

# UMLS-Query: A Perl Module for Querying the UMLS

Nigam Shah<sup>\*1</sup> and Mark A. Musen<sup>1</sup>

<sup>1</sup>Stanford Medical Informatics, Stanford University School of Medicine, Stanford, CA 94002

## 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. 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.

**Availability:** [www.stanford.edu/~nigam/UMLS](http://www.stanford.edu/~nigam/UMLS)

**Contact:** [nigam@stanford.edu](mailto:nigam@stanford.edu)

## 1 INTRODUCTION

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 a Metathesaurus which inter-connects over 100 biomedical vocabularies, the Semantic Network and the SPECIALIST lexicon. Of these three resources, the Metathesaurus is the most widely used resource. According to the latest UMLS user survey (Fung, et al.), 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.

## 2 RESULTS

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. 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). *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.

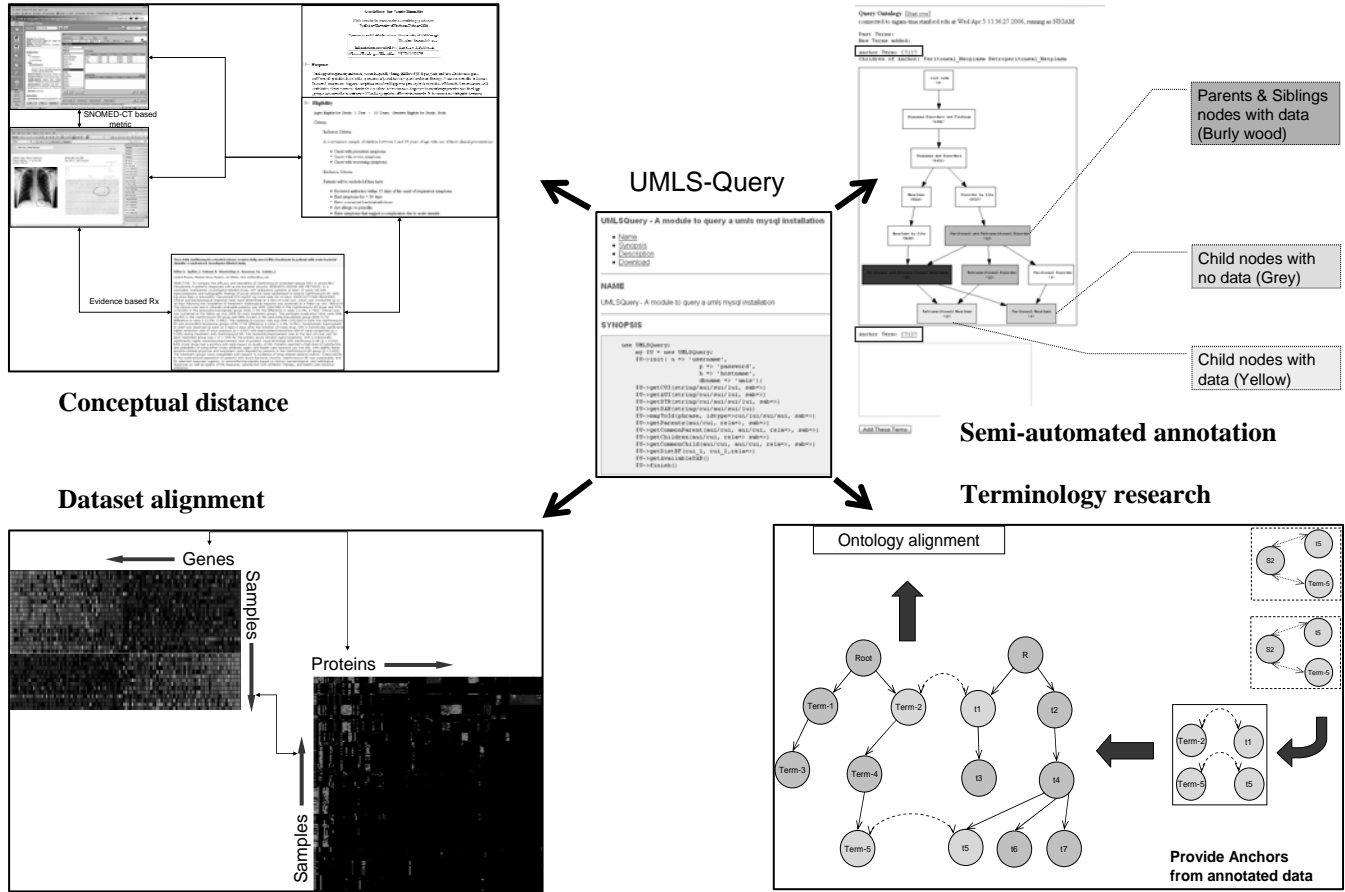
Text mapping functions: *mapTold* - this function accepts a phrase (up to 10 words) and maps it to an id type (aui,sui,lui, or cui); and can be restricted by a vocabulary if desired. The function first looks for an exact match for the phrase, if none is found, it will generate all possible permutations and attempt an exact match for each one (with right truncation of words 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 table 1.

Graph traversal: *getParents* - this function accepts a cui or aui and returns its direct parent/s (nodes linked by the PAR relationship (NLM)) 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. *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 aui of the common parent and the distance from each query node. *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. *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).

**Table 1** Results of the text mapping function

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

<sup>\*</sup>To whom correspondence should be addressed.



**Figure 1** – The UMLS-Query module can be used for: 1) Mapping text-phrases to UMLS concepts for purpose of ontology alignment in terminology research. 2) Building applications for semi-automated annotation. 3) Processing descriptions of public datasets in order to align them using UMLS CUIs. 4) Computing conceptual distance metrics.

### 3 DISCUSSION

UMLS-Query provides a versatile set of functions making it relevant for a wide range of uses shown in figure 1. Briefly, we consider these to be 1) 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. 2) Semi-automated sample annotation – Databases such as the Stanford Tissue Microarray Database use a controlled vocabulary to annotate their tissue samples. We have used the functions in UMLS-Query to automatically map these annotations to NCI thesaurus terms with a high degree of accuracy (Shah, et al.) as well as used the graph traversal functions to deploy a graphical browsing interface for the tissue samples using the NCI thesaurus. 3) Dataset alignment – 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. 4) Building distance metrics – The graph traversal functions can be used to compute conceptual distance metrics developed by Caviedes and Cimino (Caviedes and Cimino) and by Melton et al (Melton, et al.).

### ACKNOWLEDGEMENTS

We would like to acknowledge Daniel Rubin and Natasha Noy for useful discussions that led to this work. This work was funded by NIH grant U54 HG004028

### REFERENCES

Caviedes, J.E. and Cimino, J.J. (2004) Towards the development of a conceptual distance metric for the UMLS, *J Biomed Inform.* **37**, 77-85.  
 Fung, K.W., Hole, W.T. and Srinivasan, S. (2006) Who is Using the UMLS and How - Insights from the UMLS User Annual Reports. *AMIA Annual Symposium*. Washington, DC, 274-278.  
 Melton, G.B., Parsons, S., Morrison, F.P., Rothschild, A.S., Markatou, M. and Hripscak, G. (2006) Inter-patient distance metrics using SNOMED CT defining relationships, *J Biomed Inform.* **39**, 697-705.  
 NLM (2006) UMLS Metathesaurus Documentation. NLM.  
 Shah, N.H., Rubin, D.L., Supekar, K.S. and Musen, M.A. (2006) Ontology-based Annotation and Query of Tissue Microarray Data. *AMIA Annual Symposium*. Washington, DC, 709-713.