Mathematical Statistics
Stockholm University

Statistical Modeling of the Severity of Mutations from Protein and Genetic Data

Anna Olofsson

Examensarbete 2012:4

Postal address:

Mathematical Statistics
Dept. of Mathematics
Stockholm University
SE-106 91 Stockholm
Sweden

Internet:

http://www.math.su.se/matstat

Mathematical Statistics
Stockholm University
Examensarbete **2012:4**,
http://www.math.su.se/matstat

Statistical Modeling of the Severity of Mutations from Protein and Genetic Data

Anna Olofsson*

May 2012

Abstract

Single nucleotide polymorphisms (SNPs) are the most common form of genetic variation in humans. The number of SNPs identified in the human genome is growing rapidly, but attaining experimental knowledge about possible disease-associated variants is a laborious quest, and the main challenge is to narrow the list down to a few candidate genes where the mutations occur. At the moment the identification of candidate genes is quite intuitive. Current in-silico, mathematical and statistical tools provide only a very basic, sequence-based indication about the relevance of a mutation to a disease. There is a lack of multifactorial tools applying statistical, mathematical and biological knowledge to automatically estimate how interesting or relevant a mutation is to a disease by scoring it in some appropriate way: The higher the score the more likely it is that the mutation is disease-causing.

In this paper three SCoring Methods (SCM1-SCM3) are created for estimating the relevance of a mutation to a disease, separating deleterious mutations from neutral ones, each based on two types of data sets. The first one is PolyPhen-2, a web-based software tool, estimating the probability of a possible impact of a mutation on the protein level. The second one is the 1000 Genomes Project, an online catalogue storing information about variations in the population (i.e., the allele frequency). These two factors are combined in different ways for the investigated scores.

Either p-values were calculated, using training data and Fisher's exact and combined tests (SMC1), or logistic regression was used for predicting the probability that a mutation is harmful (SMC2), or a linear combination of the two factors was used as score (SMC3). In order to quantify how well benign mutations are separated from harmful ones, we used the area under the receiver operating characteristic curve, AUC.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: annaaolofsson@gmail.com. Supervisor: Ola Hössjer.

Acknowledgement

A lot of people have helped me, and I would like to relish this opportunity to thank them. I received generous encouragement and many hours of valuable discussions from my supervisor Ola Hössjer, at Stockholm University, early on in the project and I would like to express my gratitude to him for his support, valuable feedback and constructive instructions helping me move forward during this project. To the members of St Giles laboratory of Human Genetics of Infectious Diseases, thank you for an inspiring and ambitious atmosphere and for contributing with data sets to the project. Special thanks goes to Avinash Abhyankar and Yuval Itan for useful meetings, interesting and encouraging discussions, feedback, and suggestions, and also to Jean-Laurent Casanova for accepting me as a part of his laboratory. Thanks also goes to Elias Amselem for useful discussions and suggestions helping me on the way.

Contents

List of Figures

List of Tables

1 Introduction

DNA sequencing methods provide us with tens of thousands of genetic variations, so called polymorphisms, per individual, and the ability to discriminate between deleterious and benign variants could significantly improve the targeting of disease-causing mutations by filtering them into a reasonable number of sensible candidate genes, and then identifying those variations responsible for specific traits (phenotypes) from available data. Nonsynonymous single nucleotide polymorphisms (nsSNPs) is a type of SNP, believed to have the greatest impact on protein function because they often lead to mutation on the protein level. nsSNPs can be classified into two categories: those that are disease-associated (causing deleterious effect on protein level) and those that are neutral. Given the huge number of nsSNPs, a major challenge is to predict which of them are potentially disease-associated. Several computational methods have been developed for the classification of nsSNPs according to their predicted phenotypic effects, and one of them is the automatic software tool PolyPhen-2 (pph2). Also, many databases and catalogues of genomics and diseases have been established to store information about variations in the population, such as the 1000 Genomes Project (1kg) - an online catalogue created to collect human genetic variation from different population groups to represent the allele frequency in the population.

In this paper three SCoring Methods (SCM1-SCM3) are created based upon two datasets; pph2 and 1kg. For each mutation a probability is extracted from pph2 and an allele frequency from 1kg. These two factors are converted to p-values, and finally SCM1 combines them into one single p-value using Fisher's combined test. However, since we lack allele frequency data for patients in 1kg, we did not use p-values based on Fisher's combined test but rather a simplied p-value derived only from pph2 data. In SCM2 a logistic regression is performed using a set of training and testing data to estimate the cofactors and validate the model, respectively. In SCM3 we take a linear combination of mutation specific scores from pph2 and 1kg, giving us another measure of how deleterious the mutation is. For all three scoring methods, one may evaluate the scores for benign and disease-causing mutations with a set of training data and generate a Receiver Operating Characteristic (ROC) curve, for which the area under the curve (AUC) becomes a performance measure of the method.

1.1 Objectives

As the amount of mutation data and information about the genotypes of individuals increases, understanding the molecular level effects of variations and clarifying their possible disease-association is an important research challenge. The objective of this thesis is to create a statistical scoring tool estimating a mutation as damaging or benign/neutral. It is the beginning structure of a multifactorial tool to assess and process already existing factors and discriminate between deleterious mutations and neutral ones. The methodology, tools and theories that are used will be described and explained. Finally estimated values and produced figures will be presented, discussed and assessed.

1.2 Organization of the Report

Chapter 1 gives a brief introduction to the scoring project and why new tools are necessary. The main dogma in biology and genetics is explained in Chapter 2. Chapter 3 presents and describes the two parameters (1kg and pph2) used in all three scoring methods. In Chapter 4 data and technical details are brought to attention. Statistical methods and analysis is discussed and explained in Chapter 5. The results of the scoring is given in Chapter 6. The final Chapter 7 discusses important observations from the results Section, together with a brief continuation of further applications of the scoring methods.

2 Biological Background

Before we move on it can be a good idea to gain some knowledge about the biological structure and background of the genetic material discussed in this paper.

The DNA molecule is a double helix composed of two antiparallel chains joined together in a ladder-like arrangement of nucleotides; A (adenine), T (thymine), G (guanine) and C (cytosine), where A always binds to T and G always binds to C. The sequence of nucleotides on the DNA molecule encodes the genetic information, which is inside each cell in an organism.

A *gene* is a part of the DNA chain, and it is a collection of nucleotides containing the instruction for building a particular protein or specifying a specific trait of an organism; e.g., a gene responsible for your eye color. All of us have two copies of DNA and thus two copies of each gene; one inherited from the mother, and the other from the father. The specific sequences of nucleotides in a gene are called *alleles*. *One* allele thus represents *one* locus (genetic position) and consists of *one* nucleotide. When a gene is in an active state in a cell we say that the gene is *expressed*. Gene expression occurs in two steps. First, the transcription where DNA is used as a template for the creation of RNA, a molecule very similar to DNA. Second, the translation, during which the RNA strand is translated into a protein, important to cell function. Proteins are built up by 20 different kinds of *amino acids*. Each amino acid consists of one or more codons, which is a three-letter combination of nucleotides, who frequently only differ by one nucleotide.

For example, according to `Wikipedia` (2012a), the codons for the amino acid isoleucine are AUU, AUC, and AUA. A substitution changing the first nucleotide in one of the codons to either a U, C, or G would cause another amino acid to be inserted instead of isoleucine. The insertion of the wrong amino acid in a functional region of a protein may cause a disfunctional protein, which may result in a severe disease or even causing the death of the organism. If the substitution affects a less critical region of the protein, there may be no change in the resulting protein at all, according to the web page quincetree.com (2012). There can also be rare substitutions causing the protein to function in such a way that it is giving an organism a survival advantage.

Substitution resulting in a severe disfunctional protein is called a mutation, which is a change in genetic information. Genetic information is encoded by the order of the nucleotide bases of DNA, so a mutation represents a change in the order of those nucleotides. DNA sequencing includes several methods and technologies that are used for determining the order of these nucleotide bases in a molecule of DNA. Figure 1 represents two sequenced DNA fragments from different individuals. The C is substituted for a T and we say that there are two alleles (or variants) of the nucleotide at this position; C and T. If the pair {C,T} occurs at two homologous chromosomes of the same individual (inherited from the father and mother), it is called a genotype. If the more rare allele has a frequency of at least 1% then the variation at this locus is considered a *single nucleotide polymorphism* (SNP), which is a DNA sequence variation occuring among members of a population at a specific nucleotide (A, T, C or G).

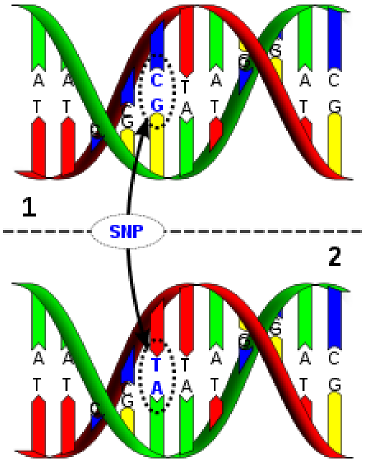

Figure 1: DNA sequences from two different individuals, Wikipedia (2012b).

If we look at the sequencies in Figure 1 vertically, between individuals (or species), we can see a SNP. SNPs can occur in protein coding regions and in non-coding regions, but in any case they can increase the risk of getting a certain disease. Coding SNPs can be further divided into: synonymous SNPs - no change in the amino acid sequence of a protein, and non-synonymous

SNPs (nsSNPs) - an amino acid substitution, that may have consequences on the structure and/or function of the encoded protein.

SNPs do not have to be strongly associated with a certain disease, but they can nevertheless help determining the probability that someone will develope the disease. For example, someone who has inherited two copies of a disease-associated allele may never develop the disease, whereas another person with the same two alleles, i.e., the same genotype, may do so. This is called incomplete penetrance of the disease-associated allele and it makes genetic testing a lot more complicated.

