

## Texas Medical Center Library DigitalCommons@TMC

---

UT GSBS Dissertations and Theses (Open Access)

Graduate School of Biomedical Sciences


---

12-2018

# Improving dbNSFP

Mingyao Lu

Follow this and additional works at: [https://digitalcommons.library.tmc.edu/utgsbs\\_dissertations](https://digitalcommons.library.tmc.edu/utgsbs_dissertations)

 Part of the [Bioinformatics Commons](#), [Computational Biology Commons](#), and the [Genomics Commons](#)

---

### Recommended Citation

Lu, Mingyao, "Improving dbNSFP" (2018). *UT GSBS Dissertations and Theses (Open Access)*. 920.  
[https://digitalcommons.library.tmc.edu/utgsbs\\_dissertations/920](https://digitalcommons.library.tmc.edu/utgsbs_dissertations/920)

This Thesis (MS) is brought to you for free and open access by the Graduate School of Biomedical Sciences at DigitalCommons@TMC. It has been accepted for inclusion in UT GSBS Dissertations and Theses (Open Access) by an authorized administrator of DigitalCommons@TMC. For more information, please contact [laurel.sanders@library.tmc.edu](mailto:laurel.sanders@library.tmc.edu).



**Improving dbNSFP**

by

Mingyao Lu B.S.

APPROVED:

---

Xiaoming Liu, Ph.D.  
Advisory Professor

---

Yunxin Fu, Ph.D.

---

Peng Wei, Ph.D.

---

Degui Zhi, Ph.D.

---

Myriam Fornage, Ph.D.

APPROVED:

---

Dean, The University of Texas  
MD Anderson Cancer Center UTHHealth Graduate School of Biomedical  
Sciences

**Improving dbNSFP**

A

Thesis

Presented to the Faculty of

The University of Texas

MD Anderson Cancer Center UTHealth

Graduate School of Biomedical Sciences

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

by

**Mingyao Lu, B.S.**

Houston, Texas

December 2018

# Acknowledgments

I would like to take this opportunity to thank those who have helped me with all aspects of my research. First of all, I would like to express my deepest appreciation to my major professor, Dr. Xiaoming Liu, for his guidance and professional knowledge, as well as his belief in my professional and personal growth. Besides my advisor, I would like to thank the rest of my committee members: Dr. Degui Zhi, Dr. Yunxin Fu, Dr. Peng Wei, and Dr. Myriam Fornage, for their insightful comments and encouragement. My sincere thanks also go to Dr. Alice Djotsa and Chang Li, for the stimulating discussions and for all the fun we have had in the last one year. In the end, I thank my family and friends. Without their support, it would not be possible to conduct this research.

## IMPROVING dbNSFP

Mingyao Lu, B.S.

Advisory Professor: Xiaoming Liu, Ph.D.

The analysis and interpretation of DNA variation are very important for the Whole Exome studies (WES). Genome research has focused on single nucleotide variants (SNVs). Since indels are as important as SNVs, especially indels in coding regions are often candidates of disease-causing variants, thus, it is necessary to expand the focus to include indel mutations.

The goal of my project is to provide an automatic annotation pipeline to the WES based disease studies project by extending the dbNSFP with a tool for automated indel annotation and deleteriousness prediction. The current sequencing results typically include both SNVs and indels. Although there have been many available tools to integrate functional prediction/annotations for SNV effects, there are no such tools for indels to my knowledge. Therefore, the aim of this thesis was to add deleteriousness prediction scores to indel annotation based on gene models, including CADD, SIFT, and PROVEAN. All those scores can be calculated on-the-fly after installing resources locally. A Docker implementing the indel annotation and deleteriousness prediction has been developed and ready to be deployed from the cloud.

**Keywords:** Indels Annotation, Functional Annotation, Whole Exome Sequencing, Deleterious Prediction Scores

# Table of Contents

<b>Approval Sheet .....</b>	<b>i</b>
<b>Title Sheet .....</b>	<b>i</b>
<b>Acknowledgments .....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>iv</b>
<b>Table of Contents .....</b>	<b>v</b>
<b>List of Illustrations.....</b>	<b>vii</b>
<b>List of tables.....</b>	<b>viii</b>
<b>1 Introduction.....</b>	<b>1</b>
<b>1.1 Context: Literature review about annotation.....</b>	<b>1</b>
<b>1.2 Aim and Motivation .....</b>	<b>3</b>
<b>1.3 Layout of this Thesis .....</b>	<b>4</b>
<b>2 Methodology .....</b>	<b>5</b>
<b>2.1 Genome Annotation .....</b>	<b>5</b>
<b>2.2 dbNSFP Introduction .....</b>	<b>5</b>
<b>2.3 WGS Annotator (WGSA).....</b>	<b>7</b>
<b>2.4 Deleterious Prediction Scores .....</b>	<b>8</b>
<b>2.4.1 Combined Annotation–Dependent Depletion (CADD) .....</b>	<b>9</b>

2.4.2 Sorting Intolerant From Tolerant (SIFT).....	10
2.4.3 Protein Variation Effect Analyzer (PROVEAN) .....	11
3 New Module .....	15
3.1 Input.....	15
3.2 Pipeline Realization .....	16
3.3 Running Pipeline .....	16
3.4 Outputs.....	17
4 Case Study .....	18
4.1 Pipeline Configuration .....	18
4.2 Results .....	18
5 Conclusion and Future Work .....	23
6 References .....	24

