XGBMut Documentation

XGBMut is a machine learning method with integrated software that uses multiple genetic and molecular data to predict whether a missense mutation is neutral or deleterious together with its probability of being deleterious for the protein's function. XGBMut was trained on a balanced dataset derived from ClinVar (https://www.ncbi.nlm.nih.gov/clinvar/) containing 86.336 labeled mutations and validated using a non-redundant dataset derived from HumsaVar (https://www.uniprot.org/docs/humsavar), which contained 69.887 labeled mutations. XGBMut is an extreme gradient boosting model containing 500 regularized boosting trees, each trained on a different subset of predictor variables. XGBMut presented 82.8% accuracy and 0.9 ROC-AUC in the validation set, which is within the performance range of ten functional prediction algorithms when analyzing the same validation set, including PANTHER (http://www.pantherdb.org/tools/), SNPs&GO (https://snps-and-go.biocomp.unibo.it/snps-and-go/), PolyPhen-2 (http://genetics.bwh.harvard.edu/pph2/), SIFT (https://sift.bii.a-star.edu.sg/), PROVEAN (https://www.jcvi.org/research/provean), FATHMM (http://fathmm.biocompute.org.uk/), PMut (http://mmb.irbbarcelona.org/PMut/), PON-P2 (http://structure.bmc.lu.se/PON-P2/), and MutPred2 (http://mutpred2.mutdb.org/), VariPred (https://github.com/wlin16/VariPred) (Figure1).

