



SOFTWARE TOOL ARTICLE

REVISED Identifier Mapping in Cytoscape [version 2; referees: 3 approved]

Adam Treister , Alexander R. Pico

Institute of Data Science and Biotechnology, Gladstone Institutes, San Francisco, CA, 94158, USA

v2 First published: 11 Jun 2018, 7:725 (doi: [10.12688/f1000research.14807.1](https://doi.org/10.12688/f1000research.14807.1))
 Latest published: 06 Aug 2018, 7:725 (doi: [10.12688/f1000research.14807.2](https://doi.org/10.12688/f1000research.14807.2))

Abstract

Identifier Mapping, the association of terms across disparate taxonomies and databases, is a common hurdle in bioinformatics workflows. The *idmapper* app for Cytoscape simplifies identifier mapping for genes and proteins in the context of common biological networks. This app provides a unified interface to different identifier resources accessible through a right-click on the table's column header. It also provides an OSGi programming interface via *Cytoscape Commands* and *CyREST* that can be utilized for identifier mapping in scripts and other Cytoscape apps, and supports integrated Swagger documentation.

Keywords

Cytoscape, ID Mapping, Identifiers, BridgeDb



This article is included in the [Cytoscape Apps gateway](#).

Open Peer Review

Referee Status:

	Invited Referees		
	1	2	3
REVISED			
version 2 published 06 Aug 2018	report		report
version 1 published 11 Jun 2018	?	report	?

- 1 **Ruth Isserlin** , University of Toronto, Canada
- 2 **Augustin Luna** , Dana-Farber Cancer Institute, USA
- 3 **Nadezhda T. Doncheva**, University of Copenhagen, Denmark

Discuss this article

Comments (0)

Corresponding author: Alexander R. Pico (apico@gladstone.ucsf.edu)

Author roles: **Treister A:** Methodology, Software, Writing – Original Draft Preparation; **Pico AR:** Conceptualization, Funding Acquisition, Methodology, Project Administration, Resources, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: We would like to acknowledge funding from National Institute of General Medical Sciences [P41GM103504 (ARP, AT)]
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2018 Treister A and Pico AR. This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

How to cite this article: Treister A and Pico AR. **Identifier Mapping in Cytoscape [version 2; referees: 3 approved]** *F1000Research* 2018, **7**:725 (doi: [10.12688/f1000research.14807.2](https://doi.org/10.12688/f1000research.14807.2))

First published: 11 Jun 2018, **7**:725 (doi: [10.12688/f1000research.14807.1](https://doi.org/10.12688/f1000research.14807.1))

REVISED Amendments from Version 1

The following reviewer comments are addressed in this version:

- * Clarification about relationship to and reliance upon BridgeDb project app, databases and web services.
- * Updates to [Table 1](#) and caption
- * Clarification of persistent selection behavior in GUI
- * Added Use Case 3: Identifiers and symbols
- * Explanation of "force single"
- * Example of R code with and without the custom function"
- * Clarification on how regular expressions are used for data source inference
- * Consistent references to "Uniprot-TrEMBL"
- * Described how results are added to Table Panel
- * Changed "singular" to "single"
- * Updated documentation on available species

See referee reports

Introduction

Cytoscape is an integrated network visualization tool and analysis platform^{1,2}. Within its common workflows, identifier mapping remains a challenge when working with biological data from different sources. This problem has been addressed by the BridgeDB project³, which created clients and services to translate between various identifiers. The original BridgeDb app⁴ for Cytoscape was written to provide an exhaustive set of functions to match the full capabilities of BridgeDb. Though this provided the needed functionality, its basic usage was unnecessarily complex. The idmapper app is a useful alternative, providing access to a commonly used subset of BridgedDb databases via web services by means of a simplified interface bundled into Cytoscape. Now, without any installation or configuration, Cytoscape users can right-click on a table header to map that column's data to a different namespace ([Figure 1](#)). Although, the breadth of coverage is smaller than the full-featured BridgeDb app, it still covers over a dozen identifier data sources maintained by BridgeDb, including Ensembl, Entrez Gene, HGNC, KEGG, Uniprot-TrEMBL and various species-specific sources. Because idmapper supports Cytoscape's new CyREST interface, identifier mapping can be included in scripted workflows, and driven from R or python programs.

