# Fundamentals of Bioinformatics Project Manual 2022: Mutation Impact Prediction Methods

Marina Diachenko, Roel van der Ploeg, Lucía Barbadilla Martínez, Lara Pozza, Will Harley, Fabienne Kick, Ren Xie, Alex van Kaam, Ignas Krikštaponis, Arthur Goetzee, Anton Feenstra, Daniël Muysken and Sanne Abeln

September 2022

# Table of Contents

# Introduction

## Aims of the Group Project

This project is an introduction to the basic theory and practice of solving common problems in bioinformatics. Bioinformatics is an interdisciplinary field combining both biology and computer science, and depending on your academic background, some parts of this project may be unfamiliar and challenging to you. However, we aim to make project groups that will include students from different BSc backgrounds. You should allocate tasks accordingly within your group but also work collaboratively as much as possible. The objective of this project is to learn to communicate scientific problems with people who speak a different scientific language, as this will be an essential skill working in the field of bioinformatics. In addition, the project allows you to see how far your current knowledge reaches and find which skills you will have to improve in the coming year. Courses scheduled later in the curriculum will delve deeper into the details of the tools and data.

## Predicting the Impact of Mutations

Nonsynonymous (missense) mutations occur where a single nucleotide in a DNA codon is substituted for another (a single nucleotide polymorphism or SNP), resulting in a change to the amino acid that the codon codes for. These missense mutations can have no or little effect on protein function and phenotype, but can sometimes result in significant changes that can cause disease. The impact prediction tools PolyPhen-2 (Adzhubeiet *et al.*, 2010) and SIFT (Vaseret *et al.,* 2016) are designed to predict which mutations in DNA will cause changes in the cell. They can be used to help interpret mutation data from patients who have genetic diseases, but these methods must be validated to assess how accurate their predictions are. Validation of these tools requires experimental data with accurate annotations (i.e., a benchmark or gold standard dataset) against which we can compare the performance of our tools. In this case, this means SNP data with annotations to indicate whether they are benign or pathogenic to compare against the impact predictions of SIFT and PolyPhen. We will use the database ClinVar for our gold standard dataset. Once we have benchmarked the predictions, we can visualise the performance of these tools by creating a ROC (Receiver Operating Characteristic) plot, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR). More will be explained about ClinVar and ROC plots in depth in the step-by-step instructions below.

# Timeline

| **Week** | **Date** | Activity/**Deadline** | **Exercises** (Green boxes) |
|---|---|---|---|
| 1 | Monday | Intake Test<br>Linux and Command Line Introduction | |
| | Tuesday | Impact Prediction Tutorial and Report Writing Introduction | Exercises 1-4 |
| 2 | Monday | Script Baseline + report writing intro + define research question | Exercises 1-4 |
| | Tuesday | Script Baseline + report writing methods | Exercise 5 |
| 3 | Monday | Script ROC plot + report writing results | Exercises 6-9 |
| | Tuesday | Script ROC plot + report writing results + abstract | Exercises 6-9 |
| 4 | Monday | Work on draft report | Exercise 10-11 |
| | Tuesday | Work on draft report | Exercise 10-11 |
| 5 | Monday | Work on draft report | |
| | Tuesday | **Draft Report Deadline** | |
| 6 | Monday | **Peer Review Deadline** | |
| | Tuesday | **FoB Exam** | |
| 7 | Monday | Discussion Sessions | |
| | Tuesday | Project report Q&A session | |
| 8 | Monday | **Deadline Final Report** | |
| | Tuesday | | |

# Grading

This project counts for 40% of the final course grade. The deliverables that are listed below all have to be handed in and are either graded or pass/fail. You will be assigned to give feedback on the draft report of another group and your group's draft report will get feedback from members from other groups. During the final weeks of the course, you will compare your results to that of other groups. This will help you to write the discussion for your report. At the end of this manual, each of these deliverables is explained in greater detail. Note that the group project will also help you to prepare for the exam.

- Progress as checked by the TAs (pass/fail)
- Draft report (pass/fail) + ROC data files (pass/fail)
- Peer Feedback of draft report (pass/fail)
- Final report + scripts (100% of final project grade)
    - Based on rubric

**>_ Exercise X | Blue Boxes**

Through this manual you will find green boxes like this one. Green boxes contain exercises that will help you understand the project. You should discuss them with the Teacher's Assistants (TAs), but you are **NOT** expected to answer them in the report directly.