# List of Illustrations

Figure 1. WGS pipeline .....	8
Figure 2. Indelanno Pipeline .....	15
Figure 3. inputsample.vcf .....	19
Figure 4. Script.sh .....	20
Figure 5. Proveanoutput.sh .....	20



# List of tables

Table 1. input.txt ..... 19

Table 2. input.txt0 ..... 20

Table 3. input.txt01 ..... 21

Table 4. siftinput.txt ..... 21

Table 5. input.txt012 ..... 22

# 1 Introduction

## 1.1 Context: Literature review about annotation

DNA sequencing is the process of determining the order of DNA nucleotides, or bases, in an individual's genome. Two large-scale DNA sequencing technologies, are whole genome sequencing (WGS) and whole exome sequencing (WES). They have been mainly used as a research tool to identify genetic variations and are currently being introduced into clinics.

Advancements in DNA sequencing technologies in throughput and quality have driven an ever-growing body of genomic sequencing data to a new level. In the meantime, reductions of the cost of DNA sequencing makes it possible to generate sequence data rapidly and with high throughput. It not only opens the door to the affordable sequencing of patients and the development of precision medicine but also democratizes the ability to collect information of a great number of genetic variations in individual laboratories. However, it remains a challenge to extract meaningful biological information from raw sequence data. Genome sequencing can only figure out the complete DNA sequence of an organism's genome but cannot directly provide the biological functions of DNA. Therefore, the genomic variant annotation is an increasingly crucial and complex step in the analysis of genome sequencing data. On one hand, functional annotation helps analysts filter a subset of elements of interest (e.g., cell type-specific enhancers), and on the other hand, annotation helps researchers improve their ability to identify phenotype-related loci (e.g., use functional prediction scores as weights for association tests) and interpret potentially interesting discoveries.

Functional annotation database dbNSFP was developed in 2011 to facilitate filtering and prioritizing SNVs observed in WES (Liu et al. 2011). Since then 32 content updates have been released including two major updates to version 2.0 and 3.0 (Liu et al. 2013, 2016b). Currently, dbNSFP only supports SNV annotation. However, in practice indels are as important as SNV, especially indels in coding regions are often candidates of disease-causing variants.

Historically, genome research has focused on single nucleotide variants (SNVs), because of their high prevalence and relatively simple detection. However, recent advances in sequencing techniques and computational methods have expanded the focus to include insertion and deletion (indel) mutations (Fang et al. 2016).

Indel mutations are defined by adding or missing of one or more nucleotides (less than 1000 base pairs) of a DNA sequence. Recent studies have shown that insertions and deletions (indels) are the second most common variant in the human genome. Researches have shown that they have a crucial impact on genetic variation by altering human characteristics and can lead to a variety of human diseases. There are two types of coding insertions/deletions (indels), frameshifting indels and non-frameshifting indels. Frameshifting indels that have lengths that are not divisible by three and subsequently result in frameshifts. The frameshift mutation is a highly destructive type of indel mutations that alter the reading framework of protein-coding sequences and are closely related to neurodevelopmental disorders, cardiovascular diseases, cancer, and many other human diseases. Indels, which is divisible by three, will cause amino acid insertion/deletion or block substitution called non-frameshifting indels.

Although indels are the second most common variant in human genomes, it remains challenging to accurately call indels from short-read sequencing data. Unlike SNVs, due to the length variability, we cannot list all potential coding indels in the human genome in dbNSFP. Therefore, we plan to annotate indel on-the-fly. WGS annotator (WGSA) pipeline already supports indel annotation on-the-fly using three annotation tools ANNOVAR, SnpEff and VEP for Gencode and RefSeq gene models (Liu et al. 2016a). Those programs in the pipeline can be easily ported to the dbNSFP to support quick indel annotation based on gene models. Adding indel annotations will make the dbNSFP a truly one-stop-shop annotation tool for WES based disease studies. The aim of this study is to add deleteriousness prediction scores to indel annotation, including CADD (Kircher et al. 2014), SIFT (Vaser et al. 2016) and PROVEAN (Choi et al. 2012). All those scores can be calculated on-the-fly after installing resources locally. A Docker implementing the indel annotation and deleteriousness prediction will be developed and ready to be deployed from the cloud.

## **1.2 Aim and Motivation**

The main goal of this thesis is to provide an automatic annotation pipeline to the WES based disease studies project by extending the dbNSFP with a tool for automated indel annotation and deleteriousness prediction. The current sequencing results typically include both SNVs and indels. Although there have been many available tools to integrate functional prediction/annotations for SNV effects, there are no such tools for indels to my knowledge. The ideal tool will have the ability to search several annotation resources in batch and produce an integrated report in return. Such a tool will be very useful and appreciated by the sequencing community.

### **1.3 Layout of this Thesis**

The layout of this thesis is as follows. In Chapter 1, we present a brief description of the context and motivations of our work. Chapter 2 introduces the genome annotation. A set of genome sequencing and annotation methods are described. Chapter 3 describes the main work developed in this project. The new annotation module is detailed as to how to run the scripts. Chapter 4 includes a test case that validates the automatic annotation module. Conclusions and future work are presented in Chapter 5.