Implementation

Inferring the data source

From within Cytoscape, a user initiates an ID mapping operation by right-clicking on the header of a column containing identifiers in the *Table Panel*. Based on the specified species a list of data sources is provided to the user. In the most common cases the type of identifier can be guessed by idmapper based on its format and is presented as the default selection. [Table 1](#) shows the supported data sources and example identifier formats.

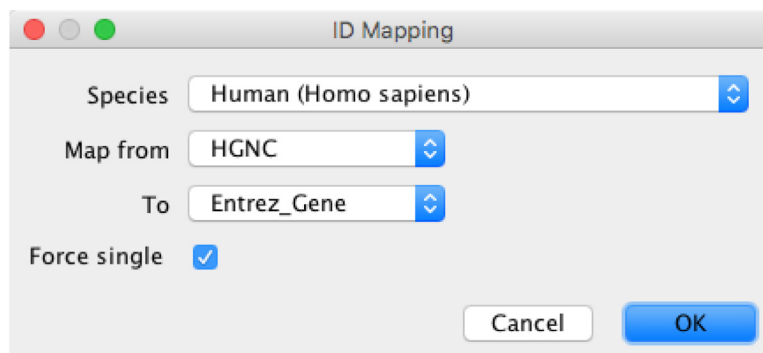


Figure 1. Simplified dialog for ID Mapping. Four options are presented to the user when accessing idmapper from within the Cytoscape GUI, each with common default or inferred values to reduce the number of steps required of the user.

Table 1. Supported Data Sources. The parameter names of supported data sources, their species exclusivity and an example identifier. Note that Ensembl support is only for gene identifiers, not proteins.

Data Source	Species	Example
Ensembl	Any	ENSG00000139618
Entrez Gene	Any	11234
KEGG Genes	Any	syn:ssr3451
UniGene	Any	Hs.553708
Uniprot-TrEMBL	Any	P62158
FlyBase	Drosophila melanogaster	FBgn0011293
HGNC	Homo sapiens	DAPK1
MGI	Mus musculus	MGI:2442292
RGD	Rattus norvegicus	2018660
SGD	Saccharomyces cerevisiae	S000028457
TAIR	Arabidopsis thaliana	AT1G01030
WormBase	Caenorhabditis elegans	WBGene00000001
ZFIN	Danio rerio	ZDB-GENE-041118-11

The app looks at the first ten entries and chooses the source that matches corresponding regular expressions provided by BridgeDb. If there is no match (or if more than one system is matched), then it simply chooses first option in the list as the default selection.

Cytoscape tasks

There are two different tasks supported by the idmapper app. *ColumnMappingTask* is activated by the right-click mouse event on a table header. It infers the current table and column from the information that comes from the mouse event, triggering a dialog (see GUI use case) that collects the information needed to make a call to BridgeDb web services. Please refer to the BridgeDb project for details about their services and sources³. In order to support automation, we added *MapColumnCommandTask* as an analog that is exposed specifically for *Commands* and *CyREST* access. These tasks eventually result in the same algorithms being invoked.

Use cases

Cytoscape graphical user interface (GUI)

The idmapper app provides the same basic functionality of the BridgeDb app with less fuss. Users do not have to install it, launch it, make configuration decisions or think about which database they are accessing. The app comes bundled with every Cytoscape release. As such its usage in Cytoscape via the interactive GUI (graphical user interface) is documented in the Cytoscape manual: http://manual.cytoscape.org/en/stable/Node_and_Edge_Column_Data.html#mapping-identifiers.

To map an identifier from one source to another, right click on the column header of your identifier. Select the option to **Map Column** to bring up the idmapper dialog (Figure 1).