# Practical Instructions and Questions

## Setup

### Logging Into the VU Servers

To make sure you can run in a linux environment, you first need to log in to the VU servers, so that you can run *python3* in a suitable environment. For instructions on how to create a suitable environment for your OS, check tutorial 0 under the Technical setup header under modules on Canvas, next to the section Programming Class. Additionally, you can install a local editor, such as PyCharm. Note that it may be wise for all group members to use the same editor, to avoid issues with tab and space settings in python.

Also if you decide to share your code through a GitHub repository, remember to make the repository private, **a public repository would be seen as enabling plagiarism**.

### Setting up a local database

Before we can start we have to create directories to store our data and outputs. You can again create these directories from the command line. We have provided you with five data

files and three skeleton scripts. To organize your working directory for the project, let's create two subdirectories in your working directory named *data* and *output*. This can be done from the command line with *mkdir* (also see programming tutorial 1 on Canvas).
From FoB Project in Canvas Modules, download the *BLOSUM62.txt* file and the *<HGVSdataset>_benchmark.tsv*, *<HGVSdataset>_sift_scores.tsv*, *<HGVSdataset>_polyphen_scores.tsv* and *<HGVSdataset>_VEP_baseline.tsv* files. Place the *BLOSUM62.txt* file and the *<HGVSdataset>_benchmark.tsv* in the *data* folder. Download the three skeleton scripts from Canvas (.py files) and place them in your working directory. Note that if you are going to be working on the compute server, you first need to copy your files there. Please see the tutorial *working at home or on your own laptop* to see how you can move/copy files to the compute servers.

Create a subdirectory in *data* called *vep* which stands for Ensembl Variant Effect Predictor.

Put the three *.tsv* data files (*<HGVSdataset>_sift_scores.tsv*, *<HGVSdataset>_polyphen_scores.tsv* and *<HGVSdataset>_VEP_baseline.tsv)* in *data/vep* (these are VEP output files which will be explained further).

## Benchmarking Impact Prediction Methods

In this project you will test the performance of the mutation impact prediction tools PolyPhen and SIFT using a benchmark (gold-standard data) from ClinVar, an NCBI database of human genomic variation and its relationship to human health. You will also create a baseline model and compare this to the ClinVar benchmark. You will do this using the BLOSUM62 matrix, which is an amino acid substitution scoring matrix used for protein sequence alignment tools such as BLASTP. This is to build a basic prediction model against which to compare the performance of PolyPhen and SIFT as measured using the benchmark. To visualise the results, you will create ROC plots for how SIFT, PolyPhen and your baseline model compare with the data from ClinVar. You will create three individual ROC plots and one plot of all three models together. You will then combine your results with those from other students to compare your data with those of your peers.

An overview of the workflow of this project is shown in *Figure 1*.

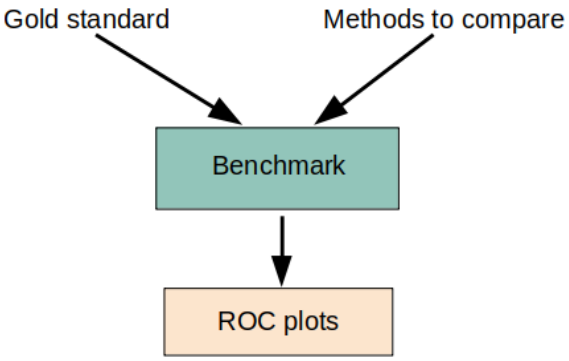


*Figure 1*. **Project workflow.** The workflow shows a simple overview of the project. A benchmark will be performed between the gold standard, VEP, and the methods to compare, baseline, SIFT and PolyPhen. The scores from the benchmark can then be used to produce a ROC plot for each predictor.

You will be given four initial *.tsv* files (tsv stands for tab separated values), one text file and three skeleton scripts.

