Dictionary of disease ontologies (DODO): a graph database to facilitate access and interaction with disease and phenotype ontologies

Liesbeth Francois, Jonathan van Eyll and Patrice Godard

Abstract

Disease ontologies assign diseases to a more formal classification that facilate the connection to biological databases containing drug information, transcriptomics and genomics information, and others. However, different ontologies employ independent identifiers and use heterogenous decisions on disease definitions. Despite ongoing efforts on integration, two challenges can still be identified. First, no resource provides a complete mapping across the multitude of disease ontologies and there is no ready to use software available to interact and explore with ontologies on the standard bioinformatics platforms. In this paper, the DODO (Dictionary of Disease Ontology) database and accompagnying R package are presented. DODO aims to deal with these two challenges by constructing a meta-database incorporating information of different publically available disease ontology resources and their annotations. Through the use of transitivity mappings build on the cross-reference relationships provided by the ontologies, indirect relations between disease and phenotype identifiers can be retrieved. The accompagnying R package contains several functions to build and interact with disease networks or convert concept identifiers between ontologies. It specifically aims to facilitate access and the exhaustive extraction of information from life science databases without the need to harmonize these upfront. The workflow for local adaption and extension of the DODO database is available as well as a docker image of the DODO database for convenience.

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 diseases; however, each ontology 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 knowledgebases, 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 knowledgebases 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. It subsequently enriches these classes with additional axioms when needed [Malone et al., 2010]. Currently, it combines information from OMIM, Orphanet, ICD9/10 and SNOMEDCT, HPO, UBERON, and MonDO [Ochoa, 2019].

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. In addition, the existing efforts for integration are not flexible to extend easily to proprietary disease ontologies. DODO combines the information provided by the different ontologies, to enrich cross-reference mappings. It also allows connecting ontologies to each other that have no direct cross-reference mapping between them by indirectly inferring relationships using transitivity. 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]. Such access could facilitate a more flexible connection to the different life science resources and 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 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 ensure 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

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

Data model

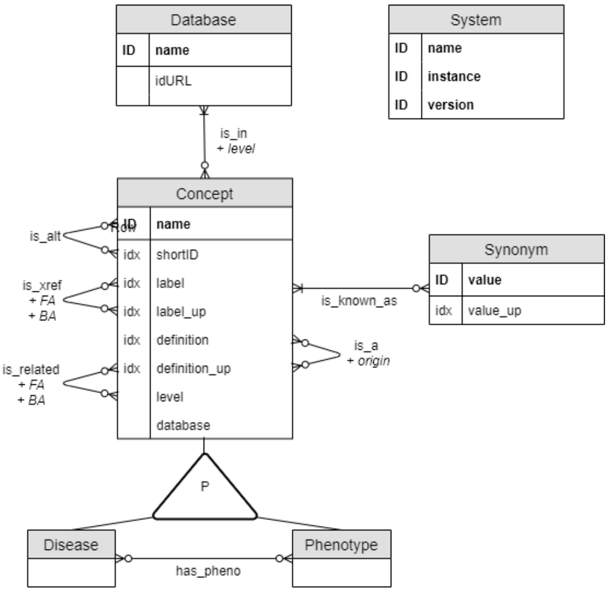

Figure 1: The DODO data model is shown as an Entity/Relationship (ER) diagram. It consist of four types of entities (concept, disease, phenotype, database) corresponding to graph nodes. Several relationships between the nodes are described referring to graph edges. 'ID' refers to an unique entity while '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 (Figure 1). It relies on four types of nodes (concept, disease, phenotype, and database) each with specific proporties. A disease or phenotype node share the same properties. *Name* is a primary property and the full length identifier of a disease or phenotype, 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. When available, a node also has the property of (hierarchical) *level* which captures the lowest level a disease has in the ontological tree. Each node with class disease or phenotype is also encoded with 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)

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

Disease ontology	GitHub
Monarch Disease Ontology (MonDO)	https://github.com/Elysheba/Monarch
Experimental Factor Ontology (EFO)	https://github.com/Elysheba/EFO
Orphanet	https://github.com/Elysheba/Orphanet
MedGen	https://github.com/Elysheba/MedGen
Medical Subject Headings (MeSH)	https://github.com/Elysheba/MeSH
Human Phenotype Ontology (HPO)	https://github.com/patzaw/HPO
ClinVar	https://github.com/patzaw/ClinVar
Disease Ontology (DO)	https://github.com/Elysheba/DO
International Classification of Diseases (ICD11)	https://github.com/Elysheba/ICD11

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 paragraph . 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.

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 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 scripts is available to load and feed a Neo4j instance. These are not exposed directly to the user instead and a general workflow is available in the *build/scripts* folder. The feeding of DODO is based on the parsed files of the different ontologies. The workflow to download and parse each included ontology is available through GitHub (Table 1).

The different steps to construct a new DODO Neo4j instance 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 [Inc., 2019]: https://hub.docker.com/repository/docker/elysheba/dodo (tag: 02.04.2020). 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, alternative identifiers, and cross-reference information) around a disease and is structured as shown in Table 2.

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

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
type	type of cross-reference edge (is_xref or is_related)
alt	
id	current identifier
alt	deprecated identifier
pheno	
disease	disease identifier
phenotype	phenotype identifier
seed	
seed	vector of disease ids used to seed the disNet

Operation

The data model is implemented using the Neo4j software which uses the Cypher query language [Inc., 2020]. An accompagnying R package *DODO* was developed to connect and provides higher level functions to query the Neo4j graph database based on the described data model (paragraph) [R Core Team, 2019].

