key features defining driver mutations in a cancer gene.

This problem has been approached before through experiments of saturation mutagenesis, in which all possible mutants of a cancer gene are generated and their impact on protein function [Kakudo *et al.*, 2005, Kato *et al.*, 2003, Kawaguchi *et al.*, 2005], or cell viability [Findlay *et al.*, 2018, Mighell *et al.*, 2018] are assessed. These experiments possess obvious technical and economic hurdles. Furthermore, due to limitations imposed by the experimental setup, these approaches do not directly measure the tumorigenic potential of mutations, but rather some proxy, such as their functional impact. For instance, in certain tumor suppressor genes, saturation mutagenesis experiments have been conducted in haploid human cells to identify mutations that abrogate cell viability [Findlay *et al.*, 2018]. Only scattered mutagenesis assays have been carried out that actually assess the tumorigenic potential of mutations affecting cancer genes, restricted to few cell types, which do not represent the wide spectrum of tissue-specific constraints. Generalizing them to cover hundreds of cancer genes across cell types representing different tissues would be a herculean task.

To address this problem we have developed a platform to train machine learning models to identify all possible driver mutations in cancer genes across cancer types. This document presents details of the methodology, features and data used to generate these gene-tumor type specific models, as well as an extensive benchmark of their performance. Then we provide an extensive description of the validation and comparison experiments to critically assess our approach, including hold-out testing with experimentally validated rare oncogenic variants, comparisons with experimental saturation mutagenesis of 5 genes and with several bioinformatics tools. We also re-trained models with subsamples to assess the growth outlook of the pipeline as more sequenced tumors become available. Finally, we include a short section where we justify the choice of the exponential function used in the definition of the discovery index.

1.3 BoostDM overview

1.3.1 Training set

boostDM is based on a supervised learning approach using a training set of driver and passenger mutations in cancer genes. How to define such training datasets is not trivial, as there is not a complete ground truth collection of driver and passenger point mutations in a cancer gene.

For some genes, the excess of observed-over-expected mutations is large enough that the vast majority of observed mutations are involved in tumorigenesis. We propose (and validate) that if there are enough observed mutations in cancer driver genes with consequence type above a certain excess will provide both sufficiently many mutations and high enough driver enrichment to render good discriminative ability by standard supervised classification techniques. Thus we define as positive set (drivers) the set of mutations with excess of observed-to-expected higher than 85% according to dNdScv [Martincorena *et al.*, 2017].

From a theoretical perspective, passenger mutations would be any other mutations. For discriminative efficiency, however, training data must reflect the fact that passenger mutations are randomly generated following tri-nucleotide specific mutation rates (neutral mutational profile). Thus a collection of simulated mutations according following the neutral mutational profile will be used as a negative set (Passengers).

1.3.2 Features

Each mutation provided for training is annotated with a vector of mutational features, which the classification task exploits to discriminate between observed drivers and passengers in tumours. Some mutational features of each cancer gene across malignancies have been derived from the systematic analysis of tens of thousands of tumor samples by IntOGen [Martínez-Jiménez *et al.*, 2020]. Other relevant features have been collected from public databases: VEP.92 [McLaren *et al.*, 2016], PhosphositePlus [Hornbeck *et al.*, 2015] and phyloP [Pollard *et al.*, 2010].

1.3.3 Learning heuristics

For each gene-tumor type pair, the learning heuristics makes use of gradient boosted trees as the base classifier. Several base classifiers are trained on subsets (bagging) of the pool of learning mutations. Then these classifiers are aggregated into a consensus model. Alongside the training, this approach yields an out-of-bag (cross-validation) assessment of the model performance. This evaluation strategy is expected to give a conservative estimation of the performance, as it reflects the typical performance of the base classifiers before aggregation.

1.3.4 Scope

As a principle, **boostDM** provides a distinct model for each driver gene-tumor type pair. In some cases, however, the models cannot be successfully rolled out due to the fact that there are not enough mutations for training or even if there are, cross-validation yields low accuracy. Within each gene, **boostDM** covers the protein coding sequence of the genome.

The effect of all mutations being considered are relative to the canonical transcript of protein coding genes according to the Ensembl Variant Effect Predictor version 92 (VEP.92 [McLaren *et al.*, 2016]). Notice that these transcripts may include mutations in untranslated regions which are splicing affecting, even if they are non-protein-altering mutations.

1.3.5 Prediction

For each cancer gene and tumor type our method fits a model representing feature rules that define driver mutations in that context. Specifically, the method yields a score $0 \leq p \leq 1$ that reflects the strength of the forecast that the mutation is a potential driver: the higher the score, the stronger the evidence. Although p is not calibrated to support a probabilistic interpretation, a score >0.5 reflects a predominant evidence in favour of the mutation being a potential driver.

1.3.6 Explanation

Each base classifier (gradient boosted tress) admits an additive explanation model that decomposes the logit prediction of each individual mutation as the sum of so-called SHAP values associated to the features [Lundberg and Lee, 2017]. These are explanatory values in the sense that a feature having positive (resp. negative) SHAP value implies that the method deems more likely (resp. unlikely) that the mutation is a driver conditioned to the feature's value (see Shapley Additive Explanations). We define the SHAP values of the aggregated model as the mean SHAP values across base classifiers.

1.4 BoostDM in detail

1.4.1 Implementation

Each **boostDM** model is an ensemble of base classifiers (n=50) each trained with a random sample subset (bagging) drawn from the learning dataset. Each base classifier is a boosted trees model (sum of regression tree functions trained with XGBoost gradient boosting implementation) with a cross-entropy loss function. The base classifiers are aggregated into a consensus model with an aggregator intended to correct for the systematic bias of each expert classifier (Section~ref{consensus}). The consensus model additive explanations are computed as the mean SHAP values base classifiers (Section~ref{explanations}).

The **boostDM** pipeline has been implemented in Python and Nextflow [Di Tommaso *et al.*, 2017]. The models were implemented with the libraries **xgboost** [Chen and Guestrin, 2016] (version 0.90) to train the base classifiers and **shap** [Lundberg and Lee, 2017] (version 0.28.5) to compute the SHAP values associated with the predictions.

Base classifier hyperparameters

The model hyperparameters specify the learning strategy to keep a balance between loss minimization and generalization. While some hyperparameter values are the result of an explicit design decision, others cannot unequivocally defined. The current values stand as a compromise resulting from a testing series. For more information about hyperparameter tuning with XGBoost, please check **xgboost** documentation xgboost.readthedocs.io.

Herein we provide the current hyperparameter set-up:

- Models are sums regression tree functions functions (`booster = "gbtree"`)
- The learning task minimizes a cross-entropy loss function (`objective = "binary:logistic"`).
- All the features are available when building each new tree (`colsample_bytree = 1`) and each new tree level (`colsample_bylevel = 1`).
- Learning rate 0.001 (`learning_rate = 0.001`) was decided on by testing in combination with the maximum number of training steps.
- Percentage of samples randomly drawn prior to growing a new tree is 70% at every iteration (`subsample=0.7`). This is a technical choice to further prevent overfitting.

- Maximum depth of trees used in the tree function is 4 (`max_depth=4`). In practice a good performance can be attained even at depth as low as 1 (stumps). However, at a higher but still tolerable computational cost, more depth gives also more room for learning complex dependencies.
- Default regularization hyperparameters were employed.
- Maximum number of training steps is 20,000 (`n_estimators = 20000`).

Empirical evidence suggests that this set-up was a convenient choice in view of performance tests, expected capacity for the models to handle feature interactions and moderate training speed. Notwithstanding the importance of this technical setup, due to the fact that our base classifiers undergo additional aggregation, biases attributable to gradient boosting training will tend to moderate, resulting in an implicit form of regularization.

1.4.2 Oncotree: tumor type ontology

A specific tumor type determines the set of sequenced samples that contribute positive mutations to the learning dataset, the neutral mutation profile required to simulate passenger mutations and the value of some features.

We aim to develop models that are capable of classifying mutations across tumor types with different degrees of generality, the main reason being that for some specific gene-tumor type pairs the number of mutations is not sufficient to render a good model fitting, but the lack of mutations to train a gene-tumor type model can potentially be mitigated by pooling mutations from other samples of similar tumor types from a histopathological perspective.

We delineate a model selection strategy based on a hierarchical organization of tumor types that allows to conduct this pooling in a systematic way. Thus we defined a tumor type ontology (Oncotree) adapted from IntOGen [Martínez-Jiménez *et al.*, 2020] that allows us to group samples according to several degrees of specificity regarding the tissue-pathology context where the mutations are reported. Then the root term **CANCER** is connected to two children terms, **SOLID** and **NON_SOLID**, which children connected at increasing levels of specificity. The leaves of this hierarchy define the most specific tumor-type terms considered in this study.