- Two of these *.tsv* files contain the results of PolyPhen-2 and SIFT, a third *.tsv* contains the information of the benchmark, and the fourth *.tsv* file contains the information that you will need to create your baseline model.
- The *.txt* file contains the BLOSUM62 matrix that you will need for the baseline model.
- The skeleton scripts have missing blocks of code that you will have to complete, you can find them between the “START CODING HERE” and “END CODING HERE”.
    - The first script you will need to complete and run is the *skeleton_script_baseline_model.py*, to create a baseline model.
    - The *skeleton_script_create_roc_plot.py* uses the output of the baseline model along with the three other *.tsv* files as its input to create ROC plots that compare the baseline model, SIFT and PolyPhen with the data from ClinVar.
    - Finally, you will run the *skeleton_script_roc_plot_tsv.py* script which needs the output.tsv files generated with the previous script from your data and from your fellow students data, which will be provided to you three weeks into the project.

Details of these steps can be found in the following sections.

Before starting, remember that you cannot import any packages apart from the ones already found in the skeleton scripts. That would make the code harder to read for those who have just started programming, and will most likely be graded as a fail. Note that even though some students in your group can focus on the programming, everyone in the group should be able to run the scripts and understand what they aim to do (also to prepare for the exam). We recommend you start reading the script in the *main()* function, and then try to read and understand each function as they are being called in the main function. To make the code understandable for everyone, make sure to comment what you are doing with the lines of code you add.

## The Benchmark Datasets

ClinVar is a database of how human genomic variants are related to phenotypes of human disease and supporting clinical evidence for these relationships, managed by the NCBI. This database has been used to obtain the HGVS IDs of the genomic variants and their clinical significance (label) you will work with (**<HGVSdataset>_benchmark.tsv**). Each group will work with one of the three datasets:

1. old version of the dataset (*HGVS_2014_<...>.tsv*)
2. short dataset of the up-to-date database (*HGVS_2020_small_<...>.tsv*)
3. long dataset of the up-to-date database (*HGVS_2020_big_<...>.tsv*).

Each of these have been mapped to the reference genome GRCh38. The up-to-date database (*HGVS_2020_small_benchmark.tsv* and *HGVS_2020_big_benchmark.tsv*) was obtained from clinvar_20200629.vcf.gz while the old database (*HGVS_2014_benchmark.tsv*) was obtained from clinvar_20141202.vcf.gz. A selection process was done for all of them: only benign or pathogenic SNPs (Single Nucleotide Polymorphisms) were selected (likely benign or likely pathogenic SNPs were excluded), to overcome ambiguities. Subsequently, the following SNPs were filtered out:

- Intron variants
- Synonymous variants (those that lead to synonymous mutations)
- Variants in mitochondrial DNA
- Variants that vary into multiple bases
- Unknown variants.

The three datasets were balanced obtaining the same number of 'Benign' and 'Pathogenic' samples.

**>_ Exercise 1 | Inspection of the data**

- How many HGVS Ids does each dataset have?
- Can you check that the benchmark dataset is indeed balanced? Why is this important?

**>_ Exercise 2 | Search for your own SNP**

In this exercise, we are interested in modifying the gene coding for Apoliprotein E (APOE) to reverse a missense mutation from a patient with APOE deficiency. We do not know how many missense mutations can be pathogenic. Go to https://www.ncbi.nlm.nih.gov/clinvar and search for APOE. From how many missense variants can this deficiency originate?

## The HGVS Format

The Human Genome Variation Society (HGVS) nomenclature is used worldwide as a standard language for the description of changes (variants/polymorphisms/mutations) in RNA and DNA sequences. It is formatted as **reference : description** – the reference sequence (e.g., NM_004006.2) is how the variant is referenced in databases, in this case RefSeq, and it is followed by a description of the variant (e.g., c.4375C>T). These descriptions are usually given in the context of a specific gene. The first (lowercase) letter stands for the context of the code: c for coding DNA, g for genomic DNA, r for RNA and p for protein. The number is the position of the polymorphism in the reference sequence (e.g., 4375) and the last two letters separated by a > symbol represent the two different nucleotides that are found in this position.

**>_ Exercise 3 | HGVS format**

- What is the meaning of the highlighted symbols in the HGVS ID *NM_004006.2:c.4375**C>T***?

- Your HGVS IDs will be genomic reference sequences based on a chromosome, which values should the highlighted Xs take?

***X**_000003.12:**X**.12599717C>G*

## Ensembl Variant Effect Predictor (VEP)

The Ensembl Variant Effect Predictor (VEP) is a tool for the analysis and annotation of genomic variants in coding and non-coding DNA. VEP can take different genomic variant formats as input, but here we will use the HGSV format. It works using an extensive collection of genomic annotation and can be adapted to different interfaces depending on the context of the project: it can be used through a web interface, a command line tool and REST API. The web interface can be used for smaller amounts of data while the command line tool is able to handle larger amounts of data and has more flexibility and a greater range of options.