## 2 Methodology

### 2.1 Genome Annotation

In the study of a particular organism, the complete genome sequence provides only partial and raw information. After obtaining DNA sequences, scientists need to find out where the genes are, what the functions of the various DNA elements are, how they interact, with each other and with environmental factors, etc. This is where the annotation process intervenes to link this information to the genome sequence. Genome annotation is thus the process of extracting important biological information from the genome sequences. Genome annotation has two interrelated types: structural annotation and functional annotation. Genome annotation starts by identifying the positions of structural genomic elements, like genes, exons, introns, repeated regions, promoters, etc. This process can be defined as structural annotation. After identifying genes and other structural sequence elements in a genome, a secondary annotation providing biochemical and biological function information to these elements is necessary and this process is called functional annotation. Function annotation is an important part of genome sequencing studies.

### 2.2 dbNSFP Introduction

With the developments in sequencing technologies, whole exome and whole genome sequencing has enabled fast and high-throughput generation of sequence data and has been used to discover genomic variations that cause human diseases. Functional prediction is a crucial step in genomic sequencing analyses and can filter or prioritize nonsynonymous SNV as well as insertions/deletions (indels) for further analysis. In

recent years, functional prediction algorithms for genomic variants continue to make progress with the developments of computational biology, structure biology, bioinformatics, and population genetics. In general, the functional prediction tools output a score to qualify the degree of how likely a mutation will affect the protein function. They used deleterious prediction scores to measure different variants to provide a more accurate result for identifying gene influencing a trait. However, different prediction tools used different methods, every prediction algorithm has its own weakness and strength. In addition, even there are lots of prediction algorithms for sequencing studies, it is still necessary to use multiple predictions tools to make a more accurate prediction for a variation instead of only relying on a single one. (Agajanian et al. 2018).

dbNSFP is a functional annotation database and was developed in 2011 to provide a comprehensive resource for functional predictions and annotations for variations studies including nonsynonymous single-nucleotide variants (nsSNVs) and splice site variants (ssSNVs). The aim of dbNSFP is to accelerate the steps of filtering and prioritizing SNVs from a great number list of SNVs discovered in an exome-sequencing study (Liu et al. 2011). Since then 32 content updates have been released including two major updates to version 2.0 and 3.0 (Liu et al. 2013, 2016b). The database compiled a list of all potential nsSNVs and ssSNVs based on the human reference sequence. Functional predictions and annotations were curated and compiled for each SNV. The current version was released in 2016 and was based on the Gencode release 22/ Ensembl version 79, including a total of 83,422,341 nsSNVs and ssSNVs (splicing-site SNVs). It does not need to connect the internet because it was designed to work locally and independently for users. Even if the users do not have good bioinformatics

training, they still can use it easily. They can use the companion Java program to search the database and the step can be done by a single command line call which is convenient. It is the first integrated database of functional annotation predictions from multiple methods for the comprehensive collection of human nsSNVs. However, till now, dbNSFP only supports SNV annotation.

### **2.3 WGS Annotator (WGSa)**

WGS annotator (WGSa) is designed as an annotation pipeline for human genome re-sequencing studies, to facilitate the functional annotation step of whole genome sequencing. It already supports indel annotation on-the-fly using three annotation tools ANNOVAR, SnpEff and VEP for Gencode and RefSeq gene models and provides a summary of variant consequences from the six annotation results (Liu et al. 2016a). Those programs in the pipeline can be easily ported to the dbNSFP to support quick indel annotation based on gene models.

Currently, WGSa supports SNV annotations and indels annotations locally without remote database requests, which can be extended for large WGS studies. An overview of the WGSa pipeline is shown in Figure 1. It was used with permission from



