**Expanding HGNChelper package for more genomic symbols correction**

by

Warner Alexis



Levi Waldron (Site Supervisor)

Hunter College Grade School of Public Health

May 22, 2016

**Integration of data in HGNChepler for broader applicability**

**Abstracts**

HGNChelper is an R package developed by Waldron Lab that has the ability to correct and format genes using the HGNC Gene Symbols and Affymetrix Probeset Identifiers. Waldron lab, a public health bioinformatics lab, previously found errors in older publications regarding the usage of HGNC gene symbols that have been converted to date format by Excel such as, DEC1 that automatically converted to 1-DEC. The goal of this project is to expand HGNChelper package so that it can correct genomic symbols to apply algorithm from the International HapMap Project to calculate correlations between any pair of single nucleotide polymorphisms (SNPs), and provide a function to identify calculate Linkage Disequilibrium (LD) for any string of SNP identifiers and integrating cell line curation and drug identifiers. We implemented an RSQLite database, which contains the HGNC datasets with more 33,000 genes symbols, the small nucleotide polymorphism, and the cell line and drug treatment identifiers. We made an entity relational diagram of how information would flow among the identifiers. The database is made with four tables including 15 Gigabytes of genomics identifiers such as the SNPs. The next step of this project is to migrate a fully operational and functional package in Bioconductor. HGNChelper will be of use to researchers who want to correct errors caused by spreadsheets software, by recognizing genomics symbols and beneficiary for mapping of publishing genomic signatures. Our package will correct any symbol that exists in our package repository.

**Introduction**

Gene symbols are very important for researchers because of their associated roles in biological function of genes, as well as relationship among other species. Waldron’s Lab previously developed HGNChelper package using R Studio software by identifying and automatically correcting errors in HGNC Gene Symbols and Affymetrix Probeset Identifiers. This package is built to convert genes that have been transmogrified by Excel date formatting. We redesigned and implemented new features in the HGNChelper package to reflect a more broad applicability in correcting different identifiers. Due to the large size of data implemented, we changed the way data flow through the package by creating an RSQLite database to add new datasets to the package, which contains the following datasets: (1) the HGNC datasets, which contain approved unique symbols and names for human loci, (a) protein coding genes, (b) ncRNA genes and (c)pseudogenes, (2) single nucleotide polymorphism identifiers obtained from NCBI Database, (3)curation cell line and, (4)drug identifiers table obtained by Dr. Need Bejamin from GitHub repository “DRUGNETS”.[2]  This informatics project intends not only to improve decisions making in genomic studies , but to increase performances and reliability of the package.

The HUGO Gene Nomenclature Committee is a public database that assigns standardized nomenclature for approved human names and genes for protein coding and other genomic features found within human genome. HGNC gives official abbreviations to all the of human genes names. The HUGO Gene Nomenclature Committee is placed at the European Bioinformatics Institute. In 1977, the Human Gene Mapping community recognized the need for a database to approve human gene names and hence, created HGNC database. Thirty years later, HGNC had more than 27,000 genes symbols at their disposal [4]. All the gene entries are curated manually and the HGNC abbreviations for names and symbols are particularly assigned to use in publications and databases where specific gene is discussed or referenced [4]. On the HGNC webpage, every gene is associated with a report that contains varieties of information about the history of the gene. The description of a gene is very simple. Once queried the gene name, he HGNC webpage will give the following information: *HGNC ID , Approved Name, Approved Symbol, Previous Gene Symbol and Name , Synonyms, Locus type, Chromosomal location, , Gene family, HCOP and all the reporting symbol reports from other databases*[5].   
The HGNC ID is a unique identification number provided for every single gene. You can query any gene symbols or name with their ID. Approved Name corresponds to the approved symbol denoted by HGNC. The report also includes the previous full history of Gene Symbol and Names, Chromosomal location, Gene family and links for other reporting databases. Genes are usually associated with SNPs. Small nucleotide polymorphisms are single nucleotide substitutions of one base for another that occur in more than one percent of the population. Small nucleotides polymorphisms database was first used by Genome-wide association studies (GWAS), which pointed out genetic loci involved in human genome as well as identified small variations. The database (dbSNP) was implemented by NCBI in collaboration with the National Human Genome Research (NHGRI)[8]. The dbSNP datasets consist of single nucleotide polymorphisms (SNP), short deletion and insertion of polymorphism, microsite markers, multi nucleotide polymorphisms (MNP).

