snptools Manual

Daniel Ortiz Martínez

Contents

1 Introduction

A single nucleotide polymorphism (SNP) is a variation in a DNA sequence affecting only one nucleobase: adenine (A), thymine (T), cytosine (C) o guanine (G), although some authors consider that changes affecting a few nucleobases as well as isolated insertions and deletions should also be identified as SNPs.

Studying SNPs may serve different purposes. One example would be the generation of genetic footprints, which allow us to distinguish between two individuals of the same species and is useful for instance in forensic science. However, the most interesting application from the perspective of this work is the relationship between diseases and genetic variation, including the susceptibility of individuals to such diseases, their severity as well as the response to medical treatments.

The rest of this document is organized as follows, first we will introduce the main features of the SNPedia database, devoted to SNPs studies (Section 2). After that, we will describe the *snptools* package we created (Section 3). The package makes use of SNPedia to automatically generate reports about SNPs of interest in different file formats. Finally, to conclude this document, we provide one example of such reports (Appendix A).

2 The SNPedia Database

The SNPedia database (Cariaso and Lennon 2012) is a collaborative web storing information about SNPs. Each database entry provides diverse information about an individual SNP, including a brief description, links to scientific articles or web pages, as well as microarray data information.

SNPedia uses the so-called *Semantic MediaWiki* open source software, which allows to annotate semantic data contained in wiki pages. SNPedia is edited and updated both in an automatic or manual fashion, and its structure has been designed so as to enable automatic report generation in an easy manner, ensuring the highest possible readability for the user. SNPedia does not embrace the so-called "No original research"[1] policy used in Wikipedia, inviting users to contribute with well documented original research in form of genotype reports.

2.1 Stored Information

SNPedia stores SNPs using a specific notation composed of the *rs* letters and a numeric identifier that has been adopted in the NCBI's dbSNP database (Wheeler et al. 2004). While dbSNP is not focused on any particular species, SNPedia is devoted on the human one. The basic unit of information handled in SNPedia are those specific nucleotide positions that are known to vary in germ and somatic cells. SNPedia not only contains entries related to individual SNPs, but also genotype-level information, incorporating studies with a large number of patients, producing studies with a high statistical power.

SNPedia also analyzes the phenotype observed as a result of the combination of a set of SNPs. For this purpose, a specific notation able to describe them has been developed.

SNPedia contributions are authored by individual users, or retrieved from public databases. Typically, data sources mention publications, incorporating certain identifiers such as Pubmed PMID or DOI that are also included in SNPedia.

[1] http://en.wikipedia.org/wiki/Wikipedia:No_original_research

SNPedia has been online in Internet from 2006. During these years the information it contains has grown from around 1000 SNPs that were stored initially to more than 70 000 that are currently available. Newly added data are verified by the database users in a regular basis. This manual verification is complemented with various levels of automatic checks, such as those provided by the *Semantic MediaWiki* templates, or certain *software bots* that regularly incorporate supplementary information (chromosome, position, allelic data, etc.) to the different database entries.

2.2 Related Webs and Databases

The vast majority of the SNPedia database content is generated automatically from software accessing Pubmed[2] daily in an automated manner, looking for articles containing the string *rs#* in their abstract (which is related to the standard SNP notation mentioned above). Other related webs would be Genome.gov, OMIM[3] o PharmGKB[4].

Additionally, SNPedia database entries also contain links to other websites such as those mentioned above, dbSNP o PharmGKB. Besides those public sources, SNPedia also incorporates links to private ones, as for example NextBio[5] or 23andMe[6].

2.3 How to Access SNPedia

There exist three different ways to access SNPedia content that are briefly described in the following sections, namely, accessing through the SNPedia web site, accessing through the SNPedia API or downloading a file in `gff` format.

2.3.1 From SNPedia Site

It is possible to access to SNPs information by means of a web page[7] that is part of SNPedia. Such a page contains a list of links to the different database entries.

2.3.2 Using SNPedia's API

SNPedia incorporates a section explaining how database queries can be automated by means of different programming languages[8].

2.3.3 Accessing `gff` file

It is also possible to download a file in `gff` (General File Format)[9] format which, according to the SNPedia web page, is updated in a *semi-regular* basis. The `gff` file format is composed of lines representing characteristics or features. Each line is divided into tab separated files. Alternatively, it is also possible to find the so-called *track definition lines*, which allow to configure feature sets. For the case of SNPedia, each feature correspond to a SNP, for which varied information is provided. This information includes the SNP identifier, a link to a web page describing it, or a brief summary.