The idmapper dialog presents a few choices the user can override before performing ID mapping. The default *Species* is determined by the previous selection made by the user per network, providing a persistent behavior across multiple searches. The available choices for the identifier data sources are determined by the species. The *Map from* data source is automatically selected based on an inspection of the first ten identifiers found in the column clicked on by the user. This can be overridden by the pull down menu. The *To* data source must be selected by the user; Ensembl is presented by default. Finally, the *Force single* checkbox offers to simplify the results of ID mapping by ignoring one-to-many cases and only keeping the first result (arbitrarily determined by the BridgeDb web service result). If the option is off, a list of results will appear in the column. This can easily be overridden by clicking the toggled checkbox. The result of the mapping is appended to the node table in a column named after the target data source, e.g., “Ensembl”. If a column by that name already exists, a parenthesized number is appended to the name to ensure it is unique, e.g., “Ensembl(1)”.

Cytoscape command line interface

The command interface does not use the same tasks as the GUI. In the GUI use case, the app knows the current context of where the command was activated, i.e., the network, table and column. This information must explicitly be provided as parameters to the command interface to perform the same operation. Thus, in addition to species, mapFrom, mapTo and forceSingle, the command line operation of idmapper also requires networkName, table and columnName (see next section for more details).

Cytoscape automation

In the scripting environment, idmapper provides all of its functionality in a single call (Figure 2). This means that identifier mapping can be incorporated into Cytoscape automation workflows with a single additional command. The scripting version of the command includes extra parameter for **columnName**, **networkName** and **table**, which are implicit in the GUI version from the location of the mouse event.

The **map column** function takes the following parameters:

- **columnName** (string): Specifies the column name where the source identifiers are located
- **forceSingle** (string, optional): When multiple identifiers can be mapped from a single term, this forces a singular result
- **mapFrom** (string): Specifies the data source describing the existing identifiers
- **mapTo** (string): Specifies the data source identifiers to be returned as a result in a new column
- **networkName** (string, optional): Which network is used in the mapping.
- **species** (string): The common *or latin* name of the species to which the identifiers apply, e.g., Human, *Homo sapiens*, Mouse, *Mus musculus*, Rat, *Rattus norvegicus*, Frog, *Xenopus tropicalis*, Zebra fish, *Danio rerio*, Fruit fly, *Drosophila melanogaster*, Mosquito, *Anopheles gambiae*, Arabidopsis, *Arabidopsis thaliana*, Yeast, *Saccharomyces cerevisiae*, E. coli, *Escherichia coli*, Tuberculosis, *Mycobacterium tuberculosis*, Worm, *Caenorhabditis elegans*
- **table** (string, optional): Which table is used as the source of the identifiers, e.g., "node" for the default node table

With Cytoscape running, the **map column** function can be called from any scripting environment or programming language that supports REST calls. In the case of R and Python scripts, there are dedicated packages to make this even easier. The *RCy3* package wraps this command in an R function called **mapTableColumn** to conform to other table functions (<https://www.bioconductor.org/packages/release/bioc/html/RCy3.html>). The *py2cytoscape* library similarly provides this command as a python function, **cyclient.idmapper.map_column** (<https://github.com/cytoscape/py2cytoscape>). The advantage of using one of these dedicated packages is having more concise syntax and language-specific conventions. In *RCy3*, for example, the custom **mapTableColumn** function simplifies the call, conforms to other *RCy3* functions and returns a dataframe with the map.from and map.to columns, while the generic **commandsPOST** function relies on the composition of a command string using the idmapper parameters defined in Figure 2:

```
(RCy3 generic): commandsPOST(paste('idmapper map column,
                                   columnName="name", forceSingle="true",
                                   mapFrom="Ensembl", mapTo="Entrez Gene",
                                   species="Human", table="node", sep=" '))
```

```
(RCy3 custom): mapTableColumn (column="name", species="Human",
                               map.from="Ensembl", map.to="Entrez Gene")
```

A sample script demonstrates how to map identifiers via *RCy3*, covering the most common use cases (<https://github.com/cytoscape/RCy3/blob/master/vignettes/Identifier-mapping.Rmd>).



