UMLS-Query: A Perl Module for Querying the UMLS

Nigam H. Shah, MBBS, PhD, Mark A. Musen, MD, PhD
Center for Biomedical Informatics Research, Stanford University, Stanford, CA

Abstract

The Metathesaurus from the Unified Medical Language System (UMLS) is a widely used ontology resource, which is mostly used in a relational database form for terminology research, mapping and information indexing. A significant section of UMLS users use a MySQL installation of the metathesaurus and Perl programming language as their access mechanism. We describe UMLS-Query, a Perl module that provides functions for retrieving concept identifiers, mapping text-phrases to Metathesaurus concepts and graph traversal in the Metathesaurus stored in a MySQL database. UMLS-Query can be used to build applications for semi-automated sample annotation, terminology based browsers for tissue sample databases and for terminology research. We describe the results of such uses of UMLS-Query and present the module for others to use.

Introduction and Background

The Unified Medical Language System (UMLS) is a 20 year old project to aid the development of systems that help researchers retrieve and integrate electronic biomedical information from a variety of sources. The UMLS consists of 1) a Metathesaurus which inter-connects over 100 biomedical vocabularies, 2) the Semantic Network and 3) the SPECIALIST lexicon. Of these three resources, the Metathesaurus is the most widely used resource.

The UMLS Metathesaurus is a very large (1.37 million concepts), multi-purpose, and multi-lingual vocabulary database that contains information on biomedical and health related concepts, their various names, and the relationships among them. The Metathesaurus is unique in terms of providing alternative names and views of the same concept and identifying relationships between different concepts based on a union of the content from multiple source vocabularies.

According to the last UMLS user survey of 2677 licensees (1427 of whom responded) [1], 89% of UMLS users use it on Windows, 55% use Java and 25% use PERL. 35% use a MySQL installation of the Metathesaurus. Most users used it for processing of clinical information and most commonly to identify concepts for findings/diagnosis, procedures and lab tests. Java tools for accessing the Metathesaurus are easily available, but same is not true for Perl. With the increasing use of ontologies in bioinformatics, there is an increased interest in using the UMLS in Perl applications.

UMLS-Query is a PERL module to query a MySQL installation of the Metathesaurus on windows. UMLS-Query provides functions for retrieving identifiers for a user provided text string, mapping text-phrases to Metathesaurus concepts and graph traversal in the Metathesaurus. We describe each of these three groups of functions and then discuss the uses that UMLS-Query has enabled.

Methods

UMLS-Query provides functions for identifier retrieval, mapping text-phrases to concepts and graph traversal. All the functions can be restricted to particular source vocabularies or by relationship types in case of graph traversal.

Id retrieval functions

getCUI - this function accepts any text string, an atom unique identifier (aui), string unique identifier (sui) or lexical unique identifier (lui) and gets its corresponding concept unique identifier (cui). For example, calling this function with the string 'Malignant neoplasm of prostate' fetches CUI C0376358 as the result.

getSTR – this function accepts any concept unique identifier (cui), an atom unique identifier (aui), string unique identifier (sui) or lexical unique identifier (lui) and gets its corresponding string.

Both functions search for an exact match and can be restricted to a particular dictionary.

Text mapping functions

mapToId – this function accepts a phrase (up to 10 words) and maps it to an id type (aui, sui, lui, or cui); and can be restricted by a vocabulary if desired. The function first looks for an exact match for the phrase, if none is found, it will generate all possible permutations and attempt an exact match for each one. The function also performs right truncation to look for partial matches. For example, calling the function to find a CUI belonging to the SNOMED-CT for 'intraductal carcinoma of prostate' will return the results shown in the table below (Table 1).

