Dictionary of disease ontologies

true true true

Abstract

To add

R version: R version 3.6.0 (2019-04-26)

Bioconductor version: 3.10

Package: 1.0.0

Introduction

Disease ontologies have been developed to meet the need to structure, classify, and describe diseases [Gruber, 1993, Haendel et al., 2018, Hoehndorf et al., 2013]. As a result of the diversity in their usage a multitude of disease ontologies exist, aiming to facilitate the integration with drug information, transcriptomics and genomics information, etc. as well as to support development of novel treatments [Haendel et al., 2018, Hoehndorf et al., 2013, Rappaport et al., 2013]. Disease ontologies allow a more formal description of disease; however, each often defines an independent identifier and will only link to a subset of independent biological databases [Hasnain et al., 2014, Hoehndorf et al., 2013, Kibbe et al., 2015, Livingston et al., 2015, Malone et al., 2010, Rappaport et al., 2013]. While this stimulated the construction of integrated biological knowledge bases, the use of independent, ontology-specific identifiers, heterogeneous decisions on disease definitions, and the inherent presence of errors complicates integrating disease ontologies [Livingston et al., 2015, Rappaport et al., 2013]. In addition, the navigation of these large integrated knowledgebase with often an inheritely complicated data model, is difficult for most, non-expert users [Hasnain et al., 2014, Hu et al., 2017, Livingston et al., 2015].

Several efforts have been made to connect the different disease ontologies themselves by generating of a single new integrative ontology [Mungall et al., 2017, Shefchek et al., 2019, Rappaport et al., 2013]. Using semantic similarity the Monarch Disease Ontology (MonDO) aggregates different sources including OMIM, Orphanet, NCiT, GARD, DO, and MF [Mungall et al., 2017, Shefchek et al., 2019]. Other examples is the Disease Ontology (DO) resource which aims to standardize disease descriptions and classification from a clinical perspective using equivalence mappings [Cheng et al., 2013, Schriml and Mitraka, 2015, Yu et al., 2015]. The Experimental Factor Ontology (EFO) also establishes an unified ontology (not limited to diseases) by re-using several reference ontologies that lie within its scope and enriches these classess with additional axioms when needed [Malone et al., 2010]. Currently, it combines information from OMIM, Orphanet, ICD9/10 and SNOMEDCT, HPO, UBERON, and MonDO [**?**].

Despite ongoing efforts, two issues remain to use disease ontologies efficiently: the issue of completeness and ease of access. To this end, the Dictionary of Disease Ontologies (DODO) was developed. The first issue of completeness concerns the exhaustiveness of disease cross-reference mappings [Hu et al., 2017, Rappaport et al., 2013]. While efforts such as the Monarch Initiative and EFO try to integrate different disease ontologies through semantic learning and manual curation, these resources, like the different disease ontologies themselves, are currently not providing a complete mapping across different disease ontologies. By combining the information provided by the different ontologies, these cross-reference mappings can be enriched. It also allows connecting ontologies to each other that have no direct cross-reference mapping between them by indirectly inferring relationships. In addition, the existing efforts for integration are not flexible to extend easily to proprietary disease ontologies. Another challenge is the availability of an efficient and straightforward manner to access disease information through well established bioinformatics platforms (R or python) [Rappaport et al., 2013, Saqi et al., 2018]. If avialble, this will facilitate a more flexible connection to the different life science resources to create a more complete disease landscape. Currently, the programmatic access provided by many ontologies often requires expertise in creating SPARQL queries and a high level of understanding of the underlying databases or data model to be able to generate more complex queries [Hasnain et al., 2014, Hu et al., 2017, Rappaport et al., 2013]. The presented graph database is accompagnied by a R package that allows easy access, exploration, and definition of disease concepts of interest. It can work as the intermediate player to facilitate access and exhaustive extraction of information from other life science databases without the need to harmonize these up front. In this paper we will present DODO graphical database and R package with use cases.

Methods

Implementation

For software tool papers, this section should address how the tool works and any relevant technical details required for implementation of the tool by other developers.

In this section, an overview is presented of the DODO graphQL+ database and the accompagnying R package.

Data model

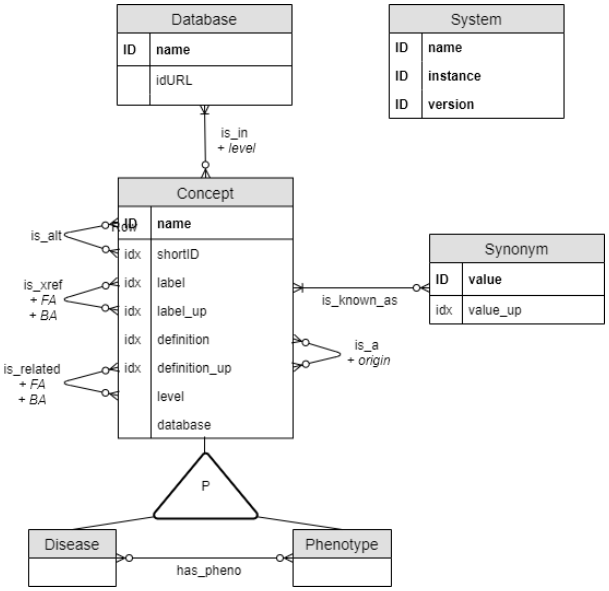

Figure 1: The DODO graph model is shown as an Entity/Relationship (ER) diagram. It consist of four types of entities (concept, disease, phenotype, database) corresponding to graph nodes. Several relationships between the nodes are described referring to graph edges. 'ID' refers to an unique entity while 'idx' indicates whether this entity is indexed.

The data model underlying DODO aims to capture the relationship between disease and phenotypes as described across different databases 1. It relies on four types of nodes (concept, disease, phenotype, and database) each with specific proporties. A disease or phenotype nodes have the same properties with *name* as their primary property. *Name* is the full length identifier of a disease or phenotype, namely a concatenation of database and identifier, e.g. "MONDO:0005027". Additional properties are (when available) a unique (canonical) *label*, disease *definition*, *database*, and node *type*. In addition, *label_up* and *definition_up* are the uppercase version of *label* and *definition* respectively. Each node with class disease or phenotype is also encoded witht the class "Concept". Synonyms of the nodes are available through the *is_known_as* relationship with *value* and *value_up* containing the (uppercase) synonyms. A database node has only two property values, its *name* (the name of the database) and url link (*idURL*).

