# MetaMap 2011 Release Notes

June 6, 2011

**MetaMap2011 includes some significant enhancements, most notably algorithmic improvements that enable MetaMap to very quickly process input text that had previously been computationally intractable.**

## 1 Algorithmic Improvements

MetaMap has always emphasized thoroughness over efficiency; recently, however, processing efficiency has become a significant concern for many members of the MetaMap user community, who have provided feedback about pieces of text that caused MetaMap to run for an extremely long time (many hours or even days) and/or run out of memory

A brief explanation of the computational issues: MetaMap first identifies candidates (UMLS concepts) mentioned in the input text, and then creates its final mappings by choosing appropriate candidates that cover as much of the input text as possible. Because each mapping is a subset of the set of candidate concepts, the set of mappings is a subset of the powerset of the candidate set. Given $N$ candidates, this computationally intensive process can in the worst case generate $2^N$ mappings, although in actual practice, the number of mappings is far smaller (although in some cases still in the millions).

To explain the algorithmic improvements, we first introduce the notion of *duplicate candidates*; we next present some information about MetaMap's mapping-construction algorithm (more details can be found here); and finally we explain how identifying duplicate candidates leads to a dramatic improvement in the efficiency of mapping construction.

### 1.1 Duplicate Candidates

Duplicate candidates are candidate concepts with **identical phrase coverage** and **candidate scores**. For example, from the text `inferior vena caval filter`, MetaMap identifies the following twenty-one candidates:

```
1.    976 Filter, Inferior Vena Cava (Vena Cava Filters) [Medical Device]
2.          Cava Filter, Vena
3.    812 Filter (Filters) [Manufactured Object]
4.    812 Filter (Optical filter) [Medical Device]
5.    812 Filter (filter information process) [Intellectual Product]
6.    812 Filter (Filter (function)) [Conceptual Entity]
```

```
 7.    812 Filter (Filter Device Component) [Medical Device]
 8.    812 FILTER (Filter - medical device) [Medical Device]
 9.    756 Inferior vena caval [Body Location or Region]
10.    733 Inferior Vena Cavas (Inferior vena cava structure) [Body Part, Organ, or Organ Co
11.    721 Inferior vena cava (Entire inferior vena cava) [Body Part, Organ, or Organ Compon
12.    694 Vena caval (Vena cava structure) [Body Part, Organ, or Organ Component]
13.    645 Vena (Structure of vein of trunk) [Body Part, Organ, or Organ Component]
14.    645 Inferior [Spatial Concept]
15.    645 inferior (inferiority) [Social Behavior]
16.    623 Vena cava (Entire vena cava) [Body Part, Organ, or Organ Component]
17.    612 Venae (Veins) [Body Part, Organ, or Organ Component]
18.    579 Kava [Plant]
19.    579 KAVA (Kava preparation) [Organic Chemical,Pharmacologic Substance]
20.    579 CAVA (CA5A gene) [Gene or Genome]
21.    579 Kava (Kava Use Code) [Intellectual Product]
```

This list of candidates includes several sets of of duplicate candidates:

1. two candidates identified from `inferior`:

```
14.    645 Inferior [Spatial Concept]
15.    645 inferior (inferiority) [Social Behavior]
```

2. four from `cava`:

```
18.    579 Kava [Plant]
19.    579 KAVA (Kava preparation) [Organic Chemical,Pharmacologic Substance]
20.    579 CAVA (CA5A gene) [Gene or Genome]
21.    579 Kava (Kava Use Code) [Intellectual Product]
```

3. and finally six from `filter`:

```
 3.    812 Filter (Filters) [Manufactured Object]
 4.    812 Filter (Optical filter) [Medical Device]
 5.    812 Filter (filter information process) [Intellectual Product]
 6.    812 Filter (Filter (function)) [Conceptual Entity]
 7.    812 Filter (Filter Device Component) [Medical Device]
 8.    812 FILTER (Filter - medical device) [Medical Device]
```

The candidates in each of the groupings above are duplicate concepts because they have the same phrase coverage and received the same candidate score.