There are different kinds of mutations. A *point mutation* is a mutation that alters a single nucleotide. It includes insertions (a base is added), transitions (a base is exchanged for another base), deletions (a base is deleted) and transversion (a base-pair is exchanged for another base-pair). Examples of point mutations include: *Missense mutations* (a type of nonsynonymous mutation), that changes a codon so that a different protein is created, and may result in a nonfunctional protein and possibly leading to a certain disease. *Nonsense mutations*, which converts an amino acid codon into a stop codon, that may lead to the protein being cut off. This can lead to a nonfunctional protein depending on how much of the protein that is lost. *Silent mutations* code for the same amino acid and has no effect on the functioning of the genome. This can also be called a synonymous change, because the old and new codon code for the same amino acid. This is possible because 64 codons specify only 20 amino acids. See `Wikipedia` (2012c).

A point mutation and a SNP are closely related concepts. Both are single-nucleotide differences in a DNA sequence, but in order to be classified as a SNP, the change must be present in at least 1% of the general population and no known disease-causing mutations are this common. Also, most disease-causing mutations occur within a gene's coding regions and affect the function of the protein encoded by the gene, but SNPs don't necessarily need to be located within genes, and they do not always affect protein function.

The *Hardy-Weinberg equilibrium* (HWE) theory serves as the basic null model for population genetics. Every individual has alleles that were passed on from their parents. If we take all of the alleles of a group of individuals of the same species (that is, a population) we have what is called the gene pool. The frequency of individuals in that population that possess a certain allele is called the *allele frequency* and it is the proportion of *one* allele relative to *all* copies of the genomic region at which the mutation has occurred in the whole population. Since each individual has two homologous copies of this region, the allele frequency is thus the number of copies of the allele divided by twice the population size. Populations can have allele frequencies, but individuals cannot. This obviously makes populations the best level in order to study evolution, as evolution is basically the study of the change in allele frequencies over time. The HWE key assumptions are:

- Random mating
- No mutations
- No migration of individuals (neither in or out)
- Infinite population size
- No selection

The simplest case is a single locus with two alleles; A and a, with respective frequencies denoted by p and q, where $p + q = 1$. The Hardy-Weinberg equilibrium holds if the genotype frequencies satisfy

$$\begin{aligned} \text{freq}(AA) &= p^2 \\ \text{freq}(Aa) &= 2pq \\ \text{freq}(aa) &= q^2 \end{aligned}$$

See `Wikipedia` (2012d).

3 Model Parameters

We have a collection of SNPs that we want to investigate using two kinds of datasets. First, we obtain allele frequencies by means of association analysis from a population of the *1000 Genomes Project* (1kg), found at `1000genomes.org` (2012). Second, we obtain variations at the protein level, i.e, *PolyPhen-2* (pph2), found at the web site `genetics.bwh.harvard.edu` (2012), containing probabilities predicting whether a variation is deleterious or benign.

3.1 The 1000 Genomes Project

The human genome consists of about 3 billion DNA base pairs and carries approximately 20,000–25,000 protein coding genes. Recall from previous Section that many SNPs have no effect on cell function, but others are believed to be disease-associated. Although more than 99% of human DNA sequences are the same, variations in the DNA sequence can have a major impact on how humans respond to disease-associated SNPs. SNPs are evolutionarily stable, not changing much from generation to generation, which makes them easier to follow in population studies. SNP maps could help identify the multiple genes associated with complex diseases. These associations are difficult to establish with conventional gene-searching methods because a single altered gene may only have a small contribution to the disease. Several research groups are working to find SNPs and ultimately create SNP maps of the human genome. Among these are 1kg, a catalogue of genetic variation in human populations, allowing for variation mapping among several different ethnicities. There are two kinds of genetic variants related to disease; 1) Rare genetic variants that have a damaging effect mostly on simple traits, such as monogenic diseases, 2) The genetic variants that are more common, having a mild effect and are thought to be involved in complex traits. 1kg tries to fill in the gaps of knowledge between these two types of genetic variants.

The project facilitates investigating the relationship between genotype and phenotype (observable characteristics, e.g., your eye color). According to the web site `1000.org` (2012), the project reached the first intermediate goal 2010 by sequencing the genomes of at least one thousand anonymous participants from a number of different ethnic groups to become a detailed catalogue of human genetic variations. This first sample consists of 1167 individuals from 13 populations. The main goal of 1kg is to sequence about 2500 samples.

The database is a useful tool in, for example, association studies relating variation to disease and understanding the underlying processes of mutation. Once the disease-associated regions are identified, the next step is to find all of the variants in those regions. 1kg provides data on almost all of the variants with a frequency of at least 1% in the individuals studied. The project aims to discover genetic variants that have frequencies of as low as 1% across the genomes and 0.1-0.5% in gene regions.

By using 1kg, researchers can save time and energy not having to sequence their own samples. The list of SNPs in the 1kg will not tell which variants that increase the risk of a disease, but it will give you the set of suspects, which might significantly narrow the list down. Then further experimental studies may only involve collection of phenotypes at 1kg SNPs from the population under study, for instance disease cases and healthy controls. Refering to the article *A map of human genome variation from population-scale sequencing*, Nature (2010).

3.2 PolyPhen-2

PolyPhen-2 (Polymorphism Phenotyping version 2) is a web-based software tool using sequence and structure-based features of the substitution site to predict nsSNPs as *damaging* - possibly affecting the protein function, or *benign* - nondamaging. Polyphen performs several steps and produces different values. We will only go into depth on how the PolyPhen-2 (pph2) probabilities, predicting a mutation as damaging or benign, are calculated. For more details about the pipeline and algorithm see the PolyPhen-2 web site referred to in the references. The pph2 probabilities are Bayes posterior probabilities produced by a Naive Bayes model that uses 11 different features to calculate the posterior probability together with an entropy-based disretization for discretizing the numeric feature values into nominal values. A Naive Bayes score close to 1 indicates a damaging mutation and a score close to 0 a benign. Sometimes zero probabilities arise and smoothing can

Table 1: The 11 selected feature variables, see Adzhubei et al (2010), *Supplementary Methods*.

Feature name	Definition
score1	PSIC score for the wild type allele
score_delta	difference of PSIC score between wild type allele and mutant allele
num_observ	number of residues observed at the position of the multiple alignment
delta_volume	change in residue side chain volume
pfam_hit	position of the mutation within/outsidde a protein domain as defined by Pfam
id_p_max	congruency of the mutant allele to the multiple alignment
id_q_min	sequence identity with the closest homologue deviating from wild type allele
cpg_transition	whether variant happened as transition in CpG context
acc_norm	normalized accessible surface area of amino acid residue
b_fact	crystallographic beta-factor
delta_prop_new	change in accessible surface area propensity for buried residues

be done with Laplace estimators. Naive Bayes requires data for training and testing, to use in 5-fold cross-validation. Two datasets are used for training and testing; $HumDiv$ contains 3,155 damaging mutations, together with 6,321 human nsSNPS assumed to be non-damaging; $HumVar$ consists of 13,032 damaging mutations and 8,946 nsSNPS treated as non-damaging. Both datasets can be downloaded from the PolyPhen-2 web site including the *Whole human exome sequence space annotations* that will be used further down in SCM2 and SCM3 (see Chapter 5).

3.2.1 Input Data

The PolyPhen-2 input is the amino acid (aa) sequence of a protein or corresponding ID, together with sequence position and two aa variants characterizing the polymorphism. One aa variant corresponds to the aa in the reference sequence and the other corresponds to the aa resulting from the nsSNP. The input can for example look like the following string *'chr1:1267483 G/A'*. Where *chr1* denotes the chromosome, and *1267483* the chromosomal position, and G/A the mutation, in this case the aa reference G and the aa variant A.

3.2.2 Features

PolyPhen-2 is classifying a mutation as damaging or benign based on a set of 11 selected features (attributes), from a number of 32 features, through stepwise regression. Table 1 gives a brief description of the selected features. Stepwise regression is a technique that can be used for selecting a subset of features available from the data, that most contribute to predicting the damaging effect of a mutation. Through either forward selection or backward elimination, 11 features were automatically extracted in order to help classifying a mutation. Each instance (i.e. mutation) in a dataset are characterized by the values of these 11 features measuring different conditions of the instance.

3.2.3 PSIC Score

The Naive Bayes model utilizes 11 different features to calculate the posterior probability, one of these features is the PSIC (Position-Specific Independent Counts) score. This score reflects how likely it is for a particular aa to occupy a specific position in the protein sequence, given the pattern of aa substitutions observed in the multiple sequence alignment, and has the form of a likelihood ratio. It is computed using the PSIC algorithm, which takes the relatedness of homologous sequences into account and uses prior probabilities derived from the aa substitution matrix (BLOSUM62). The PSIC feature contributes with about 50% of the total predictive information content to the model, it is indeed a good proximation for predicting damaging effect of the substitution. The remaining 10 features collectively contribute slightly less than the PSIC score alone. For more information about the PSIC see Sunyaev, R. et al. (1999).

3.2.4 The Naive Bayes Model

Naive Bayes (NB) is a machine learning method *naively* assuming that features are independent from one another. The input to a machine learning scheme is a set of instances (*mutations*) that are to be classified and the output is the classification (*damaging* or *benign*) of the instance. In classification learning problems, a learner attempts to construct a classifier from a given training dataset with a set of instances with known classes. The nsSNPs data from PolyPhen-2 website presents two training datasets, $HumDiv$ and $HumVar$, containing example mutations together with a decision for each as to whether this mutation is damaging or not. The problem is to learn how to classify *new* mutations. The Naive Bayes classifier works as follows:

1. Let each mutation (nsSNP) be represented by a vector

$$F = (F_1, ..., F_M)$$

consisting of $i = 1, \dots, M$ features to base our classifiers on. Based on F, the objective is to assign a class,

$$C \in \{C_1, ..., C_m\}$$

consisting of $j = 1, \ldots, m$ classes, to each mutation. For instance, the classification problem could be binary ($m = 2$) with $C_1 = \text{benign}$ and $C_2 = \text{damaging}$.