Nodes can be related to each other through different relations based on information provided by the resources. A disease/phenotype/concept node can be related to another node belonging to a different database by the *is_xref* or *is_related* relationship. This relationship indicates nodes relating to the same or highly similar disease concepts (as defined by the original resources). The relationship is directional and has the property *FA* (forward ambiguity) and *BA* (backward ambiguity). As disease definitions can be implemented in a broad or narrow sense by each disease ontology, *is_xref* and *is_related* relationships across ontologies are not always unambiguous. The concept of forward and backward ambiguity is implemented to handle transitivity mapping (see below). The *is_xref* and *is_related* relationship is dependent on the direction one is traversing through and is therefore encoded twice between each node. Each of these directions has its subsequent forward and backward ambiguity information provided as properties.

Phenotype/disease/concept nodes may also be related through *is_a* directional hierarchical relationship identifying parent/child nodes based on an ontological tree. This relationship is only available between nodes of the same database with the exception of diseases defined by EFO. EFO combines the re-use of identifiers from external resources and enriches this information with additional internal disease classifiers to construct its ontology [Malone et al., 2010]. To distinguish the origin of the *is_a* relationship, the property *origin* is a added.

Table 1: Different disease ontologies included into DODO database and link to GitHub repository.

Disease ontology	GitHub
Monarch Disease Ontology (MonDO)	NA
Experimental Factor Ontology (EFO)	NA
Orphanet	NA
MedGen	NA
Medical Subject Headings (MeSH)	NA
Human Phenotype Ontology (HPO)	NA
ClinVar	NA
Disease Ontology (DO)	NA
International Classification of Diseases (ICD10)	NA

Phenotypes are highly detailed descriptions of clinical abnormalities which are used to describe disease through *has_pheno* non-directional relationship. And finally, each disease or phenotype node belongs to a database as encoded by the *is_in* relationship. This relationship is assigned the property (hierarchical) *level* capturing the highest position a node has in the ontology tree.

Feeding the database

To construct a DODO instance, a set of script is available to load and feed a Neo4j instance. These are not exposed directly to the user instead, these scripts are available in the *build/scripts* folder. The feeding of DODO is based on the parsed files of the different ontologies, a workflow on downloading and parsing for each included ontology is available through GitHub (Table 1).

The different steps are briefly described below:

1. Creating the relationship tables based on the information from the different resources
2. Creating a new DODO instance and importing the relationship tables
3. Compiling the instance into a Dgraph image
4. Start the new instance

Availability

The DODO instance build using the workflow described above is provided as a Docker image [**?**] here. This instance is build on information from the following disease ontologies listed in (Table 1.

S3 object

The center object used through the DODO R package is the disease network or disNet S3 object. It captures all information (disease node information, hierarchical information, phenotype information, and cross-reference informatino) around a disease and is structured as shown in Table 2

In addition, a *setDisNet* S3 object is also available which consists of a list of *disNet* objects. Figure @ref{fig:disNet} shows an example disNet object for epilepsy.

Operation

The data model is implemented using the Neo4j graph database which using the Cypher query language [**?**]. One accompagnying R package *DODO* was developed to connect and query the resource. It provides higher level functions to query the Neo4j graph database based on the described data model (above) [R Core Team, 2019].

The minimal system requirements are:

- R $\geq$ 3.6
- Operating system: Linux, macOS, Windows
- Memory $\geq$ 4GB RAM

The graph database has been implemented with Neo4j 3.4.9 [**?**], the DODO R package uses the following packages:

Table 2: The center object used through the DODO R package is the disease network or disNet S3 object. It captures all information (disease node information, hierarchical information, phenotype information, and cross-reference informatino) around a disease.

Part	Content
nodes	
id	disease ids (database:shortID)
database	disease databases
shortID	disease short identifiers
label	disease labels
definition	disease descriptions
level	maximum level the identifier holds in the hierarchical ontology tree
type	type of node (disease or phenotype)
synonyms	
id	disease ids
synonym	disease synonyms
children	
parent	parent disease ids
child	child disease ids
origin	ontology of origin where the parent/child relationship is recorded
xref	
from	disease 1 ids
to	disease 2 ids
ur	unique cross-reference identifier
forwardAmbiguity	number of cross-references between disease 1 and database 2
backwardAmbiguity	number of cross-references between disease 2 and database 1
alt	
type	type of cross-reference edge (is_xref or is_related)
id	current identifier
pheno	
alt	deprecated identifier
disease	disease identifier
seed	
phenotype	phenotype identifier
seed	vector of disease ids used to seed the disNet

Table 3: Summary of the available functions in DODO with a short description and identification of their scope.

Function	Description	scope
build_disNet	Building a network of disease identifiers	Build and interact
extend_disNet	Extending a disNet by different edges	Build and interact
filter_by_id	Filtering a disNet by id	Build and interact
filter_by_database	Filtering a disNet by database	Build and interact
focus_disNet	Focus on identifiers of interest and its neighbors	Build and interact
cluster_disNet	Clustering a disNet, generates a setDisNet	Build and interact
setdiff_disNet	Substract one disNet from another	Build and interact
split_disNet	Split a disNet based on a list of identifiers into a setDisNet	Build and interact
explore_disNet	Visualizes a datatable to explore a disNet	Visualize
show_relations	Visualizes cross-reference relationships for the provided identifier	Visualize
plot.disNet	Visualizes a disNet using visNetwork	Visualize
convert_concept	Convert the provided set of identifiers to another ontology or between concepts	Conversion
get_related	Convert function for ontologies separated by *is_related* edges mainly	Conversion
check_dodo_connection	Check connection with DODO graphical database	Utility
connect_to_dodo	Establish connection with DODO graphical database	Utility
forget_dodo_connection	Forget a saved connection to DODO	Utility
list_dodo_connections	List all saved connections to DODO	Utility
call_dodo	Calls a function on the DODO graphical database	Utility
show_dodo_model	Return DODO data model	Utility
get_version	Return DODO database version	Utility
get_concept_url	Returns concept url	Utility
list_database	Lists databases in DODO	Utility
list_node_type	Lists node type in DODO	Utility
get_ontology	Returns whole ontology	Utility
describe_concept	Returns concept description	Utility

- *dplyr* [Wickham et al., 2019]
- *tibble* [Müller and Wickham, 2019]
- *neo2R* [**?**]
- *rlist* [Ren, 2016]
- *stringr* [Wickham, 2019]
- *readr* [**?**]
- *visNetwork* [**?**]
- *shinythemes* [Chang, 2018]
- *DT* [Xie et al., 2019]
- *igraph* [Csardi and Nepusz, 2006]
- *shiny* [Chang et al., 2019]
- *BiocStyle*[**?**]

Querying the database

The DODO R package combines several functions to construct, interact, and explore such a disNet object. These will be briefly outlined in the sections below. These functions also split or cluster a disNet, generating a list of disNet object, captured by the S3 object setDisNet.

DODO R package provides function to allow four different scopes: building and interacting with a disNet of setDisNet object, visualizing a disNet, converting disease and phenotype concepts to different ontologies, and several utility functions to connect to DODO graph database or obtain low level information on identifiers. The Table 3 briefly list all function available within the package, as well as a short description and identification of the scope.