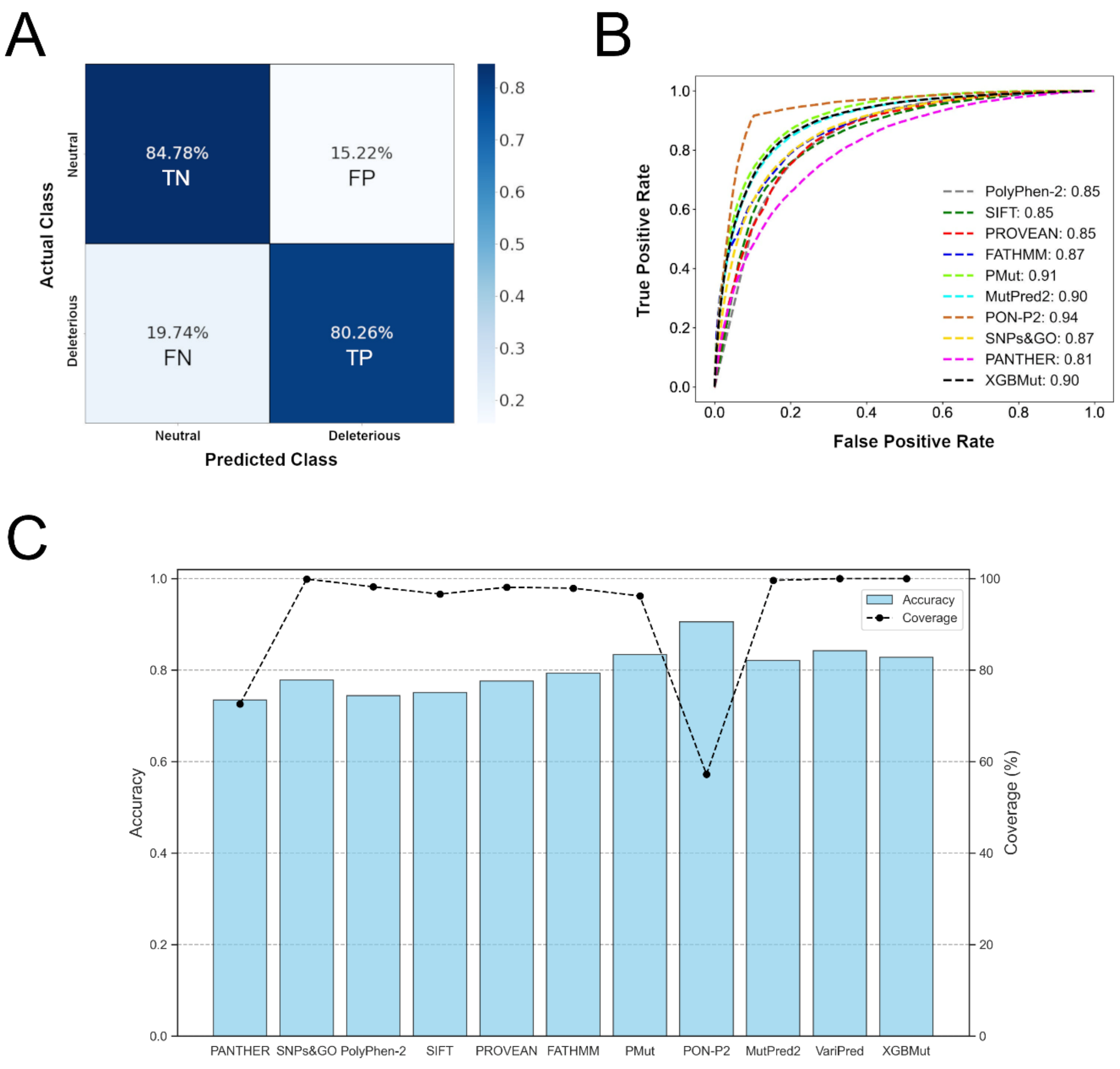


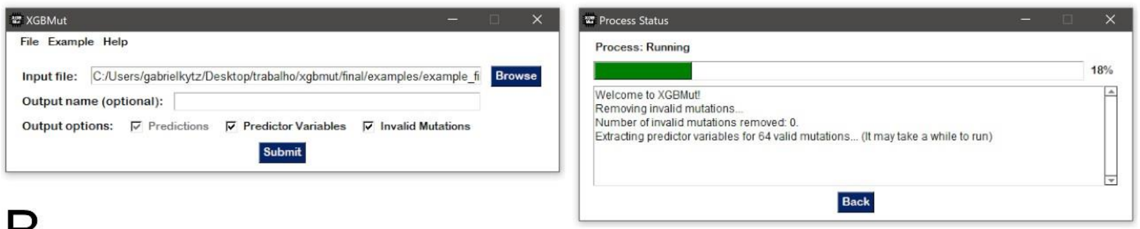
Figure 1. Performance metrics computed for XGBMut in the test set derived from HumsaVar database. (A) Confusion matrix computed for XGBMut. (B) AUC-ROC calculated for

XGBMut and well-established methods. AUC-ROC curves comparing XGBMut with other established methods. VariPred was excluded as it directly outputs the class, without providing probabilities. (C) Accuracy and mutation coverage on the test set for the compared algorithms.

The software has two user interface options: a graphical user interface (GUI) and a command line interface (CLI), which are compatible with ***Windows 10*** and ***Linux/Ubuntu* 20.04** or higher and run within an executable file with no need for a Python interpreter or modules installed. The graphical interface allows the use of XGBMut in an intuitive and user-friendly way, facilitating its use by third parties, including professionals from other areas, such as medical doctors or wet-lab researchers who are not familiar with command line tools. XGBMut was developed in Python using the *Pandas* library for data cleaning and transformation (https://pandas.pydata.org/), in addition to the *xgboost* library (https://xgboost.readthedocs.io/) for creating and deploying the predictive model. PySimpleGui library was used to develop the GUI (https://www.pysimplegui.org/), while *click* library was used to develop the CLI (https://click.palletsprojects.com/). PyInstaller was used to compile and generates the executable file (https://pyinstaller.org/).

