# Predicate Based Mapping Generation from NCI-META

# Overview

This document describes a general methodology for algorithmically generating mappings from a resource like NCI-META (or NLM’s UMLS Metathesaurus) using a criteria- or predicate-based approach. The later sections describe the specific details of a particular use of this methodology to create two sets of mappings: one between NCIt “neoplasms” and SNOMEDCT and another between NCIt “neoplasms” and ICD9CM.

# General Approach

Identifying the nature and use cases of the mappings to be generated is a good starting point. In our conceptual model, mappings exist between elements of one source vocabulary and elements of a target vocabulary. For the purposes of this approach, we are only considering *source code* to *target code* mappings (and not considering mappings between other types of identifiers or expressions of identifiers, or of free text). Portions of a source or target vocabulary can be selected (based on hierarchies, or semantic types, or other criteria) to participate in mappings (on either the *to* or *from* side), rather than trying to comprehensively map from or map to an entire vocabulary. Understanding how a mapping set will be used can guide decisions about what portions of a vocabulary should be considered for mapping.

The process of generating mappings begins by delineating the various ways in which candidate mappings might be obtained for the intended purpose. In general, every available means of finding candidates should be leveraged at this point as we can cull weaker or poorer quality matches at a later time. Thus, when using a resource like NCI-META, we can consider any possible connection between the source and target vocabularies (or portions of vocabularies) being mapped, including: synonymy, hierarchical relations, associative relations, semantic type connections, lexical matches, non-lexical semantic matching (like reasoning), and even borrowing information from known preexisting mapping sets.

As each successive level of candidate selection is performed, a determination should be made as to whether its results should be mutually exclusive with respect to previous selection criteria. For example, much stronger types of connections (like synonymy) should probably be identified before weaker kinds (like lexical matching) and candidates based on weaker criteria for which there are already stronger criteria should probably be ignored. Whether to keep all candidate mappings (and later order them by rank) depends on how the mapping set is going to be used by an application. For use cases that require greater precision, more consideration should be given to mutual exclusivity.

After all known or expected approaches to candidate selection have been applied, it is useful to review the cases of unmapped source vocabulary codes to determine what other kinds of connections they may have to target codes. For example, *parent*/*child* relations may have been used to create *broader/narrower* mappings between the source and target, but chaining *parent/child* together (to yield *grandparent/grandchild*) may be an additionally useful way to find candidate mappings. This is, of course, use-case dependent. For example, the *grandparent/grandchild* approach can be especially useful when the level of granularity of the source vocabulary is much finer than the target vocabulary and the mappings are to be used for data aggregation (for some kind of reporting).

Once a pool of candidate mappings has been found, the next step is to characterize and rank them according to relevant criteria. Relevancy depends on the use case of the eventual map set. The general idea is to characterize each criterion as positive or negative and then assign it a relative weight. A ranking score can then be computed by summing the various criteria scores yielding a sense of the overall (and relative) quality of a particular candidate.

With each candidate mapping ranked, we then have the capability to introduce a threshold score, below which mappings are not included in the final result set. We can also use ranking scores to define different kinds of output sets. For example, we could provide all mappings with their scores, allowing applications to choose dynamically (or interactively) among the various options, or we could provide a set of only the highest ranking mappings, thus providing unique (1:1) set of mappings where precision (highest quality) is required.

At this point, candidate mappings are identified, filtered, and ranked. We have determined exactly those records that should be included. The remaining step is to choose an output format. For the purposes of this work, we are focusing mostly on the UMLS Metathesaurus MRMAP.RRF file format, as it provides a high degree of flexibility and is the basis for the LexEVS 6.0 mapping model specification. Simpler forms of this (that include only source code, relationship, target code, and rank (with PT names) have also been considered and may be employed for sharing of examples to avoid the full complexity of the MRMAP view.

# Characterizing Mapping Sets

There are a variety of different reasons for creating mappings and different ways in which they may be used by an application. Some exist to be “generally useful” while others are expected to have a sufficient level of precision to support operations like (semi)automated reasoning. It is important to understand both the use-case(s) of a particular map set as well as the level of rigor required of it. Some common mapping use cases include:

* Clinical Records -> Service Reimbursement
* Lab Results Reporting -> Billing
* Clinical Records -> Adverse Drug Reaction Reporting
* Data aggregation
* Decision Support

Some use cases will require maximum recall (favoring more, potentially lower-quality mappings) and others will require maximum precision (favoring fewer, higher-quality mappings). Both use case and level of precision/rigor should be considered before choosing selection and ranking criteria. It will help understand the best algorithms to use.