According to Marie Louis, a Waldron lab intern, SNPs represent a promising tool for finding genetic determinants of complex diseases and comprehending the differences in drug response. Scientists’ main purposes were to find SNPs in the human genome that is related to a particular disease. Marie Louis reported that *reliable SNPs could serve as predictive markers that educate choices about various parts of medical care, including particular diseases, adequacy of different medications and antagonistic responses to particular medications* [9]. Researchers reported two issues on how SNPs overlap from two distinct studies. One issue is that dbSNP IDs are frequently converged in later version of dbSNP giving the same SNP for wide range of aliases and different preferred *rs* numbers for the same SNP over time[9]. The second issue is that it is very hard to update dbSNP with current information and discoveries of new variant. Single Nucleotide Polymorphism Annotation Platform (SNAP) is a web server that identify and annotate SNPs using the HapMap Project to calculate correlations and disequilibrium for any string of SNP. The SNAP sever provide a better way to understand and compare Genome-wide Association Study results, applies an algorithm for mapping boundaries and checks genes SNPs aliases across dbSNP repository *(Andrew Johnson and al.*). This informatics project is to analyze and integrate dbSNP dataset from NCBI and presented challenges to researchers attempting to re-analyze these data and use HGNC algorithm to help them correct gene symbols.

Over the past few years, the same problems have occurred with annotating cell line curation and unique drug identifiers. Cell lines are very important in biomedical research as they are widely used in diagnostic tools and therapeutic target, knowing that each line has its own features and can be used for specific studies (*Paolo and al.).* It is significant that repositories need to have the same detailed information about a unique cell line identifiers. Most websites and other resources provide vast amounts of information that are inconsistent. Cell Line Data Base (CLDB) was created to overcome these limitations so cell line identifiers can have a unique official symbol and link with multiple references. Due to inconsistency in cell line information, researchers found difficulties to uncover new biomarkers. It is very important that annotation from either cell line or unique drug database is reliable. Mathew J. Garnett and al. stated in an article that the NC160 cell line panel and associated drug screened developed the method of using cell line to use drug sensitivity with genotype data. For instance, drugs like EFGR, BRAF, EML4-ALK are subsequently linked with other cell lines. Symbols are important for identifying genes and linking them to their specific genomic features. Incorporating a simple and friendly package where invalid symbols can be reformatted to their official norms will be the main function behind creating HGNCheler R package.

**Materials and Methods**

HGNChelper is a package designed to correct gene symbols that has been converted to date format by Excel. The previous version of HGNChelper was based on the whole HGNC datasets with contains two columns. One column is called “Symbol” gives all the available genes symbols and the column is called “Approved.Symbol” contains the entire approved and corrected gene symbol. We changed the name of the second column because of the RSQLite requirements that state that column names cannot contains a period in any position. This version will continue on using the same functions, but will adhere to the role of correcting symbols including SNPs, HGNC, cell line curation and drug identifiers. The functions are *affyToR, checkGeneSymbols, findExcelGeneSymbols, hgnc.table, rToAffy rToSymbol* and, *symbolTo*. These functions are built in the package to target all HGNC invalid genes and then convert them to valid gene symbols.

**HGNChelper Function*****checkGeneSymbols*** is the main function of this package. The purpose of *checkGeneSymbols* is to identify genes that are outdated or mogrified by Excel or any spreadsheet program [1]. If a variable is assigned, the result will include three columns; one column containing the input data, another column showing whether the symbols are valid and the last column with a corrected gene list [1]. Every function built in R Studio comes with a condition. The usage of *checkGeneSymbols* requires some arguments.

checkGeneSymbols(x, unmapped.as.na=TRUE, hgnc.table=NULL)

“X” represents a vector of genes that have either outdated or mogrified values. Unmapped.as.na establish the condition if TRUE unmapped symbol will appear as NA in the Suggested.Symbol column. If FALSE, there will be no correction. Hgnc.table contains dataset that the package uses to correct genes symbol. It has all the current and withdrawn symbols from genenames.org [1]. If data in column names “*Symbol*” and *Approved.Symbol*” is similar, then it correct the gene symbol. In our improved version, Hgnc.table is replaced with HGNChelper1.sqlite . this SQLite file contains all the data to correct gene symbols, SNP , cell line curation and drug identifiers. Dr. Lewi Waldron and Markus Reister first designed these functions. We only used their function as a template to remodel the new version.

