最终报告是一份30到50页的文档，描述了项目的设计，实施，测试和评估。最后，您还应该指出一些进一步改进的建议，并提供最终的关键评估。您应该对在项目开发过程中遇到的问题以及如何解决和解决这些问题进行评论。您可能包括参考材料的附录（除了30-50页的余量）。请注意，附录将不会进行评估。最终报告包括：

* 一个单独的标题页，包含标题，您的姓名，一个100-200个单词的摘要以及一个签名的声明，指出：本人证明本论文中的所有材料（不是我自己的作品）均已被识别。
* 目录，紧跟在标题页之后。
* 报告的正文，不得超过50个A4页，且字体不少于11点且页边距不少于2cm。使用单行间距。表格和图形可以具有9点字体。
* 参考书目，字体不少于11pt；预计此部分将至少占据A4页的一半。
* 附录（可选）。这些可以用来显示例如实验数据，UML图，数据流图，用户手册或屏幕截图，出于完整性考虑，您希望将其包括在内。但是，报告本身必须是独立的，并包括主要结果和所有必要的证据。

该报告应该是一个结构化的文档，其中包括引言，方法或规范部分，数据部分，实验或验证部分以及结论。

在简介中，您应该描述问题并解释其重要性。

在方法部分，您应该描述所使用的分析技术，并说明为什么选择该特定技术的理由。相反，软件开发项目将包括一个规范部分，其中包含有关设计选择的详细信息，包括高级UML /数据流图。

在数据部分中，您应该描述数据源和正在处理的数据的特征。

在实验部分，您应该描述选择的性能评估技术，并报告相应的数值结果。在软件开发项目中，您应该报告用户评估。

在总结中，您应该总结项目中发现的问题以及如何解决和解决这些问题。您还应该提出进一步改进的建议，并对您的工作进行严格的评估。

**评分标准**

该项目将根据其满足以下标准的程度进行评估：

* 范围和复杂性：该项目具有足够的难度，并且雄心勃勃，具有大量的计算机科学或数据科学内容。
* 规格和设计：已经提出了项目的明确规格和设计，例如目的和目标，业务用例，数据描述，系统架构。
* 过程：已遵循适当的开发方法。（您可以参考用于处理数据的技术选择以及解决研究/业务问题的算法/软件开发过程。）
* 测试/实验评估：在项目开发过程中使用了适当的测试/实验方法。通过用户测试，反馈，与文献进行比较，实验等进行产品评估，视情况而定。）
* 产品和目的性：适当地使用图表，图形，表格，参考文献等，该文档的外观精美，可读性好，英文很好。
* 演示：适当地使用图表，图形，表格，参考文献等，该文件的外观精美，可读性好，英文很好。
* 长度：文档长度（不包括参考文献）超过50页的每页得分-5

**Research on new causes of genetic disease based on data analysis**

**Abstract**

Genetic disease is a health problem caused by gene mutation. For example, Down syndrome is a kind of common genetic disease resulting from a redundant chromosome 21. In general, most children with it always have mental retardation or behavior abnormalities, and they are almost impossible to be cured. Unfortunately, there are still a lot of unclear genetic diseases, not to mention a cure. This study will focus on studying the relationship between hyperinsulinism, neonatal diabetes and genetic mutations. As a result, the identification of new causes of genetic disease is a critical step for people to understand them, which will be helpful for the following clinical testing and treating. The project will be based on a database of variants and screen out the possible pathogenic variants through scripts. The whole project will be divided into three steps and eventually try to find the variants that are suspected to be pathogenic.

Content:

目录

[1. Research Background 4](#_Toc50342448)

[1.1 Introduction to hyperinsulinism 5](#_Toc50342449)

[1.2 Introduction to neonatal diabetes 5](#_Toc50342450)

[2. Basic knowledge 5](#_Toc50342451)

[2.1 Genes, DNA and chromosomes 5](#_Toc50342452)

[2.2 Genic Mutation 6](#_Toc50342453)

[2.3 Blood sugar regulation 7](#_Toc50342454)

[2.4 Genbank and OMIM 7](#_Toc50342455)

[3. Main Tasks and Method introduction 8](#_Toc50342456)

[3.1 Task 1: 9](#_Toc50342457)

[3.2 Task 2: 10](#_Toc50342458)

[3.3 Task3: 11](#_Toc50342459)

[4. Model and Results 11](#_Toc50342460)

[4.1 Task 1 11](#_Toc50342461)

[4.2 Task2 14](#_Toc50342462)

[4.3 Task3 17](#_Toc50342463)

## Research Background

Gene mutations and diseases are always inextricably linked. In the long-term research, people have made many important discoveries. First of all, it is very important to understand the basic structure of genes. By definition, genes are made up by exons and introns. During transcription, exons are the part of DNA that finally gets coded for in a section of messenger RNA and introns are noncoding regions. Messenger RNA controls the formation of proteins, which is the expression of genetic genes. There are about 180,000 exons in the human genome, accounting for 1% of the genome. However, it is estimated that 85% of the genetic diseases are derived from the exons’ mutation. For example, in the research from China, variants in exon 10 and exon 7 of the TGM6 was thought to lead cerebellar ataxia. The same variant did not occur in 500 healthy normal controls, which could further determine the pathogenicity of this TGM6 gene [1]. In recent years, many studies have suggested that mutations of introns are also likely to lead to diseases. One common situation is due to GT-AG rule, which can be simply understood as the region between GT and AG is the intron. When an intron mutates and coincidentally produces new splice sites, new exons are also created. This is why these noncoding introns still affect intron coding. Research about deep intronic cryptic ABCC8 and HAHD mutations revealed the relationship between these genes and hyperinsulinaemic-hypoglycaemia [2]. Another essay in 2014 also proved that some recessive mutations might result in isolated pancreatic agenesis [3]. Based on these outstanding studies, it can be seen that the relationship between genetic variation and disease is very complicated, and further research and exploration are needed to help people learn more about genetic diseases.

### 1.1 Introduction to hyperinsulinism

Back to the two genetic diseases studied in this article: hyperinsulinism and neonatal diabetes. As for hyperinsulinism, it is necessary to need to first understand the theory of this disease. Simply put, this is a problem about glucose homeostasis. Glucokinase (GK) can sense the concentration of glucose and instruct the β cells to secrete insulin to lower blood sugar or α cells to secrete glucagon to raise blood sugar according to small changes in blood sugar. Under normal circumstances, this mechanism is to maintain insulin balance. But after certain gene mutations, the release of insulin will be out of control, leading to hypoglycemia (HI). The more serious consequence is that it may cause brain damage in newborns. Through researches, people have found four genetic mutations related to it, but there are still 50% of cases that have not been able to determine the cause [4]. Although hyperinsulinemia is classified as a rare disease, the incidence is about one in every 50,000 newborns. But this disease is not a truly rare disease for some countries with large populations, such as India, China, the United States, and Japan. This is a real disaster for every family with sick children, so it makes sense to pay more attention to these rare diseases.

### 1.2 Introduction to neonatal diabetes

Diabetes is actually a very common disease and the main feature is high blood sugar. In people's general perception, diabetes seems to mainly occur in the elderly. Different from common diabetes, neonatal diabetes often occurs in infants under 6 months of age and is a genetic disease. At present, this disease is mainly divided into two kinds, one is temporary, and the other is permanent. Temporary neonatal diabetes may disappear after a few years, but there is a risk of recurrence in the future. And permanent neonatal diabetes will follow the patient for life. Some clinical manifestations of the patient are developmental delay, epilepsy, etc. At present, people have found more than 20 known genetic causes. Although this is still a very rare disease (about one in 90,000 or one in 160,000 incidence), it is very difficult to cure in medicine [5]. Rare diseases do not mean that they are unimportant. These patients and their families often bear huge pains of illness, economic and social pressure. At the same time, research on these rare genetic diseases can call for people to pay more attention to genetic testing, so that parents can pay more attention to the health of newborns and predict some risks.