The software can be divided into two main parts: an initial menu where the input file is loaded and the customization options selected, as well as a loading menu containing information about the algorithm's progress (Fig3). In addition to the standard analyzes, XGBMut allows output customizations, so that the user can choose the file name and which files will be generated. To guide the user experience, two usage examples are available within the software options. A help menu containing essential information about XGBMut and the required file formats was also provided. The maximum RAM usage by the software was estimated at 3GB in the tests performed, which enables its use in most computers currently available. XGBMut does not require installation, as the executable file already includes the Python interpreter and the necessary modules compiled in a single file. The software is available at the link: https://github.com/gabrielkytz2/XGBMut/.

A



Input file: C:/Users/gabrielkytz/Desktop/trabalho/xgbmut/final/examples/example_fi

Output name (optional):

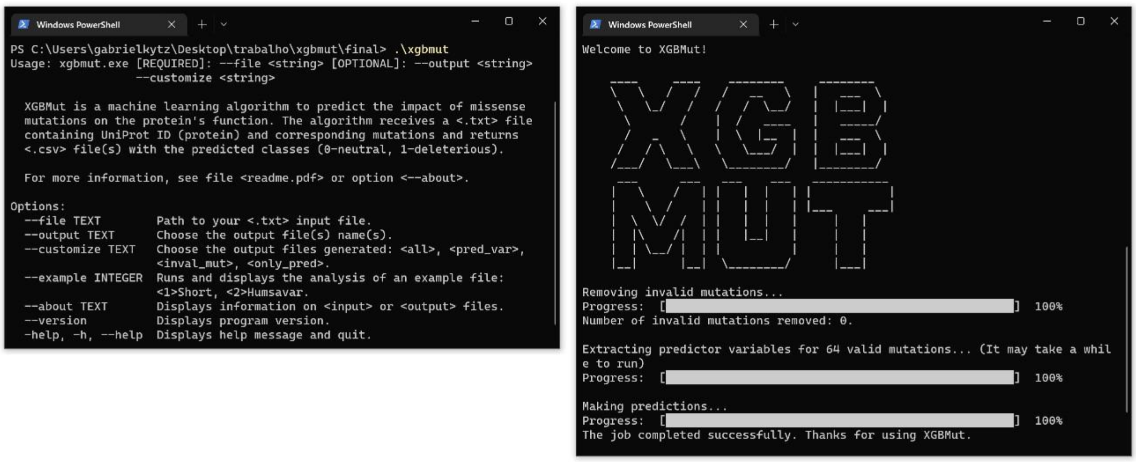
Output options: [✓ Predictions] [✓ Predictor Variables] [✓ Invalid Mutations]

Process: Running

18%

Welcome to XGBMut!
Removing invalid mutations...
Number of invalid mutations removed: 0.
Extracting predictor variables for 64 valid mutations... (It may take a while to run)

B



```
PS C:\Users\gabrielkytz\Desktop\trabalho\xgbmut\final> .\xgbmut
Usage: xgbmut.exe [REQUIRED]: --file <string> [OPTIONAL]: --output <string>
                  --customize <string>

  XGBMut is a machine learning algorithm to predict the impact of missense
  mutations on the protein's function. The algorithm receives a <.txt> file
  containing UniProt ID (protein) and corresponding mutations and returns
  <.csv> file(s) with the predicted classes (0-neutral, 1-deleterious).

  For more information, see file <readme.pdf> or option <--about>.

Options:
  --file TEXT         Path to your <.txt> input file.
  --output TEXT       Choose the output file(s) name(s).
  --customize TEXT    Choose the output files generated: <all>, <pred_var>,
                      <inval_mut>, <only_pred>.
  --example INTEGER   Runs and displays the analysis of an example file:
                      <1>Short, <2>Humsavar.
  --about TEXT        Displays information on <input> or <output> files.
  --version           Displays program version.
  -help, -h, --help   Displays help message and quit.
```

```
Welcome to XGBMut!

   ____    ____   ________   ________
   \   \  /   /  /   __   \ |   ___  \
    \   \/   /  /   /  \__/ |  |___|  |
     \      /  |   |  _____ |   _____/
     /  _   \  |   | |__   ||  |___  \
    /  / \   \  \   \___/  ||  |___|  |
   /__/   \___\  \________/ |________/
    ___    ___  ___   ___  ___________
   |   \  /   ||   | |   ||           |
   |    \/    ||   | |   ||___     ___|
   |  \    /  ||   | |   |    |   |
   |  |\  /|  ||   |_|   |    |   |
   |  | \/ |  ||         |    |   |
   |__|    |__| \_______/     |___|

Removing invalid mutations...
Progress:  [                                        ]  100%
Number of invalid mutations removed: 0.

Extracting predictor variables for 64 valid mutations... (It may take a whil
e to run)
Progress:  [                                        ]  100%

Making predictions...
Progress:  [                                        ]  100%
The job completed successfully. Thanks for using XGBMut.
```