**affyToR** converts Affymetrix probeset identifier to valid R names. According GeneChip Mouse Expression, a probeset is a collection of probes designed to interrogate a given sequence [3]. A probeset symbol look like that : “*12345\_at or 12345\_a\_at or 12345\_s\_at or 12345\_x\_at*”. It may be only digits. The last three characters identify the probe set strand. In HGNChelper, the usage of this function only requires a vector x of the Affymetrix probeset identifiers and the value will include simply prepend *“affy”*.

affyToR(x)

**findExcelGeneSymbols**

findExcelGeneSymbols tracks down all the Excel mogrified gene symbols or genes from other spreadsheets programs. This will return a vector of the same length containing all mapped symbols. According to Dr. Waldron, this function is superseded by checkGeneSymbols, which corrects Excel-mogrified gene symbols as well as aliases and outdated symbols [1].

findExcelGeneSymbols(x, mog.map = read.csv(system.file("extdata/mog\_map.csv", package = "HGNChelper"), as.is = TRUE), regex = "[0-9]\\-(JAN|FEB|MAR|APR|MAY|JUN|JUL|AUG|SEP|OCT|NOV|DEC)|[0-9]\\.[0-9] [0-9]E\\+[[0-9][0-9]")

HGNChelper is built with a data frame that has two columns “*the original and mogrified*” , which contain both correct and incorrect symbols. Mog\_map is the map of all known mogrification. Regex stands as the regular expressions that identify gene symbol. We need to integrate mog.map data in the sqlite file in order for the function to be operational. The result of this function is possible because of mog.map data inserted into the package. The *rToAffy* requires the same condition as affyToR(x). It converts back the output of affyToR(x) into its initial input.

***symbolTo(x)* and *rsymbolTo(x)***

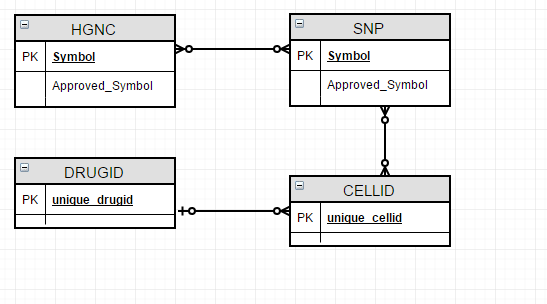
***symbolTo*** uses vector of HGNC symbols and convert it to a R names . According Waldron lab’s CRAN description , the result will include the same length vector as x which can be converted to HGNC using the command *rsymbolTo.* Those two commands use the HGNC dataset to convert the vector of genes provided. The previous version can be downloaded using command “ install.packages(“HGNChelper”) or via this link [https://cran.r-project.org/web/packages/HGNChelper/index.html](http://index.html).

This project requires great understanding in how functions are formulated in R Studio platform. The R platform offers varieties of packages that can adhere to other programming languages and supports different data science techniques. Database skills are very important for R data analysis. The platform tends to crash whenever the input file is too large, and for that particular reason, we prefer to implement a RSQLite to this package. However, several packages provide techniques to parse through file and run queries to get information out , but RSQLite allows to leverage your data analysis skills .

**RSQLite package**

RSQLite package is widely used in Bioinformatics packages in R Studio. Once installed, this database engine is embedded within R Server. It coexists inside the R server. Data do not have to be stored in a specific location or in server and are saved in files with SQLite extension. There is no need for a separate installation for a Database Interface because it is built with the package. In our case HGNChelper datasets would run the background and therefore it would only require a connection to make it functional. First, we installed the package using the following command: install.packages(“RSQLIte”). The next step is then created a database called “*HGNChelper1.sqlite*”, with the following tables. *HGNC , SNP , cellid, drugid and mog.map.* We used Structure Query Language to connect the datasets to the database. All the tables have a relationships among them. We built the database using the following command: *db <- dbConnect(SQLite(), ‘HGNChelper1.sqlite’).* It is not yet a database until we integrate data into it. db is a variable that assume the value given to the database. It will be use in all our script.