Permutation	CUI	Retrieved String
carcinoma	C0007097	Carcinoma
intraductal	C1644197	Intraductal
prostate	C0033572	Prostate
carcinoma	C0600139	Carcinoma
intraductal	C0007124	Intraductal
prostate carcinoma	C0600139	Prostate carcinoma
carcinoma of prostate	C0600139	Carcinoma of prostate

Table 1. The table shows the output of calling the *mapToId* function using the phrase 'intraductal carcinoma of prostate'. The first column shows the different permutations that resulted in a match; the second column shows the CUI for the matching concept and the third column shows the preferred text string for that concept.

Permutation generation along with right truncation is conceptually similar to using skip n-grams for matching concepts. In fact, skip bigrams have been shown to perform at or above state of the art measures with less complexity, for the purpose of identifying matching concepts[2]

Graph traversal

The Metathesaurus combines the relationships reported in various source vocabularies into a unified view keeping track of the source that asserted a given relationship. The resulting graph of concepts and relationships is highly connected and can be traversed on the basis of different relationships types from one or more source vocabularies. The following functions in UMLS-Query provide this functionality.

getParents - this function accepts a cui or aui and returns its direct parent/s (nodes linked by the PAR relationship [3]) and all the ancestor nodes comprising the path till the root of the hierarchy. The function can optionally be restricted along a particular relationship type (*rela*, in the UMLS MRHIER table, which has 188 possible values) and a source vocabulary such as NCI or SNOMEDCT.

getCommonParent - This function accepts a pair of cuis or auis and returns the common parent; optionally restricted along a particular relationship type and a source vocabulary. The function returns the identifier of the common parent and the distance from each query node. For example, calling this function with CUIs C0376358 (Malignant neoplasm of prostate) and C0346554 (Carcinoma of genitourinary organ) as inputs, returns A0740023 (Malignant tumour) as the common parent and that it is one link from each of the CUIs C0376358 and C0346554.

getChildren - this function accepts a cui or aui and returns all its direct children, optionally restricted along a particular relationship type and a source vocabulary. Similarly *getCommonChild* returns the common child node of the query nodes. For example, calling the *getCommonChild* function with CUIs C0376358 (Malignant neoplasm of prostate) and C0346554 (Carcinoma of genitourinary organ) as inputs, returns C0600139 (Carcinoma of prostate (disorder)) as the common child using SNOMEDCT as the source vocabulary.

getDistBF - this function accepts two cuis and performs a breadth first search from cui-1 to find cui-2 and reports the number of links at which cui-2 is found. The search is aborted if cui-2 is not found in a radius of links specified by the maxR parameter (maxR is set to 3 as a default). For example, For example, calling this function with CUIs C0376358 (Malignant neoplasm of prostate) and C0346554 (Carcinoma of genitourinary organ) as inputs, returns the distance between them as two links.

Availability

UMLS-Query is free for academic use. The module is tested on windows XP and Vista and is provided with full documentation and sample scripts. The module is available from www.stanford.edu/~nigam/UMLS and will be submitted to CPAN.

Results

UMLS-Query provides a versatile set of functions making it relevant for a wide range of uses shown in figure 1. We group the uses into four categories as follows:

1) Computing conceptual distances – The graph traversal functions can be used to compute conceptual distance metrics such as those developed by Caviedes and Cimino [4] and by Melton et al [5].

Using functions implemented in UMLS-Query, we are currently evaluating the appropriateness of four different conceptual distance metrics[6] for the purpose of identifying 'related results' in searches made on the BioPortal, developed by the National Center for Biomedical Ontology.

2) Semi-automated sample annotation – We have used the functions in UMLS-Query to automatically map text annotations of database records to NCI thesaurus terms with a high degree of accuracy [7] as well as used the graph traversal functions to deploy a graphical browsing interface for the tissue samples using the NCI thesaurus.

The Stanford Tissue Microarray Database (TMAD) http://tma.stanford.edu is a public resource for disseminating annotated tissue images and associated expression data[8]. Databases such as the Stanford Tissue Microarray Database allow users to annotate their tissue samples using free text. Stanford University pathologists, researchers and their collaborators worldwide use TMAD for designing, viewing, scoring and analyzing their tissue microarrays. TMAD incorporates the NCI Thesaurus ontology for browsing and searching tissues in the cancer domain.