[2] http://www.ncbi.nlm.nih.gov/pubmed
[3] http://www.ncbi.nlm.nih.gov/omim
[4] https://www.pharmgkb.org/
[5] http://www.nextbio.com/b/nextbioCorp.nb
[6] https://www.23andme.com/
[7] http://www.snpedia.com/index.php/Category:Is_a_snp
[8] http://www.snpedia.com/index.php/Bulk
[9] http://www.snpedia.com/files/gbrowse/SNPedia.gff

2.4 Related Software

SNPedia provides software allowing to automate certain tasks, such as the Promethease tool, or the code for programming software bots.

2.4.1 Promethease Tool

Promethease[10] is freely available software that allows to compare a personal genetic profile against the information stored in the SNPedia database. The result generated by means of Promethease includes information about specific features of the individual (such as susceptibility to diseases) based on the presence of certain SNPs within the genome.

2.4.2 Programming software bots

If we want to access SNPedia information in an automated manner using software bots, it is necessary to follow specific rules, since there is a limit in the number of accesses per time unit. There are GitHub software projects currently under development that try to adhere to the rules established by the SNPedia designers.

3 The *snptools* package

The *snptools* package presented in this document is useful to perform basic accessing operations to the SNPedia database. On one hand, the software allows to generate reports about SNPs in an automated manner. On the other hand, it also makes possible to select the sets of SNPs of interest for which the reports should be generated using different criteria. This functionality has been implemented by means of shell scripting, AWK and Python as programming languages.

snptools has been designed to work in Linux platforms. In the following sections we describe the necessary steps to correctly install the package, providing a general overview of the implemented tools as well as the basic functionality they incorporate. After that, a series of sample uses will be shown and finally, we will discuss possible plans to extend the package functionality in the future. Additionally, Appendix A shows a sample report in `pdf` format for a set of SNPs identified by means of the tools provided by the implemented software.

3.1 Prerequisites

The *snptools* package is based on different software libraries that should be installed so as to to access to the provided functionality. The list of such libraries as well as some basic instructions for their proper installation in shown below:

- ***wikitools***: is a Python library for manipulating information in *MediaWiki* format. It can be installed after downloading a package provided in different formats[11], including for instance the well known `rpm` format, which can be converted to a `dpkg` file for its installation in Ubuntu systems.

- ***mwparserfromhell***: is a Python package that can be used to parse *MediaWiki* code. This library can be installed by downloading it from GitHub[12].

[10] http://www.snpedia.com/index.php/Promethease
[11] https://pypi.python.org/pypi/wikitools
[12] https://github.com/earwig/mwparserfromhell

- ***pandoc***: is a tool useful to convert between different file formats that use markup languages. It can be installed in Ubuntu systems by means of the `apt` command, selecting the *pandoc* package.

- ***NLTK***: is a Python library providing varied routines for natural language processing. NLTK can be installed via the `pip` command[13].

- ***WordNet***: *WordNet* is a lexical database for the English language providing information about nouns, adjectives, verbs and adverbs in synonym sets called *synsets*, each one of them expressing different concepts. *WordNet* can be downloaded and used by means of Python. For this purpose, it is necessary to start an interactive Python session and after installing *NLTK*, typing:

  ```
  import nltk
  nltk.download()
  ```

  After that, a dialog box opens. In this box is necessary to verify whether the server address is up to date or not. If the application returns an "HTTP 404" error code, then the existing address can be replaced by the following one: `http://nltk.github.com/nltk_data`. Once the previous step has been completed, we should click on *Corpora* option and then choose the *wordnet* option. Finally, we need to click on the *Download* button.

3.2 Installation

3.2.1 Basic Installation Procedure

To install the *snptools* package, it is necessary to follow the next steps:

1. Decompress package using `tar` command and execute `cd` to the directory where decompression was made.

2. Type `./reconf` to create configuration files.

3. Type `./configure` to configure package.

4. Execute `make` to build the package.

5. Type `make install` to install all of the programs as well as some data files.

6. Finally, after installing the package it is possible to remove no longer necessary files by executing `make clean`.

By default, files are installed under the `/usr/local` folder (or similar depending on the used operating system); however, Step 5 requires super user privileges. Alternatively, a different folder can be specified at Step 3 by typing:

```
$ configure --prefix=<absolute-installation-path>
```

For instance, if the user with name `user1` wants to install the package in the `/home/user1/snptools` folder, the required sequence of commands to be executed would be the following:

[13] http://www.nltk.org/install.html

```
$ configure --prefix=/home/user1/snptools
$ make
$ make install
```

As a result of installing the package, the following two folders are created:

- `${PREFIX}/bin`: folder containing the tools that compose the *snptools* package.
- `${PREFIX}/share/snptools/examples`: folder storing sample results using *snptools* (see Section 3.6 for additional details).