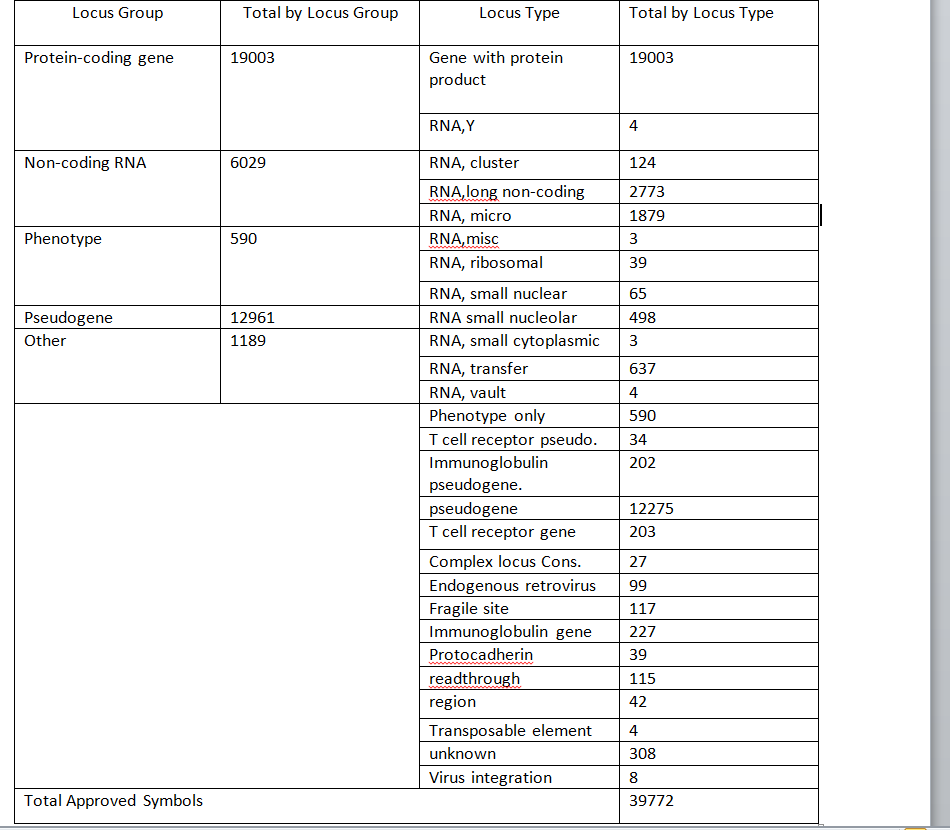
This screenshot shows a relational diagram of the HGNChelper database.



Gene from HGNC can be associated with a SNP identifier or multiple ones or not associated with one at all. The same entity relationship goes for CELLID and CELLID can either associated to DRUGIG or are not associated at all and DRUGID can have at least a CELLID associated with it.

**HGNC table**

This table contains the current HGNC datasets. It only has the gene symbols for differents genomic features . The table below provides a statistical report of the number genes associated locus groups and types [7].



We extracted the whole dataset from the previous HGNC helper package. The newly avaliable datasets can be downloaded via this <http://www.genenames.org/cgi-bin/statistics> , but needs to parse to actually get all the gene symbol in a data frame. In order to extract the gene symbols from HGNChelper backend, we used the following command :  
data("hgnc.table", package="HGNChelper", envir=environment())

hgnc <-hgnc.table

The first column extracts the file from the package and the second line reads it. We created a csv file with the result of the second command, then implemented it the database using the following command: “dbWriteTable( db, "hgnc", "hgnc.csv", overwrite = TRUE, sep = ",",eol = "\r") “. HGNC genes are formulated with Upper-case letters and Arabic numerals. The result of the executed command “hgnc <-hgnc.table” after extracting the datasets will show the full list of genes:

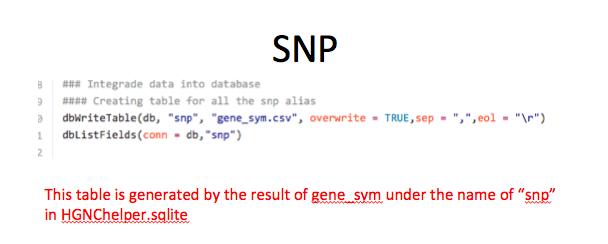


**SNP**

We wrote an R script to download thirty chromosomes from <ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606_b142_GRCh38/chr_rpts/> , then unzip them and after read the first of identifiers. We add “rs” to the beginning of their ID and arranged them so that every chromosome file has one column containing “rs”ID” such as “rs171, rs242”. To find the SNP aliases, we are manually uploaded them to SNAP Proxy Server one by one. This script produces the two column maps required by HGNChelper. The figure below gives a brief description on what all the syntaxes do:



A csv file is created under the name of *gene\_sym,* then implemented in the database with the following command:



**Unique cell line and drug Identifiers**

We download the datasets in *csv* format for both cell line and drug curation from the following link <https://github.com/bhklab/DRUGNET> , then we extracted the column of containing the identifiers from both .*csv* files then we made another csv containing only the result. The script is as follows:  
ht <- read.csv("matching\_cell.csv") ## commands that read cell line curation

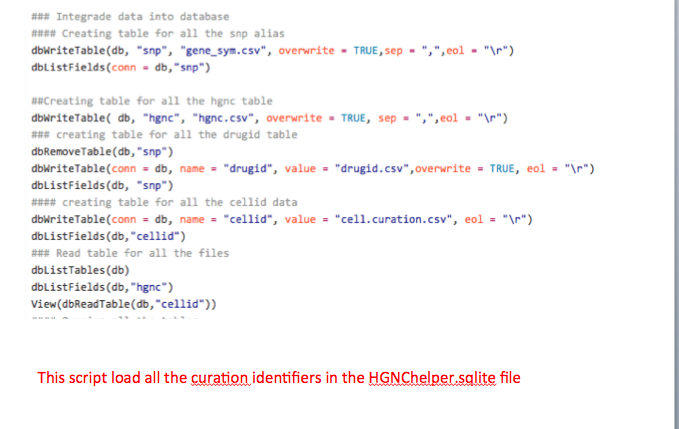
ho <-read.csv("matching\_drug.csv") ## commands that read drug identifiers

cell <- ht$unique.cellid ## select only one data

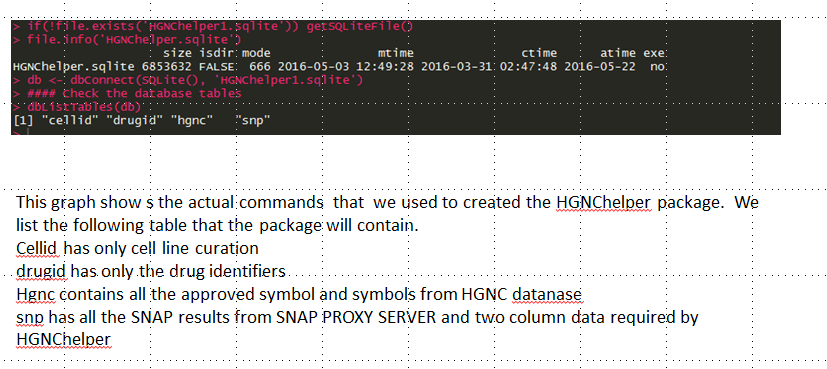
write.csv(cell, "cellid",row.names=FALSE, na="",col.names=FALSE, sep=",") ## write a csv file of the data

drug <-ho$unique.drugid

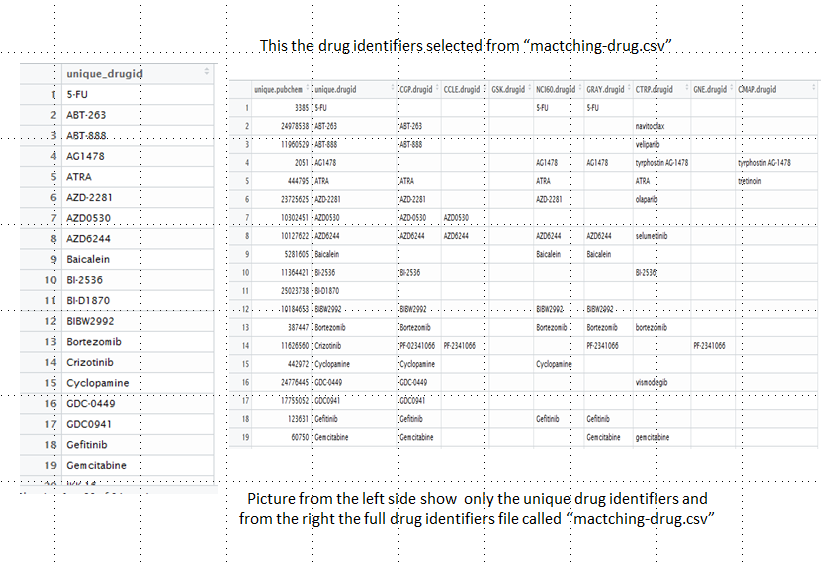
write.csv(cell, "drugid",row.names=FALSE, na="",col.names=FALSE, sep=",") ## write a csv file of the data