Figure2. XGBMut interface is available in both Windows and Linux platforms. (A) Command line interface (CLI). (B) Graphical User Interface.

XGBMut requires as the input a ***.txt*** file with two columns separated by a tab, containing: the UniProt ID of the affected protein, and the corresponding amino acid substitution under the one-letter code notation. Two input file examples are provided in the *<examples>* folder, *i.e.* ***example_file.txt*** *and* ***humsavar_example.txt***. The algorithm initially checks for and removes invalid mutations and protein IDs, which can be written into ***invalid_mutations.csv*** (optional). Then, XGBMut search for the corresponding predictor variables in the databases files, which can be saved into ***predictor_variables.csv*** (optional). At last, the algorithm predicts whether the mutation is neutral (0) or deleterious (1), together with its probability of being deleterious, which are finally written into ***output.csv***. See ***usage options*** below.

XGBMut was developed by Gabriel Rodrigues Coutinho Pereira (https://www.linkedin.com/in/gabriel-rodrigues-coutinho-pereira-biomedico/) at the University of São Paulo, Brazil, as a final project required for obtaining the title of Master in Business Administration in Data Science and Analytics. The project was oriented by professor Ricardo Limongi França Coelho, PhD.

Usage option (CLI):

Usage: xgbmut.exe [REQUIRED]: --file <string> [OPTIONAL]: --output <string>
--customize <string>

Options:

```
--file TEXT        Path to your <.txt> input file.
--output TEXT      Choose the output file(s) name(s).
--customize TEXT   Choose the output files generated: <all>, <pred_var>,
                   <inval_mut>, <only_pred>.
--example INTEGER  Runs and displays the analysis of an example file:
                   <1>Short, <2>Humsavar.
--about TEXT       Displays information on <input> or <output> files.
--version          Displays program version.
-help, -h, --help  Displays help message and quit.
```

Input file:

The input file must be a <.txt> file separated by a tab containing Uniprot ID and mutation, one mutation per row, as follows:

<Uniprot ID><tab><Mutation>

The mutation must be written in one letter code for amino acid:

<wild-type amino acid><position><mutated amino acid>

Example:

```
P04217	H52R
P04217	H395R
Q9NQ94	V555M
Q9NQ94	A558S
P01023	R704H
P01023	C972Y
```

Example inputs are provided in the <examples> folder.

Output file:

XGBMut generates up to three <.csv> files as output:

1- <predictions.csv>: a 4 columns file containing the UniProt ID, mutation, predicted class (0-neutral, 1-deleterious), and probability of being deleterious (always generated).

2- <predictor_variables.csv>: a 27 columns file containing the UniProt ID, mutation, and 25 predictor variables needed to run the predictive model.

3- <invalid_mutations.csv>: a 2 columns file containing the UniProt ID and mutation for all invalid mutations submitted to XGBMut.

The outputs can be customized by choosing <--output> and <--customize> options.

--output <string>: choose output file(s) name(s).

--customize <string>: choose output file(s) generated:

<all>: predictions.csv, predictor_variables.csv, invalid_mutations.csv.