## 2. Basic knowledge

### 2.1 Genes, DNA and chromosomes

Although this article is based on data science modeling and experiments, understanding some basic biological knowledge will help understand the experimental process and judge the results. The figure below shows the relationship between chromosomes, genes and DNA.

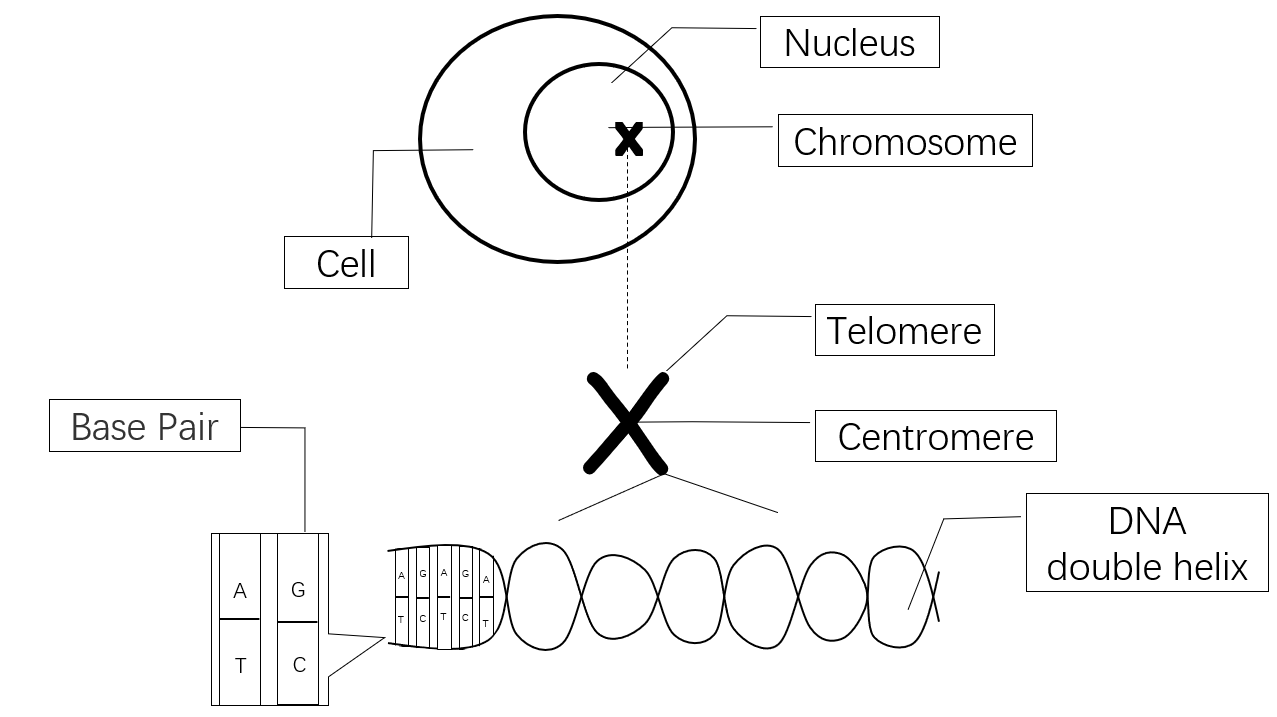


Figure 1

Human cells have 23 pairs of chromosomes in the nucleus (including 22 pairs of autosomes and a pair of sex chromosomes), which means that each cell has 46 chromatids. The stability of chromosomes is very important to human health. Although most chromosomal mutations do not have much impact on humans, Klinefelter syndrome is caused by an extra X sex chromosome. This disease may cause mental retardation or promote the development of secondary sex characteristics and cause mental instability in patients [6]. There is only one DNA molecule on each chromosome, and the chromosome is the main carrier of DNA molecules. And there are many genes on each DNA, and genes are DNA fragments with genetic effects. There are 5 common bases in organisms, namely adenine (A), guanine (G), cytosine (C), thymine (T) and uracil (U). As shown in the figure above, the bases A, G, C, and T are present in DNA, and A, G, C, and U are present in RNA. As shown in Figure 1, in double-helical DNA, each base pair contains a purine and a pyrimidine: A and T are connected by two hydrogen bonds, and C and G are paired. This phenomenon is called base complementation.

### 2.2 Genic Mutation

Genes are very important to human life. They control our life characteristics by directing the synthesis of proteins. There are two main actions of DNA on gene. One is replication: DNA is untwisted, and two strands are used as templates to synthesize two daughter strands according to the principle of complementary base pairing. The second is generating protein: DNA double-strands are opened, and one strand of the DNA is used as a template. According to the principle of base complementary pairing, single-stranded mRNA is formed, which enters the cytoplasm and combines with ribosomes. Then mRNA will direct translation to create protein. The below chart shows the processes of generating protein.

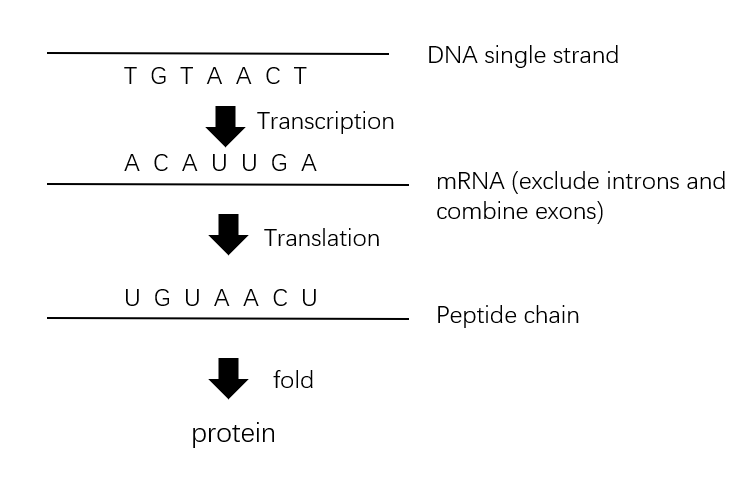


Figure 2

Gene mutation refers to a change in the base pair composition or sequence of the gene. In general, gene is very stable, it can accurately replicate itself. But under certain conditions, genes can also change. For example, a new gene suddenly appears at a locus, replacing the original gene, or deletes a certain gene. The core purpose of this article is to explore the relationship between gene mutation and disease.

### 2.3 Blood sugar regulation

Because the diseases involved in this article are closely related to blood sugar, here is a brief description of the glucose homeostasis mechanism in blood sugar regulation.

The regulation of blood sugar is mainly controlled by two hormones, insulin and glucagon. Insulin secreted by β cells will promote the conversion of blood sugar into glycogen, accelerate the oxidative decomposition of blood sugar, and inhibit the decomposition of liver glycogen and the conversion of non-sugar substances into glucose. Therefore, blood sugar will decrease. The α cells secrete glucagon to promote the conversion of non-sugar substances into glucose, leading to increased blood sugar. Glucokinase is a key regulatory enzyme that regulates β cells. Therefore, it is not difficult to judge that the mutations in the gene encoding glucokinase may cause the β cells to secrete hormones abnormally, resulting in hyperglycemia or hypoglycemia in newborns [7].