The minimal system requirements are:

- R ≥ 3.6
- Operating system: Linux, macOS, Windows
- Memory ≥ 4GB RAM

The graph database has been implemented with Neo4j 3.5.14 [Inc., 2020], the DODO R package uses the following packages:

- *dplyr* [Wickham et al., 2019]
- *tibble* [Müller and Wickham, 2019]
- *neo2R* [Godard and van Eyll, 2018]
- *rlist* [Ren, 2016]
- *stringr* [Wickham, 2019]
- *readr* [Wickham et al., 2018]
- *visNetwork* [Almende B.V. et al., 2019]
- *shinythemes* [Chang, 2018]
- *DT* [Xie et al., 2019]
- *igraph* [Csardi and Nepusz, 2006]
- *shiny* [Chang et al., 2019]

Querying the database

The DODO R package combines several functions to construct, interact, and explore a disNet object. These will be briefly outlined in the sections below. 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 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 in similar terms. 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. As mentioned before, not all nodes have label information available. This information can only be added when it is provided by the original resources.

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 wanted. This can result in an exponential increase of converted/expanded identifiers that are only distantly related to the original disease 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

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

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 similarly defined as the original identifier. However, among its cross-references it lists a relation to an ICD10 identifier (Q87.0), a broad term of 'Congenital malformation syndromes predominantly affecting facial appearance'. This identifier is highly ambiguous and links directly to 284 disease concepts. (Disease labels are only available when present in the parsed resources. Otherwise only a disease identifier is available).

Disease identifier	Disease label
ICD10:Q87.0	Congenital malformation syndromes predominantly affecting facial appearance
OMIM:303600	COFFIN-LOWRY SYNDROME
MeSH:D038921	Coffin-Lowry Syndrome
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
MedGen:C0795900	Coffin syndrome 1
MeSH:C536435	
SNOMEDCT:15182000	
OMIM:300075	
UMLS:C0795900	
GTR:GTRT000007048	
ICD9:759.89	
OMIM:300075.0011	
GTR:GTRT000000823	
NCIt:C84643	
GARD:6123	
GARD:0008589	
SNOMEDCT_US_2019_09_01:15182000	
GARD:0006123	

to each other (similar concept definitions). The *is_related* edge is used for all other cross-reference edges. The assignemnt of these edges is based on the maximum value of the sum of forward and backward ambiguities of all cross-references relationships between two databases to quantify the symmetry between them and knowledge of how ontologies deal with and define 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.

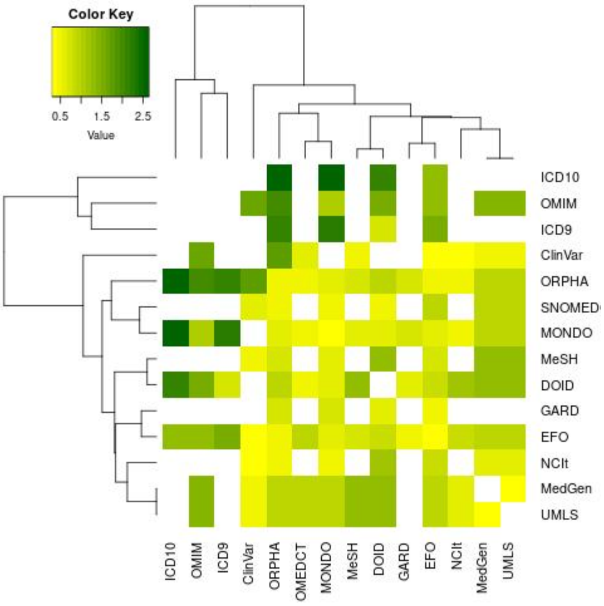

Figure 2: The heatmap shows the maximum value of total ambiguity between ontologies using a log_{10} transformation. While many ontologies are using concepts of similar level as identified by low total ambiguity, a few can be identified that are more ambiguous in their mappings. The 'optimal' ambiguity 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 ontologies of either *is_xref* or *is_related* and knowledge of the way disease concepts are defined within ontologies.

The number of conversions per identifier is shown on Figure 3 for MonDO as an example 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, more distantly related cross-reference identifiers.

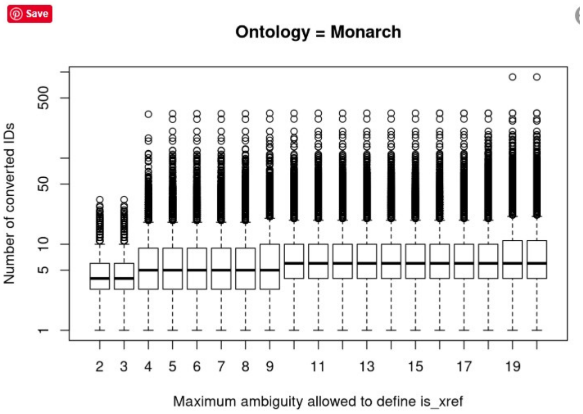

Figure 3: Number of conversions comparing incremental cutoffs in the (total) ambiguity to define *is_xref* and *is_related* cross-reference edges on an ontology scale (example of MonDO ontology).

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. Additionally, as OMIM and Orphanet define very narrow subtypes of diseases and these will also be encoded as an *is_xref* edge.

Results

The table below shows the number of nodes available through each disease ontology, in total there are 586468 nodes (Figure @ref{fig:listDB)) in this DODO instance.

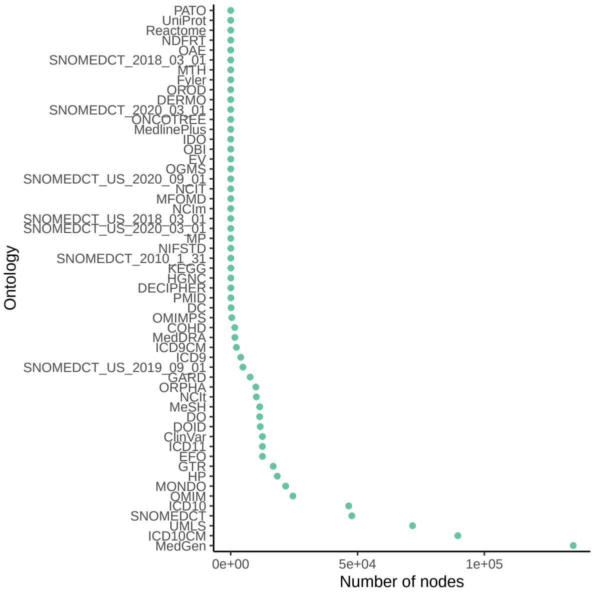

Figure 4: Overview of the number of nodes present for each disease ontology in DODO.