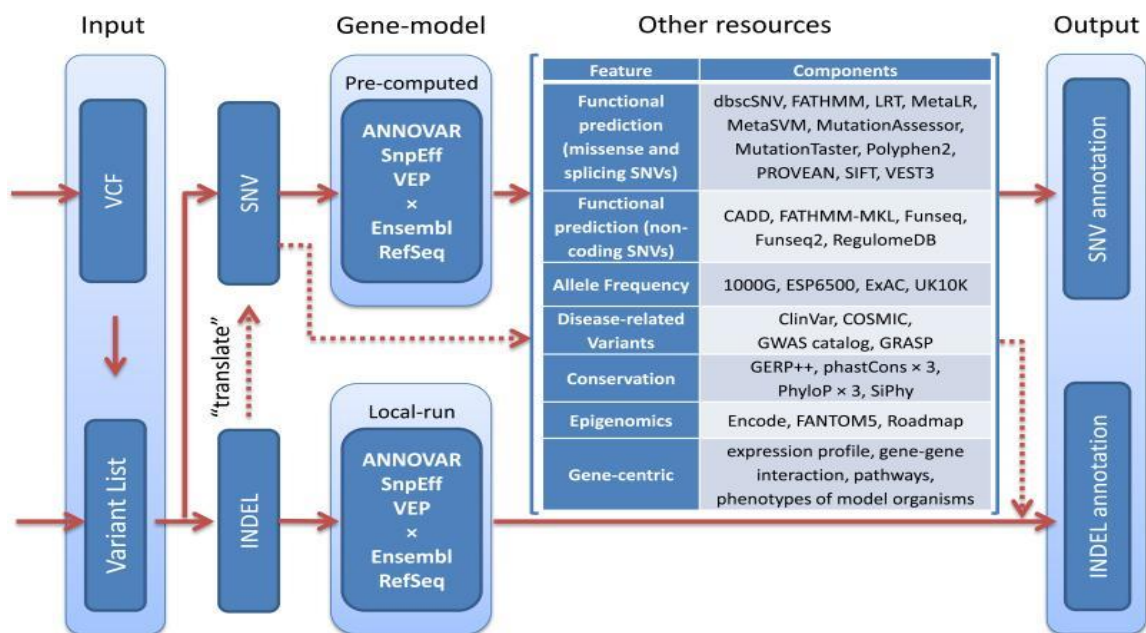


Figure 1. WGS pipeline

## 2.4 Deleterious Prediction Scores

In recent years, with the development of genetic sequencing technologies, many sequence-based disease types of research have been helped. Next-generation sequencing helped researchers get more accurate and effective results. The application of whole exome sequencing has been increased in the field of human disease studies. Predicting whether mutations are deleterious or neutral remains a challenge in interpreting all exome sequence data. Throughout exome sequencing (WES) studies, the deleterious prediction of genomic variations is critical to predicting whether mutations affect protein function and may lead to genetic diseases. With the rapid development of high-throughput experimental technologies, annotations on functional elements of the human genome have been widely used; therefore, various information can be used to study the effects of nsSNVs. Many methods have been proposed to predict deleterious nsSNVs, as well as

friendly web-based interactive software, to facilitate researchers' studies. However, most prediction algorithms only focus on SNVs but cannot deal with sequence changes such as indels and multiple amino acid substitutions. At the same time, although researchers already have many different algorithms to conduct the deleterious prediction, those methods may not have a consensus. Even when they are used to analyze the same sequence data, different prediction tools may get different results and their relative advantages in practical application are still unclear. Even when the same gene structure is implemented, predicted consequences of a given variant from different prediction tools may not be the same (Kim et al. 2016).

In order to add deleteriousness prediction scores to indel annotation, it was necessary to decide which deleteriousness prediction score should be used in the annotation. The software should fulfill certain requirements, such as, to be open source, to integrate diverse genome annotations and score the deleteriousness of insertion/deletions variants in the human genome, to be well documented and to present good results. During the process of choosing the deleteriousness prediction score, several tools were tested. Special attention was given to CADD, SIFT and PROVEAN.

#### **2.4.1 Combined Annotation–Dependent Depletion (CADD)**

CADD is an algorithm designed to provide a generalized approach to estimate the effect of genomic variants and predict the pathogenicity of human variants including single nucleotide variants (SNVs) and insertion/deletions (indels) variants in the human genome. It integrates diverse genome annotations and provides the deleteriousness of SNVs and indels in the human genome. Currently, it supported builds: GRCh37/hg19 and GRCh38/hg38 (Kicher et al. 2012).

CADD compares the annotation of fixed or almost fixed alleles with simulated variants in humans. It uses an empirical model of sequence evolution with CpG dinucleotide specific rates and mutation rates in a megabase window.

The installation of CADD requires a computer running Linux or Mac OS X. Users run CADD via the script CADD.sh which technically only requires either a VCF or VCF.gz input file within 2MB file size as last argument. In general, only the first 5 columns of a VCF file without header are needed for analysis, includes CHROM, ID, POS, REF, ALT. All other information will be ignored. Users can further specify the genome build via -g, request a fully annotated output (-a flag) and specify a separate output file via -o (else input file name.tsv.gz is used). i.e.:

```
./CADD.sh test/input.vcf
```

```
./CADD.sh -a -g GRCh37 -o output_inclAnno_GRCh37.tsv.gz test/input.vcf
```

In the output files, two distinct forms of scores are provided, namely “raw” and “scaled”. Since the scale of the combined SVM score (" C-. ") is actually arbitrary due to the use of annotations, the CADD score ranges from 1 to 99 with a cutoff of 15 based on each variable relative to all possible 8.6 billion alternative sequences in human reconstruction. If the score is greater than 15, then it is predicted to be deleterious.

CADD can provide quantitative priority to functional, harmful, and causal variants of disease across a wide range of functional categories, effect sizes, and genetic structures, and can give priority to causal variants in research and clinical settings.

#### **2.4.2 Sorting Intolerant From Tolerant (SIFT)**

SIFT is a widely used prediction tool based on sequence homology and amino acid physical properties to predict whether an amino acid substitution affects protein function. Users can prioritize substitutions for further genomic studies. SIFT can also predict coding indels that cause insertion/deletion of amino acids.

Since frameshifting indels and non-frameshifting indels are two different types of variations and reflect different biological function, the methods for the prediction of frameshifting indels and non-frameshifting indels are different. SIFT INDEL constructs a classifier based on decision tree algorithm to predict whether a non-frameshifting indel will affect the function of the gene. If it affects the function, then it is “gene-damaging”, if not then it is “neutral”. The SIFT indel classifier is trained to indel disease sets and neutral indel sets. In the training data, the disease-causing indels are from the Human Gene Mutation Database (HGMD) and the neutral indels are from comparative genomics and large sequencing projects such as Exome Sequencing Project (ESP). If a non-frameshifting indel causes an early stop or code shift, the indel will be discarded from the training and testing set (Hu et al. 2013).