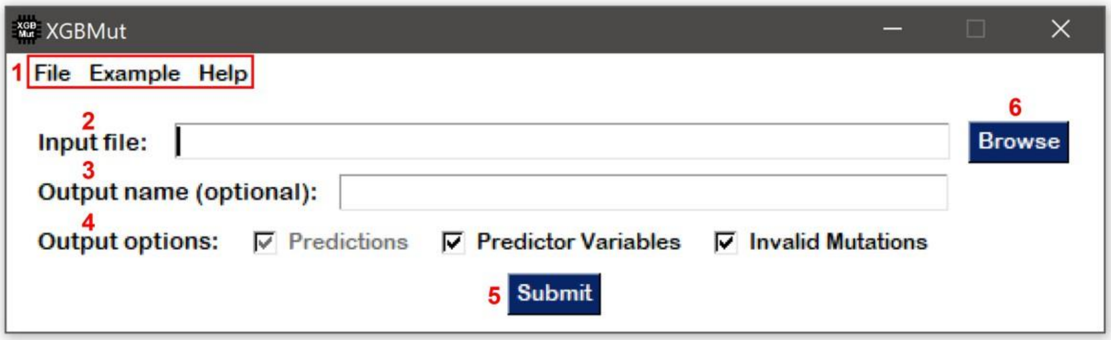
<pred_var>: predictions.csv, predictor_variables.csv.

<inval_mut>: predictions.csv, invalid_mutations.csv.

<only_pred>: predictions.csv.

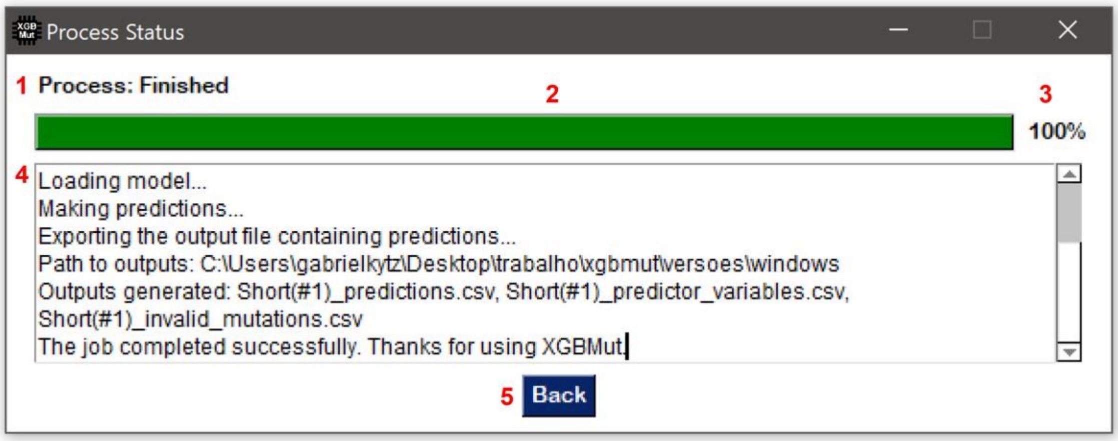
Usage option (GUI):

Initial Menu:



1- Menu: contains File, Example, and Help options;
2- Input file: write or select Browse button (6) to choose the path to input file;
3- Output name: write the output file name (optional);
4- Output options: customize the output files generated by checking the boxes.
5- Submit button: submit the analysis and open the Progress window.
6- Browse button: open a popup window to choose the path to the input file.