### 1.2 Mapping Construction

MetaMap's mappings are sets of candidates that maximize the coverage of the input phrase being analyzed. MetaMap has in the past created mappings from the full candidate set, and then deleted a mapping M1 if it is subsumed by another mapping M2, i.e., if M2 has broader phrase coverage than M1. Subsumption checking is quadratic in the number of mappings, because each of $n$ mappings must be checked, on average, against $n/2$ other mappings, which makes subsumption checking $O(n^2)$.

### 1.3 Efficiency Improvement

Mappings that differ only by duplicate candidates need not be checked for subsumption, because they will have equivalent phrase coverage, so in MetaMap2011, we reduce the number of mappings that must be checked for subsumption by temporarily ignoring all but one candidate from each set of duplicates before constructing the mappings that are checked for subsumption. In the above example, mappings are constructed from only twelve candidates instead of the full set of twenty-one because we temporarily ignore

- one of the candidates mapped from `inferior`,
- three mapped from `cava`, and
- five mapped from `filter`.

Mappings surviving the subsumption check are then duplicated using the full set of duplicate candidates. The subsumption algorithm is unchanged and still quadratic, but it is now based on a far smaller $n$, resulting in substantial efficiency gains observed while analyzing texts that generate large number of candidates.

### 1.4 Results

When analyzing text that generates such a small number of mappings as the above example, the efficiency improvements will not be noticeable; however, while processing the entire 2011 MEDLINE baseline with MetaMap2010, we encountered 116 citations that each ran for over six hours on a 3GHz Linux platform before processing was manually terminated. With our algorithmic improvements, 115 of these 116 citations now run in approximately 15 seconds—a speedup of over 1400 fold, or about 140,000%, for such extremely long-running citations. This speedup will be far more modest, if it is observable at all, for most well-behaved citations, but citations that generate a large number of candidates can now be processed far more efficiently. This algorithmic improvement is built into MetaMap2011, invoked automatically, and not subject to user control.

## 2 Pruning the Candidate Set

In spite of the efficiency improvements described above, certain pathological cases remain computationally challenging. For example, the following text (from PMID 10931555)

```
protein-4 FN3 fibronectin type III domain GSH lutathione GST
glutathione S-transferase hIL-6 human interleukin-6 HSA human
serum albumin IC(50) half-maximal inhibitory concentration Ig
immunoglobulin IMAC immobilized metal affinity chromatography
K(D) equilibrium constant
```

parses to a single phrase that generates 98 candidates. Creating mappings from 98 candidates will exceed most users' computational resources—not to mention their patience. In order to enable MetaMap to generate (perhaps suboptimal) mappings from text such as the above in a reasonable amount of time, we have implemented a mechanism of candidate pruning, which reduces the number of candidates used to construct mappings.

The candidate-pruning mechanism makes up to five passes through the candidate list; each pass examines candidates from highest to lowest scoring, and employs increasingly stringent exclusion criteria. If one pass prunes out enough candidates, the remaining passes are are not made. Suppose we want to limit the number of candidates to $N$:

Pass 1: Exclude all candidates whose phrase coverage is narrower than any previous examined candidate's. If $N$ or fewer candidates remain, stop.

Pass 2: Exclude all candidates whose phrase coverage is narrower than the aggregate phrase coverage of *all* previously examined candidates If $N$ or fewer candidates remain, stop.

Pass 3: Exclude all candidates whose phrase coverage is narrower *or the same* as the aggregate phrase coverage of all previously examined candidates If $N$ or fewer candidates remain, stop.

Pass 4: Exclude all candidates whose phrase coverage overlaps the aggregate phrase coverage of all previously examined candidates If a candidate's phrase coverage overlaps any previously examined candidate's, exclude it. If $N$ or fewer candidates remain, stop.

Pass 5: Finally, simply exclude all but the first $N$ candidates, as well as any subsequent candidates whose score is equal to that of the $N$th candidate.

Our experience has revealed that allowing MetaMap to construct mappings from more than 35–40 candidates can be extremely time consuming, even with the aforementioned efficiency improvements; conversely, restricting the candidate set to fewer than 30 is likely to exclude some useful candidates. MetaMap by default will therefore automatically prune the candidate list to 35 candidates; however, this default behavior can be overridden in two ways:

1. To set a pruning threshold different from the default 35, use the option `--prune N` with any positive integer $N$.
2. To disable candidate pruning altogether, use the option `--no_prune`; be aware, however, that using this option can cause MetaMap to run for many hours and/or run out of memory.

