Title: SIFT Missense Predictions for Genomes and 1000 Genomes Data

Course: MED263, "Bioinformatics Applications To Human Disease"

Preparer: Nathaniel Delos Santos

1) Introduction

In this practical, you are going to use linux command line tools, the SIFT4G Variant annotator, and data from the 1000 Genomes Project to predict deleterious missense mutations from human samples. Predicting variant deleteriousness is an important part of analyzing human genome variants in disease, because it provides insight into which genes have been affected by a variant, and how bad the effect might be.
After this tutorial, you should be able to:

- Download variant information from the Ensembl project
- Download aligned sequence data from the 1000 Genomes project
- Call variants from aligned sequence data
- Annotate variants from aligned sequence data with deleteriousness and amino acid change predictions
- Prioritize variants by a deleteriousness score
- Perform data manipulation with basic command line tools in BASH

1.1) Download SIFT 4G

SIFT 4G, (Sorting Intolerant From Tolerant, For Genomes) uses variant calls to predict what amino acid substitutions occur, and how deleterious they are. SIFT 4G requires Java and a reference database to run.

We will download SIFT 4G directly from their website at (http://sift.bii.a-star.edu.sg/sift4g/ (http://sift.bii.a-star.edu.sg/sift4g/)) using wget. Make sure wget is installed on your system.

In [1]:

```
%%bash
wget -q http://sift.bii.a-star.edu.sg/sift4g/SIFT4G_Annotator_v2.4.ja
r
```

1.2) Download Homo Sapiens Database (GRCh38.78) for SIFTG

We must download the reference database for GRCh38, the newest version of the human genome reference available from Ensembl. We will download SIFT 4G's version of this database directly from their website. Make sure to choose GRCh38.78. Decompress the

```
In [2]: %%bash
        wget -q http://sift.bii.a-star.edu.sg/sift4g/public/Homo_sapiens/GRCh
        38.78.zip -O GRCh38.78.zip
```

```
In [ ]: %%bash
        unzip GRCh38.78.zip
```

1.3) SAMTools

SAMTools is a general toolkit for use with aligned sequencing data. We will use it here to call variants from sequence alignments, using the 'samtools mpileup' command. We will install version 1.4 here, since the specific version matters for our purposes. Make sure that GCC and your build environment are up to date.

```
In [4]: %%bash
        wget -q https://github.com/samtools/samtools/releases/download/1.4/sa
        mtools-1.4.tar.bz2 -O samtools-1.4.tar.bz2
```

```
In [ ]: %%bash
        tar -vxjf samtools-1.4.tar.bz2
        cd samtools-1.4
        ./configure
        make
        cd ..
```

1.4) BCFTools

BCFTools is a general toolkit for use with variant call format (VCF) files. We will use it here to filter and query variants. We install version 1.4 here as we did for SAMTools

```
In [6]: %%bash
        wget -q https://github.com/samtools/bcftools/releases/download/1.4/bc
        ftools-1.4.tar.bz2 -O bcftools-1.4.tar.bz2
```

In [ ]:

```
%%bash
tar -vxjf bcftools-1.4.tar.bz2
cd bcftools-1.4
./configure
make
cd ..
```

2) Data

2.1) Craig Venter Germline Variations

Craig Venter's genome was among the first sequenced. These Variant Call Format (VCF) files summarize the variants observed in his genome from the GRCh38.78 reference.

In [8]:

```
%%bash
wget -q http://ftp.ensembl.org/pub/release-78/variation/vcf/homo_sapiens/Venter.vcf.gz -O Venter.vcf.gz
wget -q http://ftp.ensembl.org/pub/release-78/variation/vcf/homo_sapiens/Venter.vcf.gz.tbi -O Venter.vcf.gz.tbi
```

Question 1)

How many variants are in the Venter VCF?

Answer 1)

In [9]:

```
%%bash
zcat Venter.vcf.gz|grep -v '#'|wc -l
```

```
3266109
```

3266109 Variants

2.2) James Watson Germline Variations

James Watson is famous for discovering the double helix structure of DNA with Francis Crick. He has his own tribute in VCF format here.