Each model trained by **boostDM** is relative to a gene and tumor-type pair, where the tumor type corresponds to a term in the Oncotree: we denote it as (G,T)-model for brevity.

Filtering

Consequence type We restricted our analysis to single nucleotide substitutions annotated with either of the following Sequence Ontology~citep{SO} terms (according to VEP.92) with respect to the canonical transcript: splice-donor-variant, splice-acceptor-variant, splice-region-variant; missense-variant; stop-gained; stop-lost; synonymous-variant.

Multiple nucleotide variants Adjacent point mutations were excluded from our analysis due to the plausible risk that these are misannotated.

De-duplication of samples We removed duplicated samples (i.e., multiple samples from the same donor) for the training of models. If a cohort included duplicated samples, we selected the first according to its alphabetical order as in [Martínez-Jiménez *et al.*, 2020]. For additional information about the pre-processing of samples, the reader can check https://intogen.readthedocs.io.

1.4.3 Mutational features

Given a point mutation in a driver gene and tumor type (either observed or randomized) our method requires an annotation of the mutation with a set of features to train the classifiers.

Consequence type Every mutation was annotated with the following binary features, matching the obvious Sequence Ontology annotations: `missense`, `nonsense` (stop-gained; stop-lost), `synonymous` and `splicing`. For all observed and synthetic mutations in driver genes, we retrieved the annotations from VEP.92 for the canonical transcript.

Clusters of mutations in the DNA primary structure (linear clusters) For every mutation we annotated whether it overlaps a significant linear cluster identified by the method OncodriveCLUSTL [Arnedo-Pac *et al.*, 2019]. We created two annotation tiers for mutations overlapping i) linear clusters found in a cohort of the corresponding tumor type (tumor type specific: `cat_1`), or ii) clusters only identified in cohorts belonging to other tumor types (pan-cancer: `cat_2`). Additionally we create another feature that represents the OncodriveCLUSTL score for linear cluster in the tumor type (i.e., `cat_1`).

Clusters of mutations on the protein 3D structure (3D clusters) identified by the method HotMAPS [Tokheim *et al.*, 2016] in a tumor type specific (`cat_1`) and pan-cancer (`cat_2`) way. For details of the HotMAPS implementation please refer to [Martínez-Jiménez *et al.*, 2020].

The overlap with Pfam domains Pfam domains [El-Gebali *et al.*, 2019] that are significantly enriched for mutations in the gene across tumors of the cancer type, identified by the method smRegions [Martínez-Jiménez *et al.*, 2020], in a tumor type specific (`cat_1`) and pan-cancer (`cat_2`) way.

Phylogenetic conservation Conservation of the reference nucleotide across mammals measured through the PhyloP 100-way score [Pollard *et al.*, 2010].

Post-translational modifications Non-synonymous coding mutations were also annotated with post translational modifications (PTMs) when the mutation affected an amino acid that is known to be acetylated, phosphorylated, ubiquitinated, methylated or subject to any other regulatory modification according to PhosphositePlus [Hornbeck *et al.*, 2015].

NMD Whether nonsense mutations overlap the last coding exon of the canonical transcript (according to VEP.92) reflecting the potential for the truncating variant to skip nonsense-mediated RNA decay [Lindeboom *et al.*, 2016].

Two of the features (PhyloP and the OncodriveCLUSTL score) were encoded in their original floating numerical value. The rest of the features (categorical) were given with a one-hot encoding.

Excess of mutations

The so-called excess of mutations for a given coding consequence-type quantifies the proportion of observed mutations at this consequence-type that are not explained by the neutral mutation rate. The excess is inferred from the dN/dS estimate ω as $(\omega - 1) \,/\, \omega$. We computed the excess for missense, nonsense and splicing-affecting mutations.

1.4.4 Training

Training datasets

The first requirement for our supervised learning approach is to create a catalogue of mutations labeled as Drivers or Passengers. This catalogue is established globally for all the models, then for the training of each (G,T)-model only the mutations relevant to the (G,T) context are used.

Drivers

The set of Drivers used for training are observed mutations in mutational cancer genes (IntOGen) that exhibit a consequence-type specific excess of observed-over-expected mutations higher than 85% according to the method dNdScv [Martincorena *et al.*, 2017]. Repeated mutations (i.e., when the same mutation is observed in different samples) are allowed in the set of Drivers.

Passengers

For each Driver mutation mapping to a gene and cohort of samples, we randomly select one mutation from the CDS (VEP.92 canonical transcript) following probabilities the tri-nucleotide specific probabilities recorded in the cohort (see Methods). Because we are bound to train 50 base classifiers, we do this random selection 50 times with replacement.

Data Splits

For each gene-tumor type pair (G,T), the corresponding base classifiers are obtained after conducting training with two sets of annotated mutations, namely Train and Test. We will refer to a Train-Test pair as a Split. In our setting each Split is generated randomly and must satisfy the following requirements:

- Both Train and Test sets are Driver-Passenger balanced.
- The Train set comprises 70% of Driver examples.
- Each unique Driver instance belongs either to Train or Test.
- Repeated Driver mutations (i.e., when the same mutation is observed in different samples) are allowed in Train, but not in Test (to prevent spurious inflation of the cross-validation performance evaluation).
- Passenger mutations in Train and Test are randomly selected among all the mutations in the Passengers pool matching the (G,T) context.

We set the minimum average number of mutation instances in the Train size to deem the models trainable at 30.

Cross-validation and early stopping

When training a base classifier, we implemented sequential evaluation to determine the optimal number of estimators. We evaluate the performance after each learning step using the Test dataset. This sequential evaluation informs whether the training must stop due to steady or decreasing performance for a set of consecutive iterations (early stopping).

We use the cross-entropy (see scikit-learn.org/model_evaluation) as a performance measure to assess training progress sequentially with cross validation.

Given true labels $\mathbf{y} = \{y_i\}$ and forecasts $\hat{\mathbf{y}} = \{\hat{y}_i\}$, the cross-entropy is defined as:

$$J(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right].$$

We define an early-stopping requirement of 2,000 iterations. Thus each expert classifier training finishes when either of the following conditions is first met: i) the maximum number of training steps (N=20,000 iterations) is attained; ii) the model's performance on the Test data set does not improve for 2,000 consecutive iterations.

After the training we record the performance of the classifier via log-loss and a weighted F-score (F_{50}) when applied to the Test dataset. Although cross-validation and early stopping allow some extent of data leakage from the Test dataset, it turns out to provide a conservative performance evaluation compared to real performance in sample-wise hold-out tests.

1.4.5 Shapley Additive Explanations

Each base classifier yields an additive explanation model based on Shapley Additive Explanations (SHAP values) [Lundberg and Lee, 2017]. Specifically, each expert classifier can decompose the logit forecast yielded by a particular mutation m, say logit $p(m)$ into a collection of SHAP values $\{s_i(m)\}$, one per feature, in such a way that $s_1(m) + \ldots + s_n(m) =$ logit $p(m)$ for all m.

The intuition behind the concept of SHAP value is rooted and better understood in the context of cooperative game theory, where the Shapley value [Shapley, 1988] as originally described. If we think of logit(p) as a utility function that depends on the contribution of a coalition of features (thought of as players of a cooperative game) the SHAP values account for the relative contributions of the features by averaging the differences between: i) the expected forecast when the feature value is known and ii) the expected forecast when the feature value is not known.

More specifically, given an individual base classifier, denoted by B, the additive explanation model $\mathcal{A}(B)$ for B is a map from the feature space $F = F_1 \times \ldots \times F_n$ (the set of arrays representing the possible values of the features) onto the euclidean space $\mathbb{R}^n$ of dimension n equal to the number of features, which satisfies the following two requirements:

- The additive explanation model $\mathcal{A}(B)$ gives an additive decomposition of the logit forecast given by B, i.e. if $m \in F$ denotes the array of feature values for some mutation instance m and p(m) denotes the logit prediction of the classifier on m, if $\mathcal{A}(B)(z) = (\phi_1, \ldots, \phi_n)$ then $p(z) = \phi_1 + \ldots + \phi_n$.
- Given a feature array m for some mutation instance, the i-th component of $\mathcal{A}(B)$ is an estimate of the average marginal payoff of the i-th feature over all possible feature coalitions, i.e.,

  $\phi_i(m) = \sum_{S \subseteq [n] \setminus \{i\}} K(S) \cdot (f(S \cup \{i\}; m) - f(S; m))$

  where S runs through all possible subsets of the set of features excluding the i-th feature; and

  $K(S) = \frac{|S|! \cdot (n - |S|! - 1)}{n!}$

  is an averaging coefficient that depends on the size of S; and

  $f(U; m) = \mathrm{E}[p(m) \mid m \in U]$

  is the conditional expectation of the model's prediction for m, given that only the values of the features belonging to the subset U are known.