Transitivity mapping

As a consequence of the different way ontologies define disease (or phenotype) concepts, some cross-reference edges connect identifiers that are not exactly equal. Therefore, some cross-reference edges are trusted more than

Table 4: The EFO (and Orphanet) identifier for Coffin-Lowry syndrome (ORPHA:192) is cross-referenced to 21 different identifiers. Many of these cross-references are more or less equivalent to the original identifier (disease labels are only available when present in the parsed resources. Otherwise only a disease identifier is available). However, among its cross-references it also links to an ICD10 identifier (Q87.0) which is a broad term of 'Congenital malformation syndromes predominantly affecting facial appearance'. This identifier is highly ambiguous and links directly to 284 disease concepts.

Disease identifier	Disease label
ORPHA:192	Coffin-Lowry syndrome
MONDO:0010561	Coffin-Lowry syndrome
ClinVar:823	Coffin-Lowry syndrome
MedGen:C0265252	Coffin-Lowry syndrome
DOID:3783	Coffin-Lowry syndrome
UMLS:C0265252	Coffin-Lowry syndrome
MeSH:D038921	Coffin-Lowry Syndrome
OMIM:303600	COFFIN-LOWRY SYNDROME
ICD10:Q87.0	Congenital malformation syndromes predominantly affecting facial appearance
MeSH:C536435	NA
SNOMEDCT:15182000	NA
ICD9:759.89	NA
NCIt:C84643	NA
GARD:0008589	NA
GARD:0006123	NA

others. Ontologies such as MONDO or EFO use more narrow disease definitions than others like ICD10 or ICD9. If cross-reference edges are all considered equal without taking this distinction into account, it will result in the return of more distantly related concepts when traversing these edges. Table 4 shows an example of "Coffin-Lowry" syndrome (Orphanet identifier "192") for which most cross-reference identifiers are defined similarly. However, its cross-reference to ICD10 deals with a very broad term of "Conginetal malformation syndromes predominantly affecting facial appearance" ("ICD:Q87.0"). This identifier is highly ambiguous and has 284 different direct cross-referenced disease identifiers.

The concept of *ambiguity* is introduced to identify nodes that have many cross-references to the same database. Cross-reference edges are implemented in a directional manner in Neo4j, therefore both a forward (FA) and backward (BA) ambiguity is calculated and encoded on every direction. As it is often desired to move from a broader concept to a more narrow one, no filtering on forward ambiguity is put in place. However, the opposite, namely moving from a more narrow and through a broader concept is not always desired. This can result in an exponential increase of converted/expanded identifiers that are only distantly related to the original identifier.

In addition to the concept of *ambiguity*, two types of cross-reference edges encoded in DODO: *is_xref* and *is_related*. The *is_xref* edge is used for equal cross-reference relationships where the concepts relate more directly to each other (similar concept definitions). The *is_related* edge is used for all other cross-reference edges. These edges are based on the sum of forward and backward ambiguities between databases to quantify the symmetry between them and knowledge of how ontologies deal with disease concepts. Ontologies included for this quantification are: MoNDO, EFO, Orphanet, Orphanet, OMIM, MedGen, UMLS, MeSH, ICD10, ICD9, DOID, ClinVar, MEDDRA, GARD, SNOMEDct, and NCIt. The heatmap (Figure 2) shows the log_{10} transformed maximum total ambiguity between ontologies.

Figure 3 shows the number of conversions per identifier for MonDO when incrementally increasing the cutoff for the (total) ambiguity to define ontologies connected through *is_xref* and *is_related* edges. A very low cutoff would not return all cross-reference identifiers while a too high cutoff impact especially a few identifiers strongly with the return of many, lesser related, cross-reference identifiers.

By comparing the results and relationship between the different ontologies with a (total) ambiguity equal or lower than four are considered as *is_xref*. An exception is used for ICD10 and ICD9 which are never connected through an *is_xref* edge except between themselves. An additional exception is put in place for OMIM and Orphanet, which often define very narrow subtypes of diseases and therefore will be still encoded as an *is_xref* edge.

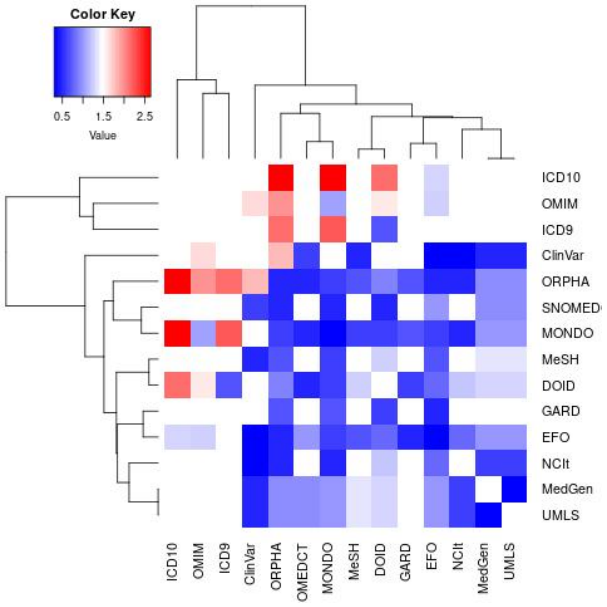

Figure 2: The heatmap shows the maximum value of total ambiguity between ontologies using a log_{10} transformation. While many ontologies are using concepts of similar level as identified by low total ambiguity, a few can be identified that are more ambiguous in their mappings. The 'optimal' ambiguity for a *is_xref* edges is determinged by comparing the number of conversions when incrementally increasing the cutoff of total ambiguity to defining a cross-reference edge between ontologie of either *is_xref* or *is_related* and knowledge of the way disease concepts are defined within ontologies.

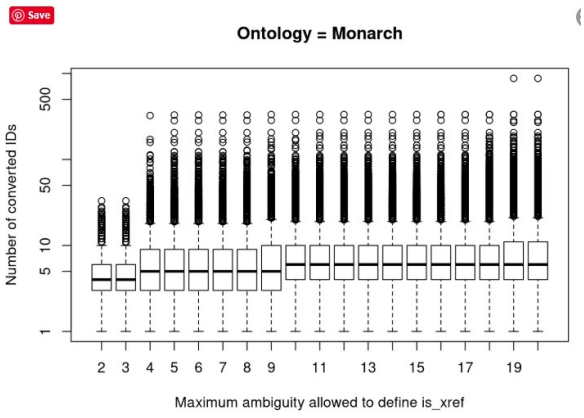

Figure 3: Number of conversions comparing incremental cutoffs in the (total) ambiguity to define *is_xref* and *is_related* cross-reference edges on an ontology scale (example of MonDO ontology).