As different groups of researchers may annotate their samples differently, there is a need to map the tissue sample metadata to NCI Thesaurus terms. We have automatically mapped text annotations of tissue samples in TMAD to NCI thesaurus terms with a high degree of accuracy [7] The annotation of the ~20000 samples is done automatically using code based on the *mapToId* function of UMLS-Query.

Such annotation with ontology terms allows a rich querying facility and offers the ability to identify "similar" or "related" tissue microarray samples, even though they may be annotated with different terms. For example, neoplasms of the adrenal medulla and neoplasms of the adrenal cortex are all related to each other by the fact that they are all retroperitoneal neoplasms.

TMAD also provides a graphical browser to the full ontology with clickable links for browsing to more general or specific terms within the NCI trees[7]. The browser (figure 1) shows a live count of the TMAD tissues present by term. Clicking on a term brings up matching stained images. This browser is implemented using code based on the getParents function[1].

3) Data integration – The text-mapping and graph traversal functionality can be used to process descriptions of experimental samples to identify corresponding gene expression and protein expression data-samples from public datasets.

Currently, the predominant genomic level data is gene expression microarrays. Recently, Tissue Microarrays, which allow for the immunohistochemical analysis of large numbers of tissue samples and are used for confirmation of microarray gene-expression results are becoming more prevalent. A single tissue microarray (TMA) can contain as many as 500 different tumors, enabling the screening of thousands of tumor samples for protein expression[9]. Little attention is being paid to the problem of integrating the results from these – gene expression microarrays and tissue microarrays – complementary data types[9, 10]. Several reviews have suggested that it is essential to address this issue [9, 10]

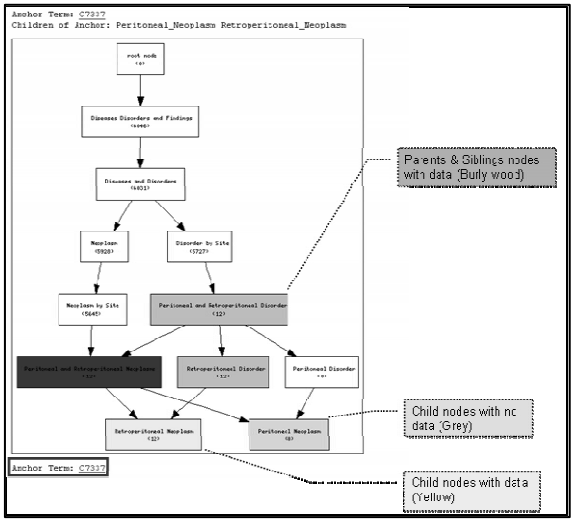

Figure 1. Semi-automated sample annotation and Graphical Browsing – the text descriptions of TMAD samples, are processed algorithmically to annotate them with ontology terms as well as browse them graphically. The graph shows a live count of the TMAD tissue samples corresponding to the selected term.

In order to develop approaches to integrate these datasets, we need to be able to identify all experiments that study a particular disease. The key query dimension for such integrative studies is the *sample*. However, because of the lack of a commonly used ontology or vocabulary to describe the diagnosis, disease studied or experimental agent applied in a given experimental dataset it is not possible to perform such a query.

The challenge is to create consistent terminology labels for each experimental dataset in the public repositories that can identify all samples that are of the same type at a given level of granularity. (e.g., *All carcinoma* samples versus *all Adenocarcinoma in situ of prostate* samples, where the former is at a coarser level of detail). Mapping the text-annotations describing the diagnoses, pathological state and experimental agents applied to a particular sample to ontology terms allowing us to formulate refined or coarse search criteria[11-13].