1.4.6 Aggregator of base classifiers

One challenge of our approach is that during training a fraction of the Driver labels may correspond to passenger mutations, even if this fraction will in general be lower than 15% (recall that we only consider those mutations matching a consequence-type specific dNdScv excess higher than 85%). This implies that each individual expert classifier may be at risk of being biased by the specific choices of Drivers made in the Splits. To work around this difficulty, we propose to use a pool of base classifiers, each trained with a different partial view of the data, so that the final forecast and explanation results from combining these classifiers, while also correcting for the systematic biases that they may bear.

Forecast aggregator

Our approach resorts to a non-linear combination of probabilities based on a logit-normal model [Satopää *et al.*, 2014]. Specifically, if a classifier B_i casts a prediction p_i that a specific mutation is a driver and $Y_i =$ logit(p_i), this modelling choice assumes that p_i arises from some latent true probability p that the mutation is a driver, which has in turn been moderated by some degree of systematic bias:

$$Y_i = \log\left(\frac{p}{1-p}\right)^{1/a} + \epsilon_i$$

with $\epsilon_i \sim \mathrm{N}(0, \sigma^2)$ being a normal random variable with standard deviation σ and $a \geq 0$ represents the systematic bias.

Intuitively, systematic bias quantifies the extent to which individual classifiers regress towards a log-odds zero due to partial information or under-confidence while issuing their respective forecasts. While a=1 would be associated with an accurate forecast, a>1 represents under-confidence.

Using the previous modelling assumptions, given a collection of classifiers and their respective forecasts p_i and logits Y_i, the latent true probability p can be estimated as $\hat{p}$ with the following estimator:

$$\hat{p}(a) = \frac{\exp(a\overline{Y})}{1 + \exp(a\overline{Y})}$$

where $\overline{Y}$ denotes the average of the logits Y_i and $a \geq 1$ is the systematic bias.

The level of systematic bias gives us a method to sharpen the consensus forecasts of a coalition of under-confident predictors. In the interest of clarity and interpretability of the outcome, we require **boostDM** forecasts to have a marked bimodal tendendency to yield forecasts close to either 0 or 1. Upon testing the method in some exemplary gene-tumor type contexts, we committed to the choice a=2.3 for all models.

SHAP aggregator

The SHAP vector associated to a specific input mutation is computed as the feature-wise mean of the SHAP vectors cast across base classifiers.

1.4.7 Model Selection

As a principle, for each (G,T) context we can compute a classification model as an aggregator of base classifiers. A weak predictive power can come about either because there are not enough observed mutations to render a clear signal by means of regression with the current features, or because regardless of the number of mutations the set of mutational features is not sufficient to learn a distinct signal for the Drivers. A weak cross-validation predictive ability is a clear sign that the model is up to the learning problem, however a strong performance is not an unequivocal sign that the model is solving the task properly. Herein we discuss the criteria to deem a model sufficient and the rule to decide which (general) model to use given a target gene-tumor type pair.

Evaluation in test sets

Train-test splits give rise to base classifiers that are ultimately merged into a consensus classifier. These splits allow performance assessment by casting the predictions of the base classifiers trained with the split on the test set (cross-validation). The performance measure used to evaluate each partial classifier is a weighted F-score. The selection of the weighted F-score is motivated by the fact that some (up to 15% given one of the thresholds set for building models) "driver" mutations may actually be passengers (see below). From the 50 splits computed to generate the consensus classifier a global performance summary is derived taking the median of the F-scores.

F-score

To establish a simple and uniform performance criterion we resort to a weighted F-score that gives more importance to precision than to recall. We used an F-beta score with beta=0.5 as defined in citep{rijsbergen-1979}, which would measure the effectiveness of retrieval from the perspective of a user who attaches twice as much importance to precision as recall. We denote this as F_{50} throughout.

First, in assessing driverness, false negatives are in general more tolerable than false positives for a classifier with practical implications, as the real set of drivers is generally assumed to be much smaller than the set of all possible mutations that can be generated under neutral selection. Second, due to the intrinsically impure set of the mutations used for learning, some driver mutations in the test set are expected not to be true driver mutations. Consequently, a theoretically perfect classifier could not even attain perfect recall, hence recall must be undervalued in order to assess the method's performance.

Representativeness

Intuitively, the performance criterion delineated in the previous section is not sufficient to establish the confidence of the assessment. For example, low confidence would follow if the repertoire of observed mutations upon which the model is trained are not representative of the set of potentially observable mutations in the gene-tumor type context. Therefore, the performance assessment must be endowed with a measure of representativeness of the observed mutation set.

To measure representativeness we make use of two properties: i) the propensity to observe the same mutations on repeated subsamples (measured by the discovery index, see online Methods) and ii) the number of mutations. We establish, as a rule, that to build a model representing a gene-tumor type with low discovery index, more observed mutations are required than those needed to build a model of a gene-tumor type with higher discovery index. Specifically, the following rule is followed: if the discovery index is higher than 0.2 at least 30 observed mutations are required to build the model; otherwise we require at least 40 observed mutations.

Model quality criteria

We deem the quality of the models sufficient if they meet the following two conditions:

Minimum training size ≥ 30 training examples on average across training splits.

Representativeness The set of observed mutations qualifies as a representative set; specifically, we require more observed mutations if the discovery index is lower, i.e., if N denotes the total number of observed mutations in a given gene and tumor type context, at least one of the following conditions must hold: $N \geq 40$; or discovery index ≥ 0.2 and $N \geq 30$; or discovery index ≥ 0.4 and $N \geq 20$; or discovery index ≥ 0.4 and $N \geq 10$.

Cross-validation performance $F_{50} \geq 0.8$ across base classifiers.

Model selection rule

We formulate the following model selection rule: for mutations mapping to gene G and cohort C, the model of choice will be the (G,T)-model where T is the most specific ancestor tumor type of C in the Oncotree such that the (G,T)-models meets the minimum quality criteria. If there is no such tumor type, we deem the forecast not feasible for that specific query.

1.5 Validation

We present a series of analyses to validate the in silico saturation mutagenesis approach implemented via **boostDM** models. We compare the performance of **boostDM** models in the classification of observed and synthetic mutations in 5 cancer genes with that of experimental saturation mutagenesis assays on the same genes. The comparison is extended across an important fraction of **boostDM** models to the performance of 8 bioinformatics methods aimed at identifying driver mutations or at assessing the functional impact of variants. Furthermore, we validate the predictions of the method using sets of observed and experimentally validated rare variants that are held out to re-train the models. We assess the ability of models trained and tested on a subset of the original data to classify the remaining held-out mutations, which constitute, in effect, an independent validation dataset. We also assess the ability of the method to reflect pathogenicity and oncogenicity in curated datasets with clinically relevant variants. Finally, we test the enrichment of positive predictions among lower frequency polymorphisms. Collectively, these tests demonstrate the satisfactory execution of **boostDM** models, which systematically outperform saturation mutagenesis experiments and other bioinformatics tools. In this document, we also describe the dependencies between the input data and the performance and complexity of the resulting models using random subsampling tests. The results of this experiment demonstrate that as more cancer sequencing data is available, more good-quality models will be attainable.

1.5.1 Hold-out with experimentally validated rare oncogenic variants

Rationale

We wanted to analyze the ability of the approach employed to train and test boostDM models to infer rules that can be subsequently applied to mutations that were never utilized at learning, i.e. the ability of the method to produce rules that can be extrapolated to biologically plausible, yet unobserved mutations. This is particularly relevant for assessing low-frequency variants of unknown significance.

Methodology

Given a probing set of missense mutations, we re-trained **boostDM** models by holding them out, then we assessed the ability of the re-trained models to correctly classify them. We carried out this analysis with two datasets separately: validated rare mutations reported in [Kim *et al.*, 2016] and [Berger *et al.*, 2016], respectively. These sets comprise mutations which have been both observed in tumors and experimentally validated.