In [10]:

```
%%bash
wget -q http://ftp.ensembl.org/pub/release-78/variation/vcf/homo_sapiens/Watson.vcf.gz -O Watson.vcf.gz
wget -q http://ftp.ensembl.org/pub/release-78/variation/vcf/homo_sapiens/Watson.vcf.gz.tbi -O Watson.vcf.gz.tbi
```

2.3) 1000 Genomes human sample exome data

The 1000 Genomes project was an international effort to catalog most variants with more than 1% frequency in the human population. It is a valuable source of human sequencing data. We will not be using the VCFs directly, but instead will be analyzing aligned sequences from a single human sample.

2.3.1) CRAM files

CRAM files are compressed sequence alignment files that use delta compression from a reference to store sequence information, rather than containining the sequence data themselves. Therefore, we must download the CRAM file, CRAM index, and the corresponding reference files to use them.

In [11]:

```
%%bash
wget -q ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/data/CEU/NA06984/exome_alignment/NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.cram -O NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.cram
wget -q ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/data/CEU/NA06984/exome_alignment/NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.cram.crai -O NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.cram.crai
```

The reference files for the CRAM file are downloaded below

In [12]:

```
%%bash
wget -q ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/GRCh38_reference_genome/GRCh38_full_analysis_set_plus_decoy_hla.fa -O GRCh38_full_analysis_set_plus_decoy_hla.fa
wget -q ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/GRCh38_reference_genome/GRCh38_full_analysis_set_plus_decoy_hla.fa.fai -O GRCh38_full_analysis_set_plus_decoy_hla.fa.fai
```

Question 2)

From the README provided by the 1000 Genomes Project (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/README.1000genomes.GRCh3
, what steps have already been performed for these CRAM files to make them ready for analysis?

Answer 2)

1. Read alignment
2. Local realignment around Indels
3. Recalibration of base quality scores
4. Marking of duplicate reads
5. Merging multiple sequencing libraries into a single sample alignment file
6. Lossless compression using CRAM

3) Methods/Results

We will now run SIFT4G to predict the deleteriousness of variants found in the Venter VCF.

3.1) Analysis of Craig Venter germline variants

First we must decompress the gzipped VCF to an uncompressed VCF using zcat.

SIFT4G is run using java, so we must call it using 'java -jar', passing the SIFT4G program as the '-jar' option. The '-c' option will run SIFT4G in command line mode, and the '-t' option will cause SIFT4G to output additional annotations for each transcript of a gene affected. The '-i' option specifies the input VCF, in this case 'Venter.vcf'. The '-d' option specifies the database we will be using, in this case the GRCh38.78 database. The '-r' option will determine where the results of the SIFT annotation will be located relative to our current directory.

In [13]:

```
%%bash
zcat Venter.vcf.gz > Venter.vcf
java -jar SIFT4G_Annotator_v2.4.jar -c -t -i Venter.vcf -d GRCh38.78
 -r Venter.SIFT4G
```

```
Start Time for SIFT4G code: Mon Mar 27 01:34:17 PDT 2017
Updates:
No updates from server!! Please go to http:sift-dna.org for updates.

Started Running .......
Running in Multitranscripts mode

Chromosome      WithSIFT4GAnnotations   WithoutSIFT4GAnnotations
Progress
MT                      1                       0
Completed : 1/25
Y                       130                     18578
Completed : 2/25
22                      2914                    22450
Completed : 3/25
20                      5175                    66630
Completed : 4/25
13                      9175                    116297
Completed : 5/25
21                      7318                    59106
Completed : 6/25
X                       3941                    79652
Completed : 7/25
10                      10038                   163272
Completed : 8/25
9                       12656                   155721
Completed : 9/25
19                      9448                    57848
Completed : 10/25
18                      11794                   79902
Completed : 11/25
7                       10687                   138041
Completed : 12/25
17                      8831                    79089
Completed : 13/25
16                      13997                   99378
Completed : 14/25
14                      12880                   86113
Completed : 15/25
6                       9155                    147054
Completed : 16/25
11                      15317                   153409
Completed : 17/25
15                      16910                   83760
Completed : 18/25
12                      17570                   132459
Completed : 19/25
8                       18309                   155222
Completed : 20/25
4                       24268                   224475
Completed : 21/25
5                       18220                   144711
Completed : 22/25
3                       20600                   205453
Completed : 23/25
1                       23661                   239132
Completed : 24/25
```

```
2                       30376                   244986
Completed : 25/25

Merging temp files....
SIFT4G Annotation completed !
Output directory:Venter.SIFT4G
End Time for parallel code: Mon Mar 27 01:43:35 PDT 2017
```

Question 3)

On Chromosome 17, how many variants are annotated? How many are unnannotated?