Finally, note that if candidate pruning is used, all original candidates will still appear in the Candidates portion of MetaMap's output.

# 3 Additional Data Models

MetaMap has historically made available two data models (Strict and Relaxed) with each UMLS Metathesaurus release; the Filtering Report on the SKR website

http://skr.nlm.nih.gov/papers/references/filtering10.pdf.

explains how we construct these models via extensive filtering of the raw UMLS data.[1]

In order to accommodate the UMLS source-vocabulary licensing permissions and processing requirements of as many users as possible, MetaMap releases beginning in 2011 will include three distinct versions of the data that are based mostly on the Restriction Categories of Metathesaurus source vocabularies. (See the *Restriction Categories* tab on the UMLS Source Release Documentation page for all UMLS sources' Restriction Categories.) Each data version includes a Strict and Relaxed model; listed from smallest to largest, the three versions are:

1. Base: The Base data version includes those source vocabularies with no associated licensing restrictions beyond those of the UMLS license; in the 2011AA release, this version includes all and only sources of Restriction Category 0.

2. USAbase: The USAbase data version includes those source vocabularies with no associated restrictions beyond a UMLS license, **and free for use for US-based projects**; in the 2011AA release, this version includes the Base vocabularies (those with Restriction Category 0), plus the five Category-4 sources and the four Category-9 sources (including, most notably, SNOMEDCT). The USAbase version is a proper superset of the Base version, and might be the most appropriate version for users with a SNOMEDCT license. To repeat: **This data version is MetaMap2011's default, but the default can be overridden using the `-V` flag.**

3. NLM: The NLM data version includes the full Metathesaurus other than the CPT, CPTSP, HCPT, and MTHCH vocabularies from the CPT family, and the HCDT, HCPCS, and MTHHH vocabularies from the HCPCS family.

The table below shows the number of distinct CUIs and CUI-LUI-SUI combinations in each of the three data versions' strict and relaxed models:

| | **Strict** | | **Relaxed** | |
|---|---|---|---|---|
| | **CUI** | **C-L-S** | **CUI** | **C-L-S** |
| **Base** | 1,171,126 | 1,990,193 | 1,704,719 | 3,309,606 |
| **USAbase** | 1,332,076 | 2,301,691 | 2,006,271 | 4,266,921 |
| **NLM** | 1,557,385 | 2,683,726 | 2,394,524 | 5,448,114 |

For comparison testing of the three data versions, we ran MTI on over 72,000 MEDLINE citations, and achieved best overall results with the USAbase data version. Our experiments showed that including vocabularies of Restriction Categories 1–3 is not necessary to achieve optimal results; however, users should decide which of these data versions best suits their specific analytical and processing requirements and is consistent with their UMLS licensing privileges. We encourage users to consult the UMLS Metathesaurus License, available at https://uts.nlm.nih.gov/license.html.

[1] The moderate model mentioned in the report is no longer created.

# 4 Single-Character Alphabetic Tokens

MetaMap2011 does not attempt to identify candidates from single-character alphabetic tokens appearing in a phrase

1. containing at least 10 tokens, and
2. more than three-quarter of whose tokens are single-character tokens.

This strategy is designed to reduce false-positive candidates that would otherwise be generated from text such as

```
The sequence was (in the standard one-letter code)
A-N-S-F-L-X-X-L-R-P-G-N-V-X-R-X-C-S-X-X-V-C-X-F-X-X-A-R-X-I-F-Q-N-T-X-D-T-
M-A-F-W-S-K-Y-S-D-G-D-Q-C-E-D-R-P-S-G-S-P-C-D-L-P-C-C-G-R-G-K-C-I-H-G-L-G-
G-F-R-C-D-C-A-E-G-W-E-G-R-F-C-L-H-E-V-R-F-S-N-C-S-A-E-B-G-G-C-A-H-Y-C-M-E-
E-E-G-R-R-H-C-S-C-A-P-G-Y-R-L-E-D-D-H-Q-L-C-V-S-K-V-T-F-P-C-G-R-L-G-K-R-M-
```