Results

The table below shows the number of nodes available through each disease ontology, in total there are 454637 nodes (Figure @ref{fig:listDB}) in this DODO instance.

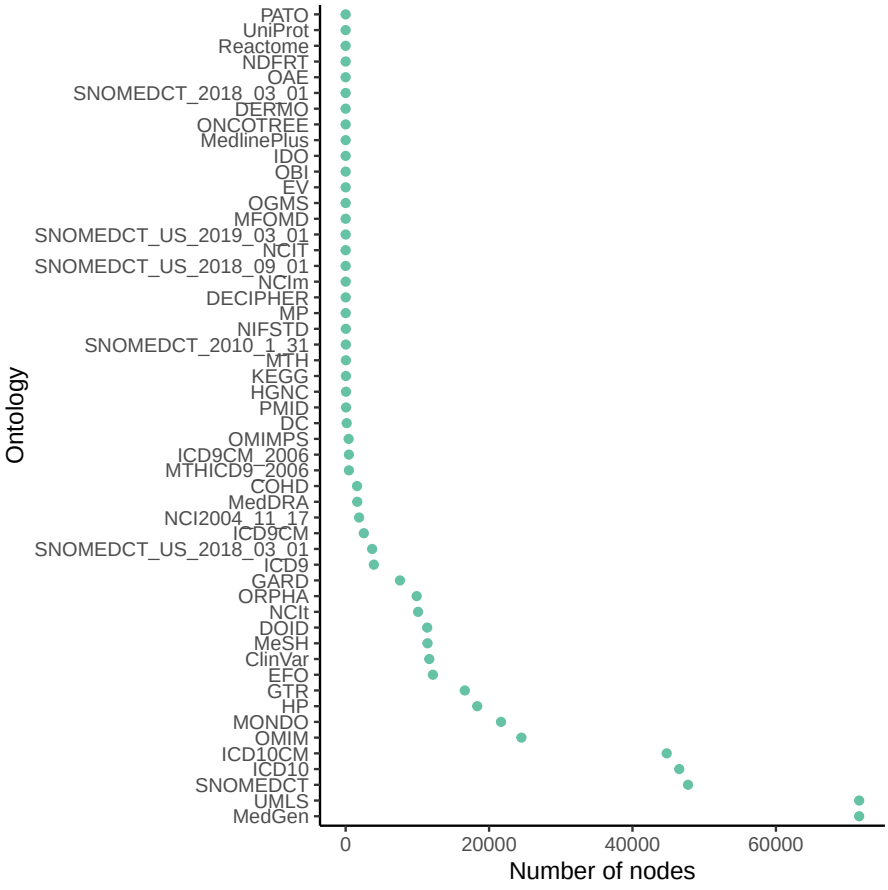

Figure 4: Overview of the number of nodes present for each disease ontology in DODO.

Use cases

Converting concepts

One of the basic functionalities of DODO is the ability to convert disease and phenotype identifiers. The conversion of identifiers generally performed in two steps based on the type of cross-reference edges and ambiguity values. Four separate use cases can be identified based on the parameters for converting within the same concept, e.g. converting one disease identifier to other disease ontologies.

A first and default conversion consists of two steps, a first using the transitivity of *is_xref* edges followed by an addition step using both types of cross-reference edges to obtain the indirect cross-references (parameters "step = NULL" and "intranstivity_ambiguity = NULL"). The default conversion can be used to get all identifiers around a disease concept whether broad or narrowly related or when converting from a more narrow concept to a broader concept. In general when the aim is to reach a broader concept related to the original identifiers but not move through it, it is recommended to put not filter on the *intransitive_ambiguity*. For transitivity mapping using *is_xref* edges only it is strongly recommended to use the default filtering on ambiguity by limiting (backward) ambiguity to one. The second step is a intransitive mapping where both *is_related* and *is_xref* cross-reference edges are traversed with one step to get the direct cross-reference identifiers. In Figure 5 an example is presented of the extension starting from the MonDO identifier "MONDO:0005027" encoding "Epilepsy". The transitive mapping with filtering on ambiguity moves between the *is_xref* mapping only in which a higher trust is placed. The final intransitive mapping step will return all nodes related through either an *is_related* or *is_xref* edge with no filtering as all ambigious relations want to be returned but not used when extending.