# Candidate Selection Predicates

Candidate mappings between source codes and target codes can arise from whatever information exists about the connections between the two sets of things. This is the basic set of predicates to consider:

* **Synonymy**: source and target code mean the same thing through either sharing of a third-party concept class identifier or explicit *synonymy* relation (e.g. sharing an NCI-META CUI).
* **Hierarchical Relations**: source code is synonymous (in the sense above) with a parent or child code of a target code. Transitivity can be considered as well (e.g. grandparent/grandchild). This approach helps harmonize differing levels of granularity.
* **Associative Relations**: source code has an associative relationship to another source code that is synonymous (in the sense above) with a target code.
* Lexical Match: here a source code and a target code have names that share a lexical similarity (either same string, or same output from a normalization operation).
* **Semantic Non-lexical Matches**: if source and target terminologies are ontologies, reasoning operations can be used to derive inferred connections that may manifest as synonymy, hierarchical, or associative relations.
* **Non-synonymous Third-Party Classification**: source and target codes are not directly synonymous but each share space with some other kind of third party code or class id. For example, an NCIt code and a SNOMEDCT code that both shared CUIs in NCI-META with the same MeSH descriptor (D#). In that case, the combination of NCI-META CUI and D# suggest a connection between the NCIt code and SNOMEDCT code that could be relevant given certain use cases.

These predicates can be parameterized in a number of different ways. We can specify the source of third party class identifiers, the types of associative relations, the source of hierarchical or associative relations, etc. As such, a complete algorithm may incorporate more than one instance of a type of selection predicate, where each one had a different set of parameters.

When choosing candidate selection predicates, it is important to order your choices. The order and types of choices are somewhat use case dependent. When ordering choices, you also have the opportunity to determine which predicates are mutually exclusive of other predicates. For example, you may want a synonymy predicate that overrides all other predicates, meaning that any code mapped through a synonymy predicate is no longer a candidate for another kind of predicate. Alternatively, you could use purely additive predicates (append-only mode), to attempt to identify all possible mappings that a source code might have. In that case, ranking algorithms would become especially important.

# Candidate Qualifying and Ranking Predicates

Candidate mappings can be of higher or lower quality depending upon various criteria. This is very use case specific and dependent upon the kind of third-party database used to derive the mappings. Finding useful and meaningful ranking predicates may involve significant research into the data to understand what makes a connection in a particular data set stronger or weaker. Application of ranking predicates can be automated, while the process of choosing them involves manual intervention.

Using NCI-META style database as the data set suggests a number of useful ranking predicates. Those predicates that contribute to a lower quality ranking include:

* Source code is split or merged (e.g. with respect to an NCI-META CUI) -
* Target code is split or merged (e.g. with respect to an NCI-META CUI) -
* Discrepancies in levels of granularity (e.g. *parent/child* is better than *grandparent/grandchild*)
* Same-concept PTs.

Those predicates that contribute to a higher quality ranking include:

* Candidate mapping has close association with preferred terms of the source/target codes.
* Semantic types (either UMLS-style or like the top-levels of the SNOMEDCT hierarchy) are aligned for the kinds of candidates we are attempting to discover.

Any and all other reasons why a candidate mapping is of higher or lower quality should be considered. Each predicate can be weighted differently – they do not all have to contribute +1 or -1. It may require some research or experimentation to devise the exact weighting and formula for computing a rank from the various predicates.

Once each candidate mapping is ranked according to this mechanism, consider applying a threshold filter. Determine whether or not a there is a score threshold below which mappings cease to be useful (for the given use case). If such a score exists, consider removing any mappings at or below that score.

# Format Considerations

For this approach, the main format considerations are:

* The information should be easily shared in a way that is human readable.
* The end products of the generation process can be easily loaded into LexEVS.

This suggests that potentially two formats should be used: one that is a supported LexEVS load format (e.g. MRMAP or the corresponding LexEVS XML representation), and a second that is human readable, in the form of an Excel spreadsheet with preferred term information included.

Other formats could be considered as needed.

# NCIt “Neoplasms” to SNOMEDCT Mappings

Our first exercise/pilot in applying this methodology was to map from the set of source codes comprised by NCI Thesaurus neoplasms (those things that are descendents of the “Neoplasm” concept – C3262) and SNOMEDCT concepts (from any part of the tree for now).

The use case was conceived of as generally useful data mapping and aggregation. Thus the mappings could be used to transform any records referencing NCIt neoplasm codes into SNOMEDCT and aggregate at the coarser level of granularity within SNOMEDCT.

We used the following candidate selection predicates (in order):

1. Synonymy based on NCI-META CUIs.
2. PAR/CHD relationships from NCIt in conjunction with NCI-META CUIs. Mutually exclusive from #1.
3. Transitive PAR/CHD relationships from NCIt (grandparent/grandchild relationships). Mutually exclusive from #1 and #2.