2. Each feature is either categorical (with a fixed number of levels) or continuous. For the most part we will assume that all features are categorical. For example, if $M = 3$ and features 1 and 3 are binary with two levels no and yes, and feature 2 has levels low, medium and high, the observed feature vector could be $F = (\text{yes}, \text{medium}, \text{no})$.

3. For a nsSNP, Bayes produces a posterior to train the classifier. Bayes' theorem can be expressed as,

$$P(C_j|F) = \frac{P(C_j)P(F|C_j)}{P(F)} \tag{1}$$

Given a specific feature vector of F, the classifier will predict that F belongs to the class having the highest *a posteriori* probability, $P(C_j|F_i)$, conditioned on F. That is, the probability that the hypothesis (e.g., benign) for the class holds given the *evidence* vector, F. The *a priori*, probability of the hypothesis, $P(C_j)$, is the probability that the outcome for the new instance belongs to class C_j without knowing any of the features F. The *goal* is to find the class, C, that maximizes the posterior, $P(C_j|F_i)$. In other words, we are looking for the probability that sample F belongs to class C, given that we know the feature values of F.

PolyPhen-2 utilizes 11 features, $F = (F_1, \ldots, F_{11})$, to base the classifiers C on, and the evidence is the particular combination of feature values for the new mutation. Suppose that $m = 2$ and the hypothesis is that the mutation is damaging. Then $P(damaging|F_1, F_2, ..., F_{11})$ is the probability that the mutation being observed is damaging given that we know $(F_1, ..., F_{11})$ of that mutation. In contrast, the *a* priori probability of the hypothesis, $P(C_j)$, is the probability of a damaging outcome without knowing F.

4. We are only interested in the numerator of (1), since the denominator $P(F)$ does not dependend on the class, and hence does not affect the maximization of the posterior. So, only $P(C_j)P(F_i|C_j)$ need to be maximized. If very little prior knowledge of the class C is available, one usually assumes an uniformative (uniform) prior

$$P(C_1) = ... = P(C_m) = 1/m$$

Then, only the likelihood, $P(F|C_j)$ needs to be maximized.

5. Given many features a simplification might be needed to make it less computationally expensive to calculate the posterior. Therefore a $Naive$ Bayes assumption can be made, that the values of the attributes are conditionally independent of one another given the class of the sample. Mathematically, we can phrase this as:

$$P(F_i|C_j, \{F_k;\, k \neq i\}) = P(F_i|C_j), \quad \text{for all } i \in \{1, \ldots, M\} \text{ and } j \in \{1, \ldots, m\}$$

For the likelihood this means that

$$P(F|C_j) = \prod_{i=1}^{M} P(F_i|C_j),$$

where F_i refers to the value of the ith feature for a specific mutation. The probabilities $P(F_1|C_j), P(F_2|C_j), ..., P(F_M|C_j)$ can easily be estimated from a training set, as will be further described in the next section. For *numeric* attributes an entropy-based discretization (Section 3.2.6) is applied before the calculation. See `Wikipedia` (2012e) and Witten et al. (2011).

3.2.5 Parameter Estimation

In order for the Naive Bayes classification to work, we must estimate the prior probabilities $P(C_j)$, and also the likelihoods $P(F_i|C_j)$. This is either done simultaneously with classification, or, if training data is available, in a preliminary step. In the simplest case all features have already been extracted (this assumption will be relaxed in section 3.2.8) and the feature vector $F^l = (F_1^l, \ldots, F_M^l)$ and the classification $C^l = (C_1^l, \ldots, C_m^l)$ is collected for a training dataset of size $l = 1, \ldots, n$. Note, that we will use superscript to describe the serial number of the entire feature vector in a training dataset, to be able to seperate it from subscripts, F_i, which denotes a component in a specific feature vector. Let n_i denote the number of observations l in this training set with $C^l = C_j$. Then the prior probabilities can be estimated by maximum likelihood as

$$\hat{P}(C_j) = \frac{n_j}{n}, \quad j = 1, \ldots, m$$

So, the class *a* priori probabilities of each class of a hypothesis (that our mutation is, e.g., benign) may be estimated with

$$\text{prior for a given class} = \frac{\text{number of samples in the class}}{\text{total number of samples}}$$

We also need to estimate the likelihood terms $P(F_i|C_j)$. For categorial features F_i with a finite but large number of possible levels, training data might be missing for some levels. Then the Laplace estimator (a Bayesian estimator based on a uniform Dirichlet prior for the probabilities of the various levels), can be used in order to guarantee that each estimated $P(F_i|C_j)$ is strictly between 0 and 1. Suppose for instance that the i:th feature has a_i levels and let n_{jik} denote the number of observations l in the training data set for which $C^l = C_j$ on one hand and F_i^l equals the k:th level on the other hand (so that $\sum_{k=1}^{a_i} n_{jik} = n_j$). Then the Laplace estimator is defined as

$$\hat{P}(F_i = \text{level } k|C_j) = \frac{n_{jik} + 1}{n_j + a_i}, \quad k = 1, \ldots, a_i.$$

Suppose for instance that F_2 is a feature with three levels (low, medium, high), and that the training samples contains $n_j = 1000$ observations at level C_j. If, for example, none have $F_2^l =$ low, 350 have $F_2^l =$ medium and 650 have $F_2^l =$ high, then, the estimated probabilities of these events, without the Laplacian correction, are 0, 0.350, and 0.650, respectively. Using the Laplacian correction, we instead obtain the following probabilities (rounded up to three decimals): $\frac{1}{1003} = 0.001$, $\frac{351}{1003} = 0.350$, $\frac{651}{1003} = 0.649$ and the problem of a zero probability value has disappeared. See Witten, Frank, and Hall.

For continous (numeric) features, one may fit a parametric model (for instance a Gaussian distribution) with few parameters to $P(F_i|C_j)$. However, since the Naive Bayes is often applied to large datasets, one typically uses a nonparametric approach by discretizing numeric feature values and then applying the above mentioned Laplace estimator.

3.2.6 Entropy-Based Discretization

To handle continous-valued attributes one can use binning to *discretize* the values into a small number of distinct ranges, while still within the range of the variable's values, so that they are reported on a nominal scale. Discretizing requires a set of training data and can be achieved by constructing a tree. Each inner node corresponds to a split of some level (corresponding to an interval) of the feature variable into two or more disjoint and smaller subintervals. New splits are generated recursively until the leaf nodes are reached, representing the final and finest level of discretization. If only one cutting point is allowed for at each node, we get two subintervals and a binary tree. An unknown instance is assigned a range or level by being guided down the tree according to the values of the attributes tested in successive nodes, and when a leaf is reached the instance is classified according to the class assigned to that leaf.

For a given training dataset we have two distinct problems to solve. First, how many splits to make? Second, where to cut an interval into two subintervals? Entropy-based discretization, by Fayyad and Irani (1993), coupled with a minimum description length (MDL) criterion answers the first question and entropy calculation answers the second one. This method is called ENT-MDL.

To simplify things, let's first introduce some useful terminology. Let S be a dataset of n instances consisting of the list

$$S = \{(X^1, C^1), \dots, (X^n, C^n)\},$$

that is sorted in ascending order of X^l, where X^l represents the continuous feature variable, and $C^l \in \{C_1, \dots, C_m\}$ is the class variable for item l. Let $S_{a,b}$ be a subset list of the first elements of S, starting at the a^{th} pair in S and ending at the b^{th} pair. For a binary split, let a threshold value T be the cutpoint partitioning the dataset $S_{a,b}$ into two branches (intervals) S_1

Table 2: Dataset, S, containing an attribute temperature and a decision, measured over 14 days. See Witt et al.

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Temperature	64	65	68	69	70	71	72	72	75	75	80	81	83	85
Decision	1	0	1	1	1	0	0	1	1	1	0	1	1	0

and S_2. Referring to Fayyad U. and Irani K. (1992), the entropy of $S_{a,b}$ is defined as:

$$\text{Ent}(S_{a,b}) = -\sum_{j=1}^{m} P(C_j, S_{a,b}) \log(P(C_j, S_{a,b}))$$

When the logarithm base is 2, $\text{Ent}(S_{a,b})$ measures the amount of information needed, in *bits*, to specify the classes in $S_{a,b}$. $P(C_j, S_{a,b})$ is the proportion of examples in $S_{a,b}$ assigned to a class C_j.

ENT-MDL recursively splits an interval, containing all known values of a feature, at the point that minimizes the information class entropy of the subintervals. A minimum description length (MDL) criterion is applied to decide whether to actually execute a split, when to stop discretization, and also to control the number of intervals partitioned. It is optimal to choose the MDL of a split with minimum number of bits.

The class information entropy for S_1 and S_2 can be expressed as

$$\text{Info}(S_1, S_2) = \frac{|S_1|}{|S_{a,b}|}\text{Ent}(S_1) + \frac{|S_2|}{|S_{a,b}|}\text{Ent}(S_2)$$

where $|S_{a,b}|$ refers to the number of elements of S, and similarly for $|S_1|$ and $|S_2|$. The entropy measures whether a new instance should be classified and it is calculated based on the number of positive and negative classes in the decision list.