### 2.4 Genbank and OMIM

In many papers about genes, genbank often appears. GenBank is a very comprehensive DNA sequence library established by the National Center for Biotechnology Information, which is calculated from biologists or large-scale gene sequencing projects [8]. For example, about the mutations in the gene encoding glucokinase, NM\_000162.2 is regarded as mutated gene. According to Genbank’s information, it is very easy for people to know its meanings: Homo sapiens glucokinase (GCK), transcript variant 1, mRNA. There are more information that can get and the following table shows some of them：

|  |  |  |
| --- | --- | --- |
| **Key words** | **Example of NM\_000162.2** | **Definition** |
| LOCUS | NM\_000162 | The identifier of the sequence |
| DEFINITION | Homo sapiens glucokinase (GCK), transcript variant 1, mRNA. | The meaning of the sequence |
| VERSION | NM\_000162.5 | Version number |
| KEYWORDS | RefSeq; MANE Select | The product of the gene sequence and other related information |
| SOURCE | Homo sapiens (human) | Source of organism or tissue |
| REFERENCE | bases 1 to 2745 | Related documents in the sequence |
| FEATURES | Source, gene, exon… | Describe sequence characteristics in detail |
| BASE COUNT | / | Count the number of times different bases appear in the sequence |
| ORIGIN | actccacacc tggctggag … | The complete sequence expressed with ‘a’, ‘t’, ‘c’, ‘g’ |

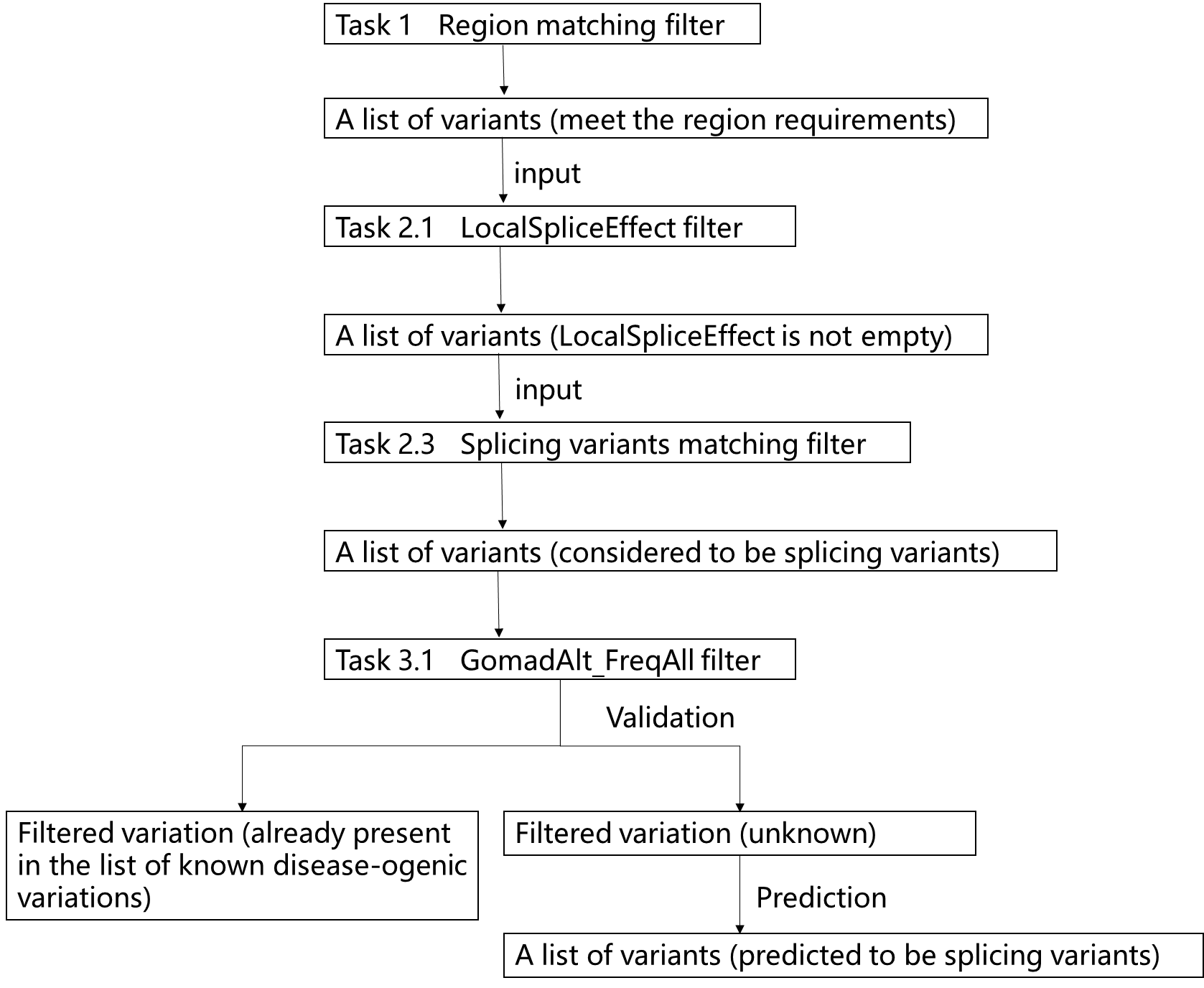
<https://www.ncbi.nlm.nih.gov/nuccore/NM_000162.5/>

In addition, people also generated the OMIM (Online Mendelian Inheritance in Man) database to query the relationships between genes and phenotypes of various genetic diseases. Still taking glucokinase mutation as an example, on the OMIM database, we can obtain the following valid information about mutation MIM # 138079, such as the phenotypes, gene location, Inheritance and so on.

<https://omim.org/entry/138079?search=MIM%20%23138079&highlight=138079%20mim#text>

## Main Tasks and Method introduction

The whole project is divided into three steps, the purpose is to find out the pathogenic genes related to the target disease. The task1 is to screen out some variants through some suspicious gene sequence regions. The Task 2 is to screen again on the basis of task one, mainly to find those variants that generate new introns and affect gene transcription. The third task is to comprehensively analyze the found variants, and then make the final screening and judgment. It needs to be stated that all data is provided by the University of Exeter School of Medicine. The following is a flowchart of the task and the expected results：



### 3.1 Task 1:

The first task is to identify the variants. There are two documents, the first is the list of variants for a patient we think it particularly likely to have a splicing variant, the second is the list of regions in the human genome and their labels. The chart shown below is part of list of variants, which contains the data of millions of variants and each variant contains 230 attributes. In Task1, only these two attributes will be considered. The first column is the chromosome number, and the second column is the location of the gene variants.

|  |  |
| --- | --- |
| chrom | inputPos |
| 2 | 88883240 |
| 11 | 17420822 |
| 11 | 17430159 |
| 11 | 17438981 |

The following chart is the part of list of regions. The first column is the chromosome number, the second and third columns form a region, and the fourth column represents the labels of the region. The document contains about 200,000 regions which are thought to control the regulation of genes. They correspond to some biological labels, such as the "Inactive\_open\_chromatin\_regions" label, which means an invalid open chromatin interval, such as the "CTCF" label, which represents the transcription factor encoded by the CTCF gene.

