Computational Methods for Annotation Analysis of Genetic Variations

Tibor Fülöp

Master's thesis

ITÄ-SUOMEN YLIOPISTO

School of Computing

Computer Science

May 2015

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Kuopio
School of Computing
Computer Science

Fülöp, Tibor: Computational Methods for Annotation Analysis of Genetic Variations
Master's Thesis, 102 p., 1 appendix (3 p.)
Supervisors of the Master's Thesis: Professor Matti Nykänen and PhD Jussi Paananen
May 2015

Abstract: Analysis and interpretation of DNA variations is very crucial for research trying to solve genetic background of heritability, diseases and other traits. Because of the massive amounts of data produced by modern genetics research technologies, computational methods are required to solve these challenges. This thesis briefly introduces the field of molecular biology and basic principles of genetics, discusses genetic variation annotation methods, genome-wide association studies (GWAS) and enrichment analysis methods with their implementation algorithms. As a part of this thesis, we introduce a novel web tool called Varanto that can be used to annotate, visualize and analyse genetic variations. The thesis describes the preparation of Varanto's background database, including descriptions of extracting and importing data from public genome databases. Varanto can be used to perform hypergeometric test based annotation enrichment analysis for a set of genetic variations, and to visualize the annotation results by heatmaps and hierarchical clustering. Varanto includes a web based user interface developed using Shiny web application framework for R. Varanto's performance and functionality is tested and showcased by performance benchmarks and by analysing and interpreting data from previously published genome-wide association studies, including examples from GWAS on body mass index (BMI) and GWAS on Chrohn's disease.

Keywords: bioinformatics, genetics, enrichment analysis, hypergeometric test, annotation methods

CR Categories (ACM Computing Classification System, 1998 version):
J.3 Biology and genetics
G.3 Statistical computing

Foreword

This thesis was done at the School of Computing, University of Eastern Finland during the academic year 2014-2015.

I would like to express my gratitude to my supervisor Dr. Jussi Paananen for the useful advices, comments, providing literature and leading me and also to my supervisor Ing. Tomáš Martinek PhD. for consultations and helpful literature recommendations. I would like to thank to all people who have supported me all the time I was working on this thesis.

List of abbreviations

A	Adenine
BH	Benjamini-Hochberg (method)
BMI	Body mass index
C	Cytosine
CDCV	Common disease-common variant hypothesis
DNA	Deoxyribonucleic acid
eQTL	Expression quantitative trait loci
FDR	False discovery rate
G	Guanine
GO	Gene ontology
GSEA	Gene Set Enrichment Analysis
GUI	Graphical user interface
GWAS	Genome-wide association studies
HGMD	Human Gene Mutation database
HGNC	HUGO Gene Nomenclature Committee
HMM	Hidden Markov Model
IL23	Interleukin 23 (cytokine)
Indels	Single-base insertions and deletions
K-S	Kolmogorov-Smirnov (test)
LD	Linkage disequilibrium
miRNA	Micro RNA
mRNA	Messenger RNA
MSigDB	Molecular Signature Database
ncRNA	Non-coding RNA
OMIM	Online Mendelian Inheritance in Man
OR	Odds ratio
PWM	Position weight matrices
RNA	Ribonucleic acid
rRNA	Ribosomal RNA
SNP	Single nucleotide polymorphism
SO	Sequence ontology
T	Thymine
TF	Transcription factors
tRNA	Transfer RNA
U	Uracil
UI	User interface
UTR	Untranslated regions

Contents

Appendices

1 Introduction

Importance of molecular biology has been increasing in the last decades. It is now known that all organisms contain huge amount of encoded genetic information in their cells and that genetics is a key element for understanding why all of us differ from each other. Characteristics of an organism, known as *phenotype*, are dependent not only on environment but also on unique heritable genetic information carried by organism, its *genotype*. Genetic information of an organism is stored in the molecules of *deoxyribonucleic acid* (*DNA*) which structure was identified in 1953 by James Watson and Francis Crick who were subsequently awarded the Nobel Prize for this discovery. DNA molecules comprise of large number of individual units, called *nucleotides*, organized in long but microscopically thin strands. Sequences of these nucleotides include genetic elements, genes being the most important ones. Every genetic element potentially contains specific information related to development of an organism or to fundamental functions which ensures the life of the organism. Because of all the time growing amount of identified genetic information it is difficult or even impossible to effectively use this information for research purposes without using computational tools. To facilitate effective processing of biological data, new bioinformatics methods and common annotation terms were created to help reference specific biologic or molecular properties, processes, functions, consequences or any other biologically related information related to biology. These annotations are also easier to store in databases and easier to process by computational methods.

DNA is not the same across the species and also across the individuals within species. Variations in DNA result in organisms having distinct genotypes. For this reason these variations are subject to research and studies which try to identify their effects on phenotypes. This knowledge can help in understanding biological processes and differences between organisms, including medical purposes. There are research efforts that try to identify genetic causes of human traits and diseases to understand their biological background in more detail or try to predict the probability of developing the trait in question. Other areas of research include studying the dependencies between a genotype and response on using a specific drug. Besides traditional genetics research which utilized only a small fraction of genetic information, for example a single gene, new

methods for obtaining the whole genetic information of an organisms and for analysing this information computational resources were designed. *Genome-wide association studies (GWAS)* are aimed on finding causal variations in the genetic information for specific traits on a genome-wide level across large number of genotyped individuals. The result of single GWAS is a list of variations and their association to a trait. For these variations finding enriched annotations is important for determining the related biological processes or causes for the trait. This can be achieved by analysing annotations for their over- and underrepresentation. For this purpose several enrichment analysis methods have been developed to determine the statistical significance of annotations enrichment. Many previously developed tools for performing this task and implementing common methods, are described in this thesis. Most of these tools are developed for enrichment analysis of genes instead of genetic variations. This work introduces a novel tool called *Varanto* that can be used for annotation analysis of genetic variations. Varanto uses an input consisting of a set of variations which can, for example, result from a GWAS. Varanto works with large background database of annotations and provides easy-to-use web interface. In addition, Varanto provides visualization of associations between variations and their annotations. Web interface allows users to set parameters such as background set, distance filter and selection of annotation classes.

The second chapter covers basic information about molecular biology and genetics. Genetics research, specially annotation methods and GWAS, are discussed in the third chapter. Enrichment analysis methods and computational algorithms implementing them are described in the fourth chapter. In the fifth chapter, Varanto tool is introduced including description of methods for preparing its background database and implementation of its web interface. Finally, in the sixth chapter, the results of performance tests are shown and use-cases of *Varanto* demonstrated on Body Mass Index trait, Crohn's disease and detecting technical bias.

2 MOLECULAR BIOLOGY

Species in the world differ from each other, and furthermore even the individuals of the same species have different traits from each other. Development of every living entity is encoded in the genetic material located in cells. Characteristics are inherited from a parent to an offspring. *Genetics* is a field that studies biologically inherited traits. Not all traits are inherited purely biologically but many of them are influenced also by the environmental or cultural factors and genetics recognizes this fact.

Information in this chapter is mostly adapted from Daniel Hartl and Elizabeth Jones (Hartl & Jones 2001).

2.1 Mechanisms of storing of information in DNA sequences

DNA was discovered by Friedrich Miescher in 1869 but at the time it was impossible to determine its biological function. Later in the beginning of 20th century it was believed that proteins are the genetic material. In 1944 Oswald Avery, Colin MacLeod, and Maclyn McCarty performed experiments determining the transforming activity in S cells of *Streptococcus pneumonia* in the presence of R cells, displaying the genetic function of DNA. In 1953 James Watson and Francis Crick examined the three-dimensional structure of DNA molecule. They found out that DNA has double-stranded helix form. Each strand consists of *nucleotides*. Each nucleotide contains *bases* connected to phosphorylated molecule of the 5-carbon sugar *deoxyribose*. There are four types of bases: *Adenine (A), Thymine (T), Guanine (G)* and *Cytosine (C),* the letters of DNA. Nucleotides have pair of another nucleotide located on the opposite strand. Paired nucleotides are *complementary* to each other. A is matching with T and G is matching with C. The sequence of bases is variable and each base can code for some information meaning that DNA can code for a huge amount of information. DNA strand has a polarity resulting from asymmetric structure of nucleotides. One end of DNA strand is marked as 5’ and another end as 3’. The strands have opposite polarity to each other. The complementarity of bases enables replication of DNA molecules. During the replication the strands are separated and then each strand serves as a template to synthesis of a new strand (Figure 2-1).

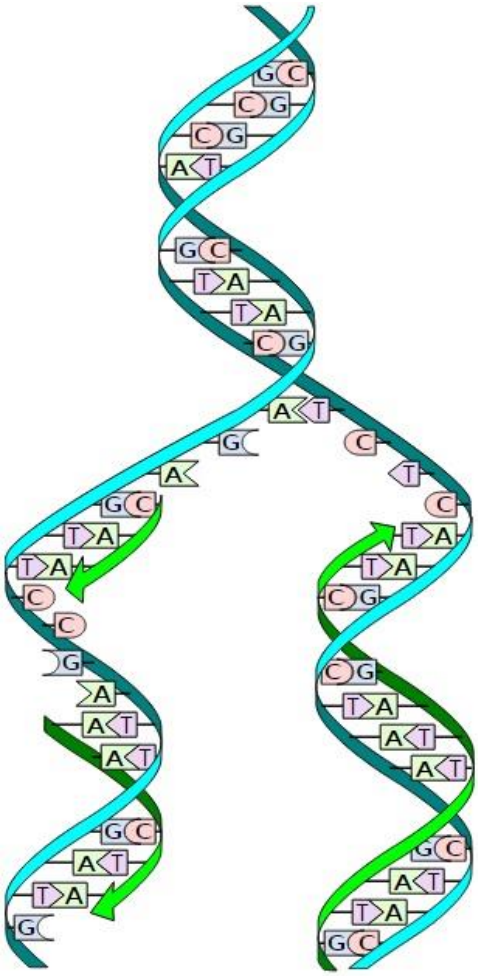


Figure 2-1. DNA replication. (I, Madprime [CC0, GFDL, CC-BY-SA-3.0 or CC BY-SA 2.5-2.0-1.0], via Wikimedia Commons)

DNA molecules in cells are commonly organized in *chromosomes*. Organisms have various numbers of chromosomes. Prokaryotes have usually a single chromosome. Eukaryotes have more chromosome pairs and also chloroplasts and mitochondria can contain additional DNA. There is often a distinction between the chromosomal structure of females and males. Humans have 23 chromosome pairs numbered from 1 to 22 with additional sex chromosome pair XY in males and XX pair in females. The paired chromosomes, except the sex chromosome pair, are complementary. Individual chromosomes of the pair are inherited from one of the parents. The complete set of DNA information is called as a *genome*. *Karyotype* is a visual presentation of chromosomes arranged in pairs. Human genome contains about 3 billion of base pairs in one copy of the 23 chromosomes.

2.2 Genes and genetics

In 1866 Gregor Mendel demonstrated the existence of *genes* and their role in transmission of information from generation to generation, becoming the father of genetics. It was before the recognition of DNA as genetic material and even before the discovery of DNA itself. Therefore he introduced his theory of inheritance in terms of abstract rules. Nowadays it is possible to obtain DNA sequence of an organism by technologies that are increasingly cheaper and more effective. This enables studying genes on molecular level and makes the field of genetics dynamic and fast developing.

DNA sequences are sequences of encoded information which can be expressed. The *central dogma* is a theory describing the process of transferring this information. The information transfer is indirect as DNA acts through an intermediary molecule of *ribonucleic acid (RNA).* The result of expression are *proteins,* many of which are enzymes, or functional *non-coding RNA* molecules. Sequences, which proteins are produced from by process of transcription and translation (see below), are called *coding genes*.

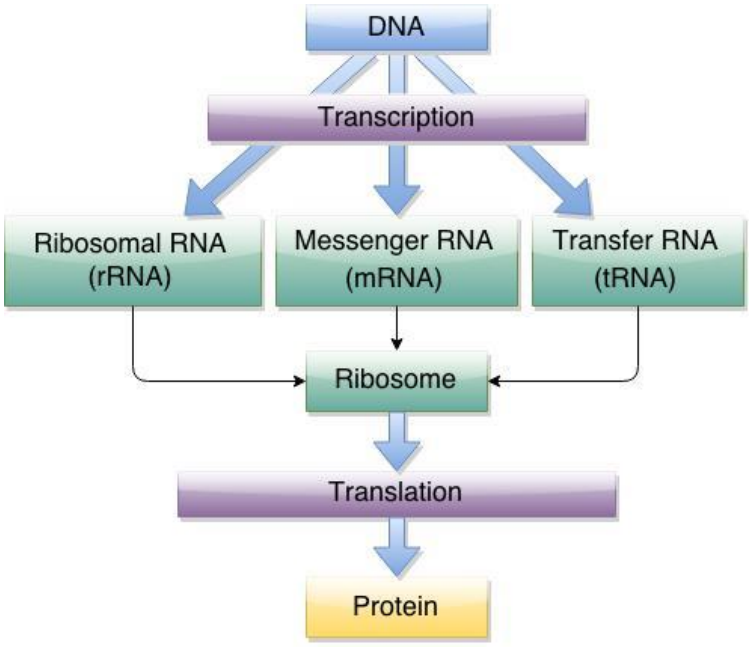


Figure 2-2. The "Central dogma" of molecular genetics.

As seen from Figure 2-2, *transcription* is a process of synthetizing RNA from a DNA strand. *Translation*, performed in *ribosomes*, is a process of synthetizing polypeptide chains from RNA. Ribosomes consist of *ribosomal RNA (rRNA)* and proteins. *Transfer*

RNA (tRNA) participates in the translation. *Messenger RNA (mRNA)* is an actual carrier of genetic information.

An important consequence of the central dogma is that this information transfer is one-directional.

2.2.1 Transcription

Before the transcription, *RNA polymerase* binds to a *promoter* site in DNA. Then the transcription starts at *transcription start site*, which is inside or near to the promoter. RNA polymerase gains access to either strand. It synthetizes RNA chain by adding new nucleotides to its 3' end. Nucleotides in RNA with thymine (T) base are replaced by *uracil (U)* base. Transcription is terminated by reaching a chain-termination sequence. The produced RNA molecule is *primary transcript*. In eukaryotes, it is processed before it becomes mRNA. This processing contains excision of untranslated sequences in 5' and 3' ends in the *untranslated regions (UTR)* and excision of *introns*. Introns are embedded between the coding sequences – *exons*. Mechanism of excision of introns is called *RNA splicing*. The patterns of splicing varies in different cell types. This feature is called *alternative splicing* (Figure 2-3).

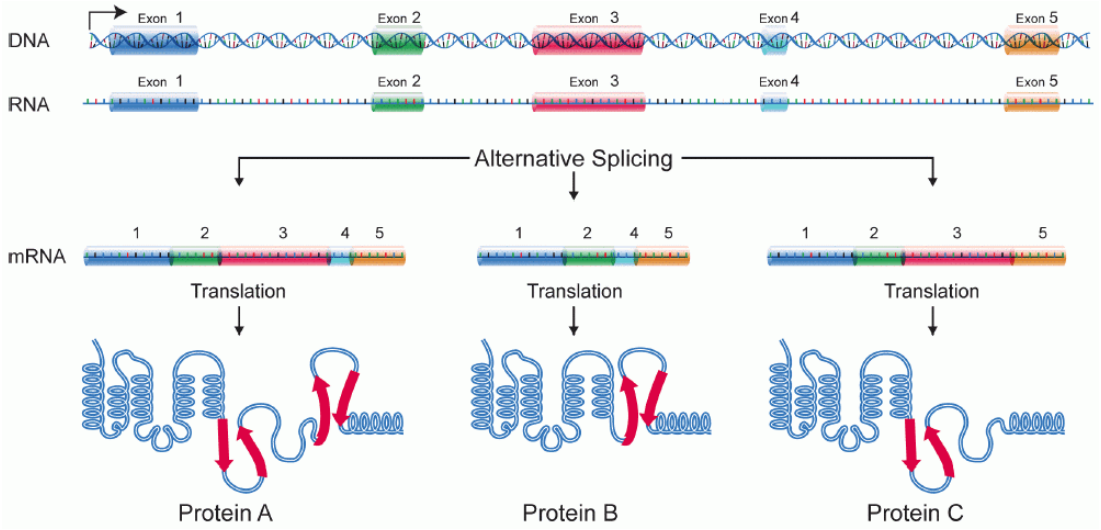


Figure 2-3. Alternative splicing. (By National Human Genome Research Institute [Public domain], via Wikimedia Commons)

Near the gene, there can be *enhancer* sequences that increase the rate of transcription when certain proteins, *transcription factors (TF)*, bind to them. S*ilencers*, on the other

hand, can bind proteins that result in protein complex blocking or slowing down the transcription.

2.2.2 Translation

An important member of translation is tRNA. Each tRNA carries a particular *amino acid*. A polypeptide chains is created by synthetizing amino acids. Three nucleotides whose bases determines the carried amino acid according to the *genetic code* (Figure 2-4) are contained in tRNA. The group of three nucleotides is called *codon*. Each codon corresponds to one of the 20 amino acids. Single amino acid can be encoded by one or more codons.

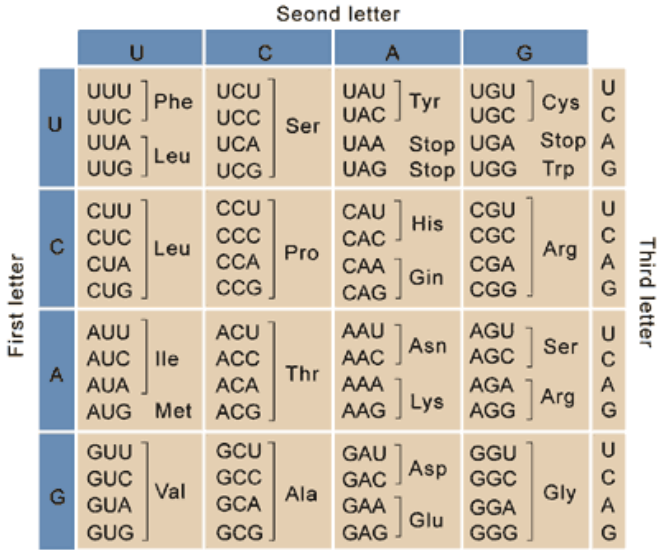


Figure 2-4. The genetic code. (By NIH [Public domain], via Wikimedia Commons)

tRNA molecules bind to codons on mRNA in direction from the 5' end to 3' end. After each connection amino acid carried by tRNA molecule is attached to growing end of the polypeptide chain (Figure 2-5). The codon AUG is "start" codon and tRNA carrying UAC codon (complementary to AUG) is the initiation tRNA. The codons UAA, UAG and UGA are "stop" codons which terminate translation and release completed polypeptide from the ribosome.

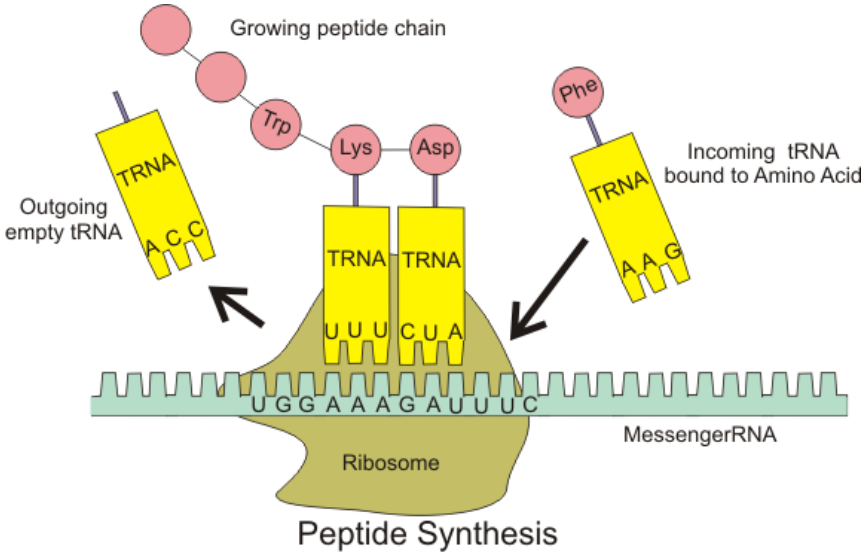


Figure 2-5. Mechanism of translation. (By Boumphreyfr (Own work) [CC BY-SA 3.0 or GFDL], via Wikimedia Commons)

2.2.3 Sequence motifs

Some sequences in various sites or between species are similar to each other and can be described by a consensus pattern. These patterns are called *sequence motifs* and describe patterns of sequences related to transcription, regulation or binding sites for proteins. One well known pattern is for example *TATA box* which is a binding site located 10 bases before transcription start site and has consensus sequence `TATAAT`. The patterns are defined by *position weight matrices (PWM)* (Stormo et al. 1982) that define probability or frequencies of nucleotides on each position of motif. For particular sequence, there are algorithms which compute alignment score to motif or which search motifs in long nucleotide sequences (Weirauch et al. 2013). The motifs can be visualized by *sequence logos* (Figure 2-6).

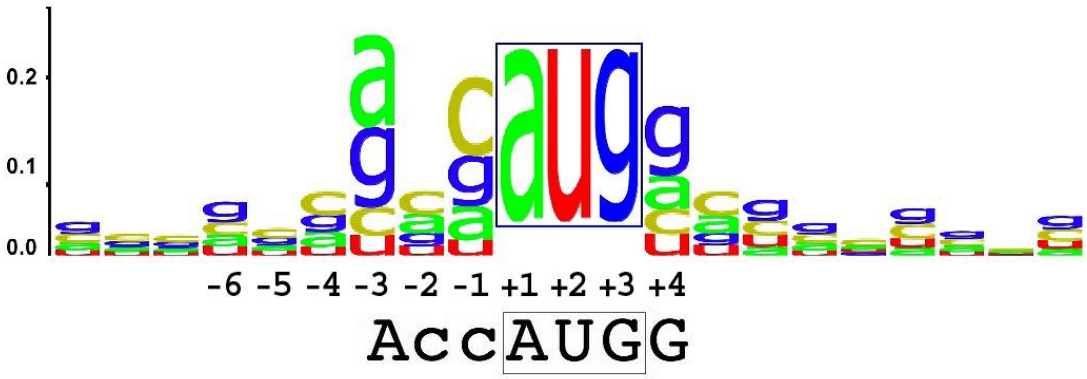


Figure 2-6. Sequence logo of bases around the initiation codon. (TransControl at the English language Wikipedia [GFDL or CC-BY-SA-3.0], from Wikimedia Commons)

2.2.4 Regulation of gene expression

Expression of genes is regulated by several factors. Firstly, the presence of alternative splicing mechanisms or enhancer and silencer sites which works as binding sites for proteins suggests that transcription activity is regulated differently in different cell tissues. Secondly, chromosomes are organised in complexes known as *chromatin* consisting of DNA, RNA, *histones* and other proteins. Histones' ability to bind to DNA also enables them to regulate gene expression (Rattray & Müller 2012). Chromatin can be affected by *chromatin-remodeling complexes*, which are multiprotein complexes restructuring chromatin. The gene expression depends also on the chromatin structure (Ralston & Brown 2008). Thirdly, the next way of regulation is through an *epigenetic* process known as *DNA methylation*, where high methylation decreases genes transcription ability. Finally, there are also non-coding RNAs, for example *Micro RNA (miRNA)*, which works in posttranscriptional regulation of the genes (Bartel 2004), or long *non-coding RNA (ncRNA)*, which are regulators of mRNA transcription (Goodrich & Kugel 2006).

With knowledge about regulation mechanisms it is possible to identify regulatory sites in DNA that affect gene expression. These sites are called *expression quantitative trait loci (eQTL)* (Rockman & Kruglyak 2006).

2.2.5 Homologous sequences

During evolution DNA retains its information content while transferred from one generation to another. It is also subject to changes which then cause DNA diversity across

species and individuals. Many methods exist for recognizing origin of species by comparing their sequences and creating *phylogenetic trees*. The genes, or certain sequences in general, are also suspect to evolution between generations. If two sequences originate from the same ancestry sequence, they are *homologous* to each other.

2.3 DNA variations

Most of the DNA in all organisms within a species is the same. In the human genome with the length of about 3 billion bases, there are around 65 million known sequence variations, most of them being short variations (Ensembl 2015). This smaller part, where there are differences among individuals, is important for genetics as genetic variation is linked to phenotype variation. These differences include short nucleotide variations with length up to few nucleotides, structural variations and chromosome abnormalities.

2.3.1 Short nucleotide variations

Short nucleotide variations are sites of length of one or several nucleotides. The simplest variation is change from one base to another. Another type of variations are single-base insertions and deletions ("indels"). These variations are commonly termed as *Single nucleotide polymorphisms (SNPs)* (Greg & Muse 2004). Two-nucleotide changes and several-nucleotide indels are by some authors considered as SNPs as well. Variations result in alternative forms of sequence called *alleles*. The group of associated alleles is a *haplotype*.

SNPs arise by event called *mutation*. When a mutation spreads within a population, its frequency changes, resulting in different population specific allele frequencies for the mutation.

