Description:

search_dbNSFP41a.jar is the java program for querying dbNSFP v4.1a on your local machine, which supports both the command-line environment and the graphic user interface (GUI). This program was developed by Dr. Xiaoming Liu at the University of South Florida. It is available for use without charge and without warranty. This document is a simple instruction of the usage of the program.

Release:

Version 4.1a, released June 16, 2019

Files:

search_dbNSFP41a.jar	- the java program for querying database
search_dbNSFP41a.java	- the source code of the java program
LICENSE.txt	- the license for using the source code
tryhg19.in	- an example input file with hg19 genome positions
tryhg18.in	- an example input file with hg18 genome positions
tryhg38.in	- an example input file with hg38 genome positions
try.vcf	- an example of vcf input file
search_dbNSFP41a.readme.pdf	- this file

Prerequisite:

A proper Java Runtime Environment should be installed on the machine which hosts dbNSFP v4.1a. To check the availability and version of the java on the machine, on the command-line type:

```
java -version
```

If the system found the Java Runtime Environment, a version number and other information will be shown on the screen. search_dbNSFP41a is written with java version 1.8, and it should work fine with java 1.8 or upper.

Input file format:

Currently, the search program supports two formats of file: vcf format and custom format. The program automatically recognizes vcf file if its extension is ".vcf" and it will query the database by "chr pos ref alt" (see below). Custom input file contains one or more lines. Each line represents a query. A query can be a

1. A genome position: A genome position is represented by "chr pos", where chr is the chromosome number and pos is position on chromosome (default as to hg38). For example,

22 15528161

2. A non-synonymous SNP (nsSNV): A nsSNV can be represented by “chr pos ref alt”, where ref is the reference allele and alt is the alternative allele. For example,

   22 15528159 A G

   That is, the SNP on chromosome 22 at position 15528159 with reference allele A and alternative allele G. A nsSNV can also be represented more specifically by “chr pos ref alt refAA altAA”, where refAA is the reference amino acid and altAA is the alternative amino acid. For example,

   22 15528159 A C M L

   Alternatively, users may specify the nsSNV based on amino acid change referring to an Ensembl transcript id or Ensembl protein id. For example,

   Ensembl:ENST00000252835:M1K
   Ensembl:ENSP00000252835:M1T

   User can also specify a nsSNV or ssSNV using a HGVS c. presentation with Ensembl transcript id or a HGVS p. presentation with Ensembl protein id or Uniprot acc or id. For example:

   HGVSc:ENST00000335137:c.43G>C
   HGVSp:ENSP00000334393:p.E15X
   HGVSp:Q8NH21:p.Gln17*
   HGVSp:A0A2U3U0J3_HUMAN:p.Met1?

3. Users may specify a dbSNP rs number, beginning with “rs”. For example,

   rs28358582

4. A gene ID: A gene ID can be a gene name (HGNC symbol), Entrez id, Uniprot id or accession number, or Ensembl gene id or transcript id. Database name is needed for Entrez, Uniprot and Ensembl. For examples,

   MT-ND1
   Ensembl:ENSG00000198763
   Ensembl:ENST00000361624
   Ensembl:ENSP00000354876
   Uniprot:NU5M_HUMAN
   Uniprot:P00846
   Entrez:4541

Command-line usage:

1. Download dbNSFP4.1a.zip on to your machine and unzip it to a directory (for example, mydbNSFP). If you want to search dbscSNV along with dbNSFP, you need also download dbscSNV.zip and unzip it to the same directory where you put your dbNSFP database files.

2. Prepare an input file, for example, “tryhg38.in”. See above for file format. Put it into mydbNSFP.
3. Enter command-line environment, change directory to mydbNSFP. Type:

   java -jar search_dbNSFP41a.jar -i tryhg38.in -o tryhg38.out

   This command specifies the input file as “tryhg38.in” and the output file as “tryhg38.out”. If you put the input file in working directory other than the directory where dbNSFP files locate, i.e. mydbNSFP, you need to put that working directory into the “PATH” variable of your system, or replace try.in with <directory>try.in. Similarly, you can use <directory>try.out to specify the destination directory of the output file.