Where `${PREFIX}` represents the installation folder that was set when configuring the package.

3.2.2 Add *snptools* to the System PATH

To end the installation process, it might be useful to add *snptools* to the system PATH. This will allow us to easily execute commands provided in the package without the necessity of providing the whole *snptools* installation path.

For this purpose, we can execute the following commands:

```
$ SNPTOOLS_HOME_DIR=<absolute-installation-path>
$ export PATH=$PATH:${SNPTOOLS_HOME_DIR}/bin
```

These variable definitions can be added to the `.bashrc` user profile file, so as to define them automatically whenever a new interactive shell session is started.

3.3 Overview of Provided Tools

Depending on their functionality, the tools implemented in the package can be classified in two categories. First, there are tools that allows us to generate reports from a set of SNPs contained in a plain text file. Second, *snptools* also incorporates tools to identify SNPs of interest without the necessity of downloading the whole SNPedia database.

Below we enumerate the tools related to report generation:

- `gen_snp_report`: it is the main executable file in this group of tools. It is useful to generate SNPs reports in different file formats given a list of identifiers. It internally uses the `retrieve_snp_info` tool described below.
- `retrieve_snp_info`: program used to establish a web connection to the SNPedia web site and retrieve the information related to a specific SNP provided as input parameter.

Regarding the programs to identify SNPs, the package incorporates the following list:

- `download_snp_gff`: very simple program that downloads the file in `gff` format mentioned in Section 2.3.3.
- `extract_gff_keywords`: given the file in `gff` format obtained with the previous tool, extracts lists of keywords associated to the SNPs that are later used to select them.
- `gen_snp_list`: starting from the output of the `extract_gff_keywords` tool, obtains the identifiers of a set of SNPs given a list of keywords.

- `filter_spec_pos`: utility to filter natural language words based on the specific *part-of-speech* (POS) in which those words are classified. It is internally used by `extract_gff_keywords`.

- `obtain_hypernyms`: tool to obtain hypernyms[14] of natural language words. In an arbitrary manner, the tool gets the first hypernym appearing in the list generated by means of the *WordNet* database. This program is internally used by `extract_gff_keywords`.

- `tokenize`: tool useful to tokenize natural language text, it is internally used by `extract_gff_keywords`.

- `get_snp_names`: program to obtain the name of all of the SNPs contained in SNPedia (this utility is just a proof of concept on accessing SNPedia content based on the instruction provided in its web site).

3.4 Generating Reports for a Set of SNPs

The *snptools* package allows to generate reports about SNPs in different file formats. For this purpose it is necessary to provide a file with SNPs identifiers as well as the path of the output file, whose extension will determine the format. Below there is a list of the supported file formats:

- `csv`: *comma separated value* file. It is a very basic plain text fie format that can be easily used by different applications and programming languages, such as R, where this kind of files can be loaded via the `read.table()` function, obtaining an object of `data frame` type.

- `md`: *markdown* format.

- `pdf`: file in `pdf` format incorporating hyperlinks to SNPs stored in SNPedia.

- `html`: file in `html` format incorporating links to SNPedia SNPs.

Below it is shown the command line used to generate a SNP report in `pdf` format. In the example we use the file with thrombosis-related SNP identifiers included in the *snptools* package (see more information regarding the folder created after package installation in Section 3.2):

```
snp_file=${PREFIX}/share/snptools/thrombosis_snps.ids
gen_snp_report -s ${snp_file} -o thrombosis_snps.pdf
```

3.5 Identifying SNPs of Interest

When a list of SNPs is already available, it is relatively easy to generate reports in an automated manner or data files that can be read within other applications or programming languages such as R. Nevertheless, identifying the list of SNPs of interest using SNPedia is not easy, at least provided that we access the information using its web site. This is because the only information available there is a list of links to SNPs labeled with an

[14] Hypernym: general term that can be used to refer to the concept represented by a more specific term called *hyponym.*

alphanumeric identifier, which is not very informative. If we wanted to collect additional information, we could download the whole database by accessing the database web. However, this is a very time consuming task, due to the great amount of stored entries, and also due to the fact that it is not permitted the implementation of software bots that carry out the download without imposing certain constraints regarding the number of database entries that can be downloaded per time unit, as it is explained in the *Frequently Asked Questions* Section provided in the SNPedia web site.