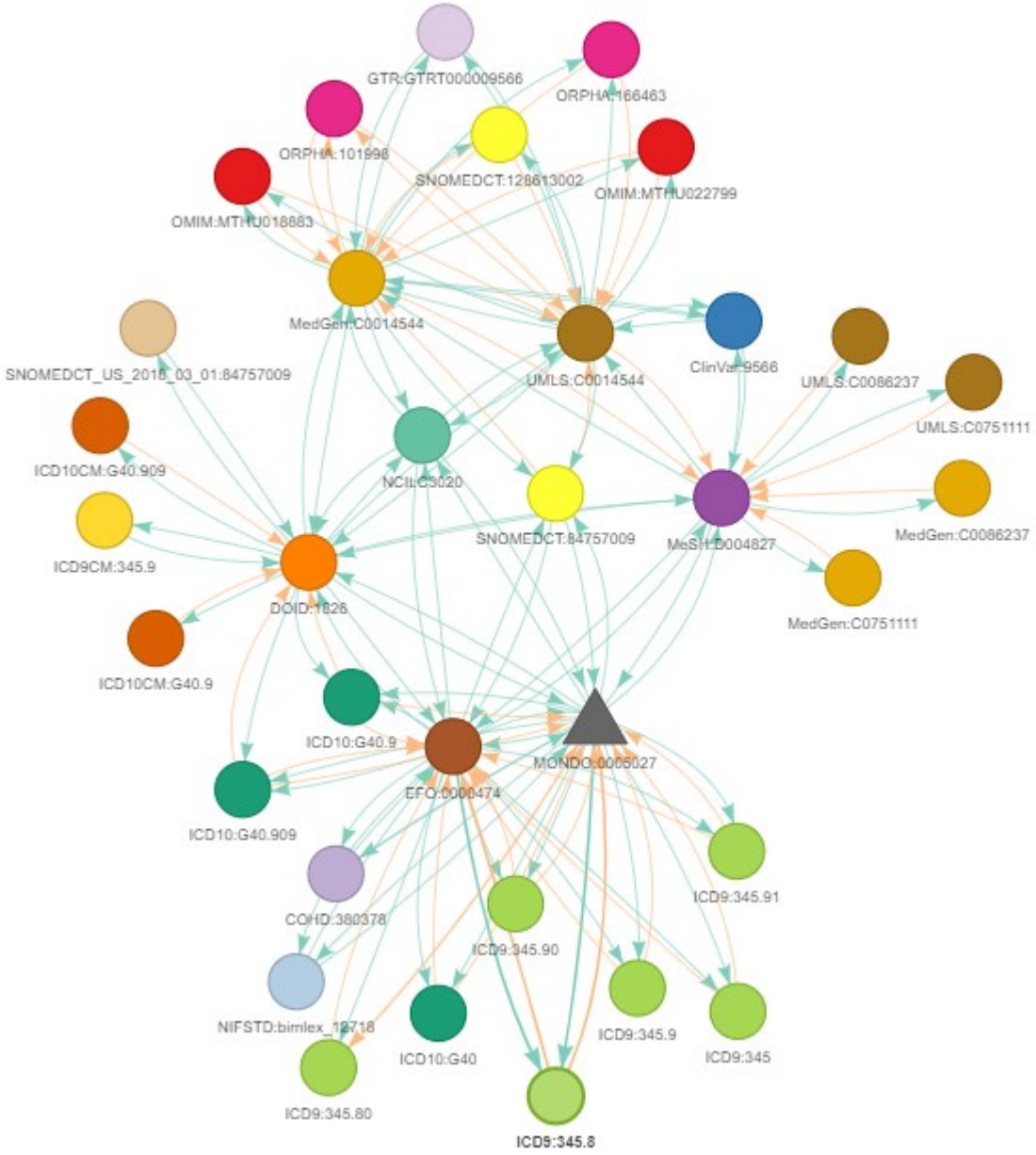

Figure 5: A disNet constructed and subsequently extended using only cross-reference relations using the MonDO identifier for 'Epilepsy' ('MONDO:0005027') as an example. This extension is performed in two phases. First, the transitive mapping uses the *is_xref* edges wich are trusted to a larger extend to obtain all relations with filtering on ambiguity (backward ambiguity = 1) (blue edges). The second steps uses no filtering as all relations (ambiguous or not) need to be returned but not used while extending. This final intransitive mapping step will return all nodes related through either an *is_related* (brown edges) or *is_xref* edge with no filtering as all.

Table 5: Conversion of 'MONDO:0005027' (epilepsy) returning only direct cross-references using parameters 'step = 1'

from	to	label
MONDO:0005027	MONDO:0005027	Epilepsy, unspecified, not intractable, without status epilepticus
MONDO:0005027	ICD9:345.8	Epilepsy, unspecified
MONDO:0005027	ICD10:G40.9	Epilepsy and recurrent seizures
MONDO:0005027	ICD10:G40.909	epilepsy
MONDO:0005027	ICD9:345.80	Epilepsy
MONDO:0005027	ICD9:345.9	epilepsy
MONDO:0005027	ICD9:345	epilepsy
MONDO:0005027	ICD9:345.90	NA
MONDO:0005027	ICD9:345.91	NA
MONDO:0005027	COHD:380378	NA
MONDO:0005027	NIFSTD:birnlex_12718	NA
MONDO:0005027	ICD10:G40	NA
MONDO:0005027	MeSH:D004827	NA
MONDO:0005027	DOID:1826	NA
MONDO:0005027	SNOMEDCT:84757009	NA
MONDO:0005027	NCIt:C3020	NA
MONDO:0005027	EFO:0000474	NA

A second case is the return of direct cross-references only without the use of transitivty to return indirect relations (parameter "step = 1"). For the same identifier "MONDO:0005027", Table @ref{tab:step1} shows only the direct cross-references.

A third use case can be used to focus on concepts that are equivalent to each other and not return more broadly defined concepts. This can be a good practise when converting between ontologies that define concept similarly. With parameters 'step = NULL' and 'intransitive_ambiguity = 1' disease identifiers are convert using transitivitity to return indirect mappings while using an additional filter on ambiguity for the final intransitive step.

Finally, a specific conversion procedure is recommended for the ontologies that consider disease concepts that are less connected through *is_xref* edges to the "core" ontologies in DODO (e.g. MonDO, EFO, MedGen, etc.). Ontologies of ICD10, ICD9, ClinVar frequently report only those ontologies that are mapped using *is_related* cross-reference edges. When converting from these identifiers as a starting point, the conversion will not be able to return cross-reference relationship to all other database (even when they are avialable, Figure **??**). It limits or removes the effect of the transitive mapping. Therefore, for these ontologies it is recommend that in addition to the standard conversion, an additional step of intransitive mapping is performed afterwards which is implemented in the function *get_related* (Figure 6).

Conversion can also be used to convert between concept, i.e. from disease identifier to phenotype identifiers or vice versa. This conversion is handled in two distinct phases. First, depending on the options listed above, identifiers are converted within the same concept. When this is not required, using parameter "step = NA" the conversion within the same concept can be avoided. The second phase converts between concepts by returning phenotype or disease nodes directly related to the original identifiers (including the converted identifiers obtain in the first phase).

Building a disNet

A core concept in DODO is the *disNet*, a S3 object containing all information on a disease and its relationship with other nodes in the object (see above). This object can be constructed using helper functions starting from disease identifiers (*build_disnet*) and specifying either identifiers or terms to build a disNet around.

```
disNet <- build_disNet(id = c("MONDO:0005144"))

disNet <- build_disNet(term = "amyotrophic lateral sclerosis",
                       fields = c("label", "synonym"))
```

Table 6: Conversion of 'MONDO:0005027' (epilepsy) using transtivity mapping but only returning equivalent concepts in the final step by filtering on intransitive_ambiguity

from	to	label
MONDO:0005027	MONDO:0005027	Epilepsy, unspecified, not intractable, without status epilepticus
MONDO:0005027	ICD9:345.91	Epilepsy, unspecified, not intractable, without status epilepticus
MONDO:0005027	ICD9:345	Epilepsy, unspecified
MONDO:0005027	ICD9:345.8	Epilepsy, unspecified
MONDO:0005027	ICD9:345.90	Epilepsy, Cryptogenic
MONDO:0005027	NIFSTD:birnlex_12718	Epilepsy, Cryptogenic
MONDO:0005027	ICD10:G40.9	Epilepsy syndrome
MONDO:0005027	ICD10:G40.909	Epilepsy and recurrent seizures
MONDO:0005027	ICD10:G40	Epilepsy
MONDO:0005027	COHD:380378	Awakening Epilepsy
MONDO:0005027	ICD9:345.9	Awakening Epilepsy
MONDO:0005027	DOID:1826	epilepsy
MONDO:0005027	NCIt:C3020	Epilepsy
MONDO:0005027	EFO:0000474	Epilepsy
MONDO:0005027	SNOMEDCT:84757009	Epilepsy
MONDO:0005027	MeSH:D004827	epilepsy
MONDO:0005027	UMLS:C0014544	epilepsy
MONDO:0005027	MedGen:C0086237	NA
MONDO:0005027	UMLS:C0751111	NA
MONDO:0005027	MedGen:C0014544	NA
MONDO:0005027	UMLS:C0086237	NA
MONDO:0005027	MedGen:C0751111	NA
MONDO:0005027	ClinVar:9566	NA
MONDO:0005027	ICD9:345.80	NA
MONDO:0005027	ICD10CM:G40.909	NA
MONDO:0005027	ICD10CM:G40.9	NA
MONDO:0005027	SNOMEDCT_US_2018_03_01:84757009	NA
MONDO:0005027	ICD9CM:345.9	NA
MONDO:0005027	ORPHA:166463	NA
MONDO:0005027	GTR:GTRT000009566	NA
MONDO:0005027	OMIM:MTHU018883	NA
MONDO:0005027	OMIM:MTHU022799	NA
MONDO:0005027	SNOMEDCT:128613002	NA