Figure 2. Swagger documented function. The functionality of idmapper is contained in this single function: *map column*.

Case 1: Species-specific considerations

The Yeast Perturbation sample network provided with Cytoscape can be loaded from the Starter Panel and provides gene identifiers of the form “YDL194W”. These are actually Ensembl-supported identifiers for Yeast, distinct from the typical “ENSXXYG00000123456” form as presented in [Table 1](#). This presents a special case that users will need to be aware of when selecting species and source database or *mapFrom* in the GUI. (Ensembl has special cases for Yeast, Worm and Fly identifiers in addition to the standard terms that start with ENS.) In terms of automation, you could generate a new column of Entrez Gene IDs in this network with these calls:

```
(RCy3): mapTableColumn(column="name", species="Yeast",
                       map.from="Ensembl", map.to="Entrez Gene")

(py2cytoscape): cyclient.idmapper.map_column(source_column="name", species="Yeast",
                                             source_selection="Ensembl", target_selection="Entrez Gene")
```

Case 2: From proteins to genes

When working with protein interaction networks, for example those from the STRING database (see <http://apps.cytoscape.org/apps/stringapp>), you may want to translate protein identifiers (e.g., Uniprot-TrEMBL) to gene identifiers. The idmapper app supports this case as well, but one should be aware of the assumptions involved when making this translation. Since most genes encode for many proteins, you may have many-to-one mappings in your results. For all human networks imported from STRING using the *StringApp*⁵, the following commands will perform an ID mapping from Uniprot-TrEMBL (proteins) to Ensembl (genes):

```
(RCy3): mapTableColumn(column="canonical name", species="Human",
                       map.from="Uniprot-TrEMBL", map.to="Ensembl")

(py2cytoscape): cyclient.idmapper.map_column(source_column="canonical name",
                                             species="Human", source_selection="Uniprot-TrEMBL",
                                             target_selection="Ensembl")
```

Case 3: Identifiers and symbols

In contrast to gene names and symbols, identifiers provide a more reliable means of specifying a particular gene. All data integration should be performed using identifiers as keys. Nevertheless, names and symbols play an important role in making results easier to read and understand. The idmapper app is primarily concerned with identifiers. However, relying on a subset of commonly used sources from BridgeDb ([Table 1](#)) it does provide one exception. HGNC symbols, when used properly, can serve as identifiers in ID mapping and more generally can be added when starting from any other human ID source:

```
(RCy3): mapTableColumn(column="canonical name", species="Human",
                       map.from="Ensembl", map.to="HGNC")

(py2cytoscape): cyclient.idmapper.map_column(source_column="canonical name",
                                             species="Human", source_selection="Ensembl",
                                             target_selection="HGNC")
```

Limitations

The idmapper app provides easy access to a critical subset of ID mapping functionality originally covered by the BridgeDb app. When users run into the limitations of idmapper, they still have the option of installing and using the full-featured BridgeDb app from <https://apps.cytoscape.org/apps/bridgedb>. Examples of limitations include support for additional species or data sources. The BridgeDb app includes more of both as well as means to access custom data sources.

Software availability

1. Software available from the Cytoscape App Store: <http://apps.cytoscape.org/apps/idmapper>
2. Latest source code: <https://github.com/cytoscape/idmapper>
3. Archived source code as at the time of publication: <https://doi.org/10.5281/zenodo.1246814>⁶
4. License: Apache License, Version 2.0: <http://www.apache.org/licenses/LICENSE-2.0.html>

Author information

AT and ARP participated in the design of the described software. AT implemented the software. AT and ARP contributed to the writing of this article.

Competing interests

No competing interests were disclosed.

Grant information

We would like to acknowledge funding from National Institute of General Medical Sciences [P41GM103504 (ARP, AT)].

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgments

Jianjiong Gao and Chao Zhang for their work on the original BridgeDb app (<https://f1000research.com/articles/3-148/v1>). Nuno Nunes for his work on the BridgeDb web service.