One possible solution to this problem would be to work with the file in `gff` format, which contains a summary of the database entries. The file is updated in a semi-regular manner. Among the different kinds of information contained in the `gff` file, there is a field that can be useful to identify sets of SNPs of interest. Specifically, the field is labeled as *Note* and seems to contain some sort of abstract related to each SNP. It is worthy of note here that in spite of the fact that the *Note* field is present for the vast majority of SNPs stored in SNPedia, there are some of them that lack this information. Moreover, the *Note* field constitutes an incomplete summary of the SNP, since it is truncated to a certain length for all of the entries. However, in the *snptools* package, this information is used to tackle the problem of SNP identification. The most recent version of the `gff` file can be downloading by typing the following:

```
download_snp_gff
```

The downloaded file is stored in the current directory.

3.5.1 Extraction of Keywords

One of the main tools included in the *snptools* package for SNP identification is `extract_gff_keywords`. This tools takes as input parameter the path of the file in `gff` format as well as the specific criterion we want to use to identify keywords. The output consists in two new files, one containing a list of keywords along with the number of times each keyword appears (in particular, the number of SNPs containing the keyword in the *Note* field), as well as a file where the SNPs are linked to the keywords. Optionally, it is possible to restrict ourselves to those keywords with a frequency equal or greater than a given threshold.

Regarding the criterion used to obtain the keywords, we defined three of them, each one with a specific identifier:

- `vocab`: words contained in *Note* field of the `gff` file are tokenized, lowercased and after that we extract the vocabulary composed of all of the tokens, associating to each one of them its frequency in the `gff` file.

- `pos`: the same as `vocab`, but the tokens are first semantically analyzed using *WordNet*, filtering only those that are classified as certain parts of speech (POS). In this particular case, we retain the tokens corresponding to nouns or adjectives. This allow us to remove non relevant information produced by the `vocab` criterion, since for instance, very frequent and non informative words such as "with" or "the" are filtered out.

- `hyper`: the same as `pos`, but *WordNet* is used again over the filtered tokens to obtain hypernyms for each keyword, allowing us to obtain more general terms as well as a smaller number of them under the same conditions, simplifying the process of SNP identification.

	#Keywords
vocab	1 475
pos	1 090
hyper	592

Table 1: Quantity of extracted keywords using three different extraction criteria: `vocab`, `pos` and `hyper`.

Once the different selection criteria have been introduced, we proceed to show sample uses of the `extract_gff_keywords` tool. If the tool is executed without parameters, a brief help message is displayed explaining the available input parameters. For instance, to extract keywords using the `vocab` criterion and focusing only in those whose frequency is higher or equal to 10 we can use the following command line:

```
extract_gff_keywords -f SNPedia.gff -c vocab -v 10 -o extract_vocab_10
```

The result of the previous command generates two different output files whose name starts with prefix `extract_pos_10` (which is specified by means of the `-o` option). The file with `kw` extension contains a list of keywords, while the `snp_kw` extension associated to the other file shows the association between SNP identifiers and keywords. The list of extracted keywords with frequency information can be obtained by means of the following command:

```
less extract_vocab_10.kw
```

We can also visualize the relationship between SNP identifiers and keywords by executing:

```
less extract_vocab_10.snp_kw
```

Alternatively, we can obtain the list of keywords using the `pos` and `hyper` criteria by means of the two following commands:

```
extract_gff_keywords -f SNPedia.gff -c pos -v 10 -o extract_pos_10
extract_gff_keywords -f SNPedia.gff -c hyper -v 10 -o extract_hyper_10
```

Analyzing the number of keywords following each one of the three criteria can be interesting to illustrate their relative differences. Table 1 shows such quantities. As it can be observed, `vocab` is the criterion generating a greater quantity of keywords, followed by the `pos` criterion and finally `hyper`.

It can also be interesting to compare the keywords obtained for three criteria. For this purpose, we visually inspected the files with `kw` extension. From this comparison, we observed that the `pos` criterion excluded non informative words that are present when applying the `vocab` criterion. On the other hand, the keywords obtained by means of the `hyper` criterion was totally different to those obtained by `vocab` y `pos`. In spite of the fact that the obtained hypernyms were not always useful, the criterion allowed us to obtain some interesting keywords, such as those corresponding to groups of diseases: *blood_disease*, *cardiovascular_disease*, *eye_disease*, *infectious_disease*, *inflammatory_disease*, etc.