In this project, you will not have to use VEP directly, as we have selected some of the results we obtained previously by running REST API.

The VEP output that we are interested in this project is:

- **ID**: Corresponds to the HGSV provided in the input.
- **Amino acids change**: Reference and variant amino acids.
- **Codon change**: Reference and variant codon sequence, the alternative codons with the variant base in upper case.
- **PolyPhen Score***:* Impact prediction of an amino acid substitution produced by PolyPhen 2.2.2. The score ranges from 0.0, being tolerated, to 1.0, being deleterious.
- **SIFT score:** Impact prediction of an amino acid substitution produced by SIFT 5.2.2. The score ranges from 0.0, being deleterious, to 1.0, being tolerated.

Each HGVS ID is a variant of a genomic sequence that can overlap multiple transcripts. Hence, when using VEP each HGVS ID has as many outputs as transcripts there are, and each one can take different PolyPhen and SIFT scores. To make things easier for you, we have selected only one transcript result for each HGVS ID. The transcript with the highest impact (most deleterious) predicted by SIFT and PolyPhen was the one selected. If several transcripts have this score, the transcript is selected at random among these ones. If PolyPhen and SIFT do not agree, that HGVS ID will be skipped to avoid bias towards either PolyPhen or SIFT.

**>_ Exercise 4 | Selecting transcripts for the VEP output**

This output has been extracted from the VEP web interface when providing three HGVS IDs. If you have understood the paragraph above, you should know which transcripts have been selected in your output.

| Uploaded variant | Feature | Amino acids | Codons | SIFT | PolyPhen |
|---|---|---|---|---|---|
| NC_000001.11:g.224424520A>C | ENST00000414423.8 | S/R | AG**T**/AG**G** | 0 | 0.992 |
| NC_000001.11:g.224424520A>C | ENST00000445239.1 | S/R | AG**T**/AG**G** | 0 | 0.972 |
| NC_000001.11:g.224424520A>C | ENST00000651911.1 | S/R | AG**T**/AG**G** | 0 | 0.972 |
| NC_000018.10:g.51078285G>C | ENST00000342988.8 | D/H | **G**AT/**C**AT | 0 | 0.899 |
| NC_000018.10:g.51078285G>C | ENST00000398417.6 | D/H | **G**AT/**C**AT | 0 | 0.899 |
| NC_000018.10:g.51078285G>C | ENST00000588745.5 | D/H | **G**AT/**C**AT | 0 | 0.629 |
| NC_000001.11:g.236885200A>G | ENST00000366576.3 | D/G | G**A**C/G**G**C | 0.2 | 0.21 |
| NC_000001.11:g.236885200A>G | ENST00000535889.6 | D/G | G**A**C/G**G**C | 0.23 | 0.21 |
| NC_000001.11:g.236885200A>G | ENST00000674797.1 | D/G | G**A**C/G**G**C | 0.18 | 0 |
| NC_000001.11:g.236885200A>G | ENST00000366577.10 | D/G | G**A**C/G**G**C | 0.15 | 0.161 |

The output is given to you in three separate *.tsv* files ("<...>" refers to your assigned dataset, as described above in The Benchmark Datasets):

- ***<HGVSdataset>_sift_scores.tsv***: contains the HGVS IDs* and the SIFT score.
- ***<HGVSdataset>_polyphen_scores.tsv***: contains the HGVS IDs* and PolyPhen score.
- ***<HGVSdataset>_VEP_baseline.tsv***: contains the HGVS IDs*, Amino acid change and Codon change.

**Note that the HGVS IDs are the same for the 3 files.*

## Create Baseline Prediction

The BLOSUM62 matrix is based on frequencies of amino acid substitutions of a collection of protein alignments with 62% identity. As you know, this matrix is being used in alignment tools such as BLAST or BLASTP. In this project, you will use the information in BLOSUM62 to obtain an insight into a substitution's expected impact, and with this create a baseline impact prediction method. You can find the BLOSUM62 matrix in the *BLOSUM62.txt* file.