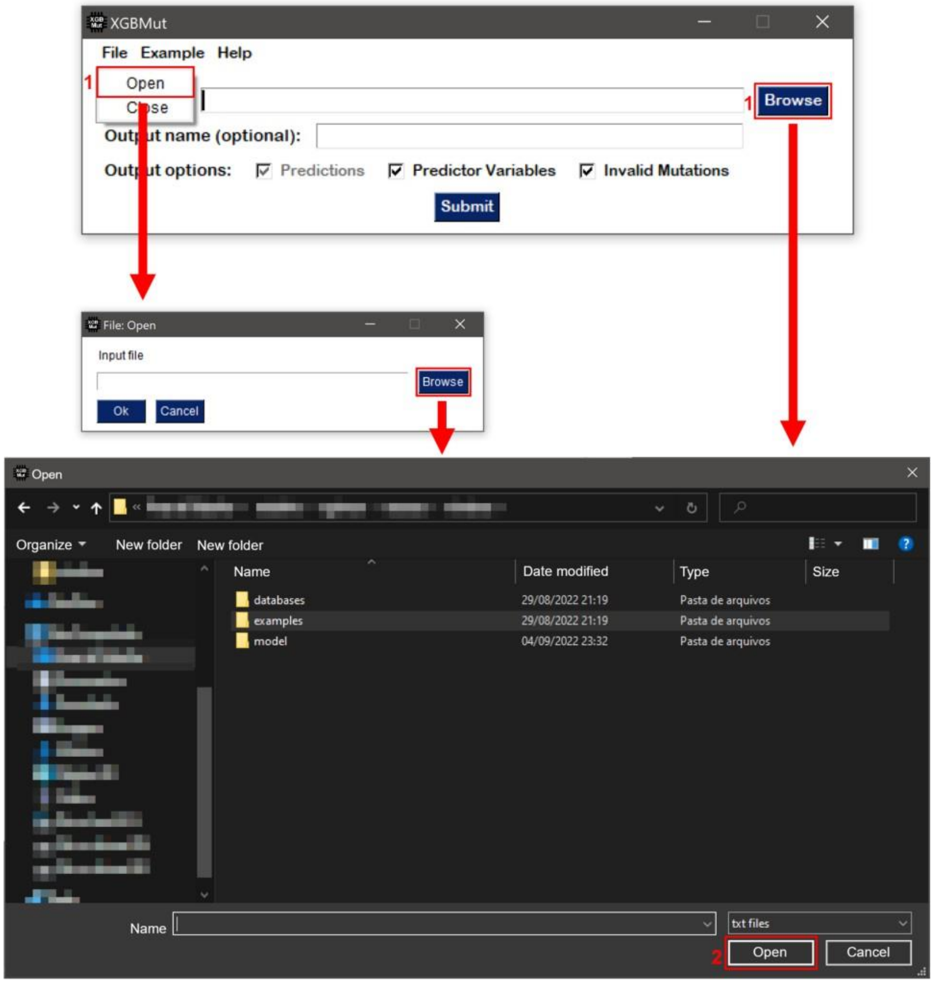
Progress menu:



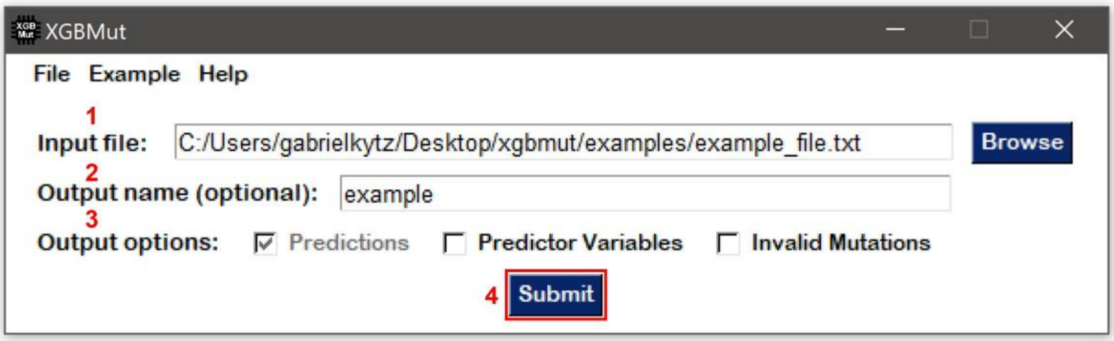
```
Loading model...
Making predictions...
Exporting the output file containing predictions...
Path to outputs: C:\Users\gabrielkytz\Desktop\trabalho\xgbmut\versoes\windows
Outputs generated: Short(#1)_predictions.csv, Short(#1)_predictor_variables.csv,
Short(#1)_invalid_mutations.csv
The job completed successfully. Thanks for using XGBMut.
```