from PMID 282610 and

```
Upon sequencing the peptides by the automated Edman method, the following
sequence was obtained: A D T N A P L C L C D E P G I L G R N Q L V T P E V
K E K I E K A V E A V A E E S G V S G R G F S L F S H H P V F R E C G K Y E
C R T V R P E H T R C Y N F P P F V H F T S E C P V S T R D C E P V F G Y T
V A G E F R V I V Q A P R A G F R Q C V W Q H K C R Y G S N N C G F S G R C
T Q Q R S V V R L V T Y N L E K D G F L C E S F R T C C G C P C R N Y
Carcinoscorpius coagulogen consists of a single polypeptide chain with a
total of 175 amino acid residues and a calculated molecular weight of 19,675.
```

from PMID 3905780.

# 5 Improved Treatment of Apostrophe-s

Previous versions of MetaMap have analyzed "`'s`" (apostrophe + s) as a contraction for the verb *is*; MetaMap2011 by default treats "`'s`" as a possessive, except when following *he she* or *it*. The previous contraction treatment of "`'s`" can be restored by specifying the `--apostrophe_s_contraction` command-line option.

# 6 New XML Command-Line Options

In previous versions of MetaMap, the XML options have been

- `--XML format`,

- `--XML format1`,
- `--XML noformat`, and
- `--XML noformat1`

Moreover, the short version "`-%`" could be used instead of "`--XML`". These options were first explained here.

We have changed these four options to

- `--XMLf`,
- `--XMLf1`,
- `--XMLn`, and
- `--XMLn1`

There is no longer a short form such as `-%` for any of the new XML options.

## 7 Composite Phrases

Previous versions of MetaMap have included an option called `--quick_composite_phrases`, which caused MetaMap to construct longer, composite phrases from the simple phrases produced by the parser. A composite phrase consists of

- a noun followed by
- any prepositional phrase, optionally followed by
- one or more prepositional phrases introduced by *of*.

Our canonical example is *pain on the left side of the chest*, which will map to `Left sided chest pain` with the composite phrases option on, but to separate concepts without.

This option is now named simply `--composite_phrases`, and can be invoked with `-Q`; it also automatically turns on `--term_processing` and `--ignore_word_order`, but only during the analysis of any constructed composite phrase (i.e., a phrase created by glomming at least one prepositional phrase onto a preceding noun). The maximum number of prepositional phrases that will be added to a noun is under user control, and can be specified by providing a mandatory integer argument to the `-Q` flag; our experiments suggest that 3 or 4 might be reasonable values to try. Examples of text that would by default be divided into multiple phrases, but analyzed as a single phrase by specifying the `-Q 4` option are

```
Limits for the Number of Solutions of Certain General Types of Equations

findings from the examination of the oral cavity of pupils of the Amaleion Orphanage

Points in the Technique of the Treatment of Fracture of the Patella

Institute of Anatomy of the Faculty of Medicine of Oporto
```

Composite phrases are necessarily longer than non-composite phrases, and the `--term_processing` and `--ignore_word_order` options will invoke more computation; consequently, an observable slowdown will result from using `--composite_phrases`. If processing with this option turned on proves to be too slow, pruning the candidate set (as described in Section 2 above) might be a worthwhile option to explore.

Much to our surprise, our internal testing of composite phrases with MTI resulted in virtually identical recall, but slightly worse precision. However, this option might prove useful for users analyzing clinical text who wish to increase recall.

## 8 Numbered Mappings

MetaMap has long provided the ability to number the candidates in its default human-readable output by using the `--number_the_candidates` (short form `-n`) command-line option. When called with `-n`, the Candidates portion of MetaMap's human-readable output is displayed as

```
Phrase: "stent"
Meta Candidates (2):
   1.   1000 Stent (Stent, device) [Medical Device]
   2.   1000 Stent (Stent Device Component) [Medical Device]
```

instead of

```
Phrase: "stent"
Meta Candidates (2):
  1000 Stent (Stent, device) [Medical Device]
  1000 Stent (Stent Device Component) [Medical Device]
```

MetaMap2011 includes the ability to number the mappings using the `--number_the_mappings` (short form `-f`) command-line option. When called with `-f`, the Mappings portion of MetaMap's human-readable output is displayed as

```
1. Meta Mapping (1000):
  1000 Stent (Stent Device Component) [Medical Device]
2. Meta Mapping (1000):
  1000 Stent (Stent, device) [Medical Device]
```

instead of

```
Meta Mapping (1000):
  1000 Stent (Stent Device Component) [Medical Device]
Meta Mapping (1000):
  1000 Stent (Stent, device) [Medical Device]
```

Invoking the `--number_the_mappings` option will not modify MetaMap Machine (MMO) or XML output.

