SOFTWARE TOOL ARTICLE

REVISED HGNChelper: identification and correction of invalid gene symbols for human and mouse [version 2; peer review: 3 approved]

Sehyun Oh[1,2], Jasmine Abdelnabi[1,2], Ragheed Al-Dulaimi[1-3], Ayush Aggarwal[4,5], Marcel Ramos[1,2], Sean Davis[6], Markus Riester[7], Levi Waldron[1,2]

[1]Epidemiology and Biostatistics, Graduate School of Public Health and Health Policy, City University of New York, New York, 10027, USA
[2]Institute for Implementation Science and Population Health, New York, 10027, USA
[3]School of Medicine, University of Utah, Utah, 84132, USA
[4]CSIR-Institute of Genomics and Integrative Biology, New Delhi, 110025, India
[5]Academy of Scientific and Innovative Research, Ghaziabad, Uttar Pradesh, 201 002, India
[6]Center for Cancer Research, National Cancer Institute, Maryland, 20892, USA
[7]Novartis Institutes for BioMedical Research Incorporation, Massachusetts, 02139, USA

v2 **First published:** 21 Dec 2020, **9**:1493
https://doi.org/10.12688/f1000research.28033.1
Latest published: 09 Jun 2022, **9**:1493
https://doi.org/10.12688/f1000research.28033.2

Abstract

Gene symbols are recognizable identifiers for gene names but are unstable and error-prone due to aliasing, manual entry, and unintentional conversion by spreadsheets to date format. Official gene symbol resources such as HUGO Gene Nomenclature Committee (HGNC) for human genes and the Mouse Genome Informatics project (MGI) for mouse genes provide authoritative sources of valid, aliased, and outdated symbols, but lack a programmatic interface and correction of symbols converted by spreadsheets. We present HGNChelper, an R package that identifies known aliases and outdated gene symbols based on the HGNC human and MGI mouse gene symbol databases, in addition to common mislabeling introduced by spreadsheets, and provides corrections where possible. HGNChelper identified invalid gene symbols in the most recent Molecular Signatures Database (MSigDB 7.0) and in platform annotation files of the Gene Expression Omnibus, with prevalence ranging from ~3% in recent platforms to 30-40% in the earliest platforms from 2002-03. HGNChelper is installable from CRAN.

Keywords

gene symbols, molecular biology, HGNC, MGI

Open Peer Review

Approval Status ✓ ✓ ✓

	1	2	3
version 2 (revision) 09 Jun 2022		✓ view	
version 1 21 Dec 2020	✓ view	? view	✓ view

1. **Mikhail G. Dozmorov**, Virginia Commonwealth University, Richmond, USA
2. **Susan Tweedie**, European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, UK
3. **Marcin Cieślik**, University of Michigan Medical School, Ann Arbor, USA

Any reports and responses or comments on the article can be found at the end of the article.

This article is included in the RPackage gateway.

Corresponding author: Levi Waldron (Levi.Waldron@sph.cuny.edu)

Author roles: Oh S: Investigation, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Abdelnabi J**: Investigation, Writing – Review & Editing; **Al-Dulaimi R**: Investigation, Writing – Review & Editing; **Aggarwal A**: Software; **Ramos M**: Software, Writing – Review & Editing; **Davis S**: Supervision, Writing – Review & Editing; **Riester M**: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Waldron L**: Formal Analysis, Investigation, Methodology, Project Administration, Software, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by the National Cancer Institute (NCI) grant U24-CA180996 to LW.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Oh S, Abdelnabi J, Al-Dulaimi R *et al.* **HGNChelper: identification and correction of invalid gene symbols for human and mouse [version 2; peer review: 3 approved]** F1000Research 2022, **9**:1493 https://doi.org/10.12688/f1000research.28033.2

First published: 21 Dec 2020, **9**:1493 https://doi.org/10.12688/f1000research.28033.1

REVISED **Amendments from Version 1**

This revision addresses comments raised by reviewers, with the most significant changes being 1) addition of a Limitations section, 2) comparison to the limma packages alias2Symbol and alias2SymbolTable functions, and 3) improvement of the readability of the figure.