Datasets of experimentally validated rare variants

Kim et al. In [Kim *et al.*, 2016] the authors validated a set of cancer mutations which comprises both high and low frequently observed mutations through in vivo tumor formation assays. Out of the positive set, 71 alleles belonging to the rarely observed category were validated by xenografts in NCR-Nu mice, as measured by the size of the tumors after 130 days. We used the "functional" (set of positive) and "neutral" (set of negative) labels as defined by the authors.

Berger et al. In [Berger *et al.*, 2016] the authors experimentally analyzed 194 mutations observed in lung adenocarcinomas, 69% of which were deemed as impactful via expression-based variant impact phenotyping (eVIP). Resistance to EGFR inhibition by erlotinib in two different concentrations was also measured. We labeled the mutation as "driver" when an eVIP-positive mutation also elicited resistance to EGFR inhibition in both conditions, and "passenger" when the allele was negative in both experiments.

Results

In computing the performance of the re-trained models on the pool of annotated missense mutations we conducted a separate analysis for oncogenes and tumor suppressors. When casting the predictions of re-trained models, only those mutations mapping to genes having a good quality models were assessed (364 variants) using the most specific tumor type models available.

Reported predictive abilities were generally high with slighlty better results for oncogenes:

Kim et al. missense mutations in oncogenes 96 driver, 16 passenger; $F_{50} = 0.95$.

Kim et al. missense mutations in tumor suppressors 47 driver, 55 passenger; $F_{50} = 0.72$.

Berger et al. missense mutations in oncogenes 131 driver, 10 passengers; $F_{50} = 0.92$.

Berger et al. missense mutations in tumor suppressors 0 drivers, 12 passengers; NPV=0.92 (negative predictive value was used instead, as per the zero driver labels).

For a reduced set of genes we also computed their gene-wise performance: those genes for which at least one mutation in each driver and passenger class was reported. See Fig. 1h and Fig. S4a.

1.5.2 Sample-wise hold-out validation

Rationale

We wanted to test boostDM models on a setting close to the real-life scenario of interpretation of mutations in newly sequenced tumors. Such experiment would thus consist on leaving a fraction of the samples of a cohort of a tumor type out of those used in the training and test of the models (i.e., held-out) to use in their subsequent validation. Thus, we held out a random subset representing 10% of the samples from each cohort, the goal being to compare the performance of the re-trained models on the held-out data per gene-tumor type against the cross-validation performance of the original models trained with the full dataset. Specifically, holding out a subset of samples seeks to evaluate how the model is expected to perform in the classification of mutations identified in newly sequenced cancer samples. This is key for the in silico saturation mutagenesis –and by extension, for the interpretation of mutations in newly sequenced cancer genomes– which by definition consists in the classification of yet unobserved mutations.

Methodology

Among the mutations held out, we kept those that were included in the pool of mutations used for training of the full models – with their respective labels. Then we filtered out those mutations not mapping to any of the 185 gene-tumor types covered by good quality re-trained gene models. Considering unique mutations, about 66% were annotated as driver and the rest as passengers.

Matching the predictions of the retrained models with the driver/passenger annotations we could evaluate the hold-out performance (F_{50}) for 161 gene-tumor type pairs. We then compared them with the respective cross-validation performance in the full models.

Results

We compare the hold-out performance with re-trained models of TP53 against the cross-validation assessment yielded by the respective full models (see Fig. S2c). We also show the hold-out vs cross-validation across gene-tumor type pairs for oncogenes and tumor suppressors, separately.

The case of TP53 is paradigmatic of a driver gene with a moderate discovery index, implying a relatively low number of recurrent mutations in sample subsets, yet it has been intensely covered by sequencing, and is frequently affected by driver mutations across many tumor types. This makes it the gene with most good-quality models available for training and most stable under perturbations of the input datasets. Consequently, the observed mutations in TP53 are highly representative across several tumor types. The comparison between our 10% sample-wise hold-out and cross-validation in TP53 is consistent with the view that cross-validation yields a reasonable proxy for the expected performance in yet unobserved sets of mutations, with a comparable performance overall.

Looking across gene-tumor types, the results suggest that cross-validation is a conservative proxy of the sample-wise hold-out validation. This can be explained by the fact that cross-validation assessment reflects the typical performance of base classifiers prior to aggregation, which are expected to be sub-optimal models reflecting only partial classification rules. A segregated analysis within oncogenes and tumor suppressors indicate that the cross-validation may be slightly more conservative in the case of oncogenes (Fig. S2c).

1.5.3 Comparison against experimental and in silico functional scores

Rationale

Our aim in the development of **boostDM** consists in the site by site identification of potential drivers. We have framed it as a supervised classification problem that takes observed mutations in high excess consequence types as positive examples. Observed mutations in these high-excess cancer genes are the best proxy for driver mutations to use as validation dataset. In this regard, tumors may be considered as the only true real-life validation –natural experiment– of oncogenic mutations. Experimental approaches in general assess the effect of mutations in artificial experimental settings and do not measure directly oncogenic potential in human tissues. This is why we have used the observed –and randomized– mutations across tumors as positive and negative sets for validation of saturation mutagenesis approaches, both experimental and in silico (via **boostDM** models or via other bioinformatics tools aimed at the identification of driver mutations or at the assessment of the functional impact of variants).

Methodology

Bearing this in mind, the **boostDM** performance we report to compare with other approaches is derived by keeping as ground truth all the test mutations used for cross-validation of **boostDM** (described above) and measuring the agreement of the prediction cast by each base classifier on its corresponding set of test mutations. As for the other methods (experimental mutagenesis or in silico functional scores) we use the same set of ground truth mutations, keeping only those for which the corresponding method has a defined functional score.

We assessed the performance of a collection of mutation-specific functional scoring methods, comprising experimental saturation mutagenesis assays (TP53, RAS domain and PTEN) and in silico functional scores, then compared them against **boostDM** performance.

First we will introduce the methodology, then for sake of interpretation we provide a brief description of the methods themselves.

Comparison criterion

Although **boostDM** establishes a natural cutoff (0.5) to tell apart predicted drivers from passengers, none of the methods we compare with allows a straightforward dichotomous classification. Hence we evaluate their performance as the area under the precision-recall curve (auPRC) when we compare the method output against a validation driver-passenger-annotated dataset.

Validation Dataset

For a specific gene-tumor type context we defined the validation dataset as the balanced set comprising all test mutation instances across the 50 base base classifier splits. This choice allows an unbiased performance estimation of **boostDM** as we can evaluate the performance of each base classifier on their respective test mutations, which were never used for training. In other words, for each test set included in the validation pool, we got the **boostDM** forecasts with the base classifier resulting from the training set of the same split.

The other scores were mapped using genomic coordinates –and the tumor type of the model in case of the per-tumor-type CHASMplus models. Note that this approach may in principle overestimate the performance of the supervised learning-based bioinformatics tools we evaluate, as we do not prevent mutations that were used at their training from being part of the validation dataset. Finally, each method was evaluated only on the set of validation mutations where the score could be mapped.

Precision-recall curves

For each score we computed the precision-recall curve from the validation dataset using the Python **sklearn.metrics.precision_recall_curve** function, which required a re-scaling step to render all the scores in a 0-1 probability range. We accomplished this with a logistic regression step that used as a single covariate the score and as response the labels of the validation set. For fair comparison, the same step was conducted for all methods, including **boostDM**.

Continuous scores from a dichotomous perspective

For the sake of completeness, in the case of the TP53 (Kato et al. [Kato *et al.*, 2003] and Giacomelli et al. [Giacomelli *et al.*, 2018]) and PTEN (Mighell et al. [Mighell *et al.*, 2018]) experimental saturation mutagenesis assays, CHASMplus and CADD, we also conducted the performance comparison against **boostDM** by using probing cutoffs that yielded a dichotomous forecast (F_{50}).

For TP53 (Kato et al.) we used a cutoff of 50 (a.u.) for the median intensities between the WAF1, MDM2, BAXnWT, h1433sn, AIP1, GADD45, NOXA, P53R2 reporters. For TP53 (Giacomelli et al.) we used the cutoff of 0 (A549 cells, p53NULL + etoposide Z-score) where values below 0 are positive. For PTEN (Mighell et al.) we used a cutoff of -2 (a.u.) for the log-scaled and wild-type normalized fitness scores. For CHASMplus we created two dichotomous classifications based on mild and stringent cutoff levels. These cutoffs were decided upon transformation of the score p-values into q-values by FDR correction across all the CHASMplus-annotated mutations mapping to the same gene-tumor type pair: the stringent (resp. mild) cutoff required a q-value=0 (resp. q-value leq 0.01). For CADD we created a dichotomous classification based on the cutoff 10. We present the results after conducting the comparison for the entire validation set and for each test set separately (Fig. S3).