Recently, we have used the UMLS-Query to map existing text descriptions in TMAD and GEO to ontology terms[14]. In this mapping work, we identified

[1] The exact code bases are different because the implementation of TMAD and development of UMLS-Query occured in parallel.

45 disease related concepts for which there were datasets in both GEO and TMAD – and hence were potential candidates to support further analysis.

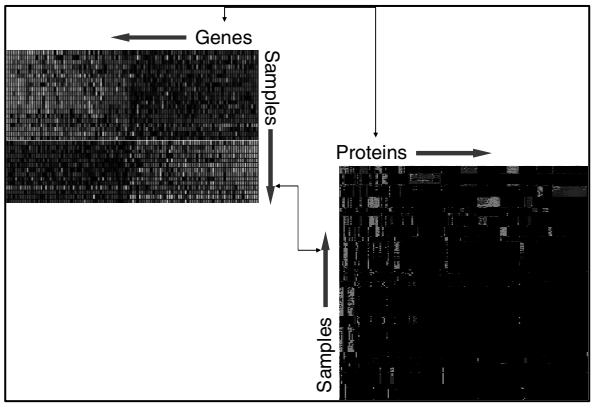

Figure 2. Dataset integration – Currently it is easy to identify all gene implicated in a process, such as cell death, using Gene Ontology terms but it is not easy to identify all datasets (samples) corresponding to a disease of a class of tumors such as retroperitoneal tumors. If datasets from multiple resources are annotated with ontology terms, queries to identify corresponding samples, from gene and protein expression datasets, for a given disease are enabled.

From this set of 45 matches, there are 23 disease related concepts that were at an appropriate level of granularity and have multiple samples in both GEO and TMAD to enable further integrative study (Table 2). Out of the 45 candidate datasets, 12 were high level terms such as *Cancer, Syndrome, and Sarcoma*. We consider these uninformative for the purpose of matching up disease related datasets across repositories. Counting such high level matches as false positives, we obtain a precision of 73% for identifying datasets for further integrative study[14].

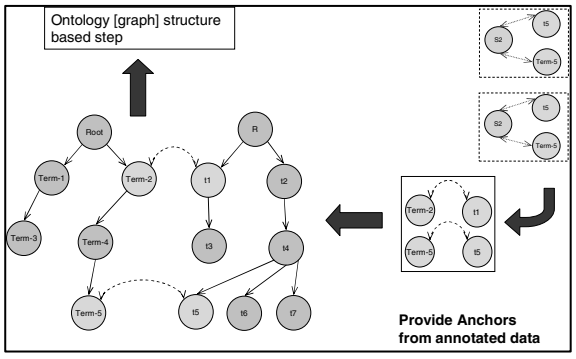

Figure 3. Terminology research – the text-mapping function (and its extensions) can map terms from different terminologies onto UMLS concepts for the purpose of aligning the terminologies. Moreover, samples annotated with the terms from different ontologies serve as potential anchor points to drive terminology alignment using graph based methods.

4) Terminology research – The text-mapping function (and its extensions) can map terms from different terminologies onto UMLS concepts for the purpose of aligning the terminologies. Just as UMLS curators map atomic strings from different terminologies to common concepts, an automated procedure can map terms from other ontologies to create draft alignments to the UMLS.

In fact alignment can also come as a byproduct of automatically annotating a large number of samples with terms from multiple ontologies. During the process of mapping described in our previous work [7], we acquire information that can be used to align terms from the two ontologies. For example, during the process of mapping the sample descriptions to the NCI and SNOMED-CT, we find samples annotated with terms from the two ontologies. These dually annotated samples serve as evidence to 'anchor' the two terms (from the two different ontologies) as candidate alignment points. Subsequently, algorithms like Anchor-Prompt [15] can be invoked with these anchors to derive a computationally generated alignment between two ontologies.