If a SNP is located in a 5' or 3' UTR, in an intron or in an intergenic region, it is a *noncoding SNP*. Noncoding SNPs can influence gene function by affecting transcriptional or translational regulatory sequences, splicing or RNA stability. *Coding SNPs* may be *non-synonymous SNP* (*replacement*) or *synonymous SNP* and they change the coding sequences which are processed to mRNA and are part of translation. Synony-

mous SNP results in a change of codon but not in a change of amino acid. Non-synonymous variations results in a change of amino acid and may have an impact on protein function.

This thesis focuses mainly on annotation analysis of SNPs.

2.3.2 Structural variations

Structural variants (Feuk et al. 2006) are genomic alterations involving large segments of DNA. One type of structural variation is a *copy-number variation* which is a repeated DNA segment occurring numerous times in comparison to the reference genome. Another type is *inversion*, occurring when a segment of DNA is reversed in orientation with respect to the rest of the chromosome. *Translocation* means position change of a segment within a genome while content is preserved. These types of variants are visualized in Figure 2-7.

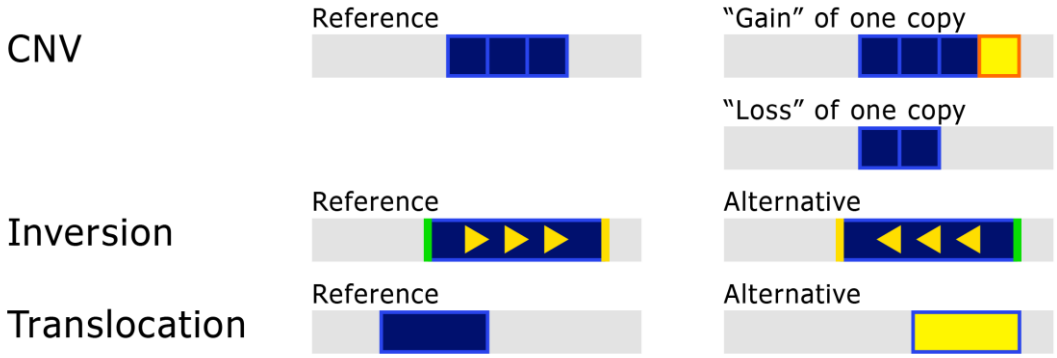


Figure 2-7. Structural variations.

2.3.3 Chromosome numerical abnormalities

Diploid organisms have chromosomes organised in pairs. Sometimes one chromosome is missing in a pair or there is an extra one. Mostly these disorders in human lead to spontaneous abortions. *Trisomy* is a disorder where an additional copy of chromosome appears. Down syndrome is the case of the trisomy on chromosome pair 21. *Monosomy* is a disorder of missing chromosome in a pair. There are also known cases where all chromosome pairs have one or two extra chromosomes. The sex chromosome pair is also susceptible to gain or loss of an X or Y chromosome.

2.3.4 Linkage disequilibrium

There is an observable correlation between alleles of a specific variations in a population, caused the common ancestry of the individual members. Because of recombination events during meiosis these variations are likely to be located near to each other. This correlation is called *linkage disequilibrium (LD)* (Greg & Muse 2004). In mammals, LD is observed for several tens or hundreds of kilobases in either direction from a variation. If frequencies of two alleles of distinct variations are p_1 and q_1, then frequency of their common incidence is $p_1 q_1$. To determine measure of LD, LD coefficient *D* can be used, as shown in **Error! Reference source not found.** with two variations and their possible alleles A_1, A_2 and B_1, B_2.

	$\boldsymbol{B_1}$	$\boldsymbol{B_2}$	**TOTAL**
$\boldsymbol{A_1}$	$p_{11} = p_1 q_1 + D$	$p_{12} = p_1 q_2 - D$	p_1
$\boldsymbol{A_2}$	$p_{21} = p_2 q_1 - D$	$p_{22} = p_2 q_2 + D$	p_2
TOTAL	q_1	q_2	1

Table 2-1. Haplotype frequencies with LD coefficient

For better interpretation coefficient D is divided by maximum possible value of D, which can be calculated by given allele frequencies:

$$D' = \frac{D}{D_{max}}. \tag{2.1}$$

The result D' from equation (**2.1** is in interval $[-1.0, 1.0]$. Another option is to use squared correlation coefficient:

$$r^2 = \frac{D^2}{p_1 p_2 q_1 q_2}. \tag{2.2}$$

The result r^2 from equation (**2.2** is in interval $[0.0, 1.0]$. The values of D' and r^2 near 0.0 means independence of variations to each other while value near 1.0 (or also -1.0 in case of D') means they are in a strong LD. The nearer to zero the value is the weaker LD is between variations. These values measures the strength of LD. Figure 2-8 shows

common way of visualizing variation LD where colour of each square represents LD distance between two particular SNPs.

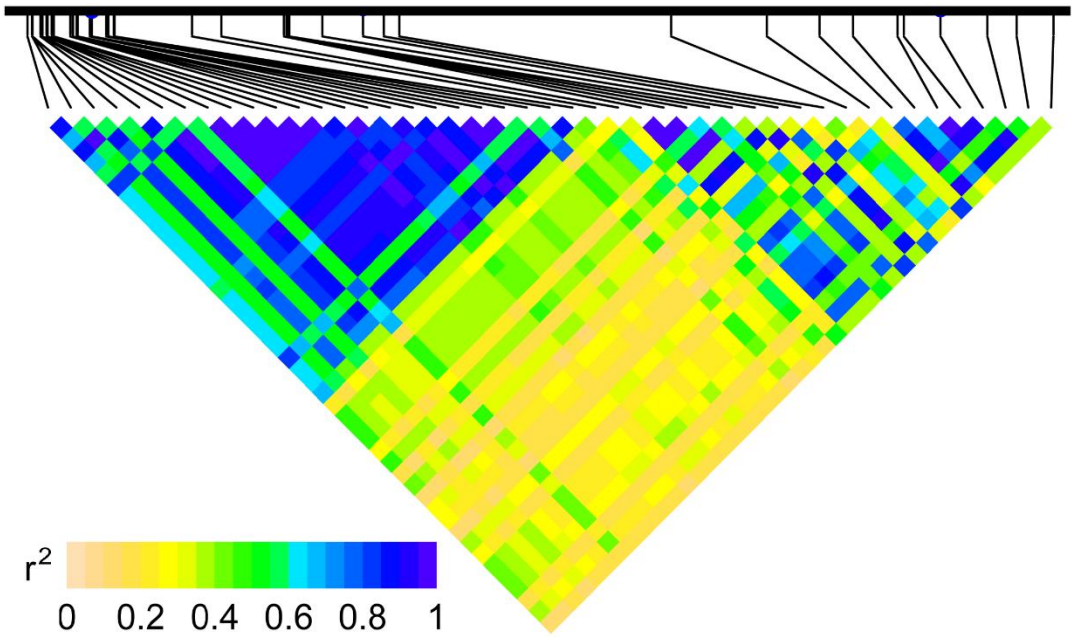


Figure 2-8. LD plot showing LD level of SNPs from each other using r^2 coefficient. Adapted from (By Weihua Shou,Dazhi Wang, Kaiyue Zhang, Beilan Wang, Zhimin Wang, Jinxiu Shi, Wei Huang [CC BY 3.0], via Wikimedia Commons).

2.4 Bioinformatics

In 1977 the first DNA sequencing techniques, now known as the Sanger method, were developed (Sanger et al. 1977). Since then we have been able to obtain DNA sequences. To analyse DNA sequences and to discover knowledge from them, computational methods are increasingly required (Orengo et al. 2003). Bioinformatics is a field which aims to develop and utilize efficient tools and methods to process vast amounts of biological data. It is an interdisciplinary field combining information technology and data analysis with biology and medicine. Statistics and mathematics are also extensively utilized in bioinformatics. Biological knowledge is gained by data analysis using statistical methods, software development of methods and tools, and data management for storing the biological data to be accessible for viewing, processing and analysis. Computational methods like alignment of sequences, clustering and machine learning methods are also common in solving bioinformatics challenges, and often they are NP-hard problems requiring heuristics approaches. Technological

advancement in genetics and information technology are rapid, making bioinformatics a quickly developing dynamic field.

3 GENETICS RESEARCH

We know how the information from DNA molecules is expressed by translating them into proteins. Genomics describe this process on molecular level in detail, though not all observable effects are explained and therefore roles of many genetics factors have not been revealed yet. To advance understanding of genetics, several genome projects have been initiated and many of them continue at present. Some of these projects have international scope, such as *GENCODE* (Harrow et al. 2012) and *RefSeq* (Pruitt et al. 2012) which aim to annotate all genes, their transcripts and proteins. They both use computational and manual methods to create annotations. Another project is *ENCODE* (The ENCODE Project Consortium 2004) which aims to identify all functional genomic elements, including regulatory sequences and all genomic locations that affect gene expression. *HapMap* (The International HapMap Consortium 2003) and *1000 Genomes* (Abecasis et al. 2012) projects are focused on describing genetic variations in human genomes. These projects form the basis for studies of genetic impact on human diseases. Besides widening our understanding of nature, genetics research is important for discovering new knowledge about diseases and their treatment. Large databases and computational methods are the necessary part of analysing effects and interactions of genes, variations or other functional elements, especially on the genome-wide level.

3.1 Genome Browsers

Genome browsers have been created to provide access to data from genome projects. Two best known genome browsers are *Ensembl* (Cunningham et al. 2014) and *UCSC Genome Browser* (Kent et al. 2002). Both of them provide access to periodically updated databases of genome elements of various species. For human they currently use genome assembly of version GRCh38 containing 20,364 coding genes, 196,345 gene transcripts and 65,897,584 short variants. Visualization of genome data is based on customizable tracks. Every track contains a specific type of information, for instance genes, transcripts, variations, regulatory elements, conserved segments or actual nucleotide sequence. They are visualized in relation to their location on the genome (Figure 3-1). Hundreds of tracks are available to visualization. Tracks with custom data

are supported as well. Each genome element has its own web page containing information about it and links to other related elements. Exporting selected data in common formats is available. *Ensembl* provides the *BioMart* interface (Kinsella et al. 2011) with filter and attribute options for accessing genes, variants and regulatory annotations of several species. The UCSC Genome Browser provides a comprehensive Table Browser (Karolchik et al. 2004) for obtaining genome data in common file formats or exporting data to another supported web tool and potentially applying filter, intersection or correlation operations on Table browser queries.

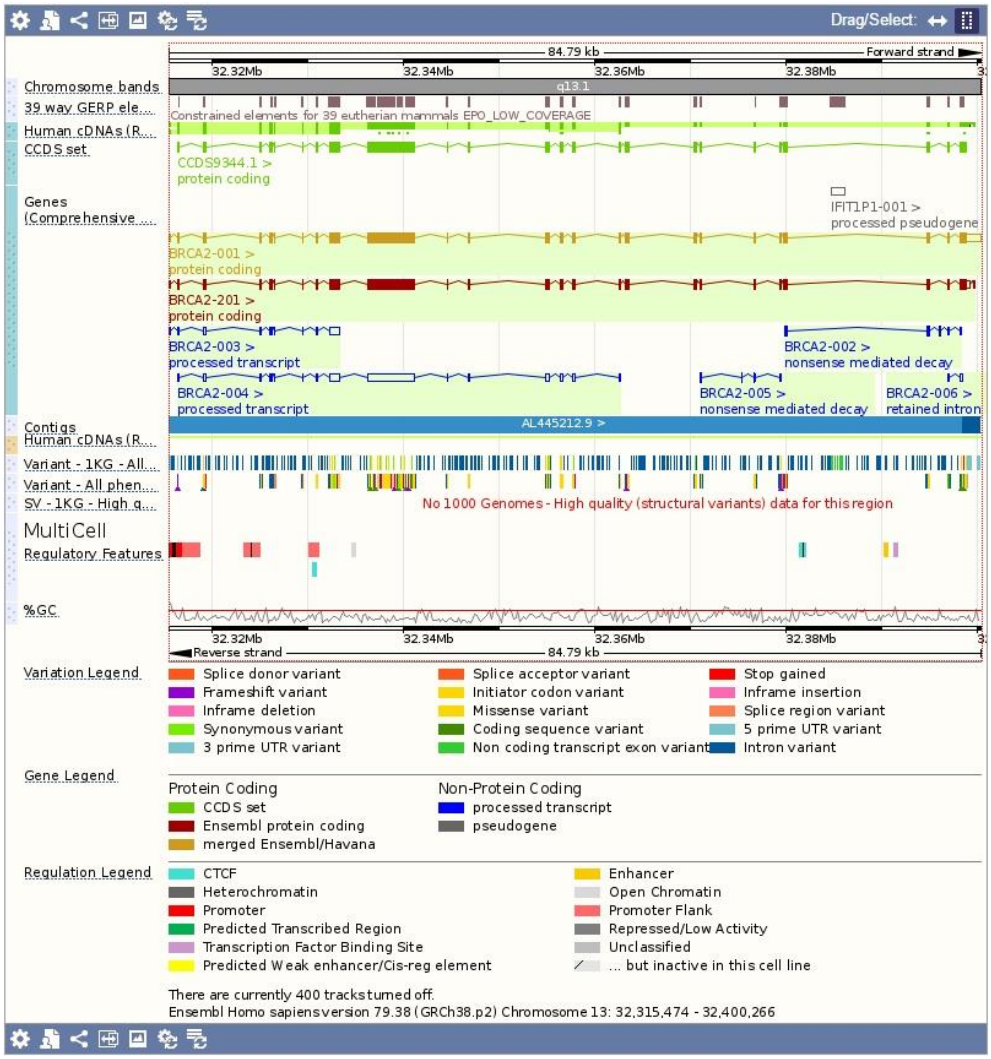


Figure 3-1. Screenshot from Ensembl browser showing various tracks in the region of BRCA2 gene.

3.2 Annotation methods

Genetics research and experiments provide enormous amount of new information. This requires definition of rules for annotating genomic elements and their relations to biological processes, functions and other elements. Every identified gene, transcript, protein or variant is assigned a unique identifier. There are more common conventions or ways to create identifiers for each type of element. For genomic elements of human and other species *Ensembl* creates an identifier by using a prefix defining the type of element and species, and then its unique numeric id. For instance, the human gene with identifier `ENSG00000139618` is processed to transcript `ENST00000380152` which is then translated to protein `ENSP00000369497.3`. This gene is also assigned

to unique symbol `BRCA2` by the *HUGO Gene Nomenclature Committee* (*HGNC*) (Gray et al. 2015). HGNC symbols are usually abbreviations of description of their function or effect on phenotype. For example, the `BRCA2` gene is associated with breast cancer. For the genetic variations the most common source of identification is *dbSNP* (Kitts et al. 2014) which uses identifiers like `rs1815739`. Variations can be also identified by an identifier that includes the absolute genomic position of the element and allele change (`13:g.32316513G>A` – nucleotide on position 32,316,513 on the chromosome 13, with alleles G and A) or including the relative position within a region of a genomic element (`ENST00000380152.6:c.4563A>G` – nucleotide on position 4,563 from the beginning of the coding sequence of transcript `ENST00000380152` of version 6, with alleles A and G).

Many annotations of genes and other genomic elements are based on genome sequence analysis involving computational methods. Gene annotations are briefly described in the following subchapter. Currently intensive efforts are performed to find associations of short variations to phenotype. Many software tools were developed to perform analysis of short variations and imputing their annotations. We will also focus on them in the following subchapters, which are based on a review of these tools and methods provided by Graham RS Flicek and Paul Flicek (Ritchie & Flicek 2014).

3.2.1 Gene annotations

Annotations can be organized into an ontology. One of the best known ontologies for gene annotations is the *Gene ontology (GO)* (Ashburner et al. 2000) which consists of its three parts: *cellular component*, *molecular function* and *biological process*. An advantage of GO is that biological terms are ordered in an easily extensible ontology tree which defines a hierarchy and relationships between individual terms (Figure 3-2). An Example of GO term is `GO:0000910` which stands for the biological process of cytokinesis.

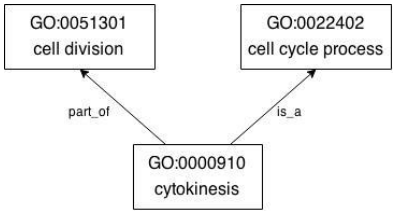


Figure 3-2. A part of the GO hierarchy.

Another well-known annotation source is *Online Mendelian Inheritance in Man (OMIM)* (Amberger et al. 2015). *OMIM* contains curated descriptions of human genes and phenotypes identified by unique six digit identifier, and their associations.

Several databases of biological pathways have also been built, for example *Reactome* (Croft et al. 2014), *BioCarta* (BioCarta LLC n.d.) and *KEGG* (Kanehisa et al. 2012). The *hiPathDB* (Yu et al. 2012) database integrates data from these databases and provides visualization of pathways by its layout algorithm.

3.2.2 Variation position in gene

When detecting a variant that could have an effect on a phenotype, the variant can be linked to nearby genes to study its effect on the genes. For this purpose, the variation needs to be annotated with information about the nearest gene or, more generally, its position to nearest one or two genes on both sides. Therefore high-performance methods based on region intersection, union or other similar set operations, are used for example in tools *Bedtools* (Quinlan & Hall 2010) and *Bedops* (Neph et al. 2012).

More specific classification of variations is based on their position inside a gene, and in case of the variation being located in coding region of a gene, also the allele change. In other words, on *consequence* of a variation to a gene. Consequence terms are standardized within the *Sequence ontology (SO) project* (Eilbeck et al. 2005) (Table 3-1).

SO term	**SO description**
`splice_acceptor_variant`	Variant at the 3’ end of an intron.
`splice_donor_variant`	Variant at the 5’ end of an intron.
`stop_gained`	Variant causing change of codon to stop codon. The transcript is shortened.

SO term	SO description
frameshift_variant	Variant causing the length of sequence to translate not divisible by three. The following codon triples are rearranged.
stop_lost	Variant causing change of stop codon. The transcript is elongated.
initiator_codon_variant	Variant causing change of initiating codon.
missense_variant	Variant causing change of amino acid (non-synonymous variation).
synonymous_variant	Change of nucleotide not causing change of amino acid.
5_prime_UTR_variant	Variant at the 5' end UTR.
3_prime_UTR_variant	Variant at the 3' end UTR.
intron_variant	Variant in the intron region.
upstream_gene_variant	Variant located near to the 5' end of a gene.
downstream_gene_variant	Variant located near to the 3' end of a gene.
TF_binding_site_variant	Variant located in the region of TF binding site.
intergenic_variant	Variant located in the intergenic region.

Table 3-1. Part of the variations consequences.

Consequences are basic prediction of variation's effect on the functionality of a gene. Tools determining consequence include *Variant Effect Predictor* (McLaren et al. 2010) or *ANNOVAR* (Wang et al. 2010).

3.2.3 Impact of non-synonymous variations

Major candidates for variants that change gene function are variants that are located in the translated section of a gene, and which in addition change the resulting amino acid sequence. Many methods have been developed for predicting the impact of this type of change.

SIFT (Kumar et al. 2009) is a predictor based on the level of conservation of the amino acid sequence. At first, it finds all homologous sequences to the given sequence and

applies the multiple-sequence alignment on them. Then, for all positions of these sequences the *conservation value* is computed. If all 20 possible amino acids are located in the column of the alignment the conservation value is zero. If there is only one amino acid the value is $\log_2 20 \ (= 4.32)$. For getting the most confident results only such a subset of found homologous sequences is chosen for prediction which has a median conservation value of ~ 3.0 . Afterwards, the probabilities of all amino acids are computed according to their occurrence in their position in the obtained subset of sequences. The probabilities are normalized by the amino acid with highest probability. The substitution is predicted as the impact to protein function if the substituted amino acid probability is under the certain threshold. The tool *FATHMM* (Shihab et al. 2013) enhances the accuracy of this method by using a Hidden Markov Model (HMM) of the surrounding sequence. Moreover,an HMM can model amino acid insertions and deletions so it is possible to predict their impact as well.

The next approach to determine damaging impacts is adopting the machine learning methods. Already known variants with certain features are used as training set. The classifier is trained on them and then it predicts the impact of novel variants. A tool using this approach is *Polyphen* (Adzhubei et al. 2010). When creating the Polyphen classifier 32 features were initially considered but after a validation process their number was reduced to 11. Three of them are structure features while remaining of them are sequence features. The conservation level is also examined in a similar manner as in SIFT. Classification is based on a Naïve Bayes approach as authors favoured it for its simplicity and few parameters. The training set of variants was collected from the *UniProt* project (Apweiler et al. 2004). Another tool *SNAP* (Bromberg & Rost 2007) uses neural networks as classifiers, a third tool *PhD-SNP* (Capriotti et al. 2006) uses support vector machines.

3.2.4 Impact of non-coding variations

It seems that the non-synonymous variations are the only functional variations as they change the resulting polypeptide sequence which likely affects phenotype. However, many variations identified by genome-wide association studies, which we will discuss below in the text, are located in UTR, intron regions or intergenic regions (Hindorff et al. n.d.). Many functional variations were also identified in locations of functional non-

coding RNA genes, TF binding sites or other regulatory elements. For this reason, several software tools are aimed on imputing functional consequences of non-coding variations.

The tool *Haploreg* (Ward & Kellis 2012) detects intersections of variations with regions of certain chromatin states, conservation as well as regulatory motifs. For regulatory motifs the change of PWM score is computed.

RegulomeDB (Boyle et al. 2012) produces heuristic assessment for variations and predicts with certain confidence if they lie in a functional location. The assessment is done by assigning the score defining confidence of prediction. The variations located in a known eQTL for genes with combination of evidences that they intersect with the region of TF binding, TF regulatory or other motif or region of increased DNAse activity have the highest score. The less score has variations without known eQTL but intersecting with mentioned regions. The more evidences are assigned to the variations the higher the score is.

A machine learning method is used in *GWAWA* (Ritchie et al. 2014) by accommodating a random forest algorithm to build the classifier. Common variants already known to be contributing to diseases from the *Human Gene Mutation database (HGMD)* (Stenson et al. 2014) were used among with three control variation sets so they created three classifiers. The first set was created by random selection of variations across the genome. As many variations from HGMD lie near to transcription start site, the second control set was created from variations near to it as well. The final control set was generated from variations from 1000 Genomes project surrounding all variations from HGMD.

3.3 Genetic diseases

Under the assumption that every disease has at least partly genetic cause, it is a substantial part of genetics research to identify genetic sites impacting phenotype traits. Several diseases were observed to be heritable within families and having high proportion of genetic origin. One of such diseases is cystic fibrosis which is caused by several mutations in *CFTR* gene (Bush & Moore 2012). The human is affected by

disease when both gene sequence pairs contain those mutations which follows an *autosomal recessive inheritance pattern*. If the mutations occur only on a single chromosome of the pair then the individual is only a carrier of the cystic fibrosis and is not affected. It means that one or more individuals from his offspring may get or may not get affected by the cystic fibrosis. If, for example, both parents are carriers then it is 25% chance that their child will suffer by cystic fibrosis, 50% chance that it will be carrier and 25% that its both gene sequences will have harmless alleles. Another genetic disease is Huntington's disease following an *autosomal dominant inheritance pattern* which means that it is sufficient to have only one harmful allele on a single chromosome of the chromosome pair to be affected. If the disease allele spreads across the next generations at least one individual in each generation is affected by disease.

Observation of spreading of genetic diseases through the pedigree suggested a large proportion of genetic cause so there were efforts to determine the causal genome site. Therefore pedigree studies were performed to identify them. The pedigree studies examines genotypes of families, in which the disease is common, and with knowledge of inheritance pattern of disease they look for the causal variations. These studies successfully identified causal mutations in the *CTFR* gene for cystic fibrosis.

3.4 Genome-wide association studies

For heart diseases or cancer, pedigree studies described above are not successful in identifying genetic causal sites. That suggests that these *common diseases* are not purely genetic and they might have multiple causal loci in the genome.

For studying common traits and diseases *Genome-wide association studies (GWAS)* (Bush & Moore 2012) were established. While traditional genetic studies were focused on a single locus, such as a gene, considering small sample of genotypes, the goal of the GWAS is to find linkage between many genetic variants and phenotype on the genome-wide level and considering hundreds of thousands or millions genotypes. This linkage is easier to find when the proportion of genetic cause to disease is high. When the proportion of environmental factors is high it is more difficult to find genetic linkage with satisfying confidence. The motivation for establishing GWAS was the assumption that genetic factors of common diseases consist of effects of variations which

are common as well. This is described by *common disease-common variant hypothesis (CDCV)* (Reich & Lander 2001).

3.4.1 Common disease-common variant hypothesis

The common diseases show heritability in pedigrees as well what indicates that their causes have some genetic component. The common disease-common variant hypothesis supposes that the genetic factor is caused by the common variants which have higher minor allele frequency than rare variants causing rare traits or diseases like cystic fibrosis. However, the common phenotype does not occur as frequently as common variants which can reach for example 40%. This indicates that if common variants cause the common diseases then their effect size (*penetrance*) is small and their correlation with disease phenotype is rather slight. The fact that even common diseases can be observable in unusual frequencies in certain families in comparison to population implies that there must be more variants or sites in genome influencing the disease risk by a certain smaller amount. Therefore the pedigree studies orienting on genetic diseases are not suitable for studying the common diseases but much larger sample of individuals are needed. The common disease studies must be done on the population level. The problem of population studies is in their extensiveness what requires vast amounts of resources (millions of euros, time required for collecting the genotypes and descriptions of the phenotypes of huge number of individuals). This requires that their design has to be effective and precise. The HapMap and 1000 Genomes projects were created for building databases of variations, their frequencies and linkage disequilibrium information in selected populations to support finding the associations between variations and diseases.