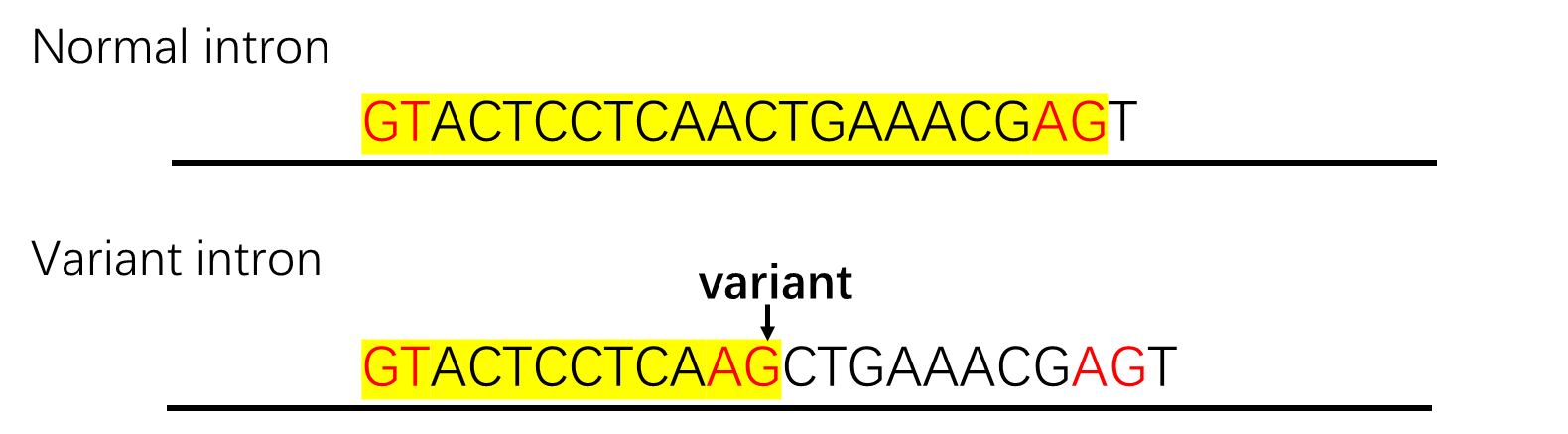
|  |  |  |  |
| --- | --- | --- | --- |
| Chrom | Start\_Pos | End\_Pos | Labels |
| 1 | 9915 | 10774 | Inactive\_open\_chromatin\_regions |
| 1 | 237616 | 237917 | Strong\_CTCF |
| 1 | 521430 | 521703 | Strong\_CTCF |
| 1 | 713698 | 714591 | Active\_promoters |
| 1 | 740024 | 740277 | Active\_enhancers\_III |
| 1 | 740380 | 740590 | Active\_enhancers\_III |
| 1 | 762202 | 763371 | Active\_promoters |
| 1 | 766176 | 766433 | Active\_enhancers\_I |
| 1 | 766734 | 767227 | Active\_enhancers\_I |

Regions

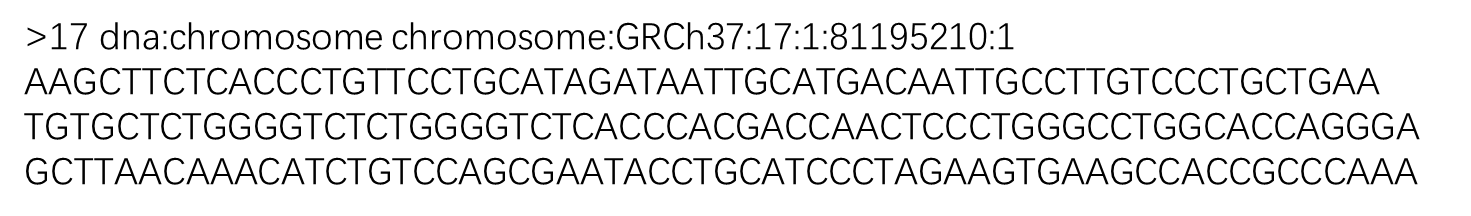
In other words, a region of DNA has been given in the region list (e.g., from position 9915 to position 10447), and then we need to find out in the first list which variants occur in this range. This task is based the studies that have been completed, which is equivalent to narrowing the range of options for variants.

### 3.2 Task 2:

As mentioned before, the boundaries of exon are encoded by specific patterns called splice sites in the DNA sequence (AG and GT). As mentioned earlier, the region between GT and AG is an intron region. When introns mutate and accidentally create new splice sites, new exons are also created. Therefore, even though introns are non-coding regions, their mutations still affect gene transcription. The following is a simple schematic:



So, the task2 is finding splicing variants. Specifically, we need to check the “localSpliceEffect” of the variant first. This attribute has the following situations: Cryptic Acceptor Strongly Activated, Cryptic Acceptor Weakly Activated, Cryptic Donor Strongly Activated, Cryptic Donor Weakly Activated, New Acceptor Site, New Donor Site and blank. If it is not blank, which means the variant creates a splice site. The next step is to check if it generates AG or GT. Assuming the variant produces AG, which is the splice site on the left. So, the following job is to use pattern matching on the DNA sequence surrounding the variant to make sure if there is a GT on the right. This task will use a fasta format database, which is the sequence of the human genome. The fasta format is a format that uses text to represent nucleic acid sequences or polymorphic sequences and is widely used in bioinformatics. The first line of each basic unit of the fasta file is the sequence description starting with "> ", and the specific sequence starts from the second line. In this task, the file contains the sequences on 22 chromosomes. The following is an example of part information of 17 chromosome.



To sum up, the task is to first screen out the variants that the label “localSpliceEffect” is not empty, and then analyze whether new introns are generated through specific variants’ situation. For example, the variant makes a GT which is known to be on the right hand splice site. We need to pattern match to the left to see if there is an AG nearby like this sequence :

TATCTAGTTATGTATCATGCTGTACATATGAGGT

On the contrary, If the variant made an AG, we need to pattern match to the right to see if there was a nearby GT.

### 3.3 Task3:

There are two parts of task3: the first part is to do the final round of screening based on biological knowledge, and the second part is to verify the screening results and make predictions.

#### 3.3.1 Last Round Filter

After finishing tasks 1 and 2, there will generate a new list of variants. In order to continue to narrow the range, some attributes can be used to filter out some variants. The screening methods is to use “gnomadAltFreq\_all”, which means the frequency of the variant in a large sample of people. As the genetic disease is very rare in the population, so some variants with higher frequency in the population can be excluded.

#### 3.3.2 Validation Design

After the final screening, it is necessary to validate the results after three filters. Here is a list of known pathogenic variants named “WG0514-WG0515” from Exeter Medical College, which is the list of already known pathogenic variants, can be used for comparison and validation. In fact, these variations have been filtered multiple times by Task 1, Task 2, and the previous part of Task 3. So we can infer that the result is quite accurate. On this basis, the remaining experiment will be divided into two parts:

1. First determine whether there are any variants in the filtered list that are already in the list of pathogenic variations, which are determined to be the correct variations for screening.

2. As for those variations that are not in the list of pathogenic variations, we will build models to predict their pathogenicity. Perhaps some unknown pathogenic genes have been screened out, and it is meaningful to predict their pathogenicity.

In order to predict disease-causing variation, a machine learning model is needed to complete the prediction work. This problem is a two-category prediction problem, from the functional point of view of the machine learning models, the decision tree and neural networks can achieve this effect.