Additionally, if we inspect the `extract_hyper_10.snp_kw` file, it is possible to see the relationship between hyponyms and hypernyms that have been established by means of the *WordNet* database for the words contained in the *Note* field for each SNP.

3.5.2 Obtaining SNP Lists

Once the list of keywords have been generated and we have selected those in which we are interested, the next step is to obtain a list of SNP identifiers associated to such keywords. From this list we will be ready to generate reports or `csv` data files in the way that was explained in Section 3.4.

To generate the SNP list, we can use the `gen_snp_list` command. Such command takes as input the prefix of the output files generated by means of the `extract_gff_keywords` tool. For instance, assuming that we chose the word *thrombosis* from the list of keywords generated by means of the `pos` criterion. The required command line to obtain the SNP list would be the following:

```
gen_snp_list extract_pos_10 thrombosis
```

The output of the tool is printed to the standard output.

It is also possible to obtain those SNPs related to the *cardiovascular_disease* hypernym by executing:

```
gen_snp_list extract_hyper_10 cardiovascular_disease
```

The `extract_gff_keywords` utility accepts more than one keyword. In addition to this, in spite of the fact that we have studied individual keywords, it is also possible to provide terms composed of more than one word, which will be searched in the *Note* field for the different SNPs. One sample use of this could be finding those SNPs related to early onset obesity:

```
gen_snp_list extract_pos_10 obesity "early onset"
```

As it can be observed, if we want to use terms composed of multiple words, we need to enclose them by means of quotes. Otherwise, they would be considered individually.

3.6 Examples

The *snptools* package incorporates a set of sample files collecting some of the results obtained by means of the commands shown in previous sections. Once the package has been installed, such sample uses are stored in the `${PREFIX}/share/snptools/examples` folder, where `${PREFIX}` refers to the installation folder specified during package configuration (see Section 3.2).

Below we briefly describe the included sample files:

- `cardiovascular_disease_snps.ids`: file containing SNPs related to cardiovascular diseases. Those SNPs were obtained by means of the criterion to extract keywords based on hypernyms (`hyper`).
- `extract_{vocab|pos|hyper}_10.kw`: file containing keywords extracted via the three extraction criteria explained above, namely, `vocab`, `pos` y `hyper` (see Section 3.5). Those keywords with frequency below 10 are discarded.
- `extract_{vocab|pos|hyper}_10.snp_kw`: files containing the correspondence between each SNP and their associated keywords, for the three implemented keyword extraction techniques.
- `obesity_early_onset_snps.ids`: file containing those SNPs related to early onset obesity.

- `thrombosis_snps.ids`: file of SNP identifiers related to the *thrombosis* keyword.
- `thrombosis_snps.{csv|md|html|pdf}`: files containing SNP reports in different formats for the *thrombosis* keyword.

3.7 Possible Extensions

Implemented software for the *snptools* package should be viewed as a first approximation to the work with SNPs provided by SNPedia. This first approximation can be extended in different ways. We consider that the most important one is the identification of SNPs of interest discussed in Section 3.5. In this regard, the tools that were implemented exclusively work with the information provided in the `gff` file. The motivation behind this decision is to avoid downloading the whole SNPedia database. However, the information contained in the `gff` file is pretty limited, and the *Note* field we have used to generate keyword lists is not always available for all of the SNPs. Additionally, the maximum length of such field seems to be restricted, and hence we could expect more interesting results if we were able to work with more complete information. Perhaps this hypothesis could motivate the download of the whole SNPedia database or using information contained in additional ones, such as the dbSNP database.

On the other hand, another aspect of the software that could be extended in many different ways is the use of *WordNet* to obtain word categories. Such categories are obtained, in current version of the package, based on the information about hypernyms contained in *WordNet*. In particular, the implemented code arbitrarily selects the first hypernym contained in the list for each word. However, it could also be interesting to find better informed ways to select it, or perhaps selecting multiple terms. In addition to this, *WordNet* incorporates other features beyond hypernym extraction that could be interesting to exploit. One example of this would be the similarity measures between words[15].