Experimental saturation mutagenesis assays

We compared the results between **boostDM** and the predictions that would be derived from an experimental saturation mutagenesis assay. Thus we parsed and mapped the outputs of several experimental saturation mutagenesis studies for TP53 [Giacomelli *et al.*, 2018, Kato *et al.*, 2003], mutations in the RAS domain [Bandaru *et al.*, 2017], and PTEN [Mighell *et al.*, 2018], then analyzed the concordance between these scores and **boostDM** across all the tumor types where a **boostDM** model was available.

Datasets

Kato et al. [Kato *et al.*, 2003] TP53 data. In the experiment, a site-directed mutagenesis was employed to construct all possible missense substitutions in the p53 protein (2569 nucleotide substitutions, 2314 different amino acid variants), followed by a promoter-specific transcriptional activity in yeast-based functional assays. Non functional sites were considered when less than the 50% wildtype transactivation was observed, as measured by the median of WAF1, MDM2, BAX, h1433s, AIP1, GADD45, NOXA and P53R2 factors.

Giacomelli et al. [Giacomelli *et al.*, 2018] TP53 data. TP53 saturation mutagenesis screens in an isogenic pair of TP53 wild-type and null cell lines. Library comprising 8258 mutant TP53 alleles, such that each allele would contain a single mutation, and each of the 20 natural amino acids and a stop codon would be represented at each codon position.

Bandaru et al. [Bandaru *et al.*, 2017] Ras function data. Analysis of a complete set of single-site mutations for the relevant variant of human H-Ras, expressed together with Raf-RBD, in a bacterial two-hybrid system.

Mighell et al. [Mighell *et al.*, 2018] PTEN data. Parallel approach in an artificial humanized yeast model to evaluate the impact of 7244 amino acid PTEN variants in the lipid phosphatase activity. The final score is defined by the mean of the functional score defined by the authors out of six experiments (two biological replicates with three technical replicates each).

Other comparisons

There are other experimental saturation mutagenesis studies that in principle would provide us with interesting comparisons to further validate and interpret our methodology. This is the case of a recent study on BRCA1 [Findlay *et al.*, 2018]. However, for comparisons to render meaningful outcomes, we require our models to reach a minimum quality for the genes concerned. In Supp. Table 2 we provide specific information about the 2080 starting cancer gene-tissue combinations entering our pipeline. The numbers provided in this Table (in particular the label "effective mutations", i.e., mutations available for training once the excess for each consequence type across each cohort of the tumor type is accounted for) constitute the rationale behind the model building flow diagram in Figure S1a. As of the preparation of this manuscript, specific models for some genes (e.g. BRCA1) did still not attain the minimum requirements: BRCA1 in Breast Adenocarcinoma only attains 7 observed mutations compatible with high excess that can be effectively used for training, thus an insufficient number to yield the 30 mutations required for the training sets. This precludes any comparisons until new somatic mutation data is available for training.

In silico functional scores

CHASMplus [Tokheim and Karchin, 2019] We run CHASMplus with the following 15 tumor type specific models according to the TCGA ontology: Bladder urothelial carcinoma, Cervical squamous cell carcinoma and endocervical adenocarcinoma, Colorectal adenocarcinoma, Esophageal carcinoma, Gliobastoma, Acute myeloid leukemia, Liver hepatocellular carcinoma, Lung adenocarcinoma, Lung squamous cell carcinoma, Ovarian serous cystadenocarcinoma, Pancreatic adenocarcinoma, Prostate adenocarcinoma, Skin cutaneous melanoma, Thyroid carcinoma, Uterine corpus endometrial carcinoma.

CHASMplus scores missense mutations observed in pan-cancer and significantly mutated driver genes via a supervised learning approach employing a set of 95 features. CHASMplus generates models encompassing all

cancer genes on a tissue specific basis. For each of the 185 gene-tumor type pair good quality models, we mapped the canonical transcript specific scores of CHASMplus onto the test mutations. Note that CHASMplus scores might in principle reflect the features of the test mutations employed in our testing setting, as they were not explicitly removed from its training set, which might bear overestimation of the method's performance.

We also downloaded from **dbNSFP** repository (version 4.1, [Liu *et al.*, 2020]) the precomputed scores for the following methods. Herein we briefly summarize the scope of each method:

CADD [Rentzsch *et al.*, 2019] Tool for scoring the deleteriousness of SNVs in the human genome, based on the distinction between simulated de novo variants and polymorphisms that have arisen and become fixed in human populations since the split between humans and chimpanzees.

SIFT and SIFT4G [Ng and Henikoff, 2003, Vaser *et al.*, 2016] Tools for scoring deleteriousness of missense amino acid substitutions based on the position and type of the amino acid change. From protein alignment information, SIFT calculates the probability that an amino acid at a position is tolerated, conditioning on the most amino acid being tolerated.

Polyphen-2 (HDIV and HVAR) [Adzhubei *et al.*, 2013] Predicts the stability and functional impact of amino acid substitutions in human proteins by exploiting structural and comparative evolutionary information.

FATHMM [Shihab *et al.*, 2013] Predicts the potentially deleteriousness of amino acid substitutions using a Hidden Markov Model-based probabilistic approach.

VEST4 [Carter *et al.*, 2013] Prioritizes likely pathogenic missense variants with a supervised learning based on the distinction between curated disease mutations and high allele frequency putatively neutral missense variants.

MutationAssessor [Reva *et al.*, 2011] Introduces a functional impact score for amino acid substitutions by exploiting evolutionary conservation patterns. These patterns are derived from aligned families and subfamilies of sequence homologs within and between species using a combinatorial entropy formalism.

1.5.4 Classification of manually curated pathogenic variants

We also used **boostDM** models to classify somatic pathogenic and germline benign somatic variants collected across the literature by ClinVar [Landrum *et al.*, 2014], a database that annotates human variation with phenotypic consequence. Although these mutations are biased towards those that have been observed more recurrently, and their annotation as oncogenic do not necessarily need to be maintained across tumor types, we use them as a further exercise of validation of **boostDM** models.

For each mutation in these dataset, **boostDM** predictions were cast using all the tumor type models corresponding to genes where a sufficiently high quality gene-tumor type models was available. We measured the precision, recall and F_{50} based on the ability of **boostDM** to correctly separate somatic pathogenic (ClinVar terms *pathogenic* and *likely pathogenic*) from germline benign (ClinVar terms *benign* and *likely benings*) variants.

1.5.5 Enrichment in low-frequency polymorphisms

We reasoned that applying **boostDM** models to germline variants detected across human populations should identify potential driver mutations depleted for frequent variants, since these have passed purifying selection. We downloaded all SNPs from gnomAD (release 2.1.1, [Karczewski *et al.*, 2020]) and mapped them to genes with **boostDM** good quality model. We computed the driver/passenger **boostDM** class. We tested the association between **boostDM** class (response) against the log allele frequency of the SNPs with a logistic regression. For each gene-tumor type with mapped SNPs we computed the p-value and effect size (log-odds-ratio) representing the propensity of lower frequency values being more enriched in potential driver mutations according to **boostDM**.

1.6 Discovery index

For the sake of completeness, we provide the rationale behind the exponential regression function we used at defining the discovery index.

1.6.1 A simple urn model

Consider an urn containing N different mutations that can occur in some gene. Each mutation has in addition one of two flavours: driver or passenger. There are d driver and p passenger mutations, hence N=p+d. If we sample n times uniformly with replacement one mutation at a time from the urn, let's denote E(n) the expected number of driver mutations drawn at least one time. We aim to provide an expression to this function.

There are two trivial base cases: E(0)=0 and E(1)=d/N. For the general case n>1 observe that we can exploit the following recurrence:

$$E(n) = E(n-1) + \frac{1}{n} \cdot (d - E(n-1)) = \frac{N-1}{N} \cdot E(n-1) + \frac{d}{N}$$

i.e., the expectation at step n is the expectation at step n-1 plus the probability to draw a mutation that remained unobserved after step n-1. Letting a=(N-1)/N this recurrence yields the following expression for E(n):

$$E(n) = \frac{d}{N} \cdot \frac{1-a^n}{1-a} = d \cdot \left[1 - \left(\frac{N-1}{N}\right)^n\right]$$

Suppose that the probability δ to draw any particular driver is uniform across drivers, but not necessarily equal to 1/N. We can build a similar recurrence:

$$E(n) = E(n-1) + \delta \cdot (d - E(n-1)) = (1-\delta) \cdot E(n-1) + \delta \cdot d$$

Whence the following expression follows if we let $a = 1 - \delta$:

$$E(n) = \delta \cdot d \cdot \frac{1-a^n}{1-a} = d \cdot [1 - (1-\delta)^n] \,.$$

For a small enough δ, notice that the following approximation holds:

$$E(n) \approx d \cdot [1 - \exp(-\delta \cdot n)]$$

All in all, the previous expressions can be both put into exponential form:

$$E(n) \approx d \cdot [1 - \exp(\beta \cdot n)]$$

where β can represent either $\log((N-1)/N)$ or $\log(1-d)$, depending on the case.