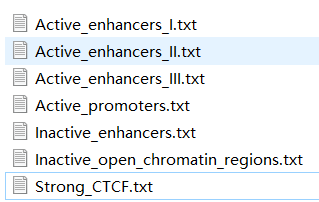
## Model and Results

### 4.1 Task 1

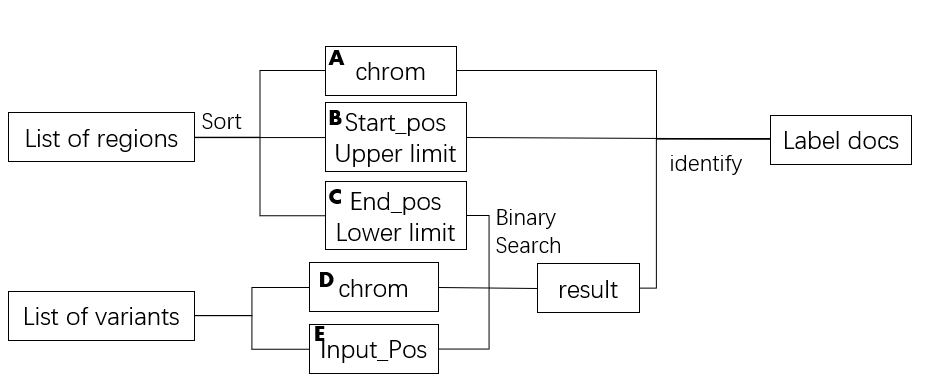
Although the data for this task is based on biology, we can temporarily ignore the biological meaning of these data to complete this task.

#### 4.1.1 Algorithm model analysis and establishment

The result I achieved was the files classified according to interval labels as shown in the figure. Each file will contain all variants that match the region labels we are interested in. And in the next task, we can use code to call the content. This result will serve as an important basis for future tasks.



When choosing a programming language, both C# and python are be considered, but the weak type characteristics of python will make it more convenient to store different kinds of data, and it happens that many data in this article are both strings and numbers. If using C#, a coding language with great emphasis on data types, it will be necessary to create an additional method based on the Object base class, so python will be the main language. The following flowchart shows my code idea.



Based on this idea, after previewing the data, it is obvious that these regions do not overlap each other and are in order. The specific rule is that on each chromosome, the regions are arranged in order. We first sort all the regions and read the region list into 3 data which are the chromosome numbers[A], the upper limit of the regions[B] and the lower limit of the regions[C]. Because the regions of about 200,000 rows is relatively small compared with the millions of variants, we read in 200,000 regions at once. At the same time, read the list of variants into a list of chromosomes[D] and a list of positions[E]. In order to ensure that the memory space of computer is sufficient to run, each time we will take out the corresponding set of data from the two lists [D] and [E] to match. The first match is the position data[E] and the lower limit of the interval[C]. Since we have sorted the intervals before, we can use a more efficient binary search method to find the lower limit of the interval where the location data is located. Then, we compare the position data[E] with the upper limit[B] corresponding to the region, and also match the chromosome number[D] corresponding to the regions’ number[A]. If the matching is successful, the variant information will be entered into the document related to the region labels. The below picture shows part of my results and the number

of variants in each region.



When designing this idea, the main consideration is how to complete the task more efficiently. In terms of time complexity, the time complexity of the binary search method is log2N, based on the matching of 20W rows of data in this article, which means that it only takes about 18 searches to match the corresponding regions. If we match line by line, then at least a few thousand times, as many as tens of thousands of times to find the target interval. In terms of space complexity, extracting the variant lists [D][E] line by line will help to save running memory and prevent memory overflow due to excessive data.

#### 4.1.2 Result analysis

After the results are obtained, a detailed analysis of the data will help later experiments. The following is the statistics about the number of variants on each chromosome.

As can be seen from the figure above, the variation of chromosomes 1 to 10 accounted for 67.71% of the total variation. Among the variants we are interested in, there are some variants on the sex chromosomes, about 1.82%. Therefore, it can be roughly inferred that gender may have some relationships with hyperinsulinemia and neonatal diabetes, although the percentage is not very large. In fact, in a set of comparative experiments, gender is considered to be one of the causes of hyperinsulinemia [9]. The sample of 67 people in this experiment may be considered a small sample in common diseases, but in fact, it is a relatively large sample in this rare disease. Of these 67 patients, 43 of them were men and 24 were women. This ratio is about 2:1. Therefore, we can make a hypothesis that there may be pathogenic variants on the Y chromosome unique to men, and subsequent experiments will pay more attention to the variants on the Y chromosome.

### 4.2 Task2

#### 4.2.1 filter “localSpliceEffect”

On the basis of the first task, we perform a second screening, leaving the variants that “localSpliceEffect” was not empty. In this step, we still use python to read each line of each region label file, treat each line as an array separated by spaces, and determine whether the corresponding attribute is empty. Finally, save all lines whose target attributes are not blank to a new file. The following chart is the statistics about this filter.

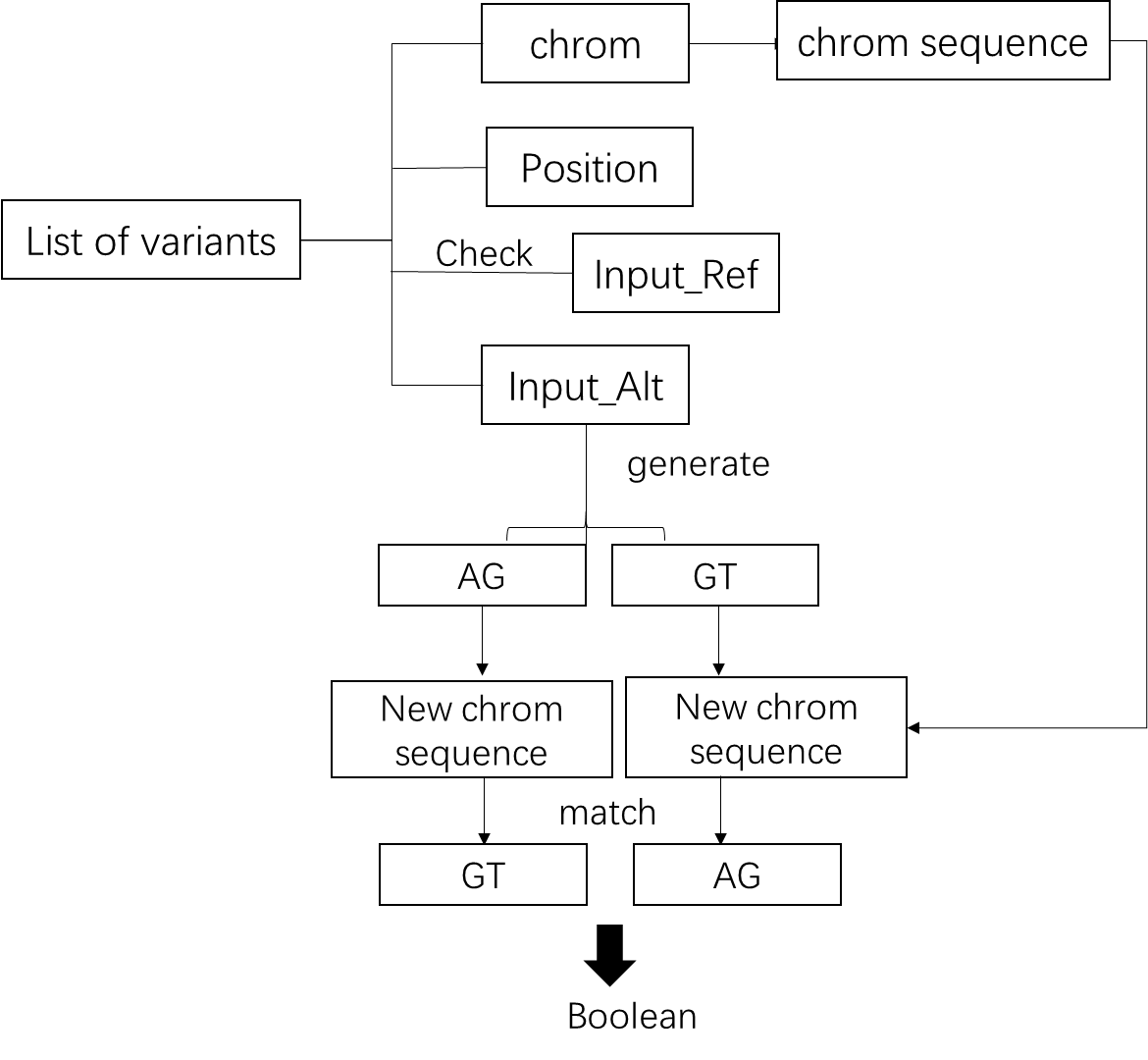


It is not difficult to find that the filtering at the beginning of the task2 is very effective, almost every region has filtered more than 80% of the variation. This is a very useful foundation for subsequent work. Based on the current situation, the fewer variants left in the later stage, the more we can find the most critical and suspicious pathogenic variants. But after this screening, the variant on the Y chromosome was eliminated.