SIFT INDEL accepts only space-based coordinates for insertion/deletion variants. To run SIFT via the script `SIFT_exome_indels.pl` which technically only space-based coordinates as input and generates a result shows whether the coding indel will affect the gene function with a confidence score. Typically, SIFT prediction scores range from 0 to 1. The amino acid substitution is predicted damaging when the score is lower than 0.05, and it is tolerated if the score is higher than 0.05. However, for SIFT INDEL, the result only shows neutral or damaging with a confidence.

#### **2.4.3 Protein Variation Effect Analyzer (PROVEAN)**

PROVEAN (Protein Variation Effect Analyzer) is an effective tool designed to predict the functional effects of protein sequence variations, including single or multiple amino acid substitutions, as well as non-frameshift insertions and deletions. PROVEAN is very useful for filtering sequence variants to identify functionally important nonsynonymous or indel variants (Choi et al. 2012).

PROVEAN predicts the effects of non-frameshift indels by measuring changes in the similarity scores of the target protein to its homologous protein sequence. Human indels extracted from UniProt's "Human Polymorphism and Disease Mutation" dataset had a deletion accuracy of 82% and an insertion accuracy of 87%.

PROVEAN used the delta alignment scale to measure the effect of amino acid variations on protein function. Delta alignment score is based not only on amino acid residues at the location of interest but also on the alignment quality of neighborhood flanking sequences. Due to the unique characteristics of scoring schemes, the new method can provide the functional prediction to assess the impact of changes in protein sequences in all categories except single amino acid substitution, including intra-frame indentation and multiple amino acid substitutions. The low delta value is interpreted as the harmful effect of amino acid variation on protein function, while high delta value is interpreted as the variation of neutral effect on protein function (Choi et al. 2012).

Its main function is to predict protein sequences from any organism. In order to annotate a sequence, PROVEAN needs some minimal inputs: a genomic sequence, of any length, in FASTA format; and amino acid variations. Run PROVEAN through script.sh technology to generate proof scores.

PROVEAN scores range from -14 to 14 with a cutoff of -2.5, a lower score indicating a higher likelihood to be deleterious. If the score is lower than -2.5, then it is predicted to be deleterious.

Since three prediction tools have three different input files, there is a need to convert user's input file to different prediction software's input file format. This procedure is finished by running Java program SIFT1, PROVEAN1 and PROVEAN2. Especially for PROVEAN, the input files are amino acid variations and protein sequence. Therefore, it is necessary to run ANNOVAR first to get the ENST ID and amino acid variants. Then running java program PROVEAN2 to get protein sequences, which is in FASTA format for PROVEAN input.

#### **2.4.3.1 ANNOVAR Annotation**

ANNOVAR is an efficient command-line driven software tool written by Kai Wang to functionally annotate single nucleotide variants (SNVs) and insertions/deletions detected from diverse genomes and filter mutations. ANNOVAR can analyze the genetic variation in various genomes by using the latest gene models. It can detect their functional consequences on genes, infer cellular genetic bands, report mutations in conserved regions or identify functional importance scores of mutations reported in the 1000 Genome Project and dbSNP (Wang et al. 2010). There are three different annotation methods: gene-based annotation, region-based annotation, and filter-based annotation. Gene-based annotations can identify whether SNVs or CNVs cause changes in protein-coding and affected amino acids, region-based annotations can identify variants in specific genome regions, filter-based annotations can identify variants recorded in specific databases, and other functions.

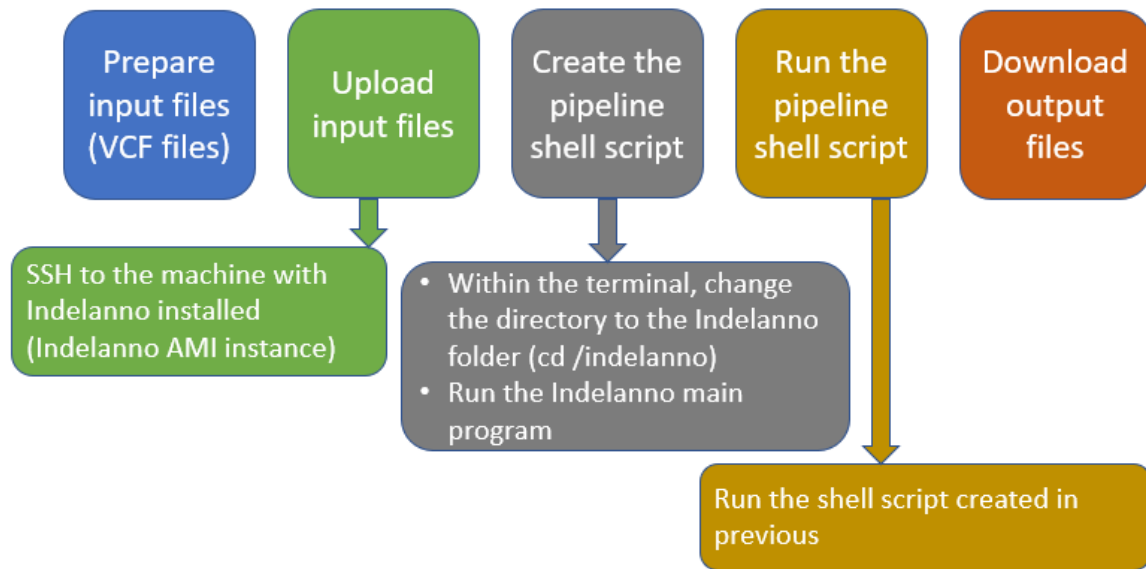
After running the ANNOVAR command, three files will be generated.