Any further responses from the reviewers can be found at the end of the article

Introduction

Gene symbols are widely used in biomedical research because they provide descriptive and memorable nomenclature for communication. However, gene symbols are constantly updated through the discoveries and re-identification of genes, resulting in new names or aliases. For example, *GCN5L2* (*G*eneral *C*ontrol of amino acid synthesis protein *5-L*ike *2*) is a gene symbol that was later discovered to function as a histone acetyltransferase and therefore renamed as *KAT2A* (*K*(lysine) *A*cetyl *T*ransferase *2A*)[1]. In addition to the rapid and constant updates on valid gene symbols, commonly used spreadsheet software, such as Microsoft Excel, modify some gene symbols, converting them into dates or floating-points numbers[2,3]. For example, '*DEC1*', a symbol for '*D*eletion in *E*sophageal *C*ancer *1*' gene, can be exported in date format, '1-DEC'. There have been attempts to rectify gene symbol issues, but they have largely been limited to Excel-modified gene symbols. Also the suggested solutions often reference static files with the corrections curated at the time of publication[3] or comprise scripts for detecting the existence of Excel-modified gene symbols without correction[2]. In recognition of the importance of the spreadsheet modification issues, HGNC offers its own symbol correction tool, the Multi-symbol checker, and also recently announced that all symbols that auto-convert to dates in Excel have been changed[4]. However, much literature and public data still contains outdated and incorrect gene symbols, motivating a convenient method of systematic detection and correction. To systematically identify historical aliases, correct for capitalization differences, and simultaneously correct spreadsheet-modified gene symbols, we built the HGNChelper R package. HGNChelper maps different aliases and spreadsheet-modified gene symbols to approved gene symbols maintained by The HUGO Gene Nomenclature Committee (HGNC) database[5]. HGNChelper also supports mouse gene symbol correction based on the Mouse Genome Informatics (MGI) database[6].

Methods

Implementation

Source data. Human gene symbols are accessed from HGNC Database ftp site (ftp://ftp.ebi.ac.uk/pub/databases/genenames/new/tsv/hgnc_complete_set.txt)[7] and mouse gene symbols are acquired from MGI Database (http://www.informatics.jax.org/downloads/reports/MGI_EntrezGene.rpt)[6]. These URLs, and their access and processing, are handled by HGNChelper so the user does not interact directly with them.

Algorithm. Human gene symbol correction is processed in three steps. First, capitalization is fixed: all letters are converted to upper-case, except the open reading frame (orf) nomenclature, which is written in lower-case. Second, dates or floating-point numbers generated via Excel-modification are corrected using a custom index generated by importing all human gene symbols into Excel, exporting them in all available date formats, and collecting any gene symbols that are different from the originals. In the last and most commonly applied step, aliases are updated to approved gene symbols in the HGNC database. Mouse gene symbol correction follows the same three steps as in human gene symbol correction, except the capitalization step since mouse gene symbols begin with an uppercase character, followed by all lowercase.

User interface. The user interface of HGNChelper does not include any local input or output files; instead it uses R data structures as function arguments and output. Base R data export functions such as write.table can be used to write results to file in whichever format required. The input arguments to the main function, checkGeneSymbols, are:

1. **x**: A character vector of gene symbols to check for modified or outdated values
2. **chromosome**: An optional integer vector the same length as x, providing chromosome numbers for each gene
3. **unmapped.as.na**: A logical value, if TRUE (default), unmapped symbols will appear as NA in the Suggested. Symbol output column. If FALSE, the original unmapped symbol will be kept.
4. **map:** An optional user-updated or non-standard gene map. The default maps can be updated by running the interactive example provided in the help page to checkGeneSymbols.
5. **species:** A required character vector of length 1, either "human" (default) or "mouse".

checkGeneSymbols returns an R data.frame with one row per input gene and three columns:

1. The first column of the data frame shows the input gene symbols.
2. The second column indicates whether the input symbols are valid.
3. The third column provides a corrected gene symbol where possible.

A message is printed indicating when the package's built-in map was last updated. Because the gene symbol databases are updated as frequently as every day, we provide the getCurrentHumanMap and getCurrentMouseMap functions for updatingthe reference map without requiring an HGNChelper software update. These functions fetch the most up-to-date version of the map from HGNC and MGI, respectively, and users can provide the output of these functions through the map argument of checkGeneSymbols function. However, fetching a new map requires internet access and takes longer than using the package's built-in index.

Operation

HGNChelper is an R package installable from CRAN on Linux, Windows, and OSX. It requires a base installation of R (> 3.5.0) and no other dependencies, and has minimal hardware requirements that should be met by any computer capable of installing the R dependency.