#### Finding splicing variants

Since the amount of data of gene sequence is very large, in order to save the time of browsing the gene sequence each time, first divide the gene sequence into 24 files according to the chromosome number. In this way, the chromosome number can be matched first, and then the corresponding file can be read, instead of reading the entire gene sequence file every time. Like the rules of fasta files mentioned earlier, the sequence of each chromosome will start with "<" + "chrom number", so it is very convenient to use python to read line by line and write the file separately. The specific result is 24 txt files containing the gene sequence of each chromosome.

The next step is that a script needs to be designed to determine whether the variants that produces the splicing site has a matching splicing site. The following figure shows the idea of the script.



This experiment will use the four attributes of the variant list: chromosome number, mutation location, reference value, and change value. First, we need to match the correct chromosome sequence file. Secondly, in order to ensure the accuracy of the subsequent experiments, it is necessary to check whether the reference value of the mutation position is consistent with the sequence file. The next step is to remove the base at the original position and replace the mutated part. In fact, according to the observation of the mutation list, there are at least four kinds of situation of changes:

* one base mutates into another,
* one base mutates a sequence of bases,
* a sequence of bases mutates a base.

- a sequence of bases mutates a sequence of bases.

The difficulty encountered at the beginning is how to deal with the four variants. The earliest scripts used wrote different methods for different situations, but this resulted in a lot of code duplication and redundancy. So in the end, in order to streamline the code, but also to make the code more universal, we no longer pay attention to the ways of mutation, but directly generate a new sequence after the any situation of changes., no matter what the mutation situation is. Then we find the AG or GT splicing site generated by the variant under the position in the new sequence. If AG is generated, then we need to look for GT on the right side of the sequence. Similarly, if a GT is generated, the AG on the left needs to be found. The problem is how many bases to the left or right are more appropriate. According to a data analysis on genes [10], it is safer to set the search length to 600, and according to manual inspection, it is often found when looking for the 20th or 30th. Finally, a boolean value is returned to reflect whether splicing variants have been found.

The following is the result of script filtering：



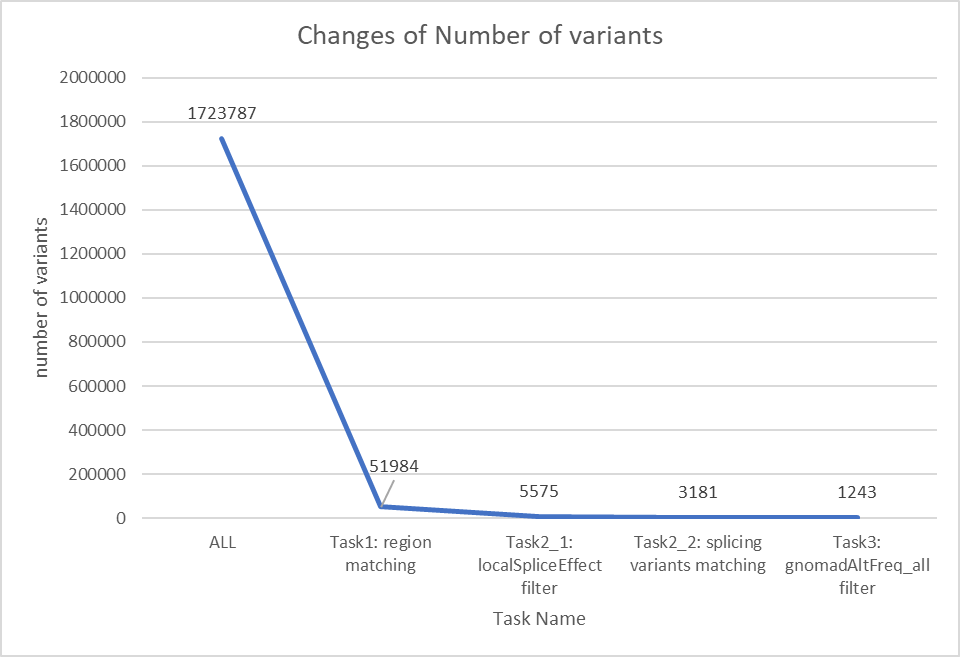
After task2's screening, only 3181 mutations were left so we no longer classifies and stores these variants, but puts them in a document. Among them, the main mutations are still concentrated on chromosomes 1, 2, and 3, accounting for 12.07%, 9.02% and 8.83% respectively.

### 4.3 Task3

#### 4.3.1 Last Round Filter

The final round of screening is mainly based on the frequency of gene mutations. There is a string of attributes starting with "gnomAD" in the list, and they represent Genome Aggregation Database. The data set provided by the database includes whole-exome sequencing data of 123,136 individuals and whole-genome sequencing data of 15,496 individuals. These data are derived from the experimental results of various disease research projects and large population sequencing projects around the world. In the list used in this article includes the frequency of all species (All), Africans, East Asians, South Asians, Jews, etc.

Since hyperinsulinemia and neonatal diabetes are rare diseases and have no obvious regional characteristics, so we use gnomadAltFreq\_all data, with 0.00004 as the limit, which means that there is one patient in 25,000 people. If the gnomadAltFreq\_all of a variant is greater than 0.00004, it will be regarded as not a rare disease and excluded. If gnomadAltFreq\_all is empty, keep the variant temporarily. The screening script this time is similar to the first part of the second task and they are all based on a certain attribute. The following is a summary and analysis of the results of all previous screenings.



After the final round of screening is completed, only 0.07% of the mutation is kept, which shows that the screening effect is very significant.

#### 4.3.2 Validation and Prediction

##### 4.3.2.1 Validation

Among the 1243 variants we got, first we need to compare with the known pathogenic variant list "WG0514-WG0515" to confirm the percentage of correct judgments. After comparison, 216 variants are known variants. Their variation position, chromosome number, and variation reference value and change value are the same, accounting for 17.37% of our total filter. In other words, our screening scripts can effectively find target variants. The following are the names of 31 genes that fully match the list of known pathogenic variants.