The common and rare variants should not be considered as two distinct groups. We can recognize variants with various frequency and effect size somewhere between typical values of rare and complex disease variants as it is seen in Figure 3-3.

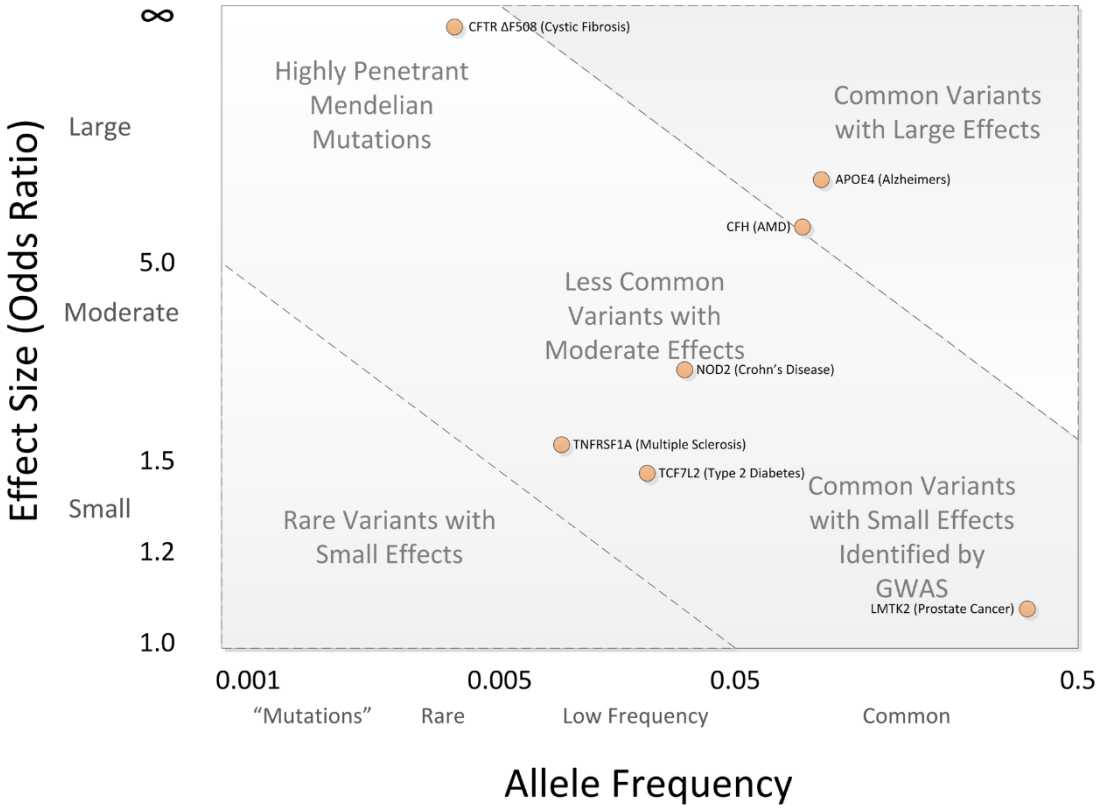


Figure 3-3. Spectrum of variants with disease effects. (By Bush WS and Moore JH [CC BY 2.5], via Wikimedia Commons)

The highest effect size have rare variants causing for example Cystic Fibrosis. The area near to upper left corner is typical for rare disease variants. In the upper right corner are common variants with large effects which are not so typical. However, the disease allele of Alzheimer is quite near to this corner so it can be recognized in either pedigree or population study. The lower right quarter is area of variants which are possible to identify only in population-level studies. The most complicated case is represented by variants near the lower left corner because their studies require huge sample to be able provide statistically significant results.

In performed studies, the CDCV hypothesis was not completely confirmed. If the hypothesis were true, the studies would explain all genetic components influencing the trait. In reality, the genetic factors are not fully revealed (Manolio et al. 2009). For example, 40 sites were identified for the complex trait of human height by large GWAS, but they comprise only about 5% of phenotypic variance when the genetic factors are estimated on about 80%. Other alternative models to CDCV hypothesis are discussed (Gibson 2011). The *infinitesimal model* is similar to the CDCV hypothesis

in the principle of multiple common sites influencing the trait. However, the infinitesimal model supposes that there are larger number of influential sites with small effects and theoretically that each site (or each gene) contributes to trait by some amount which is often difficult or impossible to measure because of significance thresholds. The *rare allele model* accepts common influencing sites from the CDCV hypothesis or environmental factors but as the main trait contributors considers the large number of rare alleles of large and multiplicative effect. Accordingly the sufficient number of rare alleles comprise a major contribution to the disease state. The *broad sense heritability model* considers the previous models as insufficient. It supposes the existence of other genetic features like interactions of genotypes between each other or interaction of genotypes with the environment so various combinations of genotypes with environment have various contributions to the trait. It also suppose the inheritance of DNA methylation patterns which have regulative impact on genes expression.

The population GWAS were inspired by the CDCV hypothesis and introduced a belief that they will lead to the revolution of human medicine (Visscher et al. 2012). Though they produced findings in several diseases it is needed to expend more effort to explain and understand the mechanisms of genetic contributions to complex traits. Methods in GWAS are still being developed to bring more results.

3.4.2 Genome-wide association study workflow

To perform a population GWAS, several steps are needed to carry out (Bush & Moore 2012). The first one is clearly definition of phenotype to study. It is needed to create exact rules of assignment of a high number of individuals (for example 10,000) to categories which are accepted in all centres collecting phenotype information not to introduce bias to the study. Electronic medical records can be a source of information as well and used by adopting natural language processing methods but which are needed to be validated and refined to avoid bias. The essential decision is the choice whether to use quantitative or case-control design. Quantitative design requires the ability to define the phenotype quantitatively. Results of GWAS with this design are more powerful and interpretable, nevertheless it is not necessary to achieve valuable results. The case-control design divides the individuals into cases, for example with disease status, specific trait or status of responding to drug, and non-cases.

The second step is collecting the genotypes from the individuals in the sample. Technologies for extracting variant alleles are available, for example, *Affymetrix* or *Illumina* SNP microarray series can sequence mostly from one hundred thousand to one million of variations per genome. They cover only a subset of variations but all causal variations may be detected via linkage disequilibrium with correlating variation in the subset. Nowadays next-generation sequencing outperforms current common technologies as they are able to sequence entire genomes but for purpose of GWAS they are still quite expensive.

When having the list of observed variations then association tests for variations are performed. The type of association tests depends on the selected study design. Generalized linear models approaches are appropriate for quantitative design, for example *Analysis of Covariance* related to linear regression. The table contingency tests and logistic regression are appropriate for dichotomous design. Table contingency tests include the chi-square test or Fisher's exact test. Logistic regression is extended from linear regression. Regression methods can be adjusted for factors and clinical covariates. The results of association tests are adjusted for multiple tests or genome-wide significance level is pre-determined because some of the very large number of association tests, for example one million, will produce false-positives. These methods will be covered in more detail in Chapter 4 where enrichment analysis methods for annotations except logistic regression, which is covered in the next subchapter, are described. Results of association tests are often visualized by *Manhattan plots* (Figure 3-4).

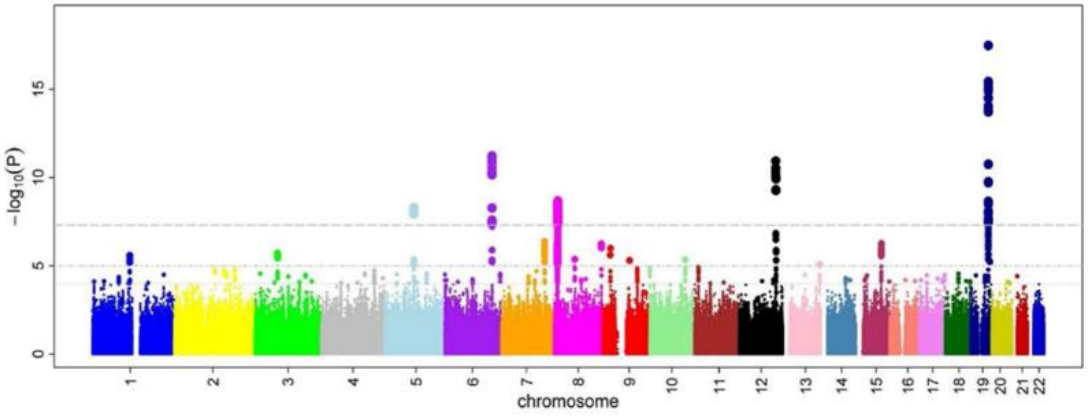


Figure 3-4. Example of Manhattan plot depicting variations p-values of association tests and showing genome-wide signifiance levels. Each filled circle represents p-value for a variation which location on genome is determined by position in the direction of the chromosome axis. (By M. Kamran Ikram et al [CC BY 2.5], via Wikimedia Commons)

The association can be *allelic* when association of one allele of SNP to phenotype is analysed or *genotypic* when association of genotype to phenotype under certain model is analysed. As an example, these models considering diallelic locus include dominant, recessive, additive or multiplicative model. In each of these models allele *a* is a non-risk allele while *A* is a risk allele. Dominant model supposes that if only one of the alleles from couple is risk allele then genotype is also considered as risk genotype. Recessive model supposes as a risk allele only *AA* while *Aa* is an unaffected carrier genotype. The dominant and recessive models were described in more detail in Chapter 3.3. Additive model supposes risk from the number of alleles so their effects in *aa*, *Aa* and *AA* may be encoded as 0, 1 and 2. Finally multiplicative model captures multiplicative effects so instead of allele encodings 0, 1 and 2 the allele effects may be for example 0, 3 and 9.

The methods above are mainly based on the CDCV hypothesis or infinitesimal model. However, for capturing the Broad sense heritability model these kinds of methods are not sufficient because they analyse only single variations one after another. However analysing combinations of alleles for a large number of variations would be too computationally expensive. From one million variations can be created $5 . 10^{11}$ pair combinations and with 100,000 genotypes it would be needed to perform $5 . 10^{16}$ association tests. For triples it is needed to perform another $3 . 10^{22}$ tests. Therefore it is more suitable to perform single tests at first and then analyse for multiple variation effects

in smaller set of variations. Also heuristics methods may decrease the number of combinations to test. The next option of analysing multiple variation effects is to include information about biochemical pathways or protein families. Bayesian or related methods also bring efficient methods to simultaneous analyse of variations (Wray et al. 2013).

Now the study has identified suspicious sites related to trait. The results should be confirmed with a *replication study* on another (and if possible larger) samples of individuals but with the smaller set of SNPs which were computed in the study as the most significant. When a phenotype trait is studied by several GWAS the *meta-analysis* can be performed to unify the results of them. This requires to pass several challenges such as the difference between sets of the genotyped variations or other heterogeneity between GWAS.

3.4.3 Regression methods

Regression methods (Neale et al. 2012) determines if independent variables are correlated with a dependent variable of interest. The simple linear regression is defined by equation 3.1:

$$y = a + bx. \tag{3.1}$$

To calculate a and b from a sample of data the ordinary least squares method is used (equations (3.2 and (**3.3**):

$$b = \frac{\sum(x - \overline{x})(y - \overline{y})}{\sum(x - \overline{x})^2}, \tag{3.2}$$

$$a = \overline{y} - b\overline{x}. \tag{3.3}$$

The simple regression can be extended to multiple regression (equation (3.4) to determine the correlation of multiple variables with a dependent variable:

$$y = a + b_1x_1 + b_2x_2 + \cdots + b_nx_n. \tag{3.4}$$

By adding covariates affecting the dependent variable as confounding factors the bias can be reduced. In practice, environmental or other factors can be added such as smoking, age or sex. The multiple regression can also capture more genotype models at one time when each genotype model corresponds to separate independent variable.

A special case of multiple regression is *Analysis of Covariance* when one of the independent variables represents a nominal parameter like if the drug was applied to an individual. In the case there is more than one of these nominal variables (for example sex), the regression is *Multifactor Analysis of Covariance*. When the dependent variable is nominal, what for dichotomous traits is typical, the term Logistic regression is applied for this regression. In that case the coefficients are usually converted to coefficients representing odds ratios (equation (3.5):

$$y = e^{a} e^{b_1 x_1} e^{b_2 x_2} \dots e^{b_n x_n}. \tag{3.5}$$

After this conversion we can interpret each value of e^{b_i} as increase of factor, by which the probability of that y equals to 1. As an example for GWAS, the dependent variable y represents disease case or non-case, x_1 can be a variable for SNP with alleles T and G, while TT can be encoded as value of 0, GT as value of 1 and GG as value of 2. Variable x_2 can represent smoking, whether individual smokes or not which is encoded in values 1 or 0. Having a list of values for each individual in the format <*allele, smoking*> the application of the logistic regression can return odds ratio of $x_1$variable as value of 2.241 and x_2 of 1.485. This means that the SNP is an influencing factor for disease. Smoking influences disease as well but not so significantly because it is nearer to value of 1.0.

3.4.4 Results from genome-wide association studies

For common diseases, the number of variations have been found to be in association with disease state. GWAS can detect multiple loci with small effects. For example, about 95 sites were identified in meta-analysis for high-density and low-density lipoprotein levels related to cardiovascular diseases in >100,000 individuals of European ancestry, 10 genes were revealed to be related with multiple sclerosis (Bush & Moore 2012) and 71 sites with Crohn's disease in meta-analysis in about 6,000 cases and 15,000 controls of European population (Franke et al. 2010). Other sites were revealed

in auto-immune and metabolic diseases (Visscher et al. 2012) and also 79 sites in recent breast cancer meta-analysis (Michailidou et al. 2015) from around 47,000 cases and 42,892 controls from European population as well. Thousands of GWAS have been already performed. The catalogue of published GWAS publications contains 15,396 SNPs (Hindorff et al. n.d.). However, only a fraction of them were biologically confirmed as functional. But many of them have helped to discover new biological mechanisms related to diseases. If GWAS have not yet fulfilled the initial expectations they at least contributed to knowledge in genetics or biologics fields (Visscher et al. 2012).

4 ENRICHMENT ANALYSIS

Enrichment analysis looks for enrichment of annotation terms associated with a set of genes or variations in comparison to a background set and computes the statistical significance of enrichment (Huang et al. 2009a). Enrichment means that some annotation terms are overrepresented or underrepresented in a provided set of genes or variations which comes for example from GWAS or from gene expression analysis. Several methods dependent on computational power exist for determining the level of enrichment and its statistical significance. They can be classified into several classes: contingency table methods, permutation methods, methods using continuous values and regression methods. Regression methods are not common in annotation enrichment analysis because they have higher computational complexity in comparison to previous methods.

4.1 Contingency table methods

Contingency tables contain information about the number of observed items or features within category groups of entities. It is possible to use them to record the number of genes or variants with and without annotation in distinct sets of genes or variants. In case of association tests in GWAS it is possible to use them to record a number of alleles in groups of individuals. The following text is aimed on annotation enrichment analysis and in the following Table 4-1. Example of contingency table with number of annotations of term "Sift: deleterious" is the example of observation of annotation counts in variations set.

	Sample variants set	Remaining variants from background set	Sum
Annotated by "Sift: deleterious"	12	1,197	1,209
Not annotated by "Sift: deleterious"	988	907,101	908,089
Sum	1,000	908,298	909,298

Table 4-1. Example of contingency table with number of annotations of term "Sift: deleterious"

Sample of variations has size of 1,000. These variations are part of the background set of total size 909,298. There are 12 variations annotated by *Sift: deleterious* term in the sample and it is known that in the background set is 1,209 annotated variations what means that there are remaining 1,197 annotated variations in the remaining variations of the background set. The task is to get information about statistical significance of overrepresentation or underrepresentation of term "Sift: deleterious" in sample variants set.

4.1.1 Hypergeometric test

The one of the most common method for computing significance of over or underrepresentation is hypergeometric test which uses hypergeometric distribution (Rice 2007). From this distribution it is possible to compute the probability of 12 annotations of "Sift deleterious" in the sample variants set according to the following equation (4.1:

$$P(X = k) = \frac{\binom{r}{k}\binom{n-r}{m-k}}{\binom{n}{m}}. \quad (4.1)$$

Let k be the sample annotated variants count, r be the total count of annotated variants to, n be the total count of variants and m be the total count of sample variants. Then that probability equals $1.535486 \cdot 10^{-8}$. For obtaining the probability of observing 12 or more annotations in sample variants we need to compute $P(X \geq 12)$, which equals to $1.70514 \cdot 10^{-8}$. This value means the probability of observing an extreme of 12 or

more hits. If we set up a null hypothesis that selection is random, that value is a p-value providing information if the null hypothesis is likely true or false. As this p-value is very low it is very little chance that so many annotations would be associated with sample variants if these variants were chosen randomly so it is possible to reject the null hypothesis. Therefore term "Sift: deleterious" is statistically significantly overrepresented. Computation of $P(X \leq k)$ would lead to a value very near to 1 so the term underrepresentation is not statistically significant or in this case it is better to say that there is no underrepresentation of the term.

The problem of hypergeometric distribution is that it is not computationally simple to compute probabilities of term occurrences in variants when using high numbers of variations in Table 4-1. Moreover numeric overflows may occur when using naïve algorithms. Therefore several methods were developed for efficient computation. The methods should be able to perform thousands or even more computations of hypergeometric test in reasonable time.

Aleš Berkopec has developed the HyperQuick algorithm (Berkopec 2007) with complexity $O(n-m)$ with assumption of unlimited size of processor registers. In case of the computation of underrepresentation the algorithm is based on the derived equation (4.2:

$$P(X \leq k) = 1 - \frac{\frac{1}{j_{r_0-1}} \cdot \frac{1}{j_{r_0-2}} \cdots \frac{1}{j_r} \cdot \frac{1}{j_{r-1}} \cdot s}{1 + \frac{1}{j_{r_0-1}} \cdot \left(1 + \frac{1}{j_{r_0-2}} \cdot \left(1 + \cdots \frac{1}{j_r}\left(1 + \frac{1}{j_{r-1}} \cdot s\right)\right)\right)}, \quad (4.2)$$

while it is set in equations (4.3,(**4.4** and (**4.5**:

$$\boldsymbol{r_0 = n - m + k}, \quad (4.3)$$

$$s = 1 + \frac{1}{j_{r-2}} \cdot \left(1 + \frac{1}{j_{r-3}} \cdots \frac{1}{j_{k+1}}\left(1 + \frac{1}{j_k}\right)\right), \quad (4.4)$$

$$j_p = \frac{1 - \frac{m-1-k}{n-1-p}}{1 - \frac{k}{p+1}}. \quad (4.5)$$

The first step of algorithm consists of computation of s, which is done by a loop of $r - 1 - k$ iterations. Then the cumulative probability is computed in another loop of $n - m + k - r + 1$ iterations or possibly lower if the accuracy of the result is set. When the accuracy is not applied this algorithm computes the exact cumulative probability.

Methods for computation of cumulative probability of hypergeometric distribution are part of the libraries in several programming languages, for example in Python in the package Scipy (The Scipy community 2015) or in R (R Documentation n.d.). In R, cumulative probability of hypergeometric distribution is computed by function `phyper`.

Many existing tools for enrichment analysis of genes use the hypergeometric test. As an example, DAVID (Huang et al. 2009b) is a web tool providing enrichment of annotation terms of various sources in gene sets. The GREAT (McLean et al. 2010) adopts hypergeometric tests for determining enrichment in genes associated with given genomic regions via genes' regulation domains.

4.1.2 Fisher's exact test

For testing the validity of null hypothesis the Fisher's exact test is widely used (Rice 2007). It also uses the hypergeometric distribution to determine the probability of finding certain number of "Sift: deleterious" annotations in sample variants set in the case of 2x2 table of Table 4-1. There are two kinds of Fisher's tests: one-tail and two-tail (Preacher & Briggs 2001). One-tail Fisher's test is equivalent to the test based on hypergeometric distribution. Two-tail Fisher's test cumulates also probabilities of possible extreme observations in the direction of another side with lower probabilities than of the experiment observation. For the purpose of defining either overrepresentation or underrepresentation significance the one-tail test is used.

4.1.3 Binomial test

While hypergeometric distribution model supposes changing of probability of hit after each test of experiment in binomial distribution (Rice 2007) the probability is still the

same. The probability of k successes in a binomial experiment is calculated by equation (4.6:

$$\boldsymbol{P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}}, \tag{4.6}$$

where n is number of trials and p probability of success. As the variants sample count is often much lower than count of remaining background set the probability of hit after each successful test changes only a little. Testing the overrepresentation or underrepresentation by binomial distribution leads to approximate but mostly sufficient accuracy. Taking the example of Table 4-1 and as n using the number of variants in sample, k as the number of the annotated sample variants and p as proportion of annotated variants from all the variants would give the value of $1.78068 \cdot 10^{-8}$ which is similar to the value from hypergeometric test.

The cumulative probability of binomial probability could be computed by the *incomplete beta function* (Press 1992) which is more convenient for evaluation of large parameters than evaluating of binomial coefficients. The incomplete beta function is defined by equations (4.7 and (**4.8**:

$$\boldsymbol{I_x(a,b) = \frac{1}{B(a,b)} \int_0^x t^{\alpha-1}(1-t)^{b-1} dt}, \quad \boldsymbol{(a,b > 0)}, \tag{4.7}$$

$$B(a,b) = \int_0^1 t^{x-1}(1-t)^{y-1} dt, \tag{4.8}$$

and the relationship between the binomial cumulative function and the incomplete beta function is defined in equations (4.9 and (**4.10** (Press 1992):

$$\boldsymbol{P(X \leq k) = 1 - I_p(k+1, n-k)}, \tag{4.9}$$

$$P(X \geq k) = I_p(k, n-k-1). \tag{4.10}$$

For numerical evaluation the incomplete beta function can be expressed by equation (4.11 (Press 1992):

$$\boldsymbol{I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \cdot \left[\frac{1}{1 + \frac{d_1}{1 + \frac{d_2}{1 + \cdots}}} \right]}, \tag{4.11}$$

where for odd d coefficients the equation (4.12 is:

$$\boldsymbol{d_{2m+1} = -\frac{(a+m)(a+b+m)x}{(a+2m)(a+2m+1)}}, \tag{4.12}$$

and for even d coefficients the equation (4.13 is:

$$\boldsymbol{d_{2m} = \frac{m(b-m)x}{(a+2m-1)(a+2m)}}. \tag{4.13}$$

The continued fraction of equation (4.11 converges at latest after $O(\sqrt{\max(a,b)})$ evaluated coefficients for $x < (a+1)/(a+b+2)$. For $x > (a+1)/(a+b+2)$ it is possible to use symmetry of incomplete beta function to achieve convergence (equation (4.14) (Press 1992):

$$\boldsymbol{I_x(a,b) = 1 - I_{1-x}(b,a)}. \tag{4.14}$$

The continued fraction can be computed by *Lentz's method* (Press 1992).

One of the most known tool using the binomial test is GREAT (McLean et al. 2010) which adopts it for determining enrichment in genes associated with given genomic regions via genes' distal regulation domains. The proportion of all distal regulatory domains of annotated genes to whole genome is considered as the probability of a binomial hit while n would denote the number of all genomic regions and k the number of genomic regions within regulatory domain of any gene.

In R (R Documentation n.d.), the function `pbinom` may be used for binomial test.

4.1.4 Pearson's chi-square test

The Chi-square distribution (Rice 2007) is a continuous distribution and can be used to test overrepresentation of categorical variables as well. Its probability density function is defined by equation (4.15:

$$P_r(x,n) = \frac{1}{2^{n/2}(n/2-1)!}x^{(n/2)-1}e^{-x/2}, \tag{4.15}$$

where n is the degree of freedom of distribution. To perform Pearson's chi-square test (Rice 2007, p. 342), which is based on the chi-square distribution, expected values of cells of Table 4-1 are needed to be determined from marginal counts. The expected values are shown in Table 4-2:

	Sample variants set	**Remaining variants from background set**	**Sum**
Annotated by "Sift: deleterious"	1.329597	1207.670403	1209
Not annotated by "Sift: deleterious"	998.670403	907090.3296	908089
Sum	1000	908298	909298

Table 4-2. Table of expected values determined by marginal counts.

For example, the expected count of annotated variants in the sample set is $(1000 \cdot 1209)/909298 = 1.329597$. To perform the test, the sum of squared differences between observed and expected counts divided by expected counts has to be computed in each cell of table (equation (4.16):

$$X^2 = \sum_{i}^{I}\sum_{j}^{J}\frac{(O_{ij}-E_{ij})^2}{E_{ij}}. \tag{4.16}$$

In case of Table 4-2 this gives the value of 85.842. We use degree of freedom of 1, as the table has two rows and two columns. In case of more rows or columns, the degree of freedom is computed by $DF = (rows-1)\cdot(columns-1)$. The p-value is determined by chi-square distribution with degree of freedom 1 by computing $P(x >$

85.842, 1) which is less than $2.2 \cdot 10^{-16}$ (function `chisq.test` in R). As it is seen this value differs from the value computed by the hypergeometric or the binomial test so in this case the result of Pearson's chi-square test differs from the exact result quite significantly. Note that Pearson's chi-square test is an approximate test while hypergeometric or binomial tests are exact ones.