Answer 3)

8831 annotated, 79089 unnannotated

3.1.1 SIFT 4G Output

The output of SIFT 4G includes a VCF file and an excel (.xls) file that describe the amino acid changes and the predicted deleteriousness of each variant. The excel file is formatted similarly to a tab-separated values file, with the exception of a carriage return ('\r') before each new line. We will use this to navigate the SIFT 4G output.

Question 4)

How many columns Does the SIFT4G output contain? What does each column contain?

Answer 4)

In [14]:

```
%%bash
cat Venter.SIFT4G/Venter_SIFTannotations.xls|head -n1|tr '\t' '\n'|ca
t -n
```

```
     1	CHROM
     2	POS
     3	REF_ALLELE
     4	ALT_ALLELE
     5	TRANSCRIPT_ID
     6	GENE_ID
     7	GENE_NAME
     8	REGION
     9	VARIANT_TYPE
    10	REF_AMINO
    11	ALT_AMINO
    12	AMINO_POS
    13	SIFT_SCORE
    14	SIFT_MEDIAN
    15	NUM_SEQS
    16	dbSNP
    17	SIFT_PREDICTION
```

17 Columns, contents are described above.

Question 5)

How many deleterious (not 'Low confidence') variants are found from these variants?

Answer 5)

In [15]:

```
%%bash
cat Venter.SIFT4G/Venter_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence'|cut -f1,2,3,4 \
|sort|uniq|wc -l
```

```
1561
```

1561 deleterious variants.

Question 6)

How many genes have deleterious variants? Output the list of genes names into a file. Display the first 10 gene names.

Answer 6)

In [1]:

```
%%bash
cat Venter.SIFT4G/Venter_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence'|cut -f7 \
|sort|uniq \
> Venter.SIFT4G.genes_with_deleterious_variants.txt
wc -l Venter.SIFT4G.genes_with_deleterious_variants.txt
head -n10 Venter.SIFT4G.genes_with_deleterious_variants.txt
```

```
1186 Venter.SIFT4G.genes_with_deleterious_variants.txt
A2ML1
ABCA10
ABCA6
ABCA7
ABCC8
ABCD1
AC008686.1
AC073657.1
ACACB
ACADS
```

1186 genes. Gene names listed above.

3.1.2) SIFT Scores

SIFT scores less than 0.05 are considered deleterious. Anything greater is considered tolerated. Lower SIFT scores are considered more deleterious.

Question 7)

What is the lowest SIFT score of the deleterious variants?

Answer 7)

In [17]:
```
%%bash
cat Venter.SIFT4G/Venter_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence' \
|cut -f1,2,3,4,13 \
|sort|uniq \
|sort -k1,1 -k2,2n \
|sort -k5,5n \
|head
```

```
10      122336645       A       G       0.000
10      125980182       C       T       0.000
10      128113592       C       G       0.000
10      26219214        C       A       0.000
10      46461688        A       C       0.000
10      46549695        C       G       0.000
10      46549695        C       T       0.000
10      48086   G       A       0.000
10      59792934        G       T       0.000
10      6224537 G       T       0.000
```

0.0 is the lowest SIFT score.

Question 8)

What variants are annotated with the lowest SIFT score? Output the chromosome, coordinate, reference base, alternate base, gene name, reference amino acid, alternate amino acid, amino acid position, and sift score into a file. Display the first 10 lines of this file.

Answer 8)

In [18]:

```
%%bash
cat Venter.SIFT4G/Venter_SIFTannotations.xls|cut -f1,2,3,4,7,10,11,1
2,13,17 \
|grep '^CHROM\|DELETERIOUS'|grep -v 'Low confidence' \
|awk '($9==0.0)||$1=="CHROM"' \
> Venter.SIFT4G.sift_score_0.txt
head -n10 Venter.SIFT4G.sift_score_0.txt
```

```
CHROM   POS     REF_ALLELE      ALT_ALLELE      GENE_NAME       REF_A
MINO    ALT_AMINO       AMINO_POS       SIFT_SCORE      SIFT_PREDICTI
ON
1       1956754 C       A       CFAP74  G       C       628     0.000
DELETERIOUS
1       3497541 C       T       MEGF6   G       R       1152    0.000
DELETERIOUS
1       11789390        A       G       C1orf167        R       G
810     0.000   DELETERIOUS
1       17334004        G       C       PADI4   G       A       112
0.000   DELETERIOUS
1       25321889        G       C       RHD     G       A       385
0.000   DELETERIOUS
1       54670856        T       C       MROH7-TTC4      V       A
534     0.000   DELETERIOUS
1       54801124        G       C       TTC22   L       V       14
0.000   DELETERIOUS
1       54801124        G       C       TTC22   L       V       14
0.000   DELETERIOUS
1       120889909       T       G       PPIAL4B L       R       30
0.000   DELETERIOUS
```