Results

To evaluate the performance of HGNChelper, we quantified the extent of invalid gene symbols present in platform annotation files in the Gene Expression Omnibus (GEO) database from 2002 to 2020. We downloaded 20,716 GEO platform annotation (GPL) files using GEOquery::getGEO[8], of which 2,044 platforms were suspected to contain gene symbol information based on matching to valid symbols. There is a clear trend of increasing proportion of invalid gene symbols with age of platform submission (Figure 1), ranging from an average of ~3% for recent platforms and increasing with age to ~20% in 2010 and 30–40% in the earliest platforms from 2002–03. The overall proportion of valid gene symbols was 79%, increasing to 92% after HGNChelper correction. We also checked the validity of gene symbols in the Molecular Signatures Database (MSigDB 7.0)[9]. Out of 38,040 gene symbols used in MSigDB version 7.0, 850 were invalid, and this number reduces to 453 after HGNChelper correction, of which the majority were lncRNA and a few withdrawn symbols.

The limma[10] Bioconductor package provides related functionality; however, limma::alias2Symbol and limma::alias2SymbolTable are intended only to translate known gene aliases, whereas HGNChelper is intended for heterogeneous input that may include aliases, valid symbols, Excel-modified symbols, incorrect capitalization, and unmappable symbols, and to provide a map between input and output. limma::alias2SymbolTable maintains the length of the output vector as same as the input, but if there are multiple aliases, it displays only the one with the lowest Entrez ID number, whereas HGNChelper returns a delimited vector of all aliases.

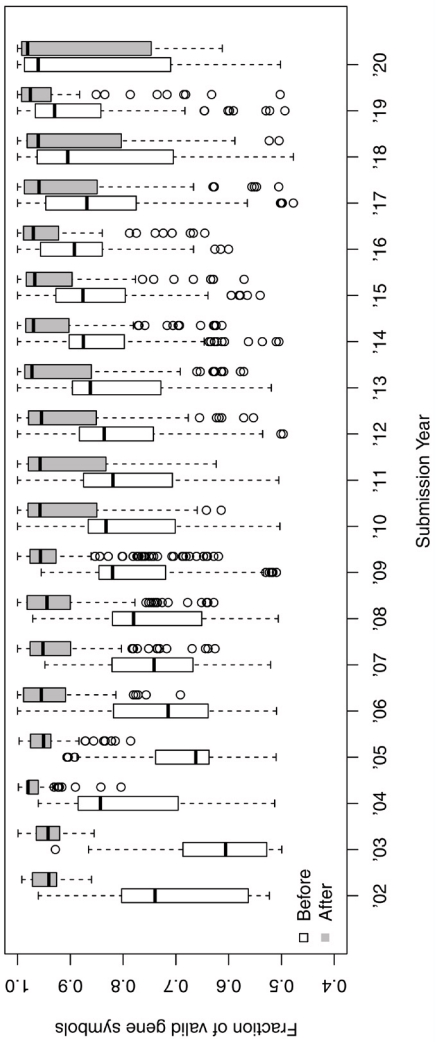

Figure 1. The fraction of valid gene symbols in GPL files grouped by year of data submission. Each dot represents a unique GPL. Older entries show a smaller fraction of valid gene symbols than more recent entries (Before, white box), but many of which are successfully corrected by HGNChelper (After, grey box).

Discussion

Gene symbols are error-prone and unstable, but remain in common use for their memorability and interpretability. Our analysis of public databases containing gene symbols emphasizes the need for gene symbol correction particularly when using symbols from older datasets and reported results. Such correction should be routinely done when gene symbols are part of high-throughput analysis, such as re-analysis of targeted gene panels for precision medicine, which tend to be annotated with gene symbols (e.g. 11), in Gene Set Enrichment Analysis using the gene symbol versions of popular databases such as MSigDB[9] or GeneSigDB[12], or when performing systematic review or meta-analysis of published multi-gene signatures (e.g. 13). HGNChelper implements a programmatic and straightforward approach to the routine identification and correction of invalid gene symbols.

Limitations

We reduced the fraction of invalid gene symbols in GPL files using HGNChelper (Figure 1), but there are still 8% remaining, invalid gene symbols. We further investigated the cases where HGNChelper failed to fix and identified the following situations:

1. Long non-coding RNAs (e.g. "*lnc-ARMCX4-1*", "*lnc-SOX11-1*")
2. Withdrawn symbol (e.g. "*OCLM*")
3. Uncharacterized gene (e.g. "*LOC644669*"): *Symbols beginning with LOC. When a published symbol is not available, and orthologs have not yet been determined, this may be represented as 'LOC' + the GeneID.*
4. Non-human gene symbol
5. Missing data
6. Commercial product name (e.g. Probe ID)

Another limitation with HGNChelper is that it cannot always provide the correct answer for which gene a symbol refers to. For example, *FHL1* is both an approved symbol and an alias of *CFH*, so unless the chromosome of *CFH* is specified, *FHL1* will be just returned as a valid symbol. Thus, we recommend users to provide as much information as possible and still be cautious in interpretation of its output.