For example, look at the dataset $S_{a,b}$ described in Table 2, where each data pair (representing a specific day) of an attribute variable X, e.g., the *temperature* in Fahrenheit, taking values in the range 0 to 100, together with a decision, say, whether we should play tennis (yes=1) or not (no=0).

For instance, we could cut the whole dataset S (so that $S = S_{a,b}$) into two branches using a threshold $T = 71.5$. This gives us two intervals

$$\begin{aligned} \text{high} &= (71.5, 100) \\ \text{low} &= (0, 71.5) \end{aligned}$$

The *low* interval contains four *yes's* and two *no's*, and the *high* interval contains five *yes's* and three *no's*. The class information entropy of the subsets is given by:

$\text{Info}([4,2],[5,3]) = \frac{6}{14}\text{Info}[4,2] + \frac{8}{14}\text{Info}[5,3] = 0.939$ bits where

Info[4,2]=Ent($\frac{4}{6}$, $\frac{2}{6}$)=-$\frac{4}{6}$log($\frac{4}{6}$)- $\frac{2}{6}$log($\frac{2}{6}$)=0.918 bits

and

Info[5,3]=Ent($\frac{5}{8}$, $\frac{3}{8}$)=-$\frac{5}{8}$log($\frac{5}{8}$)-$\frac{3}{8}$log($\frac{3}{8}$)=0.954 bits

Alternatively, suppose that in Table 2 we choose to make two splits at temperatures $T_1 = 69.5$ and $T_2 = 77.5$ simultaneously, so that the temperature range is divided into three subintervals

$$\begin{aligned} \text{hot} &= (77.5, 100) \\ \text{mild} &= (69.5, 77.5) \\ \text{cool} &= (0, 69.5) \end{aligned}$$

where *hot* consists of two yes's and two no's (i.e., [2,2]), *mild* of four yes's and two no's (i.e., [4,2]) and *cool* of three yes's and one no (i.e., [3,1]). The entropy of *temperature* = *hot* would be

Info([2,2])=Ent($\frac{2}{4}$, $\frac{2}{4}$)=-$\frac{2}{4}$log($\frac{2}{4}$)-$\frac{2}{4}$log($\frac{2}{4}$)=1 bits

and similarly for *mild* and *cool*. The expected information becomes

Info([2,2],[4,2],[3,1])=$\frac{4}{14}$·1+$\frac{6}{14}$·0.918+$\frac{4}{14}$·0.811=0.911 bits

This represents the amount of expected information to classify a new instance, given the list structure of a specific feature. The entire structure list, i.e., all data in S, of *temperature* consists of nine yes's and five no's corresponding to an information value of

Info([9,5])=-$\frac{9}{14}$log($\frac{9}{14}$)-$\frac{5}{14}$log($\frac{5}{14}$)=0.940 bits

One can choose to split at the point(s) where the information value is the smallest, and this is equal to splitting where the information *gain* is the largest. The information *gain* is defined as the difference between the information value *without* the split and the one *with* the split. So, the information *gain* for temperature when splitting the dataset S at two points $T_1 = 69.5$ and $T_2 = 77.5$ is

$$\begin{aligned} \text{gain}(\text{temperature}, (69.5, 77.5), S) &= \text{Info}([9,5]) - \text{Info}([2,2],[4,2],[3,1]) \\ &= 0.940 - 0.911 \\ &= 0.029 \text{ bits} \end{aligned}$$

A cut point that minimizes the information class entropy value never occurs between two instances of the same class. So, it is not necessary to split an interval further in the ideal case when all training samples within this interval contain the same class variable C_j. In other words, the information entropy value becomes zero.

Choosing an optimal partition is a compromise between having few subsets in the partition on one hand and subsets that discriminate well between classes C_j, i.e., when the information entropy value becomes zero. In more detail, for each leaf node of the partition tree, a decision whether a further cut should be made or not is based on the MDL Principle. For a binary tree, suppose that a cutting point T of a set $S_{a,b}$ of values of the continuous attribute X divides it into subsets S_1 and S_2. Then the cut is accepted if

$$\begin{aligned} \text{gain}(X, T, S_{a,b}) &> \frac{\log_2(|S_{a,b}|-1)}{|S_{a,b}|} \\ &= \frac{\log_2(3^{m_{a,b}}-2)-m_{a,b}\text{Ent}(S_{a,b})+m_1\text{Ent}(S_1)+m_2\text{Ent}(S_2)}{|S_{a,b}|} \end{aligned}$$

where $m_{a,b}$, m_1 and m_2 is the number of the m classes C_j that occur in $S_{a,b}$, S_1 and S_2, respectively.

When all feature values have been discretized, the dataset can continue to be divided into training and testing sets. See Witt et al. (2011), for further information about entropy discretization.

3.2.7 ROC

The receiver operator characteristic (ROC) curve for a binary classification problem plots the true positive rate (TPR) as a function of the false positive rate (FPR). The points of the curve are obtained through the various possible threshold values, as FPR varies between 0 and 1. ROC curves are applied for deciding whether a mutation is benign (the null hypothesis, $C = C_1$) against the alternative hypothesis that it is damaging ($C = C_2$), based on feature vector data F.

The *Sensitivity* equals the TPR, i.e., the proportion of positive cases that are well detected by the test. The mathematical definition is given by:

$$\text{Sensitivity} = \frac{\text{correctly classified positive}}{\text{total positive}} = \frac{\text{TP}}{\text{TP + FN}}$$

where *TP* (true positive) is the number of damaging mutations classified as damaging and *FN* (false negative) is the number of damaging mutations misclassified as benign.

The *Specificity* equals one minus the FPR, i.e., the proportion of negative cases not detected by the test. The mathematical definition is given by:

$$\text{Specificity} = \frac{\text{correctly classified negative}}{\text{total negative}} = \frac{\text{TN}}{\text{TN + FP}}$$

where *TP* (true positive) is the number of benign mutations classified as benign and *FP* (false positive) is the number of benign mutations misclassified as damaging.

The prediction accuracy is quantified as the area under the ROC curve (AUC), i.e., the average sensitivity obtained when integrating over various specificities from 0 to 1. Often one considers a value of AUC around 0.85-0.9 or higher to be sufficient for good discrimination between the null and alternative hypotheses, although this depends on the type of application.

3.2.8 Cross-Validation

Naive Bayes classification requiers data for training and testing. We have already discussed how to use training data for parameter estimation in section 3.2.5. We will now describe how to select an optimal set of features. PolyPhen-2 uses 5-fold cross-validation to decide when to stop the feature selection algorithm (i.e., the stepwise regression) and to evaluate the performance of the selected features. To predict the performance of a classifier on new data, we divide the data into training set and test set; one fold is retained as the test set, and the remaining four folds are together used as training set. 5-fold means retraining the model five times, so that each of the five folds are used exactly once as the test set. During the test procedure we must know the classification of all instances in the training and test data. For further explanation see Witten et al. (2011).

For the training data S in, for example, *HumDiv* we have recorded a number of feature variables for a set of $n = 9476$ mutations, divided into 6321 benign mutations and 3155 damaging mutations. Randomly split S into five approximately equal partitions S_1,S_2,...,S_5, i.e., with size $n/5$, and each in turn is used for testing and the remainder for training. Then, the predictors are learned based on the training data and the values that yield maximum accuracy are used. This accuracy is evaluated on the test set to give an idea of how well this model will perform on future data. At the end, the predictive performance of a given set of feature vectors is averaged over the 5 different test sets and quantified by means of the AUC.

The total list of features is

$$F_{all} = (F_1, \dots, F_N),$$

and the objective is to extract the optimal set of features F_i, all of which we assume have been discretized according to the entropy based algorithm of Section 3.2.6. Now, do the following:

a) Choose a subset F of $M \leq N$ feature vectors and a test data set S_k and estimation set $S_{(-k)} = \{S_j;\, j \neq k\}$.

b) Use Naive Bayes to calculate the posterior. First, a number of parameters have to be estimated from the estimation data set; the priors $P(F_i)$ as well as the likelihood functions $P(F_i|C_j)$, as described in Section 3.2.5.

c) Set up a threshold t for $P(C = \mathit{damaging}|F)$ so that a value $< t$ is classified as *benign* and another value $> t$ is classified as *damaging*. Control, for the test dataset S_k, what proportion out of the *benign* mutations that has been incorrectly classified, i.e., *false positive rate* and the proportion out of the *damaging* mutations that has been correctly classified, i.e., *true positive rate*.

d) Given t, taking the average of the five different values of false positive rate and true positive rate received in c), a summarized false positive rate and true positive rate is received for all five ways to choose training and testing sets.

e) Repeat steps a-d) for different thresholds t in order to get a ROC curve with sensitivity plotted against specificity.

f) Repeat a-e) for different subsets of features F and choose the one with maximum AUC(F). Two procedures was used to choose features: forward selection includes one feature at a time (the best one that has not yet been included), until AUC(F) no longer increases and backward selection starts with F_{all} and excludes one feature vector at a time (the worst one of the leftovers), until AUC(F) no longer increases. (Both forward and backward selection gave the same 11 features).

g) For the optimal feature vector $F = (F_1, ..., F_M)$ from f), use

$$\text{NB score } = P(C = \mathit{damaging}|F)$$

as performance measure. Define thresholds $t_0 < t_1 < t_2 < 1$, so that a mutation is classified as *benign* (most likely lacking any phenotypic effect) when the NB probabiliy score belongs to $[0, t_1]$, as *possibly damaging* (i.e., it is supposed to affect protein function or structure) when it belongs to $[t_1, 1]$, and *probably damaging* (i.e., it is with high confidence supposed to affect protein function or structure) when it belongs to $[t_2, 1]$.