References

1. Cline MS, Smoot M, Cerami E, *et al.*: **Integration of biological networks and gene expression data using Cytoscape**. *Nat Protoc*. 2007; **2**(10): 2366–2382.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Shannon P, Markiel A, Ozier O, *et al.*: **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome Res*. 2003; **13**(11): 2498–2504.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
3. van Iersel MP, Pico AR, Kelder T, *et al.*: **The BridgeDb framework: standardized access to gene, protein and metabolite identifier mapping services**. *BMC Bioinformatics*. 2010; **11**: 5.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Gao J, Zhang C, van Iersel M, *et al.*: **BridgeDb app: unifying identifier mapping services for Cytoscape [version 1; referees: 2 approved]**. *F1000Res*. 2014; **3**: 148.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
5. Szklarczyk D, Morris JH, Cook H, *et al.*: **The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible**. *Nucleic Acids Res*. 2017; **45**(D1): D362–D368.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. Treister A, Ono K, Zmasek C, *et al.*: **cytoscape/idmapper: 3.6.3 (Version 3.6.3)**. *Zenodo*. 2018.
<http://www.doi.org/10.5281/zenodo.1246814>

Open Peer Review

Current Referee Status:   

Version 2

Referee Report 08 August 2018

doi:[10.5256/f1000research.16859.r36844](https://doi.org/10.5256/f1000research.16859.r36844)



Nadezhda T. Doncheva

Novo Nordisk Foundation Center for Protein Research & Center for non-coding RNA in Technology and Health, University of Copenhagen, Copenhagen, Denmark

The idmapper is a simple, but very useful app for Cytoscape that significantly enhances the functionality of Cytoscape for users of the GUI and the CyREST interface. My concerns about the article have been addressed by the authors and I am very glad to hereby approve the revised version.

Competing Interests: No competing interests were disclosed.

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

Referee Report 07 August 2018

doi:[10.5256/f1000research.16859.r36842](https://doi.org/10.5256/f1000research.16859.r36842)



Ruth Isserlin 

Donnelly Centre for Cellular and Biomolecular Research, Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, ON, Canada

This version addressed all my concerns. The idmapper app will be a very user friendly, and useful addition to Cytoscape

Competing Interests: No competing interests were disclosed.

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

Version 1

Referee Report 19 July 2018

doi:[10.5256/f1000research.16116.r34907](https://doi.org/10.5256/f1000research.16116.r34907)

**Nadezhda T. Doncheva**

Novo Nordisk Foundation Center for Protein Research & Center for non-coding RNA in Technology and Health, University of Copenhagen, Copenhagen, Denmark

The paper describes idmapper, a simple yet very useful app for converting node identifiers from one data source to another. It is provided as part of the widely used network analysis and visualization software Cytoscape and the mapping functionality can be used by users in several different ways, including the GUI, Cytoscape commands, and from R or Python scripts via the Cytoscape REST interface. It supports common data sources such as UniProt and Ensembl. Both the app and the manuscript are in good quality and there are a few minor aspects to be addressed.

- In the Implementation section, it is important to clarify which extent the idmapper app relies on the BridgeDB mappings/app and also say a few words about how these mappings are done in BridgeDB and what the versions of the used data sources are. Maybe it can be a subsection called Dependencies or Backend?
- In the use Cases section, it would be useful to provide the Cytoscape command call in addition to the R and python function calls.
- Figure 2 is not very informative as it is now. Maybe it would be better if it includes a screenshot of the "Example Value" code? In addition, the term "singular function" is used in mathematics and might be confusing for some readers.