Software availability

Package available from CRAN: https://cran.r-project.org/package=HGNChelper

Source code available from: https://github.com/waldronlab/HGNChelper/

Archived source code as at time of publication: https://doi.org/10.5281/zenodo.4309985[13]

License: GPL (≥ 2.0)

Acknowledgements

An earlier version of this article can be found on bioRxiv (doi: https://doi.org/10.1101/2020.09.16.300632)

This work was supported by National Cancer Institute (NCI) grant U24-CA180996 to L.W.

References

1. Poux AN, Cebrat M, Kim CM, *et al.*: **Structure of the GCN5 histone acetyltransferase bound to a bisubstrate inhibitor.** *Proc Natl Acad Sci U S A.* 2002; **99**(22): 14065-70.
PubMed Abstract | Publisher Full Text | Free Full Text
2. Zeeberg BR, Riss J, Kane DW, *et al.*: **Mistaken identifiers: gene name errors can be introduced inadvertently when using Excel in bioinformatics.** *BMC Bioinformatics.* 2004; **5**: 80.
PubMed Abstract | Publisher Full Text | Free Full Text
3. Ziemann M, Eren Y, El-Osta A: **Gene name errors are widespread in the scientific literature.** *Genome Biol.* 2016; **17**(1): 177.
PubMed Abstract | Publisher Full Text | Free Full Text
4. Bruford EA, Braschi B, Denny P, *et al.*: **Guidelines for human gene nomenclature.** *Nat Genet.* 2020; **52**(8): 754–758.
PubMed Abstract | Publisher Full Text | Free Full Text
5. Yates B, Braschi B, Gray KA, *et al.*: **Genenames.org: the HGNC and VGNC resources in 2017.** *Nucleic Acids Res.* 2017; **45**(D1): D619–D625.
PubMed Abstract | Publisher Full Text | Free Full Text
6. Bult CJ, Blake JA, Smith CL, *et al.*: **Mouse Genome Database (MGD) 2019.** *Nucleic Acids Res.* 2019; **47**(D1): D801–D806.
PubMed Abstract | Publisher Full Text | Free Full Text
7. **Home | HUGO Gene Nomenclature Committee.** [cited 2 May 2020].
Reference Source
8. Davis S, Meltzer PS: **GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor.** *Bioinformatics.* 2007; **23**(14): 1846–1847.
PubMed Abstract | Publisher Full Text
9. Liberzon A, Subramanian A, Pinchback R, *et al.*: **Molecular signatures database (MSigDB) 3.0.** *Bioinformatics.* 2011; **27**(12): 1739–1740.
PubMed Abstract | Publisher Full Text | Free Full Text
10. Ritchie ME, Phipson B, Wu D, *et al.*: ***limma* powers differential expression analyses for RNA-sequencing and microarray studies.** *Nucleic Acids Res.* 2015; **43**(7): e47.
PubMed Abstract | Publisher Full Text | Free Full Text
11. McCabe MJ, Gauthier MEA, Chan CL, *et al.*: **Development and validation of a targeted gene sequencing panel for application to disparate cancers.** *Sci Rep.* 2019; **9**(1): 17052.
PubMed Abstract | Publisher Full Text | Free Full Text
12. Culhane AC, Schwarzl T, Sultana R, *et al.*: **GeneSigDB--a curated database of gene expression signatures.** *Nucleic Acids Res.* 2010; **38**(Database issue): D716–25.
PubMed Abstract | Publisher Full Text | Free Full Text
13. Waldron L, Haibe-Kains B, Culhane AC, *et al.*: **Comparative meta-analysis of prognostic gene signatures for late-stage ovarian cancer.** *J Natl Cancer Inst.* 2014; **106**(5): dju049.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔ ✔

Version 2

Reviewer Report 27 June 2022

https://doi.org/10.5256/f1000research.133588.r140199

✔ **Susan Tweedie**
HUGO Gene Nomenclature Committee (HGNC), European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, UK

The authors have addressed my major concern by adding a Limitations section.

Competing Interests: HGNC also provide a symbol checking tool but this is not an R package.