Ex1.variant\_function annotates the locations of variations as to genes.

Ex1.exonic\_variant\_function detailed notes on the functions, types and amino acids changes of exons. Ex1.ann.log log file, which contains the running command line and running hint, the database file used. In ex1.exonic\_variant\_function file, we get ENST ID and amino acid variants. Then searching the sequences by using ENST ID to get the protein sequences. These steps can be down by running Java program PROVEAN1 and PROVEAN2.

### 3 New Module

In order to study the indel annotations, it is necessary to annotate the regions of interest. For an easy and fast way to annotate the sequences stored in the Indelanno, an annotation module was created. The Indelanno pipeline is presented in Figure 2.



*Figure 2. Indelanno Pipeline*

#### 3.1 Input

The input provided to the annotation pipeline is expected to be a VCF file. VCF is a text file format that contains meta-information lines, a header line and the data lines each containing information about a position in the genome. The header line includes 8 mandatory columns, which are #CHROM, POS, ID, REF, ALT, QUAL, FILTER, and INFO. All data lines are tab-delimited. If there is a missing value, it will be specified with a dot (“.”). The CHROM presents chromosome that is an identifier from the reference



genome. POS means the reference position with the 1<sup>st</sup> base as position 1. REF and ALT present the reference and alternative alleles respectively.

### **3.2 Pipeline Realization**

To facilitate community access, we built an Amazon Machine Image (AMI) to run Indelanno in the cloud through Amazon Web Services (AWS). The user has access to the pipeline through an instance of the AMI. All depended software and programs, such as ANNOVAR, CADD, SIFT INDEL and PROVEAN, have already been installed in the AMI. Five java programs for this pipeline (Indelanno, Indelanno2, SIFT1, PROVEAN1, and PROVEAN2) are also in the AMI and ready to run.

### **3.3 Running Pipeline**

To run Indelanno, users only need to upload a VCF file. The annotation pipeline can be run with only two command line calls. The first call specifies the input/output files and score options and produce a shell script. The second call runs the shell script and produces the annotation results. Since Indelanno is written in Java, so most of its annotation modules can run easily across different platforms.

The users choose which prediction software they want to use in the first command line. After running the first command, two output files are generated. One output file is the input.txt file including four columns, with header CHROM, POS, REF, and ALT. Another output file is scripts.sh shell script file, include prediction score software commands. Then the user runs the second command line to add the prediction score to the genome information table to get the last output file.

The first Command format is java Indelanno Customer's input file (VCF file)  
input.txt scripts.sh Predictionscoretools

The first Command format is

*Java Indelanno [Customer's input file (VCF file)] input.txt script.sh  
Predictionscoretools.*

The input.txt is the name of the output variant list file including four columns (CHROM, POS, REF, and ALT) and scripts.sh is the name of the shell script to be run in the next step. Predictionscoretools are CADD, SIFT, PROVEAN or their combinations. For example, if the user wants to run CADD and PROVEAN with input file *my.vcf*, the command is java Indelanno *my.vcf* input1.txt scripts.sh CADDPROVEAN. The order of these words CADD, SIFT, PROVEAN in the last parameter does not matter.

The second Command format is

*bash scripts.sh.*

### **3.4 Outputs**

If the user runs one prediction score tool, the output file is input.txt0. If the user runs two prediction score tools, the output file is input.txt01. If the user runs two prediction score tools, the output file is input.txt02.

## 4 Case Study

In this chapter, we report on the results of our case study by using the implemented annotation pipeline on the human genome.

### 4.1 Pipeline Configuration

The input file provided to the annotation pipeline was `inputsample.vcf` (Figure 3), which consisted of the human genome information, including frameshift deletion/insertion and non-frameshift deletion/insertion, in VCF format. Since the user can choose three deleterious prediction scores in the first command line, in this case, we chose CADD, SIFT, and PROVEAN. All steps were accomplished by two single command line calls. The first command line was `java Indelanno CosmicCodingMuts.vcf input.txt script.sh CADD SIFT PROVEAN`. The second command line is `bash script.sh`.

### 4.2 Results

After running the first command, there are two output files, `input.txt`, and `scripts.sh`. They are shown in Table 1 and Figure 4 respectively. And there are multiple command lines in `scripts.sh` for the further analysis. First, since the ANNOVAR database have already been downloaded and so it runs ANNOVAR annotation to get three output files (`Ex1.variant_function`, `Ex1.exonic_variant_function` and `Ex1.ann.log` log file). From `ex1.exonic_variant_function` file, we got ENST IDs and amino acid variants lists from the third column. Running Java program PROVEAN1 will search protein sequences by using ENST IDs and generate a `Proveanoutput.sh` shell script, which includes the command to run PROVEAN, the `Proveanoutput.sh` is presented in Figure 5. Since PROVEAN output

files only include the amino acid variant names and the PROVEAN scores. Thus, there is a need to connect the PROVEAN scores and VCF file table by running Java program PROVEAN2. Proveanresult.txt was then generated including five columns shows the chromosome, position, ref, alt, and the PROVEAN scores. Next, run Java Program Indelanno2 to add the PROVEAN scores to the last output table and get input.txt0 file (Table 2).