Computation of the cumulative probability of the chi-square distribution can be performed by computing the lower tail of the cumulative probability of gamma distribution (Reeve 1986) which is defined in equation (4.17:

$$\boldsymbol{P_G(x,k) = \int_0^x \frac{t^{k-1}e^{-t}dt}{\Gamma(k)}\ (x \geq 0; k > 0)},\tag{4.17}$$

where according to equation (4.18:

$$\boldsymbol{\Gamma(k) = \int_0^\infty t^{k-1}e^{-t}dt}.\tag{4.18}$$

The upper tail $Q_G(x,k) = 1 - P_G(x,k)$ differs in its interval of integral which is $[x, \infty]$. The relationship between the chi-square and the gamma cumulative distribution probabilities is defined by equation (4.19 (Reeve 1986):

$$\boldsymbol{P(P_r(x,n) < x) = P_G(x/2\,, n/2)}.\tag{4.19}$$

The computation methods use the recurrence relation of the gamma distribution cumulative probability (equation (4.20) (Reeve 1986):

$$\boldsymbol{P_G(x,k) = \frac{x^k e^{-x}}{\Gamma(k+1)} + P_G(x,k+1)}.\tag{4.20}$$

Repeating the computation of $P_G(x,k)$ by recurrence relation (4.20 n times gives the following equation (4.21:

$$\boldsymbol{P_G(x,k) = \sum_{i=1}^{n} \frac{x^{k+i-1}e^{-x}}{\Gamma(k+i)} + P_G(x,k+n)}.\tag{4.21}$$

By giving the required accuracy ε there is an n such that $P_G(x, k+n) < \varepsilon$ so it is sufficient to take the sum from equation (4.21 to get an approximated result of specified accuracy. To test if the $P_G(x, k+n)$ is under the value of ε, it is possible to use equations (4.22 (Reeve 1986):

$$\begin{aligned} \boldsymbol{P_G(x, k+n)} &= \sum_{\boldsymbol{i=n+1}}^{\infty} \frac{\boldsymbol{x^{k+i-1}e^{-x}}}{\boldsymbol{(k+i-1)!}} \\ &= \frac{\boldsymbol{x^{k+n}e^{-x}}}{\boldsymbol{(k+n)!}}\left[\mathbf{1}\right. \\ &\quad \left. + \sum_{\boldsymbol{i=1}}^{\infty} \frac{\boldsymbol{x^i}}{\boldsymbol{(k+n+1)(k+n+2)\cdots(k+n+i)}}\right] \\ &\leq \frac{\boldsymbol{x^{k+n}e^{-x}}}{\boldsymbol{(k+n)!}}\left[\mathbf{1} + \sum_{\boldsymbol{i=1}}^{\infty} \frac{\boldsymbol{x^i}}{\boldsymbol{(k+n+1)}}\right] \\ &\leq \frac{\boldsymbol{x^{k+n}e^{-x}}}{\boldsymbol{(k+n)!\,[1-x/(a+n+1)]}}. \end{aligned} \tag{4.22}$$

The result converges quickly when $x - k$ is low. But with high values of $x - k$ the number of applications of the gamma cumulative recurrence probability relation (4.20) may converge too slowly. For these cases the computation of upper tail $Q_G(x, k)$ is more convenient because less iterations are needed to get the result with specific accuracy. In this case the result must be subtracted from one to get the lower tail value. The recurrence relation (equation (4.23) of upper tail for computation methods can be derived from the recurrence relation of lower tail (4.20 (Reeve 1986):

$$\boldsymbol{Q_G(x, k)} = \frac{\boldsymbol{x^{k-1}e^{-x}}}{\boldsymbol{\Gamma(k)}} + \boldsymbol{Q_G(x, k-1)}. \tag{4.23}$$

A tool using Pearson's chi-square test is for example *WEGO* (Ye et al. 2006).

4.2 Permutation methods

Permutation methods use shuffling of properties (annotations) between studied entities. Shuffling repeats many times for having good confidence of test results which

results in utilization of much computational power. By these methods enrichment significance may be tested by observing how often overrepresentation of annotation occur in shuffled samples.

4.2.1 Permutation test

The basic permutation test (Moustafa n.d.) consists of several steps. The value determining difference between two samples is computed. Samples are combined to one collection of values. In the example with variant annotations we would combine the sample variations and the remaining background variations. Afterwards the original samples are recreated by random non-replacing selection of variations from combined samples. Again the value determining difference between samples is calculated. Recreating original samples and calculation of difference between them is repeated many times, for example 1,000. The p-value is the proportion of generated samples which have the same or more extreme value of difference than difference between original samples. In case of testing overrepresentation with mentioned variant annotations the difference would be calculated from counts of annotated variants in the sample and the remaining background variants sets. However this method is computationally demanding and moreover the computed p-value is not exact.

4.2.2 Kolmogorov-Smirnov test

The *Kolmogorov-Smirnov (K-S) test* (Massey 1951) is useful for analysing variation or gene data which are ranked. The data can be ranked by their correlation to phenotype or by measure of gene expression. The K-S test is based on determining if the cumulative distribution of sample data corresponds to some reference cumulative distribution. If yes their difference should be a random walk. The maximum difference between these distributions characterizes how much the sample data are deviates from the reference distribution. One of the advantages of this test is that it is possible to visualize the results graphically (Figure 4-1).

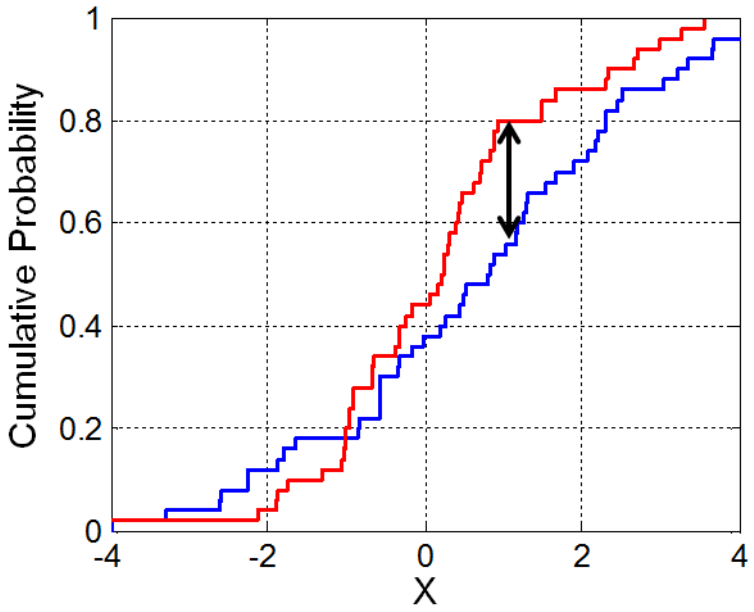


Figure 4-1. Kolmogorov-Smirnov test (By Bscan (Own work) [CC0], via Wikimedia Commons)

In enrichment analysis, this can be useful to test whether a sample of variations or genes differs from some test set. By repeating this process and creating the new test set by random shuffling the significance of enrichment can be determined similarly like in permutation methods. The Kolmogorov-Smirnov test is part of *Gene Set Enrichment Analysis (GSEA)* (Clark & Ma'ayan 2011) used for testing if set of genes are associated with an annotation or a biological category. The analysis processes genes expression data from case and control samples. The genes are ranked by differences between average measures of expression in case and control samples. Top genes are those which have less expression and bottom genes are those which have higher expression in case samples in comparison to control samples. The enrichment computation starts by performing random walk on the set of ranked genes. If the gene is part of the subset, then it is added $\sqrt{\frac{N-G}{N}}$, otherwise it is subtracted $\sqrt{\frac{G}{N-G}}$. N denotes a number of all tested genes and G denotes number of annotated genes. The maximum deviation from zero is stored. Afterwards the samples between case and control group are shuffled and a random walk is computed again. This repeats many of times. The more times are deviations from zero in the random walks less than deviation from zero in the random walk of the original set, the more the enrichment of differentially expressed genes is significant.

4.3 Methods using continuous values

Variations or genes can have assigned values indicating measure of correlation with phenotype or in case of genes measure of expression and these measures were used in the K-S test to rank genes. The following methods use these continuous values and give scores computed according to basic statistical measures of data like mean and standard deviation.

4.3.1 Student t-test

The student t-test (Rice 2007) is suitable for determining significance of means difference of two data sets. For example, it is suitable for testing expression data representing expression in two various conditions, such as temperature change. The T-profiler tool (Boorsma et al. 2005) use this test to evaluate enrichment significance of common motifs, transcription factors or GO categories. For each annotation the t-value is computed (equation (4.24):

$$t_G = \frac{\mu_G - \mu_{G'}}{s\sqrt{\frac{1}{N_G} + \frac{1}{N_{G'}}}}, \tag{4.24}$$

where according to equation (4.25:

$$s = \sqrt{\frac{(N_G - 1) \cdot s_G^2 + (N_{G'} - 1) \cdot s_{G'}^2}{N_G + N_{G'} - 2}}, \tag{4.25}$$

and where G is a set of genes with annotation, G' is a set of remaining genes, N_G is a number of genes in set G, μ is data (expression) mean and s^2 is pooled variance estimate of data. The p-value is obtained by applying the t-score to the cumulative probability distribution of t-distribution while using degree of freedom $N_G - 2$. T-distribution with n degree of freedom is defined in equation (4.26:

$$f(t) = \frac{\left[\frac{(n+1)}{2} - 1\right]!}{\sqrt{n\pi}\left(\frac{n}{2} - 1\right)!}\left(1 + \frac{t^2}{n}\right)^{-\frac{n+1}{2}}. \tag{4.26}$$

4.3.2 Z-score

Z-score indicates how much the value x deviates from a distribution with mean μ and standard deviation σ. It is computed by equation (4.27:

$$z = \frac{x - \mu}{\sigma}. \tag{4.27}$$

A feature of distribution of sample values z-scores is that it approaches to normal distribution. One of the methods using Z-score is described in (Gupta et al. 2007). This method integrates genes p-values from association experiments. Therefore bias from false-positive genes influence is reduced. Genes' p-values are transformed to the random deviates z_i by calculating from equation (4.28:

$$z_i = \Phi^{-1}(1 - p_i). \tag{4.28}$$

where Φ is the standard normal cumulative distribution function. From the deviates z_i the z-score for annotation term $\mathcal{F}$ is computed by equation (4.29:

$$z = \sum_{i \in \mathcal{F}} z_i / \sqrt{|\mathcal{F}|}. \tag{4.29}$$

where $|\mathcal{F}|$ denotes number of genes containing term $\mathcal{F}$. The integrated p-value is given by reverse transformation of z-score (equation (4.30):

$$p_{\mathcal{F}} = 1 - \Phi(z). \tag{4.30}$$

4.4 Multiple test corrections

As during enrichment analysis (or association tests in GWAS as well) thousands or more single tests may be performed some false positives may occur when some true null hypotheses are rejected. Many methods were developed to address this issue and to provide better insight to consider significance and usability of results. The multiple test corrections should be applied to experiments consisting of a large number of significance tests.

4.4.1 Bonferroni correction

The basic method of correcting significance results is Bonferroni correction (Bonferroni 1936). The idea is to divide the significance threshold by the number of single association tests performed. In case of 500,000 tests the threshold would be adjusted from the common value 0.05 to 10^{-7}. This application provides statistical significance that there is no false positive in multiple tests. The correction is considered as too conservative as it reduces statistical power and may result in producing false negatives of statistical analysis so that some untrue null hypotheses may not be revealed.

4.4.2 False discovery rate

Bonferroni correction is efficient when it is needed to prevent from any false rejection of the true null hypothesis. However, often it is possible to allow a certain proportion of null hypothesis rejections to be false while capturing more untrue null hypotheses. In practice the false positives can be often eliminated in further research of results. The proportion of these false positives is termed as *False discovery rate (FDR)* (Benjamini & Hochberg 1995). The common method for determining significant results according to required FDR is the *Benjamini-Hochberg (BH) method* (Benjamini & Hochberg 1995). At first this method orders in ascending rank the P values of performed tests. The lowest p-value is P_1 and the highest is P_m where m refers to the number of tested null hypotheses. Then it is determined for which the highest i applies the following equation (4.31:

$$P_i \leq \frac{i}{m} q, \tag{4.31}$$

where q is the acceptable level of FDR. All the P-values P_i, $i \leq k$, are set as significant.

Sometimes the computation of BH adjusted p-values is performed based on the equation (4.32:

$$P_i' = \begin{cases} \min\left(\frac{m}{i} \cdot P_i, P_{i+1}'\right), i < m \\ \frac{m}{i} \cdot P_i, i = m \end{cases} . \quad (4.32)$$

Though values of adjusted p-values range in the interval of [0,1] like standard p-values they do not have an exact meaning. They provide another way of determining significant results. When adjusted p-value P_i' is less than q, then original p-value P_i is according to value q significant.

As an example the following list of p-values from an experiment can be used:

0.7590, 1.0000, 0.0278, 0.0019, 0.0298, 0.0459, 0.0344, 0.0201, 0.0095, 0.0001, 0.0004, 0.4262, 0.6528, 0.5719, 0.3420

It is needed to reorder the p-values and compare them with values of $(i/m) \cdot q$ as it is shown in following Table 4-3. Also the adjusted p-values are in the last column. It is supposed that acceptable FDR is set to $q = 0.15$:

Rank	**P-value**	$(i/m) \cdot q$	**Adjusted P-value**
1	0.0001	0.0100	0.0015
2	0.0004	0.0200	0.0030
3	0.0019	0.0300	0.0095
4	0.0095	0.0400	0.0356
5	0.0201	0.0500	0.0603
6	0.0278	0.0600	0.0638
7	0.0298	0.0700	0.0638
8	0.0344	0.0800	0.0645
9	0.0459	0.0900	0.0765
10	0.3420	0.1000	0.5130
11	0.4262	0.1100	0.5812

Rank	P-value	$(i/m) \cdot q$	Adjusted P-value
12	0.5719	0.1200	0.7149
13	0.6528	0.1300	0.7532
14	0.7590	0.1400	0.8132
15	1.0000	0.1500	1.0000

Table 4-3. BH method of determining significance. Rows with significant p-values according to FDR of q=0.15 are underlined.

When accepting that 15% of significant p-values may be false rejections of true null hypothesis the 9 tests are determined as significant. In Bonferroni correction is possible to suppose only that no false positive is acceptable. For example, a threshold of 0.15 would be adjusted to 0.01. In this case, only the first four p-values would be significant.

5 VARANTO TOOL

As part of this thesis a novel web tool *Varanto* is created. Varanto is a tool for annotating variations according to various annotation sources. It shows information about variations and annotations and visualizes associations between them. On given input set of variations enrichment analysis is performed using hypergeometric test. Vast amount of background data are prepared for Varanto. Tens of thousands of annotation terms in many annotation classes are accessible to be analysed for overrepresentation and underrepresentation. Moreover Varanto associates input variations with near genes and their annotations. The selection of predefined background sets for enrichment analysis is available. The input set of variations can be filtered by distance filter by a selectable distance parameter so in case of many variations near to each other only one of them is considered in analysis. This can prevent from bias toward annotations related to a genomic region containing more variations in an input set. Varanto provides a possibility to select whichever combination of annotation classes for analysis and it is possible to obtain several types of analysis results.

5.1 Varanto functionality

Varanto provides results for given input reactively so whenever the input changes all dependent output information is also updated. This provides interactive interface for setting input variations, adjusting input parameters and viewing the results. Varanto provides multiple types of results to show (Figure 5-1).

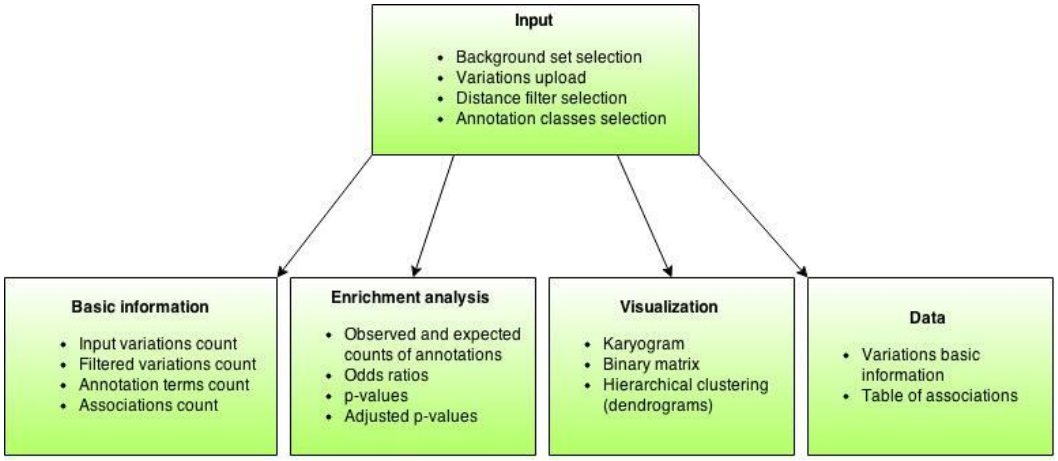


Figure 5-1. Schema of Varanto functionality

Firstly, the basic information about given input variations and selected annotation classes are shown, particularly the number of recognized variations in input set according to selected background set, the number of filtered variations by distance filter, the number of annotation terms in selected annotation classes and the number of all found associations of input filtered variations to annotation terms of selected annotation classes.

Secondly, for determining over or underrepresentation the observed and expected counts of terms are shown with calculated odds ratio. The p-values determining significance of terms are viewed along with their adjusted values computed by the BH method. These results are collected to a browsable data table which is also sortable by any of these annotations data columns. Description of annotation terms, if available, are integrated to the data table. The table interface allows to filter table rows by given input string. The entire table is available for downloading in csv format.

Thirdly, visualization of associations are shown. This feature supplements enrichment analysis in finding new biological knowledge. This makes the tool unique. The visualization is in the form of a rendered binary matrix where the rows of the matrix represent variations and columns represent annotation terms labels. The cell value is either zero when there is no association between a variation and a term or one when there is an association. Rows and columns are ordered by hierarchical clustering so patterns of variations or terms with similar associations are observable. This can be useful in identifying clusters of variations which may be connected with individual biological processes or clusters of annotation terms with similar associations. The visualization is

possible to be adjusted, for instance the rows' and columns' dendrograms visibility are configurable or the look of visualization can be optimized. It is available for downloading as a pdf file as well. Also the karyogram depicting locations of variations in chromosomes is possible to be shown.

Finally, it is also possible to be render the binary matrix of associations as a data table with sorting and filtering features like in the case of annotation terms enrichment data table. The binary matrix data table is also available for downloading in csv format.

The described features are accessible through a web interface with an intuitive graphical user interface (GUI). Employing created custom background database allows to provide results quickly for most of scenarios what enables effective usage in genetics research. To build the background database data from various sources are needed to be imported, transformed and stored to database. This is because of huge amount of source data extensive and computationally-demanding process which is supposed to be repeated in longer periods of time, for example once per few weeks or months. Nevertheless, sufficient computational power and suitable implementation are necessary to perform this process in reasonable time.

5.2 Implementation tools

For developing the Varanto, importing data and building the background database certain materials and methods are used. For data, main data source is Ensembl which is accessed via MySQL interface or *BiomaRt* package in R. The *homo_sapiens_variation.variation* MySQL table is queried for list of variation identification labels and Ensembl Variation and Ensembl Genes databases in BiomaRt are used for collecting variation and gene annotations of many annotation classes. The additional source for variation annotations is the *GET-Evidence* tool (Ball et al. 2012) and for genes the *Molecular Signature Database (MSigDB)* (Subramanian et al. 2005). All annotation sources are listed in Table 5-1 and annotation classes which the annotations are imported of are listed in Appendix 1.

Data	Sources
Variation identifiers	Ensembl
Variation annotations	Ensembl GET-Evidence
Gene annotations	Ensembl Molecular Signature Database

Table 5-1. Data sources used to building the Varanto background database.

A combination of several programming languages and development software was used to implement data import, database and *Varanto* itself as web application. The development was running under the CentOS 6.6 operating system.

5.2.1 Bash

Bash (Fox 2014) is the shell which is originally a part of the GNU operating system and currently it is used in many Unix operating systems. It provides many built-in commands performing tasks like operating system commands, manipulating and operations with files, processing files or launching scripts or programs. The commands may be pipelined or executed within another command. They can be run in background as separate process. It is also a scripting language which may be used for creating custom scripts containing logic like in other programming languages.

5.2.2 Python

Python (Python Software Foundation 2015) is a multiplatform object-oriented interpreted programming language intended to be easy and its code to be readable. Built-in data types like lists, sets and dictionaries and their interfaces provide flexible ways to manipulate the data. Therefore Python is used for processing and transforming downloaded variation and gene data which are afterwards ready to be copied to database. For programming in Python the Eclipse developing tool (The Eclipse Foundation 2015) was used.

5.2.3 PostgreSQL

As a database for storing background data of Varanto *PostgreSQL* was adopted (The PostgreSQL Global Development Group 2015). It is an object-relational database management system supporting features for data management in relational tables. The key features useful for Varanto are indexes, primary and foreign keys for efficient retrieving data from huge quantity of information. The primary keys are automatically indexed. The project of PostreSQL is open-source and has a wide community able to provide quick support. Easy definition of database schema can be done by the *SQL Power Architect tool* (SQL Power Group Inc. 2015) which was used to definition of Varanto background data schema.

5.2.4 R

R (The R Foundation 2015) is the free highly extensible GNU programming language aimed mainly on statistical operations and graphics. It is optimized for computations on arrays and matrices and writing code for manipulating them is easy and effective as well as plotting graphs or images. The extensiveness is achieved by the ability to create custom functions like in other programming languages or easy importing of external packages.

The *BiomaRt* (Durinck et al. 2005) package allows to download genomics data from Ensembl or other biomart databases. Its function `getBM` downloads data with their specified attributes and filtered by the specified filter data in the given database.

Another package which was adopted in Varanto is *dplyr* (Wickham & Francois 2015) which makes the database operations easy and returns query results in `data.frame` what is a common data type in R. Selection, filter, join, order and other operations are flexible to write and they have good performance.

Function `heatmap.2` of package *gplots* (Warnes et al. 2015) whose purpose is rendering heatmaps is used for generating graphical visualization of binary matrix of associations. Although the heatmaps are suitable for generating data with continuous values this function is also suitable for binary matrix visualization.

For the development of *R* scripts for downloading variation and gene data using Biomart and also for the development of Varanto web user interface (see next subchapter) *RStudio* integrated development environment (RStudio 2014) was applied to.

5.2.5 Shiny framework

A powerful method for creating web pages providing a user interface (UI) is the Shiny framework (Chang et al. 2015). It is another R package and programming of UI is done in R so it is easy to integrate with existing R functions. The definition of UI is performed by the `shinyUI` function which is after server initialization compiled to HTML file. Shiny provides custom functions to define layout though it still makes possible to define parts of the UI by HTML tags. The reactions on input and preparing data for output are specified by the `shinyServer` function. The input and output controls (widgets) have unique identifiers defined in the `shinyUI` function which are used in `shinyServer` for referencing the input controls to define reactions and for the output controls to define their data source.

The key feature of Shiny intensively used in Varanto is its reactiveness. It means that output is automatically updated whenever any of the input controls, which the output control depends on, changes. An expression defining output control data source can contain references to some input controls providing a value. The output control will be automatically updated each time the value of any of the input controls changes (Figure 5-2).

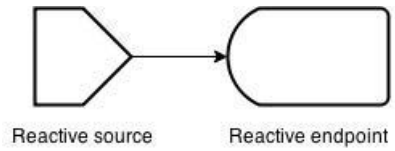


Figure 5-2. A Shiny simple program with one output depending on one input. Input is reactive source and output a reactive endpoint.

However, some functions may sometimes be used by more output controls while using the value of the same input controls as parameter. When a certain input control value is changed then such functions are called multiple times while computing the same stuff. This can be prevented by the `reactive` expression. Similarly like output definitions they automatically react to change of all input controls which are part of the expression. Their key aspect is that they remember their computed values until the next

change of any input control whose reference they contain. So if they wrap some function taking input control value as parameter and which is called by more output controls the function is executed after input control value change only once. The reactive expressions can be nested so it is possible to design a Shiny application in a way that no redundant computations are performed (Figure 5-3).

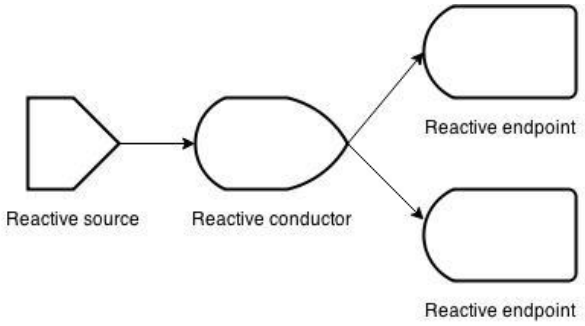


Figure 5-3. A simple reactive program including a reactive conductor defined by a reactive expression.

There are some cases when output control data definition or reactive expression use value of some input control but it is not needed or desirable to re-evaluate them reactively every time after its change. In Varanto this requirement arises when it is intended to submit the input control value by a button. The expression `isolate` ensures that changes of input controls of contained references are not causing re-evaluation of parent expression. The reactions on buttons are defined by `observe` expressions.

5.3 Preparing background database

For building background database with data several steps consisting of operations of downloading, transforming, processing data are performed. It is necessary to perform these import operations before running the web server of Varanto. The background database is designed to achieve that Varanto web interface can retrieve data in appropriate form. It is intended that the import process is repeated by user after some period of time, for example, when variation or gene sources are updated.

