Methodology

If a user has some chemical compounds dataset consisting of compound names, compound SMILEs and their activation statuses (0 or 1) on a target (here, 0 indicates that chemical compound will not activate the target, 1 indicates that chemical compound will activate the target), then below described machine learning pipeline with various parametrized settings can help the user to create machine learning-based binary classification models and use these models to discover novel chemical compounds and predict their ability to activate the target.

There are a total of 6 steps with various parameterized settings that can help the user to discover novel compounds similar to activating compounds in their training dataset.

Step 1: Provide Compound Dataset

This is a mandatory step where a user provides his/her chemical compound ***dataset*** with activation status on a target. All the classification models are trained and validated on this dataset. The dataset should be a CSV file with exactly three fields namely 1) **Compound Name** (A string indicating the name of the chemical compound), 2) **Compound SMILE (**A string-based SMILE representation of the compound) 3) **Activation Status (**An integer with allowed values as 1 or 0 based on whether the compound will activate a target or not). It is recommended to have the ratio of activating compounds to non-activating compounds to be at least 15%:85%.

Step 2: Feature Generation and Data Preprocessing

Feature Generation

Classification based machine learning algorithms can only work with numeric data, so a tool is needed to convert user compound data uploaded in step 1 from string format (SMILEs representation) to the numeric format. The machine learning pipeline uses PaDEL to generate numeric features (molecular descriptors) from compound SMILES. PaDEL generates 1875 molecular descriptors (1444 1D, 2D descriptors and 431 3D descriptors). The details of all the features can be found *[give document reference here].*

Preprocessing Steps

The generated molecular descriptors (numeric features) may have many data issues like missing values of some descriptors for some of the compounds, same or near same value of a descriptor across all compounds (zero or low variance issue) or there may be a class imbalance between activation statuses of user uploaded compounds. Preprocessing steps helps cleaning and rectify some of these issues before feeding the data to machine learning algorithms for error free training of the prediction model. The machine learning pipeline has 5 preprocessing sub-steps **1) Handling Missing Values:** It helps users to prune features which have missing values above a particular percentage threshold. It is recommended to remove features which have missing values more than 75%. The features which have missing values less than the user specified threshold are imputed with mean of the respective feature. **2) Variance Removal:** It helps remove the features which have variance lower than a threshold as they don't add any discriminatory information to classification algorithms. It is recommended to at least remove the features with 0 variance by keeping the threshold value as 0. **3) Correlation Check:** When two features are correlated above a particular threshold, one of them can be removed as it does not add much additional information gain for the classifier. It is recommended to remove one of the features out of two features having a correlation of at least 95%. **4) Normalization:** If some feature values are on scale of tens and others are on scale of thousands, the classifier will unintentionally give more importance to features having comparatively high scale values. It is thus recommended that users perform data normalization to bring every feature on the same scale. The machine learning pipeline uses sklearn's MinMaxScalar normalization to transform all the features to a range from 0 to 1. **5) Class Imbalance:** It is important to handle class imbalance (ratio of non activating to activating compounds) in a dataset, as without handling, even a simple classifier, which always predicts compounds as non activating, will have great accuracy performance. The machine learning pipeline provides users with a setting called SMOTE (Synthetic Minority Over-Sampling Technique), which is an oversampling technique that generates synthetic new samples of minority class to make the ratio of both activating and inactivating classes balanced.

Step 3: Feature Reduction

There are a total of 1875 numeric features (molecular descriptors) generated by PaDEL and the user uploaded dataset may have only a few hundred samples. The classifier may not learn appropriately when the number of features are very high compared to the number of data samples. It is thus important to filter out features which do not add much information and should only retain most important features which have discrimination power for classification. The machine learning pipeline provides two ways to reduce the number of features. **1) Feature Selection** using Boruta. The boruta method performs a top-down search for relevant features by comparing original attributes' importance with importance achievable at random, estimated using their permuted copies, and progressively eliminating irrelevant features to stabilise that test. Boruta is configured to use the default RandomForestClassifier with 100 estimators (100 trees in forest) **2) Feature Extraction** using PCA. PCA works by using fewer transformed features compared to the original features to explain most of the variance in data (i.e. retain as much information as possible). It is recommended to keep the variance threshold (also called eigen energy to 98% (0.98), meaning the reduced number of features should still be able to explain 98% of variance in the data.