In PolyPhen-2 a nsSNP is predicted as **1)** *probably damaging*, if the fraction of FPR is under the 10%-level (TPR is 78%) on HumDiv and under the 19%-level (TPR is 71%) on HumVar, i.e., when the NB score exceeds $t_2 = 0.85$, **2)** *possibly damaging*, if the fraction of FPR is above the 18%-level (TPR is 89%) on HumDiv and under the 40%-level (TPR is 90%) on HumVar, i.e., the NB score is above 0.15, **3)** *benign*, for all the remaining mutations, **4)** *unknown*, if lack of data does not allow PolyPhen-2 to make a prediction. All according to Adzhubei et al. (2010).

3.2.9 WEKA

For each instance the 11 features are fed into a web-based machine-learning tool called WEKA (Waikato Environment for Knowledge Analysis). WEKA contains many different tools, for example, classification and regression. In WEKA PolyPhen-2 uses a Naive Bayes classifier model together with supervised entropy-based discretization (see sections 3.2.4-3.2.8) to train the predictor, along with some other options. All Bayes network algorithms in WEKA allows the user to discretize them on a nominal scale. See references for more details about WEKA.

4 Data and Technical Details

We are going to use two factors in our scoring methods described in the next Chapter. PolyPhen-2 displays a dataset, *Whole human exome sequence space annotations*, that can be downloaded at the web site. The dataset consists of pph2 annotations for 149,948,690 single-nucleotide nonsynonymous (missense) SNPs, and predictions were calculated using two datasets HumDiv and HumVar presented in Section 3.2. The 1000 Genomes Project contains information about the population allele frequencies, and the variants are assumed *not* to be disease-associated.

We are going to focus on chromosome 1 in a patient dataset consisting of 4464 variations from the Mendelian susceptibility to mycobacterial diseases (MSMD). For each such variation we extract a pph2 probability (the NB score) from the whole genome exome dataset. PolyPhen-2 only takes nsSNPs into consideration, while the patient dataset consists of other protein-coding mutations and also noncoding mutations, and therefore we only found a match of 536 observations (121 damaging and 415 nondamaging mutations). Variations not matching was simply not included. We will also use the benign mutations extracted from the pph2 HumDiv training dataset consisting of 6608 variations. These datasets will be used in Section 5.1.

A training dataset was also created to be used in Section 5.2.1, and for each matching mutation a probability from pph2's whole human exome dataset, and an allele frequency from 1kg were both extracted from chromosome 1. The training dataset consists of 926 mutations.

Also, for each mutation in the patient dataset a NB score and an allele frequency was extracted, and this dataset consisting of 174 mutations can be used in Section 5.3.

When we are extracting the data from pph2 and 1kg, there are a number of different scenarios that can arise. A large number of variations in our patient dataset did not match with any entry in the 1kg database, thus limiting the number of cases that can be analysed. If no allele frequency can be found for a certain mutation, then, it is rare and more likely to be damaging. If a mutation could not be found in pph2 or if that mutation have a probability of zero, then that variation was excluded from the model and classified as unknown.

4.1 Software

Statistical anlysis and data was processed, assessed and evaluated with Python and R.

5 Statistical Models and Scoring Methods

In this Chapter three scoring methods are being described. Before further introductions we will consider some therminology that we will use.

Assume that for each mutation i we have scores $X_{i1}, \ldots, X_{iK}$, computed from K data sets. A high value of X_{ij} indicates that mutation i is damaging according to data set j, whereas a low value indicates that it is benign. We want to combine $X_{i1}, \ldots, X_{iK}$ into one single number SCM_i for mutation i, using some scoring method, where a large value of SCM_i corresponds to a mutation more likely to be damaging.

In this work, we will consider $K = 2$ datasets, with X_1 computed from pph2 and a second score X_2 computed from 1kg. We will look at three SCoring Methods SCM1-SCM3, all of which quantify the relevance of a mutation's disease-association.

We would want to give more weight to 1kg, and therefore we will make a few assumptions regarding the classification of damaging and benign variations. The "unknown" mutations we are interested in scoring are very rare, so if we find a frequency (high or low) in the population, that is, in 1kg, it is assumed to be nondamaging. Note that this holds true only with the kind of rare diseases we are studying and cannot be generalized. The reason being, no matter how deleterious the polyphen prediction is, or how important the gene is, or how important the disease pathway is, if the mutation is found at a certain level in the general population it does *not* lead to the disease. In our case this "certain level" should be zero. Though, by looking at the pph2 probability independently might give some more weight to the model when we want to combine the "scores".

5.1 Scoring Method 1

For a given mutation i, this method is based on statistical hypothesis testing,

$$\begin{array}{ll} H_0: & i \text{ is benign} \\ H_1: & i \text{ is damaging} \end{array}$$

In principle, one can use Fisher's combined probability test in order to test the null hypothesis H_0, above, against the alternative hyptothesis H_1. It is based on a test statistic

$$T_i = -2\ln(X_{i1}) - 2\ln(X_{i2}),$$

with X_{i1} a p-value for mutation i computed from pph2 and X_{i2} a p-value computed from 1kg. The calculations of these p-values are further described in Section 5.1.1 and 5.1.2, respectively.

Under the null hypothesis that the two tests are independent and the mutation has no effect, X_{i1} and X_{i2} are i.i.d. (independent identically distributed) random variables with a uniform distribution on $(0, 1)$. Then T_i has a χ^2-distribution with 4 df. Therefore we can calculate

$$\mathrm{SCM1}_i = -\log(1 - G(T_i)),$$

where G is the distribution function, a χ^2-distribution with 4 df and $1-G(T_i)$ is the p-value associated with Fisher's combined test.

However, the p-value X_{i2} defined in Section 5.1.2 requires a data set of cases and controls. While this is of interest in future studies, in this paper we only have controls from the 1kg data set. We will therefore neglect X_{i2} and define a simplified score

$$\mathrm{SCM1}_i = -\log(X_{i1}) \tag{2}$$

based only on p-values from the pph2 data set. For instance, $\mathrm{SCM1}_i = 3$ corresponds to a p-value of 10^{-3} from pph2.

5.1.1 p-Value Computed from pph2

In order to test the null hypothesis of no damaging effect of mutation i against the alternative that i is damaging from pph2, assume that we have a test dataset containing NB scores (see Section 3.2.8) $\mathrm{NB}_1, \ldots, \mathrm{NB}_n$ from n benign mutations. Then their empirical distribution function

$$\hat{F}(x) = \frac{1}{n}\sum_{k=1}^{n} 1_{\{\mathrm{NB}_k \le x\}}$$

is an estimate of the population distribution F of NB-scores of non-damaging mutations. For instance, in the Polyphen-2 HumDiv dataset, that can be downloaded from the web site, we will only consider the neutral mutations in order to estimate F. The empirical distribution can be visualized with a "stair case function", where stairs of height $1/n$ are placed at the NB scores of the training dataset. When n grows, the empirical distribution will approximate the true distribution.

If mutation i does *not* belong to the training data set, we define the Naive Bayes (NB) score for mutation i as

$$X_{i1} = p\text{-value of mutation } i = 1 - \hat{F}(\text{NB score of mutation } i)$$

is the fraction of NB scores of the training data set greater than or equal to the NB score of mutation i.

Table 3: 2x2 contingency table.

	cases	controls	
damaging	a	b	a+b
nondamaging	c	d	c+d
	a+c	b+d	a+b+c+d

5.1.2 p-Value Computed from 1kg

Suppose that we have mutation (benign or damaging) and phenotype (case or control) data from $n = a + b + c + d$ alleles for a given nsSNP. We can organize them into a contingency table, like the one in Table 3. Fisher's exact test can be used for testing H_0 versus H_1. It is a hypothesis test which explores the association of categorical data, and can be used when comparing proportions. A contingency table like the one in Table 3 is then often used. Let Y denote the number of damaging mutations among the cases. Under the null hypothesis, the conditional distribution of Y (given the marginal sums in Table 3) is hypergeometrical,

$$P(Y = a) = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}.$$

This can be visualized as an urn problem, since it asks for the probability of obtaining a damaging mutations from case alleles when drawing $a + b$ balls (corresponding to damaging mutations) from an urn of $a + c$ case alleles and $b + d$ control alleles, i.e.

$$P(Y = a) = \frac{\binom{a+c}{a}\binom{b+d}{b}}{\binom{a+b+c+d}{a+b}}.$$

Let a_i, b_i, c_i, d_i denote the entries of Table 3 for mutation or nsSNP i. Then

$$X_{i2} = p - \text{value of mutation } i = P(Y \geq a_i),$$

when Y has the above mentioned hypergeometrical distribution.

5.2 Scoring Method 2

A second scoring model was created using logistic regression. It is similar to a linear regression model but is suited to models where the dependent variable is dichotomous, i.e., when there are only two categories that we are trying to predict (e.g., yes or no, female or male, success or failure). A set of training data was created containing already known deleterious and nonharmful mutations from pph2 and 1kg. We fitted the model on the training data by estimating the regression coefficients, then, we can use the same dataset for validating the model.

The goal of logistic regression is to correctly predict the category of outcome for individual cases using the greediest model. To accomplish this goal, a model is created that includes all predictor variables that are useful in predicting the response variable; in this case *pph2* and *1kg*. A major purpose of logistic regression is to predict group membership. Since logistic regression calculates the probability of success over the probability of failure, the results of the analysis can be seen in the form of an odds ratio.

A training dataset of 926 observations, containing known mutations with pph2 probabilities and 1kg allele frequencies, was used to build the model. When the regression coefficients has been estimated, the same training dataset was also used to determine the goodness of fit of the model and also its ability to distinguish benign from damaging mutations by means of the area under a ROC-curve.