3.2) Analysis of James Watson germline variants

In [19]:

```
%%bash
zcat Watson.vcf.gz > Watson.vcf
java -jar SIFT4G_Annotator_v2.4.jar -c -t -i Watson.vcf -d GRCh38.78
 -r Watson.SIFT4G
```

```
Start Time for SIFT4G code: Mon Mar 27 01:43:46 PDT 2017
Updates:
No updates from server!! Please go to http:sift-dna.org for updates.

Started Running .......
Running in Multitranscripts mode

Chromosome      WithSIFT4GAnnotations   WithoutSIFT4GAnnotations
Progress
MT                      0                       1
Completed : 1/25
Y                       119                     20889
Completed : 2/25
22                      3226                    25060
Completed : 3/25
20                      5293                    69584
Completed : 4/25
13                      10529                   120358
Completed : 5/25
21                      6616                    52355
Completed : 6/25
X                       4172                    70126
Completed : 7/25
10                      10105                   166368
Completed : 8/25
9                       9591                    135019
Completed : 9/25
18                      11786                   83503
Completed : 10/25
19                      9681                    67255
Completed : 11/25
17                      8476                    82316
Completed : 12/25
7                       13094                   182383
Completed : 13/25
16                      13877                   101925
Completed : 14/25
14                      13495                   90997
Completed : 15/25
6                       11816                   193121
Completed : 16/25
11                      15692                   165100
Completed : 17/25
15                      16943                   82199
Completed : 18/25
12                      17819                   145339
Completed : 19/25
8                       18021                   156492
Completed : 20/25
4                       25496                   219125
Completed : 21/25
3                       20838                   223465
Completed : 22/25
5                       21886                   184437
Completed : 23/25
1                       21091                   243196
Completed : 24/25
```

```
2                       31034                   255084
Completed : 25/25

Merging temp files....
SIFT4G Annotation completed !
Output directory:Watson.SIFT4G
End Time for parallel code: Mon Mar 27 01:52:23 PDT 2017
```

Question 9)

On Chromosome 17, how many variants are annotated? How many are unnannotated?

Answer 9)

8476 annotated, 82316 unnannotated

Question 10)

How many deleterious (not 'Low confidence') variants are found from these variants?

Answer 10)

In [20]:

```
%%bash
cat Watson.SIFT4G/Watson_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence'|cut -f1,2,3,4 \
|sort|uniq|wc -l
```

```
1970
```

1970 deleterious variants.

Question 11)

How many genes have deleterious variants? Output the list of genes names into a file. Display the first 10 gene names.

Answer 11)

In [2]:
```
%%bash
cat Watson.SIFT4G/Watson_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence'|cut -f7 \
|sort|uniq \
> Watson.SIFT4G.genes_with_deleterious_variants.txt
wc -l Watson.SIFT4G.genes_with_deleterious_variants.txt
head -n10 Watson.SIFT4G.genes_with_deleterious_variants.txt
```

```
1528 Watson.SIFT4G.genes_with_deleterious_variants.txt
A2ML1
AADACL3
AASDHPPT
ABCA5
ABCA9
ABCB5
ABCC10
ABCC11
ABCC8
ABCC9
```

1528 genes. Gene names listed above.

Question 12)

What genes do Craig Venter and James Watson both have deleterious variants in? How many genes is this? Output the genes to a file and display the first 10.

Answer 12)

In [3]:
```
%%bash
join Venter.SIFT4G.genes_with_deleterious_variants.txt Watson.SIFT4G.
genes_with_deleterious_variants.txt \
> Venter_and_Watson.SIFT4G.genes_with_deleterious_variants.txt
wc -l Venter_and_Watson.SIFT4G.genes_with_deleterious_variants.txt
head -n10 Venter_and_Watson.SIFT4G.genes_with_deleterious_variants.tx
t
```

```
524 Venter_and_Watson.SIFT4G.genes_with_deleterious_variants.txt
A2ML1
ABCC8
AC073657.1
ACACB
ACAN
ADAMTSL3
ADH1C
AHNAK
AKAP13
AKR1C2
```

Gene names provided above. 524 genes in common.