[15] To find out more about additional *WordNet* functionality using NLTK, check the following page: http://www.nltk.org/howto/wordnet.html

A Sample Use: SNP Report for *thrombosis* Keyword

rs1523127

- **rsid**: 1523127
- **Chromosome**: 3
- **position**: 119782192
- **GMAF**: 0.4578
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (G;G)
- **geno2**: (G;T)
- **geno3**: (T;T)
- **StabilizedOrientation**: minus
- **Gene**: NR1I2
- **Gene_s**: NR1I2

rs13146272

- **rsid**: 13146272
- **Chromosome**: 4
- **position**: 186199057
- **GMAF**: 0.4555
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (A;A)
- **geno2**: (A;C)
- **geno3**: (C;C)
- **StabilizedOrientation**: plus
- **Gene**: CYP4V2
- **Gene_s**: CYP4V2

rs6048

- **rsid**: 6048
- **Chromosome**: X
- **position**: 139551121
- **GMAF**: 0.1578
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (A;A)
- **geno2**: (A;G)
- **geno3**: (G;G)
- **StabilizedOrientation**: plus

- **Gene**: F9
- **Gene_s**: F9

rs2227589

- **rsid**: 2227589
- **Chromosome**: 1
- **position**: 173917078
- **GMAF**: 0.1253
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (A;A)
- **geno2**: (A;G)
- **geno3**: (G;G)
- **StabilizedOrientation**: minus
- **Gene**: SERPINC1
- **Gene_s**: SERPINC1

rs3093030

- **rsid**: 3093030
- **Chromosome**: 19
- **position**: 10286727
- **GMAF**: 0.314
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (C;C)
- **geno2**: (C;T)
- **geno3**: (T;T)
- **StabilizedOrientation**: plus
- **Gene**: ICAM1
- **Gene_s**: ICAM1,ICAM4

rs670659

- **rsid**: 670659
- **Chromosome**: 1
- **position**: 240998475
- **GMAF**: 0.3365
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (C;C)
- **geno2**: (C;T)
- **geno3**: (T;T)

- **StabilizedOrientation**: plus
- **Gene**: RGS7
- **Gene_s**: RGS7

rs5988

- **rsid**: 5988
- **Chromosome**: 6
- **position**: 6151904
- **GMAF**: 0.1882
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (C;C)
- **geno2**: (C;G)
- **geno3**: (G;G)
- **StabilizedOrientation**: minus
- **Gene**: F13A1
- **Gene_s**: F13A1

rs2066865

- **rsid**: 2066865
- **Chromosome**: 4
- **position**: 154604124
- **GMAF**: 0.3186
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (C;C)
- **geno2**: (C;T)
- **geno3**: (T;T)
- **StabilizedOrientation**: minus
- **Gene**: FGG
- **Gene_s**: FGG

rs9898

- **rsid**: 9898
- **Chromosome**: 3
- **position**: 186672838
- **GMAF**: 0.4633
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (C;C)
- **geno2**: (C;T)

- **geno3**: (T;T)
- **StabilizedOrientation**: plus
- **Gene**: HRG
- **Gene_s**: HRG

rs1613662

- **rsid**: 1613662
- **Chromosome**: 19
- **position**: 55536595
- **GMAF**: 0.1309
- **Assembly**: GRCh37
- **GenomeBuild**: 37.1
- **dbSNPBuild**: 132
- **geno1**: (A;A)
- **geno2**: (A;G)
- **geno3**: (G;G)
- **StabilizedOrientation**: plus
- **Gene**: GP6
- **Gene_s**: -

rs4524

- **rsid**: 4524
- **Chromosome**: 1
- **position**: 169542517
- **GMAF**: 0.2438
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (A;A)
- **geno2**: (A;G)
- **geno3**: (G;G)
- **StabilizedOrientation**: minus
- **Gene**: F5
- **Gene_s**: F5

rs28934892

- **rsid**: 28934892
- **Chromosome**: 21
- **position**: 43058927
- **GMAF**: -
- **Assembly**: GRCh38
- **GenomeBuild**: 38.1
- **dbSNPBuild**: 141
- **geno1**: (C;C)

- **geno2**: (C;T)
- **geno3**: (T;T)
- **StabilizedOrientation**: minus
- **Gene**: CBS
- **Gene_s**: CBS

References

Cariaso, Michael and Greg Lennon (2012). "SNPedia: a wiki supporting personal genome annotation, interpretation and analysis." In: *Nucleic Acids Research* 40.Database-Issue, pp. 1308–1312. URL: http://dblp.uni-trier.de/db/journals/nar/nar40.html#CariasoL12.

Wheeler, David L. et al. (2004). "Database resources of the National Center for Biotechnology Information: update". In: *Nucleic Acids Research* 32 (Database Issue), pp. D35–D40. URL: http://nar.oxfordjournals.org/cgi/content/full/32/suppl_1/D35.