5.3.1 Master import script

To perform data import to background database easily all operations are run by the master script written in Bash. The import process is divided into 8 steps run by custom

script (Figure 5-4). The master script provides an interface for running all steps at once or only some of them. Parameters of individual scripts performing each step are possible to set in a master configuration file to change them easily.

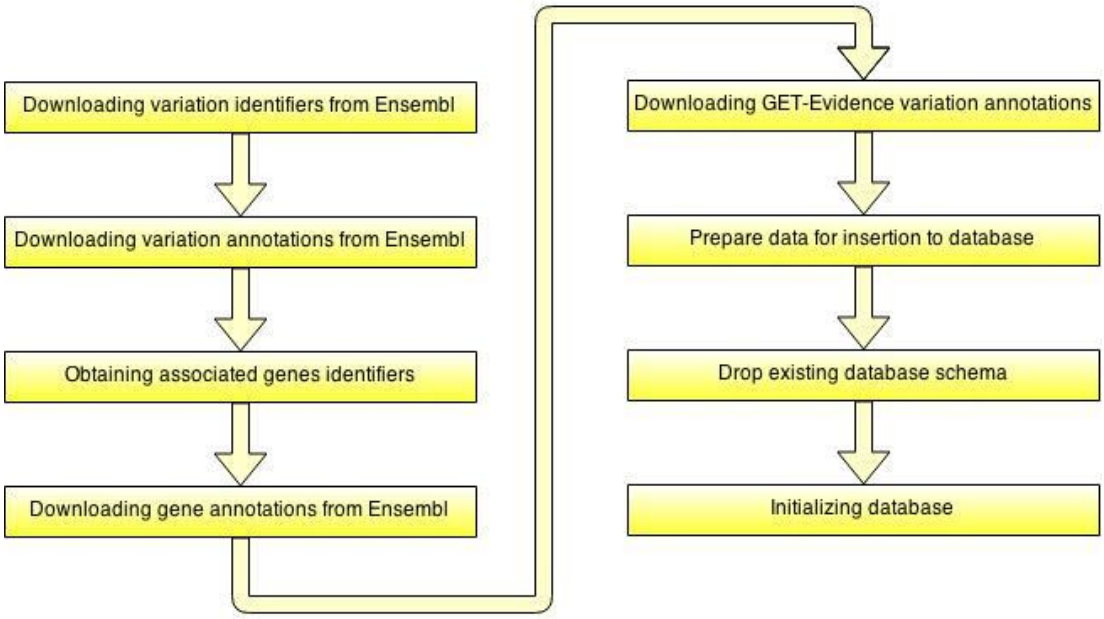


Figure 5-4. Import to background database.

During the first step, **downloading variation identifiers from Ensembl**, identifiers of all available short variations from Ensembl database are downloaded using its MySQL interface.

During the second step, **downloading variation identifiers from Ensembl**, annotations of variations of each of the identifiers are downloaded. This is achieved by running R script using BiomaRt package where the `getBM` function use list of variation identifiers as filter. The list of identifiers are chunked so function is called repeatedly for each chunk. The basic variation data of allele, chromosome, chromosome start position and chromosome strand attributes are downloaded along with functional data which are listed in Appendix 1. To speed up downloading the R script is called concurrently in separate processes while list of identifiers is split using Unix command `split`. Synchronization of processes is ensured by `wait` command which simply waits for all of the running processes to be completed.

Within the functional data downloaded in the second step are gene identifiers associated with variations. These associations are based on overlap with genes or proximity

to genes. The third step, **obtaining associated genes identifiers**, is extracting the list of gene identifiers which were associated with any of the variations. This is done by extracting the column with gene identifiers from variation annotations and making these identifiers unique.

During the fourth step, **downloading gene annotations from *Ensembl***, annotations of genes using identifiers from previous step are downloaded. This step is performed in the same manner as the second step.

During the fifth step, **downloading *GET-Evidence* variation annotations**, additional variation annotations from *GET-Evidence* web site using `wget` are downloaded.

The sixth step is **preparation data for insertion to database**. Because `INSERT` database operation is for inserting huge data overwhelming it is needed to prepare files containing complete table data and use `COPY FROM` PostgreSQL command to load them to database. The data preparation are performed by Python script and the algorithm will be described below.

The seventh step is **dropping existing database schema** as it is supposed that import might replace old data, Import can be also run for purpose of the reloading data because of performed changes in the database schema.

The final step, **initializing database**, is inserting data to database which schema will be described in the next subchapter. At first tables without any constraints are initialized. Indexes are also not yet set up. Afterwards the prepared data are inserted to database by `COPY FROM` commands and finally primary and foreign keys and indexes are set up.

5.3.2 Background database schema

The schema is designed to possibility of retrieval variation and annotation data as well as associations of annotations to variation. For computing enrichment significance annotations counts in each background sets are available. There are information whether variations are part of background sets. Following Figure 5-5 depicts database schema:

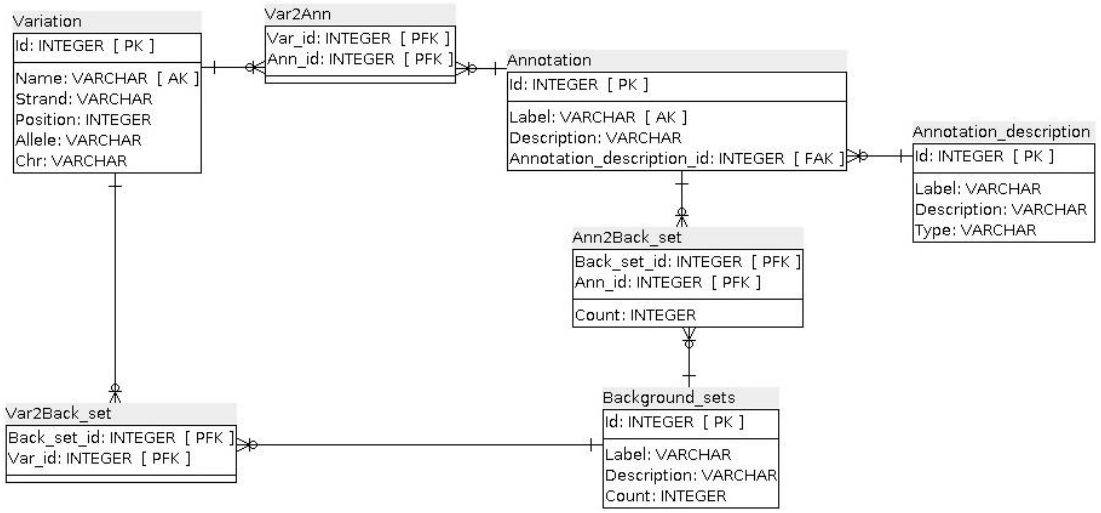


Figure 5-5. Background database schema designed in SQL Power Architect (SQL Power Group Inc. 2015)

`Variation` table contains basic information about variations like their unique name, strand, starting position, alleles and chromosome which they are located in. Besides primary key index additional index is created for name column as it is expected that user inputs any set of variation names in Varanto interface. The next two-column ascending index is built also for chromosome and starting position column and this is useful for distance filter which is simpler to apply when database returns variations in order by chromosome and starting position.

`Annotation_description` table contains information about annotation classes like their label, description and type. The type can have values *variation* or *gene* depending on whether the annotation class is gene or variation annotation class. The data for this table are created manually.

Each row representing annotation term in **`Annotation`** table contains annotation labels and their description. Moreover there is column for annotation class description reference identifier which serves as foreign key for `Annotation_description` table. The combination of `label` and `annotation_description_id` is unique and they are indexed together while `annotation_description_id` column is the first part of index. This is useful as user can query for annotations of certain annotation classes through Varanto interface. Adding `label` into index may be useful for effective retrieval of annotations of specific labels.

The Variation and Annotation tables are joined through **Var2Ann** table which models M:N relationship between those tables. The Var2Ann table contains references to primary keys of Variation and Annotation tables as foreign keys so this table describes associations of annotations to variations.

Background_sets table contains information about manually downloaded background sets. For each background set its label, description and count of its variations is defined. Except the count of variations these data are created manually. The count of variations is calculated during preparation of data to insertion to database.

Var2Back_set table contains references to primary keys of Variation and Background_sets tables so this table describes which variations belong to each background set. This table also models M:N relationship. Although set of all downloaded and inserted variations are considered as a separate background set as well this table does not contain rows describing association of variation with this background set. So when querying for variations of background set of all variations it is needed to query the Variation table without joining Var2Back_set table. For other background sets it is needed to join Var2Back_set table.

In **Ann2Back_set** table for each annotation term there is its count in variations of each background sets. It contains references to primary keys of Annotation and Background_sets and models M:N relationship but unlike Var2Ann and Var2Back_set tables this table contains table records for all combinations of records of joined tables. If there is not any annotation of term in variations of a background set the count is set to zero. Additional two-column index is set for count and background set identifier columns while count index is set as ascending for purpose of fast obtaining of annotation terms with count higher than zero.

For optimizing database performance fill factor property is set to 100% in all tables and indexes which means that there are no additional empty space in table or index pages. This is desired setting as it is not intended to modify data in the database after their import.

The indexes in several tables are clustered for better accessibility of data of all columns after filtering data based on index. In Variation table the name column index is

clustered, in `Annotation` table the index of annotation class reference and annotation term label columns, in `Ann2Back_set` table the index of count and background set reference columns and in `Annotation_description` table as well as in `Background_sets` table the primary key index.

5.3.3 Algorithm of data preparation

After downloading all annotation data it is needed to process and transform them to create files with annotation and variation data representing database table data from schema described above. The main part of the data preparation is processing of variant annotations file line-by-line. Each line contains basic information about variation with its functional annotations. Single functional annotations are after their loading represented by tuple containing annotation class label the annotation term belongs to, annotation label itself, description of annotation term and type of annotation class.

For initialization of data processing several procedures are performed. Firstly, background data sets' files are read. Their variations are stored to hash sets so it is possible to quickly test whether any variation is part of a background data set. Secondly, annotation class descriptions are loaded from manually created file. Dictionary for getting a class identifier from class label concatenated with type of annotation class is created. The concatenation is performed because some annotation classes may have same name while one of them is variation annotation class and another is gene annotation class. The dictionary is used when filling out `annotation_description_id` column for each annotation term record. Thirdly, the genes' annotations are loaded. A dictionary which returns gene functional annotations from their Ensembl identifier is built. Gene annotations are obtained by processing line-by-line genes annotation file downloaded using biomaRt package from the fourth step of import process. In each line the Ensembl identifier of gene is read and all functional annotations in that line are stored for this gene. Another data source for gene annotations is MSigDB. Data from MSigDB are arranged in files in which each line represents certain annotation term and contains list of all genes which are associated with this term. The gene identifiers are HGNC symbols. These are converted to Ensembl identifiers by a dictionary which is created during processing the annotation file downloaded by biomaRt package as the file contains also the HGNC symbol along the gene functional annotations. Finally,

the GET-Evidence variation annotations are processed which are arranged in file in similar manner like data from biomaRt package.

When initialization procedures are completed the main part of processing is started. It processes variation annotations from file obtained from the second step of import process. It is supposed that multiple lines contain annotations for same variation and these lines are clustered and that some annotations can occur in multiple lines for the same variation.

Variations which are not on the standard track of genome in Ensembl are skipped. This filtering is performed by checking the chromosome column whether it contains single number of chromosome or *X*, *Y* and *MT* of sex and mitochondrial chromosomes. When line with new variation occurs this variation with its basic data is written to `Variation` table file and it is determined which background sets contain it. Also the pre-processed GET-Evidence variation annotations are obtained. Variation allele count is incremented which is needed to obtain the most common alleles after the processing is finished. Also the variation with its allele is appended to a list which is used in the end to add annotations of most common alleles. The most common alleles is treated as separate functional annotation as the number of all alleles in the background set of all variations is so high (more than 400,000) that enrichment significance computation lasts too much time. Using the common alleles instead of all alleles allows user to get results faster while uncommon alleles are skipped.

Each line of variation annotations including GET-Evidence annotations and except the variations on the non-standard tracks is processed. If annotation label is found the first time for current variation the record in `Annotation` table file is created while using data from the tuple representing annotation. The description of annotation often contains slash characters '/' which can cause that database could interpret it with following character as escape so all slash characters are doubled to prevent this. The `annotation_description_id` is obtained from pre-processed dictionary of annotation class descriptions. Then variation and annotation term identifiers are used to write record of `Var2ann` table while it is prevented from multiple write of same combination of variation and annotation term in case that annotation term occurs in multiple lines

of the same variation. When Ensembl gene identifier is available in variation annotations also the pre-processed gene annotations are written to `Annotation` and `Var2ann` table files. The identifier of variation which is associated with gene is used for recording associations between gene and annotation term. This associates gene annotations to variations which are near to or overlaps with genes. Variation and annotation identifiers are generated for each new variation or annotation term. When writing record to `Var2ann` table the counts of the current annotation term in the background sets, which the current variation is part of, are incremented. These counts will be useful for computation of enrichment significance.

After the all lines of variation annotations are processed the additional annotations of common alleles are processed as it is already possible to determine which alleles are the most common. `Background_sets` table file is also possible to write as it is already known how many variations are in each background set. The count of variations of the custom background sets are known already after initialization processes but count of variations of background set of all variations is known only after processing variation annotations file. Also the annotation terms counts in all background sets are stored to `Ann2Back_set` file.

The pseudo code describing the algorithm in its top level is following:

```
01. PerformInitializations();
02. current_variant_name = NULL;
03. foreach (line in variant_annotations_file)
04. {
05.   if (current_variant_name != line.variant_name)
06.   {
07.     if (!IsOnStandardTrack(line.chromosome_name))
08.     {
09.       continue;
10.     }
11.     current_variant_name = line.variant_name;
12.     variant_background_sets =
          GetBackgroundSetsWithVariant(current_variant_name);
13.     ProcessVariantAndGET-EvidenceAnnotations(
          current_variant_name, variant_background_sets);
```

```
14.   }
15.
16.   foreach (functional_annotation in line)
17.   {
18.     if (functional_annotation is ensembl_gene_id)
19.     {
20.       gene_id = functional_annotation;
21.       ProcessGeneEnsemblAndMSigDBAnnotations(
            gene_id, variant_background_sets);
22.     }
23.     ProcessVariantAnnotation(
          functional_annotation, variant_background_sets);
24.   }
25. }
26. PerformConcludingActions();
```

5.3.4 Implementation of data preparation

The algorithm described above was implemented using Python language. The class diagram of implemented script is shown in the following Figure 5-6:

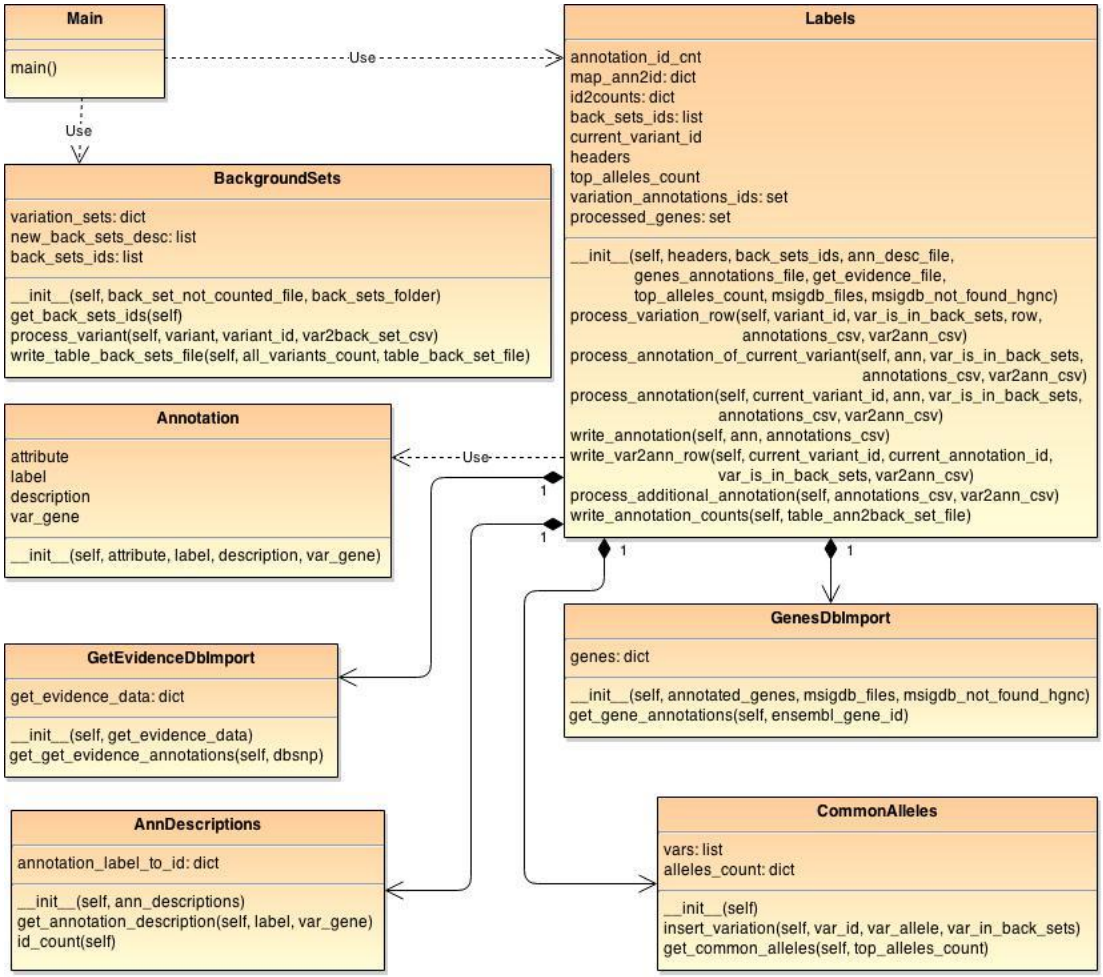


Figure 5-6. Class diagram of script preparing data for storing to database.

The main function is called when the script is started and performs the algorithm described above. It controls reading variation annotation file and writing `variations`, `annotations`, `var2ann` and `var2back_set` table files. It uses `BackgroundSets` and `Labels` classes.

`BackgroundSets` class contains and process variation background sets. Its constructor takes as parameters manually created file of `BackgroundSets` table without column of variations counts and path to folder with files representing single background sets with list of their variation identifiers. It is supposed that names of files in the folder corresponds to background sets labels in the `BackgroundSets` table file. The background sets files are read and `variation_sets` dictionary is initialized with sets of variations of background sets which ids serves as keys for the dictionary. `new_back_sets_desc` list loads data rows from `Background_sets` table file

and after processing each custom background set the count of variations in the background set is appended to this data structure. back_sets_ids list is also initialized with list of background sets primary key identifiers. process_variant method processes variation with its name and primary key identifier and according to whether the variation is part of single sets in the variation_sets dictionary adds records to the var2back_set table file. Also the method returns list of background sets identifiers which the variation is part of what is necessary to increment annotation counts of proper background sets (see Labels class below). write_table_back_sets_file is intended to be called after the variation annotation files processing is finished. This method takes as parameters number of all variations which is additionally appended to row of background set of all variations. Afterwards the list of background sets, which already contains complete information about background sets, is written to a file passed in another parameter.

Purpose of **Labels** class is to process annotation labels. Its constructor takes several parameters. headers is list of variation annotation class labels which are used for variation annotations downloaded by biomaRt package to obtain identifier of their annotation class from AnnDescriptions class (see below) to write it to annotation_description_id column. back_sets_ids is list of all background sets identifiers and it is used for initialization of records in Ann2back_set table for new annotation term. ann_desc_file is manually created Annotation_description table file and it is passed to AnnDescriptions object. genes_annotations_file is file of gene annotations downloaded by biomaRt package which is passed to GeneDbImport object and get_evidence_file is file downloaded in the fifth step of import process with GET-evidence variation annotations which is used by GetEvidenceDbImport object. top_alleles_count is number of most common alleles which will be considered as separate common allele annotation class so it is used by CommonAlleles class. msigdb_files is file with list of MSigDB files containing gene annotations. The last parameter msigdb_not_found_hgnc is list of all HGNC symbols in the MSigDB files which are not possible to convert to Ensembl identifiers during processing of annotations because no gene downloaded from biomaRt is associated with these HGNC sym-

bols. These two parameters are passed also to GeneDbImport class. write_annotation method is used for recording new annotation to Annotation table file. As parameters the tuple representing annotation is passed and Annotation table file itself. At first it is checked whether this annotation is already recorded by checking if the unique tuple representing annotation already exists in map_ann2id dictionary as key. If not it is added to dictionary with generated annotation identifier as value, the annotation is recorded to Annotation table file and also id2counts dictionary item for the annotation is initialized. In id2counts dictionary there are counts of annotation terms in all background sets so their values are initialized to zero. The keys of id2counts dictionary are background sets identifiers from back_sets_ids list. write_var2ann_row method writes new record to var2ann table file. Besides the record write itself the annotations counts in the background sets given as parameter are incremented in id2counts dictionary. process_annotation and process_annotation_of_current_variant methods calls write_annotation and write_var2ann_row methods while the process_annotation_of_current_variant method moreover prevents from multiple writing of annotation to current variation using variation_annotations_ids set which contains already recorded annotations in Var2ann table of current variation. process_variation_row method takes line from variation annotations file. If the line contains annotations for another variation than previous line the current_variant_id field is updated to identifier of new variation and variation_annotations_ids and processed_genes sets are cleared. processed_genes set prevents from processing same Ensembl gene annotation multiple times for the current variation. Also for new variation its GET-evidence annotations are obtained and its allele is processed in common_alleles class (see below). The method then processes variation and gene annotations from biomaRt package as it was described in previous subchapter, creates tuples representing annotations and writes individual records to Annotation and Var2ann table files using the process_annotation_of_current_variant method. The method process_additional_annotation is used for writing common alleles annotations after completing the processing of variation annotations file and for eachcommon allele calls process_annotation method. write_annotation_counts is

used for storing annotation counts in background sets stored in `id2counts` dictionary to `Ann2back_set` table file.

`GeneDbImport` class is used to provide gene annotations while after calling its constructor pre-processing of gene annotations is performed. As the result the `genes` dictionary with keys as Ensembl gene identifiers and values as tuples representing genes annotations is created. Method `get_genes_annotations` provides annotations for given Ensembl identifier by reading from `genes` dictionary.

Similarly class **`GetEvidenceDbImport`** provides interface for getting variation *GET-evidence* annotations where `get_evidence_data` dictionary is built during pre-processing after calling class constructor and annotation tuples are then possible to obtain by `get_get_evidence_data` method.

`AnnDescriptions` class provides conversion of the annotation class label concatenated with type of annotation to annotation class identifier. The conversion is performed by `annotation_label_to_id` dictionary which is used to return annotation class identifier in `get_annotation_description` method.

`CommonAlleles` class processes individual variations by `insert_variation` method. The individual variations are stored with their allele and background sets they are part of in `vars` list. In the end of the processing of the variations is called `get_common_alleses` method which orders the alleles counts and returns the variations from `vars` list with their common allele annotation. If their allele does not belong to most common alleles their common allele annotation is *uncommon*.

`Annotation` class serves as structure type for items of tuple representing annotation.

5.4 Varanto implementation

Varanto web tool is implemented in Shiny network. The UI is implemented by `shinyUI` function, interactions with UI in `shinyServer` function. The set of functions providing communication with database or computation tasks was implemented. The queries to database are performed by dplyr package. Layout is defined as sidebar

layout while sidebar contains input panel and main panel contains tab panels with output controls. In the following text the tool capabilities are described along with implementation.

5.4.1 Input panel and main tab

The first tab panel *Main* contains summary information about input variations and annotations.