Question 13)

What is the lowest SIFT score of the deleterious variants?

Answer 13)

```
In [23]: %%bash
         cat Watson.SIFT4G/Watson_SIFTannotations.xls|tail -n+2 \
         |grep 'DELETERIOUS'|grep -v 'Low confidence' \
         |cut -f1,2,3,4,13 \
         |sort|uniq \
         |sort -k1,1 -k2,2n \
         |sort -k5,5n \
         |head
```

```
10      113766634       T       C       0.000
10      19387657        A       G       0.000
10      26157364        C       A       0.000
10      46461688        A       C       0.000
10      48086   G       A       0.000
10      59792934        G       T       0.000
10      62376867        C       T       0.000
10      86936837        C       G       0.000
10      89307233        A       T       0.000
10      95339252        C       A       0.000
```

0.0 is the lowest SIFT score.

Question 14)

What variants are annotated with the lowest SIFT score? Output the chromosome, coordinate, reference base, alternate base, gene name, reference amino acid, alternate amino acid, amino acid position, and sift score into a file. Display the first 10 lines of this file.

Answer 14)

In [24]:

```
%%bash
cat Watson.SIFT4G/Watson_SIFTannotations.xls|cut -f1,2,3,4,7,10,11,1
2,13,17 \
|grep '^CHROM\|DELETERIOUS'|grep -v 'Low confidence' \
|awk '($9==0.0)||$1=="CHROM"' \
> Watson.SIFT4G.sift_score_0.txt
head -n10 Watson.SIFT4G.sift_score_0.txt
```

```
CHROM   POS     REF_ALLELE      ALT_ALLELE      GENE_NAME       REF_A
MINO    ALT_AMINO       AMINO_POS       SIFT_SCORE      SIFT_PREDICTI
ON
1       1956754 C       A       CFAP74  G       C       628     0.000
DELETERIOUS
1       3497541 C       T       MEGF6   G       R       1152    0.000
DELETERIOUS
1       11789390        A       G       C1orf167        R       G
810     0.000   DELETERIOUS
1       12725782        C       T       AADACL3 P       L       280
0.000   DELETERIOUS
1       17334004        G       C       PADI4   G       A       112
0.000   DELETERIOUS
1       26367769        T       C       ZNF683  D       G       48
0.000   DELETERIOUS
1       26367769        T       C       ZNF683  D       G       48
0.000   DELETERIOUS
1       26367769        T       C       ZNF683  D       G       48
0.000   DELETERIOUS
1       28490968        C       T       PHACTR4 R       C       622
0.000   DELETERIOUS
```

3.3) Analysis of 1000 Genomes Sample Human Data

3.3.1) Calling variants from aligned sequencing data

The 1000 Genomes exome sequencing data for this sample is not yet in VCF format. We must use samtools mpileup and bcftools call to convert it.

For samtools mpileup, we use the following options:

- '-u' generate uncompressed VCF/BCF output. This saves time on compression and decompression, since we pipe to bcftools.
- '-g' generate output in BCF format. This is a more compact binary format, ideal for transferring between programs.
- '-f' the FASTA file used as reference for the CRAM file. Required to determine if something varies from the reference, and to decompress the CRAM data.

For bcftools call, we use the following options to call variants:

- '-f GQ,GP' output genotype quality and genotype probability. We care about GQ for filtering.
- '-v' output variant sites only. We don't care about sites that match the reference.
- '-m' we use the multiallelic caller, upon recommendation by the samtools website.
- '-O v' output VCF formatted file.
- '-o' output variants to the specified file

We connect the output of samtools mpileup to the input of bcftools using a pipe '|'.