# 9 User-Defined Acronyms and Abbreviations

The biomedical literature is replete with acronyms and abbreviations (AAs)[2] defined by the author, for example

```
Trimethyl cetyl ammonium pentachlorphenate (TCAP)
Reticulo-endothelial immune serum (REIS)
isonicotinic acid hydrazid (INAH)
```

MetaMap has long handled such text by interpreting appearances of the AA (`TCAP`, `REIS`, `INAH`) later in the text as if the expansion had been used instead.

In MetaMap2011, we introduce user-defined AAs (UDAs),[3] which enable MetaMap users to better handle AAs and other idiosyncratic expressions that either are not in the UMLS or exhibit unwanted spurious ambiguity. For example, defining `Positron Emitting Tomography` to be an AA for `PET` and `Computerized Axial Tomography` for `CAT` could be useful in analyzing radiology reports, because doing so would suppress the identification of UMLS concepts referring to certain companion animals.

To take advantage of this functionality, simply create a plain text file in which UDAs and their expansions are separated by a vertical bar:

```
PET|Positron Emitting Tomography
CAT|Computerized Axial Tomography
DRSP|drug-resistant streptococcus pneumoniae
NIDR|national infectious disease register
```

After creating such a file, e.g., `UDAFile`, simply call MetaMap with the `--UDA` option as follows:

```
metamap --UDA UDAFile
```

and the AA expansions will be analyzed by MetaMap. The AA and the expansion can appear in the line in either order, e.g.,

```
Positron Emission Tomography|PET
CAT|Computerized Axial Tomography
DRSP|drug-resistant streptococcus pneumoniae
national infectious disease register|NIDR
```

Indeed consistency of order within the file is not necessary: MetaMap will consider the shorter string to be the AA, and the longer one the expansion, regardless of which appears first in a given line. Some formatting trivia:

- UDAs should consist of one alphanumeric token only.

[2] Of which this is an example.

[3] `UDA` is a UDA!

- Lines whose first non-blank character is "`#`" are treated as comments and are ignored.
- Multiple consecutive whitespace characters are treated as a single one.
- Whitespace is ignored
  - before the first nonblank character in a line,
  - after the last nonblank character in a line, and
  - immediately before and after the "`|`" character.
- MetaMap will stop reading the UDA file when it encounters a line containing only whitespace.

Several points to note:

- In order to respect the primacy of the text and not subvert the intentions of the original author, author-defined AAs take precedence over any defined by the user. More specifically, if both the author and the user provide expansions for the same AA, MetaMap will use the author's and not the user's; moreover, if the user provides an AA expansion, and the AA itself is part of an author-defined AA expansion, the user's expansion will be ignored. Such conflicts between author- and user-defined AAs should be uncommon, because UDAs will probably be most applicable in analyzing clinical text, which does not generally contain author-defined AAs, but does typically include idiosyncratic domain-specific AAs that are defined in neither the text nor the UMLS.
- UDA expansions should map to some Metathesaurus concept(s) in order to be useful.
- Metathesaurus concepts mapped to by UDA expansions override (not supplement) concepts that MetaMap would have identified absent any UDAs. To illustrate potential dangers of ignoring this point, we present a number of multiword Metathesaurus strings along with completely invented acronyms that are themselves Metathesaurus strings:

  ```
  ACHE|acute constitutional hand eczema
  CYST|childhood yolk sac tumor
  IRIS|inner retinal ischemic spots
  LUCITE|left upper central incisor tooth enamel
  SCAR|sodium clodronate adverse reaction
  TEAR|topical estradiol adverse reaction
  UMLS|undecylenate medicated liquid soap                (!!)
  UREA|uridylyl removing enzyme activity
  WART|WHO Adverse reaction terminology
  ```

  Defining any of the above UDAs will block the identification of any Metathesaurus concepts derived from the AAs themselves, which is probably not the desired behavior.
- UDAs are case sensitive: If `WART` is defined as above, and the input text contains `wart`, the UDA will not be expanded.