**>_ Exercise 5 | BLOSUM62 matrix**

Check out the BLOSUM62 matrix in the *BLOSUM62.txt* file. Do you think the diagonal values are going to be used on the baseline model with the data set that we have provided? Why, or why not?

**>_ Exercise 6 | BLOSUM62 matrix**

Check the BLOSUM62 matrix in *BLOSUM62.txt* again. Look at the substitution scores of cystine (C) and glutamine (Q). Can you think of reasons why glutamine seems to be more replaceable than cystine? Can you generalise your answer to the different groups of amino acids?

Complete and execute the baseline predictor skeleton script (*skeleton_script_baseline_model.py*) using *BLOSUM62.txt* and *<HGVSdataset>_VEP_baseline.tsv* as inputs. The script should create a score which is simply the raw value of the BLOSUM62 for that amino acid exchange. As output, you should obtain the scores of your baseline in the same format as in *<HGVSdataset>_sift_scores.tsv* and *<HGVSdataset>_polyphen_scores.tsv*. The output can be saved to a *data* or *output* folder, or any other folder that you might have additionally created in your working directory beforehand, by providing a file path to the ***-o*** argument on the command line. This argument is required, and the file name should be supplied with the *.tsv* extension; for example:

```
$ python3 skeleton_script_baseline_model.py data/vep/HGVS_2020_small_VEP_baseline.tsv data/BLOSUM62.txt -o data/HGVS_2020_small_baseline_scores.tsv
```

## ROC Plot

Your task here is to create a Receiver Operating Characteristic (ROC) plot by comparing the results from your predictors to the gold standard data we have obtained from ClinVar. A ROC plot is a method of visualising the performance of your predictor, it plots the True Positive Rate (TPR) against the False Positive Rate (FPR). Refer to the lecture on machine learning and benchmarking for a thorough explanation of what ROC plots are and how they

can be used to evaluate, compare, and refine classification methods. In addition http://wikipedia.org/wiki/Receiver_operating_characteristic may be a helpful resource.

Note that a threshold to classify variants as (putatively) benign or damaging is not fixed at a constant value, in order to create a ROC plot of the results. Instead, in a ROC plot you calculate the true and false positive rate for every possible threshold spanning the range of possible values for your method, from 0 until 1 for SIFT or from -2 until 9. For every threshold, this allows every variant classified by the predictor to be categorised as a True Positive (TP), False Positive (FP), False Negative (FN) or a True Negative (TN).

**>_ Exercise 7 | Confusion Matrix**

Complete the blank cells in this confusion matrix. Hint: The conclusion drawn from the predictor depends on the threshold.

| | **BENCHMARK** | | *ClinVar Benign* |
|---|---|---|---|
| **PREDICTOR** | **Conclusions** | | *Confirmed Benign* |
| | *(Putative Damaging)* | | |
| | *(Putative Benign)* | | True Negative (TN) |

## Calculating the AUC for a ROC Curve

A ROC plot can be made by varying the threshold, counting the TPs, FPs, TNs and FNs, and calculating the TPR and FPR. First, it is important to define what positive and negative assignations are. Think of a covid test, a positive result means you probably have the virus. Here the same consensus applies, thus a mutation with a predicted deleterious effect will be a positive result.

When working with ROC plots, the Area Under the Curve (AUC) is often taken as a measure to evaluate performance. To calculate the AUC you have to approximate the integral of the function *f(x)* that describes the shape of the curve of the ROC-plot. We do not know the function that describes the curve, thus we have to evaluate the integral numerically. A method that approximates the integral is the trapezoidal rule.
Think of a clever way to implement this rule, and complete the provided skeleton script: *skeleton_script_create_roc_plot.py*. This script will parse your predictor and benchmark results, count the number of TP, FP, FN and TN, calculate your ROC plot's line coordinates, create the corresponding figure, and integrate the AUC.

Complete and execute the skeleton script *skeleton_script_create_roc_plot.py*. For a better understanding of the ROC plot, the script produces a color gradient indicating the score range.