In [25]:

```
%%bash
date
samtools-1.4/samtools mpileup \
-ugf GRCh38_full_analysis_set_plus_decoy_hla.fa \
NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.cram \
| bcftools-1.4/bcftools call \
-f GQ,GP \
-vmO v \
-o NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.gq.gp.vcf
date
```

```
Sun Mar 26 01:52:24 PDT 2017
Sun Mar 26 04:28:51 PDT 2017
```

```
Note: none of --samples-file, --ploidy or --ploidy-file given, assumi
ng all sites are diploid
[mpileup] 1 samples in 1 input files
<mpileup> Set max per-file depth to 8000
```

3.3.2) Filtering variants by read depth, quality, and genotype quality

Not all variant calls are made equal. We want to avoid predicting the deleteriousness of variants that may not be real. So we use filtering to filter for the depth of sequencing at each variant coordinate, and the confidence the variant caller has in the variant. This is encapsulated in the DP, QUAL, and GQ fields.

The command bcftools filter is used to implement these filters.

- '-i' specifies an expression for variants to include.
- 'INFO/DP>10': We want raw read depth to be greater than 10
- 'QUAL>20': We want the quality of any variant called here to be greater than 20
- 'FMT/GQ>20': We want the genotype to be called with a confidence greater than 20.

We then combine these criteria using logical AND ('&&') to yield the final filter inclusion statement, '(QUAL>20)&&(INFO/DP>10)&&(FMT/GQ>20)'.

For more details on DP, QUAL, and GQ, see the guide from GATK (http://gatkforums.broadinstitute.org/gatk/discussion/1268/what-is-a-vcf-and-how-should-i-interpret-it (http://gatkforums.broadinstitute.org/gatk/discussion/1268/what-is-a-vcf-and-how-should-i-interpret-it)).

In [26]:

```
%%bash
date
bcftools-1.4/bcftools filter -i '(QUAL>20)&&(INFO/DP>10)&&(FMT/GQ>2
0)' \
-O v \
-o NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_1
0.gq_gt_20.vcf \
NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.gq.gp.vcf
date
```

```
Sun Mar 26 04:28:51 PDT 2017
Sun Mar 26 04:28:56 PDT 2017
```

Question 15)

How many variants are in the VCF before filtering? How many after filtering?

Answer 15)

In [27]:

```
%%bash
cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.gq.gp.vcf|grep -v
'^#'|wc -l
cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_1
0.gq_gt_20.vcf|grep -v '^#'|wc -l
```

```
2254572
93617
```

2254572 variants before filtering. 93617 variants after filtering.

3.3.3) Annotating variants with SIFT4G

In [26]:
```
%%bash
java -jar SIFT4G_Annotator_v2.4.jar -c -t \
-i NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.gq_gt_20.vcf \
-d GRCh38.78  \
-r NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.gq_gt_20.SIFT4G \
> NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.gq_gt_20.SIFT4G.log
```

Question 16)

On Chromosome 17, how many variants are annotated? How many are unnannotated?

Answer 16)

1571 annotated, 2169 unnannotated

In [27]:
```
%%bash
cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.gq_gt_20.SIFT4G.log|grep -w '^17'
```

```
17                        1571                      2169
Completed : 1034/1047
```

Question 17)

How many deleterious (not 'Low confidence') variants are found from these variants?

Answer 17)

In [28]:

```
%%bash
cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_1
0.gq_gt_20.SIFT4G\
/NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.g
q_gt_20_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence'|cut -f1,2,3,4 \
|sort|uniq|wc -l
```

```
1365
```

1365 deleterious variants.

Question 18)

How many genes have deleterious variants? Output the list of genes names into a file. Display the first 10 gene names.

Answer 18)

In [29]:

```
%%bash
cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_1
0.gq_gt_20.SIFT4G\
/NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.g
q_gt_20_SIFTannotations.xls|tail -n+2 \
|grep 'DELETERIOUS'|grep -v 'Low confidence'|cut -f7 \
|sort|uniq \
> 1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
wc -l 1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
head -n10 1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
```

```
1154 1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
A2ML1
ABCA12
ABCA4
ABCA9
ABCC11
ABCC4
ABCD4
ABHD11
ABO
AC244230.1
```

1154 genes. Gene names listed above.

Question 19)

What genes do Craig Venter, James Watson, and this 1000 Genomes sample All have deleterious variants in? How many genes is this? Output the genes to a file and display the first 10.

Answer 19)

In [30]:

```
%%bash
join Venter_and_Watson.SIFT4G.genes_with_deleterious_variants.txt \
1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt \
> Venter_and_Watson_and_1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
wc -l Venter_and_Watson_and_1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
head -n10 Venter_and_Watson_and_1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
```

```
322 Venter_and_Watson_and_1KGenomesSample.SIFT4G.genes_with_deleterious_variants.txt
A2ML1
ACACB
ACAN
ADAMTSL3
AHNAK
AKAP13
AKR1C2
ALDH1B1
ALPK2
ALPK3
```

Gene names provided above. 322 genes in common.