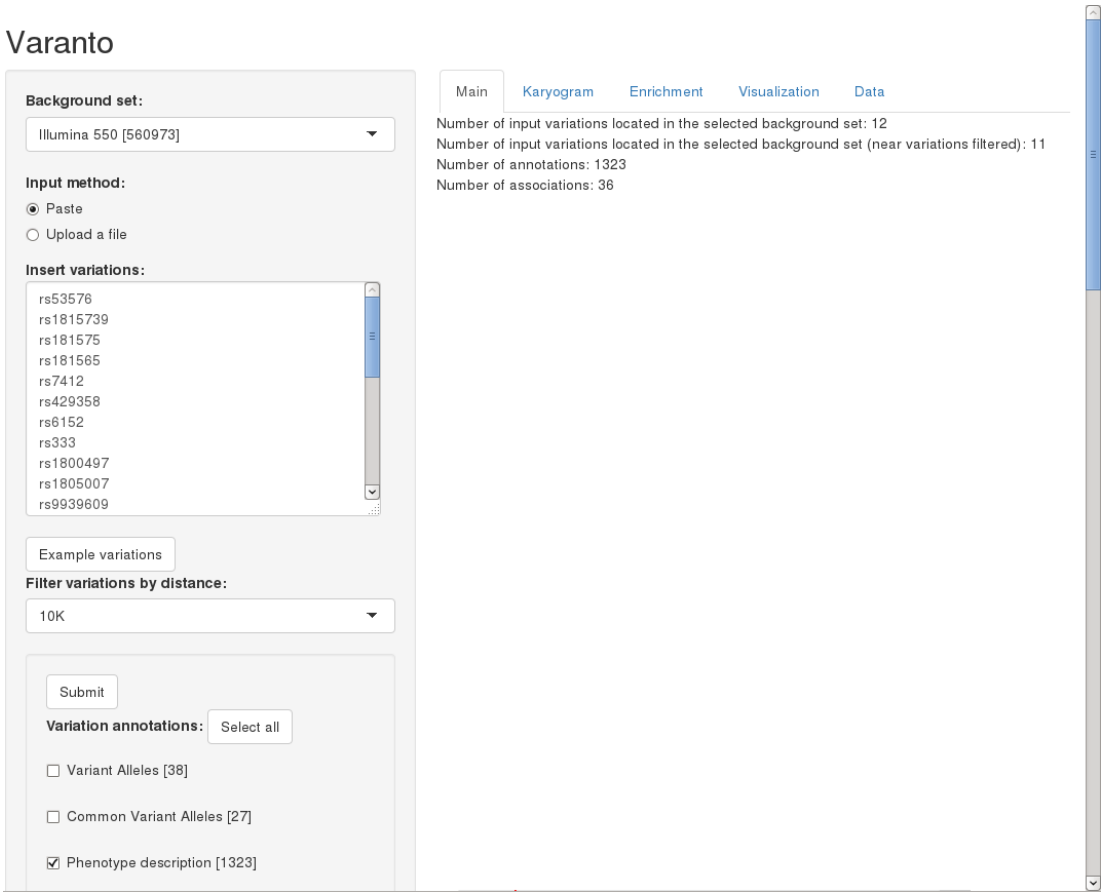


Figure 5-7. Main panel of Varanto tool

As it is seen in the Figure 5-7, Illumina 550 variations are used as background set. The number in the brackets is the number of variations in the set. Background set selection options are loaded automatically from database each time the Varanto server starts. Function `get_back_set` loads background sets by selecting entire `Background-Sets` table.

Distance filter is set to 10 kilobases so if multiple variations are located to each other in less than this distance only one of them will be considered.

Phenotype description variation annotations are chosen for analysis but any combination of annotation classes can be chosen by user. The numbers in brackets next to the annotation class labels are counts of their annotation terms associated at least to one variation in the chosen background set. The input checkbox controls for annotation classes are also generated automatically when Varanto server starts. Annotation classes are loaded from database by function `get_ann_desc` while function `get_var_ann_desc` filters classes which are variation classes and `get_gene_ann_desc` filters classes which are gene classes. The counts of annotations in single annotation classes for each background set is computed from database by `get_total_count_of_annotations_in_back_set` function. This function joins `Annotation` and `Ann2back_set` tables by `inner_join` function of dplyr package but rows in `Ann2back_set` with count equal to zero are excluded before joining. Then `back_set_id` and `annotation_description_id` columns are selected from joined tables. From these two columns the contingency table using R function `table` is created. As a result this table contains counts of annotations for each combination of background set identifier and annotation class identifier which are used to fill out numbers in brackets next to the annotation class labels.

The *Main* tab panel shows basic information about input variations and selection of annotation classes. The first information is number of variations from input which are part of the chosen background set. This number may be lower than number of input variations when some input variations are not part of the chosen background set. The variations of the chosen set are obtained from database by joining `Variation` and `Var2Back_set` tables in function `get_variations_of_names_of_back_set`. Function `semi_join` of dplyr package is used to get variation rows for which exists association with chosen background set in `Var2Back_set` table.

The next information is number of variations filtered by distance filter. In the Figure 5-7 it is seen that one variation was filtered. Distance filter is implemented by function `filter_variations`. It is supposed that variations are ordered by chromosome and position what is already ensured in `get_variations_of_names_of_back_set` function using the arrange function of dplyr. The algorithm used in this function creates window of desired size (for instance 10

kilobases) starting from the first variation. All variations which are inside this window except the first one are excluded. The next window is created from the first variation which is outside of the previous window. This repeats until all variations are processed and for the next operations only the not excluded variations are considered.

Annotation terms are obtained by `get_annotations_of_desc_id_of_back_set` function which performs join operation of `Annotation` and `Ann2Back_set` table using `inner_join` dplyr function. In case of custom background set the rows with count of zero are excluded before the join. The count of annotation terms in the output is sum of counts in brackets of selected annotation classes.

Finally number of associations are shown. The associations between variations and annotations are obtained by `get_association_pairs` which returns pairs of variation and annotation identifiers loaded from `Var2ann` table.

The reactivity graph of Main tab is shown in Figure 5-8.

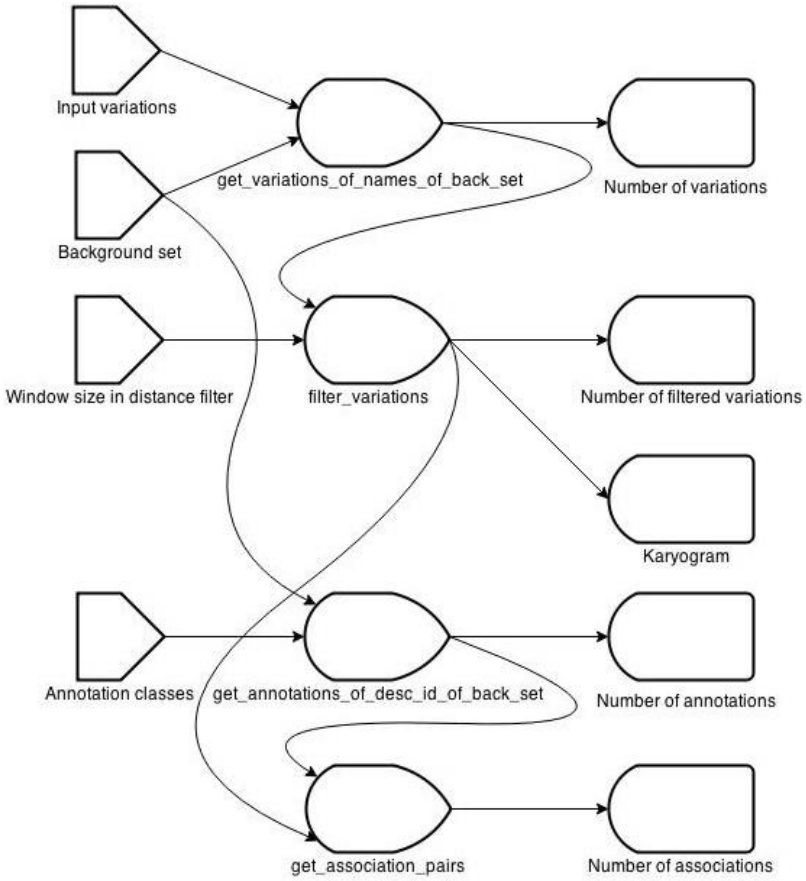
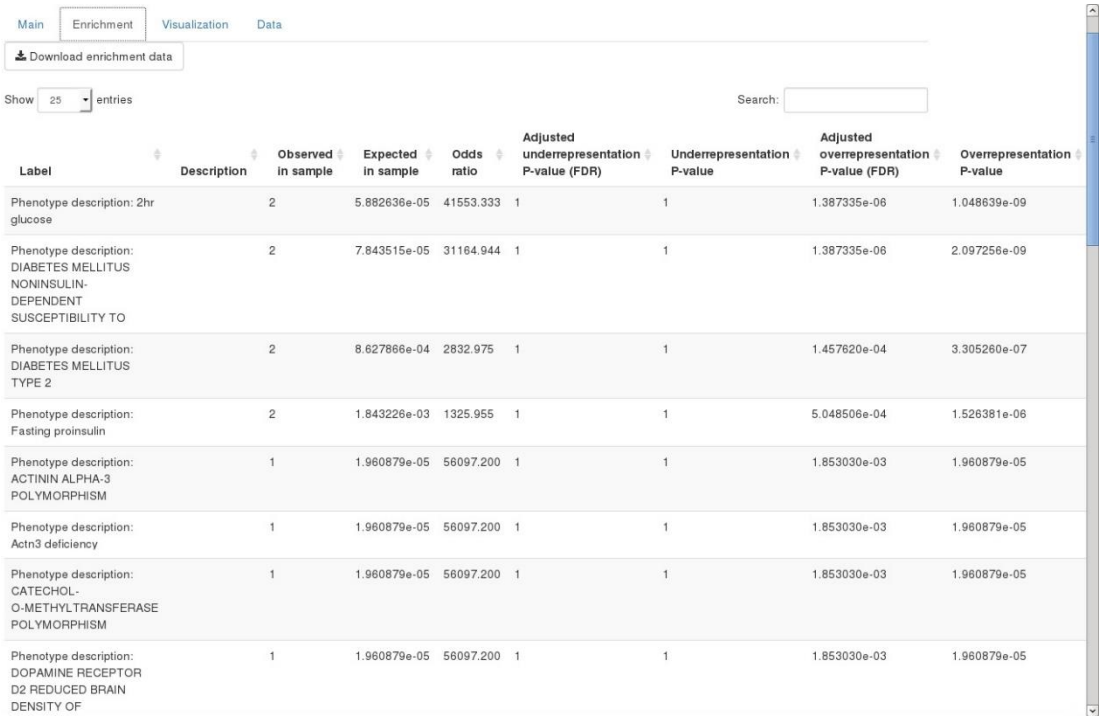


Figure 5-8. Reactivity graph of Main tab.

5.4.2 Enrichment tab

Enrichment tab shows table where each row contains values of observed and expected counts, odds ratio, over and underrepresentation significance expressed as p-value and significance expressed as adjusted p-value using BH method for each annotation term (Figure 5-9). The table is ordered by overrepresentation p-value. Data shown by this table is also possible to download as csv file.



Label	Description	Observed in sample	Expected in sample	Odds ratio	Adjusted underrepresentation P-value (FDR)	Underrepresentation P-value	Adjusted overrepresentation P-value (FDR)	Overrepresentation P-value
Phenotype description: 2hr glucose		2	5.882636e-05	41553.333	1	1	1.387335e-06	1.048639e-09
Phenotype description: DIABETES MELLITUS NONINSULIN-DEPENDENT SUSCEPTIBILITY TO		2	7.843515e-05	31164.944	1	1	1.387335e-06	2.097256e-09
Phenotype description: DIABETES MELLITUS TYPE 2		2	8.627866e-04	2832.975	1	1	1.457620e-04	3.305260e-07
Phenotype description: Fasting proinsulin		2	1.843226e-03	1325.955	1	1	5.048506e-04	1.526381e-06
Phenotype description: ACTININ ALPHA-3 POLYMORPHISM		1	1.960879e-05	56097.200	1	1	1.853030e-03	1.960879e-05
Phenotype description: Actn3 deficiency		1	1.960879e-05	56097.200	1	1	1.853030e-03	1.960879e-05
Phenotype description: CATECHOL-O-METHYLTRANSFERASE POLYMORPHISM		1	1.960879e-05	56097.200	1	1	1.853030e-03	1.960879e-05
Phenotype description: DOPAMINE RECEPTOR D2 REDUCED BRAIN DENSITY OF		1	1.960879e-05	56097.200	1	1	1.853030e-03	1.960879e-05

Figure 5-9. Enrichment tab

Label column shows annotation class label concatenated with annotation term label. *Observed in sample* column is number of filtered input variations which are associated with annotation term. *Expected in sample* column shows expected number of annotations in a sample calculated using size of a variation sample, variations count in a background set and annotations associations count in a background set. Odds ratio is computed by following equation:

$$OR = \frac{sample\ associations/(sample\ variations - sample\ associations)}{total\ associations/(total\ variations - total\ associations)}.$$

Function computing expected count of associations and odds ratio is `get_odds_ratio`.

Over and underrepresentation p-values are computed by hypergeometric test which is performed by R function `phyper`. Adjustment of p-values by BH method is performed by R function `p.adjust`. These operations are performed in `get_overrepresentation_significance` and `get_underrepresentation_significance` functions. For getting associations counts in sample function `get_sparse_matrix` was implemented to build sparse binary matrix from variation and annotation identifier pairs which are returned by

`get_association_pairs` function. Finally `get_enrichment_data` function merges odds ratio and significance data and adjusts table data to be in readable form.

The reactivity graph of Enrichment tab is shown in Figure 5-10.

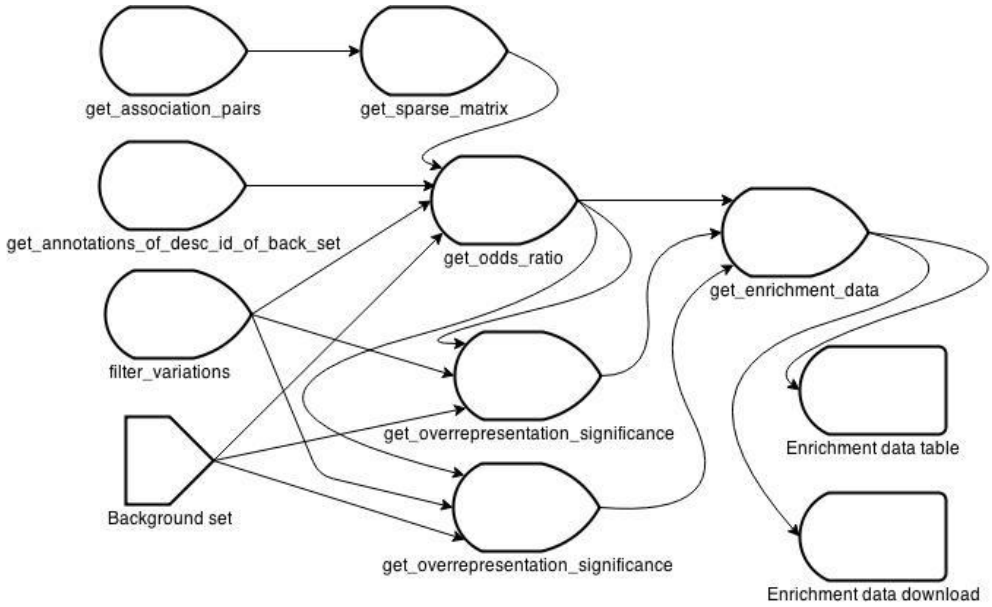


Figure 5-10. Reactivity graph of Enrichment tab.

5.4.3 Karyogram tab

This tab contains figure with graphical view of all chromosomes with illustrated positions of input variations. Each variation is depicted by red vertical line above the site where it is located (Figure 5-11).

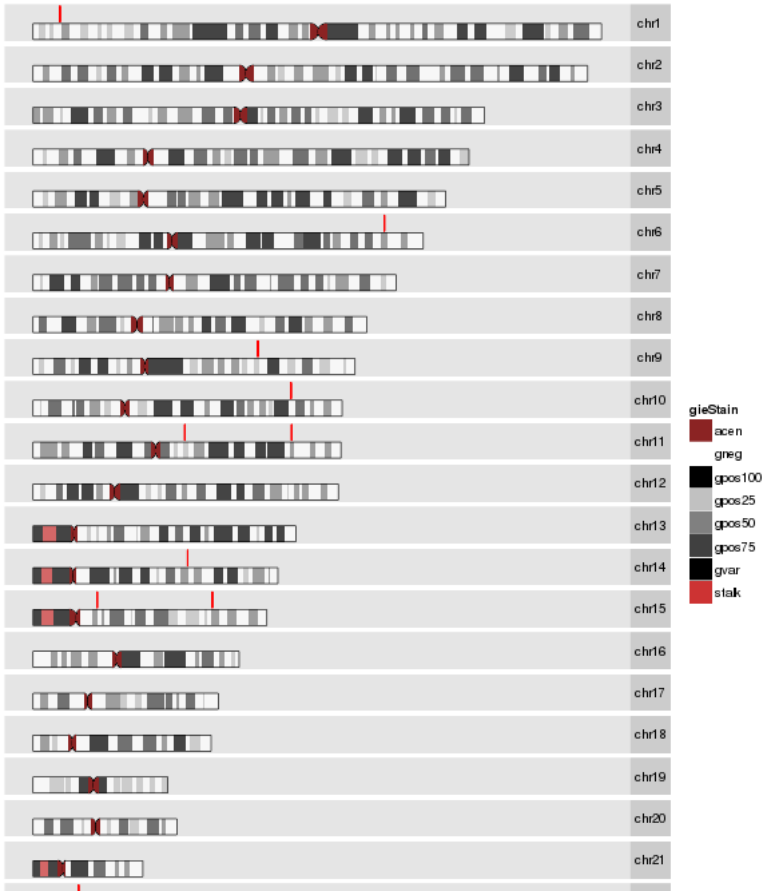


Figure 5-11. Karyogram showing location of variations.

5.4.4 Visualization tab

In visualization tab the binary matrix of variations associations with annotations is possible to show (Figure 5-12). Visualization is configurable to reach optimal view and multiple parameters are possible to adjust, for example angle of column labels, dendrograms visibility or grid visibility.

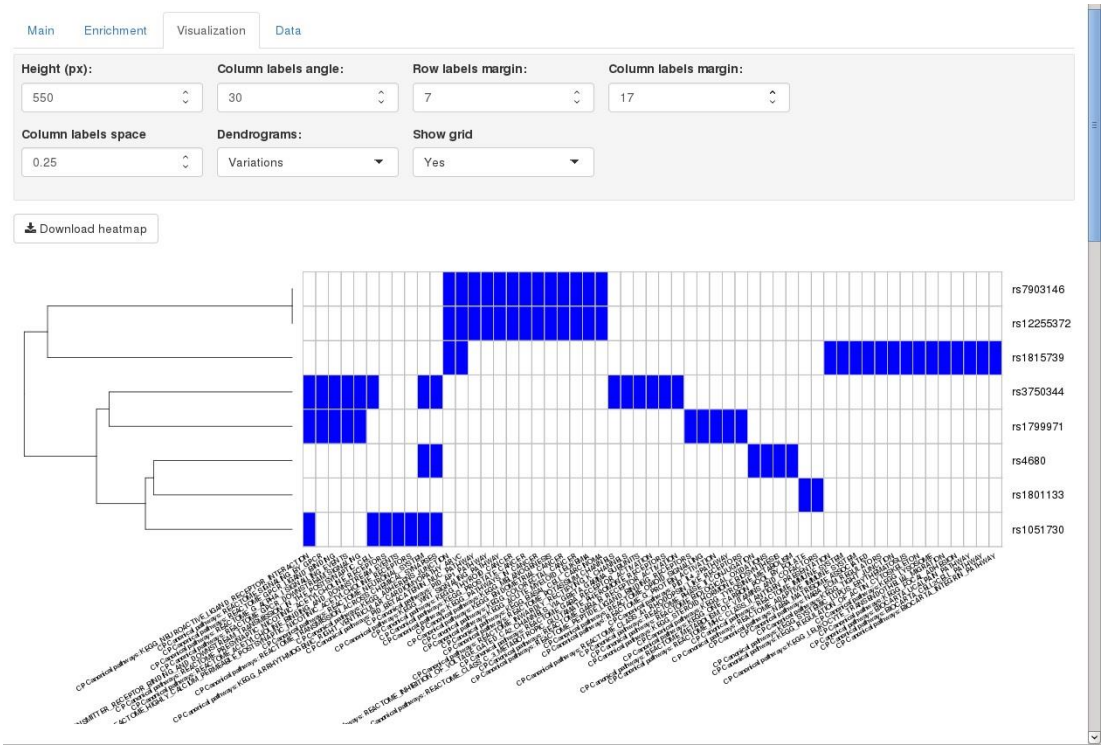


Figure 5-12. Visualization tab.

As data source for `heatmap.2` function a contingency table is created in `get_contingency_table` function using association pairs from `get_association_pairs` function. Annotation term labels in the contingency table are created by concatenation of annotation class label and term label to clearly present annotation label to user.

The reactivity graph of Visualization tab is shown in Figure 5-13.

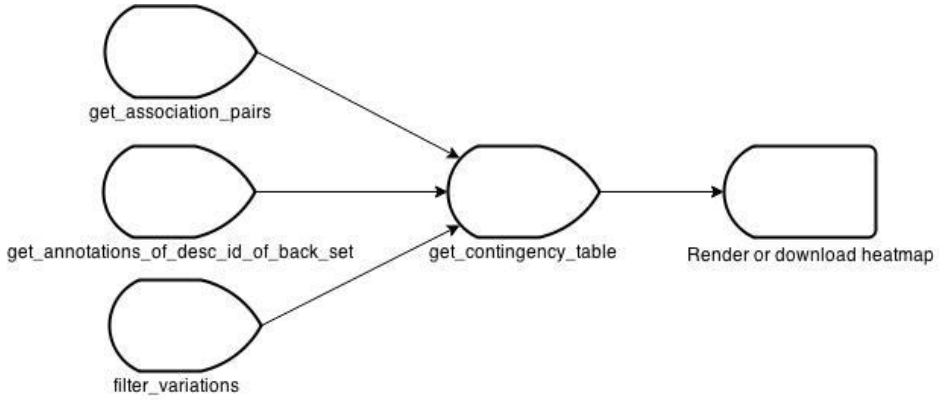


Figure 5-13. Rreactivity graph of visualization data.

5.4.5 Data tab

Data tab shows table of variations with their basic information and binary data whether variation is associated with annotation terms (Figure 5-14). In Varanto the view is limited to at most 100 annotation terms but in case of download of results all annotation terms are possible to obtain.

Main Enrichment Visualization Data

Maximum columns to show: 100

Download all data

Show 25 entries Search:

Name	Strand	Position	Allele	Chromosome	Consequence to transcript: 3_prime_UTR_variant	Consequence to transcript: 5_prime_UTR_variant	Consequence to transcript: coding_sequence_variant	Consequence to transcript: downstream_gene_variant
rs1801133	1	11796321	G/A	1	0	0	0	1
rs7903146	1	112998590	C/T	10	0	0	0	0
rs12255372	1	113049143	G/T	10	0	0	0	0
rs1815739	1	66560624	C/T	11	0	0	0	1
rs1800497	1	113400106	G/A	11	0	0	0	1
rs181575	1	67806862	C/T	14	0	0	0	0
rs7495174	1	28099092	A/G	15	0	0	0	0
rs1051730	1	78601997	G/A	15	0	0	0	0
rs4680	1	19963748	G/A	22	0	0	0	1
rs1799971	1	154039662	A/G	6	0	0	0	0
rs3750344	1	98578034	T/C	9	0	0	0	0
Name	Strand	Position	Allele	Chromosome	Consequence to transc	Consequence to transc	Consequence to transcript:	Consequence to transcript:

Figure 5-14. Data tab

The annotation terms columns have values of zero or one depending whether they are associated with variation. Function `get_variation_binary_data` builds data

source of the data table. It also uses sparse matrix from `get_sparse_matrix` function and for each annotation adds column of zeros and ones to table according to values in sparse matrix.

The reactivity graph of Data tab is shown in Figure 5-15.

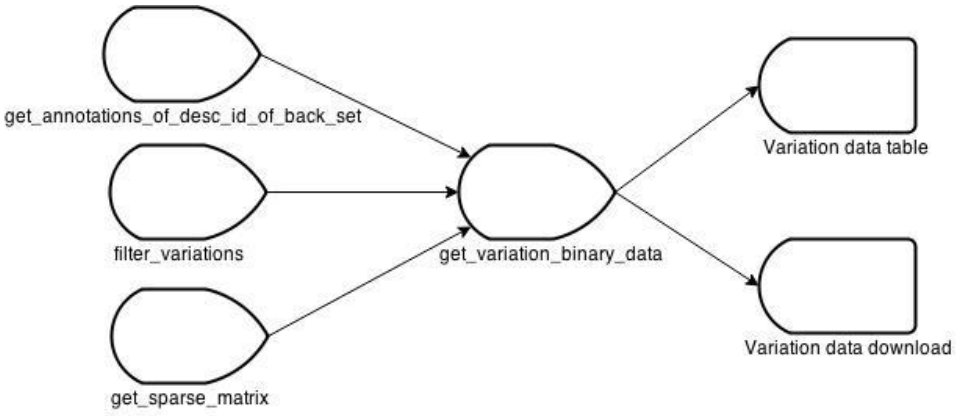


Figure 5-15. Reactivity graph of data tab.

6 RESULTS AND USE CASES

In this chapter Varanto tool is demonstrated. At first performance of import and Varanto tool is analysed. Afterwards its usability in finding new biological knowledge from GWAS results is demonstrated. Varanto is also useful in detecting technical bias in background sets caused by technology errors. In this chapter performance results and use cases are demonstrated on data downloaded from Ensembl release 79 and genome assembly *GRCh38.p2*.

6.1 Performance results

In this subchapter the performance of import process and several queries to background database is evaluated. All performance tests are performed on virtualized computer cluster with following specification (Table 6-1):

Processor:	AMD Opteron 6348 with 36 cores
Memory:	576 GB
Operating system:	CentOS 6.6

Table 6-1. Specification of virtualized computer cluster where the performance tests were performed.

6.1.1 Import

Database server has following specification (Table 6-2):

Processor:	AMD Opteron 6348 with 4 cores
Memory:	32 GB
Operating system:	CentOS 7.1
Database:	PostgreSQL 9.2.10

Table 6-2. Specification of computer where the database server is installed.

The durations of database import steps are depicted in the following Table 6-3:

Import step	**Duration**
Ensembl variation identifiers	~ 3 minutes (downloading from Ensembl)
Ensembl variation annotations	~ 12 hours (downloading from Ensembl)
Associated genes identifiers	~ 12 minutes
Associated genes annotations	~ 2 minutes (downloading from Ensembl)
GET-evidence annotations	~ 15 seconds (downloading from homepage)
Preparing data for insertion to database	~ 11 hours
Dropping existing database schema	~ 1 second
Initializing database	~ 10 hours
Total	**~ 1.5 day**

Table 6-3. Durations of database import steps.