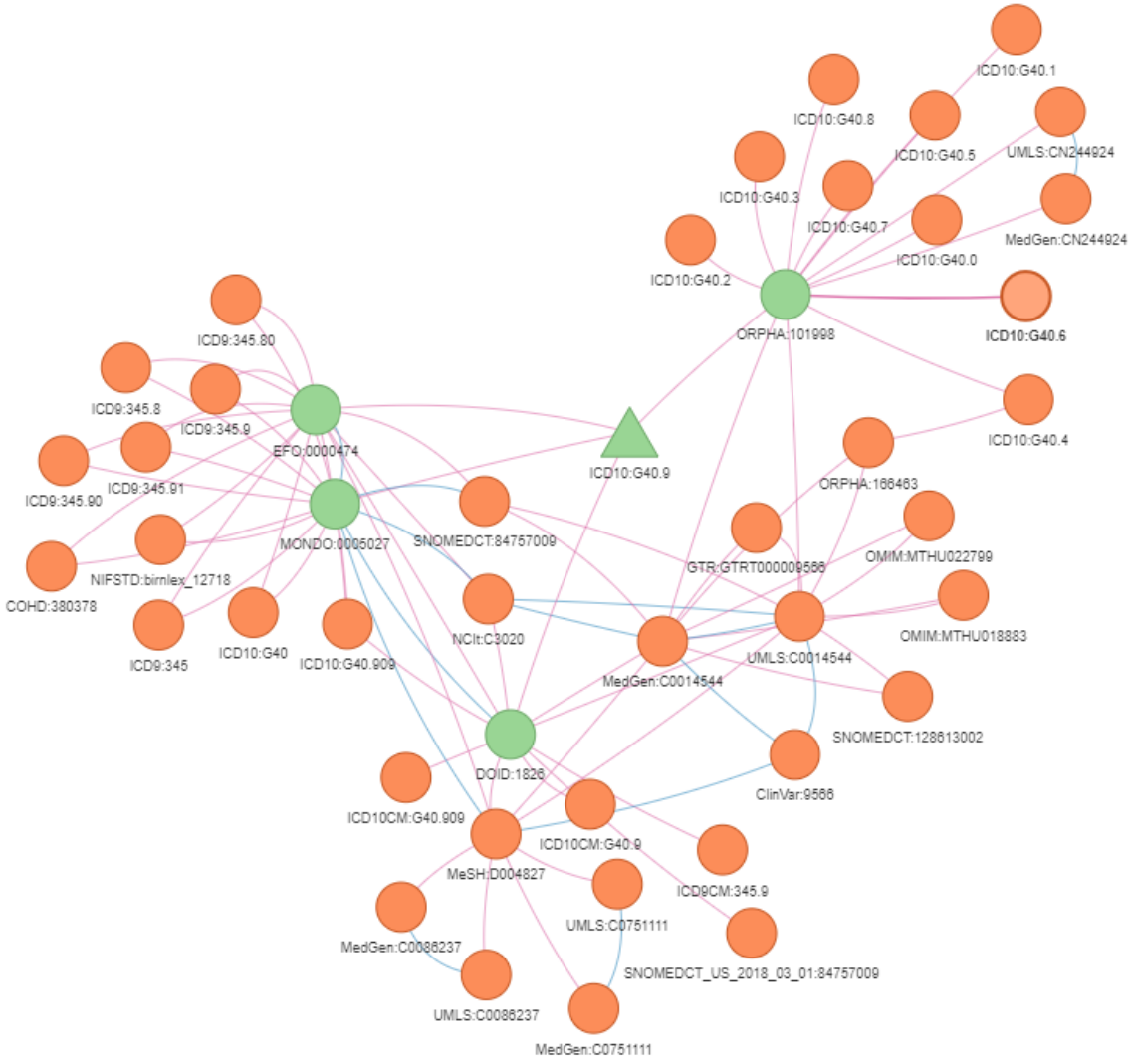

Figure 6: Comparison of the default conversion results with the output of the specific conversion for ontologies with limited connections through is_xref edges. The example is the ICD10 identifier G40.9 (Epilepsy). The nodes in green signify those return through both methods, while the nodes in orange are only returned using the more specific approach for this type of ontology. In addition, the edges in blue represent is_xref edges, while those in pink present is_related edges between nodes (with no specification for ambiguity for simplicity).

Extending a disNet

When building a *disNet* by either identifier or search term, the resulting *disNet* will likely not contain the complete information on that particular disease landscape and might be missing some cross-reference relationships. Alternatively, it might be necessary to extend the returned nodes to include their child terms, annotated phenotypes/disease, or alternative identifieres when available. To this end, the *extend_disnet* function in DODO allows extending a disNet object and returning additional cross-references edges, parent/child information, alternative identifierse, and/or phenotype annotation.

DODO is build as a meta-database incorporating several disease ontologies and their listed relationships. Not all relationships might be required or needed and exploring (*explore_disNet*) or plotting a *disNet* visualizes the sometimes more fuzzy concept of a disease as different ontologies employ hetergeneous definitions and cross-reference axes are not always exact. This may help to understand how disease are identified across ontologies and visualize the relationship between different resources.

Extension follows the same two-step approach as converting between ontologies. Please refer to paragraph @ref(Converting concepts) for more details and the different use cases.

Reviewing

Often after extending it is necessary to review the returned *disNet* to assess whether all nodes are of interest. This workflow decribed below aims to establish a process of reviewing a *disNet* following a top/down approach. Clustering the *disNet* nodes based on cross-reference edges, allows the user to focus on the disease he/she wishes to retain and takes care of these cross-reference information in the back.

For a smaller-sized *disNet* the clustering step and exploring these clusters might be sufficient to review a given object.

```
# ## Cluster disNet on "xref"
# clDisNet <- cluster_disNet(disNet = disNet,
#                            clusterOn = "xref")
#
# ## Explore clusters
# # explore_disNet(disNet = clDisNet)
# disNet <- c(list = clDisNet[c(1:2, 5, 9)])
```

When establishing a *disNet* based on search terms, some returned nodes have no parent/child edges or cross-reference edges. These are considered as *singletons* and are treated separately as these can still be of interest when aiming to link to different external resources.