Reviewer Expertise: Gene nomenclature, Genomics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 02 February 2021

https://doi.org/10.5256/f1000research.31006.r76417

✔ **Marcin Cieślik**
Michigan Center for Translational Pathology, University of Michigan Medical School, Ann Arbor, USA

HGNChelper is a particularly valuable tool in the toolbox of a bioinformatics practicioneer. It addresses a real problem, which while superficially trivial, actually affects the quality of analyses.

I use HGNChelper ALL THE TIME especially if a dataset ends up being garbled by Excel. It is easy to use, in most recent versions reasonably fast, and frankly just gets the job done

The paper is a short description of the motivation and implementation. With a short study detailing the evolution of gene symbolss.

What I miss in the paper is an assessment of HGNChelper failures (to increase my confidence in the tool). For example how often are symbols converted incorrectly because the same gene symbol was used, at different times, to denote different genes.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Cancer Genetics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 27 Apr 2022

Levi Waldron, Graduate School of Public Health and Health Policy, City University of New York, USA

Thank you for reviewing our manuscript and for your encouraging comments.

Comment 1: The main issue raised is when HGNChelper fails to map symbols, it is important for users to understand the limitations of what the tool can and cannot map.

To address this point we manually reviewed many cases where HGNChelper correction efficiency is low in Figure 1 and almost all unmapped symbols fell into one of the following categories:

1. Long non-coding RNAs (e.g. “lnc-ARMCX4-1”, “lnc-SOX11-1”)
2. Withdrawn symbol (e.g. “OCLM”)
3. Uncharacterized gene (e.g. “LOC644669”): Symbols beginning with LOC. When a published symbol is not available, and orthologs have not yet been determined, this may be represented as ‘LOC’ + the GeneID.
4. Non-human gene symbol
5. Missing data
6. Commercial product name (e.g. Probe ID)

We have added this to the manuscript section “Limitations”.

Competing Interests: No competing interests were disclosed.

Reviewer Report 20 January 2021

https://doi.org/10.5256/f1000research.31006.r76418

?

Susan Tweedie
HUGO Gene Nomenclature Committee (HGNC), European Molecular Biology Laboratory, European Bioinformatics Institute, Hinxton, UK

The paper describes an R package for that checks whether human and mouse symbols match an HGNC or MGI approved symbol and if not suggests a replacement by, correction of capitalization, correction of Excel date and floating-point transformations and matching to alias symbols.

The rationale for developing the new software tool is generally clearly explained. As the authors point out, symbols do change (although the HGNC are now committed to making as few symbol changes as possible) and there is a need to check symbols are valid. As the human symbols that convert to dates in Excel have all been changed, this should be less of a problem going forward. However, these mangled symbols persist in historic data sets and some authors will undoubtedly continue to use problematic aliases such as OCT3 so that aspect of the tool is helpful. It may be worth adding that conversion of gene symbols to floating-point numbers in Excel is more of an issue for mouse genes with RIKEN identifiers than human gene symbols.

The authors should also address whether there are any other R packages that have similar functionality. The name HGNChelper could lead some to think this is an HGNC endorsed tool; given that a symbol checking tool is already available from the HGNC (https://www.genenames.org/tools/multi-symbol-checker/) (albeit not an R package) the authors

should mention this exists and ideally compare the functionality of their tool versus the HGNC tool.

While this is likely to be a useful tool for R users it should come with a few words of caution given that you cannot always be completely sure which gene a symbol refers to in the absence of confirmation via an ID or other additional information. For example, FHL1 is both an approved symbol and an alias of CFH so while FHL1 is a valid symbol the input data may refer to CFH. There are also cases where a symbol is an alias for several genes but not an approved symbol itself e.g. NIP (or Nip) which not an approved symbol but is an alias of GIPC1, DUOXA1, and CRPPA. The authors should clarify how the algorithm deals with cases where an input symbol matches more than one gene. This would address my concerns about whether this is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool.

I note that the input contains optional chromosome information but there is no mention of how this is used – does the algorithm take this information into account when determining whether a symbol is valid or not?

The authors note that some symbols in their test that could not be updated were lncRNAs and pseudogenes. As both of these classes of gene are named by HGNC it would be good to expand on why the tool failed with these genes – do these particular genes lack an HGNC symbol?

Figure 1 is hard to interpret - it is unclear what the white boxes mentioned in the legend refer to. Improving the clarity of the figure would address my concerns about whether the conclusions about the tool and its performance adequately supported by the findings presented in the article.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: The HGNC has a symbol checking tool with some of the functionality of the tool described in the paper.