The most time consuming steps are obtaining Ensembl variation annotations, preparing data for insertion to database and initializing database. Ensembl variation annotations performance depends on workload of Ensembl biomaRt server and Internet connection. Preparing data for insertion to database manipulates large annotation files which are transformed to table files and initializing database includes copying all data from table files to database and building indexes and foreign keys. The row counts for each database table are in the following Table 6-4:

Table	**Row count**
Ann2Back_set	1,777,287
Annotation	592,429
Annotation_description	38

Table	Row count
Background_sets	3
Var2Ann	5,351,873,567
Var2Back_set	1,464,869
Variation	62,492,516

Table 6-4. Database tables row counts.

Annotation classes from `annotation_description` table and background sets from `Background_sets` table are listed in Appendix 1.

6.1.2 Database queries

The performance of database queries is important for obtaining results in reasonable time for web user interface. Several tests were performed using *microbenchmark* package of R (Mersmann 2014). All tests are repeated 100 times and plots with line ranges of measured times are generated for depicting performance of queries with input of various size. Median values are marked as filled circles. The single R script was used to run all of the tests.

The first test consist of getting the variations information while only variations which are part of specified background set are considered. This corresponds to function `get_variations_of_names_of_back_set`. The function is benchmarked with several various inputs. Variation names parameters are of size 1, 10, 100, 1,000, 10,000 and 100,000. Background set parameter is set to Affymetrix SNP 6.0. The set of input variations used in these tests are chosen randomly from set of Affymetrix SNP 6.0 variations. In the following plot (Figure 6-1) the performance of this function is visualized.

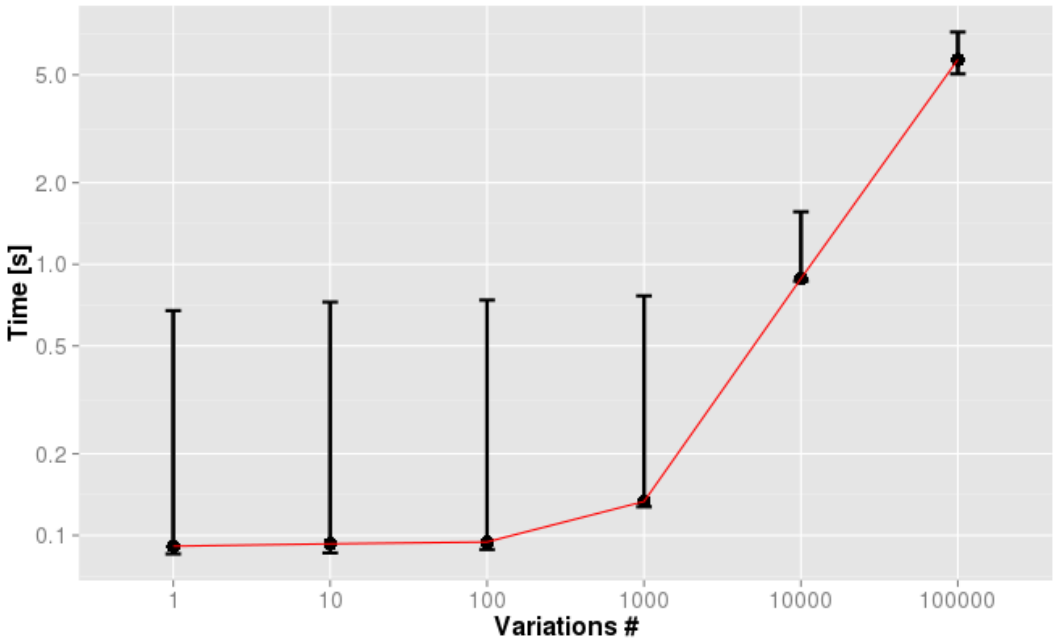


Figure 6-1. Performance of query for variations which is performed by function `get_variations_of_names_of_back_set`.

From the graph it is possible to see that each query has some little initialization delay and the time gets increasing from input size of 100 variation names up to 5 seconds when getting 100,000 variations. Median values are close to low boundaries so sometimes the time of the query is higher than usual.

Following test visualized in Figure 6-2 is aimed on query for annotation terms while only annotation terms which are associated with at least one variation of specified background set are considered. This corresponds to function `get_annotations_of_desc_id_of_back_set`. Background set parameter is again set to Affymetrix SNP 6.0. Annotation class parameter is set to identifiers of annotation classes with sizes of 4, 20, 189, 15,289 and 127,709 annotation terms. The size of annotation class is considered as number of its annotation terms associated with at least one variation of the Affymetrix background set.

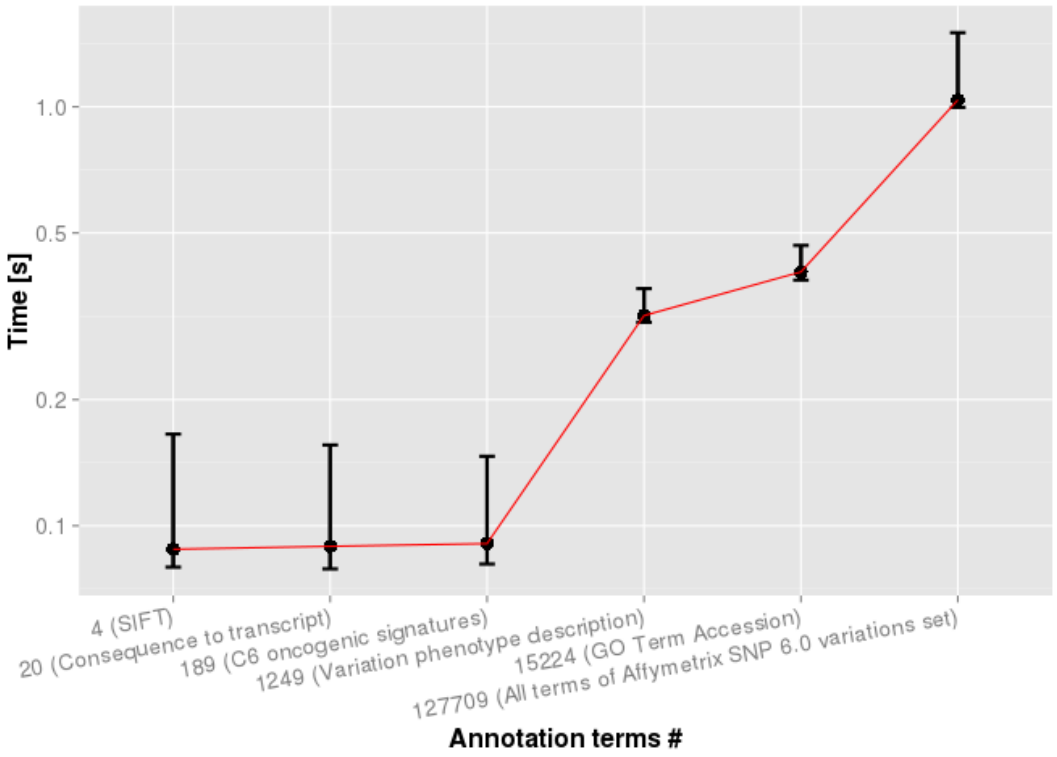


Figure 6-2. Performance of query for annotations which is performed by function `get_annotations_of_desc_id_of_back_set`.

Similarly also in this case there is some initialization delay in most cases of running the query and time of query increases when getting of about 200 or more annotation terms.

The next test measures performance of getting the association pairs of variations and annotation terms identifiers. This corresponds to function `get_association_pairs`. The function is benchmarked with several combinations of various variations input size and annotation class sizes. Variation names are of size 10, 100 and 1,000, annotation classes are of size 4, 20, 189, 1,249 and 15,224 so 15 combinations will be benchmarked.

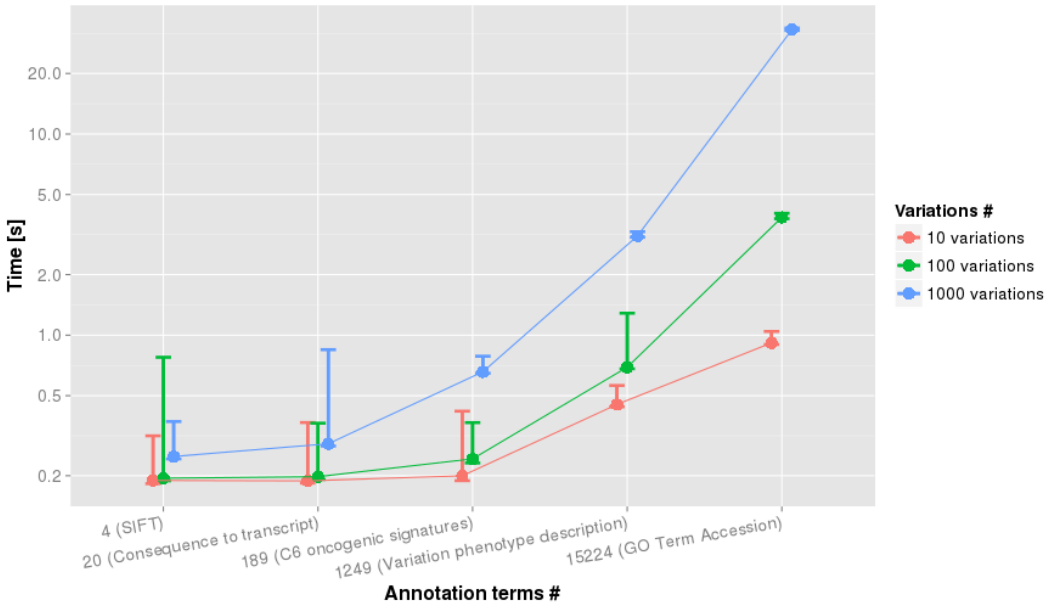


Figure 6-3. Performance of query for association pairs which is performed by function `get_association_pairs`.

As it is seen in the Figure 6-3 when getting associations of large sets of variations and annotation terms the time of the query processing gets high. The more input variations there are the more steeply the time for higher counts of annotation terms increases. In case of 1,000 variations and about 15,000 annotation terms it is more than 20 seconds.

Final test measures performance of enrichment which is computed by `get_enrichment_data` function and using the same input parameters like in the previous test with association pairs (Figure 6-4).

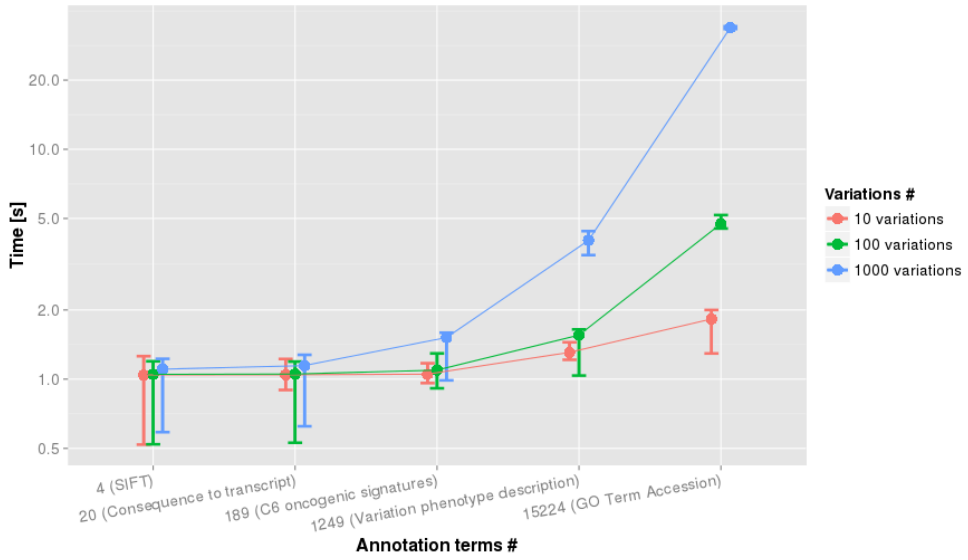


Figure 6-4. Performance of enrichment_analysis performed by function `get_enrichment_data`.

Results are similar for most of the input combinations to results of performance test of query for getting associations. Duration of queries are only slightly higher. Proportionally the duration is increased at most for inputs with low number of annotation terms as they last mostly about one second while for query for associations they last only about 0.2 second. For inputs with high number of annotation terms the difference is only negligible so `phyper` function shows good performance.

6.2 Use-case in GWAS

For demonstration of Varanto web tool functionality several use-cases in GWAS are described. The list of input variations are collected from public sources and examined in Varanto.

6.2.1 Body Mass Index

One phenotype trait which genetic background has been intensively studied is body mass index. In this use case variations associated with BMI (n=151) were obtained from GWAS catalogue (Hindorff et al. n.d.). Distance filter of 1 kilobase reduces count to 145 to prevent overestimation originating from multiple variations being located at

the same locus. Because the variations comes from several studies the set of all variations is used as the background set. The variations are located across almost the whole human genome as it is seen in the following Figure 6-5.

Figure 6-5. Locations of variations from the GWAS catalog associated to BMI.

In the following Table 6-5 results from enrichment analysis including several annotation classes for these variations are shown. 30 most significantly overrepresented annotations are shown from several annotation classes.

Label	Observed variations #	Expected variations #	Odds ratio	p-value (adjusted by FDR)	p-value
Phenotype description: BODY MASS INDEX	129	0.00327623	356823	0	0
Phenotype description: BMI	53	0.00179125	46633.0	6.3e-217	1.8e-221
Qualified impact: Insufficiently evaluated pathogenic	38	0.00499556	10307.9	5.5e-131	2.3e-135
Impact: pathogenic	38	0.00520902	9885.44	2.0e-130	1.2e-134
Inheritance pattern: unknown	41	0.09410119	607.074	6.05e-91	4.29e-95
Phenotype description: Weight	16	7.65692e-05	234879	4.10e-78	3.49e-82
Phenotype description: OBESITY	16	0.00027379	65686.4	7.01e-68	6.96e-72
Associated gene with phenotype: FTO	10	6.9608e-05	154302	8.72e-46	9.89e-50
Associated gene with phenotype: NR	11	0.00363123	3277.86	1.85e-31	2.37e-35
Associated gene with phenotype: BDNF	5	2.5523e-05	202898	2.04e-22	2.90e-26
Associated gene with phenotype: TMEM18	5	2.78433e-05	185989.6	3.18e-22	4.97e-26

Label	Observed variations #	Expected variations #	Odds ratio	p-value (adjusted by FDR)	p-value
GO Term Accession: GO:0035516	10	0.01956226	548.980	7.10e-21	1.61e-24
GO Term Accession: GO:0070350	10	0.01956226	548.980	7.10e-21	1.61e-24
Phenotype description: GROWTH RETARDATION DEVELOPMENTAL DELAY COARSE FACIES AND EARLY	10	0.01956226	548.980	7.10e-21	1.61e-24
Phenotype description: GROWTH RETARDATION DEVELOPMENTAL DELAY COARSE FACIES AND EARLY DEATH	10	0.01956226	548.980	7.10e-21	1.61e-24
Phenotype description: Lethal polymalformative syndrome Boissel type	10	0.01956226	548.980	7.10e-21	1.61e-24
GO Term Accession: GO:0035515	10	0.02126767	504.953	1.45e-20	3.71e-24
GO Term Accession: GO:0035553	10	0.02126767	504.953	1.45e-20	3.71e-24
Associated gene with phenotype: MC4R	6	0.00039212	15961.6	1.54e-20	4.16e-24
GO Term Accession: GO:0042245	10	0.02208440	486.276	1.88e-20	5.41e-24
GO Term Accession: GO:0035552	10	0.02216097	484.595	1.88e-20	5.60e-24
GO Term Accession: GO:0043734	10	0.02300788	466.755	2.61e-20	8.14e-24
GO Term Accession: GO:0044065	10	0.02434900	441.042	4.39e-20	1.43e-23
GO Term Accession: GO:0010883	10	0.02986430	359.577	3.23e-19	1.10e-22
GO Term Accession: GO:0070989	10	0.03621258	296.528	2.12e-18	7.52e-22
GO Term Accession: GO:0080111	10	0.05630386	190.690	1.66e-16	6.11e-20
GO Term Accession: GO:0008198	10	0.05654981	189.860	1.67e-16	6.38e-20
GO Term Accession: GO:0006307	10	0.06079360	176.601	3.30e-16	1.31e-19
GO Term Accession: GO:0001659	10	0.06506755	164.997	6.27e-16	2.58e-19
BP GO biological process: PATTERN_SPECIFICATION_PROCESS	11	0.13412366	88.66425	8.70e-15	3.83e-18

Table 6-5. 30 most significant annotations for SNPs associated with BMI.

The phenotype descriptions like *BODY MASS INDEX*, *BMI*, *Weight* and *OBESITY* are between the most enriched annotations what is expected as the input variations are associated to BMI according to GWAS catalogue. Some GET-evidence annotations are also strongly enriched like *Qualified impact: Insufficiently evaluated pathogenic*, *Impact: pathogenic* and *Inheritance pattern: unknown*. Annotations of associations to several genes, particularly *FTO*, *NR*, *BDNF*, *TMEM18* and *MC4R*, have also statistically significant p-values. Other annotation terms in the Table 6-5 are related to Growth retardation or Lethal polymalformative syndrome so some variations are associated with these diseases as well. However, further research is needed to determine if there are some common real biological foundations between them and BMI. There are also several GO annotation terms in the Table 6-5. Many of them are according to their descriptions related to DNA or RNA demethylation activities. Other GO terms,

for instance GO:0070350, is annotation for regulation of white fat cell proliferation or GO:0010883 is annotation for regulation of lipid storage.

6.2.2 Crohn's disease

Among diseases studied using GWAS approach, one of the most researched one is Crohn's disease. It was found that the genetic factors explain half of its overall risk (Tysk et al. 1988). Variations associated with Crohn's disease (n = 194) were obtained from the GWAS catalogue, similarly as with the BMI use case (Hindorff et al. n.d.). Distance filter of 1 kilobase reduces the number to 145, to help preventing overestimating effect of nearby variations. The variations comes from several studies so the set of all variations is used as the background. In the following Table 6-6 results from enrichment analysis including several annotation classes for these variations are shown. 30 most significantly overrepresented annotations are shown from several annotation classes.

Label	Observed variations #	Expected variations #	Odds ratio	p-value (adjusted by FDR)	p-value
Phenotype description: Crohns Disease	183	3.1831e-03	Inf	0	0
Qualified impact: Insufficiently evaluated pathogenic	58	6.3047e-03	13467.50	8.15e-207	1.8e-211
Impact: pathogenic	58	6.5741e-03	12915.58	6.35e-206	2.1e-210
Inheritance pattern: unknown	76	1.1876e-01	1093.76	5.50e-186	2.4e-190
Associated gene with phenotype: Intergenic	31	4.9196e-03	7586.22	3.14e-103	1.7e-107
Phenotype description: Inflammatory bowel disease	20	3.4262e-04	65536.57	1.86e-85	1.24e-89
Associated gene with phenotype: IL23R	16	7.3209e-05	239492.1	3.91e-79	3.04e-83
Phenotype description: Inflammatory Bowel Diseases	13	2.6941e-04	51943.83	1.89e-53	1.68e-57
Qualified impact: Insufficiently evaluated pharmacogenetic	13	3.8215e-03	3661.87	3.59e-38	3.61e-42
Impact: pharmacogenetic	13	3.8508e-03	3634.02	3.59e-38	3.99e-42
C2 curated: PID_IL23PATHWAY	21	1.1733e-01	202.05	9.39e-37	1.53e-40
CP Canonical pathways: PID_IL23PATHWAY	21	1.1733e-01	202.05	9.39e-37	1.53e-40
HGNC symbol: IL23R	14	9.0193e-03	1680.73	9.39e-37	1.56e-40
Phenotype description: INFLAMMATORY BOWeL DISeASe 17	14	9.0193e-03	1680.73	9.39e-37	1.56e-40
Phenotype description: PSORIASIS SUSCePTIBILITY 7	14	9.0193e-03	1680.73	9.39e-37	1.56e-40

Label	Observed variations #	Expected variations #	Odds ratio	p-value (adjusted by FDR)	p-value
Phenotype description: Behcet disease	18	5.2871e-02	377.48	3.66e-36	6.50e-40
GO Term Accession: GO:0005143	16	2.6991e-02	649.50	9.86e-36	1.86e-39
Phenotype description: Ulcerative colitis	16	2.7307e-02	641.97	1.12e-35	2.24e-39
GO Term Accession: GO:0032740	17	4.2113e-02	444.92	2.44e-35	5.14e-39
GO Term Accession: GO:0042510	14	1.1787e-02	1286.10	3.00e-35	6.66e-39
GO Term Accession: GO:0042520	14	1.2978e-02	1167.99	1.10e-34	2.57e-38
GO Term Accession: GO:0038155	14	1.3309e-02	1138.95	1.32e-34	3.65e-38
GO Term Accession: GO:0042019	14	1.3309e-02	1138.95	1.32e-34	3.65e-38
GO Term Accession: GO:0042020	14	1.3309e-02	1138.95	1.32e-34	3.65e-38
GO Term Accession: GO:0072536	14	1.3309e-02	1138.95	1.32e-34	3.65e-38
Associated gene with phenotype: intergenic	7	2.9284e-05	248549.7	6.83e-34	1.97e-37
GO Term Accession: GO:0051135	14	1.5825e-02	957.89	1.38e-33	4.13e-37
GO Term Accession: GO:2000330	14	1.6077e-02	942.89	1.66e-33	5.15e-37
GO Term Accession: GO:0032725	14	1.6194e-02	936.06	1.77e-33	5.70e-37
GO Term Accession: GO:0010535	14	1.6680e-02	908.78	2.59e-33	8.62e-37

Table 6-6. 30 most significant annotations for SNPs associated with Crohn's disease.

Phenotype description of Crohn's Disease is the most significantly overrepresented and this can indicate that enrichment analysis works well for this input. *GET-evidence* annotations are again strongly enriched while in this case *Impact: pharmacogenetic* term is also enriched. *Inflammatory bowel disease* annotation of *Phenotype description* class is more general term for Crohn disease. Many annotations are related to *Interleukin 23 (IL23)* cytokine and its receptor. From annotations it is observable that many variations are associated with gene *IL23R* or pathway associated with IL23. Also GO terms *GO:0038155*, *GO:0042019*, *GO:0042020*, *GO:0072536* are related to IL23 mediated signalling pathway, its binding, receptor activity and complex. These findings highlight the role of IL23 induced inflammation in Chron's disease. In the Table 6-6 are another *Phenotype description* terms referencing Psoriasis, Behcet disease and Ulcerative colitis and they may be related with Crohn's disease. Another GO terms in the Table 6-6 references regulations of *Interleukin-17*, tyrosine phosphorylation of *Stat1* or *Stat3* protein or *T cells*.

For this use-case the visualization of associations of variations and annotations is shown in the following Figure 6-6 where several clusters of variations with similar annotations can be identified.

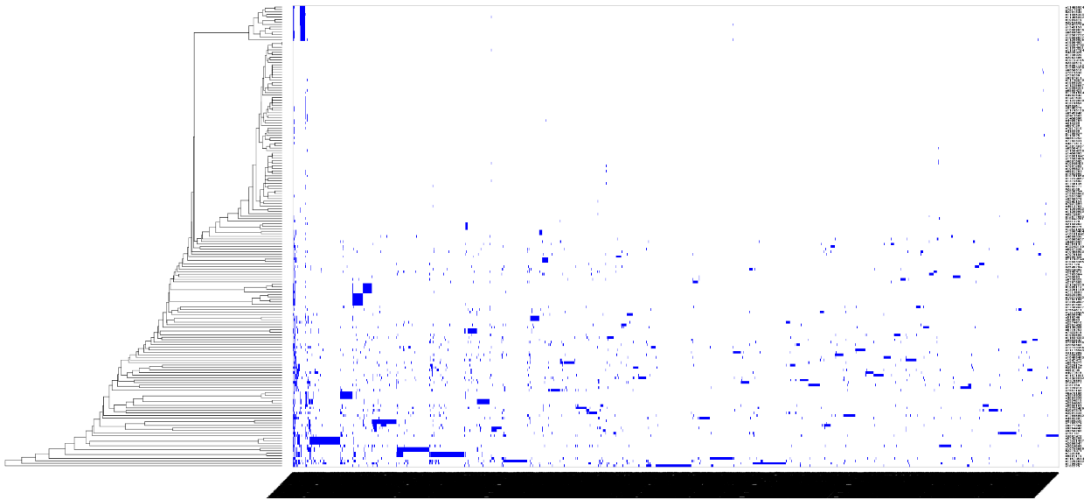


Figure 6-6. Visualization of associations.

6.3 Use-case in detecting technical bias

The Varanto can be used to detect technical biases in input or background sets. We showcase this functionality by studying how distribution of different allele combinations may differ between different technological platforms presented by our background sets. In this case technical bias means that some alleles can be over- or underrepresented in a sample of variations with evenly distributed alleles when comparing them to different background sets.

For this use-case the testing background set is Affymetrix SNP 6.0 set which is available in Varanto. The sample of 1,200 variations are chosen so 100 variations have one of the 12 combinations of alleles as is seen in the following tables Table 6-7 and Table 6-8. The first table contains enrichment analysis data of alleles using set of all variations and another table of alleles using Affymetrix SNP 6.0 background set.