4. When the input file is large, Java may report memory insufficiency. In that case, try specify a larger memory for Java, for example:

   java -Xmx5g -jar search_dbNSFP41a.jar -i tryhg38.in -o tryhg38.out

Advanced command-line usage:

1. Specify the human genome reference sequence: By default, search_dbNSFP41a use human genome reference sequence version hg38 to interpret the chromosome position. To query genome positions or nsSNVs according to version hg19 (or hg18), you can specify the human genome reference sequence by using the option “-v hg19” (or “-v hg18”). For example,

   java -jar search_dbNSFP41a.jar -i tryhg19.in -o tryhg19.out -v hg19

2. Specify the chromosomes to search: By default, search_dbNSFP41a searches all chromosomes if some queries do not contain the chromosome information (for example, when querying a gene id). You can specify the chromosomes to search by using the option “-c list_of_chromosomes_to_search”, where list_of_chromosomes_to_search is a list of chromosome numbers separated by commas. For example,

   java -jar search_dbNSFP41a.jar -i tryhg19.in -o tryhg19.out -v hg19 -c 1,2,3,10,Y

3. Specify the columns to output: By default, search_dbNSFP41a outputs all columns. You can specify the columns to output by using the option “-w list_of_columns_to_write”, where list_of_columns_to_write is a list of column numbers separated by commas, and continuous number block can be simplified by begin-end. For example,

   java -jar search_dbNSFP41a.jar -i tryhg38.in -o tryhg38.out -w 1-6,8

   will output columns 1 to 6 and 8.

4. Specify whether all input columns will be preserved in the output file (for vcf input only): By default, search_dbNSFP41a will not output any columns from the vcf input file. You can choose

to output all columns of the vcf input file by using the option "-p". For example,

java -jar search_dbNSFP41a.jar -i try.vcf -o try.vcf.out –p

This option may require large memory to run for large vcf files.

5. Specify whether to search attached databases: By default, search_dbNSFP41a will not search any attached databases. You can turn on querying dbscSNV and SPIDEX by using the option "-s", and/or querying dbMTS by using the option "-m". For example,

   java -jar search_dbNSFP41a.jar -v hg38 -i tryhg38.in -o tryhg38.out -m

   Currently, this option only searches the input variants with formats of "chr pos", "chr pos ref alt" or "chr pos ref alt refAA altAA".

   Available attached databases:

   dbscSNV: dbscSNV includes all potential human SNVs within splicing consensus regions (−3 to +8 at the 5' splice site and −12 to +2 at the 3' splice site), i.e. scSNVs, related functional annotations and two ensemble prediction scores for predicting their potential of altering splicing. It is freely available at https://sites.google.com/site/jpopgen/dbNSFP. To enable search, download the zipped database and unzip all files to the same folder as the dbNSFP files. The coordinates in the input file must be in hg19 or hg38. Matching dbscSNV1.1 entries will be output to a file with the user specified output file name and an extension of ".dbscSNV".

   SPIDEX: SPIDEX free non-commercial version 1.0 can be downloaded from ANNOVAR (http://www.openbioinformatics.org/annovar/spidex_download_form.php). To enable search, first download hg19_spidex.txt, and put it in the same folder as the dbNSFP files. The coordinates in the input file must be in hg19. Matching SPIDEX entries will be output to a file with the user specified output file name and an extension of ".SPIDEX".

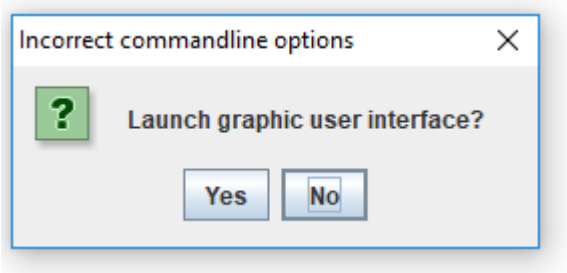
   SpliceAI: SpliceAI non-commercial version can be downloaded from https://basespace.illumina.com. After login, find the project "Predicting splicing from primary sequence". Then from the tab "FILES" click folder "genome_scores_v1.3". Download file spliceai_scores.masked.snv.hg38.vcf.gz for querying hg38 based input file and spliceai_scores.masked.snv.hg19.vcf.gz for querying hg19 based input file. Put the files in the same folder as the dbNSFP files. Matching SpliceAI entries will be output to a file with the user specified output file name and an extension of ".SpliceAI".

   dbMTS: dbMTS collects all potential SNVs microRNA target seed regions in human 3'UTRs and provides their functional predictions and annotations to facilitate the steps of filtering and prioritizing SNVs from a huge list of all SNVs discovered in a whole exome sequencing (WES) study. It is freely available at https://sites.google.com/site/jpopgen/dbNSFP. To enable search, download the zipped database and unzip all files to the same folder as the dbNSFP files. The coordinates in the input file must be in hg19 or hg38. Matching dbMTS entries will be output to a file with the user specified output file name and an extension of ".dbMTS".