This is the expression we used to define an index to measure the extent to which more driver mutations are to be discovered provided more sequenced tumors. Intuitively, d is the asymptotic value that the function E(n) would take for n large, i.e., the total number of driver mutations to be discovered in the Urn. Notice also that the smaller is β the faster the exponential term approaches zero, i.e., the fewer samples are required to approach the asymptotic value.

1.7 Pipeline

1.7.1 Availability

A public version of the pipeline is available in this git repo.

1.7.2 Pre-requisites and running

Follow the instructions of the repo README.

1.7.3 Input

The full pipeline feeds on several data sources, mainly coming from IntOGen pipeline, but also others. A full installation of IntOGen is required to properly handle the full preprocessing of mutations and features that ultimately yields the training data.

For the public version we provide a simplified pipeline that starts with a preprocessed input that summarizes the training data alongside all the necessary feature annotations, driver genes and cohort data.

regression data

`create_datasets/<cohort-code>.regression_data.tsv`

Collection of positive and negative training mutations alongside their features for a given cohort.

For each mutation it encompasses the following information:

chr: chromosome

pos: position in genomic coordinates (hg38)

ref: reference allele

gene: gene

alt: alternate allele

PhyloP: PhyloP conservation score,

aachange: amino acid change

nmd: whether a stop mutation maps to the last coding exon

Acetylation, Methylation, Phosphorylation, Regulatory_Site, Ubiquitination: whether the mutation maps to a residue subject to post-translational modification.

CLUSTL_SCORE, CLUSTL_cat_1, CLUSTL_cat_2: oncodriveCLUSTL scores

motif: whether the mutation maps to a significantly enriched Pfam domain according to smRegions,

csqn_type_missense, csqn_type_nonsense, csqn_type_splicing, csqn_type_synonymous: simplified protein coding consequence type

HotMaps_cat_1, HotMaps_cat_2: HotMAPs features (tumor type specific and pan-cancer)

smRegions_cat_1, smRegions_cat_2: smRegions features (tumor type specific and pan-cancer)

role_Act, role_LoF: oncogenic mode of action of the gene (oncogene or tumor-suppresor)

response: whether the mutation is labeled as positive or negative mutation for supervised learning

saturation annotation files

`saturation/annotation/<tumor-type>/<gene>.annotated.out.gz`

Comprehensive catalogue of all the mutations mapping to the canonical transcript for each gene in each tumor type context according to the results of IntOGen pipeline, VEP and PhosphositePlus.

For each mutation it encompasses the following information:

chr, pos hg38 genomic coordinates

alt alternate allele

ENSEMBL_GENE, ENSEMBL_TRANSCRIPT, Feature_type, cDNA_position, CDS_position, Protein_position, Amino_acids, Codons Existing_variation, IMPACT, DISTANCE, STRAND, FLAGS, gene, SYMBOL_SOURCE, HGNC_ID, CANONICAL, ENSP, EXON, INTRON

customary VEP annotations, a description can be found here: VEP feature description

boostDM features: PhyloP, aachange, nmd, Acetylation, Methylation, Phosphorylation, Regulatory_Site, Ubiquitination, CLUSTL_SCORE, motif, csqn_type_missense, csqn_type_nonsense, csqn_type_splicing, csqn_type_synonymous, CLUSTL_cat_1, CLUSTL_cat_2, HotMaps_cat_1, HotMaps_cat_2, smRegions_cat_1, smRegions_cat_2

role_Act, role_LoF: one hot encoding of the tumorigenic mode of action

drivers

`datasets/drivers.tsv`

Lists the driver genes that have been found in each cohort by *IntOGen <https://www.intogen.org>_*.

Each item has the following fields:

SYMBOL, TRANSCRIPT, COHORT, CANCER_TYPE, METHODS, MUTATIONS, SAMPLES, %_SAMPLES_COHORT, QVALUE_COMBINATION, ROLE, CGC_GENE, CGC_CANCER_GENE, DOMAIN, 2D_CLUSTERS, 3D_CLUSTERS, EXCESS_MIS, EXCESS_NONEXCESS_SPL

Please, check IntOGen FAQs and Extended Documentation.

cohorts

`datasets/cohorts.tsv` provides some more description of each cohort used.

Each item has the following fields:

COHORT, CANCER_TYPE, CANCER_TYPE_NAME, SOURCE, PLATFORM, PROJECT, REFERENCE, TYPE, TREATED, AGE, SAMPLES, MUTATIONS, WEB_SHORT_COHORT_NAME, WEB_LONG_COHORT_NAME

Please, check IntOGen FAQs and Extended Documentation.

oncotree

Tumor type ontology defining high specific tumor type categories and a hierarchical structure to combine them. A detailed descrition is provided in the section *Oncotree: tumor type ontology*.

It consists of two files:

datasets/tree_cancer_types.json: ontology hierarchy as a JSON dictionary

datasets/definitions.json: definitions of all the tumor type acronyms used in the oncotree

discovery

discovery/discovery.tsv

Pre-computed summary information for each gene and tumor type where the following features are provided:

gene, ttype: gene and tumor type

n_muts: number of observed mutations in gene in samples matching the tumor type; more general terms in the oncotree ontology will comprise more samples

n_unique_muts: among the mutations considered for n_muts, how many unique mutations are comprised

n_samples: total number of samples matching tumor type with at least one mutation in the gene

discovery_index: discovery index corresponding to each gene and tumor type

discovery_high: discovery index *upper* (IQR) confidence bound

discovery_low: discovery index *lower* (IQR) confidence bound

1.7.4 Output

Data splits for training

splitcv/<cohort-id>.cvdata.pickle.gz

Dictionaries indexed by the set of driver genes in the cohort.

For each gene there is a list of tuples (as many elements as base models) with 4 elements each:

- x_train (pandas.DataFrame, feature data per training instance)
- x_test (pandas.DataFrame, feature data per testing instance)
- y_train (pandas.Series, response labels per training instance, binary)
- y_test (pandas.Series, response labels per testing instance, binary)

The feature information has the following columns:

chr, pos, ref, alt, CLUSTL_SCORE, CLUSTL_cat_1, CLUSTL_cat_2, HotMaps_cat_1, HotMaps_cat_2, smRegions_cat_1, smRegions_cat_2, PhyloP, nmd, Acetylation, Methylation, Phosphorylation, Regulatory_Site, Ubiquitination, csqn_type_missense, csqn_type_nonsense, csqn_type_splicing, csqn_type_synonymous, role_Act, role_LoF

splitcv_meta/<tumor-type>/<gene>.cvdata.pickle.gz

Pickled list of tuples (as many elements as base models) of 4 elements each following the same structure as described above. These are the result of aggregating the splitting information tumor-type-wise.

Gradient boosting base classifiers

`training_meta/<tumor-type>/<gene>.models.pickle.gz`

Pickled dictionaries where the set of trained base models for a given gene and tumor type are kept. They comprise the following levels of information:

models: list of trained base classifiers (instances of boostwrap.methods.Classifier)

x_test: list of pandas.DataFrame instances with feature information used for testing at each base classifier

y_test: list of pandas.Series instances with response labels used for testing at each base classifier

learning_curves: list of dictionaries with performance evaluation vectors at train and test (`validation_0` and `validation_1`) for representation graphical evaluation of the learning progress as a function of the number of estimators for each base classifier.

Evaluation of base classifiers

`evaluation/<tumor-type>/<gene>.eval.pickle.gz`

Performance information of the base models kept separately. Each data instance is a dictionary with correlative lists of values computed for each base classifier. We computed the following performance metrics:

auc: area under the ROC curve

mcc: Matthews correlation coefficient

logloss: Log-loss or cross-entropy

precision: Precision, a.k.a. *positive predictive value* (PPV), true-over-positive rate.

npv: Negative predictive value, i.e. the false-over-negative rate.

recall: Recall, a.k.a. *sensitivity*, i.e. the positive-over-true rate.

fscore: F-score, i.e. harmonic mean of precision and recall.