Phenotypes

Finally, it is also possible to extend to phenotype information using either *convert_concept* and specify the concept to convert to. Additionally, the *extend_disNet* can be used to extend identifiers to either “Phenotype” or “Disease” relations. While this extension can be performede simultaneously with extension using “xref”, “parent/child”, or “alt” parameters, it is not possible to extend through both “disease” and “phenotype” at the same time. Please note that extending through either phenotype or disease in combination with other options for extension (e.g. xref, parent, child, or alt) will not use transitivity for phenotype or disease extension. This is only performed as a final step after two-step extension (similar to conversion, please refer to paragraph @ref{Converting concepts} for more details).

```
disNet <- build_disNet(id = c("HP:0003394", "HP:0002180", "HP:0002878"))
disNet <- extend_disNet(disNet = disNet, relations = "disease")
nrow(disNet)
```

```
## [1] 140
```

Connecting to external resources

The aim of DODO is to facilitate the connection with external resources. Defining a disNet around a disease of interest annotated with its cross-references and child terms ensures the connection to an external resources returns a list that is exhaustive.

Below this use with DODO will be exemplified by connecting to two different external resources ClinVar and CHEMBL for “Amyotropic Lateral Sclerosis”. To compare this usage, the resources themselves (ClinVar and

CHEMBL) are queried directly for this indication and the results compared to the use of a disNet. In addition the use of a disNet across multiple resources ensures transitivity by defining a network of equivalent diseases that allows tracing the connecting between these different resources easily.

CHEMBL

Both approaches identify the same drug compounds but an additional disease was available through use of a disNet.

This disease is a child term identified through extending the disNet:

The table below shows the disease labels:

ClinVar

The table below shows the genes carrying a disease variant only identified through ClinVar directly:

These terms from MedGen and OMIM are related to term "MONDO:0014178" () itself a child term of "MONDO:0000507" (). This additional variant was identified through extending the disNet and obtaining al child terms and related cross-references.

Tracing connections

A additional feature from connecting resources through the use of a disNet, is the possibility to identify if and how diseases returned from each resource are connected to each other. This does not only allow a better understanding of disease, but also facilitates downstream analyses.

Bridging external resources

Connecting ClinVar to CHEMBL

–>

Conclusions

Disease ontologies have allowed a more formal classification of diseases and could facilitate the integration of several biological databases to increase disease understanding and develop novel treatments. However, efforts to integrate biological databases or the ontologies directly are complicated by ontology-specific identifiers, heterogeneous decisions on disease definitions, and the inherent presence of errors. Despite this efforts, we identified two remaning challenges:

- Currently no resource provides a flexible and complete mapping across the multitude of disease ontologies
- Availability of efficient and straightforward software on bioinformatics platform (R or python)

DODO R package contains several functions to build and interact with a disNet or convert concept identifiers between ontologies. The workflow to construct a custom, local DODO graphical database is provided with the intent to allow adaptation to specific needs and additional resources can be easily added without adaptation to the data model or R package.

We believe that DODO can help clarify and define conditions of interest and explore and visualize relationships between disease concepts. This allows the exploration of a disease and how correct the different mappings between ontologies are. This hopefully helps to increase disease understanding to people inside and outside of the medical field. In addition, connecting different biological database (information on compounds, text mining, transcriptomics, etc.) through a disNet ensures these resources are queried using different equivalent identifiers of the disease of interest without the user being required to make decisions on these mappings. In addition, it also allows visualizing the connection between these resources directly.

Through the aggregation of different ontologies and their mappings, DODO functions as a meta-database with easy access and possibility to estabblish a more exhaustive disease landscape. The code to build and query DODO is provided under open source license to allow further improvement by other developers.

–>

–>

Software availability

This section will be generated by the Editorial Office before publication. Authors are asked to provide some initial information to assist the Editorial Office, as detailed below.

1. URL link to where the software can be downloaded from or used by a non-coder (AUTHOR TO PROVIDE; optional)
2. URL link to the author's version control system repository containing the source code (AUTHOR TO PROVIDE; required)
3. Link to source code as at time of publication (*F1000Research* TO GENERATE)
4. Link to archived source code as at time of publication (*F1000Research* TO GENERATE)
5. Software license (AUTHOR TO PROVIDE; required)

Author information

In order to give appropriate credit to each author of an article, the individual contributions of each author to the manuscript should be detailed in this section. We recommend using author initials and then stating briefly how they contributed.

Competing interests

All financial, personal, or professional competing interests for any of the authors that could be construed to unduly influence the content of the article must be disclosed and will be displayed alongside the article. If there are no relevant competing interests to declare, please add the following: 'No competing interests were disclosed'.

Grant information

Please state who funded the work discussed in this article, whether it is your employer, a grant funder etc. Please do not list funding that you have that is not relevant to this specific piece of research. For each funder, please state the funder's name, the grant number where applicable, and the individual to whom the grant was assigned. If your work was not funded by any grants, please include the line: 'The author(s) declared that no grants were involved in supporting this work.'

Acknowledgments

This section should acknowledge anyone who contributed to the research or the article but who does not qualify as an author based on the criteria provided earlier (e.g. someone or an organization that provided writing assistance). Please state how they contributed; authors should obtain permission to acknowledge from all those mentioned in the Acknowledgments section.

Please do not list grant funding in this section.

Additional advice to authors

The content above is either taken directly from the F1000Research LaTeX template, or has been adpated to suit R Markdown. Here we provide some additional advice to authors based on our own and other users' experiences of article submission to F1000Research, which will hopefully make your own journey smoother. If you would like to contribute to this, please contact us at https://github.com/grimbough/BiocWorkflowTools/issues/4

- When writing figure or tables captions do not use acronyms or abbreviations without explanation in the legend itself, even this has been addressed in the main text. F1000Research require each figure and table to be able to stand alone from the rest of the manuscript. *Thanks to Mike Love @mikelove*

References

Winston Chang. shinythemes: themes for shiny, 2018. URL https://cran.r-project.org/package=shinythemes.

Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. shiny: Web Application Framework for R, 2019. URL https://cran.r-project.org/package=shiny.

Liang Cheng, Guohua Wang, Jie Li, Tianjiao Zhang, Peigang Xu, and Yadong Wang. SIDD: A Semantically Integrated Database towards a Global View of Human Disease. *PLoS ONE*, 8(10):1–9, 2013. ISSN 19326203. doi: 10.1371/journal.pone.0075504.

Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Compex Sys:1695, 2006. URL http://igraph.org.

Thomas R Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Aquisition*, 5(2): 199–220, 1993. URL https://ebiquity.umbc.edu/get/a/publication/501.pdf.