Running CADD by using CADD.sh command, and then run Java program Indelanno2 to add the CADD scores to the last output table and get input.txt01 (Table 3).

```
X      2828729      1179924      TG      T
X      36162684     rs112700338,967118 C      CTG
X      114425181     972622      A      AGAGGCCGCTCGCCCAACGCCACAGCG
X      119070327     115103      CGAT     C
```

*Figure 3. inputsample.vcf*

*Table 1. input.txt*

#CHROM	POS	REF	ALT
X	2828729	TG	T
X	36162684	C	CTG
X	114425181	A	AGAGGCCGCTCGCCCAACGCCACAGCG
X	119070327	CGAT	C

```
#!/bin/sh
perl annovar/convert2annovar.pl -format vcf4 inputsample.vcf > ex1.avinput
perl annovar/annotate_variation.pl -geneanno -dbtype ensGene -out ex1 -build hg19 ex1.avinput humandb/
java Provean1 ex1.exonic_variant_function /indelanno/Proveanoutput.sh
bash Proveanoutput.sh
java Provean2 input.txt ex1.exonic_variant_function /indelanno/PROVEAN/provean-1.1.5/examples/ENST.res
Proveanresult.txt
java Indelanno2 input.txt Proveanresult.txt input.txt0 PROVEAN
bash /indelanno/CADD/CADD-scripts-master/CADD.sh inputsample.vcf
gunzip inputsample.tsv.gz
java Indelanno2 input.txt0 inputsample.tsv input.txt01 CADD
java Sift1 inputsample.vcf /indelanno/siftinput.txt
perl /indelanno/SIFT/SIFTINDEL/bin/SIFT_exome_indels.pl -i /indelanno/siftinput.txt -c hs37 -d
/indelanno/SIFT/SIFT_INDEL_HG37 -m 1 -o /indelanno/SIFT/output
siftout=$(find /indelanno/SIFT/output -name `ls -ltrR /indelanno/SIFT/output |
grep predictions.tsv | tail -n 1 | awk '{print $9}'`)
java Indelanno2 input.txt01 $siftout input.txt012 SIFT
```

*Figure 4. Script.sh*

```
cd /indelanno/PROVEAN/provean-1.1.5/examples/
provean.sh -q /indelanno/PROVEAN/provean-1.1.5/examples/sep/ENST00000424776 -v
/indelanno/R393delinsRGRSPNAHSG.var --save_supporting_set ENST00000424776R393delinsRGRSPNAHSG.sss
--num_threads 2| tail -n 1 > ENST00000424776R393delinsRGRSPNAHSG.res
cd /indelanno/PROVEAN/provean-1.1.5/examples/
provean.sh -q /indelanno/PROVEAN/provean-1.1.5/examples/sep/ENST00000371410 -v
/indelanno/S201del.var --save_supporting_set ENST00000371410S201del.sss
--num_threads 2| tail -n 1 > ENST00000371410S201del.res
```

*Figure 5. Proveanoutput.sh*

*Table 2. input.txt0*

#CHROM	POS	REF	ALT	PROVEAN
X	2828729	TG	T	N/A
X	36162684	C	CTG	N/A
X	114425181	A	AGAGGCCGCTCGCCCAACGCCCACAGCG	3.427
X	119070327	CGAT	C	-1.629

Table 3. *input.txt01*

#CHROM	POS	REF	ALT	PROVEAN	CADD
X	2828729	TG	T	N/A	6.200
X	36162684	C	CTG	N/A	1.943
X	114425181	A	AGAGGCCGCTCGCCCAACGCCCACAGCG	3.427	1
X	119070327	CGAT	C	-1.629	0.625

For SIFT, since SIFT only accepts space-based coordinates for insertion/deletion variants. It is necessary to convert the VCF file to SIFT input file format by running Java program SIFT1 and get siftinout.txt, which was presented in Table 4. After running SIFT Indel predictions, run Java Program Indelanno2 to add the SIFT result to the last output table and get input.txt012 (Table 5), which is the last output table showed three indel prediction results.