F_{50} **(fscore50):** F_β with $\beta = 0.5$; in particular, this score primes more precision over recall than the F-score.

accuracy: Accuracy, i.e. rate of correctly classified mutations over all predictions.

balance: Test dataset balance, i.e. deviation of the proportion of labels from 0.5; perfect balance should yield 0.

calibration: Calibrarion, i.e. extent to which the average boostDM predicted score matches the proportion of positive labels – which in case of a balanced set would give 0.5. This is computed as $(\overline{\hat{y}_i} - \overline{y}_i)/\overline{y}_i$, where $\hat{y}_i$ are the boostDM predicted values, y_i are the true labels, and bars denote averaging.

size: Test size, i.e. total number of mutations in the test set.

Saturation predictions

`saturation/prediction/<gene>.<tumor-type>.prediction.tsv.gz`

Predictions for all possible mutations in the canonical transcript for a specific gene in a tumor type context. The file includes the following columns:

gene, ENSEMBL_TRANSCRIPT, ENSEMBL_GENE, chr, pos, alt, aachange, CLUSTL_SCORE, CLUSTL_cat_1, CLUSTL_cat_2, HotMaps_cat_1, HotMaps_cat_2, smRegions_cat_1, smRegions_cat_2, PhyloP, nmdAcetylation, Methylation, Phosphorylation, Regulatory_Site, Ubiquitination, csqn_type_missense, csqn_type_nonsense, csqn_type_splicing, csqn_type_synonymous, role_Act, role_LoF, selected_model_ttype, selected_model_gene, boostDM_score, boostDM_class, shap_CLUSTL_SCORE, shap_CLUSTL_cat_1, shap_CLUSTL_cat_2,

shap_HotMaps_cat_1, shap_HotMaps_cat_2, shap_smRegions_cat_1, shap_smRegions_cat_2, shap_PhyloP, shap_nmd, shap_Acetylation, shap_Methylation, shap_Phosphorylation, shap_Regulatory_Site, shap_Ubiquitination, shap_csqn_type_missense, shap_csqn_type_nonsense, shap_csqn_type_splicing, shap_csqn_type_synonymous, shap_role_Act, shap_role_LoF

Columns with shap_ prefix: They denote the SHAP values corresponding to the prefixed features. The meaning of these values are explained in the section *Shapley Additive Explanations*.

selected_gene_model, selected_model_ttype: Represent the gene and tumor type context that was used to cast the predictions, in other words, which model was employed to cast the predictions in this case.

boostDM_score, boostDM_class: Denote the prediction score and whether this score is higher than 0.5 which the method established as the threshold for driver potential.

Deprecation note: At the moment the columns selected_gene_model, role_Act, role_LoF, shap_role_Act, shap_role_LoF are not informative. They were used in preliminary versions of the pipeline to handle analyses where pools of mutations from distinct genes were used for training meta-gene models. The current approach precludes this analysis and these columns will not be supported in forthcoming versions of the tool.

1.8 References

BIBLIOGRAPHY

[AJS13] I. Adzhubei, D. M. Jordan, and S. R. Sunyaev. Predicting functional effect of human missense mutations using PolyPhen-2. *Curr Protoc Hum Genet*, Chapter 7:Unit7.20, Jan 2013.

[APMMuinos+19] C. Arnedo-Pac, L. Mularoni, F. Muiños, A. Gonzalez-Perez, and N. Lopez-Bigas. OncodriveCLUSTL: a sequence-based clustering method to identify cancer drivers. *Bioinformatics*, 35(22):4788–4790, Nov 2019.

[BSB+17] P. Bandaru, N. H. Shah, M. Bhattacharyya, J. P. Barton, Y. Kondo, J. C. Cofsky, C. L. Gee, A. K. Chakraborty, T. Kortemme, R. Ranganathan, and J. Kuriyan. Deconstruction of the Ras switching cycle through saturation mutagenesis. *Elife*, 07 2017.

[BBW+16] A. H. Berger, A. N. Brooks, X. Wu, Y. Shrestha, C. Chouinard, F. Piccioni, M. Bagul, A. Kamburov, M. Imielinski, L. Hogstrom, C. Zhu, X. Yang, S. Pantel, R. Sakai, J. Watson, N. Kaplan, J. D. Campbell, S. Singh, D. E. Root, R. Narayan, T. Natoli, D. L. Lahr, I. Tirosh, P. Tamayo, G. Getz, B. Wong, J. Doench, A. Subramanian, T. R. Golub, M. Meyerson, and J. S. Boehm. High-throughput Phenotyping of Lung Cancer Somatic Mutations. *Cancer Cell*, 30(2):214–228, 08 2016.

[CDS+13] H. Carter, C. Douville, P. D. Stenson, D. N. Cooper, and R. Karchin. Identifying Mendelian disease genes with the variant effect scoring tool. *BMC Genomics*, 14 Suppl 3:S3, 2013.

[CG16] T. Chen and C. Guestrin. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 785–794. New York, NY, USA, 2016. ACM. URL: http://doi.acm.org/10.1145/2939672.2939785, doi:10.1145/2939672.2939785.

[DTCF+17] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame. Nextflow enables reproducible computational workflows. *Nat. Biotechnol.*, 35(4):316–319, 04 2017.

[EGMB+19] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, E. L. L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. C. E. Tosatto, and R. D. Finn. The Pfam protein families database in 2019. *Nucleic Acids Res.*, 47(D1):D427–D432, 01 2019.

[FDM+18] G. M. Findlay, R. M. Daza, B. Martin, M. D. Zhang, A. P. Leith, M. Gasperini, J. D. Janizek, X. Huang, L. M. Starita, and J. Shendure. Accurate classification of BRCA1 variants with saturation genome editing. *Nature*, 562(7726):217–222, 10 2018.

[GYL+18] A. O. Giacomelli, X. Yang, R. E. Lintner, J. M. McFarland, M. Duby, J. Kim, T. P. Howard, D. Y. Takeda, S. H. Ly, E. Kim, H. S. Gannon, B. Hurhula, T. Sharpe, A. Goodale, B. Fritchman, S. Steelman, F. Vazquez, A. Tsherniak, A. J. Aguirre, J. G. Doench, F. Piccioni, C. W. M. Roberts, M. Meyerson, G. Getz, C. M. Johannessen, D. E. Root, and W. C. Hahn. Mutational processes shape the landscape of TP53 mutations in human cancer. *Nat Genet*, 50(10):1381–1387, 10 2018.

[HZM+15] P. V. Hornbeck, B. Zhang, B. Murray, J. M. Kornhauser, V. Latham, and E. Skrzypek. PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. *Nucleic Acids Res.*, 43(Database issue):D512–520, Jan 2015. https://www.phosphosite.org/ (release: 4 October 2018).

[KSO+05] Y. Kakudo, H. Shibata, K. Otsuka, S. Kato, and C. Ishioka. Lack of correlation between p53-dependent transcriptional activity and the ability to induce apoptosis among 179 mutant p53s. *Cancer Res.*, 65(6):2108–2114, Mar 2005.

