# Project title: Predicting conflicts in genetic variant classification

Group members: Alex Ajangu, Kaspar Põder, Miriam Nurm

## Task 1

https://github.com/poderkas/clinvar_projekt

## Task 2

- Identifying your business goals
    - Background - ClinVar database is used in clinical practice to estimate whether a genetic variant contributes to a disease or not. A consensus is reached based on classification submissions from various laboratories, but these submissions may not overlap and thus create a classification conflict which makes clinical interpretation difficult.
    - Business goals - Developing an algorithm that would correctly predict whether the variant has a conflicting classification and determining which features influence a classification conflict the most.
    - Business success criteria - accuracy score, list of features in order of importance

- Assessing your situation
    - Inventory of resources - kaggle, course materials
    - Requirements, assumptions, and constraints - we must complete the project (code, video, poster) before the 14th of December.
    - Risks and contingencies - the biggest risk for our project would be if our current choice of algorithm does not work, in that case we could try another classification algorithm like SVM.

    - Terminology - snp, gene, dna, point mutations, genetic variants, genomics, chromosomes, pathogenic, benign

- Defining your data-mining goals
    - Data-mining goals - Obtaining the ability to classify the clinical significance of a given genetic variant.
    - Data-mining success criteria - accuracy score, logically appraising the features (e.g whether it makes sense that the chromosome number would affect classification

Our group consists of three members, two of which are mostly with a genetic engineering and one with fully IT background. In our project we use a dataset from Kaggle, a crowd-sourced platform often referred to as “AirBnB for data scientists” which trains and challanges data scientists all around the world with all kinds of data analysis, machine learning and predictive analytics problems. Our dataset is based on annotations about human genetic variants from ClinVar, a database used in clinical practice to estimate whether a genetic variant contributes to a disease or not. A consensus is reached based on classification submissions from various laboratories, but these submissions may not overlap and thus create a classification conflict which makes clinical interpretation difficult.

To better understand this, we will explain the frequently used terminology in this project. We will be dealing with human genes, functional units of heredity which are made up of DNA and act as instructions to make amino acid chains which then are turned into proteins (building blocks for cells) and enzymes (proteins that carry out tasks in the body other than just being building material). Genes can have defects caused by various mutations. The ones we will be dealing in this project are called single nucleotide polymorphisms, also known as SNPs (pronounced “snips”), which are the most common type of genetic variation among people. Each SNP represents a difference in a single DNA building block, called a nucleotide. For example, a SNP may replace the nucleotide cytosine (C) with the nucleotide thymine (T) in a certain stretch of DNA. SNP’s can vary in their impact on our organism: some may be

pathogenic (cause for disease(s)), neutral (with no impact) or even benign (with a positive impact). Classification of SNP's is a rather difficult task, since the prediction of their effect on the body can most clearly be seen only on the patients they have been discovered in and there can be great variability in patients.

The problem posed in Kaggle, which is also our projects goal, is to determine which features from the given dataset influence a classification conflict the most (we would create a list of features in order of importance) and to develop an algorithm which would correctly predict whether the variant has a conflicting classification (success criteria of which would be the accuracy score). In order to achieve all this, we will have to find, and possibly modify, algorithms which would fit our needs the most since one of the biggest risk for our project is that our current choice of algorithm, Random forest classifier, would not work. If that would be the case, we could try another classification algorithm like SVM etc. To solve any problems which arise we will use information gathered from the course materials, from our genetic engineering and other IT courses, kaggle and google.

This problem is topical since many problems can arise from misinterpretation of possible disease vectors. In the future, this project, if the goals are to be achieved, could potentially help biomedicine scientists and medical workers to better understand the cause of these classification conflicts and predict them beforehand.

# Task 3

Gathering data:

Our data comes from an open dataset which was uploaded to Kaggle. In this dataset we have an outtake from the ClinVar database that has compiled rows of individual genetic variants with their corresponding features. Since our whole project is built on this one dataset, we are not specifically collecting the data ourselves, but we are using this particular dataset. In order to train our classification algorithm, we need as many different genetic variants and features as possible to have enough data to split into test and training sets and to reduce overfitting. The most practical data format for us would be .csv file where we have the genetic variants as rows and features as columns properly indexed.

As stated before, our data comes entirely from a publicly available dataset from Kaggle which itself has been compiled from the publicly available database ClinVar. This amount of data should be sufficient for the task (over 65 000 genetic variants) so we would not need to look for data elsewhere.

Our primary information for the features comes from their descriptions on Kaggle and from the prior experience of one of the team members. We can also use databases such as Ensembl or dbSNP or ClinVar itself for looking up information on genetic variants or to better understand the interpretation of the several pathogenicity scores given as features in the dataset. This information would also be helpful to us for selecting the features that we eventually want to keep for training our algorithm.

Describing data:

Our source dataset of ClinVar genetic variants is a .csv file, originating from Kaggle. The dataset has a size of 29.2 MB, 65 189 rows and 46 columns. Each row represents a genetic variant and the columns have the following significations:

- CHROM - Chromosome the variant is located on
- POS - Position on the chromosome the variant is located on.
- REF - Reference Allele
- ALT - Alternate Allele
- AF_ESP - Allele frequencies from GO-ESP (variant database)
- AF_EXAC - Allele frequencies from ExAC (variant database)
- CLNDISDB - Tag-value pairs of disease database name and identifier, e.g. OMIM:NNNNNN
- CLNDISDBINCL - For included Variant: Tag-value pairs of disease database name and identifier, e.g. OMIM:NNNNNN
- CLNDN - ClinVar's preferred disease name for the concept specified by disease identifiers in CLNDISDB
- CLNDNINCL - For included Variant : ClinVar's preferred disease name for the concept specified by disease identifiers in CLNDISDB
- CLNHGVS - Top-level (primary assembly, alt, or patch) HGVS (variant database) expression.