- A UDA will be expanded only if it appears as a discrete token in the input, and not as a substring of a longer word. For example, if `WART` is defined as above, it will not be expanded from, say, `WARTS` or `WARTHOG`.

| Base/strict | | USAbase/strict | | NLM/strict | |
|---|---|---|---|---|---|
| NCBI | 539,685 | NCBI | 539,333 | NCBI | 539,312 |
| MSH | 482,182 | MSH | 480,673 | MSH | 476,577 |
| LNC | 133,652 | SNOMEDCT | 224,714 | MEDCIN | 222,445 |
| FMA | 127,251 | LNC | 133,178 | LNC | 132,915 |
| HUGO | 107,179 | FMA | 127,177 | RCD | 127,983 |
| NCI | 101,407 | HUGO | 107,178 | FMA | 125,953 |
| OMIM | 83,077 | NCI | 97,012 | SNOMEDCT | 116,691 |
| CHV | 73,648 | OMIM | 82,493 | HUGO | 107,178 |
| GO | 67,353 | SNMI | 78,375 | NCI | 94,460 |
| RXNORM | 64,981 | RXNORM | 67,616 | OMIM | 81,972 |
| MTH | 63,998 | GO | 67,328 | SNMI | 71,983 |
| NDFRT | 21,352 | MTH | 65,203 | RXNORM | 70,364 |
| MTHFDA | 17,932 | CHV | 55,919 | GO | 67,323 |
| MTHSPL | 11,396 | ICD10CM | 25,842 | MTH | 62,607 |
| AOD | 11,017 | NDFRT | 21,095 | CHV | 50,100 |
| ICD9CM | 10,590 | MTHFDA | 17,922 | MMSL | 38,171 |
| ICD10PCS | 10,527 | MTHSPL | 11,340 | MDR | 30,209 |
| MTHICD9 | 10,254 | SNM | 10,766 | NDFRT | 21,217 |
| CSP | 7,327 | ICD10PCS | 10,512 | ICD10CM | 20,443 |
| UWDA | 6,785 | AOD | 10,251 | UMD | 18,237 |
| PDQ | 5,933 | MTHICD9 | 8,386 | MTHFDA | 17,729 |
| DXP | 5,832 | ICD9CM | 8,095 | ICD10AM | 11,727 |
| HL7V3.0 | 4,943 | CSP | 6,946 | GS | 11527 |
| LCH | 3,448 | UWDA | 6,624 | MTHSPL | 11,334 |
| SPN | 3,168 | PDQ | 5,869 | NDDF | 10,749 |
| CST | 2,720 | DXP | 5,018 | ICD10PCS | 10,506 |
| HL7V2.5 | 2,673 | HL7V3.0 | 4,917 | AOD | 10,085 |
| COSTAR | 1,764 | LCH | 3,394 | SNM | 9,971 |
| MTHMST | 1,414 | SPN | 3,057 | MTHICD9 | 7,656 |
| VANDF | 1,289 | HL7V2.5 | 2,648 | MDDB | 7,586 |
| MEDLINEPLUS | 1,194 | CST | 2,326 | RCDAE | 7,561 |
| HCPCS | 1,078 | COSTAR | 1,437 | CCPSS | 7,372 |
| CCS | 526 | MTHMST | 1,418 | ICD9CM | 7,317 |
| MTHCH | 519 | VANDF | 1,264 | CSP | 6,897 |
| AOT | 427 | MEDLINEPLUS | 1,150 | UWDA | 6,622 |
| ICPC | 319 | HCPCS | 1,075 | PDQ | 5,736 |
| QMR | 303 | ICF-CY | 760 | HL7V3.0 | 4,904 |
| AIR | 279 | CCS | 532 | DXP | 4,621 |
| SRC | 229 | MTHCH | 511 | ICNP | 4,454 |
| RAM | 178 | LNC_MDS20 | 473 | MMX | 3,812 |
| USPMG | 173 | AOT | 298 | ICPC2P | 3,506 |
| MTHHH | 170 | ICPC | 295 | LCH | 3,058 |
| MCM | 20 | AIR | 264 | SPN | 3,022 |
| MTHHL7V2.5 | 1 | SRC | 264 | HL7V2.5 | 2,634 |