To obtain the individual ROC plot for one predictor, the optional argument *-ipred* should be included once with the *.tsv* file with the scores of one of your methods (SIFT, PolyPhen or baseline). The *-ibench* should be included with the *<HGVSdataset>_benchmark.tsv* file. You can use the help function *-h* or *--help* for explanation of these and other options. You will have to specify a path for the output *.png* file with the argument *-o* (including the *.png* extension). A *.tsv* file with the ROC *x*- and *y*-coordinates will be saved automatically to the same output directory. For example, to call the script for the PolyPhen ROC plot:

```
$ python3 skeleton_script_create_roc_plot.py -ibench
data/HGVS_2020_small_benchmark.tsv -ipred
data/vep/HGVS_2020_small_polyphen_scores.tsv -color -o
output/ROCplot_HGVS_2020_small_polyphen.png
```

To show the ROC curves of the three predictors in one figure, the script can be run with the *-ipred* argument three times for each of the three prediction *.tsv* files (SIFT, PolyPhen and baseline). (In this case, the ROC plot coordinates file will not be created). A command line example is provided below:

```
$ python3 skeleton_script_create_roc_plot.py -ipred
data/vep/HGVS_2020_small_polyphen_scores.tsv -ipred
data/vep/HGVS_2020_small_sift_scores.tsv -ipred
data/HGVS_2020_small_baseline_scores.tsv -ibench
data/HGVS_2020_small_benchmark.tsv -o output/ROCplot_all.png
```

**>_ Exercise 8 | Curve details**

- In your ROC plots you will probably find that some of the methods provide a much more detailed curve than others. Can you explain why this is?
- What will a ROC-plot look like if you have extremely unbalanced data? Will the ROC-plot be representative?

## Comparing Your ROC Curve with Other Data Sets

The performance of the predictors depends not only on the predictor itself but can also be influenced by the data you work with. When you hand in your draft report we will also ask you for the data you have used to generate the ROC plots. This data will be shared with the other groups for comparison. In order to read in this data, and make a new plot, there is a

third script. To test how different data sets can influence the ROC plot results, you will run the third script *skeleton_script_roc_plot_tsv.py*.

This script contains only one coding block which is the same you have found in the last script to calculate the AUC – please use the same code. As inputs, you will use the coordinates that your fellow students have obtained with the other two datasets by providing paths to -itsv. These two sets of coordinates will be given to you through Canvas. You will have to run the code twice, once for each type of dataset. As output, you will get a .png file with the ROC plot comparing the same dataset on the three predictors, just as the previous script (skeleton_script_create_roc_plot.py). This command line illustrates how can it be run:

```
$ python3 skeleton_script_roc_plot_tsv.py -itsv output/ROCplot_HGVS_2020_small_sift_xy.tsv -itsv output/ROCplot_HGVS_2020_small_polyphen_xy.tsv -itsv output/ROCplot_HGVS_2020_small_baseline_xy.tsv -o output/ROCplot_comparison.png
```

The ROC plots from this script will be discussed in the discussion session and you will have to add them to your final report as well.

**>_ Exercise 9 | Default thresholds**

SIFT and PolyPhen define default thresholds for their score to classify a mutation as benign or pathogenic. Do you think the default thresholds make sense according to your ROC plot (look at the FPR and TPR)? What would happen if you changed the threshold?

# Instructions for Submitting Draft Report (Submit in PDF format via Canvas)

The draft report must contain between 1000 and 1500 words and contain following sections:

- Abstract
- Introduction
- Methods; and
- (Preliminary) results.

The results section should include a ROC plot and its interpretation. You must clearly state your research question in the introduction and answer it in the results and discussion sections. Note that in the section below, and in the rubric (you will receive for the peer review), more details are provided about what the report should contain.

Please add word counts in square brackets [ ] behind the title of each section, before you submit. Your draft report needs to be handed in via Canvas, and will be peer reviewed by students of other groups. Note that you do not yet need to write the discussion and conclusion sections for your draft.

## Peer Review of Draft Reports

Everyone should peer review the report of one other group, meaning that each group should get around 4 peer reviews back. The peer review should be handed in on Canvas. Note that the peer review should be based on the rubrics provided. You need to write a peer review in order to pass the course.

# Use Cases: Investigating two SNPs in detail

Now we would like you to think more about the biological aspect of impact prediction. You will do this by comparing two SNPs from the same gene, where one is known to be a benign SNP and one is known to be a pathogenic SNP. You will report on your findings in a section called "**Use cases**". The section needs to contain the answers to the exercises below and you may add additional information to support your findings. Also see the rubrics.