Reviewer Expertise: Gene nomenclature, Genomics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 27 Apr 2022

Levi Waldron, Graduate School of Public Health and Health Policy, City University of New York, USA

Thank you for reviewing our manuscript and for your constructive comments. Below are our responses to the individual comments.

Comment 1: It may be worth adding that conversion of gene symbols to floating-point numbers in Excel is more of an issue for mouse genes with RIKEN identifiers than human gene symbols.

The reviewer is correct that human gene symbols prone to Excel conversion have now been changed, but many still exist in the literature and public databases as demonstrated in Figure 1. HGNChelper does not currently fix RIKEN identifiers, so we don't draw this comparison in the manuscript.

Comment 2: The authors should also address whether there are any other R packages that have similar functionality. The name HGNChelper could lead some to think this is an HGNC endorsed tool; given that a symbol checking tool is already available from the HGNC (https://www.genenames.org/tools/multi-symbol-checker/) (albeit not an R package) the authors should mention this exists and ideally compare the functionality of their tool versus the HGNC tool.

We now compare HGNChelper with the function alias2Symbol from the limma package. This is described in the response to comment 1 from Reviewer 1.
Thank you for pointing us to the HGNC's own tool, the Multi-symbol checker. We understand that our package name, HGNChelper, can potentially imply the endorsement from HGNC, so we clarified in the manuscript that Multi-symbol checker is the tool supported by HGNC. We also compared the HGNChelper and Multi-symbol checker from HGNC. Here are the major points that differentiate these tools:

1. Implementation: Multi-symbol checker is a web-based UI tool. Users can provide an input as a comma- or space- separated list of gene symbols, directly typing-in or uploading the file. Outputs are displayed in the interface as a sortable table and users can choose to download it as a csv file. HGNChelper is a R package, which takes an input as a character vector and outputs the result as a data frame, which can be saved and exported in a different format, such as csv, tsv, rds, etc.
2. Excel-modified gene symbols: HGNChelper corrects inputs in a potentially excel-modified format (e.g. "9/7", "1-Mar", "Oct4"), and suggest the original symbol (e.g. "SEPTIN7", "MARCHF1 /// MTARC1", "POU5F1"). This functionality is not part of multi-symbol checker - it marks them as 'Unmatched' as with any other unmatchable

symbol.

3. Chromosome location: Multi-symbol checker provides the chromosome location as a part of the default output, if the approved symbol is available for a given input. HGNChelper provides the chromosome information only if it is provided with the input gene symbol - it validates whether the input chromosome information is correct or not, and if it's wrong, gives the correct chromosome location.

Comment 3: While this is likely to be a useful tool for R users it should come with a few words of caution given that you cannot always be completely sure which gene a symbol refers to in the absence of confirmation via an ID or other additional information. For example, FHL1 is both an approved symbol and an alias of CFH so while FHL1 is a valid symbol the input data may refer to CFH.

In this example, FHL1 will be returned as a valid symbol, unless the chromosome of CFH is specified. For example:

```
> checkGeneSymbols("FHL1")
    x Approved Suggested.Symbol
1 FHL1     TRUE           FHL1
> checkGeneSymbols(c("FHL1", "FHL1"), chromosome = c("X", "1"))
    x Approved Suggested.Symbol Input.chromosome Correct.chromosome
1 FHL1     TRUE           FHL1              X                X
2 FHL1    FALSE            CFH              1                1
```

We have added this discussion to the "Limitations" section.

Comment 4: There are also cases where a symbol is an alias for several genes but not an approved symbol itself e.g. NIP (or Nip) which is not an approved symbol but is an alias of GIPC1, DUOXA1, and CRPPA. The authors should clarify how the algorithm deals with cases where an input symbol matches more than one gene. This would address my concerns about whether this is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool.

When an input symbol matches ambiguously to more than one gene, HGNChelper displays all the matching genes. For example, HGNChelper::checkGeneSymbols("NIP") will return "CRPPA /// DUOXA1 /// GIPC1'' as the suggested symbols.