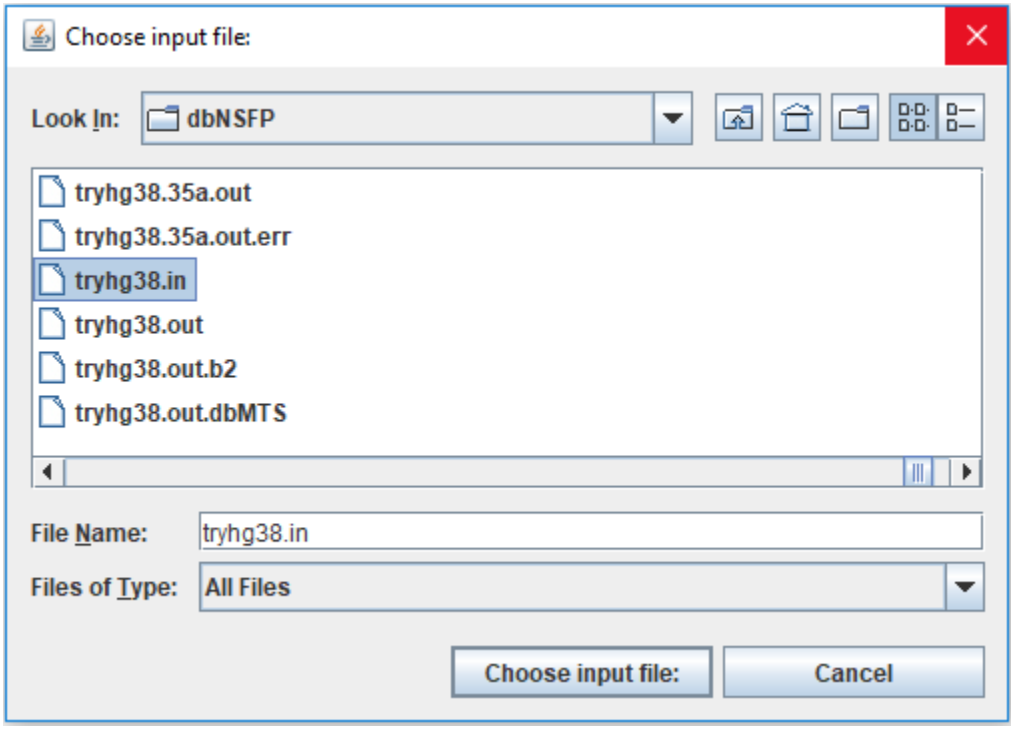
6. Specify whether to search complete gene annotation: By default, search_dbNSFP41a will search dbNSFP4.0_gene.gz. With option "-g", search_dbNSFP41a will search dbNSFP4.0_gene.complete.gz, which contains complete gene interaction annotations.

Graphic user interface (GUI) usage:

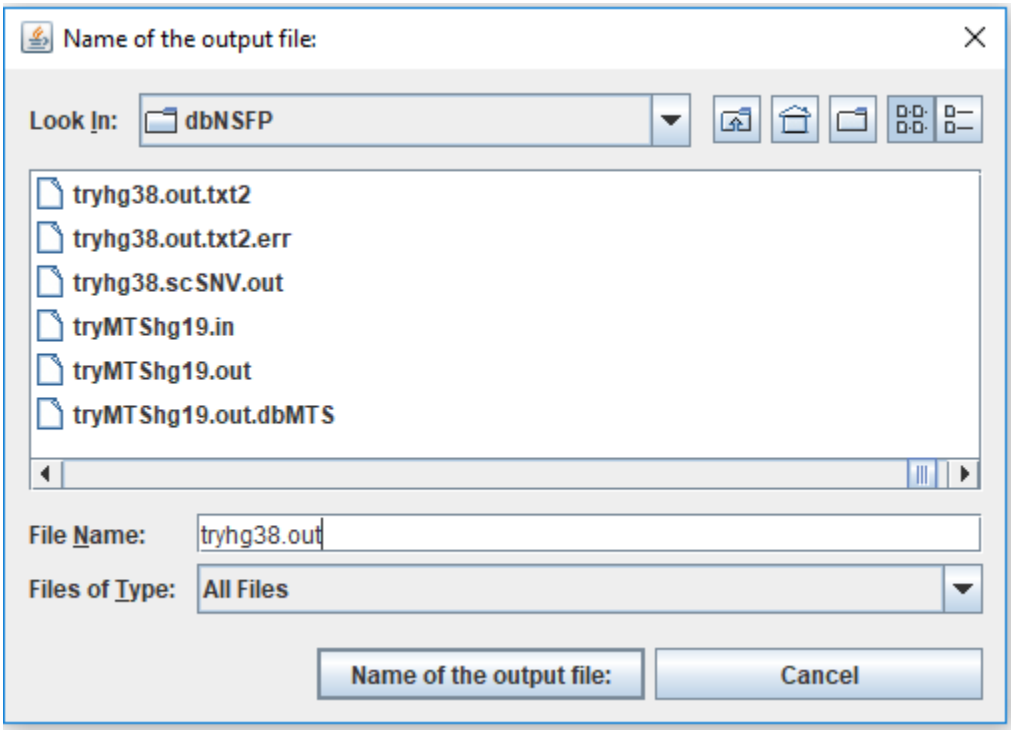
1. Double-click search_dbNSFP41a.jar. A dialog will pop-up and asks whether to launch GUI. Click "Yes".