[KFT+20] K. J. Karczewski, L. C. Francioli, G. Tiao, B. B. Cummings, J. Alföldi, Q. Wang, R. L. Collins, K. M. Laricchia, A. Ganna, D. P. Birnbaum, L. D. Gauthier, H. Brand, M. Solomonson, N. A. Watts, D. Rhodes, M. Singer-Berk, E. M. England, E. G. Seaby, J. A. Kosmicki, R. K. Walters, K. Tashman, Y. Farjoun, E. Banks, T. Poterba, A. Wang, C. Seed, N. Whiffin, J. X. Chong, K. E. Samocha, E. Pierce-Hoffman, Z. Zappala, A. H. O'Donnell-Luria, E. V. Minikel, B. Weisburd, M. Lek, J. S. Ware, C. Vittal, I. M. Armean, L. Bergelson, K. Cibulskis, K. M. Connolly, M. Covarrubias, S. Donnelly, S. Ferriera, S. Gabriel, J. Gentry, N. Gupta, T. Jeandet, D. Kaplan, C. Llanwarne, R. Munshi, S. Novod, N. Petrillo, D. Roazen, V. Ruano-Rubio, A. Saltzman, M. Schleicher, J. Soto, K. Tibbetts, C. Tolonen, G. Wade, M. E. Talkowski, B. M. Neale, M. J. Daly, D. G. MacArthur, C. A. Aguilar Salinas, T. Ahmad, C. M. Albert, D. Ardissino, G. Atzmon, J. Barnard, L. Beaugerie, E. J. Benjamin, M. Boehnke, L. L. Bonnycastle, E. P. Bottinger, D. W. Bowden, M. J. Bown, J. C. Chambers, J. C. Chan, D. Chasman, J. Cho, M. K. Chung, B. Cohen, A. Correa, D. Dabelea, M. J. Daly, D. Darbar, R. Duggirala, J. Dupuis, P. T. Ellinor, R. Elosua, J. Erdmann, T. Esko, M. Färkkilä, J. Florez, A. Franke, G. Getz, B. Glaser, S. J. Glatt, D. Goldstein, C. Gonzalez, L. Groop, C. Haiman, C. Hanis, M. Harms, M. Hiltunen, M. M. Holi, C. M. Hultman, M. Kallela, J. Kaprio, S. Kathiresan, B. J. Kim, Y. J. Kim, G. Kirov, J. Kooner, S. Koskinen, H. M. Krumholz, S. Kugathasan, S. H. Kwak, M. Laakso, T. Lehtimäki, R. J. F. Loos, S. A. Lubitz, R. C. W. Ma, D. G. MacArthur, J. Marrugat, K. M. Mattila, S. McCarroll, M. I. McCarthy, D. McGovern, R. McPherson, J. B. Meigs, O. Melander, A. Metspalu, B. M. Neale, P. M. Nilsson, M. C. O'Donovan, D. Ongur, L. Orozco, M. J. Owen, C. N. A. Palmer, A. Palotie, K. S. Park, C. Pato, A. E. Pulver, N. Rahman, A. M. Remes, J. D. Rioux, S. Ripatti, D. M. Roden, D. Saleheen, V. Salomaa, N. J. Samani, J. Scharf, H. Schunkert, M. B. Shoemaker, P. Sklar, H. Soininen, H. Sokol, T. Spector, P. F. Sullivan, J. Suvisaari, E. S. Tai, Y. Y. Teo, T. Tiinamaija, M. Tsuang, D. Turner, T. Tusie-Luna, E. Vartiainen, M. P. Vawter, J. S. Ware, H. Watkins, R. K. Weersma, M. Wessman, J. G. Wilson, and R. J. Xavier. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature*, 581(7809):434–443, 05 2020.

[KHL+03] S. Kato, S. Y. Han, W. Liu, K. Otsuka, H. Shibata, R. Kanamaru, and C. Ishioka. Understanding the function-structure and function-mutation relationships of p53 tumor suppressor protein by high-resolution missense mutation analysis. *Proc. Natl. Acad. Sci. U.S.A.*, 100(14):8424–8429, Jul 2003.

[KKO+05] T. Kawaguchi, S. Kato, K. Otsuka, G. Watanabe, T. Kumabe, T. Tominaga, T. Yoshimoto, and C. Ishioka. The relationship among p53 oligomer formation, structure and transcriptional activity using a comprehensive missense mutation library. *Oncogene*, 24(46):6976–6981, Oct 2005.

[KIS+16] E. Kim, N. Ilic, Y. Shrestha, L. Zou, A. Kamburov, C. Zhu, X. Yang, R. Lubonja, N. Tran, C. Nguyen, M. S. Lawrence, F. Piccioni, M. Bagul, J. G. Doench, C. R. Chouinard, X. Wu, L. Hogstrom, T. Natoli, P. Tamayo, H. Horn, S. M. Corsello, K. Lage, D. E. Root, A. Subramanian, T. R. Golub, G. Getz, J. S. Boehm, and W. C. Hahn. Systematic Functional Interrogation of Rare Cancer Variants Identifies Oncogenic Alleles. *Cancer Discov*, 6(7):714–726, 07 2016.

[KWJ+14] M. Kircher, D. M. Witten, P. Jain, B. J. O'Roak, G. M. Cooper, and J. Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nat. Genet.*, 46(3):310–315, Mar 2014.

[LLR+14] M. J. Landrum, J. M. Lee, G. R. Riley, W. Jang, W. S. Rubinstein, D. M. Church, and D. R. Maglott. ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res.*, 42(Database issue):D980–985, Jan 2014.

[LSL16] R. G. Lindeboom, F. Supek, and B. Lehner. The rules and impact of nonsense-mediated mRNA decay in human cancers. *Nat Genet*, 48(10):1112–1118, 10 2016.

[LLM+20] X. Liu, C. Li, C. Mou, Y. Dong, and Y. Tu. dbNSFP v4: a comprehensive database of transcript-specific functional predictions and annotations for human nonsynonymous and splice-site SNVs. *Genome Med*, 12(1):103, Dec 2020.

[LL17] S. M. Lundberg and S. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[MartinezJimenezMuinosLopezA+20] F. Martínez-Jiménez, F. Muiños, E. López-Arribillaga, Lopez-Bigas N., and Gonzalez-Perez A. Systematic analysis of alterations in the ubiquitin proteolysis system reveals its contribution to driver mutations in cancer. *Nat. Cancer*, 1:122–135, 2020.

[MRG+17] I. Martincorena, K. M. Raine, M. Gerstung, K. J. Dawson, K. Haase, P. Van Loo, H. Davies, M. R. Stratton, and P. J. Campbell. Universal Patterns of Selection in Cancer and Somatic Tissues. *Cell*, 171(5):1029–1041, Nov 2017.

[MJMS+20] F. Martínez-Jiménez, F. Muiños, I. Sentís, J. Deu-Pons, I. Reyes-Salazar, C. Arnedo-Pac, L. Mularoni, O. Pich, J. Bonet, H. Kranas, A. Gonzalez-Perez, and N. Lopez-Bigas. A compendium of mutational cancer driver genes. *Nat Rev Cancer*, 20(10):555–572, 10 2020.

[MGH+16] W. McLaren, L. Gil, S. E. Hunt, H. S. Riat, G. R. Ritchie, A. Thormann, P. Flicek, and F. Cunningham. The Ensembl Variant Effect Predictor. *Genome Biol.*, 17(1):122, 06 2016.

[MEDORoak18] T. L. Mighell, S. Evans-Dutson, and B. J. O'Roak. A Saturation Mutagenesis Approach to Understanding PTEN Lipid Phosphatase Activity and Genotype-Phenotype Relationships. *Am. J. Hum. Genet.*, 102(5):943–955, 05 2018.

[MuinosMJP+21] F. Muiños, F. Martinez-Jimenez, O. Pich, A. Gonzalez-Perez, and N. Lopez-Bigas. In silico saturation mutagenesis of cancer genes. *Nature*, 2021.

[NH03] P. C. Ng and S. Henikoff. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Res*, 31(13):3812–3814, Jul 2003.

[PHRS10] K. S. Pollard, M. J. Hubisz, K. R. Rosenbloom, and A. Siepel. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.*, 20(1):110–121, Jan 2010.

[RWC+19] P. Rentzsch, D. Witten, G. M. Cooper, J. Shendure, and M. Kircher. CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Res*, 47(D1):D886–D894, 01 2019.

[RAS11] B. Reva, Y. Antipin, and C. Sander. Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic Acids Res*, 39(17):e118, Sep 2011.

[SBF+14] Ville A. Satopää, Jonathan Baron, Dean P. Foster, Barbara A. Mellers, Philip E. Tetlock, and Lyle H. Ungar. Combining multiple probability predictions using a simple logit model. *International Journal of Forecasting*, 30(2):344–356, 2014. URL: https://EconPapers.repec.org/RePEc:eee:intfor:v:30:y:2014:i:2:p:344-356.

[Sha88] Lloyd S. Shapley. *A value for n-person games*, pages 31–40. Cambridge University Press, 1988. doi:10.1017/CBO9780511528446.003.

[SGC+13] H. A. Shihab, J. Gough, D. N. Cooper, P. D. Stenson, G. L. Barker, K. J. Edwards, I. N. Day, and T. R. Gaunt. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Hum Mutat*, 34(1):57–65, Jan 2013.

[TBN+16] C. Tokheim, R. Bhattacharya, N. Niknafs, D. M. Gygax, R. Kim, M. Ryan, D. L. Masica, and R. Karchin. Exome-Scale Discovery of Hotspot Mutation Regions in Human Cancer Using 3D Protein Structure. *Cancer Res.*, 76(13):3719–3731, 07 2016.

[TK19] C. Tokheim and R. Karchin. CHASMplus Reveals the Scope of Somatic Missense Mutations Driving Human Cancers. *Cell Syst*, 9(1):9–23, 07 2019.

[VAL+16] R. Vaser, S. Adusumalli, S. N. Leng, M. Sikic, and P. C. Ng. SIFT missense predictions for genomes. *Nat Protoc*, 11(1):1–9, Jan 2016.