| Base/relaxed | | USAbase/relaxed | | NLM/relaxed | |
|---|---|---|---|---|---|
| NCBI | 654,158 | SNOMEDCT | 771,970 | MEDCIN | 706,782 |
| MSH | 624,855 | NCBI | 653,765 | NCBI | 653,744 |
| RXNORM | 417,134 | MSH | 623,200 | MSH | 619,037 |
| LNC | 312,547 | RXNORM | 416,132 | SNOMEDCT | 601,411 |
| ICD10PCS | 249,861 | LNC | 312,080 | RXNORM | 398,862 |
| FMA | 137,568 | ICD10PCS | 249,845 | LNC | 311,809 |
| NCI | 134,086 | FMA | 137,498 | RCD | 293,367 |
| HUGO | 120,949 | NCI | 129,508 | ICD10PCS | 249,839 |
| OMIM | 108,599 | HUGO | 120,948 | FMA | 136,208 |
| CHV | 91,468 | OMIM | 108,030 | NCI | 126,716 |
| GO | 89,206 | SNMI | 104,739 | HUGO | 120,948 |
| MTH | 81,098 | ICD10CM | 98,774 | OMIM | 107,486 |
| NDFRT | 72,547 | GO | 89,180 | SNMI | 97,729 |
| ICD9CM | 36,170 | MTH | 80,492 | ICD10CM | 90,629 |
| MTHFDA | 31,769 | CHV | 73,187 | GO | 89,174 |
| MTHSPL | 26,640 | NDFRT | 72,270 | ICPC2ICD10ENG | 78,499 |
| MTHICD9 | 17,444 | ICD9CM | 33,165 | MTH | 73,035 |
| AOD | 13,734 | MTHFDA | 31,759 | NDFRT | 72,193 |
| PDQ | 12,340 | MTHSPL | 26,584 | CHV | 66,919 |
| HCPCS | 11,083 | MTHICD9 | 15,258 | NDDF | 60,855 |
| CSP | 8,408 | SNM | 13,986 | MMSL | 58,166 |
| VANDF | 8,383 | AOD | 12,961 | MDR | 48,009 |
| UWDA | 7,928 | PDQ | 12,273 | ICD9CM | 31,996 |
| HL7V,3.0 | 7,033 | HCPCS | 11,080 | MTHFDA | 31556 |
| DXP | 6,387 | VANDF | 8,365 | MTHSPL | 26,563 |
| SPN | 4,730 | CSP | 8,025 | ICD10AM | 23,983 |
| HL7V2.5 | 4,595 | UWDA | 7,764 | UMD | 22,185 |
| CST | 4,562 | HL7V3.0 | 7,001 | GS | 18,131 |
| LCH | 3,652 | DXP | 5,568 | MTHICD9 | 14,437 |
| COSTAR | 1,852 | SPN | 4,630 | RCDAE | 13,850 |
| MTHMST | 1,801 | HL7V,2.5 | 4,570 | SNM | 13175 |
| MEDLINEPLUS | 1,348 | CST | 4,242 | AOD | 12,790 |
| MTHCH | 1,015 | LCH | 3,599 | MDDB | 12,347 |
| CCS | 971 | LNC_MDS20 | 2,540 | PDQ | 12,149 |
| ICPC | 933 | MTHMST | 1,806 | CCPSS | 11,387 |
| AIR | 600 | ICF-CY | 1,642 | ICPC2P | 11,251 |
| QMR | 588 | COSTAR | 1,525 | NIC | 10,813 |
| AOT | 452 | MEDLINEPLUS | 1,302 | MMX | 9,008 |
| MTHHH | 318 | MTHCH | 1,008 | ALT | 8,845 |
| USPMG | 282 | CCS | 979 | VANDF | 8,361 |
| SRC | 259 | ICPC | 918 | CSP | 7,973 |
| RAM | 228 | AIR | 588 | UWDA | 7,762 |
| MCM | 23 | QMR | 534 | HL7V3.0 | 6,987 |
| MTHHL7V2.5 | 2 | AOT | 318 | RCDSY | 6,590 |