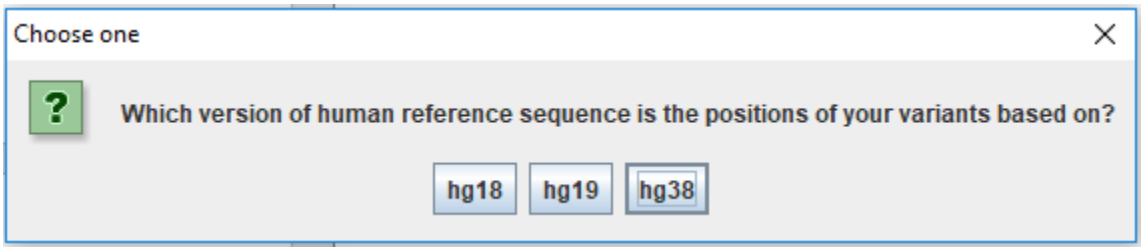
2. A file-chooser window similar to the figure below will appear and ask for choosing the input file. Find your input file and click "Choose input file:".



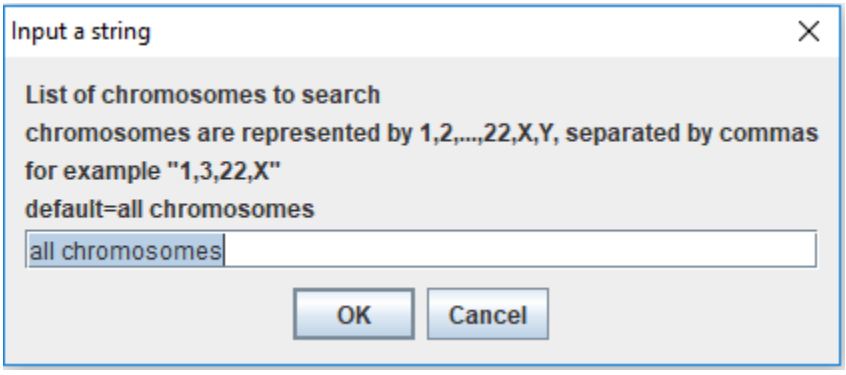
3. Another file-chooser window similar to the figure below will appear and ask for the name of the output file. Type in the output file name and click “Name of the output file:”.



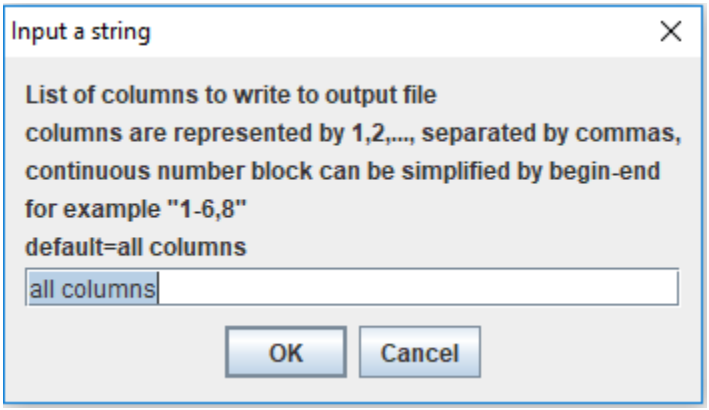
4. A dialog window appears and asks for the version of human reference sequence, based on which the coordinates of the variants in the input file are based on. Click the button corresponding to the correct version.