5.2.1 The Logistic Regression Model

Logistic regression can be used when we have one dependent (response) variable that is dichotomous, and one or several independent variables. The response function has a binary outcome, coded as 0 or 1, where 1 could be seen as a *success* and 0 as a *failure*. We want to model the probability of success given the value of explanatory variables, $\pi = \Pr(Y = 1|X = x)$. For example, consider a vector of predictor or explanatory variables X, containing for instance risk factors that may contribute to a disease. Then, probability of success will depend on levels of these risk factors. Further, let Y be a binary response variable

$$\begin{aligned} Y_i &= 1 \quad \text{if the trait is present in observation } i \\ Y_i &= 0 \quad \text{if the trait is \textit{not} present in observation } i \end{aligned}$$

Suppose a training data set $\{(X_i, Y_i); i = 1, \dots, n\}$ of size n is available, with a response variable. Let $X = (X_{i1}, \dots, X_{iK})$ be a set of explanatory variables, and let x_{ir} be the observed value of the explanatory variables for observation i and parameter $r = 1, \dots, K$.

The logistic distribution constrains the estimated probabilities to lie between 0 and 1. To keep it within this interval a sigmoid response function called the logistic function is used, as can be seen in Figure 2, and its function can be expressed as $\frac{1}{1+e^{-x}}$. If we have $r = 2$ predictor variables and replace x with $\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2}$ we get the logistic regression model:

$$P(Y_i = 1|X_i) = \pi_i = \frac{e^{\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}}}{1 + e^{\beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}}} \tag{3}$$

where β_0, β_1 and β_2 are unknown regression parameters. If X_{ir} increases by one, the odds of $Y_i = 1$ increases by a factor e^{β_r}.

If estimates $\hat{\beta}_r$ of these are computed from the training data set and then plugged into (3), we obtain estimates of the success or disease probabilities

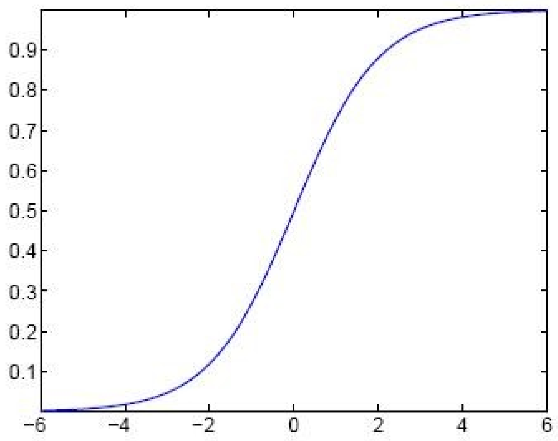

Figure 2: The logistic function with its characteristic S-shaped curve

of all observations, and this gives a score

$$\mathrm{SCM2}_i = \hat{\pi}_i = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_{i1}+\hat{\beta}_2 X_{i2}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X_{i1}+\hat{\beta}_2 X_{i2}}}$$

of mutation i. A logit (or logarithm of odds) transformation

$$\hat{\pi}_i' = log\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \hat{\beta}_2 X_{i2}$$

makes them linearly dependent on the estimated regression parameters. For generalized linear models such a transformation is referred to as a link function, since it links the mean function to a linear combination of the regression parameters.

Assuming that $\{Y_i\}$ are conditionally independent given all explanatory variables $\{X_i\}$, a likelihood function

$$L = L(\beta_0, \beta_1, \beta_2) = \prod_{i=1}^{n} \frac{1}{1+\exp(-\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2})}$$

is obtained. The maximum likelihood estimator (MLE) of $(\beta_0, \beta_1, \beta_2)$ is found as the parameter vector that maximizes the likelihood function. The MLE can be found numerically by applying some iterative algorithm, such as the Newton-Raphson, to the log likelihood function $l = \log(L)$.

For $r = 1, 2$ we can also test the null hypothesis H_0 that the explanatory variable X_r has an affect on the dependent variable Y against the alternative hypothesis H_1 that it has not, i.e.,

$$\begin{aligned} H_0 :&\quad \beta_r = 0, \\ H_1 :&\quad \beta_r \neq 0, \end{aligned}$$

Except for very small samples, we can test H_0 using a z test statistic by dividing the maximum likelihood estimate $\hat{\beta}_r$ by its standard error. Some softwares reports the square of this statistic, called the Wald statistic. Asymptotically for large samples it has a chi-squared distribution with 1 df.

In order to test goodness-of-fit, i.e., the validity of the logistic regression model, we can use the Hosmer-Lemeshow (H-L) test statistic. In order to compute it, we first partition the observations into (for instance) 10 equally sized groups based on their percentile ranks of the fitted risk values and then comparing the observed number O_j of cases in each group j with its expected number E_j, as predicted by the logistic regression model. In more detail, the H-L statistic is defined as

$$G^2_{HL} = \sum_{j=1}^{10} \frac{(O_j - E_j)^2}{E_j(1 - E_j/n_j)}$$

where n_j is the number of observations in the j:th group. Under the null hypothesis that data follow a logistic regression model, G^2_{HL} has a χ^2-distribution with $10 - 2 = 8$ degrees of freedom asymptotically for large samples, see for instance Hosmer and Lemeshow (2000). If a p-value computed from this χ^2-distribution is greater than 0.05, we fail to reject the null hypothesis that there is no difference between observed and model predicted values, implying that the model fits the data at an acceptable level. According to Hosmer, D.W. and Lemeshow, S. (2000).

5.3 Scoring Method 3

In this third scoring method the estimated parameter values, of the logistic regression model in previous Section, could help us classify new cases. In this subsection, we look at a simpler class of linear combinations

$$\mathrm{SMC3}_i = aX_{i1} + bX_{i2},$$

of the NB-score X_{i1} of mutation i computed from the pph2 data set and some other score X_{i2} (we will use the frequency of the mutated allele) computed from the 1kg data set, with a and b as weights or co-factors. Without loss of generality we can normalize the co-factors so that $a + b = 1$. Then, since X_{i1} and X_{i2} are both between 0 and 1, the same is true for SCM3.

We can use training data and ROC curves (see Section 3.2.7), in order to investigate which (a, b) that maximize AUC. Alternatively, we can use the logistic regression model, putting $a = \hat{\beta}_1/(\hat{\beta}_1 + \hat{\beta}_2)$ and $b = \hat{\beta}_2/(\hat{\beta}_1 + \hat{\beta}_2)$.

6 Results

In this Chapter the results of the three scoring methods, described in the previous Section, will be presented and assessed, respectively.

6.1 Scoring Method 1

We will use the simplifed version (2) of SCM1, based only on pph2 data. For each mutation in the pph2 patient dataset (which consists of both neutral and damaging mutations) we extracted a probability (that is, a NB score). 536 observations were included in the patient dataset in the range of 0 to 1 for which we calculated the empirical p-values. Another set of test data was used to predict the empirical distribution of $\hat{F}$. Because of the null hypothesis that the mutation has no effect, this dataset was filtered to only consist of 6608 neutral mutations from the HumDiv dataset. For example, 5993 observations in the test dataset of known mutations is less than or equal to an observed value of 0.02 in the patient dataset. Thus, $\hat{F}(0.02) = 5993/6608 = 0.907$, and the p-value becomes 1-0.9069=0.0931. An empirical p-value < 0.05 was regarded as statistically significant. If we take the logarithm of this p-value we get the corrected p-value, presented in the previous Chapter, -log(p)=-log(0.093)=1.031. This was done for all of the 536 variations.

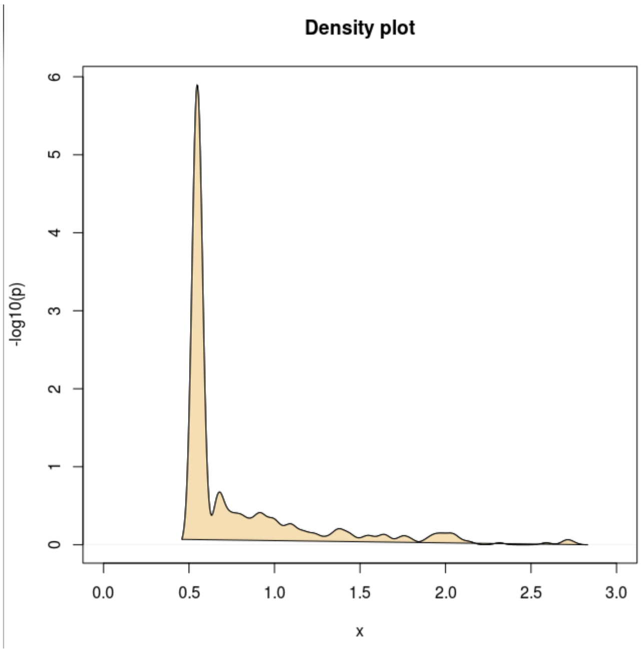

Figure 3: Kernel density estimate of SCM1 for a patient data set containing 121 mutations, neutral and deleterious (mean=0.9240, sd=0.1129, bandwidth=0.03).

In Figure 3 we plot the empirical p-values of the patient dataset on a logarithmic scale.

The graph is plotted in R using Kernel estimates that produce a smoothed estimate of the probability density function. To get the best optimal graph with Kernel a free parameter, bandwidth, can be used. This parameter has a strong influence on the resulting estimate. We choose a Kernel function $k \geq 0$ satisfying $\int_{-\infty}^{\infty} k(x)dx = 1$, concentrated around 0. Each observation x_i is then replaced by a copy of the function k shifted so that it is centred at x_i, and scaled by the bandwidth.