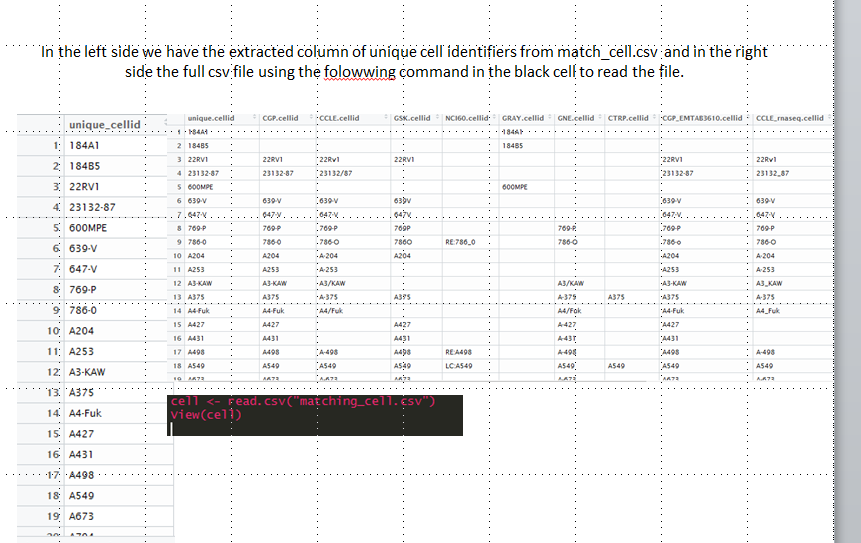
**Result**



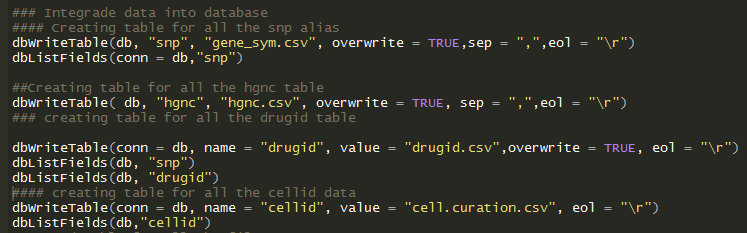
After downloading from DRUGNET GitHub repository, we select online the unique identifiers from matching-drug.csv



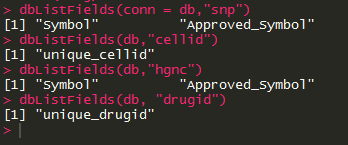
After we load the cell line curation and extracted the column for its identifiers



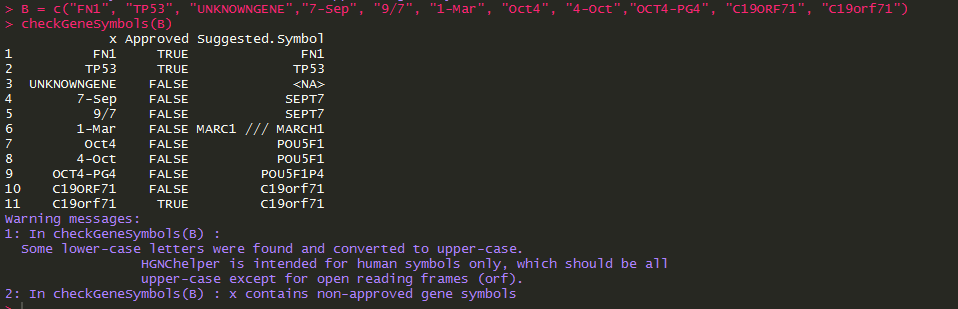
We create table using this specific command