In case of the TMAD, 3208 samples were annotated by *both* NCI thesaurus and SNOMED-CT terms. Analysis of these terms showed that for 2810 samples these terms were appropriately aligned as evidenced by their identical (or very close) CUIs in the UMLS.

Conclusion

Based on the UMLS user survey, we believe there is a need for a PERL programming interface to the MySQL installation of the UMLS and we have developed such a Perl Module called UMLS-Query. We have used this module in several applications that have been peer reviewed and published on. We have described the key functionality of UMLS-Query, the different ways in which we have used it; and present the module for others to use.
We believe that as the interactions between bioinformatics and medical informatics increase[16, 17], providing easy access to the UMLS (a crucial medical informatics resource) in a programming language of choice of the bioinformatics community is required; and UMLS-Query accomplishes that objective.

Acknowledgements

This work was funded by NIH grant U54 HG004028.

References

1 Fung KW, Hole WT, Srinivasan S. Who is Using the UMLS and How - Insights from the UMLS User Annual Reports. AMIA Annual Symposium; 2006; Washington, DC; 2006. p. 274-8.

2 Reeve LH, Han H. CONANN: An Online Biomedical Concept Annotator. LECTURE NOTES IN COMPUTER SCIENCE. 2007;4544:264.

3 NLM. UMLS Metathesaurus Documentation. 2006 [cited 2006 Dec]; Available from: http://www.nlm.nih.gov/research/umls/meta2.html

4 Caviedes JE, Cimino JJ. Towards the development of a conceptual distance metric for the UMLS. J Biomed Inform. 2004 Apr;37(2):77-85.

5 Melton GB, Parsons S, Morrison FP, et al. Inter-patient distance metrics using SNOMED CT defining relationships. J Biomed Inform. 2006 Dec;39(6):697-705.

6 Lee WN, Shah NH, Sundlass K, et al. Comparison of Ontology-based Semantic-Similarity Measures. AMIA Annual Symposium; 2008; Washington, D.C.; 2008. p. accepted.

7 Shah NH, Rubin DL, Supekar KS, et al. Ontology-based Annotation and Query of Tissue Microarray Data. AMIA Annual Symposium; 2006; Washington, DC; 2006. p. 709-13.

8 Marinelli RJ, Montgomery K, Liu CL, et al. The Stanford Tissue Microarray Database. Nucleic Acids Res. 2008 Jan;36(Database issue):D871-7.

9 Rimm DL, Camp RL, Charette LA, et al. Tissue microarray: a new technology for amplification of tissue resources. Cancer J. 2001 Jan-Feb;7(1):24-31.

10 Basik M, Mousses S, Trent J. Integration of genomic technologies for accelerated cancer drug development. Biotechniques. 2003 Sep;35(3):580-2, 4, 6 passim.

11 Spasic I, Ananiadou S, McNaught J, et al. Text mining and ontologies in biomedicine: making sense of raw text. Brief Bioinform. 2005 Sep;6(3):239-51.

12 Moskovitch R, Martins SB, Behiri E, et al. A Comparative Evaluation of Full-text, Concept-based, and Context-sensitive Search. J Am Med Inform Assoc. 2007 March-April;14(2):164-74.

13 Shah NH, Rubin DL, Espinosa I, et al. Annotation and query of tissue microarray data using the NCI Thesaurus. BMC Bioinformatics. 2007;8:296.

14 Shah NH, Chiang AP, Butte AJ, et al. Ontology-driven Indexing of Public datasets for Translational Bioinformatics. *AMIA 2008 STB Submission*. Stanford 2007.

15 Noy NF, Musen MA. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. International Journal of Human-Computer Studies. 2003;59(6):983-1024.

16 Altman RB. The interactions between clinical informatics and bioinformatics: a case study. J Am Med Inform Assoc. 2000 Sep-Oct;7(5):439-43.

17 Altman RB, Klein TE. Challenges for biomedical informatics and pharmacogenomics. Annual review of pharmacology and toxicology. 2002;42:113-33.