Melissa A Haendel, Julie A Mcmurry, Rose Relevo, Christopher J Mungall, N Peter, and Christopher G Chute. A Census of Disease Ontologies. *Annual Review of Biomedical Data Science*, 1:305–331, 2018.

Ali Hasnain, Maulik R. Kamdar, Panagiotis Hasapis, Dimitris Zeginis, Claude N. Warren, Helena F. Deus, Dimitrios Ntalaperas, Konstantinos Tarabanis, Muntazir Mehdi, and Stefan Decker. Linked biomedical dataspace: Lessons learned integrating data for drug discovery. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8796:114–130, 2014. ISSN 16113349. doi: 10.1007/978-3-319-11964-9.

Robert Hoehndorf, Michel Dumontier, and Georgios V. Gkoutos. Evaluation of research in biomedical ontologies. *Briefings in Bioinformatics*, 14(6):696–712, 2013. ISSN 14675463. doi: 10.1093/bib/bbs053.

Wei Hu, Honglei Qiu, Jiacheng Huang, and Michel Dumontier. BioSearch: a semantic search engine for Bio2RDF. *Database : the journal of biological databases and curation*, 2017:1–13, 2017. ISSN 17580463. doi: 10.1093/database/bax059.

Warren A. Kibbe, Cesar Arze, Victor Felix, Elvira Mitraka, Evan Bolton, Gang Fu, Christopher J. Mungall, Janos X. Binder, James Malone, Drashtti Vasant, Helen Parkinson, and Lynn M. Schriml. Disease Ontology 2015 update: An expanded and updated database of Human diseases for linking biomedical knowledge through disease data. *Nucleic Acids Research*, 43(D1):D1071–D1078, 2015. ISSN 13624962. doi: 10.1093/nar/gku1011.

Kevin M Livingston, Michael Bada, William A. Baumgartner, and Lawrence E Hunter. KaBOB: ontology-based semantic integration of biomedical databases. *BMC Bioinformatics*, 16(1):1–21, 2015. ISSN 14712105. doi: 10.1186/s12859-015-0559-3.

James Malone, Ele Holloway, Tomasz Adamusiak, Misha Kapushesky, Jie Zheng, Nikolay Kolesnikov, Anna Zhukova, Alvis Brazma, and Helen Parkinson. Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics*, 26(8):1112–1118, 2010. ISSN 13674803. doi: 10.1093/bioinformatics/btq099.

Kirill Müller and Hadly Wickham. tibble: simple data frames, 2019. URL https://cran.r-project.org/package=tibble.

Christopher J. Mungall, Julie A. McMurry, Sebastian Kohler, James P. Balhoff, Charles Borromeo, Matthew Brush, Seth Carbon, Tom Conlin, Nathan Dunn, Mark Engelstad, Erin Foster, J. P. Gourdine, Julius O.B. Jacobsen, Dan Keith, Bryan Laraway, Suzanna E. Lewis, Jeremy Nguyen Xuan, Kent Shefchek, Nicole Vasilevsky, Zhou Yuan, Nicole Washington, Harry Hochheiser, Tudor Groza, Damian Smedley, Peter N. Robinson, and Melissa A. Haendel. The Monarch Initiative: An integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Research*, 45(D1):D712–D722, 2017. ISSN 13624962. doi: 10.1093/nar/gkw1128.

R Core Team. R: A Language and Environment for Statistical Computing, 2019. URL https://www.r-project.org/.

Noa Rappaport, Noam Nativ, Gil Stelzer, Michal Twik, Yaron Guan-Golan, Tsippi Iny Stein, Iris Bahir, Frida Belinky, C. Paul Morrey, Marilyn Safran, and Doron Lancet. MalaCards: An integrated compendium for diseases and their annotation. *Database*, 2013:1–14, 2013. ISSN 17580463. doi: 10.1093/database/bat018.

Kun Ren. rlist: a toolbox from non-tabular data manipulation, 2016. URL https://cran.r-project.org/package=rlist.

Mansoor Saqi, Artem Lysenko, Yi-ke Guo, Tatsuhiko Tsunoda, and Charles Auffray. Navigating the disease landscape: knowledge representations for contextualizing molecular signatures. *Briefings In Bioinformatics*, (November 2017):1–15, 2018. ISSN 1467-5463. doi: 10.1093/bib/bby025. URL http://fdslive.oup.com/www.oup.com/pdf/production{_}in{_}progress.pdf.

Lynn M Schriml and Elvira Mitraka. The Disease Ontology : fostering interoperability between biological and clinical human disease-related data. *Mammalian Genome*, 26(9):584–589, 2015. ISSN 1432-1777. doi: 10.1007/s00335-015-9576-9.

Kent A Shefchek, Nomi L Harris, Michael Gargano, Nicolas Matentzoglu, Deepak Unni, Matthew Brush, Daniel Keith, Tom Conlin, Nicole Vasilevsky, Aaron Zhang, James P Balhoff, Larry Babb, Susan M Bello, Hannah Blau, Yvonne Bradford, Seth Carbon, Leigh Carmody, Lauren E Chan, Valentina Cipriani, Alayne Cuzick, Maria D Rocca, Nathan Dunn, Shahim Essaid, Petra Fey, Chris Grove, Jean-phillipe Gourdine, Ada Hamosh, Midori Harris, Ingo Helbig, Maureen Hoatlin, Marcin Joachimiak, Simon Jupp, M Pendlington, Clare Pilgrim, B Lett, Suzanna E Lewis, Craig Mcnamara, Tim Putman, Vida Ravanmehr, Justin Reese, Erin Riggs, Sofia Robb, Paola Roncaglia, James Seager, Erik Segerdell, Morgan Similuk, Andrea L Storm, Courtney Thaxon, Anne Thessen, Julius O B Jacobsen, Julie A Mcmurry, Tudor Groza, K Sebastian, Damian Smedley, Peter N Robinson, J Mungall, Melissa A Haendel, Monica C Munoz-torres, and David Osumi-sutherland. The Monarch Initiative in 2019 : an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Research*, 1:1–12, 2019. doi: 10.1093/nar/gkz997.

Hadly Wickham. stringr: simple, consistent wrappers for common string operations, 2019. URL https://cran.r-project.org/package=stringr.

Hadly Wickham, Romain François, Lionel Henry, and Kirill Müller. dplyr: a grammar of data manipulation, 2019. URL https://cran.r-project.org/package=dplyr.

Yihui Xie, Joe Cheng, and Xianying Tan. DT: a wrapper for the JavaScript Library "DataTables", 2019. URL https://cran.r-project.org/package=DT.

Guangchuang Yu, Li Gen Wang, Guang Rong Yan, and Qing Yu He. DOSE: An R/Bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics*, 31(4):608–609, 2015. ISSN 14602059. doi: 10.1093/bioinformatics/btu684.