We then used the following candidate ranking predicates:

* Starting score: 5
* (-1 point): NCIt code of mapping is split across multiple NCI-META CUIs
* (-1 point): NCIt code of mapping is merged within NCI-META CUIs
* (-1 point): SNOMEDCT code of mapping is split across multiple NCI-META CUIs
* (-1 point): SNOMEDCT code of mapping is merged within NCI-META CUIs
* (+1 point): NCI-META CUIs involved in selection criteria for NCIt and SNOMEDCT code both contain the PTs for those codes
  + E.g. in the synonymy case, it means the NCIt PT and the SNOMEDCT PT were both in the same NCI-META CUI. In the PAR/CHD case, it means the SNOMEDCT PT was in the same concept as the parent NCIt PT.

You might imagine grandparent/grandchild should be scored as well (with a -1). Given that those mappings are mutually exclusive from the other sets and are generated last, the lowered ranking would not actually help differentiate them from other potential candidates.

We did not use a threshold score; we allowed all candidate mappings to remain in the result.

For output, we chose an Excel spreadsheet with four tabs

1. MRMAP.RRF-style representation of ALL entries (where the rank is modeled as MAPRANK).
2. A representation mapping back to fields in the suggested LexEVS specification for mappings (so that one could see how the data looks in the potential LexEVS 6.0 mapping model)
3. Same as #1 except only the mappings with the max(rank) for each NCIt code were kept (lower ranking ones were discarded).
4. Same as #2 except only the mappings with the max(rank) for each NCIt code were kept (lower ranking ones were discarded).

It may be worth considering a 5th and 6th page that would include the preferred terms of each code, to make the result more human readable. Alternatively, we could replace the LexEVS 6.0 model tables with the simplified form.

*Statistics*

* Total number of codes in the "source" set (e.g. NCI Neoplasms): 7503
* Total number of codes in the "target" set (e.g. SNOMEDCT): 2877
* Total number of NCIt codes with mappings: 6968
* Cases of NCIt codes with exactly 1 mapping: 3325
* Cases of NCIt codes with >1 mapping: 3643
* Cases of NCIt codes with zero mappings: 535
* Tally of mappings by relationship
* 10254 narrower
* 3062 synonymous
* 1710 broader

# NCIt “Neoplasms” to ICD9CM Mappings

Our second exercise/pilot in applying this methodology was to map from the set of source codes comprised by NCI Thesaurus neoplasms (those things that are descendents of the “Neoplasm” concept – C3262) and ICD9CM concepts (from any part of the tree for now).

We assumed that the use cases are similar to that of the NCIt to SNOMEDCT mappings, so we used the same candidate selection predicates. (in order):

1. Synonymy based on NCI-META CUIs.
2. PAR/CHD relationships from NCIt in conjunction with NCI-META CUIs. Mutually exclusive from #1.
3. Transitive PAR/CHD relationships from NCIt (grandparent/grandchild relationships). Mutually exclusive from #1 and #2.

We then used the same candidate ranking predicates:

* Starting score: 5
* (-1 point): NCIt code of mapping is split across multiple NCI-META CUIs
* (-1 point): NCIt code of mapping is merged within NCI-META CUIs
* (-1 point): ICD9CM code of mapping is split across multiple NCI-META CUIs
* (-1 point): ICD9CM code of mapping is merged within NCI-META CUIs
* (+1 point): NCI-META CUIs involved in selection criteria for NCIt and ICD9CM code both contain the PTs for those codes
  + E.g. in the synonymy case, it means the NCIt PT and the ICD9CM PT were both in the same NCI-META CUI. In the PAR/CHD case, it means the ICD9CM PT was in the same concept as the parent NCIt PT.

We did not use a threshold score; we allowed all candidate mappings to remain in the result.

For output, we again chose an Excel spreadsheet with four tabs

1. MRMAP.RRF-style representation of ALL entries (where the rank is modeled as MAPRANK).
2. A representation mapping back to fields in the suggested LexEVS specification for mappings (so that one could see how the data looks in the potential LexEVS 6.0 mapping model)
3. Same as #1 except only the mappings with the max(rank) for each NCIt code were kept (lower ranking ones were discarded).
4. Same as #2 except only the mappings with the max(rank) for each NCIt code were kept (lower ranking ones were discarded).

*Statistics*

* Total number of codes in the "source" set (e.g. NCI Neoplasms): 7503
* Total number of codes in the "target" set (e.g. ICD9CM): 406
* Total number of NCIt codes with mappings: 3184
* Cases of NCIt codes with exactly 1 mapping: 2205
* Cases of NCIt codes with >1 mapping: 979
* Cases of NCIt codes with zero mappings: 4319
* Tally of mappings by relationship
* 3765 narrower
* 456 broader
* 337 synonymous
* 21 other