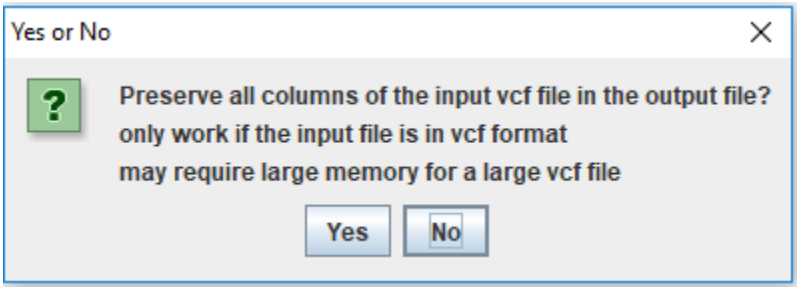
5. A string input window appears for the value of the “-c” option, i.e. the chromosomes to search. The default is “all chromosomes”. Accept the default choice or change the value then click “OK”.



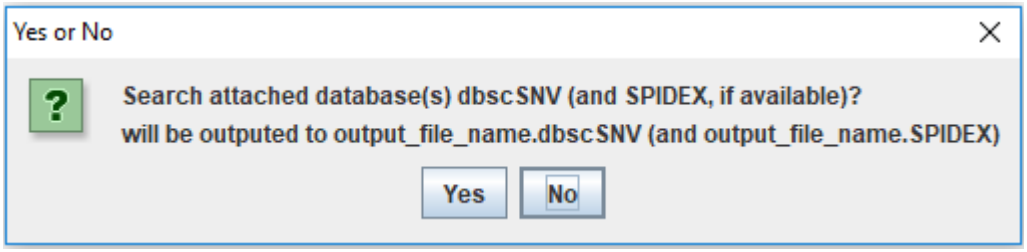
6. Another string input window appears for the value of the "-w" option, i.e. the columns to output. The default is "all columns". Accept the default choice or change the value then click "OK".



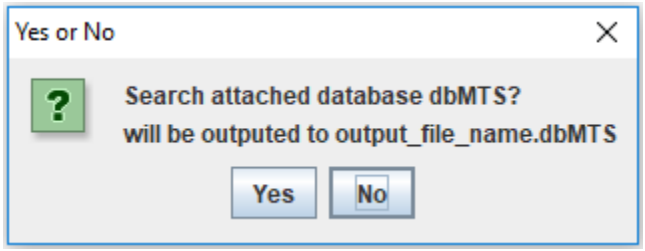
7. A "Yes or No" dialog appears and asks whether to preserve all the columns in the input vcf file, i.e. the "-p" option. This option is only valid if the input file is in vcf format. Click "Yes" or "No".



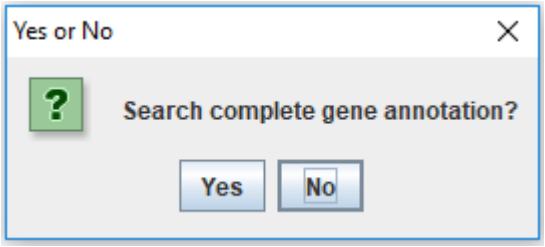
8. Another "Yes or No" dialog appears and asks whether to search the attached databases dbscSNV and SPIDEX, i.e. the "-s" option. Click "Yes" or "No".



9. Another "Yes or No" dialog appears and asks whether to search the attached database dbMTS, i.e. the "-m" option. Click "Yes" or "No".



10. The final “Yes or No” dialog appears and asks whether to search the complete gene annotation, i.e. the “-g” option. Click “Yes” or “No”.



11. The output will be shown in a text window. Close the window when the search is finished.

Output:

The output file contains all nsSNVs that match the query. By default that includes all columns of dbNSFP4.1a_variant and most columns of dbNSFP4.0_gene (except the first three columns). User can specify the columns to output (see above). All queries that do not have a match in dbNSFP4.1a_variant will be written to an “.err” file. If you search dbscSNV, SPIDEX, SpliceAI and dbMTS along with dbNSFP, separate output files will be produced containing all corresponding SNVs that match the query.

Contact:

Xiaoming Liu, Ph.D.

Associate Professor,
USF Genomics,
College of Public Health,
University of South Florida

Email: xmliu.uth@gmail.com