Step 4: Classification

The machine learning pipeline provides users with a choice to create models using various classifiers such as Support Vector Machine (SVM), Extra Tree (ET), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Gradient Boosting Machine (GBM), Random Forest (RF), Multi-Layer Perceptron (MLP). *TODO: Add more details on each classifier about their hyperparameters and bagging settings.*

Step 5: Validation Scheme

3-Fold Cross Validation, 5-Fold Cross Validation, Leave One Out Cross Validation *TODO: Add more details here*

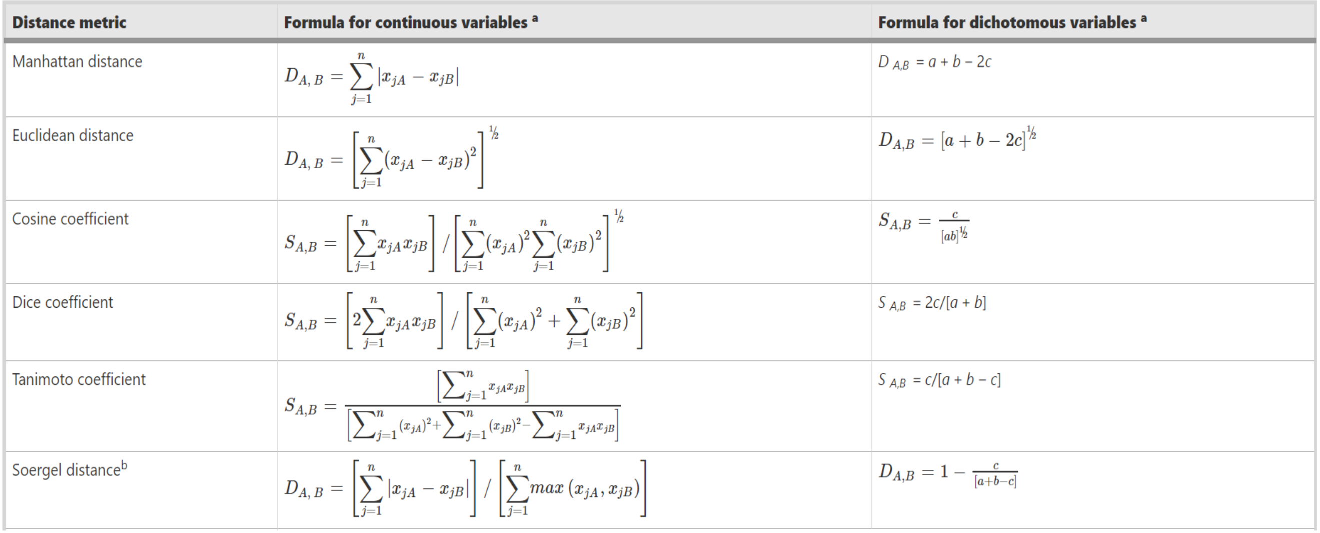
Step 6: Test Set Generation

The software performs a similarity search on various databases (details of databases given below) in order to find the novel compounds which are similar to activating compounds uploaded by the user in his/her dataset. The similarity is measured using various similarity metrics (as described below). The compounds which are found similar (similarity higher than that of a threshold) are tested against the target using the trained classification models and their probability of activating the target is returned as a result. This process helps discover novel compounds which can activate a target with high probability and are similar to activating compounds in training dataset.