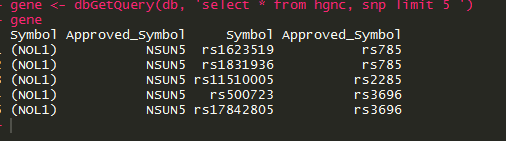
After the creation of table in the database, we run those queries to know if tables are created with the right table name and the right data



The function has been tested and evaluated by Wadron to see if it need more work. We found out that this function need to be remodeled because of the condition it was constructed. It can only read Capital letter genes symbol. Based on our available data, SNP ,curation cell line and drug identifiers contains lower case letter. We have reached our main objective. Further works need to be addressed to make the package fully operational.



This screenshot shows that SQL executed perfectly to give the first approved column of both snp and hgnc table



**Discussion**

Working with large amounts of data requires a lot of attention on how they are organized and filter. There were some challenges in generating files from the output of the identifiers. We believed that the use of spreadsheet programs to analyze genomic data is not efficient in managing large amounts of data. First, every file created in csv format automatically added x column containing the number of each row. This was an issue that affected the function of the package. We filter the file to delete all the hidden characters that keep changing during reformatting from text to csv. One of our objectives was to produce a table that has both the cell line and drug identifiers. At first, we did it, but both csv files have different rows of data. Cell line identifiers file has 681 rows while drug identifiers only contain 98 rows. Data from the drug identifiers keep copying rows multiple times until the 680th column is filled up. We had to rewrite the script so we could have two separate tables. One has the cell line identifiers and the other one the drug identifiers.

Another challenge was that relational database systems couldn’t contain a period in the construction of the table. This affects the Data Manipulation Language within the database system. At the end of the project, we couldn’t query any data from any table, except the “Symbol” field. We redesigned the table so that queries no longer contained any hidden character such “/r” and “.” in their column name.

We believe that informatics project can advance further.  If we look at the outcomes of HGNChelper in the past years, the package was tested against highly accessed database. It showed all the incorrect gene symbols and correct majority of the gene. Waldron should continue this project to increase the outcomes of this package. We integrate SNPs, cell line and drug curation in a specific database to have more features to work with in the HGNChelper. In near future, Waldron will be able to integrate more genetics features and tables to the existing database. The purpose of the Waldron Lab is to make HGNChelper capable of correcting any genetic symbols that are widely used in computational biology.

**Future Work**Functions were tested during the last step this project. HGNChelper is built using an algorithm that allows input of only capital letter gene symbol. Small nucleotide polymorphisms, cell line and drug identifiers contain not only capital letters and also lower letters. HGNChelper is expanding to facilitate users with more applicability in correcting symbols. Certain functions need to be rewritten so they could meet HGNChelper old requirement. The final stage of this informatics is to migrate this package from Comprehensive R Archive Network (CRAN) to Bioconductor for computational biology.

**Reference:**

1- HGNChelper. (n.d.). Retrieved from <https://cran.rproject.org/web/packages/HGNChelper/index.html>

2- DRUGNET GitHub Repository. Retreived from

<https://github.com/bhklab/DRUGNET>

3- Affymetrix probeset identifier <http://www.affymetrix.com/support/help/faqs/mouse_430/faq_8.jsp>

4- Gray KA, Daugherty LC, Gordon SM, Seal RL, Wright MW, Bruford EA. Genenames.org: the HGNC resources in 2013. *Nucleic Acids Research*. 2013;41(Database issue):D545-D552. doi:10.1093/nar/gks1066

5- Approved Symbol and Gene Symbol. Retrieved from  
[http://www.genenames.org/help/symbol-report#Approved\_Symbol](http://www.genenames.org/help/symbol-report" \l "Approved_Symbol)

6- HGNC Gene name . Retrieved from.

http://www.genenames.org

7- HGNC helper Statistics and Download repository. Retrieved from .

<http://www.genenames.org/cgi-bin/statistics>

8- Smigielski EM, Sirotkin K, Ward M, Sherry ST. dbSNP: a database of single nucleotide polymorphisms. *Nucleic Acids Research*. 2000;28(1):352-355.

9- Baranov VS. Genome Paths: A Way to Personalized and Predictive Medicine. *Acta Naturae*. 2009;1(3):70-80.