The app works nicely both through the Cytoscape GUI and the command/swagger interface. However, it would be great if the documentation can be improved a little bit so it is more consistent and informative. In particular:

- Could a list of all possible species and all possible data sources names be provided both in the manuscript and in the documentation of the command in Cytoscape/Swagger? It could be included as part of the description of the map column function in the Cytoscape automation section.
- The species are most likely the ones listed in the species parameter description but it is clearly stated anywhere. Is there a comma missing in the sentence: "The combined common or latin name"? It seems that one can use all three of those: the common name, the latin name or the combined one (although there is a warning if anything but the combined one is used).
- Are the data sources exactly and only the ones in Table 1? There is an example with "Uniprot" (GUI, swagger documentation and example), one with "Uniprot-TrEMBL" (Case 2 in the paper) and in the table it is written "Uniprot TrEMBL". Are all of those the same or not and if not what are the differences?
- For the table, it says "node" in the command documentation and "default node" in the swagger example. Both of them work, but it would be good to have a list of possible values as it is the case for the species parameter.
- Make clear (maybe in table 1) that Ensembl refers to Ensembl Gene identifiers and not Proteins.
- Would it be possible that the parameters of the python function have the same names as the parameters in all other functions? Only recommended if it does not break already existing scripts.

There were two minor issues while testing the app:

- When running the following command from within Cytoscape or the swagger documentation on the galFiltered network (or on a STRING network), a warning and an error message come up in the Task Manager. It seems to work nonetheless, so could you check what is going on there?
- `idmapper map column columnName="name" mapFrom="Ensembl" mapTo="Entrez" species="Yeast"`
- warning: value not contained in list of possible values possible items = [] and error: networkTable not found.

- The data source inferring works well for the identifiers, but in the case that the species is wrong (e.g. switching from a yeast to a human network), changing the species resets the Map from column to the first entry in the list.

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?

Partly

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: No competing interests were disclosed.

I have read this submission. I 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 22 Jul 2018

Alexander Pico, The Gladstone Institutes, UCSF, USA

Nadya, thank you for the review. We have prepared a version 2 of the paper that addresses your main comments and corrections. A couple additional notes that are not reflected in the updated text:

- The R and Python library projects are independently developed, so coordinating on parameter names is impossible and in many cases not desired since consistency within each is much more important than across them, i.e., most users will pick one or the other and want maximum consistency with the conventions of that language, etc.
- Figure 2 is a requirement for all Cytoscape Automation papers. Even for cases like ours where there is just one operation. It's our fault for making such a simple app!
- Thanks for the bug reports. We will try to reproduce and fix these for future releases of the app.

Competing Interests: No competing interests were disclosed.

Referee Report 03 July 2018

doi:10.5256/f1000research.16116.r34906



Augustin Luna 

cBio Center, Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA, USA

This Cytoscape app provides functionality that is widely useful for Cytoscape users for converting network identifiers to different databases. Some points not completely clear:

1. The list of regular expressions used for inference, do they come from identifiers.org? What happens if the inference fails? Does the app try to pick the closest matching regular expression?
2. For use of RCy3, is RCy3 a generic package to interact with any REST function in any Cytoscape package? Or did the developers of RCy3 specially include the mapTableColumn function to access the idmapper app?

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: No competing interests were disclosed.

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

Author Response 03 Jul 2018

Alexander Pico, The Gladstone Institutes, UCSF, USA

Augustin, thanks for the review.

1. The regular expressions come from BridgeDb (which in turn gets them from identifiers.org). If there isn't a match to the regular expressions (or if more than one system is matched), then it just picks the first option in the list. A few of the system types aren't well specified. It's a simple matter to override this in the UI. We will add a sentence or two in the next version of the paper in response to all reviewers.

2. Right. RCy3 supports both a generic function call and a specific `mapTableColumn` function call. Since this is a "core" app, the RCy3 package supports custom convenience functions to make this operation easier to use and better documented. We will add a more detailed description and contrasting example in the next version of the paper.

Competing Interests: No competing interests were disclosed.

Referee Report 12 June 2018

doi:10.5256/f1000research.16116.r34908



Ruth Isserlin 

Donnelly Centre for Cellular and Biomolecular Research, Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, ON, Canada

The paper entitled "**Identifier Mapping in Cytoscape: idmapper**" by Adam Treister and Alexander Pico presents a new implementation of id mapping available directly through Cytoscape with no additional configuration.