Below a list with three genes is shown, with corresponding SNPs and the variant. Depending on your dataset used you will do the following steps for two SNPs from a single gene.

| **Gene** | **HGVS** | **Feature** (or transcript_id in rest API without the . and last number) | **Group** |
|---|---|---|---|
| TP53 | NC_000017.11:g.7674220C>A | ENST00000413465.6 | Old dataset |
| | NC_000017.11:g.7673751C>T | ENST00000269305.9 | Old dataset |
| BRCA2 | NC_000013.11:g.32362595G>C | ENST00000380152.8 | Small dataset |
| | NC_000013.11:g.32396905A>G | ENST00000380152.8 | Small dataset |
| BRCA1 | NC_000017.11:g.43067628G>A | ENST00000478531.5 | Big dataset |
| | NC_000017.11:g.43063368T>C | ENST00000586385.5 | Big dataset |

As a first step, go to the Ensembl Variant Effect Predictor (VEP) website. Click on the Web interface option. In your input data, paste each of your SNPs in HGVS format on a separate line. You can leave all other settings on default, and click "Run" at the bottom. Please be aware that a job can take a few minutes to complete. Click on "view results" when your job is complete. You might need to change the shown columns by clicking on the "Show/hide columns" button in the blue bar.

Alternatively you can use the REST API you can type the following url in your browser "https://rest.ensembl.org//vep/human/hgvs/" followed by the HGVS code. See also: https://rest.ensembl.org/documentation/info/vep_hgvs_ge

We will start with comparing the sequence conservation between the two SNPs. The web server of PolyPhen-2 is the tool we will use for this. Before going to the website, find the rsID in the VEP output under “Existing variant” for each SNP (be sure to look for the right feature as indicated in the table above). Use the rsID one by one as input for the query WHESS.db. If you get multiple options in the results screen, choose the results with the same protein position as in the VEP output. On the report page you can find the sequence conservation under the Multiple sequence analysis tab.

**>_ Exercise 10 | Sequence conservation**

- Do you see differences in sequence conservation for the region around the SNP and for the SNP itself?
- Do you see differences in sequence conservation between the SNPs? Is this as expected when considering evolution laws?

On the same report page produced by PolyPhen-2, you can also find a tab called 3D visualization. Use this to find where in the protein structure the mutation occurs.

**>_ Exercise 11 | Protein structure**

- What is the structural environment around the SNP, when focusing on where in the protein the secondary structure occurs?
- Does the place of the SNP in the protein make sense considering it’s impact as predicted by VEP?

# Discussion Sessions

In the discussion sessions, you will be matched with students from other groups to discuss your findings about the Use cases. The discussion points you need to prepare are the questions from exercises 9 and 10. Additionally we want you to tell something about the biological background of the SNPs, this is however not required for the report. The discussion session will be moderated by teachers and TAs.

Make sure you can show figures of the sequence conservation and the structure. This can be done in a small presentation.

## Sharing your data

When you execute the create_roc_plot.py script, the coordinates of the ROC plot will automatically be exported to '[your_custom_plotname]_xy.tsv'. You need to share the .tsv files generated by the skeleton_script_create_roc_plot.py with other students for your specific benchmark datasets, for all three methods.

Please follow the instructions posted on Canvas to share your data.

# Discussion Questions

Your discussion section in the report should contain the answers to the following questions. The questions under discussion within your group should be based on your own results, and the questions under discussion with other groups should be based on your own results and the results from other groups.

## Discussion Within Your Group:

1. Do you observe a difference in performance between SIFT, PolyPhen and the baseline script? How can you explain the difference in performance? [A1]
2. SIFT and PolyPhen define default threshold(s) for their score to classify a mutation as benign or pathogenic. Do you think the default thresholds make sense? [A2]

## Discussion with Other Groups:

1. Are there any clear differences between the different benchmark datasets in terms of the AUC and the shape of the ROC curves? What is the effect of having more benchmark data available? [B1]
2. Is the relative performance the same in all three datasets for SIFT, PolyPhen and the baseline script, how could you test this? [B2]

# Format of the Final Report

The final report must contain the following sections:

- Abstract (max 250 words):
  - Motivation, results & impact
- Introduction:
  - Include references to previous studies from literature related to your research question
  - Make sure to explain why impact prediction is typically performed
  - Make sure to explain why bioinformatics methods need to be benchmarked
  - State your research question explicitly
- Methods and Data:
  - Describe the methods you use
  - Describe the datasets you use
  - Include a flow chart or scheme
- Results:
  - Include 4 ROC plots and interpret them
- Use Cases:
  - Observations of sequence conservation and protein structure
  - Discussion about the impact from VEP in comparison with your observations
- Discussion
  - Discuss the points listed as questions in the discussion session
  - You can add results from other groups, with a reference, and/or cite other studies
- Conclusions
  - Answer your research question
  - Explain the impact of the work on its application areas
- Tables & Figures
  - Explain all axes, labels, lines and points in the caption of your figure/table.
  - Refer to each figure/table in the main text, and explain in the main text what can be seen from the figure/table.
- References
  - We expect between 3 and 15 citations to other papers (author-year citations are preferred). Some essential literature for the project is provided on Canvas and in the lecture slides. Note that 15 citations is not a limit.

The final report must contain between 3000 and 3500 words. We count everything (including figure text) except references.

Please read the rubric for the final report and make sure you include every requirement listed.

## Handing in the Final Report

The final report must clearly state what each group member contributed to the project. Individual students' grades may be adjusted according to the reported and observed differences in workload.

## Handing in the Final Code

All scripts you produce or complete during the practical should be handed in on CodeGrade. Your code must be readable (which means it should be well structured), use self-explanatory variable and function names, and be sufficiently commented. File names, paths and query entries may not be hard coded.

Note that you need to add comments to the code you write, so that all group members can understand what is going on in the code blocks.

The code will count 10% towards the final project grade, the final report 90%.

### CodeGrade

We use CodeGrade, an automatic grading system, to evaluate your code. This means that coding outside the code block for editing (“START CODING HERE” and “END CODING HERE”) is generally not allowed (otherwise CodeGrade may not work). Importing additional packages is also not allowed.

You can submit your code to CodeGrade multiple times to check if your code works correctly.

# References

Adzhubei, I. A., Schmidt, S., Peshkin, L., Ramensky, V. E., Gerasimova, A., Bork,P., Kondrashov, A. S. and Sunyaev, S. R. (2010) A method and server for predicting damaging missense mutations. *Nature methods*, **7** , 248–249

Vaser, R., Adusumalli, S., Leng, S. N., Sikic, M. and Ng, P. C. (2016) Sift missense predictions for genomes. *Nature protocols*,**11**, 1.

# Report grading rubrics

| **Criteria** | **Pts** |
|---|---|
| Give an accurate and concise summary of the introduction and clearly state the research question. | 2 pts |
| Give an accurate and concise summary of the materials and methods section. | 2 pts |
| Give the most important results and answer the research question. | 4 pts |
| Mention the potential impact of these results on future research and/or practical applications. | 2 pts |
| Introduce the importance of impact prediction | 2 pts |
| In the context of impact prediction explain the relevant biology | 2 pts |
| Introduce existing methodology (impact prediction methods, and benchmarking) | 2 pts |
| Clearly state your research question, which should be accurate to the details of the project and be falsifiable by your results. | 2 pts |
| Cite the most relevant previous research. | 2 pts |
| Give an overview of the workflow in a scheme. | 2 pts |
| Describe the properties of the data used. | 2 pts |
| Describe the scores for the different methods. | 3 pts |

| | |
|---|---|
| Describe your benchmarking strategy. | 3 pts |
| Explain what you are trying to test and how you are testing this | 2 pts |
| Provide the ROC plots for each of the benchmarked methods together with the AUC | 6 pts |
| Compare the three different methods | 2 pts |
| Describe the plots and what they represent in the main text | 5 pts |
| Explain whether the results conform to your initial expectations. | 5 pts |
| The figures are readable. | 2 pts |
| The figures are correct. | 2 pts |
| The captions provide all the information to understand the data shown in the figures. | 6 pts |
| Explain the difference in performance between the three different methods | 4 pts |
| Discuss the default values for SIFT and Polyphen. | 3 pts |
| Describe any clear differences between the different benchmark datasets, and provide an explanation and give the consequences of these findings. Describe what the effect is of having more benchmark data available. | 5 pts |
| Describe if the relative performance is the same in all three datasets for SIFT, Polyphen and the baseline script, and how you could test this. | 3 pts |
| Give your main conclusions and answer your research question. Consider what can and can not be concluded from your results. | 5 pts |
| Discuss the potential impact of your results in a practical/medical context. | 5 pts |