- If there is only one valid gene symbol matched with the input, HGNChelper simply evaluates whether the provided chromosome information is correct or not, and if it's incorrect, outputs the correct chromosome location under the 'Correct.chromosome' column. For example, HGNChelper::checkGeneSymbols("NIP", chromosome = 1) will return "CRPPA /// DUOXA1 /// GIPC1'' as the suggested symbol and "7 /// 15 /// 19'' as the correct chromosome.
- If the input matches more than one gene, the chromosome information is used to specify the suggested gene symbol. For example, HGNChelper::checkGeneSymbols("NIP", chromosome = 7) will return "CRPPA'' as the suggested symbol and "7'' as the correct chromosome.
- lncRNAs and pseudogenes can be updated as long as they are not 'uncharacterized

genes', whose symbols start with 'LOC'. Based on NCBI, when a published symbol is not available and orthologs have not yet been determined, gene will provide a symbol that is constructed as 'LOC' + the GeneID. So HGNChelper can not update them because there are no approved gene symbols for them.

Comment 5: Figure 1 - Improve the clarity of the figure
We apologize for the confusing color display. Color schema for Figure 1 is fixed in the updated manuscript.

Competing Interests: No competing interests were disclosed.

Reviewer Report 18 January 2021

https://doi.org/10.5256/f1000research.31006.r76531

Mikhail G. Dozmorov
Biostatistics Department, Virginia Commonwealth University, Richmond, VA, USA

The manuscript "HGNChelper: identification and correction of invalid gene symbols for human and mouse" by Oh S. et al. describes the HGNChelper R package that corrects the common problem of misformatted gene symbols and aliases. The package works with both human and mouse gene symbols. The manuscript describes the procedure for obtaining current gene symbols and creating a mapping between the correct and invalid gene symbols. Analysis of >20K GEO datasets demonstrated the pervasive presence of invalid gene symbols, with the proportion of such symbols decreasing over time. Applied to the recent version of MSigDB, it shows the presence of 850 invalid gene symbols; half of them can be corrected with HGNChelper. The manuscript is well-written, and the package is made using R best practices.

Minor comments about the manuscript include:

- The limma R package has the alias2Symbol function with similar functionality. How the functionality of HGNChelper differs or improves upon this function?

- Figure 1 - Before/after boxplots look identical in the black and white version. Please, correct.

The following comments about the package interface are suggestive.

- The current output of the checkGeneSymbols() function returns a data frame with three columns (x, Approved, Suggested.Symbol). Suggesting including an argument "simplify" (TRUE by default) that will return one vector of the same length and order as the original vector of gene symbols, with NAs replacing non-mappable symbols. The rationale is to use this function as a wrapper around the original vector of gene symbols, e.g., checkGeneSymbols(my_genes), returning a drop-in replacement vector of corrected gene symbols. An example is the p.adjust() function that, given a vector of p-values, returns a vector of p-values corrected for multiple testing.

- The package provides separate functions to update gene maps. These updated gene maps can then be used in the checkGeneSymbols() function. Suggesting including an argument "update.map" (FALSE by default) to checkGeneSymbols(), that, if TRUE, will automatically update gene maps.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: Co-organize the Bioconductor conference. I declare that I provided an impartial review.

Reviewer Expertise: Bioinformatics, genomics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 27 Apr 2022

Levi Waldron, Graduate School of Public Health and Health Policy, City University of New York, USA

Thank you for reviewing our manuscript and for your constructive comments. Below are our responses to the individual comments.

Comment 1: The limma R package has the alias2Symbol function with similar functionality. How does the functionality of HGNChelper differ or improve upon this function?

The biggest difference is that limma::alias2Symbol and limma::alias2SymbolTable are intended only to translate known gene aliases, whereas HGNChelper is intended for heterogeneous input that may include aliases, valid symbols, Excel-modified symbols,

incorrect capitalization, and unmappable symbols, and to provide a map between input and output. limma::alias2SymbolTable maintains the length of the output vector as same as the input, but if there are multiple aliases, it displays only the one with the lowest Entrez ID number, whereas HGNChelper returns a delimited vector of all aliases. The following example demonstrates these differences:

```
> library(HGNChelper)
> input = c("FN1", "TP53", "UNKNOWNGENE",
+         "7-Sep", "9/7", "1-Mar",
+         "Oct4", "4-Oct", "OCT4-PG4",
+         "C19ORF71", "C19orf71", "NIP")
> checkGeneSymbols(input)
Maps last updated on: Mon Sep 28 18:31:21 2020
           x Approved          Suggested.Symbol
1         FN1     TRUE                       FN1
2        TP53     TRUE                      TP53
3  UNKNOWNGENE    FALSE
4       7-Sep    FALSE                   SEPTIN7
5        9/7    FALSE                   SEPTIN7
6       1-Mar    FALSE        MARCHF1 /// MTARC1
7        Oct4    FALSE                    POU5F1
8       4-Oct    FALSE                    POU5F1
9    OCT4-PG4    FALSE                  POU5F1P4
10   C19ORF71    FALSE                   C19orf71
11   C19orf71     TRUE                   C19orf71
12        NIP    FALSE CRPPA /// DUOXA1 /// GIPC1
Warning messages:
1: In checkGeneSymbols(input) :
 Human gene symbols should be all upper-case except for the 'orf' in open reading frames.
The case of some letters was corrected.
2: In checkGeneSymbols(input) : x contains non-approved gene symbols
> library(limma)
> library(org.Hs.eg.db)
> alias2Symbol(alias = input)
[1] "FN1"      "TP53"     "C19orf71" "GIPC1"    "DUOXA1"
> alias2SymbolTable(alias = input)
 [1] "FN1"      "TP53"     NA         NA         NA         NA
 [7] NA         NA         NA         NA         "C19orf71" "GIPC1"
Warning message:
In alias2SymbolTable(alias = input) :
 Multiple symbols ignored for one or more aliases
```