Use cases

Converting concepts

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

A first and default conversion consists of two steps. The first step uses the transitivity on *is_xref* edges followed by one step expansion using both types of cross-reference edges (parameters "step = NULL" and "intranstivity_ambiguity = NULL"). In Figure 5 an example is presented of the extension starting from the MonDO identifier "MONDO:0005027" encoding for "Epilepsy". The transitive mapping with filtering on ambiguity moves between the *is_xref* edges (blue) with ambiguity equal to one. The final intransitive mapping step will return all nodes related through either an *is_related* or *is_xref* edge with no filtering (orange) so ambigious relations can also to be returned but not used when extending.

This conversion can be used to get all identifiers around a disease concept whether broadly 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 no filter on the *intransitive_ambiguity*. However, for the first step using transitivity mapping on *is_xref* edges it is strongly recommended to use the default filtering on ambiguity by limiting (backward) ambiguity to one.

A second use case can be identified when the aim is to only get direct cross-references of the identifier of interest withouth using transitivity to return indirect relations (parameter "step = 1"). While transitivity allows the incorporation of information from different ontologies and allows users to connect indirectly related concepts, these relationships might be more vague. This would limit the returned conversion to the nodes indicated in green on Figure 5.

A third use case can be identified when the aim of conversion is to only return equivalent concepts and not return broader disease concepts. This can be a good practise when converting between ontologies that define concept similarly. Here the transitivity mechanism can be employed by parameter 'step = NULL' and supplying 'intransitive_ambiguity = 1' to ensure that only equivalent edges are traversed in the final step. Figure 6 shows the difference between the default conversion and filtering on the final one step extension.

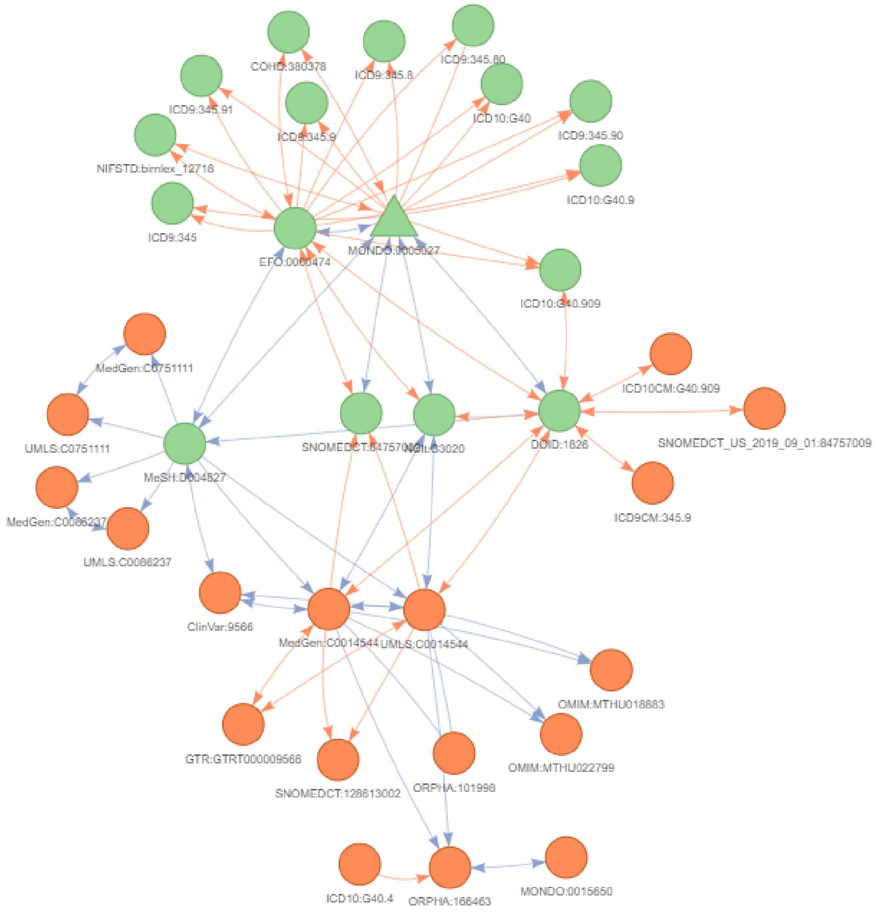

Figure 5: A disNet constructed and subsequently extended using only cross-reference relations using the MonDO identifier for 'Epilepsy' ('MONDO:0005027') as an example. This extension is performed in two phases. First, the transitive mapping uses the *is_xref* edges (blue edges) wich are trusted to a larger extend to obtain all relations with filtering on ambiguity (backward ambiguity = 1) (green nodes). The second steps uses no filtering as all relations (ambiguous or not) need to be returned but not used while extending (red nodes). This final intransitive mapping step will return all nodes related through either an *is_related* (orange edges) or *is_xref* edge (blue) with no filtering. The arrows show how nodes can be reached when taking backward ambiguity filtering into account. When there are no arrows present the ambiguity is higher than one and this node can only be reached during the final step of intransitive mapping with no filtering put in place.

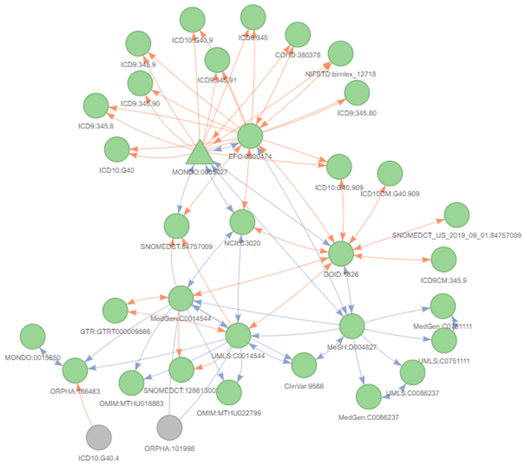

Figure 6: Conversion of 'MONDO:0005027' (epilepsy) using transtivity mapping but only returning equivalent concepts in the final step by filtering on intransitive_ambiguity

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.) such as ontologies like ICD10 or ICD9. When converting from these identifiers as a starting point, the conversion will not be able to return cross-reference relationships to all other database. It limits or removes the transitivity mechanism and, therefore, we recommend an additional step to the standard conversion. Implemented in the *get_related* function, an addition expension step through *is_related* and *is_xref* edges is performed before the standard conversion procedure. The ambiguity on this additional step is the same for the final intransitive step in the standard conversion procedure (modified by the *intransitive_ambiguity* parameter). Figure 7 shows the difference between the standard conversion procedure and the recommended procedure on ICD10:G40.9 (Epilepsy). When using the standard conversion only the nodes indicated in green are returned and it is not possible to make use of the transitivity mechanism. By using the recommend *get_related* procedure with an additional expansion step followed by standard conversion, the transitivity mapping can be employed (nodes in orange).