1- Process: progress status information;
2- Loading bar containing a visual indicator of the analysis progress;
3- Percentage of the analysis progress;
4- Output console containing information about the analysis.
5- Back menu: returns to the Initial window.

Loading File:


XGBMut
File Example Help
1
Open
Close
1
Browse
Output name (optional):
Output options: ☑ Predictions ☑ Predictor Variables ☑ Invalid Mutations
Submit
File: Open
Input file
Browse
Ok
Cancel
Open
Organize
New folder
New folder
Name
Date modified
Type
Size
databases 29/08/2022 21:19 Pasta de arquivos
examples 29/08/2022 21:19 Pasta de arquivos
model 04/09/2022 23:32 Pasta de arquivos
Name
txt files
2
Open
Cancel

Running Analysis:



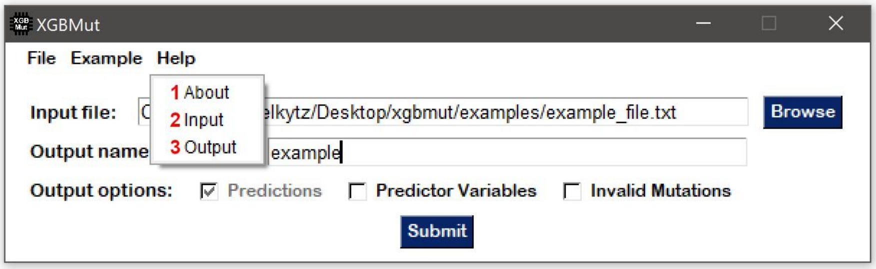
Input file: C:/Users/gabrielkytz/Desktop/xgbmut/examples/example_file.txt

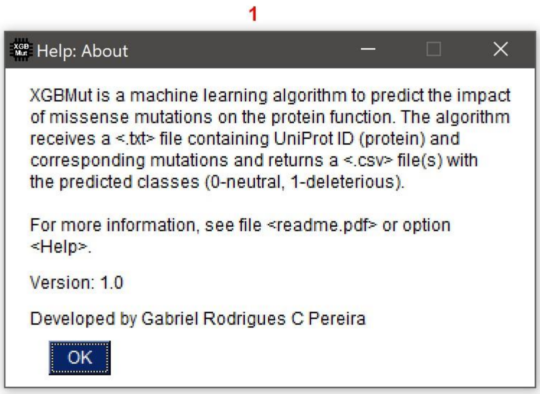
Output name (optional): example

Output options: ☑ Predictions ☐ Predictor Variables ☐ Invalid Mutations

1- Write the path to the input file (mandatory);
2- Choose an output name (optional);
3- Customize the output files generated (optional)
4- Submit the analysis (mandatory);

Help options:




Help: About

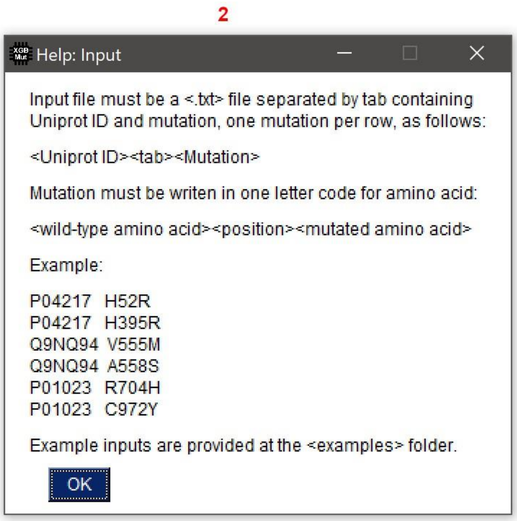
XGBMut is a machine learning algorithm to predict the impact of missense mutations on the protein function. The algorithm receives a <.txt> file containing UniProt ID (protein) and corresponding mutations and returns a <.csv> file(s) with the predicted classes (0-neutral, 1-deleterious).