6.2 Scoring Method 2

In our logistic regression model with covariate variables from pph2 and 1kg (see Section 5.2), we first estimate the regression parameters with a training dataset, that can be seen in Table 4, consisting of 463 known mutations extracted from pph2 and 1kg, respectively. The larger the first covariate X_{i1} (i.e. the NB score from pph2) is, the larger is the chance for the mutation to be damaging. For 1kg we first take X_{i2} as one minus the allele frequency, since, with this choice of covariate, we would expect that a larger X_{i2} increases the risk of a damaging mutation. The predicted coefficients of the logistic regression can be seen in Table 5 giving the following fitted regression model:

Table 4: The first few and last few rows of pph2 probabilities and one minus the 1kg allele frequencies in the training dataset. A high pph2 probability indicates a damaging mutation and a low probability a benign. Whereas a low allele frequency indicates a damaging mutation and a high frequency a nondamaging. Since, 1 minus the allele frequency is reported, a higher value corresponds to a mutation more likely to be damaging.

mutation damaging=1 nondamaging=0	NB score	1 − the allele frequency
1	0.890	0.9995
1	0.996	0.9995
1	0.966	0.9986
0	0.002	0.3292
...	...	...
1	1.000	1.000
1	1.000	1.000
1	1.000	1.000

$$\text{SCM2}_i = \hat{P}(Y_i = 1|X_i) = \hat{\pi}_i = \frac{e^{-2.54+7.50X_{i1}-1.05X_{i2}}}{1 + e^{-2.54+7.50X_{i1}-1.05X_{i2}}}$$

Notice that $\hat{\beta}_1$ is positive, as expected. On the other hand, contrary to our expectation $\hat{\beta}_2$ is negative, indicating that a lower frequency of the mutated allele decreases the risk of the mutation being damaging.

A statistic for significance (Wald) was produced for each predictor. The larger the better. Each Wald value is associated to a p-value (the lower the better). Even if the entire model is significant, it does not mean that all

Table 5: Fit of Logistic Regression Model for the training data, with covariates one minus the population allele frequency for 1kg and probability of a damaging mutation (NB score) for pph2.

	$\hat{\beta}_i$	S.E.	Wald	df	p-value	$e^{\hat{\beta}_i}$
constant	-2.54	1.82	1.90	1	0.16	0.08
NB score	7.50	0.61	150.50	1	0.00	1808.04
$1 -$ the allele frequency	-1.05	1.87	0.32	1	0.57	0.35

the predictors are significant (in that case, we could drop a nonsignificant predictor, or enter it in some modified form if justified, e.g. its square or logarithm).

If the probability of the column "p-value" is less than 0.05 we would reject the hypothesis at the 5% level that the parameter is zero. For instance, since the p-value for testing $\beta_2 = 0$ is 0.57, we cannot exclude (and still believe) that a decrease in allele frequency of the mutated variant increases the risk that the mutation is damaging, although the effect is very small. If we had more parameters we could try removing each of these, one at a time, to see the effect on our correct classification rate.

A 1-unit increase of the 1kg coefficient X_{i2} decreases this risk, as quantified by the estimated odds ratio $e^{-1.05} = 0.35$. The estimated odds ratio for a 1-unit increase of the pph2 coefficient X_{i1} is much higher, $e^{7.50} = 1808.04$. These are the most extreme cases, since our logistic regression coefficients have to be in the interval [0,1]. If we would increase pph2 with for instance 0.1 we would get a much lower odds ratio.

In order to test goodness-of-fit, we evaluated our results with the Hosmer-Lemeshow test, that compares the predicted and observed probabilities for each decile of probabilities under the linear model. Our Hesmer-Lemeshow statistic has a p-value of 0.19, meaning it is not statistically significant and we can say that our test dataset does fit the model well.

In order to quantify the ability of SCM2 to distinguish benign and damaging mutations, we evaluated the receiver operating characteristic (ROC) curve (see Section 3.2.7). The AUC under the ROC ranges from 0.5 and 1.0 with larger values indicating a better separation between benign and damaging mutations. Figure 4 shows the output of a ROC curve based on the predicted probabilities in the logistic regression model for the training dataset mentioned above. The area under the curve is 0.965. The AUC is significantly different from 0.5, with a p-value of 0.000, meaning that the logistic regression classifies the two groups of mutations significantly better than by chance.

We investigated a second logistic regression model by redefining X_{i2} so that it equals 1 if the allele frequency of the mutated variant is zero, and 0 if the

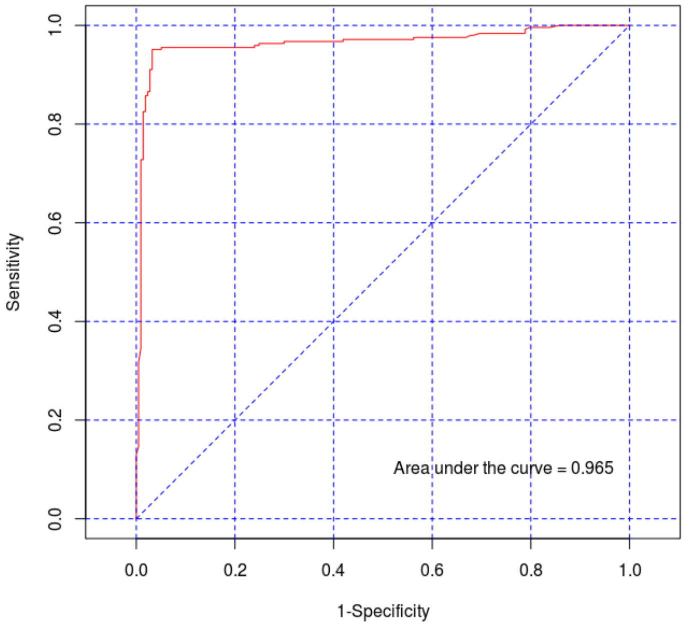

Figure 4: ROC curve of the predicted probabilities in the logistic regression model with 1kg covariate one minus the allele frequency for a data set of 463 NB scores and allele frequencies, respectively. The red curve corresponds to the discrimination analysis between damaging and benign mutations, and the blue line represents ”the line of no-discrimination diagonal” dividing the ROC space. Points above the diagonal represent good discrimination (better than random), points below the line poor discrimination (worse than random). The two groups are almost totally separated, since the estimated AUC = 0.965 is close to 1.

allele frequency is greater than zero. In Table 6 the predicted coefficients can be seen, and our new logistic regression model becomes:

$$\hat{P}(Y_i = 1|X_i) = \hat{\pi}_i = \frac{e^{-3.56+7.48X_{i1}+0.58X_{i2}}}{1 + e^{-3.56+7.48X_{i1}+0.58X_{i2}}}$$

The appearance of this new model has some slight differences compared to the first method. The NB score coefficient is still positive, but the allele frequency is positive instead of negative. Also, the H-L test gave a p-value of 0.002 and hence the logistic regression model is rejected.

Table 6: Fit of Logistic Regression Model for the training data, using a categorical covariate for 1kg and the NB score as covariate for pph2.

	$\hat{\beta}_i$	S.E.	Wald	df	p-value	$e^{\hat{\beta}_i}$
constant	-3.56	0.35	101.80	1	0.000	0.03
NB score	7.48	0.61	151.60	1	0.000	1772.24
the allele frequency (categorical)	0.58	1.35	0.18	1	0.67	1.79

In Figure 5, on the next page, we can see that the AUC, for this second method, is 0.964. A value also very close to 1, meaning that the classifier scores every positive higher than every negative. We can say that this logistic regression model discriminates as well as the first model, meaning that SCM2 even for this choice of 1kg covariate provides an adequate discrimination between deleterious and benign mutations in the dataset of pph2 and 1kg.

For comparison, we also performed fits of data to a logistic regression model with only one covariate, either NB-scores (Table 7), 1-the allele frequency from 1kg (Table 8), or only the categorical allele frequency from 1kg (Table 9).

As can be seen from Table 9 below, the estimated regression coefficent of the categorical 1kg variable is positive as before (cf. Table 6). On the other hand, the estimated regression coefficient for the model with a 1−the allele frequency covariate is now positive (in line with what we would expect), whereas it was negative in Table 5, when the NB score covariate was also present. Moreover, the AUC of the model with a single NB score covariate, in Table 7, is similar to Figures 4 and 5, whereas the AUC of the models in Tables 8 and 9 are much smaller, 0.515 and 0.504 respectively. This means that the pph2 covariate predicts the mutation class much better than the allele frequency of the 1kg data.

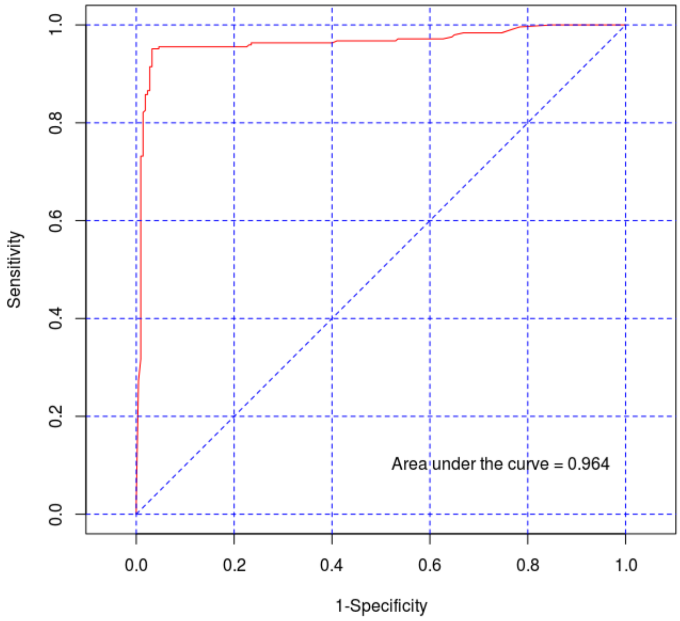

Figure 5: ROC curve of the predicted probabilities in the logistic regression with the NB score as covariate from pph2 and a categorical covariate from 1kg which is 1 if the allele frequency is 0 and 0 otherwise. The estimated AUC = 0.964.