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).

```
## From disease to phenotype
toPhenotype <- convert_concept(from = "MONDO:0012391",
                               to = "HP",
                               from.concept = "Disease",
                               to.concept = "Phenotype")
head(toPhenotype)
```

```
## # A tibble: 6 x 3
##   from          to         deprecated
##   <chr>         <chr>      <lgl>
## 1 MONDO:0012391 HP:0000737 FALSE
## 2 MONDO:0012391 HP:0002069 FALSE
## 3 MONDO:0012391 HP:0001249 FALSE
## 4 MONDO:0012391 HP:0001272 FALSE
## 5 MONDO:0012391 HP:0000529 FALSE
## 6 MONDO:0012391 HP:0002059 FALSE
```

```
## From phenotype to disease
toDisease <- convert_concept(from = "HP:0002384",
                             from.concept = "Phenotype",
                             to.concept = "Disease")
head(toDisease)
```

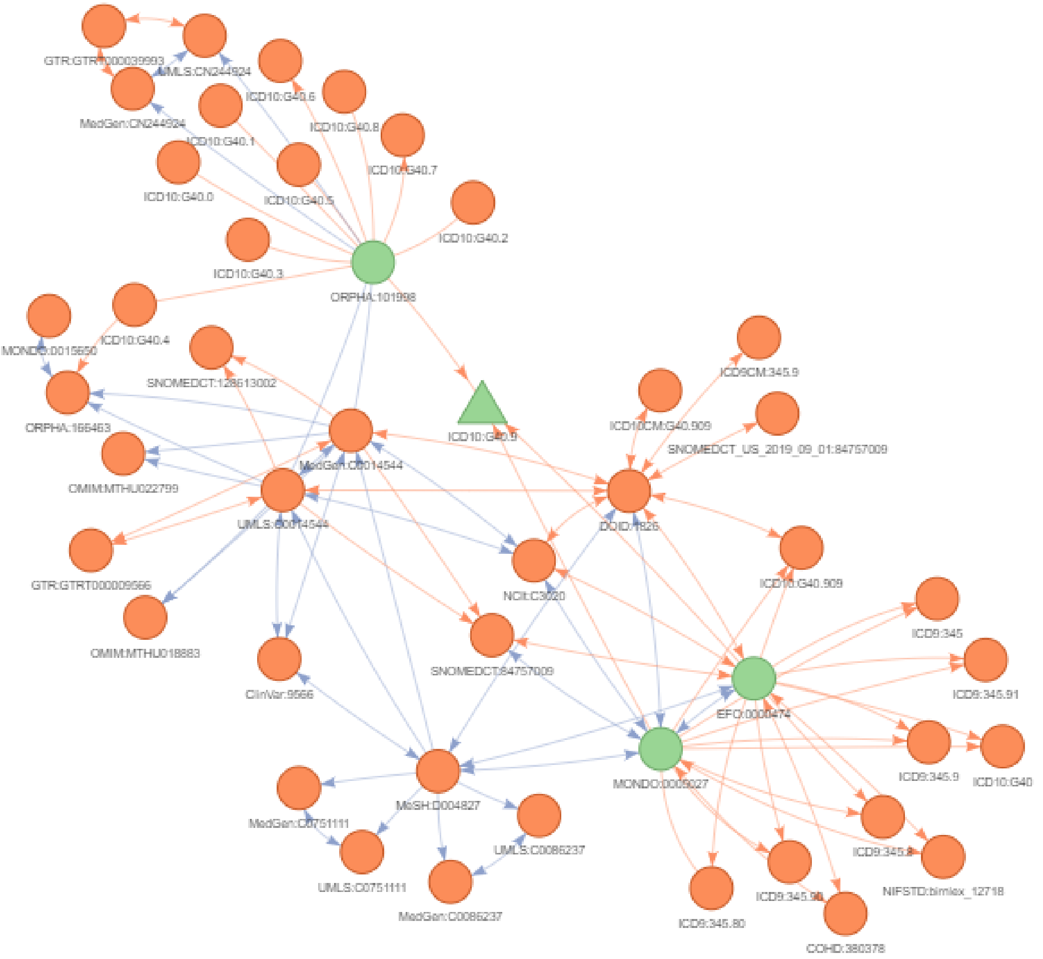

Figure 7: Comparison of the default conversion results with the output of the specific conversion for ontologies with limited connections through is_xref edges (blue edges). The example is the ICD10 identifier G40.9 (Epilepsy). The nodes in green signify those return when using the standard conversion. This removes the functionality of the transitive mapping between nodes. Using the recommended procedure implemented in get_related() function with an additional step of expansion before the standard conversion, the transitivity can be employed (nodes in orange). The edges in blue represent is_xref edges, while those in pink present is_related edges between nodes. The arrows show how nodes can be reached when taking backward ambiguity filtering into account. When there are no arrows present the ambiguity is higher than one and this node can only be reached during the final step of intransitive mapping with no filtering put in place.

```
## # A tibble: 6 x 3
##   from       to             deprecated
##   <chr>      <chr>          <lgl>
## 1 HP:0002384 HP:0002384     FALSE
## 2 HP:0002384 ClinVar:17141  FALSE
## 3 HP:0002384 UMLS:C0270834  FALSE
## 4 HP:0002384 MONDO:0009697  FALSE
## 5 HP:0002384 MONDO:0009079  FALSE
## 6 HP:0002384 MONDO:0016825  FALSE
```