Question 20)

What is the lowest SIFT score of the deleterious variants?

Answer 20)

```
In [31]: %%bash
         cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_1
         0.gq_gt_20.SIFT4G\
         /NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.g
         q_gt_20_SIFTannotations.xls|tail -n+2 \
         |grep 'DELETERIOUS'|grep -v 'Low confidence' \
         |cut -f1,2,3,4,13 \
         |sort|uniq \
         |sort -k1,1 -k2,2n \
         |sort -k5,5n \
         |head
```

```
chr10   100506090       A       C       0.000
chr10   11755501        G       A       0.000
chr10   26219214        C       A       0.000
chr10   46549695        C       A       0.000
chr10   48086   G       A       0.000
chr10   73378933        C       T       0.000
chr10   97465888        G       A       0.000
chr1    11046609        T       C       0.000
chr11   108593482       T       C       0.000
chr11   26508237        C       T       0.000
```

0.0 is the lowest SIFT score.

Question 21)

What variants are annotated with the lowest SIFT score? Output the chromosome, coordinate, reference base, alternate base, gene name, reference amino acid, alternate amino acid, amino acid position, and sift score into a file. Display the first 10 lines of this file.

Answer 21)

In [32]:

```
%%bash
cat NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_1
0.gq_gt_20.SIFT4G\
/NA06984.alt_bwamem_GRCh38DH.20150826.CEU.exome.qual_gt_20.dp_gt_10.g
q_gt_20_SIFTannotations.xls \
|cut -f1,2,3,4,7,10,11,12,13,17 \
|grep '^CHROM\|DELETERIOUS'|grep -v 'Low confidence' \
|awk '($9==0.0)||$1=="CHROM"' \
> 1KGenomesSample.SIFT4G.sift_score_0.txt
head -n10 1KGenomesSample.SIFT4G.sift_score_0.txt
```

```
CHROM   POS     REF_ALLELE      ALT_ALLELE      GENE_NAME       REF_A
MINO    ALT_AMINO       AMINO_POS       SIFT_SCORE      SIFT_PREDICTI
ON
chr1    1956754 C       A       CFAP74  G       C       628     0.000
DELETERIOUS
chr1    11046609        T       C       MASP2   D       G       120
0.000   DELETERIOUS
chr1    17334004        G       C       PADI4   G       A       112
0.000   DELETERIOUS
chr1    18483281        T       C       KLHDC7A L       S       767
0.000   DELETERIOUS
chr1    25342976        T       G       TMEM50A W       G       37
0.000   DELETERIOUS
chr1    26043403        G       T       SLC30A2 N       K       189
0.000   DELETERIOUS
chr1    26043403        G       T       SLC30A2 N       K       140
0.000   DELETERIOUS
chr1    54653861        C       T       MROH7   S       F       312
0.000   DELETERIOUS
chr1    54653861        C       T       MROH7   S       F       312
0.000   DELETERIOUS
```

4) References

1. Vaser R, Adusumalli S, Leng SN, Sikic M, Ng PC. SIFT missense predictions for genomes. Nat Protoc. 2016;11(1):1-9. (https://www.ncbi.nlm.nih.gov/pubmed/26633127 (https://www.ncbi.nlm.nih.gov/pubmed/26633127))
2. Aken BL, Achuthan P, Akanni W, et al. Ensembl 2017. Nucleic Acids Res. 2017;45(D1):D635-D642. (https://www.ncbi.nlm.nih.gov/pubmed/27899575 (https://www.ncbi.nlm.nih.gov/pubmed/27899575))
3. Auton A, Brooks LD, Durbin RM, et al. A global reference for human genetic variation. Nature. 2015;526(7571):68-74. (https://www.ncbi.nlm.nih.gov/pubmed/26432245 (https://www.ncbi.nlm.nih.gov/pubmed/26432245))
4. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078-9. (https://www.ncbi.nlm.nih.gov/pubmed/19505943 (https://www.ncbi.nlm.nih.gov/pubmed/19505943))
5. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011;27(21):2987-93. (https://www.ncbi.nlm.nih.gov/pubmed/21903627 (https://www.ncbi.nlm.nih.gov/pubmed/21903627))