Table 7: Fit of Logistic Regression Model for the training data, with only the covariate NB score from pph2.

	$\hat{\beta}_i$	S.E.	Wald	df	p-value	$e^{\hat{\beta}_i}$
constant	-3.55	0.35	102.3	1	0.00	0.03
NB score	7.48	0.61	151.5	1	0.00	1772.24

Table 8: Fit of Logistic Regression Model for the training data, with only the covariate one minus the population allele frequencies from 1kg.

	$\hat{\beta}_i$	S.E.	Wald	df	p-value	$e^{\hat{\beta}_i}$
constant	0.072	0.83	0.008	1	0.93	1.075
$1 -$ the allele frequency	0.055	0.85	0.004	1	0.95	1.057

Table 9: Fit of Logistic Regression Model for the training data, with only the categorical covariate allele frequency from 1kg.

	$\hat{\beta}_i$	S.E.	Wald	df	p-value	$e^{\hat{\beta}_i}$
constant	0.12	0.09	1.5	1	0.22	1.13
the allele frequency (categorical)	0.29	0.54	0.29	1	0.59	1.34

6.3 Scoring Method 3

The third scoring method is the simplest, a linear combination of the pph2 and 1kg scores with weights a and b. We can look at the logistic regression not only as a tool for predicting the probability that a mutation is damaging in SCM2, but also as a method for generating good weights a and b in SCM3 of our factors pph2 and 1kg. As mentioned in Section 5.3, we can achieve this by choosing a and b proportional to the estimated regression parameters $\hat{\beta}_1$ and $\hat{\beta}_2$. We can use the same training dataset from HumDiv as for SCM2, containing damaging as well as benign mutations, in order to calculate the ROC curve. Given that the weights are taken from the logistic regression we get the same plot as in Figure 4 or Figure 5, depending on which covariates we choose. SCM3$_i$ is just a monotone transformation of SCM2$_i$ for all mutations i with this choice of weights a and b.

7 Discussion and Project Outline

The still increasing amount of genetic variation data requires computational tools for prediction of the impact of disease-associated variants and to possibly alter the most interesting and likely pathogenic cases for experimental analysis. The aim of this paper was to show a novel method to evaluate how reliably the pathogenicity of missense mutants can be predicted.

To this end, we created scoring models with two factors containing data from PolyPhen-2 (pph2) and the 1000 Genomes Project (1kg) as a first step towards creating an in-silico multifactorial tool for estimating the relevance of a mutation to a certain disease. The preliminary idea was to create a basic model scoring a mutation as damaging or nondamaging, and to start with, only taking two factors into consideration, pph2 and 1kg. We focused on one disease, Mendelian susceptibility to mycobacterial disease (MSMD) for one patient.

For each polymorphism found we first computed the pph2 Naive Bayes probabilities, predicting how harmful the polymorphism is. Then, we computed allele frequencies of controls in 1kg, taking into consideration if a mutation can't be found in 1kg then it is rare and more likely to be harmful. If we find any frequency (high or low) in 1kg then that mutation is most likely nondamaging, at least for Mendelian diseases. If a mutation is not found in pph2 then it is either not a missense mutation or if the lack of data did not allow to make a prediction, then the status of the mutation is unknown.

The purpose was to assign a score to each mutation by combining the information we have on the population allele frequency of the mutation (from 1kg) together with the probability that the mutation is harmful (NB scores from pph2). We therefore proposed various ways of putting these two pieces together.

With SCM1 we get a prediction for each mutation in form of a p-value. Since we lack allele frequency data for patients, we didn't use the 1kg data for this method but computed the simplified formula in (2) based only on the NB scores.

In SCM2 we performed a logistic regression predicting the probability that the mutation is damaging, with covariates from pph2 (NB score) and 1kg. We used two options for the 1kg covariate. Either a continuous covariate, for which one minus the allele frequency gives a gradual indication of the mutation's harmfulness. Alternatively, a categorical covariate, in which case the mere presence of the mutated variant in 1kg indicates a benign mutation. In any case, it is necessary with at least one mutated variant among some of the controls in 1kg. We also found that the logistic regression model fitted the test data set well with the Hosmer-Lemeshow test, that compares the predicted and observed probabilities for each decile of probabilities.

With SCM3 we used a simpler linear combination of the two scores from pph2 and 1kg.

For all three scoring methods, the area under the ROC curve (AUC) was used in order to quantify how well the mutation status could be predicted. We got values of AUC very close to 1, indicating that for the test data sets we considered, the screening measure reliably distinguishes between deleterious and benign mutations.

As a conclusion, it is important to take the pph2 data into account, since the variants we study are rare. Because of the rareness conventional association studies are then much less powerful, since a huge number of individuals are needed to estimate allele frequencies. When comparing the exome data with the reference ("healthy") genome we identify for each patient many thousands of genetic variations, and the main challenge is to narrow down the list to a few candidate genes where these mutations occur that would be further investigated in the laboratory, in order to validate at least one of them as disease-causing.

The next step would be to put the project on a larger scale and generalize it to a more complicated statistical model that includes more types of data and biological knowledge, in order to estimate how likely it is that the mutation is disease-causing. We have only considered one type of DNA-evolution, nucleotide substitution, but the evolutionary process involves several other factors.

For example, other important factors to take into consideration, for scoring on the single individual level, could be selection value (higher score for highly conserved), mutation type (for example in the following order missense, nonsynonymous, synonymous), mutation location (coding, noncoding). In a broader perspective we could observe the copy number variation (CNV) checking if the polymorphism in question has more or less copies for patients compared to healthy individuals, take a systems biology approach and check whether the mutated gene belongs to some pathway of the disease, integrating the scoring with other data. For example, if we have a mutated gene and we in microarray also confirm that it has a significantly lower expression for patients compared to controls, then it is much more likely to be disease-causing. We could also check if the gene in question is close to any gene known to be in a disease pathway, for example by using Connectome.

We could also extend the scoring method by adding more features of the mutation. For instance, if we had an index of the severity of the harmful effects of each mutation, we may compute a score combining not only the relative frequency of the mutation and the probability of it being harmful, but also the severity of the expected harm. This would be an enrichment of our score, for which severity is not included. For instance, two mutations may have the same frequency and the same probability of being harmful, but the harm is lethal in one case and only aesthetic in the other, for instance it may cause some extra skin blotches. Though, any combination of several features into one single score or index would imply a loss of information.

Also, we would want to give more weight to 1kg than pph2. This is because sometimes a mutation is misclassified in pph2 as deleterious when it is in fact was neutral (or the opposite), but if a variation is found in 1kg, no matter how high or low the frequency is, it exists in the population and is therefore most likely nondamaging.

The scores we have considered refer to the mutations and their consequences, and not to the causal factors determining them to appear. For instance, a particular mutation can either be caused by exposure to sunlight or to some chemical in food or water. The degree of exposure to a causal factor is something conceptually different from the nature of the mutation and its consequences, and these aspects could also be taken into account in a statistical analysis.

We simply want to add as much information as possible to end up with a scheme where name of gene, biological background of that gene (e.g., is it known in any disease pathway(s)?) and other valuable available information is taken into consideration. Then of course we could extend the model statistically in order to incorporate biological background information and handle more types of data, and computationally, for instance more extensive simulations based on empirical data. By looking at the statistical significance we want to be able to draw conclusions about the biological significance.

References

Adzhubei et al. (2010). *A method and server for predicting damaging missense mutations*, Nature Methods, vol.7 no.4

Adzhubei et al. (2010). *Supplementary Methods*, Nature Methods, vol.7. no.4

Fayyad U. and Irani K. (1993). *Multi-interval discretization of continuous-valued attributes for classification learning*. Proc 13th International Joint Conference on Artificial Intelligence. Vol.2, San Mateo, CA: Morgan Kaufmann. p.1022-1027.

Fayyad U. and Irani K. (1992). *Technical Note, On the Handling of Continuous-Valued Attributes in Decision Tree Generation*, Kluwer Academic Publishers, Boston. Machine Learning, 8, p.87-102.

Hosmer, D.W. and Lemeshow, S. (2000). *Applied logistic regression*, Wiley, New York

Sunyaev, R. et al. (1999). *PSIC: profile extraction from sequence alignments with position-specific counts of independent observations*, Protein Engineering, vol.12 no.5 p.387–394

The PolyPhen-2 web site, `http://genetics.bwh.harvard.edu/pph2/dokuwiki/start`. (2012). /Downloads, Datasets *Whole human exome sequence space annotations*, HumDiv and HumVar training sets

The 1000 Genomes Project web site, `http://www.1000genomes.org/about` (2012a)

The WEKA web site, *http://www.cs.waikato.ac.nz/ml/weka/*

Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge CB10 1SA, UK. (2010). *A map of human genome variation from population-scale sequencing*, Nature Methods, vol.467 1061–1073

Witten H., Frank, and Hall A. (2011). *Data Mining and Practical Machine Learning Tools and Techniques*, 3rd ed. Burlington, USA, p.90-94, p.99-104, p.148-154

`http://www.quincetree.com/family-tree-dna-explained/`

recloh-important-genetic-genealogy (2012)

http://en.wikipedia.org/wiki/Genetic_code (2012a)

http://en.wikipedia.org/wiki/Single-nucleotide_polymorphism (2012b)

http://en.wikipedia.org/wiki/Point_mutation (2012c)

http://en.wikipedia.org/wiki/Hardy-Weinberg_principle (2012d)

http://en.wikipedia.org/wiki/Naive_Bayes_classifier (2012e)