Finally, conversion can also be used to return deprecated identifiers when these are available.

```
deprecated <- convert_concept(from = "HP:0009638",
                              deprecated = TRUE,
                              from.concept = "Phenotype",
                              to.concept = "Phenotype")
head(deprecated)
```

```
## # A tibble: 2 x 3
##   from       to         deprecated
##   <chr>      <chr>      <lgl>
## 1 HP:0009638 HP:0004079 TRUE
## 2 HP:0009638 HP:0006073 TRUE
```

Building a disNet

A core concept in DODO is the S3 *disNet* object which contains all information on diseases/phenotypes and their internal relationships. This object can be constructed using helper function *build_disnet* and supplying it with either a vector of disease identifiers or search terms.

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

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

Extending a disNet

When building a *disNet* by either identifier(s) or search term(s), the resulting *disNet* will likely not contain the complete information on that particular disease landscape. The *extend_disNet* function enriches the disNet and extends it to cross-reference identifiers, child/parent terms, annotated phenotypes/disease, and/or alternative identifieres when available. In concordance with the conversion procedure, extension follows the same two-step approach. Please refer to paragraph for more details and the different use cases.

Of specific note is the extension to (or from) phenotype information. Within one extension all different parameters (xref, child, parent, alt, and disease/phenotype) can be supplied with the exception that it is not possible to extend to both disease and phenotype simultaneously. Extending to or from phenotypes does not employ the transitivity mechanism but is performed as a final step (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")
disNet
```

```
## # A tibble: 397 x 7
##    id      label            definition            shortID level type    database
##    <chr>   <chr>            <chr>                 <chr>   <int> <chr>   <chr>
##  1 MONDO:~ glycogen storag~ "Phosphoglycerate ki~ 0010392    11 Concep~ MONDO
##  2 OMIM:6~ NEURODEGENERATI~ "NEURODEGENERATION W~ 610217     NA Concep~ OMIM
##  3 MONDO:~ adult-onset dis~ "Adult-onset distal ~ 0018006    10 Concep~ MONDO
##  4 MONDO:~ Charcot-Marie-T~ "Autosomal dominant ~ 0011675     9 Concep~ MONDO
##  5 OMIM:6~ MOTOR NEURON DI~ "MOTOR NEURON DISEAS~ 600333     NA Concep~ OMIM
##  6 MONDO:~ pure mitochondr~ "Pure mitochondrial ~ 0016807    11 Concep~ MONDO
##  7 ORPHA:~ Congenital musc~ "Congenital muscular~ 258         8 Concep~ ORPHA
##  8 OMIM:3~ MITOCHONDRIAL C~ "MITOCHONDRIAL COMPL~ 301021     NA Concep~ OMIM
##  9 ORPHA:~ Nocardiosis      "Nocardiosis is a lo~ 31204       4 Concep~ ORPHA
```

Table 5: Annotation of the different cross-reference clusters of nodes identified for a disNet around 'amyotrophic lateral sclerosis'. The annotation is obtained from node within the cluster with the highest order in its disease ontology ('level') and avialability of label information.

cluster	clusterSize	id	label
1	150	ICD10CM:G12.21	Amyotrophic lateral sclerosis
2	22	MONDO:0017161	frontotemporal dementia with motor neuron disease
3	2	MedGen:C1862940	Amyotrophic Lateral Sclerosis, Autosomal Recessive
4	9	ORPHA:357043	Amyotrophic lateral sclerosis type 4
5	3	MedGen:C3542025	Amyotrophic lateral sclerosis 1, autosomal recessive
6	8	MONDO:0005145	sporadic amyotrophic lateral sclerosis
7	6	MONDO:0014640	FTDALS3
8	5	MONDO:0008781	juvenile amyotrophic lateral sclerosis with dementia
9	6	MONDO:0011632	amyotrophic lateral sclerosis type 21
10	2	MedGen:C4302169	Amyotrophic lateral sclerosis plus syndrome
11	3	UMLS:C2750729	Amyotrophic lateral sclerosis 6, autosomal recessive
12	3	UMLS:CN239196	Amyotrophic Lateral Sclerosis, Recessive
13	2	MedGen:CN260033	Amyotrophic lateral sclerosis 10, with or without FTD
14	4	ORPHA:52430	Inclusion body myopathy with Paget disease of bone and frontotemporal dementia
15	3	UMLS:C2931441	Infantile-onset ascending hereditary spastic paralysis
16	2	MedGen:C2931786	Amyotrophic lateral sclerosis, type 6
17	3	ORPHA:98756	Spinocerebellar ataxia type 2
18	2	MedGen:C3662062	Restrictive lung disease due to amyotrophic lateral sclerosis
19	3	MedGen:CN239175	Amyotrophic Lateral Sclerosis, Dominant
20	2	MedGen:C4551993	Amyotrophic Lateral Sclerosis, Familial
21	3	UMLS:CN239211	Amyotrophic Lateral Sclerosis/Frontotemporal Dementia
22	1	ClinVar:10103	Amyotrophic lateral sclerosis, typical
23	1	ClinVar:10438	Amyotrophic lateral sclerosis, susceptibility to
24	1	ClinVar:10387	Amyotrophic lateral sclerosis 13
25	1	ClinVar:32676	Amyotrophic lateral sclerosis 22 with frontotemporal dementia
26	1	MONDO:0008178	inclusion body myopathy with Paget disease of bone and frontotemporal dementia type 1
27	1	ClinVar:10104	Amyotrophic lateral sclerosis-parkinsonism/dementia complex 1, susceptibility to
28	1	ClinVar:16925	Amyotrophic lateral sclerosis 14 without frontotemporal dementia
29	1	HP:0007354	Amyotrophic lateral sclerosis