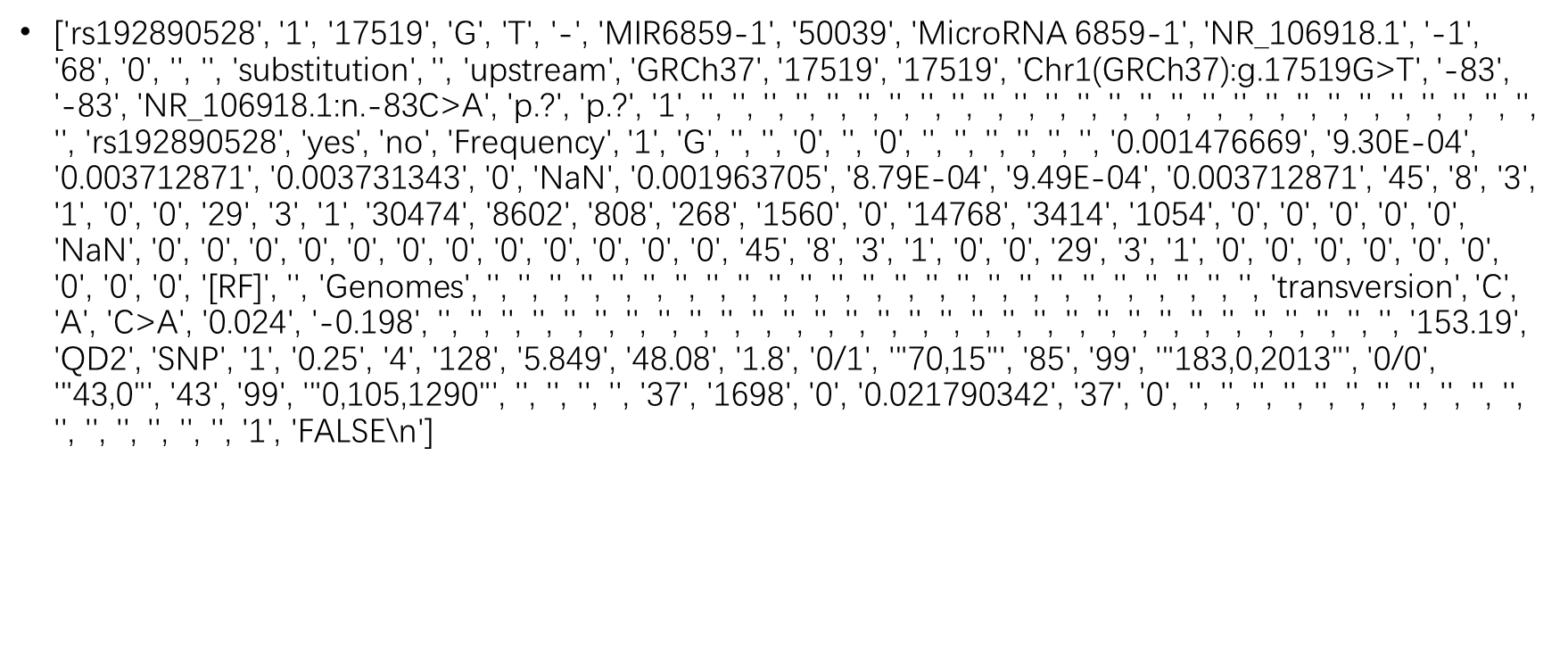
The following are the remaining genes are still considered that may have pathogenic variants.



##### 4.3.2.1 Prediction Model

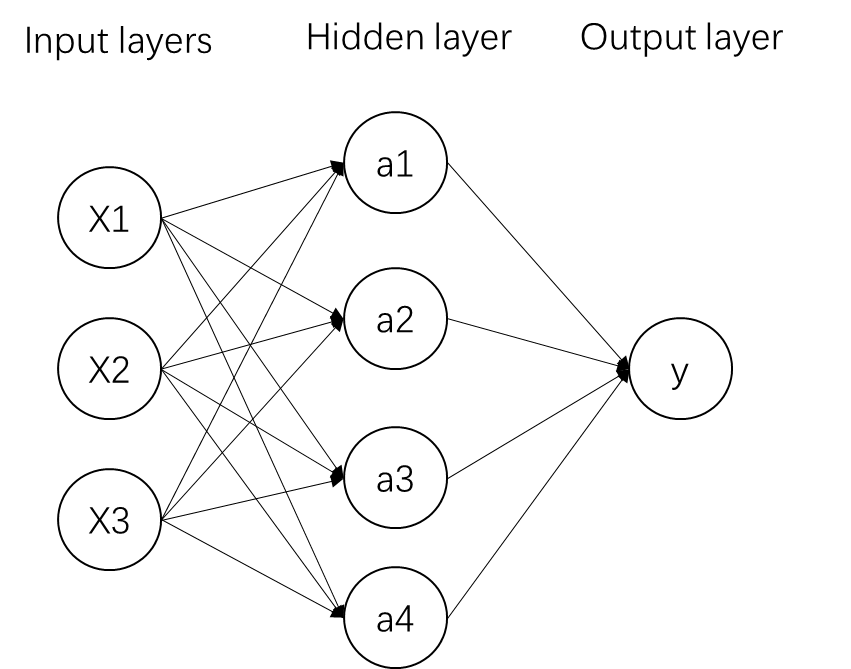
Although these genetic variants are biologically difficult to confirm whether there may be pathogenic mutations, we can still use machine learning algorithms to make judgments and predictions. When modeling, a very important question is which machine learning algorithm is used to accomplish this task. First of all, it is confirmed that we need to adopt a supervised learning method, because these variants and characteristics may be difficult to distinguish clearly, not to mention that we have a list of known pathogenic variants that may help us make more accurate judgments. So, we will not consider the use of unsupervised learning algorithms for this time.

The purpose of this question is to classify the variant, that is, to predict whether it is a pathogenic variant. The choice of algorithm needs to be determined according to the characteristics of the data used in this article, that is, the high-dimensional sparse matrix. The following shows a piece of mutated information, which means a row of the matrix. It can be seen that there is a lot of empty information and the dimension is as high as 251.



Therefore, some linear programming, such as some tree algorithms, which are good at dealing with low latitudes, are not suitable for this problem. The high-dimensional sparse matrix will make the training efficiency of the tree model extremely inefficient and easy to overfit. This is because the tree model training process is a greedy feature selection algorithm, and a feature that maximizes the information gain after splitting is selected from the candidate feature set to split. When processing high-level sparse matrices, the number of subtrees is very large, the amount of calculation will be very large, and the training will be very slow. Based on an article on diabetes gene analysis [11], it can be seen that neural networks MLP have excellent performance in variant data analysis.

Multi-Layer Perceptron (MLP) is a supervised learning algorithm, which means that it predicts data with unknown labels based on a part of data with known labels. As its name says, it is a multi-level structure, mainly divided into input layer, hidden layer and output layer, and there can be multiple hidden layers. Therefore, the simplest MLP model is a three-layer structure as shown in the figure below.



As shown in the figure, the input layer is the 251-dimensional information of each variation, and there are different weights between the input layer and the hidden layer. As for the hidden layer, the calculation formula is as follows

Among them, X is the input vector, W is the weight, and b is the bias coefficient. f is the common sigmoid function or tan function. The relationship between the output layer and the hidden layer is also similar, except that the weight and bias coefficient have changed:

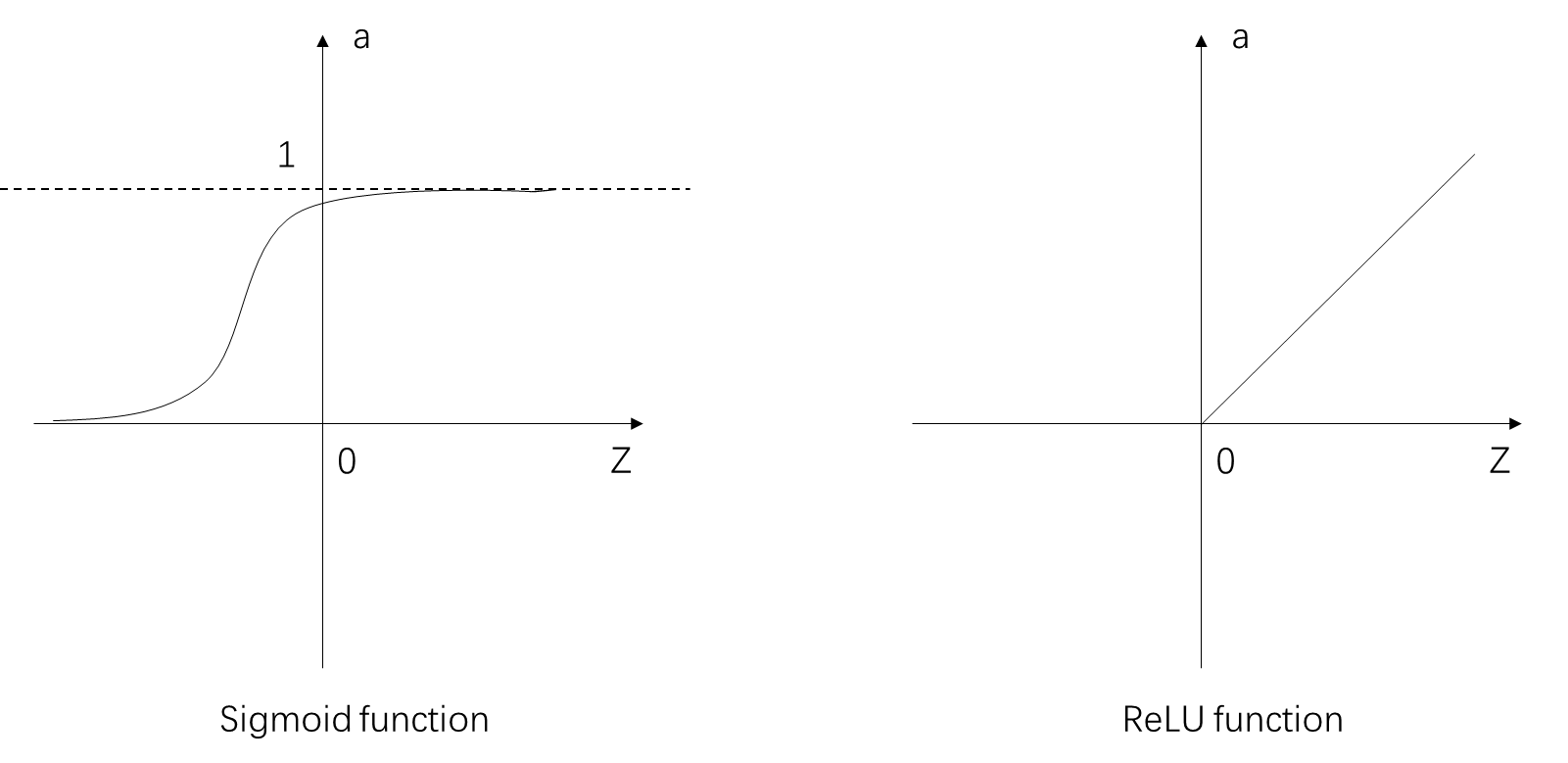
The following is an analysis of the some important parameter settings of the MLP classifier:

* Solver

In order to suggest a complete MLP model, we need to determine the weights and bias coefficients between the layers. In order to obtain the optimal parameters, that is, to minimize the cost function of training, the gradient descent method (sgd) is the most direct solution. The gradient descent method first uses random parameters, and then iteratively calculates and updates the parameters until it meets the threshold of the set number of training times or the threshold of error. According to sklearn's instructions, "adam" is an optimization algorithm based on gradient descent. And it has good performance on some relatively large training sets. So we choose “adam” in the first time experiment and also try “lbfgs” and “sgd” to compare the results.

●Activation

This parameter determines the use of the activation function. Take the sigmoid function image as an example. As shown in the figure below, the value of a remains between [0,1]. When calculating some probabilities, the sigmoid function will be more suitable for expression. However, the default value of this parameter ReLU is also shown in the figure.



When a is greater than zero, the gradient remains at 1, so the calculation speed of the neural network will be very fast. When z is less than 0, the gradient is zero, which will cause the gradient to disappear. However, in actual training, ReLU has always been the preferred parameter, and the effect of the defect of gradient disappearance is not very large.

●Alpha

This value is generally a constraint on the error, and the default value is 0.00001. In fact, the setting of this parameter is generally based on training experience, so if there are no other special requirements, the default value can be used directly.

##### 4.3.2.1 Prediction Results

Due to the strong randomness of neural network training, the results of each training are different. The following result analysis is only based on a model with “adam”,”sgd” and “lbfgs”.



【数据介绍】\*1

【task流程图】+ 预计生成的文件分析。\*1

【python编程+分析，excel表格统计】\*0.5

【扩充每个task的介绍任务以及内容】\*0.5

总结：

【遇到的问题】\*1

【改进措施】\*1

**References:**

[1] LIN Zheng-Wei, ZHAO Xiao-Hang. (2012). Application of exome sequencing in the research of human diseases. Chinese Bulletin of Life Science, Vol 24. No. 1 95-97.

[2] Flanagan, S. E., Xie, W., Caswell, R., Damhuis, A., Vianey-Saban, C., Akcay, T., ... & Ocal, G. (2013). Next-generation sequencing reveals deep intronic cryptic ABCC8 and HADH splicing founder mutations causing hyperinsulinism by pseudoexon activation. The American Journal of Human Genetics, 92(1), 131-136.

[3] Weedon, M. N., Cebola, I., Patch, A. M., Flanagan, S. E., De Franco, E., Caswell, R., ... & Houghton, J. A. (2014). Recessive mutations in a distal PTF1A enhancer cause isolated pancreatic agenesis. Nature genetics, 46(1), 61.

[4] Glaser B, Thornton P, Otonkoski T, et alGenetics of neonatal hyperinsulinismArchives of Disease in Childhood - Fetal and Neonatal Edition 2000;82:F79-F86.

[5] Lemelman, M. B., Letourneau, L., & Greeley, S. (2018). Neonatal Diabetes Mellitus: An Update on Diagnosis and Management. *Clinics in perinatology*, *45*(1), 41–59. https://doi.org/10.1016/j.clp.2017.10.006

[6] Smyth, C. M., & Bremner, W. J. (1998). Klinefelter syndrome. *Archives of Internal Medicine*, *158*(12), 1309-1314.

[7] Osbak, K. K., Colclough, K., Saint‐Martin, C., Beer, N. L., Bellanné‐Chantelot, C., Ellard, S., & Gloyn, A. L. (2009). Update on mutations in glucokinase (GCK), which cause maturity‐onset diabetes of the young, permanent neonatal diabetes, and hyperinsulinemic hypoglycemia. *Human mutation*, *30*(11), 1512-1526.

[8] Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2010). GenBank. *Nucleic acids research*, *38*(Database issue), D46–D51. https://doi.org/10.1093/nar/gkp1024

[9] Mathilde, L. , Stephanie, R. , Emmanuelle, C. , Isabelle, S. , Yanis, M. , & Migraine, B. A. , et al. (2018). Association of maternal nutrition with transient neonatal hyperinsulinism. Plos One, 13(5), e0195383.

[10] Piovesan, A., Caracausi, M., Antonaros, F., Pelleri, M. C., & Vitale, L. (2016). GeneBase 1.1: a tool to summarize data from NCBI gene datasets and its application to an update of human gene statistics. *Database : the journal of biological databases and curation*, *2016*, baw153. <https://doi.org/10.1093/database/baw153>

[11] Pociot F, Karlsen AE, Pedersen CB, et al. Novel analytical methods applied to type 1 diabetes genome-scan data. American Journal of Human Genetics. 2004 Apr;74(4):647-660. DOI: 10.1086/383095.