Table 4. *siftinput.txt*

X,2828729,2828730,1,G/
X,36162684,36162684,1,TG
X,114425181,114425181,1,GAGGCCGCTCGCCCAACGCCCACAGCG
X,119070327,119070330,1,GAT/

Table 5. input.txt012

#CHROM	POS	REL	ALT	PROVEAN	CADD	SIFT	SIFT Confidence Score
X	2828729	TG	T	N/A	6.200	damaging	0.858
X	36162684	C	CTG	N/A	1.943	neutral	0.918
X	114425181	A	AGAGGCCGCTCGCCCAACGCCACAGCG	3.427	1	neutral	0.696
X	119070327	CGAT	C	-1.629	0.625	neutral	0.961

## 5 Conclusion and Future Work

Currently, researchers already have many different algorithms to do the deleterious prediction, however, those methods may not have a consensus. Even when they are used to analyze the same sequence data, different prediction tools may get different results and their relative advantages in practical application are still unclear. Therefore, collecting prediction scores from multiple algorithms can contribute to more accurate SNVs and indels functional prediction. The goal of this thesis is to extend the functional annotation database, dbNSFP, which have already supported SNV annotations. In this project, we have written five Java program codes and added three deleterious prediction scores (CADD, SIFT, and PROVEAN) to indel annotation and integrated them to an automatic annotation pipeline called Indelanno. We built an Amazon Machine Image (AMI) to run Indelanno in the cloud through Amazon Web Services (AWS). Users can upload a VCF file and run Indelanno with two command line calls. Since Indelanno can search several annotation resources in batch and produce an integrated report in return, it will be a useful tool for the researcher of sequencing-based studies.



## 6 References

- Choi Y, Sims GE, Murphy S, Miller JR, Chan AP. 2012. Predicting the functional effect of amino acid substitutions and indels. *PloS One* 7: e46688.
- Kircher M, Witten DM, Jain P, O’Roak BJ, Cooper GM, Shendure J. 2014. A general framework for estimating the relative pathogenicity of human genetic variants. *Nat Genet* 46: 310–315.
- Li Q, Wang K. 2017. InterVar: Clinical Interpretation of Genetic Variants by the 2015 ACMG-AMP Guidelines. *Am J Hum Genet* 100: 267–280.
- Liu X, Jian X, Boerwinkle E. 2011. dbNSFP: A lightweight database of human nonsynonymous SNPs and their functional predictions. *Hum Mutat* 32: 894–899.
- Liu X, Jian X, Boerwinkle E. 2013. dbNSFP v2.0: a database of human non-synonymous SNVs and their functional predictions and annotations. *Hum Mutat* 34: E2393-2402.
- Liu X, White S, Peng B, Johnson AD, Brody JA, Li AH, Huang Z, Carroll A, Wei P, Gibbs R, et al. 2016a. WGSA: an annotation pipeline for human genome sequencing studies. *J Med Genet* 53: 111–112.
- Liu X, Wu C, Li C, Boerwinkle E. 2016b. dbNSFP v3.0: A One-Stop Database of Functional Predictions and Annotations for Human Nonsynonymous and Splice-Site SNVs. *Hum Mutat* 37: 235–241.

- Quang D, Chen Y, Xie X. 2015. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics* 31: 761–763.
- Vaser R, Adusumalli S, Leng SN, Sikic M, Ng PC. 2016. SIFT missense predictions for genomes. *Nat Protoc* 11: 1–9.
- Hu J, Ng PC. “SIFT Indel: Predictions for the Functional Effects of Amino Acid Insertions/Deletions in Proteins.” *PLoS ONE*, vol. 8, no. 10, 2013, doi: 10.1371/journal.pone.0077940.
- Hu J, Ng PC. “Predicting the Effects of Frameshifting Indels.” *Genome Biology*, vol. 13, no. 2, 2012, doi:10.1186/gb-2012-13-2-r9.
- Agajanian, S, Odeyemi O, Bischoff N, Ratra S, Verkhivker GM. “Machine Learning Classification and Structure–Functional Analysis of Cancer Mutations Reveal Unique Dynamic and Network Signatures of Driver Sites in Oncogenes and Tumor Suppressor Genes.” *Journal of Chemical Information and Modeling*, vol. 58, no. 10, 2018, pp. 2131–2150., doi:10.1021/acs.jcim.8b00414.
- Kim, JK, Yeom M, Hong JK, Song I, Lee YS, Guengerich FP, Choi JY. “Six Germline Genetic Variations Impair the Translesion Synthesis Activity of Human DNA Polymerase  $\kappa$ .” *Chemical Research in Toxicology*, vol. 29, no. 10, 2016, pp. 1741–1754., doi:10.1021/acs.chemrestox.6b00244.
- Fang, H, Bergmann EA, Arora K, Vacic V, Zody MC, Iossifov I, O’Rawe JA, Wu Y, Jimenez Barron LT, Rosenbaum J, Ronemus M, Lee YH, Wang Z, Dikoglu E, Jobanputra V, Lyon GJ, Wigler M, Schatz MC, Narzisi G. “Indel Variant

Analysis of Short-Read Sequencing Data with Scalpel.” *Nature Protocols*, vol. 11,  
no. 12, 2016, pp. 2529–2548., doi:10.1038/nprot.2016.150.

Mingyao Lu was born in Songyuan, Jilin, China. She went to Songyuan experimental high school, Songyuan, Jilin, China in 2008 and after she completed her study, she entered Shanxi University in Taiyuan, Shanxi Province China. After two years, she entered University of Minnesota Duluth (UMD) in Duluth, MN to continue her bachelor's degree. She got her bachelor's degree in Biology major with Chemistry minor from UMD in May 2016. She entered the University of Texas MD Anderson Cancer Center UTHHealth Graduate School of Biomedical Sciences for her Master of Sciences Degree in Bioinformatics studies.

Permanent address:

1885 El Paseo St

Houston, TX, 77054