```
## 10 ORPHA:~ Mitochondrial e~ " mutation is charac~ 1194        9 Concep~ ORPHA
## # ... with 387 more rows
##
## The disNet contains:
##  -  394 disease nodes from 6 ontologies and 3 phenotype nodes from 1 ontologies
##  -  2206 synonyms of the disease nodes
##  -  0 parent/child edges
##  -  0 crossreference edges
##  -  0 alternative edges
##  -  398 phenotype edges
##  -  The disNet was build based on 3 seeds
```

Reviewing

It may be required to review the returned *disNet* after building and/or extending it to assess whether all nodes are of interest. This process can be simplified by considering clusters of cross-references (nodes dealing with similar concepts) using the *cluster_disNet* functionality. Instead of reviewing each node, the different cross-reference clusters can be reviewed to identify those of interest while using the relationships between nodes to handle equivalent nodes simultaneously without the need to review them separately (Table 5). In addition, the process of using cross-reference clusters also allows the revision of identifiers that have no label information attached.

Visualizations

DODO is build as a meta-database incorporating several disease ontologies and their listed relationships. As disease concepts and definitions are not auto-generated but rather a human effort, concepts might not always be clearly defined or related to each other in a straightforward manner. The different ontologies employ hetergeneous definitions, cross-reference axes are not always exact, and errors present in the original ontologies will impact DODO as well. Using the *explore_disNet* and plotting function may increase understanding in how disease are defined across the different ontologies and visualize the relationship between different resources.

Connecting to external resources

The aim of DODO is to facilitate the connection with external resources. By defining a *disNet* around a disease of interest and annotating it with all its cross-references and/or child terms across different ontologies, the return will be an exhaustive network of disease identifier to connect to external resources. In addition, the use of a disNet across multiple resources ensures transitivity by defining a network of diseases allows tracing the connecting between these different resources more easily.

Below this use case with DODO will be exemplified by connecting to two different external resources: ClinVar and CHEMBL for “amyotropic lateral sclerosis” (ALS). To compare this usage, the resources themselves (ClinVar and CHEMBL) are queried directly for this indication and the results compared to using DODO.

The *disNet* is constructed by querying for the term “amyotrophic lateral sclerosis” on labels and synonyms provided in DODO. This disNet is then extended to use the transitivity mappings to obtain indirect cross-references. The CHEMBL resource using the ontology of EFO and MeSH to annotated compounds with indication, while ClinVar uses a variety of ontologies such as SNOMEDCT, MedGen/UMLS, Orphanet, OMIM. Therefore, we are interested in equivalent cross-references edges only and have no need for broader terms, so the parameter *intransitive_ambiguity* to put to one.

disNet

The *disNet* and consequent extension around “amyotrophic lateral sclerosis” returns 494 disease concepts across 25 ontologies. Figure 8 shows the retrieval of additional nodes not identified when querying directly for “amyotrophic lateral sclerosis” through either labels or synonyms. 251nodes match the query term directly in either label or synonym information attached to the nodes.

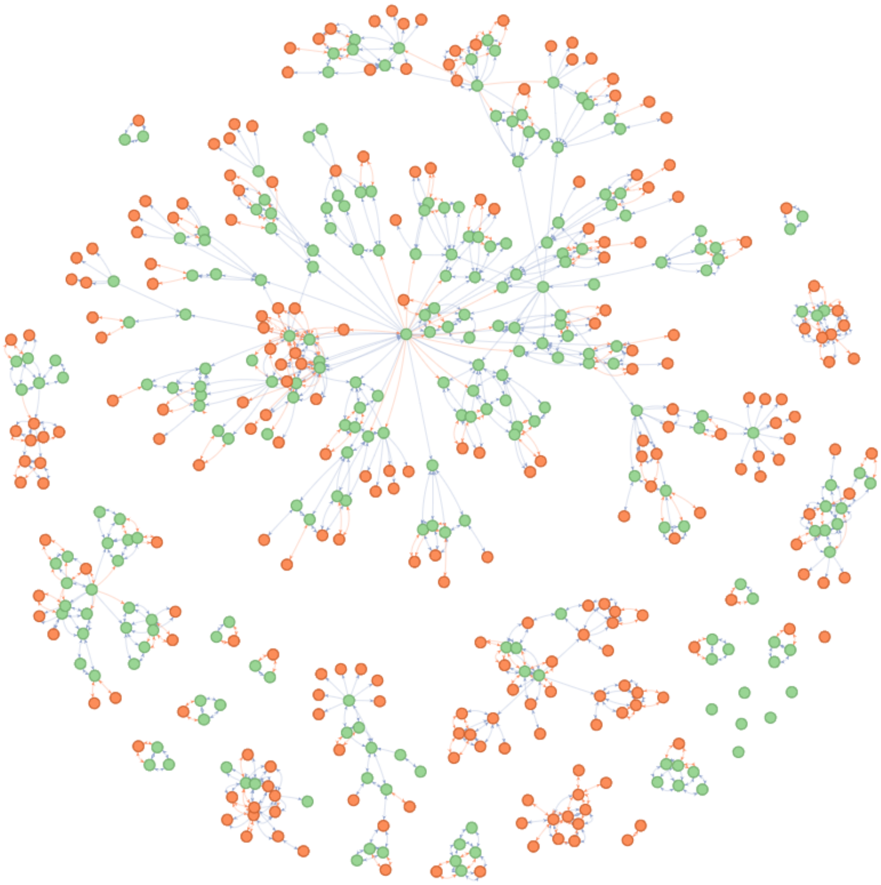

Figure 8: The disNet build on the term ’amyotrophic lateral sclerosisr querying both labels and synonyms provided in DODO (green nodes). This disNet is subsequenctly extended to return all cross-reference identifiers and child terms using the extend_disNet function (orange nodes) (parameters ’relations = c(’xref’, ’child’)’ and ’intransitive_ambiguity = 1’ to return only equivalent identifiers).

By using the disNet to connect to the CHEMBL resource, 96 unique compounds are identified as being documented for amyotrophic lateral sclerosis in the database for 4 different disease identifiers (Table 6).

Table 6: Using the disNet to connect to CHEMBL results identifies compounds available for different disease identifiers listed here.

dbid	name
MeSH:D000690	Amyotrophic Lateral Sclerosis
EFO:0000253	amyotrophic lateral sclerosis
ORPHA:98756	Spinocerebellar ataxia type 2
EFO:0001356	familial amyotrophic lateral sclerosis

Similarly, the *disNet* is used to identify disease variants reported for amyotrophic lateral sclerosis in ClinVar. `rlength(unique(clinvar_disnet$entrez))` unique Entrez gene identifiers are reported for 44 different disease identifiers (Table 7).

CHEMBL

There are 96 compounds identified through 3 disease identifiers when querying CHEMBL directly for amyotrophic lateral sclerosis. Both the direct query of CHEMBL and the use of the *disNet* identify the same drug compounds but an additional disease was available through use of a *disNet*.