Additionally, limma::alias2Symbol uses Bioconductor org*.db packages to map aliases for multiple organisms. org*.db packages in turn pull data from NCBI and update it with each Bioconductor release. HGNChelper is a CRAN package with no dependency on non-base packages, and instead downloads aliases directly from the HUGO and MGI projects. limma::alias2Symbol however provides an advantage of supporting any organism for which

an org*.db package is available, whereas HGNChelper supports only human and mouse.

Comment 2: Figure 1 - Before/after boxplots look identical in the black and white version. Please, correct.

We apologize for the confusing color display. The color schema for Figure 1 is fixed in the updated manuscript.

Comment 3: The current output of the checkGeneSymbols() function returns a data frame with three columns (x, Approved, Suggested.Symbol). Suggesting including an argument "simplify" (TRUE by default) that will return one vector of the same length and order as the original vector of gene symbols, with NAs replacing non-mappable symbols. The rationale is to use this function as a wrapper around the original vector of gene symbols, e.g., checkGeneSymbols(my_genes), returning a drop-in replacement vector of corrected gene symbols. An example is the p.adjust() function that, given a vector of p-values, returns a vector of p-values corrected for multiple testing.

This is a good use case, but we are reluctant to allow a function argument to change the class (and the contract) of what the function returns. Motivated by arguments for "Type consistency" such as by Gillespie and Lovelace (Efficient R programming, https://csgillespie.github.io/efficientR/, section 3.5.2), we think it is less error-prone to require a simple but explicit step to change data class. We've added an example to the checkGeneSymbols help page to provide a straightforward solution in this use case:

```
> human
 [1] "FN1"        "TP53"       "UNKNOWNGENE" "7-Sep"      "9/7"        "1-Mar"
 [7] "Oct4"       "4-Oct"      "OCT4-PG4"    "C19ORF71"    "C19orf71"
> checkGeneSymbols(human)$Suggested.Symbol
Maps last updated on: Thu Mar 25 08:36:49 2021
 [1] "FN1"              "TP53"             NA                 "SEPTIN7"
 [5] "SEPTIN7"          "MARCHF1 /// MTARC1" "POU5F1"          "POU5F1"
 [9] "POU5F1P4"         "C19orf71"         "C19orf71"
Warning messages:
1: In checkGeneSymbols(human) :
  Human gene symbols should be all upper-case except for the 'orf' in open reading frames.
The case of some letters was corrected.
2: In checkGeneSymbols(human) : x contains non-approved gene symbols
```

Comment 4: The package provides separate functions to update gene maps. These updated gene maps can then be used in the checkGeneSymbols() function. Suggesting including an argument "update.map" (FALSE by default) to checkGeneSymbols(), that, if TRUE, will automatically update gene maps.

We are reluctant to add an argument that automatically updates the data resource first because results from the same code and the same version of HGNChelper would produce different results at any time since the HGNC and MGI databases change frequently, and secondly because the potential for unnecessary heavy load on the HGNC database could

result in restrictions on the bulk downloads we rely on. To maintain reproducibility by default we require the user to download and save the map if they want a version newer than what HGNChelper has, an approach also compatible with caching programs like BiocFileCache. We have added the following explanation to the vignette under the title, "Updating maps of aliased gene symbols":

We intentionally avoid automatic update of the map to maintain reproducibility, because the same code from the same version of HGNChelper could produce different results at any time with automatic map update.

Competing Interests: No competing interests were disclosed.