- CLNSIGINCL - Clinical significance for a haplotype or genotype that includes this variant. Reported as pairs of VariationID:clinical
- CLNVC - Variant Type
- CLNVI - the variant's clinical sources reported as tag-value pairs of database and variant identifier
- MC - comma separated list of molecular consequence in the form of Sequence Ontology ID|molecular_consequence
- ORIGIN - Allele origin. One or more of the following values may be added: 0 - unknown; 1 - germline; 2 - somatic; 4 - inherited; 8 - paternal; 16 - maternal; 32 - de-novo; 64 - biparental; 128 - uniparental; 256 - not-tested; 512 - tested-inconclusive; 1073741824 - other
- SSR - Variant Suspect Reason Codes. One or more of the following values may be added: 0 - unspecified, 1 - Paralog, 2 - byEST, 4 - oldAlign, 8 - Para_EST, 16 - 1kg_failed, 1024 - other
- CLASS - The binary representation of the target class. 0 represents no conflicting submissions and 1 represents conflicting submissions.
- Allele - the variant allele used to calculate the consequence
- Consequence - Type of consequence: https://useast.ensembl.org/info/genome/variation/prediction/predicted_data.html#consequences
- IMPACT - the impact modifier for the consequence type
- SYMBOL - gene name
- Feature_type - type of feature. Currently one of Transcript, RegulatoryFeature, MotifFeature.
- Feature - Ensembl (variant database) stable ID of feature
- BIOTYPE - Biotype of transcript or regulatory feature
- EXON - the exon number (out of total number)
- INTRON - the intron number (out of total number)
- cDNA_position - relative position of base pair in cDNA sequence
- CDS_position - relative position of base pair in coding sequence
- Protein_position - relative position of amino acid in protein
- Amino_acids - only given if the variant affects the protein-coding sequence
- Codons - the alternative codons with the variant base in upper case
- DISTANCE - Shortest distance from variant to transcript

- STRAND - defined as + (forward) or - (reverse).
- BAM_EDIT - Indicates success or failure of edit using BAM file
- SIFT - the SIFT prediction and/or score, with both given as prediction(score)
- PolyPhen - the PolyPhen prediction and/or score
- MOTIF_NAME - the source and identifier of a transcription factor binding profile aligned at this position
- MOTIF_POS - The relative position of the variation in the aligned TFBP
- HIGH_INF_POS - a flag indicating if the variant falls in a high information position of a transcription factor binding profile (TFBP)
- MOTIF_SCORE_CHANGE - The difference in motif score of the reference and variant sequences for the TFBP
- LoFtool - Loss of Function tolerance score for loss of function variants: https://github.com/konradjk/loftee
- CADD_PHRED - Phred-scaled CADD score.
- CADD_RAW - Score of the deleteriousness of variants: http://cadd.gs.washington.edu/
- BLOSUM62 - score indicating the impact of the amino acid change. See: http://rosalind.info/glossary/blosum62/

The data quality in general is good, but we will have to discard several columns (like SSR) due to missing values. Missing values appear at a lesser rate in other columns as well, but that can be overcome with one-hot-encoding.

# Task 4

**Step 1. Analysing our obtained dataset and determining parts of the data which are complete and useful in our task of reaching our data mining goal.**

Many columns have information which has nothing to do with the end result of classifying the genetic variants, it is of little use to include this data in the training of our classifier. For looking at the data like this manually we've been using MS Excel.

**Step 2. Based on decisions made in the previous step we will clean up the data and prepare it for analysis such that there are no erroneous entries or faulty**

**values which would prevent us from training an effective classifier which works on good data.**

Again, this is manual work to an extent, eliminating clearly faulty values which would skew any training results down the line.

**Step 3. After cleaning up the data we will prepare training and test data sets for training our classification algorithm. Using these datasets we will try to train the most accurate classifier we can, with the end-goal being the ability to give useful results in line with our data mining goals.**

Using pandas and python machine learning libraries we will try to create the most accurate classifier we can, trying different classifiers and approaches to see which will get us the best results.

**Step 4. Determine the features most impactful in the result of the classification of a given genetic variant. Assign weights to the results based on these values in order to try to further enhance the accuracy of our model.**

Using the trained classifier, we will extract the weights it assigned internally to each feature and try to make decisions as to why these are the most likely to determine whether a conflict of opinion will occur on the matter of clinical significance.

**Step 5. Based on our final results we will prepare a poster presentation and video describing our work and the results we obtained.**

Using what will most likely be graphs generated in python and an image processing tool like GIMP we will create a large poster detailing the end results of our project.

**Time allotment**

Up until this point we have always worked together in a video call when working on the project, we do not see this trend changing much, so for the most part our time spent on each task will come out to be about equal. At this point we have reached the end of step 2 in the project with each of us having contributed around 18 hours of work cumulatively. We predict that step 3 and step 4 together will be the brunt of the

work with each of us working around 15-20 hours to complete the classifier and finish the data science part of the project. The poster and presentation together will most likely take around 2-5 hours of individual work from all members to finish.