There is a lot of discussion throughout the paper about BridgeDB. It is not clear what the relationship between idmapper and BridgeDB is. It is understood that idmapper is a simplification and alternative of the aforementioned tool but does idmapper rely on BridgeDB? Do they share a codebase?

One thing that might be a good addition to the implementation section is a short discussion on the backend. Where are the mappings coming from? Is the app dependent on any online resources or are the mappings static and stored within the Cytoscape instance? This is related to the previous BridgeDB question as given that there is no discussion about the backend I thought maybe the info was coming from BridgeDB and that was the reason it was included.

In the implementation section, Table 1,

- it might be useful to have a column specifying the data source names that the app recognizes. (for example in the table one of the data sources listed is "Uniprot TrEMBL" and later in Case 2 in the command the `map.from(RCy3)` and `source_selection(py2cytoscpae)` the db is specified as "Uniprot-TrEMBL" (with a dash)).
- Given that this is the implementation section it might be good to list the regular expressions that each identifier recognizes (provided that they aren't too messy) and any exceptions they look for.
- Later on in the case 1 you mention an exception to the basic regular expression behaviour for yeast. Are there more exceptions that the app handles?
- It is also unclear why the "Code" column is required.
- Might be nice to separate the data sources that can handle any species and those that are species specific.

In the use cases section "The default **Species** is determined by the previous selection made per network, providing a "smart and sticky" behavior. " It is unclear but the previous selection was?

In the use cases section in the specific cases two example use cases are presented, species specific and protein to gene conversions. It would be helpful to list these and other common use cases at the start of the use case section as well. One of the most common use cases being going from non-descriptive

identifiers (like entrez gene ids, and ensembl ids) to something more understandable such as species specific IDs (HGNC or MGD, Is it possible to map to proper gene symbols?)

In the use case section it is stated that if the ID maps to multiple identifiers there is an option, "*Force single*", that when selected the app selects the first result. How are the returned IDs sorted? Is the first match the "best" match, alphabetical, random?

In the Cytoscape automation section in the parameters section for the species option all the available mappings are listed but for the mapFrom and mapTo no options are listed. (if the recognized data source name is add to Table 1 you can just reference the table here or if the first column of Table 1 are the recognized names it would be good to reference it here). Also, the parameters listed for Cytoscape automation section are very different from the parameters used in the use cases which can be very confusing. Maybe adding an example using the RCy3 commandsGet option under RCy3 and py2cytoscape examples just showing how the user can use all the parameters as specified using the command directly.

Minor comments/questions:

- In the introduction "Uniprot and various species-specific sources" should be "Uniprot and various species-specific sources"
- In the implementation section "The app looks at the first ten entries and choose the source" would be better as "The app looks at the first ten entries and chooses the source"
- In the implementation section - "This number of identifiers iteratively sampled is set by a static variable called *N_Iterations*. The algorithm for inferring the data source is implemented in *IdGuess.java*." - This is a little confusing why this is needed. Is this a parameter the user can control or tweak?
- In Use cases section - "As such it usage in Cytoscape via" should be "As such its usage in Cytoscape via"
- In use case 2 "you may want to translate to gene identifiers" might be better as "you may want to translate **protein identifiers (for example: Uniprot-TrEMBL)** to gene identifiers"
- Can idmapper convert a list column? (in the example use case where the network is an enrichment map and each node contains a set of genes as opposed to each node being a gene)
- What is the resulting column name? What if a column with that name already exists?

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?

Partly

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: No competing interests were disclosed.

I have read this submission. I 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 26 Jun 2018

Alexander Pico, The Gladstone Institutes, UCSF, USA

Ruth, thank you so much for your thorough review. These clarifications, fixes and additions have greatly improved the article. Version 2 should be released soon, addressing all the issues you raised. We decided not to include the regular expressions in Table 1, however. They are messy and are what you'd expect from the example identifiers provided, which we feel do a better job of communicating what to expect from each data source.

We hope you'll have a chance to look over version 2 and find that it meets your expectations.

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research