The similarity comparison is performed using SMILES of chemical compounds with the help of the RDKit package. For a given chemical compound and it's SMILE, the software is first generating its fingerprint using RDKit Fingerprint. This fingerprint is then compared against the fingerprints of all compounds in user selected databases using different similarity metrics like “Tanimoto”, “Dice”, “Cosine” etc. at different levels of thresholds. The similarity computation is also performed with the help of the RDKit package.

The machine learning pipeline allows users to select any of four databases for search of similar compounds. For each compound in these databases, three information fields namely compound identifier, compound Name and compound SMILES are downloaded and stored. The databases are **1) HMDB (Human Metabolome Database)** is a database having detailed information about small molecule metabolites found in the human body. The current version has information of 1,14,005 metabolites. **2) ChEBI (Chemical Entities of Biological Interest)** is a database of molecular entities focused on 'small' chemical compounds. The current version has information of 1,01,607 small compounds **3) FOODB** is the world's largest and most comprehensive database on food constituents, chemistry and biology. It consists of chemical composition data on common, unprocessed foods. The current version has information on 26,490 food related compounds. **4) IMPPAT (Indian Medicinal Plants, Phytochemistry And Therapeutics)** is the largest database on phytochemicals of Indian medicinal plants to date. The current version of the software consists of information of 1,512 phytochemicals of indian medicinal plants.

The machine learning pipeline also allows users to select any of six similarity/distance metrics to compare similarity between two compounds. Similarity measures work by measuring similarity in molecular fingerprints of the compounds generated by RDKit. List of all the similarity metrics along with their formulas is described in the table below as an image. The fingerprints generated by RDKit are dichotomous in nature and hence the formula corresponding to dichotomous variables columns are used for similarity calculation.



Distance metric	Formula for continuous variables [a]	Formula for dichotomous variables [a]
Manhattan distance	$D_{A,\,B} = \sum_{j=1}^{n} \lvert x_{jA} - x_{jB} \rvert$	$D_{A,B} = a + b - 2c$
Euclidean distance	$D_{A,\,B} = \left[\sum_{j=1}^{n} (x_{jA} - x_{jB})^2 \right]^{1/2}$	$D_{A,B} = [a + b - 2c]^{1/2}$
Cosine coefficient	$S_{A,B} = \left[\sum_{j=1}^{n} x_{jA} x_{jB} \right] / \left[\sum_{j=1}^{n} (x_{jA})^2 \sum_{j=1}^{n} (x_{jB})^2 \right]^{1/2}$	$S_{A,B} = \frac{c}{[ab]^{1/2}}$
Dice coefficient	$S_{A,B} = \left[2 \sum_{j=1}^{n} x_{jA} x_{jB} \right] / \left[\sum_{j=1}^{n} (x_{jA})^2 + \sum_{j=1}^{n} (x_{jB})^2 \right]$	$S_{A,B} = 2c/[a + b]$
Tanimoto coefficient	$S_{A,B} = \frac{\left[\sum_{j=1}^{n} x_{jA} x_{jB} \right]}{\left[\sum_{j=1}^{n} (x_{jA})^2 + \sum_{j=1}^{n} (x_{jB})^2 - \sum_{j=1}^{n} x_{jA} x_{jB} \right]}$	$S_{A,B} = c/[a + b - c]$
Soergel distance[b]	$D_{A,\,B} = \left[\sum_{j=1}^{n} \lvert x_{jA} - x_{jB} \rvert \right] / \left[\sum_{j=1}^{n} max\,(x_{jA}, x_{jB}) \right]$	$D_{A,B} = 1 - \frac{c}{[a+b-c]}$

Here, a is the number of *"on bits"* in molecule A, b is the number of *"on bits"* in molecule B, while c is the number of bits that are on in both molecules. S denotes similarities, while D denotes distances (according to the more commonly used formula for the given metric). x_{jA} means the j-th feature of molecule A. [b]The Soergel distance is the complement of the Tanimoto coefficient. Note that distances and similarities can be converted to one another using below equation

$$similarity = \frac{1}{1 + distance}$$