```
##
##  EFO:0000253  EFO:0001356 MeSH:D000690  ORPHA:98756
##           95            2           97            2
```

```
##
##  EFO:0000253  EFO:0001356 MeSH:D000690
##           95            2           97
```

This disease is a child term identified through extending the *disNet* (Figure 9). This term 'ORPHA:98756' was identified as a child term of 'EFO:0001356' with the extension of the *disNet* using DODO. The disease could not be identified through querying CHEMBL directly as it's labelled 'Spinocerebellar ataxia type 2'". While this enriched extended network did not return additional compounds from CHEMBL in this example, the annotation of compounds used for treatment of ALS shows more granularity.

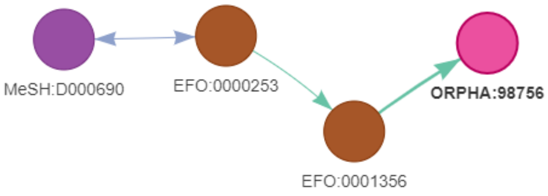

Figure 9: The visualized disNet shows the relations between the different diseases with compounds available in CHEMBL resource. This term 'ORPHA:98756' was identified as a child term of 'EFO:0001356' through extension (green edges). Cross-reference edge is_xref are indicated in blue. The disease could not be identified through querying CHEMBL directly as it's labelled 'Spinocerebellar ataxia type 2'. Using disease relationships present in DODO this node was identified additionally.

ClinVar

All entrez gene identifiers identified by querying the ClinVar resource directly for "amyotrophic lateral sclerosis" were also returned when using the disNet disease network. However, the use of the *disNet* around ALS returns 1 additional Entrez gene identifiers reported for ClinVar identifier ClinVar:18286 with label: Inclusion body myopathy with early-onset Paget disease with or without frontotemporal dementia 2. This identifier was identified through extending through cross-references edges through transitivity mapping related to "ORPHA:52430" which carries "Pagetoid amyotrophic lateral sclerosis" as a synonym (Figure 10).

Tracing connections

Understanding the relation between disease identifiers obtained when querying a resources directly through a search term is not a trivial thing. The question remains whether these identifiers are dealing with the same disease concepts. An 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. Figure 11 shows the original ALS extended disNet with disease identifiers matching in CHEMBL (orange) and those matching in ClinVar (blue). As both resources use different ontologies as references, there is a necessity to use cross-reference to understand their

Table 7: Using the disNet to connect to CHEMBL results identifies compounds available for different disease identifiers listed here.

id	label
ClinVar:16012	Amyotrophic lateral sclerosis 14, with or without frontotemporal dementia
ClinVar:38753	AMYOTROPHIC LATERAL SCLEROSIS, SUSCEPTIBILITY TO, 25
ClinVar:10103	Amyotrophic lateral sclerosis, typical
ClinVar:6638	Amyotrophic lateral sclerosis type 12
ClinVar:10438	Amyotrophic lateral sclerosis, susceptibility to
ClinVar:219	Amyotrophic lateral sclerosis type 5
ClinVar:38525	AMYOTROPHIC LATERAL SCLEROSIS 23
ClinVar:6240	Spinocerebellar ataxia 2
ClinVar:21297	Frontotemporal dementia and/or amyotrophic lateral sclerosis 2
ClinVar:36393	Amyotrophic Lateral Sclerosis, Recessive
ClinVar:10387	Amyotrophic lateral sclerosis 13
ClinVar:18288	Amyotrophic lateral sclerosis 20
ClinVar:297	Amyotrophic lateral sclerosis type 2
ClinVar:17308	Amyotrophic lateral sclerosis 17
ClinVar:1898	Amyotrophic lateral sclerosis and/or frontotemporal dementia 1
ClinVar:33549	Frontotemporal dementia and/or amyotrophic lateral sclerosis 3
ClinVar:4575	Amyotrophic lateral sclerosis type 8
ClinVar:32676	Amyotrophic lateral sclerosis 22 with frontotemporal dementia
ClinVar:2302	Inclusion body myopathy with early-onset Paget disease with or without frontotemporal dementia 1
ClinVar:239	Amyotrophic lateral sclerosis type 11
ClinVar:4157	Infantile-onset ascending hereditary spastic paralysis
ClinVar:393	Amyotrophic lateral sclerosis type 9
ClinVar:10104	Amyotrophic lateral sclerosis-parkinsonism/dementia complex 1, susceptibility to
ClinVar:16925	Amyotrophic lateral sclerosis 14 without frontotemporal dementia
ClinVar:15814	TARDBP-related frontotemporal dementia
ClinVar:3145	Amyotrophic lateral sclerosis 21
ClinVar:241	Amyotrophic lateral sclerosis type 4
ClinVar:32675	Amyotrophic lateral sclerosis 22 with or without frontotemporal dementia
ClinVar:36372	Amyotrophic Lateral Sclerosis, Dominant
ClinVar:38578	AMYOTROPHIC LATERAL SCLEROSIS, SUSCEPTIBILITY TO, 24
ClinVar:298	Amyotrophic lateral sclerosis type 6
ClinVar:238	Amyotrophic lateral sclerosis type 1
ClinVar:296	Amyotrophic lateral sclerosis type 10
ClinVar:9457	Amyotrophic lateral sclerosis
ClinVar:16560	Amyotrophic lateral sclerosis 15, with or without frontotemporal dementia
ClinVar:16701	Amyotrophic lateral sclerosis 16, juvenile
ClinVar:17352	Amyotrophic lateral sclerosis 18
ClinVar:18364	Amyotrophic lateral sclerosis 19