For more information, see file <readme.pdf> or option <Help>.

Version: 1.0

Developed by Gabriel Rodrigues C Pereira

OK


Help: Input

Input file must be a <.txt> file separated by tab containing Uniprot ID and mutation, one mutation per row, as follows:

<Uniprot ID><tab><Mutation>

Mutation must be writen in one letter code for amino acid:

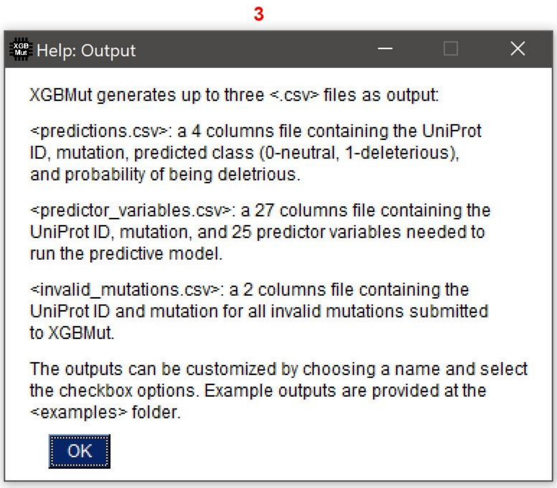
<wild-type amino acid><position><mutated amino acid>

Example:

```
P04217  H52R
P04217  H395R
Q9NQ94  V555M
Q9NQ94  A558S
P01023  R704H
P01023  C972Y
```

Example inputs are provided at the <examples> folder.

OK


Help: Output

XGBMut generates up to three <.csv> files as output:

<predictions.csv>: a 4 columns file containing the UniProt ID, mutation, predicted class (0-neutral, 1-deleterious), and probability of being deletrious.

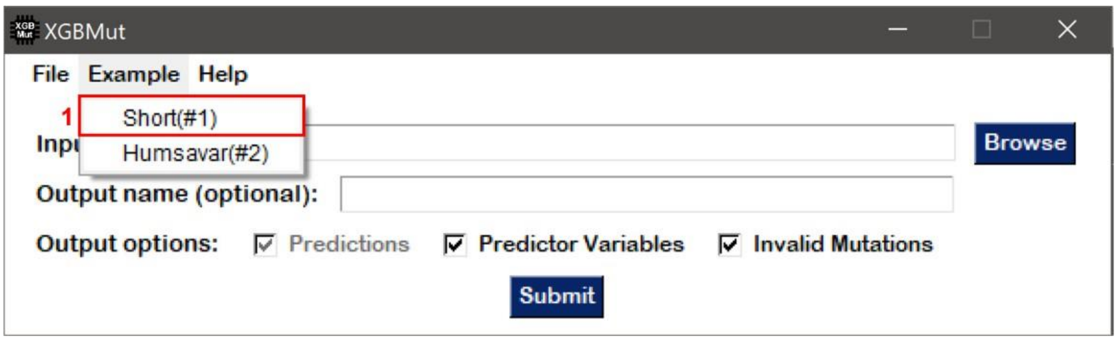
<predictor_variables.csv>: a 27 columns file containing the UniProt ID, mutation, and 25 predictor variables needed to run the predictive model.

<invalid_mutations.csv>: a 2 columns file containing the UniProt ID and mutation for all invalid mutations submitted to XGBMut.

The outputs can be customized by choosing a name and select the checkbox options. Example outputs are provided at the <examples> folder.

OK

Running examples:



1- Choose an example option: Short(#1) or Humsavar(#2).