Label	Observed variations #	Expected variations #	Odds ratio	Underrepresentation p-value	Overrepresentation p-value
T/A	100	38.71	2.73	1.00e+00	3.13e-17
A/T	100	39.37	2.68	1.00e+00	9.19e-17
T/G	100	40.57	2.60	1.00e+00	6.08e-16
A/C	100	41.42	2.54	1.00e+00	2.19e-15

Label	Observed variations #	Expected variations #	Odds ratio	Underrepresentation p-value	Overrepresentation p-value
G/C	100	44.18	2.38	1.00e+00	1.05e-13
C/G	100	45.21	2.32	1.00e+00	3.99e-13
C/A	100	49.51	2.11	1.00e+00	6.27e-11
G/T	100	50.60	2.07	1.00e+00	1.99e-10
T/C	100	140.20	0.69	9.83e-05	1.00e+00
A/G	100	143.69	0.67	2.90e-05	1.00e+00
C/T	100	207.69	0.43	2.43e-19	1.00e+00
G/A	100	204.49	0.44	1.80e-18	1.00e+00

Table 6-7. Enrichment analysis of alleles using set of all variations as the background set.

Label	Observed variations #	Expected variations #	Odds ratio	Underrepresentation p-value	Overrepresentation p-value
T/A	100	37.53	2.82	1.00e+00	4.01e-18
A/T	100	37.78	2.80	1.00e+00	6.19e-18
T/G	100	44.01	2.39	1.00e+00	8.02e-14
A/C	100	44.33	2.37	1.00e+00	1.21e-13
G/C	100	53.71	1.94	1.00e+00	4.07e-09
C/G	100	53.98	1.93	1.00e+00	5.24e-09
C/A	100	48.71	2.15	1.00e+00	2.52e-11
G/T	100	48.76	2.15	1.00e+00	2.67e-11
T/C	100	190.77	0.48	6.21e-15	1.00e+00
A/G	100	191.13	0.48	5.05e-15	1.00e+00
C/T	100	219.03	0.41	1.45e-22	1.00e+00
G/A	100	218.52	0.41	2.04e-22	1.00e+00

Table 6-8. Enrichment analysis of alleles using *Affymetrix SNP 6.0* as the background set.

From the tables it is observable that most of the alleles have similar odds ratios or p-values in both tables. The largest differences are seen for G/C or C/G alleles when odds ratio differs in their case by 0.40 and p-value by 4 orders. The odds ratio is higher in the Table 6-7 of set of all variations so there is a bias originated from selection of too many variations with G/C and C/G alleles to set of genotyped variations. The similar bias is discoverable in case of T/C and A/G alleles.

7 CONCLUSION

This work described basics of molecular biology, genetics research, annotation methods and enrichment analysis. Annotation methods are important for processing large collections of biological data. The thesis described how many annotations are already available and that some of them are generated computationally and others are created by manual curation. Wide variety of computational methods are adopted to annotate genomic elements while this work mainly described methods for annotating variations. During preparation of background database of Varanto huge amount of annotation data was used, the number of stored associations between variations and annotation terms in the background database being more than 5 billion. GWAS create many useful results in the form of list of variations associated with different traits. Described enrichment methods can be used to determine whether and how these variations affect studied trait.

Created tool Varanto is aimed on enrichment analysis of variations, especially for sets of variations resulting from GWAS. Hypergeometric method implementation by function `phyper` in R showed good performance for an input of that we estimate represent a common usage. For example for 100 variations and 10,000 annotation terms, the computation lasts about few seconds. Also it was shown that hypergeometric method is able to detect enrichment of relevant annotations related to studied trait or disease, and this was further demonstrated in the use cases. However these results should be further analysed and interpreted by researchers with knowledge of biological background to make final conclusions from the results. It is also possible to examine the technical bias present in the background sets. The karyogram visualization with input variations also provides possibility to have good insight about variation locations in genome. The binary matrix visualization helps to recognize clustered variations with similar annotations or clustered annotations which are part of a similar set of variations.

In the future the Varanto functionality could be enhanced, for example, by possibility to show list of observed variations for any annotation term in table of enrichment anal-

ysis or by possibility to define custom background set. Also applying methods for considering hierarchical relationships between annotation terms, such as GO ontology terms, may be useful. These methods are already mentioned in literature and may provide more accurate results. In presented use cases some annotations in the table were duplicated, resulting from some annotations being associated with variations as well as genes. Unifying these annotations may be an option to consider in future. Varanto as a tool can be used to annotate variations and perform enrichment analysis. All data table results are available for download in csv format and all visualizations are available in pdf format enabling using them for further processing or analysis.

References

Abecasis, G.R. et al., 2012. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422), pp.56–65. Available at: http://dx.doi.org/10.1038/nature11632 [Accessed July 9, 2014].

Adzhubei, I.A. et al., 2010. A method and server for predicting damaging missense mutations. *Nature methods*, 7(4), pp.248–9. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2855889&tool=pmcentrez&rendertype=abstract [Accessed July 9, 2014].

Amberger, J.S. et al., 2015. OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic acids research*, 43(Database issue), pp.D789–98. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4383985&tool=pmcentrez&rendertype=abstract [Accessed April 1, 2015].

Apweiler, R. et al., 2004. UniProt: the Universal Protein knowledgebase. *Nucleic acids research*, 32(Database issue), pp.D115–9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/14681372 [Accessed February 6, 2015].

Ashburner, M. et al., 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1), pp.25–9. Available at: http://dx.doi.org/10.1038/75556 [Accessed July 10, 2014].

Ball, M.P. et al., 2012. A public resource facilitating clinical use of genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 109(30), pp.11920–7. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3409785&tool=pmcentrez&rendertype=abstract [Accessed April 22, 2015].

Bartel, D.P., 2004. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2), pp.281–97. Available at: http://www.ncbi.nlm.nih.gov/pubmed/14744438 [Accessed July 17, 2014].

Benjamini, Y. & Hochberg, Y., 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. , 57, pp.289–300. Available at: http://www.researchgate.net/publication/221995234_Controlling_the_False_Discovery_Rate_A_Practical_and_Powerful_Approach_to_Multiple_Testing [Accessed April 8, 2015].

Berkopec, A., 2007. HyperQuick algorithm for discrete hypergeometric distribution. *Journal of Discrete Algorithms*, 5(2), pp.341–347. Available at: http://www.sciencedirect.com/science/article/pii/S1570866706000499 [Accessed March 23, 2015].

BioCarta LLC, BioCarta. Available at: http://www.biocarta.com/ [Accessed May 15, 2015].

Bonferroni, C.E., 1936. *Teoria statistica delle classi e calcolo delle probabilità*, Available at: http://books.google.fi/books/about/Teoria_statistica_delle_classi_e_calcolo.html?id=3CY-HQAACAAJ&pgis=1 [Accessed April 21, 2015].

Boorsma, A. et al., 2005. T-profiler: scoring the activity of predefined groups of genes using gene expression data. *Nucleic acids research*, 33(Web Server issue), pp.W592–5. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1160244&tool=pmcentrez&rendertype=abstract [Accessed April 6, 2015].

Boyle, A.P. et al., 2012. Annotation of functional variation in personal genomes using RegulomeDB. *Genome research*, 22(9), pp.1790–7. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3431494&tool=pmcentrez&rendertype=abstract [Accessed February 26, 2015].

Bromberg, Y. & Rost, B., 2007. SNAP: predict effect of non-synonymous polymorphisms on function. *Nucleic acids research*, 35(11), pp.3823–35. Available at: http://nar.oxfordjournals.org/content/35/11/3823.full [Accessed March 5, 2015].

Bush, W.S. & Moore, J.H., 2012. Chapter 11: Genome-wide association studies. *PLoS computational biology*, 8(12), p.e1002822. Available at: http://www.ncbi.nlm.nih.gov/pubmed/23300413 [Accessed July 11, 2014].

Capriotti, E., Calabrese, R. & Casadio, R., 2006. Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. *Bioinformatics (Oxford, England)*, 22(22), pp.2729–34. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16895930 [Accessed March 15, 2015].

Clark, N.R. & Ma'ayan, A., 2011. Introduction to Statistical Methods for Analyzing Large Data Sets: Gene-Set Enrichment Analysis. *Science Signaling*, 4(190), pp.tr4–tr4. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3205944&tool=pmcentrez&rendertype=abstract [Accessed April 6, 2015].

Croft, D. et al., 2014. The Reactome pathway knowledgebase. *Nucleic acids research*, 42(Database issue), pp.D472–7. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3965010&tool=pmcentrez&rendertype=abstract [Accessed October 17, 2014].

Cunningham, F. et al., 2014. Ensembl 2015. *Nucleic acids research*, 43(D1), pp.D662–669. Available at: http://nar.oxfordjournals.org/content/43/D1/D662 [Accessed November 25, 2014].

Durinck, S. et al., 2005. BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics (Oxford, England)*, 21(16), pp.3439–40. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16082012 [Accessed February 17, 2015].

Eilbeck, K. et al., 2005. The Sequence Ontology: a tool for the unification of genome annotations. *Genome biology*, 6(5), p.R44. Available at: http://genomebiology.com/2005/6/5/R44 [Accessed March 9, 2015].

Ensembl, 2015. Ensembl genome browser 79: Homo sapiens -. Available at: http://mar2015.archive.ensembl.org/Homo_sapiens/Info/Annotation [Accessed May 15, 2015].

Feuk, L., Carson, A.R. & Scherer, S.W., 2006. Structural variation in the human genome. *Nature reviews. Genetics*, 7(2), pp.85–97. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16418744 [Accessed July 9, 2014].

Fox, B., 2014. GNU Bash. Available at: https://www.gnu.org/software/bash/ [Accessed April 22, 2015].

Franke, A. et al., 2010. Genome-wide meta-analysis increases to 71 the number of confirmed Crohn's disease susceptibility loci. *Nature genetics*, 42(12), pp.1118–25. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21102463 [Accessed February 26, 2015].

Gibson, G., 2011. Rare and common variants: twenty arguments. *Nature reviews. Genetics*, 13(2), pp.135–45. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22251874 [Accessed July 9, 2014].

Goodrich, J.A. & Kugel, J.F., 2006. Non-coding-RNA regulators of RNA polymerase II transcription. *Nature Reviews Molecular Cell Biology*, 7(8), pp.612–616. Available at: http://dx.doi.org/10.1038/nrm1946 [Accessed March 15, 2015].

Gray, K.A. et al., 2015. Genenames.org: the HGNC resources in 2015. *Nucleic acids research*, 43(Database issue), pp.D1079–85. Available at: http://www.ncbi.nlm.nih.gov/pubmed/25361968 [Accessed January 23, 2015].

Greg, G. & Muse, S. V., 2004. *A Primer of Genome Science* 2nd ed., Sunderland, Massachusetts, USA: Sinauer Associates, Inc. Available at: http://www.amazon.com/Primer-Genome-Science-2nd-Edition/dp/0878932321 [Accessed March 15, 2015].

Gupta, P.K. et al., 2007. *Bioinformatics Research and Applications* I. Măndoiu & A. Zelikovsky, eds., Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: http://www.researchgate.net/publication/221462472_Statistical_Absolute_Evaluation_of_Gene_Ontology_Terms_with_Gene_Expression_Data [Accessed April 7, 2015].

Harrow, J. et al., 2012. GENCODE: the reference human genome annotation for The ENCODE Project. *Genome research*, 22(9), pp.1760–74. Available at: http://europepmc.org/articles/PMC3431492 [Accessed July 12, 2014].

Hartl, D.L. & Jones, E.W., 2001. *Genetics: Analysis of Genes and Genomes, Nide 1* 5th ed., Sudbury, Massachusets, USA: Jones and Bartlett Publishers. Available at: http://books.google.fi/books/about/Genetics.html?id=-qBqAAAAMAAJ&pgis=1 [Accessed March 15, 2015].

Hindorff, L. et al., A Catalog of Published Genome-Wide Association Studies. Available at: http://www.genome.gov/gwastudies/ [Accessed March 15, 2015].

Huang, D.W., Sherman, B.T. & Lempicki, R.A., 2009a. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1), pp.1–13. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2615629&tool=pmcentrez&rendertype=abstract [Accessed July 9, 2014].

Huang, D.W., Sherman, B.T. & Lempicki, R.A., 2009b. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols*, 4(1), pp.44–57. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19131956 [Accessed July 9, 2014].

Chang, W. et al., 2015. shiny: Web Application Framework for R. Available at: http://cran.r-project.org/package=shiny.

Kanehisa, M. et al., 2012. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic acids research*, 40(Database issue), pp.D109–14. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3245020&tool=pmcentrez&rendertype=abstract [Accessed July 9, 2014].

Karolchik, D. et al., 2004. The UCSC Table Browser data retrieval tool. *Nucleic acids research*, 32(Database issue), pp.D493–6. Available at: http://nar.oxfordjournals.org/content/32/suppl_1/D493.abstract?ijkey=06tIQcBr2VZNz&keytype=ref [Accessed December 24, 2014].

Kent, W.J. et al., 2002. The Human Genome Browser at UCSC. *Genome Research*, 12(6), pp.996–1006. Available at: http://genome.cshlp.org/content/12/6/996.abstract [Accessed July 11, 2014].

Kinsella, R.J. et al., 2011. Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database : the journal of biological databases and curation*, 2011, p.bar030. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21785142 [Accessed January 28, 2015].

Kitts, A. et al., 2014. The Database of Short Genetic Variation (dbSNP). Available at: http://www.ncbi.nlm.nih.gov/books/NBK174586/ [Accessed March 15, 2015].

Kumar, P., Henikoff, S. & Ng, P.C., 2009. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nature protocols*, 4(7), pp.1073–81. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19561590 [Accessed July 14, 2014].

Manolio, T.A. et al., 2009. Finding the missing heritability of complex diseases. *Nature*, 461(7265), pp.747–53. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19812666 [Accessed July 9, 2014].

Massey, F.J.J., 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253), pp.68–78. Available at: http://www.jstor.org/discover/10.2307/2280095?sid=21106358052353&uid=3737976&uid=4&uid=2 [Accessed April 5, 2015].

McLaren, W. et al., 2010. Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinformatics (Oxford, England)*, 26(16), pp.2069–70. Available at: http://bioinformatics.oxfordjournals.org/content/26/16/2069 [Accessed July 16, 2014].

McLean, C.Y. et al., 2010. GREAT improves functional interpretation of cis-regulatory regions. *Nature biotechnology*, 28(5), pp.495–501. Available at: http://www.ncbi.nlm.nih.gov/pubmed/20436461 [Accessed July 14, 2014].

Mersmann, O., 2014. microbenchmark: Accurate Timing Functions. Available at: http://cran.r-project.org/package=microbenchmark.

Michailidou, K. et al., 2015. Genome-wide association analysis of more than 120,000 individuals identifies 15 new susceptibility loci for breast cancer. *Nature Genetics*. Available at: http://www.ncbi.nlm.nih.gov/pubmed/25751625 [Accessed March 9, 2015].

Moustafa, A., Permutation Test. Available at: http://spark.rstudio.com/ahmed/permutation/ [Accessed March 28, 2015].

Neale, B. et al., 2012. *Statistical Genetics*, Garland Science. Available at: https://books.google.com/books?id=2ykWBAAAQBAJ&pgis=1 [Accessed April 23, 2015].

Neph, S. et al., 2012. BEDOPS: high-performance genomic feature operations. *Bioinformatics (Oxford, England)*, 28(14), pp.1919–20. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22576172 [Accessed March 15, 2015].

Orengo, C., Jones, D.T. & Thornton, J.T., 2003. *Bioinformatics: Genes, Proteins and Computers*, BIOS Scientific. Available at: https://books.google.com/books?id=9kslngEACAAJ&pgis=1 [Accessed March 23, 2015].

Preacher, K.J. & Briggs, N.E., 2001. Calculation for Fisher's Exact Test: An interactive calculation tool for Fisher's exact probability test for 2 x 2 tables. Available at: http://quantpsy.org/fisher/fisher.htm [Accessed March 25, 2015].

Press, W.H., 1992. *The Art of Scientific Computing*, Cambridge University Press. Available at: https://books.google.com/books?id=7vuNLcQhg8UC&pgis=1 [Accessed March 31, 2015].

Pruitt, K. et al., 2012. The Reference Sequence (RefSeq) Database. Available at: http://www.ncbi.nlm.nih.gov/books/NBK21091/ [Accessed March 15, 2015].

Python Software Foundation, 2015. Welcome to Python.org. Available at: https://www.python.org/ [Accessed April 24, 2015].

Quinlan, A.R. & Hall, I.M., 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)*, 26(6), pp.841–2. Available at: http://www.ncbi.nlm.nih.gov/pubmed/20110278 [Accessed July 9, 2014].

R Documentation, R: The Hypergeometric Distribution. Available at: https://stat.ethz.ch/R-manual/R-patched/library/stats/html/Hypergeometric.html [Accessed March 25, 2015].

Ralston, A. & Brown, W., 2008. Chromatin Remodeling and DNase 1 Sensitivity. *Nature Education*, 1(1), p.15. Available at: http://www.nature.com/scitable/topicpage/chromatin-remodeling-and-dnase-1-sensitivity-1054 [Accessed March 15, 2015].

Rattray, A.M.J. & Müller, B., 2012. The control of histone gene expression. *Biochemical Society transactions*, 40(4), pp.880–5. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22817752 [Accessed January 15, 2015].

Reeve, C.P., 1986. *An Algorithm for Computing the Gamma C.D.F. to a Specified Accuracy*, Gaithersburg, Maryland, USA. Available at: http://www.nist.gov/itl/sed/upload/SED_Note_86-2-2.pdf [Accessed April 3, 2015].

Reich, D.E. & Lander, E.S., 2001. On the allelic spectrum of human disease. *Trends in genetics : TIG*, 17(9), pp.502–10. Available at: http://www.ncbi.nlm.nih.gov/pubmed/11525833 [Accessed March 15, 2015].

Rice, J.A., 2007. *Mathematical Statistics and Data Analysis*, Thompson/Brooks/Cole. Available at: https://books.google.com/books?id=b6XHAAAACAAJ&pgis=1 [Accessed March 23, 2015].

Ritchie, G.R. & Flicek, P., 2014. Computational approaches to interpreting genomic sequence variation. *Genome Medicine*, 6(10), p.87. Available at: http://genomemedicine.com/content/6/10/87 [Accessed November 20, 2014].

Ritchie, G.R.S. et al., 2014. Functional annotation of noncoding sequence variants. *Nature methods*, 11(3), pp.294–6. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24487584 [Accessed July 23, 2014].

Rockman, M. V & Kruglyak, L., 2006. Genetics of global gene expression. *Nature reviews. Genetics*, 7(11), pp.862–72. Available at: http://dx.doi.org/10.1038/nrg1964 [Accessed January 8, 2015].

RStudio, 2014. R Studio. Available at: http://www.rstudio.com/.

Sanger, F., Nicklen, S. & Coulson, A.R., 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12), pp.5463–7. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=431765&tool=pmcentrez&rendertype=abstract [Accessed July 10, 2014].

Shihab, H.A. et al., 2013. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. *Human mutation*, 34(1), pp.57–65. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3558800&tool=pmcentrez&rendertype=abstract [Accessed March 15, 2015].

SQL Power Group Inc., 2015. Data Modeling & Profiling Tool: SQL Power Architect. Available at: http://www.sqlpower.ca/page/architect [Accessed April 24, 2015].

Stenson, P.D. et al., 2014. The Human Gene Mutation Database: building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine. *Human genetics*, 133(1), pp.1–9. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3898141&tool=pmcentrez&rendertype=abstract [Accessed January 16, 2015].

Stormo, G.D. et al., 1982. Use of the "Perceptron" algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Research*, 10(9), pp.2997–3011. Available at: http://nar.oxfordjournals.org/content/10/9/2997 [Accessed February 15, 2015].

Subramanian, A. et al., 2005. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43), pp.15545–50. Available at: http://www.pnas.org/content/102/43/15545.abstract [Accessed July 10, 2014].

The Eclipse Foundation, 2015. Eclipse - The Eclipse Foundation open source community website. Available at: https://eclipse.org/ [Accessed April 24, 2015].

The ENCODE Project Consortium, 2004. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science (New York, N.Y.)*, 306(5696), pp.636–40. Available

at: http://www.sciencemag.org/cgi/doi/10.1126/science.1105136 [Accessed July 10, 2014].

The International HapMap Consortium, 2003. The International HapMap Project. , 426(6968), pp.789–796.

The PostgreSQL Global Development Group, 2015. PostgreSQL: The world's most advanced open source database. Available at: http://www.postgresql.org/ [Accessed April 24, 2015].

The R Foundation, 2015. R: The R Project for Statistical Computing. Available at: http://www.r-project.org/ [Accessed April 22, 2015].

The Scipy community, 2015. Statistics (scipy.stats) — SciPy v0.15.1 Reference Guide. Available at: http://docs.scipy.org/doc/scipy-0.15.1/reference/tutorial/stats.html [Accessed March 25, 2015].

Tysk, C. et al., 1988. Ulcerative colitis and Crohn's disease in an unselected population of monozygotic and dizygotic twins. A study of heritability and the influence of smoking. *Gut*, 29(7), pp.990–6. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1433769&tool=pmcentrez&rendertype=abstract [Accessed April 20, 2015].

Visscher, P.M. et al., 2012. Five years of GWAS discovery. *American journal of human genetics*, 90(1), pp.7–24. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3257326&tool=pmcentrez&rendertype=abstract [Accessed July 11, 2014].

Wang, K., Li, M. & Hakonarson, H., 2010. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic acids research*, 38(16), p.e164. Available at: /pmc/articles/PMC2938201/?report=abstract [Accessed July 15, 2014].

Ward, L.D. & Kellis, M., 2012. HaploReg: a resource for exploring chromatin states, conservation, and regulatory motif alterations within sets of genetically linked variants. *Nucleic acids research*, 40(Database issue), pp.D930–4. Available at: http://nar.oxfordjournals.org/content/early/2011/11/07/nar.gkr917.long [Accessed July 10, 2014].

Warnes, G.R. et al., 2015. gplots: Various R Programming Tools for Plotting Data. Available at: http://cran.r-project.org/package=gplots.

Weirauch, M.T. et al., 2013. Evaluation of methods for modeling transcription factor sequence specificity. *Nature biotechnology*, 31(2), pp.126–34. Available at: http://dx.doi.org/10.1038/nbt.2486 [Accessed December 20, 2014].

Wickham, H. & Francois, R., 2015. dplyr: A Grammar of Data Manipulation. Available at: http://cran.r-project.org/package=dplyr.

Wray, N.R. et al., 2013. Pitfalls of predicting complex traits from SNPs. *Nature reviews. Genetics*, 14(7), pp.507–15. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4096801&tool=pmcentrez&rendertype=abstract [Accessed July 11, 2014].

Ye, J. et al., 2006. WEGO: a web tool for plotting GO annotations. *Nucleic Acids Research*, 34(Web Server), pp.W293–W297. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1538768&tool=pmcentrez&rendertype=abstract [Accessed March 11, 2015].

Yu, N. et al., 2012. hiPathDB: a human-integrated pathway database with facile visualization. *Nucleic acids research*, 40(Database issue), pp.D797–802. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3245021&tool=pmcentrez&rendertype=abstract [Accessed May 15, 2015].

Appendix 1: Annotation classes and their source.

Annotation class	**Genomic element**	**Downloaded from**
Variant Alleles	Variation	Ensembl
Common Variant Alleles	Variation	- (Generated during import)
Phenotype description	Variation	Ensembl
Phenotype description	Gene	Ensembl
Study External Reference	Variation	Ensembl
Study Description	Variation	Ensembl
Consequence to transcript	Variation	Ensembl
Ensembl Gene ID	Variation	Ensembl
Associated gene with phenotype	Variation	Ensembl
PolyPhen prediction	Variation	Ensembl
SIFT prediction	Variation	Ensembl
GO Term Accession	Gene	Ensembl
HGNC symbol	Gene	Ensembl
Gene type	Gene	Ensembl
MIM Morbid Description	Gene	Ensembl
Impact	Variation	GET-Evidence

Annotation class	**Genomic element**	**Downloaded from**
Qualified impact	Variation	GET-Evidence
Inheritance pattern	Variation	GET-Evidence
H hallmark	Gene	MSigDB
C1 positional	Gene	MSigDB
C2 curated	Gene	MSigDB
CGP chemical and genetic perturbations	Gene	MSigDB
CP Canonical pathways	Gene	MSigDB
CP BIOCARTA BioCarta	Gene	MSigDB
CP KEGG KEGG	Gene	MSigDB
CP REACTOME Reactome	Gene	MSigDB
C3 motif	Gene	MSigDB
MIR microRNA targets	Gene	MSigDB
TFT transcription factor targets	Gene	MSigDB
C4 computational	Gene	MSigDB
CGN cancer gene neighborhoods	Gene	MSigDB
CM cancer modules	Gene	MSigDB
C5 GO	Gene	MSigDB
BP GO biological process	Gene	MSigDB

Annotation class	Genomic element	Downloaded from
CC GO cellular component	Gene	MSigDB
MF GO molecular function	Gene	MSigDB
C6 oncogenic signatures	Gene	MSigDB
C7 immunologic signatures	Gene	MSigDB

Table 1: List of annotation classes.