ClinVar:11395	Amyotrophic lateral sclerosis 1, autosomal recessive
ClinVar:9091	Amyotrophic lateral sclerosis 6, autosomal recessive
ClinVar:18286	Inclusion body myopathy with early-onset Paget disease with or without frontotemporal dementia 2
ClinVar:36408	Amyotrophic Lateral Sclerosis/Frontotemporal Dementia
ClinVar:33572	Frontotemporal dementia and/or amyotrophic lateral sclerosis 4
ClinVar:18287	Inclusion body myopathy with early-onset paget disease with or without frontotemporal dementia 3

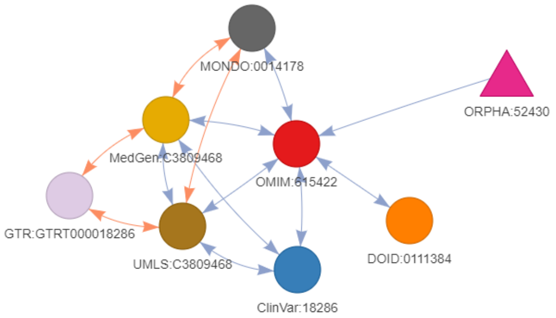

Figure 10: The visualized disNet shows how the identifier 'ClinVar:18286' ('Inclusion body myopathy with early-onset Paget disease with or without frontotemporal dementia 2') was returned through the use of transitivity on cross-reference edges starting from 'ORPHA:52430' ('Pagetoid amyotrophic lateral sclerosis'). With the is_xref edges are indicated in blue and the is_related edges indicated in orange.

relationship to each other. Indirect relationships are used and recorded when extending and can facilitate the understanding and integration of different biological resources.

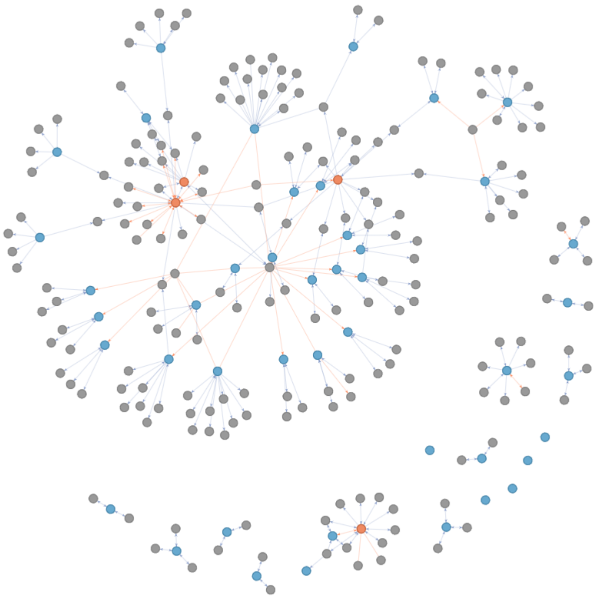

Figure 11: A additional feature from connecting resources through the use of a disNet, is the possibility to identify if and how diseases returned from each resource are connected to each other. This does not only allow a better understanding of disease, but also facilitates downstream analyses.

Conclusions

Disease ontologies have allowed a more formal classification of diseases. They facilitate the integration of biological databases thereby increasing disease understanding and supporting the development of 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 ongoing 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 standard bioinformatics platform (R or python)

DODO database and R package aims to tackle these two challenges by constructing a meta-database containing information on disease identifiers and their relationships across different ontologies. Through the transitivity mechanisms, the information across resources can be used to identify indirect cross-references. The R package contains several functions to build and interact with a disNet or convert concept identifiers between ontologies. While the workflow to construct a custom, local DODO database is provided with the intent to allow adaptation and addition, a docker image with the presented ontologies is provided for ease of use.

We believe that DODO can help clarify and define conditions of interest as well as aid in the understanding of relationships between disease concepts. We hope it improves accessibility of disease ontologies for a standard user. In addition, connecting different biological database (information on compounds, text mining, transcriptomics, etc.) through a disNet facilitates integration. It also ensures these resources are queried using 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

The source code for parsing disease ontologies are available at:

- https://github.com/Elysheba/Monarch
- https://github.com/Elysheba/EFO
- https://github.com/Elysheba/Orphanet
- https://github.com/Elysheba/MedGen
- https://github.com/Elysheba/MeSH
- https://github.com/patzaw/HPO
- https://github.com/patzaw/ClinVar
- https://github.com/Elysheba/DO
- https://github.com/Elysheba/ICD11

The source code for DODO is available at: https://github.com/Elysheba/DODO

A docker image of the DODO Neo4j instance is available at: https://hub.docker.com/repository/docker/elysheba/dodo (tag:02.04.2020) Software is available to use under a GPL-3 license

Competing interests

No competing interests were disclosed.

Grant information

This work was entirely supported by UCB Pharma. The authors declared that no grants were involved in supporting this work.

References

Almende B.V., Benoit Thieurmel, and Titouan Robert. visNetwork: network visualization using vis.js library, 2019. URL https://cran.r-project.org/package=visNetwork.

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.

Patrice Godard and Jonathan van Eyll. BED: A Biological Entity Dictionary based on a graph data model [version 2; referees: 2 approved]. *F1000Research*, 7:1–32, 2018. ISSN 1759796X. doi: 10.12688/f1000research.13925.2.

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.

Docker Inc. Docker Community Edition, 2019.

Neo4j Inc. Neo4j Community Edition, 2020.

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.

David Ochoa. EFO3: A community-driven ontology to advance clinical discoveries, 2019. URL https://blog.opentargets.org/2019/12/19/efo3-a-community-driven-ontology-to-advance-clinical-discoveries/.

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, Jim Hester, and Romain François. readr: Read Rectangular Text Data, 2018.

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.